Ethical Hacking System - Web and Linux Toolkits

B. L. Naveen Mayantha Wijesinghe

IT2020077

Faculty of Information Technology

Horizon Campus

December - 2024

This dissertation is submitted in partial fulfillment of the requirement of the Degree of BIT (Hons) in Networking and Mobile Computing of the Horizon Campus

DECLARATION

I certify that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a degree or diploma in any university, and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Student: . . . . . .

Date: / ….. / …...

Name of Student: B.L. Naveen Mayantha Wijesinghe

Countersigned by: Signature of Supervisor(s): . . . . . .

Date: /…... / …...

Name(s) of Supervisor(s): Mr. Thilina Samarasinghe

Abstract

The Integrated Ethical Hacking System (IEHS) addresses critical gaps in current ethical hacking tools and educational resources by providing a unified platform aimed at both practical utility and educational enhancement. Ethical hacking plays a crucial role in today's cybersecurity landscape, yet the fragmentation and complexity of existing resources create inefficiencies that hinder the effectiveness of professionals and educators alike. The IEHS seeks to overcome these challenges by integrating a comprehensive suite of ethical hacking tools with a robust educational component within a single, user-friendly interface.

This venture follows a structured development approach, beginning with an extensive literature review to identify key limitations in existing ethical hacking systems. The review highlights the prevalence of fragmented, difficult-to-navigate tools that complicate workflows and hinder newcomers' ability to learn efficiently. Following this, a detailed requirement analysis was conducted, gathering input from a broad spectrum of stakeholders, including ethical hackers, educators, and students. By leveraging surveys, interviews, and other fact-finding methods, the analysis identified essential platform features, such as a diverse range of integrated tools, intuitive user interfaces, advanced reporting capabilities, and a comprehensive educational section. This approach ensures that the IEHS meets both professional demands and educational objectives.

The system design phase of the IEHS was meticulously planned to optimize usability, consistency, and accessibility. Use case diagrams, class diagrams, sequence diagrams, activity diagrams, and interface mockups were developed to map out and visualize the platform's functionality. The database design was also carefully structured to support key features, ensuring efficient data management across various system components. A feedback loop involving user surveys and usability testing provided valuable insights, allowing for iterative improvements that align the design with user needs.

Once implemented, the IEHS will undergo rigorous testing to validate its functionality and effectiveness, with a strong emphasis on user experience. By integrating tools and educational resources into a single platform, the IEHS enhances the efficiency and effectiveness of ethical hacking practices, offering users a streamlined, cohesive, and engaging experience. Continuous updates will be made to align with advancements in cybersecurity, ensuring the platform remains relevant and capable of supporting users' long-term skill development.

In conclusion, the Integrated Ethical Hacking System represents a significant step forward in the field by providing a unified, user-centric platform that facilitates both professional and academic pursuits in ethical hacking. This project not only improves the efficiency of ethical hacking processes but also fosters continuous learning and skill development, laying a strong foundation for future enhancements in cybersecurity education and practice.

Acknowledgment

I would like to express my heartfelt gratitude to everyone who supported and guided me throughout the development of this project, The Integrated Ethical Hacking System.

First, I extend my deepest thanks to my advisor, Mr. Thilina Samarasinghe, for his invaluable guidance, expertise, and constant encouragement. His insightful feedback and suggestions were crucial to shaping this project and refining it at each stage.

I am also incredibly grateful to the professors and faculty members of Horizon Campus, whose support, resources, and mentorship provided me with a strong foundation in ethical hacking and cybersecurity.

Special thanks go to my brother, Ravindu, and his professor at Northumbria University, Professor Ryan, for their advice and encouragement, as well as to Shankar, a cybersecurity expert in India, and Daniel Lowrie, a lecturer at ITProTV in USA, for their expertise and guidance. I am equally appreciative of the Reddit and GitHub cybersecurity communities, whose shared insights and experiences during the requirement analysis phase were instrumental in identifying the core needs and challenges that shaped the development of this project.

I am deeply thankful to the participants who engaged with the surveys and usability tests, providing valuable feedback that significantly enhanced the final outcome. Without their willingness to share their perspectives, this project would not have been as well-rounded or effective.

