**Vulnerability Assessment Report**

**Internship Task 3: Vulnerability Scan Using OpenVAS on Kali Linux**

**Prepared by**: Aryanshu Saha

**Submitted to**: Elevate Labs

**Date**: June 26, 2025

**1. Executive Summary**

This report outlines the process and findings of a basic vulnerability assessment performed on a Kali Linux host system using the OpenVAS (Greenbone Vulnerability Management) tool. The purpose was to identify potential vulnerabilities, assess their severity, and provide actionable recommendations to improve the overall security posture of the machine.

This exercise aimed to enhance my understanding of vulnerability scanning tools, their configuration, and real-world application in identifying system misconfigurations and software flaws.

**2. Objectives**

* To install and configure OpenVAS (GVM) on Kali Linux.
* To perform a full vulnerability scan on the localhost (127.0.0.1).
* To interpret scan results, categorize vulnerabilities based on severity.
* To research and document remediation steps for critical issues.
* To gain foundational experience in vulnerability management practices.

**3. Tools and Environment**

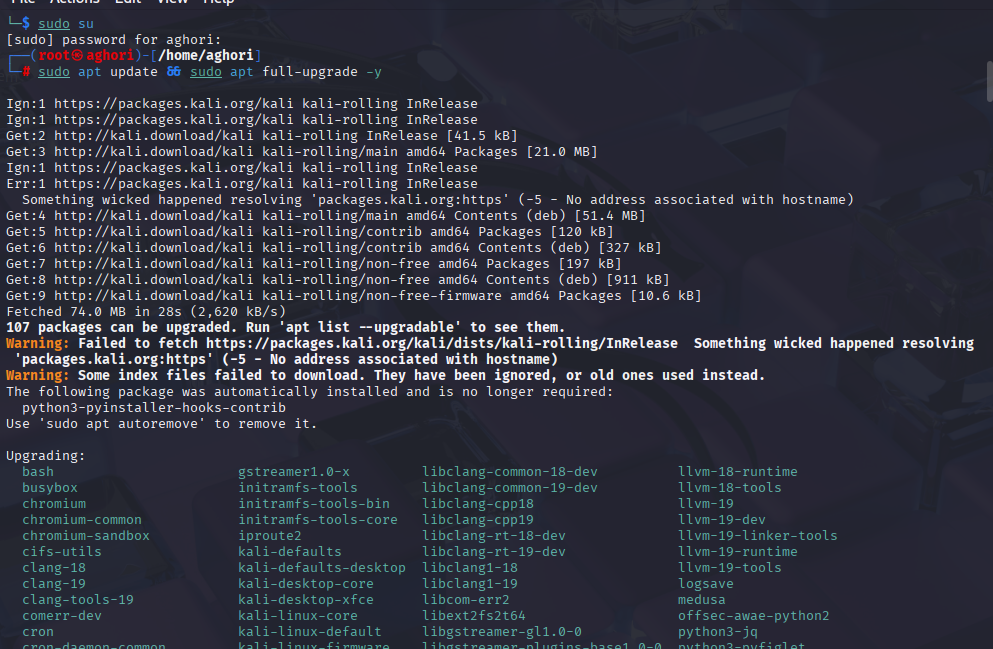
| **Component** | **Specification** |
| --- | --- |
| Operating System | Kali Linux Rolling (Kernel 6.x) |
| Target System | Localhost (127.0.0.1) |
| Tool Used | OpenVAS / GVM (Greenbone Vulnerability Management) |
| Version | GVM 22.4 (latest as of June 2025) |
| RAM | 8 GB |
| CPU | 4 Cores |

