BreachBuster AI Pentesting Assistant

By

M. Umar Usman

M. Yawar Naseem Satti

Saad Sultan

Supervised by:

Maj Bilal Ahmed

Submitted to the faculty of Department of Information Security,

Military College of Signals, National University of Sciences and Technology, Islamabad,

in partial fulfillment for the requirements of B.E Degree in Information Security.

June 2024

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"BreachBuster AI Pentesting Assistant"

is carried out by

M. Umar Usman

M. Yawar Naseem Satti

Saad Sultan

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of

Engineering in Information Security in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor

Maj Bilal Ahmed

Department of IS, MCS

Date: _____

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

Allah Subhan'Wa'Tala is the sole guidance in all domains. Our parents, colleagues, and most of all our supervisor, Maj Bilal Ahmed, we acknowledge your invaluable guidance. We also appreciate the dedication of our group members, who worked steadfastly through all adversities.

Plagiarism Certificate (Turnitin Report)

This thesis has ____ similarity index. Turnitin report endorsed by Supervisor is attached.

_____

M. Umar Usman

NUST Serial no 359291

_____

M. Yawar Naseem Satti

NUST Serial no 358978

_____

Saad Sultan

NUST Serial no 358993

_____

Signature of Supervisor

Maj Bilal Ahmed

ABSTRACT

The "BreachBuster AI Pentesting Assistant" is a pioneering project aimed at revolutionizing the field of cybersecurity and penetration testing. This project introduces an innovative chatbot solution powered by the NLTK Python library, specifically designed to assist users in conducting comprehensive penetration tests on web applications. Unlike existing tools that often require significant expertise and lack tailored guidance, our chatbot offers a user-friendly interface and step-by-step instructions, making it accessible even to individuals with limited penetration testing knowledge.

The core functionality of the chatbot includes database integration through APIs, where it retrieves information on web technologies and vulnerabilities to provide accurate assessments. By leveraging machine learning algorithms for natural language processing, the chatbot can understand user queries, help analyze web application technologies, and recommend specific payloads and scripts to check for vulnerabilities effectively.

This project addresses the challenges of complexity, resource intensiveness, and rapid technology changes in web security by providing a guided approach to penetration testing. The deliverables include a fully functional chatbot interface, customized payloads and scripts, and detailed reports outlining identified vulnerabilities. The "BreachBuster AI Pentesting Assistant" aims to empower security professionals and organizations with an efficient and effective tool for enhancing the security posture of their systems.

Table of Contents

Chapter 1: Introduction

In the realm of cybersecurity, the constant evolution of threats demands innovative solutions that can adapt and respond effectively to emerging challenges. The BreachBuster AI Pentesting Assistant represents a guided approach to penetration testing, leveraging the power of artificial intelligence (AI) to fortify digital defenses.

Traditional penetration testing approach confronts difficulties that are principally manual processes, difficult times, and the lack of efficiency in scaling accompanied by multiple web applications' growing sophistication. The main objective of this project is to tackle the soft spot of AI in performing manual tasks, such as guiding security engineers during security audits. Hence, the Chatbot provides the guidance in the often-complex domain of web application security.

The core of BreachBuster AI Pentesting Assistant lies in making the pen testing workflow more manageable and powerful. The integration of libraries such as NLTK and Flask would enable the conversational agent to recognize inputs from the user, give live responses on a web application and to be capable of seeking for information from a wide database of vulnerabilities and exploits through the use of APIs.

The project's objectives are twofold: to focus on the development of a human-centered tool centered on simplification of the highly complex domain of penetration testing and to possibly contribute to the artificial intelligence-based cybersecurity revolution. We implement data annotation, trainings of algorithms and continuous learning mechanisms to make the chatbot adaptable to the everchanging threats and build an environment where it can provide timely insights to secure web applications.

1.1 Overview

A wide range of issues had to be considered prior to the design of the BreachBuster AI Penetration testing Assistant to minimalize the existing flaws and gaps present in the current penetration testing solutions. The choice to spearhead this campaign is one that was informed by an in-depth assessment of the cybersecurity milieu, the MITRE ATT&CK Framework, which is especially applicable, as well as a contemplation of the dynamic nature of cyber threats that target web applications.

The lifecycle of penetration testing delineated with scanning, analysis, invasion, and attack phase gave the framework in a general outline to build the chatbot's functionality. Though, existing Pentesting tools offer the structured procedure towards attack vectors detection and response; still, with lack of the inherent information, they may not be able to detect the vulnerabilities in software systems environment precisely. The fact that there was no specification of certain common exploits, script, and precise guidance for exploiting vulnerabilities provided a cause for the need for a more refined, unique, and in-depth solution.

Our solution, BreachBuster AI Pentesting Assistant wants to fill the gap by offering the all-round way to a guided and well-thought-out penetration testing. Unlike the traditional discovery tools which are usually programmed only to scan and report, our chatbot is different since it also guides, it does it step by step, and provides payloads to test. This thorough level of specificity enables covering the whole area and helps the computer security engineers as well as penetration testers to find unknown system vulnerabilities that artificial automated scanners have limited ability to spot.

Part of our chatbot feature is the capability to help users of varying degrees of Pentesting skills to our advantage. People who are new in the profession of penetration testing may use the

chatbot as an educational tool which offers clear explanations, guidelines using features that provide user-friendly interface that eliminates the complexities of penetration testing. It reliably performs the duties of seasoned instructions as an assistant, supplementing their expertise with AI powered insides, real-time help, and access to a database of known vulnerabilities and exploits through APIs.

It is our goal to replace the standard and manual Pentesting process by merging the artificial intelligence components together with a clear and interactive approach process of online security audits. This synopsis introduces the deep dive into the technical details, design principles and planned results. These constitute the project's key attributes related to futureproofing of cybersecurity software and infrastructure against emerging threats with consequences.

Following are the issues identified:

1. Complexity of Web Security: The fact that web security threats tend to become more complex and variegated, using new attack vectors and methods, requires a robust and versatile testing process to make sure that there will be a thorough examination and identifying of the vulnerabilities prior to the application of measures of prevention.
2. Resource Intensiveness: Regular yet exhausting, regular pen testing methods usually make excess deployment of human resources, previously scheduled time, and special expertise. This, accordingly, raises questions about companies with finances dieting or short schedule.
3. Rapid Technology Changes: The fast upgrade of web technologies, frameworks and platforms creates separate vectors of security for malicious actions, which, in turn, increases the number of threats. In this context, rapidly encountering intelligent machines in the security

sector requires a proactive approach to testing strategies and adjusting to new security threats.

4. Diverse Web Application Ecosystem: The different web applications, custom-built ones, others from third-party, and some that are written in a very old way, pose a problem concerning the implementation of systemwide security assessments. It is possible that every app has its own vulnerabilities. To identify them testing techniques must be adjusted on a case-by-case basis.
5. Human Error and Oversight: A human factor and security oversight during manual penetration testing can cause remaining vulnerabilities, leaving them undiscovered, incomplete assessments, or risk reviews inaccurate. To address the shortcoming in precision and eliminate the errors, automation and intelligence systems seem to be the key

1.2 Problem Statement

The increasing complexity and dynamism of web application environments, coupled with the high demand for resources and expertise in traditional manual penetration testing, pose significant challenges to ensuring robust cybersecurity. These challenges necessitate an innovative solution that integrates artificial intelligence and natural language processing to streamline the penetration testing process, enhance efficiency, and maintain up-to-date defenses against emerging threats.

1.3 Proposed Solution

The proposed solution is “BreachBuster AI Pentesting Assistant”, which takes into account the challenges concerning the multidimensionality of web security, high resource requirements, constant technological changes, the variety of web application technologies employed, and the drop in the efficiency of manual penetration testing. It addresses this concern by bringing in the

use of artificial intelligence through the natural language processing technology to bridge the gap between the penetration testers' actions in the field with the use of a chatbot mechanism system.

The AI based chatbot system differentiates itself from manual Pentesting by offering programmed, step-by-step instructions that save time. By easing the performance of repetitive tasks, providing application specific scripts, and delivering insights and recommendations, the chatbot assists users of varying skill levels in conducting comprehensive vulnerability testing.

Moreover, the chatbot links to a comprehensive database of known web vulnerabilities and potential exploits, ensuring it remains up to date with the latest threats and attack vectors. This continuous update capability ensures that penetration tests conducted with the chatbot are based on current and accurate information, enhancing the accuracy of vulnerability detection and analysis.

A key function of the chatbot application is to produce detailed reports about the pentest of the tested servers and web applications. The reports outline the information about identified vulnerabilities, their level of impact, exploitability, complexity, and how the vulnerability can be mitigated. This functionality does not only help to implement effective risk management strategies but also helps in effective communication and collaboration among various stakeholders involved in cybersecurity assessments.

1.4 Working Principle

The project mainly works on the principles of database extraction with machine learning algorithms used for Natural Language Processing. The project is divided into different modulus and every module is interconnected with the next module. The list of modules is as under:

- Chat Bot

- Data extraction from Database through APIs
- Decision based upon Inputs
- Report Generation
- Access Control

1.4.1 Chat Bot:

The Chat Bot system utilizes Natural Language Processing (NLP) to render the user interface, provide optimal response based on input and to make the operation of the system easier.

1.4.2 Data Extraction from Database:

The data will be extracted from the professionally maintained databases in the form of NVD's CVE repository of known vulnerabilities and Exploit DB's repository of known exploits against those vulnerabilities, which will be presented to the user upon demand.

1.4.3 Decision based upon Inputs:

Following the inputs from users, the system will custom responses that contain detailed instructions to follow, in order to perform the PT (penetration testing) activity.

1.4.4 Report Generation:

When the entire activity including the sequence of steps has been done there is the option to generate a Report detailing what web technology has been penetration tested, and with what vulnerabilities and exploits.

1.4.5 Access Control:

An ordinary user would only be assigned the chat bot frontend, an administrator who would be in charge of the user management, would approve or reject users.

1.5 Objectives

1.5.1 General Objectives:

The project specific aims are to design and create an AI driven penetration testing helper that can help enhance the efficiency and effectiveness of security audits of web applications. Developing an assistant that enables streamlining the process of testing web systems and providing cyber experts with optimum guidance and reports is the main purpose of this tool.

1.5.2 Academic Objectives:

i. Using the machine learning techniques for natural language understanding (NLP) that enable machines to understand human's style of communication and then further provide guidance within the chatbot interface.
ii. To link with the external databases through APIs for getting updated information on vulnerability and becoming precise in the process of identifying vulnerabilities.
iii. With functionality, on the one hand, the user-friendly interface should be optimized, such that both novice and advanced users can enjoy good navigation experience and ease of use on the other hand.

iv. The AI powered chatbot's performance and efficiency is to be evaluated via testing scenarios and through comparison against other penetration testing methods (traditional penetration testing) that are widely used already.

1.6 Scope

This project's scope includes the information security field where security is of prime importance. Its application would cut across sectors of the economy that are ICT dependent like finance, health care, ecommerce, and government organizations. The project's scope includes:

1. Industry Applications: Artificial intelligence supported penetration testing is crucial for many industries for to the enhancement of the security posture of servers and applications. It also strengthens security by providing critical information on potential system exposures and provides stronger defenses.
2. Educational Institutions: Universities imparting cybersecurity courses and creating training programs can use this tool as an approach to practical exercises with students in capability building. This type of training gives students an opportunity to practice all the techniques used to attack web applications and test their skills.
3. Security Consulting Firms: The technology employed here can be employed by the security consulting companies to perform in-depth and strong penetration tests for their clients. It harmonizes and speeds up the testing process, enables smooth teamwork between cybersecurity professionals, and provides professional reports to clients.
4. Government and Regulatory Bodies: The tool can enable regulatory bodies and government agencies responsible for cybersecurity to track the web application security status of web apps within their control. It improves the observance of cyber security laws and standards, protects data privacy, and sustains it.

Overall, the scope of our project covers many companies that are either seeking to improve their digital security or that want to identify and remedy vulnerabilities so that cybersecurity risk can be raised.

1.7 Deliverables

1.7.1 Chatbot Interface for Penetration Testing Assistance:

The key project's result is the creation of a chatbot whose conversational user interface will help web app Pentesting by easing the process. This chatbot which has been created using the NLTK python library with the support of Natural Language Processing (NLP) algorithms will provide the users with step-by-step guidance, tactical strategies, and help through all the phases of penetration testing. It will give the features of sentence input, report generation, report downloading, launching webshell, loading previous chats, starting new chats etc.

1.7.2 Comprehensive Vulnerability Database and Exploit Library Linkage:

CVE’s Vulnerability database and Exploit DB’s exploit library are the two most important worktables that we are going to link to, using APIs. These database will cover a multitude of techs, vulnerabilities, and known exploits. Such integrations will provide all necessary standards for chatbot's advice and recommendations systems. Penetration testing assistant will provide specific payloads, scripts, and approaches that will be used during target’s assessment; that aim is to increase effectiveness of Pentesting activities.

1.7.3 Automated Report Generation Tool:

Besides construction of a chatbot, the program is capable of generating automated reports. The Pentesting assistant will analyze results of penetration tests that are conducted by users through the chatbot and aggregate detailed reports summing up vulnerabilities found, their level of severity, impact, available remedial measures into a single document. One of the tools' major aims is to simplify documentation process and provide actionable data for feasible use by various users and stakeholders.

