

Module 5 : Vulnerability Analysis

Lab 1: Perform Vulnerability Research with Vulnerability Scoring Systems and Databases

Lab Scenario

As a professional ethical hacker or pen tester, your first step is to search for vulnerabilities in the target system or network using vulnerability scoring systems and databases. Vulnerability research provides awareness of advanced techniques to identify flaws or loopholes in the software that could be exploited. Using this information, you can use various tricks and techniques to launch attacks on the target system.

Lab Objectives

- Perform vulnerability research in Common Weakness Enumeration (CWE)

Overview of Vulnerabilities in Vulnerability Scoring Systems and Databases

Vulnerability databases collect and maintain information about various vulnerabilities present in the information systems.

The following are some of the vulnerability scoring systems and databases:

- Common Weakness Enumeration (CWE)
- Common Vulnerabilities and Exposures (CVE)
- National Vulnerability Database (NVD)

Task 1: Perform Vulnerability Research in Common Weakness Enumeration (CWE)

Common Weakness Enumeration (CWE) is a category system for software vulnerabilities and weaknesses. It has numerous categories of weaknesses that means that CWE can be effectively employed by the community as a baseline for weakness identification, mitigation, and prevention efforts. Further, CWE has an advanced search technique with which you can search and view the weaknesses based on research concepts, development concepts, and architectural concepts.

Here, we will use CWE to view the latest underlying system vulnerabilities.

1. By default, **Windows 11** machine is selected, click Ctrl+Alt+Delete to activate the machine and login with **Admin/Pa\$\$w0rd**.

Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.

2. Launch any web browser, and go to **https://cwe.mitre.org/** website (here, we are using **Mozilla Firefox**).

If the **Default Browser** pop-up window appears, uncheck the **Always perform this check when starting Firefox** checkbox and click the **Not now** button.

If a **New in Firefox: Content Blocking** pop-up window appears, follow the step and click start browsing to finish viewing the information.

3. **CWE** website appears. Navigate to **Search** tab, in the **Google Custom Search** under **CWE List Quick Access** section and search for **SMB** in the search field.

Here, we are searching for the vulnerabilities of the running services that were found in the target systems in previous module labs (Module 04 Enumeration).

CWE Common Weakness Enumeration
A community-developed list of SW & HW weaknesses that can become vulnerabilities

Top 25 Top HW CWE New to CWE? Start here!

Home | About | CWE List | Mapping | Top-N Lists | Community | News

Search

CWE Top 10 KEV Weaknesses

This [list](#) identifies the top ten CWEs in the Cybersecurity and Infrastructure Security Agency's (CISA) "Known Exploited Vulnerabilities (KEV) Catalog," a database of security flaws in software applications that have been exposed and leveraged by attackers. Our analysis/key insights about the list are available [here](#), and our methodology for creating the list is [here](#).

CWE List Quick Access

Search CWE

View CWEs by

Software Development

Hardware Design

Community Engagement

Hardware CWE Special Interest Group [Join HW CWE SIG](#)

ICS/OT Special Interest Group [Join ICS/OT SIG](#)

REST API Working Group [Join REST API WG](#)

User Experience Working Group [Join UE WG](#)

CWE/CAPEC Board [Read meeting minutes](#)

CWE News

News [CWE Version 4.14 Now Available](#)

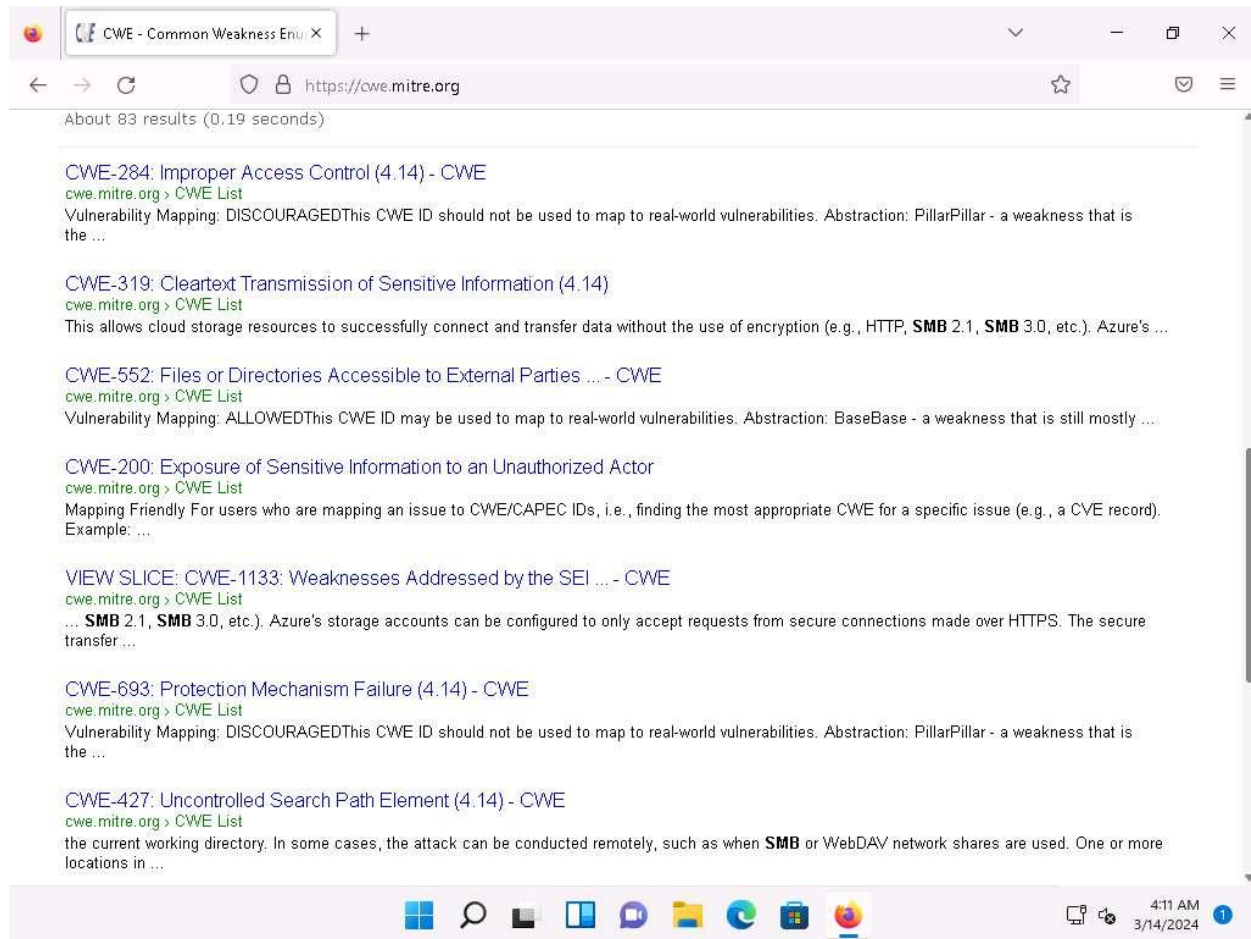
Podcast [Red Hat's CWE Journey](#)

News ["2023 CWE Top 10 KEV Weaknesses" List Now Available](#)

News [Follow CWE on Mastodon!](#)

- The search results appear, scroll-down to view the underlying vulnerabilities in the target service (here, **SMB**). You can click any link to view detailed information on the vulnerability.

