

VULNERABILITY ASSESSMENT & PENETRATION TESTING REPORT

PREPARED BY

NITHIN KRISHNA A

TABLE OF CONTENTS

1.....	Introduction
1.1.....	CIA Triad
1.2.....	OWASP Top 10
2.....	Scope of Testing
3.....	Tools Used for Pen Testing
4.....	Finding Severity ratings
5.....	Detailed Vulnerabilities
5.1.....	Vulnerabilities by host
5.2.....	Exploitable Vulnerabilities Report
5.3.....	Service Enumeration
5.4.....	Security Misconfiguration
5.5.....	Information Exposure Through Directory Listing
5.6.....	Sensitive data exposure
5.7.....	Apache outdated version 2.4.38
5.8.....	Vulnerable JavaScript dependency PORT:3873
5.8.1.....	Unencrypted communications
5.9.....	Unencrypted communications PORT:4449
5.10.....	Unencrypted communications PORT:3971
5.10.1.....	Frameable response (potential Clickjacking) PORT:3971
5.11.....	Vulnerable server nginx 1.19.7
6.....	Remediations for securing your system
7.....	Conclusion

1. INTRODUCTION

VAPT stands for Vulnerability Assessment and Penetration Testing. It is a process of identifying, assessing, and exploiting security vulnerabilities in computer systems, networks, applications, and other digital assets.

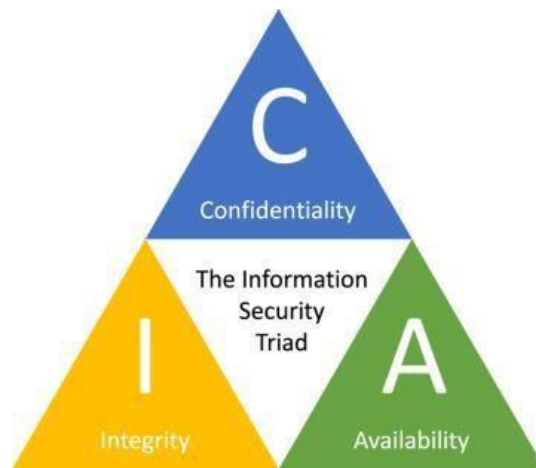
VAPT is important because it helps to identify vulnerabilities before they can be exploited by attackers. By performing VAPT, organizations can proactively identify security weaknesses and take steps to remediate them. This can help to prevent data breaches, theft of sensitive information, and other cyber-attacks.

Vulnerability Assessment (VA) is the process of identifying and assessing vulnerabilities in a system or network. It involves scanning the system or network for known vulnerabilities and misconfigurations, and then reporting on the vulnerabilities found. The output of a VA is a report that includes a list of identified vulnerabilities, along with their severity and recommendations for remediation.

Penetration Testing (PT), on the other hand, involves attempting to exploit the vulnerabilities identified in the VA to gain unauthorized access to the system or network. This is done in a controlled environment, and the results are used to identify vulnerabilities that could be exploited by attackers. The output of a PT is a report that includes a list of vulnerabilities successfully exploited, along with recommendations for remediation together, VA and PT provide a comprehensive approach to identifying and remediating security vulnerabilities. By performing VAPT on a regular basis, organizations can ensure that their digital assets are secure and protected from cyber-attacks.

1.1 CIA TRIAD IN CYBER SECURITY

The CIA Triad is an information security model, which is widely popular. It guides an organization's efforts towards ensuring data security. The three principles—confidentiality, integrity, and availability which is also the full for CIA in cybersecurity, form the cornerstone of a security infrastructure. In fact, it is ideal to apply these principles to any security program.



- **Confidentiality** makes sure that only authorized personnel are given access or permission to modify data.
- **Integrity** helps maintain the trustworthiness of data by having it in the correct state and immune to any improper modifications.
- **Availability** means that the authorized users should be able to access data whenever required.

The CIA Triad is so elementary to information security that anytime data violation or any number of other security incidents occur, it is due to one or more of these principles being compromised. So, the CIA Triad is always on top of the priority list for any infosec professional.

Security experts assess threats and vulnerabilities thinking about the impact that they might have on the CIA of an organization's assets. Based on that assessment, the security team enforces a specific set of security controls to minimize the risks within that environment.

1.2 OWASP Top 10

OWASP Top 10 is a list of the top 10 most critical web application security risks. The Open Web Application Security Project (OWASP) is a non-profit organization that aims to improve the security of software through open-source tools, resources, and guidance. The OWASP Top 10 is updated periodically to reflect changes in the threat landscape and to provide guidance on how to address common vulnerabilities.

The OWASP Top 10 is connected to VAPT in that it provides a framework for identifying and assessing security vulnerabilities in web applications. VAPT can help to identify and address the specific risks identified in the OWASP Top 10, such as injection flaws, broken authentication and session management, cross-site scripting (XSS), and other security issues.

By using the OWASP Top 10 as a reference, VAPT teams can ensure that they are covering the most critical web application security risks in their testing. They can also use the OWASP Top 10 to prioritize their remediation efforts, focusing on the most critical vulnerabilities first.

In addition, the OWASP Top 10 can be used as a guide for developers to build more secure web applications. By incorporating the security recommendations in the OWASP Top 10 into their development practices, developers can help to prevent vulnerabilities from being introduced in the first place. This can help to reduce the need for VAPT and improve the overall security of web applications.

[Here are the OWASP Top 10 web application security risks as of the latest release in 2021](#)

- [Injection](#)

Injection flaws occur when untrusted data is sent to an interpreter as part of a command or query, which can trick the interpreter into executing unintended commands or accessing unauthorized data.

