S7L5

Exploit Java Rmi (Porta 1099)

Cosa è java rmi?

Java RMI (Remote Method Invocation) è una tecnologia che permette a un'applicazione Java di invocare metodi di oggetti situati su una macchina remota, come se fossero oggetti locali. È una delle soluzioni di Java per l'implementazione di sistemi distribuiti.
È come avere un telefono con cui chiedere a qualcuno di fare un'operazione per te, senza dover fare tutto da solo.
Quindi, **Java RMI** è un modo semplice per i programmi Java su computer diversi di comunicare e lavorare insieme, utilizzando funzioni e metodi che non sono fisicamente presenti sulla macchina locale ma su un'altra macchina.

Perché è vulnerabile?

Il **Java RMI Registry** sulla porta **1099**, per impostazione predefinita, **non richiede autenticazione**. Questo significa che chiunque conosca l'indirizzo IP del server e la porta può provare a collegarsi al servizio e vedere quali oggetti sono registrati.
In alcuni casi, i metodi esposti tramite RMI possono avere accesso al filesystem, al database o ad altre risorse sensibili del server. Se un attaccante riesce a invocare questi metodi, potrebbe:

- Leggere o modificare file sensibili.
- Accedere a dati riservati.
- Effettuare modifiche non autorizzate al sistema.

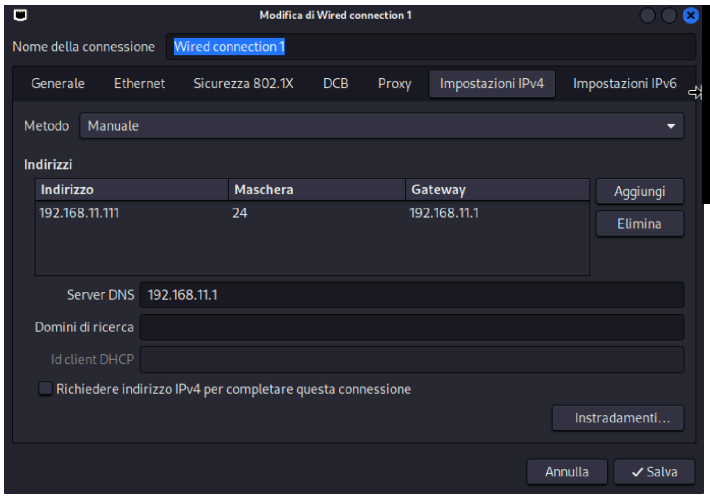
Come la sfrutterò?

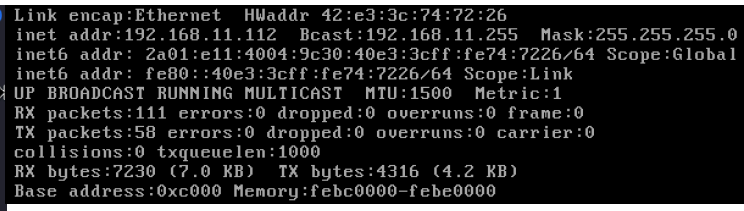
Il mio obiettivo è sfruttare una vulnerabilità nel servizio Java RMI esposto sulla porta 1099 della macchina target per ottenere l'accesso remoto. Una volta ottenuto l'accesso, avvierò una sessione di **Meterpreter** per prendere il controllo del sistema. Da lì, inizierò a esplorare la configurazione di rete, verificando gli indirizzi IP configurati, le interfacce di rete attive e la tabella di routing. Questo mi permetterà di capire meglio l'architettura della rete interna e individuare possibili vie di accesso verso altre macchine o servizi presenti nella rete target.

Esercizio pratico

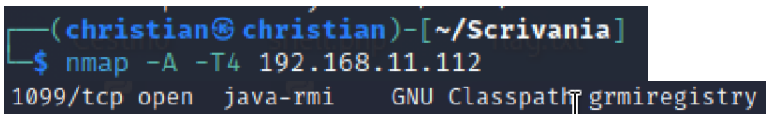
Configurazione macchine e scan nmap

Il primo passaggio che andrò a svolgere sarà posizionare la macchina target (Metasploitable2) e la macchina attaccante (Kali linux) nella stessa rete con due indirizzi IP differenti.





Dopo aver configurato le due macchine andrò ad utilizzare il comando **nmap** per fare una scansione sulla macchina target, in modo da visionare i protocolli attivi e eventuali versioni vulnerabili dei protocolli attivi. In questo caso specifico andrò a controllare soprattutto la porta 1099.



Fase exploit

Che cos'è la fase di exploit?

In questa fase andrò ad utilizzare un'exploit; ovvero del codice che va a sfruttare una vulnerabilità già presente all'interno del servizio che sto decidendo di attaccare. Per farlo utilizzerò il framework **Metasploit.**

Metasploit è un tool che mette a disposizione una serie di exploit conosciuti e permette di lanciarli verso la macchina target con estrema facilità.

