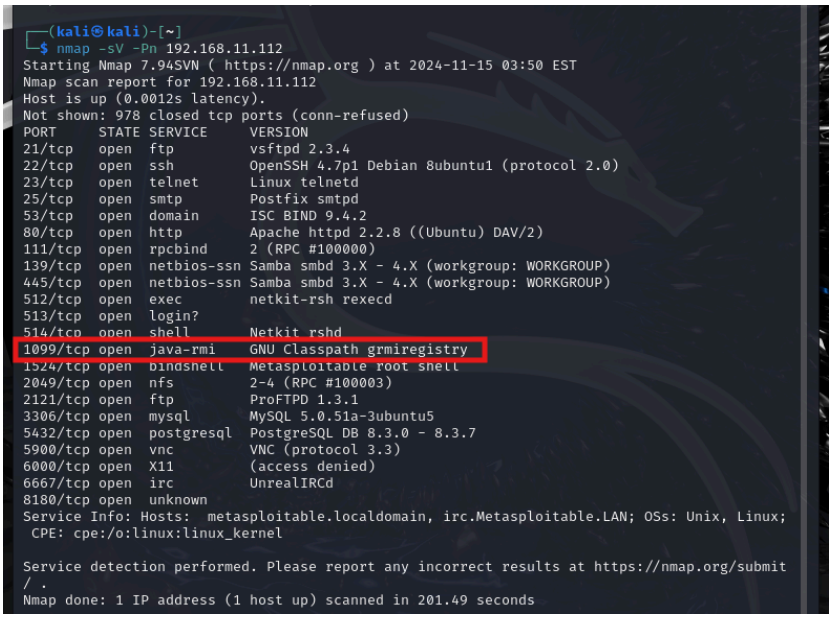
EXPLOIT JAVA-RMI

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.
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:
Il progetto odierno ci chiede di effettuare un attacco alla vulnerabilità nota Java RMI (Remote Method Invocation), si tratta di una falla di sicurezza tra le più conosciute che coinvolge le applicazioni Java, si manifesta principalmente quando i server RMI espongono oggetti non protetti o permettono l'esecuzione di codice remoto non autorizzato.
Exploit
Per prima cosa ho cambiato gli IP, impostandoli come richiesto dalla traccia.
Una cosa importante da fare sicuramente è una scansione delle porte grazie ad Nmap, per avere una visione globale delle porte aperte, e verificare se la porta 1099 (ovvero quella dedicata al servizio Java RMI) sia effettivamente attaccabile.



Il passo successivo sarà avviare Metasploit e cercare l'exploit utilizzando le parole chiave, ovvero java_rmi, l'otuput che ci restituirà saranno gli exploit disponibili per questa vulnerabilità.

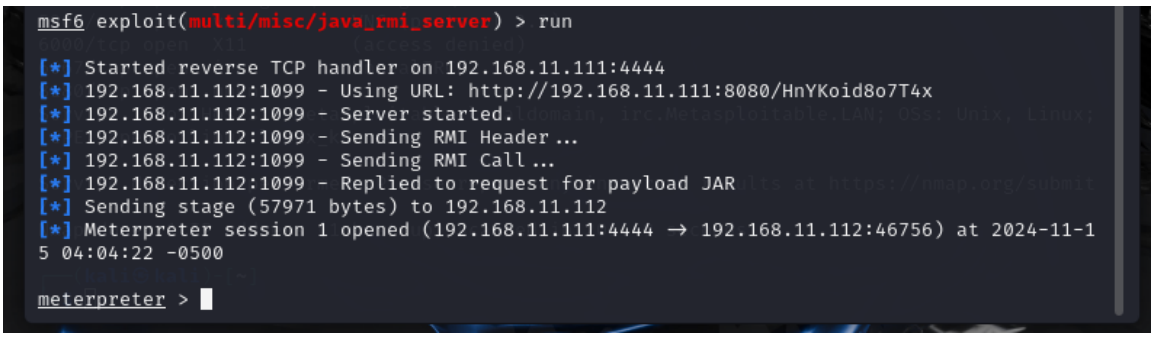
Cos'è Java RMI?
Java RMI è una tecnologia che consente ad un programma Java di eseguire, su oggetti presenti su server remoti come se fossero locali, dei metodi. Ciò permette la costruzione di applicazioni distribuite, ma può provocare potenziali vulnerabilità di sicurezza, quando non configurato correttamente.
Ora andremo ed eseguire l'exploit:

/exploit/multi/misc/java_rmi_server

Questo exploit punta ad attaccare il Java RMI Registry, consentendo l'esecuzione di un codice arbitrario (RCE, Remote Code Execution) su un server remoto che rende visibile l'RMI senza un' adeguata autenticazione o controlli di sicurezza. La porta predefinita per questo servizio è la 1099.
Il Java RMI Registry è un servizio che consente ad un client remoto di provare a richiedere metodi su oggetti remoti, la sua falla a livello di sicurezza ha origine quando tale servizio permette di caricare classi non conosciute o non controlla in maniera adeguata le chiamate eseguite dal client, per renderlo più sicuro andrebbe configurato in maniera corretta, se la configurazione corretta è assente è possibile sfruttarlo per inviare oggetti malevoli oppure la deserializzazione di oggetti Java, ciò che permette, appunto, l'esecuzione del codice arbitrario.

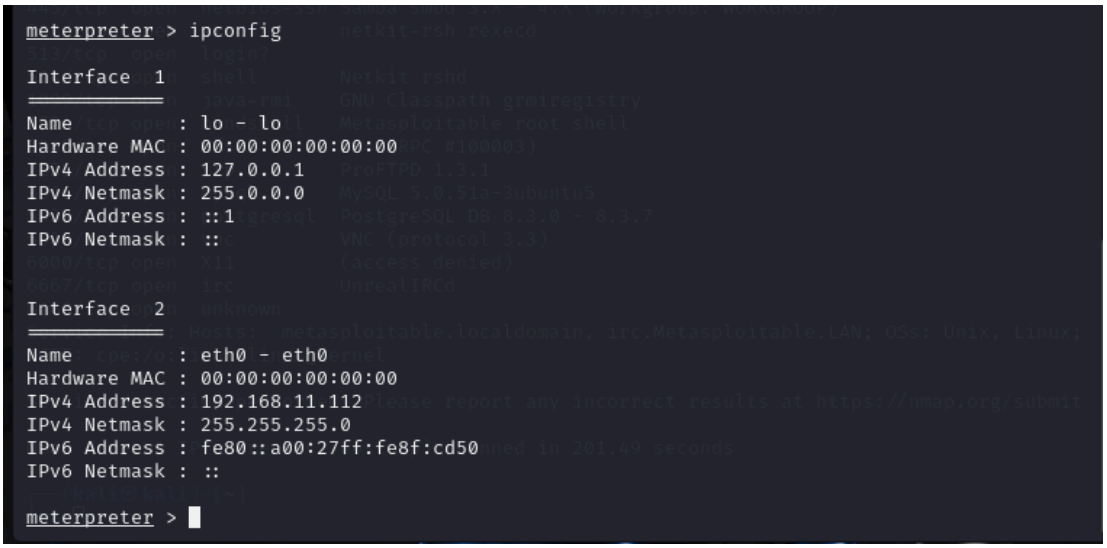
Payload
Per avviare l'exploit si richiederà di impostare il payload in maniera adeguata, sarà obbligatorio, in questo caso, inserire l'indirizzo IP del target.
Un payload, nel contesto di sicurezza informatica, è un codice che viene eseguito dopo che una vulnerabilità è stata sfruttata con successo, ovvero il "carico utile" che l'attaccante inserisce in un exploit per ottenere specifici risultati, ad esempio l'apertura di una backdoor, una connessione remota o l'esecuzione di comandi arbitrari.

Meterpreter
Una volta impostato in maniera corretta potremo andare ad eseguire l'exploit vero e proprio per poter ottenere una sessione Meterpreter.
La sessione Meterpreter è una connessione interattiva tra attaccante e sistema bersaglio compromesso, per ottenerla serve l'utilizzo del payload Meterpreter. Serve a fornire una shell invisibile che utilizzerà i comandi ed operazioni sulla macchina bersaglio senza lasciare tracce sul disco, quindi essendo difficile da individuare.