- [Broken Authentication](#)

Broken authentication occurs when authentication and session management functions are not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to assume other users' identities.

- [Security Misconfiguration](#)

Security misconfiguration occurs when security settings are not configured properly, leaving security holes that can be exploited by attackers.

- [Insecure Design](#)

Insecure design occurs when a web application is designed in such a way that it can be easily exploited by attackers, for example, by allowing unauthenticated access to sensitive functionality.

- [Sensitive Data Exposure](#)

Sensitive data exposure occurs when sensitive data, such as passwords or credit card numbers, is not properly protected, leaving it vulnerable to attackers.

- [Vulnerable and Outdated Components](#)

Vulnerable and outdated components occur when a web application relies on third-party components that are not kept up to date, allowing attackers to exploit known vulnerabilities in these components.

- [Identification and Authentication Failures](#)

Identification and authentication failures occur when a web application does not properly identify or authenticate users, allowing attackers to assume other users' identities.

- **Software and Data Integrity Failures**

Software and data integrity failures occur when a web application does not properly validate or sanitize data, allowing attackers to modify or delete data or execute arbitrary code.

- **Security Logging and Monitoring Failures**

Security logging and monitoring failures occur when a web application does not properly log or monitor security events, making it difficult to detect and respond to attacks.

- **Server-Side Request Forgery (SSRF)**

Server-side request forgery (SSRF) occurs when a web application accepts user input that can be used to make server-side requests, which can be used by attackers to access internal resources or launch attacks against other systems.

2. SCOPE OF TESTING

Penetration testing is a type of security testing that involves simulating attacks on a computer system, network, or application to identify potential vulnerabilities and weaknesses that attackers may exploit. The scope in penetration testing involves the following:

Identifying the scope and objectives: The first step in penetration testing is to define the scope and objectives of the testing. This involves identifying the assets that need to be tested, such as applications, networks, servers, and databases.

Reconnaissance: The next step is to gather information about the target system, such as IP addresses, domain names, operating systems, and applications. This is done using various techniques, such as network scanning, port scanning, and OS fingerprinting.

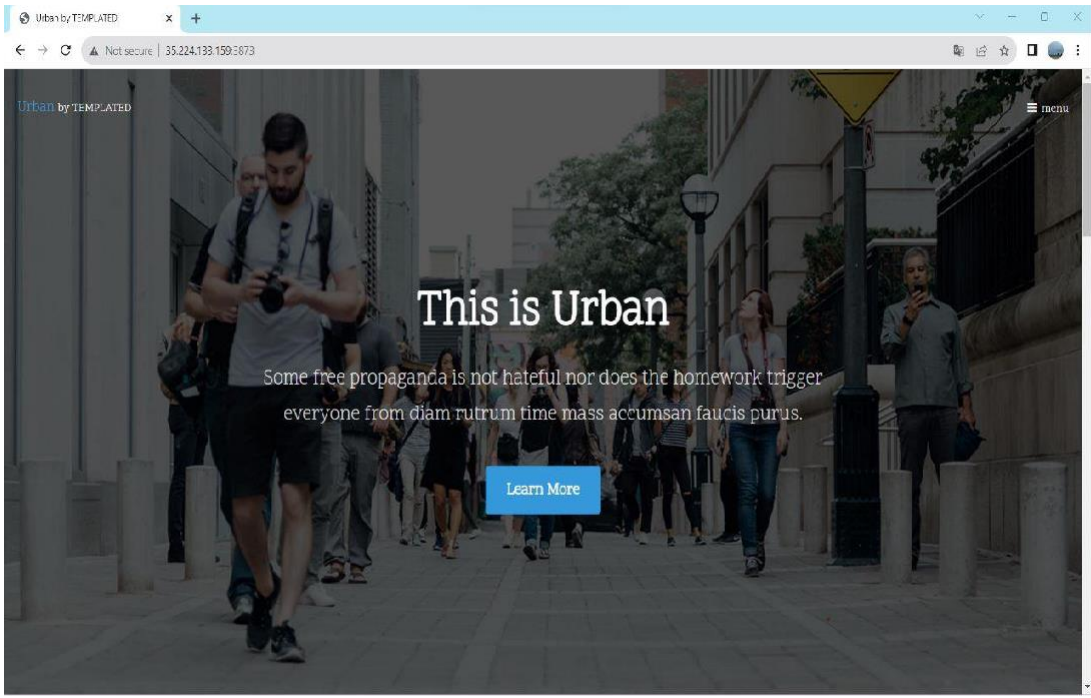
Vulnerability assessment: Once the reconnaissance phase is complete, the penetration tester will perform a vulnerability assessment to identify any weaknesses or vulnerabilities in the target system. This involves using tools such as vulnerability scanners, network analyzers, and web application scanners.

Exploitation: Once vulnerabilities are identified, the penetration tester will attempt to exploit them to gain unauthorized access to the target system. This involves using various techniques, such as SQL injection, cross-site scripting (XSS), and buffer overflow attacks.

Reporting: Finally, the penetration tester will document the results of the testing in a detailed report that includes the vulnerabilities found, their severity, and recommendations for remediation.

Overall, the scope of testing in penetration testing is to identify and assess the security posture of a target system, and to provide recommendations for improving its overall security.

SL NO	IP ADDRESS	IN SCOPE/OUT OF SCOPE
ASSET NO.1	35.224.133.159:3873	IN SCOPE



3. TOOLS USED FOR PENTESTING

Kali Linux is a popular Linux distribution used for digital forensics and penetration testing, and it comes with a variety of tools pre-installed. Here are some of the most used tools to penetrate the machine.