Lastly, I would like to thank my family and friends for their unwavering support, understanding, and encouragement throughout this journey. Their belief in my abilities gave me the motivation to persevere and bring this project to completion.

Thank you to everyone who played a part in this project, directly or indirectly. Your support has been invaluable, and this achievement would not have been possible without you.

Table of Contents

CHAPTER 1 – INTRODUCTION

As cyber threats continue to rise in frequency and sophistication, the need for effective and well-rounded tools for ethical hacking and penetration testing has never been greater. Cybersecurity is now a cornerstone of digital infrastructure protection, with ethical hacking, or penetration testing, at the forefront. Ethical hacking involves simulating cyber-attacks on systems, networks, and applications to identify vulnerabilities and strengthen defenses against potential attacks. By using tactics and techniques similar to those of malicious hackers, ethical hackers—professionals skilled in identifying and neutralizing threats—are essential in preventing data breaches and ensuring system security.

Despite the growing demand for cybersecurity defenses, the current suite of tools for ethical hacking remains fragmented. This fragmentation forces cybersecurity professionals to rely on various disjointed tools, each performing a single task or addressing a specific area of vulnerability assessment. The resulting lack of integration and cohesive workflow presents challenges, causing delays and increasing the likelihood of overlooking critical vulnerabilities.

To address these issues, this project introduces Net-Caerus: an Integrated Ethical Hacking System designed to consolidate the necessary tools within one user-friendly platform. Net-Caerus provides both a web-based platform and a downloadable Linux toolkit, empowering cybersecurity professionals to efficiently assess vulnerabilities without navigating multiple applications. While the project includes minimal educational resources to aid practical usage, its primary focus is a toolset that enables precise and comprehensive penetration testing.

1.1 Current System

In the current cybersecurity landscape, ethical hackers face significant challenges due to the lack of a centralized, multifaceted toolkit. The existing range of tools requires professionals to operate across several platforms, complicating their workflow and leaving room for oversight. Current systems for penetration testing are generally segmented into specialized tools that focus on specific areas, such as network scanning, vulnerability assessment, social engineering, or digital forensics. As a result, cybersecurity professionals often have to move between multiple tools, increasing the complexity of their work and affecting productivity.

Moreover, many of these tools lack a user-friendly interface, requiring a steep learning curve and advanced expertise. While there are some educational resources available for ethical hacking, these often do not integrate well with tool usage, making it difficult for users to understand how to effectively apply these resources within real-world contexts. Additionally, the tools currently available are either web-based or confined to specific operating systems, limiting their usability across diverse environments.

Given these gaps, the current system does not adequately support cybersecurity professionals in conducting comprehensive assessments. This fragmented approach is ill-suited to the dynamic and rapidly evolving nature of cybersecurity threats. An integrated solution that brings together these disparate tools into a cohesive, accessible interface is needed to enhance workflow efficiency, reduce oversight, and promote comprehensive vulnerability assessments.

1.2 Motivation and Significance of the Project

The concept for Net-Caerus was inspired by the rising demand for comprehensive and adaptable tools in cybersecurity. With cyber-attacks becoming increasingly sophisticated, there is an urgent need for ethical hackers to be equipped with all-encompassing toolkits that not only facilitate efficient assessments but also address a broad spectrum of potential vulnerabilities. Current resources are limited in scope and usability, often failing to provide the depth required to keep pace with modern cyber threats.

Net-Caerus is motivated by the need to streamline the workflow of cybersecurity professionals and to address the limitations posed by current systems. It aims to provide ethical hackers with a centralized, efficient, and accessible toolkit that consolidates essential functions. A critical influence for the project is platforms like HackTheBox, which primarily offer educational content for users. In contrast, Net-Caerus prioritizes the integration of practical tools over education, emphasizing a real-world application of tools and techniques rather than theoretical learning alone.

The project's significance lies in its potential to enhance cybersecurity defenses by equipping professionals with a unified toolkit that is both efficient and user-friendly. By addressing the fragmentation in ethical hacking resources, Net-Caerus seeks to contribute to the overall advancement of cybersecurity practices. With minimal but targeted educational resources to guide practical tool usage, this project also aims to encourage responsible and ethical hacking practices.

