Vulnerability Assessment and Penetration Testing of “Riipen” Web Application.

Ankan Garg
Department of Computer Science New York Institute of Technology
Vancouver,
Canada NYIT ID:
1299560
agarg07@nyit.edu

Felix Majemite
Department of Computer Science New York Institute of Technology
Vancouver,
Canada NYIT ID:
1275854
fmajemit@nyit.edu

Taiwo Mebude
Department of Computer Science New York Institute of Technology
Vancouver,
Canada NYIT ID:1305695
tmebude@nyit.edu

***Abstract*— This report presents the current progressive development, success, and failure of vulnerability assessment and Penetration testing of Riipen. Riipen is a web application that provides services to bridge the gap between the Company and the University regarding availability and accessibility to enterprise projects. This website helps both parties, such as employers, Educators, and Students, to have continuous opportunities to work together with everyone accomplished.**

Keywords*— *Riipen, WebApp, OWASP Top10, ZAP, recon, Attacks, Vulnerabilities, remediations, Tools, nmap, reverse lookup, Kali Linux, Scripts, Cyber Killchain

I. INTRODUCTION

This report presents a comprehensive web penetration test conducted on the web application of Riipen, a leading experiential learning platform. Riipen distinguishes itself by providing educational institutions and students with a wide range of remote-friendly opportunities, bridging the gap between educators and employers1.

The primary objective of this web penetration test was to identify potential vulnerabilities in Riipen's web application and propose remediation strategies to enhance its security. This is crucial as web application vulnerabilities, such as SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF), can be exploited by attackers to compromise the application.

II. PROBLEM STATEMENT

The RIIPEN web application could be susceptible to various logical and technical vulnerabilities. Technical vulnerabilities such as SQL injection, cross-site scripting, broken authentication, and security misconfiguration are potential threats that could compromise the web application's security. These vulnerabilities are significant as they can undermine the security of web applications. The causes of these vulnerabilities can range from poor programming practices to outdated systems.

III. PROJECT OBJECTIVES

The central objective is integrating penetration testing practices within an educational framework. This encompasses acquiring knowledge, implementing practical exercises, and iteratively enhancing ethical hacking skills. The overarching goal is to systematically dissect attack patterns, identify vulnerabilities, and develop effective mitigation strategies through hands-on experience and learning,

fostering a robust understanding of cybersecurity in an educational context.

IV. REVIEW OF RELATED WORK

In every project, related work is usually integral; we shall extensively use the MITRE framework, considered the Bible for pen testers. We shall utilize various pentest reports, online best practices, and other related frameworks to solidify our endeavors.

V. METHODOLOGY AND DESCRIPTIONS

The primary aim of this project is to conduct a comprehensive penetration test on a web application system, encompassing five pivotal tasks: reconnaissance, enumeration, exploitation, post-exploitation/maintaining access, and covering tracks.

These objectives are

Reconnaissance: Thoroughly gather information and intelligence about the target web application to

understand its structure, architecture, and potential vulnerabilities.

Enumeration: Systematically identify any exposed services, applications, or weaknesses within the web application, aiming to catalog possible entry points.

Exploitation: Actively assess and exploit identified vulnerabilities to ascertain the extent of potential security breaches, emphasizing ethical and controlled exploitation.

Post-Exploitation/Maintaining Access: After gaining initial access, establish mechanisms to maintain a persistent presence within the system, simulating how a malicious actor might persist within a compromised environment.

Pre-Engagement: Permission seeking or request is essential to ethical hacking practices. Personnel from Riipen with authority gave the go-ahead for the possibility of forms of pentesting done and recorded in this paper.

VI. SUB-TECHNIQUES IN VARIOUS METHODOLOGY

A. Reconnaissance Concepts

Spidering: a concept of crawling through a website and following all the links on the site, creating a working construct of the web application's functionalities[1].

Forced Browsing: though similar to crawling, more importantly, it finds hidden connected interfaces. This helps to build a profile of how the web application is put together[1].

Directory Transversal: this goes beyond the web application construction information gathering and towards the host machine the WebApp is running on[1].

Banner grabbing: collecting information on the web server and framework[1].

Server Fingerprinting: using tools that help scan Host details with a narrower focus, such as Nmap properties[1].

Documentation look-up: this is a good way to identify APIs and know if their elements are vulnerable[1].

Vulnerability Database: these accounts of real-life documented vulnerabilities are available for look-up and cross-reference analysis, such as Common Vulnerability and Exposure, Exploit Database, and Shodan[1].

B. Reconnaissance Tools Used And Relationship To Concepts

Numerous tools are available and used during this project; some can accomplish multiple concepts due to the improvement of technology and their automated ability. Furthermore, these tools can conduct reconnaissance in passive and active states[2].
Some of these tools figures can be found in Appendix B for illustration.

I. **Host:** The `host` command serves the purpose of querying Domain Name System (DNS) servers to perform domain name-to-IP address or IP address-to-domain name resolution.

II. **Nslookup:** "Nslookup" is a shortened form of "name server lookup," a utility that facilitates the querying of Domain Name System (DNS) services. This tool is commonly employed through a command line interface (CLI) to retrieve domain names, retrieve IP address mapping information, and perform DNS record lookups.

III. **Wappalyzer:** Wappalyzer is a technology profiling tool that provides insights into the technological stack powering websites. It can identify the Content Management System (CMS), frameworks, e-commerce platforms, JavaScript libraries, and other components used to construct a website.

