



# REPORT OF VULNERABILITY ANALYSIS

BY SACHCHITANAND YADAV

# VULNERABILITY ANALYSIS MODULE - 5

## Learning Objectives -

- Summarize Vulnerability Assessment Concepts
- Vulnerability Research with Vulnerability Scoring Systems and Databases
- Perform Vulnerability Assessment using Various Vulnerability Assessment Tools
- Perform Vulnerability Analysis using AI
- Vulnerability Analysis – Report Section

# Table of Contents

## 1. Introduction to Vulnerability Assessment

- 1.1 Definition of Vulnerability Analysis
  - 1.2 Objectives of Vulnerability Assessment
  - 1.3 Types of Vulnerabilities
  - 1.4 Vulnerability Severity Levels (CVSS)
  - 1.5 Vulnerability Analysis Process
  - 1.6 Popular Vulnerability Assessment Tools
- 

## 2. Vulnerability Research with Scoring Systems & Databases

- 2.1 Overview of Vulnerability Research
  - 2.2 CWE (Common Weakness Enumeration) – Definition
  - 2.3 How to Use CWE Website
  - 2.4 Exploring SMB-Related Weaknesses
  - 2.5 Conclusion on CWE-Based Research
- 

## 3. Vulnerability Analysis Using ZAP

- 3.1 Introduction to ZAP
  - 3.2 How to Install and Use ZAP
  - 3.3 Automated Scan & Manual Testing
  - 3.4 Viewing Alerts and Reports
  - 3.5 Conclusion on ZAP
- 

## 4. Vulnerability Analysis Using Nessus

- 4.1 Introduction to Nessus
  - 4.2 Download and Setup Instructions
  - 4.3 Running a New Scan
  - 4.4 Target Selection & Scanning Process
  - 4.5 Conclusion on Nessus
-

## 5. Vulnerability Analysis Using Smart Scanner

- 5.1 Definition of Smart Scanner
  - 5.2 How to Use Smart Scanner
  - 5.3 Report Output
  - 5.4 Conclusion on Smart Scanner
- 

## 6. Vulnerability Analysis Using MBSA (Microsoft Baseline Security Analyzer)

- 6.1 Introduction to MBSA
  - 6.2 Benefits of MBSA
  - 6.3 How to Download & Run a Scan
  - 6.4 Saving/Printing Reports
  - 6.5 Conclusion on MBSA
- 

## 7. Vulnerability Analysis Using Acunetix

- 7.1 Introduction to Acunetix
  - 7.2 Benefits of Acunetix
  - 7.3 Download & Setup Instructions
  - 7.4 Running a Scan Using Acunetix
  - 7.5 Conclusion on Acunetix
- 

## 8. Performing Vulnerability Analysis Using AI (Gemini-CLI)

- 8.1 Introduction to AI-Based Vulnerability Analysis
  - 8.2 Benefits of Using Gemini-CLI
  - 8.3 Sample Prompts (Nmap, Nikto, Skipfish)
- 

## 9. Detailed Vulnerability Analysis Notes

- 9.1 Vulnerability Analysis – Overview
  - 9.2 Core Steps in Vulnerability Analysis
  - 9.3 Types of Vulnerability Assessments
  - 9.4 Common Tools Used for Vulnerability Analysis
  - 9.5 Key Outcomes of Vulnerability Analysis
-

## 10. Conclusion

- 10.1 Summary of Findings
- 10.2 Importance of Continuous Security Monitoring
- 10.3 Final Thoughts

SACHCHITANAND

# Summarize Vulnerability Assessment Concepts: -

## VULNERABILITY ANALYSIS

### Definition

Vulnerability analysis is the systematic process of identifying, evaluating, and prioritizing security weaknesses within networks, systems, and applications. The primary goal is to detect potential threats before malicious attackers can exploit them, ensuring the protection and integrity of organizational assets.

### Objectives

- Detect network vulnerabilities.
- Identify potential security weaknesses.
- Discover software errors or misconfigurations.
- Determine operating system versions running on systems or devices.
- Identify accounts using weak or default passwords.
- Evaluate application and service configurations.

### Types of Vulnerabilities

Category	Examples
Software	SQL Injection, Buffer Overflow

Category	Examples
<b>Network</b>	Open Ports, Weak Firewall Rules
<b>Configuration</b>	Default Credentials, Outdated Services
<b>Web</b>	Cross-Site Scripting (XSS), CSRF, Insecure Cookies
<b>OS-related</b>	Unpatched Operating Systems, Kernel Issues

### Vulnerability Severity Levels (CVSS)

The *Common Vulnerability Scoring System (CVSS)* is used to assess the severity of discovered vulnerabilities based on their potential impact.

#### Severity CVSS Score Range Meaning

<b>Low</b>	0.1 – 3.9	Minor risk with limited impact
<b>Medium</b>	4.0 – 6.9	Moderate risk requiring attention
<b>High</b>	7.0 – 8.9	Serious risk that could lead to significant compromise
<b>Critical</b>	9.0 – 10.0	Immediate threat, easily exploitable

## Vulnerability Analysis Process

### 1. Asset Discovery

Identify all systems, applications, and networks to be scanned.

### 2. Vulnerability Scanning

Perform automated scans using tools such as *Nessus*, *Nikto*, or *Acunetix*.

### 3. Result Analysis

Examine detected vulnerabilities and filter out false positives.

### 4. Prioritization

Rank vulnerabilities based on CVSS scores, exploitability, and business impact.

### 5. Remediation

Patch the vulnerabilities or implement appropriate security controls.

### 6. Reporting and Monitoring

Document findings, generate reports, and continuously monitor systems for new threats.

## Popular Tools for Vulnerability Analysis

Tool	Use Case
Nessus	Comprehensive system and network vulnerability scanning
Nikto	Web server vulnerability assessment
Acunetix	Automated web application security testing
MBSA (Microsoft Baseline Security Analyzer)	Windows-based system vulnerability scanning
OWASP ZAP (Zed Attack Proxy)	Manual and automated web application testing
Trivy	Container and cloud environment vulnerability scanning

Tool	Use Case
Qualys SSL Labs	SSL/TLS configuration and security testing

## Summary

Vulnerability assessment is the process of identifying, analyzing, and prioritizing weaknesses in systems, networks, applications, and devices. It helps organizations understand their security posture by scanning for outdated software, misconfigurations, missing patches, and known vulnerabilities. Using tools like Nessus, Acunetix, ZAP, Smart Scanner, and MBSA, analysts can detect risks early, evaluate their severity, and apply recommended fixes. Overall, vulnerability assessment ensures systems remain protected by finding weaknesses before attackers exploit them.

# Vulnerability Research with Vulnerability Scoring Systems and Databases

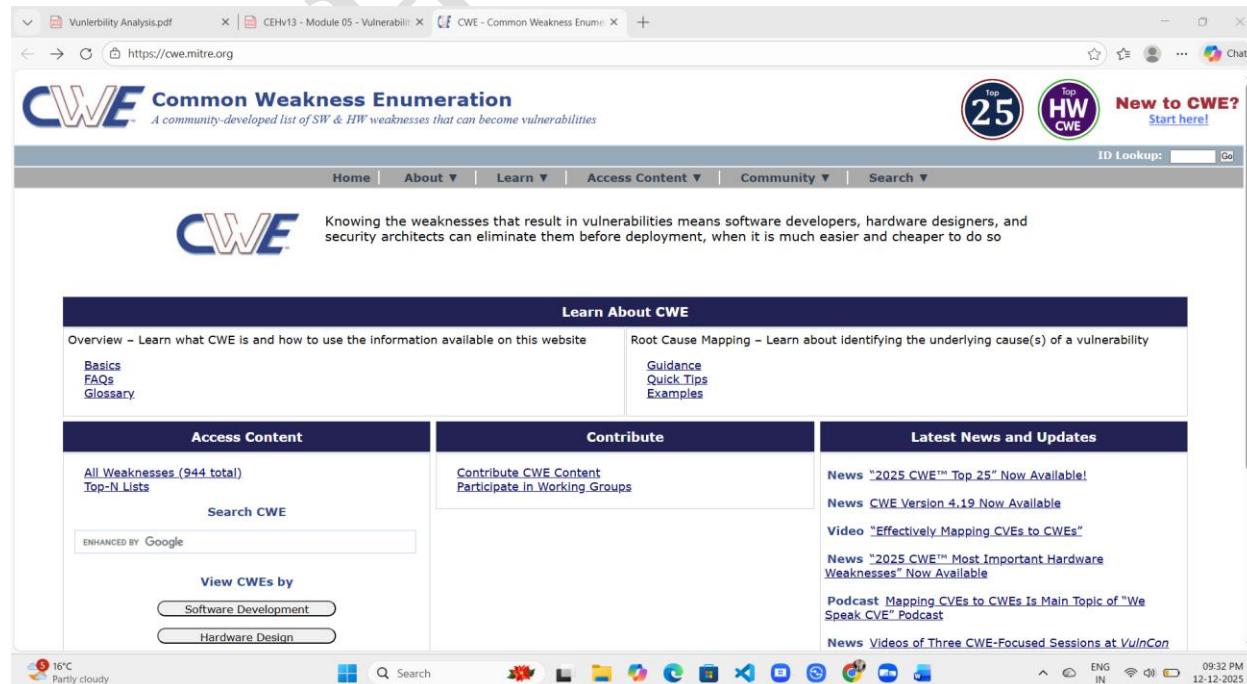
## Vulnerability Research in Common Weakness Enumeration (CWE)

### Definition

CWE is a standardized list of common software weaknesses. Using it for vulnerability research helps you classify issues, understand their root cause, and connect them to known weakness patterns. Instead of just finding a bug, CWE explains *why* the bug exists. This makes analysis clearer, improves secure coding, and helps predict similar flaws in the future.

Launch any web browser, and go to <https://cwe.mitre.org/>

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.