3.1 Nmap

Nmap is one of the most popular tools in Kali Linux and is widely used for network reconnaissance and vulnerability scanning. It can be used to scan networks and hosts for open ports, services running on those ports, and potential vulnerabilities in those service.

3.2 Dirsearch

Dirsearch is a command-line tool used for brute-forcing directories and files on web servers. It is pre-installed in Kali Linux and is a popular tool used in web application security testing.

3.3 Burpsuite

Burp or Burp Suite is a set of tools used for penetration testing of web applications. It is the most popular tool among professional web app security researchers and bug bounty hunters. Its ease of use makes it a more suitable choice.

3.4 Nessus

Nessus is a tool that checks computers to find vulnerabilities that hackers could exploit. Nessus works by testing each port on a computer, determining what service it is running, and then testing this service to make sure there are no vulnerabilities in it that could be used by a hacker to carry out a malicious attack.

4. FINDING SEVERITY RATINGS

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

5 DETAILED VULNERABILITIES

5.1 Vulnerabilities by host

35.224.133.159



Vulnerabilities					Total: 40
SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME	
CRITICAL	9.8	7.4	166677	PHP 8.0.x < 8.0.25 Multiple Vulnerabilities	
CRITICAL	9.1	6.3	169630	PHP 8.0.x < 8.0.27	
HIGH	8.8	6.7	161991	PHP 8.0.x < 8.0.20 Multiple Vulnerabilities	
HIGH	8.3	-	149348	PHP 7.4.x < 7.4.18 / 8.x < 8.0.5 Integer Overflow	
HIGH	7.5	3.6	146311	PHP 7.3.x < 7.3.27 / 7.4.x < 7.4.15 / 8.x < 8.0.2 DoS	
HIGH	7.5	3.6	171436	PHP 8.0.x < 8.0.28	
HIGH	7.5	-	179364	PHP 8.0.x < 8.0.30 Multiple Vulnerabilities	
HIGH	7.0	6.7	154296	PHP 8.0.x < 8.0.12	
MEDIUM	6.5	4.4	165622	PHP 8.0.x < 8.0.24 Multiple Vulnerabilities	
MEDIUM	6.1	5.7	136929	jQuery 1.2 < 3.5.0 Multiple XSS	
MEDIUM	5.3	2.9	155625	PHP 8.0.x < 8.0.13	
MEDIUM	4.3	2.9	177509	PHP 8.0.x < 8.0.29	
INFO	N/A	-	48204	Apache HTTP Server Version	
INFO	N/A	-	166602	Asset Attribute: Fully Qualified Domain Name (FQDN)	
INFO	N/A	-	39521	Backported Security Patch Detection (WWW)	
INFO	N/A	-	45590	Common Platform Enumeration (CPE)	
INFO	N/A	-	54615	Device Type	
INFO	N/A	-	43111	HTTP Methods Allowed (per directory)	
INFO	N/A	-	10107	HTTP Server Type and Version	

INFO	N/A	-	12053	Host Fully Qualified Domain Name (FQDN) Resolution
INFO	N/A	-	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	-	106658	JQuery Detection
INFO	N/A	-	11219	Nessus SYN scanner
INFO	N/A	-	19506	Nessus Scan Information
INFO	N/A	-	11936	OS Identification
INFO	N/A	-	117886	OS Security Patch Assessment Not Available
INFO	N/A	-	48243	PHP Version Detection
INFO	N/A	-	66334	Patch Report
INFO	N/A	-	122364	Python Remote HTTP Detection
INFO	N/A	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	149334	SSH Password Authentication Accepted
INFO	N/A	-	10881	SSH Protocol Versions Supported
INFO	N/A	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	10267	SSH Server Type and Version Information
INFO	N/A	-	22964	Service Detection
INFO	N/A	-	25220	TCP/IP Timestamps Supported
INFO	N/A	-	110723	Target Credential Status by Authentication Protocol - No Credentials Provided
INFO	N/A	-	10287	Traceroute Information
INFO	N/A	-	11154	Unknown Service Detection: Banner Retrieval
INFO	N/A	-	106375	nginx HTTP Server Detection

Suggested Remediations

Taking the following actions across 1 hosts would resolve 37% of the vulnerabilities on the network.

ACTION TO TAKE	VULNS	HOSTS
PHP 8.0.x < 8.0.30 Multiple Vulnerabilities: Upgrade to PHP version 8.0.30 or later.	12	1

5.2 Exploitable Vulnerabilities Report

Exploitable vulnerabilities create gaps in the network's integrity, which attackers can take advantage of to gain access to the network. Once inside the network, an attacker can perform malicious attacks, steal sensitive data, and cause significant damage to critical systems. This report provides a summary of the most prevalent exploitable vulnerabilities.

- Exploitable Vulnerabilities: Top 25

The Exploitable Vulnerabilities: Top 25 table uses the plugin attribute "exploit_available" to identify software that has working exploits in the wild. The data is then sorted using the count, which is a representation of the affected hosts. While some plugins may be present more than one time on a single host, for the most part a plugin will only be present once on each host. This list of vulnerabilities exposes the organization to many different attach frameworks and script kiddie attacks. These vulnerabilities should be prioritized and the software removed or updated to a supported version as soon as possible.

