



LOVELY
PROFESSIONAL
UNIVERSITY

Transforming Education Transforming India

INT 301 – Open Source Technologies

CA3 Project Report

On

*Listing all the security updates missing and system information in your
Window OS using OpenVAS (open source software)*

Submitted by

Name: Aryan Kumar Saxena

Reg. No: 11903134

Roll No: 04

Under the Guidance of

Rajeshwar Sharma (29484)

School of Computer Science & Engineering,
Lovely Professional University,
Phagwara

(April 2023)

1. INTRODUCTION

Welcome to this project on using Open Source Software to identify missing security updates in Windows OS and gathering system information. In today's world, where cyber threats are becoming more and more sophisticated, it is crucial to keep our operating systems up to date with the latest security patches. However, it can be challenging to identify which security updates are missing from our Windows OS.

This project aims to use Open Source Software to make this process easier by identifying all the missing security updates in the Windows OS. Additionally, this project will also collect free system information about the hardware and software installed in the system.

By using Open Source Software, we can achieve this task without having to rely on proprietary tools that can be costly and restrictive. The software we will be using is free, open source, and readily available, making it accessible to everyone.

The open source software used in this project is OpenVAS, to identify missing security updates in Windows OS and gather system information. In today's world, where cyber threats are becoming more and more sophisticated, it is crucial to keep our operating systems up to date with the latest security patches. However, it can be challenging to identify which security updates are missing from our Windows OS.

OpenVAS is an open-source vulnerability scanning and management tool that can help us identify missing security updates in Windows OS. Additionally, OpenVAS can gather system information about the hardware and software installed in the system. By using OpenVAS, we can achieve this task without having to rely on proprietary tools that can be costly and restrictive.

1.1 Objective of the Project

The objectives of this project are:

- To identify all the missing security updates in Windows OS by using Open Source Software.
- To gather free system information about the hardware and software installed in the system.
- To provide a comprehensive list of missing security updates that can be easily addressed to improve the security of the Windows OS.
- To increase awareness about the importance of keeping the Windows OS up to date with the latest security patches.
- To promote the use of Open Source Software as a viable alternative to proprietary tools for system security and information gathering.
- To configure and run OpenVAS to scan for missing security updates in Windows OS.
- To gather system information about the hardware and software installed in the system using OpenVAS.
- To provide a comprehensive list of missing security updates that can be easily addressed to improve the security of the Windows OS.
- To increase awareness about the importance of keeping the Windows OS up to date with the latest security patches.
- To promote the use of Open Source Software, specifically OpenVAS, as a viable alternative to proprietary tools for system security and information gathering.

1.2 Description of the Project

In this project, we will be using OpenVAS to scan for missing security updates in Windows OS and gather system information. OpenVAS is an open-source vulnerability scanning and management tool that allows us to identify potential vulnerabilities in our systems and applications.

To get started, we will first need to download and install OpenVAS on our system. Once we have installed OpenVAS, we will need to configure it to scan for missing security updates in Windows OS. This can be

done by configuring OpenVAS to scan for vulnerabilities specific to the Windows OS and then initiating a scan.

After the scan is complete, OpenVAS will generate a report that includes all the missing security updates in the Windows OS. This report will include details such as the severity of the vulnerability, the patch that needs to be installed, and any other relevant information.

In addition to identifying missing security updates, OpenVAS can also gather system information about the hardware and software installed in the system. This information can be used to determine the system's overall health and identify any potential areas for improvement.

Overall, this project will demonstrate how to use OpenVAS to improve the security of our Windows OS by identifying missing security updates and gathering system information. It will also showcase the benefits of using Open Source Software as a viable alternative to proprietary tools for system security and information gathering.

1.3 Scope of the project:

The scope of this project is to use OpenVAS to identify missing security updates in Windows OS and gather system information. The project will focus on the following areas:

1. Installation and configuration of OpenVAS: The project will cover the installation and configuration of OpenVAS to scan for missing security updates in Windows OS and gather system information.
2. Scanning for missing security updates: The project will demonstrate how to initiate a scan using OpenVAS to identify missing security updates in the Windows OS.
3. Gathering system information: The project will cover the process of gathering system information using OpenVAS, including information about the hardware and software installed in the system.
4. Generating reports: The project will show how to generate reports using OpenVAS that summarize the results of the vulnerability scan and provide information about missing security updates and system information.
5. Recommendations for improving system security: Based on the results of the vulnerability scan, the project will provide recommendations for improving the security of the Windows OS.

The project's scope is limited to using OpenVAS for vulnerability scanning and system information gathering in Windows OS. The project will not cover other tools or methods for system security or information gathering. The project's aim is to demonstrate the benefits of using Open Source Software, specifically OpenVAS, for system security and information gathering.

2. SYSTEM DESCRIPTION

The system for this project will require the following components:

- ✓ Operating system: The system will require a Windows OS to scan for missing security updates and gather system information using OpenVAS.
- ✓ OpenVAS: OpenVAS is an open-source vulnerability scanning and management tool that will be used to scan for missing security updates and gather system information.
- ✓ Internet connection: The system will require an internet connection to download and install OpenVAS and to retrieve the latest security patches for the Windows OS.
- ✓ User interface: The system will use a graphical user interface to interact with OpenVAS, initiate scans, and view scan results.
- ✓ System resources: The system will require sufficient resources, such as CPU, RAM, and storage, to support the installation and operation of OpenVAS and the Windows OS.

