ISCG7431 – CAPSTONE PROJECT
Semester 2, 2024

Comparative Analysis of Penetration Testing Tools in Assessing Web Application Vulnerabilities

Prepared For:

Course Coordinator: **Dr Masoud Shakiba**
Sponsor: **Andrew David**

Prepared By: TARUHO

Tanveer Ahmed Osman ID# 1421320
Jun Russell Ragasa ID# 1555646
Hope Pose-Anesone ID# 1541361

Date of Submission: 10 November 2024

Table of Contents

List of Figures

List of Tables

1.0 Document Control

1.1 Version History

Table 1 contains the list of version history.

Title	Comparative Analysis of Penetration Testing Tools in Assessing Web Application Vulnerabilities		
Created By	Tanveer, Russell, Hope (Taruho)		
Date Created	20/10/2024		
For	Final Capstone Report Document		
Version Number	**Modified By**	**Modification Made**	**Date Modified**
1.0.0	Tanveer	Create a format for the report	20/10/2024
1.0.1	Tanveer	Start compiling executive part	21/10/2024
1.0.2	Russell	Writing Introduction	21/10/2024
1.0.3	Hope	Problem Statement	22/10/2024
1.0.4	Russell	Objectives	23/10/2024
1.0.5	Tanveer	Scope	25/10/2024
1.0.6	Hope	Literature Review	24/10/2024
1.0.7	Hope	Client's Problem and Requirements	28/10/2024
1.0.8	Hope	Project Management Methodology	27/10/2024
1.0.9	Tanveer	Project Design	25/10/2024
1.1.0	All	Project Implementation	29/10/2024
1.1.1	All	Testing	30/10/2024
1.1.2	All	Results & Evaluation	31/10/2024
1.1.3	Hope	Impacts	1/11/2024
1.1.4	Russell	Change/Risk Management	27/10/2024
1.1.5	Tanveer	Conclusion	3/11/2024
1.1.6	Tanveer	Reference	3/11/2024
1.1.7	Tanveer	Formatting and Editing	4/11/2024
1.1.8	Tanveer & Russell	Final Checking	5/11/2024

Table 1 - Version History

1.2 Distribution List

Table 2 contains the list of distributions.

Project Role	Name
Capstone Co-Ordinator	Masoud Shakida
Project Sponsor/ Supervisor	Andrew David
Client/Stakeholders	Unitec| Masoud | Andrew
Technical Advisor	Justin Ng | Senior Penetration Tester
Tanveer Ahmed Osman	Project Manager
Jun Russell Ragasa	Penetration Tester
Hope Pose-Anesone	Penetration Tester

Table 2 - Distribution List

1.3 Glossary

Table 3 contains the list of glossary.

Abbreviation	Definition
PT	Penetration Testing
OWASP ZAP	Zed Attack Proxy tool
SQL Injection	Code injection attack consists of the insertion of SQL query.
DVWA	Damn Vulnerable Web Application
SWOT	Strengths, Weaknesses, Opportunities, and Threats
SQL Map	Pen Testing tool detecting and exploiting SQL Injection
IT	Information Technology
GUI	Graphical User Interfaces
SME	Small Middle Enterprises
GDP	Gross Domestic Product
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
Fuzzer	An automated software technique
IP address	Internet Protocol Address
WFUZZ	Brute Force Pen Testing Tool
SMB	Server Message Block
SQL Map	Tool used for detecting and exploiting SQL injection
Hydra	Brute Force Pen Testing Tool
Burp Suite	Application Security Testing Tool Software
Nikto	Automatic Vulnerability Scanner
NCSC	National Cybersecurity Center
Portswigger	Company provides Web Security Software
Kali Linux	Open-source Linux software design for Pen Testing
Apache	A free open-source web server
ms	Millisecond
RTT	Real travel time
VMware workstation	A desktop hypervisor software used to run virtual machine
MacOS Sequina	Apples MacOS operating system
Hybrid Methodology	Blended project management combines of Waterfall and Agile
Wifi	Wireless network technology
Router	Physical or virtual appliance used in networking system
Kanban Jira	Framework for Agile hosted in Jira
PlatformPi	Ecommerce website
Get	Method of retrieving data
MySQL	Database Management System
CPU	Central Processing Unit

Table 3 - Glossary

1.4 Contribution List

This project is a collaborative effort, and we would like to thank everyone who contributed to this project generously.

First and foremost, we would like to thank our project coordinator, Masoud Shakiba, for providing us with the necessary support throughout the project and guiding us in the right direction.

We would also thank PortSwigger Company for providing us with the paid version of the Burp Suite Professional to test the software during this capstone project. This contribution immensely helped us to find more in-depth results.

Our extended gratitude goes to Andrew David, who is our project sponsor and supervisor. We would really like to thank him for being so supportive. His guidance and mentorship were invaluable, and it would not have been possible for us without his guidance to complete our project.

We also thank Justin Ng, our technical advisor, for providing technical support and advice. His support for setting up the DVWA environment was helpful in conducting our test effectively.

Finally, to those who supported our project, directly and indirectly, we would like to convey our deepest thanks for their contribution.

Project Confidentiality

Distribution List

This document will remain confidential until published by the sponsor and/or team. However, this document will be accessed by the Capstone Project Coordinator, Unitec internal assessors including Judging Panel members, and Unitec's internal and external academic moderators for the purposes for ensuring that students have completed the requirements necessary to achieve the learning outcomes of the ISCG7431 Capstone Project course and any other academic requirements.

Name	Role	Email Address
Andrew David	Project Sponsor	adavid@unitec.ac.nz
Andrew David	Project Supervisor	adavid@unitec.ac.nz
Dr Masoud Shakiba Jesse Nortey	Capstone Coordinator	mshakiba@unitec.ac.nz
Student 1	Development/Project Team	osmant01@myunitec.ac.nz
Student 2		ragasj01@myunitec.ac.nz
Student 3		taimah01@myunitec.ac.nz
All assessors and panel members	Assessor	
All moderators (internal and external)	Moderator	

Declaration and Individual Contribution

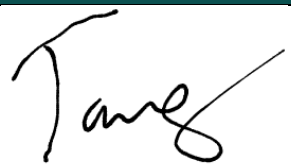
Assignment Details	
Course	ISCG 7431 Capstone Project
Due date and time	10 November 2024 – 23:59 NZ time zone
Delivery	Online Moodle Submission
Semester	2 – 2024
Course Coordinator	Dr Masoud Shakiba
Student Details	
Name / Student ID	**Contribution to Project and Final Report**
Tanveer/1421320	*Planning* *Tool Selection* *SQL Map Tool* *Documentation*
Russell/1555646	Tool Selection Wfuzz tool Burp Suite Professional Burp Suite Community Documentation

Hope/1541361	Problem Statement Requirement Analysis Literature Review Project Analysis Impact & Tools - Hydra, Nikto, ZAP

Project Declaration

Collectively as a group, we confirm that:
This is an original assessment and is entirely our own work.
Where I have used ideas, tables, diagrams or other media of other authors, I have acknowledged the source in every situation
This assignment has not previously been submitted as assessed work for any academic course.

Signatures **Date Signed: 10/11/2024**



Tanveer

Russell

Hope

Sponsor Signature	**Supervisor Signature**
 *Andrew David – 10/11/2024*	 *Andrew David – 10/11/2024*

2.0 Executive Summary

This research project compares various penetration testing tools for evaluating web application vulnerabilities in a secure environment. As cyberattacks grow more complex and sophisticated, selecting the best tools becomes the number one priority for accurately identifying, assessing, and mitigating security risks in web applications in the cyber world. This study focuses on the effectiveness of popular tools like Burp Suite, Wfuzz, OWASP ZAP, Hydra, and SQL Map in identifying common vulnerabilities such as SQL Injection and authentication errors with different security levels.

Each tool was assessed based on criteria including accuracy, detection capabilities, usability, integration options, and reporting features. Among the selected tools, Burp Suite and OWASP ZAP demonstrated high accuracy and adaptability, which are particularly valuable for manual testing and advanced customisation in a secure environment.

By outlining the strengths and limitations of each tool, this research guides testing professionals and businesses in making informed decisions based on project requirements, budget, and security goals. The findings emphasise the importance of a multi-tool approach to ensure comprehensive and effective vulnerability assessments, ultimately enhancing the security posture of web applications.

3.0 Introduction

In today's digital world, we can see that web applications are one of the main services and interfaces of many organisations in interacting with their clients. Many hackers use this opportunity to exploit the system and get delicate data from organisations, which they use to access the system. This is why securing web applications is very important. In this project, we aim to address these issues by evaluating the most common penetration testing tools to check the effectiveness of the chosen tools in identifying vulnerabilities in web applications.

Our project's main goal is to perform a comparative analysis of the most popular tools: the Burp Suite Professional and Community version, OWASP ZAP, Wfuzz , SQL map, and Hydra. We have chosen these tools because they are used mostly by people doing penetration testing and just people trying to test web applications. These tools have unique strengths in identifying vulnerabilities and issues in web applications. The assessment will focus on some key metrics, detection accuracy, performance speed, and whether they have a feature to generate reports. We have set up a real-world-like simulated testing environment to determine how well these tools perform. We tested each tool on DVWA (Damn Vulnerable Web Application) multiple times and recorded each result, and tested them in low, medium, and high levels of security.

By conducting an in-depth analysis, our project will provide insights about each tool and recommend the most effective and reliable tools for cybersecurity professionals seeking to secure web applications. Furthermore, the study will highlight which tools are the most balanced regarding speed, accuracy, and reliability, which would guide securing web interfaces for any security threats.

Ultimately, this project will help cybersecurity professionals and organisations by presenting a thoroughly researched comparison of the chosen penetration testing tools. This resource will contribute to providing information to help web applications be more secure through all the insights shared in this project for practitioners and the selection of tools recommended for web application security.

4.0 Problem Statement

4.1 Problem Scenario – Tahuro e-commerce business

Identifying vulnerabilities in web applications is crucial, specifically for a small startup business proliferating and concerned about its online platform's security. Organisations often face challenges with selecting practical tools in the context of penetration testing, leading to compatibility issues and operational inefficiencies.

4.2 Current State Scenario

Tool Selection: The IT manager is responsible for various penetration testing tools but finds the information scattered and inconsistent. Many tools need comprehensive documentation, and there are no clear comparisons available. After spending several hours sifting through user reviews and outdated articles, the IT manager selects a tool based on positive user comments despite needing to understand its features or limitations fully.

Execution: The penetration test is conducted, but the results show a mix of critical and minor vulnerabilities. The IT manager notices several alerts that indicate potential issues, but without proper documentation or a clear understanding of the tool's accuracy, there's confusion about which vulnerabilities are legitimate.

Outcome: Relying on the tool's findings, the IT team prioritises remediation efforts based on what they assume to be the most critical vulnerabilities. However, some flagged vulnerabilities are false positives, while significant issues are overlooked. This oversight leaves the web application exposed to potential exploits. Also, current manual testing methods are time-consuming and inconsistent.

4.3 Addressing the problem

To understand and address the security challenges facing Taruho's IT Team, the problem is analysed through SWOT analysis, which helps to visualise the current state, identify gaps, and establish a structured approach for improving the penetration testing process [1].

Table 4 shows the identified challenges in our team using SWOT analysis.

Strengths	**Weaknesses**
Growing e-commerce platform	Inconsistent penetration testing methods
Access to various testing tools	Reliance on unverified information sources
Motivated IT, team	Limited tool documentation and guidance
Opportunities	**Threats**
Improved tool selection framework	Rising cyber threats targeting web application
Streamlined testing and remediation	False positives leading to resource waste
Enhanced platform security	Undetected vulnerabilities

Table 4 - SWOT Analysis

4.4 Desired State Scenario and Solution

With proper updated research review and reliable resources, the IT team could confidently conduct and analyse the penetration test, ultimately enhancing the security of the web application platform. Access to evaluation frameworks for penetration testing tools and could make an informed decision, ensuring the selected tool is well suited for their specific needs. Comprehensive documentation and recommendations would clarify how to interpret the test results, minimising the chances of false positives and ensuring that all critical vulnerabilities are addressed.

A defined comparison of performance metrics, test accuracy, and evaluation of tools is necessary to ensure efficiency and security gaps. Establishing performance metrics will allow the IT team to systematically evaluate the effectiveness of penetration testing tools, streamline their processes, and enhance the platform's overall security posture.

5.0 Objectives

5.1 Objectives for Penetration Testing

Objectives for Comparative Analysis of Web Application Tools

The project assesses the efficiency of selected penetration testing tools, including Hydra, Burp suite professional and community, OWASP ZAP, Wfuzz, and SQL Map, in identifying web application vulnerabilities, particularly brute force technique and SQL injection. The tools will be evaluated with performance metrics such as accuracy, ease of use, and report-generated features. Using the structured framework for guidance in comparison. The result will then be analysed, and recommendations will be provided for cybersecurity professionals and organisations to enhance web application security.

5.2 SMART Objective

Specific:

Produce a well-researched report comparing multiple penetration testing tools, including Hydra, Burp suite professional and community, OWASP ZAP, and SQL Map. Detailed information on each tool's capabilities will be provided to recommend which tool has the best features in detecting web application flaws. The ultimate goal is to guide cybersecurity professionals and organisations in securing web applications by selecting the most effective tools.

Measurable:

The tools are evaluated on:

- The accuracy of each tool in detecting web application vulnerabilities
- The ease of use of each tool
- Reporting capabilities
- The speed performance of each tool

The structured framework will guide the comparison, and each tool will check the success rate of identifying security flaws and ensure that the result is consistent in the test environment.

Achievable:

We can achieve this objective by choosing highly used penetration testing tools and known methodologies. We used simulated web application environments that we will use to test web application vulnerability. We will use DVWA (Damn Vulnerable Web Application), which is hosted by our stakeholder Justin Ng (Spark Senior Penetration tester), and the Platform PI application. We ensure a controlled and practical approach while testing. Having the right people and available resources, the project's scope is well within reach.

Realistic:

The analysis will focus on several penetration testing tools we can analyse for three months of the project, ensuring a thorough comparison and research and that we achieve the project's scope. Recommendations will be provided after testing real-world web security scenarios. Making the testing more realistic and beneficial to stakeholders.

Time-Bound:

The project will be completed and presented in 15 weeks. The key phases include tool testing, which is done in 4-8 weeks. Framework development, performance criteria, and metrics are created during Weeks 9-11, and the final report will be on weeks 12-14; the final presentation is scheduled for November 15, 2024.

5.3 Stakeholder's Understanding

This project aims to provide a comprehensive analysis of the efficacy of selected penetration testing tools to stakeholders or anyone seeking some effective tools that they might use to assess the vulnerabilities of web applications. This report focuses on the most used penetration testing tools: Burp Suite versions, OWASP ZAP, SQLMap, and Hydra. By carefully evaluating these tools with testing and performance criteria created by the team, we aim to provide insights into each tool's strengths and limitations and how they are suitable in any scenario. These results can help stakeholders select appropriate tools for testing web applications and secure them by providing mitigations. A structured framework and detailed document creation will guide stakeholders in understanding each tool, and it is easier to adopt recommendations for enhancing cybersecurity in their organisation.

6.0 Scope

6.1 In Scope

- **Analysis of at least five widely used penetration testing tools for web applications.**

  This project will include five popular penetration testing tools to assess their effectiveness in terms of identifying web vulnerabilities. The capabilities, efficiency and limitations of each tool will be assessed to determine its functionality.

- **The selected tools will be used in a controlled web application environment (DVWA) with known vulnerabilities.**

  A secure and isolated testing environment is Damn Vulnerable Web Application (DVWA). The testing will be held in this controlled environment to allow consistent and reproducible testing conditions to accurately compare the testing tools.

- **The tools used in the project are open-sourced and possible commercial penetration testing tools if the company provides them in the analysis.**

  Both open-source and paid commercial tools will be used to analyse different functionalities.

- **Develop a test plan for comparing the tool's performance.**

  A structured testing plan will be outlined with specific testing criteria, including detection accuracy, ease of use, and other parameters.

- **Data collection from testing and proper documentation.**

  Data collected from each tool will be documented to identify vulnerabilities and possible overall efficiency.

- **Preparation of presentation and provide recommendations.**

  The final summarization of the findings is where each tool's performance is evaluated. The project will include recommendations on which tools are best for specific types of web testing and their strengths.

6.2 Out of Scope

- **Testing the penetration tools in a Live Production Environment.**

  The testing will not be implemented in real-world environments as some legal issues are involved.

- **Creation or Modification of new tools.**

  No creation or modification of the tools will be executed.

- **Mitigation of identified Vulnerabilities.**

  No security patches or configurations will be altered while testing.

- **Creation of Web application for testing.**

  No new web application will be created for testing purposes.

- **Long-term monitoring and maintenance of web applications.**

  There will be no long-term monitoring and vulnerability check on the web application.

- **Assessing the overall security of the system.**

  Only the web application vulnerability will be assessed.

- **Cost-benefit analysis of the tools.**

  This penetration testing phase will not include a cost-benefit analysis of different paid tools.

7.0 Literature Review

Taruho's literature review illustrates primary and secondary sources to enhance understanding of the crucial tools in assessing vulnerabilities and the necessity to improvise. Penetration testing is a critical set of tools for checking system security vulnerabilities. As web-based services and applications grow, significant security concerns have also developed.

7.1 Evaluation of literature from primary and secondary sources

Primary sources emphasise the urgent need for continuous assessment and adaptation as the ongoing evolution of cyber threats is advanced. A study by Albahar [2] accounts for empirical research on tools' performance in identifying vulnerabilities. [3] Research on the unique challenges that arise within virtual environments. Study by Khan/bang [4] emphasises the importance of standardised reporting. In secondary sources, all papers above combine findings from primary sources to provide a broader perspective on the capabilities and limitations of different pen-testing tools.

7.2 Comparison and Contrasting of Different Sources

Literature reviews from the sources above offer distinct but complementary insights into PT; Kim and Lee's work specialises in a virtualised context [3]. In contrast, research studies aid in tool selection for specific needs. Moreover, the MDPI study presents an overall performance benchmark, making it ideal for general web security assessments.

7.3 Connection of Literature to the Project

The literature review is related and understood to be relevant to the Taruho project in recognition of the need for a comprehensive evaluation process that includes multiple tools or methodologies to ensure proficiency.

Ultimately, integrating the findings and recommendations enhances its effectiveness, ensuring that chosen penetration testing strategies are well-suited to the complexities of the web applications being assessed and provide reliable insights for improving overall security [2].

7.4 State of the Art: Evaluation of Current Technologies

1. Testing Environment - Modern Virtualization enables maximising resource utilisation. However, it is still prone to complications that testers must be aware of.
2. Tools Platform – Commercial tools like Burp Suite simulate real-world attacks, but the free community provides an excellent alternative for low-budget organisations.
3. Tool Integration – Modern PT tools are increasingly incorporated to stay updated. However, allowing proper evaluation documentation and software updates can optimise testing efforts based on their specific environment.
4. Evaluation Criteria – The diversity of Penetration Test Technology can cater to the needs of SMEs; however, several criteria should be considered

- Ease of use
- Integration
- Scalability
- Cost-effectiveness

7.5 Comparison and Contrast of Current Technologies

The state-of-the-art analysis reveals a distinct contrast in the current PT technologies compared to the Taruho project approach, evaluating the efficacy and comparative analysis of tools, such as OWASP ZAP, reduced the learning curve and made them accessible even to less experienced testers.

7.6 Connection to our Project

Similarly, understanding the trade-offs between free tools like ZAP and commercial options like Burp Suite Pro helps align the tool selection with the project's budget constraints to individuals.

7.7 Summary of literature review

The literature review and state-of-the-art analysis underscore the urgency of addressing the comparative analysis of penetration testing tools in web applications, which is paramount. The value of virtualised environments and precise tool selection enhances testing accuracy and creates a safe, scalable framework for cybersecurity assessment and training. Please refer to the “Literature Review” appendix for a literature overview.

8.0 Problem and Requirements Analysis

Breakdown of Project Objectives into Requirements

The initial project analysis requirement follows these phases.

- Planning, Research, and Tool Selection
- Design Testing and Environment Setup
- Implementation, Testing and Document
- Analysis and Reporting
- Deployment and Presentation

To achieve the above phases, objectives were divided into detailed requirements below:

1. Select practical penetration testing tools.
    - Conduct a literature review and comparative analysis to evaluate tool suitability.
    - Establish selection criteria, such as detection accuracy, latest updates, performance speed, and feature completeness.
2. Create a safe environment for vulnerability testing
    - Set up a controlled web application (e.g., DVWA) with intentional security flaws.
    - Isolate the environment from production to avoid interference with the live system.
3. Ensure testing accuracy and minimise false positives
    - Track and analyse metrics like false positive rate, vulnerability detection accuracy, and execution speed for each tool.
    - Provide clear documentation on tool usage and interpretation of test results.
4. Provide actionable and recommendation insights and recommendations.
    - Document a detailed report of findings, including each tool's strengths, weaknesses, and suitability of each tool
    - Compile a set of recommendations to address critical based on priority
5. Share findings with stakeholders effectively.
    - Prepare a presentation summarising the key findings, tools, comparison insights, and recommended security improvements.

Aligns with the initial objectives and addresses specific challenges:

- The Taruho IT Team will have a structured selection process, minimising reliance on outdated or inconsistent information
- Setup of a controlled testing environment allows for repeatable vulnerability analysis.

- Performance metrics ensure tool accuracy and reliability, reducing false positives and undetected vulnerabilities
- Comprehensive documentation and a final presentation ensure stakeholders are well informed, guiding future security strategies effectively.

Software, Hardware and Tools Used

Software: Testing Virtual Environment

- Operating Systems: Windows 10/11 64-Bit,
- Kali Linux Version 2024
- VMWare Workstation

Software: Pen Testing Tool (some tools already installed within Kali)

- Burp Suite Professional
- Burp Suite Community
- Hydra
- Wfuzz
- Nmap
- Zap
- SQL Map

Testing Web Application:

- DVWA (enable to whitelist and host online privately on Cloud Server for Testing purposes)
- Web Server Installation: Apache 2.1

Hardware:

- HP Laptop - Intel core i5
- Apple Laptop – MacOS Sequoia
- HP Laptop – Elitebook
- Router
- Wi-Fi

Rationale for selected technology

The selected testing tools and software were based on their ability to:

- **Enhance Security Assessment:** Tools like Burp Suite and ZAP provide in-depth testing capabilities using the Brute Force attack, allowing the team to assess web application security comprehensively.

- **Support Automation and Efficiency:** Tools like SQL map and Hydra streamline repetitive tasks, allowing the team to focus on analysis and remediation.
- **Provide Isolation and Safety:** Virtualization ensures testing activities are isolated from live systems, minimising risks to production environments and reliable testing scenarios.
- **Facilitate Documentation and Reporting:** Tools that offer structured output, such as Burp Suite, make documenting findings easier and provide transparent reporting to stakeholders.

9.0 Project Management Methodology

Hybrid Methodology

Taruho project employs a hybrid project management approach as we utilise Gantt Charts to monitor and ensure each phase is complete.

Also, we prioritise SecSDLC as our foundational framework in which we also utilise OWASP Pen Testing workflow to guide our PT testing activities that emphasise on phases below [5].

- Information Gathering

  Purpose: This stage sets the penetration test's foundation, which helps identify potential entry points.
- Review Possible Test Method

  Purpose: This is the critical stage where different testing tools can be compared based on their ability to detect various vulnerabilities.
- Have the Attack Methods been Analyzed and Investigated?

  Purpose: Executing an attack, the Tester evaluates whether potential methods have been explored.
- Did the Attack succeed?

  Purpose: To confirm that the vulnerability is real and exploitable rather than a false positive.
- Evaluate the Tool /Risk Assessment of the Attack

  This stage can be used to compare how different tools are prioritised, which is essential for practical evaluation.

With the Agile structure, we utilised the Kanban Jira Platform as an ideal methodology required in our implementation testing phase, with continuous testing, research, and evaluation as essential when dealing with evolving threats and vulnerabilities. Please see the appendix for the Project Gantt Chart and Jira board.

10.0 Project Design

10.1 Structure Design

The structural design of this penetration testing project includes a Damn Vulnerable Web Application (DVWA) environment setup is a safe mode and critical analysis of the web application vulnerabilities. Within a controlled environment, DVWA allows penetration testers to test weaknesses in the system with different levels of security. Different penetration testing tools such as Hydra, OWASP ZAP, Wfuzz, Burp Suite and SQL Map can be implemented to test the system's vulnerability through SQL Injection and brute force attacks.

Each tool has a different approach and reacts to this controlled and secure structure. The penetration tester controls each tool's functionality to initiate the attack based on the situation. The attack is done on the Web application – DVWA. The appropriate response comes from the vulnerable website, where the security levels are determined. Based on the response, it is analysed, weakness is discovered, and the loophole is identified. Finally, the report and documentation are done with the help of a structured design.

Figure 1 shows how our testing is done using a systematic approach.

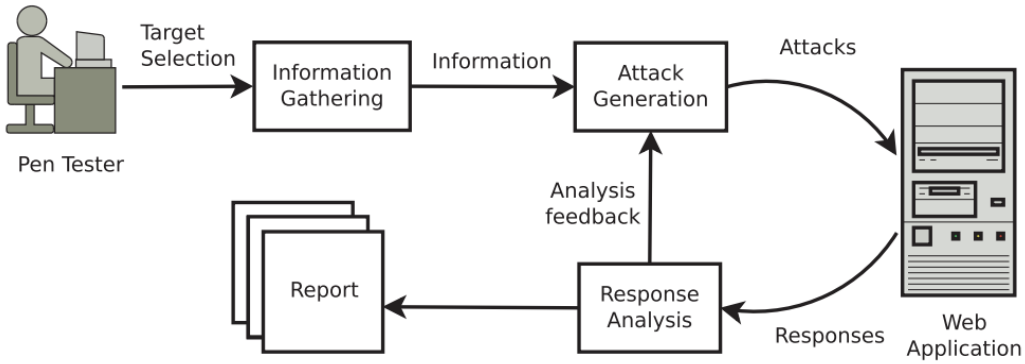


Figure 1 - Process of doing the Penetration testing. [6]

This structured approach provides detailed insight into the weaknesses and strengths of each tool, along with the enhancement of the testing. This project will help security professionals to choose the right tool to assess the web application in a secure environment.

10.2 Conduct Initial Research on The Penetration Testing Framework

To understand the development of Penetration Tests over the last few years, various researchers in the field examine different aspects of web application testing where most projects implement the OWASP Framework guidelines [7].

10.3 Review Academic and Industry Reports

We have conducted a literature review based on the work done by various researchers in Web Application Testing tools. Our Research Methodology for comparing and evaluating the selected Web application pen testing tools was developed following the criteria below.

10.4 Selection of Penetration Testing Tools

Selecting and implementing the right penetration testing tools is crucial to carry out the expected result when it comes to testing web applications. When selecting the appropriate tool, there must be a balance between usability and functionality, which will meet project requirements. Supporting a range of applications and finding specific vulnerabilities within security levels are also considered when selecting tools. For example, as there are automation features, SQL Map can be used for SQL injection attacks on the database.

Security experts may customise attacks and conduct in-depth analysis of HTTP/S traffic with the help of Burp Suite and OWASP ZAP, which are well-known for thorough web application scanning and include proxy-based testing, interception, and custom payload choices [3].

Furthermore, some penetration testing tools come in both paid and open-source versions. Open-source tools such as ZAP, Nikto, Hydra, Wfuzz, and SQL Map have different functionalities, and paid tools such as Burp Suite have some more features compared to the community version. The paid version includes some automation and improved reporting functionality.

10.5 Setting Up Test Environments:

Effective and efficient penetration testing projects require setting up a secure test environment, which enables testers to recreate actual attacks on online applications without endangering live data. The Damn Vulnerable Web Application (DVWA) is one of the controlled environments where testers may safely practice exploiting deliberate flaws. To prevent any security threats, installing DVWA usually entails setting it up on a local or virtual server. This setup is separated from any live production environment. This secure and stable environment provides insight into how vulnerabilities react to different attack tactics and may be adjusted by the tester from low to high-security levels [8].

Setting the environment along with the tools is a crucial and important task. For example, intercepting HTTP requests through Burp Suite and setting up login credentials for testing with Hydra is a step-by-

step process. In order to work properly, these tools need to be set in such a way that on different security levels, they can perform at their best effort. A properly configured testing environment helps the security team to identify weaknesses in the system and mitigate future exploitation [9].

Figure 2 shows the team's design and implementation of the project, having 3 different virtual machines and tools used to test web applications.