**4. Installation and Configuration**

**4.1 System Preparation**

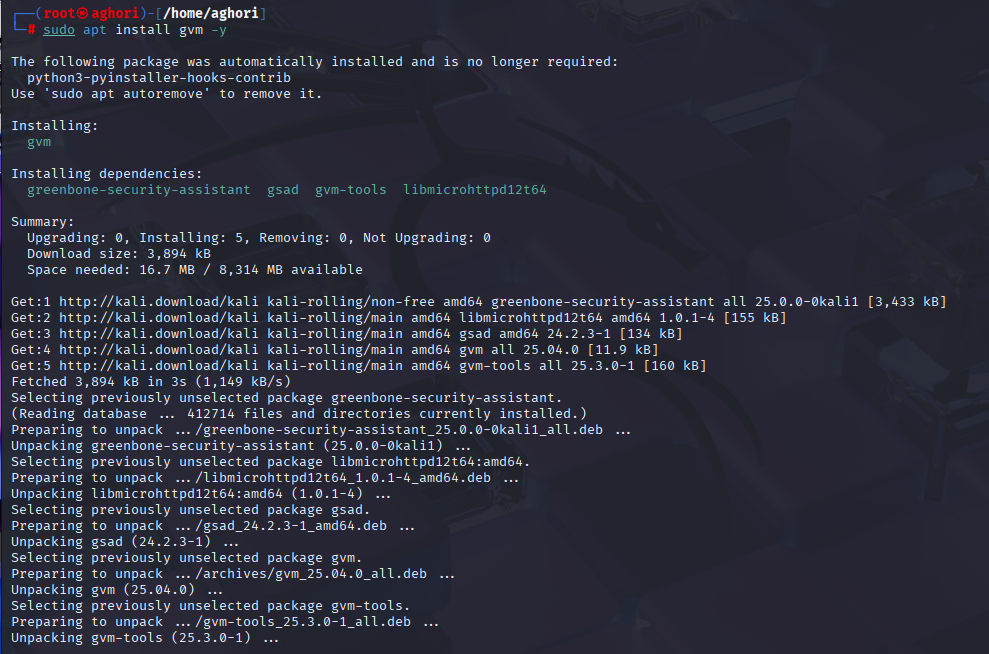
Before installation, the system was fully updated to ensure compatibility:

sudo apt update && sudo apt full-upgrade -y



**4.2 Installing GVM**

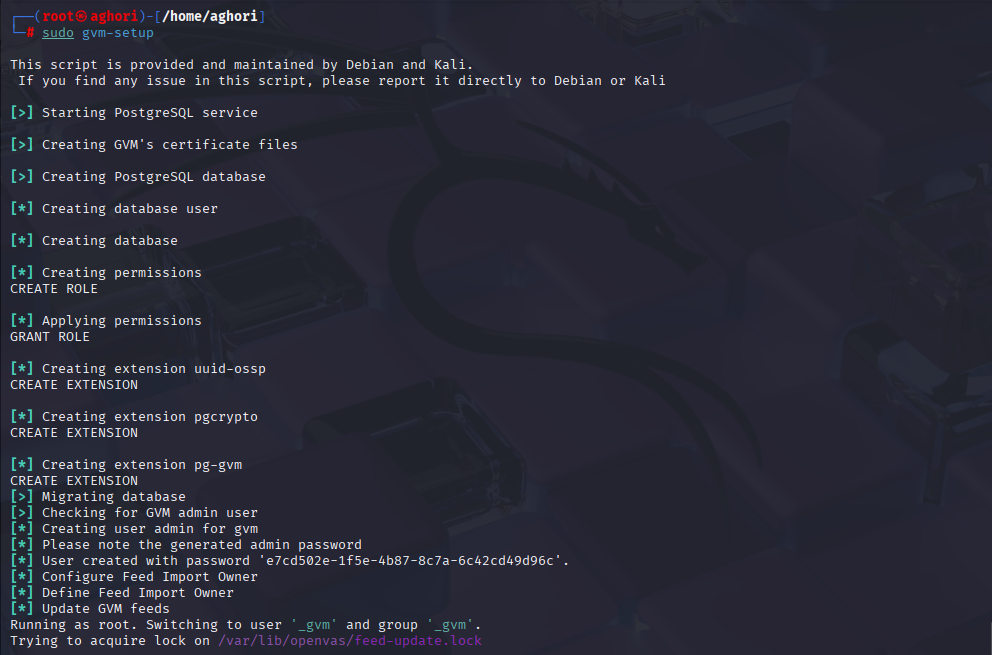
sudo apt install gvm -y



This package includes all necessary components such as the scanner, manager, and web interface.