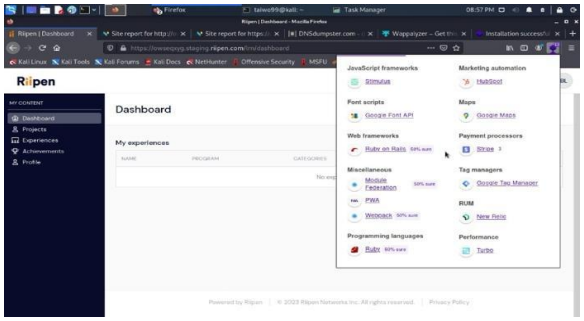


Fig 1: Wapalyzer demonstrates web technologies utilized by Riipen.

IV. **Nikto** is an open-source command-line vulnerability scanning tool that conducts thorough assessments of web servers to detect potentially hazardous files, Common Gateway Interfaces (CGIs), obsolete server software, and other security-related issues. Fig 5. Nikto carried out the depicted vulnerability scanning result

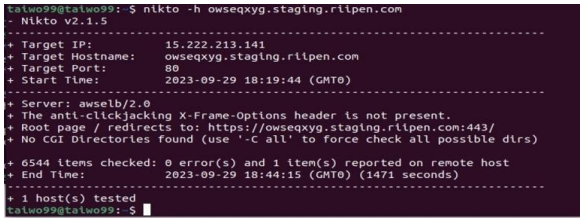


Fig 2. Depict vulnerability scanning result carried by Nikto

V. **Wafw00f:** Wafw00f is an open-source tool specifically developed for identifying and fingerprinting Web Application Firewalls (WAFs).

Fig 3. Showing recon results for a WAF scan

VI. **Nessus**, a creation by Tenable, is a robust platform meticulously crafted to scan for security vulnerabilities across an extensive spectrum of targets, including devices, applications, operating systems, cloud services, and various network resources.

VII. **Sniper:** This is an automated scanner utilized within the context of a penetration test. Its purpose is to enumerate and scan for vulnerabilities systematically.

Enumeration/ findings

In pentesting, a close relationship exists between reconnaissance and enumerations, which refers to the next step after recon. Documentation of notable properties of Riipen as the target website after recon scans are as follows:

SSL/TLS vulnerability: The web URL is not SSL vulnerable. It is not based on TLS1.1 and TLS1.2, which are vulnerable versions. The overall ranking of the certificate looks good.

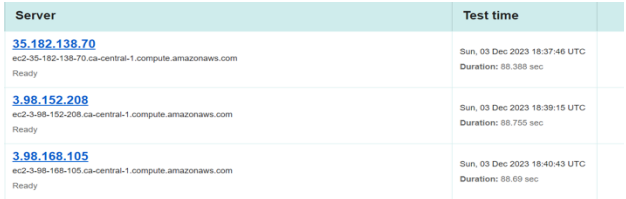


Fig 6. SSL/TLS check

EMAIL/SPAM Spoofing: Email spoofing is a technique used by cybercriminals to send emails that appear to come from a known, trusted source to trick the recipient. It exploits the fact that most email protocols do not authenticate the sender's address, allowing it to be easily forged.

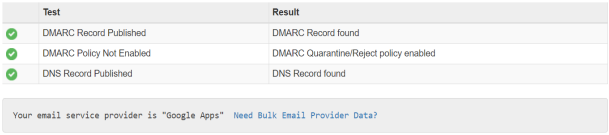


Fig 7. EMAIL Protocol checking

Result : DKIM, DMARC, and SPF are all set. Hence, it's not vulnerable to email spoofing or spamming attacks.

Clickjacking: Clickjacking is a cyber-attack where a malicious website tricks a user into clicking on something different from what they think they are clicking. The attacker overlays transparent buttons or Other interface elements over a legitimate website in a hidden iframe.

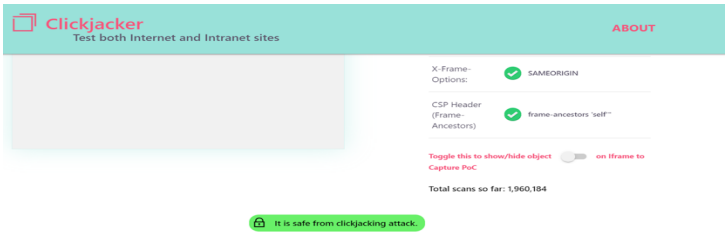


Fig 8. Clickjacking attack

Result: Upon performing header analysis, it's clear that the website is not prone to clickjacking attacks.

Broken link Hijacking: Broken link hijacking is an attack where cybercriminals exploit expired or inactive external links on websites to carry out malicious activities like defacement, phishing, and impersonation. Attackers hijack the broken links by registering expired domains or replacing passive resources.

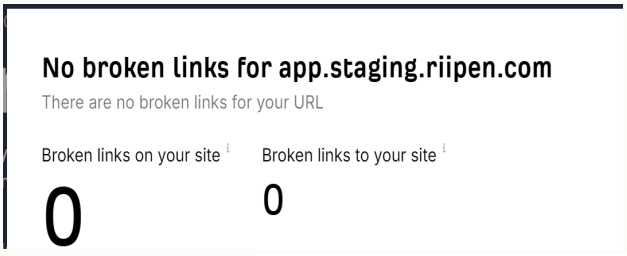


Fig 9. Broken link check

Result: The web URL is free from dead or unclaimed links. Hence, it is safe from this attack.