Per poter utilizzare questo tool basta digitare il comando **msfconsole** all'interno di un terminale in Kali linux.

Ricerca del giusto exploit

Metasploit mette a disposizione una quantità di exploit enorme. Per poter filtrare meglio e scegliere con più facilità il codice di cui ho bisogno andrò ad utilizzare il comando **search** seguito dal nome del servizio che voglio andare a violare in questo caso come viene mostrato dalla scansione fatta prima con **nmap** il servizio sarà **java-rmi.**

```
msf6 > search java_rmi

Matching Modules
================

   #  Name                                     Disclosure Date  Rank       Check  Des
cription
   -  ----                                     ---------------  ----       -----  ---
-----
   0  auxiliary/gather/java_rmi_registry                        normal     No     Jav
a RMI Registry Interfaces Enumeration
   1  exploit/multi/misc/java_rmi_server       2011-10-15       excellent  Yes    Jav
a RMI Server Insecure Default Configuration Java Code Execution
   2  auxiliary/scanner/misc/java_rmi_server   2011-10-15       normal     No     Jav
a RMI Server Insecure Endpoint Code Execution Scanner
   3  exploit/multi/browser/java_rmi_connection_impl  2010-03-31       excellent  No     Jav
a RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/brow
ser/java_rmi_connection_impl
```

Come si può vedere dall'immagine metasploit mi fornisce quattro diversi exploit utilizzabili. È buona prassi provare ad utilizzare tutti gli exploit a disposizione per vedere qual'è il più efficace tra tutti. In questo caso però mi limiterò ad usare il primo exploit con rank **excellent.**

Per poter utilizzare tale exploit posso scrivere il comando **use** seguito dal numero di exploit desiderato, in questo caso **1,** oppure seguito dall'intero **path** dell'exploit ovvero **exploit/multi/misc/java_rmi_server** (scrivere l'intero path è la scelta consigliata perché decisamente più precisa).

Configurazione exploit

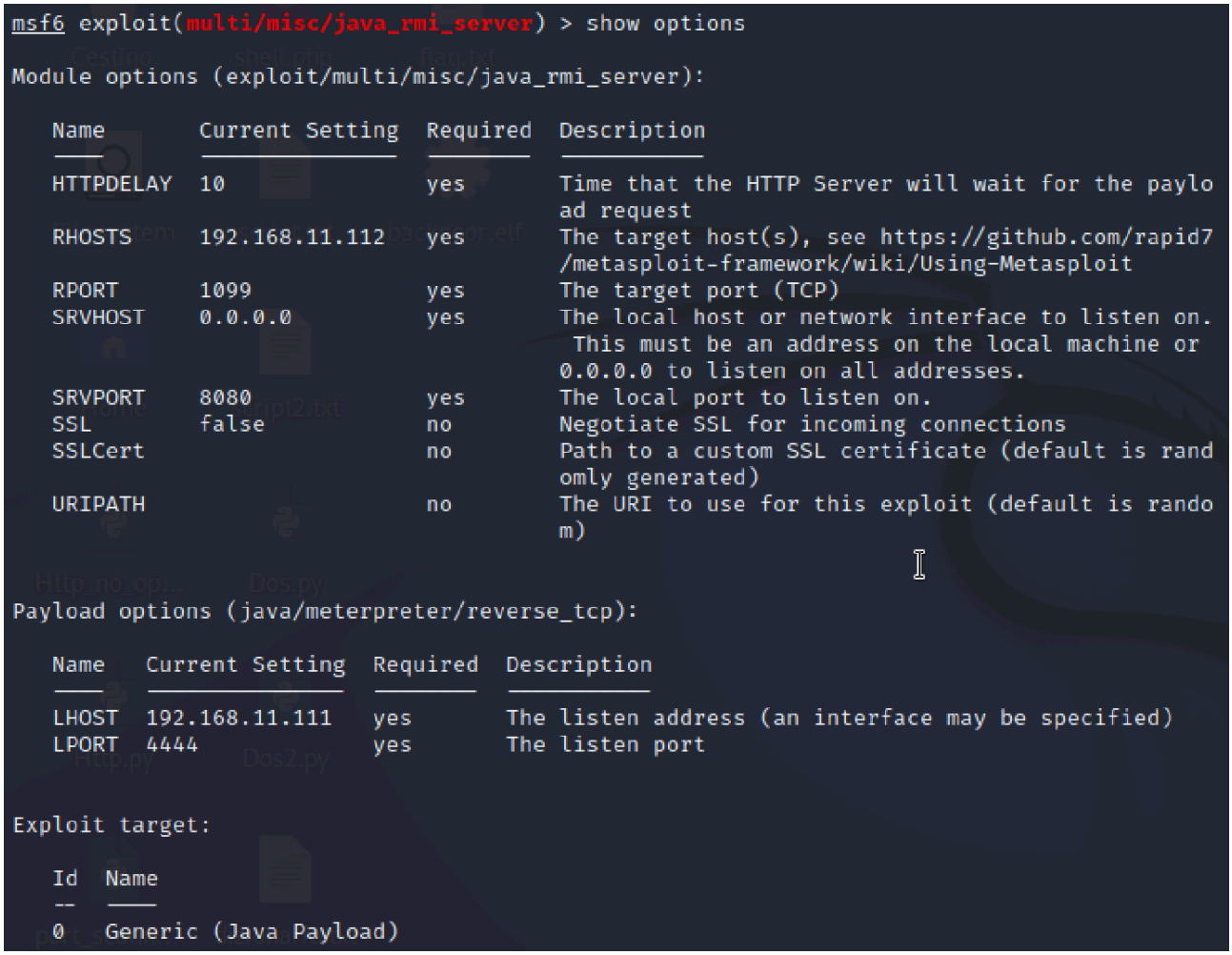
Dopo aver scelto l'exploit una parte fondamentale è andarlo a configurare.