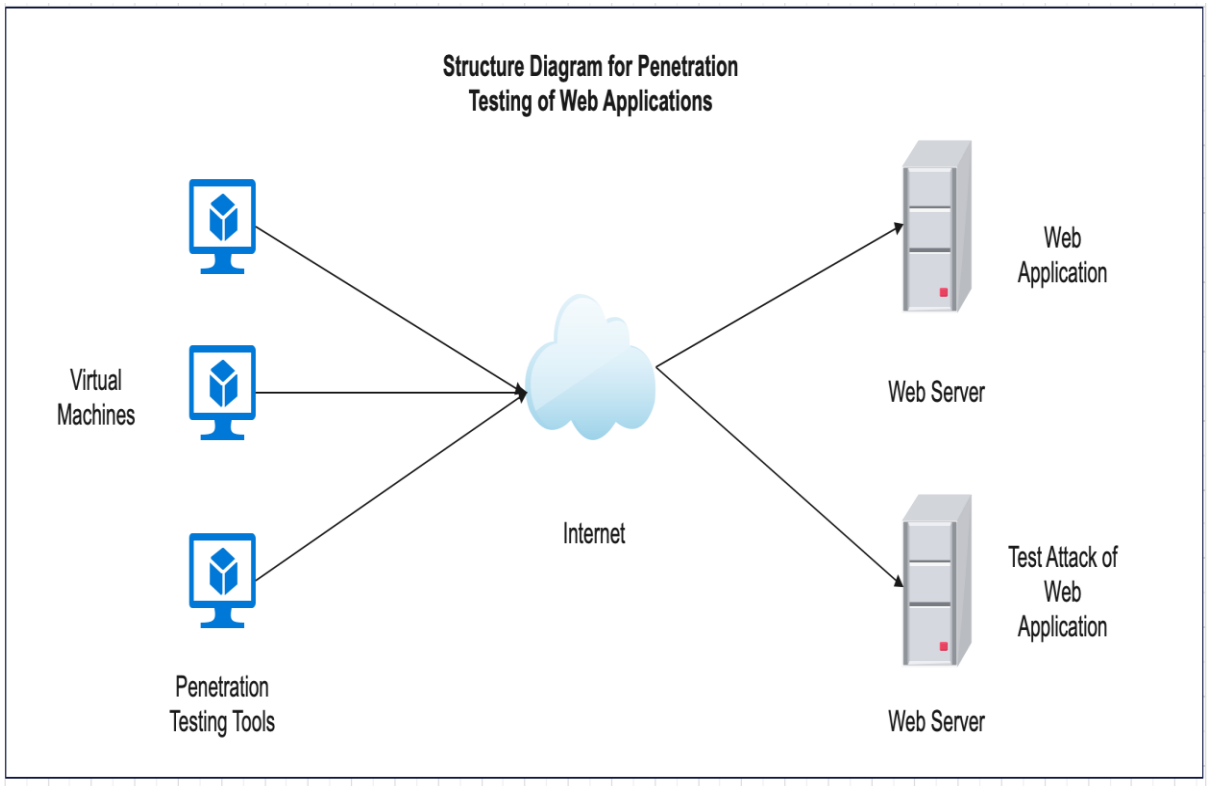


Figure 2 - Process of doing the Penetration testing [6]

10.6 Test Implementations

In penetration testing projects, test implementation entails prearranged steps to find, exploit, and record security flaws in an online site with a secured and controlled environment. The first step in this phase is to choose the target regions in a controlled setting, such as a Damn Vulnerable Web Application (DVWA), which is designed to highlight typical security vulnerabilities at different levels. Tools are tested with commands relevant to each attack for every vulnerability found. For instance, Burp Suite or OWASP ZAP allow both automatic and manual scanning to find hidden vulnerabilities and analyse HTTP/S requests. At the same time, SQL Map may automate SQL injection testing by focusing on fields with known vulnerabilities [10].

Penetration testers investigate how programs or applications react to harmful inputs during implementation using various strategies, including input manipulation. Every test has thorough documentation that includes information on successful exploits, unsuccessful efforts, and unexpected system reactions. Testers mostly use a variety of factors and variables to validate results from various approaches, assuring comprehensive and reliable outcomes for each test. Collecting screenshots and other evidence at every stage thoroughly records the application's security posture [11].

10.7 Selection of evaluation criteria

Assessment criteria are one of the important factors to choose when it comes to evaluating the efficiency of the penetration testing tools. Based on the testing, the assessment criteria are categorised into different factors. The assessment criteria for each tool are:

Tool Type: It defines open-source or paid version

Version No: It defines the version of the tool.

Latest Update: It defines the date of the newest update

Cost Effective: It defines if the tool is a paid version and how much it costs.

Integration: It defines whether this tool can be implemented with other web application penetration testing tools.

Ease of Use: Comprehensive and needs tutorials or a manual to use the application.

Accuracy: This gauges how consistently a tool detects real vulnerabilities without producing false positives.

Performance/speed: It defines how quickly it can perform its operation.

Usability: It defines how the tool is used in terms of installation.

Scalability: It refers to the adaptability of the tool in different operating systems.

Reporting: It defines the reporting format and understandability of the outcome.

Out of these criteria, accuracy, integration, performance, and scalability are most important. In addition to these, usability is also important because when it comes to complex testing, usability becomes the prime factor. Performance criteria could be very effective in terms of identifying vulnerabilities in a customised environment, which explains how it can adjust according to the needs [12].

Furthermore, the tool's integration capacity evaluates if it can complement other systems or tools to improve testing workflows for different test environments. Reporting features are also crucial since they enable penetration testers to record their results thoroughly to identify actual output. Testers may choose to use the finest technologies to do precise, thorough, and effective security assessments

that are adapted to particular project requirements and circumstances for any project by using these evaluation criteria [6].

We ensured the most recent versions of each tool were used for evaluation, verifying updates until the project deadline. The base model of our performance criteria is below. Figure 3 shows the design of our performance criteria used to test each tool used in the project.

Test Tool Name	[Tool Name]	**Test Case Version:**	**[BS_01]**
Date	[Date of Testing]	**Test Input:**	
Tester Name:		**Test Environment:**	[Virtual Kali OS]
Criteria	**Metrics**	**Test Comments**	
Tool Type:	[Community, Paid, Free Trial]	[Provide a short explanation]	[Provide screenshots or explanation]
Version No:	[Version used in Testing]		
Latest Update:	[New update available]	[How often does the tool updates]	
Accuracy:	[Low/Moderate/High]	[Provide a short explanation of the tool's vulnerability detection accuracy]	
Update	[Frequent/Seldom/out of Date]	[Describe the frequency of updates]	
Ease of Use:	[Low/Moderate/High]	[How user-friendly is the tool? Consider interface design Learning curve, support]	
Cost Effective:	[Paid/Open Source]	[Specify whether the tool is free or open source and compare the cost with value]	
Usability:	[Low/Moderate/High]	[Rate the overall user experience in terms of ease of navigation and functionality]	
Performance/Speed:	[Low/Moderate/High]	[How well does the tool when performing scans on web applications?]	
Scalability:	[Low/Moderate/High]	[How well does the tool handle large-scale web applications and enterprise environments?]	
Reporting:	[Low/Moderate/High]	[Evaluate the quality customisation of reports generated]	

Figure 3 - Our Performance Criteria

11.0 Project Implementation

Figure 4 shows detailed information on the settings used in our testing. Hardware and software used and tools to test web applications.

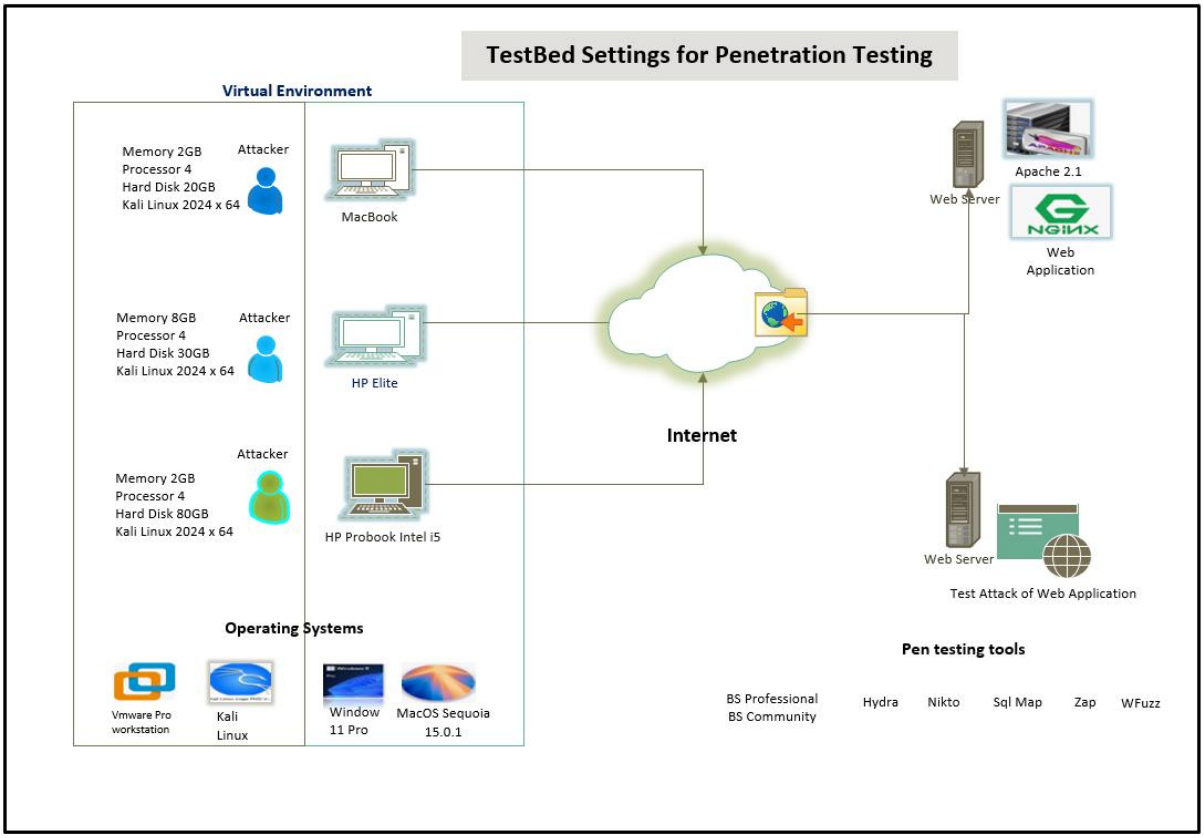


Figure 4 - Test Bed for Penetration Testing

Environment Setup:

Virtualisation and OS Setup

This project will have a testing environment setup. A Kali Linux operating system will be installed in a virtual machine, specifically VMware Fusion, and will be dedicated to this project for penetration testing. Kali Linux is a famous operating system for security professionals because it has pre-installed security and penetration testing tools. By deploying this to VM, we ensure an isolated environment where the tools we tested do not affect other network devices. Also, using virtualisation, there is a feature where we can reverse to a previous state if issues arise during testing.

Separately, DVWA (Damn Vulnerable Web Application) will be installed on a distinct VM which is handled and hosted by our stakeholder, who is working as a Senior Penetration tester Justin Ng. DVWA is a known vulnerable application that is ideal for penetration testing. It allows a real-world simulation of web application vulnerabilities. DVWA is in another VM to protect the other system by not exposing its vulnerabilities.

Figure 5 shows the log-in page of our target web application for testing, which is hosted by our stakeholders.

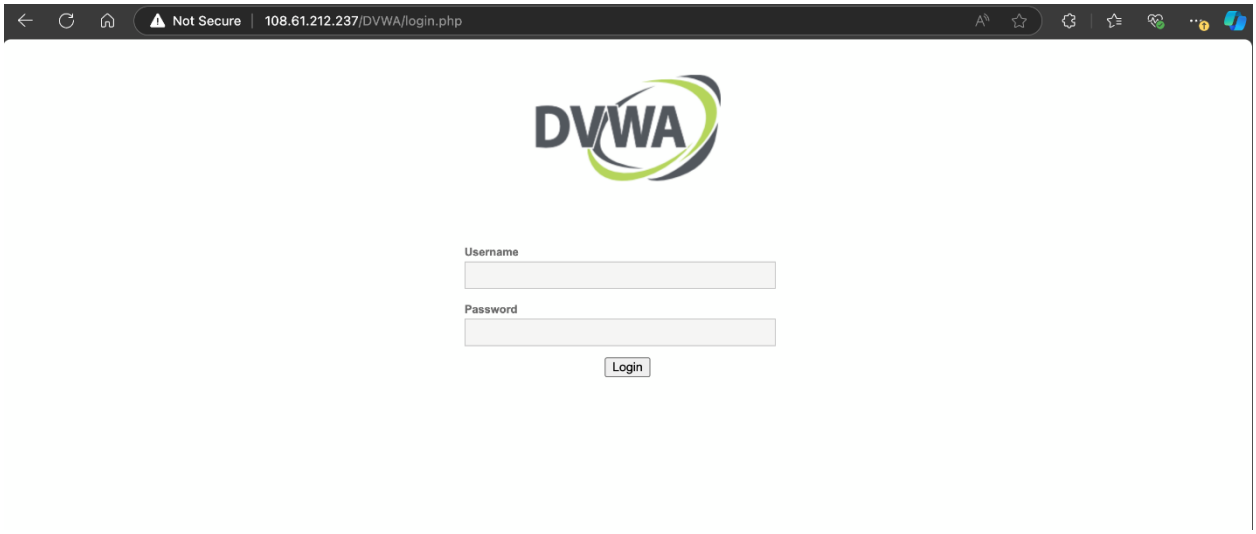


Figure 5 - This figure is the login page of the DVWA with IP 108.61.212.237.

Network Configuration:

Our stakeholders have whitelisted our IP addresses so that we are the only ones who can test the web application he created. We can provide reliable results from all our testing with the selected tools. It also prevents any unauthorised access to the web application. By limiting only team members to test the web application, we are confident that we can execute penetration testing without the risk of exposing other network assets.

Figure 6 shows the login page where our stakeholders whitelisted our IP address, which means that we are the only authorised person to test the system.

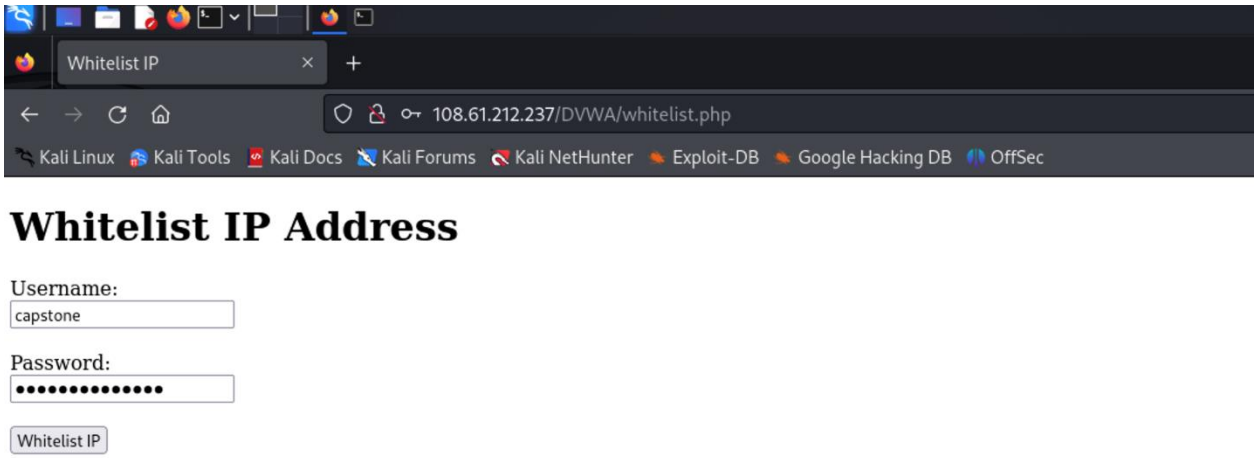


Figure 6 - IP Whitelisting Process

Figure 7 shows that after we log in then our IP already been whitelisted.

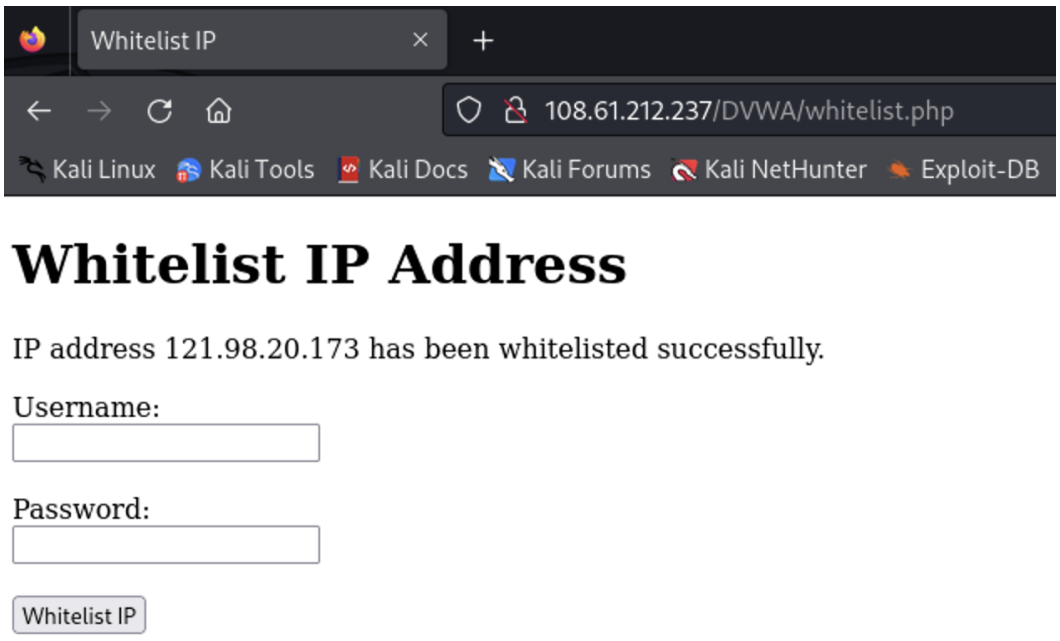


Figure 7 - The IP address has been whitelisted successfully

11.1 Implementation of Burp Suite Professional Tool

Implementation of Burp Suite Professional

The project implementation for the Burp Suite Professional tool will include the following steps:

- Set up Burp Suite Professional
- Login into DVWA
- Capture the Login Request
- Set Attack Positions in the Intruder
- Configure Payload for Brute-Force Attack
- Start the Attack and Analyze the Results
- Document the findings and generate a report

Set up Burp Suite Professional

To start, we need to install Burp Suite Professional from Portswigger website. This version is a paid version with advanced features like automated scanning and detailed reporting of result findings and is ideal for penetration testers and individuals. For this project, we emailed Portswigger support to access the software, and they provided us with three months to use the full version of the Burp Suite Professional. After installation of the software in Kali Linux, configure the proxy settings to listen to a specific port (usually 127.0.0.1:8080). With this setting, Burp Suite can intercept HTTP between the browser and the web application target. The professional version has its own configured browser to save the time of manually changing settings on another web browser.

Login into DVWA

Open the DVWA web application in the browser and log in with the test credentials. The login page is the primary target because it has information vulnerable to brute-force attacks. This step will help us establish a session, and we can identify the request for testing. It is also essential to set the level of security in the DVWA from low security to high to test each successful result at every level of security.

Capture the Login Request

When the Burp Suite' proxy intercept feature is turned on, an attempt to log in will be captured to the Burp Suite interface, which includes login credentials. This capture request is an essential step because we need to have a structure for launching brute-force attacks, and we can define parameters. (e.g. username and password) that the intruder will target. These capture requests need to be sent to the intruder for further analysis.

Set Attack Positions in the Intruder

In the Intruder section, defining the attack position is essential by selecting specific parameters like username and password and choosing password fields as attack positions; when a password is highlighted and added, it allows Burp Suite to substitute values during the attack. This setup is a critical element in the login process during brute forcing.

Configure Payload for Brute-Force Attack

When the login credential is added to the payload, input a list of potential usernames or passwords. For this project, we use password lists provided by Burp Suite Professional. Using a text file with a possible password is also allowed. This version allows multiple payload types and configurations to be added, which maximises the tool's effectiveness and reliable results.

Start the Attack and Analyze the Results

When the payload is set, Click “Start Attack.” Burp Suite will try to brute force the login credential by attempting to combine usernames and passwords from payload lists. By analyzing the results, we can identify the successful login attempts by character length that differ from other parameters.

Document the findings and generate a report

Record all relevant information, which includes the successful login parameters with each level of security and speed of the testing. This tool can generate reports of known vulnerabilities.

11.2 Implementation of Burp Suite Community Tool

Implementation of Burp Suite Community

The project implementation for the Burp Suite Community tool will include the following steps:

- Set up Burp Suite Professional
- Login into DVWA
- Capture the Login Request
- Set Attack Positions in the Intruder
- Configure Payload for Brute-Force Attack
- Start the Attack and Analyze the Results
- Document the findings and generate a report

Set up Burp Suite Community

First, download and install Burp Suite Community Edition from Portswigger website. This free version has essential tools for manual testing, including Proxy, Intruder, and Repeater. This version lacks some features that are only available to the professional version. After installation of the software in Kali Linux, configure the proxy settings to listen to a specific port (usually 127.0.0.1:8080). Configuring the website manually and getting certificate installed in the browser and Burp Suite can then listen and capture requests back to the tool interface.

Login into DVWA

Open the DVWA web application in the browser and log in with the test credentials. The login page is the primary target because it has information vulnerable to brute-force attacks. This step will help us establish a session, and we can identify the request for testing. It is also essential to set the level of security in the DVWA from low security to high to test each successful result at every level of security

Capture the Login Request

Upon enabling Intercept in the Burp Suite, open the configured browser to log in to DVWA. Upon login, the Burp Suite will capture the request log, which is shown as raw request data in Burp Suite. This step is essential as the capture details are needed in brute force attacks. This captured request will then be submitted to the intruder to highlight the parameter for inserting payloads.

Set Attack Positions in the Intruder

In the Intruder section, defining the attack position is essential by selecting specific parameters like username and password and choosing password fields as attack positions; when a password is highlighted and added, it allows Burp Suite to substitute values during the attack. This setup is a critical element in the login process during brute forcing.

Configure Payload for Brute-Force Attack

In this edition, additional configurations are required to handle the payload effectively. Password lists for testing are not readily available. Looking for a compiled password list from GitHub is essential and copying it to the Payload section is necessary for brute forcing.

Start the Attack and Analyze the Results

Launching the attack by pressing the Attack button, Burp Suite will attempt each combination from the list of passwords provided. The community version needs to monitor the response of codes to manually identify successful attacks. It lacks automated analysis features.

Document the findings and generate a report

Document the results after brute force and record the speed for each level of security. Successful brute force will show unusual characters from other password lists. The community edition now has built-in reporting, so every finding is manually compiled. Results from testing are essential for further review and remediation efforts.

11.3 Implementation of Wfuzz Tool

The project implementation for the Wfuzz tool will include the following steps:

- Define the Target and Test Scope
- Prepare Wordlist
- Brute Force Login credentials
- Brute Force command in the terminal
- Document Findings

Define the target

First, we need to identify the target web application we can test. For this project our website is http://108.61.212.237/DVWA/login.php. Determining the test scope could be identifying hidden directories and files or brute-forcing login credentials. For this testing, we tried to brute force the system. Ensure that you are permitted to test the target when doing web fuzzing.

Prepare Wordlists

You can create a wordlist suitable for each type of test. In this testing, we use the wordlists that are available in Kali Linux, which is in the rockyou.txt file that consists of passwords that can be used to brute force the web application.

Example: /usr/share/seclists/Discovery/Web-Content/common.txt

Brute Force login credentials

This step should only be used on a permitted website upon logging into the DVWA by providing credentials. We need to get the Session ID, which is a unique character that can be found when you

right-click the space in the web application, go to the “Inspect” tab, and then click Storage. This is where you will find the “Value”, the web application's session ID.

Figure 8 shows where to get the application session ID.

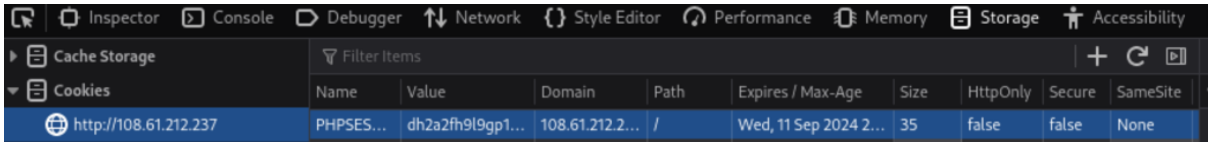


Figure 8 - Application's Session ID

Brute force command in the terminal

Correct use of wfuzz command is very important to generate successful brute forcing on a web application. There are important commands that need to be put into the terminal, which is the directory of the wordlist, setting up the security level, providing PHP session ID, the web application address, special command use in this tool /?username=admin&password=FUZZ&Login=Login.

Sample image:

Figure 9 shows the whole command use in testing the brute force for the web application.

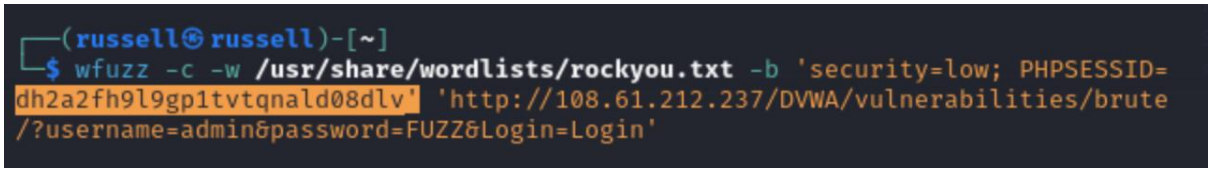


Figure 9 - Command for WFuzz testing

Document Findings

Successful credentials have unique parameters among other wordlists or different status codes or response lengths. Document report if successful in each level of security, which are the low, medium, and high.

Figure 10 shows the result of the successful attack and shows that the length of the correct password is different from other passwords.

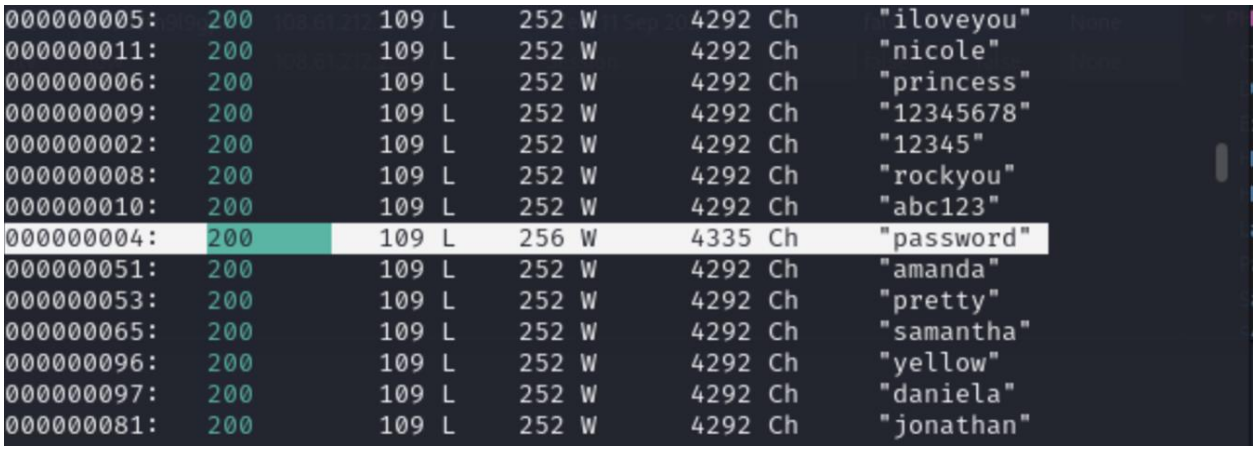


Figure 10 - WFuzz Successful Attack

11.4 Implementation of SQL Map Tool

The project implementation for the SQL Map tool will include the following steps:

- Environment Setup for SQL Map
- Installation and Verification of SQL Map
- Identify Vulnerable in DVWA with URL
- SQL Map Basic test run
- Database Enumeration Performance
- Extraction of Table names from the database
- Retrieve information from a specific table
- Export the result and save the output file
- Documentation and reporting

Environment Setup for SQL Map:

Security Level Configuration: From the homepage, DVWA's security level needs to be set to low, medium, and high from the "DVWA Security" settings tab. This configuration will allow SQL Injection to be accepted on DVWA, which will be ideal for testing SQL Map.

Connectivity with database: To ensure DVWA can connect to the database, we need to check that it can connect with MySQL Database by injecting a command in the field.

Figure 11 shows where the user id field for injection input.

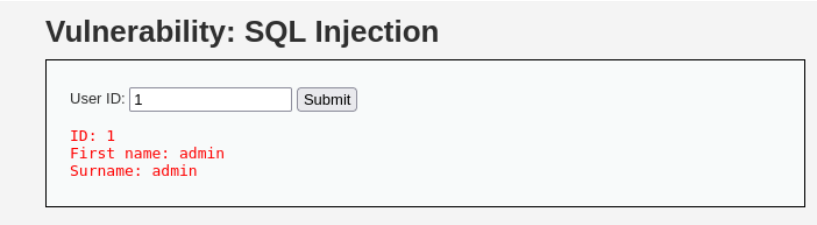


Figure 11 - SQL Injection Input

Installation and Verification of SQL Map:

As we are using Kali Linux, SQL Map has already been installed on the testing machine. In order to verify SQL Map in the testing system, we need to run the command in the terminal: “sqlmap – h”. It will confirm the installation.

Identify Vulnerable in DVWA with URL

To access DVWA’s SQL Injection section, navigate to the left side of the menu and select “SQL Injection”. To test user input, we need to insert a command in the input field: “1' OR '1'='1”

Figure 12 shows the input of SQL command to the field to generate some information from the database.