1.8 Relevant Sustainable Development Goals

The project "BreachBuster AI Pen testing Assistant" aligns with several Sustainable Development Goals (SDGs):

- Goal 9: Industry, Innovation, and Infrastructure

  Enhancing cybersecurity infrastructure through innovative tools and methodologies.
- Goal 16: Peace, Justice, and Strong Institutions

  Promoting cybersecurity and ensuring digital stability for strong institutions.
- Goal 4: Quality Education

  Providing educational resources for cybersecurity professionals and learners.
- Goal 8: Decent Work and Economic Growth

  Supporting economic growth by fostering a secure digital environment.

1.9 Structure of Thesis

Chapter 2 contains the literature review and the background and analysis study this thesis is based upon.

Chapter 3 contains the design and development of the project.

Chapter 4 analyzes details of working and of the GUI/frontend.

Chapter 5 contains the conclusion of the project.

Chapter 6 highlights the future work needed to be done for the commercialization of this project.

Chapter 2: Literature Review

2.2 Penetration Testing Methodologies

2.2.1 Traditional Penetration Testing Approaches

The methods of traditional penetration testing are basic techniques that are applied to assess the security of systems. Some of the frameworks that are commonly used are the Open-Source Security Testing Methodology Manual (OSSTMM), NIST SP 800115, and the OWASP Testing Guide.

- OSSTMM: This methodology offers a holistic security testing methodology, which stresses operational security metrics and a comprehensive approach for performing a complete security assessment. It is about a well-organized testing process that covers all security areas starting from physical to process controls (Herzog, 2015).
- NIST SP 800115: The Special Publication 800115 of the National Institute of Standards and Technology provides a technical security testing measure, which includes network, web application, and wireless security testing. It strives to make the process a standard and to allow the consistency among different testing engagements (Scarfone et al., 2008).
- OWASP Testing Guide: This manual offers an elaborate approach intended for web application security testing. It includes a variety of testing methods and tools, focusing on systematic methods of finding and reducing security vulnerabilities (OWASP, 2017).

The weaknesses of the traditional, manual testing methods include, time consuming, high resource requirement, and chances for human errors. In a dynamic and complex environment, the

limitation can prevent the penetration testing from working effectively and efficiently (AlShaer et al. , 2017).

2.2.2 Automated Penetration Testing Tools

Some of the inefficiencies of manual penetration testing are dealt with by the use of automated tools. Automated tools include Metasploit, Nessus, and Burp Suite.

- Metasploit: An opensource framework that supplies security vulnerabilities details and helps in penetration testing and IDS signature development. Metasploit makes vulnerability exploitation process automated, although, the user needs strong knowledge to interpret the results and introduce complex attacks (Bejtlich, 2013).
- Nessus: An all-inclusive vulnerability scanner that detects possible weaknesses in systems and networks. Nessus includes an auto discovery of vulnerabilities and configuration issues however it usually requires manual validation of findings (Ruff, 2012).
- Burp Suite: A web application security testing graphical tool featuring scanning, spidering, and vulnerability analysis tools. Although Burp Suite automates many tasks, for thorough analysis and exploitation manual intervention is necessary (Stuttard & Pinto, 2011).

These tools streamline specific aspects of penetration testing, such as vulnerability discovery and initial assessment. However, they do not eliminate the need for skilled testers to interpret results, conduct complex tests, and devise remediation strategies (Engebretson, 2013).

2.3 Artificial Intelligence in Cybersecurity

2.3.1 AI for Threat Detection and Prevention

Artificial intelligence and machine learning have been increasingly applied to enhance threat detection and prevention mechanisms.

- Anomaly Detection: The algorithms of machine learning are used to identify the pattern of anomalies which would be used to detect the malicious activities. Clustering and neural networks are used in this process (Laskov et al. , 2005).
- Intrusion Detection Systems (IDS): AI techniques help the IDS to ensure its accuracy and efficiency by learning from the historical data in order to identify the repetitive patterns that are a sign of an attack (Patcha & Park, 2007).
- Predictive Analytics: Predictive analytics are another type of an AI system that identifies trends in data to predict future security incidents so that appropriate security measures can be taken before an attack (Dua & Du, 2016).

Case studies have demonstrated the effectiveness of AI driven security solutions in detecting sophisticated threats that traditional methods may miss. For instance, a study by Sommer and Paxson (2010) highlighted how AI could significantly reduce false positives in IDS.

2.3.2 AI in Penetration Testing

Research on AI assisted penetration testing explores the use of AI models and frameworks to enhance various aspects of the testing process.

- AI Models: There are a number of AI models which has been used to automate the process of discovery and exploitation of vulnerabilities like reinforcement learning and genetic algorithms (Fraunholz et al., 2018).
- Frameworks: There are a number of other frameworks available which incorporate AI to improve or enhance the process of penetration testing. For example, the priority of vulnerabilities based on the potential impact by using machine learning algorithms (Ghafir et al. , 2018).

2.4 Natural Language Processing in Security Applications

2.4.1 NLP for Threat Intelligence

NLP is utilized to process huge volumes of data that are unstructured across different channels to obtain useful threat intelligence.

- Data Analysis: Many NLP techniques are applied in threat intelligence feeds and dark web sources to transform the available information into security intelligence and trends (Liu et al. , 2011)
- Insight Extraction: NLP applications also harnesses the necessary security report and logs in order to better value or meaning in security operation (Bridges et al. , 2013).

Examples include the use of NLP to parse threat intelligence reports for indicators of compromise (IOCs) and other critical information, which can enhance the situational awareness of security teams (Cheng et al., 2019).

2.4.2 Chatbots and Virtual Assistants

IT support and IT security are two areas in which chatbots, and virtual assistants are becoming more and more commonly employed to facilitate and better interactions.

- IT Support: In the field of cybersecurity, chatbots are used to perform certain tasks including identifying an incident, notifying the employer about certain types of threats, and educating users. They make security teams more efficient and effective – more responsive (Pereira & Díaz, 2019).
- Security Operations: In the field of cybersecurity, chatbots are used to perform certain tasks including identifying an incident, notifying the employer about certain types of threats, and educating users. They make security teams more efficient and effective – more responsive (Pereira & Díaz, 2019).

The following are specific examples: Security awareness training chatbots and incident response designed security awareness training chatbots that play a part in the security postures of organizations by ensuring that information is shared promptly, and action is undertaken (Van Rooyen & Smit, 2017).

2.5 Integration of AI and NLP in Penetration Testing

2.5.1 Existing Solutions and Research

There have been different research papers and industry publications about effective use of the penetration testing with the integration of AI and NLP.

- Technologies and Approaches: Some of the Latest technologies used in these solutions involve A deep learning models for vulnerability detection, NLP algorithms to parse and

analyze security data and AI based frameworks for automated testing (Brown et al. , 2018).

- Academic Research: Research has been conducted into the potential of integrating AI research and NLP for improving penetration testing tools' automation and intelligence. For example, the study by Apruzzese et al. (2018) shows how machine learning is allowing vulnerability assessments to be more accurate.

2.5.2 Gaps and Challenges

Despite advancements, significant gaps and challenges remain in the integration of AI and NLP for penetration testing.

- Data Quality: Predictive models within AI are mostly data driven and it is needless to say that the quality and quantity of the training data is crucial for the model's performance. Frequently it is not enough to use all the available data, or the data might be biased, and this could lead to biased results (Nguyen et al. , 2019).

- Model Accuracy: This oversight process can become practically impossible in complex and dynamic environments in which AI models are used. New models need to be made available to satisfy challenges in security (Apruzzese et al. , 2018).

- Domain Specific Knowledge: Integration requires knowledge about biases within the data and how to better tune AI and NLP specific models for different environments and applications. This web of challenges requires the involvement of cybersecurity specialists and AI scientists (Ghafir et al. , 2018).

Future research can contribute further to the advancement of AI and NLP in penetration testing with the aim of addressing the abovementioned gaps and improving the capabilities and effectiveness of these tools.

2.6 Comparison Table

PentestGPT is an advanced AI-driven platform designed to automate and enhance penetration testing activities. Utilizing natural language processing, it simplifies complex security assessments, providing comprehensive scanning, vulnerability detection, and exploitation capabilities across various domains, including web applications, networks, and hosts. Despite its sophisticated AI and user-friendly interface, PentestGPT falls short in several areas. It lacks the detailed customization and specific phase-oriented approach of BreachBuster, which provides tailored guidance through each phase of penetration testing. Additionally, PentestGPT's reliance on pre-defined scripts and models can limit its adaptability to novel or unique security challenges, making it less flexible in dynamic testing environments compared to BreachBuster's more interactive and comprehensive approach.

A potential third competitor to the BreachBuster project could be Nexpose by Rapid7. Nexpose is a well-known vulnerability management and penetration testing tool that offers extensive scanning capabilities and integrates with Metasploit for exploitation testing. But while Nexpose is a robust tool with extensive features, its lack of AI and NLP integration, along with a more complex user interface, makes it less competitive compared to the innovative, user-friendly approach of BreachBuster.

The comparison table is given below:

Features	**PentestGPT**	**BreachBuster**	**Nexpose**
Target Audience	Medium to large	Low scale companies	Medium to large

	scale companies		scale companies
Cost	Higher	Lower	Higher
User Interface	CLI-based	GUI-based	GUI-based
Ease of Use	Requires technical knowledge	User-friendly, suitable for less technical users	User-friendly, requires some technical knowledge
Target Selection	Manual	Automated target validation and user prompt	Manual and automated
Chat History	Limited or None	Comprehensive, with timestamped sessions	Not available
Kali Integration	Not available	Integrated Kali WebShell for seamless pentesting	Not available
User Privileges	Basic user roles	Admin and regular user roles with approval system	Comprehensive user roles and permissions
Pentesting Stages Guidance	Not specified	Step-by-step guidance through four color-coded stages	Detailed, step-by-step guidance
Tool Installation	Manual	Pre-installed tools, no separate installation required	Pre-installed tools, requires some configuration
Reporting	Basic reports	Detailed reports with vulnerabilities, exploits, CVEs	Detailed reports with vulnerabilities, exploits, CVEs
Session Management	Limited	Save and load chats for session continuity	Comprehensive session management
Accessibility	Requires setup	Easy access through a web interface	Web and desktop interface
Documentation	Basic	Detailed and automated documentation stage	Automated documentation generation
Flexibility in Target Testing	Limited	User prompt for continuing with unresponsive targets	High flexibility in target testing
New User Registration	Basic user creation	Admin approval required for added security	Detailed user registration process
Learning Curve	Steep	Minimal, designed for users with less technical knowledge	Moderate, suitable for technical and semi-technical users

2.7 Pentesting Survey

We have analyzed a Pentesting survey “The State Of Pentesting 2023 Survey Report” by Pentera. (it can be found at https://pentera.io/wp-content/uploads/2024/01/2023-state-of-pentesting-survey-report-1.pdf)

2.7.1 Introduction

The report was conducted by Pentera, a leader in Automated Security Validation. The research was undertaken to understand the current state of security validation in organizations of different sizes across Europe and the USA. The report provides a snapshot of how security leaders in 2023 perceive and choose to adopt security validation strategies.

2.7.2 Methodology

The report is based on data from over 3,100 pentests and responses from more than 1,000 security practitioners in the United States, the United Kingdom, and Germany. The survey was administered online by Global Surveys Research, a global research firm. The respondents were recruited through a global B2B research panel, invited via email to complete the survey, with all responses collected during December 2022.

2.7.3 Key Findings

- Defense in Depth Is Not Sufficient Anymore: Despite large investments in defense-in-depth strategies, where multiple security solutions are layered to best protect critical

assets, 88% of organizations admit to being compromised by a cyber incident over the past two years. On average, companies have almost 44 security solutions in place.

- Cybersecurity Budgets Aren't Impacted by the Financial Slowdown: Despite the current economic slowdown, 92% of organizations are raising their overall IT security budgets.
- Layoffs in Security Teams: 77% of security teams have experienced layoffs.
- Vulnerability Management Struggles: 73% of affected teams struggle to manage vulnerabilities.
- Slower Patching of Critical Vulnerabilities: 96% of security teams were slower to patch critical vulnerabilities compared to 2021.
- Survey Respondents: The survey was conducted among 300 security executives who hold VP or C-level positions in companies with more than 1,000 employees. The respondents were split between the U.S., the UK, and Western Europe.
- Survey Administration: The survey was administered online by Global Surveys Research, a global research firm. The average amount of time spent on the survey was 6 minutes and 44 seconds.
- Top Vulnerabilities: The top vulnerabilities found during pentesting include components with known vulnerabilities, broken access control, server security misconfiguration, and cross-site scripting (XSS).

2.7.4 Conclusion

The report provides insights on current IT and security budgets, cybersecurity validation practices, and how cyber exposure is being managed. It showcases differences between the regions and enterprise sizes. The report also explores how operational changes can jeopardize security.

2.8 Literature Review Summary

The Literature Review chapter provides a comprehensive overview of existing research, technologies, methodologies, and tools related to penetration testing, artificial intelligence (AI), and natural language processing (NLP). This chapter contextualizes the proposed solution by examining current advancements and identifying gaps that the BreachBuster AI Pentesting Assistant aims to address.

Chapter 3: Components of BreachBuster

3.1 Intent detection Model

Intent detection is the focused area for Natural Language Processing (NLP) for Conversational models at the moment. The basic algorithm behind any of the chatbots, AI Based personal assistant, and other platforms is the intent recognition model an algorithm that is trained on the real text in the request or question asked by the user. The training data that is utilized in intent recognition includes intent information that is assigned based on features of sentences. In the mapping algorithm an input is mapped to a correct intent which is already defined. The first requirement is to look for the most likely intent that connects to the system and then decide the action to take based on whether there is a match with the user. For instance: if there's input 'I want to run an Nmap scan' the intent generated for this would be Nmap scan. This addresses the problems associated with the user request and its collaborating factor to boost the system's capability to handle request made by them.