Severity (CVSS v3.0)	Plugin ID	Plugin Name	Count
MEDIUM	136929	JQuery 1.2 < 3.5.0 Multiple XSS	2

- Exploitable Vulnerabilities: Hosts by Plugin

The Exploitable Vulnerabilities: Hosts by Plugin table provides the IT operations team with an action plan and the identified hosts for each vulnerability. IT managers are able to use this information in planning patch deployments and in working with the information security team in risk mitigation efforts. The table also uses the plugin attribute "exploit_available" to identify exploitable software and then sorts the scan results using severity, then plugin ID. The entries in the "Hosts" column are then sorted in ascending order.

Severity (CVSS v3.0)	Plugin ID	Plugin Name	Hosts
MEDIUM	136929	JQuery 1.2 < 3.5.0 Multiple XSS	35.224.133.159

5.3 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

IP Address	Severity	Status
35.224.133.159	Medium	Open

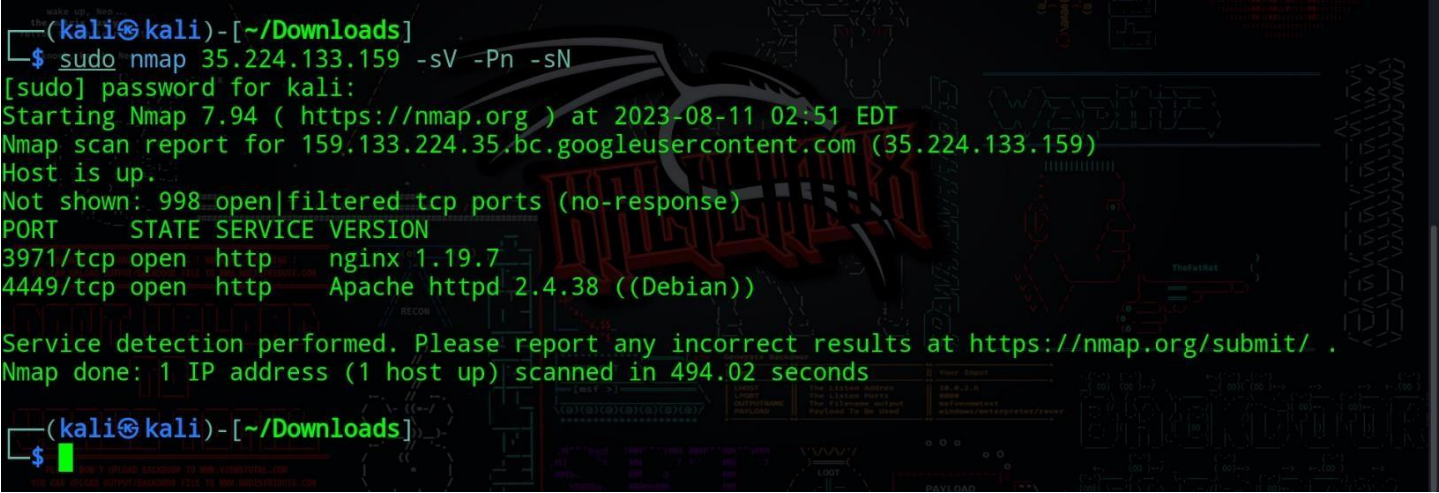
Impact:

Attacker can perform wide range of scanning like Services Enumeration, Ports, OS, Versions and Subnets.

Result of Service Enumeration:

- Host discovery
- Open ports
- Service detection
- Version detection

Screenshot:

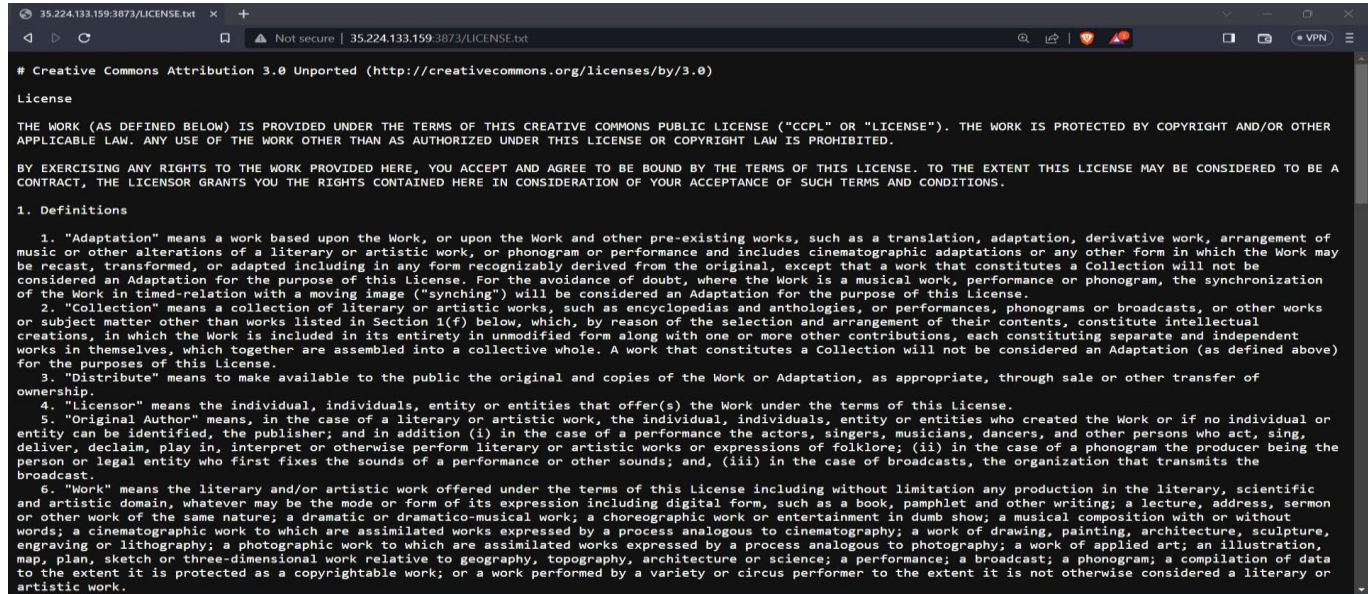


5.4 Security Misconfiguration

A Security Misconfiguration can cause web directory database data to be accessible through a basic web search or web page source code search if this includes sensitive data like License, administrator credentials an attacker can be able to launch another attack to the entire machine.

