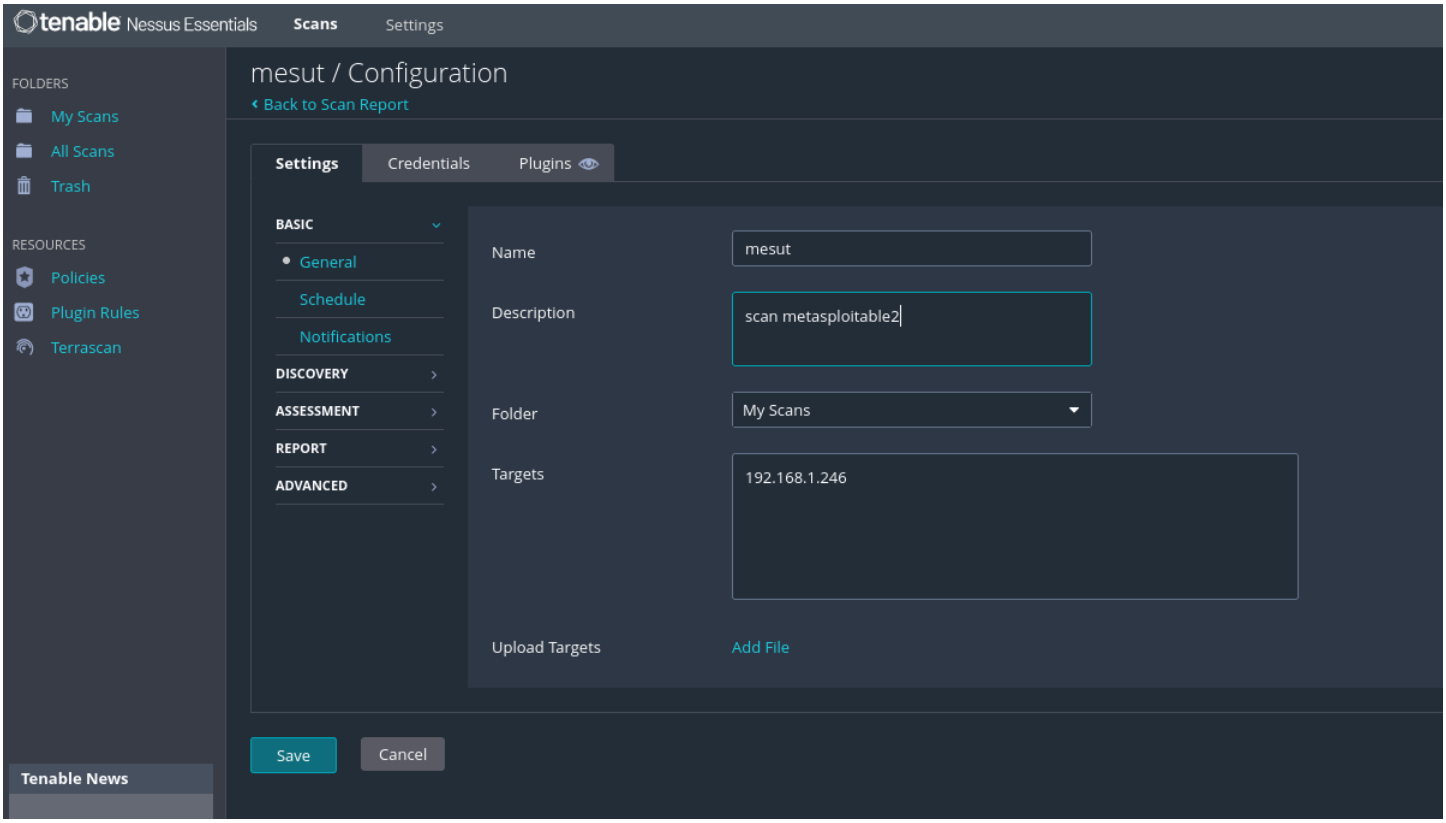
Mattia Montis

Vulnerability Scanning

Relazione sull'Uso di Nessus per l'Analisi delle Vulnerabilità di Metasploitable

1. Introduzione Nell'esercizio, ho configurato e lanciato una scansione di vulnerabilità su un target di rete specifico (192.168.1.246) utilizzando Nessus Essentials. L'obiettivo era identificare eventuali vulnerabilità presenti e classificarle per livello di rischio, per poi analizzare le vulnerabilità critiche e suggerire possibili soluzioni.

2. Configurazione della Scansione Per iniziare, ho creato una nuova scansione all'interno della dashboard di Nessus, nominata "mesut", come visibile nello screenshot di configurazione. Ho fornito una descrizione per identificarla ("scan metasploitable2") e ho impostato il target specifico. L'analisi è stata eseguita con una scansione di rete di base (Basic Network Scan), tramite un "Local Scanner".

