Nessus Vulnerability Assessment Report – Kali Linux Virtual Machine

1. Executive Summary

In this assessment, I performed a credentialed vulnerability scan on a Kali Linux virtual machine using Tenable Nessus Essentials. By providing SSH credentials, Nessus was able to perform deeper checks on installed packages and services, which allowed me to identify several critical, high, and medium vulnerabilities.

The goal of this assessment was to simulate a real internal vulnerability review, understand how different weaknesses affect a system, and document practical remediation steps that improve the overall security posture of the VM.

2. Environment Details

- **Operating System:** Kali Linux (Kernel 6.12.25-amd64)
- **Scanner:** Tenable Nessus Essentials v10.8.3
- **Scan Type:** Credentialed SSH Scan
- **Target IP Address:** 10.0.2.15
- **Lab Setup:** Local VM running in a home lab environment

This setup allowed me to safely perform security testing without affecting production systems.

3. Assessment Objectives

During this assessment, I focused on the following:

- Identifying vulnerabilities affecting installed software on the VM
- Understanding the severity of each issue using CVSS scores and plugin details
- Reviewing exposure around Node.js, PostgreSQL, Tornado, SSL, and other Linux services
- Recommending specific updates or configuration changes to reduce risk
- Building practical experience with credentialed scans and vulnerability management workflows

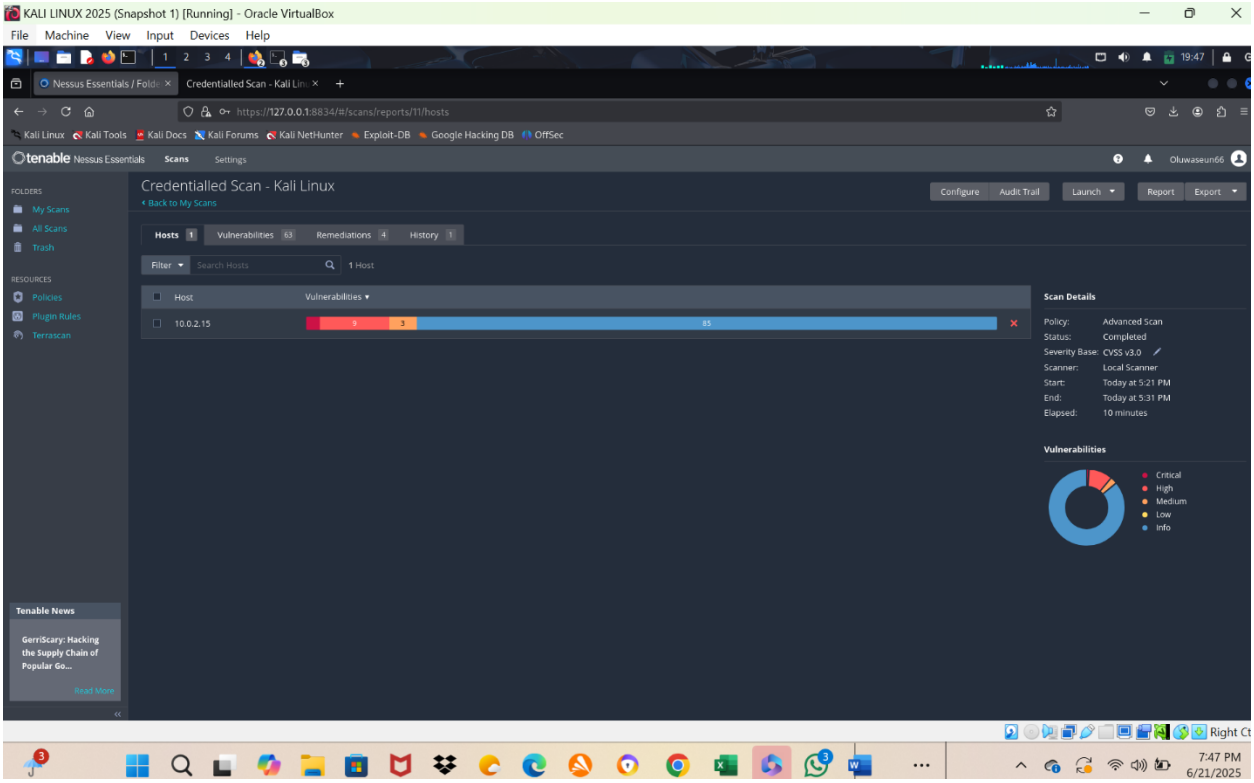
4. Scan Overview

The credentialed scan completed successfully and returned a mix of **critical**, **high**, and **medium** vulnerabilities. The most severe issues were related to:

- Outdated Node.js components
- PostgreSQL security weaknesses
- Tornado web framework vulnerabilities
- Weak SSL configuration using self-signed certificates

These findings gave me a clear view of how outdated dependencies and misconfigurations can increase system exposure.

Overall Scan Dashboard



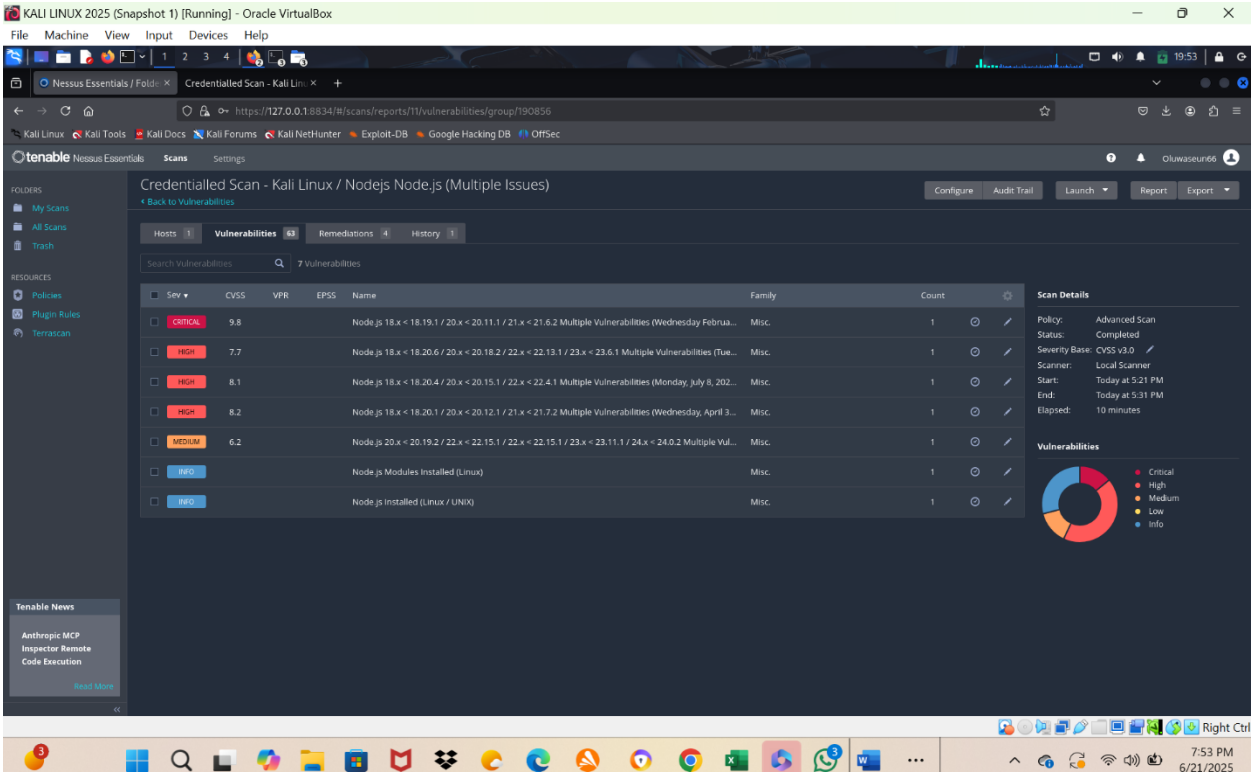
Screenshots/nessus-dashboard.png

5. Key Findings

5.1 Node.js Vulnerabilities

The scan flagged two major CVEs affecting the installed Node.js version:

- **CVE-2024-21892 (Critical):** A privilege escalation flaw that could let an attacker gain elevated access and run arbitrary commands.
- **CVE-2024-27980 (High):** A command injection vulnerability that could lead to remote code execution.



