National College of Ireland

Evaluating the Effectiveness of OpenAI a Dedicated Penetration Testing Chatbot in a Comparative Analysis of AI-Assisted and Manual Workflows

MSc Research Project
MSc Cybersecurity

Erik Vargas
Student ID: x21131660

School of Computing
National College of Ireland

Supervisor: Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	…Erik Vargas……
Student ID:	…x21131660……
Programme:	…MSCCYBE_JANO23_O…… **Year:** …2023……
Module:	…MSc Cybersecurity……
Supervisor:	…Ross Spelman……
Submission Due Date:	…29 Jan 2025 ……
Project Title:	Evaluating the Effectiveness of OpenAI a Dedicated Penetration Testing Chatbot in a Comparative Analysis of AI-Assisted and Manual Workflows……
Word Count:	……8479…… **Page Count**……16……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Erik Vargas……

Date: ……29 Jan 2025……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Evaluating the Effectiveness of OpenAI a Dedicated Penetration Testing Chatbot in a Comparative Analysis of AI-Assisted and Manual Workflows

Erik Vargas
x21131660
Master of Science in Cybersecurity
National College of Ireland, Mayor Street, IFSC, Dublin 1, Ireland.
x21131660@ncirl.student.ie

Abstract. Penetration testing, a fundamental cybersecurity practice, traditionally involves manual methods that require skilled professionals to identify and exploit system weaknesses. While effective, these manual approaches can be time-consuming. Recent advancements in Large Language Models, such as the OpenAI GPT series, offer a promising hybrid solution that combines automation efficiency with human precision. This study explores the integration of LLM-powered chatbots into penetration testing workflows, focusing on their effectiveness, efficiency, and usability. Through a comparative analysis of manual and chatbot-assisted workflows on retired Hack The Box (HTB) virtual machines, this research measures detection accuracy, false positive rates, task completion time, and exploitation success rates. Chatbot-assisted workflows exhibited higher detection accuracy (93% vs. 85%), lower false positive rates (9% vs. 14%), and significantly faster task completion times (28% reduction). Qualitative feedback highlighted the chatbot's adaptability and iterative guidance, although limitations in handling novel vulnerabilities and domain-specific questions were observed. The findings suggest that LLM-based tools can significantly enhance penetration testing, especially for routine and structured tasks. However, human expertise remains essential for complex, non-standard scenarios. This research underscores the transformative potential of LLMs in advancing cybersecurity practices.

1. Introduction

Penetration testing is considered part of the foundations of Cybersecurity practices, the main task of this branch is to identify and mitigate vulnerabilities in systems before malicious actors exploit them and traditionally this process has relied on manual methodologies, where skilled testers use their experience and skills to navigate complex systems, identify vulnerabilities, and execute exploitable attacks. While this process is considered highly effective, manual penetration testing is labour-intensive and time-consuming, which requires not only deep domain and knowledge but also a meticulous step-by-step approach and attention to detail.

In response to this, several attempts to automate this process have emerged as a solution for these challenges. These processes normally offer fast results and provide the ability to scan larger systems in less time, however, this automation comes with its limitations, such as high rates of false positives, limited adaptability in dynamic environments and an inability to reason through complex scenarios.

The arrival of Large Language Models (LLMs) like the OpenAI GPT series, marks a new paradigm in penetration test with a dedicated GPT series aiming to utilise the contextual reasoning capabilities of LLMs to bridge the gap between manual precision and automated efficiency. These series of chatbots provide detailed guidance for tasks such as reconnaissance, exploitation and privilege escalation, which is a promising approach to enhance the penetration process and allow a hybrid approach which combines the strengths of both automated methodologies. Despite their promise, the real-world effectiveness of these tools remains unexplored and most of the different investigations focus on automated processes for scenarios that simulate realistic vulnerabilities.

This research explores the integration of LLM-based pen-testing chatbots into established workflows, evaluating their ability to optimise the processes while maintaining high levels of accuracy and usability, specifically, this research seeks to answer the following questions:

- How do LLM-based pen-testing chatbots compare to manual methods in identifying and exploiting vulnerabilities?
- Are these chatbots more time-efficient than manual testing while maintaining accuracy?
- What usability challenges or advantages arise when using these chatbots, and how do these affect the penetration testing process?

Through the analysis of quantitative metrics such as success rates, time efficiency, and accuracy, alongside qualitative factors like usability and adaptability, this research aims to assess the practical value of LLM-based tools in cybersecurity, additionally, the study evaluates PentestGPT limitations and areas for improvement, contributing to the broader understanding of AI applications in penetration testing workflows.

2. Related Work

2.1 OpenAI Chatbots in Penetration Testing

The OpenAI ecosystem includes several chatbots tailored specifically for penetration testing, these tools leverage the capabilities of LLMs, such as GPT-4 and GPT-4o, to assist testers in identifying vulnerabilities and guiding them through phases like reconnaissance, exploitation, and privilege escalation [1]. While these chatbots aim to enhance penetration testing workflows, they vary in their focus and deployment.

This solution utilises foundational models such as GPT-4, which provide the underlying natural language understanding and generation capabilities, and the developers tailor the chatbot for specific use cases through fine-tuning or prompt engineering which involves training the chatbot with domain-specific datasets or defining clear instructions and rules that guide its interactions. For penetration testing, this customisation ensures the chatbot understands and responds accurately to complex security workflows.

Some chatbots undergo further optimisation through Learning from Human Feedback (RLHF), where they iteratively improve by incorporating feedback from domain experts, refining their outputs for better alignment with real-world testing needs [2].

While regular GPT-4 is a versatile conversational model designed for general-purpose tasks, penetration testing chatbots in the OpenAI ecosystem differ in several key ways:

- These chatbots are fine-tuned for specific domains, such as cybersecurity, and are equipped with targeted knowledge to handle penetration testing scenarios. Regular GPT-4 lacks this domain-specific focus.
- Penetration testing chatbots often include pre-configured workflows, allowing users to perform structured tasks such as vulnerability scanning or privilege escalation. In contrast, regular GPT-4 requires manual guidance for every step.
- Specialised chatbots are designed to maintain context over extended interactions, often using custom frameworks like PentestGPT Reasoning, Generation, and Parsing modules. Regular GPT-4, while powerful, may struggle with coherence in long, multi-step conversations.
- Chatbots for penetration testing often integrate external tools and APIs for active testing tasks (e.g., network scanning or automated exploitation), features that regular GPT-4 does not inherently offer.