Security Headers:

HTTP response headers instruct browsers on how to handle web content securely. Implementing headers like Content-Security-Policy, X-Frame-Options, and Strict-Transport-Security can mitigate common attacks like cross-site scripting, clickjacking, and man-in-the-middle attacks. Security headers restrict dangerous functionality in browsers and limit the attack surface for malicious actors. Using proper security headers is a simple yet effective way to strengthen the security posture of a website. They are an essential line of defense that reduces application vulnerabilities and protects users' data and privacy.

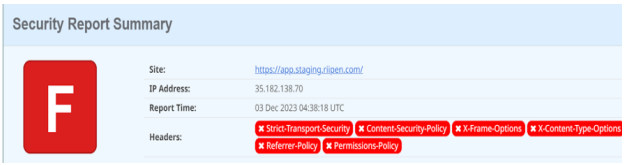


Fig 10. Security headers validation

Results: As per the data shared, the web URL is missing a few security headers. which are as follows: Strict-transport-security, content-security-policy, X-Frame-options, X-Content-Type-options, Referrer-policy and Permission- policy. explanations are found in the Appendix C section.

DNS spoofing The classic DNS poisoning attack is to send a DNS server a query that you think will cause the server to do a recursive lookup and then blast it with spoofed responses for that lookup before the real ones can arrive. The server will cache the first response, which will be yours.

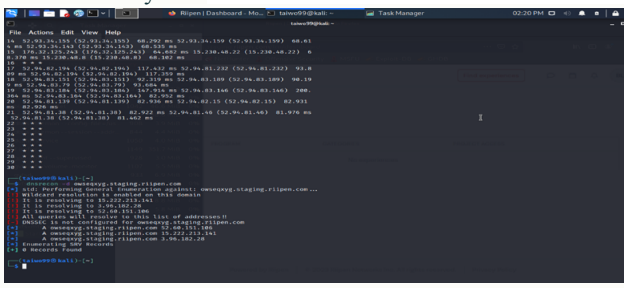


Fig 11. DNSSEC check

Result: It can be observed from our enumeration that DNSSEC is not enabled. This should be enabled to prevent such kinds of attacks and reduce the overall attack surface.

Reverse IP domain check: This was performed on the various IP addresses hosted by the website to understand if the domain is also getting shared with other customers.

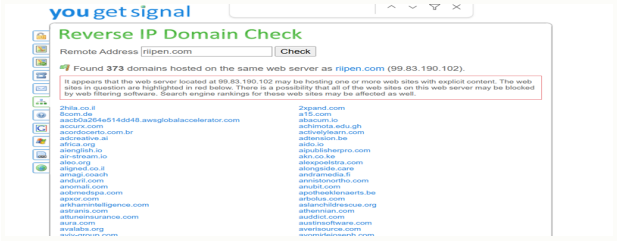


Fig 12. Reverse lookup IP

Result: It was noticed that the production website is sharing the hosting space with other real-world applications, which is a risk. All the applications listed above seem to be hosted on the same AWS instance.

Identity and Access Management misconfig: Authentication and authorization are important mechanisms in any organization. It enforces the "Principle of Least Privilege" and ensures the right user can access an organization.

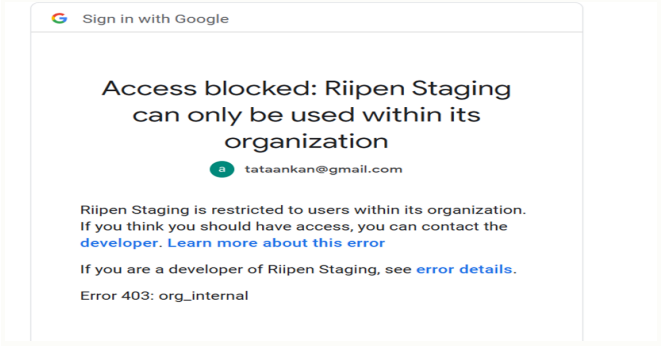


Fig 13. Identity check

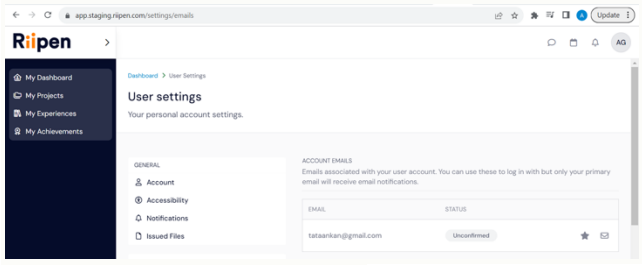


Fig 14. Identity management issues

Result: As can be seen, the in-staging environment, only .riipen.com accounts, should have access, but since it was misconfigured, we were logged in via gmail.com as well.

Pre-account takeover: As part of the initial survey, we tried exploiting the create account field to create various accounts.

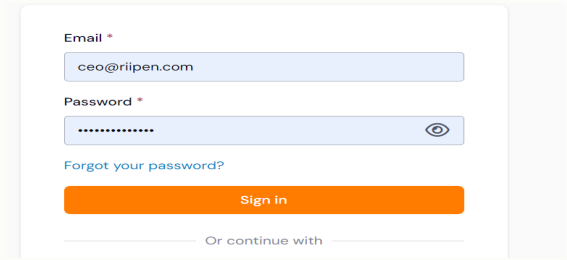


Fig 15. Pre-account of CEO/CTO

Result: We were able to create an account using ceo@riipen.com and cto@riipen.com successfully. The same vulnerability has been informed to the customer, and they are working on it.

The session does not expire after changing the password: It's a vulnerability where a user is not logged out from the session even if the password for the account is changed.