Figure 1 Net-Caerus Logo

1.2.1 Aim of the Project

The primary aim of Net-Caerus is to create a revolutionary ethical hacking toolkit that merges essential tools within a cohesive and intuitive interface. This toolkit is designed to serve as a versatile, practical resource for cybersecurity professionals, improving their workflow and enhancing their ability to identify and mitigate vulnerabilities. Net-Caerus strives to be both an adaptable and advanced solution, providing users with the means to conduct comprehensive penetration testing through a streamlined approach.

1.2.2 Objectives

To achieve its aims, Net-Caerus focuses on several core objectives:

- Toolset Integration: Develop a centralized toolkit that encompasses a wide range of ethical hacking tools, allowing professionals to conduct assessments without switching between multiple applications. This integration will enhance workflow efficiency and reduce oversight in vulnerability assessments.
- User-Focused Design: Create a user-friendly interface that facilitates a seamless experience. The design will aim to reduce the learning curve, making the toolkit accessible to both novice and experienced users in the cybersecurity field.
- Linux Compatibility: Ensure compatibility with the Linux operating system by developing a specialized Linux version of the toolkit. This will cater to the distinct needs of Linux users and support a platform that is widely preferred by cybersecurity professionals.
- Advanced Tools and Unique Features: Incorporate innovative tools, such as an Arduino-powered Wi-Fi jammer and a Payload Generator. These tools, which are not commonly found in other toolkits, will set Net-Caerus apart and enhance its effectiveness in addressing contemporary cybersecurity challenges.
- Minimal Educational Component: Include a small educational module to aid in practical application, providing tutorials or guides that help users understand the correct usage of each tool within the toolkit.

1.3 Scope and Limitations of the Project

The scope of the Net-Caerus project is focused on developing a comprehensive toolkit that consolidates the essential tools needed for effective ethical hacking. The toolkit will be available as both a web-based platform and a downloadable Linux version, providing flexibility for cybersecurity professionals who may need to operate across different environments.

The toolkit will include a range of tools and utilities for various aspects of penetration testing, such as:

IP Address Utilities: Tools that assist with IP address detection, hostname identification, and network scanning.

Hashing and Encryption: Functions for secure data handling, helping ethical hackers assess encryption strength.

Network and Port Scanners: Tools for analyzing network activity and detecting open ports, essential for identifying potential vulnerabilities.

Web Interaction Tools: Tools to facilitate interactions with web servers, such as reading protected URLs and handling HTTP responses.

Social Engineering and Remote Access Tools: Aiding in more advanced penetration testing scenarios that may require social engineering or remote access.

Arduino-Based Tools: Including an Arduino-powered Wi-Fi jammer, which provides unique functionality not commonly found in other toolkits.

The limitations of the project include:

Resource Constraints: Budget, time, and personnel limitations may impact the depth and breadth of features included in Net-Caerus. These constraints may restrict the number of tools integrated into the initial release of the toolkit.

Compatibility Challenges: Ensuring compatibility across various operating systems and network configurations may present technical challenges. While the toolkit is designed to be Linux-compatible, certain functionalities may encounter limitations depending on the user's setup.

Ethical and Legal Considerations: Developing tools for ethical hacking entails navigating ethical and legal boundaries. There is an inherent risk of misuse; therefore, stringent safeguards must be implemented to prevent unauthorized access and unethical use.

Maintenance and Updates: Net-Caerus will require continuous updates to remain effective against evolving cybersecurity threats. Ensuring ongoing maintenance may present resource challenges.

Scope Creep: As development progresses, there may be a tendency to introduce additional features. Managing this scope creep is essential to prevent resource overruns and delays.

1.4 Chapter Outline

The remainder of this report is organized as follows:

Chapter 2 – Literature Review: This chapter provides an in-depth review of existing literature on ethical hacking tools and methodologies. It examines the strengths and weaknesses of available resources, identifies gaps, and discusses how Net-Caerus aims to address these shortcomings.

Chapter 3 – Methodology: This chapter outlines the methodology used to develop Net-Caerus, detailing the research, planning, and development process. It discusses requirement gathering, system design, and integration, with an emphasis on tool compatibility and usability.

