Performing a Vulnerability Scan with Greenbone

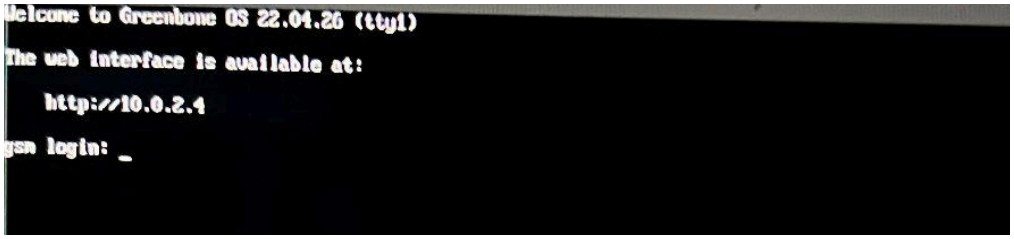
By

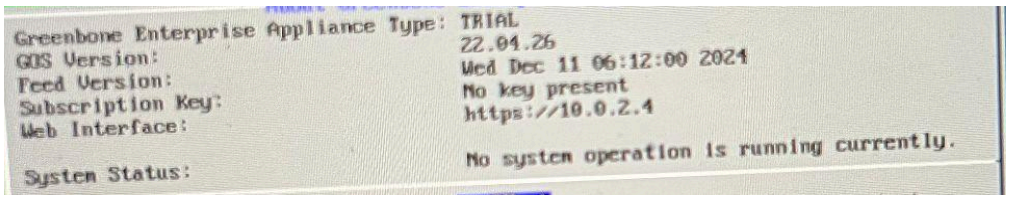
Russell Robinson

Vulnerability scanning is a critical step in identifying security weaknesses within a system or network. Using Greenbone Vulnerability Manager, you can efficiently detect, analyze, and address vulnerabilities in your target systems. This guide walks through the process of setting up and running a vulnerability scan, including configuring your Greenbone VM, connecting to your target machine, and interpreting the results.

1. **Start Greenbone VM**

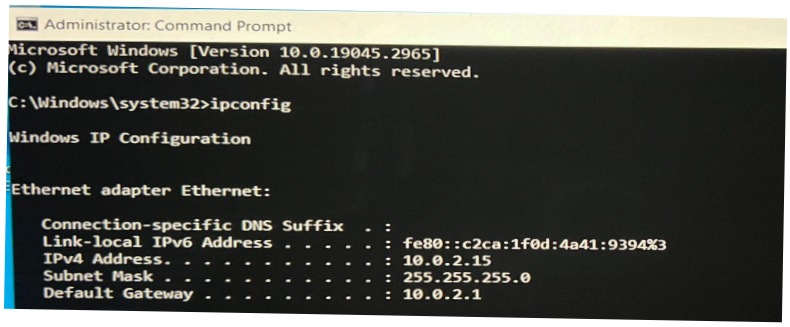
I started my Greenbone VM, logged in with my username and password, and located the IP address to access my Greenbone account for performing a vulnerability scan.