## MODULE – 5 VULNERABILITY ANALYSIS

The screenshot shows the CWE (Common Weakness Enumeration) website. In the search bar at the top, the term "SMB" has been entered. The search results are displayed in the main content area, listing various CWE IDs and their descriptions. The first result is "CWE-284: Improper Access Control (4.19)". Other visible results include "CWE-319: Cleartext Transmission of Sensitive Information", "CWE-693: Protection Mechanism Failure (4.18)", "CWE-200: Exposure of Sensitive Information to an Unauthorized Actor", "CWE-427: Uncontrolled Search Path Element (4.19)", "CWE-552: Files or Directories Accessible to External Parties", "CWE-294: Authentication Bypass by Capture-replay (4.19)", "CWE-130: Improper Handling of Length Parameter Inconsistency", and "CWE VIEW: Weaknesses Addressed by ISA/IEC 62443 Requirements". The page also features navigation links for "Access Content", "Contribute", and "Latest News and Updates".

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.

**Note:** The search results might differ when you perform this task

This screenshot is identical to the one above, showing the same search results for "SMB" on the CWE website. The results list various CWE IDs and their descriptions, including "CWE-284: Improper Access Control (4.19)" and other related weaknesses. The interface and layout are consistent with the previous screenshot.

## MODULE – 5 VULNERABILITY ANALYSIS

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

The screenshot shows a web browser window with the URL <https://cwe.mitre.org/data/definitions/284.html>. The page is titled "CWE-284: Improper Access Control". It includes sections for "Description", "Extended Description", and "Alternate Terms". The "Description" section states: "The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor." The "Extended Description" section provides more detail about access control mechanisms and their failure points. The "Alternate Terms" section lists terms like "SMB", "File sharing", and "Network sharing". The top of the page features the "Common Weakness Enumeration" logo and navigation links for Home, About, Learn, Access Content, Community, and Search. There are also "Top 25" and "Top HW CWE" badges.

## Conclusion

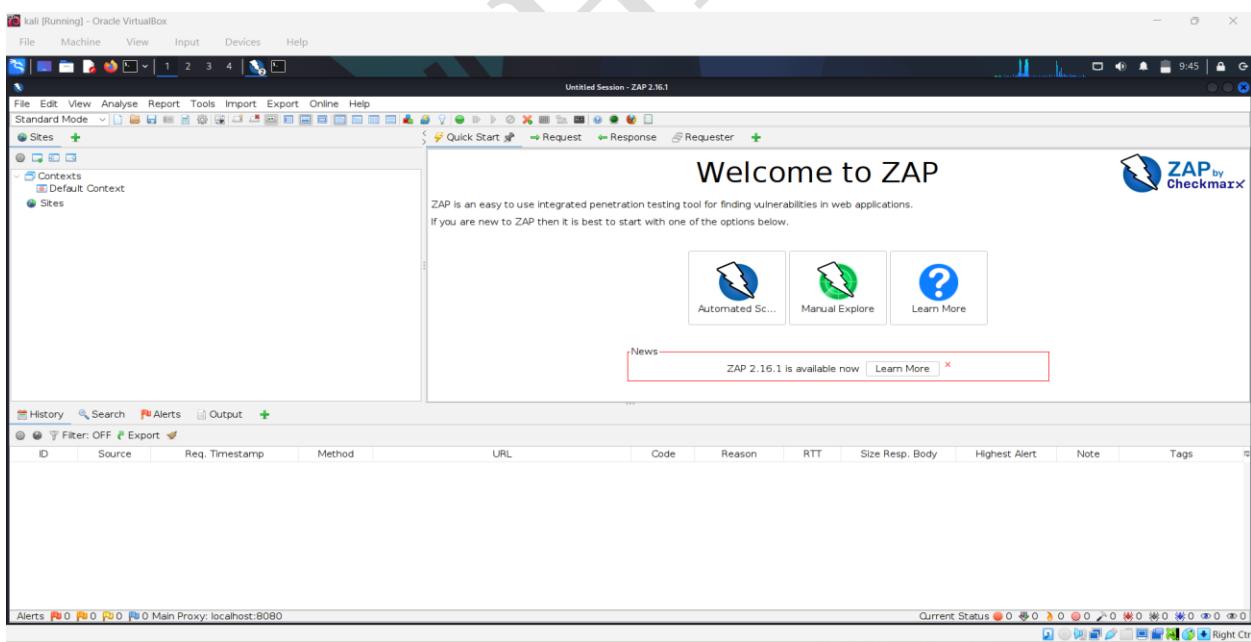
Using CWE for vulnerability research provides a structured way to understand software weaknesses at their core. By exploring known patterns and root causes, analysts can identify issues more accurately and strengthen security practices. Searching CWE for terms like **SMB** helps map specific vulnerabilities to their underlying weaknesses, improving both analysis and prevention.

# Vulnerability Analysis Using Checkmarx ZAP

ZAP (Zed Attack Proxy) is a dynamic application security testing tool published under the Apache License. It allows users to manipulate all of the traffic that passes through it, including HTTPS encrypted traffic, when used as a proxy server.

## How to use it –

- ❖ Open kali Linux / Parrot OS
- ❖ Open Terminal And type sudo apt install zaproxy
- ❖ Then type zaproxy
- ❖ Click on automated scan



- ❖ If you want to see there is any vulnerability spotted on your target just click on alerts option
- ❖ See our target are vulnerable with many things.

## MODULE – 5 VULNERABILITY ANALYSIS

**Cross Site Scripting (Reflected)**

URL: http://testfire.net/sendFeedback

Risk: High

Confidence: Medium

Parameter: name

Attack: <div><<script>alert(1)</script></div>

CWE ID: 79

WASC ID: 8

Source: Active (40012 - Cross Site Scripting (Reflected))

Input Vector: Form Query

Description: Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

**PERSONAL**

- Personal Products
  - Checking
  - Loan Products
  - Cards
  - Investments & Insurance
  - Other Services

**SMALL BUSINESS**

- Deposit Products
- Lending Services
- Bank
- Insurance
- Retirement
- Other Services

**INSIDE ALTORO MUTUAL**

- Find Us
- Contact Us
- Locations
- Press Releases
- Press Room
- Careers
- Subscribe

[Privacy Policy](#) | [Security Statement](#) | [Server Status Check](#) | [REST API](#) | © 2005 Altoro Mutual, Inc.

This web application is open source! Get your copy from [GitHub](#) and take advantage of advanced features.

The Altoro2 website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to [http://www.hcl-software.com/privacy/](#).

Copyright © 2008, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd. All rights reserved.

## MODULE – 5 VULNERABILITY ANALYSIS

**Cross Site Scripting (Reflected)**

URL: http://testfire.net/sendFeedback

Risk: High

Confidence: Medium

Parameter: name

Attack: <div><<!</>><script>alert(1);</script></div>

CWE ID: 79

WASC ID: 8

Source: Active (40012 - Cross Site Scripting (Reflected))

Description: Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

**Online Banking Login**

Username:

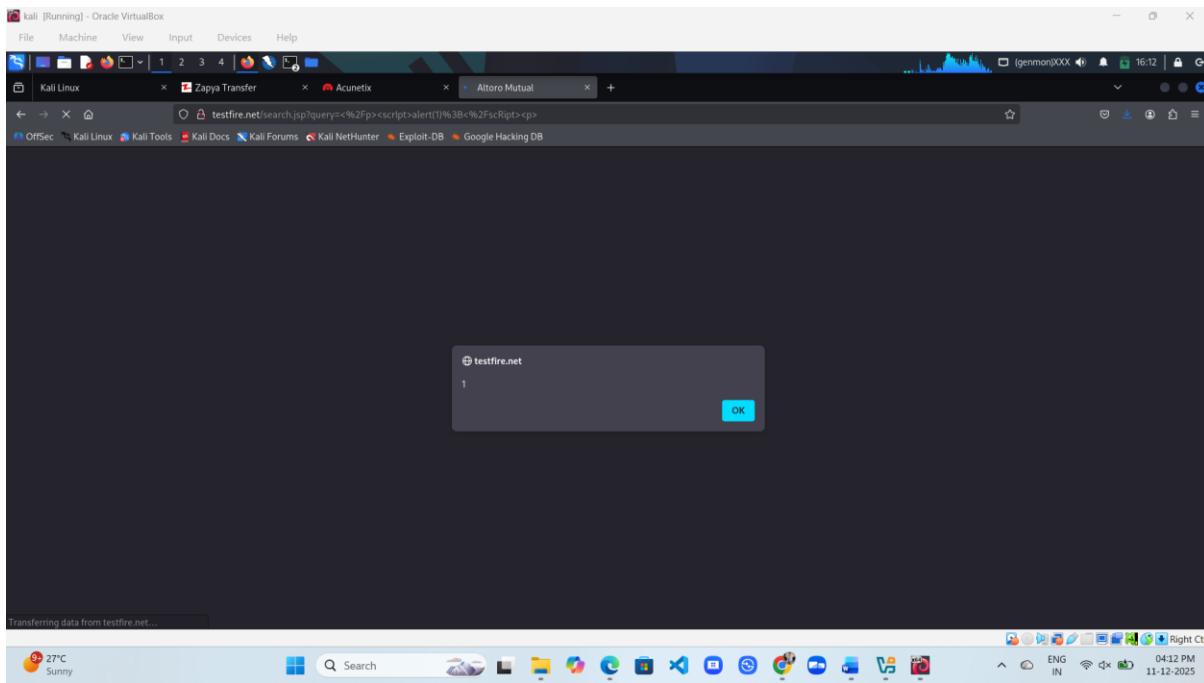
Password:

Login

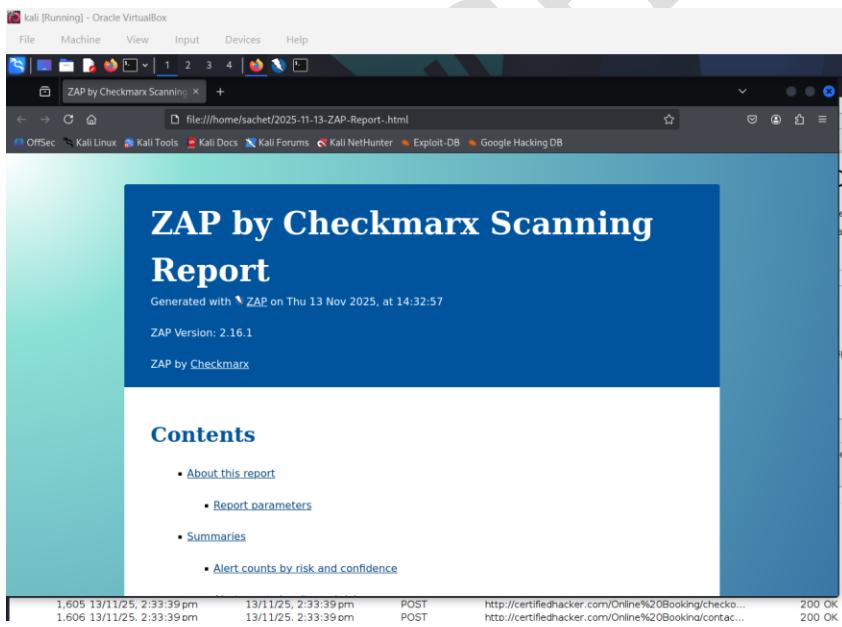
The Altoro2 website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <http://www.hcl-software.com/privacy/>.