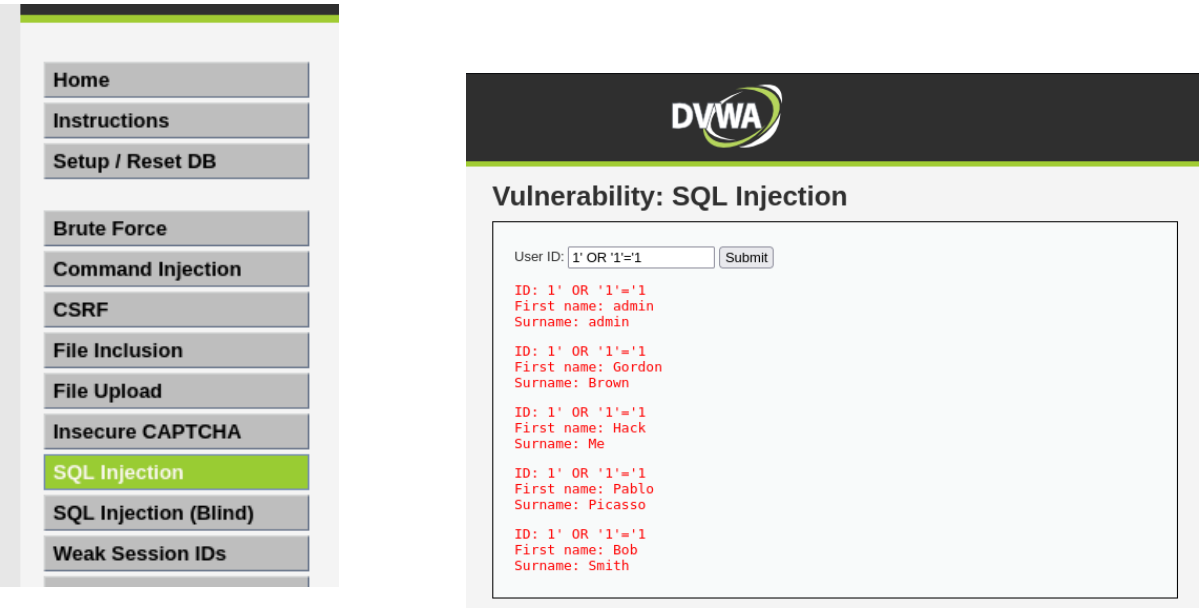


Figure 12 - DVWA SQL Injection on the home Page

SQL Map Basic test run

To test the SQL injection vulnerability, we need to run a basic command in Kali Linux for SQL Map:

“sqlmap -u "http://108.61.212.237/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=2gbp16c0a7lngtkbhe681cm23g; security=low" --banner --current-user --current db –dump”

Explanation: The -u option determines the target URL, and cookies are utilized to maintain the current session token in DVWA.

Database Enumeration Performance

If the detection of SQL Injection exists, the enumeration of database names will appear next. It will list all the available database names, which confirms the vulnerability.

Extraction of Table names from the database

Once the name of the database is available, the extraction of table names can be gathered with the following command:

“sqlmap -u "http://108.61.212.237/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=2gbp16c0a7lngtkbhe681cm23g; security=low" -D dvwa --tables”

Retrieve information from a specific table

To retrieve data from a specific table, SQL Map will display data from the user table with the following command:

“sqlmap -u "http://108.61.212.237/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=2gbp16c0a7lngtkbhe681cm23g; security=low" -D -T users --dump”

As the retrieved information will be used for brute force attack, the value for each user will be in hash values, which will be decrypted by the dictionary attack on that specific output. It will display the user and other details from the database table, and the passwords will be displayed in hash values. SQL Map will be able to crack the hash values during the testing phase.

Export the result and save the output file

To save the output file, we need to run the command in the terminal during the same session. The command for saving the output file:

“sqlmap -u "http://108.61.212.237/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=2gbp16c0a7lngtkbhe681cm23g; security=low" --dump -o”

Documentation and reporting

After executing the command in the terminal, we review the output results. A thorough analysis is performed to determine the most effective way to gather all relevant information, followed by recommendations based on our findings.

11.5 Implementation of Hydra Tool

Hydra is a type of Brute-Force tool used by Penetration Testers. It is designed to crack passwords for various network services, including FTP, HTTP, HTTPS, and SMB databases. More info about this tool can be found at https://github.com/vanhauser-thc/thc-hydra.

Figure 13 shows that Hydra is installed in Kali and provided some key information that will help using the tool.

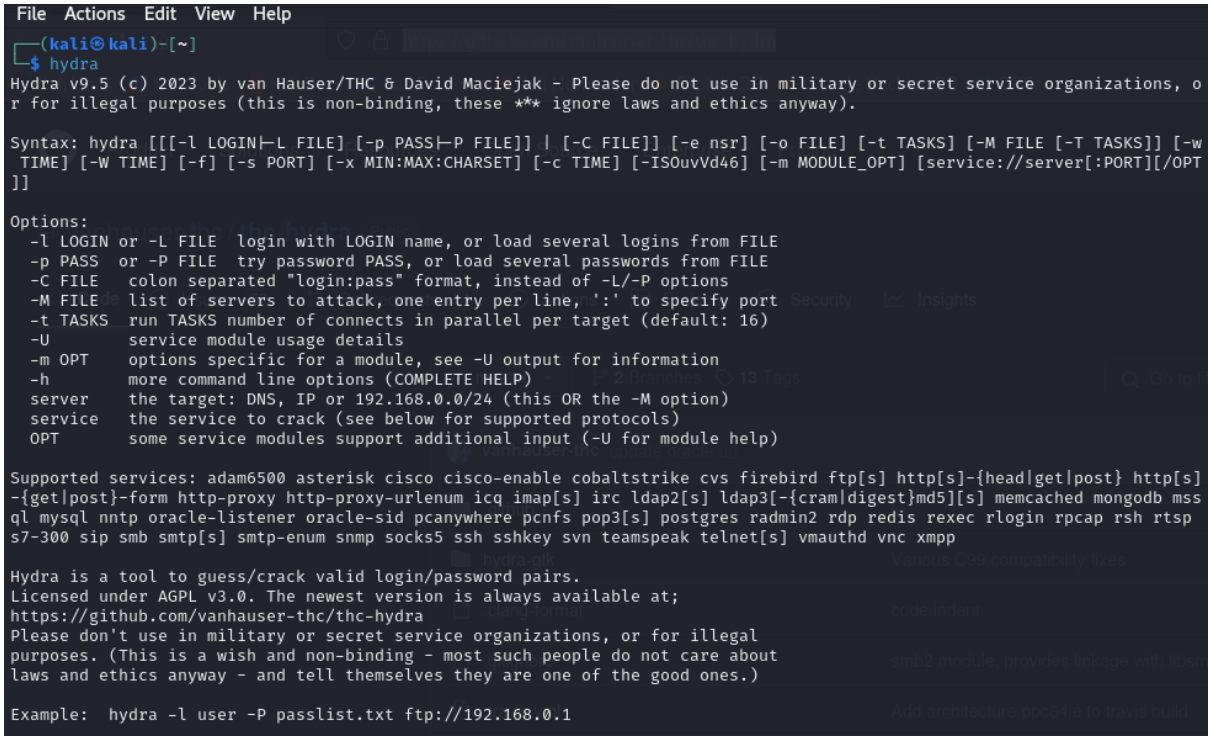


Figure 13 - Confirm Hydra is installed inside Kali Linux with updated version v9.5

1. For Hydra's implementation to fully function, it must implement a) a username list, b) the password, and c) a remote port to be attacked.
2. In this implementation, the rockyou.txt file is utilised, and ensure it is already installed.

Figure 14 shows the location of the text file that will be used to brute force the target web application.

```
┌──(root㉿kali)-[/home/kali]
└─# cd /usr/share/wordlists

┌──(root㉿kali)-[/usr/share/wordlists]
└─# ls
amass  dirbuster   fasttrack.txt  john.lst  metasploit  rockyou.txt  sqlmap.txt
dirb   dnsmap.txt  fern-wifi      legion    nmap.lst    seclists     wfuzz
```

Figure 14 - Confirm that the rockyou.txt file is available in the above directory

Figure 15 shows the command used to brute force the web application.

```
┌──(root㉿kali)-[/]
└─# hydra -l admin -P /usr/share/wordlists/rockyou.txt  108.61.212.237 http-get-form "/DVWA/vulnerabilit
ies/brute/index.php:userfield=^USER^:passwordfield=^PASS^" -s80
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service
 organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).
```

Figure 15 - For Testing on the DVWA website

11.6 Implementation of Nikto Tool

Nikto is an open–source Web Server scanner that identifies vulnerabilities, outdated server software, and security misconfiguration. It also has comprehensive testing and security issues [13].

Figure 16 shows that the tool Nikto is already installed in Kali Linux as a free tool for penetration testing.

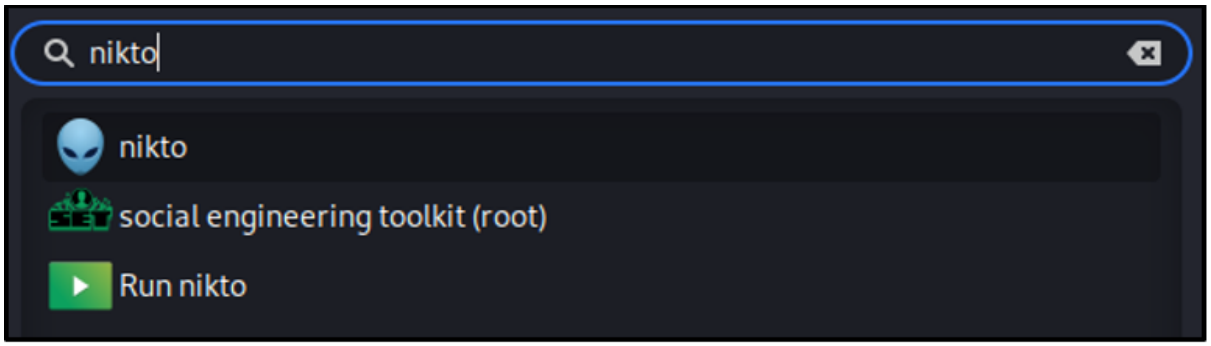


Figure 16 - This is Nikto preinstalled within Kali Linux

Nikto tool can be accessed by using the Terminal command in Kali. At the time of this implementation, the tool version is shown below in Figure 17:

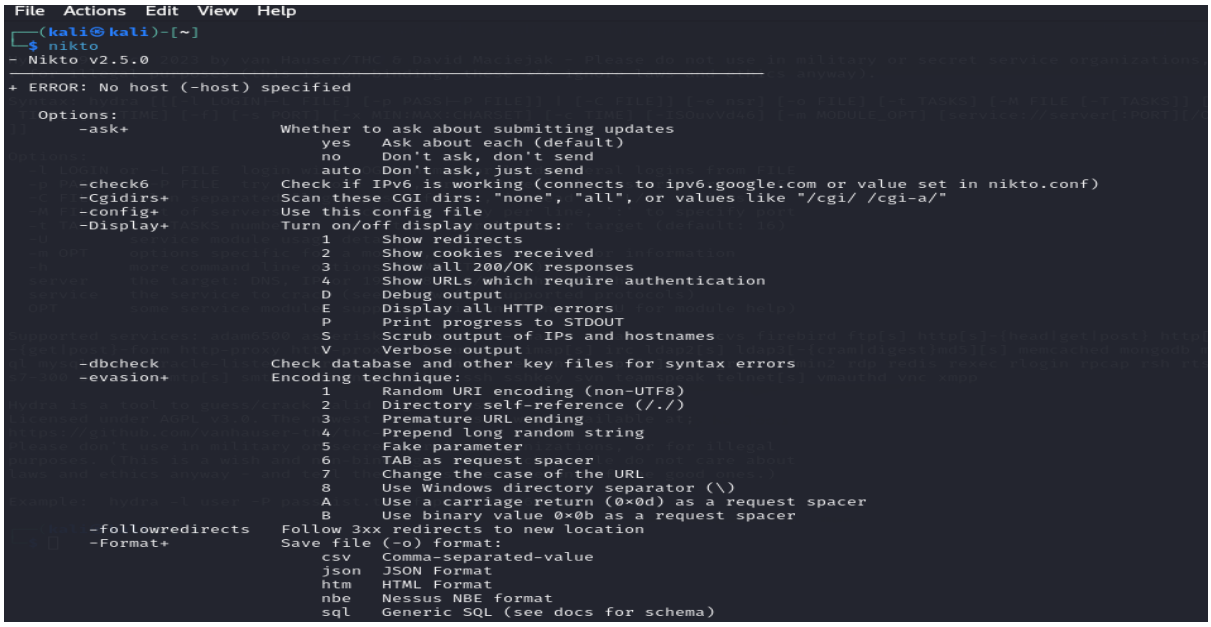


Figure 17 - Confirm Nikto tool already implemented in Kali Linux

Figure 18 provide us information about the target web application and open ports in the system.

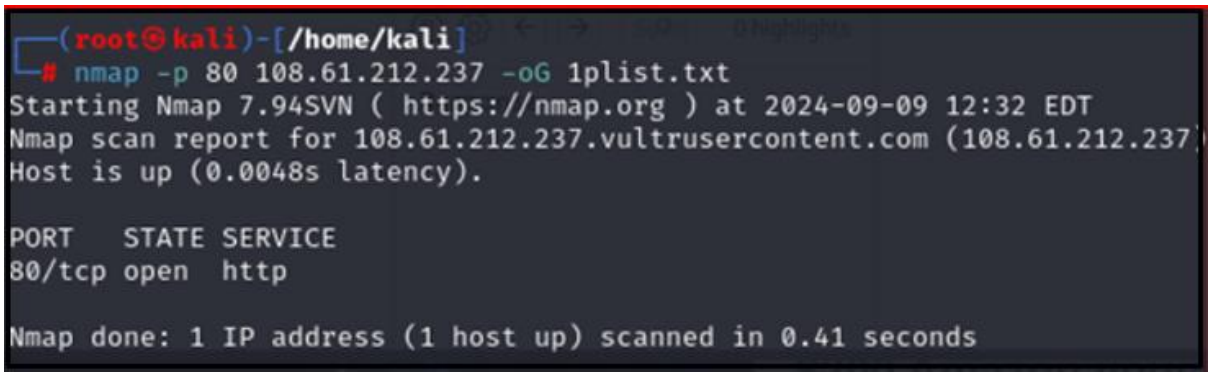


Figure 18 - Nikto Output

For testing purposes of the DVWA website using Nikto, we also utilize the Nmap command to scan open ports in the DVWA website address (108.61.212.237), targeting port use(p80), and report files (1plist).

Figure 19 shows some report findings from the web application and show signs that the web application is ready for testing.

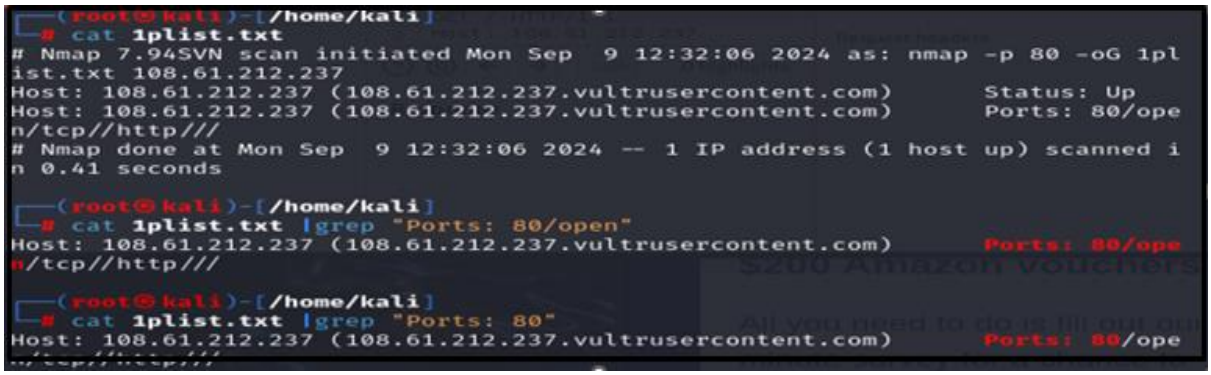


Figure 19 - Confirm Nikto implementation is ready for the Testing Session

11.7 Implementation of ZAP Tool

ZAP is an open-source web application security testing tool developed by OWASP. Preinstalled within Kali Linux or can be downloaded from the OWASP download website, available in Windows or Linux [14].

Figure 20 shows the front page of the Zap tool after installation and with the tool version.

Figure 20 - Installation of Zap tool version used v2.15.0

Figure 21 shows the Manual Explore section, where settings are configured to test the web application.

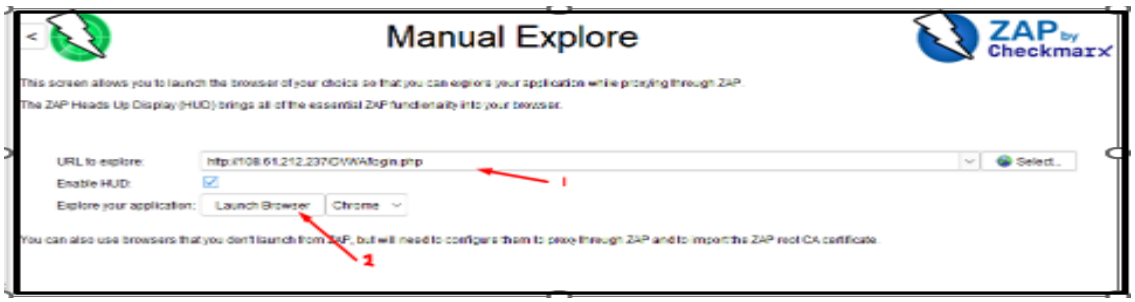


Figure 21 - Manual Explore Setting where the setting of the 1) targeting URL website before 2) Launch Browser is entered either Chrome or Firefox platform

Browser is entered either Chrome or Firefox platform

Figure 22 shows some folder locations of the target and getting parameters.

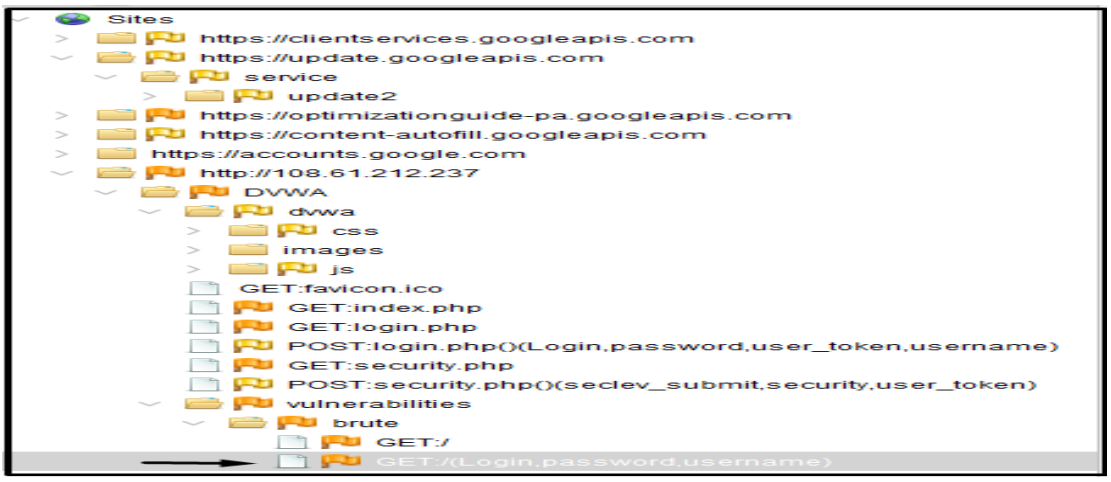


Figure 22 - To set the target site before testing

12.0 Testing

12.1 Burp Suite Professional Tool Testing

Few steps to follow in Burp Suite Community Testing

- Prerequisite and setup
- Testing Step

Prerequisites and setup

The Burp Suite Community should be installed in Kali Linux inside Vmware Fusion for testing.

Testing Steps:

Step 1: Set up Burp Suite Professional

1. Open the Burp Suite Professional and go to the “Proxy” section.
2. Burp Suite has its browser, which has already configured the settings and has a default port of 8080.
3. If using a different browser (e.g., Firefox or Chrome), use Burp’s proxy. The setting should be changed with HTTP Proxy to 127.0.0.1; the port number is 8080.
4. Additionally, in different browsers, install the Burp Suite CA certificate in the browser for HTTPS interception, which is under the “Proxy” > “Options” section.

Step 2: Login into DVWA

1. Open the DVWA website in the Burp Suite Browser and log in using the provided credentials (admin/password).

Figure 23 shows the main page of the DVWA web application and provides credentials to access the web.

Figure 23 - DVWA Login Page

2. For the initial testing, we put the DVWA security level in the “Low” setting to check vulnerabilities and for testing purposes.

Figure 24 shows that the web application allows us to change security level for testing.

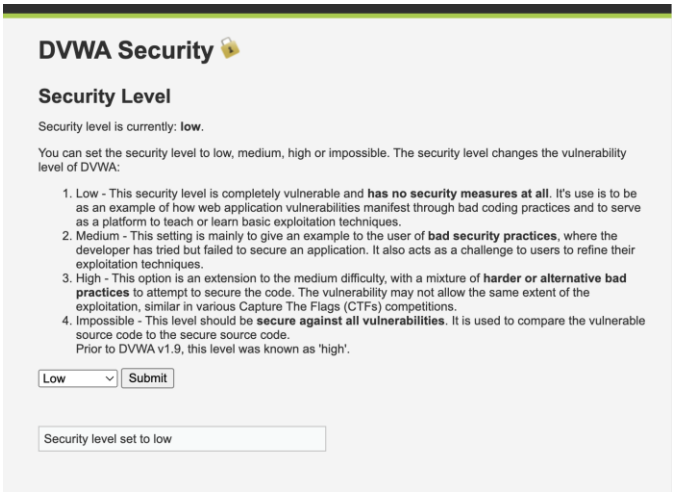


Figure 24 - DVWA Security Level

3. Under “Brute Force” on the DVWA main page, the Brute-force testing takes place.

Figure 25 shows where brute force section for testing and input of credentials.

Figure 25 - Brute Force Page

Step 3: Capture the Login Request

1. Under Burp Suite, make sure that “Intercept” is on by clicking the “Intercept” button under the “Proxy” section.

2. In DVWA, under Brute Force, insert the credentials (e.g., admin/password), then click “log in.”

Figure 26 shows where we need to input the credentials for brute forcing the system.

Figure 26 - User input

3. Burp Suite Professional will then intercept the request under the “Proxy” > “Intercept” section; we can check that we have already captured the HTTP request.

Figure 27 shows that after the login, it was then captured to the Burp Suite for further analysis.

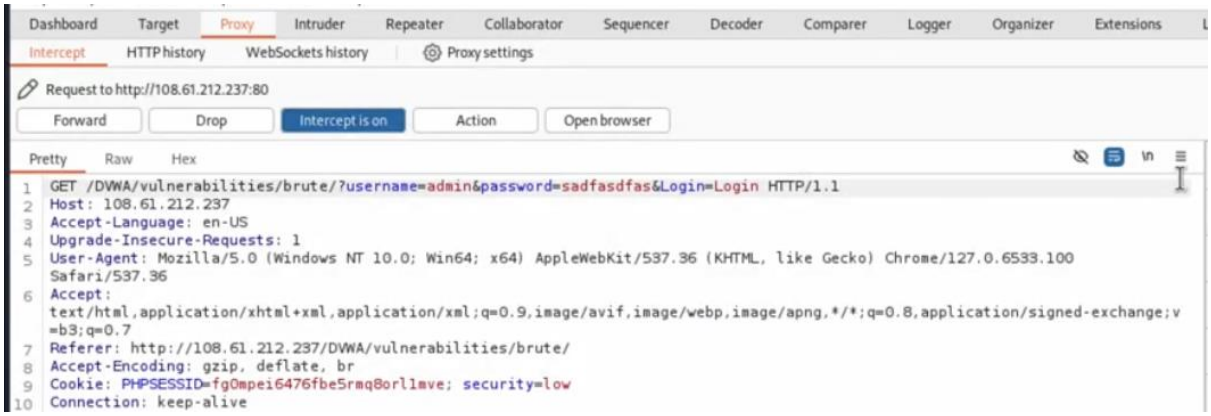


Figure 27 - Intercept login

4. Right-click on the page and select “Send to Intruder” to begin the brute force attack.

Step 4: Set Attack Positions in the Intruder

1. Under the “Intruder” section, open the request that is sent from the proxy.
2. Click the “Position” tab to check attack points. On this page, we can see the request placeholder for username and password parameters.

3. We should clear the preset parameters and manually highlight the password parameters to set it as the attack parameter.

Figure 28 shows after the captured information was sent to the intruder section to highlight important parameters.

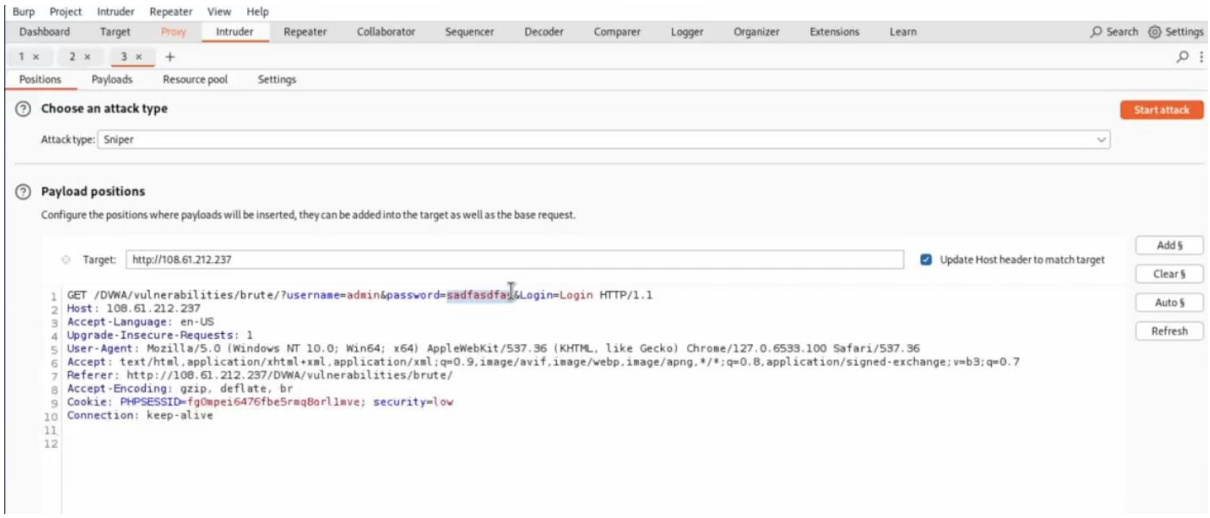


Figure 28 - Clear preset parameters

4. After highlighting the password, click the add icon and set this attack as a “sniper” attack.

Figure 29 shows that we used sniper attack for brute force method.

Figure 29 - Sniper attack

Step 5: Configure Payload for Brute-Force Attack

1. Click the “Payload” section and choose a list of common passwords. (e.g., from Burp’s list of passwords, or you can use your own wordlist of passwords).
2. In the “Payload Option” section, add the password list that you want to use and then begin the testing.

Figure 30 shows sample of a payload that is used for web testing.

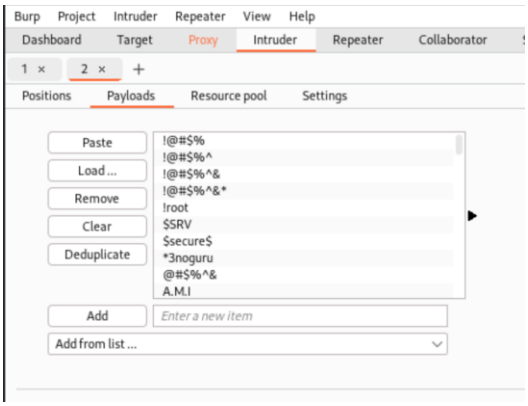


Figure 30 – Payload

Figure 31 shows another payload that is getting some error phrases that can be used to analyze and get more reliable results.

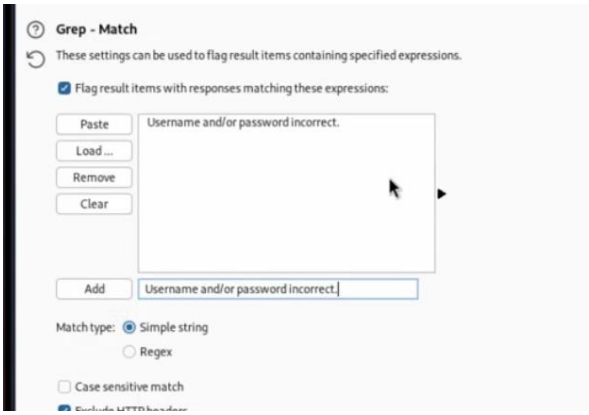


Figure 31 - Payload Grep match

Step 6: Start the Attack and Analyze the Results

1. Upon clicking “Start Attack,” it will begin the process of brute force.
2. Burp Suite Professional will attempt each list and display it in the status field.
3. We can Identify if it is successful by checking the HTTP response code and response length. The correct password will show a different response length than other failed attempts.

Figure 32 shows the result of a successful password attack where the length of the character is different from other password lists.

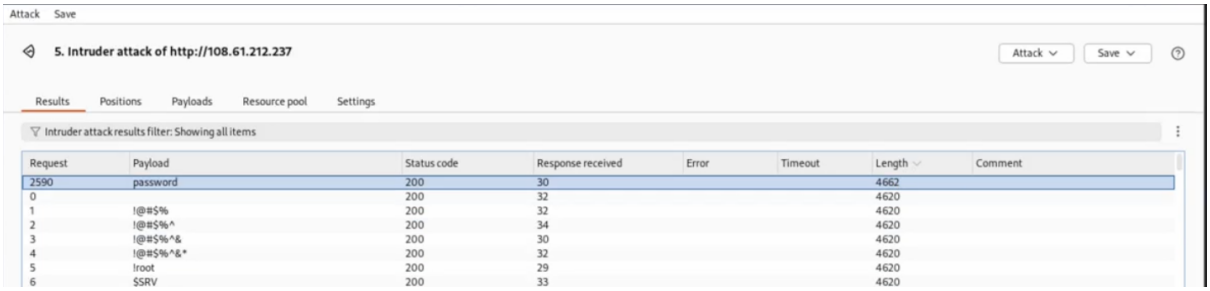


Figure 32 - Password response length

Figure 33 shows another result, which includes another payload to confirm the password is reliable.