Steps to reproduce and results - The application is vulnerable to this attack. The steps followed are listed below:
- You need to log in with the same ID in two browsers
- Attacker browser - firefox
- Victim browser - chrome
- You can change the password in Chrome and go back to Firefox
- Update the first name or any information in firefox browser
- If the information is updated or the session is still live.

Exploitation Trails

After conducting reconnaissance and enumeration, three targeted attacks were executed based on the findings. These included a website directory traversal attack, a login bypass using SQLmap, and API fuzzing. The purpose was to identify and address vulnerabilities related to unauthorized access and potential weaknesses in the login system and application programming interfaces.

- **Website Directory Traversal Attack**: A Web Directory Traversal attack involves an attacker exploiting a vulnerability in a website's server to gain unauthorized access to restricted files and directories. This is achieved by manipulating the URL with special characters, allowing the attacker to access files and directories not intended for public access.
- To execute the Web Directory Traversal attack, we employed **dotdotpwn**. This tool serves as a versatile and intelligent fuzzer.
  Designed to identify traversal directory vulnerabilities in various software, including HTTP/FTP/TFTP servers and web platforms such as CMSs, ERPs, and Blogs. It enables systematic testing for directory traversal vulnerabilities, enhancing the attack's efficiency.

Figures 15 & 16 illustrate the outcomes of conducting fuzzing on the Ripen website to identify potential vulnerabilities. The results include the output of malicious special characters that can be utilized to manipulate the website, making it susceptible to a web directory traversal attack. These findings highlight potential weaknesses that should be addressed to enhance the website's security. Following the active fuzzing of the website, a minor timeout was experienced. However, further analysis will determine whether this was a successful denial of service attack or a potential hoax.

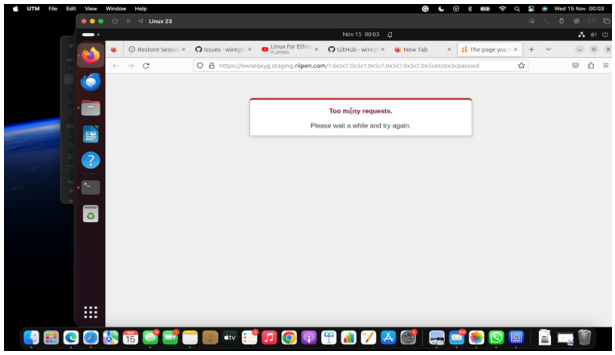


Fig 17. The image indicates a URL timeout, suggesting that the website was inaccessible or took too long to respond during the testing.

Further Analysis: After testing the Ripen URL with the potentially harmful characters identified by dotdotpwn in a web browser, it was observed that most URLs could not be accessed. This indicates that the website has some protection against the web directory traversal attack. all images are found in Appendix B according to the figure numbers.

Internal IP exposure: The disclosure of internal IP addresses on public-facing web applications is a security risk that can provide attackers with valuable information about an organization's internal network and systems. Exposed internal IPs enable attackers to gather details about the network architecture, which can then inform exploitation attempts.

Results: We discovered a few IPs that are not public IP addresses. They are of internal IP addresses and pose a great risk. Upon further investigation, we found out that those are hosted as HTTP, which makes it more vulnerable since the traffic is all unencrypted. Fig 18 captures the same as explained above.

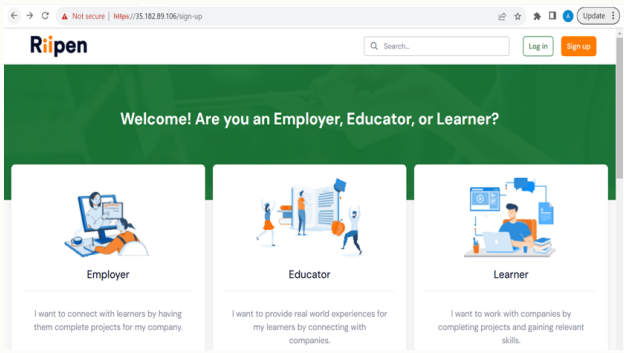


Fig 18. Internal IP exposure.

SQL Injection

SQL injection is a code injection technique aimed at data-driven applications. In this attack, the goal is to bypass the login page of the Riipen website by injecting malicious SQL codes.

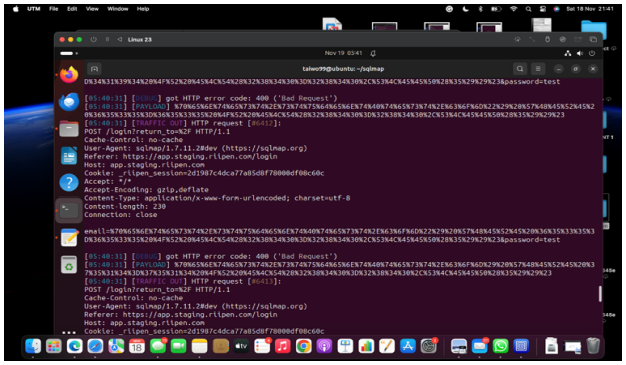


Fig 22: The image indicates an HTTP error 400 after injecting SQL payloads . This outcome may suggest that the website has security measures to detect and prevent such attacks, making it resistant to the SQL injection attempt. Further analysis and testing are essential to understand the website's robustness against security threats.