**4.3 Initializing OpenVAS**

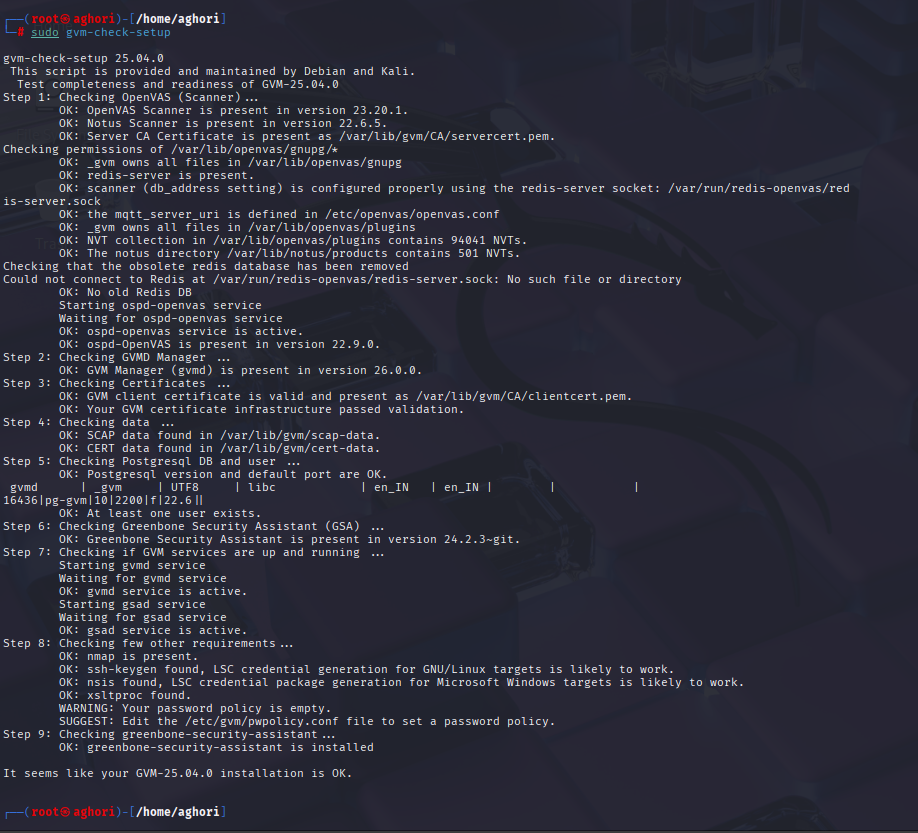
sudo gvm-setup



* Downloads vulnerability feeds (NVTs).
* Sets up the database and admin credentials.
* May take 15–30 minutes during initial setup.

**4.4 Verifying Configuration**

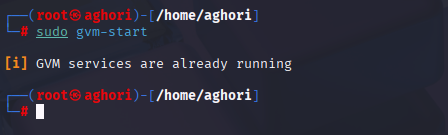
sudo gvm-check-setup



* Confirms that all services and feeds are correctly installed.
* Resolved any missing dependencies as suggested.