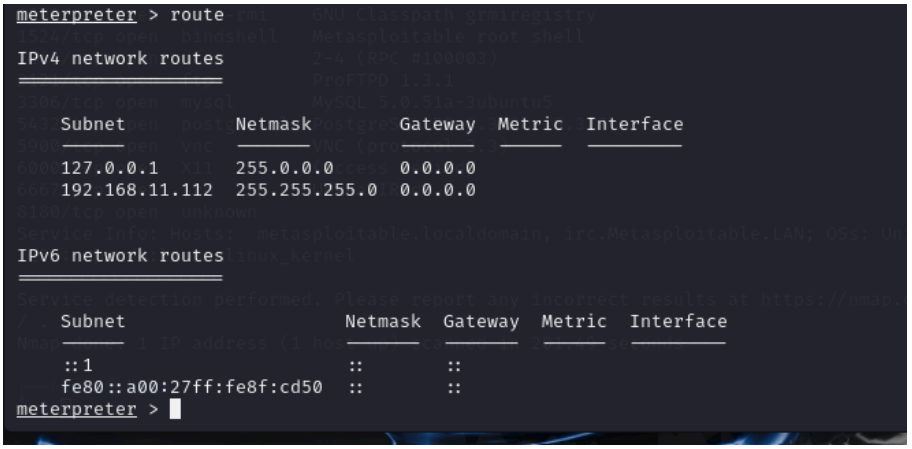
Una volta aperta la sessione possiamo dire di essere entrati all'interno del dispositivo attaccato e potremo eseguire dei comandi come se fossimo al suo interno.

Configurazione di rete
Per ottenere la configurazione di rete della macchina target, ci basterà effettuare un semplice comando *ifconfig* che ci restituirà i dati di rete di Metasploitable.



Come si può notare, l'IP che ci mostra è esattamente quello impostato precedentemente sulla macchina Metasploitable, ciò ci conferma che la sessione è stata aperta con successo.

Tabella di routing
Si richiede anche di visualizzare la tabella di routing, sempre sfruttando la sessione Meterpreter appena aperta, per ottenerla basterà il comando *route*.



La tabella di routing è una struttura presente nei router e nei dispositivi che indica il modo in cui vengono instradati i pacchetti verso la giusta destinazione. Le informazioni che contiene riguardano i possibili percorsi che i pacchetti potranno seguire per raggiungere reti o dispositivi all'interno della LAN (Local Area Network).
Gli elementi visualizzabili sulla tabella di routing sono:

- Indirizzo di destinazione;
- Decisione del percorso;
- Instradamento.

Errore HTTPDELAY
In alcuni casi, durante l'esecuzione di questo exploit, è possibile che si manifesti un errore HTTPDELAY, che ha a che fare con una delle impostazioni del payload appena utilizzato.
L'opzione HttpDelay è un parametro configurabile nei moduli di attacco basati sul protocollo HTTP, la sua funzione è quella di introdurre un ritardo tra le risposte HTTP. L'errore si manifesta quando il parametro è stato configurato in modo errato e può causare un comportamento indesiderato durante l'avvio dell'exploit, non consentendo l'apertura della sessione meterpreter.
Con il parametro HttpDelay si può specificare il numero di millisecondi che Metasploit inserisce tra le risposte HTTP, è utile per simulare un comportamento più naturale del server ed evitare che l'attacco sia facilmente individuabile, soprattutto in presenza di sistemi IDS/IPS, inoltre può aiutare ad evitare di sovraccaricare il sistema della macchina bersaglio.
L'errore che può manifestarsi spesso è un Timeout, ciò significa che potrebbe essere stato impostato un valore troppo alto, che Metasploit non è in grado di gestire bene

questa opzione, quindi va modificata o cancellata, se non è uno dei parametri considerati obbligatori.
L'errore potrebbe manifestarsi anche dal lato della macchina bersaglio se il ritardo inserito è troppo lungo, con conseguente mancata risposta da parte del server target.

Conclusioni
L'esercizio è servito a dimostrare come sfruttare una vulnerabilità nota del servizio Java RMI su un sistema vulnerabile, utilizzando Metasploit. Il modulo scelto, combinato col payload Meterpreter, ha permesso di ottenere un accesso remoto alla macchina bersaglio. Una volta ottenuto l'accesso, è stato possibile raccogliere i dati richiesti, ovvero la configurazione di rete e la tabella di routing, grazie ai comandi integrati di Meterpreter.
Questo scenario serve a mostrare l'importanza di tenere monitorati i servizi considerati vulnerabili, ed aggiornare regolarmente i sistemi per evitare che vulnerabilità come questa vengano sfruttate da hacker malintenzionati.

Progetto a cura di
Sonia Laterza