Chapter 4 – System Design and Development: This chapter presents the detailed system architecture, user interface design, and database integration. It covers each stage of the development process and explains how specific tools were integrated into the platform.

Chapter 5 – Implementation and Testing: This chapter covers the implementation of each tool within Net-Caerus, describing the testing protocols used to validate functionality and ensure the toolkit meets cybersecurity standards.

Chapter 6 – Conclusion and Future Work: This chapter summarizes the project's contributions to ethical hacking, discusses the limitations faced, and outlines potential areas for future development and enhancement.

CHAPTER 2 – LITERATURE REVIEW

In this chapter, we critically evaluate existing systems similar to the proposed Ethical Hacking System. We analyze their strengths and weaknesses, user satisfaction, and requirement satisfaction. This review places the new system in the context of existing literature, identifies gaps, and highlights the need for an integrated approach.

2.1. Critical Evaluation of Existing Solutions

In this section, we behavior an in-depth evaluation of 7 most important systems related to ethical hacking equipment and academic resources. We summarize their advantages and drawbacks,

emphasizing person pleasure and requirement fulfillment. Tables and figures are used to spotlight vital factors and comparisons among these systems.

1. **The Autonomous Security Analysis and Pentesting (ASAP)**

The Autonomous Security Analysis and Pentesting (ASAP) [1] tool is specifically designed to deploy automated exploit tools such as Metasploit to target and test corporate networks. Its primary function is to streamline penetration testing by reducing the manual effort typically required, increasing efficiency through the use of predefined attack protocols and automated tools.

Key Advantages: The main advantage of ASAP lies in its automation capabilities, allowing repetitive or routine tasks to be performed without constant human intervention. By using predefined attack guidelines, ASAP can quickly identify vulnerabilities and attempt to exploit them, significantly speeding up the testing process compared to traditional, manual penetration testing methods. This automation enables penetration testers to focus on other higher-level tasks while relying on ASAP for a basic layer of network security testing. But this is temporary.

Limitations: However, automation is also a limitation. The device's reliance on a fixed set of predefined attack guidelines can make it rigid in its operation, and some vulnerabilities may not be controlled if outside these predefined rules. Moreover, a lack of human oversight can lead to suboptimal results, especially in complex attack scenarios that require nuanced understanding to interpret and respond to complex security flaws. This lack of rigidity and adaptive intelligence limits ASAP's ability to conduct comprehensive assessments, as it lacks the flexibility to change its approach based on unique, unpredictable security contexts.

User satisfaction and needs: User satisfaction with ASAP is generally moderate, and the device receives positive feedback for the convenience and time savings it provides through automation. However, users often express a desire for a more adaptable solution. ASAP's performance meets the requirements for efficient, rapid testing but falls short of providing the thoroughness required for comprehensive security assessments. As a result, users who prioritize deep, adaptive testing for complex environments may find ASAP's functionality partially inadequate because it doesn't quite match their expectations for a comprehensive security tool.

2. **Bash inside the Wild: Language Usage, Code Smells, and Bugs**

The paper “Bash inside the Wild: Language Usage, Code Smells, and Bugs” [2] presents an extensive empirical study of usage patterns, code quality issues, and common errors associated with the Bash scripting language. The study is based on an analysis of over a million open source scripts obtained from a GitHub repository, making it one of the most comprehensive assessments of real-world Bash scripting practices.

Main Contributions and Findings: This study is notable for its in-depth study of how Bash is used in practice, documenting frequent coding errors, and providing insights into common “code smells”. By analyzing such a large dataset, the study identified a series of best practices and

common pitfalls, providing practical guidance on how to write more effective and maintainable Bash scripts. A key finding of the study is that larger and more complex Bash scripts are more prone to errors, highlighting the challenges associated with the complexity and maintainability of Bash scripts. These insights emphasize the need for caution and additional testing when using extended scripts, as they are more error-prone and more difficult to manage.

Limitations: The study focuses exclusively on Bash scripts, limiting its applicability to other scripting languages or tools commonly used in Unix-like environments. While it provides valuable insights for Bash, its findings are not easily transferable to other shell languages or to broader ethical hacking tools that may use alternative scripting languages.

