S7/L5

Traccia:

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.

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.

Ci troviamo nella fase 3 di un penetration testing, nella fase dell’exploit, ovvero il momento in cui si cerca di sfruttare le vulnerabilità identificate nel sistema target. L'obiettivo è utilizzare l’exploit più appropriato per ottenere accesso non autorizzato al sistema e nel caso per eseguire codice malevolo.

L’exploit a differenza del malware, è un codice che va a sfruttare una vulnerabilità già presente nel software, mentre il malware si va ad immettere in un codice legittimo andando ad “azionare” una vulnerabilità. Possiamo dire che l’exploit risulta più efficace del malware, in quanto nel malware è anche complice l’essere umano, mentre nell’exploit, se vengono rispettate determinate condizioni (tra cui per esempio che il programma sulla macchina vittima sia attivo) è una via a senso unico, ovvero non vi è interazione con la vittima.

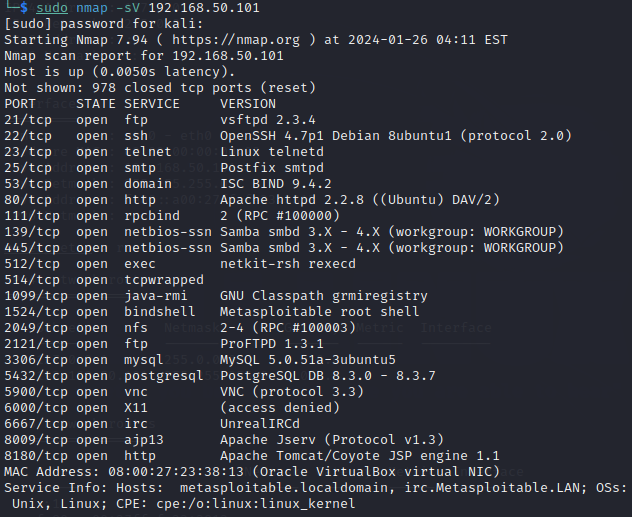
Di solito questo problema si risolve correggendo il codice del programma attraverso degli upgrade/aggiornamenti. In una situazione ideale, la soluzione migliore è quella di testare prima il nuovo aggiornamento su un dispositivo clone (sia a livello software, sia hardware).

Step 1: Per prima cosa apriamo le nostre macchine kali (attaccante) e metasploitable2 (vittima), che ovviamente dovranno comunicare (stessa rete).

Step 2: Un tool utilissimo per questo tipo di attività è Metasploit, ovvero una piattaforma open-source usata per lo sviluppo e utilizzo di exploit.

Possiamo attivare l’interfaccia di Metasploit attraverso il comando MSFCONSOLE.

Step 3: Utilizziamo nmap + l’IP della macchina vittima per fare una scansione di rete e verificare quali porte siano aperte:





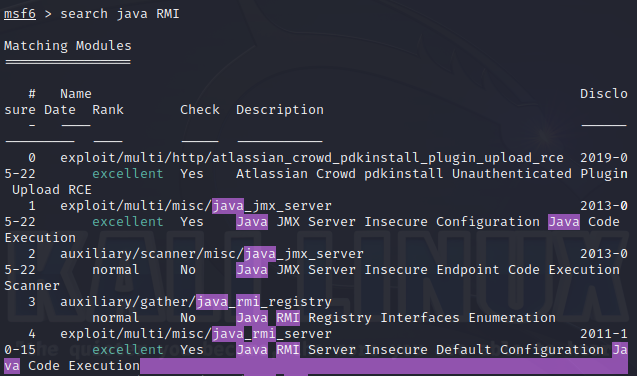
Step 4: Andiamo ora ad identificare il servizio vulnerabile.