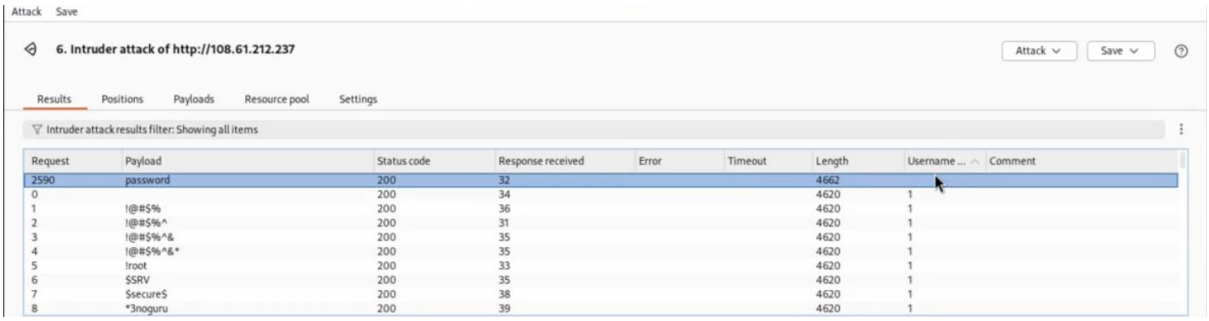


Figure 33 - Password Grep match output

Step 7: Document the finding and generate a report

1. Once we get the password verified. Document it using the performance metrics created in this project.
2. Include recommendations for the application and securing log-in from brute force attacks.

12.2 Burp Suite Community Tool Testing

Few steps to follow in Burp Suite Community Testing

- Prerequisite and setup
- Testing Step

Prerequisites and setup

The Burp Suite Community should be installed in Kali Linux inside Vmware Fusion for testing.

Testing Steps:

Set up Burp Suite Community

1. Open the Burp Suite Community and go to the “Proxy” section.
2. Set up the proxy for interception to port 8080.

3. Configure the browser proxy setting and use HTTP proxy to 127.0.0.1 and port 8080. During interception, click Manual.

Figure 34 shows the browser's manual method of configuring before the testing in the Burp Suite.

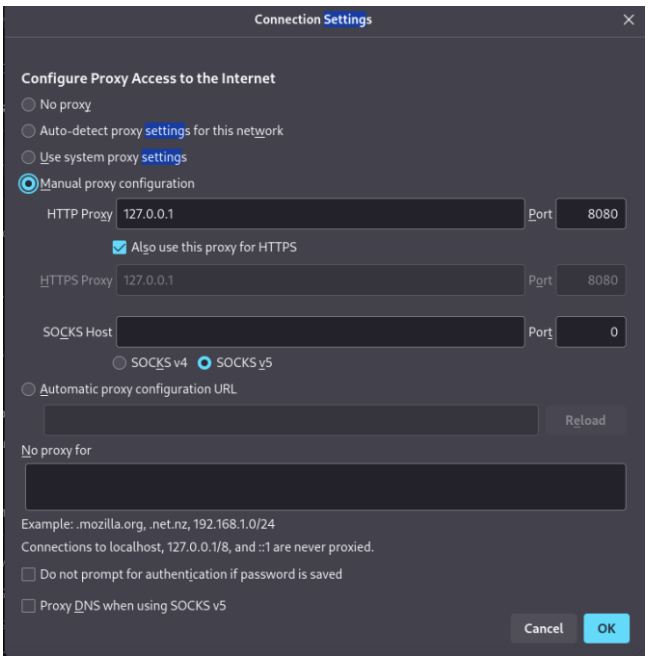


Figure 34 - Configuring Proxy

4. Install the Burp Suite CA certificate in the browser for HTTPS interception, which is under the “Proxy” > “Options” section.

Figure 35 shows the certificate location that is essential to capture request to the Burp Suite tool.

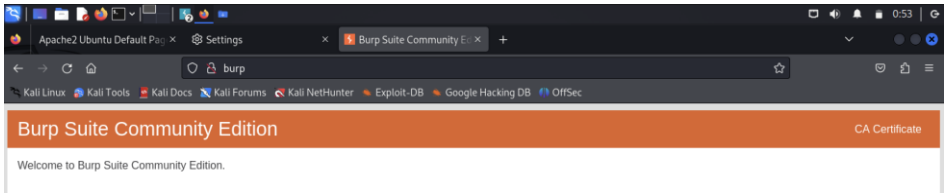


Figure 35 - Burp Suite CA certificate

5. 6. Go to Download and find the download certificate /downloads/cacert, then pre ok

Figure 36 shows that the certificate is downloaded and provides authorization for collecting requests from the web application.

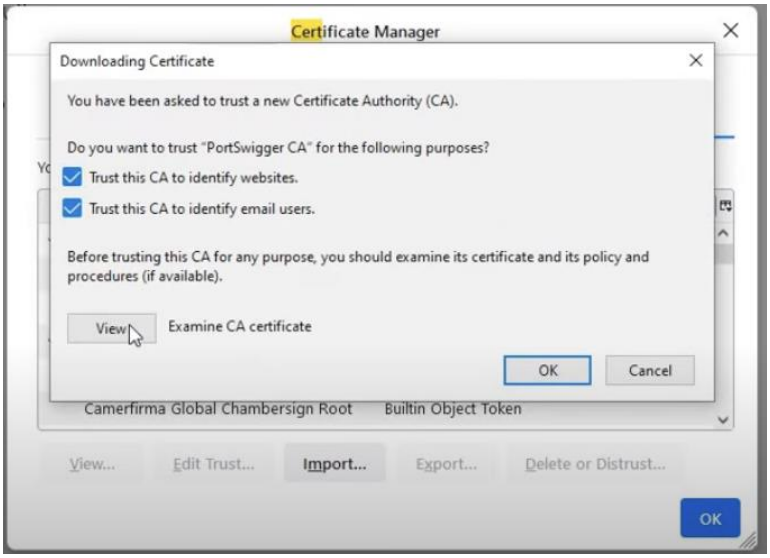


Figure 36 - Download Certificate

Login into DVWA

1. Open the DVWA website in the Burp Suite Browser and log in using the provided credentials (admin/password).

Figure 37 shows the location of the section for the Brute Force technique in the Burp Suite Community.

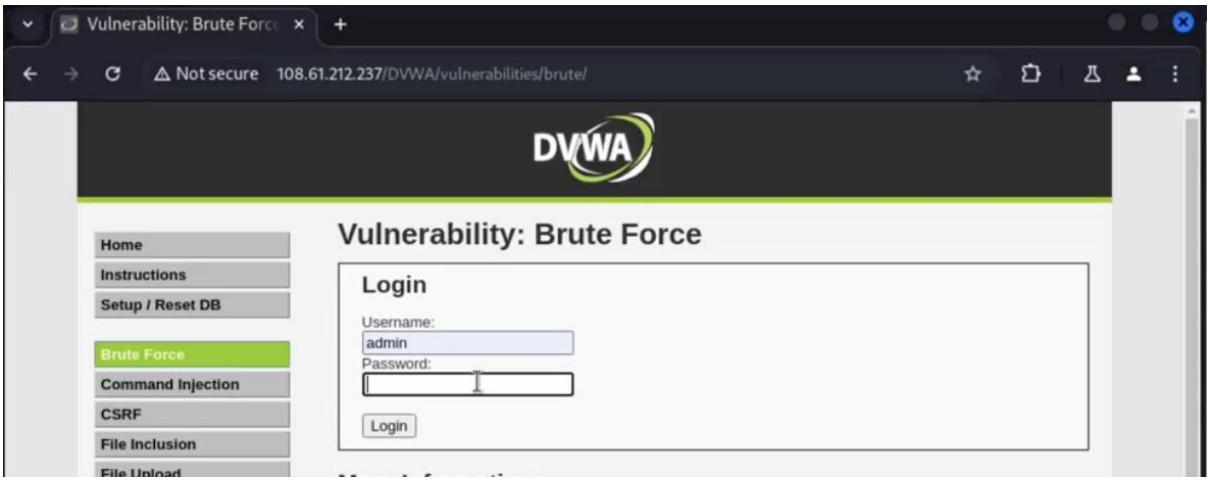


Figure 37 - DVWA Homepage

2. For the initial testing, we put the DVWA security level in the "Low" setting to check vulnerabilities and for testing purposes.
3. Under "Brute Force" on the DVWA main page, the Brute-force testing takes place.

Capture the Login Request

1. Under Burp Suite Community, make sure that "Intercept" is on by clicking the "Intercept" button under the "Proxy" section.
2. In DVWA, under Brute Force, insert the credentials (e.g., admin/password), then click "log in."

3. Burp Suite Community will then intercept the request under the “Proxy” > “Intercept” section; we can check that we have already captured the HTTP request.

Figure 38 shows that the intercept is on to collect or capture request from the target web application.

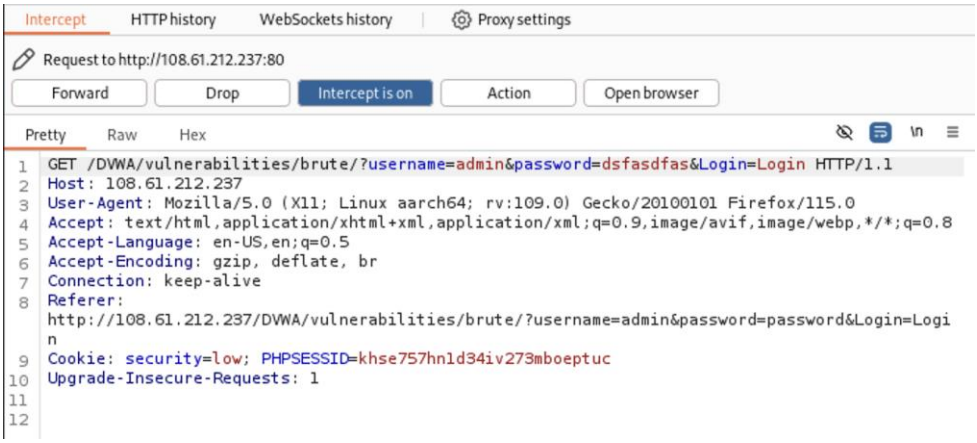


Figure 38 - Intercept request

4. Right-click on the page and select “Send to Intruder” to begin the brute force attack.

Set Attack Positions in the Intruder

1. Under the “Intruder” section, open the request sent from the proxy.
2. Click the “Position” tab to check attack points. On this page, we can see the request placeholder for username and password parameters.
3. We should clear the preset parameters and manually highlight the password parameters to set them as attack parameters.

Figure 39 shows the parameters cleared and then highlights the credentials that are going to be used for brute force.

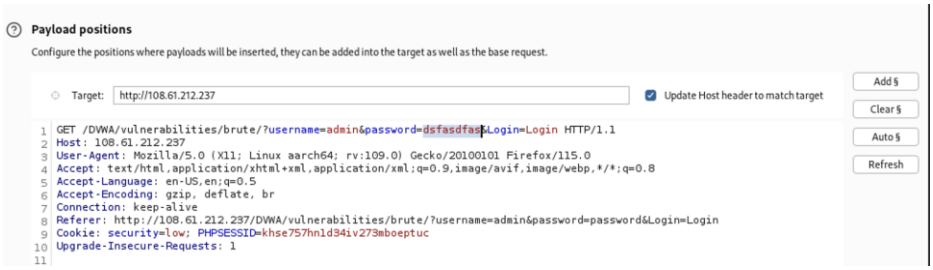


Figure 39 - Clear the preset parameters

4. After highlighting the password, click the add icon and set this attack as a “sniper” attack.

Configure Payload for Brute-Force Attack

1. Click the “Payload” section and choose a “Simple list” as a payload type. (e.g. can get from github user content password list and only a simple list due to the limitation of the community version).

Figure 40 shows that the payload is pasted from website lists that are used in this testing.

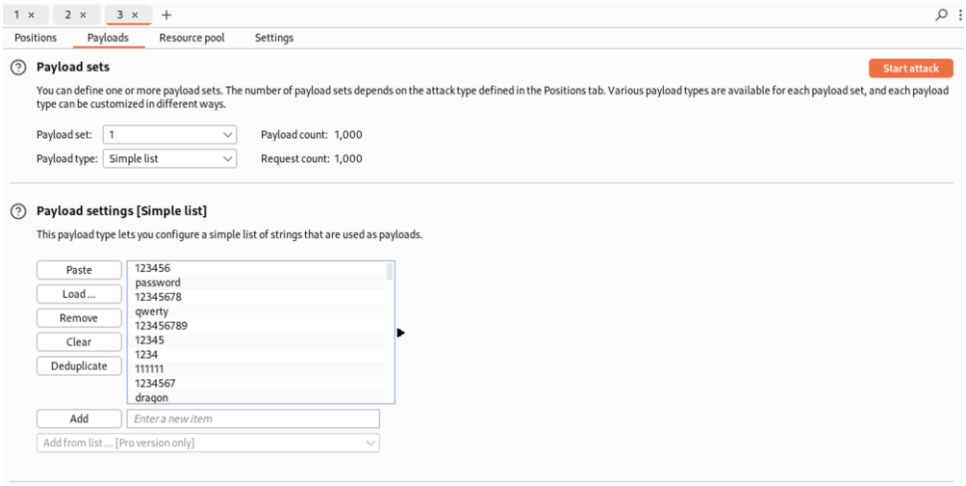


Figure 40 - Payload sample list

2. In the “Payload Option” section, add the password list that you want to use and then begin the testing.

Start the Attack and Analyze the Results

1. Upon clicking “Start Attack,” it will begin the process of brute force.
2. Burp Suite Community will attempt each list and display it in the status field.
3. We can Identify if it is successful by checking the HTTP response code and response length. The correct password will show a different response length than other failed attempts.

Figure 41 shows some details about the attack, and the results are successful.

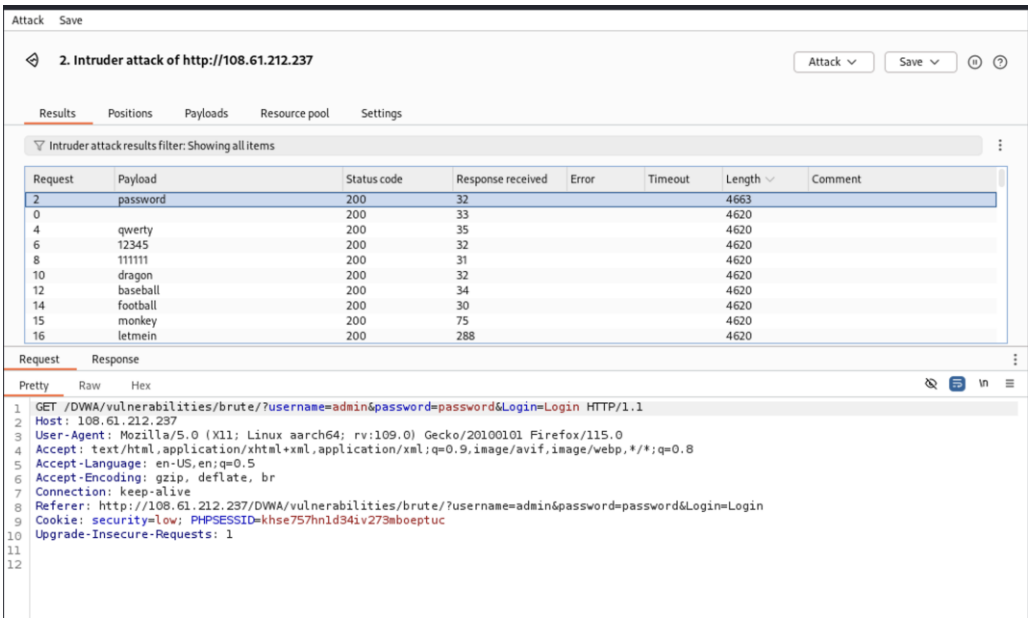


Figure 41 - Start the Attack and get a response

Document the findings and generate a report

1. Once we get the password verified. Document it using the performance metrics created in this project.
2. Include recommendations for the application and securing log-in from brute force attacks.

12.3 Wfuzz Tool Testing

Steps of use:

1. Open terminal
2. Select a password list in the drive.
   /usr/share/wordlist/rockyou.txt
3. Log in to DVWA and provide credentials.
4. Under security, change the security level to low for this testing to check vulnerabilities.
5. Get the session ID by right-clicking somewhere on the screen. Select Inspect and click in the storage section.

Figure 42 shows the location of the session ID that is essential for using Wfuzz.

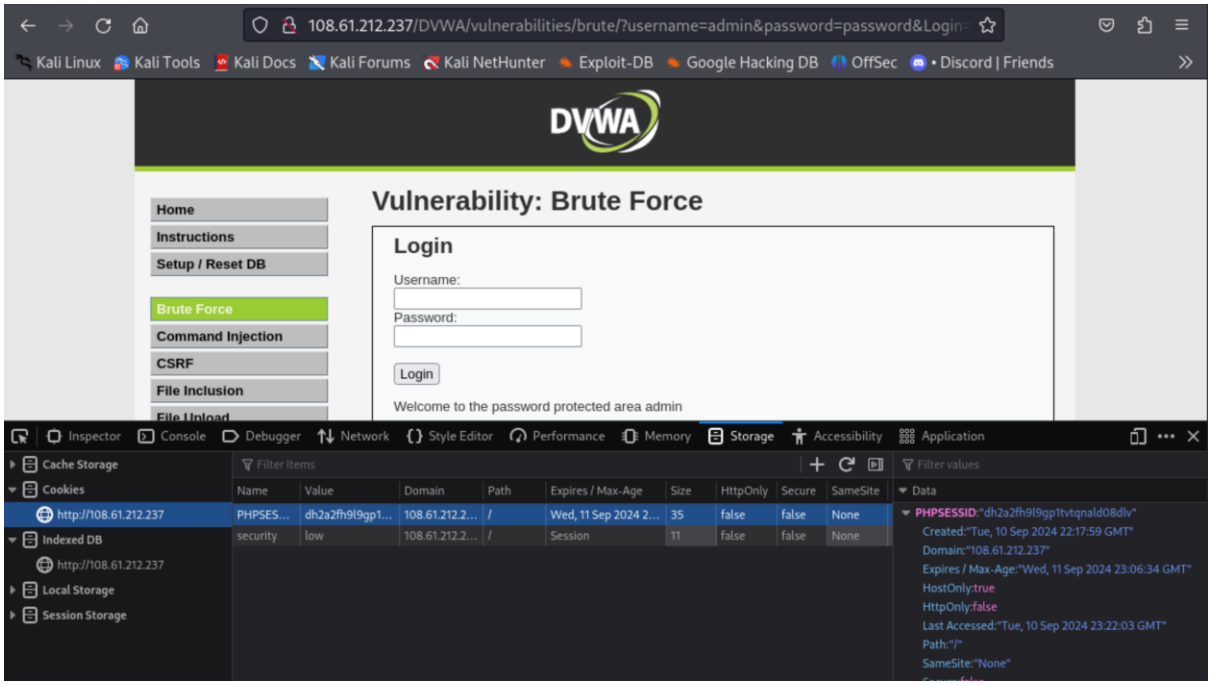


Figure 42 - Session ID for WFuzz

6. Copy the session ID that need to be put on the command in the terminal for web fuzzing.

Figure 43 shows that the session ID is under the "Value" section and copying it to the Wfuzz command.

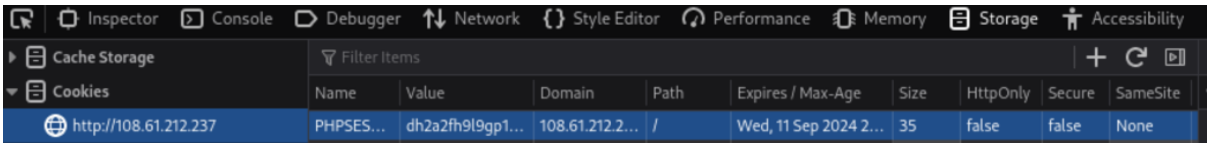


Figure 43 - Copy Session ID

7. In the terminal use the command wfuzz -c -w /usr/share/wordlist/rockyou.txt followed by the security level, session id, and the web application address to begin the brute forcing.

Figure 44 shows the command used in the testing where we put the location of the text lists of passwords and setting security level, providing session ID, address of the target, and some commands used in Wfuzz.

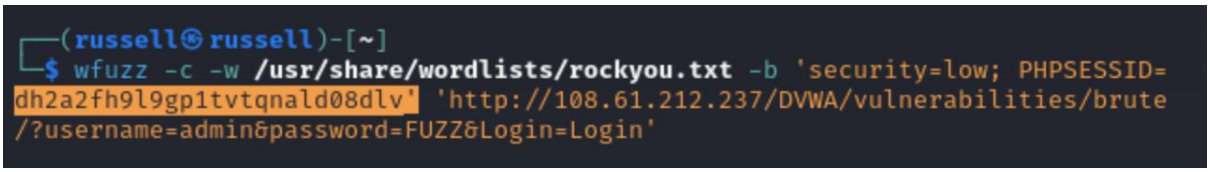


Figure 44 - WFuzz command

8. Add this command after the web address.

Figure 45 shows the web fuzzing command used in Wfuzz.

Figure 45 - Command after web address

9. Result will be shown, parameters are different for the correct password.

Figure 46 shows the successful result where the character length is different from the rest of the password list.

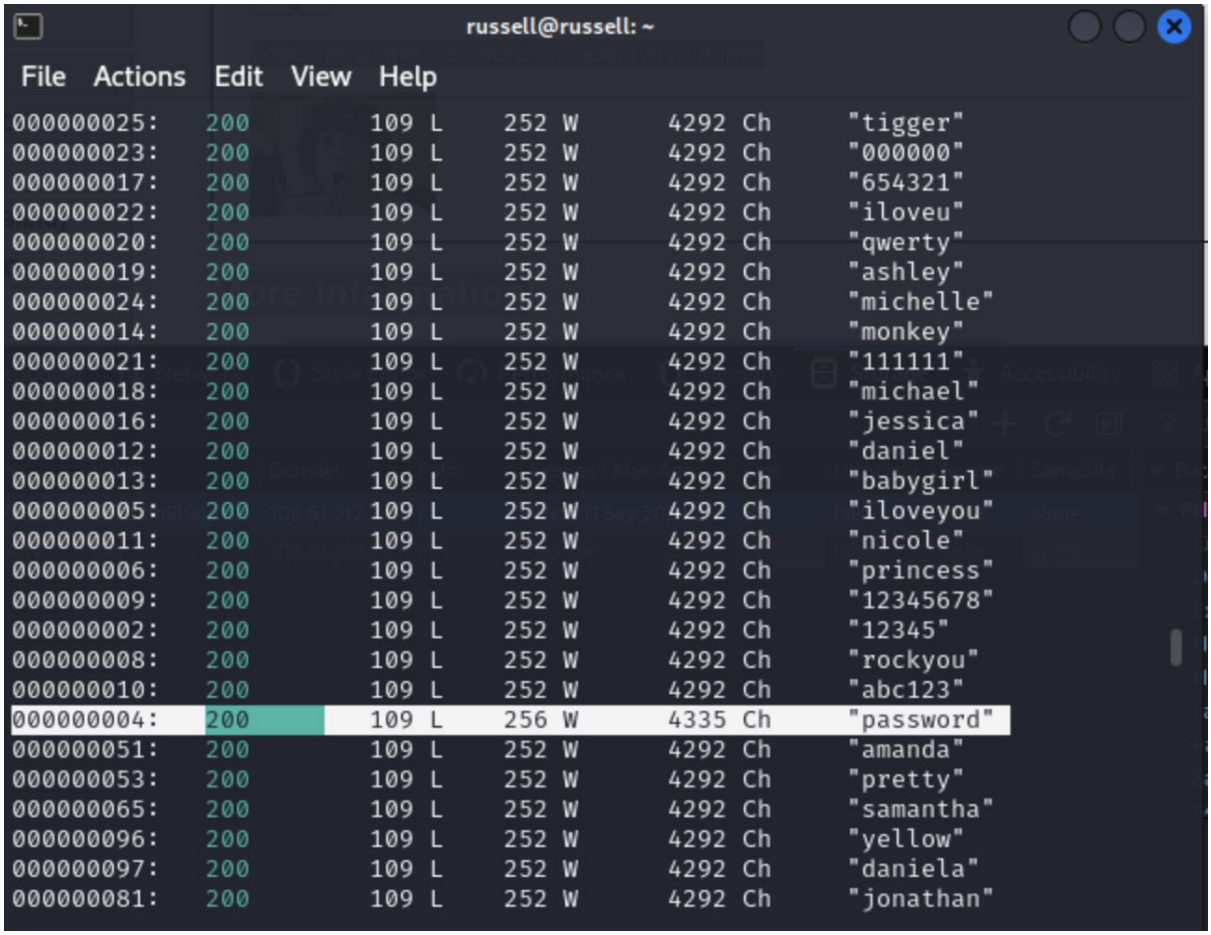


Figure 46 - WFuzz successful output

12.4 SQL Map Tool Testing

There are a few steps for SQL Map Testing.

- Prerequisites and setup
- Testing Steps

Prerequisites and setup

The current session ID is required while testing on DVWA. Also, the SQL Injection command field needs to be tracked to use in SQL Map.

Testing Steps:

Step 1:

Confirmation of SQL Injection in DVWA and retrieve the URL for further testing.

URL: http://108.61.212.237/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#

Figure 47 shows the successful injection in DVWA.

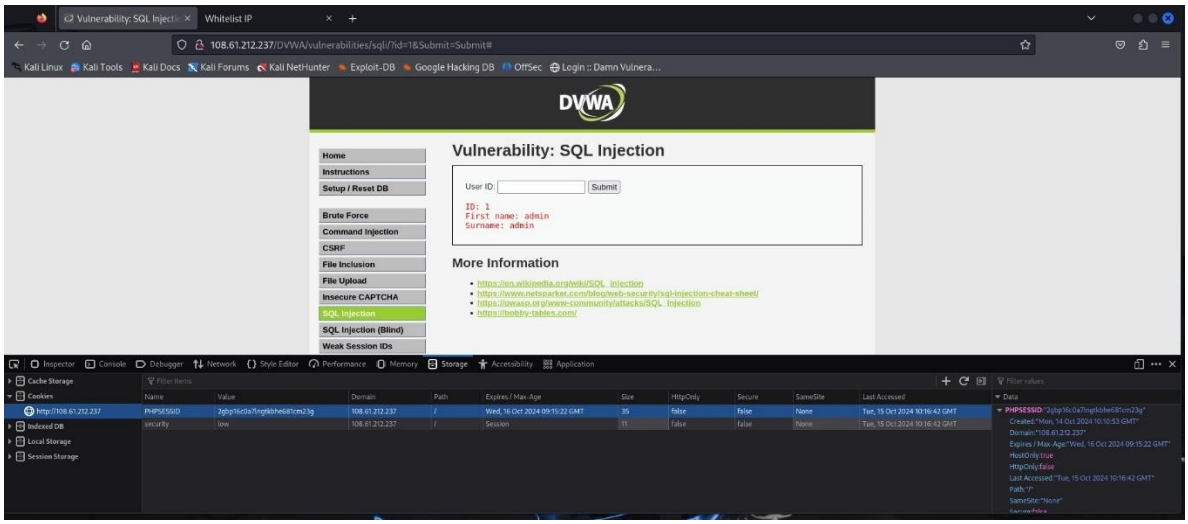


Figure 47 - Confirmation of SQL Injection in DVWA

Step 2:

Inject command in SQL Map for Penetration testing.

Command: sqlmap -u "http://108.61.212.237/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=2gbp16c0a7lngtkbhe681cm23g; security=low" --banner --current-user --current db –dump

Figure 48 shows that in the terminal, this command is use to activate SQL Map tool and injecting command for collecting information from the web application.

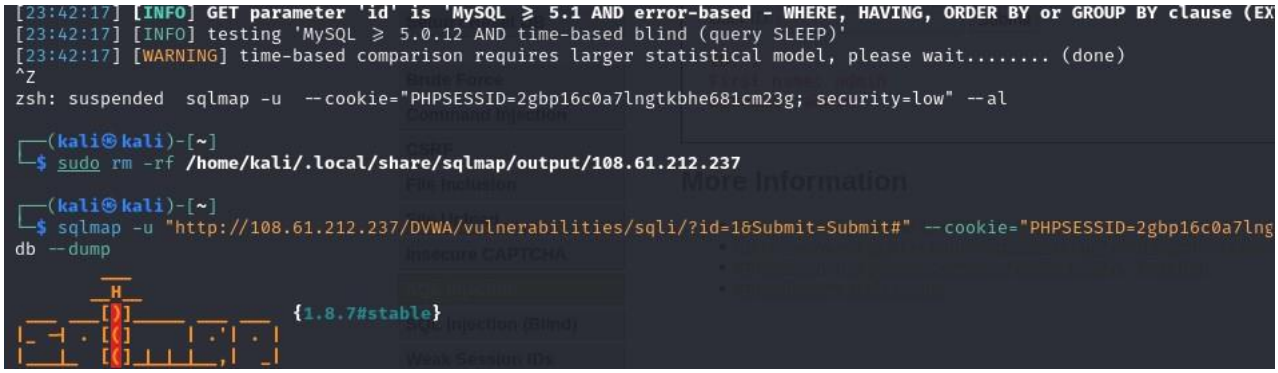


Figure 48 - Inject command in SQL Map for Penetration testing

Step 3:

Testing connection to the target URL

Testing if GET parameter is dynamic

Basic test shows that GET parameter might be injectable – MySQL

GET parameter appears to be Boolean based blind – injectable

Figure 49 shows that the tool is connected to the target web application using some GET parameters.

