TRACCIA ESERCIZIO S7-L5

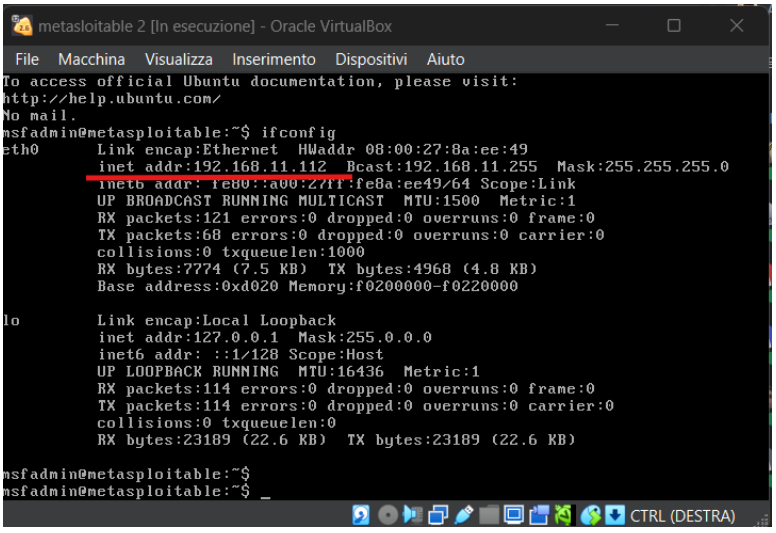
La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 -Java RMI. Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

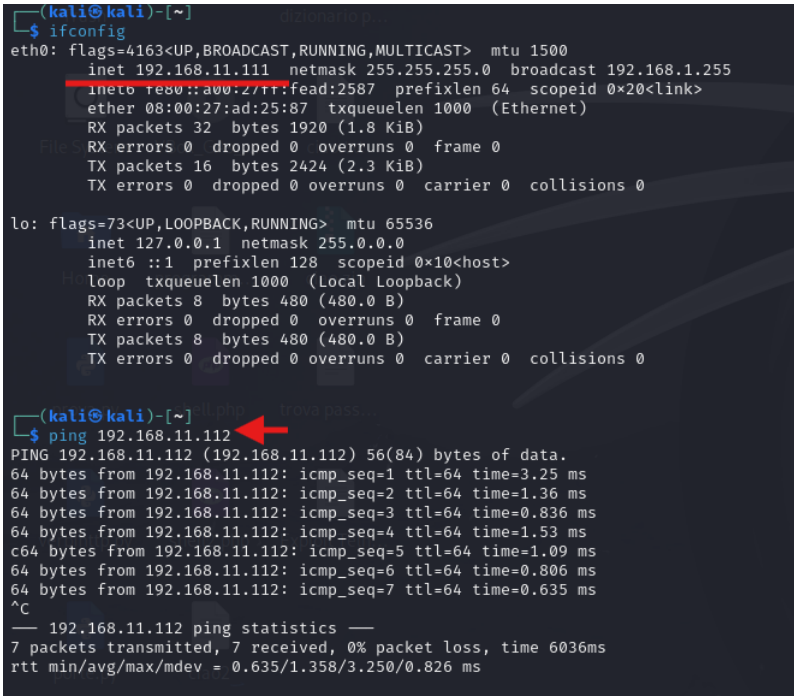
I requisiti dell'esercizio sono:

- 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
- 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.