Once the above components are installed and configured, the system will be able to initiate a scan using OpenVAS to identify missing security updates in the Windows OS. The system will also be able to gather system information about the hardware and software installed in the system. After the scan is complete,

OpenVAS will generate a report that includes details about the missing security updates and system information. The report can then be used to improve the security of the Windows OS by installing the missing security updates and taking other recommended measures. The system can be used by individuals, organizations, or system administrators looking to improve the security of their Windows OS and gain insight into their system's hardware and software.

2.1 Target System Description:

The target system for this project will be a Windows OS-based computer that requires vulnerability scanning and system information gathering. The system should meet the following requirements:

1. Operating system: The system should be running a Windows OS version that is compatible with OpenVAS.
2. Internet connection: The system should have an internet connection to download and install OpenVAS and retrieve the latest security patches for the Windows OS.
3. User access: The user accessing the target system should have sufficient privileges to install and configure OpenVAS and initiate a vulnerability scan.
4. System resources: The target system should have sufficient resources, such as CPU, RAM, and storage, to support the installation and operation of OpenVAS and the Windows OS.
5. Installed software: The target system should have all the required software installed, such as the .NET Framework, to support the installation and operation of OpenVAS.

Once the target system meets the above requirements, the user can install and configure OpenVAS to initiate a vulnerability scan and gather system information. The results of the scan can then be used to identify missing security updates and improve the overall security of the target system. The target system can be used by individuals, organizations, or system administrators looking to improve the security of their Windows OS-based computer and gain insight into their system's hardware and software.

2.2 Assumptions and Dependencies:

The following assumptions and dependencies should be considered before implementing the project:

Assumptions:

1. The user has a basic understanding of the Windows OS and the concepts of system security.
2. The user has access to a Windows OS-based computer that meets the requirements for installing and running OpenVAS.
3. The user has administrative privileges to install and configure OpenVAS and initiate a vulnerability scan.
4. The user has a stable internet connection to download and install OpenVAS and retrieve the latest security patches for the Windows OS.
5. The user has knowledge of the hardware and software installed on the target system.

Dependencies:

1. .NET Framework: OpenVAS requires the .NET Framework to be installed on the target system to run. The user must ensure that the .NET Framework is installed on the target system before installing OpenVAS.
2. OpenVAS installation package: The user must have access to the OpenVAS installation package for Windows OS, which can be downloaded from the OpenVAS website.
3. Latest security patches: To accurately identify missing security updates, the user must ensure that the Windows OS on the target system is up-to-date with the latest security patches.
4. Internet connection: The system requires a stable internet connection to download and install OpenVAS and retrieve the latest security patches for the Windows OS.
5. User interface: The user will require a user interface to interact with OpenVAS, initiate scans, and view scan results.

By considering these assumptions and dependencies, the user can ensure that the project is implemented successfully, and the desired results are achieved.

2.3 Functional/Non-Functional Dependencies

The functional dependencies for this project are:

- OpenVAS software must be able to access the internet to download the latest security updates.
- The system information tools used must be able to gather accurate and up-to-date information about the hardware and software installed on the system.

The non-functional dependencies for this project are:

- The speed and performance of the system may affect the time taken to complete the scan.
- The accuracy of the system information gathered may be affected by the quality of the system information tools used.

2.4 Data set used in support of your project

There is no specific data set required for this project. OpenVAS software and free system information tools will be used to gather the necessary information.

3. ANALYSIS REPORT

3.1 System snapshots and full analysis report

OpenVAS/GVM

OpenVAS/GVM is a fully-featured vulnerability scanner, but it's also one component of the larger "Greenbone Security Manager" (GSM).



OpenVAS dates back to 2009 and the project is maintained by a commercial/open-source company. With its focus on the enterprise market and its long history, any risks of enterprises adopting a technology that might become abandoned are greatly reduced.

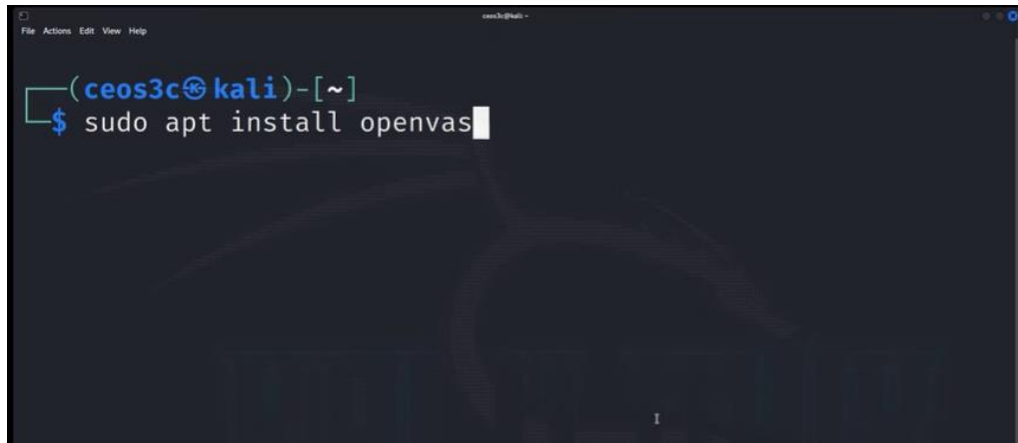
Here are some notable positives of OpenVAS/GVM:

- Has a long history (since 2009) with daily updates and over 50,000 vulnerability tests
- Is backed by an enterprise software-security company
- Can perform various types of authenticated/unauthenticated tests
- Supports a variety of high- and low-level Internet and industrial protocols
- Has an internal programming language that can be used for implementing custom vulnerability tests

3.1.1 Installation and Configuration of OpenVAS in kali Linux

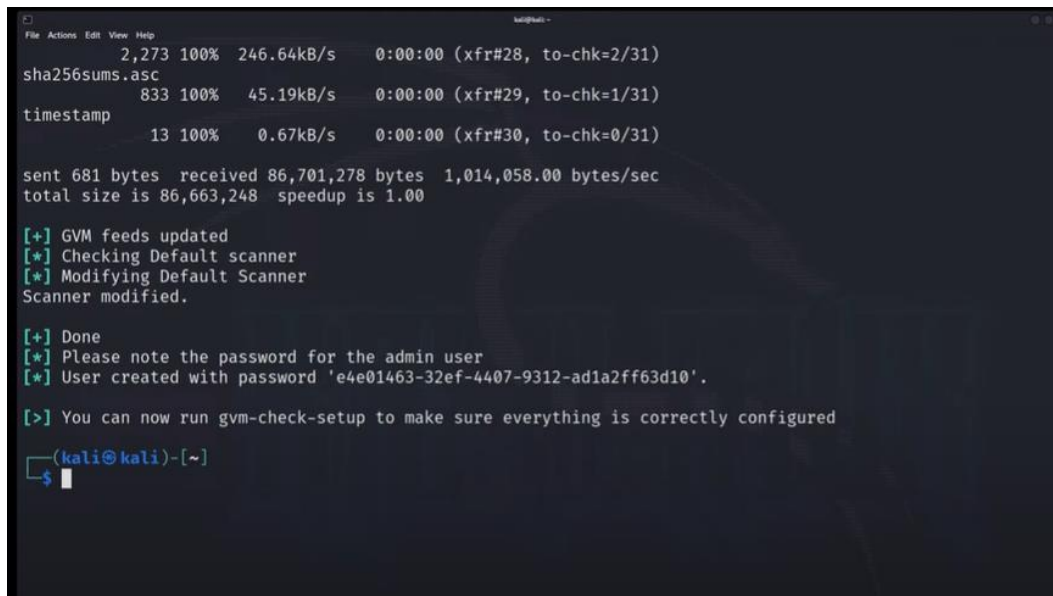
We can install OpenVAS on Kali Linux by following these steps:

1. Open a terminal window in Kali Linux by pressing Ctrl+Alt+T or by searching for "terminal" in the applications menu.
2. Update the package lists by running the following command:
 `sudo apt update`
3. Install the OpenVAS package by running the following command:
 `sudo apt install OpenVAS`



```
File Actions Edit View Help
(ceos3c@kali)-[~]
$ sudo apt install openvas
```

4. During the installation process, you will be prompted to create a password for the OpenVAS user. Enter a strong password and remember it, as you will need it to log in to the OpenVAS web interface.
5. Once the installation is complete, run the following command to update the OpenVAS database:
 🚀 `sudo openvas-feed-update`
 OR
 🚀 `sudo openvas-setup`



```
File Actions Edit View Help
2,273 100% 246.64kB/s 0:00:00 (xfr#28, to-chk=2/31)
sha256sums.asc
833 100% 45.19kB/s 0:00:00 (xfr#29, to-chk=1/31)
timestamp
13 100% 0.67kB/s 0:00:00 (xfr#30, to-chk=0/31)

sent 681 bytes received 86,701,278 bytes 1,014,058.00 bytes/sec
total size is 86,663,248 speedup is 1.00

[+] GVM feeds updated
[*] Checking Default scanner
[*] Modifying Default Scanner
Scanner modified.

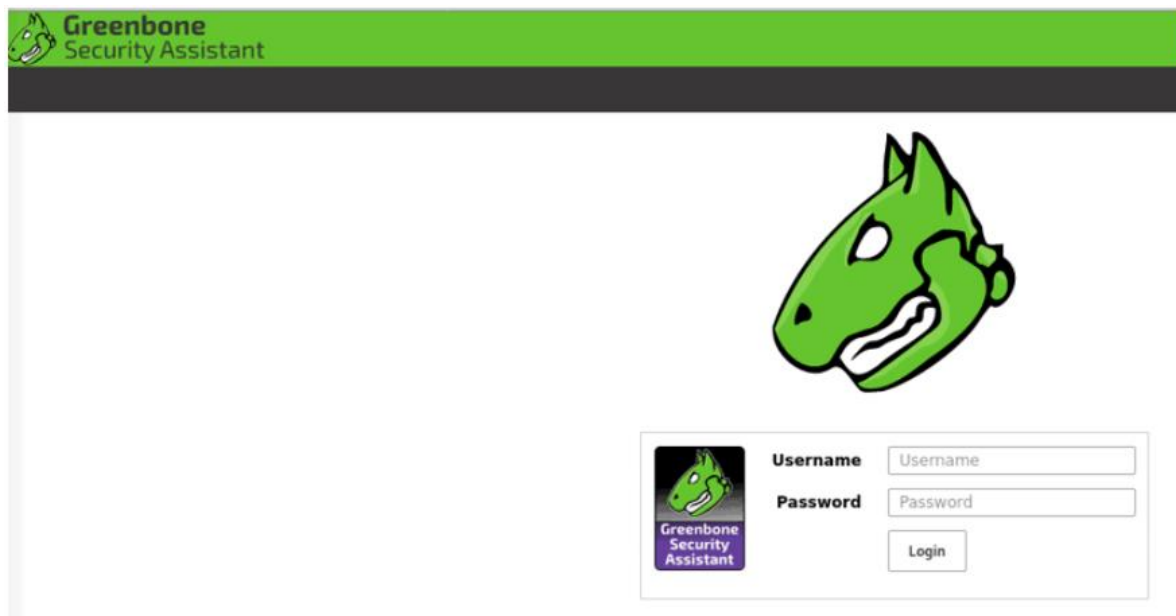
[+] Done
[*] Please note the password for the admin user
[*] User created with password 'e4e01463-32ef-4407-9312-ad1a2ff63d10'.

[>] You can now run gvm-check-setup to make sure everything is correctly configured

(kali@kali)-[~]
$
```