3.1.1 Main Elements of an Intent Detection Framework

1. Data Acquisition:

Creating and synthesizing data assets with diverse language patterns and users' objectives. Features in training data sets for NLP are created through feature engineering which include syntax-based feature types, word embedding and character-based feature.

2. Model Architecture:

Training the intent detection system with machine learning algorithms such as neural networks, support vector machines (SVMs).

3.2 CVE API

3.2.1 Overview

CVE is the systematic way in which the security vulnerabilities for global system are exposed to the world in a structured and processable format. The CVE is a standard for representing common identification of vulnerabilities and exposures cross platform and tools with unique identifiers in the CVE space administered by the MITRE Corporation which is certified by NIST.

3.2.2 Functionality

One of the possible and recommended ways of using the CVE database is the Scripting Interface which allows users to interact with the CVE through API calls. That implies that instead of getting a general blurb of information, it has a specific approach that can be used to search for information about a specific vulnerability as well as information concerned with description and metadata about it. This ability is required to keep security tools and processes up to date with the assistance of the information regarding the risk of the vulnerability.

3.2.3 Usage in Pentesting

1. Vulnerability Identification:

   Penetration testers can easily use the CVE API to identify what vulnerability they might stumble upon in the particular program or system before testing it. The good thing when they incorporate the API, they can thus determine whether a given hard or soft system component is a threat by identifying the state of the hard or soft system component as having CVEs or not.
2. Exploit Research:

The attacker can then use the CVE API to feed the vulnerabilities details into the CVE and find out if an exploit of the CVE exists. It has also facilitated the creation of types of attack which has addressed the security status of the system.

3. Reporting:

An additional meaning of the index is it offers nonsuggestive reporting when vulnerabilities are identified during the penetration test in a consistent manner. The inclusion of CVE IDs in the reports allows for easier and quicker identification of the issues at hand and the search for the optimal methods to solve them.

3.2.4 What CVE API provides

The CVE API provides the following key data and functionalities:

CVE Details: The severity and the reference to a list with information on each of the threats – the description of the threat, its status, and the links to external sources.

CPE Information: CPE enumeration gives us software and versions of software affected by the vulnerability.

Metadata: Metadata information such as CVSS scores including impacts and exploitability.

3.3 Exploit-DB API

3.3.1 Overview

The Exploit-DB is a global database that is considered a source of vulnerability and exploit data or applications. It acts as a central location for coordinating the information on security vulnerabilities that secure uses to oversee specific software programs. The database is usually

managed and updated by the security experts and organizations to make ensure that relevant details of security attacks are included.

3.3.2 Functions of Exploit-DB

1. Centralized Information:

It includes all the obvious hazards such as CVSS scores, affected issues, affected versions, patches.

2. Security Insights:

It helps in the identification of weaknesses and informative tips that tell an entity what possible loopholes exist and how they can be protected.

3.3.3 Risk Management

Aids in making decisions concerning the impact of a particular vulnerability to provide for risk identification and risk ranking services. Exploit-DB is very helpful in the field of Pentesting when a user is scanning for exploits linked with vulnerabilities of a target system or web technology.

3.3.4 Vulnerability Identification

Its uses are for the penetration tester to determine whether there are known vulnerabilities in systems and software that the targets are using. The database will then be compared with details on the system in order to achieve the aim of identifying any exploits for vulnerability in the system.

3.3.5 Exploitation Planning

The recommendation from Exploit-DB is specific which provides some of the exploit codes once you have identified the vulnerability and it entails what the modern-day hackers would use as a malicious user.

3.3.6 API Features of Exploit-DB

API will also be effective in integrating Exploit-DB with other internal tools as well as/or systems/ methods for the automation process.

3.4 Web scraper

A web scraper is a tool designed to extract data from websites. It automates the process of visiting web pages, extracting specific information, and storing it for later use. Web scrapers are widely used for various purposes, such as data mining, market research, and competitive analysis. Exploit-DB API provides links of data to exploits not the exact exploit code. So, in order to get the data, web scarping is used.

3.4.1 Working

Access Web Pages: Access the web page by making a web request to that page, and use the returned web request response.

Parse HTML: HTML parser to read the HTML content and identify different elements of the web page.

Extract Data: Extract data based on where it is present in the web page, using the identifiers and HTML tags.

Store Data: The data extracted is then stored in a way that it can be accessed in different forms such as CSV JSON or database as a case for what the operation will do to the data resources.

3.4.2 Prominent features

Customizable Extraction Rules:

It implies that perspective of encoding information and special tags for defining the area in which the desired information is present on the page has to be communicated.

Pagination Handling: They tend to follow link supporting pages in order to locate some of the other succeeding pages that would complete the extraction of data.

Data Storage: to create the right format to store values as comma separated columns (CSV), JavaScript Object Notation (JSON) or Extensible Markup Language (XML) or a database.

Rate Limiting: Uniformly assign the requests to be delivered to the target web servers to prevent them from becoming overloaded.

Error Handling: Reporting component needs to be responsible for receiving notifications from the scrapping so that the it could raise alerts for issues they have met during the scrapping process.

3.5 Report Generator

After Pentesting is completed then there is an option to generate a report which can include targets, date, time, vulnerabilities, exploits and the detailed chat with the Pentesting assistance which helped in Pentesting a technology.

3.5.1 Reporting Features

Targets: It will show the targets that are assessed for the purpose of Pentesting.

Exploits Used: It will include details of exploits used in the Pentesting process.

Date and Time: Provide date and time of the report generation.

3.6 Chat history

Chat History is an essential element of the user interface of Breach Buster AI Pentesting Assistant to help users quickly view their previous conversations and manage it effectively. This section describes the feature's functionality, interface design, and workflow of use of the Chat History.

3.6.1 Functionality

The Chat History option lets the users load and view past chats made by the same AI Pentesting Assistant with Breach Buster. It is created to enable users to easily go back to their previous activities in order to read through the conversations that were made, recall the progress of the Pentesting processes they were doing or even to recall the information that was needed once again instead of asking them twice.

3.6.2 Interface Design

The Chat History interface can be accessed through a button above reading "Load Chats" on the lefthand side of the user interface. When the user clicks this button, the panel is opened with a list of the previously saved chats. This type of list consists of the timestamp for each chat entry in the list, which may then be referred to by the user to pinpoint a certain discussion thread.

Key elements of the Chat History interface include:

1. Load Chats Button: It is located on the left of the interface and easily noticeable. It is used to open the Chat History panel where the saved conversations are stored.

2. Chat List: On clicking the 'Load Chats' button, a list of stored chats will appear. Each piece of content in this list is also associated with a time stamp that allows users to easily find a particular exchange according to the specified date and time.

3. Chat Entries: All the chat entry comes with a ‘clickable’ icon. Clicking on the list entry launches expanded view presenting timestamped interaction of the user queries and bot responses.

3.6.3 Operational Workflow

The flow of operations carried out by the developed Chat History is made simple and natural. The process involves the following steps:

1. Saving Chats: The user after finishing a chat with the agent is able to save the conversation with the help of the ‘Save chat’ button that is located at the right corner of the chat window. It captures the whole row of all the users’ questions and bot’s answers in the chat conversation under the Chat History.

2. Loading Chats: Attached to the chat is a button – “Load Chats” through which the saved chats can be accessed as and when required by the user. This action displays all saved messages in the Chat History panel that has each message with the attached timestamp.

3. Viewing Chats: This is why when any chat is saved users can simply scroll down to the Chat History list and click on a specific entry for details of that chat. It generates a new page that enables a customer to read through and further reviews the specific conversation.

3.6.4 Importance of Chat History

The Chat History function is one of the key elements of the Breach Buster AI Pentesting Assistant and responsible for the best effectiveness and solvability.

It provides several key benefits:

1. Reference and Review: Chatbot as a service can be helpful to refer back to previous conversations and read on the details or instructions or guidance given in the previous discussion which is very useful for a complex PENTEST work.

2. Continuity: The history of chats is useful for the continuation of the process of pentest so that users can maintain this possibility and not lose the information they discussed in previous sessions.

3. Efficiency: This is very helpful for the users because they do not have to scroll to the log of the chat history to ask the same question again or carry out the same task over and over again.

Overall, it is a well thought out and very useful feature of the Breach Buster AI Pentesting Assistant in terms of helping the user to guide their conversation or access the known history for the conversation with the chatbot. This is very important for the overall UX because it implies access to historical data while using the tool for maintaining continuity of a penetration test.

3.7 Kali WebShell

One of the most important features of the Breach Buster AI Pentesting Assistant and one of the main benefits of using Kali WebShell for the web server penetration test, as I think will provide maximum assist in the from the web server penetration test. This section explains what the Kali WebShell tool does and details are provided about the layout of Kali WebShell home page, how the Kali web shell works and looks like as well as the advantages of using Kali WebShell service for Pentesting instead of using the Kali Linux directly.

3.7.1 Functionality

The Kali WebShell feature allows for Web Server Pen Testing from within the Breach Buster AI Assistant Interface. Users can access a virtual desktop on a full command line interface (CLI) using the Kali Linux GUI on a web browser through a remote Kali Linux machine connected through SSH. This setup ensures that users don’t need to worry about manually installing Kali on

their systems for testing with the framework which makes setup testing much easier, and all the tools are at hand.

3.7.2 Interface Design

Main elements of the interface are:

1. Launch Kali Button: On the right top side corner of the user interface above the log out button there is a primary window which provides access to the Kali WebShell. It is useful whenever a necessity crops up and at the same time it is not a strategic option.

2. Kali CLI Interface: Linux Kali widely known as kali is a Debian based penetration testing Linux distribution. This application also displays the command line interface for those users who may want to perform some operations of penetration testing via several scripts and commands.

3.7.3 Operational Workflow

Kali WebShell operational workflow has a streamlined processing system to make users' experience simple. The process involves the following steps:

1. Launching Kali WebShell: The principle behind the Kali WebShell starts with the users' clicking of the "Launch Kali" button. This separates from the previous action opens up the web browser with the Kali Linux CLI interface an SSH window to a remote machine that is running Kali Linux.

2. Performing Pentesting Tasks: This chatbot is known as the Breach Buster and it can be controlled using the Kali CLI interface and used to run scans, exploit vulnerabilities, and use tools for Pentesting as instructed by the AI.

3. Tool Availability: This is one of the benefits of the Kali WebShell because all the tools that are required for anything that is related to Pentesting are already available on the remote Kali

machine. Different tools do not need to be installed separately and this saves time and helps with the bugs that come from installations and dependencies.

3.7.4 Advantages of Kali WebShell

The Kali WebShell feature offers several significant advantages over traditional methods of using Kali Linux for Pentesting:

1. Ease of Access: Kali Webshell is a web-based Kali Linux framework used to make the Kali Linux available to an individual through remote access eliminating the need for downloading and configuring Kali Linux locally. This is of great benefit to the user in the sense that it becomes easy for the user to carry out Pentesting activities.
2. Preinstalled Tools: WebShell comes with the Kali machine that comes with a number of preinstalled penetration testing tools. This ensures that users can get a ready to use isolated set of tools to use without cumbersome steps involved in installing, configuring, and even depending on the execution of the application.
3. Seamless Integrat**ion**: You can use Breach Buster which is an AI Pentesting assistant & Kali Webshell together effectively. Users of manual Kali will have to be instructed by the chatbot while at the same time enter commands on the Kali CLI interface.
4. Efficiency: The features such as Kali WebShell mean that one does not have to install the tools in the system or any further setting of the system, thus saving time that is used to undertake the said activities. It is related to the fact that they are no longer thinking about system configuration as well as managing testing tools as it is required in the actual testing process.

3.7.5 Conclusion

Another very useful tool added to the Breach Buster AI Pentesting Assistant is the Kali WebShell feature that enables the users to penetrate web servers in an efficient and practical manner. Its robust, easy-to-use interface, efficiency, and procreated asset platform enable the pentester to apply efficient tools without frustration. When using the Kali WebShell users are able to do all activities for Pentesting in one application instead of having to switch between various Pentesting tools and able to identify any weak areas in web servers.

3.8 Account Management

The Account Management feature is a critical component of the Breach Buster AI Pentesting Assistant, ensuring secure and controlled access to the system. This section outlines the functionality, user roles, interface design, and operational workflow of the Account Management feature, highlighting its significance in maintaining the integrity and security of the platform.

3.8.1 Functionality

Account Management is a right management mechanism Task that helps the system maintain registration, approval, and removal of users as a systematic way of controlling access. It supports two types of user privileges: administrator and other customers that have different tasks or functionalities.

3.8.2 User Roles

1. Admin: The nominal users may have limited power as compared to an admin.

They both have rights, but the rights given to an admin are high. Amongst such tasks that are

paramount when performing this role are the verification of the new users who would want to register and the deactivation of users. All users in the system are managed by the one administrative account that the system provides.

2. Regular User: Normal user has lesser right to access functions compared to admin user. Both be available to those who register accounts of the Breach Buster AI Pentesting Assistant Market.

3.8.3 Interface Design

The interface design of the Account Management feature includes components for user registration, admin approval, and user removal: Account Management feature's user interface includes the provision of user enrollment site, admin site, and deleted user site.

1. Registration Page: Now that Registered users access the registration page and complete the registration process. This page consists of the fields that one has to key in one's name, email address, and password. As a further form of validation, the request for registration is forwarded to the admin for approval.