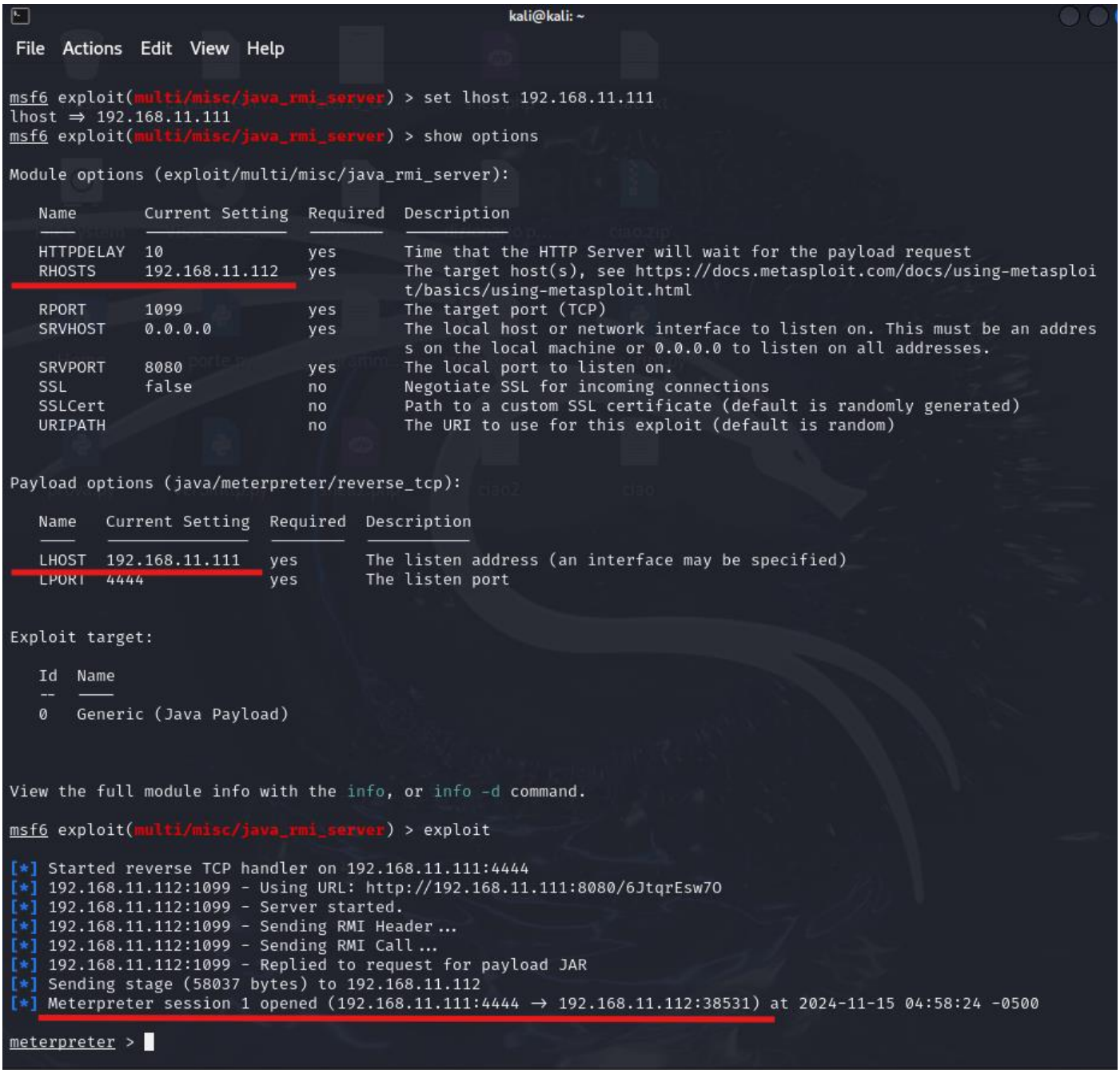
SVOLGIMENTO ESERCIZIO

SETTIAMO LE MACCHINE CON I NUOVI IP ED ACCERTIAMOCI CHE COMUNICHINO





APRIAMO LA SESSIONE TRA LE MACCHINE



SESSIONE

Aprire una sessione tra Kali e Metasploitable significa stabilire una connessione di comunicazione remota tra la tua macchina Kali Linux (che è l'attaccante) e la macchina Metasploitable (la vittima), in modo che tu possa interagire con essa e sfruttare vulnerabilità.

Nel contesto di un esercizio di penetration testing, una sessione è una connessione attiva che ti permette di eseguire comandi e raccogliere informazioni sulla macchina vittima.