Copyright © 2008, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

## MODULE – 5 VULNERABILITY ANALYSIS



- ❖ Now You can also Generate report
- ❖ Click on report section and click generate report



1.605 13/11/25, 2:33:39 pm	13/11/25, 2:33:39 pm	POST	http://certifiedhacker.com/Online%20Booking/checko...	200 OK
1.606 13/11/25, 2:33:39 pm	13/11/25, 2:33:39 pm	POST	http://certifiedhacker.com/Online%20Booking/contact...	200 OK

## Conclusion

Checkmarx ZAP makes it easy to find web application weaknesses by intercepting and analyzing traffic, even over HTTPS. It helps identify vulnerabilities early and improves overall application security.

## Vulnerability Analysis Using Nessus.

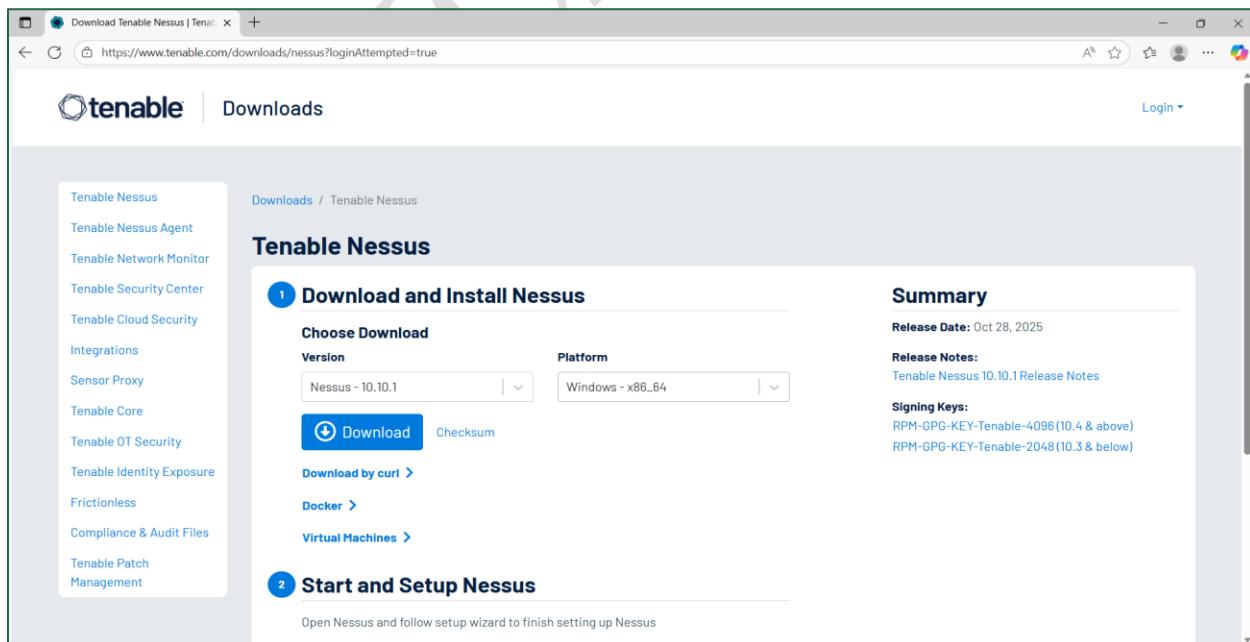
Nessus is a popular vulnerability assessment tool used to scan networks, systems, and applications for security weaknesses. It is developed by Tenable, Inc. and is widely used for penetration testing, compliance auditing, and risk assessment.

### Nessus Download Link –

<https://www.tenable.com/downloads/nessus?loginAttempted=true>

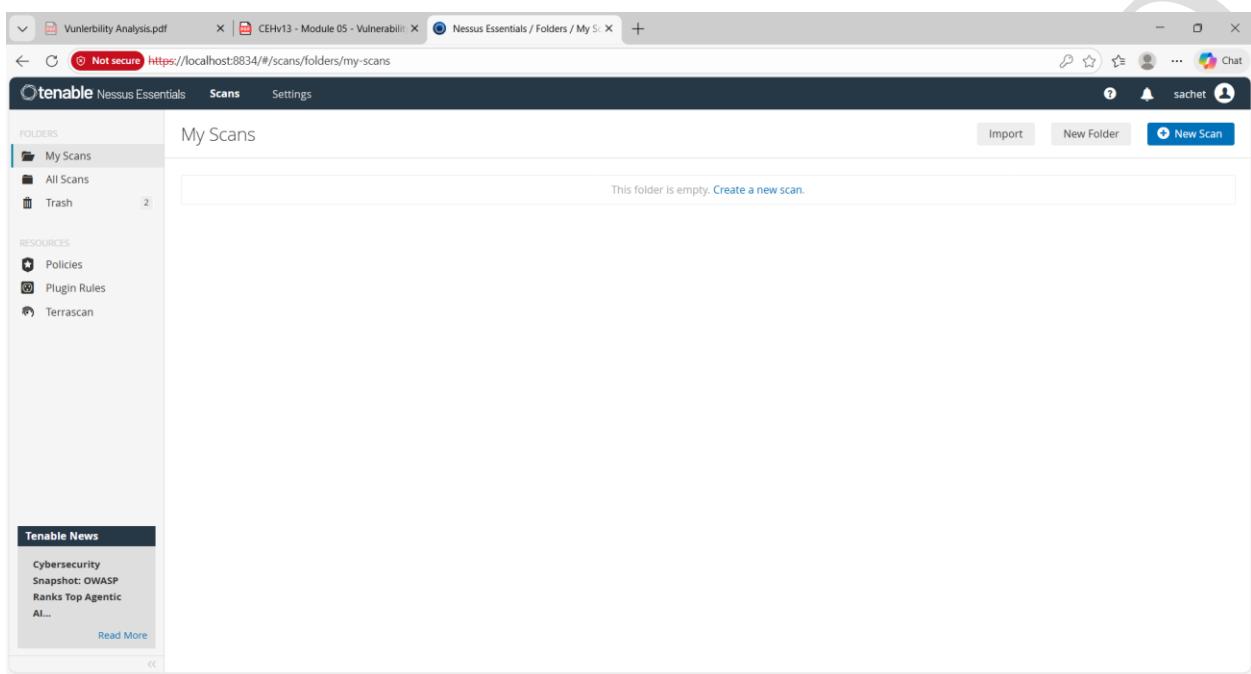
### How To Download Nessus –

- ❖ Open Browser
- ❖ Search Nessus Download
- ❖ Click first website for official Tenable website and you can redirected to download page



## MODULE – 5 VULNERABILITY ANALYSIS

- ❖ After completing the setup, the Nessus interface appears. To begin a vulnerability scan, click on the “New Scan” option.



- ❖ Provide target

Note – In my case , I'm scan Entire network , you can also scan a entire network as well as single ip address its all depend on you

- ❖ Click on save and then click on launch
- ❖ Here , scanning start

## MODULE – 5 VULNERABILITY ANALYSIS

The screenshot shows the Otenable Nessus Essentials interface. The left sidebar includes 'My Scans' (2), 'All Scans', and 'Trash'. Under 'RESOURCES', there are 'Policies', 'Plugin Rules', and 'Terrascan'. A 'Tenable News' section highlights 'Google Cloud Platform (GCP)', 'Google Security Operat...', and a 'Read More' link. The main content area displays a scan titled 't1' with 16 vulnerabilities found. The table columns are 'Sev', 'CVSS', 'VPR', 'EPSS', 'Name', 'Family', 'Count', and 'Scan Details'. The 'Scan Details' pane shows the policy as 'Basic Network Scan', status as 'Aborted', severity base as 'CVSS v3.0', scanner as 'Local Scanner', start at 'Today at 4:38 PM', and end at 'Today at 4:45 PM'. A pie chart titled 'Vulnerabilities' indicates the distribution by severity: Critical (red), High (orange), Medium (yellow), and Info (blue). The bottom taskbar shows various application icons and the system clock at 04:53 PM on 11-12-2025.

## Conclusion

Nessus makes vulnerability analysis straightforward and reliable, giving you a clear view of the security weaknesses hiding inside networks, systems, and applications. Its scanning engine is fast, its results are detailed, and its interface keeps things simple even when the risks aren't. Once downloaded and set up through the official Tenable site, it becomes a powerful part of any security workflow — helping you spot threats early, strengthen configurations, and maintain a strong security posture with confidence.