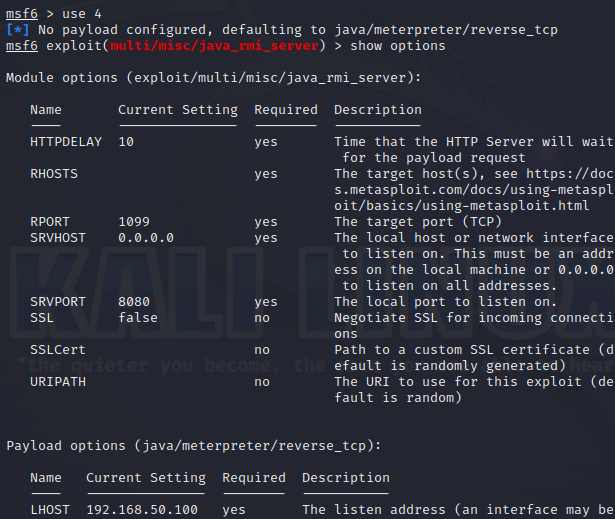
La traccia ci chiede di sfruttare la vulnerabilità sulla porta aperta 1099 – Java RMI.

La Java Remote Method Invocation, permette a programmi scritti in Java di comunicare tra loro su reti diverse, ma anche su computer diversi. La porta 1099 è spesso utilizzata come porta di default per il registro RMI, un servizio che funge da registro centrale per le informazioni e consente la comunicazione tra le applicazioni Java.

Step 5: Inseriamo la keyword “search Java RMI” e da qui otteniamo un elenco di exploit che sono correlati al servizio Java\_rmi. Dopo aver scelto l’exploit più adatto da utilizzare (o meglio dopo averli testati tutti, o fino a quando non troviamo il primo che funzioni), lo attiviamo con il comando “use” seguito dal percorso dell’exploit, in questo caso selezioniamo “use 4”:

exploit/multi/misc/java\_rmi\_server

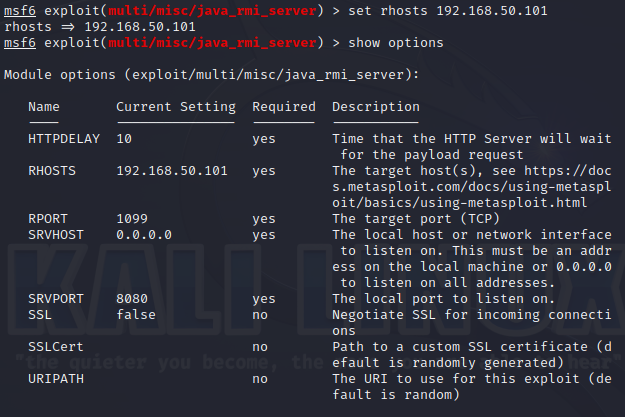




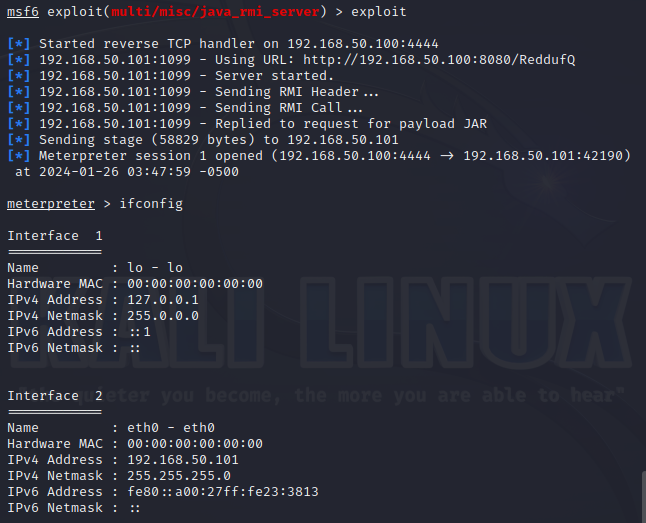
Qui troviamo un payload flessibile, molto utile e potente nella fase dell’exploit, ovvero “meterpreter”. Meterpreter può essere utilizzato per accedere in maniera non autorizzata e quindi ad avere accesso remoto e completo sul sistema target.

Step 6: Vediamo ora le opzioni richieste tramite il comando “show options” e notiamo che alcune sono obbligatorie da inserire per utilizzare l’exploit.