Affected Location	Severity	Status
35.224.133.159:3873	Critical	Open

Screenshot:



Recommendation:

Restrict Apache default configuration and only admin can access.

5.5 Information Exposure Through Directory Listing

The Information Exposure Through Directory Listing can be done using brute forcing attack. It is the process of requesting files and directories to which there are no direct links in the application or web server, this is usually done by getting the directory and filenames from common name list. Some of the details that can be obtained through directory brute forcing includes

- Directory and file names
- Hidden or sensitive content
- File types
- Web server configuration

Affected Location	Severity	Status
35.224.133.159:3873	Critical	Open

Evidence:

In Apache Configuration file .http access not purely configured with 403 this is the reason Using the world list -w/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt the attacker can do the directory brute forcing

Screenshot:

```
obadijah@je:~$
$ dirsearch -u 35.224.133.159:3873 -w /home/obadijah/.dirsearch/wordlists/rockyou.txt
/home/obadijah/.local/lib/python3.11/site-packages/requests/_init_.py:102: RequestsDependencyWarning: urllib3
(1.26.16) or chardet (5.1.0)/charset_normalizer (2.0.12) doesn't match a supported version!
warnings.warn("urllib3 ({}) or chardet ({})/charset_normalizer ({}) doesn't match a supported "
chir3z0z0i1ch v0.4.2

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /home/obadijah/.dirsearch/reports/35.224.133.159-23-08-10-12-35-24.txt
Error Log: /home/obadijah/.dirsearch/logs/errors-23-08-10-12-35-24.log
Target: http://35.224.133.159:3873/

[12:35:25] Starting:
[12:35:36] 403 - 201B - /.ht_wsr.txt
[12:35:36] 403 - 201B - /.htaccess.orig
[12:35:36] 403 - 201B - /.htaccess.sample
[12:35:36] 403 - 201B - /.htaccess.bak1
[12:35:36] 403 - 201B - /.htaccess.save
[12:35:36] 403 - 201B - /.htaccess_extra
[12:35:36] 403 - 201B - /.htaccessBAK
[12:35:36] 403 - 201B - /.htaccessOLD
[12:35:36] 403 - 201B - /.htaccessOLD2
[12:35:36] 403 - 201B - /.htaccess_sc
[12:35:36] 403 - 201B - /.htaccess.orig
[12:35:36] 403 - 201B - /.htm
[12:35:36] 403 - 201B - /.htpasswd
[12:35:36] 403 - 201B - /.htpasswd_test
[12:35:36] 403 - 201B - /.httr-oauth
[12:35:36] 403 - 201B - /.html
[12:35:50] 200 - 17KB - /LICENSE.txt
[12:36:16] 403 - 201B - /assets/
[12:36:17] 301 - 324B - /assets/ → http://35.224.133.159:3873/assets/
[12:36:38] 200 - 6KB - /home.html
[12:36:39] 301 - 324B - /images/ → http://35.224.133.159:3873/images/
[12:36:39] 403 - 201B - /images/
[12:36:40] 200 - 6KB - /index.php
[12:36:40] 200 - 6KB - /index.php/login/
[12:37:04] 403 - 201B - /server-status/
[12:37:04] 403 - 201B - /server-status

Task Completed
```

Reason For Brute Force Attack:

Brute-Force attacks are often used for attacking authentication and discovering directories within the web application. These attacks are usually sent via GET and POST request to the machine. In regard to authentication, brute force attacks are often mounted when an account lockout policy is not in place.

Recommendations:

- ◇ Purely Configure 403 in configuration file
- ◇ Place an WAF(Web Application Firewall)
- ◇ Implement authentication mechanism
- ◇ Prevent directory listing for all paths beneath the web root

5.6 Sensitive data exposure

Sensitive data exposure is a type of vulnerability that occurs when inadvertently exposes sensitive data such as passwords, credit card data, session tokens, or other authentication credentials. It differs from a data breach, in which an attacker accesses and steals information. Most major security breaches worldwide result in some kind of sensitive data exposure.

Affected Location	Severity	Status
35.224.133.159:3971	High	Open

Screenshot:



Under Construction!

Recommendations:

- ◇ Encrypt All Sensitive Data
- ◇ Don't Hold on to Sensitive Data:
- ◇ Classify data and ensure that no sensitive data is accessed by unauthorized users.

5.7 Apache outdated version 2.4.38

A vulnerability in Apache 2.4.38 is discovered it allow attackers to potentially perform HTTP Request Smuggling vulnerability in Apache HTTP Server via mod_proxy_uwsgi. This issue affects Apache HTTP Server: from 2.4.30 through 2.4.55. Special characters in the origin response header can truncate/split the response forwarded to the client.