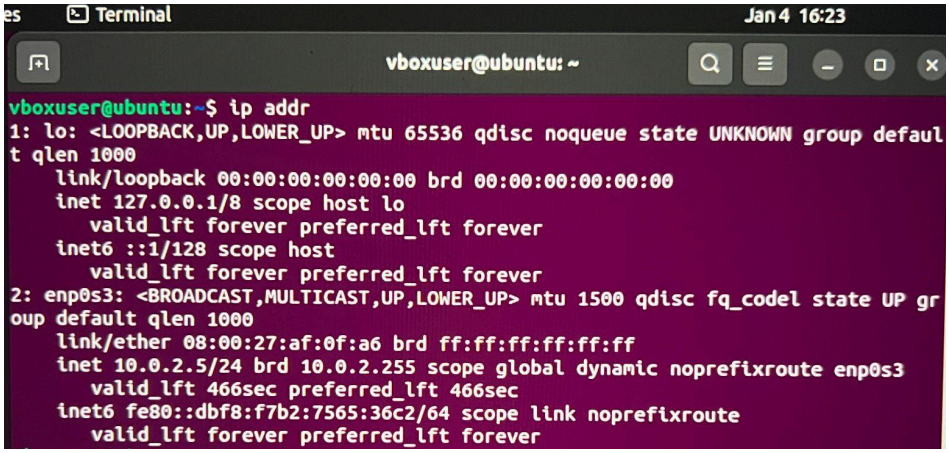
2. **Start Windows 10 Target Machine**

I started my Windows 10 target machine, opened the terminal under administrator, and typed ipconfig at the prompt to retrieve the IP address. I saved this IP address in my notes.



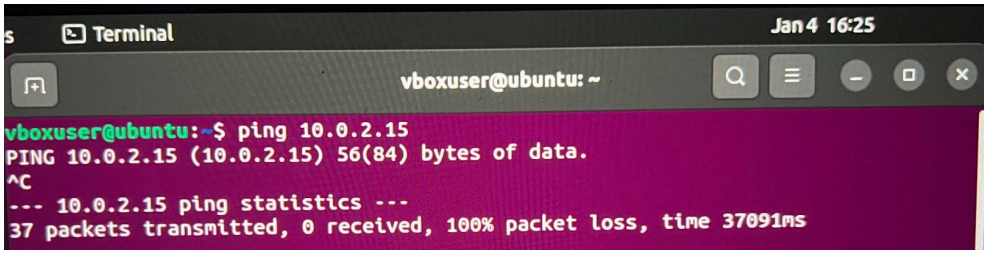
3. **Start Ubuntu Machine**

I started my Ubuntu machine, logged in, opened the terminal, and entered the command ip addr to get my IP address. I saved it in my notes and closed the terminal.