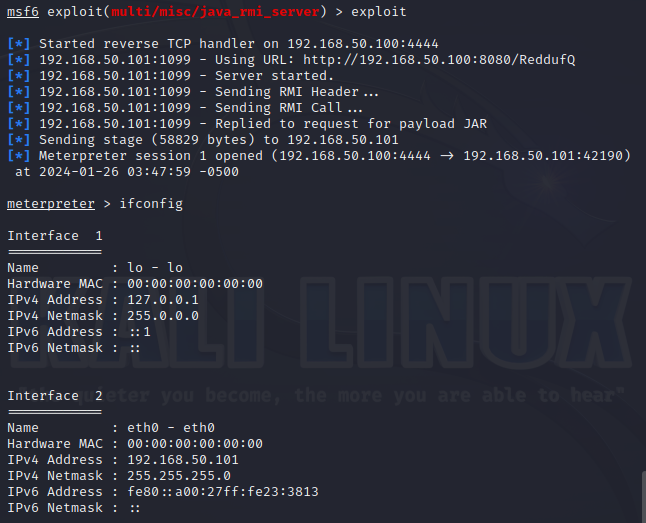
Ad esempio RPORT, ovvero la porta sulla macchina target dove il servizio è in ascolto (1099 già inserita) ed RHOSTS, ovvero l’indirizzo IP della macchina vittima, in questo caso da inserire (required = yes), Per cui settiamo l’IP tramite il comando “set rhosts + IP metasploitable” e per controllare se viene salvato rimandiamo un altro “show options”:



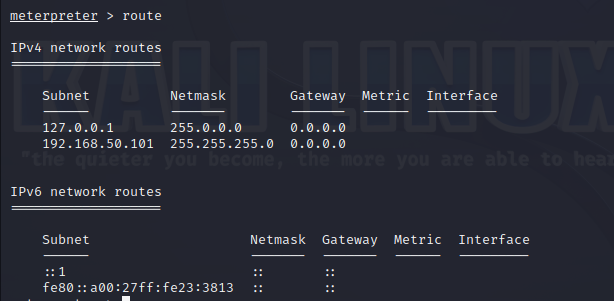
Step 7: Adesso attiviamo l’exploit con il relativo comando “exploit”; se ha successo otteniamo l’accesso al target/macchina vittima, con una sessione aperta di meterpreter:



Step 8: L’esercizio ci chiede di raccogliere informazioni sul target, ovvero sulla sua configurazione di rete, tramite il comando “ifconfig”: (IP di metasploitable + subnet mask)



Step 9: Dopo di che, verifichiamo le informazioni sulla tabella di routing tramite il comando “route”:



Come possiamo vedere ci torna la tabella di routing di Metasploitable, ovvero la tabella che contiene informazioni sulle rotte di rete che la macchina conosce e utilizza per instradare il traffico di rete.

In sintesi, tramite un exploit su Metasploit (approccio automatico) siamo andati a sfruttare una vulnerabilità (in questo caso la porta 1099 del servizio Java rmi) già presente nel sistema target, al fine di ottenere un accesso non autorizzato, remoto e completo attraverso una sessione di meterpreter. Infatti, dopo che l’exploit ha avuto successo, si può attivare un “payload” ovvero si esegue il file malevolo, cioè ciò che l’attaccante desidera ottenere o eseguire sul target specificato. Possiamo quindi eseguire comandi sulla macchina target, raccogliere informazioni (dati utente, configurazione di rete, ecc…) ed esplorare la rete interna per ampliare l’accesso e compromettere eventualmente ulteriori sistemi. Importante ricordare che affinché l’exploit funzioni il software/programma sulla macchina target deve essere in esecuzione.

Alla fine dunque si è creata una shell/connessione tra le due macchine. Se è l’attaccante che crea la connessione verso la vittima si parla di bind shell; se invece è la vittima che crea una connessione verso l’attaccante parliamo di reverse shell (questo è dovuto al fatto che il firewall dinamico blocca solo le connessioni in entrata, ma non in uscita).