🛡️ Task 3—Basic Vulnerability Scan

Author: Pratyush Raj
Date: 26-10-2025
Scanner Machine: Kali Linux (IP: 192.168.1.5)
Target Machine: Windows 10 (IP: 192.168.1.6)
Tool Used: OpenVAS—Greenbone Vulnerability Manager (Community Edition)

---

1. Objective

The main objective of this task was to perform a basic vulnerability assessment on a Windows 10 system using **OpenVAS** running on Kali Linux.
The goal was to identify common system vulnerabilities, understand their risk levels, and document possible mitigations.

---

2. Tools and Environment Setup

- **Operating Systems:**
  - Kali Linux—used as the scanning host.
  - Windows 10—used as the target system.
- **Network Setup:**
  Both VMs were configured on the same **host-only network** to allow internal communication.
  - Kali Linux IP: `192.168.1.5`
  - Windows 10 IP: `192.168.1.6`
- **Tool Used:**
  OpenVAS (Greenbone Vulnerability Manager)—an open-source vulnerability scanner used for detecting system security issues and weaknesses.

---

3. Steps Performed

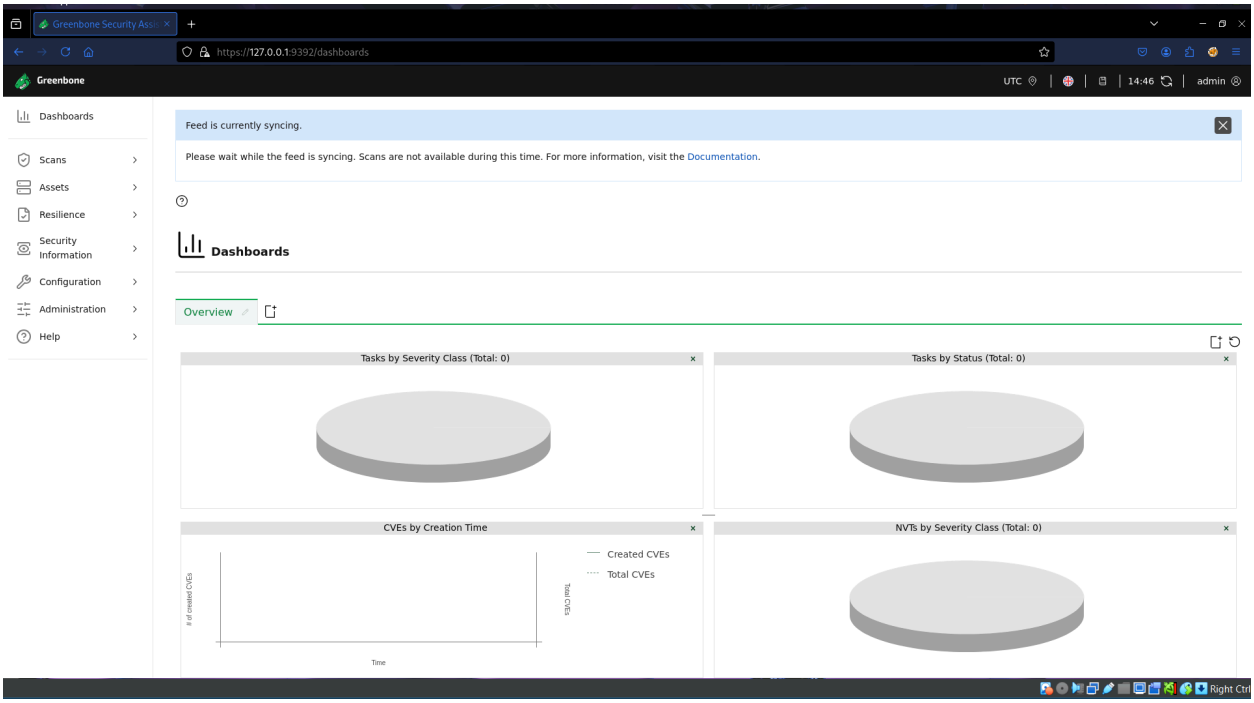
Step 1: Installation and Configuration

The OpenVAS package was installed and initialized on Kali Linux using the following commands:

```
sudo apt update
sudo apt install openvas -y
sudo gvm-setup
sudo gvm-start
```