The search results might differ when you perform this task

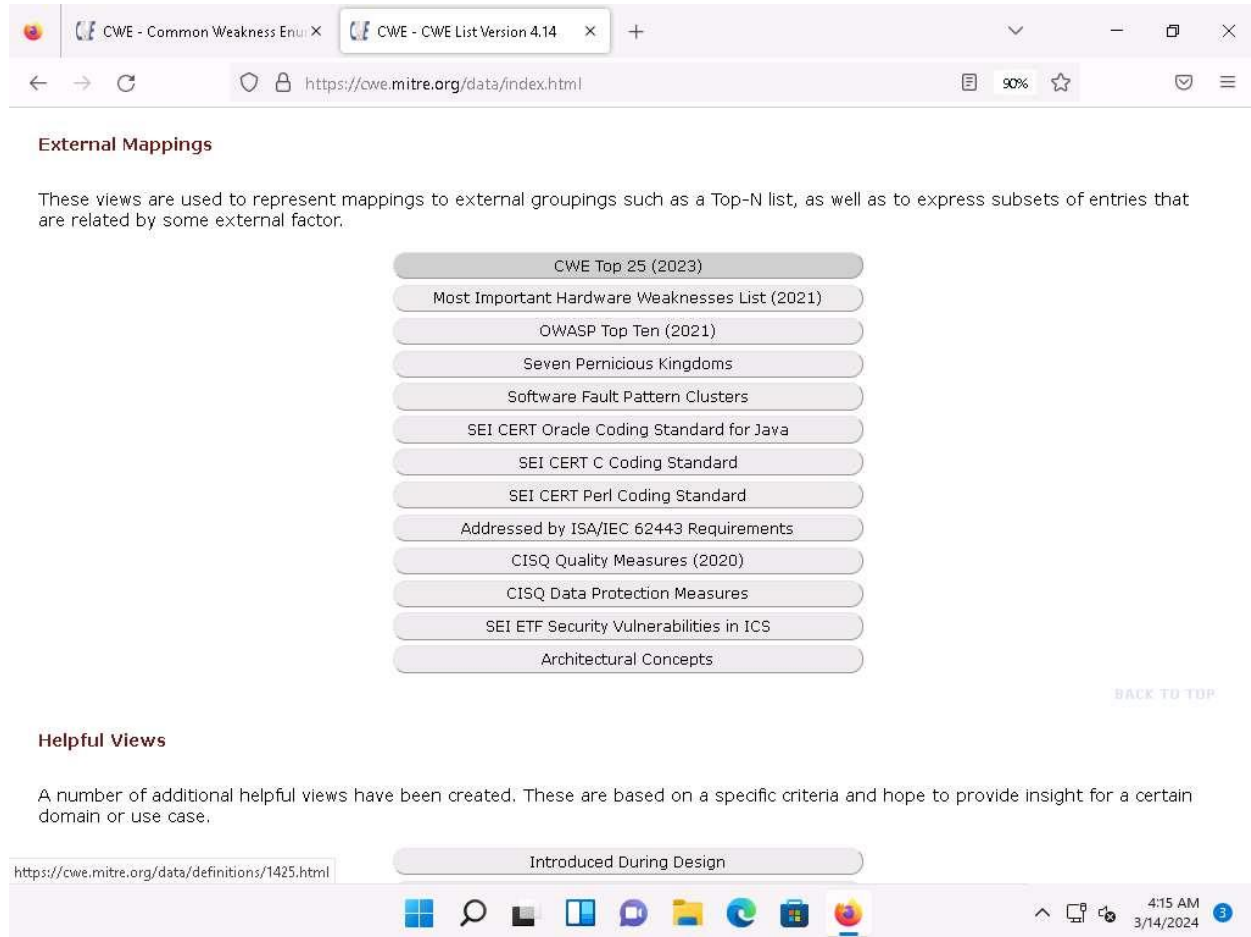


5. Now, click any link (here, **CWE-284**) to view detailed information about the vulnerability.

The screenshot shows a web browser window with two tabs: 'CWE - Common Weakness Enum...' and 'CWE - CWE-284: Improper Acc...'. The address bar shows the URL 'https://cwe.mitre.org/data/definitions/284.html'. The page header features the 'CWE Common Weakness Enumeration' logo, a tagline 'A community-developed list of SW & HW weaknesses that can become vulnerabilities', and two circular badges: 'Top 25' and 'Top HW CWE'. A 'New to CWE? Start here!' link is also present. Below the header is a navigation bar with links: Home, About, CWE List, Mapping, Top-N Lists, Community, News, and Search. The main content area is titled 'CWE-284: Improper Access Control'. It includes a 'Weakness ID: 284', 'Vulnerability Mapping: DISCOURAGED', and 'Abstraction: Pillar'. A 'View customized information:' section has buttons for 'Conceptual', 'Operational', 'Mapping Friendly', 'Complete' (selected), and 'Custom'. The 'Description' section states: 'The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.' The 'Extended Description' section explains that access control involves several protection mechanisms: Authentication, Authorization, and Accountability. It further details that when a mechanism fails, attackers can compromise security by gaining privileges, reading sensitive information, or evading detection. Two distinct behaviors are identified: Specification (incorrect privileges, permissions, ownership, etc.) and Enforcement (mechanism contains errors that prevent it from properly enforcing the specified access control requirements).

6. Similarly, you can click on other vulnerabilities and view detailed information.
7. Now, navigate to the **CWE List** tab. **CWE List Version** will be displayed. Scroll down, and under the **External Mappings** section, select **CWE Top 25 (2023)**.

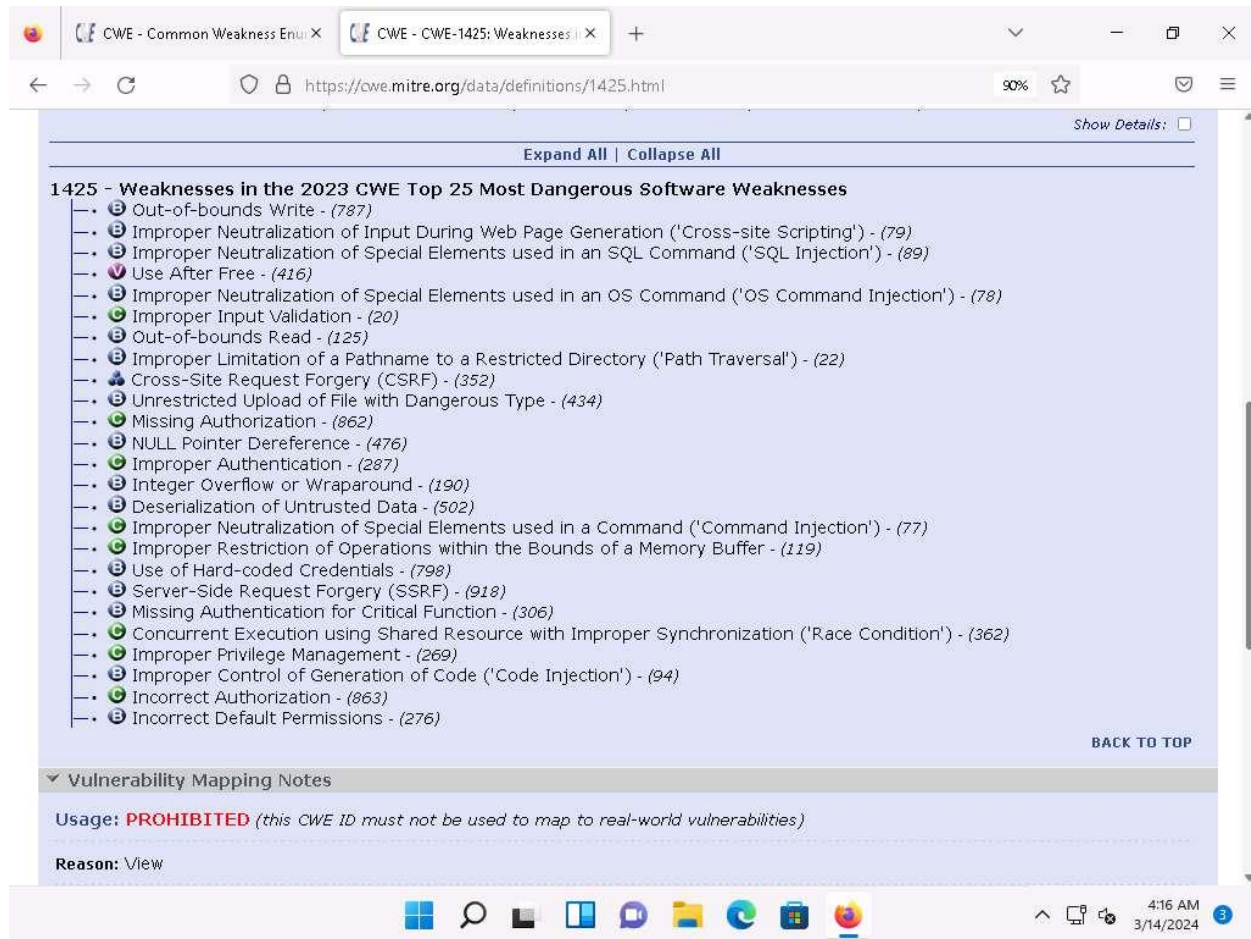
The result might differ when you perform this task.



8. A webpage appears, displaying **CWE VIEW: Weaknesses in the 2023 CWE Top 25 Most Dangerous Software Weaknesses**. Scroll down and view a list of **Weaknesses in the 2023 CWE Top 25 Most Dangerous Software Weaknesses** under the **Relationships** section. You can check each weakness to view detailed information on it.

This information can be used to exploit the vulnerabilities in the software and further launch attacks.

The result showing publishing year might differ when you perform this task.



9. Similarly, you can go back to the CWE website and explore other options, as well.
10. Attacker can find vulnerabilities on the services running on the target systems and further exploit them to launch attacks.
11. This concludes the demonstration of checking vulnerabilities in the Common Weakness Enumeration (CWE).
12. Close all open windows and document all the acquired information.

Question 5.1.1.1

Search the Common Weakness Enumeration (CWE) list and find the name of the vulnerability with the CWE ID 591.

Question 5.1.1.2

Search the Common Weakness Enumeration (CWE) list and find the top weakness in the list “Weaknesses in the 2023 CWE Top 25 Most Dangerous Software Weakness.”

Lab 2: Perform Vulnerability Assessment using Various Vulnerability Assessment Tools

Lab Scenario

The information gathered in the previous labs might not be sufficient to reveal potential vulnerabilities of the target: there could be more information available that may help in finding loopholes. As an ethical hacker, you should look for as much information as possible using all available tools. This lab will demonstrate other information that you can extract from the target using various vulnerability assessment tools.

Lab Objectives

- Perform vulnerability analysis using OpenVAS

Overview of Vulnerability Assessment

A vulnerability assessment is an in-depth examination of the ability of a system or application, including current security procedures and controls, to withstand exploitation. It scans networks for known security weaknesses, and recognizes, measures, and classifies security vulnerabilities in computer systems, networks, and communication channels. It identifies, quantifies, and ranks possible vulnerabilities to threats in a system. Additionally, it assists security professionals in securing the network by identifying security loopholes or vulnerabilities in the current security mechanism before attackers can exploit them.

There are two approaches to network vulnerability scanning:

- Active Scanning
- Passive Scanning

Task 1: Perform Vulnerability Analysis using OpenVAS

OpenVAS is a framework of several services and tools offering a comprehensive and powerful vulnerability scanning and vulnerability management solution. Its capabilities include

unauthenticated testing, authenticated testing, various high level and low-level Internet and industrial protocols, performance tuning for large-scale scans, and a powerful internal programming language to implement any vulnerability test. The actual security scanner is accompanied with a regularly updated feed of Network Vulnerability Tests (NVTs)-over 50,000 in total.

Here, we will perform a vulnerability analysis using OpenVAS.

In this task, we will use the **Parrot Security (10.10.1.13)** machine as a host machine and the **Windows Server 2022 (10.10.1.22)** machine as a target machine.

1. Click on Parrot Security to switch to the **Parrot Security** machine and login with **attacker/toor**.

If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.