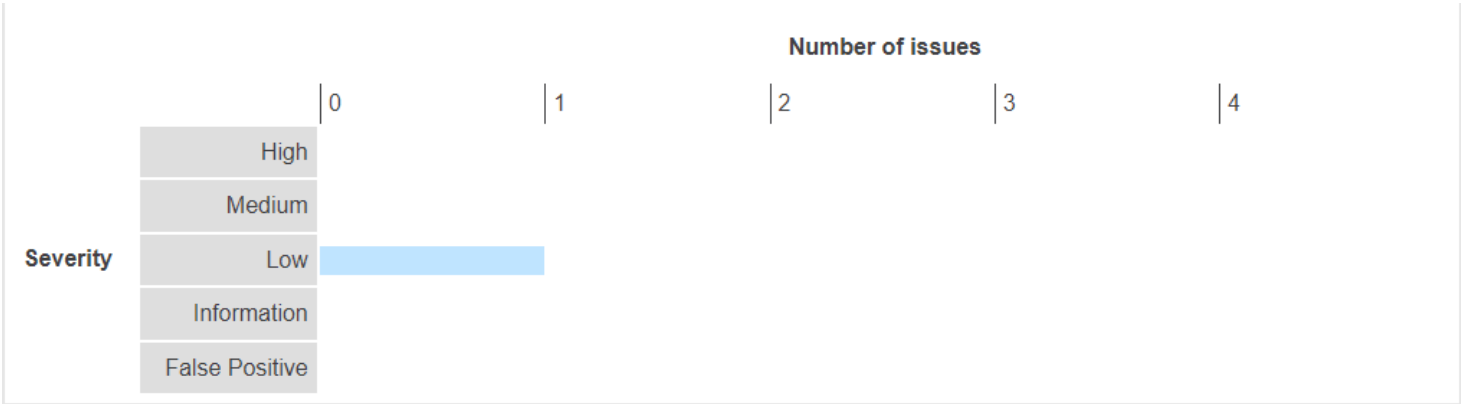
Affected Location	Severity	Status
35.224.133.159	Critical	Open

Severity: Low

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low, Information or False Positive. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.

		Confidence			Total
		Certain	Firm	Tentative	
Severity	High	0	0	0	0
	Medium	0	0	0	0
	Low	0	0	1	1
	Information	0	0	0	0
	False Positive	0	0	0	0

The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.



	Severity:	Low
	Confidence:	Tentative
	Host:	http://35.224.133.159:3873
	Path:	/assets/js/jquery.min.js

Request:

```
GET /assets/js/jquery.min.js HTTP/1.1
Host: 35.224.133.159:3873
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.199 Safari/537.36
Accept: */*
Referer: http://35.224.133.159:3873/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 11 Aug 2023 05:50:58 GMT
Server: Apache/2.4.38 (Debian)
Last-Modified: Sat, 16 Jan 2021 04:20:12 GMT
ETag: "176d5-5b8fcce105700-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Length: 95957
Connection: close
Content-Type: application/javascript

/*! jQuery v1.11.3 | (c) 2005, 2015 jQuery Foundation, Inc. | jquery.org/license */
!function(a,b){("object"==typeof module&&"object"==typeof module.exports?module.exports=a.document?b(a,!0):function(a){if(!a.document)th
...[SNIP]...
```

Remediations:

Develop a patch-management strategy to ensure that security updates are promptly applied to all third-party libraries in your application. Also, consider reducing your attack surface by removing any libraries that are no longer in use.

5.8.1 Unencrypted communications

The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.

To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a

compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

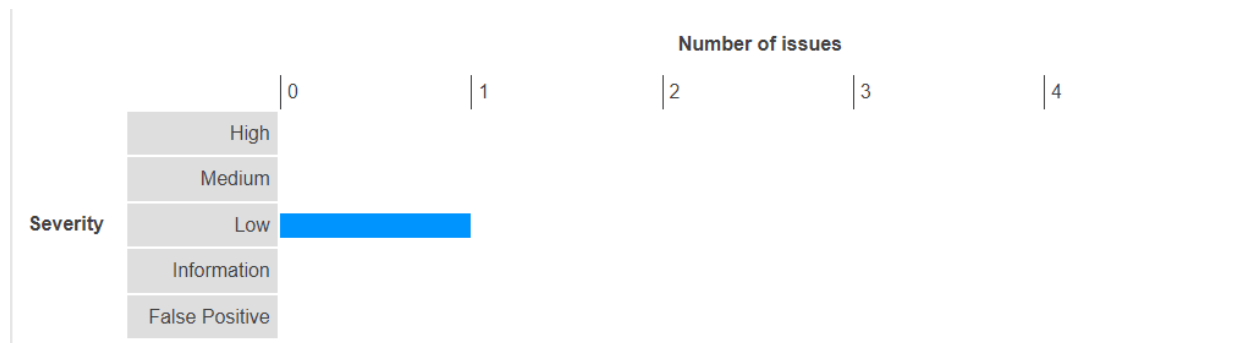
Please note that using a mixture of encrypted and unencrypted communications is an ineffective defense against active attackers, because they can easily remove references to encrypted resources when these references are transmitted over an unencrypted connection.

Severity: Low

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low, Information or False Positive. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.

		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	0	0	0	0
	Medium	0	0	0	0
	Low	1	0	0	1
	Information	0	0	0	0
	False Positive	0	0	0	0

The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.



	Severity:	Low
	Confidence:	Certain
	Host:	http://35.224.133.159:3873
	Path:	/

Remediations:

Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection.

5.9 Unencrypted communications PORT:4449

The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.