6. Finally, start the OpenVAS service by running the following command:
 🚀 `sudo systemctl start openvas-scanner.service`
 🚀 `sudo systemctl start openvas-manager.service`
 🚀 `sudo systemctl start greenbone-security-assistant.service`

We can now access the OpenVAS web interface by opening a web browser and navigating to <https://localhost:9392> or <https://127.0.0.1:9392>. Enter the username OpenVAS and the password you created during the installation process.



We can login to the dashboard using the following username/password details:

- admin
- admin

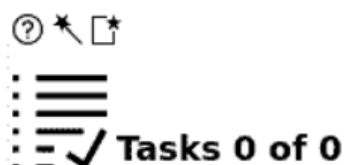
You should then see the dashboard of OpenVAS/GVM as shown here:



Our first test will be to configure a simple scan using OpenVAS/GVM on a single IP address. We have chosen one of the malvertising IPs we track as a test: 192.243.59.20

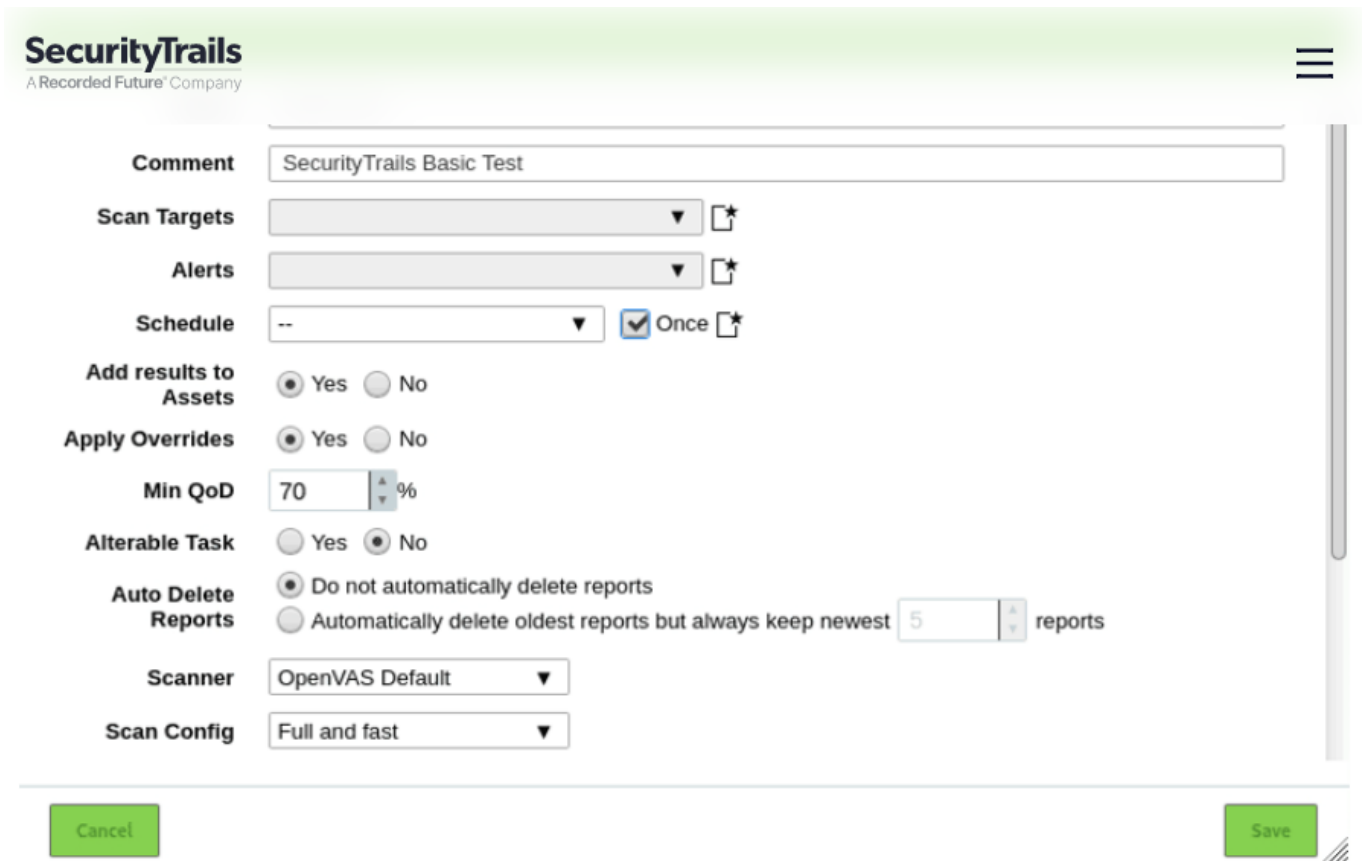
To conduct a new scan, we follow the path of: Scans > Tasks