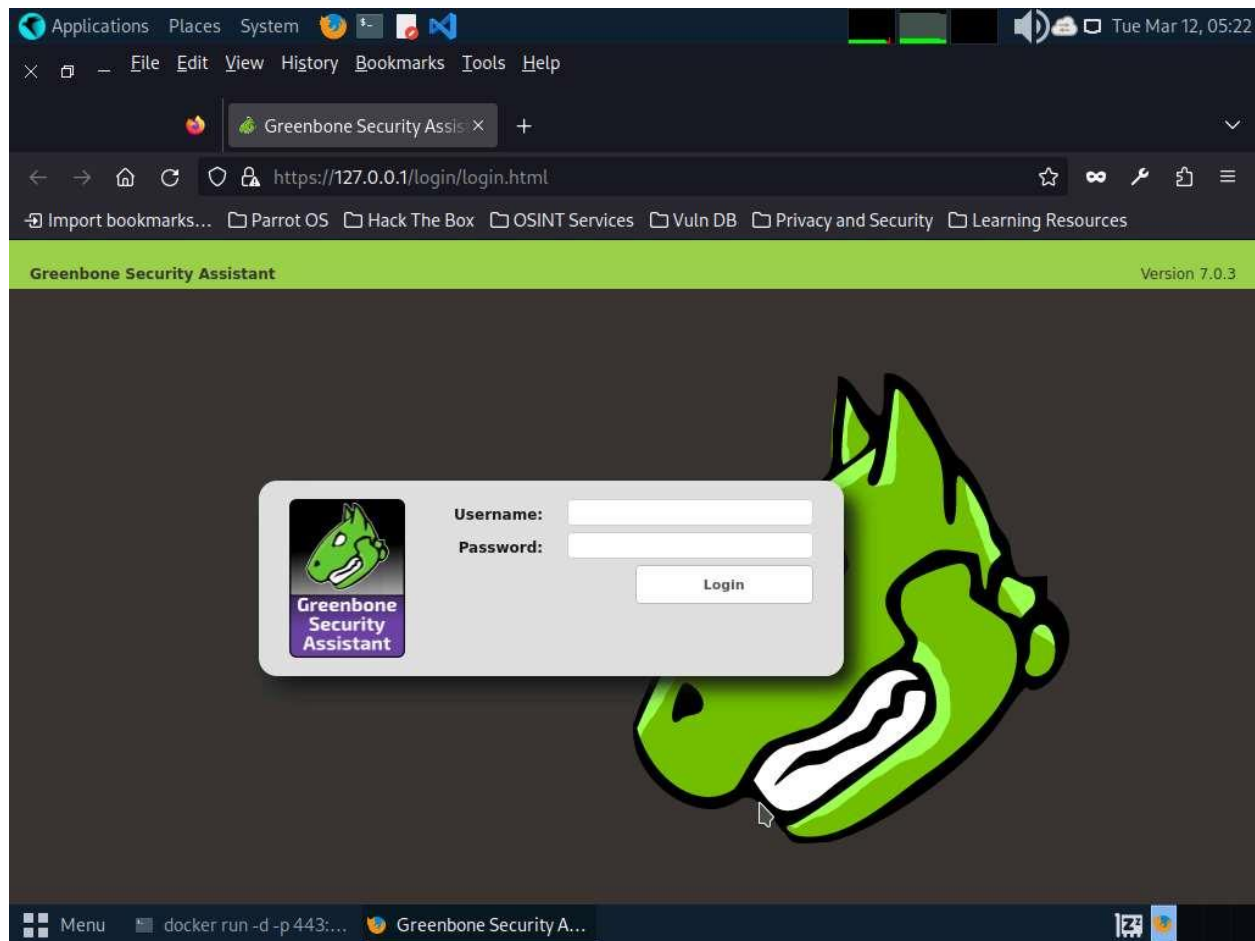
If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

2. Open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).

The password that you type will not be visible.

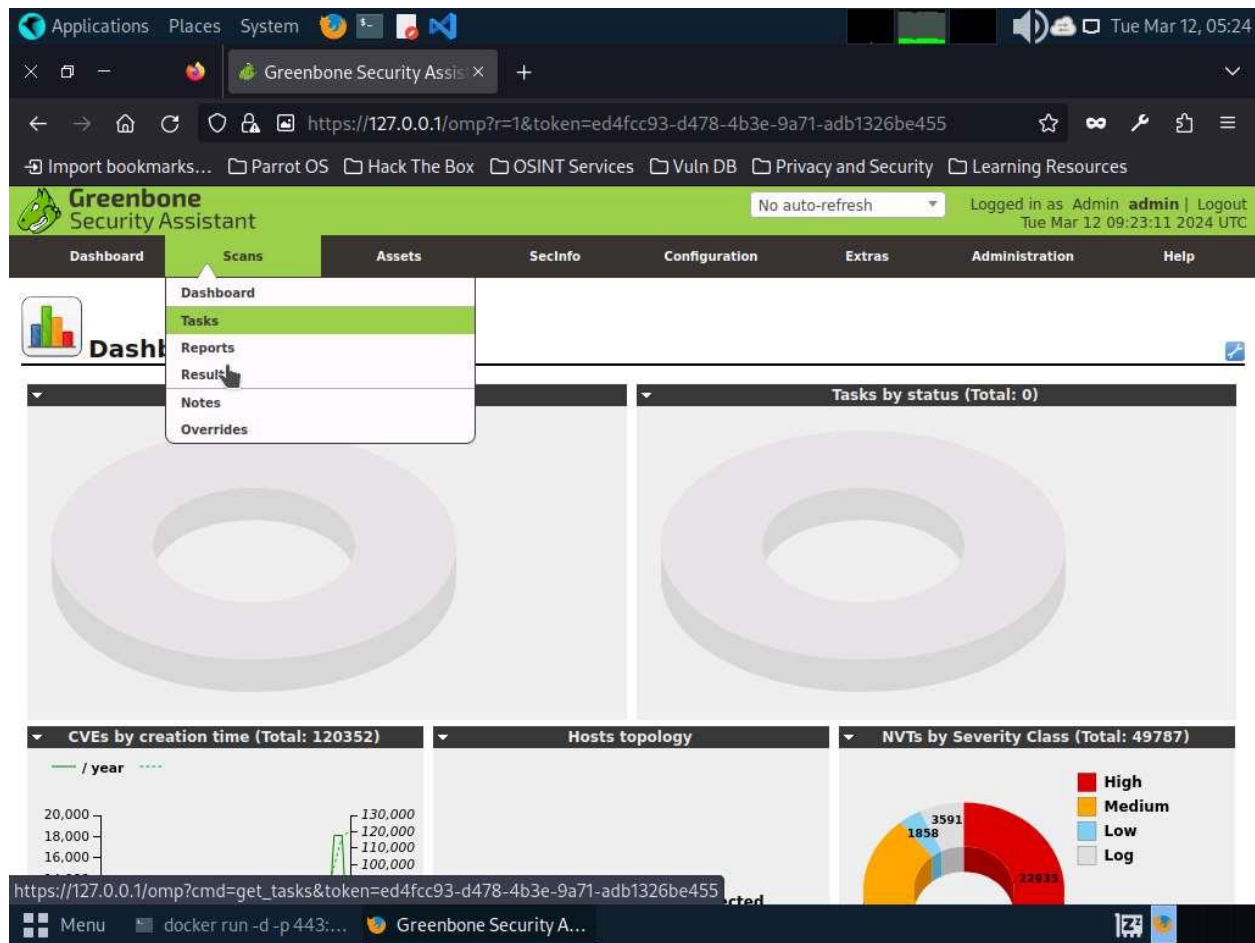
3. Run **docker run -d -p 443:443 --name openvas mikesplain/openvas** command to launch OpenVAS.
4. After the tool initializes, click **Firefox** icon from the top-section of the **Desktop**.
5. The **Firefox** browser appears, go to **https://127.0.0.1/**. OpenVAS login page appears, log in with **admin/admin**.

If a **Warning** page appears, click **Advanced** and select **Accept the Risk and Continue**.



6. The **OpenVAS Dashboards** appears. Navigate to **Scans --> Tasks** from the **Menu** bar.

If a **Welcome to the scan task management!** pop-up appears, close it.



7. Hover over wand icon and click the **Task Wizard** option.

Applications Places System Tue Mar 12, 05:24

Greenbone Security Assistant

https://127.0.0.1/omp?cmd=get_tasks&token=ed4fcc93-d478-4b3e-9a71-adb1326be455

Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources

Greenbone Security Assistant No auto-refresh Logged in as Admin admin | Logout Tue Mar 12 09:24:24 2024 UTC

Dashboard Scans Assets Secinfo Configuration Extras Administration Help

Task Wizard
Advanced Task Wizard
Modify Task Wizard

Filter: min_qod=70 apply_overrides=1 rows=10 first=1 sort=name

Tasks (0 of 0)

Tasks by Severity Class (Total: 0)

