**S7/L5**

**Exploit di java**

Ho iniziato con una scansione Nmap sul server target, utilizzando lo script rmi-vuln-classloader per verificare la presenza di vulnerabilità sulla porta 1099, dove è attivo il servizio rmiregistry. Questo script verifica se la configurazione di default del servizio Java RMI permette l’esecuzione di codice remoto caricando classi da URL esterni. La scansione ha confermato la vulnerabilità: la porta era aperta e la configurazione del servizio consentiva il caricamento di codice, rendendo possibile un exploit da remoto. Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Dopo aver identificato questa vulnerabilità, ho aperto Metasploit e selezionato il modulo java\_rmi\_server, specificamente progettato per sfruttare la configurazione vulnerabile dei server RMI. Questo exploit consente di eseguire codice sul server target utilizzando un payload che stabilisce una connessione di tipo "reverse TCP" (java/meterpreter/reverse\_tcp). In questo momento dopo che ho eseguito il "payload" , il sistema target si connette automaticamente alla mia macchina, permettendomi di accedere da remoto.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Successivamente ho configurato i parametri dell’exploit: RHOSTS, impostato sull’indirizzo IP del target (192.168.11.112), e LHOST, impostato sull’IP della mia macchina (192.168.11.111), che sarebbe stato il destinatario della connessione di ritorno. Ho anche specificato LPORT, scegliendo la porta 4444 per ricevere la connessione del payload.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Con tutte le configurazioni pronte, ho eseguito l’exploit. Dopo qualche istante, la connessione è stata stabilita, e mi sono ritrovato con un accesso diretto al sistema remoto. Da qui ho iniziato a raccogliere informazioni sulle interfacce di rete, sulle rotte e sulle configurazioni del sistema, ottenendo una panoramica completa dell’architettura di rete e delle configurazioni attive. Questi dati mi avrebbero permesso, se necessario, di valutare eventuali movimenti laterali verso altre macchine all'interno della stessa rete o di stabilire un accesso persistente.

Immagine che contiene testo, schermata

Descrizione generata automaticamente

Questa procedura ha mostrato chiaramente come una configurazione di default non sicura su un servizio Java RMI possa aprire la porta a un attaccante per ottenere accesso completo al sistema.

Note:   
  
L’HTTP Delay a 10 secondi garantisce che il payload venga inviato solo quando il server target è pronto, evitando errori di trasmissione.

Il servizio java\_rmi\_server, che sta per Java Remote Method Invocation (RMI), permette a un’applicazione di eseguire comandi su un'altra macchina in modo remoto. In pratica, consente di invocare metodi su un altro sistema attraverso la rete. Questo è utile per far interagire due o più macchine, ma in termini di sicurezza può diventare un rischio se non è configurato correttamente.