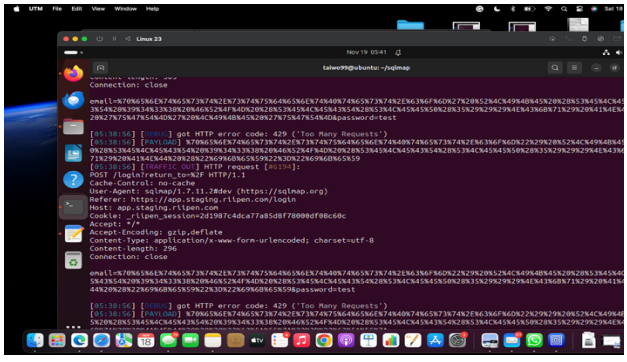


Fig 23: The image indicates an HTTP error 429, which typically signifies a "Too Many Requests" status.

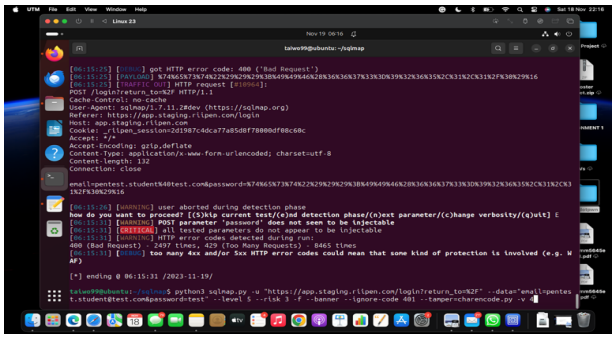


Fig 24: After injecting over 10,000 SQL injections, SQLmap observed that all parameters were marked non-injectable. This outcome suggests the presence of a firewall or other security measures that effectively blocked the SQL injection attempts.

Conclusion

Riipen Vulnerability assessment and penetration testing were performed to the best of all participants' knowledge through learning curves and already acquired knowledge. Based on the methodology of the MITRE framework, various attacks and vulnerabilities were found from the P1 to P3 categories. We have completed our assessment based on the OWASP Top10 vulnerabilities and have found that the website is pretty safe and has a relatively good security posture.

References

1. Cyber Technical Knowledge.“Web Application Penetration Testing Tutorial. ” YouTube.com.https://www.youtube.com/watch? v=gqMZx5H_Om0&t=7210s (accessed Sept. 14, 2023).

2. HackerSploit.“RedTeam Reconnaissance Techniques.”youtube.com.https://www.youtube.com/watch?v=BWaGnsRirtU (accessed Oct. 2, 2023).

3. C. M. J. Aileen and G. Bacudio, "AN OVERVIEW OF PENETRATION TESTING," *International Journal of Network Security & Its Applications (IJNSA)*.

4. V. S. KUMAR, "Ethical Hacking and Penetration Testing Strategies," *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, vol. 11

5. P. Ami and A. Hasan, "Seven Phrase Penetration Testing Model," *International Journal of Computer Applications*, vol. 59.

Appendix A: Project Proposal

Vulnerability Assessment and Penetration Testing of "Riipen" Web Application.

Name: Taiwo Mebude 1305695

Name: Felix Majemite 1275854

Name: Ankan Garg 1299560

PROBLEM STATEMENT

The primary aim of this project entails conducting a comprehensive penetration test (pentest) on a web application referred to as "riipen," with the overarching goal of identifying and delineating any existing vulnerabilities. Furthermore, the project will elucidate secure methodologies and techniques that can be implemented to fortify the security posture of the aforementioned web application.

I.REVIEW OF RELATED WORK

We shall be extensively using the MITRE framework as a source, as it's considered the Bible for pen testers. Along with the framework, we shall be utilizing various pentest reports and best practices available online to solidify our learnings.

II.PROJECT OBJECTIVE

The central objective revolves around integrating penetration testing practices within an educational framework. This encompasses acquiring knowledge, implementing practical exercises, and iteratively enhancing ethical hacking skills. The overarching goal is to systematically dissect attack patterns, identify vulnerabilities, and develop effective mitigation strategies through hands-on experience and learning, fostering a robust understanding of cybersecurity in an educational context.

III.DESCRIPTION AND METHODOLOGY OF RELATED WORKS

The primary aim of this project is to conduct a comprehensive penetration test on a web application system, encompassing five pivotal tasks: reconnaissance, enumeration, exploitation, post-exploitation/maintaining access, and covering tracks.

The specific objectives are as follows:

Reconnaissance: Thoroughly gather information and intelligence about the target web application to understand its structure, architecture, and potential vulnerabilities.

Enumeration: Systematically identify and enumerate any exposed services, applications, or weaknesses within the web application, aiming to catalog potential points of entry.

Exploitation: Actively assess and exploit identified vulnerabilities to ascertain the extent of potential security breaches, emphasizing ethical and controlled exploitation.

Post-Exploitation/Maintaining Access: After gaining initial access, establish mechanisms to maintain a persistent presence within the system, simulating how a malicious actor might persist within a compromised environment.

Covering Tracks: Implement countermeasures and techniques to obfuscate the penetration testing activities, mimicking the efforts an actual attacker might employ to evade detection.

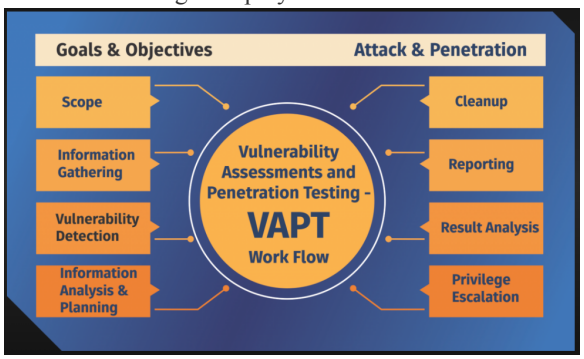


Fig: Capturing the flow of VAPT from end to end

IV. RESOURCES

In the context of an educational project, accomplishing the designated tasks necessitates a comprehensive approach involving the study and acquisition of specialized tools for identifying and exploiting vulnerabilities. Therefore, the essential resources for this project encompass online courses accessible via platforms like YouTube, various books, articles on medium.com, and journals which provide explicit guidance on diverse techniques and methodologies for conducting penetration testing on the targeted application.