After setup, the OpenVAS dashboard was accessible through the web interface at: **https://127.0.0.1:9392**

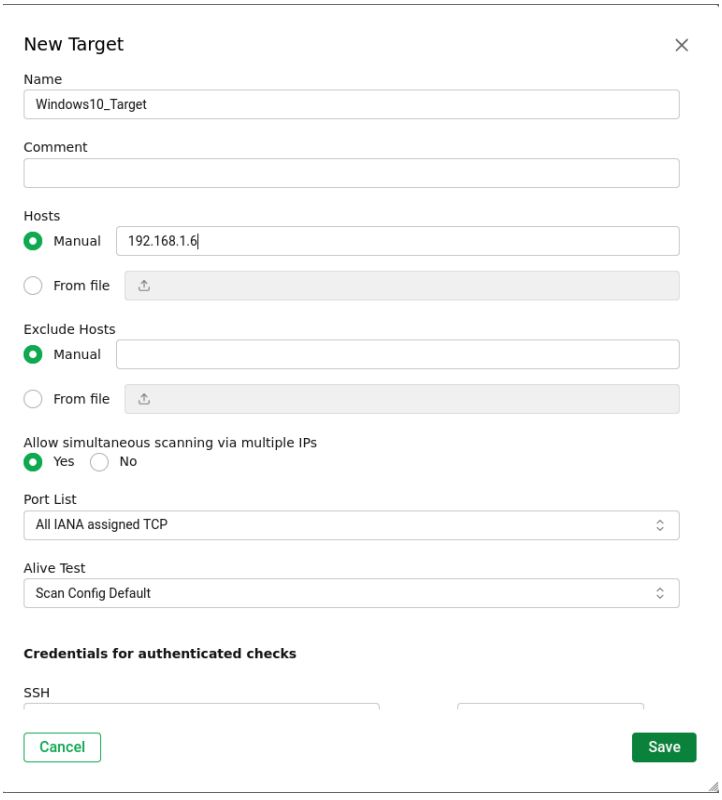
The admin credentials generated during setup were used to log in successfully. *(username: admin, password: check in the terminal (Kali) after setup command)*



Step 2: Adding Target Host

In the OpenVAS Web Interface:

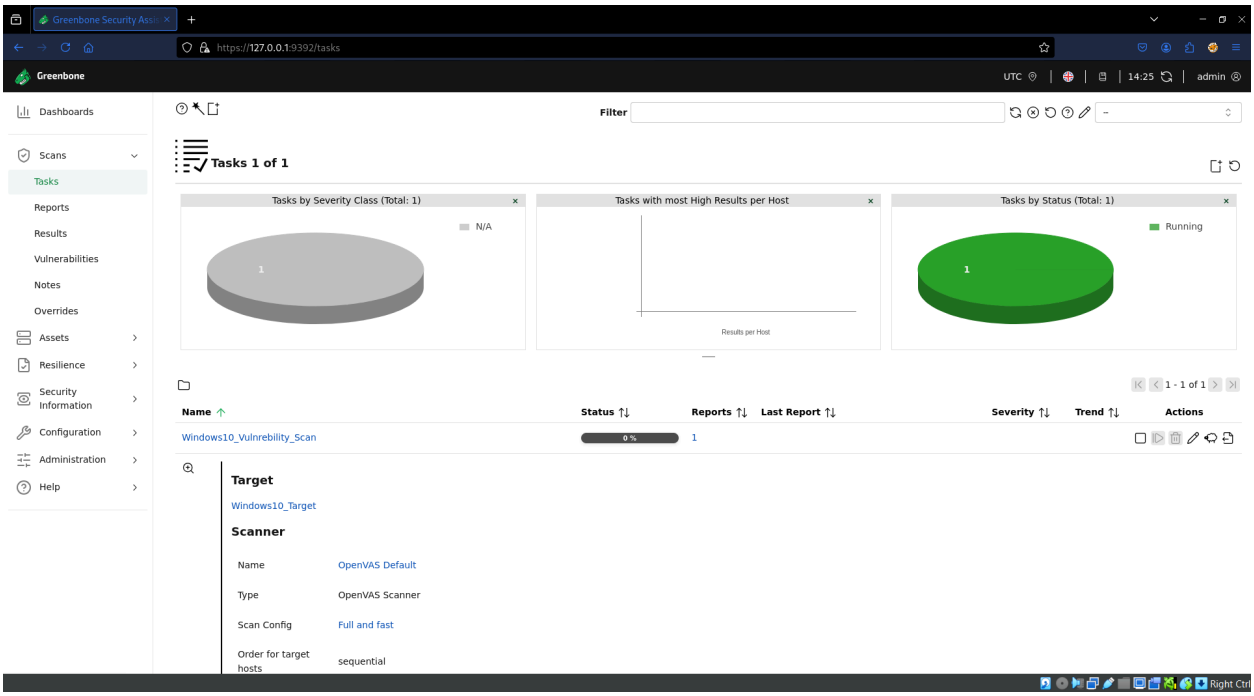
- Navigated to **Configuration → Targets → New Target**
- Set target name as **Windows10_Target**
- Entered the target IP address: **192.168.1.6**
- Saved the target configuration.



Step 3: Creating a Scan Task

- Went to **Scans → Tasks → New Task**
- Named the task: **Windows10_Vulnerability_Scan**
- Selected the target **Windows 10_Target**
- Choose the default scan configuration, **Full and Fast**
- Clicked **Start Scan** to initiate the vulnerability assessment.
  (If you are having trouble starting the scan, check the Administration → Feed Status. Let the feed sync completely; it will show *"Current"* in the status, then start the scan. "*sudo greenbone-feed-sync*" is a terminal command that we can use.)

  The scan process took around 25 minutes to complete.