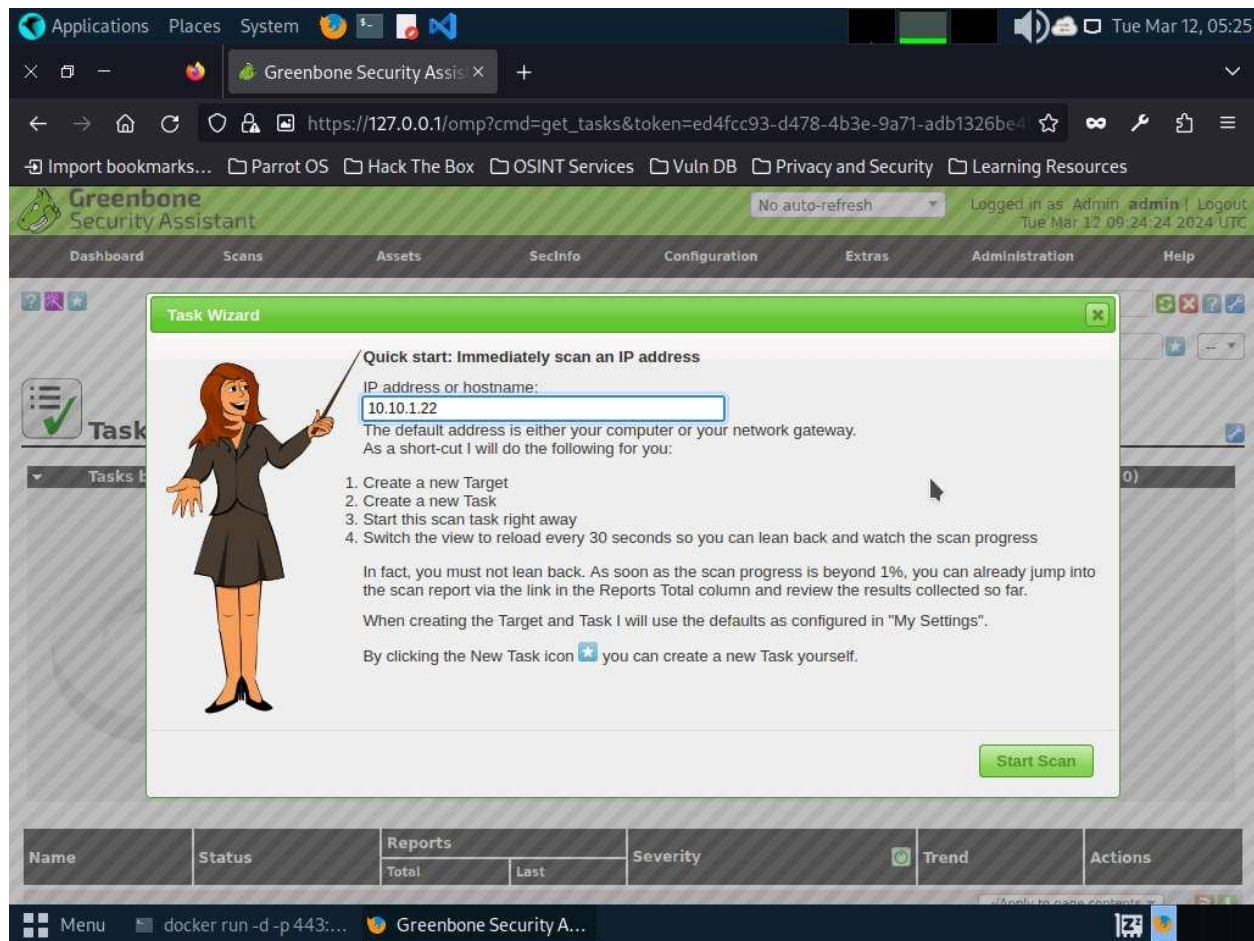
Tasks with most High results per host
No Tasks with High severity found

Tasks by status (Total: 0)

Name	Status	Reports		Severity	Trend	Actions
		Total	Last			
https://127.0.0.1/omp?cmd=wizard&name=quick_first_scan&filter=&filt_id=&token=ed4fcc93-d478-4b3e-9a71-adb1326be455						

Menu docker run -d -p 443:... Greenbone Security A...

- The **Task Wizard** window appears; enter the target IP address in the **IP address or hostname** field (here, the target system is **Windows Server 2022 [10.10.1.22]**) and click the **Start Scan** button.



9. The task appears under the **Tasks** section; OpenVAS starts scanning the target IP address.
10. Wait for the **Status** to change from **Requested** to **Done**. Once it is completed, click the **Done** button under the **Status** column to view the vulnerabilities found in the target system.

It takes approximately 20 minutes for the scan to complete.

If you are logged out of the session then login again using credentials **admin/admin**.

Applications Places System Tue Mar 12, 06:19

Greenbone Security Assis x +

https://127.0.0.1/omp?cmd=get_tasks&token=521d747f-cb51-46e8-abbe-837c64e84e... ☆ ∞ 🔑 ☰

Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources

Dashboard Scans Assets SecInfo Configuration Extras Administration Help

Tasks (1 of 1)

Tasks by Severity Class (Total: 1)

Medium

1

Tasks with most High results per host

No Tasks with High severity found

Tasks by status (Total: 1)

Done

1

Name	Status	Reports		Severity	Trend	Actions
		Total	Last			
Immediate scan of IP 10.10.1.22	Done	1 (1)	Mar 12 2024	9.0 (Medium)		

(Applied filter: min_qod=70 apply_overrides=1 rows=10 first=1 sort=name)

Backend operation: 0.01s Greenbone Security Assistant (GSA) Copyright 2009 - 2018 by Greenbone Networks GmbH, www.greenbone.net

Menu docker run -d -p 443:... Greenbone Security A...

11. **Report: Results** appear, displaying the discovered vulnerabilities along with their severity and port numbers on which they are running.

The results might differ when you perform this task.

Applications Places System

Greenbone Security Assistant

https://127.0.0.1/omp?cmd=get_report&report_id=cd423c0b-daa4-4885-b01c-e30452f57...

Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources

Greenbone Security Assistant

Logged in as Admin admin | Logout 09:51:48 2024 UTC

Dashboard Scans Assets Secinfo Configuration Extras Administration Help

Anonymous XML Done

Filter: autofp=0 apply_overrides=1 notes=1 overrides=1 result_hosts_only=1 first=1 rows=100 sort-reverse=severity levels=hml min_qod=70

Report: Results (2 of 61)

ID: cd423c0b-daa4-4885-b01c-e30452f579de
Modified: Mon Jul 8 09:45:25 2024
Created: Mon Jul 8 09:45:25 2024
Owner: admin

Vulnerability	Severity	QoD	Host	Location	Actions
DCE/RPC and MSRPC Services Enumeration Reporting	5.0 (Medium)	80%	10.10.1.22	135/tcp	
TCP timestamps	2.6 (Low)	80%	10.10.1.22	general/tcp	

(Applied filter: autofp=0 apply_overrides=1 notes=1 overrides=1 result_hosts_only=1 first=1 rows=100 sort-reverse=severity levels=hml min_qod=70)

Backend operation: 0.40s

Greenbone Security Assistant (GSA) Copyright 2009 - 2018 by Greenbone Networks GmbH, www.greenbone.net

Menu docker run -d -p 443:... Greenbone Security A...

12. Click on any vulnerability under the **Vulnerability** column to view its detailed information.
13. Detailed information regarding selected vulnerability appears, as shown in the screenshot.

The screenshot shows the Greenbone Security Assistant web interface. The browser address bar displays the URL: `https://127.0.0.1/omp?cmd=get_result&result_id=373743ce-f786-4204-bcae-92b19420f49e`. The interface includes a navigation bar with tabs: Dashboard, Scans, Assets, Secinfo, Configuration, Extras, Administration, and Help. The user is logged in as 'Admin'.

The main content area displays a vulnerability report titled "Result: DCE/RPC and MSRPC Services Enumeration Reporting". The report details are as follows:

Vulnerability	Severity	QoD	Host	Location	Actions
DCE/RPC and MSRPC Services Enumeration Reporting	5.0 (Medium)	80%	10.10.1.22	135/tcp	[Icons]

Summary
Distributed Computing Environment / Remote Procedure Calls (DCE/RPC) or MSRPC services running on the remote host can be enumerated by connecting on port 135 and doing the appropriate queries.

Vulnerability Detection Result
Here is the list of DCE/RPC or MSRPC services running on this host via the TCP protocol:

Port: 2103/tcp

```

UUID: 1088a980-eae5-11d0-8d9b-00a02453c337, version 1
Endpoint: ncacn_ip_tcp:10.10.1.22[2103]
Annotation: Message Queuing - QM2QM V1

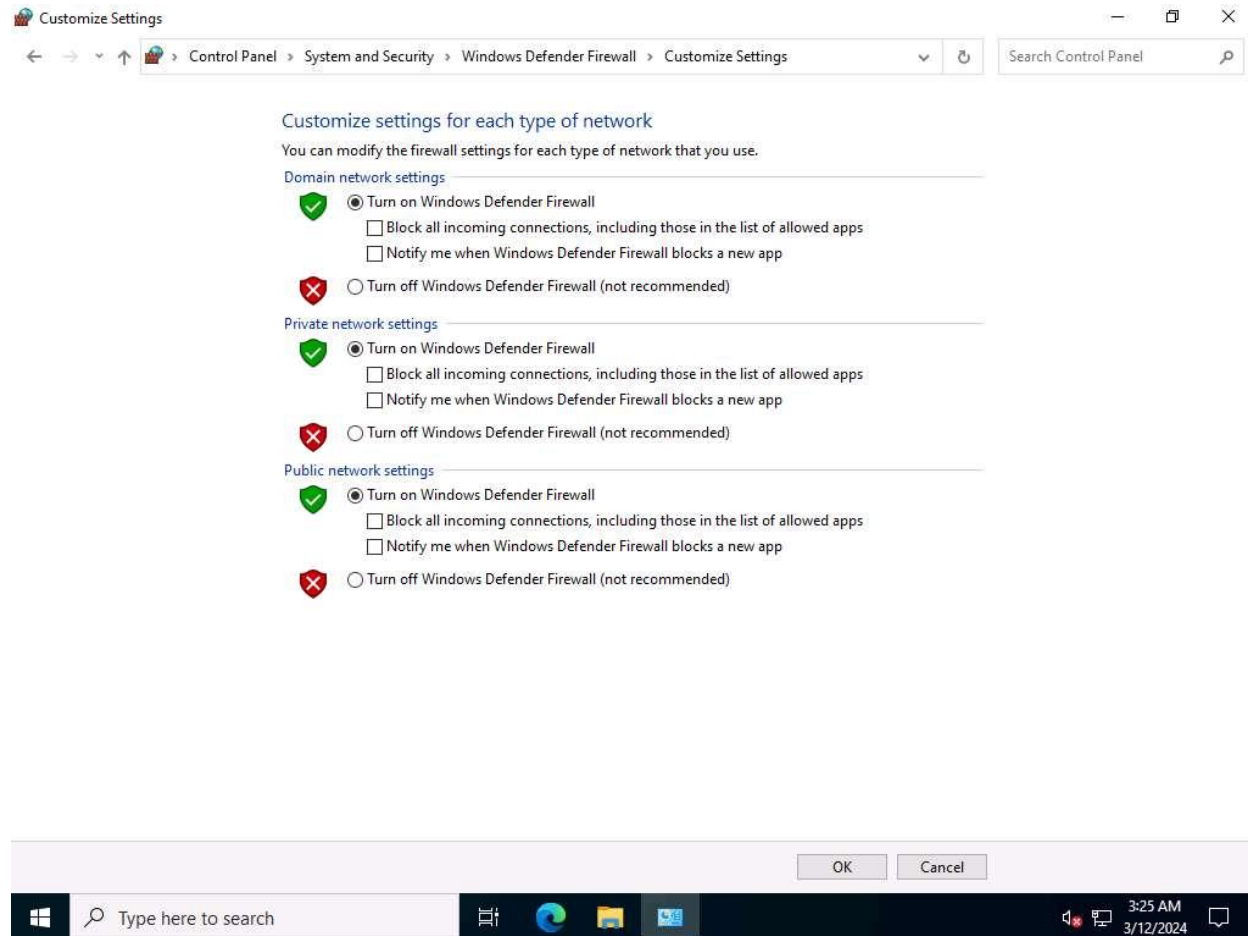
UUID: 1a9134dd-7b39-45ba-ad88-44d01ca47f28, version 1
Endpoint: ncacn_ip_tcp:10.10.1.22[2103]
Annotation: Message Queuing - RemoteRead V1