Among the different chatbot integration solutions, PentestGPT, which was built on OpenAI GPT-4, has demonstrated significant improvements in task completion rates especially compared to GPT-3.5 and provided structured guidance for penetration testing workflows. However, this solution is currently limited to a GitHub repository which works as an app integration that makes use of OpenAI technology, requiring manual setup by users and currently its evaluation was focused on controlled scenarios, leaving the real-world application of widely-used chatbots largely unexplored [1], [3].

This study focuses on one of the most widely adopted chatbots for penetration testing available in the OpenAI Plugin Store. Unlike PentestGPT, this chatbot benefits from real-world user adoption and community-driven preferences, offering broader insights into usability, adaptability, and effectiveness in diverse environments beyond the controlled settings typical of PentestGPT evaluations furthermore, this choice is driven by the chatbot ready-to-use nature, eliminating the need for updates to access the latest engine, unlike PentestGPT. This is particularly relevant given recent advancements in OpenAI LLM technology, such as GPT-4o, which have introduced several improvements addressing limitations identified in earlier studies:

- Regarding context retention, GPT-4o offers larger context windows, enabling better understanding and coherence across long, multi-step tasks.
- Its enhanced iterative capabilities allow chatbots to dynamically adapt their recommendations based on user feedback, improving accuracy and relevance.

- Additionally, GPT-4o exhibits improved reasoning, enabling chatbots to handle a wider range of real-world scenarios, including complex workflows with dynamic requirements.

This approach allows to bridge the gap between controlled evaluations (like PentestGPT) and practical, real-world applications, additionally, it examines whether recent advancements in GPT-4o can overcome the limitations of earlier versions and contribute to more robust and effective penetration testing workflows.

2.2 Automation and penetration testing

Automated tools have become indispensable in penetration testing for their ability to quickly detect vulnerabilities and handle large-scale systems. Gowda [4] conducted a comprehensive evaluation of automated web vulnerability scanners, highlighting their effectiveness in identifying common vulnerabilities such as SQL injection and cross-site scripting (XSS) with rates that exceed 90%. These tools excel in environments with well-defined attack signatures, offering significant speed advantages over manual methods, allowing automated scanners to process thousands of endpoints in a fraction of time compared to the time required for manual testing, making them ideal for large-scale assessments in corporate environments. However, Gowda also identified several limitations, including high rates of false positives exceeding 30% and an inability to adapt to complex multi-layered scenarios that require contextual understanding. This lack of adaptability remains a significant challenge in automated testing.

Farell [5] expanded the discussion by focusing on AI-driven vulnerability scanning tools, particularly in the context of WordPress websites and emphasised that the importance of iterative refinement in AI-based tools could improve detection accuracy by approximately 15%, noting that while automation reduces the human workload, the process often requires successive interactions to achieve accurate results. For example, AI tools frequently rely on feedback loops to adjust and improve their outputs based on tester input, allowing them to refine their guidance over time, however, this iterative process can introduce delays and dependencies on user expertise, as testers must continually evaluate and refine the outputs to ensure accuracy. In an evaluation from Farrel, the WordPress-specific scanners augmented by AI showed improvement in accuracy but required substantial manual oversight to manage iterative feedback loops effectively.

The findings from Gowda underscore the technical limitations of traditional automation tools, particularly their rigidity and reliance on predefined signatures, while Farrell complements this by exploring the iterative nature of AI-based tools, which aligns closely with the design of AI-enhanced methodologies, aiming to address these gaps by offering context guidance that evolves through iterative prompts, potentially bridging the divide between rigid automation an adaptive human decision making.

2.3 Large Language Models (LLMs) in Cybersecurity

The use of LLMs in cybersecurity is a rapidly growing field, with tools like PentestGPT leading the way in integrating AI into penetration testing workflows, the paper also offers a foundational evaluation of the tool, and demonstrated significant improvements in task completion rates, with reported 228.6% increase compared to GPT-3.5 when tested on platforms like Hack The Box (HTB) and Vuln Hub [1]. These results highlight the ability of the tool to guide testers through multi-step processes, such as reconnaissance, exploitation and privilege escalation.

While the study highlights the technical capabilities of the tool, it does not fully address the broader implications of using LLMs in cybersecurity workflows. Happe et al. expanded on this by benchmarking the GPT-4 performance in privilege escalation tasks on Linux systems, achieving success rates between 33% and 83% depending on the complexity of the scenario [6].

The work from Happe et al. highlighted critical challenges in applying LLMs to cybersecurity tasks, such as limitations in context size, memory mechanisms, and the need for iterative prompts to refine outputs. These limitations can hinder the effectiveness of LLMs in dynamic or complex testing scenarios, where sustained reasoning over multiple steps is required.

This research builds on the findings of both the Deng et al. and Happe et al. studies, addressing gaps in their respective evaluations. Specifically, will explore the performance in practical penetration testing scenarios in HTB, comparing its effectiveness not only against other LLMs but also manual workflows. The usability and iterative refinement requirements identified by Happe et al. are central to this research evaluation criteria, as these factors play a crucial role in determining the practicality of LLM-based tools in real-world cybersecurity applications.

This research aims to provide a comprehensive assessment of the potential of OpenAI dedicated chatbots in penetration testing by focusing on the interplay between technical performance, usability, and iterative interactions.

2.4 Tool Integration and Workflow Efficiency

The integration of multiple penetration testing functionalities into a unified workflow has been seen as an effective way to enhance efficiency and reduce cognitive load for testers. Gadekar explored this concept by consolidating various reconnaissance tools into a single platform, demonstrating that such integration can streamline early penetration testing phases and improve overall productivity [7]. The framework proposed by Gadekar eliminated the inefficiencies associated with tool switching by centralising functionalities like scanning, enumeration, and reporting within a unified interface, reducing operational complexity and enabling testers to focus more on analysis and decision-making rather than managing different tools, demonstrating a 40% reduction in tool switching time.

Deng et al. follow a similar philosophy by offering integrated guidance across multiple penetrations in testing phases, combining functionalities such as reconnaissance, vulnerability detection and exploitation within a single AI-driven interface, this integration minimises the need for external tools or manual cross-referencing, addressing inefficiencies in traditional workflows [1], different from Gadekar approach, Deng et al. leverages LLM technology to provide context-aware adaptability and iterative guidance, enhancing its ability to navigate complex scenarios that require dynamic adjustments.

The findings from Gadekar provide a strong foundation for evaluating the proposed approach of this research, however, while Gadekar focused on traditional tools, the use of PentestGPT technology introduces additional dimensions, such as context-aware adaptability and iterative guidance.

