Day 3 - Task 3

Objective: Use free tools to identify common vulnerabilities on your computer.
Tools: OpenVAS Community Edition (free vulnerability scanner) or Nessus Essentials.
Deliverables: Vulnerability scan report with identified issues

Vulnerability Assessment Report

Tool Used: Tenable Nessus Essentials
Scan Name: `basic_scan`
Date of Scan: 08-Aug-2025
Scan Target: `10.0.2.15` (Local Machine)
Policy: Basic Network Scan
Scanner: Basic Vulnerability Scan
Duration: 19 minutes

---

1. Objective

The objective of this assessment was to use a free vulnerability scanner to identify potential security weaknesses on the local machine, evaluate their severity, and document findings for remediation.

---

2. Methodology

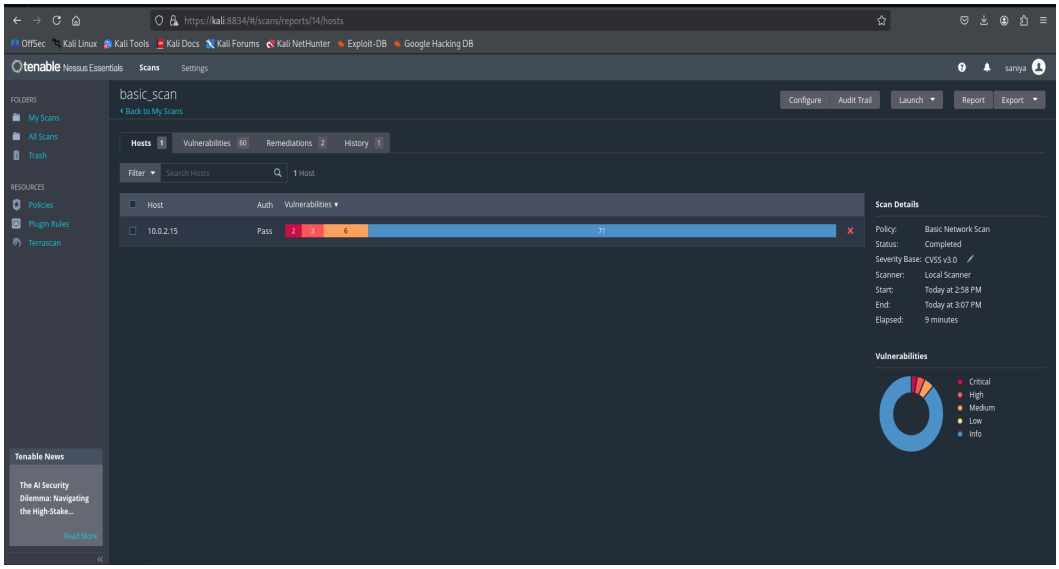
1. Installed **Nessus Essentials** on Kali Linux.
2. Set up the scan target as the local machine IP (`10.0.2.15`).
3. Selected **Basic Network Scan** template.
4. Launched the scan and allowed it to complete.
5. Exported the scan results in PDF format and reviewed them.

---

3. Summary of Findings

The scan identified a total of **75 vulnerabilities**:

- **Critical:** 2
- **High:** 3
- **Medium:** 6
- **Low:** 0
- **Informational:** 64