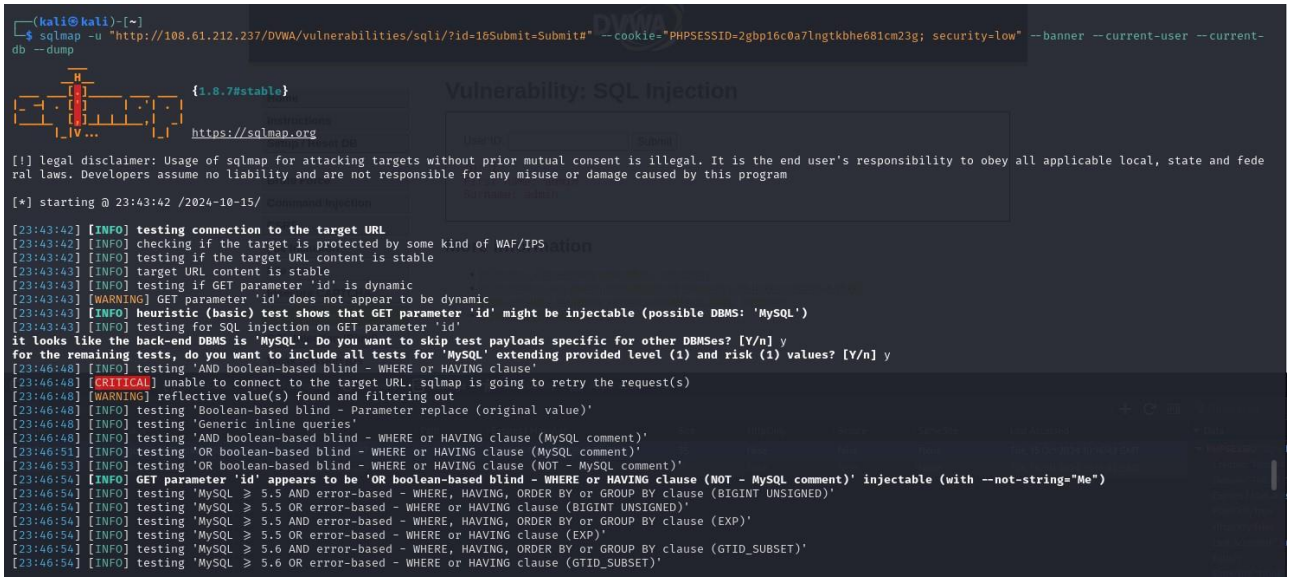


Figure 49 - Testing connection to the target URL

Step 4:

GET parameter appears to be MySQL >5.0.12 and time-based blind – injectable

GET parameter is MySQL Union query – 1 to 20 columns – injectable

System asking for other vulnerabilities – Yes

Figure 50 shows information about the Get parameter and “id” vulnerable and uses MySQL. The system asks if the user wants to get more information or further testing.

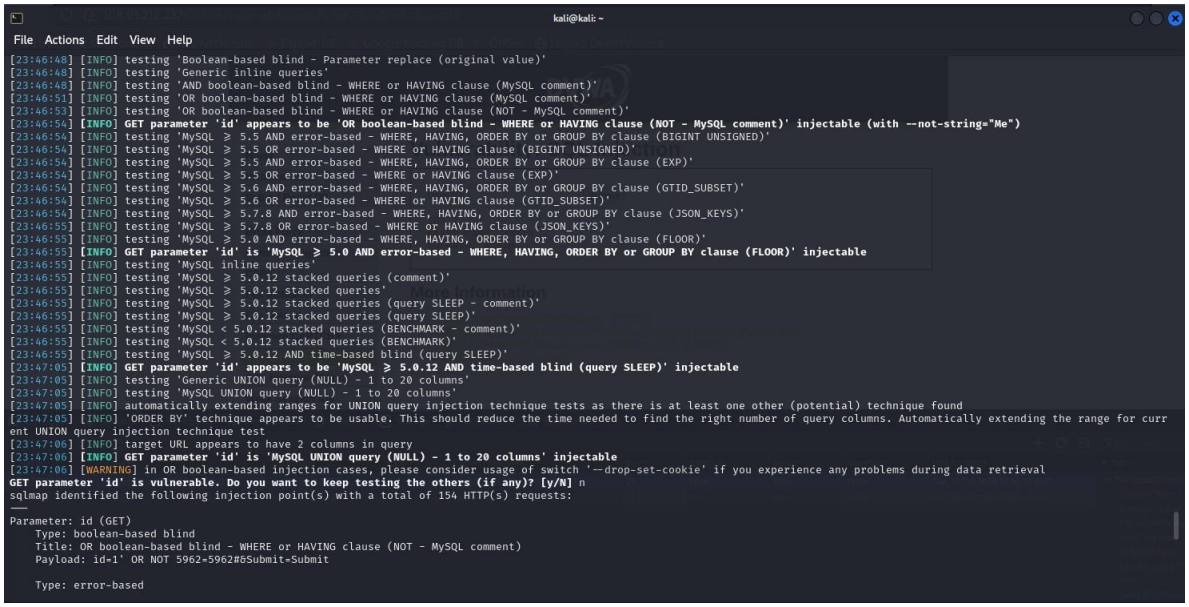


Figure 50 - GET parameter appears to be MySQL

Step 5:

Back-end information details were retrieved.

Storing hash values to a temporary file - yes

Cracking via dictionary-based attack – Yes

Figure 51 shows information about the hash value that is stored temporarily in a file location and asks if you want to proceed to crack the hash value into the correct password.

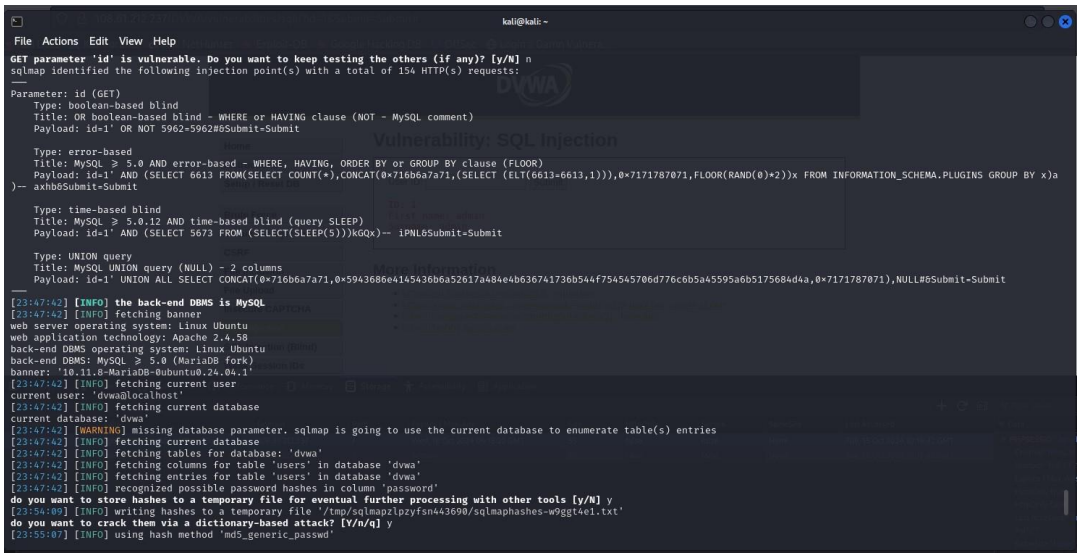


Figure 51 - Back-end information details were retrieved

Step 6:

Default dictionary file: '/usr/share/sqlmap/data/txt/wordlist.txt

Use common password suffix – Yes

Output:

Database: DVWA

Tables: users

Entries: 5

Hash Values: the password is retrieved as hash values for every user.

Cracking the hash values for every user's password with details.

Figure 52 shows that the hash value has been converted to the correct password and successfully penetrated the system with the tool.

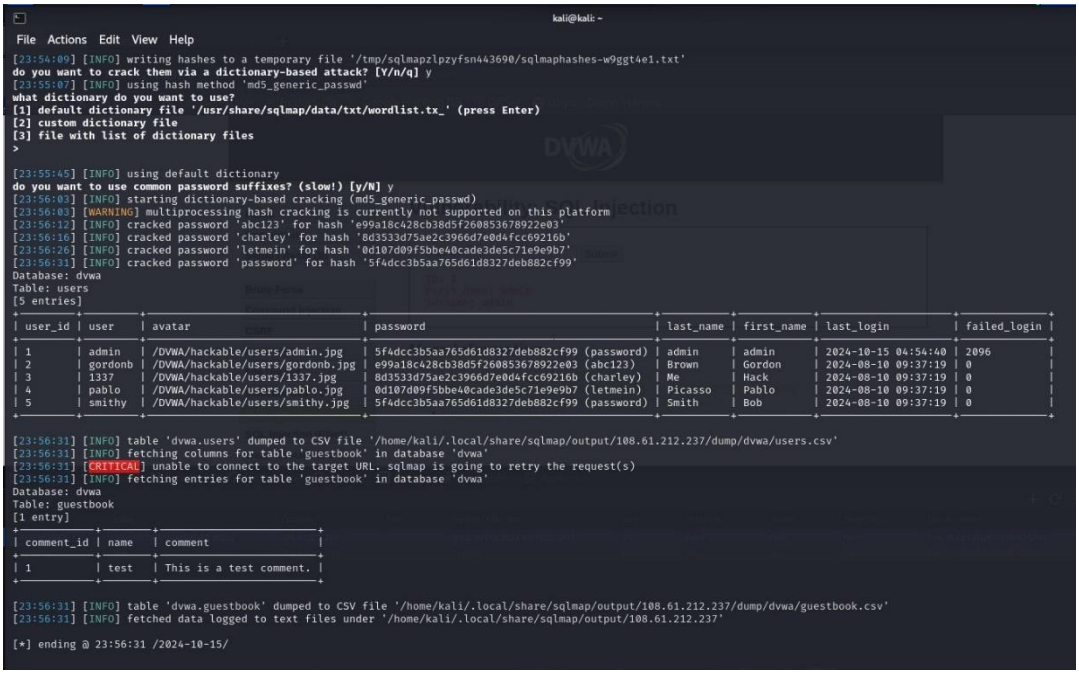


Figure 52 - Cracking the hash values for every user's password with details

Step 7: To locate and remove the directory file for SQL Map output result (if required):

Command: sudo rm -rf /home/kali/.local/share/sqlmap/output/108.61.212.237

For Medium and High Security, "--data="id=1&Submit=Submit"" specifies HTTP Post parameters which is sent in the request. SQL Payload will try to inject into the id parameter. Submit=Submit is really important if this parameter is expected to process.

For Medium and High Security the command is:

sqlmap -u "http://108.61.212.237/DVWA/vulnerabilities/sqli/" --cookie="PHPSESSID=9e6mnhtrmkfco5v4e3o1i9nnkg; security=medium" --data="id=1&Submit=Submit" --batch db -dump

Figure 53 shows the medium security level parameters.

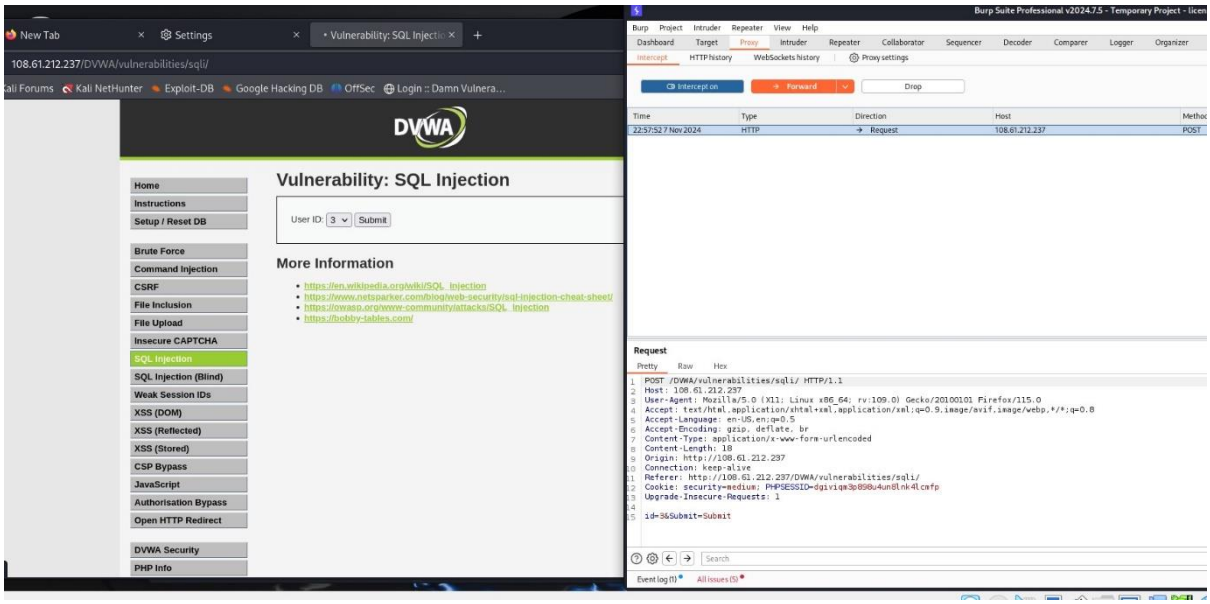


Figure 53 - Medium security level parameters

Figure 54 shows the high security level parameters.

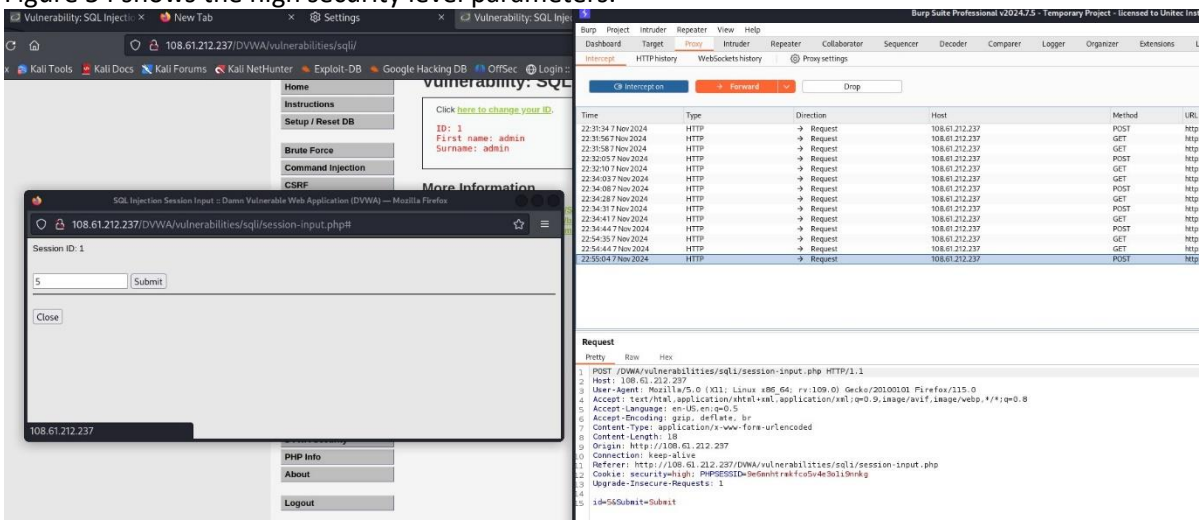


Figure 54 - High security level parameters

12.5 Hydra Tool Testing

Brute Force testing of the DVWA website using the Hydra tool

Prerequisites:

1. Hydra tool with rockyou.txt file preinstall in Kali Linux
2. DVWA website access login
3. Wifi access login

Steps use:

1. Set up the target website and launch the hydra tool in the Kali terminal.
2. Target **Login**: Admin **Password**: Admin

Figure 55 shows that the tester provides some wrong credentials to check system credentials and security.

Figure 55 - Wrong Password credential

3. Configure and use command

Type: hydra -l admin -P /usr/share/wordlists/rockyou.txt 108.61.212.237 http-get-form "/DVWA/vulnerabilities/brute/index.php: userfield=USER:passwordfield=PASS -s80

Figure 56 shows the command used in the testing to brute force the system admin password credential.

Figure 56 - Hydra target command

Figure 57 shows the successful attack and time it was started and it was finished.

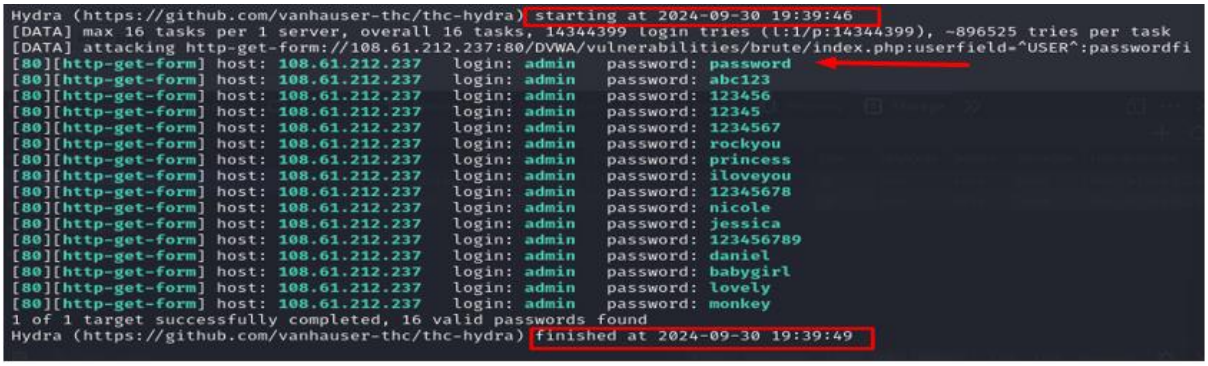


Figure 57 - Hydra successful scan

12.6 Nikto Tool Testing

Prerequisites

1. http://108.61.212.237/DVWA/login.php

2. Launch Nikto in Kali Terminal

3. Wifi login access

Testing Steps:

1. Set up

Action: Login to http://108.61.212.237/DVWA/login.php

Type: nikto -h scanme, nmap.org

2. Find port scan

Type: Nmap 108.61.212.237 -oG iplist.txt

Type: cat iplist.txt |grep “Ports: 80/open”

Figure 58 shows the command to use to get open port in the system.

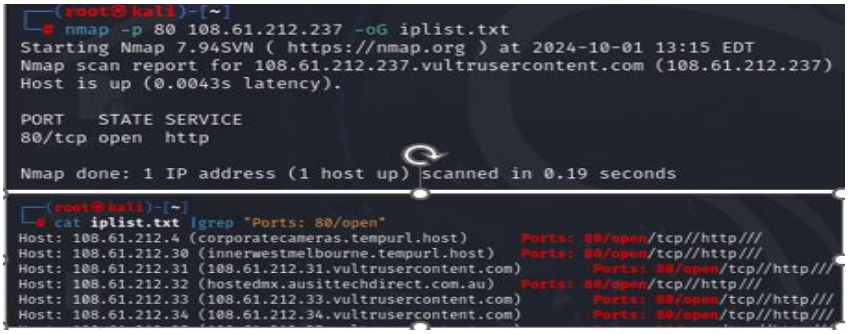


Figure 58 - Port scan results through Nmap

3. Create a script file using nano as seen below

Figure 59 command used to create script that will be use in the scanning.

Figure 59 - script of target website:108.61.212.237, script filename: iplist2.txt

4. Initial Target Scan

Type Command: nikito -h iplist2.txt -o nikto_result.csv -Format csv

Figure 60 shows information about the target IP, Hostname, Port and time and date of the scan.

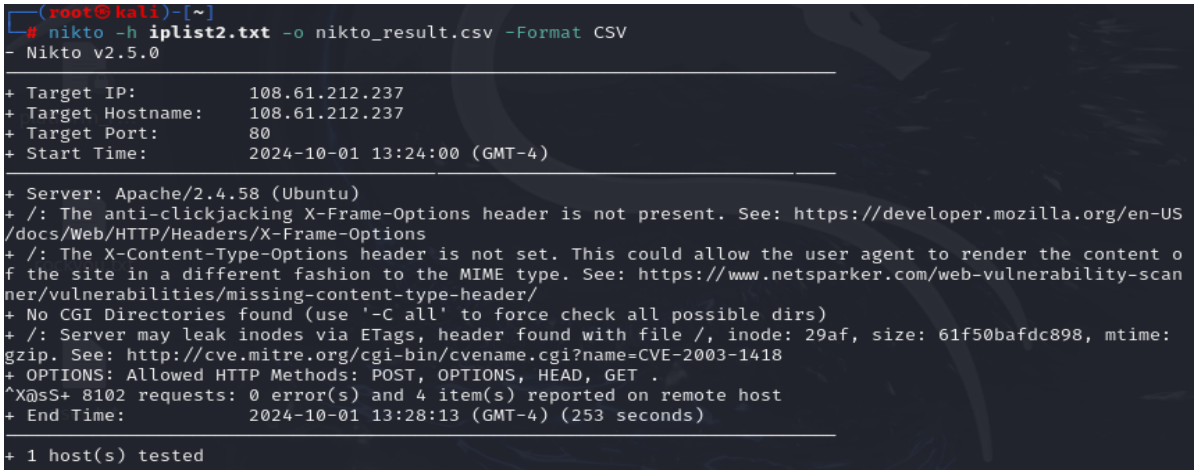


Figure 60 - Nikto scan result of website 108.61.212.237 in CSV format

5. Report and Review

Figure 61 shows the scanned results from the target web application, and the report was formatted in CVS.

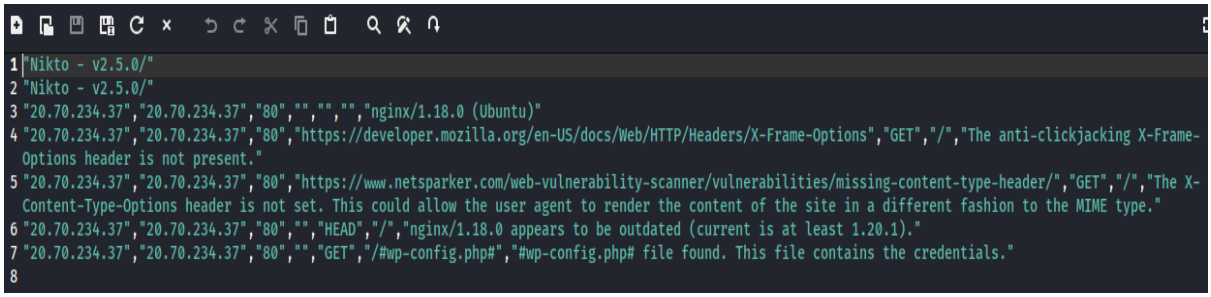


Figure 61 - iplist2.txt folder where scan report saves in CVS format

12.7 ZAP Tool Testing

To perform Testing using the ZAP tool, the steps as follows below:

Prerequisites:

1. DVWA website access login: https
2. OWASP Zap tool launch and rockyou.txt file implemented in Kali Linux/Window
3. Wifi Access:

Steps use:

1. Set up Launch Zap and load the target website to log in with the wrong credential below
2. Target Inputs: username: **admin** password: **admin**

Figure 62 shows the location of captured information from DVWA.

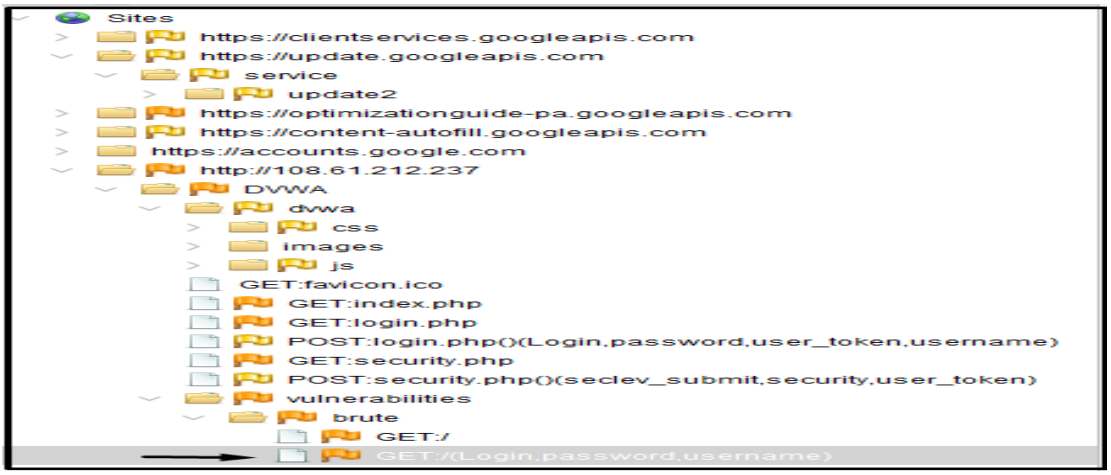


Figure 62 - Screenshot capture of DVWA Login monitoring in Zap tool

3. Launch Brute Force attack as follows: a) In the Sites tab, find the login URL b) Right-click and select **Attack>Fuzze**r function

Figure 63 shows the Manual Execution of a brute-force attack

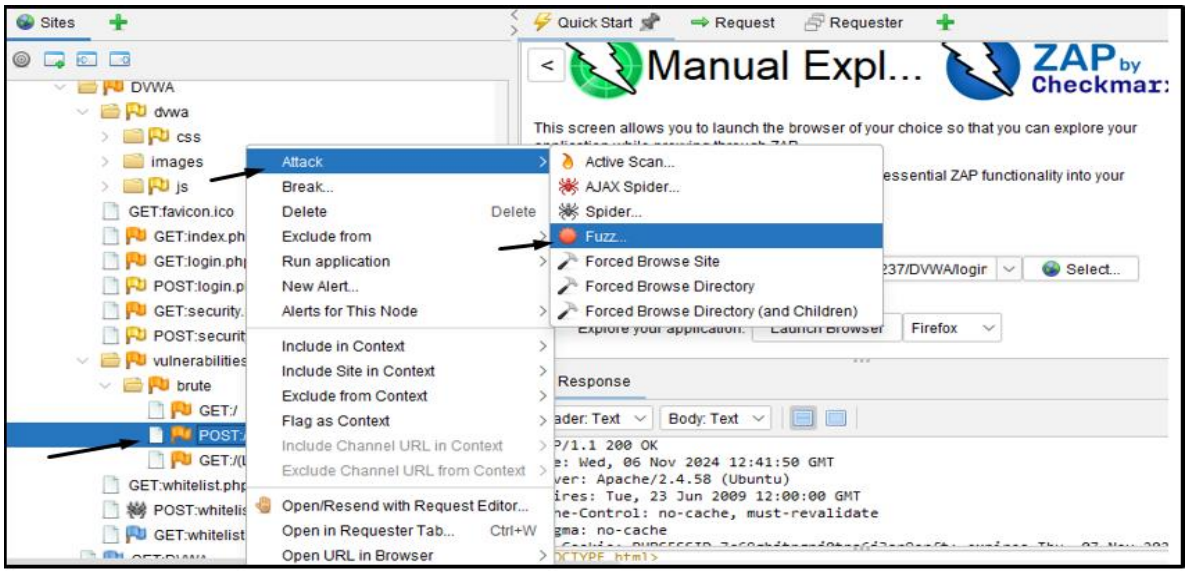


Figure 63 - Manual Execution of Brute Force Attack using Fuzzer

4. Highlight 1) username and password parameter. 2) Add to the Fuzz Location 3) Add Payload

Figure 64 shows that Zap uses a Fuzzer technique to brute force the system.

Figure 64 - Configure the brute force attack using Fuzzer

Initial scan

5. Steps: a) Browse to rockyou.txt. file. b) Select Add Payload. c)Select Start Fuzzer

Figure 65 shows where the location of the text file used for brute force attack.

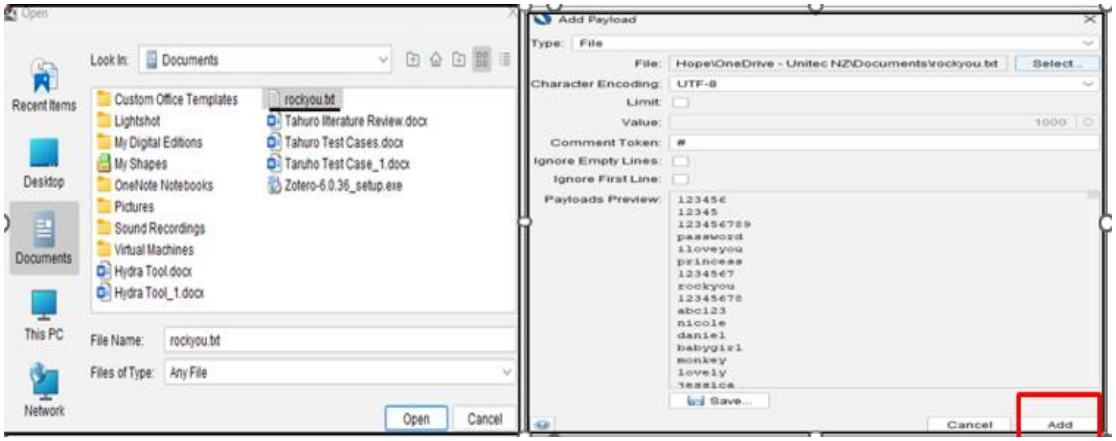


Figure 65 - Add rockyou.txt file to add payload for the initial attack

Figure 66 shows that after everything is configured then, click “Start Fuzzer” to proceed with the attack.