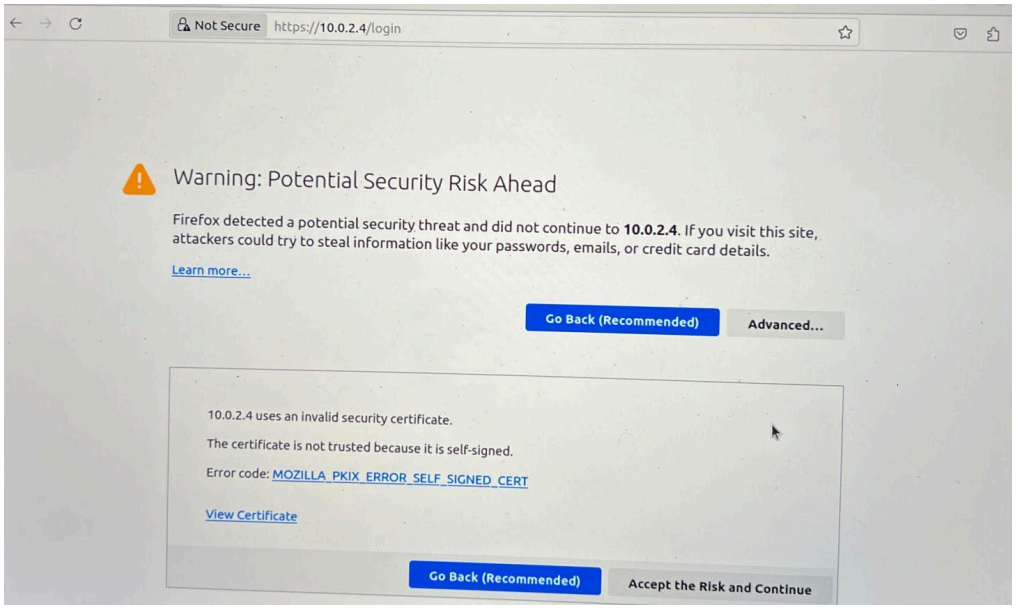
4. **Ping the Target Machine**

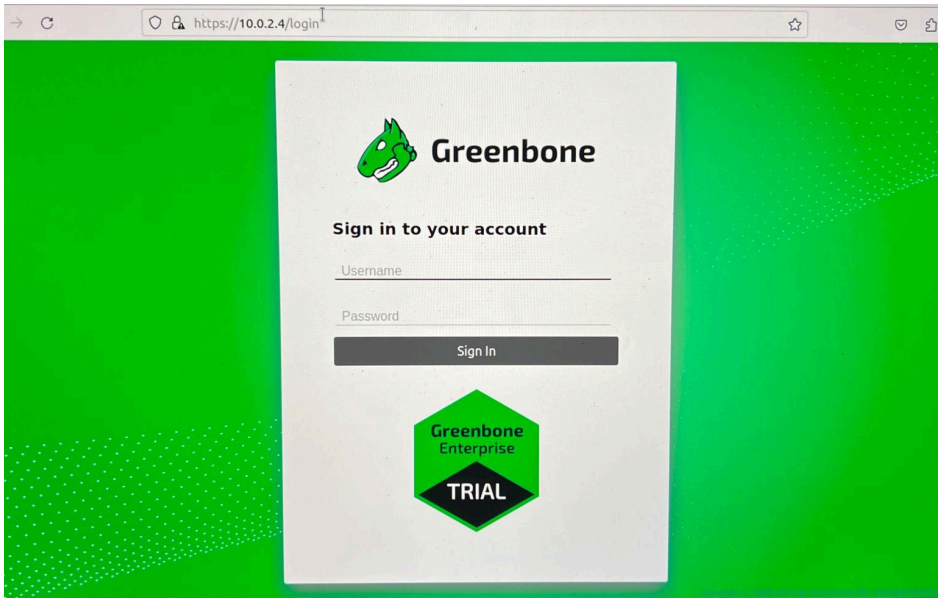
I opened a new terminal and entered the command ping <IP address> (using the IP address of the target machine) to establish a connection. Once the connection was established, I pressed Ctrl + C to stop the ping.



5. **Access Greenbone VM**

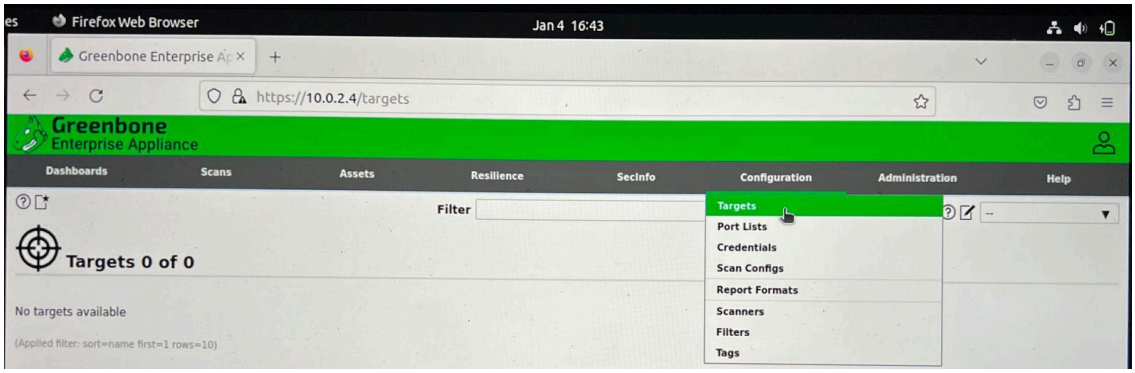
In my Firefox browser, I used the IP address of my Greenbone VM: https://<IP address>. A warning screen appeared; I scrolled down and clicked "Accept Risk" to access the Greenbone login page.