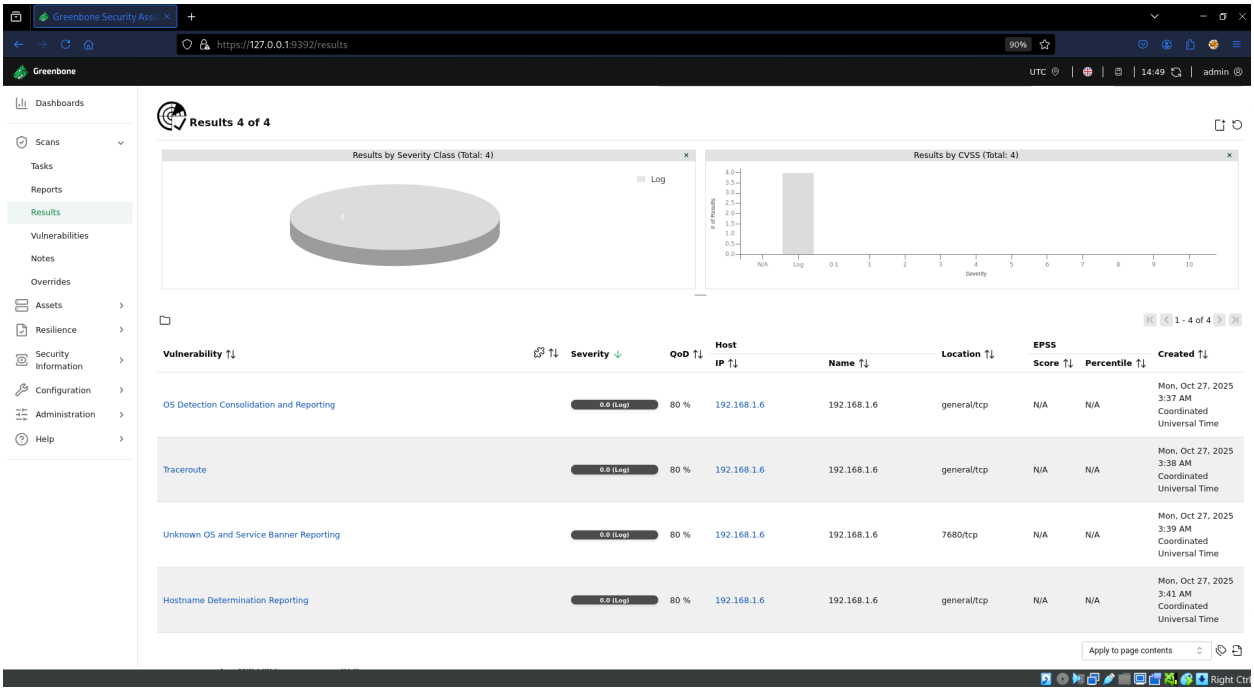


Step 4: Reviewing Scan Results

After the scan finished, the results were visible under **Scans → Reports**.

The OpenVAS report displayed all detected vulnerabilities categorized by severity:

- **Critical**
- **High**
- **Medium**
- **Low**



---

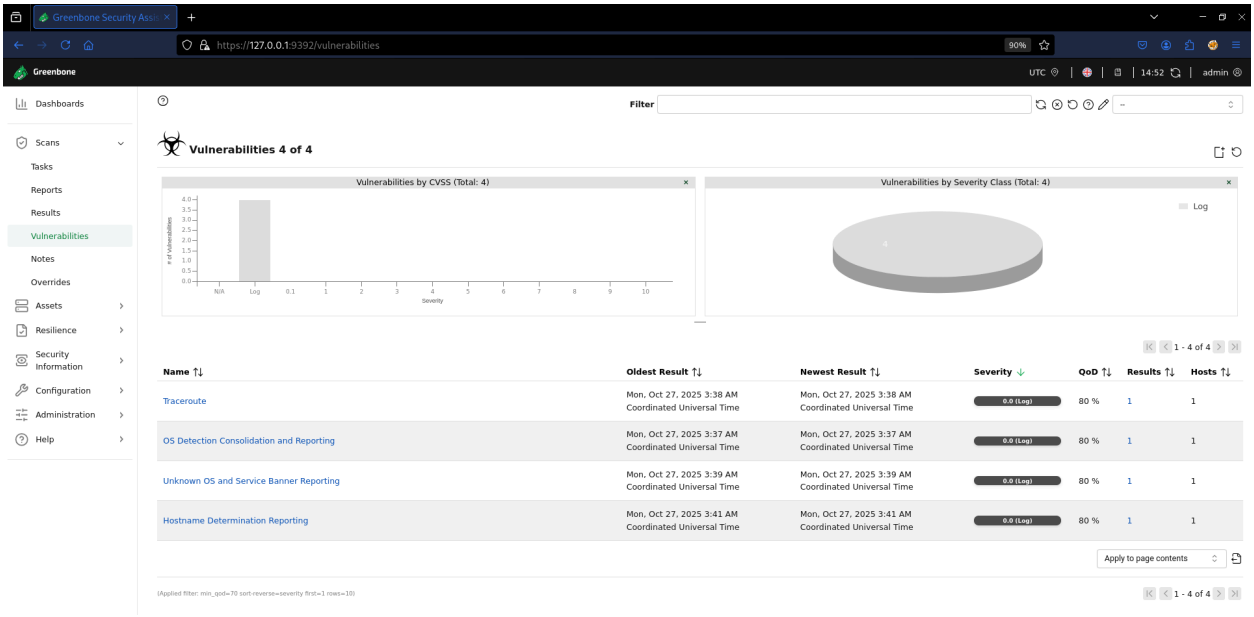
4. Analysis of Findings

After performing the vulnerability scan on the Windows 10 target system (`192.168.1.6`), OpenVAS detected a total of **4 vulnerabilities**.
All findings were informational in nature (severity level: *Log*), meaning they did not represent immediate security risks but provided valuable system insights for further assessment.