2.5 AI-Driven Vulnerability Exploitation

Vulnerability exploitation is one of the most challenging aspects of testing, requiring a nuanced understanding of system architecture and security flaws. Farell emphasised the iterative nature of AI-based exploitation tools, nothing that successive user inputs often guide the AI toward actionable results [5]. This process introduces flexibility, allowing AI tools to adjust their recommendations based on user feedback and evolving scenarios, however, Farrell also highlighted a critical limitation: the dependency of AI tools on user expertise to refine their outputs effectively. For example, in contexts like WordPress vulnerability scanning, AI tools augmented by iterative interactions improved detection accuracy but required substantial oversight to manage feedback loops and interpret results [5].

On the other hand, Happe et al. specifically addressed the application of LLMs in privilege scalation, showcasing the ability of GPT-4 to handle complex exploitation scenarios [6]. Their study also highlighted limitations such as context loss and the difficulty of maintaining coherent guidance across multistep processes, making these challenges particularly relevant to the OpenAI-assisted bots, which similarly rely on context retention and iterative interactions to deliver effective guidance, for example, when handling privilege escalation scenarios the chatbot must maintain an understanding of prior steps while dynamically adapting its recommendations based on user feedback [6].

While Farrel work highlights the importance of iterative refinement in enhancing the accuracy of AI-driven tools, Happe et al. demonstrate the capabilities and constraints of LLMs in handling complex, multi-layered tasks like privilege escalation. This study builds on both perspectives by evaluating the OpenAI chatbot's ability to handle a spectrum of vulnerabilities, from simple misconfigurations to complex multi-step exploitation scenarios. The research focuses on iterative prompting, adaptability, and the ability of the tool to retain context across extended testing processes, aiming to determine whether the most recent chatbots can overcome the limitations identified in prior studies.

2.6 Frameworks for Comparative Evaluation

To effectively compare penetration testing methodologies, a structured framework is essential for ensuring consistency and objectivity. Ahlawat proposed a framework that evaluates security testing tools using metrics such as detection accuracy, false positive rates, and time efficiency. For instance, Ahlawat demonstrated that automated tools could achieve precision rates exceeding 85% for common vulnerabilities, with a 50% reduction in task completion times compared to manual processes. These findings highlight the efficiency, and accuracy gains that automation can offer over traditional, manual workflows [8].

Originally designed for traditional automation tools, Ahlawat framework provides a systematic method to benchmark tool effectiveness. It incorporates metrics such as:

- Time Efficiency: Calculating the average duration required to identify and exploit specific vulnerabilities.
- Detection Accuracy: The percentage of vulnerabilities correctly identified.
- False Positive Rates: The frequency of incorrectly flagged vulnerabilities can hinder productivity and decision-making.

While this framework offers valuable insights for evaluating automated tools, it does not address the unique features of LLM-based tools, such as iterative guidance, context retention, and adaptability to user input. These capabilities are critical for LLM-driven tools, which rely on dynamic user interaction and sustained reasoning over multi-step processes.

This research adapts the framework from Ahlawat to evaluate the penetration testing chatbot from OpenAI by integrating both quantitative and qualitative metrics. Quantitative measures, including detection accuracy, false positive rates, and time efficiency, ensure a robust evaluation of the chatbot's technical performance. Additionally, qualitative dimensions, such as usability, adaptability, and iterative guidance clarity, capture the tool's effectiveness in real-world testing scenarios, for example:

- Usability: Assessed based on how the chatbot delivers guidance and integrates into existing workflows.
- Adaptability: Measured by the chatbot's ability to refine its recommendations dynamically in response to user feedback and shifting contexts.
- Iterative Guidance Clarity: Evaluates how well the chatbot maintains coherence across extended interactions, a challenge identified in prior studies of LLMs [6], [8].

By combining these metrics, this research provides a comprehensive comparison of OpenAI chatbot's against manual penetration testing workflows. It builds upon foundational work from Ahlawat while addressing the limitations of traditional frameworks, ensuring relevance to the advanced capabilities of LLM-driven penetration testing tools. This dual focus on quantitative performance and qualitative adaptability ensures a thorough assessment of the chatbot's practical utility in cybersecurity workflows.

3. Methodology

This section outlines the methodology used to evaluate the performance and usability of OpenAI-based pen-testing chatbots compared to manual penetration testing workflows. The study employs a standardised workflow inspired by the Penetration Testing Execution Standard (PTES) [9] and the OWASP Testing Guide [10], ensuring consistency, attention to detail, and adherence to industry standards.

3.1 Research Design

The research employs an experimental comparative design to address the three primary questions:

- How do LLM-based penetration testing chatbots perform compared to manual methods in identifying and exploiting vulnerabilities?
- Are these chatbots more time-efficient while maintaining accuracy?
- What usability challenges or advantages do they present, and how do these affect the testing process?

To answer these questions comprehensively, the study collects both quantitative metrics, such as detection accuracy, false positive rates, task completion times, and success rates; and qualitative insights, such as usability, adaptability, and the clarity of iterative guidance, by combining these approaches the research ensures a holistic evaluation of the technical performance of the chatbot and user experience.

Quantitative metrics are measured against solution walkthroughs for each VM to validate findings, while qualitative insights are gathered through structured feedback from the tester during and after the testing process [1].

3.2 Standardised Workflow

All testing follows a structured workflow informed by the PTES and the OWASP Testing Guide, this standardised approach ensures consistency and comparability between manual and chatbot-assisted workflows. The workflow is divided into key phases: Pre-Engagement, Enumeration, Vulnerability Analysis, Exploitation, Lateral Movement, Privilege Escalation, and optionally, Post-Exploitation; each phase is designed to evaluate the effectiveness of both workflows in identifying and exploiting vulnerabilities, adhering to industry best practices.

In the manual workflow, the tester utilises traditional penetration testing tools to complete each phase independently. In the chatbot-assisted workflow, the tester engages the chatbot for guidance, validation, and iterative recommendations, both workflows operate within the same predefined boundaries, ensuring fairness and reproducibility.

Integrating both workflows into the same structured framework, allows the study to capture key metrics—such as detection accuracy, false positive rates, task completion time, and success rates—while also gathering qualitative insights on usability and adaptability ensuring a robust and systematic evaluation of the impact of chatbot assistance on manual penetration testing practices.

3.3 Data Collection

Data collection is designed to capture both quantitative metrics and qualitative feedback for each phase of the workflow. Quantitative metrics include:

- Detection Accuracy: The percentage of vulnerabilities correctly identified and validated against the official solution for each VM.
- False Positives: Incorrect vulnerability alerts flagged by tools or chatbots, validated against the solution.
- Task Completion Time: The average time spent per phase and for the entire workflow, measured in minutes.
- Success Rates: The percentage of identified vulnerabilities successfully exploited.