User Satisfaction and Relevance: Satisfaction was significantly higher among users who rely heavily on Bash for automation, configuration, and administration tasks in Unix-like environments. These users appreciated the study's insights into Bash-specific best practices and error prevention techniques. However, the study's relevance to those seeking guidance on scripting in a broader context or other ethical hacking tools beyond Bash is limited. While it successfully addresses the need for a focused, data-driven exploration of Bash scripting, its usefulness is limited to Bash, leaving a broad knowledge gap in ethical hacking or scripting for a variety of tools in other domains.

3. Backdoor Remote Access Trojan (BRAT) Phishing Threat Analysis

The research paper "Backdoor Remote Access Trojan (BRAT) Phishing Threat Analysis"[3] explores the development and in-depth analysis of Remote Access Trojan (RAT) malware created using Python. The study presents a comprehensive approach to creating and detecting RAT malware, drawing attention to the potential risks and vulnerabilities associated with Python-based malware development.

Key Findings and Methodology: The study highlights how Python's versatility and accessibility make it an effective language for both crafting offensive tools such as RATs and defensive measures. The study details the entire lifecycle of Python-based RATs, from coding and deployment to detection strategies, allowing for a comprehensive understanding of the mechanisms and countermeasures of this type of malware. A key insight drawn from this analysis is the limited ability of traditional antivirus software to effectively detect these RATs. Testing showed that only 11 out of 71 antivirus solutions could identify the created RATs, highlighting the challenges that standard antivirus software faces in detecting custom Python-based malware.

Limitations: However, because the study does not expand its scope to other forms of malware or cybersecurity challenges beyond RATs, the paper's narrow focus on RATs limits its broad applicability to other types of cyberthreats. While the study provides valuable insights into RAT-specific tactics, it may lack the versatility required for a more comprehensive malware analysis approach or to address a wider range of cybersecurity threats.

User Satisfaction and Practical Relevance: Because the paper provides a comprehensive technical examination of Python-based RATs, readers interested in understanding specific cyberthreats have a high user satisfaction rate for this paper. Its findings are particularly important for cybersecurity professionals, researchers, and ethical hackers seeking a detailed understanding of RAT development and detection methods. However, the study falls short for users seeking a comprehensive toolkit or broader threat insights. While it meets the need for in-depth RAT threat analysis, it does not provide a general solution for broader cybersecurity applications or multi-threat defense strategies.

4. Role of Ethical Hacking in System Security

'Role of Ethical Hacking in System Security' [4] refers to understanding the fundamental roles fulfilled by ethical hacking, its meaning, and particularly, its importance in enhancing computer system security. This study conceptualizes ethical hacking as the authorized proactive process of search and eliminating the loopholes in systems to avert any security breach The situation is further aggravated by the ability of the internet to connect many people, which has seen an increase in cyber threats. Due to this situation, the paper stresses that the services of ethical hackers in protecting malicious assails have become a basic prerequisite.

Key Findings and Methodology: The research analysis presents the over reign significance of ethical hacking in the present day, especially with organizations increasingly becoming targets to sophisticated hackers. In this manner, ethical hackers are essential in advance efforts to find and fix holes to prevent unwarranted intrusions and attacks. The current research study furthers the appreciation and understanding concerning ethical hacking fundamentals by addressing the ways in which ethical hackers all over the world work towards enhancing the security of systems within the confines of the law.

Limitations: The aforementioned positive aspects are tempered by the fact that the research does not contain specific, practical implementations of ethical hacking. The study is more concerned with explaining what ethical hacking is or providing basics about it instead of addressing powerful tools, advanced techniques, or even samples of practical situations. This narrows down the reach of the study to the readers who are looking for any practical use of ethical hacking as a business.

User satisfaction and practical relevance: User satisfaction is moderate, especially for those seeking a theoretical examination of the nature of ethical hacking. The research justifies providing elements that help to better understand.

5. Some Ethical Hacking Possibilities in Kali Linux Environment

The document entitled "Some Ethical Hacking Possibilities in Kali Linux Environment" [5] investigates the functionality of hacking operating systems, in particular, the Kali Linux operating system designed for ethical hacking. The main objective here is to demonstrate how Kali Linux is

a functional tool for security hacking, emphasizing the availability of various strong weaponry for all forms of security attacks and assessment simulations.