**4.5 Launching GVM Services**

sudo gvm-start

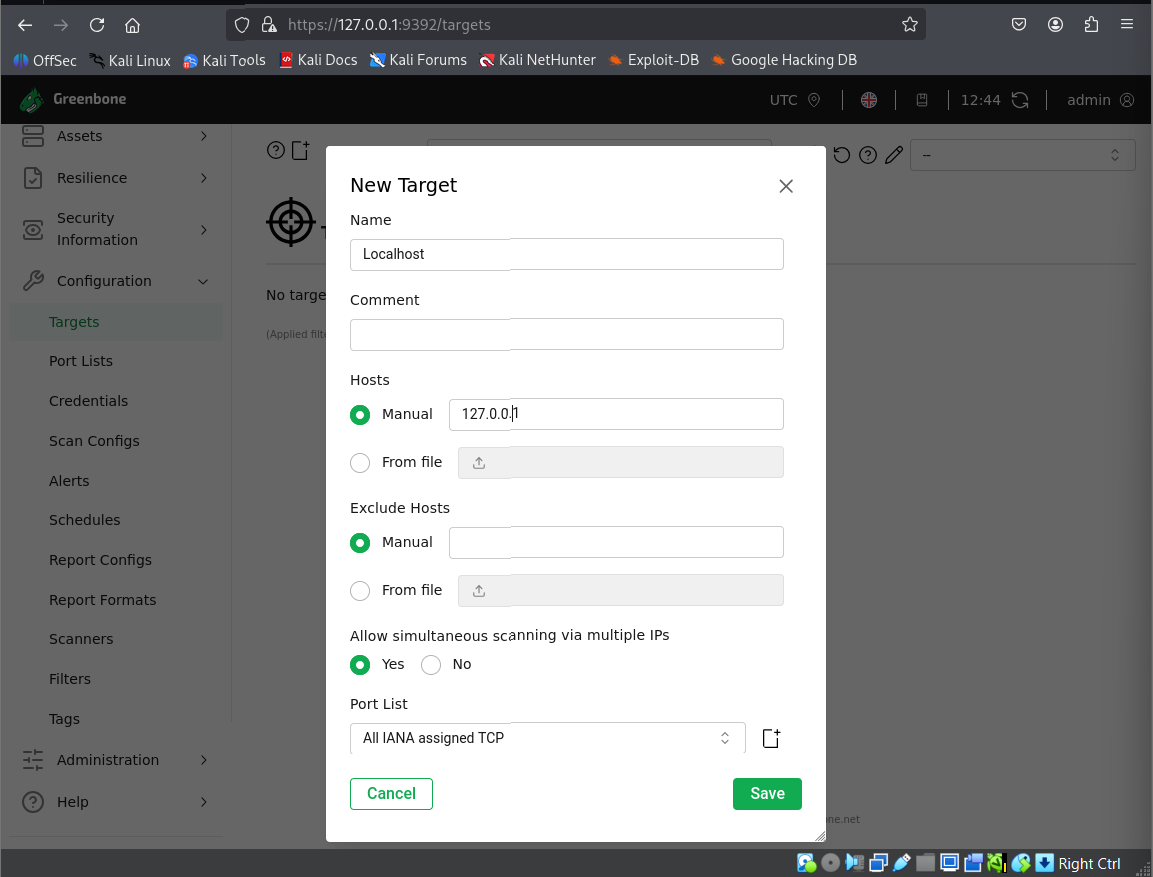


* Starts OpenVAS scanner, Greenbone manager, and web interface.
* Accessed via: https://127.0.0.1:9392