To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

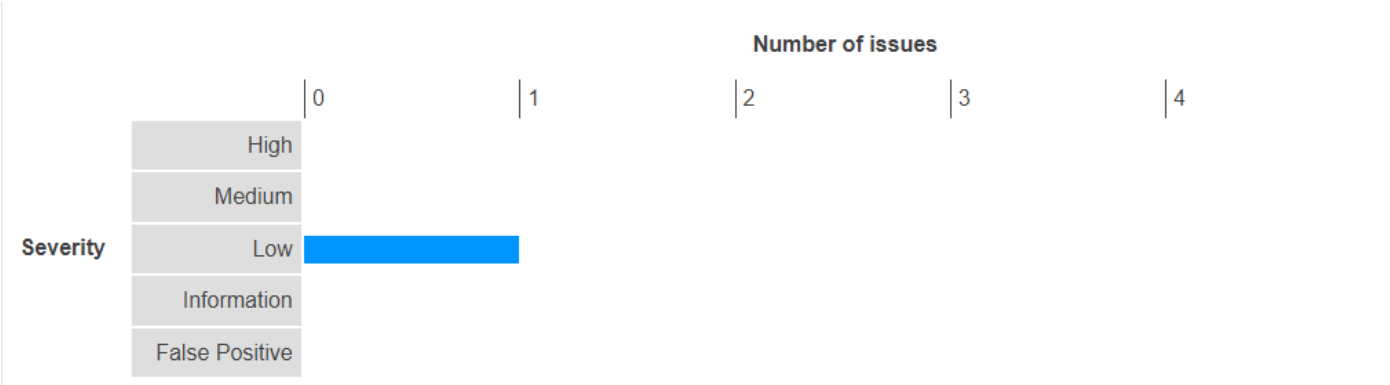
Please note that using a mixture of encrypted and unencrypted communications is an ineffective defense against active attackers, because they can easily remove references to encrypted resources when these references are transmitted over an unencrypted connection.


Severity: Low

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low, Information or False Positive. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.

		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	0	0	0	0
	Medium	0	0	0	0
	Low	1	0	0	1
	Information	0	0	0	0
	False Positive	0	0	0	0

The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.



	Severity:	Low
	Confidence:	Certain
	Host:	http://35.224.133.159:4449
	Path:	/

Remediations:

Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection.

5.10 Unencrypted communications PORT:3971

The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.

To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

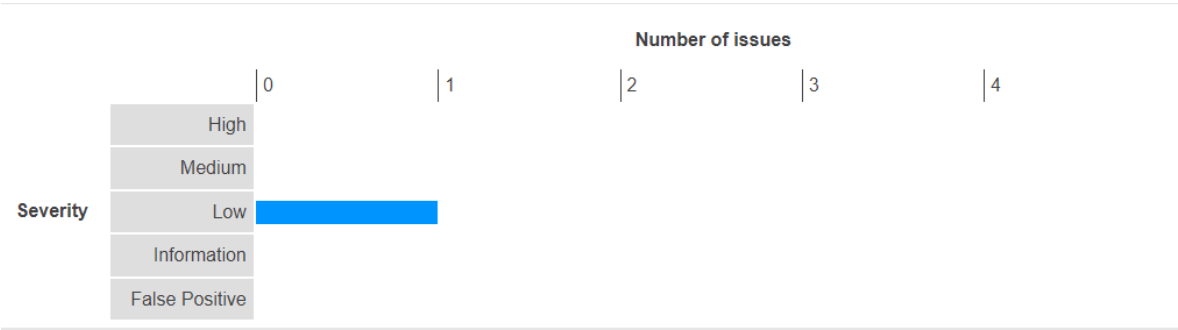
Please note that using a mixture of encrypted and unencrypted communications is an ineffective defense against active attackers, because they can easily remove references to encrypted resources when these references are transmitted over an unencrypted connection.


Severity: Low

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low, Information or False Positive. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.

		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	0	0	0	0
	Medium	0	0	0	0
	Low	1	0	0	1
	Information	0	0	0	0
	False Positive	0	0	0	0

The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.





Severity:	Low
Confidence:	Certain
Host:	http://35.224.133.159:3971
Path:	/

Remediations:

Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection.

5.10.1 Frameable response (potential Clickjacking) PORT:3971

If a page fails to set an appropriate X-Frame-Options or Content-Security-Policy HTTP header, it might be possible for a page controlled by an attacker to load it within an iframe. This may enable a clickjacking attack, in which the attacker's page overlays the target application's interface with a different interface provided by the attacker. By inducing victim users to perform actions such as mouse clicks and keystrokes, the attacker can cause them to unwittingly carry out actions within the application that is being targeted. This technique allows the attacker to circumvent defenses against cross-site request forgery, and may result in unauthorized actions.

Note that some applications attempt to prevent these attacks from within the HTML page itself, using "framebusting" code. However, this type of defense is normally ineffective and can usually be circumvented by a skilled attacker.

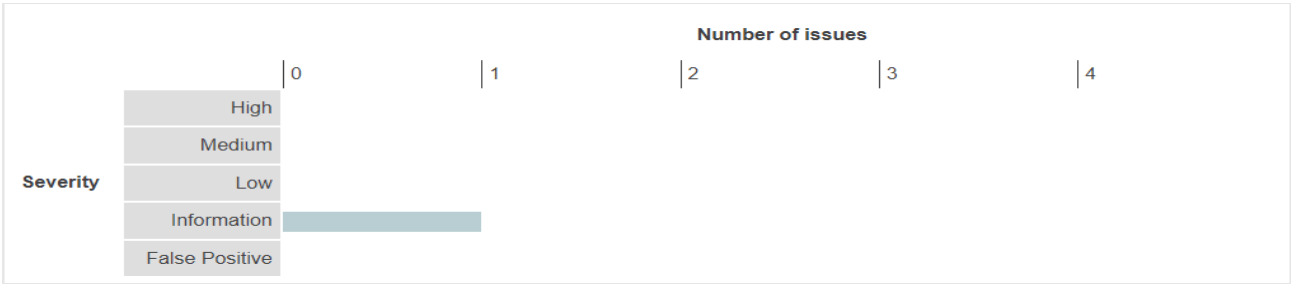
You should determine whether any functions accessible within frameable pages can be used by application users to perform any sensitive actions within the application.