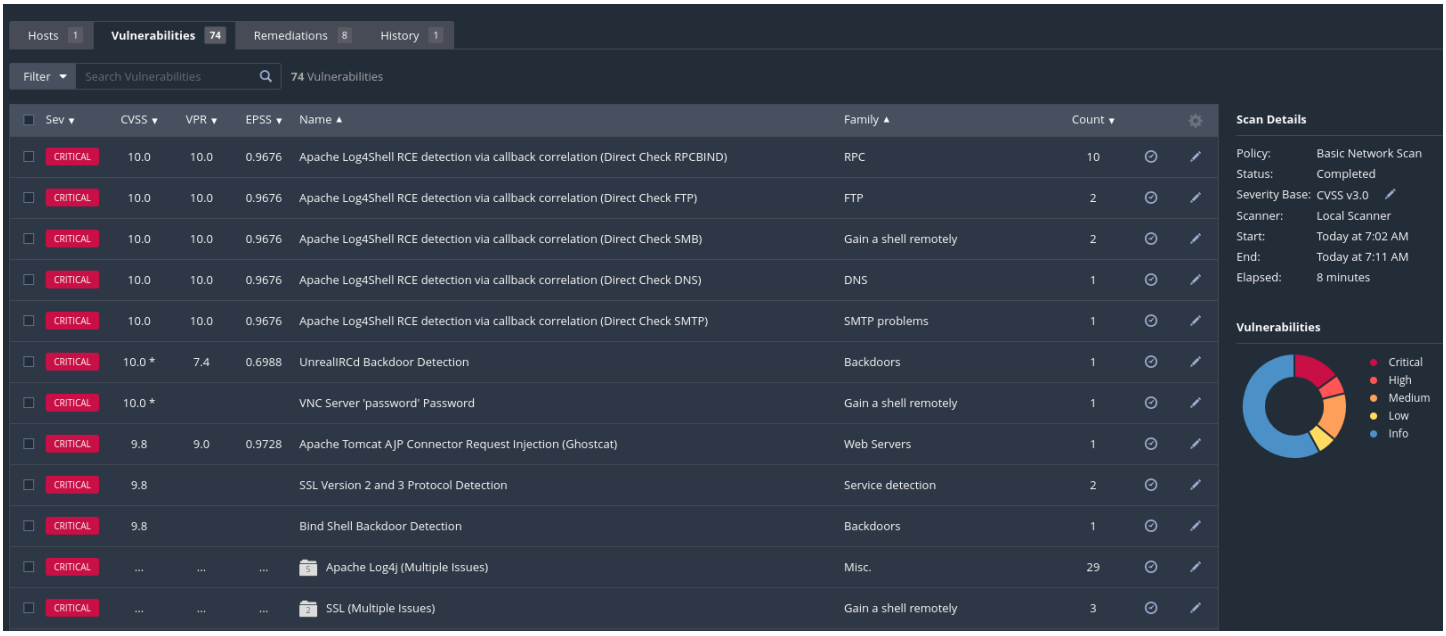


3. Risultati della Scansione Al termine della scansione, Nessus ha rilevato un totale di 74 vulnerabilità, classificate come segue:

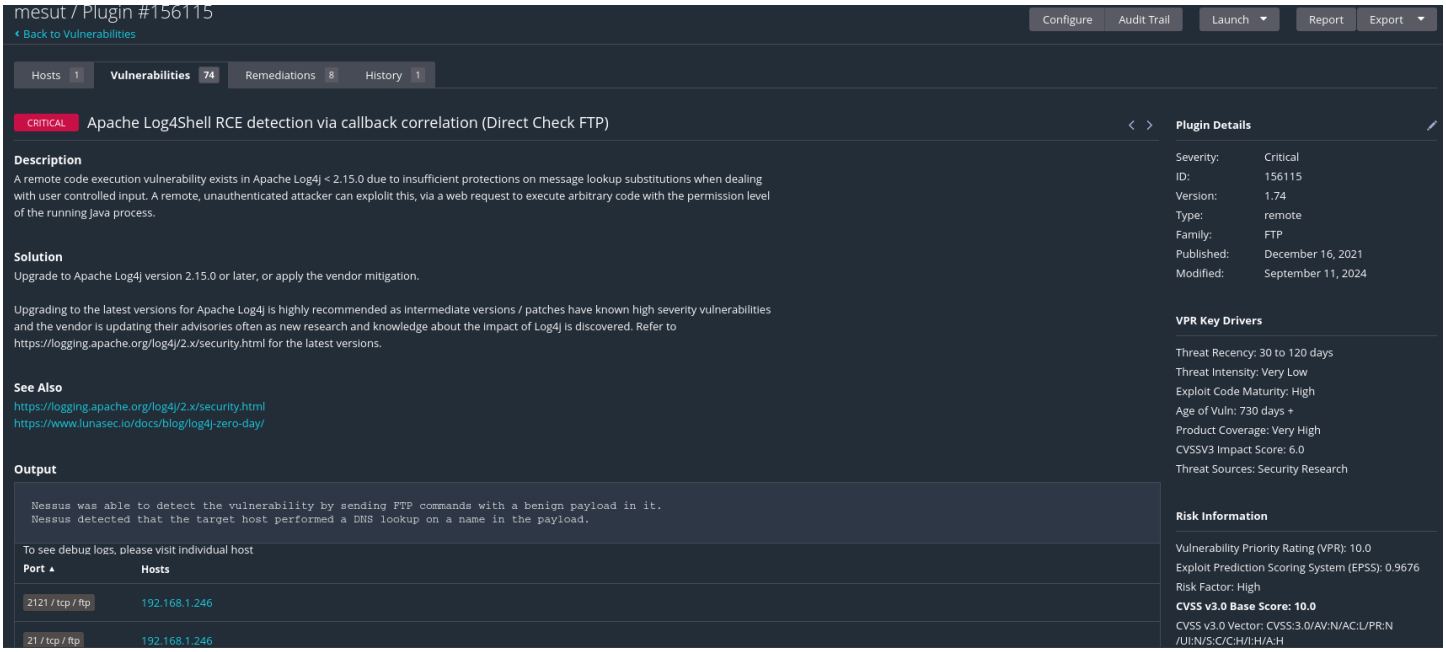
- **Critiche**: 56
- **Alte**: 8
- **Medie**: 25
- **Basse**: 9
- **Informative**: 126

Questa distribuzione di vulnerabilità suggerisce un livello di rischio elevato, con diverse criticità che richiedono un'attenzione immediata.

4.Tabella criticità Nel mentre o dopo la fine della scansione è possibile consultare la tabella delle criticità, nella quale c'è un elenco dettagliato e ordinato in maniera decrescente di rischio.



Entrando dentro una di esse è possibile leggere un report con eventuale descrizione e soluzione per la criticità.



5. Tipi di report In alto a destra dell'ultimo screen è possibile notare il 'button' repot, cliccanto su di esso si potrà scegliere che tipo di report avere, come una lista delle vulnerabilità, una lista dettagliata di quest'ultima.

Nel caso specifico si ha una lista completa delle vulnerabilità.

192.168.1.246

19	7	19	8	72
CRITICAL	HIGH	MEDIUM	LOW	INFO

Severity	CVSS v3.0	VPR Score	EPSS Score	Plugin	Name
CRITICAL	9.8	9.0	0.9728	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	-	-	51988	Bind Shell Backdoor Detection
CRITICAL	9.8	-	-	20007	SSL Version 2 and 3 Protocol Detection
CRITICAL	9.0	8.1	0.973	156164	Apache Log4Shell CVE-2021-45046 Bypass Remote Code Execution
CRITICAL	10.0	10.0	0.9676	156056	Apache Log4Shell RCE detection via Raw Socket Logging (Direct Check)
CRITICAL	10.0	10.0	0.9676	156257	Apache Log4Shell RCE detection via callback correlation (Direct Check DNS)
CRITICAL	10.0	10.0	0.9676	156115	Apache Log4Shell RCE detection via callback correlation (Direct Check FTP)
CRITICAL	10.0	10.0	0.9676	156014	Apache Log4Shell RCE detection via callback correlation (Direct Check HTTP)
CRITICAL	10.0	10.0	0.9676	156669	Apache Log4Shell RCE detection via callback correlation (Direct Check MSRPC)
CRITICAL	10.0	10.0	0.9676	156197	Apache Log4Shell RCE detection via callback correlation (Direct Check NetBIOS)
CRITICAL	10.0	10.0	0.9676	156559	Apache Log4Shell RCE detection via callback correlation (Direct Check RPCBIND)
CRITICAL	10.0	10.0	0.9676	156232	Apache Log4Shell RCE detection via callback correlation (Direct Check SMB)
CRITICAL	10.0	10.0	0.9676	156132	Apache Log4Shell RCE detection via callback correlation (Direct Check SMTP)

6. Analisi delle Vulnerabilità Critiche Per le vulnerabilità critiche identificate, ho approfondito la descrizione e le informazioni fornite nel report di Nessus e attraverso risorse aggiuntive online. Ecco un'analisi dettagliata di cinque di queste vulnerabilità critiche:

Punto 1:

Vulnerabilità Critica: Apache Log4Shell RCE via FTP

Descrizione: La libreria Apache Log4j sul server è vulnerabile a Log4Shell, permettendo a un attaccante remoto di eseguire codice arbitrario tramite input controllato dall'utente. Questo può compromettere il sistema, eseguendo codice con i permessi del processo Java.