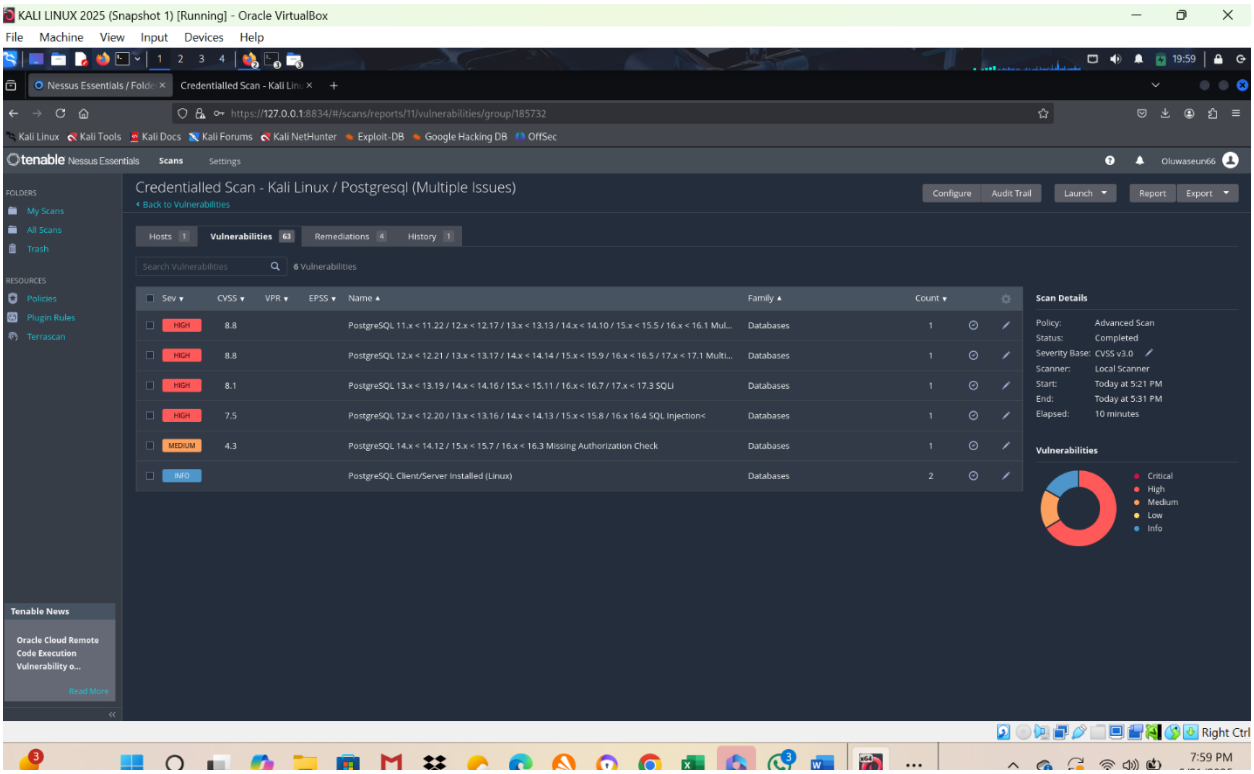
Screenshots/critical-nodejs-vuln.png

These issues highlight how quickly Node.js versions become outdated and how important it is to keep development runtimes patched.

5.2 PostgreSQL Vulnerabilities

Two PostgreSQL-related issues were also identified:

- **CVE-2023-5869 (High):** A memory overflow problem that could lead to service crashes or unexpected behavior.
- **CVE-2024-4317 (Medium):** An information disclosure issue tied to extended statistics.

