

S5L3

Obiettivo: Esercizio Traccia Lo studente effettuerà un Vulnerability Scanning sulla macchina Metasploitable utilizzando Nessus, concentrandosi sulle porte comuni. Questo esercizio ha lo scopo di fare pratica con lo strumento Nessus, la configurazione delle scansioni, e di familiarizzare con alcune delle vulnerabilità note.

Step1. Configurazione della scansione

Come primo step sono andato su Nessus basic network Scan per poi impostare il target che in questo caso è la mia Metasploitable con ip 192.168.64.3.

The screenshot shows the 'New Scan / Basic Network Scan' configuration window in Nessus. The interface is dark-themed. On the left, there is a sidebar with a 'Settings' tab selected, and sub-tabs for 'Credentials' and 'Plugins'. The 'Settings' tab is further divided into 'BASIC', 'DISCOVERY', 'ASSESSMENT', 'REPORT', and 'ADVANCED'. The 'BASIC' section is expanded, showing 'General', 'Schedule', and 'Notifications'. The 'General' sub-tab is selected, displaying fields for 'Name' (L5S3 Metaspotable), 'Description' (analisi delle porte comuni), 'Folder' (Epicode), and 'Targets' (192.168.64.3). At the bottom of the 'Targets' field, there is an 'Upload Targets' button and an 'Add File' link. At the very bottom of the window, there are 'Save' and 'Cancel' buttons.

Nella finestra Discovery ho impostato le porte che mi interessavano ovvero quelle comuni.

The screenshot shows the 'Scan Type' configuration window in Nessus. The 'Scan Type' dropdown menu is set to 'Port scan (common ports)'. Below this, there are three sections of settings: 'General Settings' (Always test the local Nessus host, Use fast network discovery), 'Port Scanner Settings' (Scan common ports, Use netstat if credentials are provided, Use SYN scanner if necessary), and 'Ping hosts using:' (TCP, ARP, ICMP (2 retries)).

Step2. Esecuzione della scansione.

Una volta configurata la scansione e premendo su salva mi appare nel reparto scansioni questa barra:

<input type="checkbox"/>	Name	Scan Type	Schedule	Last Scanned ▼	
<input type="checkbox"/>	Epicode L5S3 Metaspotable	Vulnerability	On Demand	Today at 1:03 PM	

A questo punto ho la possibilità di avviare la scansione e il programma mi segnalerà il tempo di attesa e le vulnerabilità che troverà in tempo reale.

Step3. Analisi del report

Una volta completata la scansione mi apparirà una una barra tipo questa:

Hosts1

Vulnerabilities81

Remediations8

History1

Filter

Search Hosts

1 Host

<input type="checkbox"/>	Host	Auth	Vulnerabilities
<input type="checkbox"/>	192.168.64.3	Fail	<div><div>65</div><div>9</div><div>27</div><div>9</div></div> <div>135</div> <div></div>

Cliccandoci sopra avrò la possibilità di analizzare le varie vulnerabilità
Prenderò come esempio una delle più gravi ovvero:

<input type="checkbox"/>	Sev ▼	CVSS ▼	VPR ▼	EPSS ▼	Family ▲	Count ▼		
<input type="checkbox"/>	CRITICAL	10.0	10.0	0.9436	RPC	10		

Step4. Possibile risoluzione del problema

Cliccando sopra ad una delle vulnerabilità segnalate mi darà tutte le informazioni al riguardo.

Vulnerabilities 81

CRITICAL Apache Log4Shell RCE detection via callback correlation (Direct Che... < >

Description

A remote code execution vulnerability exists in Apache Log4j < 2.15.0 due to insufficient protections on message lookup substitutions when dealing with user controlled input. A remote, unauthenticated attacker can exploit this, via a web request to execute arbitrary code with the permission level of the running Java process.

Solution

Upgrade to Apache Log4j version 2.15.0 or later, or apply the vendor mitigation.

Upgrading to the latest versions for Apache Log4j is highly recommended as intermediate versions / patches have known high severity vulnerabilities and the vendor is updating their advisories often as new research and knowledge about the impact of Log4j is discovered. Refer to <https://logging.apache.org/log4j/2.x/security.html> for the latest versions.