Quando si parla di "aprire una sessione", generalmente si intende:

- Compromettere la macchina vittima sfruttando una vulnerabilità.
- Stabilire una connessione attiva che ti consente di interagire con la macchina compromessa e raccogliere informazioni o eseguire altre azioni.

COME SI APRE UNA SESSIONE TRA KALI E METASLOITABLE

Eseguire un exploit :

Ad esempio, una vulnerabilità Java RMI (come nel nostro esercizio) per compromettere il sistema Metasploitable.

Creare una sessione Meterpreter:

Una volta che l'exploit ha avuto successo, Metasploit stabilisce una sessione di Meterpreter sulla macchina vittima. Meterpreter è uno shell avanzato che ti permette di eseguire comandi sulla macchina vittima, trasferire file, fare screenshot, raccogliere informazioni di sistema, etc.

Interagire con la sessione:

Dopo che la sessione è stata creata, puoi interagire con essa usando comandi come sessions in Metasploit per visualizzare e gestire le sessioni attive.

EXPLOIT

Un exploit è un programma o un codice che sfrutta una vulnerabilità presente in un sistema o applicazione per ottenere un comportamento non previsto, spesso con l'obiettivo di ottenere l'accesso non autorizzato a una macchina o di eseguire altre azioni dannose.

In altre parole, un exploit “approfitta” di un difetto o di una debolezza nel software, nel sistema operativo o in un servizio per eseguire azioni che normalmente non sarebbero consentite.

Tipi di Exploit:

La **vulnerabilità** è una debolezza o un difetto in un sistema, applicazione o rete che può essere sfruttata da un attaccante per compromettere la sicurezza di un sistema. Un **exploit**, come già spiegato, è un mezzo attraverso il quale un attaccante sfrutta una vulnerabilità per compiere azioni dannose, come ottenere l'accesso non autorizzato, eseguire codice arbitrario, o causare malfunzionamenti.

Gli exploit possono variare a seconda della vulnerabilità che sfruttano, ecco alcuni esempi:

1. Buffer Overflow Exploit:
   Approfitta di un errore nella gestione della memoria (quando i dati vengono scritti oltre i limiti di un buffer) per sovrascrivere dati critici, come l'indirizzo di ritorno di una funzione, e prendere il controllo del flusso del programma.

2. Vulnerabilità di Servizi di Rete:
   Un exploit può sfruttare una vulnerabilità in un servizio di rete (come HTTP, FTP, SSH) per ottenere accesso a un sistema. Ad esempio, un buffer overflow in un servizio web potrebbe consentire a un attaccante di eseguire codice arbitrario.

3. Zero-Day Exploit:
Si riferisce a un exploit che sfrutta una vulnerabilità che **non è ancora conosciuta** dal pubblico o dal produttore del software. Questi exploit sono particolarmente pericolosi, poiché non esistono ancora patch di sicurezza disponibili.

4. Web Application Exploit:
Gli exploit delle applicazioni web si concentrano su vulnerabilità come Cross-Site Scripting (XSS), SQL Injection, e Remote File Inclusion (RFI), che possono essere sfruttate per compromettere applicazioni web e ottenere l'accesso ai sistemi sottostanti.

5. Privilege Escalation Exploit:
Si tratta di un exploit che permette a un utente con privilegi limitati di ottenere **privilegi più alti** (ad esempio, amministratore o root) su un sistema. Questo tipo di exploit è molto comune una volta che un attaccante ha ottenuto l'accesso a una macchina con privilegi minimi.

Come Funziona un Exploit?

1. Individuazione della vulnerabilità: L'exploit è basato su una vulnerabilità conosciuta in un software o in un servizio. La vulnerabilità può essere un errore di programmazione, una configurazione errata o una debolezza del sistema.
2. Preparazione del codice di attacco: L'exploit è scritto per sfruttare questa vulnerabilità in un modo specifico. Questo può includere l'invio di pacchetti malformati a una porta di rete, l'inserimento di codice malevolo in un'applicazione, o la manipolazione del comportamento di un servizio.
3. Esecuzione dell'attacco: Quando l'exploit viene eseguito, la vulnerabilità viene attaccata, consentendo all'attaccante di eseguire operazioni non autorizzate (ad esempio, ottenere l'accesso, eseguire comandi, rubare dati, ecc.).