```

14. Similarly, you can check other Reports by hovering over the **Report: Results** section to view other Reports regarding the vulnerabilities in the target system.
15. Next, go through the findings, including all high or critical vulnerabilities. Manually use your skills to verify the vulnerability. The challenge with vulnerability scanners is that they are quite limited; they work well for an internal or white box test only if the credentials are known. We will explore that now: return to your OpenVAS tool, and set up for the same scan again; but this time, turn your **firewall ON** in the **Windows Server 2022** machine.
16. Now, we will enable **Windows Firewall** in the target system and scan it for vulnerabilities.
17. Click on Windows Server 2022 to switch to the **Windows Server 2022** machine and click Ctrl+Alt+Delete and login with **CEH\Administrator / Pa\$\$w0rd**.

18. Navigate to **Control Panel --> System and Security --> Windows Defender Firewall --> Turn Windows Defender Firewall on or off, enable Windows Firewall**, and click **OK**.

By turning the Firewall ON, you are making it more difficult for the scanning tool to scan for vulnerabilities in the target system.



19. Click on Parrot Security to switch to **Parrot Security** machine and perform **Steps# 7-9** to create another task for scanning the target system.
20. A newly created task appears under the **Tasks** section and starts scanning the target system for vulnerabilities.
21. After the completion of the scan, click the **Done** button under the **Status** column.
- It takes approximately 15-20 minutes for the scan to complete.
22. **Report: Results** appears, displaying the discovered vulnerabilities along with their severity and port numbers on which they are running.

The results might differ when you perform this task.

Applications Places System

Greenbone Security Assistant

https://127.0.0.1/omp?cmd=get_report&report_id=8a9680fe-5bb8-4207-9631-8d009f8bf273

Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources

Greenbone Security Assistant

Logged in as Admin **admin** | Logout 10:10:07 2024 UTC

Dashboard Scans Assets Secinfo Configuration Extras Administration Help

Anonymous XML Done

Filter: autofp=0 apply_overrides=1 notes=1 overrides=1 result_hosts_only=1 first=1 rows=100 sort-reverse=severity levels=hml min_qod=70

Report: Results (2 of 43)

ID: 8a9680fe-5bb8-4207-9631-8d009f8bf273
Modified:
Created:
Owner: admin

Vulnerability	Severity	QoD	Host	Location	Actions
DCE/RPC and MSRPC Services Enumeration Reporting	5.0 (Medium)	80%	10.10.1.22	135/tcp	
TCP timestamps	2.6 (Low)	80%	10.10.1.22	general/tcp	

(Applied filter: autofp=0 apply_overrides=1 notes=1 overrides=1 result_hosts_only=1 first=1 rows=100 sort-reverse=severity levels=hml min_qod=70)

Backend operation: 0.40s

Greenbone Security Assistant (GSA) Copyright 2009 - 2018 by Greenbone Networks GmbH, www.greenbone.net

Menu docker run -d -p 443:443 Greenbone Security Assistant

23. The scan results for the target machine before and after the Windows Firewall was enabled are the same, thereby indicating that the target system is vulnerable to attack even if the Firewall is enabled.
24. This concludes the demonstration performing vulnerabilities analysis using OpenVAS.
25. Close all open windows and document all the acquired information.
26. Click on Windows Server 2022 to switch to the **Windows Server 2022** machine and click Ctrl+Alt+Delete login with **Administrator/Pa\$\$w0rd**.
27. Navigate to **Control Panel --> System and Security --> Windows Defender Firewall --> Turn Windows Defender Firewall on or off**, disable Windows Firewall, and click **OK**.

Question 5.2.1.1

Perform vulnerability analysis for the target machine (10.10.1.22) using OpenVAS and find the number of vulnerabilities in the system. Enter the Severity level of the DCE/RPC and MSRPC Services Enumeration Reporting vulnerability.

Lab 3: Perform Vulnerability Analysis using AI

Lab Scenario

As a professional ethical hacker or pen tester, you must acknowledge the limitations of conventional approaches in revealing all potential vulnerabilities. Therefore, you will utilize AI-driven vulnerability analysis tools to identify and assess security weaknesses in a simulated network environment.

Lab Objectives

- Perform vulnerability analysis using ShellGPT

Overview of vulnerability analysis using AI

Vulnerability Analysis with AI employs advanced algorithms to unearth hidden security flaws in networks. AI-driven tools extract comprehensive data, prioritize risks, and fortify defenses, empowering ethical hackers to anticipate and mitigate emerging threats effectively. This innovative approach enhances cybersecurity readiness by leveraging AI's precision and adaptability.

Task 1: Perform Vulnerability Analysis using ShellGPT

ShellGPT swiftly interprets and executes commands, conducting scans, identifying weaknesses, and suggesting mitigation strategies in real-time. Its adaptive nature facilitates dynamic navigation through complex systems, enhancing efficiency and precision in vulnerability analysis. By integrating ShellGPT, you can gain a powerful ally in their quest to safeguard digital ecosystems, leveraging AI's capabilities to uncover and address security risks with unparalleled speed and accuracy.

Here, we will use ShellGPT to discover potential vulnerabilities in the target.

The commands generated by ShellGPT may vary depending on the prompt used and the tools available on the machine. Due to these variables, the output generated by ShellGPT might differ from what is shown in the screenshots. These differences arise from the dynamic nature of the

AI's processing and the diverse environments in which it operates. As a result, you may observe differences in command syntax, execution, and results while performing this lab task.

1. Click Parrot Security to switch to Parrot machine, and login with **attacker/toor**. Open a Terminal window and execute **sudo su** to run the program as a root user (When prompted, enter the password **toor**).

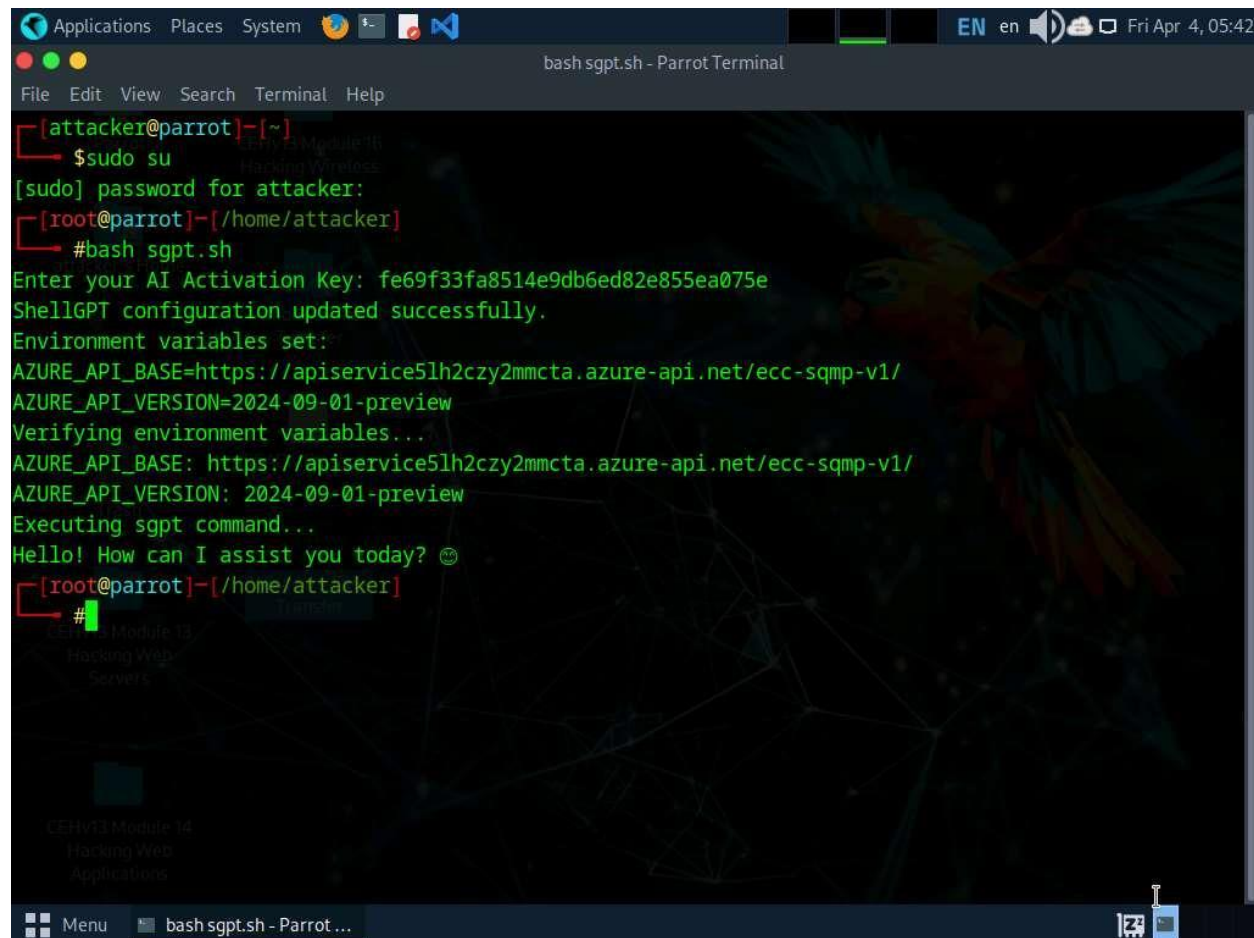
The password that you type will not be visible.