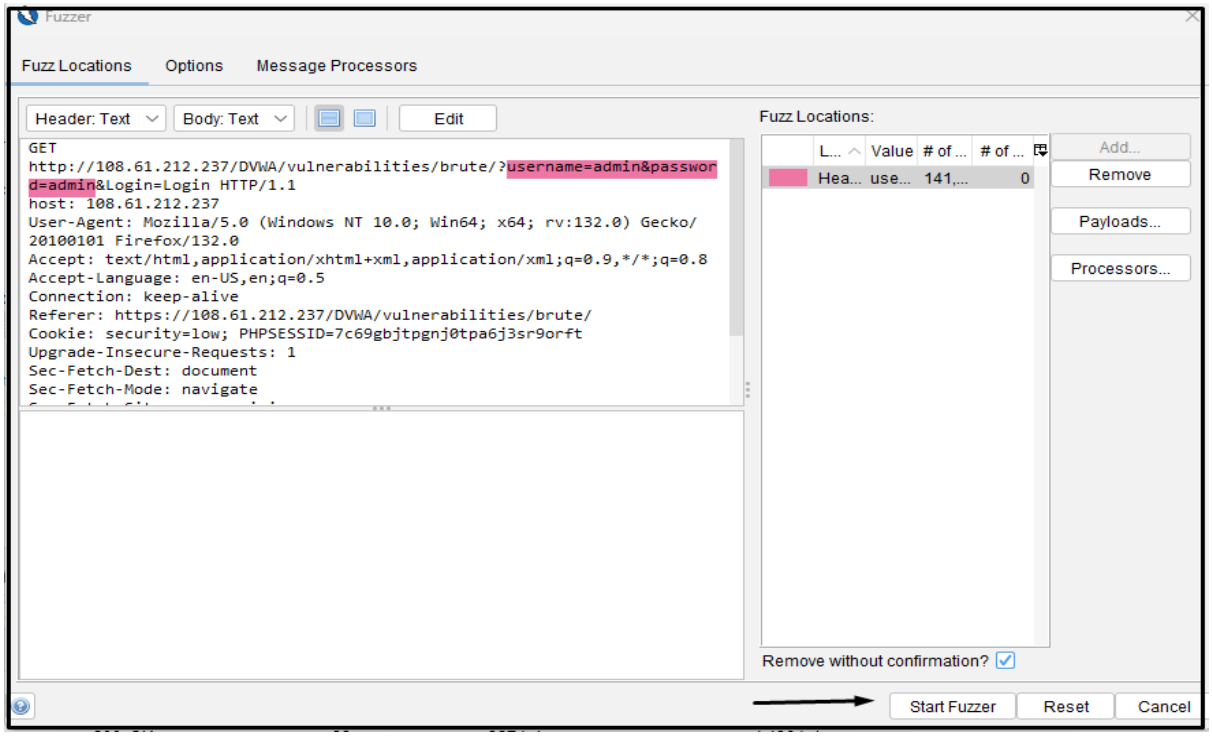
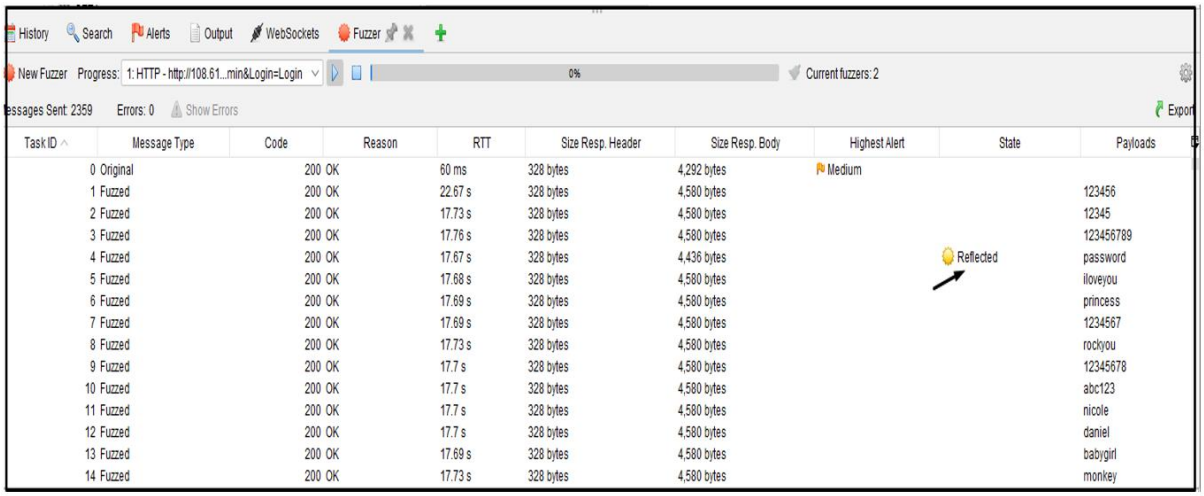


Figure 66 - Start Fuzzer Attack

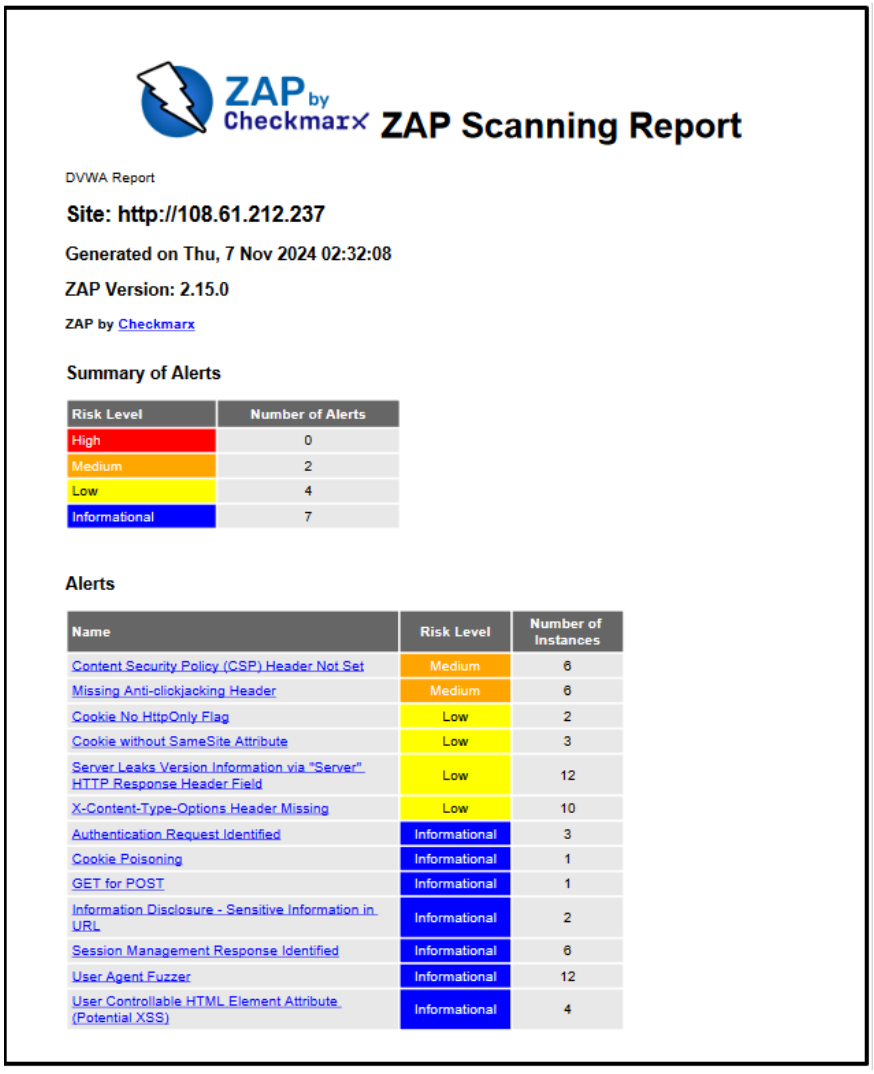
Figure 67 shows the successful brute force of the web application using the Zap tool.



Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
0	Original	200	OK	60 ms	328 bytes	4,292 bytes	Medium		
1	Fuzzed	200	OK	22.67 s	328 bytes	4,580 bytes			123456
2	Fuzzed	200	OK	17.73 s	328 bytes	4,580 bytes			12345
3	Fuzzed	200	OK	17.76 s	328 bytes	4,580 bytes			123456789
4	Fuzzed	200	OK	17.67 s	328 bytes	4,436 bytes		Reflected	password
5	Fuzzed	200	OK	17.68 s	328 bytes	4,580 bytes			iloveyou
6	Fuzzed	200	OK	17.69 s	328 bytes	4,580 bytes			princess
7	Fuzzed	200	OK	17.69 s	328 bytes	4,580 bytes			1234567
8	Fuzzed	200	OK	17.73 s	328 bytes	4,580 bytes			rockyou
9	Fuzzed	200	OK	17.7 s	328 bytes	4,580 bytes			12345678
10	Fuzzed	200	OK	17.7 s	328 bytes	4,580 bytes			abc123
11	Fuzzed	200	OK	17.7 s	328 bytes	4,580 bytes			nicole
12	Fuzzed	200	OK	17.7 s	328 bytes	4,580 bytes			daniel
13	Fuzzed	200	OK	17.69 s	328 bytes	4,580 bytes			babygirl
14	Fuzzed	200	OK	17.73 s	328 bytes	4,580 bytes			monkey

Figure 67 - Brute Force Scan Result Success

Figure 68 shows the scanning report of the Zap tool from the target web application including alerts and version of the tool during the testing.



ZAP by Checkmarx **ZAP Scanning Report**

DVWA Report

Site: http://108.61.212.237

Generated on Thu, 7 Nov 2024 02:32:08

ZAP Version: 2.15.0

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	4
Informational	7

Alerts

Name	Risk Level	Number of Instances
Content Security Policy (CSP) Header Not Set	Medium	6
Missing Anti-clickjacking Header	Medium	6
Cookie No HttpOnly Flag	Low	2
Cookie without SameSite Attribute	Low	3
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	12
X-Content-Type-Options Header Missing	Low	10
Authentication Request Identified	Informational	3
Cookie Poisoning	Informational	1
GET for POST	Informational	1
Information Disclosure - Sensitive Information in URL	Informational	2
Session Management Response Identified	Informational	6
User Agent Fuzzer	Informational	12
User Controllable HTML Element Attribute (Potential XSS)	Informational	4

Figure 68 - Zap Test Result of DVWA website

13.0 Results & Evaluation

13.1 Burp Suite Professional Tool Results

Burp Suite Professional is an advanced penetration testing tool and web vulnerability scanner version of a Burp Suite. IT security professionals use it to identify and exploit security weaknesses in web applications. This version offers extra features for automated scanning and a toolkit for manual testing. Additionally, it offers customizable scan configuration and a powerful intruder for brute-force attacks and detailed reporting.

Table 5 provides detailed results and information from the Burp Suite Professional tool testing.

Test Case Type:	**Performance Criteria Evaluation**	**Test Case Version:**	BurpSuite_01
Test Environment:	Virtual Testing using Kali Operating System in VMWare Professional Machine	**Test Input:**	**Target Website:** IP: 108.61.212.237 http://108.61.212.237/DVWA
Tester Name:	**Russell**	**Date of Testing:**	3/10/2024
Criteria Type	**Tool Name: Burp Suite Professional**	**Test Comments**	
Tool Type:	Paid	This is license under PortSwigger Company	
Version No:	Used version for testing 2024.6.6 New Version 2024.8.3	Latest Version automatically pop up when the app is opened. The new update is also available in the website. New version requires authorization from PortSwigger for our trial version.	
Latest Update:	October 7, 2024	Frequently updated. Monthly or Bi-monthly	
Cost Effective:	Licensed $449 USD	Need to manually Download and install in Kali for testing.	
Integration:	**High**	Can implement with other web application penetration testing tools.	
Ease of Use:	**Moderate**	Comprehensive and needs tutorials or manual to use the application.	
Accuracy:	**High**	Identified web vulnerabilities thru brute force attack in web application DVWA.	

Performance/Speed:	**High**	Started 11:20:21 Finished 11:20:32	
Usability:	**Moderate**	Installation in Kali is done using a command after downloading .sh installer.	
Scalability:	**High**	Burp Suite Professional can be install with different OS like Windows, macOS and Linux	
Reporting:	**High**	Detailed reporting sample from PortSwigger website.	
Tool Strengths		**Tool Weaknesses**	
• Comprehensive toolset • Automated scanning • Customizability • User-Friendly interface • Real-time result		• Cost for the tool is expensive • Resource Intensive that cause low end computer to get issues • With too much features it requires lot of time to study and learn all. • Limited support for non-limited web application. • A possible false positive result and need other verification tools to get the right findings.	

Table 5 - Burp Suite Professional Tool Results

3.2 Burp Suite Community Tool Results

Burp Suite Community Edition

Burp Suite Community Edition is a free version of the penetration testing tool. This is primarily used by beginners and individuals doing basic web security testing. This tool includes core tools like intercepting proxy, repeater, and manual testing features. There is a lack of advanced features, such as automated scanning, extension tools, and customizable options only offered to the professional version, limiting the tool's scope.

Table 6 provides detailed results and information from the Burp Suite Community Edition tool testing.

Test Case Type:	**Performance Criteria Evaluation**	**Test Case Version:**	BurpSuite_02
Test Environment:	Virtual Testing using Kali Operating System in VMWare Professional Machine	**Test Input:**	**Target Website:** IP: 108.61.212.237 http://108.61.212.237/DVWA
Tester Name:	**Russell**	**Date of Testing:**	2/10/2024
Criteria Type	**Tool Name: Burp Suite Community Edition**	**Test Comments**	**Artifacts**
Tool Type:	Free Version	This is free download on Portswigger website https://portswigger.net/burp/communitydownload	
Version No:	Latest Version: 2024.8.4	Latest Version redownload from the website	
Latest Update:	October 8,2024	Frequently updated, monthly of bi-monthly	
Cost Effective:	**Free**	Download free on Portswigger Website	
Integration:	**High**	Can implement with other web application penetration testing tools	
Ease of Use:	**Moderate**	Need to follow reliable tutorials and steps to use it.	
Accuracy:	**High**	Identified web vulnerabilities thru brute force attack in web application DVWA.	
Performance/Speed:	**High**	Started: 12:16:00 Finished: 14:33:39	
Usability:	**High**	Burp Suite Professional can be install with different OS like Windows, macOS and Linux	
Scalability:	**High**	Used by beginners and individual for basic test of web application vulnerabilities.	
Reporting:	**Moderate**	Limited compared to Professional version	
Tool Strengths		**Tool Weaknesses**	
• Free and Accessible • Comprehensive Manual testing capabilities • Proxy for Intercepting traffic • Intruder for brute force • Support for basic web vulnerabilities • Platform for multiple OS (Windows, macOS, Linux)		• No Automated scanning • Limited Intruder functionality • No extension support • No save and load functionality • No report generation • Manual testing only • No collaborator client • Limited project and task management	

Table 6 - Burp Suite Community Tool Results

Table 7 shows the test results and during each test. Each test was manually timed as the tool does not have a start and finish timer.

Burp Suite Professional	Test Duration and Result
Low Security	Successful after 11 secs
Medium Security	33minutes and 35 secs
High Security	Unsuccessful after 1 hour and 25 minutes and 33 seconds
Burp Suite Community	
Low Security	Successful after 2 hours and 15 mins
Medium Security	Successful after 3 hours and 33 minutes
High Security	Unsuccessful after 1 hour and 25 minutes and 33 seconds

Table 7 – Both version of Burp Suite Results

13.3 Wfuzz Tool Results

Table 8 provides detailed results and information from the Wfuzz tool testing.

Test Case Type:	**Performance Criteria Evaluation**	**Test Case Version:**	1
Test Environment:	Virtual Testing using Kali Operating System in VMWare Fusion software	**Test Input:**	**Target Website:** IP: 108.61.212.237 http://108.61.212.237/DVWA
Tester Name:	**Russell**	**Date of Testing:**	10/10/2024
Criteria Type	**Tool Name: Webfuzz**	**Test Comments**	**Artifacts**
Tool Type:	Open source	Already Installed in Kali Linux Operating system	
Version No:	Version 3.1.0	Command in the terminal to check the version is wfuzz --version	
Latest Update:	August 2024	https://www.edge-security.com/wfuzz.php	
Cost Effective:	Free	Package software application and brute-forcing tool	
Integration:	**High or moderate**	Can be integrated with another penetration testing tools.	
Ease of Use:	**High**	Need to get proper command to run and collect data from the application	
Accuracy:	**High**	Data was collected from providing proper command in the terminal	
Performance/Speed:	**Moderate**	15:01	

		15:14 With multiple crashing the sytem.	
Usability:	**Moderate**	Only available in Kali Linux.	
Scalability:	**High**	Need to know commands in Linux terminal to use this tool.	
Reporting:	**Moderate**	Collection of data by screenshot in the terminal to get the password cracked. Parameter length of the correct password is different from other password lists.	
Tool Strengths		**Tool Weaknesses**	
• Multiple Injection points capability with multiple dictionaries • Recursion (When doing directory bruteforce) • Post, headers and authentication data brute forcing • Output to HTML • Colored output • Hide results by returning code, word numbers, line numbers, and regex. • Cookies fuzzing • Multithreading • Proxy support • SOCK support • Time delays between requests • Authentication support (NTLM, Basic) • All parameters brute forcing (POST and GET) • Multiple encoders per payload		• Challenging beginners with the proper command line • No GUI • Limited in reporting • No SSL certificate handling • Slower than other alternatives. • Multiple crashes when testing the tools	

Table 8 - Wfuzz Tool Results

Table 9 shows the test results and during each test. Each test was manually timed as the tool does not have a start and finish timer.

Wfuzz	**Test Duration and Result**
Low Security	Successful after 14 minutes, tested a few times with few system crashes.
Medium Security	Successful after 20 minutes, tested a few times with multiple crashes in the system.
High Security	Unsuccessful after 30 minutes, no filtering of parameters.

Table 9 - WFuzz test result

13.4 SQL Map Tool Results

Table 10 provides detailed results and information from the SQL Map tool testing.

Test Case Type:	**Performance Criteria Evaluation**	**Test Case Version:**	
Test Environment:	Live Test using Windows Desktop and SQL Map Application	**Test Input:**	**Target Website**: 108.61.212.237 (DVWA)
Tester Name:	**Tanveer**	**Date of Testing:**	20/10/2024
Criteria Type	**Tool Name: SQL Map**	**Test Comments**	
Tool Type:	**Free**	Open-Source tool	
Version No:	Test version: 5.0.12	This was the latest update available at the time of testing	
Latest Update:	N/A	N/A	
Accuracy:	**High**	Highly accurate for common vulnerabilities and manual verification	
Integration:	**High**	With multiple databases	
Ease of Use:	**Medium**	Available in Linux – easy to use	
Cost Effective:	**Free**	Free to Linux	
Performance/Speed	**Medium**	Success attempt 5.5 minutes	
Usability:	**Medium**	Operation through command	
Scalability:	**Medium**	Medium adaptable	
Reporting:	**High**	Output Details have many information	
Tool Strengths		**Tool Weaknesses**	
• Automation • Multiple database support • Free and Open-source		• Limited customisation • Detection risk	

Table 10 - SQL Map Tool Results

For SQL Map, the most important part is getting the HTTP post parameters after injecting them. After the parameter is received, the SQL Map tool is able to get information with different security levels at the same time.

Table 11 compares different security level results of SQL Map Test.

SQL Map Tool – Security Level	**Test Duration and Result**
Low Security	Successful after 5.5 minutes
Medium Security	Successful after 8.5 minutes
High Security	Successful after 12 minutes

Table 11 - SQL Map Tool security level comparison

13.5 Hydra Tool Results

Tester Experience – The Hydra scan result was noted as the fastest to complete; however, multi-testing is to be considered and requires manual comparison of result if not false positive.

Table 12 provides detailed results and information from the Hydra tool testing.

Test Case Type:	**Performance Criteria Evaluation**	**Test Case Version:**		Hydra_01
Date of Testing:	1/10/2024	**Test Input:**	Target Website: 108.61.212.237 Tool use: Hydra	
Tester Name:	**Hope**	**Test Environment:**	VMWare Kali Machine	
Criteria Type	**Tool Name: HYDRA**	**Test Comments**		**Artifacts**
Tool Type:	Open-Source Tool	This is Licensed AGPLv3		
Version No:	Hydra 9.5	Latest Version found in GitHub		
Latest Update:	Jun 12, 2023	Frequently updated by the Developer		
Accuracy:	**High**	Identified valid credentials without false positive		
Integration:	**High**	Automate can brute-force, with Nmap, to find the open port -s80		
Ease of Use:	**High**	Requires understanding of command syntax		
Cost Effective:	**Free**	It is already installed in Kali		
Performance/Speed:	**High**	Started 2024-09-30 19:39:46 Finished 2024-09-30 19:39:49		
Usability:	**High**	Also available in the Hydra-graphical feature		
Scalability:	**Moderate**	Not natively designed for commercial or large distributed system		
Reporting:	**Moderate**	For CLI only		
Tool Strengths		**Tool Weaknesses**		
• Hydra Supports Multiple Protocols • Fast and efficient • Open Source • Integration with other tools • Supports Large Wordlists		• Limited Reporting and Output • Consumes significant CPU, Memory, and Network bandwidth • Requires careful ethical/legal consideration • Requires Manual Handling of Complex Logins		

Table 12 - Hydra Tool Results

Table 13 compares different security level results of Hydra Tool.

Performance Criteria on DVWA Website Level of Security				
Description: Testing scan of Level of Security	**Preconditions:** • VMware Kali Linux • Internet Access • DVWA website login		**Target Method:** Brute Force on weak password	
Tool Type	**Low**	**Medium**	**High**	
Hydra	**Success attempt** Start 09:05:54 Done 09:05:56	**Success** attempt Start 09:06:55 Done: 09:06:58	**Success** attempt Start 09:08:47 Finished 09:08:49	

Table 13 - Security level results of Hydra Tool

Figure 69 shows medium level results for Hydra tool

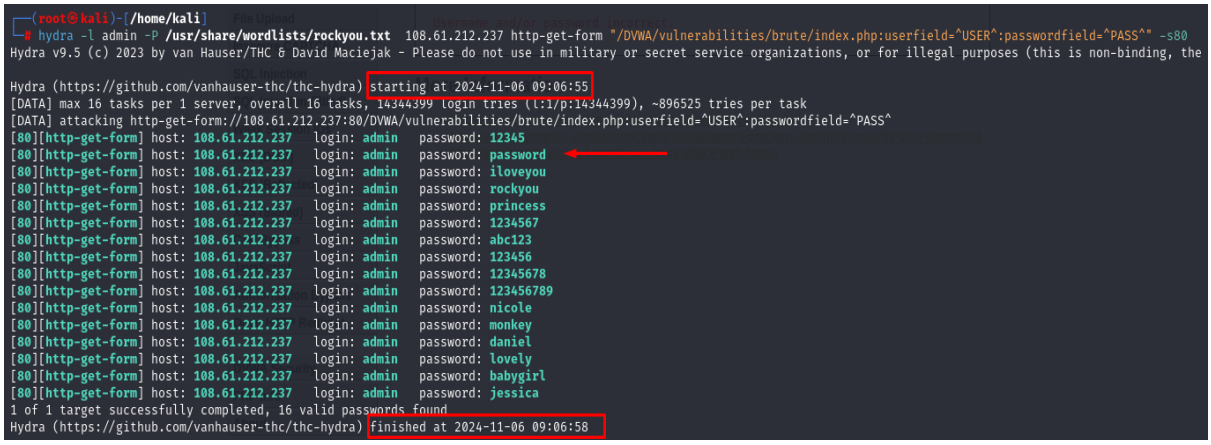


Figure 69 - Medium Level Result

Figure 70 shows medium level results for Hydra tool

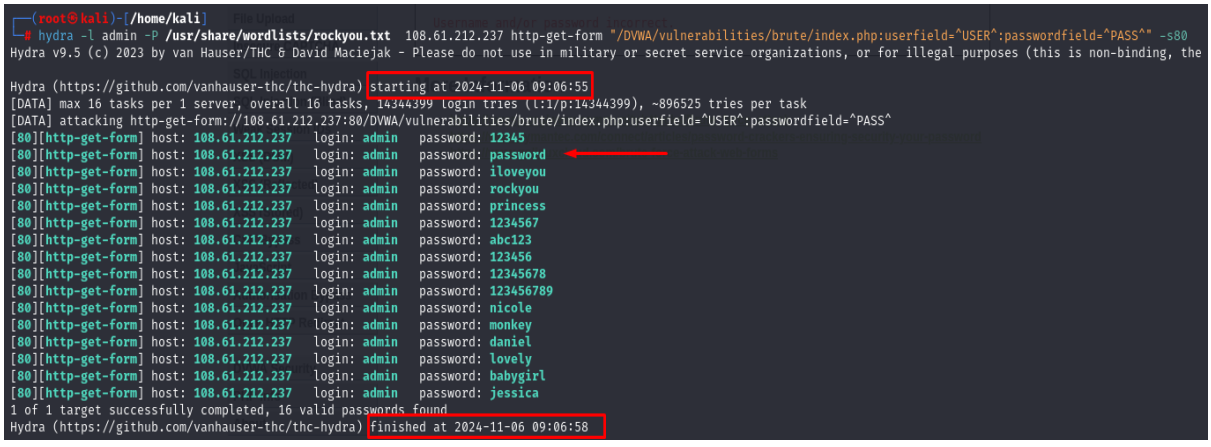


Figure 70 - High-Level Result

13.6 Nikto Tool Results

Table 14 provides detailed results and information from the Nikto tool testing.

Test Case Type:	**Performance Criteria Evaluation**	**Test Case Version:**	
Test Environment:	Virtual Testing using Kali Operating System in VMWare Professional Machine	**Test Input:**	**Target Website:** 108.61.212.237 Scanme.nmap.org Iplist.txt, iplist2.txt, nikto_result.csv
Tester Name:	**Hope**	**Date of Testing:**	2/10/2024
Criteria Type	**Tool Name: Nikto**	**Test Comments**	
Tool Type:	Open-Source Tool	This is licensed under the GNU copyright	
Version No:	Nikto 2.5.0	Latest Version found in GitHub	
Latest Update:	Dec 4, 2023	Not Frequently updated	
Cost Effective:	**Free**	It is already installed in the Kali	
Integration:	**High**	It can be implemented with Burp Suite, Nmap	
Ease of Use:	**High**	Requires understanding of command syntax	
Accuracy:	**High**	We identified web vulnerabilities but required manual validation of the findings.	
Performance/ Speed:	**High**	Started 13:24:00 Finished 13:28:13	
Usability:	**Moderate**	Easy to install, CLI operates only, which may be a disadvantage to beginners.	
Scalability:	**Moderate**	Not natively designed for commercial or large distributed system	
Reporting:	**High**	Available in HTML, text as well	
Tool Strengths		**Tool Weaknesses**	
• Open-Source tool • Supports various output format • Integrate with other tools, systems, and reporting platform • Fast Set up run with Windows, Linux, MacOS		• Prone to False Positives needs to verify the result manually • Can be time-consuming and slow • No Graphic User interface • Outdated Findings might miss the latest vulnerabilities	

Table 14 - Nikto Tool Results

13.7 ZAP Tool Results

Tester experience – Before the full scan was completed, high-level results in 69ms RTT success attempts faster than low-security results; however, it does not fully tell whether a password has been found.
Scan completion was long; unfortunately, the resulting output didn't record a start or finish time.

Table 15 provides detailed results and information from the Zap tool testing.

4 Fuzzed	302 Found	69 ms 337 bytes	0 bytes	password

Test Case Type:	**Performance Criteria Evaluation**	**Test Case Version:**		
Test Environment:	Live Test using Windows Desktop and Zap Application	**Test Input:**	**Target Website**: 108.61.212.237 Rockyou.txt file Low-Level Security	
Tester Name:	**Hope**	**Date of Testing:**	15/10/2024	
Criteria Type	**Tool Name: ZAP**	**Test Comments**		
Tool Type:	**Free**	Open-Source tool		
Version No:	Test version: 2.15.0	This was the latest update available at the time of testing		
Latest Update:	Oct 1,2024	Introduced new features support for third-party		
Accuracy:	**High**	Highly accurate for common vulnerabilities and manual verification		
Integration:	**High**	Has a GitLab CI and Jenkins plugin		
Ease of Use:	**High**	Available in Windows, macOS, and Linux. It is recommended for both beginners and professionals.		
Cost Effective:	**Free**	Free to download on WinOS, macOS and Linux		
Performance/Speed	**High**	Success attempt 4.04 secs		
Usability:	**High**	GUI is user-friendly compared to other tools		
Scalability:	**High**	Highly adaptable, allows functions add-ons, and can suit more complex testing.		
Reporting:	**High**	Customisable Reports details in low, medium, high		
Report Findings	Go to **Reports > Generate Report**	**Successful brute force attempts** return a **200 OK status**		

Tool Strengths	**Tool Weaknesses**
• Free and Open Source • Ease of use – also has Quick Start feature • Enabling automated testing • Active Community and Regular updates	• False Positives where vuln are flagged but do not exist • Performance limitations depth for advanced exploits • Consume a lot of system resource

Table 15 - ZAP Tool Results

Table 16 compares different security level results of the ZAP Tool.

Performance Criteria on DVWA Website Level of Security			
Description: Testing scan of Level of Security	**Preconditions:** • VMware Kali Linux • Internet Access • DVWA website login		**Target Method:** Brute Force on weak password
Tool Type	**Low**	**Medium**	**High**
ZAP	**Success attempt** in 78ms RTT Start 12.47 pm Completed scan 75% 12 am	**Success** attempt In 65ms RTT	**Success** attempt in 69ms RTT Start 3.30 with 1% scan complete

Table 16 - Security level results of ZAP Tool

Figure 71 shows the start Time 12:47 pm of the attack.