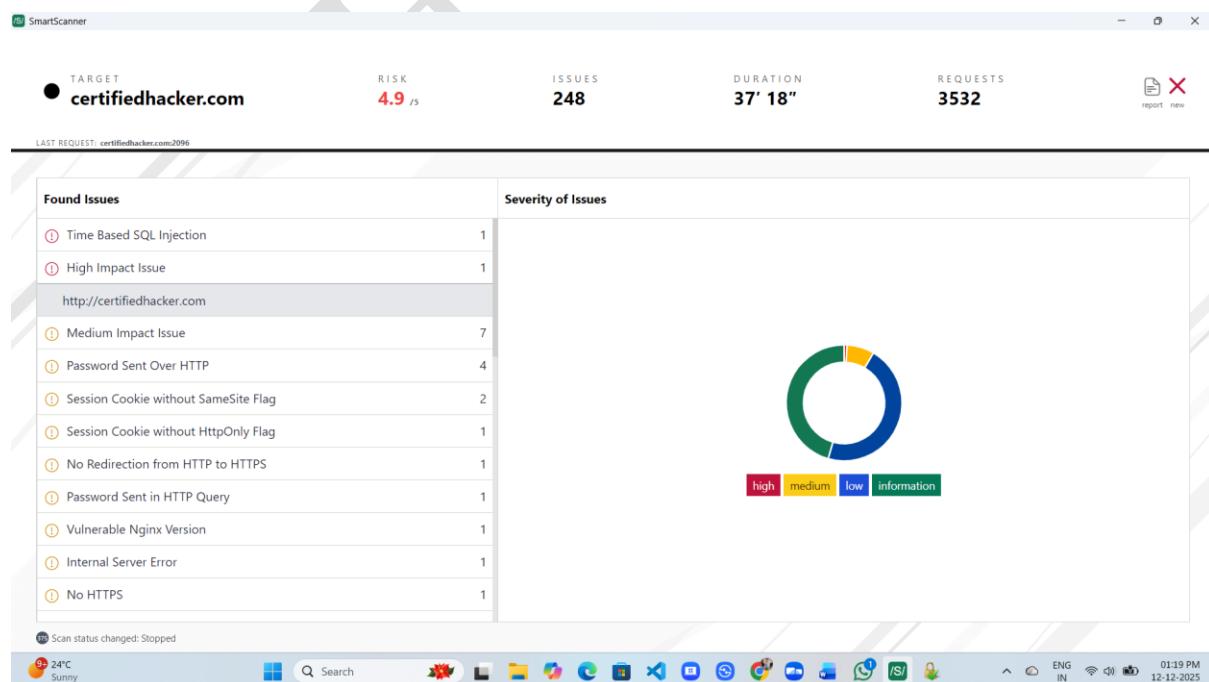
# Vulnerability Analysis Using Smart Scanner

## Definition

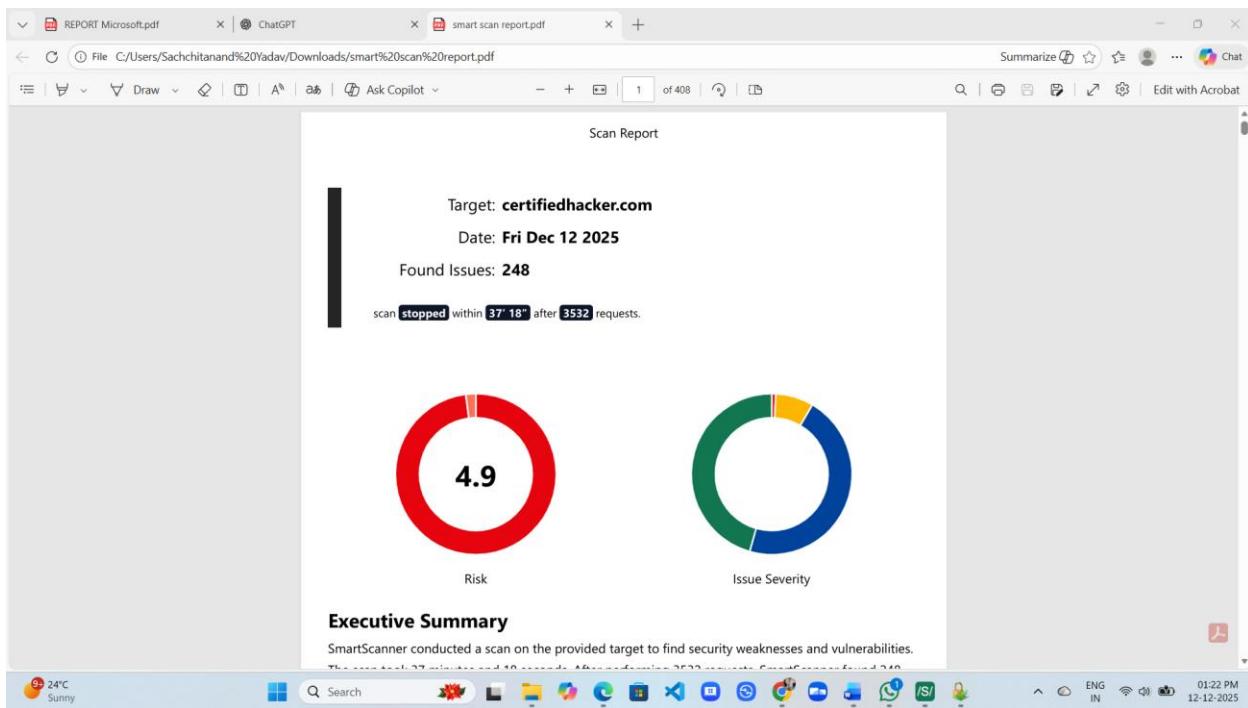
A **Smart Scanner** is an automated vulnerability assessment tool that uses intelligent scanning techniques to detect security weaknesses in systems, networks, and applications. It speeds up analysis by identifying outdated software, misconfigurations, exposed ports, and known vulnerabilities with minimal manual effort.

## How to Use

- Open the Smart Scanner application.**
- Select the target system or network** you want to analyze.
- Choose the **scan type** (quick scan, deep scan, or custom scan).
- Click **Start Scan** to begin the vulnerability assessment.
- Wait for the tool to analyze your target and **review the results**, which include detected vulnerabilities, severity levels, and recommended fixes.
- Apply the suggested remediation steps** to secure the system.



## Report -



## Conclusion

Smart Scanner makes vulnerability analysis fast and efficient by automating the detection of security flaws. Its easy-to-use interface, clear reports, and intelligent detection methods help users quickly understand risks and strengthen system security. It's a reliable tool for anyone who wants accurate results without complicated setup or manual analysis.

# Vulnerability Analysis Using MBSA (Microsoft Baseline Security Analyzer )

The Microsoft Baseline Security Analyzer (MBSA) was a free tool developed by Microsoft to help users scan their Windows systems for common security misconfigurations and missing security updates.

## Benefits

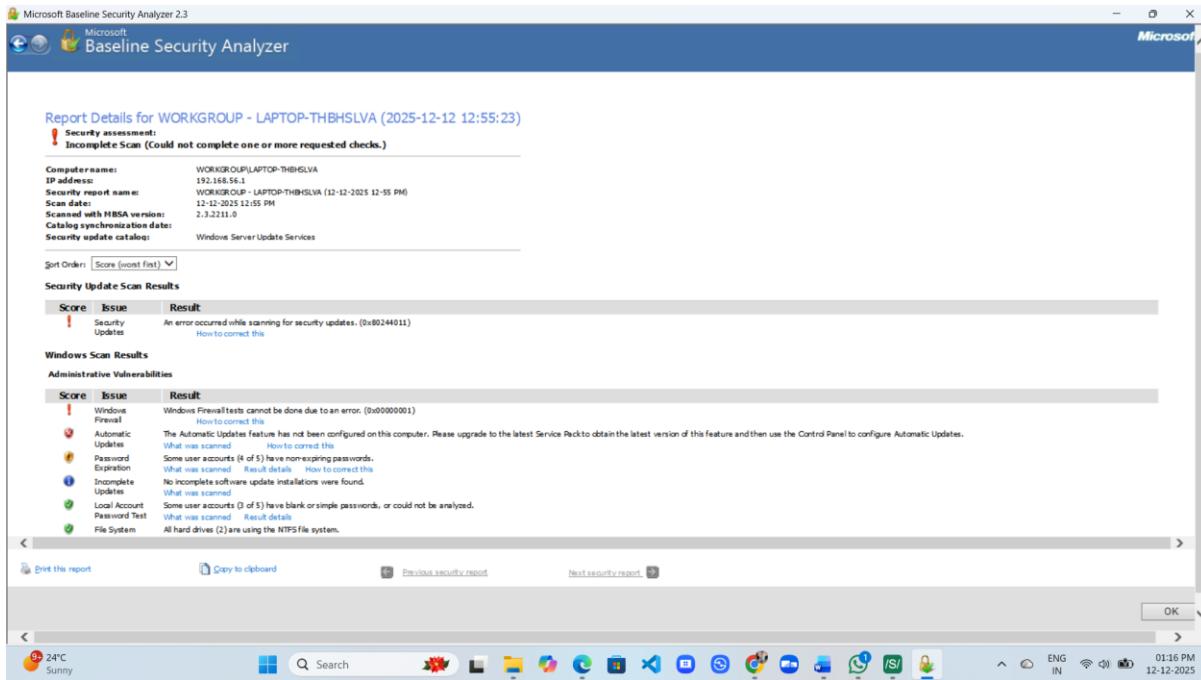
- **Easy to use** — no complex setup; even beginners can run scans smoothly.
- **Quick detection** — finds missing Windows updates, weak passwords, and insecure settings fast.
- **Gives clear recommendations** — shows what's wrong and how to fix it.
- **Focused on Windows security** — perfect for auditing Microsoft environments.
- **Lightweight** — runs without slowing down the system.