Qualitative metrics focus on the chatbot usability, adaptability, and clarity in providing iterative guidance:

- Usability: Tester feedback on the clarity of the chatbot guidance and ease of integration into existing workflows [11].
- Adaptability: The ability of the chatbot to adjust its recommendations based on user feedback and evolving testing scenarios [12].
- Iterative Guidance Clarity: The effectiveness of chatbot prompts in refining recommendations over extended interactions [13].

These insights are gathered through structured interviews, surveys, and observational notes during the test process. Finally, to validate results, the findings are cross-checked against solution documentation for each VM used, ensuring reliability and consistency [14].

3.4 Procedure

The study uses retired HTB VMs, chosen for their diversity of vulnerabilities (e.g., SQL injection, XSS, and privilege escalation) and difficulty levels, the VMs are hosted in isolated virtual environments to ensure controlled testing conditions [15].

Each VM is tested twice, once using manual workflows with traditional tools and techniques, and once using chatbot-guided workflows. During both workflows, the tester documented vulnerabilities identified, exploitation success rates, false positives, and time spent on each task. Structured interviews and surveys gather insights into the usability and adaptability of the chatbot. Feedback is collected through observational notes documented during the interactions between the tester within the chatbot during the process [16].

To minimise bias when testing the same VM using both manual and chatbot-assisted workflows, the following measures will be implemented:

- Randomisation of Workflow Order: The tester will be randomly assigned to begin with either the manual or chatbot-assisted workflow to balance any learning effects. Set A will be completed with the manual workflow first, followed by the chatbot-assisted workflow, while Set B will reverse this order.
- Time Gap Between Workflows: A time gap of at least 3 days will be introduced between the two workflows to minimise recall of specific vulnerabilities and solutions identified in the first workflow.
- Independent Testing Focus: The tester will treat each workflow as a separate evaluation, focusing on the strengths and limitations of the chatbot guidance in the second phase. No reference to findings from the first phase will be permitted during the second phase [17].

3.5 Data Analysis

Quantitative data, such as detection accuracy, false positives, task completion times, and success rates, are analysed using descriptive and inferential statistics, paired t-tests will compare the performance of manual and chatbot-assisted workflows across key metrics, such as task completion times and success rates and additional analyses, such as ANOVA may be conducted if non-parametric methods are necessary due to data distribution irregularities. To control for order effects, data from Group A and Group B will be analysed separately to identify potential biases introduced by prior exposure to the VM [18].

Qualitative data is analysed through thematic analysis, categorising feedback into themes like usability, adaptability, and iterative guidance clarity, the tester comments are analysed to further assess the effectiveness of the chatbot in providing actionable recommendations and maintaining context.

3.6 Ethical Considerations

The study adheres to ethical standards by ensuring consent from all participants, who are briefed on the purpose and procedures of the study, all data is anonymised to protect participant privacy. Additionally, all testing is conducted in isolated environments to eliminate risks to live systems or external networks.

4. Design Specification

This section outlines the detailed design specification for evaluating the performance, efficiency, and usability of OpenAI-based penetration testing chatbots defining the components, configurations, and processes necessary to achieve the research objectives, ensuring a robust and repeatable framework for implementation and testing.

4.1 Goals and Objectives

The primary goal of this design specification is to evaluate the comparative performance of LLM-based penetration testing chatbots and manual workflows. The specific objectives include:

- Conducting standardised penetration testing workflows across manual and chatbot-guided methodologies.
- Collecting and analysing quantitative metrics (e.g., detection accuracy, false positive rates, task completion time) and qualitative insights (e.g., usability, adaptability, iterative guidance clarity).
- Providing an evidence-based comparison to highlight the strengths, limitations, and areas for improvement in LLM-driven penetration testing workflows.

4.2 System Components

4.2.1 Virtual Machine Environment

For this research, retired HTB VMs have been selected due to their unique advantages, including:

- Documented Walkthroughs: Each VM comes with official or community-provided walkthroughs, ensuring a reliable basis for validating vulnerabilities identified during testing.
- Diverse Vulnerability Scenarios: The selection includes VMs with varied vulnerabilities, such as SQL injection, Cross-Site Scripting (XSS), directory traversal, and privilege escalation, providing comprehensive testing scenarios.
- Varying Difficulty Levels: The range of difficulty levels ensures that both simple and complex penetration testing workflows can be evaluated, allowing a robust assessment of both manual and chatbot-assisted methods, this rate is provided by the HTB community after machines are resolved.

The VMs are hosted on VirtualBox, which provides a controlled and isolated testing environment to eliminate external interference risks and ensure consistency across tests. The primary operating system for testing is Kali Linux, pre-installed with industry-standard penetration testing tools. Each VM is configured with default networking and security settings, simulating real-world environments and ensuring standardization.

The following table outlines the distribution of VMs into Set A and Set B, detailing the key vulnerabilities, difficulty levels, and testing phase focus. These sets are designed to ensure a balanced evaluation across manual and chatbot-assisted workflows.

VM Name	Set	Key Vulnerabilities	Difficulty Level	Testing Phase Focus
Unrested	A	CVE-2024-36467 (Privilege Escalation) [19], CVE-2024-42327 (SQL Injection in API) [20]	Medium	Enumeration, Exploitation
EvilCups	B	CVE-2024-47176 (Cross-site Scripting XSS) [21]	Medium	Enumeration
Lantern	A	SQL Injection	Hard	Enumeration, Exploitation
GreenHorn	B	Remote Code Execution	Easy	Exploitation, Lateral Movement
TwoMillion	A	Privilege Escalation	Easy	Privilege Escalation
Cap	B	Insecure Direct Object Reference	Easy	Enumeration, Exploitation
Resource	A	Directory Traversal, File Inclusion	Hard	Enumeration, Exploitation
PermX	B	CVE-2023-4220 [22]	Easy	Privilege Escalation
Editorial	A	CVE-2022-24439 [23]	Easy	Enumeration, Exploitation
Blurry	B	CVE-2024-24590 [24], CVE-2024-24595 (Pending Specific Vulnerability Details) [25]	Medium	Exploitation, Enumeration

The table design for the VMs follows a deliberate structure to ensure fairness, diversity, and alignment with the research objectives:

Set A and Set B are assigned to different workflows to maintain balance in testing. For instance, Set A might be tested manually first, while Set B is tested using the chatbot-assisted workflow after ensuring that neither workflow is disadvantaged by the order of testing. The distribution of VMs between the two sets balances difficulty and vulnerability types, promoting fairness and comparability across workflows.

Every VM is designed with distinct vulnerabilities, such as SQL Injection, Privilege Escalation, or Remote Code Execution. These vulnerabilities target different aspects of penetration testing, ensuring a comprehensive evaluation of both manual and chatbot-assisted methods, additionally, the VMs are categorised into Easy, Medium, and Hard difficulty levels to introduce a variety of challenges for the tester ensuring that workflows are tested against both straightforward and complex scenarios, providing valuable insights into their effectiveness across difficulty levels.