2. Admin Dashboard: The page Users implies the special site page for the admin to manage the user accounts. It displays if a user enrolls and if admin have to register for enrollment. This dashboard also has the current users and the options existing to delete any user from the list.

3.8.4 Operational Workflow

The operational workflow of the Account Management feature involves the following steps:

1. User Registration:

A customer opts to have an account and inputs true details.

The previous information is thus the form of the request created by a user who wishes to register, and can is relayed to the administrator for the verification process.

2. Admin Approval:

In the web interface of the panel the admin observes the list of customers who are willing to register.

He has to check and accept each proposal or reject it. Only an after-approval request can be made for the prevalent users, and they will gain access to the Breach Buster AI Assistant for Pentesting After the decision is communicated to the other users.

3. User Access:

After these approval forms have been filled out, users only need to log in and operate the chatbot to execute Pentesting on problems that have been resolved.

4. User Removal:

Admin can remove users through the use of the admin console.

3.8.5 Importance of Account Management

In this case, it is the Account Management feature that assists in preserving the secure condition and impact of the Breach Buster AI Pentesting Assistant. Key benefits include:
1. Controlled Access: Another way by which it promotes privacy is by limiting registration of users in the system because only users who are approved can be allowed to submit details to the system.
2. Centralized User Management: The ease of doing so as all the user accounts centrally located, and so administrating users can be done with much ease and also the level of administration is

improved.

3. Security: The feature of deleting users is necessary for the admin in the case when any inappropriate incidents which threaten other users, or the system as a whole begin to occur, and they need to be tackled.

3.8.6 Conclusion

The provision of the account management as a component of the Breach Buster AI Pentesting Assistant enables the user to manage what and how people access the system securely. In providing the user registration form for the membership by the structured sign up, the admin approval, and the user deletion it provides good structure for the management of users. It not only enhances the security and safety of the platform but also contributes to the familiarity and convenience of a well-organized and streamlined interface. A lot should be understood about the feature known as Account Management as it deals heavily with managing the account protecting the information and may be said to be a very secure and well-organized account.

3.9 Graphical User Interface (GUI)

Breach Buster AI Pen testing Assistant is penetration testing application which has been designed to provide user easy to perform Pentesting tasks on an America graphical user interface. The following section provides a detailed summary of the GUI interface, for instance how its structured and displaying all of the main GUI components and the task executed by each unit.

3.9.1 Layout and Design

The GUI is designed in a way that it is user friendly and makes it possible for any user to identify the various functions in the program as well as perform the tasks in a more comfortable way. The

layout is divided into distinct sections, each serving a specific purpose: All the required parts are separated by the layout into separate sections to perform particular tasks.

1. Left Panel: This section is made up of the ‘Load Chats’ button and the history of the previous chat sessions. In the end the list posts are saved with timestamps and easy to reach for users to look up the past interactions.

2. Top Right Corner: Underneath the center on the right and upper side there are two crucial buttons.

Launch Kali: If pressed then it will launch a new tab with the Kali Linux CLI interface as described in the Kali WebShell section.

Logout: This button can be used by an admin or even by the user to sign out from his or her account.

3. Right Panel: Within this section, section 4 uses a dark shade of blue to indicate this stage following red section 3, section blue section 2 and section white section 1. The stages are:

Scanning and Reconnaissance: These include the first stage which involves needs analysis.

Enumeration: Nmap scanning at the level of known open ports which are used by servers and possible entry points when hacking.

Exploitation: While it aims to strike at the found weaknesses.

Documentation: Storage and information to the users.

Each of the phases is designed in a color, and as the user proceeds to a stage the whole interface switches to one of the outlined color automatically. Visual representation that helps the user understand the current stage of the Pentesting process.

4. Lower Right Section: The highlighted working buttons on the buttons section in this section include three buttons.

Download Report: Pentesting users can also use the system to access information about the vulnerabilities found, the exploits reported, as well as respective CVEs once total Pentesting was completed.

Save Chat: Later lets a person archive the current conversation and place it into the previously discussed chats in the sidebar of the left side of the screen.

New Chat: This button launches a new chat session and is pressed to create a new dialogue with a new end user interaction and introduce the end user to the chatbot.

3.9.2 Functional Overview

The GUI helps in outlining complete and structured way in which penetration testing can be conducted. Each aspect has the capacity of bringing some sort of convenience or practicality concerning the procedure in the development and completion of the Pentesting project.

1. Load Chats: With the help of this, users may record the sessions they have gone through and make a reference for them to recheck any of their past activities and the results it might have. It also provides the service guarantee of keeping the option of users to keep their pen testing activities going.

2. Launch Kali: It is an added advantage among other features to the pentesters as they do not have to install Kali Linux and stress themselves with the configurations; rather they just have a hand of one simple touch on the browser.

3. Logout: Promotes security by facilitating the exiting of an authenticated individual in a secured way while making it possible for the authenticated to check for the existence of something or even a threat to the security before exiting the system.

4. Pentesting Stages: The color codes through stages enable a user to follow the standard

procedure of Pentesting and these stages are complete but not more effective. Same colors change with time to help the user navigate the program or demonstrate progress of something.

5. Download Report: A must for documentation and reporting purposes to document investigation task and to create comprehensive and precise summaries of Pentesting tasks both for internal use and for customer feedback.

6. Save Chat: A feature that guarantees that the chat function becomes user friendly by creating means of storing of important conversations and retrieving such conversations at the right time.

7. New Chat: Dedicates users' time directly to the content of the chatbot without the need to input the context for the new session and avoids repeating the information that was entered in the previous session; therefore, it presents a major improvement compared with the original chatbot.

3.9.3 Conclusion

What you will also realize is that the GUI of the Breach Buster AI Pentesting Assistant has been designed in an attractive way for the Pentesters at a beginner level as well as the professional Pentesters. Therefore, it refers to the natural flow of the laid-out application with some extra utilities like Load Chats, Launch Kali, and the color-coded Pentesting stages to make the application acceptable for the user. Integration of these elements into a user-friendly interface contributes to the fundamental fulfilment of the market demand for AI based Pentesting through Breach Buster AI Pentesting Assistant.

3.10 Target Selection

Target Selection is among the function of the Breach Buster AI – Pentesting Assistant in the process of aligning on the right penetration testing target. Target Selection process follows the

processes as discussed in section on what it is and the big picture of workflow as it relates to all Pentesting activities.

3.10.1 Functionality

The Target Selection functionality is used to specify the target URL from where the user wants to launch the Pentesting. This is an efficacy which helps to identify the specified target and its appropriateness for the purpose of testing before further course of action. This process involves invoking an HTTP gateway to pass an HTTP request to the target URL and retrieving the response status message from the HTTP gateway.

3.10.2 Workflow

The operational workflow of the Target Selection feature includes the following steps:

1. User Input:

The user begins with entering the required or intended URL in the specified input field. For example, one can apply a block for the domain name as www.nust.edu.pk for their penetration testing operations.

2. Backend Validation:

This will include the HTTP message that is sent upon submission and is by an HTTP request made to the specified target URL on the backend. This request is made in order to try and establish if the target is available and online.

3. Response Evaluation:

It also ensures that request to the target URL returns the right HTTP response code. The following criteria are used to determine the target's status:

Live Target: A number inside the status code of the response represents a code of status of target according to the RFC – if the code is inside the range from 100 to 399 it means that the target is up and may be tested. After the duck testing stage, the user is therefore guided as to what is required of him/her.

Inactive or Inappropriate Target: The status code of the response can also help in identifying the status of the target as the indicated status that indicates that the target is an active and a suitable target for Pentesting and anything above the range of 1399 means that the target is inactive. In this case, the system prompts the user with a message: “Target set for now is not suitable for an arrest, do you want to continue?”

4. User Decision:

This is based on the system’s prompt message that immediately follows query if the user is willing to proceed with the Pentesting session for the flagged target in spite of its unsuitability. This is good for the usability test because it provides the room for a use case where the target may not retract completely or may only revert according to some reasons.

Significance

The Target Selection feature is crucial for several reasons:

i. Initial Validation: Confirming the live and responsiveness of the target is an initial prerequisite of every pentest. It helps the users from wasting valuable time and effort pursuing the wrong targets or unfocused initiatives.

ii. User Guidance: Using such a system, users receive real-time information on the target's health, which is essential to make a decision. This is especially helpful as some of the codes in Pentesting require understanding of HTTP response codes and most newbies lack that.

iii. Flexibility: The system identifies inappropriate targets, but it also provides options to the users if they wish to proceed in order to save time and lighten their workload. This feature is useful because it recognizes that there may be reasons that the target does not respond conventionally legitimate explanation for test targets that do not respond conventionally.

iv. Efficiency: Automation of the target validation procedure significantly improves and simplifies workflow by relieving users from the burden of manually checking the targets set for penetration tests.

3.10.2 Conclusion

The ability to set a target for running penetration testing activities and validate the operations performed is the key feature of the Target Selection trait of the Breach Buster AI Pentesting Assistant. But by automating the target verification process and giving real time feedback to the user it improves the efficiency of penetration testing by making the Pentesting process more accurate and its interface more effective. This feature does not only keep the users from confusing live and appropriate targets but also allows irregular testing to take place which can be regarded as a strong and flexible tool for the system.

Chapter 4: Working and GUI

4.1 Technical Working Diagram

4.1.1 Block Diagram

A block diagram is a great way to visualize the high-level architecture of the BreachBuster project. By identifying the main components and their interactions, we created a clear and concise representation of the system that is easy to understand and communicate to others.

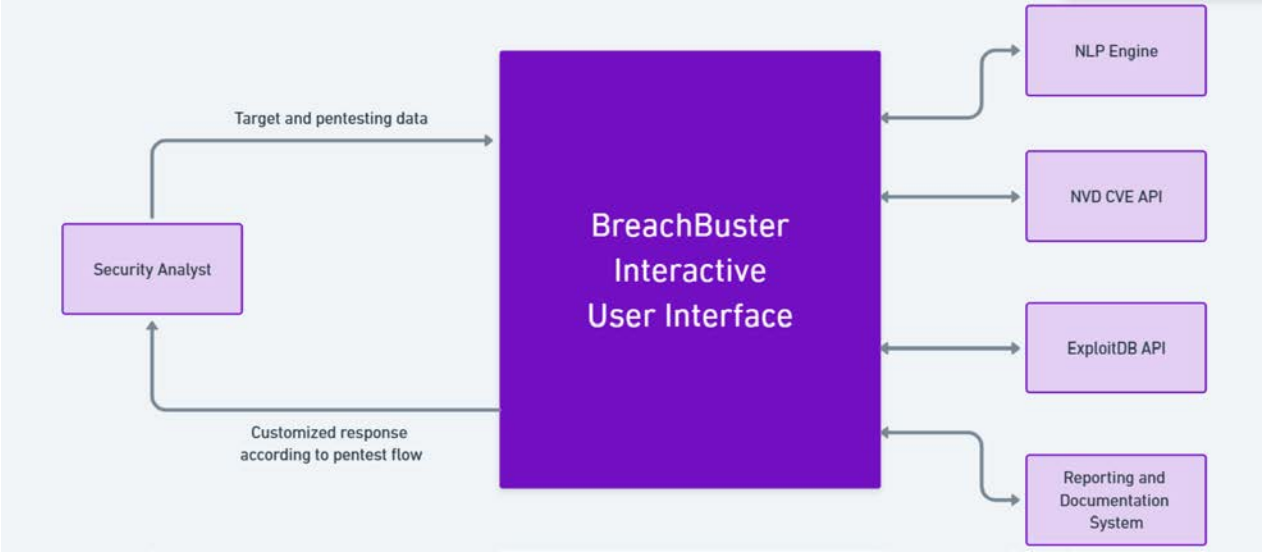

Figure 1 – Block Diagram of BreachBuster

4.1.2 Architectural Diagram

Following is the use architectural diagram based on functional requirements:

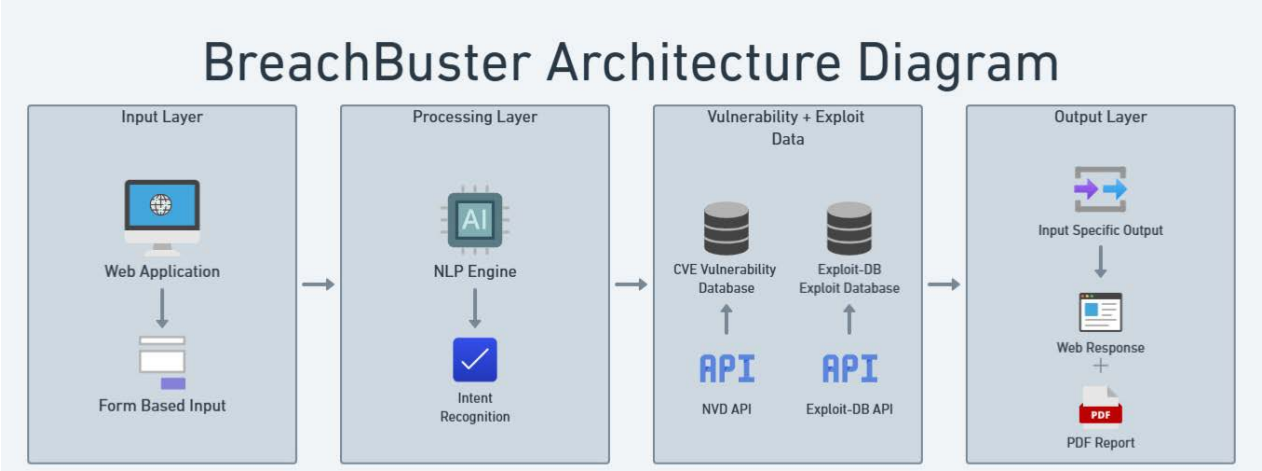

Figure 2 - Architectural Diagram of BreachBuster

4.1.3 Use Case Diagram

Following is the use case diagram based on functional requirements:

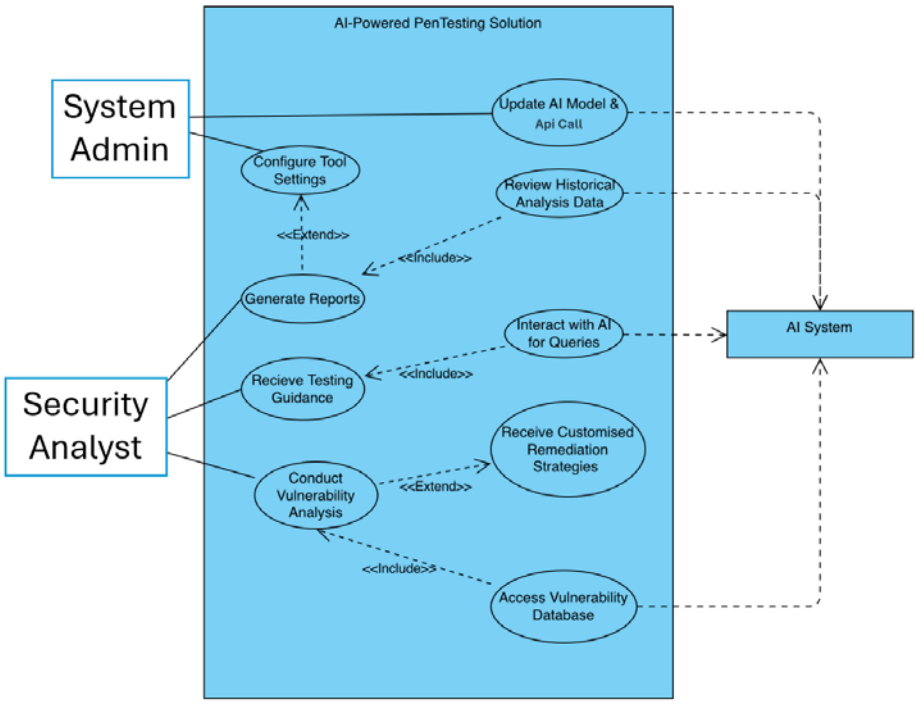

Figure 3 – Use Case Diagram of BreachBuster

The technical part of the BreachBuster chatbot, the process flow and the GUI features will be covered in this chapter. The GUI is designed to be simple and user-friendly to allow users to easily move from one phase of penetration testing (Pentesting) to another. Attached are the images providing UI elements and their functionality breakdown.

4.2 UI

BreachBuster has a user interface that is split into its various components to make the Pentesting process easier. Each of these sections will be described in more detail, including the top bar, sidebars, main window of the chat, and buttons.

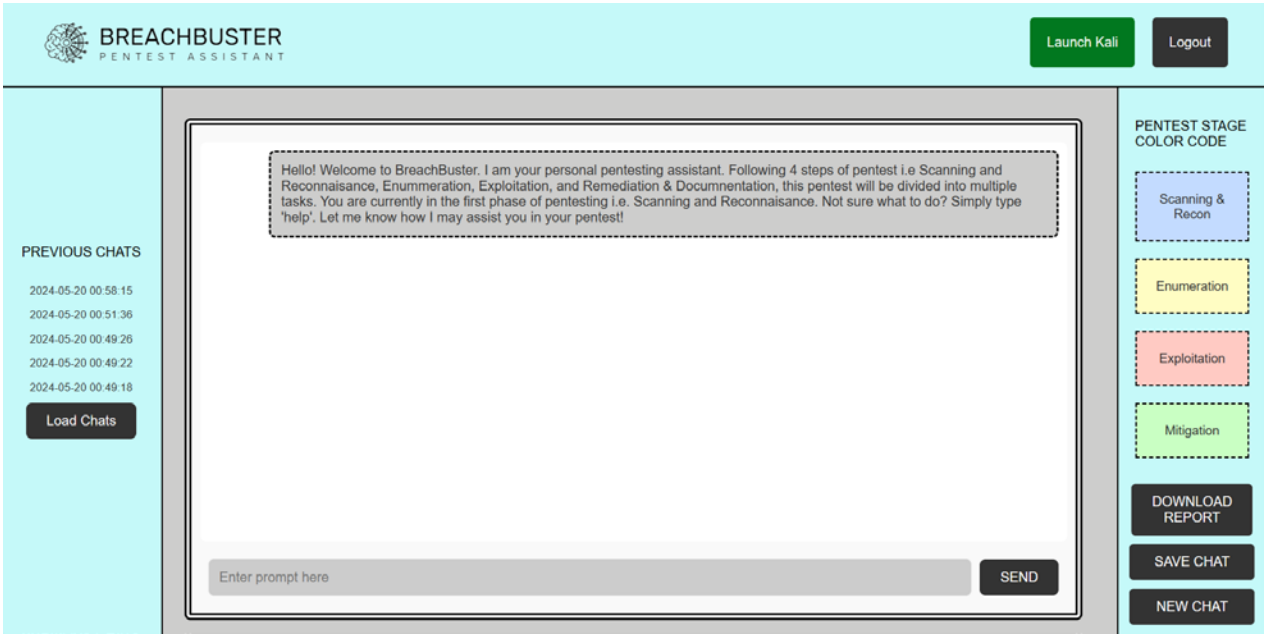

Figure 4– UI Overview

4.2.1 Top Bar

The top bar, positioned at the top of the page and spans the entire width of the screen.

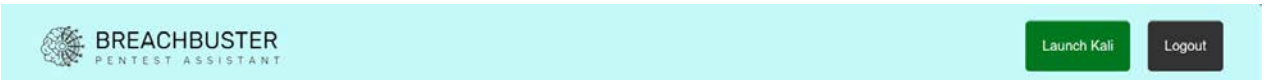

Figure 5 – UI Top/Title Bar

It includes the following elements:

- Logo: Next to the left side of the top bar is the BreachBuster logo. This logo not only serves as a visual signification of the application but also help in building the credibility of the application. Gives users confidence that they are engaging in the BreachBuster site.
- Launch Kali Button: This webshell button is located right below the top bar and it opens the webshell in a new page. This feature is especially useful for expert users who want to use the Kali Linux tools directly for more rigorous jobs. It enhances efficiency by bundling these features in the software to be incorporated in the BreachBuster software instead of having to use different applications.
- Logout Button: Located beside the Launch Kali button is this button that logs out the user of the account. This feature guarantees session security and security against unauthorized interference and continued Pentesting. The ability to log out with a single click further increases the user's confidence in the platform's safety.

4.2.2 Left Sidebar

The left sidebar contains elements that help users manage their chats and view previously saved sessions.

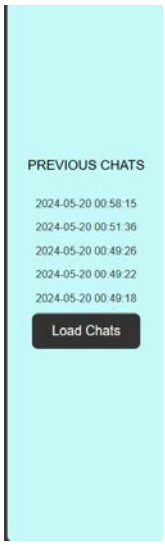

Figure 6 – Left Sidebar Showing Previous Chats

It includes:

- Load Chats Button: This button is placed at the top of the sidebar and is used by users to load previous chat history. This feature is important in ensuring that Pentesting processes continue uninterrupted, and users can go back and forth from previous sessions without losing their work.
- Previous Chats List: There is a list of timestamps under the Load Chats button that store previous chats. Every entry displayed has the date and time saved of the specific session and this makes it easier to keep record of the past activities. These timestamps allow users to click and load specific chats that would be helpful in keeping previous work contentions.

4.2.3 Right Sidebar

The right sidebar provides options related to the Pentesting stages and report generation.

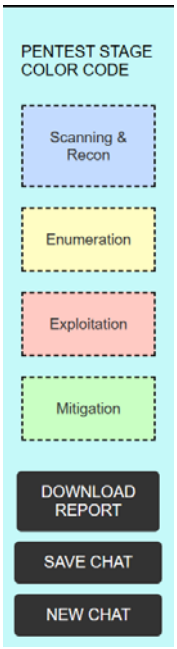

Figure 7 – Right Sidebar Showing Pentesting Stage Colors and Buttons

It includes:

- Pentest Stage Color Code: This section displays the color coding for different stages of the Pentesting process. Each stage has a distinct color, aiding in visual recognition and quick identification of the current phase:
    - Scanning & Recon: Light Blue
    - Enumeration: Light Yellow
    - Exploitation: Light Red
    - Documentation/Mitigation: Light Green
- The use of the various color codes makes navigation easy and makes users to always understand where they are in the process of Pentesting.

- New Chat Button: This button loads a new page, allowing users to start a fresh chat session. It allows us to initiate a new Pentesting chat session without closing the previous session.
- Save Chat Button: This button saves the current chat session to the database. Saving chats also helps in keeping records of the interactions and findings that can be used for later reference purposes. These saved conversations can be accessed from the Previous Chats tab in the left side of the application.
- Download Report Button: This button is located at the bottom of the sidebar, and it generates a PDF document report for the current Pentesting session. This feature is crucial for documenting information about the Pentesting process, and the results derived. The capability to download a report is crucial for compliance and auditing purposes.

4.2.4 Main Chat Window

The main chat window is the central part of the UI, where the interaction between the user and the BreachBuster chatbot takes place.

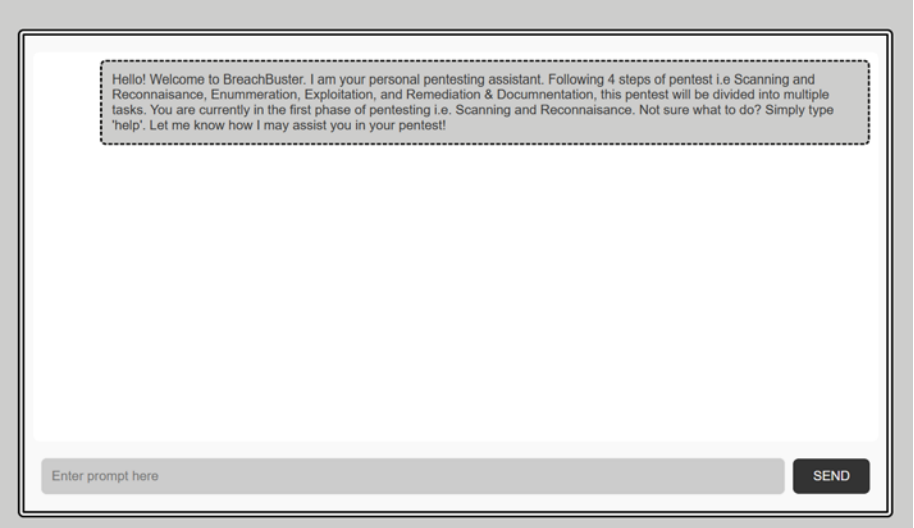

Figure 8 – Main Chat Window

It includes:

- Prompt Input Box: It is positioned at the end of the interface and enables users to type and send queries to the chatbot. The input box has a logical design and incorporates a concise “SEND” button for requests. Its ease of use means that even an inexperienced user can operate the chatbot.
- Chat Display Area: This area shows the flow of the interaction between the user and the chatbot. The dialogue box format is used for each message with the user inputs on the left side and chatbot reply on the right side. This allows for a clean separation of dialogue and improves readability because the users are able to follow the flow of conversations better.

4.2.5 UI Color Scheme

The UI color scheme changes dynamically based on the current stage of the Pentesting process, providing a visual cue to the user. The stages and their corresponding colors are:

- Scanning & Recon: UI background is light blue suggesting the first stage reconnaissance is informative. This color has a soothing effect that makes the user concentrate on the accuracy and careful work needed in this phase.

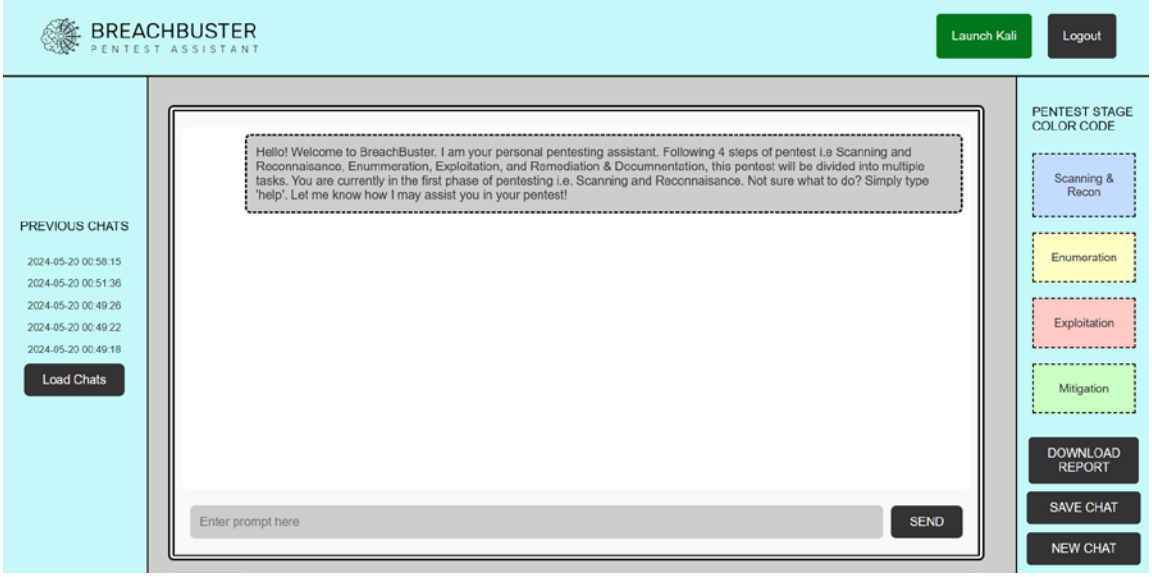

Figure 9 – Blue Color for Scanning Phase

- Enumeration: Background color changes to light yellow, indicates the enumeration stage where the background information of the target is obtained. The color makes people aware, and this is important when they need to analyze and organize data.