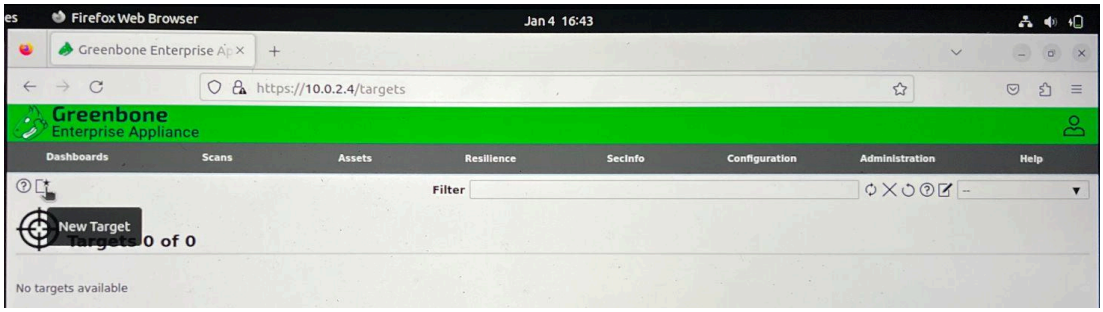


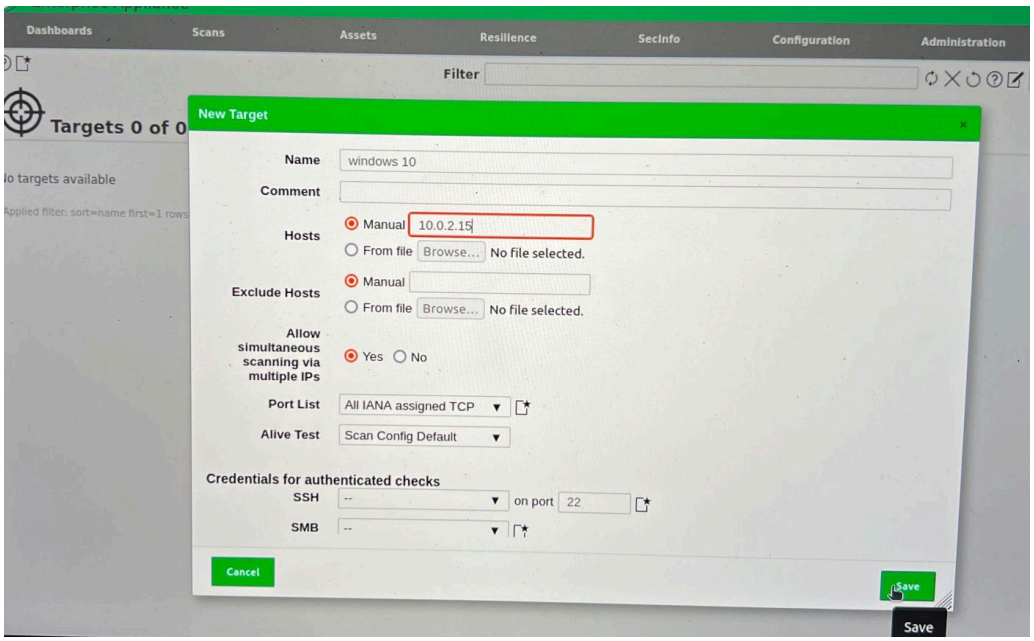


6. **Configure Greenbone Targets**

I navigated to the **Configuration** section and selected **Targets**. I clicked **Create New Target** and gave the target a name (*Windows 10*).In the **Host** section, I entered the IP address of the target machine and clicked **Save**.