## How To Download MBSA

- ❖ Open Browser
- ❖ Search MBSA Download and click on Techspot website
- ❖ Link - : <https://www.techspot.com/downloads/3886-microsoftbaseline-security-analyzer.html>
- ❖ Click on Download Now
- ❖ After successfully install and setup mbsa , open it and click on scan computer
- ❖ As you can see MBSA automatically get your system name
- ❖ Click on start scan

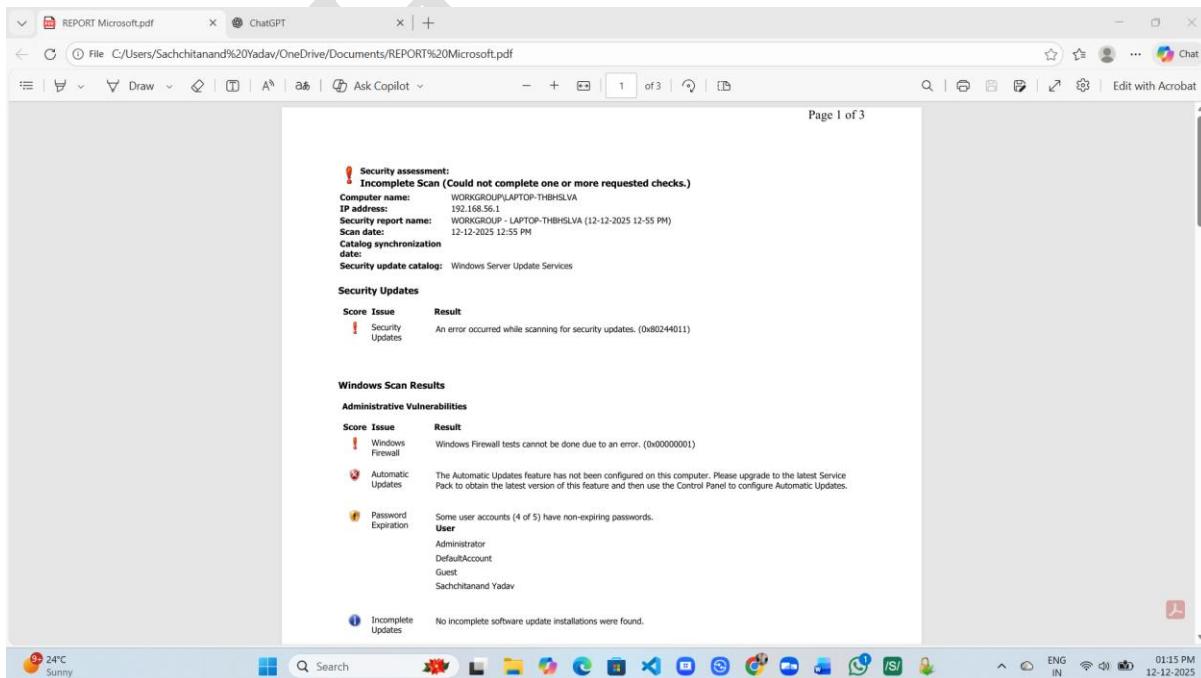
## MODULE – 5 VULNERABILITY ANALYSIS

❖ Here , MBSA scan the device



❖ You can also save or print this report , just click on print this report

❖ Click on print (save as a pdf )



## Conclusion

MBSA provides a simple and effective way to identify basic security issues in Windows systems. By highlighting missing patches, weak settings, and misconfigurations, it helps users quickly strengthen their system's security. Even with modern tools available, MBSA remains useful for fast, straightforward vulnerability checks in Microsoft environments.

# Vulnerability Analysis Using Acunetix

## Definition

Acunetix is a web vulnerability scanner designed to identify and help fix security issues in websites, web applications, and APIs. It automates the process of checking for vulnerabilities.

## Benefits of Using Acunetix

- **Automated scanning** finds vulnerabilities in websites, web apps, and APIs quickly.
- **Detects critical flaws** like SQL injection, XSS, misconfigurations, and weak authentication.
- **Easy to use** with a web-based dashboard that clearly shows issues and fixes.
- **Fast and accurate** scanning engine reduces false positives.
- **Supports continuous testing** for developers and security teams.
- **Generates detailed reports** useful for auditing, compliance, and patching.

**Download link :-:** <https://github.com/xiv3r/Acunetix-v24.10.241106172>

## How to use it :-:

- Open kali linux / parrot OS
- type git clone and paste git link
- copy and paste command one by one
- Now open a firefox and type in url section  
<https://127.0.0.1:3443>
- Successfully login in Acunetix
- Then click on scan and then click on new scan
- Click on scan

## MODULE – 5 VULNERABILITY ANALYSIS

- Here , scanning start

The screenshot shows the Acunetix web application interface. On the left is a sidebar with navigation links: Dashboard, Targets, Vulnerabilities, **Scans** (which is selected), Reports, Users, Scan Types, Network Scanner, Issue Trackers, Email Settings, Engines, Excluded Hours, Proxy Settings, and About. The main content area has tabs for Scan Information, Vulnerabilities, Site Structure, and Events. The Scan Information tab is active, displaying "Acunetix Threat Level 0" with a note that the threat level is not available yet. It shows a scan duration of 0m 30s, 140 requests, an average response time of 398ms, and 2 locations. Under Target Information, it lists the address as 162.241.216.11, server as Apache, operating system as Unknown, identified technologies as Responsive, and responsiveness as Yes. The Activity section shows the overall progress as "In Progress" with 0% completion. It lists three events: an initial request redirection, the start of the scan, and an antivirus not found. The status bar at the bottom shows the date as Dec 12, 2025, and the time as 1:57:06 PM.

The screenshot shows the Acunetix web application interface with the Reports tab selected in the sidebar. The main content area displays a table of reports. The table has columns for Report Template (with a dropdown arrow), Report Type, Target, Created On, Status, Download, and Delete. There is one entry: "Executive Summary" (Report Type), N/A (Target), Dec 12, 2025, 4:43:00 PM (Created On), Queued (Status), and a download icon (Download). The status bar at the bottom shows the date as Dec 12, 2025, and the time as 05:05 PM.

## **Conclusion**

Acunetix makes vulnerability assessment simple, fast, and reliable by automating the discovery of web application security flaws. Its powerful scanning engine, clear reporting, and easy setup help users quickly identify risks and strengthen their web security posture. Whether used for pentesting or routine audits, Acunetix provides a solid foundation for keeping applications secure.

# Perform Vulnerability Analysis using AI

## Use Gemini-CLI

Gemini-CLI lets you use AI to analyze scan results, logs, and configs with speed and clarity. Instead of digging through walls of raw data, the AI highlights vulnerabilities, ranks their severity, and recommends fixes instantly. It's like having a smart security assistant who never gets tired and doesn't miss the small stuff.

## Benefits:

- Faster and more accurate vulnerability detection
  - Explains issues in simple, context-aware language
  - Automates report creation and reduces manual effort
  - Helps beginners learn while boosting expert productivity

**Prompt :-** Perform vulnerability scan on target url <http://www.moviescope.com> with Nmap

kali [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Gemini - gemini-cli

(genimon)XXX 20:47

Session Actions Edit View Help

[sachet@kali:~] ~

[sachet]\$ id

[sachet]\$ password for sachet:

[root@kali:~/home/sachet]

[root@kali:~/home/sachet]

gemini-cli

Loaded cached credentials.

# > GEMINI

Tips for getting started:

1. Ask questions, browse files, or run commands.
2. Be specific for the best results.
3. Create `GEMINI.md` files to customize your interactions with Gemini.
4. `/help` for more information.

> Perform vulnerability scan on target url `http://www.moviescope.com` with Nmap

\* I will run an Nmap vulnerability scan on `'www.moviescope.com'`. This scan will probe for open ports, identify service versions, and check for known vulnerabilities using Nmap's scripting engine. This is an active scan that will send network packets to the target host.

v Shell nmap -sV -script vuln www.moviescope.com (Running Nmap vulnerability scan on `www.moviescope.com` to detect open ports, service versions, and known vulnerabilities.)

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-12 20:00 IST

Nmap scan report for `www.moviescope.com` (172.237.146.38)

Host is up (0.30s latency).

Other addresses for `www.moviescope.com` (not scanned): 172.233.219.123 172.237.146.0 172.233.219.49 172.233.219.78 172.237.146.25

Not shown: 908 filtered tcp ports (no-response)

PORT	STATE	SERVICE	VERSION
80/tcp	open	http	OpenResty web app server
443/tcp	open	https	OpenResty web app server, version: 2.4.41.1-1.1.10.1-1.1.10.1, built: 2023-07-03T10:55:00Z, SSLv3/TLSv1.3 support, KSS, _http-stored-xss: Couldn't find any stored XSS vulnerabilities.
_http-csrf:	closed		Couldn't find any CSRF vulnerabilities.
_http-aspect-debug:	closed		ERROR: Script execution failed (use -d to debug)
_https-aspect-debug:	closed		ERROR: Script execution failed (use -d to debug)
_jmx-console/1	closed		Authentication was not required
_http-trace-info:	closed		Problem with XML parsing of /evo/about
_http-vuln-cve2017-1001000:	closed		ERROR: Script execution failed (use -d to debug)
_http-vuln-cve2018-3704:	closed		ERROR: Script execution failed (use -d to debug)
443/tcp	open	https	OpenResty web app server
_http-enum:	closed		
_ /blog/	closed		
_ /news/	closed		
_ /weblogs/	closed		
_ /wordpress/:/ Blog	closed		