V. CONTRIBUTION TO KNOWLEDGE

The objective is to conduct a comprehensive penetration test on the web application to identify

potential vulnerability points for exploitation. Additionally, it involves validating the security measures implemented within the web application,

assessing their effectiveness, and accrediting its assurance and reliability.

How will your project contribute to that knowledge?

·Our project may uncover vulnerabilities or security issues specific to the company's systems, contributing to a deeper understanding of potential threats.

Based on our assessments, we shall offer practical recommendations for improving the company's security posture, contributing to actionable insights

Are there any foreseeable limitations that might be addressed?

While every project has limitations, some common ones in VAPT projects are

Time Constraints: Limited timeframes for assessments may restrict the depth of testing. Consider documenting what couldn't be tested due to time limitations.

Scope Constraints: A narrow scope may miss potential attack vectors. Make explicit the scope of your assessment.

Resource Limitations: Limited access to certain tools or resources may affect the comprehensiveness of your assessment.

Ethical Constraints: Ethical considerations may prevent testing certain systems or techniques. Clearly state any ethical boundaries.

VI. TOOLS AND TECHNOLOGY

Kali Linux as an OS and various tools that come along with it
Wireshark for snipping traffic and capturing packets
Burp suite to intercept traffic, perform various requests and response
Metasploit framework

VII. PROJECT SCHEDULE AND MILESTONE DESCRIPTIONS

Week1	Pre-engagement with the Industry Learning and development
Week2	
Week3	
Week4	
Week5	
Week6	Reconnaissance
Week7	
Week8	Discovery/ Vulnerability analysis
Week9	
Week10	Exploitatation
Week11	
Week12	
Week13	
Week14	Final Report and recommendations
Week15	Closure

Fig: Capturing the weekly plan of action

VII. REFERENCES

[1] Johan Nilsson (2006) Vulnerability Scanners

[2] Ronald W. McCarty Jr. (2015, June 17) Looking for Vulnerabilities with OpenVAS and Greenbone

[3] Pawan Kesharwani, Sudhanshu Shekhar Pandey, Vishal Dixit, Lokendra Kumar Tiwari (2018, December). A Study on Penetration Testing Using Metasploit Framework

[4] youtube.com

[5] medium.com

Supervisor's comment(Prof. Sara Khanchi)

Thanks for preparing the proposal. There are some comments you need to consider:
Format: - The format is changing throughout the proposal. - The references are not written in the correct format.
Content: - The proposal is too generic; a more detailed explanation is required. - It seems that two weeks for exploitation might not be enough as it takes time to figure out how to exploit the vulnerability if there is one.

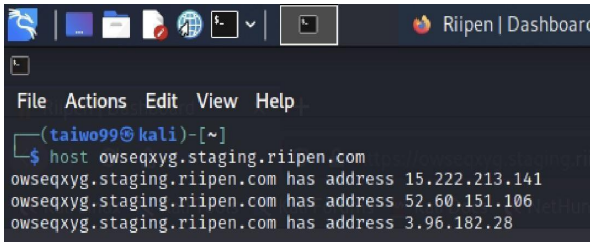
Appendix B: GitHub and supporting files.

This is just an opportunity to state-specific plans and documents. All commands used in this project will be documented and submitted with the report and also to the repository.
Repository:
https://github.com/FelixUfuoma/Final-project

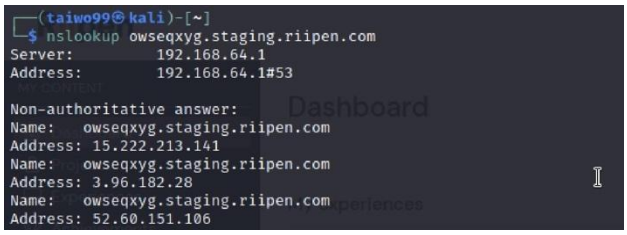
Appendix C: Images

Illustrations in this section are mentioned according to usage.

1. **Kali embedded**:

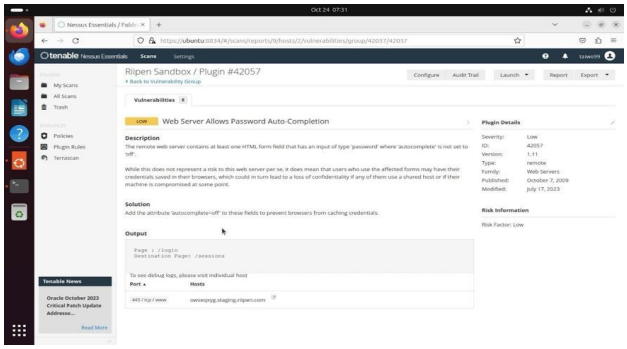


Host



Nslookup

.Automated tool:



Nessus

0. Number Images: exploitation Trails

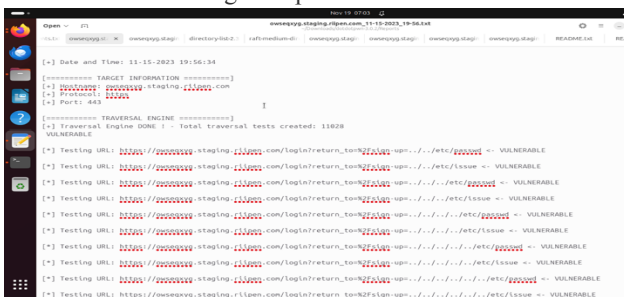


Fig 15.