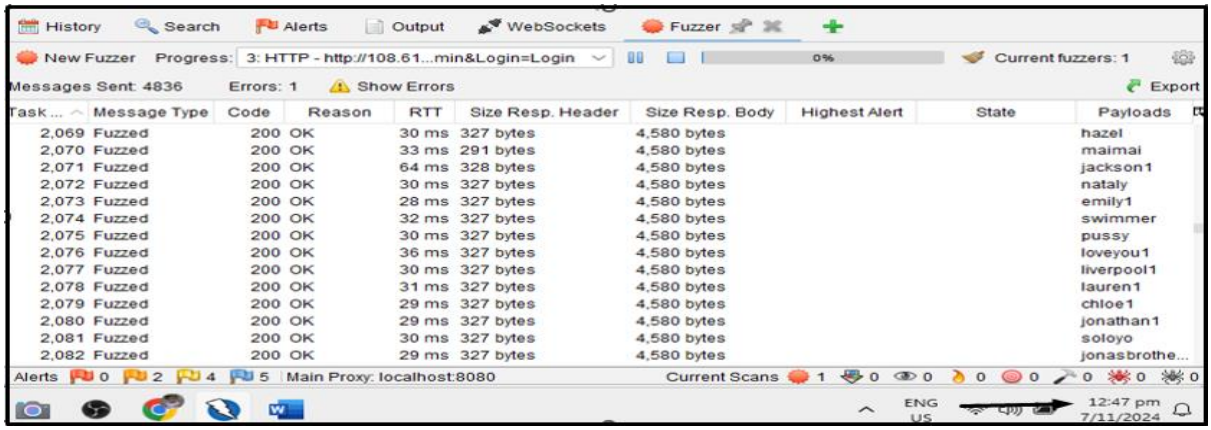


Figure 71 - Attack start Time 12:47 pm

Figure 72 shows the high-level scan of the attack.

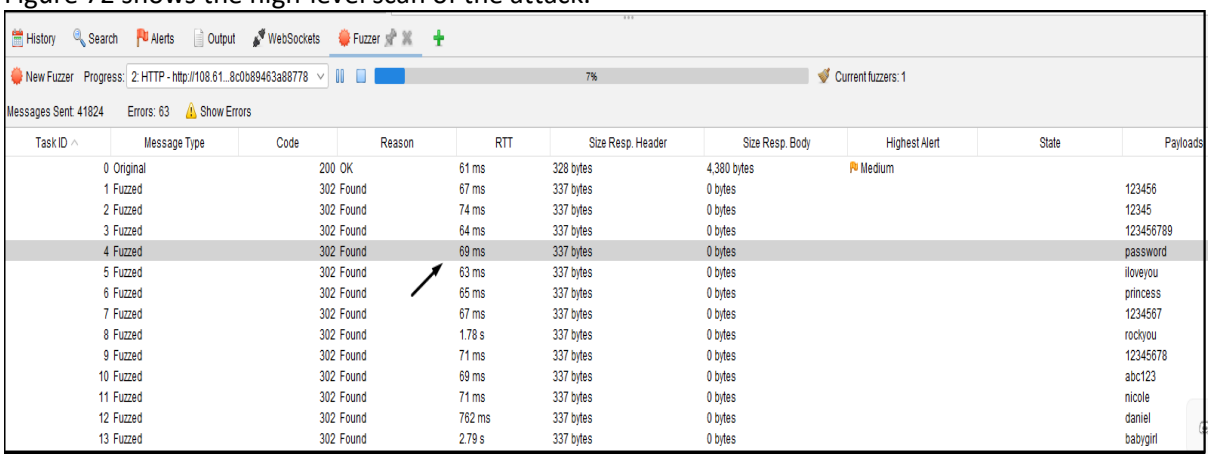


Figure 72 - High-Level scan Result

14.0 Impacts

14.1 Research Benefits for New Zealand

Our project design aims to empower New Zealand organisations in selecting appropriate testing tools, a critical step toward strengthening cybersecurity frameworks for individuals and small to medium-sized enterprises [15] [16], which account for 97% of all businesses and contribute approximately 28% of the country's GDP to defend against the growing risks of cyber threats [17].

As 43% of cyberattacks target SMEs globally, Taruho's research identifies and assesses practical, user-friendly tools to enhance penetration testing for web application vulnerabilities and support scalable best practice-aligned security [18]. By focusing on free and open-source tools, this project is designed to support a pathway for small businesses to protect digital assets affordably.

Figure 73 shows Industry statistics for Maori and Pasifika community.

Figure 73 - Digital Tech Industry Statistics for Maori and Pasifika according to Digital Skill Aotearoa

14.2 Research benefits for communities such as Māori and Pacific

There is a more vital need for education and upskilling in cybersecurity. Statistics show that Maori comprise 5% of the IT workforce, while Pacific people account for 4.2%. [19]. Furthermore, our research strives to build and be incredibly impactful for Maori and Pacific communities, who face increased risks due to limited access to cybersecurity resources and are also part of the SME innovations. [20]

14.3 Summary of Impacts

Beyond technology, this initiative addresses New Zealand's social landscape by working to bridge the digital divide, promoting a more inclusive, skilled, and resilient digital ecosystem. Through this approach, we strengthen cybersecurity frameworks for SMEs and individual startups and contribute to developing a diverse talent pipeline essential for New Zealand’s broader economic and social resilience. Please refer to the “Impact” section of the appendix for more charts.

15.0 Change/Risk Management

15.1 Change Management Plan

The purpose of a Change Management plan is to manage and control any adjustments that might occur in our project scope, timeline, or resources and to maintain the aim of the project objectives. We have established a structured Change Request Process if we need to change something. This begins with Initiation, which is done when team members or stakeholders submit a Change Request Form and explain the detailed reason for the change and a proposal for the new scope. It is then evaluated during the evaluation phase, during which the project manager will assess the impact of the proposal change's scope, timeline, budget, and resources. If this needs approval, it will be presented to the project sponsor, Andrew David, and key stakeholders for review and authorisation.

When the changes are approved, implementation can take place, which involves the adjustment of the project plan, and documents should also be updated with the new scope. Lastly, the changes should be communicated to all the team members, and a meeting should be held to discuss the changes in the project's scope.

The plan should monitor some key areas: first is the adjustment of the project scope, which could involve adding or removing tools from the analysis; the timeline for testing might change in the work breakdown structure and may require some modification in our schedule due to unexpected results and technical problems; resource allocation if there are some delays might also change the schedule of our testing and analysis, there might be an unforeseen technical requirement within the environment of the testing. With a prepared and structured approach, we ensure that if there are changes, we can manage them efficiently and still produce a successful project outcome.

15.2 Risk Management Plan

Our project has identified some key risks that might impact the success that we want to aim. Tool Evaluation Inaccuracy might be a risk due to the inconsistency of a tool’s performance with different environments, which can cause inconsistent evaluation results. This risk is considered possible and has a major impact on our project. We created a standard testing environment to mitigate this problem so that we have consistent results across different devices of each team member. The project team is responsible for this risk and remains open.

If one of the team members is injured, a risk management plan prioritises the continuity of the project and the unwell member's well-being. The injured team member's responsibility will be temporarily assigned and divided to another member to ensure that the project is still on time with the task. To manage the workload, we plan to distribute the task to each member's skills and availability, adjust the deadline if needed, and inform the project manager about the situation. If the injured person returns soon, they will be gradually given a task as their responsibility. This approach minimises delays in team projects and provides good team morale and project momentum.

Furthermore, technical issues with the tools or the test environment are another risk. There might be compatibility or configuration issues that might occur in testing tools or the DVWA setup, which could delay testing some tools. This risk has a moderate impact and will be addressed by conducting preliminary testing and troubleshooting; project stakeholder Justin Ng is ready to help us with the web application and give us suggestions for tool troubleshooting.

Another risk is data loss or corruption. Although it might not occur, the impact would be high if the testing data and configuration files are lost, as we need to redo it again. To mitigate this, we suggested that each team member have backup files and version control for testing and results. Scope creep is also a possible risk; this means that the project scope needs to be expanded or needs to have additional tool comparisons or metrics, which leads to a high impact on the project. To manage this risk, we created a change management plan which the project manager can check for approval.

Lastly, the risk of insufficient testing time will likely happen, given the challenges of tutorials from different resources and what we should use for the testing. This has a major impact on the project, and to mitigate this, we created a well-defined timeline for testing and distributed tools to test with each team member and share the results with everyone. Please see the appendix for the Project Risk Register, Issue Register and Risk Matrix

16.0 Conclusion and Lessons Learned

16.1 Project Summary and Significant Project Outcome

This penetration testing followed a methodical way to find out the vulnerability of web applications. Proper testing methods were used to identify in a safe environment. This project aimed to test Damn Vulnerable Web Applications (DVWA) with the help of well-known penetration testing tools such as Burp Suite, SQL Map, WFuzz, Hydra and OWASP ZAP. Each tool was assessed based on multiple criteria based on different security levels, and the source code of each security level is carefully analysed for future reference. When it comes to finding vulnerabilities such as brute force and SQL injection, this research project illustrates the actual importance of choosing the right penetration testing tools for web application testing. Based on the findings and comparison of the testing tools, it is evident that Burp Suite will be highly recommended among the tools. As this research project is properly documented, it is ensured that these testing procedures are followed according to the framework.

16.2 Evaluation of current and future Projects

Continuously new threats are emerging, and it is really important to improve the efficiency of the tools after evaluating the performance of this project. In future, mobile applications and other platforms can be tested to extend this project work to find more vulnerable aspects of the program. The implementation of cloud-based testing frameworks, such as machine learning security, can be included, which will bring more meaningful results. This project will work as proactive threat prevention for current and upcoming cybersecurity projects as it is based on complete penetration testing procedures.

17.0 References

[1] E. Gürel, "SWOT ANALYSIS: A THEORETICAL REVIEW," *J. Int. Soc. Res.*, vol. 10, pp. 994–1006, Aug. 2017, doi: 10.17719/jisr.2017.1832.

[2] M. Albahar, D. Alansari, and A. Jurcut, "An Empirical Comparison of Pen-Testing Tools for Detecting Web App Vulnerabilities," *Electronics*, vol. 11, no. 19, Art. no. 19, Jan. 2022, doi: 10.3390/electronics11192991.

[3] T. Guarda, W. Orozco, M. F. Augusto, G. Morillo, S. A. Navarrete, and F. M. Pinto, "Penetration Testing on Virtual Environments," in *Proceedings of the 4th International Conference on Information and Network Security*, Kuala Lumpur Malaysia: ACM, Dec. 2016, pp. 9–12. doi: 10.1145/3026724.3026728.

[4] B. Khan, J. Bangash, M. Tariq, N. Gul, S. Zahir, and A. Kamal, "A Comparative Model to Analyze Various Web Application Penetration Testing Tools for Different Vulnerabilities," Jul. 2023.

[5] "OWASP Top Ten | OWASP Foundation." Accessed: Nov. 05, 2024. [Online]. Available: https://owasp.org/www-project-top-ten/

[6] W. G. J. Halfond, S. R. Choudhary, and A. Orso, "Improving penetration testing through static and dynamic analysis," *Softw. Test. Verification Reliab.*, vol. 21, no. 3, pp. 195–214, Sep. 2011, doi: 10.1002/stvr.450.

[7] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, and N. Crespi, "A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions," *Appl. Sci.*, vol. 12, no. 8, p. 4077, Apr. 2022, doi: 10.3390/app12084077.

[8] J. Fonseca, M. Vieira, and H. Madeira, "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks," in *13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007)*, Melbourne, Australia: IEEE, Dec. 2007, pp. 365–372. doi: 10.1109/PRDC.2007.55.

[9] M. Albahar, D. Alansari, and A. Jurcut, "An Empirical Comparison of Pen-Testing Tools for Detecting Web App Vulnerabilities," *Electronics*, vol. 11, no. 19, p. 2991, Sep. 2022, doi: 10.3390/electronics11192991.

[10] H. M. Z. A. Shebli and B. D. Beheshti, "A study on penetration testing process and tools," in *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY: IEEE, May 2018, pp. 1–7. doi: 10.1109/LISAT.2018.8378035.

[11] S. Nagpure and S. Kurkure, "Vulnerability Assessment and Penetration Testing of Web Application," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, PUNE, India: IEEE, Aug. 2017, pp. 1–6. doi: 10.1109/ICCUBEA.2017.8463920.

[12] K. Barik, A. Abirami, S. Das, K. Konar, and A. Banerjee, "Penetration Testing Analysis with Standardized Report Generation:," presented at the 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021), Bangalore, India, 2021. doi: 10.2991/ahis.k.210913.045.

[13] sullo, *sullo/nikto*. (Nov. 06, 2024). Perl. Accessed: Nov. 06, 2024. [Online]. Available: https://github.com/sullo/nikto

[14] "The ZAP Homepage," ZAP. Accessed: Nov. 06, 2024. [Online]. Available: https://www.zaproxy.org/

[15] A. Hunt, "Check out the NZ 2024 SME Cyber Behaviour Tracker," IT Solutions and Managed Services. Accessed: Nov. 02, 2024. [Online]. Available: https://kinetics.co.nz/check-out-the-nz-2024-sme-cyber-behaviour-tracker/

[16] "SMEs urged to act now as cyber threats surge in New Zealand." Accessed: Nov. 02, 2024. [Online]. Available: https://www.insurancebusinessmag.com/nz/news/cyber/smes-urged-to-act-now-as-cyber-threats-surge-in-new-zealand-505416.aspx

[17] N. Z. M. of F. A. and Trade, “Supporting SMEs,” New Zealand Ministry of Foreign Affairs and Trade. Accessed: Nov. 02, 2024. [Online]. Available: https://www.mfat.govt.nz/en/trade/free-trade-agreements/free-trade-agreements-in-force/cptpp/supporting-smes
[18] “OWASP Web Security Testing Guide | OWASP Foundation.” Accessed: Nov. 02, 2024. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/
[19] “Digital Skills for Tomorrow, Today.” [Online]. Available: https://nztech.org.nz/wp-content/uploads/sites/8/2023/08/NZTech-Digital-Skills-Report_final.pdf
[20] J. Hendy, “Digital inclusion and wellbeing in New Zealand”.

18.0 Appendices

18.1 Signed Individual Contribution Statement

Project Title: Comparative Analysis of Penetration Testing Tools in Assessing Web Application Vulnerabilities

Project Start: 22/07/2024

Project Finish:22/11/2024

Contributor: Tanveer Ahmed Osman

Role: Project Manager

Contribution Statement:

In our Capstone project, I conducted overall web application security vulnerabilities. I used industry-standard tools like Burp Suite and SQL Map. Based on different security levels, I tested the tools which stimulated real-life testing. All the testing results are also documented. I identified different areas of improvement and made recommendations for the proper tool. Also, I focused on a cost-effective solution for the businesses. I designed the project in such a way that it will be helpful for cybersecurity professionals and contribute to society.

Tanveer Ahmed Osman

10/11/2024

Project Title: Comparative Analysis of Penetration Testing Tools in Assessing Web Application Vulnerabilities

Project Start: 22/07/2024

Project Finish:22/11/2024

Contributor: Jun Russell Ragasa

Role: Penetration Tester

Contribution Statement:

I actively contributed to this project as a tools tester, mostly utilising both Burp Suite Professional and Community editions and assessing these tools with different levels of security in the target web application. My responsibilities included thorough documentation of implementation, testing, and finding results in my allocated tools. I supported the team by contributing a part of the document report and also conducted research on relevant papers to enhance the quality of our report. Additionally, I always maintained good communication with the stakeholder penetration tester to ensure that our project was aligned and provided guidance to our project. My role also involved parts of the project report and participated in meetings to keep track of the progress and achieve deliverables.

Jun Russell Ragasa

10/11/2024

Project Title: Comparative Analysis of Penetration Testing Tools in Assessing Web Application Vulnerabilities

Project Start: 22/07/2024

Project Finish:22/11/2024

Contributor: Hope Pose-Anesone

Role: Penetration Tester

Contribution Statement:

As a Team Member and Penetration Tester for this Project, I contributed significantly to various phases of the engagement. My primary responsibilities included:
1. Problem Statement & Requirement Analysis
2. Literature Review
3. Project Management Methodology
4. Tool usage and documentation – leverage Nikto, Hydra and OWASP ZAP to perform brute force attacks, including web application scanning. The detailed documentation I produced contributed to the final report.
5. Impact – analyze findings and its impact towards NZ Society.

Additionally, this experience not only enhanced my technical skills, but my contributions played a key role in improving the application's security posture and the project's overall success.

Hope Pose-Anesone
10/11/2024

Supervisor and Sponsor Approval

Andrew David
Project Supervisor and Sponsor
Date: 10/11/2024

18.2 Team meeting minutes

Group Meeting Minutes 1

Team Name:	Taruho		
Date of Meeting:	29/7/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Discussion about planning and penetration Project.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Penetration testing	Discussion about the steps and planning

4. Action Items		
Tasks	Assigned to	Due Date
Project Planning	Tanveer	30/7/2024

5. Next Meeting					
Date	12/7/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Discussion about the types of penetration testing.				

Group Meeting Minutes 2

Team Name:	Taruho		
Date of Meeting:	12/7/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Discussion about the types of penetration testing.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Types of Pen Test	Discussed about different types of penetration testing

4. Action Items		
Tasks	Assigned to	Due Date
Type selection	Tanveer	15/7/2024

5. Next Meeting					
Date	26/8/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Discussion about vulnerability testing and penetration testing.				

Group Meeting Minutes 3

Team Name:	Taruho		
Date of Meeting:	26/8/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Discussion about vulnerability testing and penetration testing.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Selecting method	Discussed about which method is more effective

4. Action Items		
Tasks	Assigned to	Due Date
Tool Selection	Tanveer	28/8/2024

5. Next Meeting					
Date	9/9/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Discussion about exploitation techniques.				

Group Meeting Minutes 4

Team Name:	Taruho		
Date of Meeting:	9/9/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Discussion about exploitation techniques.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
exploitation techniques	Discussed about various exploitation techniques

4. Action Items		
Tasks	Assigned to	Due Date
exploitation techniques checklist	Tanveer	11/9/2024

5. Next Meeting					
Date	23/9/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Discussion about DVWA				

Group Meeting Minutes 5

Team Name:	Taruho		
Date of Meeting:	23/9/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Discussion about DVWA

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
DVWA	Discussed the implementation and testing of DVWA

4. Action Items		
Tasks	Assigned to	Due Date
DVWA Test	Tanveer	26/9/2024

5. Next Meeting					
Date	7/10/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Discussion about penetration testing reporting				

Group Meeting Minutes 6

Team Name:	Taruho		
Date of Meeting:	7/10/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Discussion about penetration testing reporting

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Testing reporting	Discussion about penetration Testing reporting for different tools

4. Action Items		
Tasks	Assigned to	Due Date
Performance criteria	Tanveer	10/10/2024

5. Next Meeting					
Date	28/10/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Discussion about the use of tools effectively				

Group Meeting Minutes 7

Team Name:	Taruho		
Date of Meeting:	28/10/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Discussion about the use of tools effectively

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Tool efficiency	Discussed about the tool's criteria to effectively get the outcome

4. Action Items		
Tasks	Assigned to	Due Date
Tool efficiency criteria	Tanveer	30/10/2024

18.3 Meeting minutes with Project Supervisor and Sponsor

Supervision Meeting Minutes 1

Team Name:	Taruho		
Date of Meeting:	29/7/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Discussion about the Capstone project and forming a team.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:30 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:30 pm	ragasj01@myunitec.ac.nz	02108389423
Andrew David	9:30 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Brainstorming ideas	Discussed our interest in different topics and guided us through the process.

4. Action Items		
Tasks	Assigned to	Due Date
Team finalise	Tanveer	1/8/2024
Capstone project choose	Russell	1/8/2024

5. Next Meeting					
Date	1/8/2024	Time:	9:30 pm	Location:	Online - Team App
Objectives:	Finalise Team and capstone project ideas discussion				

Supervision Meeting Minutes 2

Team Name:	Taruho		
Date of Meeting:	1/8/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Hope	Location:	Online - Team App

1. Meeting Objective(s)
Discuss the Capstone project focused on implementing the Penetration testing toolkit. Address any questions or concerns regarding the steps needed for the team to progress efficiently on the project.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:30 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:30 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:30 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:30 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
1. Capstone project outcomes	The team will focus on specific tools. Research and study previous papers for more ideas.
2. Project topic	" Penetration testing on Linux Debian Servers. " this is not official.
3. Discussion of Penetration software tools	We will focus on the specific technology tools, such as Metasploit.
4. Discussion of Project Charter	This is to be submitted by Tanveen before Monday.
5. Business Requirement	Talk about what Operating System to use. Choose tools, types of attacks, and targets.

4. Action Items		
Tasks	Assigned to	Due Date
Agenda minutes meeting	Hope	4/8/24
Project Charter - Submission	Tanveer	4/8/24
Individual Timesheet	Tanver|Jun|Hope	4/8/24

5. Next Meeting					
Date	3/8/2024	Time:	9:30 pm	Location:	Online - Teams
Objectives:	Project Charter – Review and Approval by Andrew before the submission due date.				

Supervision Meeting Minutes 3

Team Name:	Taruho		
Date of Meeting:	3/8/2024	Time Duration:	1.5 Hours
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Project Charter – Review and Approval by Andrew before the submission due date.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:30 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:30 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:30 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:30 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Project Charter Review	Overall review and approval of project charter.

4. Action Items		
Tasks	Assigned to	Due Date
Submission of Documents	Tanveer	4/8/2024
Research about the chosen topic	Tanver|Jun|Hope	8/8/2024

5. Next Meeting					
Date	8/8/2024	Time:	9:30 pm	Location:	Online - Team App
Objectives:	Discuss the idea and plan of the project in detail.				

Supervision Meeting Minutes 4

Team Name:	Taruho		
Date of Meeting:	6/8/2024	Time Duration:	1 Hour
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Discussion about the Capstone project and proposal defence.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:30 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:30 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:30 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:30 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Proposal defence	Discussion about the Capstone project and proposal defence.

4. Action Items		
Tasks	Assigned to	Due Date
Proposal defence finalise	Tanveer| Russell	7/8/2024
Defence slide	Hope	7/8/2024

5. Next Meeting					
Date	8/8/2024	Time:	9:30 pm	Location:	Online - Team App
Objectives:	Feedback about the proposal defence				

Supervision Meeting Minutes 5

Team Name:	Taruho		
Date of Meeting:	8/8/2024	Time Duration:	1 Hour
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Feedback about the proposal defence

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:30 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:30 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:30 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:30 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Project proposal defence feedback	Feedback from the supervisor for presenting the proposal defence

4. Action Items		
Tasks	Assigned to	Due Date
Agenda minutes meeting	Hope	11/8/24
Project Charter - Submission	Tanveer	11/8/24
Individual Timesheet	Tanver|Jun|Hope	11/8/24

5. Next Meeting					
Date	12/8/2024	Time:	9:30 pm	Location:	Online - Teams
Objectives:	Allocation Tasks for Project Proposal				

Supervision Meeting Minutes 6

Team Name:	Taruho		
Date of Meeting:	12/8/2024	Time Duration:	1 Hour
Minutes Prepared By:	Hope	Location:	Online - Team App

1. Meeting Objective(s)
Allocation Tasks for Project Proposal

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:30 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:30 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:30 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:30 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Project proposal review	Discuss and modification of project proposal

4. Action Items		
Tasks	Assigned to	Due Date
Executive Summary, Deliverables, WBS, Gantt Chart	Tanveer	14/8/2024
Title, Business Requirements, Objectives, Scope, Proposed Solution	Jun	14/8/2024
Problem Statement, Impacts	Hope	14/8/2024

5. Next Meeting					
Date	15/8/2024	Time:	9:30 pm	Location:	Online - Team App
Objectives:	Discuss the project proposal document				

Supervision Meeting Minutes 7

Team Name:	Taruho		
Date of Meeting:	15/8/2024	Time Duration:	1 Hour
Minutes Prepared By:	Hope	Location:	Online - Team App

1. Meeting Objective(s)
Discuss the project proposal document.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:30 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:30 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:30 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:30 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Project Proposal Review	Feedback and overview of the project proposal document

4. Action Items		
Tasks	Assigned to	Due Date
Submission of Documents	Tanveer	17/8/2024
Correction and modification of document	Jun|Hope	17/8/2024

5. Next Meeting					
Date	19/8/2024	Time:	9:30 pm	Location:	Online - Team App
Objectives:	Discuss project implementation				

Supervision Meeting Minutes 8

Team Name:	Taruho		
Date of Meeting:	23/8/2024	Time Duration:	1 Hour
Minutes Prepared By:	Hope	Location:	Online - Team App

1. Meeting Objective(s)
To review and assess the progress made on project tasks. Catch up on individual and team development.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	5:30 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	5:30 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	5:30 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	5:30 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Weekly meeting with Andrew	Confirmed on Date: Thursday Time: 9:00 pm
Week 5 Project Milestone	Research Review is nearly complete, VMWare and Burp Suite are all set and complete, and DVWA setup is complete by (Professional PenTester)
Week 6 Progress Presentation	This is a 5-minute Presentation per Team

4. Action Items		
Tasks	Assigned to	Due Date
Research on VM, CVE Vulnerabilities, Pen Test Tools	All	29/08/24
Presentation Slides	All	28/08/24
Individual Weekly Timesheet	All	25/08/24

5. Next Meeting					
Date	29/8/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Research Review – Burp Suite				

Supervision Meeting Minutes 9

Team Name:	Taruho		
Date of Meeting:	29/8/2024	Time Duration:	1 Hour
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Research Review – Burp Suite

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:00 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Discussion on Research Approach	Focus on how effective are those pentesting tools on VMware environment

4. Action Items		
Tasks	Assigned to	Due Date
Research on VM, CVE Vulnerabilities, Pen Test Tools	All	31/08/24
Presentation Slides	All	31/08/24
Individual Weekly Timesheet	All	31/08/24

5. Next Meeting					
Date	5/9/2024	Time:	9:00 pm	Location:	Online - Teams
Objectives:	Burp Suite – Penetration testing methodology				

Supervision Meeting Minutes 10

Team Name:	Taruho		
Date of Meeting:	5/9/2024	Time Duration:	1 Hour
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
To review and assess the progress made on project tasks. Catch up on individual and team development.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:00 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Weekly meeting with Andrew	Confirmed on Date: Thursday Time: 9:00 pm
Discussion about Outputs	Discussed and demonstrated various tools implementation

4. Action Items		
Tasks	Assigned to	Due Date
Research on VM, CVE Vulnerabilities, Pen Test Tools	All	7/09/24
Presentation Slides	All	7/09/24
Individual Weekly Timesheet	All	7/09/24

5. Next Meeting					
Date	29/8/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Research Review – DVWA and other penetration tools.				

