PROGETTO S7E5

Lo scopo del progetto di oggi era quello di sfruttare il servizio vulnerabile della nostra macchina vittima, aperto sulla porta 1099, per avviare una sessione di Meterpreter ed ottenere

* configurazione di rete
* tabella di routing

di Metasploitable che in questo caso rappresenta la nostra macchina target.

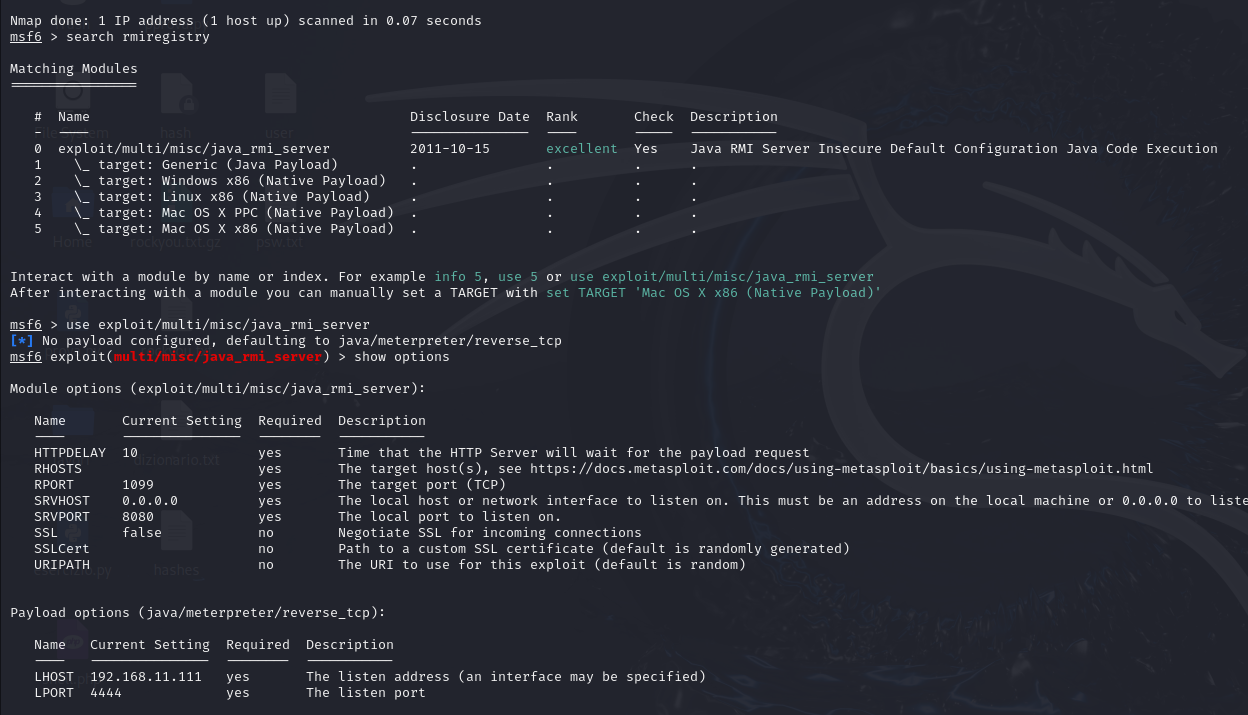
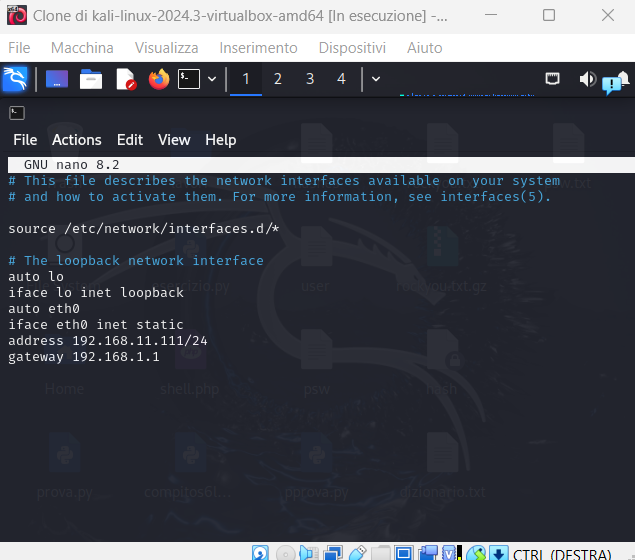
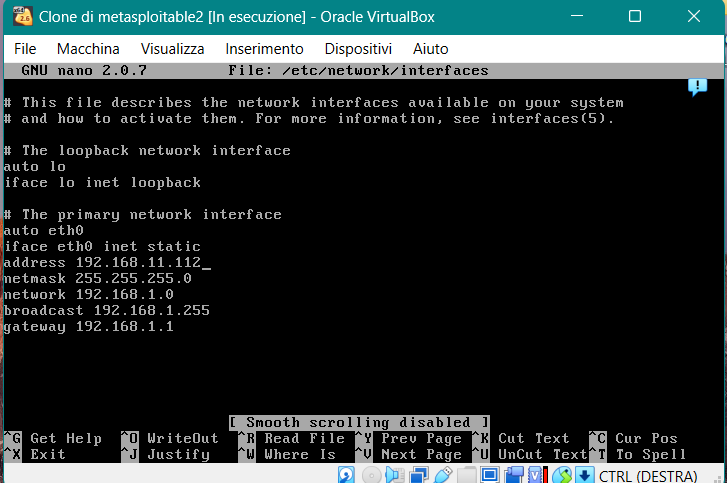
Come primo punto focale c’è stato chiesto di modificare gli indirizzi IP delle nostre macchine configurandoli come