Esempi di Exploit:

- Shellshock (2014): Una vulnerabilità nei comandi di shell Bash che consentiva agli attaccanti di eseguire comandi arbitrari sui sistemi vulnerabili.
- Heartbleed (2014): Una vulnerabilità nel software di crittografia OpenSSL che permetteva di leggere la memoria di un server remoto, rivelando informazioni sensibili come password e chiavi private.

Exploit in Metasploit:

Nel contesto di Metasploit, un exploit è un modulo predefinito che sfrutta una vulnerabilità conosciuta. Metasploit fornisce una vasta gamma di exploit che possono essere usati per attaccare sistemi vulnerabili. Ad esempio:

- Si può scegliere un exploit per una vulnerabilità di Java RMI.
- Configurare il modulo con l'indirizzo IP della macchina vittima e dell'attaccante.
- Eseguire l'exploit per ottenere una sessione di Meterpreter sulla macchina compromessa.

PAYLOAD

Un payload è una componente di un attacco informatico che esegue un'azione specifica una volta che un exploit ha avuto successo e ha compromesso un sistema. In altre parole, il payload è il "carico utile" che viene eseguito su un sistema vulnerabile dopo che è stata sfruttata una vulnerabilità (un exploit).

Cos'è il Payload in Sicurezza Informatica?

In termini più semplici, il *ayload è il codice che viene iniettato ed eseguito sulla macchina vittima una volta che l'exploit ha avuto successo. Il suo obiettivo può variare: dall'esecuzione di un comando, al furto di dati, all'installazione di un programma dannoso (come un backdoor), o addirittura al controllo completo del sistema compromesso.

Tipi di Payload:

Esistono diversi tipi di payload a seconda dell'azione che si vuole eseguire sulla macchina compromessa. I payload possono essere suddivisi in due categorie principali:

Esempi di Payload Comuni:

1. Meterpreter:

- Descrizione: Meterpreter è uno dei payload più utilizzati in Metasploit. È una shell interattiva avanzata che consente all'attaccante di eseguire una vasta gamma di operazioni sulla macchina vittima. Una volta che Meterpreter è in esecuzione sulla vittima, l'attaccante può eseguire comandi, trasferire file, fare screenshot, registrare tastiera, raccogliere informazioni di rete, e molto altro.

- Esempio di comando: Una volta che Meterpreter è attivo, l'attaccante può eseguire il comando `sysinfo` per ottenere informazioni sul sistema vittima.

2. Reverse Shell:

- Descrizione: Il payload reverse shell crea una connessione dalla macchina vittima a quella dell'attaccante, permettendo all'attaccante di eseguire comandi sulla macchina compromessa. In genere, il payload reverse shell apre una porta sulla macchina vittima e "invita" l'attaccante a connettersi ad essa.

- Esempio: Una reverse shell può essere usata per avere un controllo remoto sulla macchina, eseguire comandi, o trasferire file.

3. Bind Shell:

- Descrizione: A differenza del reverse shell, il bind shell fa in modo che la macchina vittima apra una porta sulla sua macchina e attenda una connessione in ingresso dall'attaccante. Una volta connesso, l'attaccante può inviare comandi al sistema.

- Esempio: Se l'attaccante non può connettersi direttamente alla macchina vittima (ad esempio, per motivi di rete), un bind shell può consentire alla vittima di "avviare" la connessione verso l'attaccante.

4. Command Execution:

- Descrizione: Questo payload esegue comandi specifici sulla macchina vittima. Può essere usato per eseguire comandi di sistema o per raccogliere informazioni come l'indirizzo IP o il sistema operativo.