See Also

<https://logging.apache.org/log4j/2.x/security.html>
<https://www.lunasec.io/docs/blog/log4j-zero-day/>

In questo caso posso vedere che questa vulnerabilità viene segnalata come “Log4Shell RCE”.

Il programma mi dà anche una descrizione del problema dicendomi che una vulnerabilità di esecuzione di codice da remoto (RCE) è presente in Apache Log4j versione inferiore a 2.15.0.

L'attacco sfrutta input controllati dall'utente che vengono usati nelle lookup di messaggi (JNDIlookup), permettendo a un attaccante remoto (senza autenticazione) di eseguire comandi arbitrari con i permessi del processo Java.

Pericolosità:

Impatto: L'attaccante può eseguire comandi come se fosse il servizio Java stesso, quindi potenzialmente ottenere controllo completo del sistema.

Diffusione: È stato uno degli zero-day più devastanti degli ultimi anni, usato in attacchi reali su larga scala.

Oltre alla descrizione mi da anche delle soluzioni possibili ovvero:

1. Aggiornare la versione del Apache
2. Applicare la mitigazione fornita dal fornitore

Nessus ci da anche dei siti per consultare il problema e trovare la soluzione come questo:

The Logging Services Security Team takes security seriously. This allows our users to place their trust in Log4j for protecting their mission-critical data. In this page we will help you find guidance on security-related issues and access to known vulnerabilities.

WARNING

Log4j 1 has [reached End of Life](#) in 2015, and is no longer supported. Vulnerabilities reported after August 2015 against Log4j 1 are not checked and will not be fixed. Users should [upgrade to Log4j 2](#) to obtain security fixes.

Getting support

If you need help on building or configuring Logging Services projects or other help on following the instructions to mitigate the known vulnerabilities listed here, please use our [user support channels](#).

TIP

If you need to apply a source code patch, use the building instructions for the project version that you are using. These instructions can be found in [BUILDING.adoc](#), [BUILDING.md](#), etc. files distributed with the sources.

Reporting vulnerabilities

If you have encountered an unlisted security vulnerability or other unexpected behaviour that has a security impact, or if the descriptions here are incomplete, please report them **privately** to [the Logging Services Security Team](#).

IMPORTANT

We urge you to **carefully read the threat model** detailed in following sections before submitting a report. It guides users on certain safety instructions while using Logging Services software and elaborates on what counts as an unexpected behaviour that has a security impact.

Common threat model

Below we share the threat model shared by all Logging Services projects.

Code signing

All Logging Services software release distributions are signed using GPG using a key from the Logging Services PMC [KEYS file](#). Information on how to verify releases is provided further in the [Developer page](#). These GPG signatures should be validated in your build process.

Una delle possibili soluzioni che ho trovato ricercando sul web è quella di

1. Verifica quale applicazione sulla tua Metasploitable usa Log4j
2. Andare a cambiare il `.jar` Log4j nella directory `/lib` del progetto.
3. Usare un container Docker sicuro o una VM aggiornata per sostituire quella vulnerabile.

Extra: Abbiamo anche la possibilità di creare un report per agevolare e facilitare future analisi.

Basta andare su Report in alto a destra e potremmo selezionare il tipo di report che vogliamo (consigliato Detailed Vulnerabilities by host) e il formato desiderato (HTML, PDF, CSV)

Generate Report

Report Format: ☐ HTML ☒ PDF ☐ CSV

Select a Report Template:

SYSTEM

Complete List of Vulnerabilities by Host

Detailed Vulnerabilities By Host

Detailed Vulnerabilities By Plugin

Vulnerability Operations

Generate Report

Cancel

Conclusioni:

Nessus è un programma che permette di individuare, tramite scansioni sul target richiesto , le varie vulnerabilità spiegando come poterle risolvere e segnalandole in base alla loro pericolosità. Estremamente utile per capire se una rete o dispositivo è al sicuro da attacchi esterni.