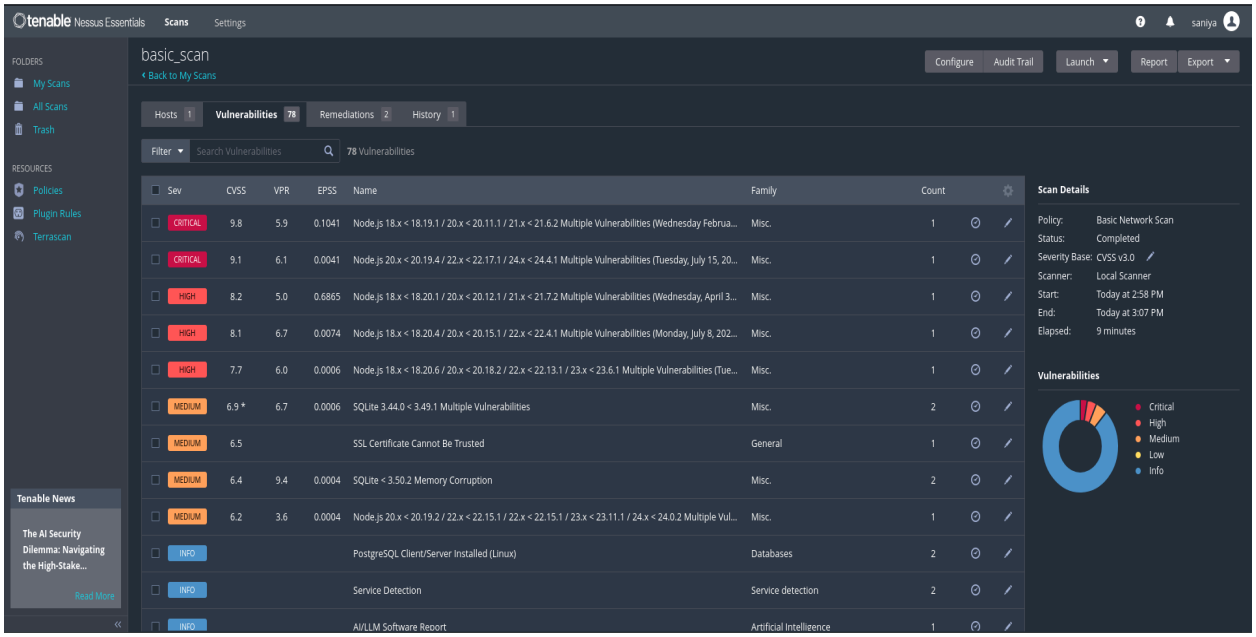


---

4. Top Critical & High Vulnerabilities

Severity	CVSS v3.0 Score	Vulnerability	Description
Critical	9.8	Node.js < 18.19.1 / 20.11.1 / 21.6.2 Multiple Vulnerabilities	Multiple security flaws in Node.js that may allow remote code execution or denial of service.
Critical	9.1	Node.js < 20.19.4 / 22.17.1 / 24.4.1 Multiple Vulnerabilities	Potentially exploitable flaws that may compromise the system.
High	8.2	Node.js < 18.20.1 / 20.12.1 / 21.7.2 Multiple Vulnerabilities	Remote exploitation possible via crafted input.

High	8.1	Node.js < 18.20.4 / 20.15.1 / 22.4.1 Multiple Vulnerabilities	May allow attackers to bypass security controls.
High	7.7	Node.js < 18.20.6 / 20.18.2 / 22.13.1 Multiple Vulnerabilities	Weaknesses in Node.js security updates.



---

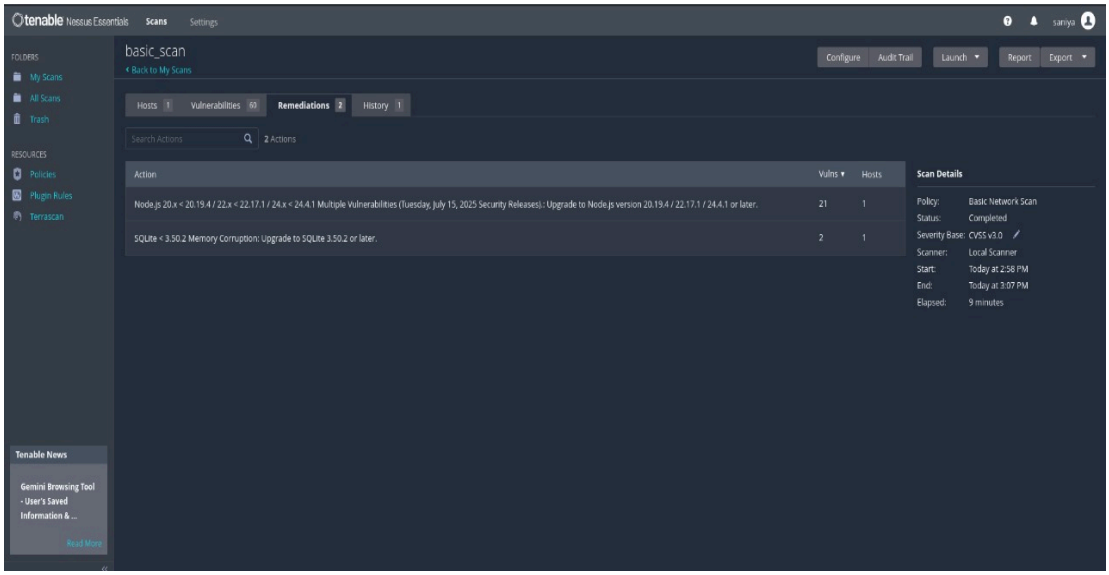
5. Other Notable Issues

- **Medium:** SSL Certificate Cannot Be Trusted.
- **Medium:** SQLite < 3.50.2 Memory Corruption vulnerability.
- Multiple informational findings such as software enumeration, service detection, and SSL/TLS configuration details.

---

6. Recommendations

- **Update Node.js** to the latest stable version to patch all listed vulnerabilities.
- **Update SQLite** to a secure version (> 3.50.2).
- Replace or reconfigure **SSL certificates** to use a trusted Certificate Authority (CA).
- Review and harden services revealed by **service detection** to minimize exposure.



---

7. Conclusion

The scan revealed multiple severe vulnerabilities, primarily due to outdated Node.js and SQLite versions. Immediate patching is required to prevent potential exploitation. Informational findings should also be reviewed to reduce unnecessary exposure of system details.