17'C Partly cloudy

Search

Right Click

## MODULE – 5 VULNERABILITY ANALYSIS

```
kali: [Running] - Oracle VirtualBox
File Machine View Input Devices Help

Session Actions Edit View Help
[?] open http:// OpenResty web app server
[!] http-dom-based-xss: Couldn't find any stored DOM based XSS.
[!] http-stored-xss: Couldn't find any stored XSS vulnerabilities.
[!] http-csrf: Couldn't find any CSRF vulnerabilities.
[!] http-aspect-debug: ERROR: Script execution failed (use -d to debug)
[!] http-jmx-console: Authentication was not required
[!] http-trame-info: Problem with XML parsing of /evo/about
[!] http-vuln-cve2018-18075: ERROR: Script execution failed (use -d to debug)
[!] http-vuln-cve2018-18076: ERROR: Script execution failed (use -d to debug)
443/tcp open ssl/HttpResty web app server
[!] http ENUM:
[+] /blog/ Blog
[+] /blog/ Blog
[+] /weblogs/ Blog
[+] /wordpress/ Blog
[+] /vhosts/ Blog
[+] /mediawiki/ Wiki
[+] /wiki/Main_Page: Wiki
[+] /tikiwiki/: Tikiwiki
[+] /gdm/ MajorDomo2: MajorDomo2 Mailing List
[+] /majordomo/majordomo: Majordomo2 Mailing List
[+] /j2ee/examples/servlets/: Oracle J2ee examples
[+] /j2ee/examples/jsp/: Oracle J2ee examples
[+] /syslog/ Logstash: Logstash LSSP Prevention Virtual Appliance
[+] /reg_1.html: Polycom IP Phone
[+] /adr.htm: Snom IP Phone
[+] /line_login.html#:1: Snom IP Phone
[+] /line_login.html#:1: Snom IP Phone
[!] http-jmx-console: Authentication was not required
[!] http-dom-based-xss: Couldn't find any DOM based XSS.
[!] http-csrf: Couldn't find any CSRF vulnerabilities.
[!] http-aspect-debug: ERROR: Script execution failed (use -d to debug)
[!] http-server-header: openresty
[!] http-majordomo-dir-traversal: ERROR: Script execution failed (use -d to debug)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 829.26 seconds

* The Nmap scan revealed that ports 80 (HTTP) and 443 (HTTPS) are open, both running an OpenResty web server. A potential vulnerability was identified: the JMX console at /jmx-console/ may not require authentication.

I can investigate this further by attempting to access the JMX console. Would you like me to proceed?

* Type your message or @path/to/file
/usr/lib/gemini-cli no sandbox (see /docs) gemini-2.5-pro (99% context left)

17°C Partly cloudy
 Search               Right Click ENG 12-12-2025
```

**Prompt :-** “Launch nikto to execute a scan against the URL <http://www.moviescope.com> to identify potential vulnerabilities.”

kali [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Gemini - gemini-cli

Session Actions Edit View Help

[sachet@kali:~] ~

[sachet]\$ sudo su

[sudo] password for sachet:

[root@sachet:~/]# /home/sachet

[root@sachet:~/]# gemini-cli

Loaded cached credentials.

# > GEMINI

Tips for getting started:

1. Type commands, files, or URLs to search for files, or run commands.
2. Be specific for the best results.
3. Create `GEMINI.md` files to customize your interactions with Gemini.
4. `/help` for more information.

> "Launch nikto to execute a scan against the URL <http://www.movieoscope.com> to identify potential vulnerabilities."

+ I will run nikto to scan <http://www.movieoscope.com> for vulnerabilities. This is a non-intrusive scan that checks for common web server issues.

v Shell nikto -h <http://www.movieoscope.com> (Run a Nikto scan against the specified URL to identify potential web server vulnerabilities.)

- Nikto v2.5.0

+ Multiple IPs Found: 172.233.219.78, 172.237.146.25, 172.237.146.38, 172.233.219.40, 172.233.219.123, 172.237.146.8

+ Target IP: 172.233.219.78

+ Target Hostname: [www.movieoscope.com](http://www.movieoscope.com)

+ Threads: 40

+ Start Time: 2025-12-12 21:03:16 (GMT5.5)

+ Server: openSUSE

+ 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 "accept-ch-", found, with contents: Sec-CH-UA, Sec-CH-UA-Arch, Sec-CH-UA-Bitness, Sec-CH-UA-Full-Version, Sec-CH-UA-Full-Version-List, Sec-CH-UA-Mobile, Sec-CH-UA-Model, Sec-CH-UA-Platform, Sec-CH-UA-Platform-Version, Sec-CH-UA-Media,

+/- The X-Content-Type-Options header is not set. This could force the user agent to render the content of the site in a different fashion to the MIME type. See: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options>

+ Scan terminated: 20 error(s) and 3 item(s) reported on remote host

+ End Time: 2025-12-12 21:06:39 (GMT5.5) (205 seconds)

+ 1 host(s) tested

+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response

\* Of course, I will save the results of the Nikto scan to the file `nikto_report.txt`.