Once the page loads, there is an option to create a new task on the top left of the screen:



➤ Create a new task

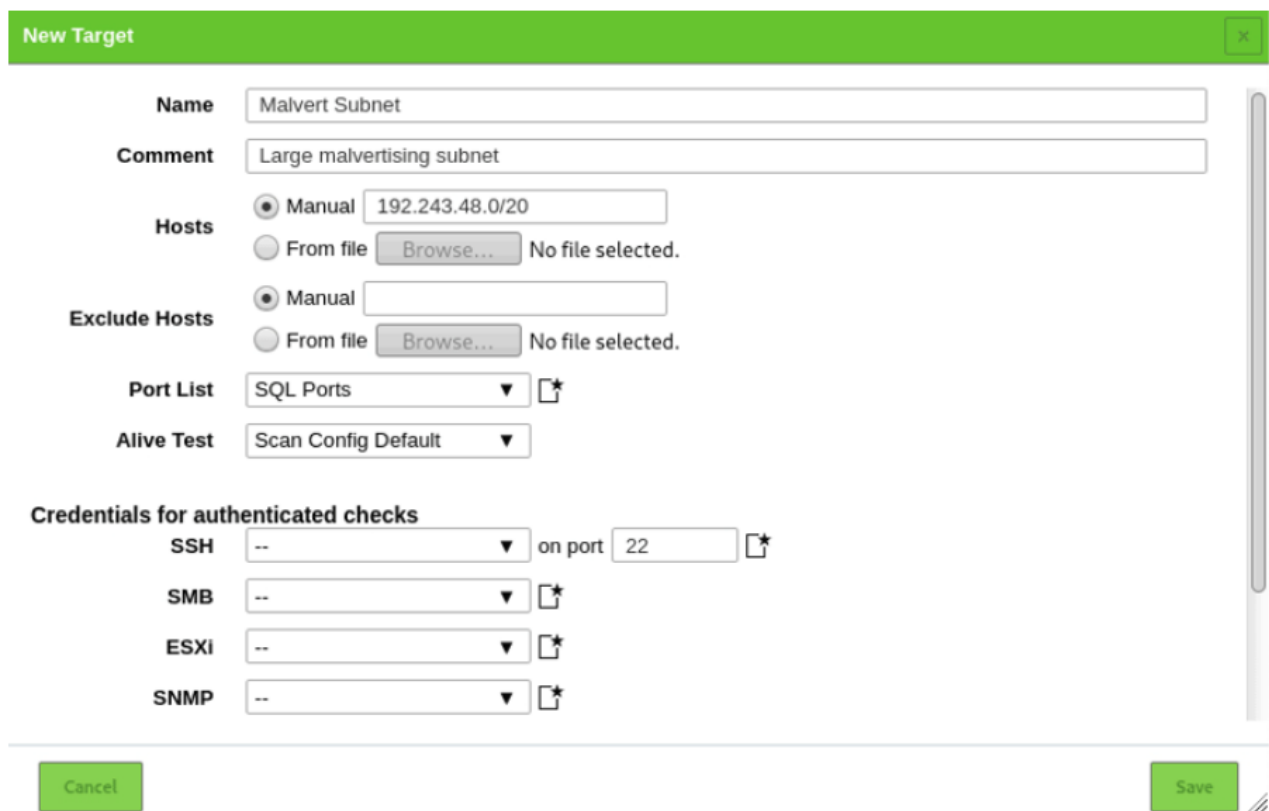
We can click on "New Task" and fill in the details as follows:



The screenshot shows the 'New Task' form in the SecurityTrails interface. The form is titled 'SecurityTrails Basic Test' in the comment field. It includes several configuration options: 'Scan Targets' (greyed out), 'Alerts' (greyed out), 'Schedule' (set to 'Once'), 'Add results to Assets' (radio buttons for 'Yes' and 'No'), 'Apply Overrides' (radio buttons for 'Yes' and 'No'), 'Min QoD' (set to 70%), 'Alterable Task' (radio buttons for 'Yes' and 'No'), 'Auto Delete Reports' (radio buttons for 'Do not automatically delete reports' and 'Automatically delete oldest reports but always keep newest 5 reports'), 'Scanner' (set to 'OpenVAS Default'), and 'Scan Config' (set to 'Full and fast'). At the bottom, there are 'Cancel' and 'Save' buttons.

The "Scan Targets" option is where the IP is added. It is currently greyed out because only existing scans can be selected in the drop-down, but next to it we can create a new target.

Clicking on it, we can fill in the details as follows:



The screenshot shows the 'New Target' form in the SecurityTrails interface. The form is titled 'New Target' in the header. It includes several configuration options: 'Name' (set to 'Malvert Subnet'), 'Comment' (set to 'Large malvertising subnet'), 'Hosts' (radio buttons for 'Manual' and 'From file', with 'Manual' selected and IP '192.243.48.0/20' entered), 'Exclude Hosts' (radio buttons for 'Manual' and 'From file', with 'Manual' selected), 'Port List' (set to 'SQL Ports'), 'Alive Test' (set to 'Scan Config Default'), and 'Credentials for authenticated checks' (radio buttons for 'SSH', 'SMB', 'ESXi', and 'SNMP', with 'SSH' selected and port '22' entered). At the bottom, there are 'Cancel' and 'Save' buttons.