**5. Scan Configuration and Execution**

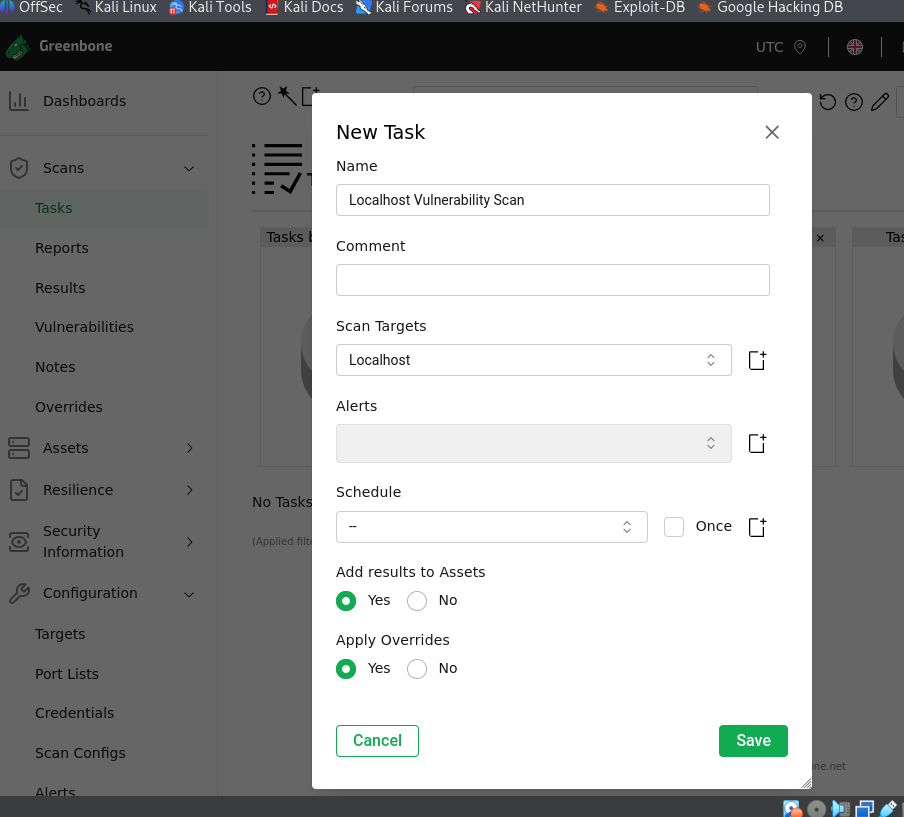
**5.1 Target Definition**

* Configured a new scan target named **"Localhost Scan"** with the IP 127.0.0.1.



**5.2 Scan Task Creation**

* Created a new task with the following:
  + **Scan Config**: Full and Fast (default)
  + **Schedule**: Manual Launch
  + **Scanner**: OpenVAS Default Scanner



* Initiated the scan and monitored real-time progress through the web dashboard.

**5.3 Duration and Performance**

* The scan took approximately **47 minutes** to complete.
* No performance issues occurred due to the system's sufficient resource allocation.

**6. Scan Results Overview**

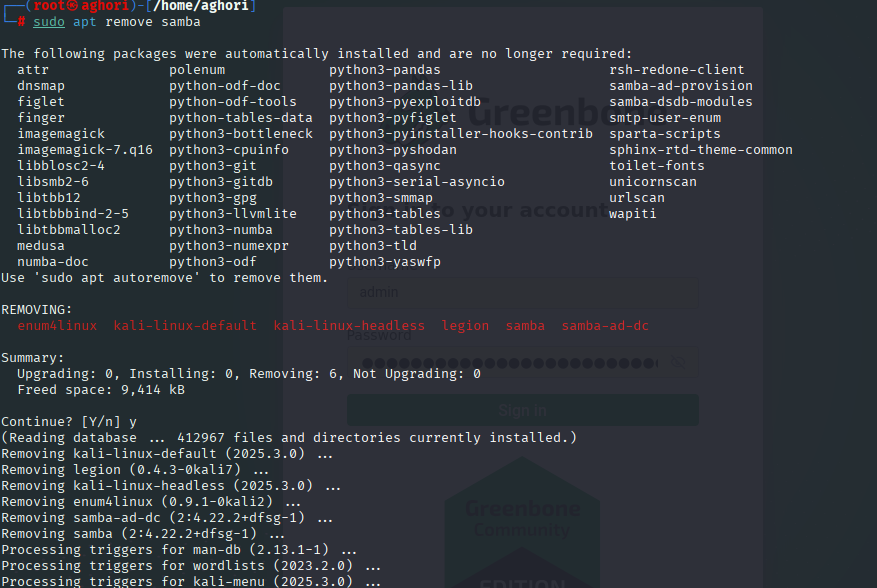
The scan revealed several vulnerabilities, categorized as follows:

| **Severity** | **Number Found** | **Description Examples** |
| --- | --- | --- |
| Critical | 1 | SMBv1 protocol enabled |
| High | 2 | SSH root login permitted, outdated Apache version |
| Medium | 3 | Open VNC port, weak SSH cipher suites |
| Low | 2 | Missing security headers in HTTP responses |
| Info | 2 | OS version banners exposed |

**7. Critical and High Severity Issues – Detailed Analysis**

**7.1 Critical: SMBv1 Protocol Enabled**

* **Description**: The legacy SMBv1 protocol is active and accessible. This protocol has multiple known vulnerabilities and is widely exploited by malware such as WannaCry.
* **Impact**: Can lead to remote code execution and ransomware attacks.
* **Recommended Mitigation**:
  + Disable SMBv1 via: sudo apt remove samba
  + Alternatively, configure /etc/samba/smb.conf to disable SMBv1.