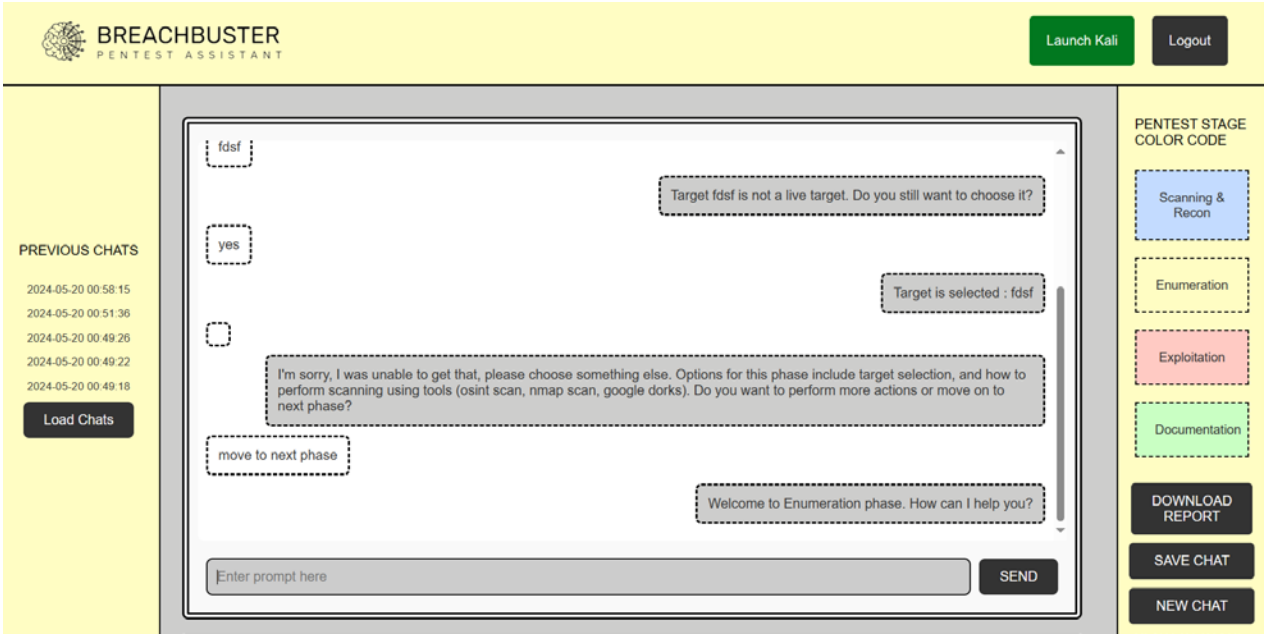

Figure 10 – Light Yellow Color for Enumeration Phase

- Exploitation: During the exploitation phase the background turns light red it means that this is when the vulnerabilities are being tested using exploits. The red color conveys the message that this stage is very important and highlights the risky nature of exploits.

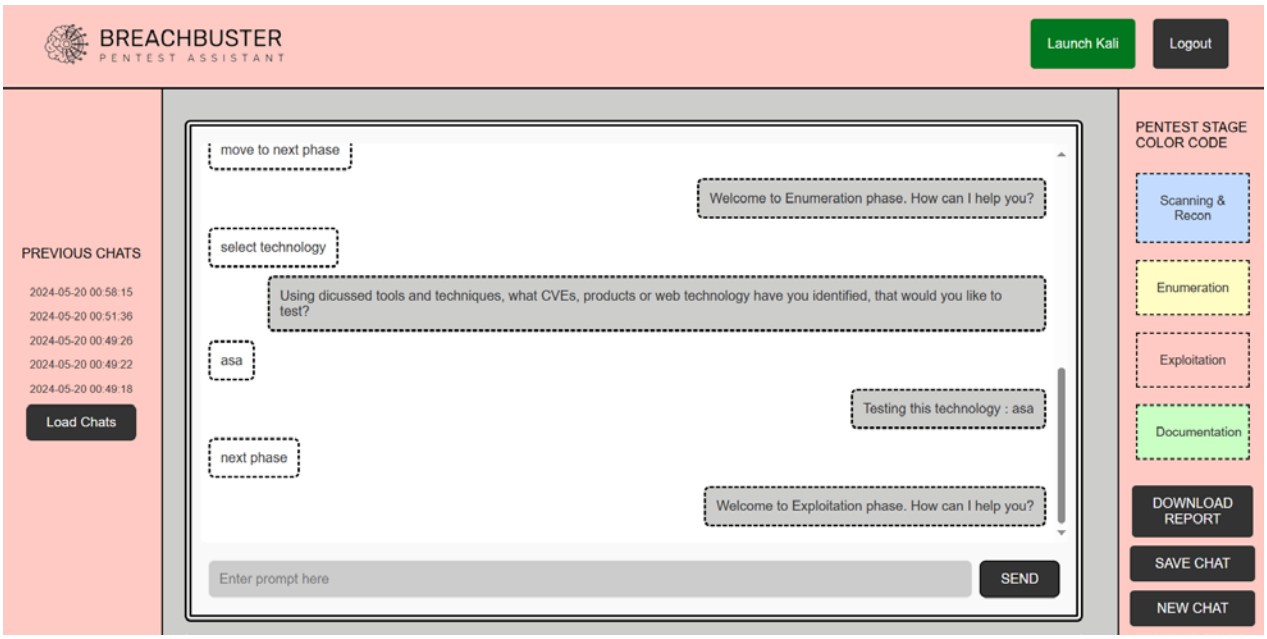

Figure 11 – Red Color for Exploitation Phase

- Documentation/Mitigation: The final stage has a light green background to represent the last phase of the presentation process when findings are recorded, and reports are produced. The green color stands completion and readiness and therefore makes the viewer feel accomplished and satisfied.

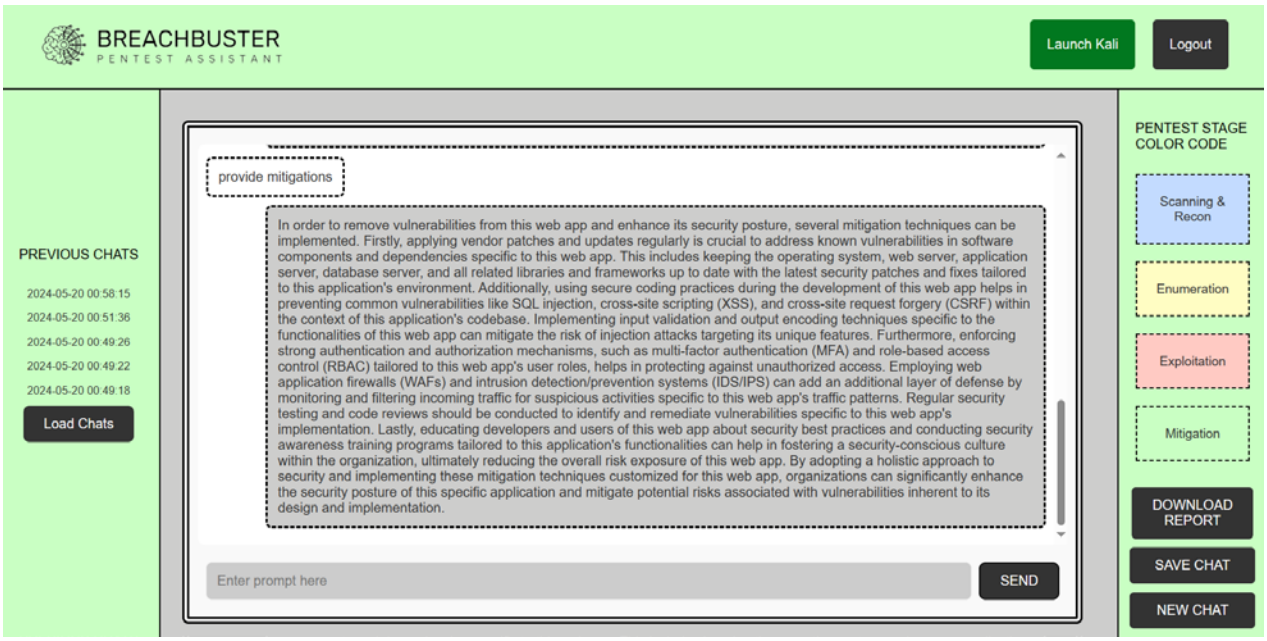

Figure 12 – Green Color for Documentation Phase

These color changes enable the users to easily identify the phase of their current Pentesting process and thus allows them to maintain an organized workflow. In addition, the color changing feature adds a psychological aspect to the interface that makes navigation easier and faster.

4.3 Kali Webshell

Kali Webshell is one of the main components of the BreachBuster toolkit which allows the user to work with a functional web-based shell that enables carrying out various penetration testing activities through the Web. This section will aim at breaking down the products of Kali Webshell further to highlight the description of each component, the reasons for the design decisions made, as well as the utility of the whole to the user, especially with regard to its ease of use and

enhanced functionality.

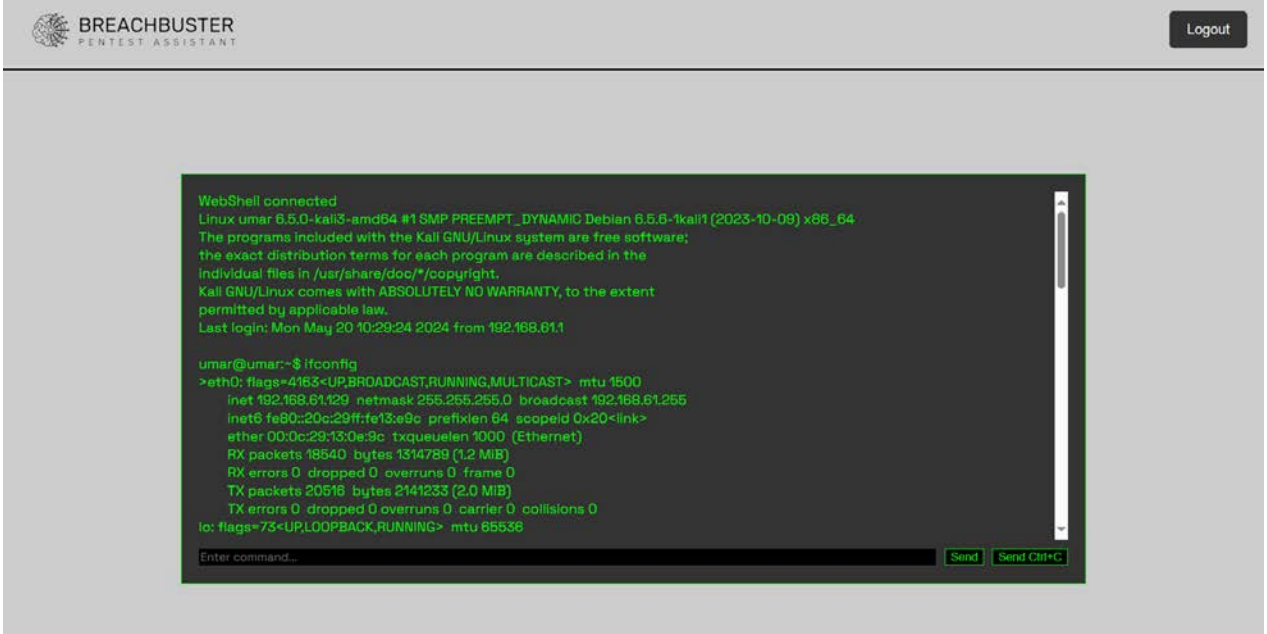

Figure 13 - Kali Webshell Layout

4.3.1 Overview of the Kali Webshell

The Kali Webshell is an in-browser penetration testing shell developed in the BreachBuster application that lets users access the full power of Kali Linux over a browser. Kali Linux remains one of the most popular distros in the hacking field, with a host of tools for doing penetration testing, security assessment, and hacking in general. The webshell used in BreachBuster allows users to easily access these tools and does not involve switching between different environments or installing additional software on their personal computers.

4.3.2 Components of the Kali Webshell

The Kali Webshell within BreachBuster consists of several key components, each designed to enhance the user experience and streamline the penetration testing process:

- Interactive Terminal Interface: The main part is a terminal that allows running commands in the emulation of a general shell of the operating system of Kali Linux. The webshell has a command input where users can run commands and execute scripts from the webshell directly.
- Tool Integration: The webshell offers access to a number of critical tools that are usually required for penetration tests. Some of these are Nmap, Metasploit, Burp Suite, Wireshark among others. The inclusion of these tools in the webshell means that the users have the option of having everything that they need to use within one interface.