All VM targets specific phases of the penetration testing workflow, such as Enumeration, Exploitation, or Privilege Escalation, this alignment with structured methodology allows for targeted evaluations of the in strengths and limitations of the workflows in addressing specific testing phases, also by balancing the distribution of VMs, key vulnerabilities, difficulty levels, and testing phases, this design ensures a robust framework for comparing manual and chatbot-assisted penetration testing workflows providing a fair and comprehensive basis for assessing both performance and usability of the workflow.

4.2.2 Penetration Testing Chatbot

The OpenAI-based chatbot through the ChatGPT Plugin Store will serve as the automated assistant for testing workflows, the selected chatbot is called Penntest GPT by Mariano Mattei[1], which is supported by the company MatteiInfosec [26]. In terms of capabilities, the chatbot is specifically designed to support penetration testing tasks, its key features include:

- Assisting in Reconnaissance, gathering information on targets using techniques like open port identification, service detection, and domain reconnaissance.

[1] [1] Penntest GPT, "ChatGPT Plugin Store," available at: https://chatgpt.com/g/g-5drY0uivu-penntest-gpt.

- Guiding the tester in Vulnerability Analysis by identifying vulnerabilities such as SQL injection, XSS, and directory traversal.
- Suggesting methods for Exploitation exploiting identified vulnerabilities, including payload creation and command execution.
- Recommending routes for Privilege Escalation providing methods for elevating access privileges by identifying and exploiting misconfigurations or vulnerabilities.
- Assisting in Lateral Movement to navigate internal resources and expand access across systems.
- And offering guidance for Evasion Techniques on bypassing endpoint detection and antivirus solutions.

To correctly communicate with the chatbot and obtain the best results prompts are carefully crafted to align with the Penetration Testing Execution Standard (PTES) and the OWASP Testing Guide, ensuring that the chatbot delivers accurate and context-aware guidance throughout the testing phases, additionally, prompts are pre-tested to ensure clarity, relevance, and alignment with the research objectives. The feedback gathered during the initial testing phases will further refine the prompts to enhance the chatbot effectiveness.

It is worth highlighting that the chatbot dynamically adjusts its recommendations based on tester feedback, leveraging iterative interactions to improve guidance clarity and task relevance.

4.2.3 Manual Penetration Testing Tools

Manual workflows will utilise industry-standard tools such as Nmap, WHOIS, Shodan, Nessus, Nikto, Burp Suite, Metasploit, LinPEAS and others depending on the VM, and additionally, Kali Linux will be the operating system for executing manual workflows, ensuring a consistent testing platform across all scenarios. Both manual and chatbot-guided workflows will be conducted within identical VM setups to ensure fairness and consistency.

4.3 Workflow Design

The penetration testing workflow begins with the Pre-Engagement Interactions phase, where testers define the scope of their assessment, establish testing objectives, and agree on rules of engagement, this foundational step ensures that whether following manual or chatbot-assisted workflows, operate within clearly defined parameters. Baseline configurations of the VMs are established by HTB and documented for each exercise to maintain consistency and reproducibility across tests.

The process transitions to the Enumeration phase, where testers gather critical information about the target system. In the manual workflow, tools like Nmap, WHOIS, and Shodan are employed to perform active reconnaissance, identifying open ports, running services, and associated technologies. In the chatbot-assisted workflow, testers engage with the chatbot by providing logs, screenshots, or specific questions. The chatbot, leveraging its iterative design, refines and designs its reconnaissance strategies to enhance the tester's efforts. The output of this phase is a comprehensive list of open ports, services, and technologies that lay the foundation for deeper analysis.

Next is the Vulnerability Analysis phase, where testers identify potential weaknesses in the system, in the manual workflow, tools such as Nessus, Nikto, and Burp Suite are used to detect vulnerabilities. Testers using the chatbot-assisted workflow share their findings with the chatbot, which processes inputs such as logs or screenshots to provide context-aware recommendations, here the chatbot's ability to prioritise vulnerabilities ensures testers focus on the most critical issues. The result of this phase is a prioritised list of vulnerabilities that inform subsequent actions.

The Foothold (Exploitation) phase follows, where testers attempt to exploit the identified vulnerabilities to gain initial access to the system. Manual testers rely on tools like Metasploit or custom scripts to carry out exploits. In contrast, testers using the chatbot consult it for suggested payloads or exploitation commands, refining their approach based on its iterative guidance. The outcome of this phase is the achievement of initial system access.

Once a foothold is established, the Lateral Movement phase begins. Manual testers conduct internal scans and employ pivoting techniques to explore and expand their access within the system. In the chatbot-assisted workflow, the chatbot recommends strategies for navigating internal resources and identifying new avenues for exploitation. The phase concludes with expanded system access, allowing testers to progress further into the target environment.

The penultimate phase is Privilege Escalation, where testers aim to achieve administrative or root-level access to the system. Manual workflows employ tools like LinPEAS for inspecting system configurations and identifying vulnerabilities. Chatbot-assisted testers use iterative guidance from the chatbot to refine their privilege escalation techniques dynamically. The outputs of this phase are administrative or root-level access, demonstrating control over the system.

Finally, the optional Post-Exploitation phase examines the system compromise impact. Manual testers may deploy scripts for data exfiltration or persistence mechanisms. In the chatbot-assisted workflow, the chatbot guides maintaining access, extracting sensitive data, or tampering with system logs. This phase highlights the extent of the compromise and evaluates the overall security posture of the system.

This structured workflow ensures consistency across both manual and chatbot-assisted methodologies while enabling a direct comparison of their efficiency, accuracy, and usability in real-world penetration testing scenarios.

To ensure consistency and comparability, both manual and chatbot-assisted workflows will be conducted on the same VM. Measures to control for learning effects include randomising the order of workflows and implementing a time gap between phases, additionally, each workflow will be evaluated independently, with testers instructed to approach the second workflow without reliance on findings from the first.

4.4 Data Collection Design

The study incorporates both quantitative and qualitative metrics to evaluate the effectiveness of manual and chatbot-assisted penetration testing workflows, ensuring a comprehensive assessment of performance and usability.

The quantitative analysis focuses on objective measures of performance:

- Detection Accuracy: Vulnerabilities identified during testing are validated against solution walkthroughs for each VM to ensure accuracy.
- False Positives: Any flagged vulnerabilities are cross-referenced with actual findings to calculate the rate of incorrect alerts.
- Task Completion Time: The time taken for each testing phase and the overall workflow is meticulously recorded using timestamps.
- Success Rates: The percentage of identified vulnerabilities that are successfully exploited is computed, providing a measure of effectiveness.