- Esempio: Un attaccante potrebbe utilizzare un payload per eseguire un comando come `ipconfig` sulla macchina vittima per raccogliere informazioni di rete.

5. Keylogger:

- Descrizione: Un keylogger è un tipo di payload che registra tutto ciò che l'utente digita sulla macchina vittima, permettendo all'attaccante di ottenere informazioni sensibili come password, numeri di carte di credito e altro.

- Esempio: Un keylogger potrebbe essere usato per raccogliere credenziali di login in un'applicazione web.

6. Download and Execute:

- Descrizione: Questo payload viene usato per scaricare e eseguire un file da una fonte esterna sulla macchina vittima. Può essere utilizzato per scaricare software dannoso, come malware, rootkit, o trojan.

- Esempio: Un payload di questo tipo potrebbe scaricare e avviare un malware che permette all'attaccante di mantenere un controllo permanente sulla macchina vittima.

Come Funziona un Payload?

1. Sfruttamento della Vulnerabilità: Un exploit sfrutta una vulnerabilità in un sistema (ad esempio, un buffer overflow o un'attacco di SQL injection).

2. Attivazione del Payload: Dopo che l'exploit ha avuto successo, il payload viene iniettato nel sistema vittima. A seconda del tipo di payload, questo esegue un'azione come aprire una shell, rubare informazioni, o installare un backdoor.

3. Completamento dell'Azione: Il payload esegue il comando o le azioni per cui è stato progettato. Può inviare i risultati a un server di comando e controllo (C&C), eseguire comandi, o aprire una connessione per consentire l'accesso remoto.

SHELL

Una shell è un programma che consente all'utente di interagire con il sistema operativo tramite una linea di comando. In pratica, è un'interfaccia che permette di inviare comandi al sistema e ottenere risposte. La shell può essere un'interfaccia a riga di comando (CLI) o anche un ambiente di scripting che esegue operazioni automatizzate.

Tipi di Shell:

1. Shell a Riga di Comando (CLI):

- Descrizione: È un tipo di shell che riceve input sotto forma di testo (comandi) e restituisce i risultati sempre come testo. È la forma tradizionale di interazione con il sistema operativo.

- Funzionalità: Esecuzione di comandi di sistema, navigazione nel file system, gestione di file e processi.

2. Shell Grafica:

- Descrizione: Anche se meno comune nel contesto della sicurezza informatica, alcune shell permettono un'interazione grafica tramite finestre e icone, ma restano legate all'idea di "interfaccia utente" del sistema operativo.

- Esempio: Ambienti desktop come GNOME, KDE su Linux, o l'interfaccia grafica di Windows.

Cos'è una Shell Remota?

In contesti di sicurezza informatica, quando parliamo di una shell remota, ci riferiamo alla possibilità di ottenere l'accesso alla shell di un sistema remoto tramite una connessione di rete. Un attaccante potrebbe riuscire a ottenere una shell remota su una macchina vulnerabile sfruttando un exploit o una backdoor, e utilizzare la shell per eseguire comandi sul sistema target.

Esistono principalmente due tipi di shell remota:

1. REVERSE SHELL:

- Descrizione: Una reverse shell è un tipo di shell remota in cui la macchina vittima si connette all'attaccante. In altre parole, la vittima apre una connessione verso l'attaccante, che può poi inviare comandi alla macchina compromessa.

- Come funziona: Dopo che un exploit ha avuto successo, la macchina vittima si collega a una porta specificata sull'indirizzo IP dell'attaccante. L'attaccante può quindi inviare comandi alla vittima attraverso questa connessione.

- Esempio: Un attaccante potrebbe usare una reverse shell per eseguire comandi sulla macchina compromessa, come ottenere una lista di file o installare malware.

2. BLIND SHELL:

- Descrizione: Una bind shell è l'opposto di una reverse shell. In questo caso, la macchina vittima apre una porta e attende una connessione dall'attaccante.

- Come funziona: La vittima apre una porta specifica e ascolta per una connessione in ingresso. L'attaccante si connette quindi a quella porta per interagire con la shell della vittima.