Now we can click on Save, which will display "Malvert1" under the "Scan Targets" option. We can click on Save to save the task. "Once" has been chosen as the schedule option to run the scan only once.

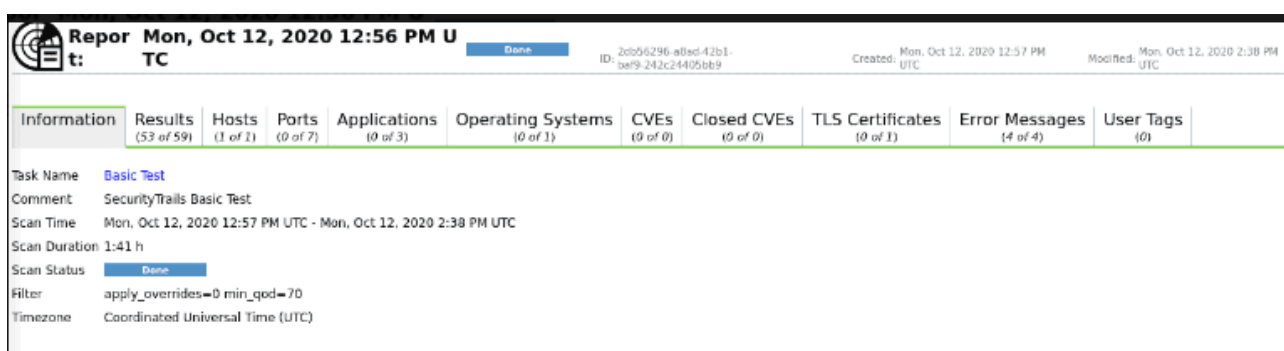
The Schedule option is useful when your scans are targeting your own infrastructure and you want it continuously monitored. The other options on the task have been left as default, as an exercise to see the outcome of the scan.



Name	Status	Reports	Last Report	Severity	Trend	Actions
Basic Test (SecurityTrails Basic Test)	None					

➤ Start scan

The bottom of the Task screen should look like the above. Now we click on the "Start" option to run the scan. The scan should take some time to run, as it looks through multiple threats and scans multiple ports. Once the scan is complete, we can look at the results under:
Scans > Reports.



Information	Results (53 of 59)	Hosts (1 of 1)	Ports (0 of 7)	Applications (0 of 3)	Operating Systems (0 of 1)	CVEs (0 of 0)	Closed CVEs (0 of 0)	TLS Certificates (0 of 1)	Error Messages (4 of 4)	User Tags (0)
<p>Task Name: Basic Test</p> <p>Comment: SecurityTrails Basic Test</p> <p>Scan Time: Mon, Oct 12, 2020 12:57 PM UTC - Mon, Oct 12, 2020 2:38 PM UTC</p> <p>Scan Duration: 1:41 h</p> <p>Scan Status: Done</p> <p>Filter: apply_overrides=0 min_qod=70</p> <p>Timezone: Coordinated Universal Time (UTC)</p>										

➤ Analyzing the results

The Results tab provides us with a broad outlook of what occurred. "Services" checks for web servers running on ports other than 80/443. "CPE Inventory" tells us what software/OS is running on a system. In our case, the results were:

```
192.243.59.20|cpe:/a:nginx:nginx:1.17.9
192.243.59.20|cpe:/a:openbsd:openssh:7.4p1
192.243.59.20|cpe:/a:python:python:2.7.13
192.243.59.20|cpe:/o:debian:debian_linux:9
```

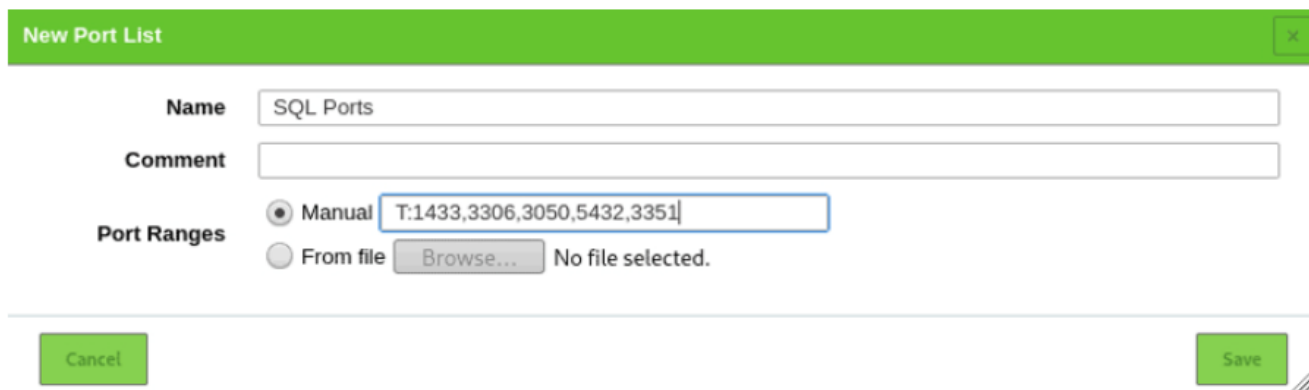
➤ Performing an advanced vulnerability scan

We can now create a more advanced scan by using the different configuration options to add custom details. In this case we'll add custom ports and a larger IP subnet to scan.