v `writetile` writing to `nikto_report.txt`

16°C Partly cloudy

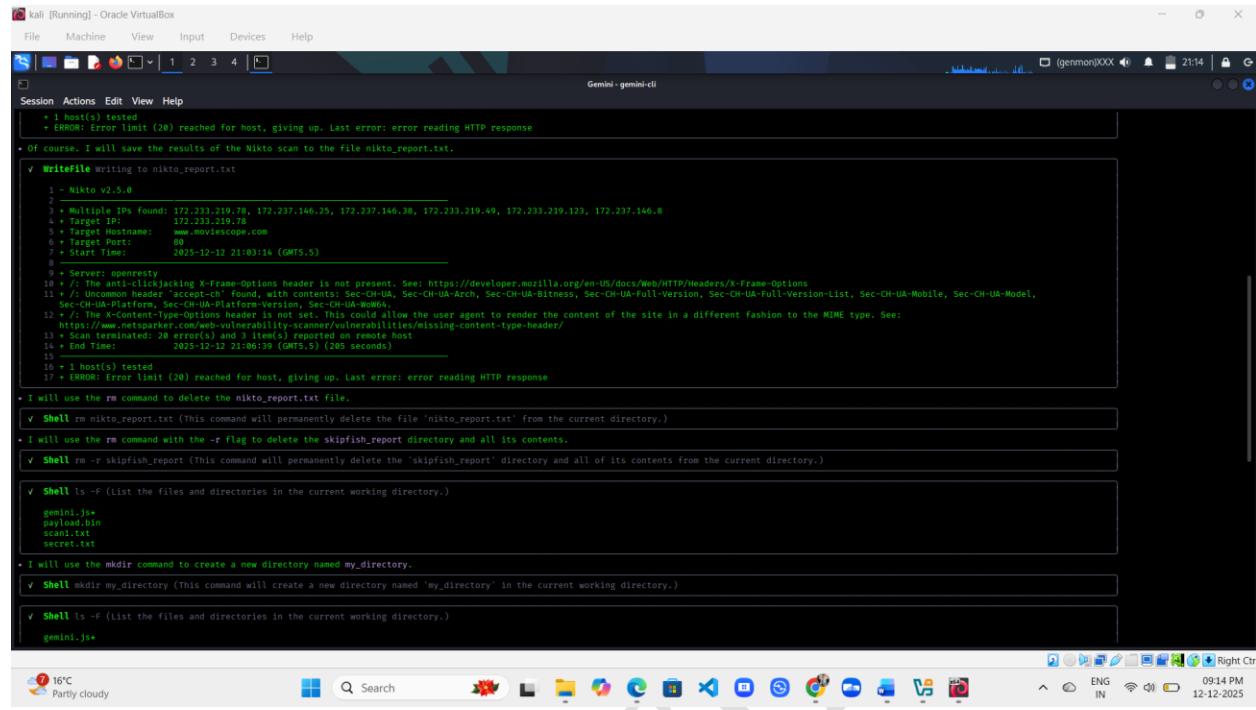
Search

EN IN

Right Ctrl

(genmon)XXX 21:12 12-2025

## MODULE – 5 VULNERABILITY ANALYSIS

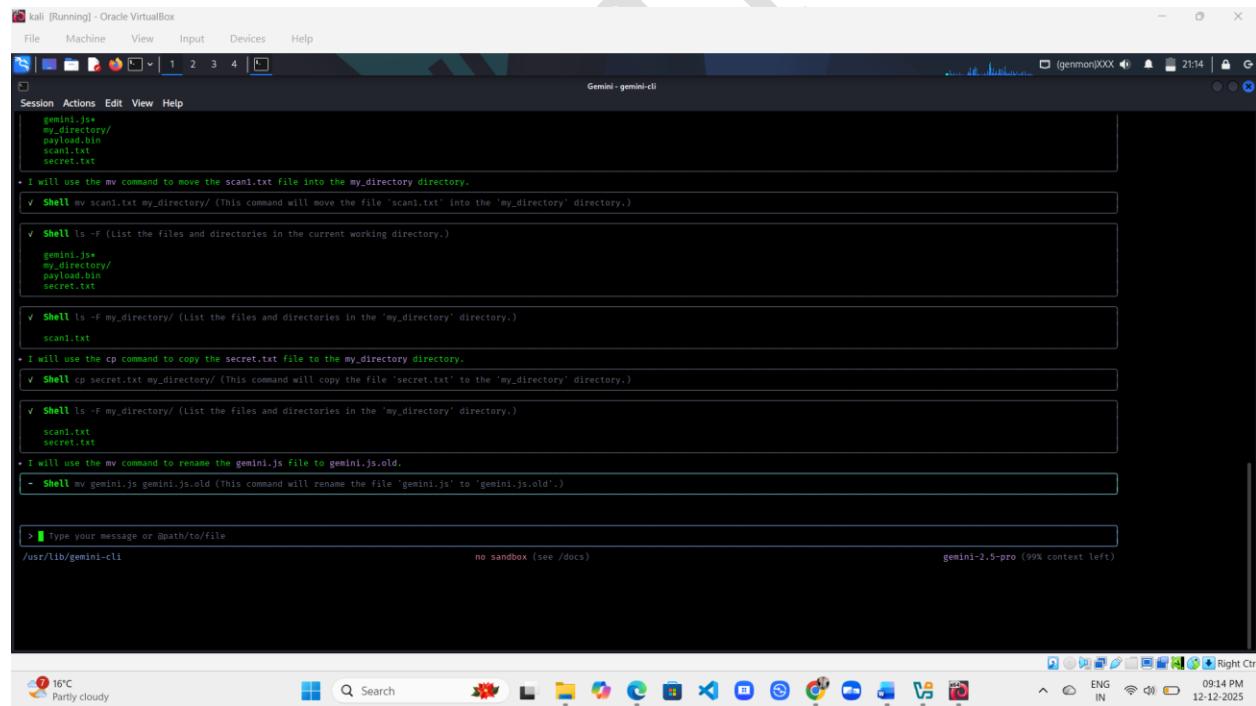


The terminal window shows the following session:

```

kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
Session Actions Edit View Help
+ 1 host(s) tested
ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
- Of course, I will save the results of the Nikto scan to the file nikto_report.txt.
v WriteFile writing to nikto_report.txt
1 - Nikto v2.5.0
2
3 + Multiple IPs found: 172.23.219.26, 172.237.146.25, 172.237.146.38, 172.233.219.49, 172.233.219.123, 172.237.146.8
4 + Target IP: 172.233.219.26
5 + Target Hostname: www.moviescope.com
6 + Target Port: 80
7 + Start Time: 2025-12-12 21:03:14 (GMTS.5)
8 + Server: openresty
9 + /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
10 + /: The Content-Security-Policy header is not present, with contents: Sec-CM-UA, Sec-CH-UA-Arch, Sec-CH-UA-Bitness, Sec-CH-UA-Full-Version, Sec-CH-UA-Full-Version-List, Sec-CH-UA-Mobile, Sec-CH-UA-Model,
11 + /: The Content-Security-Policy-Report-Only header is not present, with contents: Sec-CSP-RO-Arch, Sec-CSP-RO-Bitness, Sec-CSP-RO-Full-Version, Sec-CSP-RO-Full-Version-List, Sec-CSP-RO-Mobile, Sec-CSP-RO-Model,
12 + /: 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-vulnerability-scanner/vulnerabilities/missing-content-type-header/
13 + scan terminated: 20 errors(s) and 3 items(s) reported on remote host
14 + End Time: 2025-12-12 21:06:39 (GMTS.5) (205 seconds)
15
16 + 1 host(s) tested
17 + Known: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
- I will use the rm command to delete the nikto_report.txt file.
v Shell rm nikto_report.txt (This command will permanently delete the file 'nikto_report.txt' from the current directory.)
- I will use the rm command with the -r flag to delete the skipfish_report directory and all its contents.
v Shell rm -r skipfish_report (This command will permanently delete the 'skipfish_report' directory and all of its contents from the current directory.)
v Shell ls -F (List the files and directories in the current working directory.)
gemini.js*
payload.bin
scan1.txt
secret.txt
- I will use the mkdir command to create a new directory named my_directory.
v Shell mkdir my_directory (This command will create a new directory named 'my_directory' in the current working directory.)
v Shell ls -F (List the files and directories in the current working directory.)
gemini.js*

```



The terminal window shows the following session:

```

kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
Session Actions Edit View Help
gemini.js*
my_directory/
payload.bin
secret.txt
secret.txt
- I will use the mv command to move the scan1.txt file into the my_directory directory.
v Shell mv scan1.txt my_directory/ (This command will move the file 'scan1.txt' into the 'my_directory' directory.)
v Shell ls -F (List the files and directories in the current working directory.)
gemini.js*
my_directory/
payload.bin
secret.txt
secret.txt
- I will use the cp command to copy the secret.txt file to the my_directory directory.
v Shell cp secret.txt my_directory/ (This command will copy the file 'secret.txt' to the 'my_directory' directory.)
v Shell ls -F my_directory/ (List the files and directories in the 'my_directory' directory.)
scan1.txt
secret.txt
- I will use the mv command to rename the gemini.js file to gemini.js.old.
- Shell mv gemini.js gemini.js.old (This command will rename the file 'gemini.js' to 'gemini.js.old'.)

> Type your message or @path/to/file
/usr/lib/gemini-cl no sandbox (see /docs) gemini-2.5-pro (99% context left)

```

## MODULE – 5 VULNERABILITY ANALYSIS

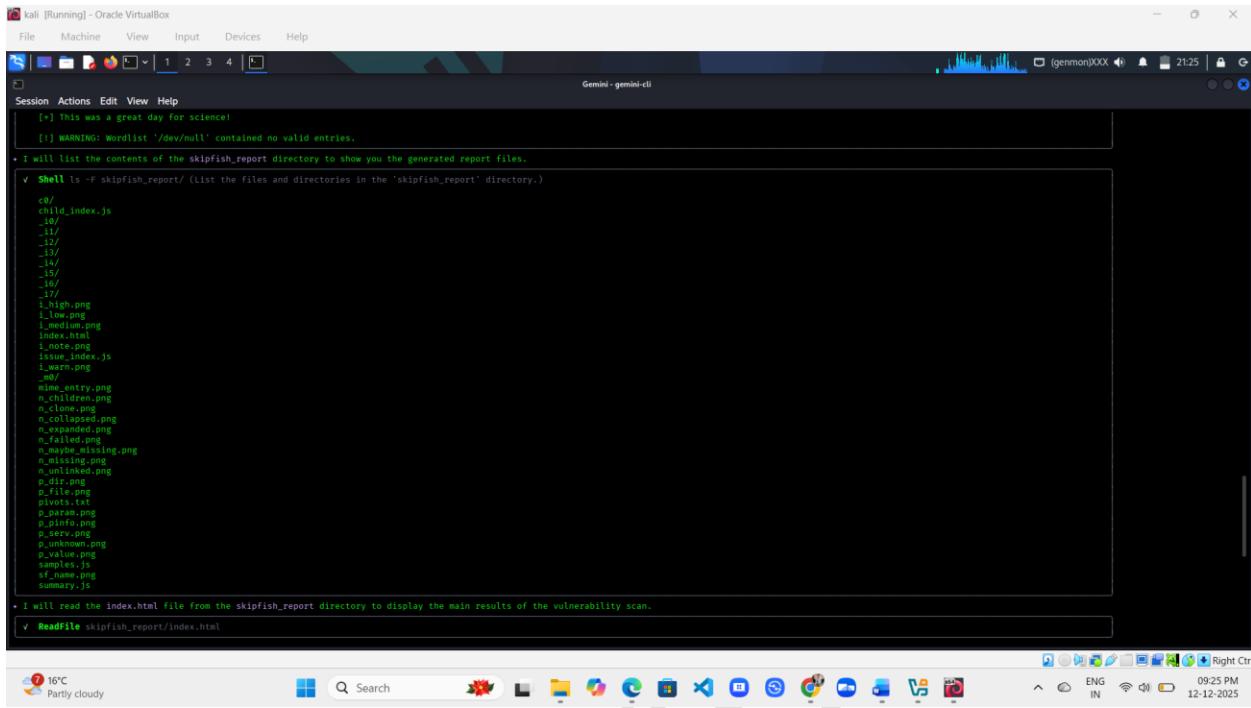
**Prompt :-:** Perform a vulnerability scan on target url <http://www.moviescope.com> with skipfish

```

kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
Shell skipfish -o skipfish_report http://testphp.vulnweb.com (This command will run a web application security scan with "skipfish" against the specified URL and save the report in the "skipfish_report" directory)
> Perform a vulnerability scan on target url http://www.moviescope.com with skipfish
* Will run a web application security scan using skipfish against the target URL http://www.moviescope.com. The scan results will be saved in a directory named skipfish_report.
V Shell skipfish -o skipfish_report http://www.moviescope.com (This command will run a web application security scan with "skipfish" against the specified URL and save the report in the "skipfish_report" directory)
skipfish web application scanner - version 2.10b
Welcome to skipfish. Here are some useful tips:
1) To abort the scan at any time, press Ctrl-C. A partial report will be written to the specified location. To view a list of currently scanned URLs, you can press space at any time during the scan.
2) Watch the number requests per second shown on the main screen. If this figure drops below 100-200, the scan will likely take a very long time.
3) The scanner does not auto-limit the scope of the scan; on complex sites, you may need to specify locations to exclude, or limit brute-force steps.
4) There are several new releases of the scanner every month. If you run into trouble, check for a newer version first, let the author know next.
More info: http://code.google.com/p/skipfish/wiki/KnowIssues
Press any key to continue (or wait 60 seconds) ... skipfish version 2.10b by lcamtuf@google.com
- www.moviescope.com -
Scan statistics:
  Scan time : 0:01:01.042
  HTTP requests : 0 (0.0/s)
  Compression : 0 KB in, 0 KB out (0.0 KB/s)
  HTTP Faults : 0 net errors, 0 proto errors, 0 retried, 0 drops
  TCP handshakes : 1 total (1.0 req/conm)
  TCP Faults : 0 failures, 0 timeouts, 0 purged
  External links : 0 skipped
  Reqs pending : 1
Database statistics:
  Pivots : 2 total, 1 done (50.00%)
  In progress : 0 pending, 1 init, 0 attacks, 0 dict
  Missing nodes : 0 spotted
  Node types : 1 serv, 1 dir, 0 file, 0 info, 0 unkno, 0 par, 0 val
  Issues found : 0 info, 0 warn, 0 low, 0 medium, 0 high impact
16°C Partly cloudy  Search  Right Ctrl
Gemini - gemini-cli
File Machine View Input Devices Help
Session Actions Edit View Help
TCP handshakes : 15 total (1.0 req/conm)
  TCP Faults : 0 failures, 0 timeouts, 0 purged
  External links : 0 skipped
  Reqs pending : 1
Database statistics:
  Pivots : 2 total, 1 done (50.00%)
  In progress : 0 pending, 1 init, 0 attacks, 0 dict
  Missing nodes : 0 spotted
  Node types : 1 serv, 1 dir, 0 file, 0 info, 0 unkno, 0 par, 0 val
  Issues found : 0 info, 0 warn, 0 low, 0 medium, 0 high impact
  Issues new : 0 new, 0 extensions, 0 candidates
  Signatures : 77 total
skipfish version 2.10b by lcamtuf@google.com
- www.moviescope.com -
Scan statistics:
  Scan time : 0:01:08.995
  HTTP requests : 29 (0.4/s), 38 KB in, 6 KB out (0.6 KB/s)
  Compression : 24 KB in, 45 KB out (30.0% gain)
  HTTP Faults : 2 net errors, 2 proto errors, 2 retried, 0 drops
  TCP handshakes : 15 total (1.0 req/conm)
  TCP Faults : 0 failures, 0 timeouts, 1 purged
  External links : 1 skipped
  Reqs pending : 0
Database statistics:
  Pivots : 3 total, 1 done (33.33%)
  In progress : 0 pending, 0 init, 0 attacks, 0 dict
  Missing nodes : 0 spotted
  Node types : 1 serv, 1 dir, 1 file, 0 info, 0 unkno, 0 par, 0 val
  Issues found : 0 info, 0 warn, 2 low, 2 medium, 0 high impact
  Dict size : 3 words (3 new), 0 extensions, 100 candidates
  Signatures : 77 total
(*) Copying static resources...
(*) Sorting and annotating crawl nodes: 3
(*) Looking for duplicate entries: 3
(*) Counting unique nodes: 3
(*) Generating static resources for third-party tools...
(*) Writing scan description...
(*) Writing crawl tree: 3
(*) Generating static resources views...
(*) Report saved to 'skipfish_report/index.html' [0x3a67c972].
(*) This was a great day for science!
(*) WARNING: Wordlist '/dev/null' contained no valid entries.
16°C Partly cloudy  Search  Right Ctrl
Gemini - gemini-cli
File Machine View Input Devices Help
Session Actions Edit View Help