- Esempio: Un attaccante potrebbe utilizzare un bind shell per connettersi a una macchina compromessa e ottenere una shell che permette l'esecuzione di comandi.

Cosa Fa una Shell?

Una shell consente di:

- Eseguire comandi: L'utente o l'attaccante può inviare comandi per interagire con il sistema operativo.

- Navigare nel file system: La shell permette di spostarsi tra le directory, visualizzare file, e modificarli.

- Gestire processi: È possibile avviare, terminare o monitorare i processi in esecuzione sul sistema.

- Automatizzare attività: Tramite script, è possibile automatizzare una serie di operazioni complesse o ripetitive.

In Sicurezza Informatica:

- Quando un attaccante ottiene l'accesso a una macchina remota tramite una shell, può utilizzarla per raccogliere informazioni, eseguire comandi, manipolare i file, o addirittura fare eseguire comandi per elevare i privilegi dell'utente (privilege escalation).

- Un'attività comune che avviene tramite una shell è l'uso di Meterprete in Metasploit. Una volta che un attaccante ha una shell Meterpreter, può eseguire molteplici operazioni sulla macchina compromessa, come raccogliere informazioni di rete, fare screenshot, o trasferire file.

EXPLOIT / PAYLOAD /SHELL

Un **exploi**t è un codice o una tecnica usata per sfruttare una vulnerabilità in un sistema e ottenere accesso non autorizzato o eseguire azioni specifiche. Una volta che l'exploit ha successo, viene eseguito un **payload**, ovvero il "carico utile" che può compiere varie operazioni, come aprire una connessione remota o eseguire comandi. Una **shell**, infine, è l'interfaccia che consente di interagire con il sistema operativo, spesso usata dal payload per dare all'attaccante il controllo del sistema compromesso.

Insieme, exploit, payload e shell formano un attacco: l'exploit apre la porta, il payload la sfrutta, e la shell fornisce il controllo.

HTTPDELETE

Il parametro HTTPDELAY in Metasploit è utilizzato per configurare un ritardo in secondi tra le richieste HTTP inviate dal payload e il server attaccante. Questo parametro è particolarmente utile quando si utilizzano payload che si basano su comunicazioni HTTP o HTTPS, come quelli di tipo reverse_http o reverse_https.

Perché impostare HTTPDELAY?

1. Rallentare la frequenza delle richieste:

- Quando un payload, come un reverse_http, stabilisce una connessione al server dell'attaccante, potrebbe inviare richieste periodiche (chiamate "beacon") per mantenere la connessione attiva.

- Configurare un ritardo tra queste richieste riduce la frequenza del traffico, utile in situazioni in cui si vuole evitare di sovraccaricare la rete o di attirare l'attenzione dei sistemi di monitoraggio.

2. Risoluzione di problemi di instabilità:

- In alcune reti lente o instabili, una comunicazione HTTP senza ritardi potrebbe fallire a causa di timeout o congestione. Incrementare HTTPDELAY può garantire che le richieste abbiano il tempo di completarsi.

3. Evitare rilevamenti da sistemi di sicurezza:

- Riducendo la frequenza delle richieste HTTP, si minimizza il rischio che i sistemi di rilevamento (IDS/IPS) notino un traffico anomalo.

Come configurarlo:

Durante la configurazione di un payload in Metasploit, puoi impostare il parametro HTTPDELAY così:

```
set HTTPDELAY 20
```

In questo esempio:

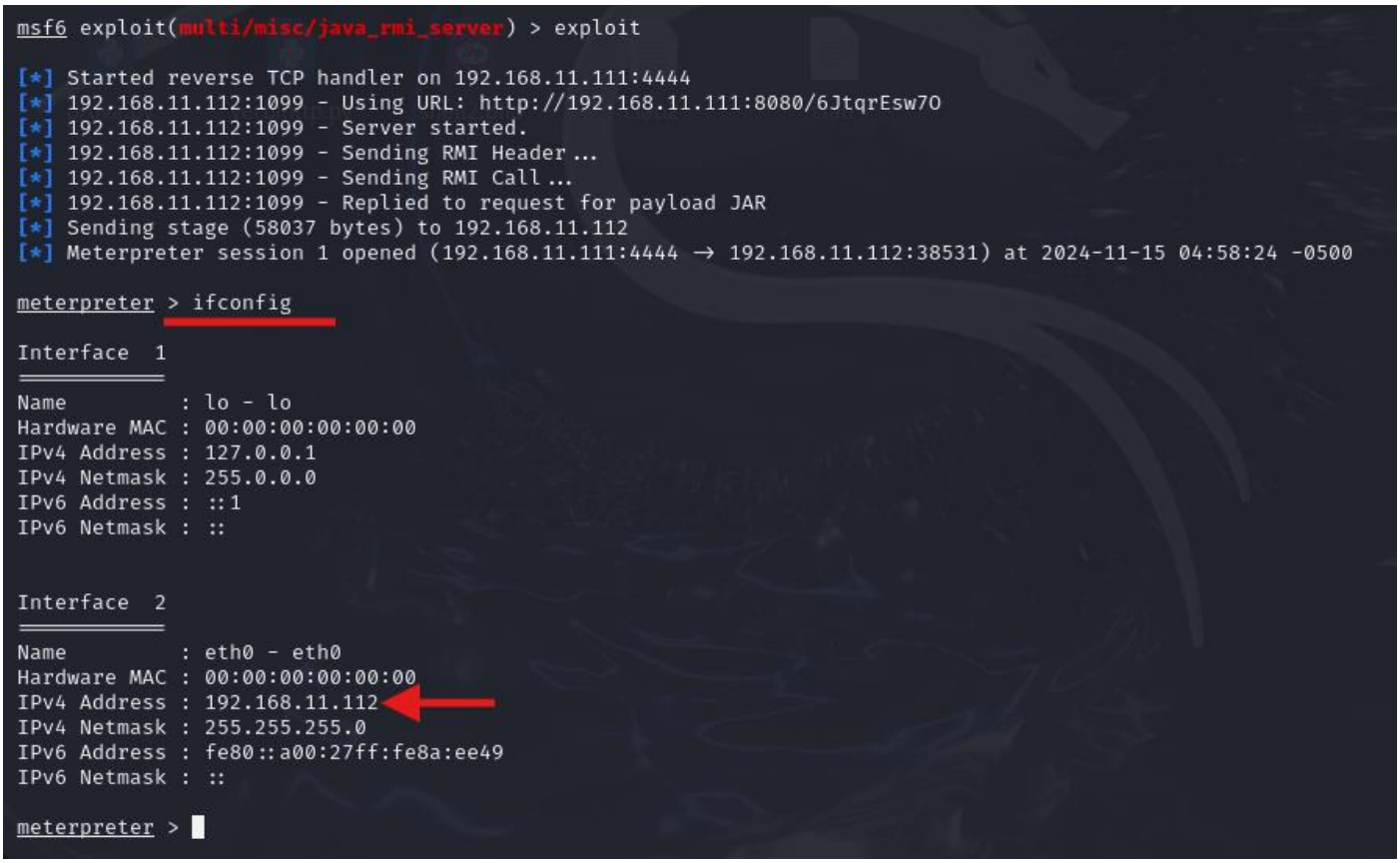
- Ogni richiesta HTTP inviata dal payload alla macchina attaccante avrà un ritardo di 20 secondi.

Quando usarlo:

Se durante un attacco con un payload reverse_http o reverse_https ottieni errori legati alla connessione, instabilità nella sessione, o se il tuo obiettivo è ridurre il "rumore" generato dall'attacco sulla rete, modificare HTTPDELAY è una strategia valida.

Di default, il valore potrebbe essere basso (ad esempio, 5 secondi), ma in situazioni particolari, un valore più alto, come 20 o 30 secondi, potrebbe aiutare a risolvere i problemi.