* Indirizzo IP Kali **192.168.11.111/24**
* Indirizzo IP Meta **192.168.11.112/24**

Per attuare la configurazione richiesta viene usato il comando

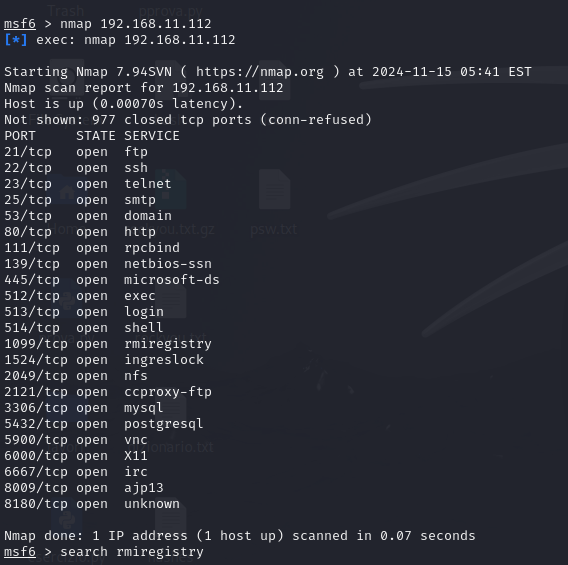
**sudo nano /etc/network/interfaces** dove

* **Sudo** ci dà i privilegi di root che ci permettono effettivamente di modificare il file in questione
* **Nano** è l’editor di testo che utilizziamo per modificare il file
* la parte successiva agli slash rappresenta il percorso svolto attraverso le directory, per arrivare all’interno del file interfaces



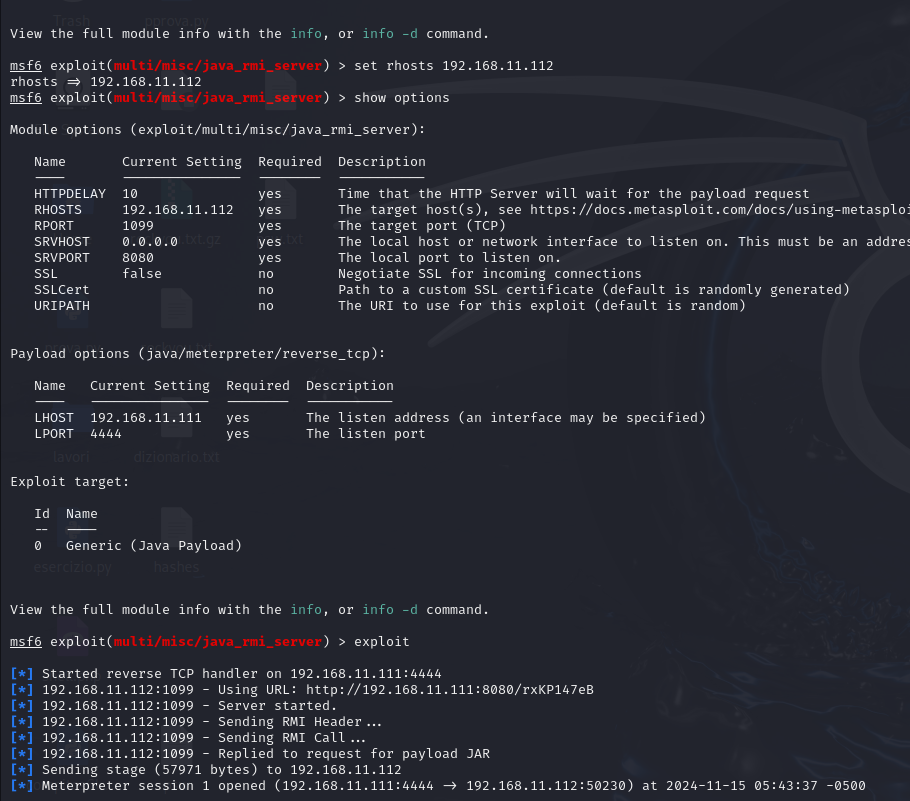
Cerchiamo quindi l’exploit adatto per sfruttare le vulnerabilità del servizio che gira dietro la porta 1099

Dopo esserci assicurati che le macchine comunichino tra loro; avviamo il tool Metasploit con il comando **msfconsole** ed effettuiamo uno scan di rete, per assicurarci che la porta necessaria per sfruttare la vulnerabilità del nostro target, sia aperta.