Il primo dato da scegliere con attenzione è il **Payload**.

Il **Payload** è un codice malevolo che mi permetterà di creare una **shell** tra me e la macchina vittima. Una **shell** invece è una connessione tra attaccante e vittima e mi permette di controllare, con vari comandi, il target.

In questo caso utilizzerò il payload **java/meterpreter/reverse_tcp** ciò significa che grazie all'exploit scelto, aprirò una connessione meterpreter con una shell in reverse tcp, ovvero la connessione partirà dalla macchina target alla macchina attaccante. Tale tecnica si utilizza per evitare che un eventuale firewall a filtraggio dinamico possa bloccare la comunicazione.

Per poter visionare la restante parte di parametri da configurare il comando da utilizzare è **show options.**



Come si può vedere dalla schermata ci sono una serie di parametri obbligatori da dover settare, in particolare l'**RHOSTS** ovvero l'indirizzo IP della macchina target e l'**HTTPDELAY.**

Cosa è l'HTTPDELAY?

L'httpdelay si riferisce al ritardo che si verifica nel processo di richiesta e risposta HTTP tra un client (ad esempio, un browser web) e un server. Questo ritardo può essere percepito come il tempo che intercorre tra il momento in cui l'utente invia una richiesta a un sito web e il momento in cui riceve una risposta dal server. In un contesto di sicurezza informatica, come nel caso di un exploit contro una macchina vulnerabile come **Metasploitable 2**, l'aumento dell'**HTTP delay** o di altri tipi di ritardo nelle comunicazioni può essere utilizzato in vari modi, specialmente durante un attacco sfruttando una vulnerabilità. In questo caso dilazionare questo tempo di risposta mi potrebbe permettere di eludere eventuali sistemi di sicurezza. Però come prima prova lascerò **HTTPDELAY** con il valore **10.**

Lancio exploit e controllo della macchina target

Dopo aver configurato tutti i parametri correttamente utilizzerò il comando **exploit** per lanciare l'attacco, se l'attacco andrà a buon fine si creerà una sessione meterpreter e grazie a tale sessione io potrò utilizzare tutti i comandi che desidero sulla macchina target. In questo caso utilizzerò i comandi **ipconfig** e **rout list** per poter visionare rispettivamente configurazione di rete e informazioni sulla tabella di Routing.