Soluzione: Aggiornare Log4j alla versione **2.15.0** o superiore. In alternativa, applicare le mitigazioni temporanee fornite da Apache.

Punto 2:

Vulnerabilità Critica: Apache Log4Shell RCE via DNS

Descrizione: Questa versione di Apache Log4j è vulnerabile a Log4Shell, permettendo l'esecuzione di codice remoto tramite richieste DNS manipolate da un attaccante non autenticato, sfruttando input controllati dall'utente.

Soluzione: Aggiornare Apache Log4j alla versione **2.15.0** o successiva, o applicare le mitigazioni consigliate dal fornitore.

Punto 3:

Vulnerabilità Critica: Apache Log4Shell RCE via SMTP

Descrizione: La versione di Apache Log4j sul server permette l'esecuzione di codice remoto tramite richieste SMTP manipolate da un attaccante non autenticato. Questa vulnerabilità sfrutta input controllati dall'utente per eseguire codice arbitrario con i permessi del processo Java.

Soluzione: Aggiornare Apache Log4j alla versione **2.15.0** o successiva, o applicare le mitigazioni temporanee suggerite dal fornitore.

Punto 4:

Vulnerabilità Critica: UnrealIRCd Backdoor Detection

Descrizione: Alcune versioni di UnrealIRCd, in particolare la **versione 3.2.8.1**, contengono una backdoor che permette a un attaccante di eseguire codice arbitrario sul server IRC, compromettendo la sicurezza e consentendo accesso non autorizzato.

Soluzione:

- **Ridistribuire il software** da una fonte affidabile.
- **Verificare l'integrità** usando i checksum MD5 o SHA1 forniti ufficialmente.
- **Reinstallare** il software verificato per eliminare la backdoor e garantire la sicurezza del sistema.

I checksum MD5 e SHA1 sono stringhe di caratteri che vengono generate applicando specifici algoritmi di hashing (MD5 e SHA-1) a un file o a un insieme di dati.

Utilità dei Checksum MD5 e SHA1

1. **Verifica dell'Integrità**: Quando scarichi un file da internet (es. software, aggiornamenti), il sito web spesso fornisce anche i checksum MD5 o SHA1. Dopo il download, puoi generare il checksum del file e confrontarlo con quello pubblicato. Se combacia, significa che il file non è stato alterato.
2. **Rilevamento di Modifiche o Manomissioni**: Anche una minima modifica del file (anche un singolo bit cambiato) cambia il checksum in modo significativo. Questo aiuta a verificare che il file sia autentico e non sia stato corrotto o manomesso.

Purtroppo sono algoritmi di criptazione insicuri quindi sarebbe meglio affidarsi ad algoritmi più moderni: **SHA-256.**

Punto 5:

Vulnerabilità Critica: Samba Badlock Vulnerability

La **Samba Badlock Vulnerability** (CVE-2016-2118) è una vulnerabilità critica che colpisce le versioni di Samba 3.x e 4.x prima di 4.2.11, 4.3.x prima di 4.3.8, e 4.4.x prima di 4.4.21. Questa vulnerabilità riguarda il modo in cui Samba gestisce le connessioni, permettendo a un attaccante man-in-the-middle di eseguire attacchi di downgrade del protocollo e impersonare utenti modificando il flusso di dati client-server1.

Soluzioni:

1. **Aggiornare Samba**: Assicurarsi di aggiornare Samba alla versione 4.2.11, 4.3.8, o 4.4.2 o successiva2.
2. **Verificare le Patch**: Applicare tutte le patch di sicurezza rilasciate dal fornitore3.
3. **Monitoraggio Continuo**: Utilizzare strumenti di sicurezza per monitorare continuamente la rete e rilevare eventuali attività sospette.

CONCLUSIONI

L'analisi delle vulnerabilità critiche evidenzia la necessità di un'attenzione costante nella gestione della sicurezza informatica. Vulnerabilità come Log4Shell, che colpisce la libreria Apache Log4j attraverso diversi canali (FTP, DNS, SMTP), dimostrano quanto possa essere pericoloso l'accesso remoto non autorizzato. È fondamentale aggiornare le librerie e le applicazioni alla versione più recente o applicare mitigazioni temporanee suggerite dai fornitori.

La presenza di backdoor, come quella in UnrealIRCd, sottolinea l'importanza di scaricare software da fonti affidabili e di verificare l'integrità dei file attraverso checksum. Infine, la vulnerabilità di Samba evidenzia i rischi associati alla gestione delle connessioni, rendendo essenziale l'implementazione di aggiornamenti regolari e un monitoraggio attivo della rete. Investire nella sicurezza informatica non è solo una misura reattiva, ma un approccio proattivo di protezione.