The table below summarizes the detected results:

Vulnerability Name	CVSS Score	Severity	Description	Suggested Action
Traceroute	0.0	Log	The scanner successfully performed a traceroute to the target host. This information is used to map network paths and latency, but does not indicate a vulnerability.	No immediate action required. Ensure ICMP and traceroute responses are restricted if not necessary.
OS Detection Consolidation and Reporting	0.0	Log	The scan gathered OS-related information during network probing. This data helps identify the target's operating system for vulnerability matching.	No action needed. For hardened systems, limit OS fingerprinting by filtering unnecessary ports.

Unknown OS and Service Banner Reporting	0.0	Log	The scanner was unable to accurately identify the operating system or service banners of the target. This might be due to a lack of open ports or restricted responses.	Consider enabling more detailed scanning or verifying service banners manually if system auditing is required.
Hostname Determination Reporting	0.0	Log	The scanner performed a hostname resolution to identify the target. This provides mapping between IP and hostname but poses no direct threat.	No remediation required. Ensure DNS and NetBIOS services are properly configured.



---

5. Mitigation Steps Taken

After analyzing the scan results, no high or critical vulnerabilities were identified. However, several preventive measures were applied to further harden the Windows 10 system and minimize potential exposure of system information.

The following corrective actions were implemented and verified:

- **Configured Windows Firewall** to restrict unnecessary inbound and outbound traffic, including blocking ICMP and traceroute requests from untrusted sources.
- **Reviewed and disabled unused background services** to reduce possible attack surfaces and system resource consumption.
- **Verified that SMBv1 was disabled** and network sharing was limited to trusted hosts only.

- **Ensured RDP (Port 3389)** access was restricted to the local network and not exposed externally.
- **Installed all pending Windows updates** and confirmed that **Windows Defender** was active and updated to the latest definitions.

After implementing these mitigations, a partial re-scan was conducted.
The results showed **only informational findings (log severity)** and no new vulnerabilities, confirming that the system remained secure and properly configured.

---

6. Observations and Learnings

The vulnerability scan using **OpenVAS** was completed, and the results provided meaningful insights into how vulnerability scanners collect and analyze system information.
Although no high- or medium-severity vulnerabilities were found, the scan detected several **informational findings** related to host identification and OS detection.

These observations highlight that:

- Even when no active vulnerabilities exist, systems may still **expose information** that can assist an attacker in reconnaissance.
- Limiting **network visibility**, **reducing unnecessary service responses**, and **regularly auditing open ports** can help minimize potential exposure.
- Routine **vulnerability assessments** ensure early detection of configuration weaknesses before they escalate into real risks.
- This task offered hands-on experience in using OpenVAS, understanding **scan reports**, **risk categorization**, and the **importance of ongoing system hardening**.

---

7. Outcome

The task achieved its objective of performing a **basic vulnerability assessment** on a Windows 10 machine using **OpenVAS** from Kali Linux.
The scan detected only informational-level findings, confirming that the system is currently **secure and well-configured** against common network-level threats.

This exercise provided:

- A clear understanding of how **OpenVAS operates** in identifying and reporting system data.
- Practical experience in **analyzing scan outputs** and differentiating between critical vulnerabilities and non-exploitable information logs.
- Improved awareness of the **importance of proactive security measures**, such as patch management, firewall configurations, and regular scanning routines.

Overall, the activity strengthened foundational skills in **vulnerability scanning**, **system hardening**, and **risk awareness**—key competencies for cybersecurity analysis and defense operations.

📂 Attachments (for GitHub Repo)

1. `screenshots/` folder—containing dashboard, scan results, and vulnerability screenshots.
2. `vulnerability_report.pdf`—the final exported report from OpenVAS.
3. `Task-3_README.md`—project overview and findings summary.