Severity: Information

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low, Information or False Positive. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.

		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	0	0	0	0
	Medium	0	0	0	0
	Low	0	0	0	0
	Information	0	1	0	1
	False Positive	0	0	0	0

The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.



	Severity:	Information
	Confidence:	Firm
	Host:	http://35.224.133.159:3971
	Path:	/

Request:

```
GET / HTTP/1.1
Host: 35.224.133.159:3971
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.199 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response:

```
HTTP/1.1 200 OK
Server: nginx/1.19.7
Date: Fri, 11 Aug 2023 05:53:59 GMT
Content-Type: text/html
Content-Length: 748
Last-Modified: Sun, 07 Mar 2021 10:44:24 GMT
Connection: close
ETag: "6044ae88-2ec"
Accept-Ranges: bytes

<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/c
...[SNIP]...
```

Remediations:

To effectively prevent framing attacks, the application should return a response header with the name **X-Frame-Options** and the value **DENY** to prevent framing altogether, or the value **SAMEORIGIN** to allow framing only by pages on the same origin as the response itself. Note that the SAMEORIGIN header can be partially bypassed if the application itself can be made to frame untrusted websites.


5.11 Vulnerable server nginx 1.19.7

Exposure of Resource to Wrong Sphere

When doing HTTP(S) transfers, lib curl might erroneously use the read call back `CURL_OPT_READFUNCTION` to ask for data to send, even when the `CURLOPT_POSTFIELDS` option has been set, if the same handle previously was used to issue a PUT request which used that callback. This flaw may surprise the application and cause it to misbehave and either send off the wrong data or use memory after free or similar in the subsequent POST request. The problem exists in the logic for a reused handle when it is changed from a PUT to a POST.

Screenshot:

CRITICAL SEVERITY

 Exposure of Resource to Wrong Sphere

Vulnerable module: curl

Introduced through: curl@7.64.0-4+deb10u2 and curl/libcurl4@7.64.0-4+deb10u2

Fixed in: 7.64.0-4+deb10u4

Detailed paths

Introduced through: nginx@1.19 › curl@7.64.0-4+deb10u2

Introduced through: nginx@1.19 › curl/libcurl4@7.64.0-4+deb10u2

Directory Traversal

Dpkg: Source:Archive in dpkg, the Debian package management system, before version 1.21.8, 1.20.10, 1.19.8, 1.18.26 is prone to a directory traversal vulnerability. When extracting untrusted source packages in v2 and v3 source package formats that include a debian.tar, the in-place extraction can lead to directory traversal situations on specially crafted orig.tar and debian.tar

Screenshot:

CRITICAL SEVERITY

 **Directory Traversal**

Vulnerable module: dpkg
Introduced through: dpkg@1.19.7
Fixed in: 1.19.8

Detailed paths
Introduced through: nginx@1.19 > dpkg@1.19.7

6. Remediations for securing your system

Keep software up to date: Ensure that all software on your machine, including operating system, web browsers, and other applications, are up to date with the latest security patches and updates.

Use strong passwords: Use strong and unique passwords for all your accounts and change them regularly. Consider using a password manager to store your passwords securely.

Use a firewall: Use a firewall to control incoming and outgoing traffic to your machine and configure it to block unnecessary ports and services.

Enable automatic updates: Enable automatic updates for your operating system and other applications to ensure that you receive critical security updates in a timely manner.

Use anti-virus and anti-malware software: Use reputable anti-virus and anti-malware software to protect your machine from viruses, malware, and other threats.

Use encryption: Use encryption to protect sensitive data on your machine, including full disk encryption for your hard drive, and encryption for sensitive files and communications.

Disable unnecessary services: Disable unnecessary services and ports that are not required for your machine's operation and consider removing or disabling software that you no longer need.

Backup important data: Regularly backup important data on your machine to ensure that you can recover from a security incident or hardware failure.

By taking these steps, you can help protect your machine from vulnerabilities and reduce the risk of security incidents. Additionally, it's important to stay informed about new security threats and vulnerabilities and to regularly review and update your security practices to ensure that you are adequately protected.

7. Conclusion

In conclusion, machine penetration testing is a critical aspect of any comprehensive security program. By identifying vulnerabilities and weaknesses in your systems and applications, you can take proactive steps to improve security and reduce the risk of a security breach.

During a machine penetration test, various tools and techniques are used to simulate an attack on your system and identify potential security weaknesses. These may include vulnerability scanning, network mapping, exploitation of known vulnerabilities, and brute-force attacks on login credentials.

To ensure that your machine is secure, it's important to take the findings of the penetration test seriously and take appropriate actions to address any vulnerabilities identified. This may include applying security patches, configuring access controls, implementing strong authentication mechanisms, and performing regular security testing.

Additionally, it's important to work closely with your security team or a trusted third-party security consultant to develop a comprehensive security plan that addresses your specific needs and risks.

Overall, machine penetration testing is an important tool in the ongoing effort to maintain a strong and secure computing environment. By being proactive about security and addressing vulnerabilities before they can be exploited, you can help protect your systems and data from unauthorized access and theft.