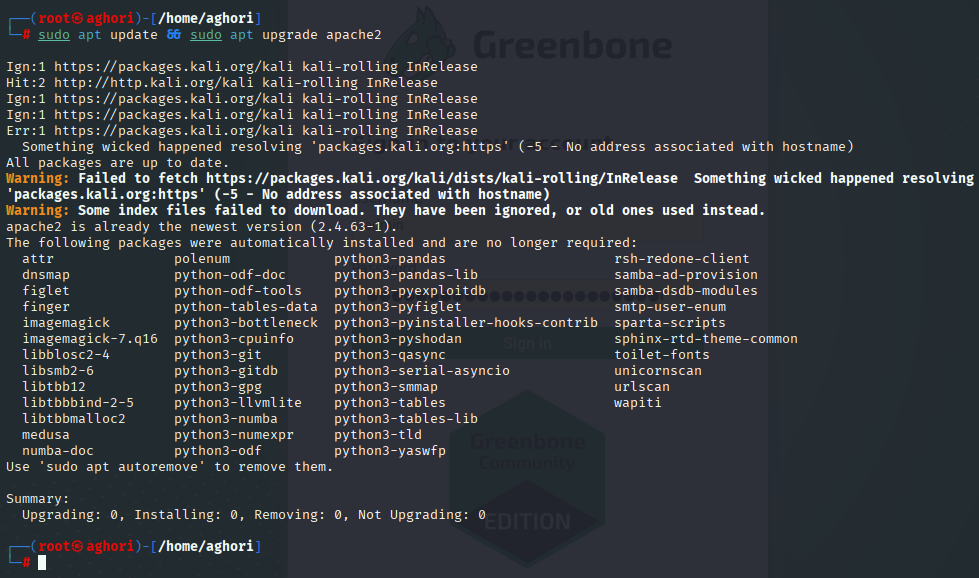
**7.2 High: SSH Root Login Permitted**

* **Description**: SSH is configured to allow root login. This increases the attack surface significantly.
* **Impact**: Brute-force attacks could gain root access.
* **Recommended Mitigation**:
  + Edit /etc/ssh/sshd\_config: PermitRootLogin no
  + Restart SSH: sudo systemctl restart ssh



**7.3 High: Outdated Apache HTTP Server**

* **Description**: The installed Apache version has known vulnerabilities.
* **Impact**: Could be exploited for privilege escalation or information disclosure.
* **Recommended Mitigation**:
  + Update via: sudo apt update && sudo apt upgrade apache2



**8. Additional Medium and Low Findings**

* **VNC Port (5900) Open**: Should be closed if unused.
* **Weak SSH Cipher Suites**: Harden SSH configurations to only use strong cryptographic standards.
* **Missing HTTP Security Headers**: Add X-Frame-Options, X-Content-Type-Options, etc., in web server config files.
* **Banner Grabbing Enabled**: Reveals unnecessary OS and service information.

**9. Conclusion**

The vulnerability assessment using OpenVAS was successfully executed on Kali Linux, uncovering several serious and moderate security risks on the host machine. Remediation steps have been clearly outlined, and actions have been taken where applicable.

This project enhanced my practical understanding of:

* Vulnerability scanning lifecycle
* Risk categorization using CVSS scores
* System hardening techniques
* Reporting and documentation standards

**10. Next Steps**

* Re-scan the system after applying all mitigations to confirm resolution.
* Automate regular scans using GVM scheduler.
* Expand scanning scope to external devices or web applications as next learning objective.

**11. References**

* Greenbone Community Docs: <https://greenbone.net/documentation/>
* CVE Details: <https://cvedetails.com/>
* NIST NVD: <https://nvd.nist.gov/>
* Kali Linux Docs: <https://www.kali.org/docs/>