Immagine 1 - connessione avvenuta correttamente

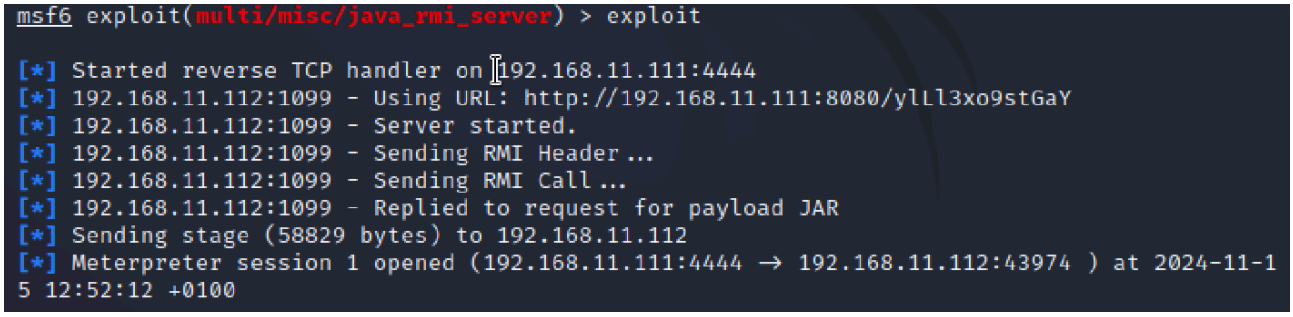


Immagine 2 - comando ifconfig

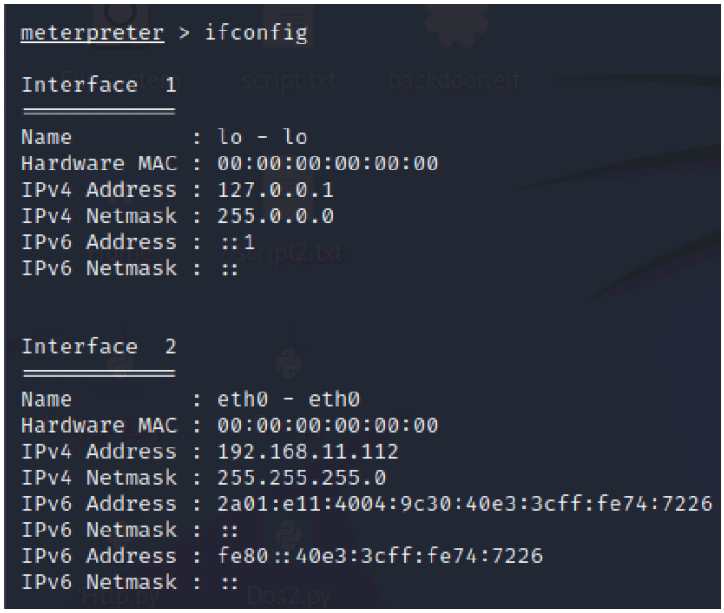
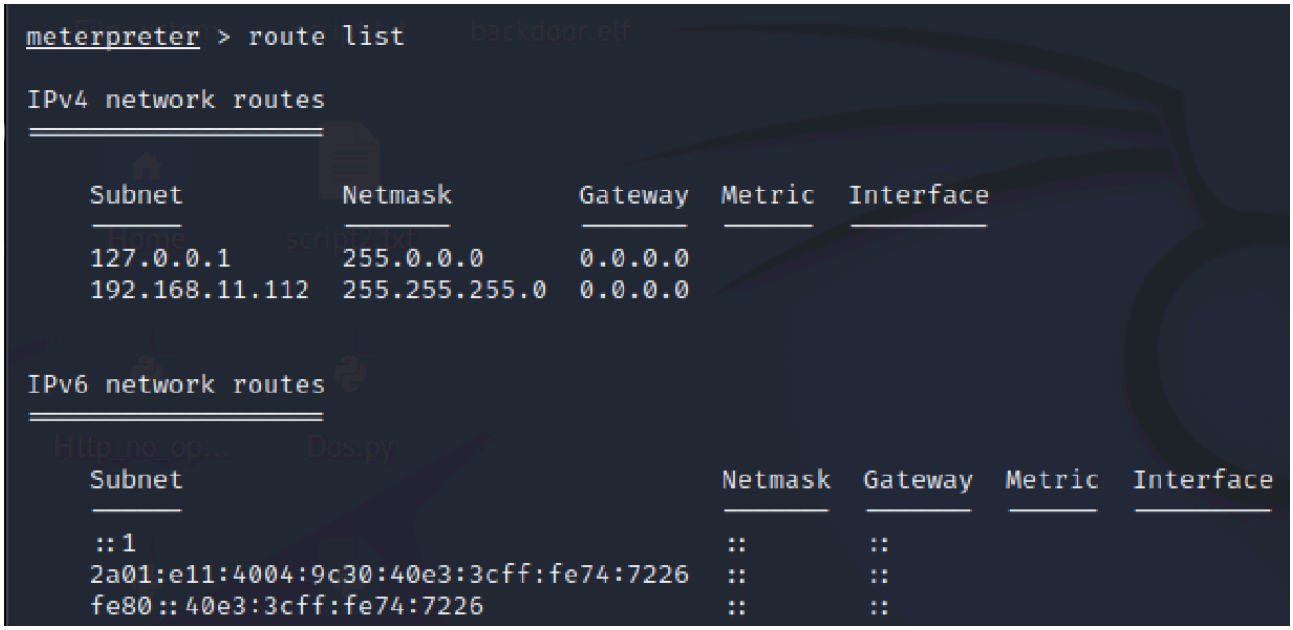


Immagine 3 - comando route List



Conclusioni

Sfruttare un servizio vulnerabile può rivelarsi piuttosto semplice: è sufficiente configurare alcune variabili specifiche per riuscire a compromettere facilmente la macchina target. Tuttavia, per mitigare efficacemente queste vulnerabilità, è fondamentale mantenere aggiornati i protocolli di sicurezza all'ultima versione disponibile. Questo permette di correggere le falle conosciute e di ridurre il rischio di exploit.