4.3.3 Design Choices and Benefits

The design of the Kali Webshell is centered around accessibility, usability, and security. Here are the key design choices and their corresponding benefits:

- Web based Access: It provides the Kali shell through a browser and enables the user to use it from any device and to use all the power of the Kali penetration testing tools. This will eliminate the problem of complex installations and allows the users to work from different places which also helps in improving the mobility factor.
- User-friendly Interface: Those who are not versed with command line operations will find the application terminal very user-friendly. Commands' history and autocomplete are familiar examples that help lower the barrier and minimize time waste due to misunderstanding commands.
- Integrated Security: The web shell is added inside within the outmost security of the BreachBuster environment. This part of the integration helps in achieving secure

interaction and confidentiality of user data. The platform's security features like session management and access control systems reduce risk of unauthorized access and proliferation for abuse.

- Efficient Workflow: This is important as the webshell organizes the tasks that help in penetration testing by allowing users to access the Kali Linux tools in the same interface used for navigation. Users do not need additional applications to switch between various activities in the application and therefore eliminate multiple factors of friction and optimizing the process.
- Comprehensive Toolset: The broad spectrum of preinstalled tools also entails the fact that those functionalities are readily at hand. This is a complete set of tools to use for penetration testing and contains modules for scanning and enumeration, exploit and post exploit, as well as reporting.

4.3.4 How the Kali Webshell Eases Interactions

The Kali Webshell significantly eases interactions during the penetration testing process through several key features:

- Simplified Access to Tools: BreachBuster makes it possible for a user to run advanced commands as well as heavy-duty tools without having to leave the BreachBuster environment. This integration will allow for less applications switching connected with a narrower focus on user's task.
- Realtime Feedback and Results: Webshell's highly interactive allows the user to get the feedback of the commands or scripts being entered immediately. This real-time

communication is needed during the practical testing of the websites, as it often requires quick changes.

- Consistent Environment: One of the features of the webshell is that it standardizes the system in use while providing service to a user hence minimizing incongruousness in systems that a user works on. I think this is very important in order to maintain the reliability and replicability of penetration testing.
- Enhanced Collaboration: The webshell helps the members of the team to work in a shared environment so that any changes and outputs of commands are easily accessible. The session management and chat features enable communicating improvements for this collaborative capability.

In summary, the Kali Webshell within BreachBuster is a powerful and versatile tool that enhances the penetration testing process by providing seamless, web-based access to the extensive capabilities of Kali Linux. Its thoughtful design and integration within the BreachBuster platform make it an invaluable resource for both novice and experienced penetration testers, streamlining workflows, improving efficiency, and ensuring a secure and consistent testing environment.

4.4 Intent Detection Model

The intent model in BreachBuster is designed to classify user inputs into predefined categories or intents, enabling the application to understand and respond appropriately to user queries. This section describes the specifics of the new intent model, its structure choices, elements, it to improve user experience.

4.4.1 Overview of the Intent Model

The intent model employs the use of NLP for preprocessing of the user's inputs, generation of labelled datasets, training of a classification model, and inferring the intent of a new user query. The model uses the NLTK library for text processing, the TFIDF vectorizer for feature extraction, and the Linear Support Vector Classifier (SVC) for classification.

4.4.2 Components and Workflow

The intent model consists of several key components and follows a systematic workflow:

1. Text Preprocessing:
    - Tokenization: Splits text into individual words (tokens).
    - Lowercasing: Converts all characters to lowercase to ensure uniformity.
    - Lemmatization: Reduces words to their base or root form.
    - Filtering: Removes nonalphabetic tokens to clean the text.
2. Training Data Loading:
    - CSV File Reading: Reads labeled training data from a CSV file, where each row contains a user query and its corresponding intent.
3. Feature Extraction:
    - TFIDF Vectorization: Converts text data into numerical features using Term Frequency Inverse Document Frequency (TFIDF), capturing the importance of words in the context of the entire dataset.

4. Model Training:

    - Pipeline Creation: Combines the TFIDF vectorizer and the Linear SVC classifier into a single pipeline.
    - Model Fitting: Trains the pipeline on the preprocessed training data.

5. Intent Prediction:

    - Input Preprocessing: Applies the same preprocessing steps to new user inputs.
    - Decision Function: Computes scores for each intent using the trained model.
    - Probability Calculation: Converts scores into probabilities using the SoftMax function.
    - Intent Selection: Identifies the intent with the highest probability as the predicted intent.

4.4.3 Detailed Explanation of the Code

Let's break down the code to understand each component and its role in the intent model.

```
nltk.download('punkt')
nltk.download('wordnet')

# Function to preprocess text
def preprocess(text):
    lemmatizer = WordNetLemmatizer()
    tokens = word_tokenize(text.lower())
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in tokens if token.isalpha()]
    return ' '.join(lemmatized_tokens)
```

Figure 14 – Downloading NLTK library Resources for Intent Models

- nltk.download: Downloads necessary NLTK resources for tokenization and lemmatization.

- preprocess: This function lowers the case of the text, tokenizes it into words, lemmatizes each word, and filters out nonalphabetic tokens. It returns a clean, processed string.

1. Loading Training Data:

```
# Load training data
def load_training_data(file_path):
    training_data = []
    with open(file_path, 'r', newline='', encoding='utf-8') as csvfile:
        csv_reader = csv.reader(csvfile)
        for row in csv_reader:
            training_data.append((row[0], row[1]))
    return training_data
```

Figure 15 – Function to load Training Data

- load_training_data: Reads the training data from a CSV file and returns it as a list of (text, intent) tuples.

2. Defining the Training Data File Path:

```
# Define the file path for the training data CSV file
training_data_file = 'intents.csv'
```

Figure 16 – Specifying Training Data File

- Specifies the file path for the training data.

2. Generating or Loading Training Data:

```
# Generate or load training data
try:
    training_data = load_training_data(training_data_file)
except FileNotFoundError:
    print("Unable to open training data file")
```

Figure 17 – Loading Training Data File

- Attempts to load the training data. If the file is not found, it prints an error message.

2. Preprocessing Training Data:

```
# Preprocess training data
X_train = [preprocess(text) for text, intent in training_data]
y_train = [intent for text, intent in training_data]
```

Figure 18 – Training Data Preprocessing Before Model Training

- X_train: Contains the preprocessed user queries.
- y_train: Contains the corresponding intents.

2. Creating the Pipeline:

```
# Create pipeline
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer(ngram_range=(1, 2), max_df=0.9, min_df=2)),
    ('clf', LinearSVC(C=0.5)),
])
```

Figure 19 – Define Pipeline features

- Pipeline: Combines the TFIDF vectorizer and the Linear SVC classifier.

- TFIDFVectorizer: Converts text to numerical features, considering unigrams and bigrams, and filters terms based on document frequency.
- LinearSVC: A linear support vector classifier with a regularization parameter C.

2. Training the Model:

```
# Train the model
pipeline.fit(X_train, y_train)
```

Figure 20 – Model Training

- fit: Trains the pipeline on the preprocessed training data.

2. Intent Prediction Function:

```
# Function to test user input against the model and print all intents with matching percentages
def test_user_intent(user_input):
    preprocessed_input = preprocess(user_input)
    intent_scores = pipeline.decision_function([preprocessed_input])[0]
    exp_scores = np.exp(intent_scores - np.max(intent_scores))  # Avoid overflow
    intent_probabilities = exp_scores / np.sum(exp_scores)
    intents = pipeline.classes_

    max_index = np.argmax(intent_probabilities)
    predicted_intent = intents[max_index]
    return predicted_intent
```

Figure 21 – Function Predicts Intent Based on Sentence Features

- test_user_intent: This function processes new user input, calculates intent scores using the decision function, converts scores to probabilities using SoftMax, and identifies the intent with the highest probability.

4.4.4 Benefits and Ease of Interaction

The intent model offers several benefits that enhance user interaction with the BreachBuster platform:

- Accurate Intent Recognition: Through textual preprocessing, and employment of an effective classifier to execute the classification of the input into possible intents, the intent model guarantees that user queries are correctly identified to ensure that the right answers are provided.
- Efficient Text Processing: Lemmatization and filtering eliminate all the unnecessary data and thus enhances both the speed and accuracy of the model.
- Realtime Predictions: The model is able to analyze and predict the user's input intents in real-time and give instantaneous feedback to the user for their action.
- Scalability: The use of TFIDF and Linear SVC allows the model to handle large datasets and complex queries efficiently.
- User-friendly Interface: The intent model when integrated into the BreachBuster platform gives the user an access to a high-end system where they can communicate with the system accurately and efficiently.

4.4.4 Sequence Diagram

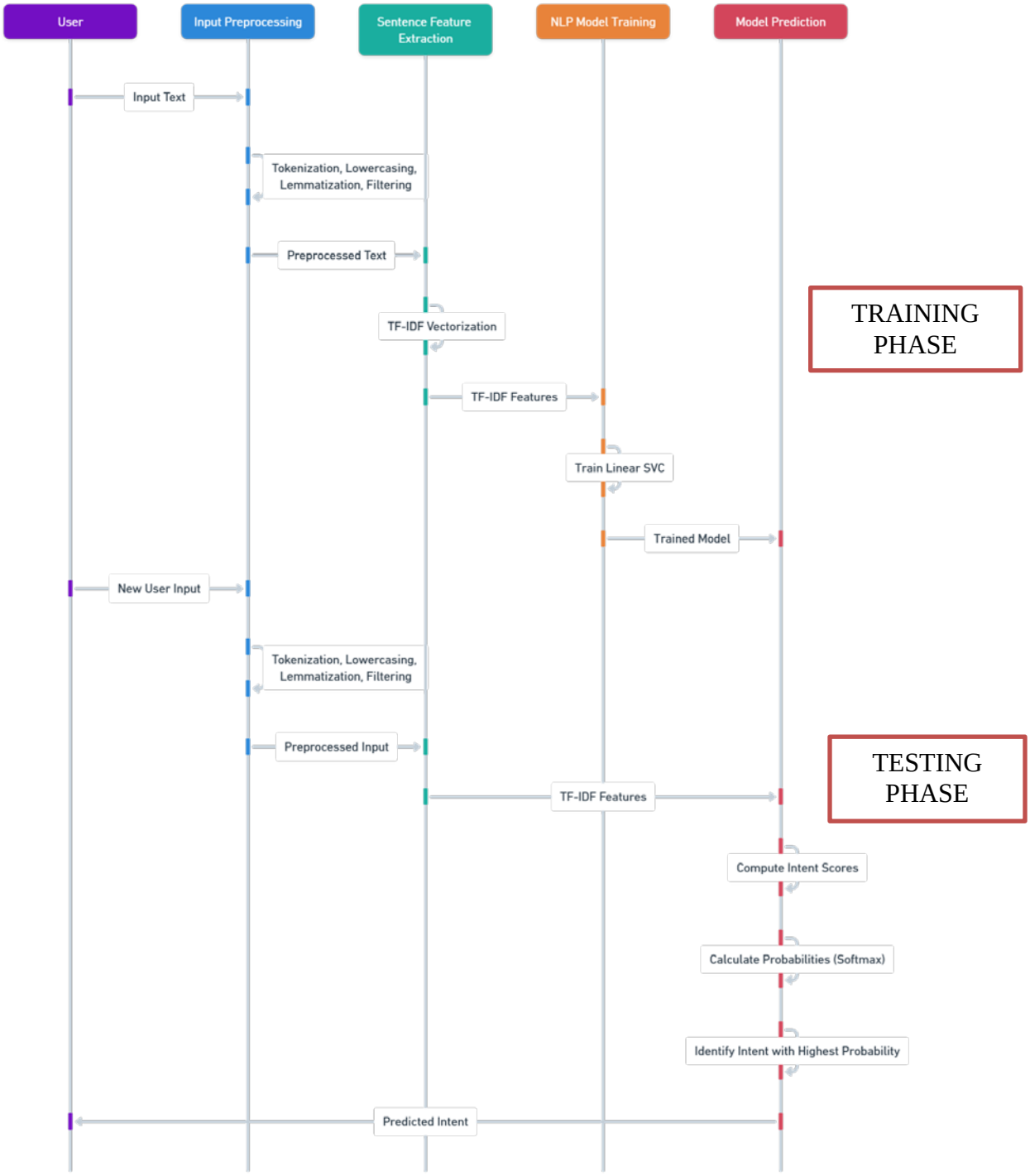

Figure 22 – Intent Model High Level Overview

4.5 Chatbot Working Procedure

This section explains the working sequence of the BreachBuster Project; a Chatbot Application for PT. The process is divided into four main phases: scanning and foot printing, mapping, exploitation and post exploitation, and documentation. In each phase, there are certain activities and interaction with the users utilizing the chatbot, aiding in the efficient completion and providing documentation of activities of the penetration test.

4.5.1 Phase 1: Reconnaissance and Scanning

The first stage incorporates research about the target system which is useful for planning the next phase of the test. This phase builds on active and passive scanning to determine potential access points and weaknesses.

1. Target Selection:
    - User Input: The user starts by choosing a target which may be website address or a web app running on local environment.
    - Web Request Validation: The chatbot will send a request to the target via a web. When the response status code is between 0 and 399 the target is assessed to be valid. If not, the chatbot asks the user to confirm their target.
2. Active Scanning: A few components are:
    - Network Scanning: The chatbot guides the user through performing network scans using tools like Nmap. It explains how to identify open ports, running services, and other network information.

- OSINT (Open-Source Intelligence): The chatbot provides instructions on using tools and techniques for gathering publicly available information about the target, such as WHOIS lookups and social media profiling.

3. Passive Scanning:

   A few components are:

   - Google Dorking: The chatbot instructs the user on how to use advanced Google search queries to find sensitive information or hidden pages on the target's website.
   - Automated Tools: The chatbot may suggest using tools like Shodan or Censys to passively gather information about the target without directly interacting with it.