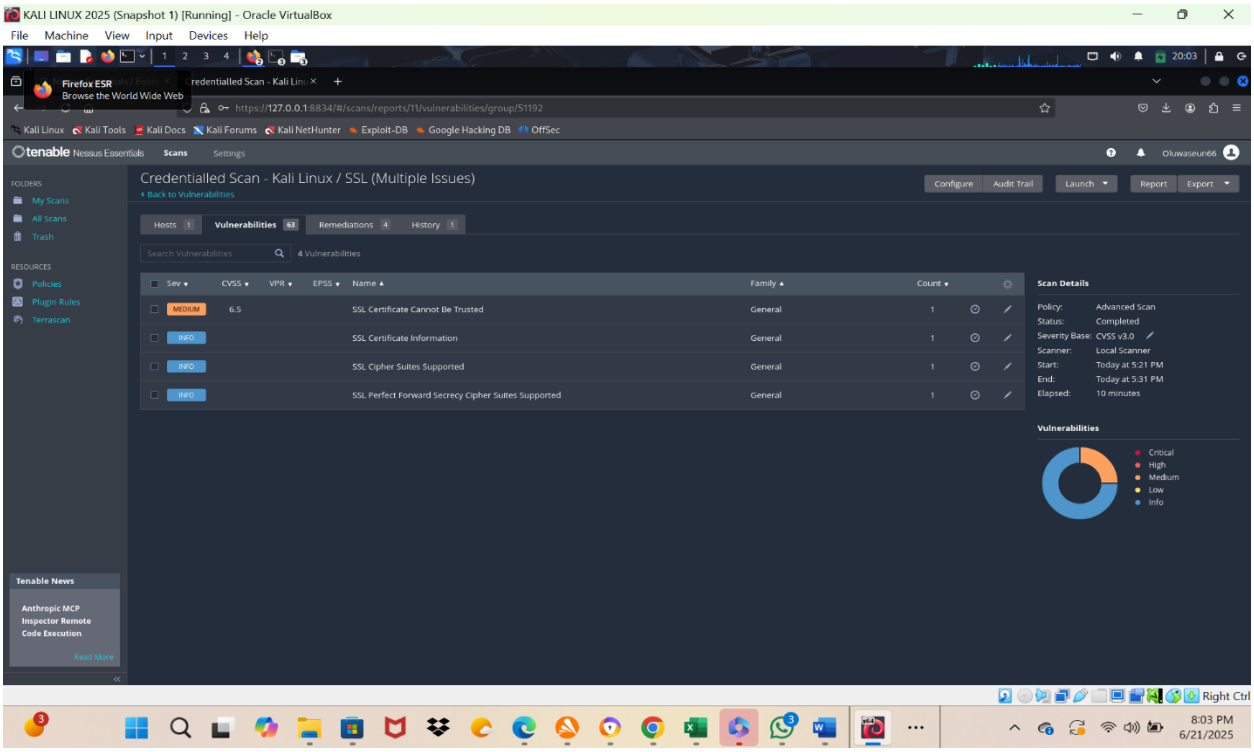


Screenshots/postgres-high-vuln.png

These findings show how database components can quietly introduce risks if not patched regularly.

5.3 SSL and Certificate Findings

Nessus also detected that the system was using **self-signed SSL certificates**. While expected in lab environments, this is still a weakness because SSL clients cannot verify the authenticity of the server.