First we will add a custom port list. Our targets will simply be the different SQL databases. Here is the full list:

- Microsoft SQL Server: 1433
- MySQL: 3306
- Firebird: 3050
- PostgreSQL: 5432
- Pervasive SQL: 3351

All of these are TCP ports, but the Port Lists option supports both TCP and UDP. Let's add this list under: Configuration > Port Lists:



New Port List

Name: SQL Ports

Comment:

Port Ranges:

- ☒ Manual: T:1433,3306,3050,5432,3351
- ☐ From file: Browse... No file selected.

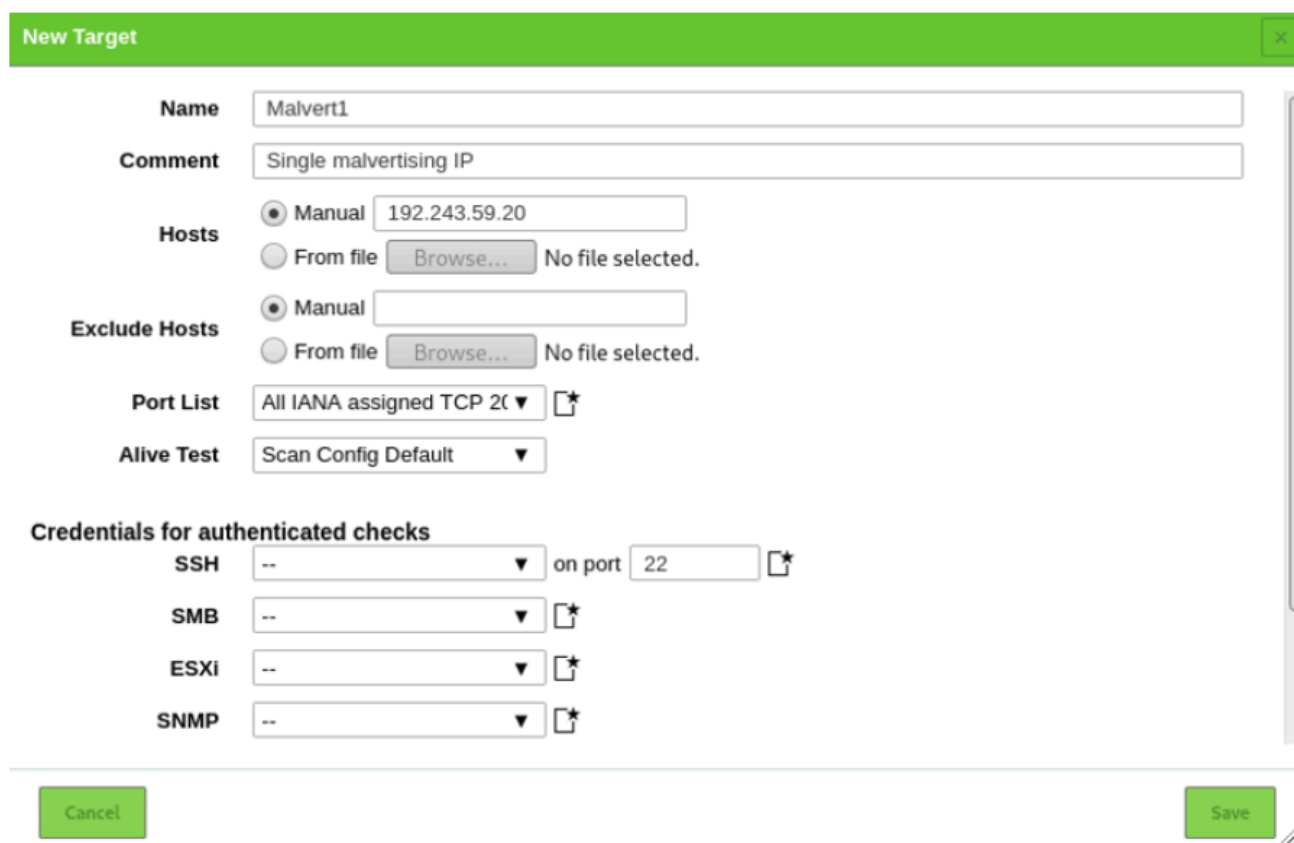
Buttons: Cancel, Save

We expect to see quite different results from the default scan above, by narrowing down our focus to the SQL ports only.

- **Create a new target**
Now we can move to creating a new target, which will be the larger subnet of the malvertising IP mentioned above:

192.243.48.0/20

Under Configuration > Targets, we can add the details of the subnet and our custom SQL Ports port-list:



New Target

Name: Malvert1

Comment: Single malvertising IP

Hosts:

- ☒ Manual: 192.243.59.20
- ☐ From file: Browse... No file selected.

Exclude Hosts:

- ☒ Manual:
- ☐ From file: Browse... No file selected.

Port List: All IANA assigned TCP 20 (dropdown) [icon]

Alive Test: Scan Config Default (dropdown)

Credentials for authenticated checks


SSH	-- (dropdown)	on port	22 (input)	[icon]
SMB	-- (dropdown)			[icon]
ESXi	-- (dropdown)			[icon]
SNMP	-- (dropdown)			[icon]

Buttons: Cancel, Save

- **Other notable features**
The Resilience tab contains some interesting features. Here you can view existing Remediation tickets, and create and view both Compliance Policies and Compliance Audits.
The SecInfo tab is probably one of the most exciting features we discovered when testing OpenVAS. Under the different options like NVT (Network Vulnerability Tests), CVE (Common Vulnerabilities and Exposure), CERT-Bund Advisories and more, you can browse through the listed items and click on each to provide a quick summary of the vulnerability/test/item.