```

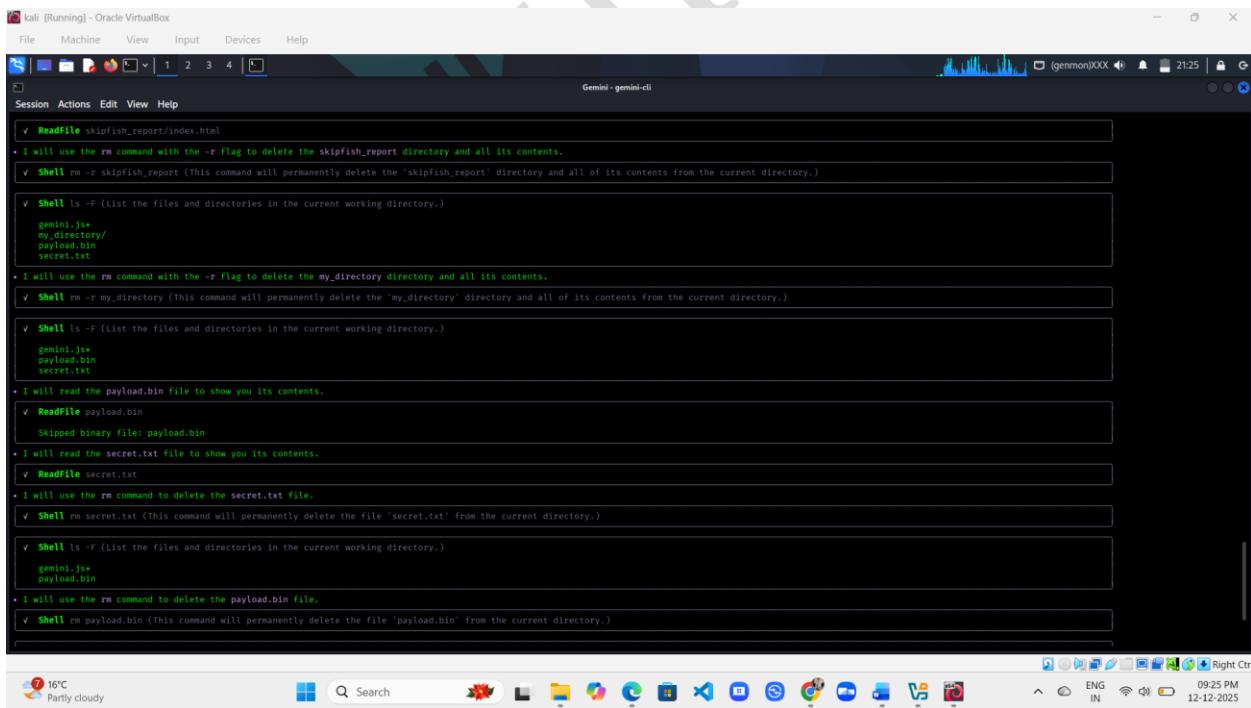
## MODULE – 5 VULNERABILITY ANALYSIS



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
[+] This was a great day for science!
[!] WARNING: Wordlist '/dev/null' contained no valid entries.

* I will list the contents of the skipfish_report directory to show you the generated report files.
v Shell ls -F skipfish_report/ (List the files and directories in the 'skipfish_report' directory.)
cd
child_index.js
.
11/
12/
13/
14/
15/
16/
1_low.png
1_high.png
1_low.png
1_medium.png
index.html
1_low.png
issue_index.js
1_warn.png
mine_entry.png
n_children.png
n_clone.png
n_expanded.png
n_expanded.png
n_failed.png
n_maybe_missing.png
n_low.png
n_unlinked.png
o_dir.png
p_file.png
p_low.png
p_param.png
p_pinfo.png
p_serv.png
p_value.png
samples.js
sf_name.png
summary.js

* I will read the index.html file from the skipfish_report directory to display the main results of the vulnerability scan.
v Readfile skipfish_report/index.html
```



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
[+] I will use the rm command with the -r flag to delete the skipfish_report directory and all its contents.
v Shell rm -r skipfish_report (This command will permanently delete the 'skipfish_report' directory and all of its contents from the current directory.)

v Shell ls -F (List the files and directories in the current working directory.)
gemini.js*
my_directory/
secret.txt

* I will use the rm command with the -r flag to delete the my_directory directory and all its contents.
v Shell rm -r my_directory (This command will permanently delete the 'my_directory' directory and all of its contents from the current directory.)

v Shell ls -F (List the files and directories in the current working directory.)
gemini.js*
payload.bin
secret.txt

* I will read the payload.bin file to show you its contents.
v Readfile payload.bin
Skipped binary file: payload.bin

* I will read the secret.txt file to show you its contents.
v Readfile secret.txt

* I will use the rm command to delete the secret.txt file.
v Shell rm secret.txt (This command will permanently delete the file 'secret.txt' from the current directory.)

v Shell ls -F (List the files and directories in the current working directory.)
gemini.js*
payload.bin

* I will use the rm command to delete the payload.bin file.
v Shell rm payload.bin (This command will permanently delete the file 'payload.bin' from the current directory.)
```

# Vulnerability Analysis – Report Section

Vulnerability analysis is that timeless craft of checking whether a system is as strong as it *claims* to be — kind of like stress-testing the walls of an old fortress before the next storm rolls in. Even though the digital world moves fast, this step has always been the backbone of real security work. And yeah, we question everything here, because trusting blindly is how breaches happen.

## What Is Vulnerability Analysis?

It's the systematic hunt for weaknesses in networks, applications, devices, and configurations. The goal is simple: spot the cracks before someone else slides through them. Think of it as checking the locks, windows, and doors of a digital house, but with scanners, scripts, and a sharp eye.

## Why It Matters

Because threats don't wait. A system left untested is basically an open invitation. Organizations need to know their flaws not for drama, but for survival. Fixing weak points early saves money, reputation, and a whole lot of headache.

## Core Steps in Vulnerability Analysis

### 1. Information Gathering

Old-school wisdom says: know your battlefield before you swing your sword. Collect details about systems, services, versions, and configurations.

### 2. Identifying Vulnerabilities

Using tools and manual checks, compare system data with known CVEs, misconfigurations, outdated software, weak passwords, and insecure services.

### 3. Risk Evaluation

Not all vulnerabilities are created equal. Some are minor annoyances, while others are digital landmines. Prioritize based on impact and likelihood.

### 4. Reporting & Recommendations

This is where clarity wins. Document findings, rating their severity, and suggest practical fixes — patching, reconfiguring, segmenting, hardening, or straight-up removing risky components.

## Types of Vulnerability Assessments

- **Network-based** – Scanning IP ranges and network devices for weak points
- **Host-based** – Digging into operating systems and local configurations
- **Application-based** – Checking web apps and software logic flaws
- **Wireless assessments** – Testing WLAN setups, encryption, rogue APs
- **Database assessments** – Permissions, injection risks, weak queries

- **Cloud assessments** – Modern needs meet old principles: access, storage, and service misconfigurations

## Common Tools Used

Traditional methods meet modern powerhouses:

- **Nessus**
- **Smart Scanner**
- **Zap**
- **Nmap (with NSE scripts)**
- **Acunetix**

These tools automate the grind but you still need human judgment — scanners can find doors, but only you can judge whether they matter.

## Key Outcomes

A proper vulnerability analysis gives:

- A mapped-out list of weaknesses
- Severity ranking
- Proof or evidence of issues
- Practical remediation steps
- A clearer security roadmap

# Conclusion

---

## Summary of Findings

The vulnerability assessment revealed that different tools—such as Nessus, Acunetix, ZAP, Smart Scanner, MBSA, and AI-based scanners—play unique roles in identifying security weaknesses across networks, systems, and web applications. Each tool helped uncover misconfigurations, outdated software, weak authentication, open ports, and other potential risks. By analyzing results through CVSS scoring and CWE references, the assessment provided a structured understanding of vulnerabilities, their severity, and the steps required for remediation.

---

## Importance of Continuous Security Monitoring

Security is not a one-time task—it's a constant cycle. New vulnerabilities appear daily, systems change, and attackers evolve. Continuous monitoring ensures that weaknesses are discovered early, patches are applied quickly, and threats are neutralized before they can be exploited. Regular scans, log monitoring, configuration reviews, and automated AI-based analysis help maintain a strong and adaptive security posture.

---

## Final Thoughts

A strong security strategy depends on both reliable tools and consistent human judgment. While scanners automate detection, it's the analyst who interprets risks, prioritizes fixes, and strengthens defenses. Continuous improvement, timely patching, and awareness of emerging threats keep systems resilient. Ultimately, vulnerability assessment is not just a technical practice—it's an ongoing commitment to protecting digital assets and ensuring long-term security.

**THANK YOU**

SACHCHITANAND