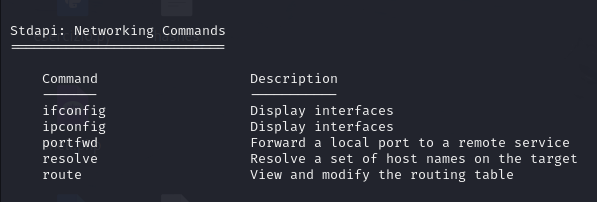


Avviamo quindi l’exploit ed apriamo una sessione di Meterpreter tra la macchina attaccante, Kali, e la macchina attaccata, Metasploitable.

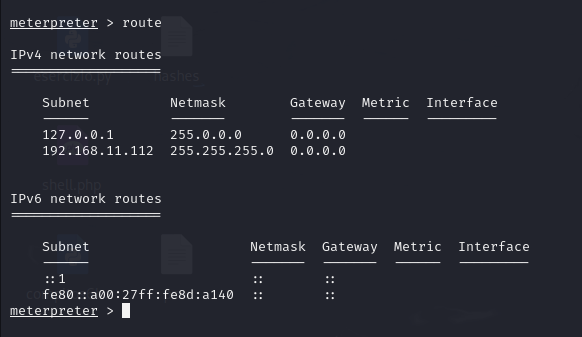
Una volta scelto l’exploit appropriato settiamo le impostazioni necessarie affinché la shell avvenga; in questo caso solamente l’rhost che corrisponde all’indirizzo IP della macchina vittima.



Aperta la shell, con il comando **help** possiamo visionare tutti i comandi utili per il payload Meterpreter ed utilizzare quello più consono al nostro scopo.



Avremo quindi l’interfaccia della configurazione di rete con il comando **ifconfig;**



E le informazioni relative alla tabella di routing con il comando **route**.

**Relazione sugli exploit**

La fase 3 del penetration test è forse la fase più importante e cruciale del pentest perché riguarda l’exploit, ovvero l’attacco vero e proprio.

Quando si parla di exploit si intende un codice malevolo che va ad agire su una vulnerabilità che è già presente all’interno del sistema; per questo motivo sono cruciali le fasi successive affinché l’exploit funzioni.

In questa fase, infatti, vengono confermate tutte le informazioni che sono state raccolte durante le fasi successive e sperimentate le vulnerabilità trovate per cercare di eliminarle o mitigare.

Lo scopo di un exploit è quello di creare una shell con la macchina vittima, e affinché abbia successo, è necessario che si rispettino 4 parametri:

* il software da attaccare deve essere in esecuzione sulla macchina vittima
* lo stesso software non deve presentare un aggiornamento che invalidi l’exploit
* l’exploit deve essere progettato per quella versione esatta del software da attaccare
* l’attaccante deve trovarsi all’interno della stessa rete della macchina target

se una di queste 4 condizioni non risulta vera, l’exploit non avrà successo; per questo motivo per proteggersi dagli exploit è molto importante tenere i sistemi costantemente aggiornati.

Il quarto punto è fondamentale perché se l’attaccante dovesse trovarsi fuori dalla rete della macchina vittima, entrerebbe in gioco il NAT-PAT che associa ad un IP privato un IP pubblico con cui quel dispositivo può uscire dalla sua rete LAN e navigare in rete WAN; essendoci appunto questo sistema, l’indirizzo IP che l’attaccante vedrebbe dall’esterno della LAN del target, sarebbe un IP pubblico che non necessariamente corrisponde al dispositivo da attaccare. Per tale motivo quando si cerca di attaccare un dispositivo rimanendo esterni alla sua rete, non si può usare un exploit classico, ma l’attaccante potrà mandare un link malevolo tramite ingegneria sociale che presenterà al suo interno un payload che permetterà la creazione di una shell.

Per capire bene la fase d’exploit è necessario suddividerla in tre sotto fasi:

1. la prima sotto fase è quella di **exploit** dove l’attaccante sceglie un codice malevolo che gli permetta di bucare il sistema della macchina target
2. la seconda è quella del **payload** che corrisponde ad un file malevolo che svolge una funzione da ponte tra l’attaccante e la vittima, permettendo di creare la shell vera e propria all’interno del nostro target
3. l’ultima sotto fase è quella della **shell**, ovvero della connessione tra attaccante e vittima; la shell può essere di due tipi:
   * + bind shell dove è l’attaccante a creare la connessione con la vittima
     + reverse shell dove la vittima crea la connessione con l’attaccante.

Quest’ultima è la più utilizzata in quanto permette di eludere il firewall che non vedrà la connessione come malevola essendo in uscita.

**HTTPDELAY**

Uno dei parametri fondamentali per la riuscita della shell è il parametro HTTPDelay che corrisponde al tempo di attesa che corre tra una richiesta e l’altra inviate dal payload; più grande è il numero corrispondente all’HTTPDelay, maggiore è la latenza tra le richieste, più sarà facile per un attaccante passare inosservato; questo perché i pacchetti mandati lentamente faranno meno rumore e riusciranno più facilmente ad eludere sistemi come IDS e IPS, che non le riconosceranno come richieste malevole.