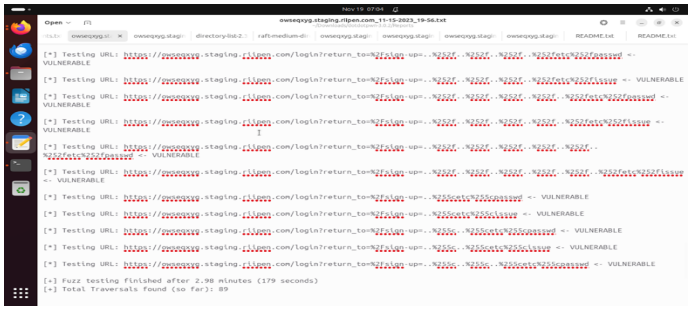


Fig 16.

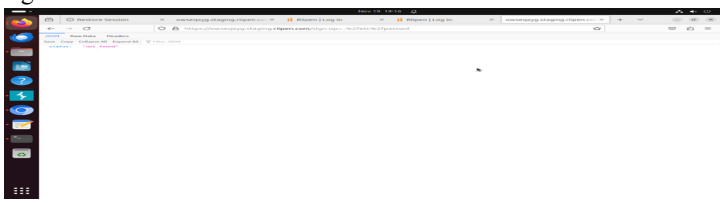


fig 18.

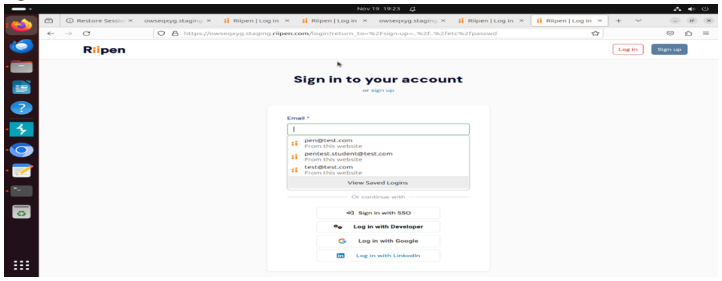


Fig 19.

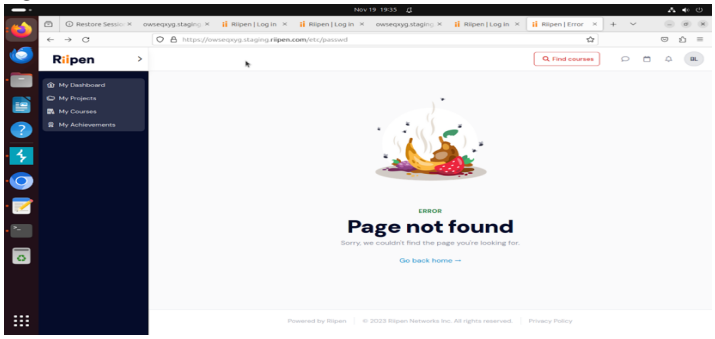


Fig 20.

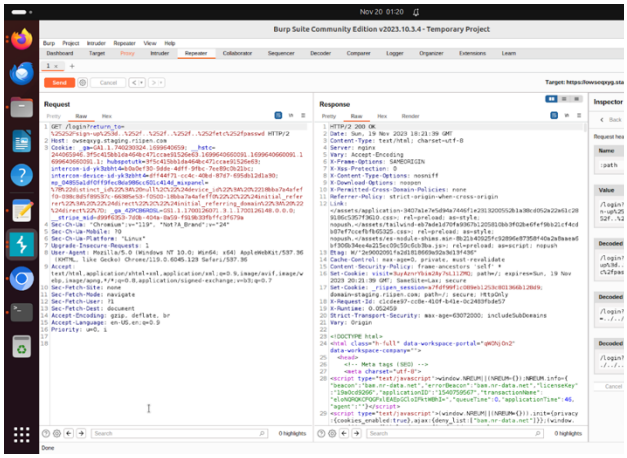


Fig 21.

Appendix C: index

- Strict-Transport-SecurityHTTP Strict Transport Security is an excellent feature to support your site. It strengthens the implementation of TLS by getting the User Agent to enforce HTTPS. Recommended value "Strict-Transport-Security: max-age=31536000; includeSubDomains".

- Content-Security-Policy: Content Security Policy is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets.

- X-Frame-Options: X-Frame-Options tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site, you can defend against attacks like clickjacking. Recommended value "X-Frame-Options: SAMEORIGIN"

- X-Content-Type-Options: X-Content-Type-Options stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content type. The only valid value for this header is "X-Content-Type-Options: nosniff".

- Referrer-Policy: Referrer Policy is a new header that allows a site to control how much information the browser includes with navigations away from a document and should be set by all sites.

- Permissions-Policy: Permissions Policy is a new header that allows a site to control which features and APIs can be used in the browser.

Appendix D: enumeration and findings

Load Balancer and Reverse proxy: We discovered that the application is hosted on three web servers and is supported by a load balancer configuration.