Supervision Meeting Minutes 11

Team Name:	Taruho		
Date of Meeting:	12/9/2024	Time Duration:	1 Hour
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Research Review – DVWA and other penetration tools.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:00 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Discussion on Research Reports	Discussion about the structure of the report
Weekly meeting with Andrew	Confirmed on Date: Thursday Time: 9:00 pm

4. Action Items		
Tasks	Assigned to	Due Date
Research on VM, CVE Vulnerabilities, Pen Test Tools	All	14/09/24
Presentation Slides	All	14/09/24
Individual Weekly Timesheet	All	14/09/24

5. Next Meeting					
Date	19/9/2024	Time:	9:00 pm	Location:	Online - Teams
Objectives:	Development of the project and Report Writing				

Supervision Meeting Minutes 12

Team Name:	Taruho		
Date of Meeting:	19/9/2024	Time Duration:	1 Hour
Minutes Prepared By:	Hope	Location:	Online - Team App

1. Meeting Objective(s)
Presentation on Impact towards the Project. Focus on Impact, Problem Statement and Problem Solution.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:00 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Weekly meeting with Andrew	Confirmed on Date: Thursday Time: 9:00 pm
Week 9 Project Presentation	Discussion on Presentation Preparation before it dues in Monay
Week 9 Testing Tasks	Set up meeting with Justin - Discussion on Pentesting Tools

4. Action Items		
Tasks	Assigned to	Due Date
Russell to set up a meeting with Justin	Russell	18/09/24
Pentesting Task – start learning SQL Map, Nikto,	All	26/09/24
Set up Meeting with Masoud on Friday	Tanveer	20/09/24
Individual Weekly Timesheet	All	22/09/24
Hope will do the Impact	Hope	21/09/24
Russell will do the Problem Statement	Russell	21/09/24
Tanveer will do the Intro,Objective	Tanveer	21/09/24

5. Next Meeting					
Date	26/9/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Test Cases, Analysis and Status Report				

Supervision Meeting Minutes 13

Team Name:	Taruho		
Date of Meeting:	26/9/2024	Time Duration:	1 Hour
Minutes Prepared By:	Hope	Location:	Online - Team App

1. Meeting Objective(s)
Feedback Review on Impact Presentation, progress plan towards Final Report Preparation.

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:00 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Weekly meeting with Andrew	Confirmed on Date: Thursday Time: 9:00 pm
Discussion with Andrew	To set up a meeting with Norah (Librarian for Research tips).
Week 10 Project Progress	
Team Discussion	Design Performance Criteria for evaluation tools
	Research Paper to use as the project guidelines Masouds Website can be used (with anon disclosure) for the project

4. Action Items		
Tasks	Assigned to	Due Date
Set up a Meeting with Norah	Tanveer	
Pentesting Task – start learning SQL Map, Nikto,	All	26/09/24
Week 10 Timesheet Submission	All	30/09/24

Supervisor Meeting Submission	Tanveer	30/09/24
Peer Review Submission	All	30/09/24
Test Cases – 1, 2	Russell	7/10/24
Test Cases – 3, 4	Tanveer	7/10/24
Test Cases – 5, 6	Hope	7/10/24

5. Next Meeting					
Date	17/10/2024	Time:	9:00 pm	Location:	Online - Teams
Objectives:	Test Cases, Analysis and Status Report				

Supervision Meeting Minutes 14

Team Name:	Taruho		
Date of Meeting:	17/10/2024	Time Duration:	1 Hour
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Progress Report, SQL map testing results

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Andrew David	9:00 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Progress Report	Updating supervisor about weekly progress
Testing Demonstration	Discussed the output result of SQL map

4. Action Items		
Tasks	Assigned to	Due Date
Drafting the final report	All	20/10/24

5. Next Meeting					
Date	24/10/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Final reprt draft discussion				

Supervision Meeting Minutes 15

Team Name:	Taruho		
Date of Meeting:	24/10/2024	Time Duration:	1 Hour
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Final reprt draft discussion

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:00 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Draft report	Discuss about the format of the draft
Allocation	Allocation of work for each team member

4. Action Items		
Tasks	Assigned to	Due Date
Final report draft allocation	All	27.10.2024

5. Next Meeting					
Date	31/10/2024	Time:	9:00 pm	Location:	Online - Teams
Objectives:	Updating about the draft report				

Supervision Meeting Minutes 16

Team Name:	Taruho		
Date of Meeting:	*31*/10/2024	Time Duration:	1 Hour
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Final report structure and details discussion

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122

Andrew David	9:00 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Final Report Structure	Detail discussion about the final report and paper submission

4. Action Items		
Tasks	Assigned to	Due Date
Final report draft completion	All	4/11/2024

5. Next Meeting					
Date	6/11/2024	Time:	9:00 pm	Location:	Online - Team App
Objectives:	Final report draft discussion and review				

Supervision Meeting Minutes 17

Team Name:	Taruho		
Date of Meeting:	*6/11*/2024	Time Duration:	1 Hour
Minutes Prepared By:	Tanveer	Location:	Online - Team App

1. Meeting Objective(s)
Final report draft discussion and review

2. Attendance at Meeting			
Name	Time of Arrival	E-mail	Phone
Tanveer Osman	9:00 pm	osmant01@myunitec.ac.nz	0278268337
Jun Russel Ragasa	9:00 pm	ragasj01@myunitec.ac.nz	02108389423
Hope Pose	9:00 pm	taimah01@myunitec.ac.nz	02110955122
Andrew David	9:00 pm	adavid@unitec.ac.nz	+64 9 892 7371

3. Agendas and Notes, Decisions, Issues	
Topic	Discussion
Final Report	Discussed and checked the final report draft

4. Action Items		
Tasks	Assigned to	Due Date
Final report draft completion	All	8.11.2024
Report Submission	Tanveer	10.11.2024

5. Next Meeting					
Date	N/A	Time:	N/A	Location:	N/A
Objectives:	N/A				

18.4.0 Supplementary materials relevant to the project

18.4.1 Project Risk Register, Issue Register and Risk Matrix

RISK REGISTER

A	Risk Name	Date Raised	Risk & Impact Description	Likelihood	Consequences	Overall	Proposed Mitigation	Owner
1.1	**Software Corruption**	12-Sep-24	Loss / corruption	Possible	Extra workloads, tasks being not achieved or unable to hand in on time.	High	Cloud storage, multiple back ups	Team Taruho
2.1	**Computer Hardware**	12-Sep-24	Broken or damaged laptop	Possible	Loss of work and time, emotional distress and financial impact.	High	using a good quality computer system with a large storage unit for back ups.	Team
2.2	**Potential Virus or Malware infection**	13-Sep-24	Loss of project	Unlikely	Loss of work and time	Low	using updated software and antivirus/ malware protection	Team
3.1	**Project Scope Change**	14-Sep-24	Scope changes the requirements	Possible	Would distrupt timeline and possible extra work to be delegated.	Moderate	ensuring project requirements are stipulate in contract and if deviated will require renegotiation of contract.	Team
4.1	**Injured Member/ ill Team member (Low impact)**	6-Aug-24	Team member falls ill or gets injured.	Possible	Possible extra work to be delegated or reallocated to other team members to meet deadline.	High	Arrange to virtual meetings for catch up so team member can continue to work whilst away from class.	Team
4.2	**Injured Member/ ill Team member (High impact)**	6-Aug-24	Team member falls ill or gets injured.	Possible	Possible extra work to be delegated or reallocated to other team members to meet deadline.	High	Project has an order of priorty to be worked on if a team member is unable to continue	Team

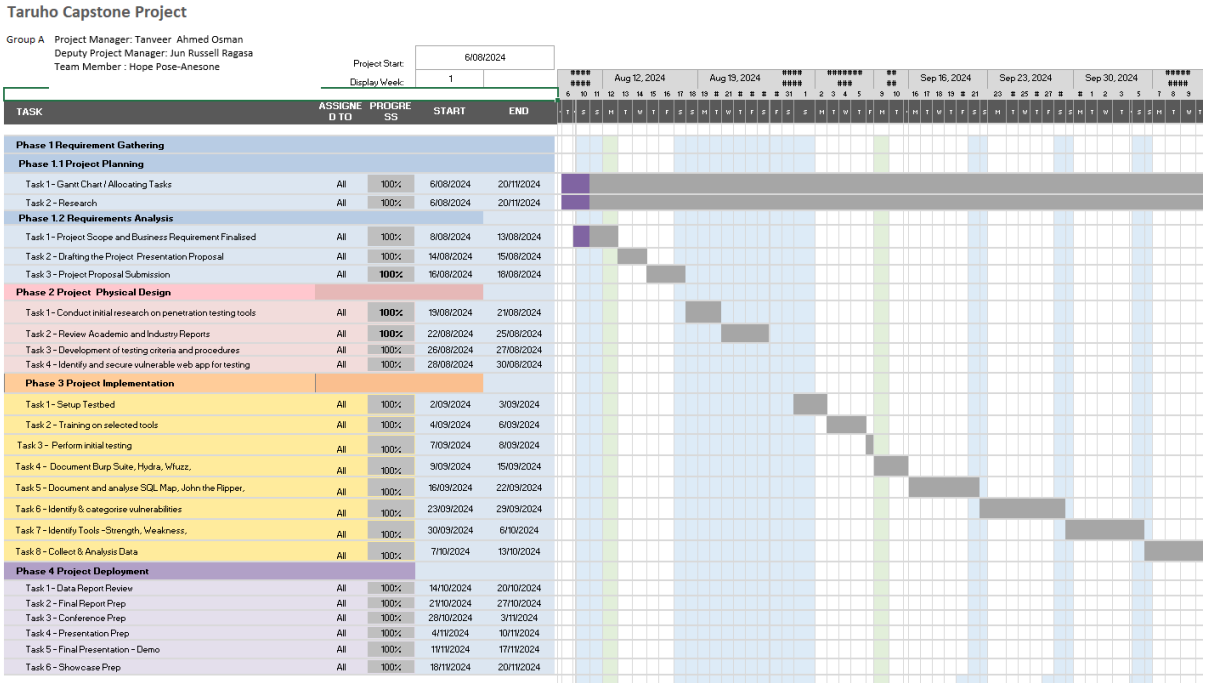
ISSUES REGISTER

ID	Date Raised	Issue Title	Issue Description	Specific Deliverable/Milestone affected	Consequences	Proposed Resolution / Mitigating Action	Issue Owner	Target Date to Resolve	Progress Status / Commentary	Open / Closed
4.1	12-Sep	ill team member (low impact)	team member ill for several days away from class.	Phase 3 Project Implementation	Possible delay to meeting next milestone deadline if work load is backed up.	Strong communication between team and ill team member. Work load reallocation not required. Team member able to work from home remotely. Will reasses with ill team member daily to check in if they are having difficulty to produce work remotely and if work load is to be reallocated to other team members to ensure deadlines are kept.	Team Taruho	5-Nov	Team member recovered from illness able to stay upto date with the project remotely. Continued support from team as they checked on ill team member daily to inquire if help is need.	Closed
4.1	12-Sep	ill team member (low impact)	team member ill for several days away from class.	Phase 3 Project Implementation	Possible delay to meeting next milestone deadline if work load is backed up.	Strong communication between team and ill team member. Work load reallocation not required. Team member able to work from home remotely. Will reasses with ill team member daily to check in if they are having difficulty to produce work remotely and if work load is to be reallocated to other team members to ensure deadlines are kept.	Team Taruho	5-Nov	Team member continues to be very unwell and unable to work on project. Team member updates team what is left to be action for redistribution to of tasks to get completed.	Open (continued ID# 4.2)
4.2	12-Sep	ill team member (high impact)	team member very unwell for several days unable to complete work remotely	Phase 3 Project Implementation & Phase 4 Project Deployment	Additional work load on other team members who have taken on additional responsiblities of ill team member.	Prioritise work load and continue with strong communication with team as to where extra support may be needed.	Team Taruho	5-Nov	Team member recovered and able to resume work. Team has assisted with ill team members workload and worked on prioritising tasks to be completed. Ill team member was also able to catch up and complete	Closed
2.1	13-Sep	Computer software	This happening during Pentesting of Hydra Tool	Testing and Implementation Phase	Computer was in a frozen mode and couldn't able to test the tool also some works were unable to retrieve	Virtual Machine needs to reset and ensure always backup works	Team Taruho	6-Nov	Issue was resolved asap. Team were able resume on the next day.	Closed

Risk Matrix

Likelihood/Impact	Insignificant	Minor	Moderate	Major	Catastrophic
Almost Certain	Moderate	High	High	Extreme	Extreme
Likely	Moderate	Moderate	High	Extreme	Extreme
Possible	Low	Moderate	High	High	Extreme
Unlikely	Low	Low	Moderate	High	High
Rare	Low	Low	Moderate	Moderate	High

18.4.2 Gantt Chart



Taruho Capstone Project

Group A Project Manager: Tanveer Ahmed Osman
Deputy Project Manager: Jun Russell Ragasa
Team Member : Hope Pose-Anesone

Project Start: 6/08/2024
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 Requirement Gathering				
Phase 1.1 Project Planning				
Task 1 - Gantt Chart / Allocating Tasks	All	100%	6/08/2024	20/11/2024
Task 2 - Research	All	100%	6/08/2024	20/11/2024
Phase 1.2 Requirements Analysis				
Task 1 - Project Scope and Business Requirement Finalised	All	100%	8/08/2024	13/08/2024
Task 2 - Drafting the Project Presentation Proposal	All	100%	14/08/2024	15/08/2024
Task 3 - Project Proposal Submission	All	100%	16/08/2024	18/08/2024
Phase 2 Project Physical Design				
Task 1 - Conduct initial research on penetration testing tools	All	100%	19/08/2024	21/08/2024
Task 2 - Review Academic and Industry Reports	All	100%	22/08/2024	25/08/2024
Task 3 - Development of testing criteria and procedures	All	100%	26/08/2024	27/08/2024
Task 4 - Identify and secure vulnerable web app for testing	All	100%	28/08/2024	30/08/2024
Phase 3 Project Implementation				
Task 1 - Setup Testbed	All	100%	2/09/2024	3/09/2024
Task 2 - Training on selected tools	All	100%	4/09/2024	6/09/2024
Task 3 - Perform initial testing	All	100%	7/09/2024	8/09/2024
Task 4 - Document Burp Suite, Hydra, Wfuzz,	All	100%	9/09/2024	15/09/2024
Task 5 - Document and analyse SQL Map, John the Ripper,	All	100%	16/09/2024	22/09/2024
Task 6 - Identify & categorise vulnerabilities	All	100%	23/09/2024	29/09/2024
Task 7 - Identify Tools -Strength, Weakness,	All	100%	30/09/2024	6/10/2024
Task 8 - Collect & Analysis Data	All	100%	7/10/2024	13/10/2024
Phase 4 Project Deployment				
Task 1 - Data Report Review	All	100%	14/10/2024	20/10/2024
Task 2 - Final Report Prep	All	100%	21/10/2024	27/10/2024
Task 3 - Conference Prep	All	100%	28/10/2024	3/11/2024
Task 4 - Presentation Prep	All	100%	4/11/2024	10/11/2024
Task 5 - Final Presentation - Demo	All	100%	11/11/2024	17/11/2024
Task 6 - Showcase Prep	All	100%	18/11/2024	20/11/2024

18.4.3 Impact

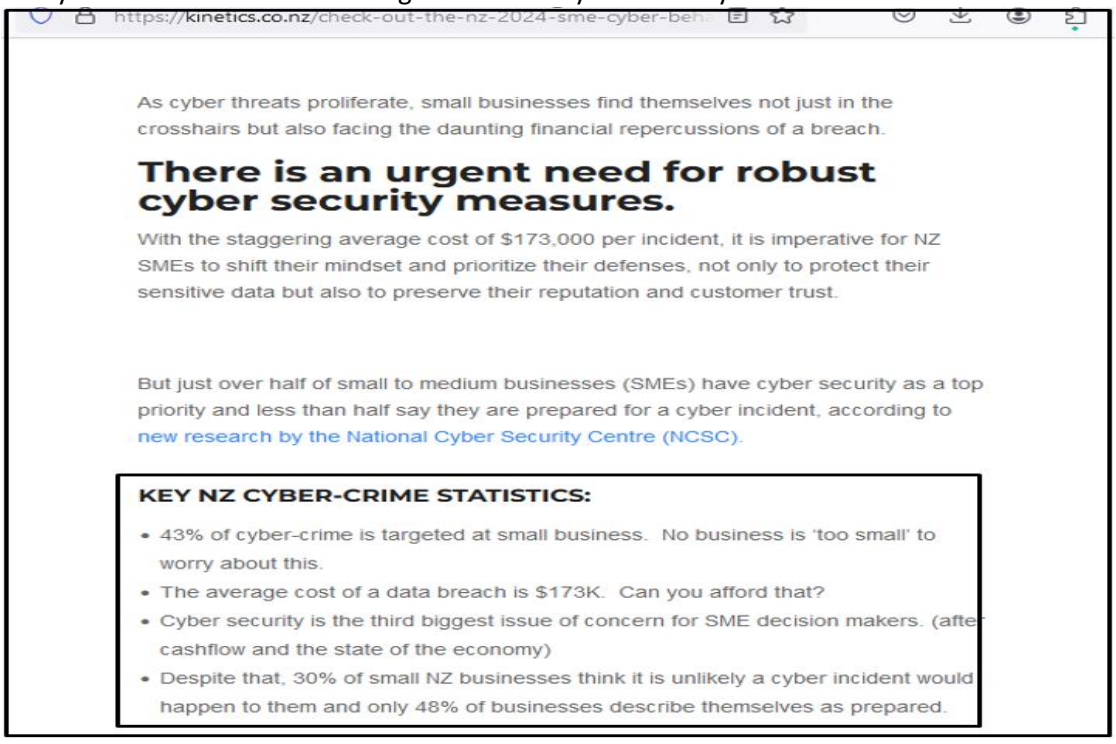
Diversity shortfalls according to Digital Aoteroa Report 2021:

TABLE 3: Diversity shortfalls

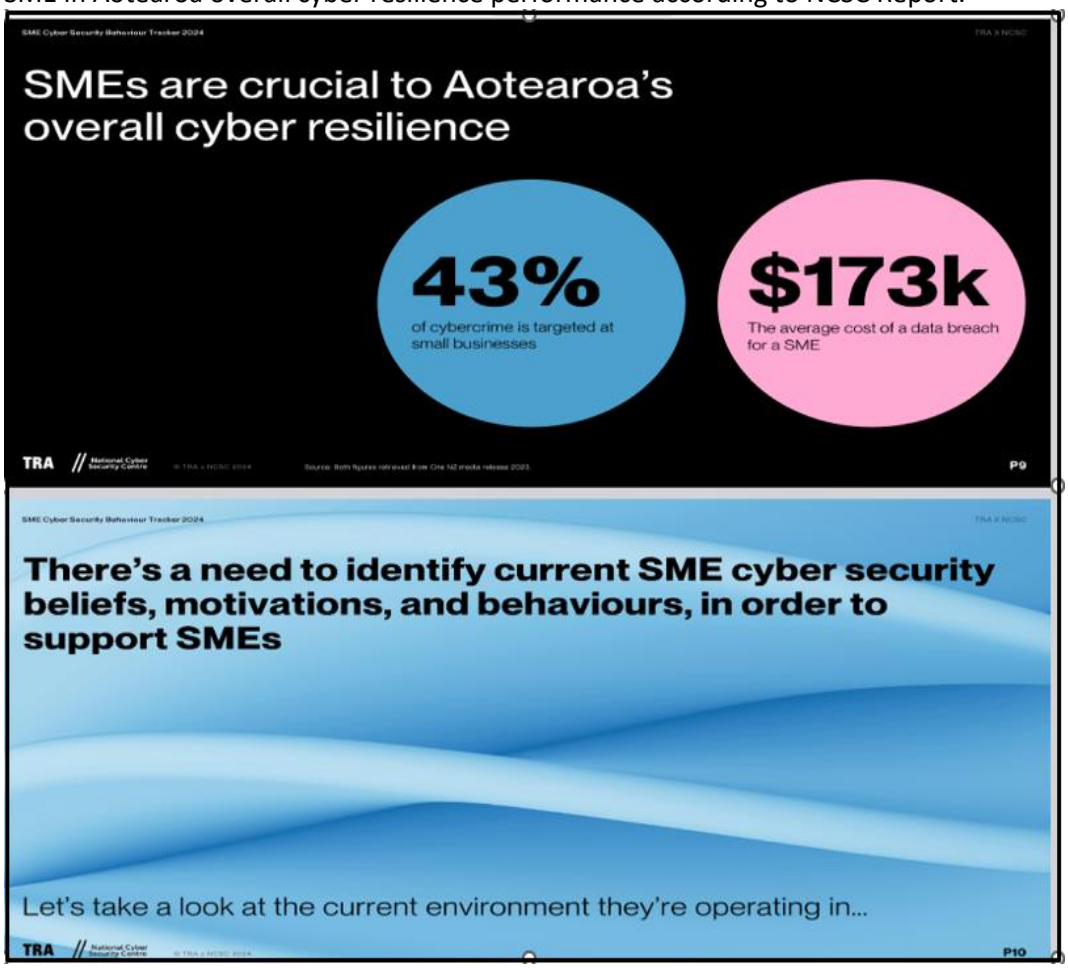
	All Students Yrs 11-13		NCEA Technology Yrs 11-13		IT Degree Graduates		Digital Tech Workforce	
	2015	2021	2015	2021	2015	2021	2015	2021
Female	50%	50%	41%	40%	22%	24%	27%	29%
Māori	20%	21%	15%	14%	3.8%	4.5%	4.1%	4.8%
Pasifika	9%	10%	8%	8%	3.6%	4.2%	2.8%	4.4%

Source: Ministry of Education, Ministry of Business, Innovation and Employment, NZTech.

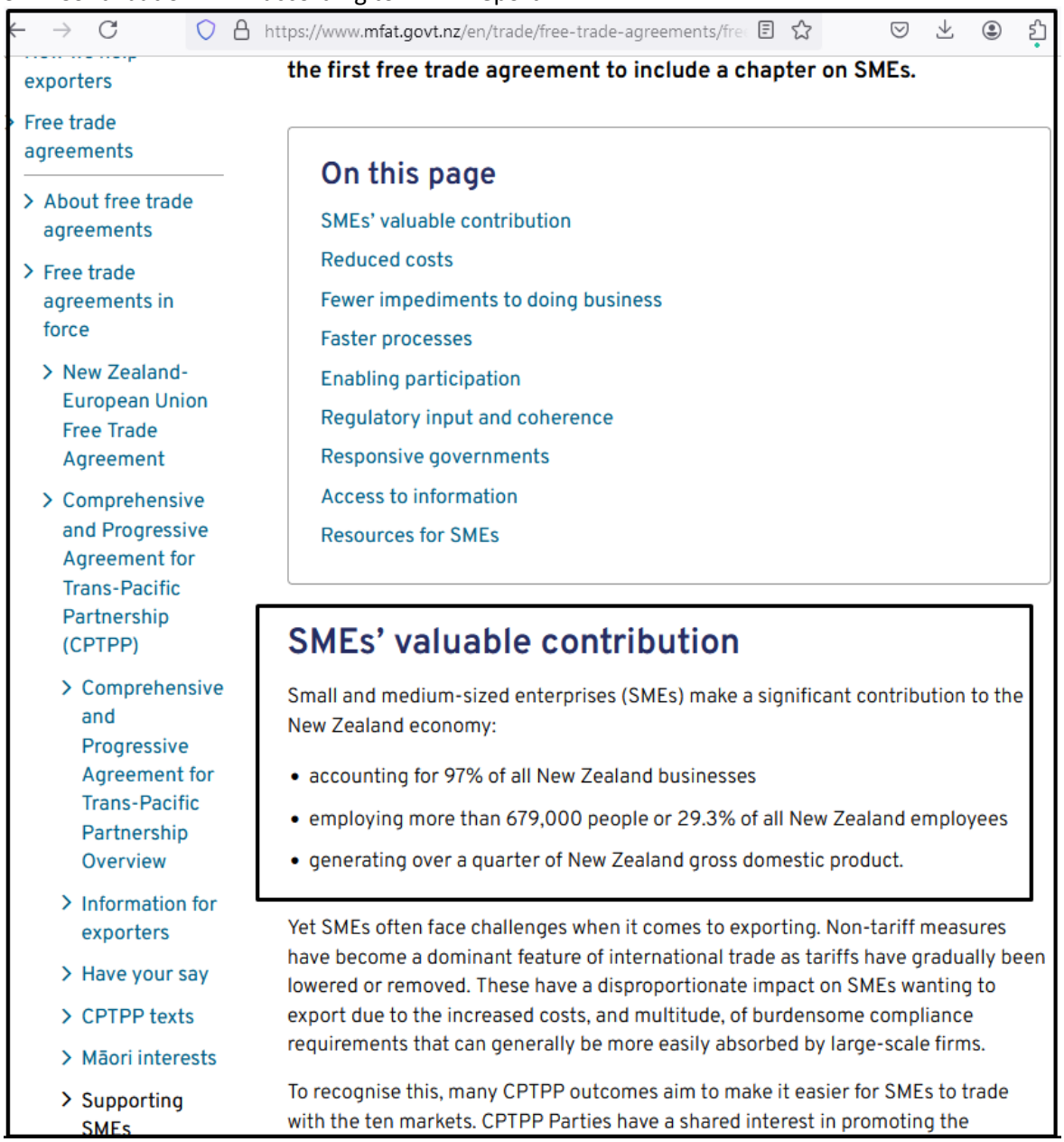
NZ Cybercrime Statistics according to National Cyber Security Centre:



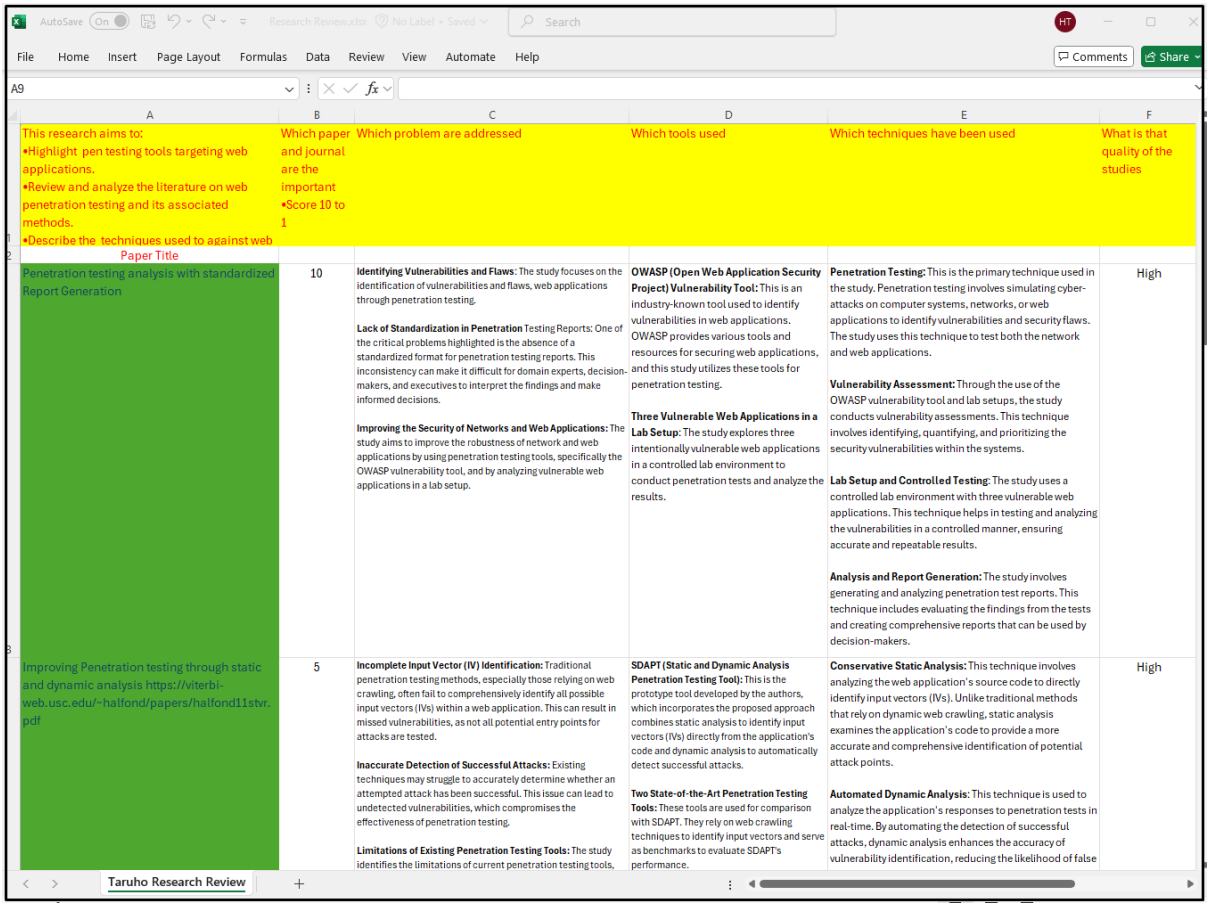
SME in Aotearoa overall cyber resilience performance according to NCSC Report:



SME Contribution in NZ according to MFAT report:



18.4.4 Literature Review:



This research aims to: •Highlight pen testing tools targeting web applications. •Review and analyze the literature on web penetration testing and its associated methods. •Describe the techniques used to against web	Which paper and journal are the important •Score 10 to 1	Which problem are addressed	Which tools used	Which techniques have been used	What is that quality of the studies
Paper Title					
Penetration testing analysis with standardized Report Generation	10	**Identifying Vulnerabilities and Flaws**: The study focuses on the identification of vulnerabilities and flaws, web applications through penetration testing. **Lack of Standardization in Penetration** Testing Reports: One of the critical problems highlighted is the absence of a standardized format for penetration testing reports. This inconsistency can make it difficult for domain experts, decision-makers, and executives to interpret the findings and make informed decisions. **Improving the Security of Networks and Web Applications:** The study aims to improve the robustness of network and web applications by using penetration testing tools, specifically the OWASP vulnerability tool, and by analyzing vulnerable web applications in a lab setup.	**OWASP (Open Web Application Security Project) Vulnerability Tool:** This is an industry-known tool used to identify vulnerabilities in web applications. OWASP provides various tools and resources for securing web applications, and this study utilizes these tools for penetration testing. **Three Vulnerable Web Applications in a Lab Setup:** The study explores three intentionally vulnerable web applications in a controlled lab environment to conduct penetration tests and analyze the results.	**Penetration Testing:** This is the primary technique used in the study. Penetration testing involves simulating cyber-attacks on computer systems, networks, or web applications to identify vulnerabilities and security flaws. The study uses this technique to test both the network and web applications. **Vulnerability Assessment:** Through the use of the OWASP vulnerability tool and lab setups, the study conducts vulnerability assessments. This technique involves identifying, quantifying, and prioritizing the security vulnerabilities within the systems. **Lab Setup and Controlled Testing**: The study uses a controlled lab environment with three vulnerable web applications. This technique helps in testing and analyzing the vulnerabilities in a controlled manner, ensuring accurate and repeatable results. **Analysis and Report Generation:** The study involves generating and analyzing penetration test reports. This technique includes evaluating the findings from the tests and creating comprehensive reports that can be used by decision-makers.	High
Improving Penetration testing through static and dynamic analysis https://viterbi-web.usc.edu/~halfond/papers/halfond11stvr.pdf	5	**Incomplete Input Vector (IV) Identification:** Traditional penetration testing methods, especially those relying on web crawling, often fail to comprehensively identify all possible input vectors (IVs) within a web application. This can result in missed vulnerabilities, as not all potential entry points for attacks are tested. **Inaccurate Detection of Successful Attacks:** Existing techniques may struggle to accurately determine whether an attempted attack has been successful. This issue can lead to undetected vulnerabilities, which compromises the effectiveness of penetration testing. **Limitations of Existing Penetration Testing Tools:** The study identifies the limitations of current penetration testing tools,	**SDAPT (Static and Dynamic Analysis Penetration Testing Tool):** This is the prototype tool developed by the authors, which incorporates the proposed approach combines static analysis to identify input vectors (IVs) directly from the application's code and dynamic analysis to automatically detect successful attacks. **Two State-of-the-Art Penetration Testing Tools:** These tools are used for comparison with SDAPT. They rely on web crawling techniques to identify input vectors and serve as benchmarks to evaluate SDAPT's performance.	**Conservative Static Analysis:** This technique involves analyzing the web application's source code to directly identify input vectors (IVs). Unlike traditional methods that rely on dynamic web crawling, static analysis examines the application's code to provide a more accurate and comprehensive identification of potential attack points. **Automated Dynamic Analysis**: This technique is used to analyze the application's responses to penetration tests in real-time. By automating the detection of successful attacks, dynamic analysis enhances the accuracy of vulnerability identification, reducing the likelihood of false	High

Taruho Research Review

18.4.5 Jira Board