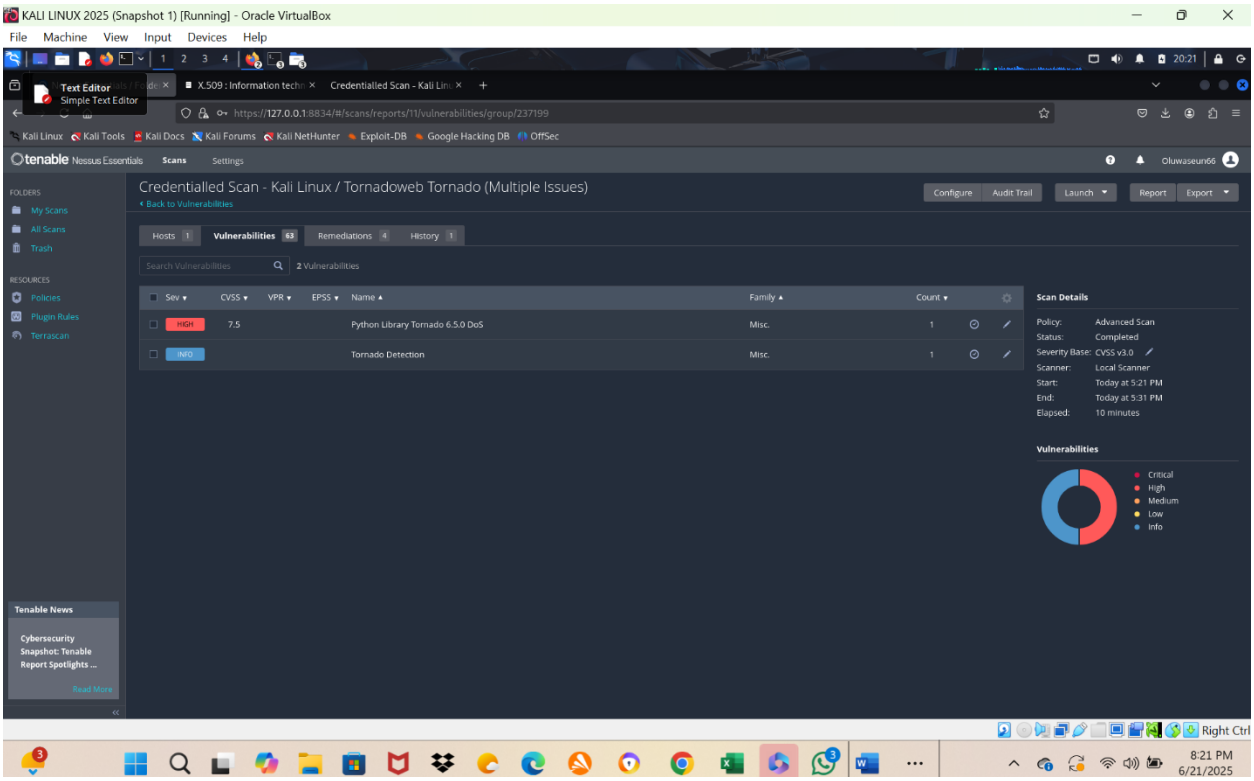


Screenshots/ssl-self-signed.png

If this were a production system, this would be a high-priority issue because attackers could abuse it for man-in-the-middle attacks.

5.4 Tornado Web Framework

The installed version of the Tornado framework was outdated and linked to several known vulnerabilities involving request handling and asynchronous operations. Although the system is not running a large application, it still creates potential exposure.



Screenshots/ tornado-vuln.png

6. Recommended Remediation Actions

To harden the VM and eliminate the identified weaknesses, I recommend the following steps:

Node.js

- Upgrade Node.js to v20.19.2 or later
- Review all dependencies for compatibility after upgrading

PostgreSQL

- Upgrade to version 16.7 or later
- Restart services and check application/database connections

Tornado Framework

- Update Tornado to v6.5.0
- Test any related applications to confirm they still run properly

SSL/TLS

- Replace self-signed certificates with trusted CA-signed certificates
- Strengthen TLS settings by disabling weak ciphers and enforcing modern protocol versions

Applying these fixes would significantly reduce the attack surface of the Kali Linux machine.

7. Conclusion

This vulnerability assessment helped me understand how credentialed scanning works and how it reveals deeper system issues compared to non-credentialed scans. I was able to identify meaningful weaknesses, analyze CVE details, and propose realistic remediation steps.

This project strengthened my hands-on experience with Nessus, CVSS scoring, Linux system hardening, and vulnerability reporting, skills that directly support security analyst, SOC, and GRC work.

8. Supporting Files

- Full screenshot set available in: Screenshots/
- PDF version of this report : Files/Nessus-Vulnerability-Report.pdf