Name	Family	Created	Modified	CVE	Severity	QoD
Huawei EulerOS: Security Advisory for httpd (EulerOS-SA-2020-2175)	Huawei EulerOS Local Security Checks	Mon, Oct 12, 2020 10:21 AM UTC	Mon, Oct 12, 2020 10:35 AM UTC	CVE-2020-9490	5.0 (Medium)	97 %
Huawei EulerOS: Security Advisory for nfs-utils (EulerOS-SA-2020-2164)	Huawei EulerOS Local Security Checks	Mon, Oct 12, 2020 10:21 AM UTC	Mon, Oct 12, 2020 10:35 AM UTC	CVE-2020-14342	5.0 (Medium)	97 %
Huawei EulerOS: Security Advisory for kernel (EulerOS-SA-2020-2166)	Huawei EulerOS Local Security Checks	Mon, Oct 12, 2020 10:21 AM UTC	Mon, Oct 12, 2020 10:35 AM UTC	CVE-2019-0147 CVE-2020-0404 CVE-2020-14314 CVE-2020-14385 CVE-2020-14386 CVE-2020-14390 CVE-2020-25212 CVE-2020-25284 CVE-2020-25285	7.3 (High)	97 %
Huawei EulerOS: Security Advisory for brotli (EulerOS-SA-2020-2163)	Huawei EulerOS Local Security Checks	Mon, Oct 12, 2020 10:21 AM UTC	Mon, Oct 12, 2020 10:35 AM UTC	CVE-2020-8927	5.0 (Medium)	97 %
Huawei EulerOS: Security Advisory for bind (EulerOS-SA-2020-2162)	Huawei EulerOS Local Security Checks	Mon, Oct 12, 2020 10:21 AM UTC	Mon, Oct 12, 2020 10:35 AM UTC	CVE-2020-8622	5.0 (Medium)	97 %
Huawei EulerOS: Security Advisory for libx11 (EulerOS-SA-2020-2167)	Huawei EulerOS Local Security Checks	Mon, Oct 12, 2020 10:21 AM UTC	Mon, Oct 12, 2020 10:35 AM UTC	CVE-2020-14344 CVE-2020-14363	5.0 (Medium)	97 %
Huawei EulerOS: Security Advisory for net-snmp (EulerOS-SA-2020-2169)	Huawei EulerOS Local Security Checks	Mon, Oct 12, 2020 10:21 AM UTC	Mon, Oct 12, 2020 10:35 AM UTC	CVE-2020-15862	7.2 (High)	97 %

The Administration tab also provides a lot of useful functionality if you're running OpenVAS among your DevOps/infosec team. You can add or modify users, groups, roles and permissions. For example, if you needed to create a view-only instance for external auditors to see your investigations after a possible breach, you could configure them as a guest account with viewing permissions only.

Permissions 321 of 321						
Name	Description	Resource Type	Resource	Subject Type	Subject	Actions
authenticate	6b Role Guest may login	Role	Guest	Role	Guest	  
authenticate	6b Role Info may login	Role	Info	Role	Info	  
authenticate	6b Role Monitor may login	Role	Monitor	Role	Monitor	  
authenticate	6b Role Observer may login	Role	Observer	Role	Observer	  
authenticate	6b Role User may login	Role	User	Role	User	  
commands	6b Role Info may run multiple GMP commands in one	Role	Info	Role	Info	  
commands	6b Role Guest may run multiple GMP commands in one	Role	Guest	Role	Guest	  
commands	6b Role Monitor may run multiple GMP commands in one	Role	Monitor	Role	Monitor	  
commands	6b Role User may run multiple GMP commands in one	Role	User	Role	User	  
create_agent	6b Role User may create a new Agent	Role	User	Role	User	  

On each page/tab there is a button that links you to the documentation regarding that section. This is very informative as it allows you to quickly understand the purpose of each part of OpenVAS.

Conclusion

In conclusion, this project demonstrated the use of Open Source Software to enhance the security of a Windows operating system. OpenVAS software was used to scan the system and identify missing security updates, and free system information tools were used to gather information about the hardware and software installed on the system. The analysis report provided recommendations for addressing the security vulnerabilities identified during the scan. By implementing these recommendations, the security of the Windows operating system can be significantly improved.

References/Bibliography

- OpenVAS, "OpenVAS - Vulnerability Scanning Framework," OpenVAS, 2021. [Online]. Available: <https://www.openvas.org/>.
- "HWiNFO - Hardware Information, Analysis and Monitoring Tools," HWiNFO, 2021. [Online]. Available: <https://www.hwinfo.com/>.
- "Speccy - System Information - Free Download," Speccy, 2021. [Online]. Available: <https://www.ccleaner.com/speccy>.
- Microsoft, "Windows 10 Pro Specifications," Microsoft, 2021. [Online]. Available: <https://www.microsoft.com/en-us/windows/windows-10-specifications>.
- Microsoft, "Windows 10 Security," Microsoft, 2021. [Online]. Available: <https://www.microsoft.com/en-us/windows/comprehensive-security>.
- "NIST National Vulnerability Database," NIST, 2021. [Online]. Available: <https://nvd.nist.gov/>.