4.5.2 Phase 2: Enumeration

Enumeration involves probing the target in greater detail to uncover more specific information that can be used to identify vulnerabilities.

1. Web Application Enumeration:

   - Hidden Directories: The chatbot helps the user to search for hidden files and directories from the web server using the tools such as DirBuster or Gobuster etc.
   - Web Components: The user can recognize and count different web parts like plugins and themes and so on which are building web sites under examine.

2. Technology and Vulnerability Selection:

   - User Input: The user selects a specific technology (e.g., Apache, WordPress) or vulnerability type to focus on.

- NVD CVE API Search: The chatbot uses the NVD (National Vulnerability Database) CVE (Common Vulnerabilities and Exposures) API to search for vulnerabilities related to the selected technology or criteria. It returns a list of relevant CVEs.

3. Detailed Vulnerability Information:

    - The chatbot provides detailed information about each CVE, including descriptions, severity levels, and potential impacts. This helps the user prioritize which vulnerabilities to focus on in the next phase.

At the end of the enumeration phase, the user has identified specific vulnerabilities and gathered detailed information about them, enabling targeted exploitation efforts.

4.5.3 Phase 3: Exploitation

In this phase, the user attempts to exploit the identified vulnerabilities to gain unauthorized access or demonstrate potential impacts.

1. CVE Based Exploit Search:

    - Exploit DB API: The chatbot uses the Exploit DB API to search for exploits related to the CVEs identified in the enumeration phase. It retrieves links to exploit details and POC code.

2. Exploit Code Retrieval:

    - Web Scraping: The chatbot uses web scraping techniques to fetch the actual exploit code from Exploit DB pages. This code can be used to test the vulnerability.

3. Exploitation Execution:

    - Guidance: The chatbot provides step-by-step instructions on how to execute the exploit code against the target. This may include setting up the exploit environment, executing commands, and verifying successful exploitation.

By the end of this phase, the user has attempted to exploit identified vulnerabilities, potentially gaining insights into the security posture of the target system.

4.5.4 Phase 4: Mitigation and Documentation

The final stage involves fixing the vulnerabilities and applying the detailed documentation on the way the penetration testing procedure was carried out.

1. Mitigation Strategies:

    - User Queries: The user can inquire the chatbot for the countermeasures for the threats described above.
    - Recommendations: The chatbot offers specific advice and strategies for addressing each vulnerability and includes advice on patching, configuration changes and various security options.

2. Report Generation:

    - Documentation Request: The user asks the chatbot to provide a detailed penetration test report.
    - Downloadable Report: The chatbot informs the user when the report is prepared and can be downloaded by pressing the download button.

Completing the mitigation and documentation phase helps the user to address the vulnerabilities that have been identified during the pentest process and helps in ensuring that all the required documentation of the pentest process is done for later reference or assure compliance later.

4.5.5 Conclusion

The working sequence of the BreachBuster project ensures a structured and comprehensive approach to penetration testing. By guiding the user through reconnaissance, enumeration, exploitation, and mitigation, the chatbot not only simplifies complex tasks but also enhances the accuracy and efficiency of the testing process. This sequence enables users to perform thorough penetration tests, identify and exploit vulnerabilities, and implement effective mitigation strategies, ultimately improving the security posture of their systems.

Chapter 5: Conclusion

In this thesis, we presented the design, development, and implementation of the BreachBuster project, a chatbot based application aimed at streamlining the penetration testing process. The project is designed to assist users in performing comprehensive penetration testing activities through an interactive and user-friendly interface, leveraging the capabilities of artificial intelligence and automated tools. The BreachBuster project was divided into several key phases, each addressing specific aspects of penetration testing: reconnaissance and scanning, enumeration, exploitation, and mitigation and documentation.

5.1 Summary of Key Contributions

1. Automated Penetration Testing Guidance: The BreachBuster chatbot provides step-by-step instructions and automated tools to guide users through various penetration testing activities, significantly reducing the complexity and expertise required to perform effective penetration tests.
2. Interactive Kali Webshell: An innovative web-based interface was developed to allow users to execute penetration testing commands and scripts in a familiar Kali Linux environment, enhancing accessibility and ease of use.
3. Intent Model for User Interaction: An advanced intent model was implemented to understand user inputs and provide relevant responses, ensuring smooth and efficient interactions between the user and the chatbot.

4. Structured Penetration Testing Phases: The project was organized into four distinct phases, each focusing on specific tasks and objectives, thereby providing a systematic approach to penetration testing.
5. Detailed Reporting and Mitigation: The project includes functionalities for generating detailed penetration testing reports and providing mitigation strategies, ensuring that users can document their findings and take appropriate actions to secure their systems.

5.2 Achievements

The BreachBuster project successfully demonstrated the feasibility of using chatbot technology to automate and simplify penetration testing tasks. By integrating various tools and APIs, such as Nmap for network scanning, the NVD CVE API for vulnerability identification, and Exploit DB for exploit retrieval, the project showcased the potential of combining AI with existing cybersecurity resources to create a comprehensive and efficient penetration testing solution.

5.3 Limitations

While the project achieved significant milestones, it also encountered certain limitations. The current scope is primarily focused on web application vulnerability scanning, which restricts its applicability to other types of systems. Additionally, the project relies heavily on the accuracy and availability of external APIs and databases, which may introduce dependencies and potential points of failure.

Chapter 6: Future Work

To realize the full potential of the BreachBuster project and enhance its commercial viability, several areas of future work have been identified. These enhancements aim to broaden the scope of the project, improve its capabilities, and ensure a more comprehensive and effective penetration testing process.

6.1 Expanding Vulnerability Scanning Domains

The current focus on web application vulnerability scanning should be expanded to include a wider range of vulnerability types, such as:

1. Network Vulnerability Scanning: Integration with tools like Nessus or OpenVAS to identify vulnerabilities in network infrastructure, including routers, switches, and firewalls.
2. Host Vulnerability Scanning: Scanning individual hosts for vulnerabilities using tools like Qualys or Microsoft Baseline Security Analyzer (MBSA).
3. Mobile Application Scanning: Extending capabilities to include the assessment of mobile applications, leveraging tools like MobSF (Mobile Security Framework).

6.2 Integration of Cyber Threat Intelligence (CTI)

Incorporating CTI into the BreachBuster project would enable advanced threat detection and remediation by leveraging real-time threat intelligence feeds and databases. This integration could provide:

1. Threat Intelligence Feeds: Realtime updates on emerging threats, vulnerabilities, and exploits, sourced from trusted CTI providers.
2. Advanced Threat Detection: Utilizing CTI to identify sophisticated threats and attack patterns, enabling proactive defense measures.
3. Automated Remediation: Recommendations and automated actions based on threat intelligence to mitigate identified risks promptly.

6.3 Complete Integration of MITRE ATT&CK Framework

Integrating the MITRE ATT&CK framework would enhance the project's ability to map discovered vulnerabilities and exploitation techniques to known adversary tactics, techniques, and procedures (TTPs), thereby providing a more comprehensive and structured approach to penetration testing.

1. TTP Mapping: Correlating identified vulnerabilities and exploits with the relevant TTPs from the MITRE ATT&CK framework.
2. Attack Simulation: Simulating attack scenarios based on MITRE ATT&CK techniques to assess the effectiveness of security controls and incident response capabilities.
3. Detailed Reporting: Including ATT&CK mappings in penetration testing reports to provide a clearer understanding of potential attack vectors and mitigation strategies.

6.4 Enhanced User Experience and Accessibility

To ensure the widespread adoption and usability of the BreachBuster project, the following enhancements to user experience and accessibility are recommended:

1. Multilingual Support: Expanding language support to cater to a global user base.
2. Mobile Application: Developing a mobile version of the application to allow users to perform penetration testing tasks on the go.
3. User Training and Documentation: Providing comprehensive training materials and documentation to help users understand and effectively utilize the platform.

6.5 Commercialization Strategy

To transition BreachBuster from a research project to a commercially viable product, the following steps should be considered:

1. Market Analysis: Conducting a thorough market analysis to identify potential customers, competitors, and market demands.
2. Business Model Development: Defining a sustainable business model, including pricing strategies, subscription plans, and value-added services.
3. Partnerships and Collaborations: Establishing partnerships with cybersecurity firms, tool providers, and threat intelligence companies to enhance the platform's capabilities and reach.

References:

- Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., & Marchetti, M. (2018). On the effectiveness of machine and deep learning for cybersecurity. In 2018 10th International Conference on Cyber Conflict (CyCon) (pp. 371390). IEEE. Available at: https://ieeexplore.ieee.org/document/8405026/;jsessionid=3DA818FA1F3737B26EFEC78B724FAEA6

- Bejtlich, R. (2013). The Practice of Network Security Monitoring: Understanding Incident Detection and Response. No Starch Press. Available at: https://nostarch.com/nsm

- Bridges, R. A., GlassVanderlan, T. R., Iannacone, M. D., & Vincent, H. L. (2013). Threat detection and classification with semantic analysis of threat reports. In 2013 IEEE Security and Privacy Workshops (pp. 5359). IEEE. Available at: https://www.researchgate.net/publication/326280667_Semantic_integration_of_security_knowledge_sources

- Brown, C., Gommers, J., & Serrano, O. (2018). From cyber security information sharing to threat management. In Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security (pp. 3541). Available at: https://www.semanticscholar.org/paper/From-Cyber-Security-Information-Sharing-to-Threat-Brown-Gommers/8ea77da409b4a5497bb19d1e7fbc962d62b45f80

- Engebretson, P. (2013). The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy. Elsevier. Available at: https://wqreytuk.github.io/Patrick+Engebretson+The+Basics+of+Hacking+and+Penetration+Testing,+Second+Edition+%282013%29.pdf

- Følstad, A., & Brandtzaeg, P. B. (2017). Chatbots and the new world of HCI. Interactions, 24(4), 3842. Available at: https://www.researchgate.net/publication/317920872_Chatbots_and_the_new_world_of_HCI

- Kwon, Donghwoon & Kim, Hyunjoo & Kim, Jinoh & Suh, Sang & Kim, Ikkyun & Kim, Kuinam. (2019). A survey of deep learning-based network anomaly detection. Cluster Computing. 22. 10.1007/s10586-017-1117-8. Available at: https://www.researchgate.net/publication/320066760_A_survey_of_deep_learning-based_network_anomaly_detection

- Ghafir, I., Prenosil, V., Hammoudeh, M., Han, L., & Svoboda, J. (2018). Botnet Command and Control Traffic Detection Challenges: A Machine Learning Approach. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 50405045). IEEE. Available at: https://www.researchgate.net/publication/323973146_Botnet_Command_and_Control_Traffic_Detection_Challenges_A_Correlation_based_Solution

- Herzog, P. (2015). OSSTMM 3—The Open-Source Security Testing Methodology Manual. ISECOM. Available at: https://www.isecom.org/OSSTMM.3.pdf

- Liu, Bing & Xia, Yiyuan & Yu, Philip. (2000). Clustering Through Decision Tree Construction. In Proceedings of the ACM International Conference on Information and Knowledge Management. 10.1145/354756.354775. Available at: https://www.researchgate.net/publication/2805324_Clustering_Through_Decision_Tree_Construction

- NIST. (2008). Technical Guide to Information Security Testing and Assessment (NIST SP 800115). National Institute of Standards and Technology. Available at: https://csrc.nist.gov/pubs/sp/800/115/final
- Padmasiri, Avishka & Herath, Salitha & Ganepola, Vishmi & Vekneswaran, Prathieshna & Ganepola, Nipuna & Welagedara, Lahiru. (2020). Survey on Deep learning-based Network Intrusion Detection and Prevention Systems. Available at: https://www.researchgate.net/publication/346518190_Survey_on_Deep_learning_based_Network_Intrusion_Detection_and_Prevention_Systems
- OWASP. (2017). OWASP Testing Guide v4.0. OWASP Foundation. Available at: https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf
- Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks, 51(12), 34483470. Available at: https://www.sciencedirect.com/science/article/abs/pii/S138912860700062X
- Pereira, J., & Díaz, O. (2019). Using health chatbots for behavior change: A mapping study. Journal of medical systems, 43(5), 110. Available at: https://pubmed.ncbi.nlm.nih.gov/30949846/
- Jay Beale, Haroon Meer, Charl van der Walt, Renaud Deraison (2004). Nessus Network Auditing. Elsevier. Available at: https://shop.elsevier.com/books/nessus-network-auditing/beale/978-1-931836-08-1

- Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). Technical Guide to Information Security Testing and Assessment. NIST Special Publication, 800(115), 115126. Available at: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-115.pdf
- Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In 2010 IEEE Symposium on Security and Privacy (pp. 305316). IEEE. Available at: https://www.semanticscholar.org/paper/Automotive-Network-Protocol-Detection-for-Testing-Sommer-D%C3%BCrrwang/bfd8497c44e0864c6b47cea06a6fd0f68a63b39b
- Stuttard, D., & Pinto, M. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. John Wiley & Sons. Available at: https://edu.anarcho-copy.org/Against%20Security%20-%20Self%20Security/Dafydd%20Stuttard,%20Marcus%20Pinto%20-%20The%20web%20application%20hacker's%20handbook_%20finding%20and%20exploiting%20security%20flaws-Wiley%20(2011).pdf
- El Bakkouri, Bouchra & Raki, Samira & Belgnaoui, Touhfa. (2022). The Role of Chatbots in Enhancing Customer Experience: Literature Review. Procedia Computer Science. 203. 432-437. 10.1016/j.procs.2022.07.057. Available at: https://www.researchgate.net/publication/362662541_The_Role_of_Chatbots_in_Enhancing_Customer_Experience_Literature_Review