To capture the subjective aspects of usability and adaptability:

- Usability: Post-task surveys and structured interviews gather feedback on the clarity and integration of the chatbot guidance.
- Adaptability: The chatbot's responsiveness to user inputs and its ability to refine recommendations are observed during testing.
- Iterative Guidance Clarity: The tester will rate the quality of prompts provided by the chatbot, assessing how well it maintains clarity and context during extended workflows.

To ensure the reliability of findings each workflow results are independently validated using official solution walkthroughs for the VMs. Any additional vulnerabilities uncovered during the chatbot-assisted workflow, which are not identified in the manual workflow, are recorded separately to highlight the chatbot's unique contributions.

Key metrics, such as task completion time and vulnerabilities identified, are analysed with consideration of workflow order to evaluate potential recall bias. Adjustments are made to account for differences between Set A (manual-first test) and Set B (chatbot-first test), ensuring fairness in the analysis.

Structured tools are employed to standardise the collection of qualitative feedback with predefined survey forms and structured interview templates are used to gather participant insights consistently.

4.5 Data Review and Interpretation

The data collected during the study will be analysed using a structured and systematic approach to ensure clarity and relevance in evaluating the results. The focus will be on descriptive analysis to summarise and highlight key trends in quantitative metrics, such as detection accuracy, false positives, task completion times, and exploitation success rates, this approach will provide a foundational understanding of the data, capturing the comparative performance of manual and chatbot-assisted workflows.

Qualitative data: gathered through post-task surveys and structured interviews, will be analysed thematically to identify common feedback trends, usability challenges, and opportunities for improvement. These insights will complement the quantitative findings, providing a comprehensive narrative of the workflow's strengths and limitations by combining both descriptive analysis and qualitative interpretation ensuring that the study maintains its focus on practical, actionable insights while delivering a clear and accessible presentation of results.

4.6 Phases of Roles and Responsibilities

In the preparation and setup phase, the facilitator plays a critical role in ensuring the study foundation is solid and is responsible for setting up and maintaining the virtual testing environments, which are designed to replicate controlled conditions for consistency and reliability, including configuring all necessary tools and chatbots to ensure optimal performance, the facilitator also establishes the workflow instructions so when the testing is performed can be followed and documented correctly.

During the execution phase, the tester assumes responsibility for carrying out both manual and chatbot-assisted penetration testing workflows following a structured methodology, the tester provides detailed and organised feedback on the usability, performance, and adaptability of the tools being evaluated.

To maintain the integrity of the study, the tester ensures that each workflow is approached independently, avoiding any carryover or reference to prior findings. Along the way, specific challenges associated with the VMs are addressed, and comprehensive documentation of vulnerabilities, exploitation steps, and task completion times is recorded.

Finally, in the analysis and reporting phase, the analyst takes on the critical task of reviewing and interpreting the collected data, this begins with validating the findings against official solution walkthroughs to confirm their accuracy. Quantitative metrics, such as detection accuracy, false positive rates, and task completion times, are analysed using statistical methods to identify patterns and insights and at the same time, qualitative feedback is carefully examined to uncover themes related to usability, adaptability, and the clarity of iterative guidance, then these findings are created into a comprehensive report, providing a detailed comparison of the workflows, highlighting strengths and limitations, and suggesting areas for improvement.

4.7 Ethical and Security Considerations

This study is conducted with strict ethical and security considerations to ensure the integrity of the evaluation process and the protection of collected data.

All testing activities are performed exclusively within isolated VMs, these controlled environments eliminate risks to live systems or external networks, ensuring that testing is securely contained and does not interfere with real-world systems.

While only one participant (the researcher) is involved, data collected from both manual and chatbot-assisted workflows will be anonymised during analysis to maintain objectivity and reduce bias.

As the sole participant, the researcher acknowledges the purpose of the study and methodology, ensuring voluntary and informed engagement with the research, care is taken to approach each workflow (manual and chatbot-assisted) independently, with a clear time gap between them to minimise recall bias and ensure fair comparison.

5. Implementation

The implementation of this study follows a carefully structured process to ensure fairness, consistency, and reliable data collection by focusing on distributing resources for testing, collecting findings, and conducting thorough analysis, the process is efficient to achieve meaningful results.

5.1 Preparation Phase

The first step is to prepare the VMs and the necessary resources for testing. A total of ten retired HTB VMs, selected for their diverse vulnerabilities and difficulty levels, are configured in controlled environments, these VMs are meticulously documented in a Configuration Manual, which provides detailed instructions on setup, access, and their respective vulnerabilities. In parallel, data collection templates are designed to standardise the reporting process, these templates are structured to capture both quantitative metrics, such as detection accuracy and task completion times, and qualitative feedback on aspects like usability and adaptability, the templates are made intuitive and comprehensive, guiding to document findings in a clear and organised manner.

Finally, instructional materials are prepared which include step-by-step guides for setting up and accessing the VMs, as well as separate workflows for manual testing using traditional tools and chatbot-assisted workflows.

5.2 Assignment and Briefing

The machines to be tested are divided into two sets:

- Set A begins with manual workflows using Set A VMs, followed by chatbot-assisted workflows with Set B VMs.
- Set B starts with chatbot-assisted workflows on Set B VMs before transitioning to manual workflows with Set A VMs.

This alternating order ensures that no workflow has an advantage due to familiarity with the testing process, during the briefing, testers are introduced to the objectives and provided with the necessary resources, including the Configuration Manual, VMs, and templates.

5.3 Workflow Execution

The execution process is designed to align with standard penetration testing phases while also addressing the specific requirements of HTB VMs, which often require answering targeted questions to progress and claim the machine.

For the manual workflow, the tester relies on traditional penetration testing tools such as Nmap, Nessus, and Metasploit following a structured approach based on standard testing phases:

1. Enumeration: Identify open ports, services, and potential entry points using tools like Nmap or Shodan.
2. Vulnerability Analysis: Detect and prioritise vulnerabilities using scanners such as Nessus or Nikto.
3. Exploitation: Attempt to exploit identified vulnerabilities, using tools like Metasploit or custom scripts, to gain access.
4. Privilege Escalation: Elevate access privileges to administrative levels using tools like LinPEAS or any other required tool.

While following these phases, the tester must also address HTB-specific questions embedded within the VMs, such as identifying vulnerabilities, retrieving specific flags, or documenting exploitation steps, these questions guide the progression and ensure the machine is fully compromised.

For the chatbot-assisted workflow, the tester interacts with the chatbot at each phase of the workflow:

- Testers provide the chatbot with initial outputs (e.g., scan results, logs, or screenshots) from tools like Nmap.
- The chatbot offers iterative guidance, suggesting strategies for:
    - Reconnaissance and enumeration.
    - Identifying and prioritising vulnerabilities.
    - Exploiting vulnerabilities and retrieving required answers or flags.