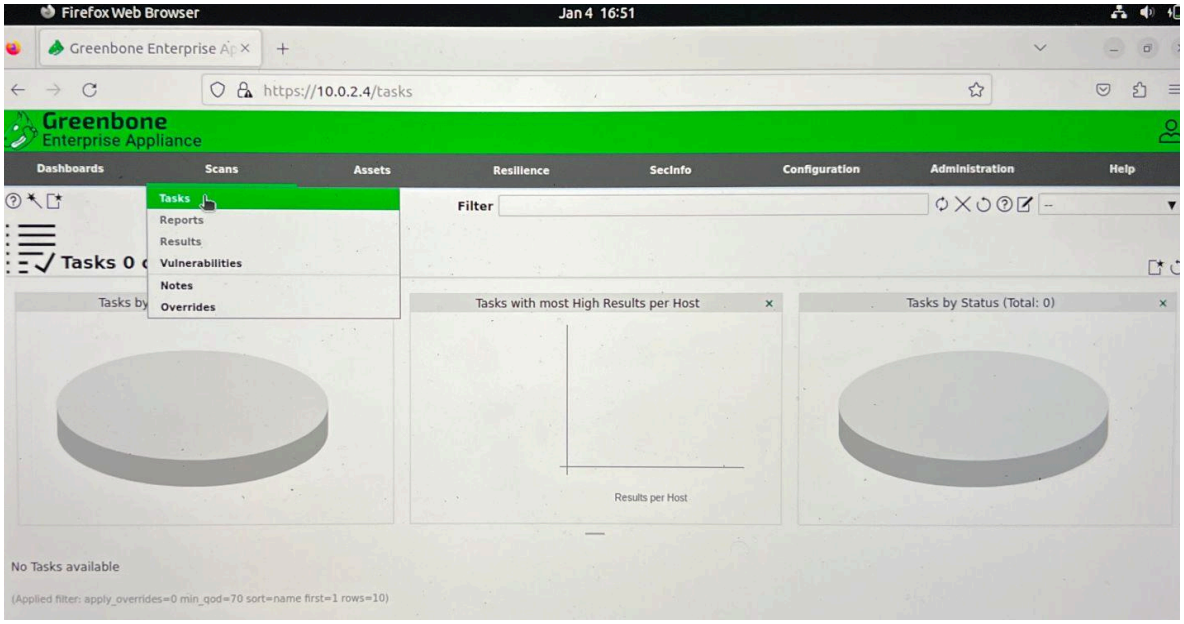


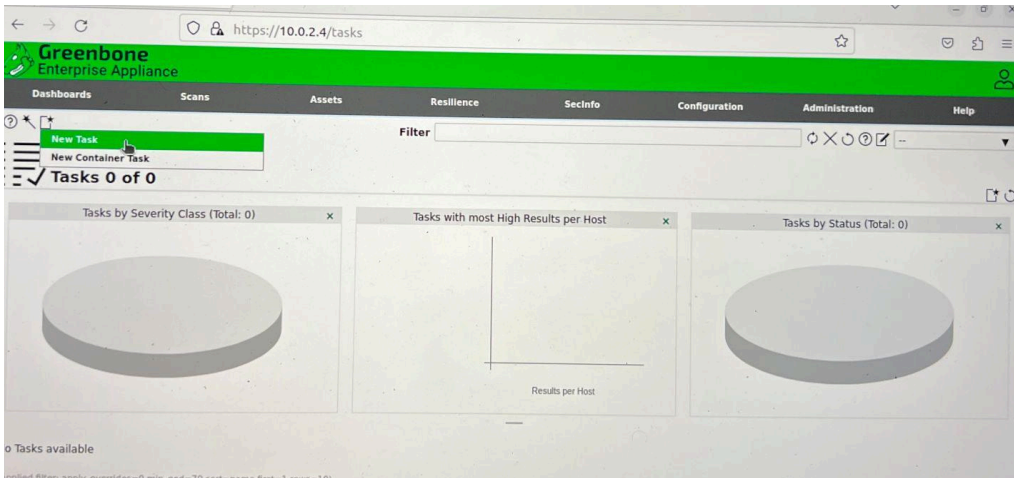


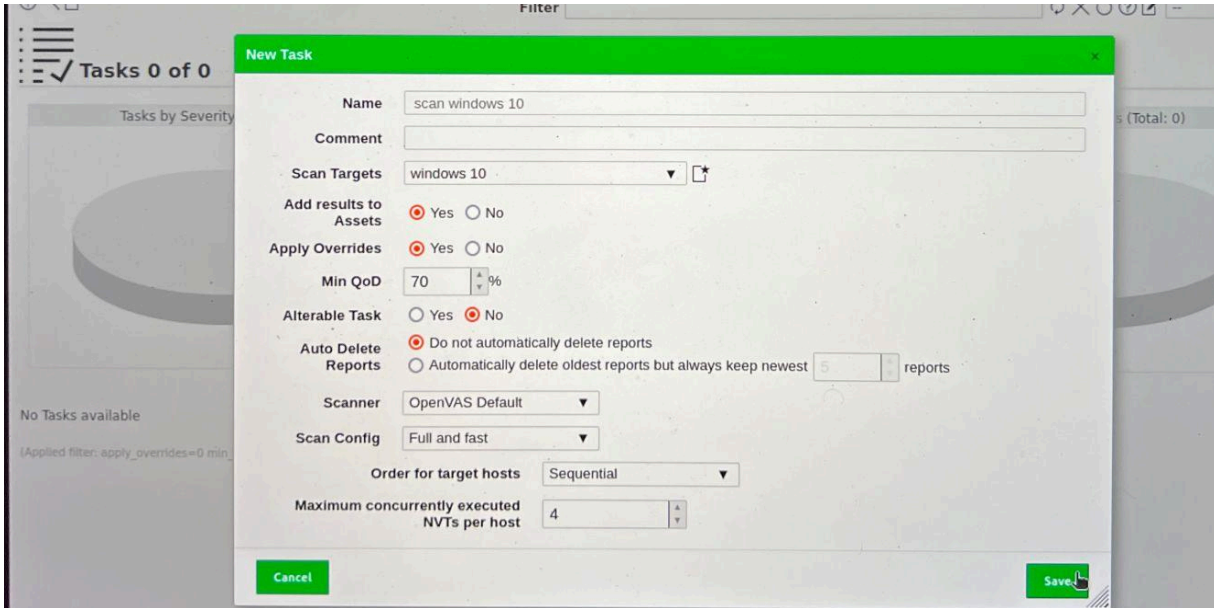


7. **Create a Scan Task**

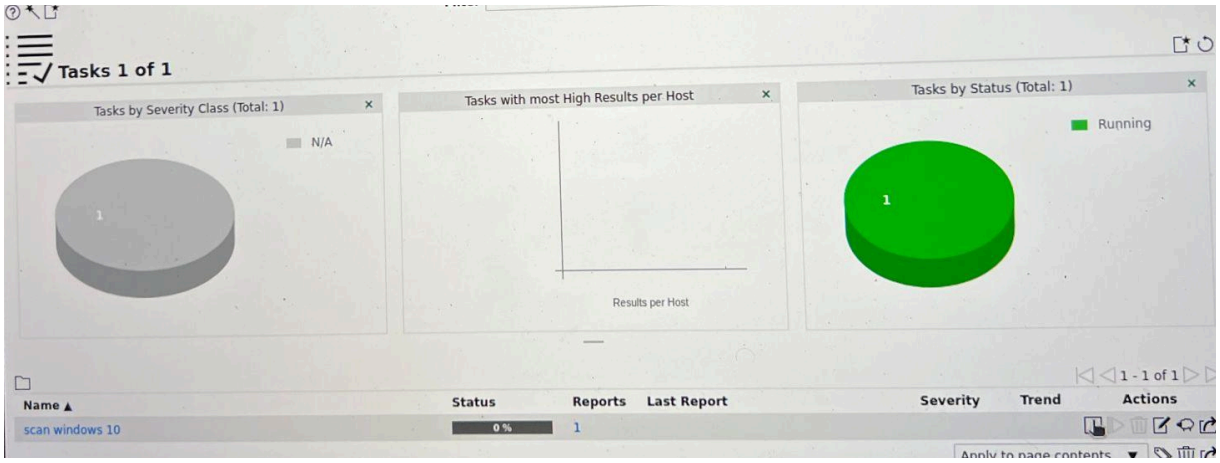
I selected the **Scans** section, then chose **Tasks** and clicked **Create New Task**. named the task (*Scan Windows 10*) and selected the target I created (*Windows 10*) from the **Scan Targets** dropdown. Finally, I saved the task.