2. Run **bash sgpt.sh** command to configure ShellGPT and the AI activation key.

You can follow the **Instructions to Download your AI Activation**

Key in **Module 00: CEH Lab Setup** to obtain the AI activation key.

Alternatively, follow the instructions available in the file, Instructions to Download your AI Activation Key - CEHv13.

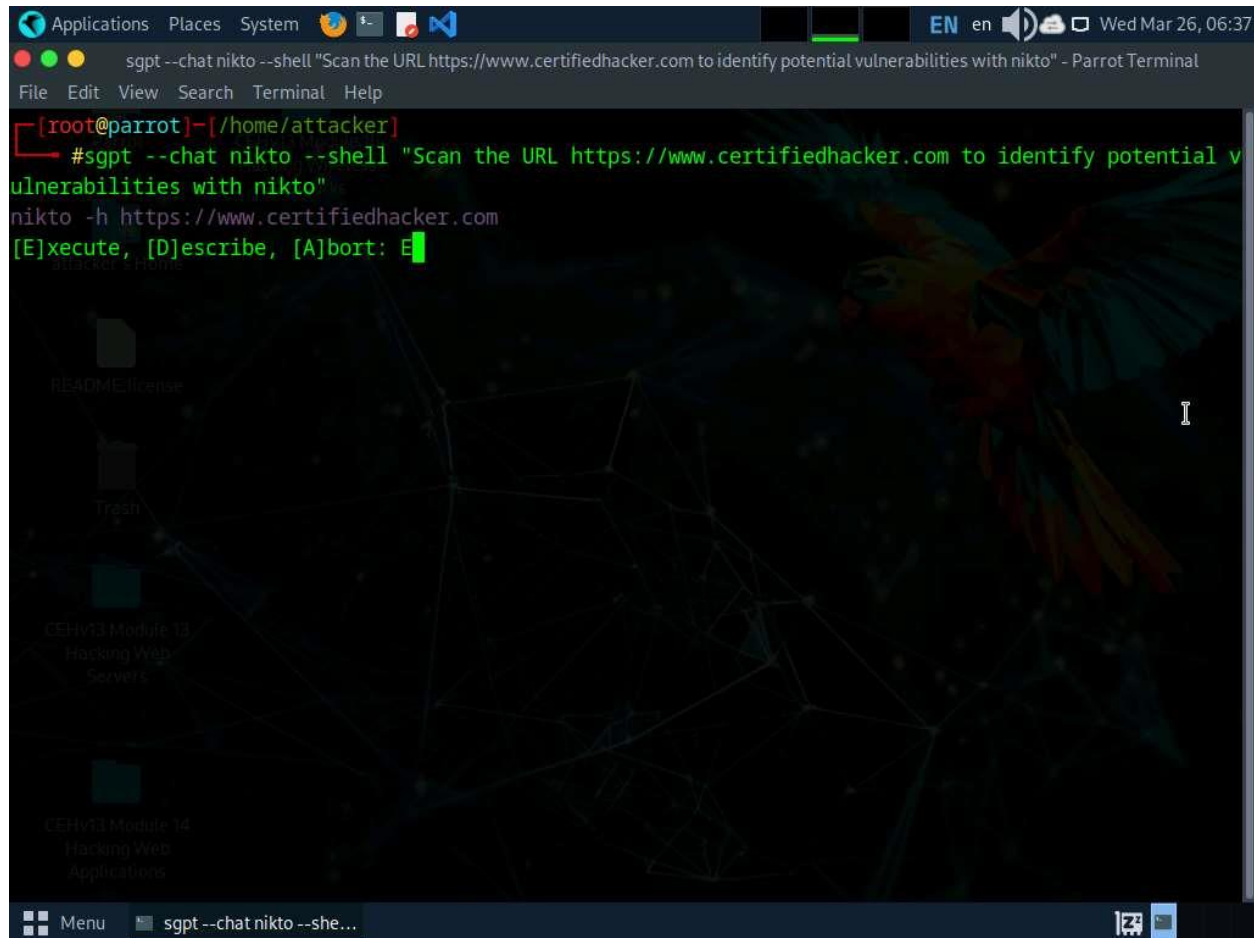


```
Applications Places System bash sgpt.sh - Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~/home/attacker# bash sgpt.sh
Enter your AI Activation Key: fe69f33fa8514e9db6ed82e855ea075e
ShellGPT configuration updated successfully.
Environment variables set:
AZURE_API_BASE=https://apiservice5lh2czy2mmcta.azure-api.net/ecc-sqmp-v1/
AZURE_API_VERSION=2024-09-01-preview
Verifying environment variables...
AZURE_API_BASE: https://apiservice5lh2czy2mmcta.azure-api.net/ecc-sqmp-v1/
AZURE_API_VERSION: 2024-09-01-preview
Executing sgpt command...
Hello! How can I assist you today? 😊
[root@parrot]~/home/attacker#
```

3. After configuring the ShellGPT in Parrot Security machine, in the terminal window, run ****sgpt**

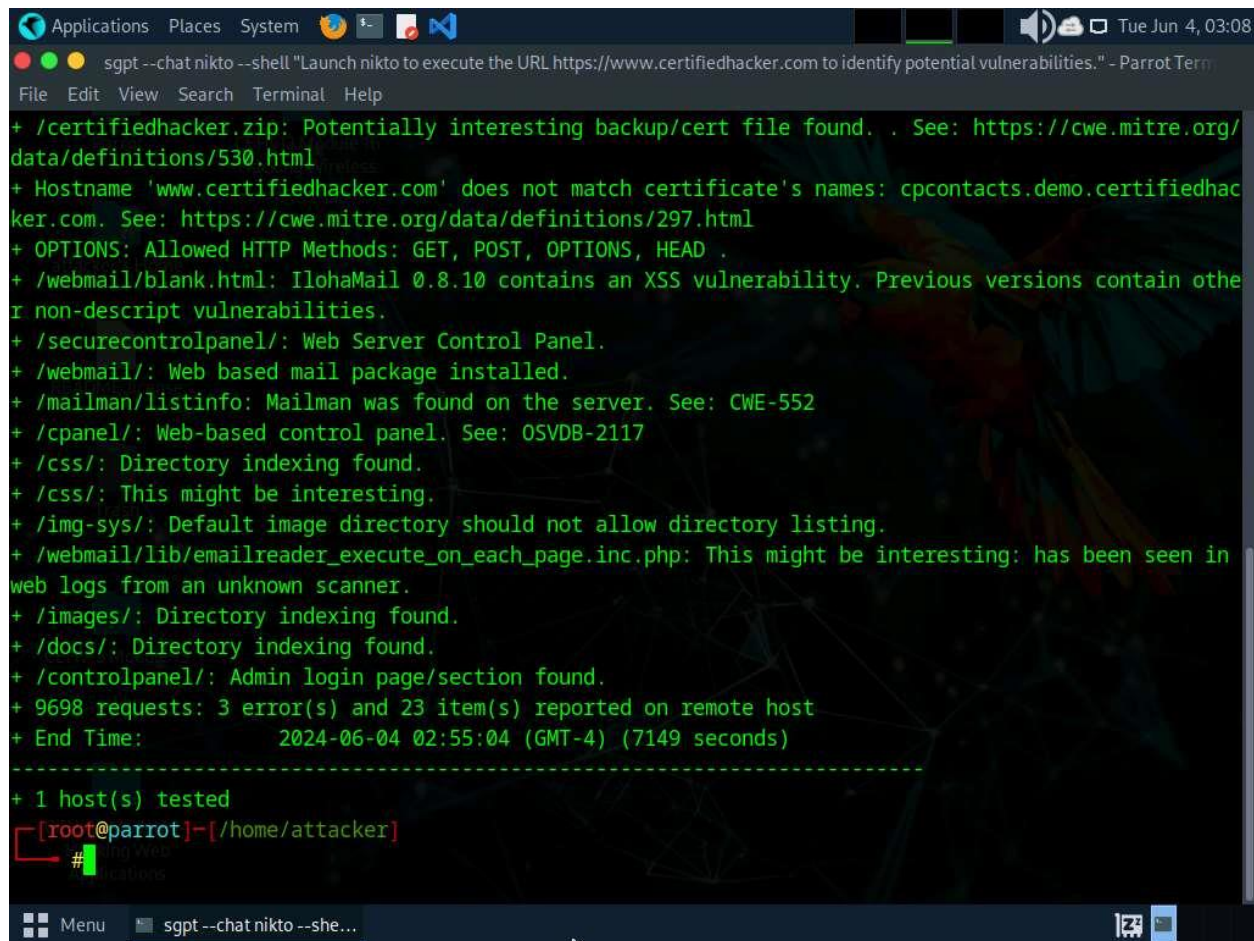
4. `--chat nikto --shell "Scan the URL https://www.certifiedhacker.com to identify potential vulnerabilities with nikto"*` to launch Nikto scan on the target website.

In the prompt, type **E** and press **Enter** to execute the command.



5. Scan result appears displaying the discovered vulnerabilities in the target website (here, **www.certifiedhacker.com**), as shown in the screenshot.