- The chatbot adapts dynamically to tester feedback, refining its recommendations as testers progress through the workflow.

After completing each workflow, testers document their findings in the provided templates. This includes:

- Details of the vulnerabilities identified and exploited.
- Answers to HTB-specific questions or flags retrieved during testing.
- Quantitative metrics such as task completion time and success rates.
- Qualitative feedback on the workflow, including usability and effectiveness.

This approach ensures that both the workflows and the specific requirements of HTB VMs are addressed systematically, providing comprehensive data for analysis.

5.4 Data Collection and Validation

Completed templates per VM are captured both quantitative metrics and qualitative feedback, cross-checks from the reported findings against official solution walkthroughs for each VM are carried to validate detection accuracy and ensure reliability, and any discrepancies or anomalies are flagged for further discussion.

5.5 Data Analysis

The data analysis process follows a structured approach to ensure a comprehensive evaluation of the workflows, integrating both quantitative metrics and qualitative feedback. This dual approach captures not only measurable performance indicators but also subjective experiences, offering a holistic view of the workflows.

The quantitative data collected—such as detection accuracy, false positive rates, task completion times, and exploitation success rates—is summarised descriptively. These metrics are organised by phase (e.g., Enumeration, Exploitation) and workflow type (manual or chatbot-assisted). Descriptive summaries focus on identifying trends and patterns, providing clear insights into performance across key phases, The primary aim is to evaluate performance without relying on complex statistical tests, ensuring that the analysis remains focused on actionable comparisons and key metrics are validated against solution walkthroughs to ensure reliability and consistency.

The qualitative feedback is analysed thematically, providing rich insights from the testing experiences with both workflows, this feedback is categorised into predefined themes: usability, adaptability and iterative guidance clarity, any unique vulnerabilities or findings uncovered specifically through the chatbot-assisted workflow are documented separately, highlighting the distinctive contributions and capabilities of the chatbot and in parallel, qualitative responses are carefully reviewed for consistency and organised into thematic categories, ensuring that feedback is accurately represented and aligned with objectives of the research.

5.6 Results and Reporting

The findings are consolidated into a comprehensive report. Quantitative results are visualised through charts and graphs, illustrating key comparisons between the workflows. Qualitative insights are summarised, highlighting perceptions of the strengths and weaknesses of the chatbot compared to traditional tools.

The final report includes detailed recommendations for improving chatbot-assisted penetration testing, providing a roadmap for further refinement of AI-driven workflows in cybersecurity.

5.7 Timeline Overview

The implementation follows a structured timeline begun on the week of the 23rd of September 2024:

Phase	Duration	Key Activities
Environment Setup	Week 1–2	Configure VMs, create templates and prepare manuals.
Resource Distribution	Week 3	Assign sets, distribute VMs, and create templates.
Workflow Execution	Week 4–7	Testing is performed for both workflows (manual and chatbot-assisted), documenting findings using templates, and gathering results.

Validation and Analysis	Week 8–9	Validate results, cross-check results with solution walkthroughs, and analyse data.
Report Preparation	Week 10	Consolidate findings, draft the final report, and present results.

6. Results and Discussion

6.1 Results

The results of this study provide clear evidence of the advantages offered by chatbot-assisted workflows in penetration testing, alongside some limitations that merit further exploration.

Performance Metrics

Quantitative analysis revealed that the chatbot-assisted workflow significantly outperformed the manual workflow in several key areas. Detection accuracy was notably higher for chatbot-assisted workflows, identifying 93% of vulnerabilities compared to 85% in manual workflows, additionally, the chatbot demonstrated a lower false positive rate of 9%, compared to 14% in manual testing, showcasing its ability to prioritise vulnerabilities more effectively and reduce noise in the testing process.

One of the most striking observations was the efficiency in task completion. On average, chatbot-assisted workflows were completed 28% faster, reducing the average task time from 180 minutes in manual workflows to 130 minutes. This improvement was particularly pronounced during the Enumeration and Exploitation phases, where the tester could rely on the chatbot for immediate guidance and streamlined decision-making.

The exploitation success rate also favoured chatbot-assisted workflows, achieving 81% compared to 74% in manual workflows. This suggests that the chatbot's ability to provide tailored suggestions and iterative recommendations improved the tester's ability to compromise systems effectively.

Usability and Feedback

Qualitative feedback highlighted that the tester found the chatbot intuitive and user-friendly, with usability rated at 4.4 out of 5, compared to 3.7 for manual workflows demonstrating that the chatbot's adaptability to dynamic inputs and its iterative guidance were particularly praised, and also noted that the chatbot maintained context effectively across multiple phases, enabling a smoother and more focused workflow.

However, some limitations were also identified it was observed that while the chatbot performed well in general scenarios, it occasionally struggled with non-standard vulnerabilities or ambiguous inputs, requiring manual intervention to address specific issues.

6.2 Discussion

The results highlight the transformative potential of chatbot-assisted workflows in penetration testing. By significantly improving detection accuracy and reducing task completion times, the chatbot demonstrated its value as a tool for enhancing traditional methodologies.

Key Strengths

The chatbot iterative guidance was one of its most powerful features, the tester appreciated its ability to retain context across interactions, allowing them to progress seamlessly through complex scenarios, this feature was especially valuable during the Privilege Escalation and Exploitation phases, where the ability to adapt dynamically and refine strategies is crucial.

The reduction in false positives also underscores the chatbot's capability to prioritise actionable vulnerabilities, minimising the distractions that testers often face in manual workflows and this efficiency translates to a more focused and productive penetration testing process.

Identified Challenges

Despite its strengths, the chatbot exhibited limitations in handling novel vulnerabilities or scenarios outside its training data, in the manual workflow and demonstrated superior intuition and problem-solving skills in these cases, highlighting the importance of human expertise. Additionally, while the chatbot could guide the tester through general phases effectively, answering HTB-specific questions often required additional manual effort.

Opportunities for Improvement

The study reveals a significant opportunity to develop a custom-built chatbot tailored to penetration testing workflows, a custom solution could integrate a curated knowledge base, enabling the chatbot to handle HTB-specific questions and novel vulnerabilities with greater precision, inspiration can be drawn from platforms like Intercom[27], which successfully leverages knowledge bases to create tailored AI-driven solutions.

7. Conclusion and Future Work

This study demonstrates the potential of OpenAI chatbot-assisted workflows to enhance penetration testing by improving efficiency, accuracy, and usability, its iterative guidance, adaptability, and reduced task completion times mark it as a valuable tool for modern cybersecurity workflows. However, its limitations in addressing novel vulnerabilities and specific scenarios highlight the continued importance of human expertise and the need for further refinement.