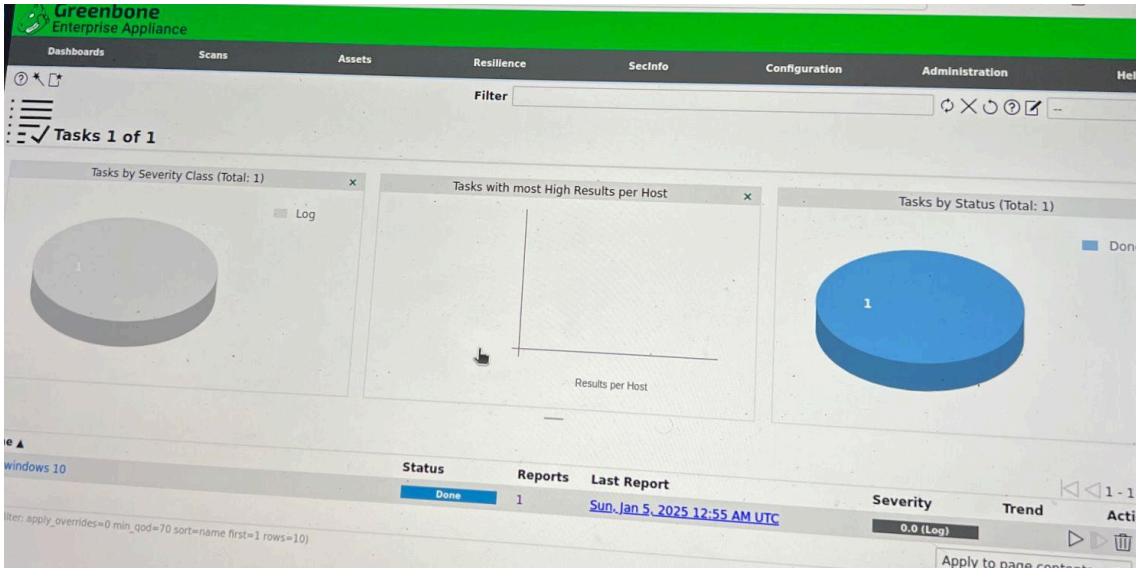

Firefox Web Browser
Jan 4 16:51
Greenbone Enterprise Ap
https://10.0.2.4/tasks
Greenbone
Enterprise Appliance
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Tasks
Reports
Results
Vulnerabilities
Notes
Overrides
Filter
Tasks 0
Tasks by
Tasks with most High Results per Host
Results per Host
Tasks by Status (Total: 0)
No Tasks available
(Applied filter: apply_overrides=0 min_qod=70 sort=name first=1 rows=10)


https://10.0.2.4/tasks
Greenbone
Enterprise Appliance
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
New Task
New Container Task
Filter
Tasks 0 of 0
Tasks by Severity Class (Total: 0)
Tasks with most High Results per Host
Results per Host
Tasks by Status (Total: 0)
o Tasks available