RACCOLTA CONFIGURAZIONE DI RETE



Raccogliere evidenze sulla macchina remota significa **ottenere informazioni utili** sulla configurazione e sulle caratteristiche del sistema compromesso. Questo è un passo fondamentale per comprendere meglio l'ambiente di rete della macchina vittima e pianificare eventuali mosse successive, come la lateralizzazione o l'esfiltrazione di dati.

Configurazione di rete

Raccogliere la configurazione di rete significa ottenere informazioni relative a:

- Indirizzi IP assegnati alla macchina (sia IPv4 che IPv6).
- Interfacce di rete disponibili (ad esempio, Ethernet, Wi-Fi, ecc.).
- Maschere di sottorete associate agli indirizzi IP.
- Gateway predefinito utilizzato per accedere ad altre reti (es. Internet).
- Eventuali DNS configurati.

Queste informazioni servono per capire:

- Su quali reti opera la macchina.
- Come comunica con altre macchine.
- Quali porte o servizi potrebbero essere attivi e vulnerabili.

RACCOLTA INFORMAZIONI SULLA TABELLA DI ROUTING DELLA MACCHINA VITTIMA

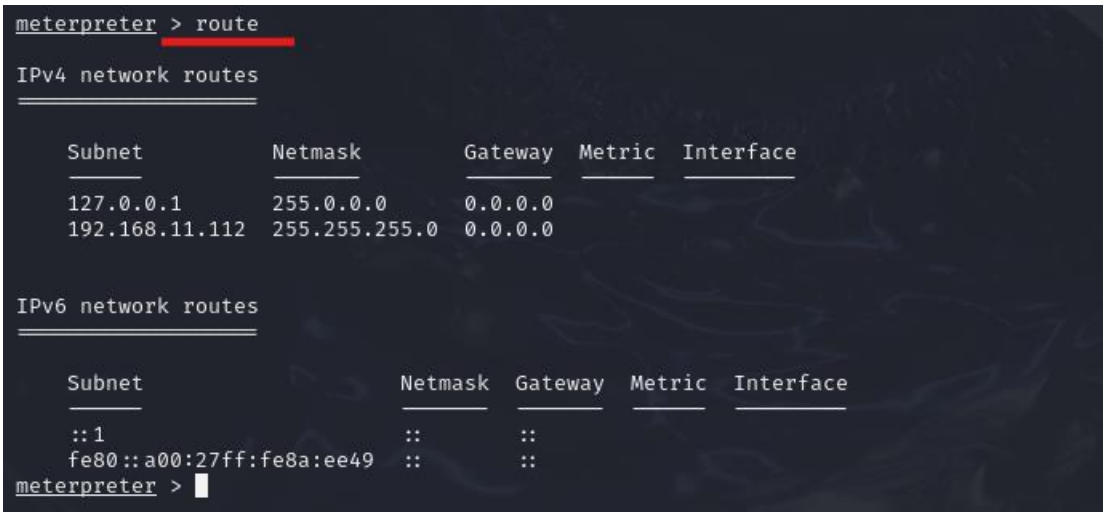


Tabella di routing

La tabella di routing è un elenco delle rotte conosciute dalla macchina, che definiscono come inviare pacchetti di rete verso altre destinazioni. Ogni riga nella tabella di routing include:

- La rete di destinazione (es. 192.168.1.0/24).
- Il gateway utilizzato per raggiungere quella rete.
- L'interfaccia di rete specifica attraverso cui i pacchetti verranno inviati.

Questa informazione serve per:

- Determinare quali altre reti possono essere raggiunte dalla macchina compromessa.
- Identificare eventuali punti di ingresso o uscita verso reti più critiche.
- Comprendere l'architettura della rete target.
-

In sintesi, raccogliere queste evidenze è un'operazione di ricognizione interna al sistema compromesso, essenziale per un penetration test o per attività post-sfruttamento, che consente di valutare le opportunità di proseguire l'attacco o fornire report dettagliati.