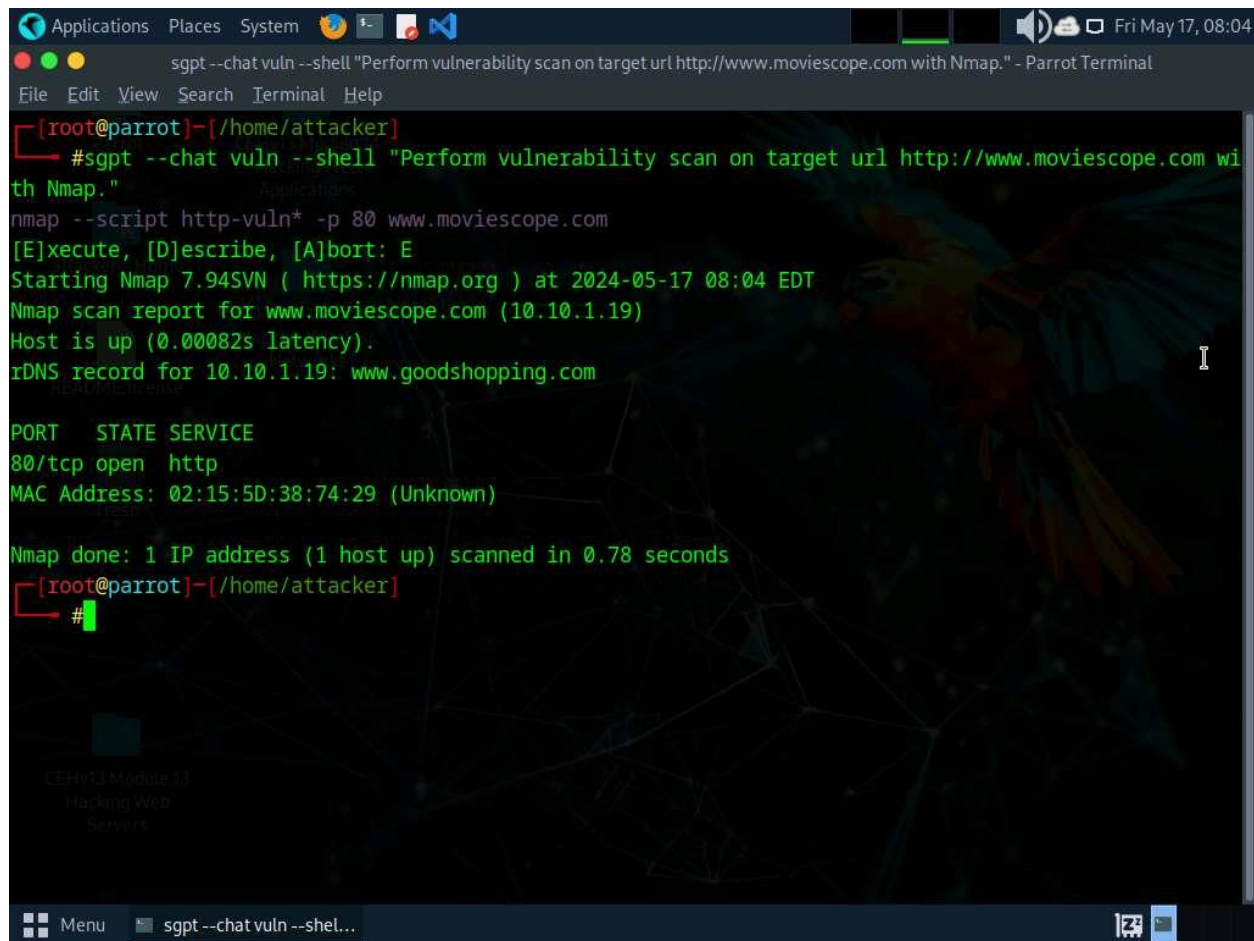
```
Applications Places System sgpt --chat nikto --shell "Scan the URL https://www.certifiedhacker.com to identify potential vulnerabilities with nikto" - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]-[/home/attacker]
#sgpt --chat nikto --shell "Scan the URL https://www.certifiedhacker.com to identify potential vulnerabilities with nikto"
nikto -h https://www.certifiedhacker.com
[E]xecute, [D]escribe, [A]bort: E
- Nikto v2.5.0
-----
+ Target IP: 162.241.216.11
+ Target Hostname: www.certifiedhacker.com
+ Target Port: 443
-----
+ SSL Info: Subject: /CN=webdisk.certifiedhacker.com
Ciphers: TLS_AES_256_GCM_SHA384
Issuer: /C=US/O=Let's Encrypt/CN=R10
+ Start Time: 2025-03-26 06:37:53 (GMT-4)
-----
+ Server: nginx/1.25.5
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Uncommon header 'x-proxy-cache' found, with contents: HIT.
+ /: Uncommon header 'x-server-cache' found, with contents: true.
+ /: Uncommon header 'host-header' found, with contents: c2hhcmVkJsdWVob3N0LmNvbQ==.
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerab
```



```
Applications Places System sgpt --chat nikto --shell "Launch nikto to execute the URL https://www.certifiedhacker.com to identify potential vulnerabilities." - Parrot Tern
File Edit View Search Terminal Help
+ /certifiedhacker.zip: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ Hostname 'www.certifiedhacker.com' does not match certificate's names: cpcontacts.demo.certifiedhacker.com. See: https://cwe.mitre.org/data/definitions/297.html
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ /webmail/blank.html: IlohaMail 0.8.10 contains an XSS vulnerability. Previous versions contain other non-descript vulnerabilities.
+ /securecontrolpanel/: Web Server Control Panel.
+ /webmail/: Web based mail package installed.
+ /mailman/listinfo: Mailman was found on the server. See: CWE-552
+ /cpanel/: Web-based control panel. See: OSVDB-2117
+ /css/: Directory indexing found.
+ /css/: This might be interesting.
+ /img-sys/: Default image directory should not allow directory listing.
+ /webmail/lib/emailreader_execute_on_each_page.inc.php: This might be interesting: has been seen in web logs from an unknown scanner.
+ /images/: Directory indexing found.
+ /docs/: Directory indexing found.
+ /controlpanel/: Admin login page/section found.
+ 9698 requests: 3 error(s) and 23 item(s) reported on remote host
+ End Time: 2024-06-04 02:55:04 (GMT-4) (7149 seconds)
-----
+ 1 host(s) tested
[root@parrot]-[/home/attacker]
#
```

Nikto scan takes long time to complete. You can terminate the scan, by pressing **Ctrl + Z**.

6. In the terminal, run **sgpt --chat vuln --shell "Perform vulnerability scan on target url http://www.moviescope.com with Nmap"** command to perform vulnerability scan on the target website. The result appears displaying open ports and services running on the target website.



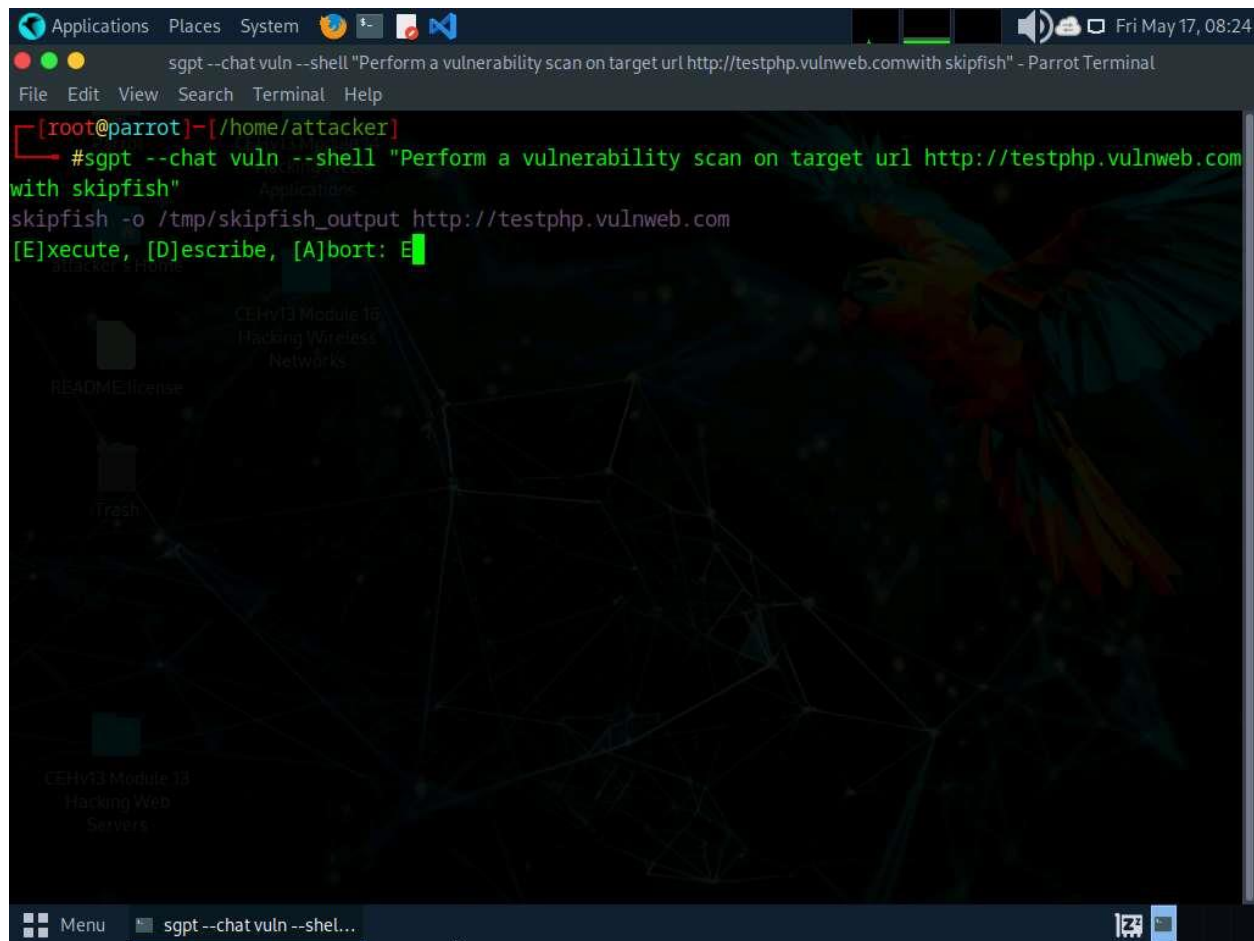
```
Applications Places System Fri May 17, 08:04
sgpt --chat vuln --shell "Perform vulnerability scan on target url http://www.moviescope.com with Nmap." - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~/home/attacker
#sgpt --chat vuln --shell "Perform vulnerability scan on target url http://www.moviescope.com with Nmap."
nmap --script http-vuln* -p 80 www.moviescope.com
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-17 08:04 EDT
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.00082s latency).
rDNS record for 10.10.1.19: www.goodshopping.com

PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:15:5D:38:74:29 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.78 seconds
[root@parrot]~/home/attacker
#
```

7. Run **sgpt --chat vuln --shell "Perform a vulnerability scan on target url <http://testphp.vulnweb.com> with skipfish"** to scan the target URL using skipfish tool.