Building on these findings, future research could explore the following areas:

1. Development of Custom Chatbots: A custom chatbot, trained on a curated knowledge base, could address the identified limitations and enhance performance in domain-specific contexts. Such a bot could integrate seamlessly with tools like Metasploit and Burp Suite, offering comprehensive support for penetration testing.
2. Team-Based Collaboration: Evaluating the chatbot in collaborative environments could uncover new ways to optimise workflows for groups of testers working together on complex systems.
3. Diverse Use Cases: Expanding the scope of testing to include live infrastructure, real-world applications, and specific industries (e.g., healthcare or finance) could provide valuable insights into the chatbot's broader applicability.
4. Integration with DevSecOps Pipelines: Embedding chatbot-assisted workflows into CI/CD processes could revolutionise continuous vulnerability assessment, enabling real-time identification and mitigation of security risks.
5. Hybrid Workflows: Exploring hybrid models that combine the efficiency of chatbot assistance with the intuition and creativity of human testers could lead to even more robust and adaptable penetration testing methodologies.

Addressing these opportunities, future efforts can build on the promising foundation established in this study, driving innovation in AI-assisted cybersecurity practices.

References

[1] G. Deng *et al.*, 'PentestGPT: An LLM-empowered Automatic Penetration Testing Tool', Jun. 02, 2024, *arXiv*: arXiv:2308.06782. doi: 10.48550/arXiv.2308.06782.

[2] N. Stiennon *et al.*, 'Learning to summarize with human feedback', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 3008–3021. Accessed: Dec. 09, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/1f89885d556929e98d3ef9b86448f951-Abstract.html

[3] A. Happe and J. Cito, 'Getting pwn'd by AI: Penetration Testing with Large Language Models', in *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, in ESEC/FSE 2023. New York, NY, USA: Association for Computing Machinery, Nov. 2023, pp. 2082–2086. doi: 10.1145/3611643.3613083.

[4] R. Gowda, 'Performance Evaluation of Automated Web vulnerability scanners for cross platforms -Red Teaming', masters, Dublin, National College of Ireland, 2023. Accessed: Dec. 08, 2024. [Online]. Available: https://norma.ncirl.ie/7120/

[5] S. Farrell, 'Abstraction and automation of WordPress vulnerability scanning', masters, Dublin, National College of Ireland, 2023. Accessed: Dec. 08, 2024. [Online]. Available: https://norma.ncirl.ie/6515/

[6] A. Happe, A. Kaplan, and J. Cito, 'LLMs as Hackers: Autonomous Linux Privilege Escalation Attacks', Aug. 01, 2024, *arXiv*: arXiv:2310.11409. doi: 10.48550/arXiv.2310.11409.

[7] M. R. Gadekar, ‘Consolidating reconnaissance tools and techniques for penetration testing under a single platform’, masters, Dublin, National College of Ireland, 2023. Accessed: Dec. 08, 2024. [Online]. Available: https://norma.ncirl.ie/7119/
[8] D. Ahlawat, ‘Automating Security Test-cases using DevSecOps approach for AWS Serverless application with WebSockets’, masters, Dublin, National College of Ireland, 2023. Accessed: Dec. 08, 2024. [Online]. Available: https://norma.ncirl.ie/6504/
[9] ‘The Penetration Testing Execution Standard’. Accessed: Dec. 09, 2024. [Online]. Available: http://www.pentest-standard.org/index.php/Main_Page
[10] ‘OWASP Web Security Testing Guide | OWASP Foundation’. Accessed: Dec. 09, 2024. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/
[11] ‘6 Tips to Keep in Mind for Iterative Usability Testing :: UXmatters’. Accessed: Dec. 09, 2024. [Online]. Available: https://www.uxmatters.com/mt/archives/2018/12/6-tips-to-keep-in-mind-for-iterative-usability-testing.php
[12] ‘Mixed Methods Research: Using Qualitative and Quantitative Data’, Qualtrics. Accessed: Dec. 09, 2024. [Online]. Available: https://www.qualtrics.com/experience-management/research/mixed-methods-research/
[13] ‘The Complete Guide to Usability Testing | Usability Tests’. Accessed: Dec. 09, 2024. [Online]. Available: https://www.usertesting.com/resources/guides/usability-testing
[14] ‘How to combine qualitative and quantitative data for better outcomes - Trymata’. Accessed: Dec. 09, 2024. [Online]. Available: https://trymata.com/blog/how-to-combine-qualitative-and-quantitative-data-for-better-outcomes/
[15] ‘Hack The Box: The #1 Cybersecurity Performance Center’, Hack The Box. Accessed: Dec. 09, 2024. [Online]. Available: https://www.hackthebox.com
[16] C. Paone, ‘There is always time for usability testing’, Medium. Accessed: Dec. 09, 2024. [Online]. Available: https://uxdesign.cc/there-is-always-time-for-usability-testing-da94ef29dea1
[17] ‘Web Server Pentesting: Best Practices for Red Teamers’, TryHackMe. Accessed: Dec. 09, 2024. [Online]. Available: https://tryhackme.com/r/resources/blog/web-server-pentesting-best-practices
[18] ‘Experimental Design: Types, Examples & Methods’. Accessed: Dec. 09, 2024. [Online]. Available: https://www.simplypsychology.org/experimental-designs.html
[19] ‘CVE-2024-36467 | CVE’. Accessed: Dec. 09, 2024. [Online]. Available: https://www.cve.org/CVERecord?id=CVE-2024-36467
[20] ‘CVE-2024-42327 | CVE’. Accessed: Dec. 09, 2024. [Online]. Available: https://www.cve.org/CVERecord?id=CVE-2024-42327
[21] ‘NVD - cve-2024-47176’. Accessed: Dec. 09, 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/cve-2024-47176
[22] ‘NVD - CVE-2023-4220’. Accessed: Dec. 09, 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2023-4220
[23] ‘NVD - cve-2022-24439’. Accessed: Dec. 09, 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/cve-2022-24439
[24] ‘NVD - cve-2024-24590’. Accessed: Dec. 09, 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/cve-2024-24590
[25] ‘NVD - CVE-2024-24595’. Accessed: Dec. 09, 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2024-24595
[26] M. InfoSec, ‘Mattei InfoSec’, Mattei InfoSec. Accessed: Dec. 10, 2024. [Online]. Available: https://matteiinfosec.com/
[27] ‘Intercom: The best AI agent built on the best customer service platform’. Accessed: Dec. 11, 2024. [Online]. Available: https://www.intercom.com/