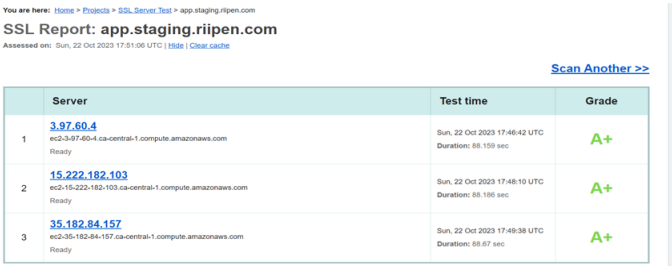


Fig 1. Capturing load-balancer details

Fig 2. Capturing reverse proxy

Result: This ensures that the website can handle massive requests and will be challenging to bring down via DOS attacks.

Tech Stack: This helps us understand what technology the application is based on, what OS it's using to host, and the various versions of the software being deployed at the application end.

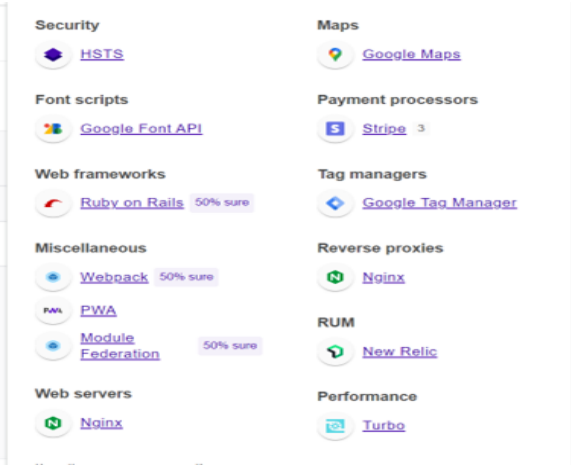


Fig 3. Technical stack of the application

Result: Based on the data gathered, software and OS versions are up-to-date. The web is safe from any software outdated vulnerability attack.

NON-HTTPS IP exposure: The disclosure of internal IP addresses on public-facing web applications is a security risk that can provide attackers with valuable information about an organization's internal network and systems. Exposed internal IPs enable attackers to gather details about the network architecture, which can then inform exploitation attempts and other targeted attacks. Internal systems and services not intended to be public should have their IP addresses hidden. Proper network segmentation, access controls, and avoiding IP address leaks in web content are key to mitigating this vulnerability.

Fig 4. Internal IP exposure

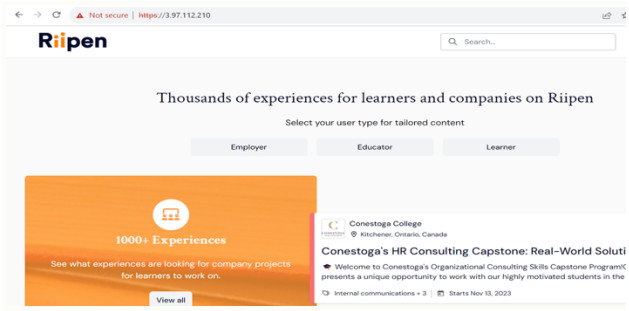


Fig 5. No HTTPS, only HTTP

Result: This is a huge vulnerability, as attackers can easily penetrate internal networks. We have informed the customer, and they are working on it now.

OWASP ZAP

This tool was created by the Open Worldwide Application Security Project (OWASP), an open-source tool for vulnerability assessment of web applications using URLs and IP addresses. This tool does an in-depth analysis using almost all concepts mentioned in this paper, such as spidering, forced browsing, directory transversal, and server fingerprinting, just to name a few. Proxy properties can sit between the browser and the user and are used to create HTTP requests and receive responses.

Owasp Zap Scan Result of Riipen

1. Initial login page scan:

Using passive scanning properties such as crawling

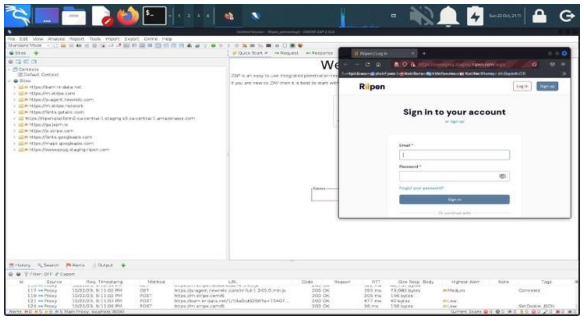


Fig 4. Showing login page scan info.

Potential alerts and flags detected in these scans:

Owasp Zap is an automated vulnerability assessment tool that is highly effective in exhausting all possible links using all forms of libraries inside. It also color codes alert flags according to the level of risk:

Red Alert: high level of risk found in a web application.

Orange Alert: medium level of risk found after scans.

Yellow Alert: low level of risk of risk

Blue Alert: without the three main levels, an information alert creates more clarity.

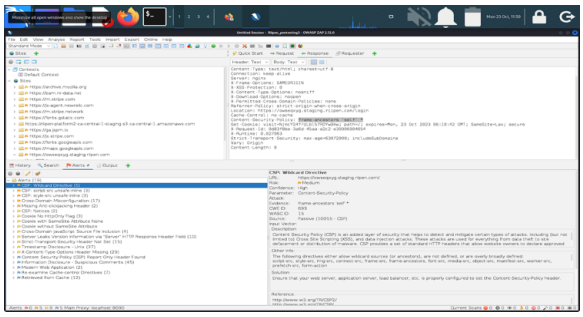


Fig 5. Showing Alert including flags and all the attributes.

Ultimately, Owasp Zap scanned every inch of the Riipen website for tangible vulnerabilities but found none. There is an HTML report documenting the record of the scans done by Owasp Zap as a pentesting tool. This report will be submitted and posted in the repository.