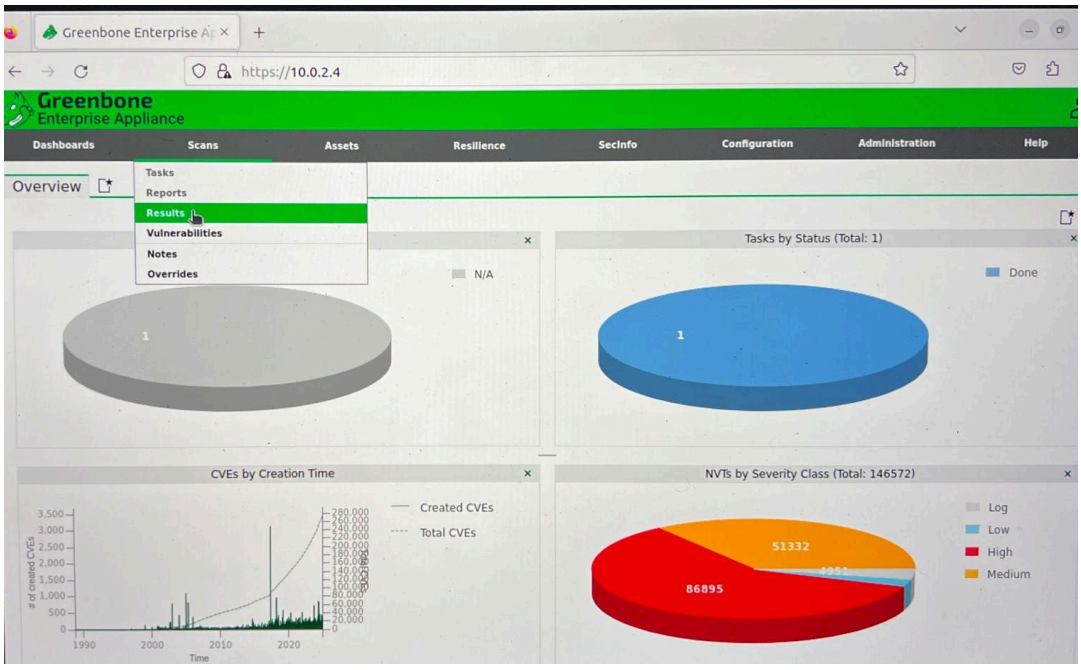


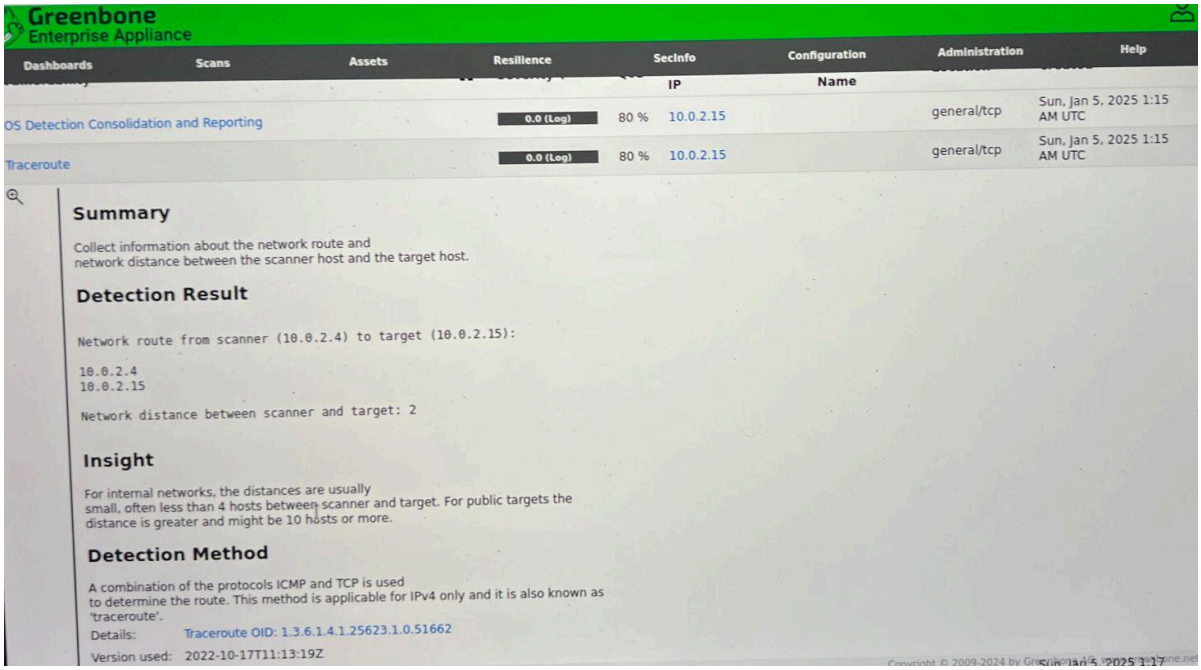
8. **Run the Scan** From the dashboard, I started the scan I had created. I monitored the progress, and once the status reached **100%**, the scan was complete.





9. **Review Results** Under the **Scans** section, I selected **Results** to view any vulnerabilities discovered during the scan. Vulnerabilities were listed by their severity. By double-clicking on a vulnerability, I accessed a summary with references such as CVE, CERT, and other resources providing instructions on how to address the issue.





Conclusion

This project demonstrated a full vulnerability scanning workflow using Greenbone. Key skills included configuring virtual environments, performing network tests, and analyzing scan results. With zero severity findings, the project emphasized the importance of regular vulnerability assessments and proactive security measures.