If a prompt appears, enter any key to continue the scanning process.

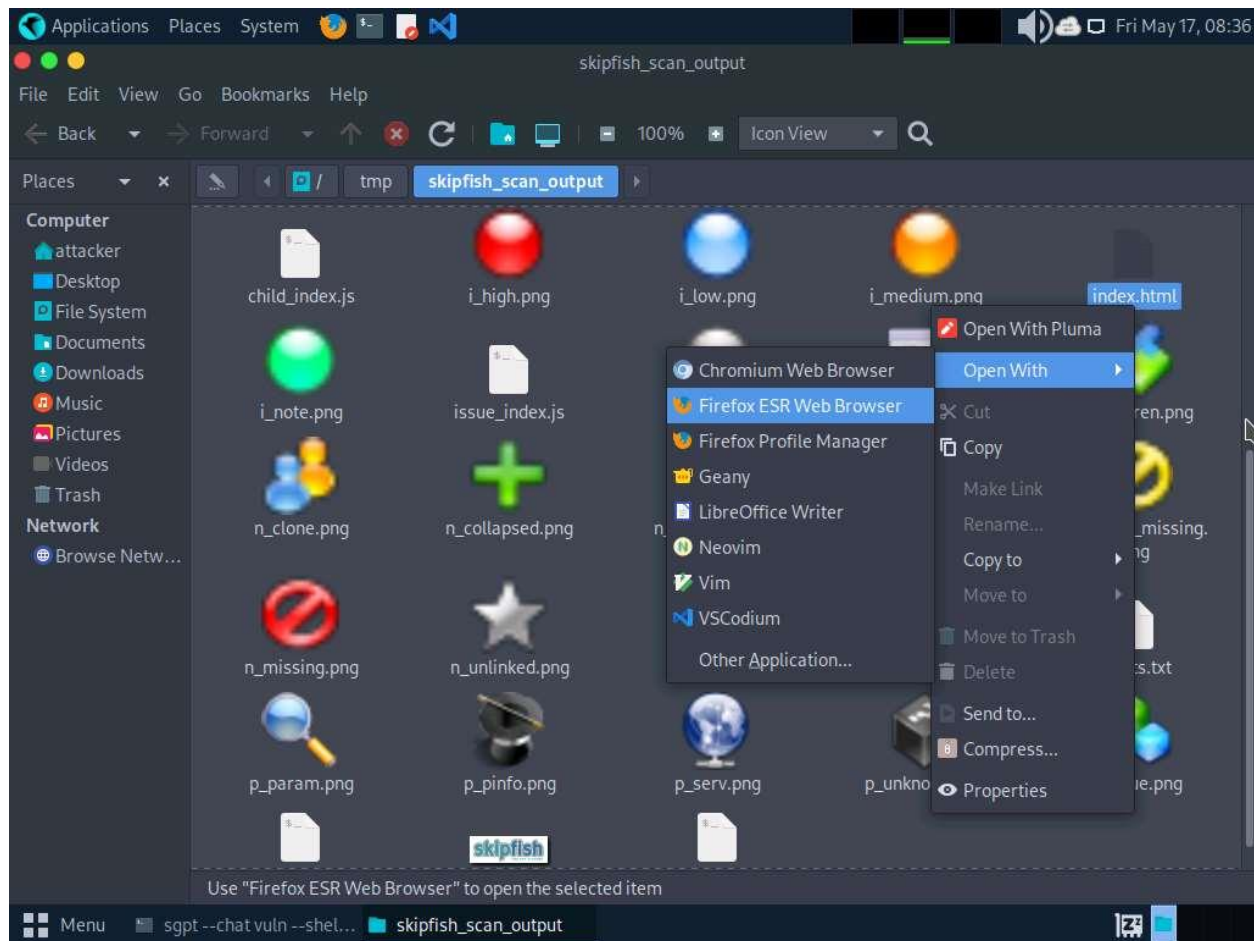


The screenshot shows a Parrot OS desktop environment with a terminal window open. The terminal title bar reads "sgpt --chat vuln --shell 'Perform a vulnerability scan on target url http://testphp.vulnweb.com with skipfish' - Parrot Terminal". The terminal content shows the user is root at the parrot machine in the /home/attacker directory. They enter the command: `#sgpt --chat vuln --shell "Perform a vulnerability scan on target url http://testphp.vulnweb.com with skipfish"`. The terminal then shows the skipfish command being executed: `skipfish -o /tmp/skipfish_output http://testphp.vulnweb.com`. The prompt changes to `[E]xecute, [D]escribe, [A]bort: E`, indicating the scan has started. The background of the terminal window features a dark theme with a parrot illustration and text labels for "CEHv13 Module 18 Hacking Wireless Networks" and "CEHv13 Module 13 Hacking Web Servers".

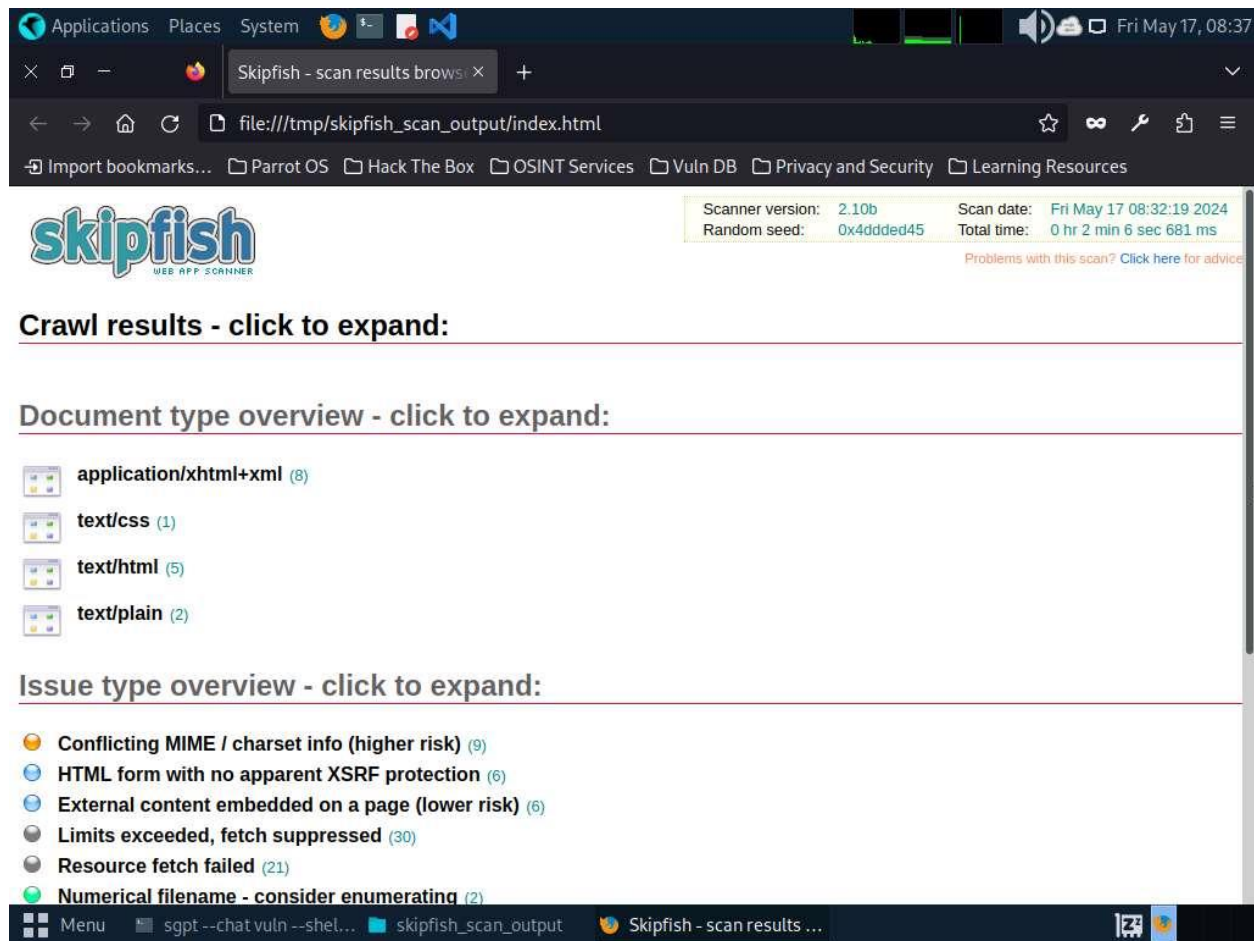
```
[root@parrot]-[/home/attacker]
#sgpt --chat vuln --shell "Perform a vulnerability scan on target url http://testphp.vulnweb.com
with skipfish"
skipfish -o /tmp/skipfish_output http://testphp.vulnweb.com
[E]xecute, [D]escribe, [A]bort: E
```

8. The skipfish begins scanning the target url. After the successful completion of the scan, report is saved at the **/tmp/skipfish_scan_output/** location, named as **index.html**. Navigate to the location, right-click on **index.html** and open with **Firefox ESR Web Browser**, as shown in the screenshot.

The location of scan report might differ. You can view the location in the skipfish command generated by ShellGPT.



9. Firefox browser window appears displaying the complete scan report, as shown in the screenshot.



10. Apart from the aforementioned commands, you can further explore additional options within the ShellGPT tool and utilize various other tools to conduct vulnerability assessments on the target.
11. This concludes the demonstration of performing vulnerability assessment on the target system using ShellGPT.
12. Close all open windows and document all the acquired information.

Question 5.3.1.1

Write a prompt using ShellGPT to perform vulnerability scan on www.certifiedhacker.com website using Nikto vulnerability scanner. Enter the contents of Uncommon header 'host header' found during the vulnerability scan.