USPs and Benefits: The current analysis focuses on the distinct advantage which potassium chloride and malay halfmoon together can be viewed not only as simple operating systems but as something far more complex, a complete package filled with strategically pluripotent tools aimed at achieving penetration testing, risk assessment, and network evaluation among other activities. Each tool within Kali Linux is meant for a certain security activity, thus ethical extremists have the possibility of many types of 'virtual' attacks and analyses within certain limits. This Auxiliary weaponry increases the degree of usability of Kali Linux, particularly for those who have been in the field of ballet, or rather the art of hacktivism for quite a while.

Efficacy and Challenges: Among the many tools that are found in Kali Linux, and that all have numerous benefits, there is hardly a simple and straightforward one that would be particularly beneficial to the beginners. Most of the tools present its usage with no particular instructions other than the general ones, which command the user in a certain hierarchy. This sophistication and usefulness may be a barrier to entry for such users since it is usually difficult to operate the tools as well as to make studies with them without prior cyber security skills.

User Experiences and Market-Location: Experienced ethical hackers do not complain, rather express joy in the fact that Kali Linux has all the tools that are required to address very challenging security problems. The extensiveness of the software system is not a problem to them.

6. Web Application Vulnerability Scanning Tools

This case also tells of the case study 'Web Application Vulnerability Scanning Tools' [7] where the various web application vulnerability scanning tools in common use are assessed based on their effectiveness particularly their merits demerits and the best practices that can be adopted by users to enhance the accuracy of detection of such vulnerabilities. This elaboration shows how important it is not only to rely on only one tool but several tools along with techniques in order to enhance and increase the chances of detecting every single vulnerability in web applications.

Key Findings and Best Practices: In the research, it was observed that the use of different scanning tools is effective as each tool is capable to detect certain types of vulnerabilities, hence the more the tools the more the chances of earlier detection of vulnerabilities. Another key point is enhancing the efficacy of scanning where scan policies are tailored to suit specific conditions, especially for tools like Nessus, helps to narrow down the parameters of interest and reduce the chances of misdiagnoses. Additionally, scans are more likely to yield a greater number of vulnerabilities when performed on different occasions due to the fact that certain vulnerabilities may not always be present unless under particular system conditions or loads.

Challenges and Limitations: Despite the benefits, the study identifies limitations in both the tools and the scanning process. Certain tools fail to detect specific vulnerabilities, indicating that no single tool can provide complete coverage. Additionally, setting up customized Nessus policies

can be complex and time-consuming, requiring a high level of technical expertise. The timing of scans also presents challenges; while multiple scan sessions increase coverage, they may disrupt normal operations and miss vulnerabilities that only appear sporadically or under unique circumstances.

User Satisfaction and Practical Relevance - The satisfaction level is really up there looking at the case of users who are interested in carrying out extensive as well as all-inclusive vulnerability detection. The case study is on a multi-scanning method which I find is appropriate for the effective dosages used and the customization of policies as well. But the setup and scan scheduling problems may hinder the user who doesn't have much technical know-how or is simply seeking for a user-friendly feature. In as much as the case study provides a useful way of enhancing the current way of vulnerability scanning, it also stresses the need to consider the feasibility of doing too much of comprehensive scans that are intrusive.

Table 1 literature results

System	Pros	Cons	User Satisfaction	Requirement Satisfaction
ASAP	Reduces human effort, enhances efficiency through automation.	Limited attack policies, potential suboptimal outcomes.	Moderate	Partial
Bash in the Wild	Comprehensive real-world usage analysis, identifies common errors.	Focused on Bash, error-prone larger scripts.	High	Specific to Bash
BRAT Phishing Threat Analysis	Detailed RAT creation and detection methodology, highlights malware risks.	Narrow focus on RATs, limited antivirus detection.	High	Specific to RATs
Role of Ethical Hacking in System Security	Highlights ethical hacking importance, emphasizes need for countermeasures.	Lacks practical application details, general overview.	Moderate	Theoretical concepts only
Ethical Hacking in Kali Linux	Comprehensive hacking tool collection, specialized for attack types.	Steep learning curve, overwhelming tools.	High for experts	High for experienced users
Web Application Vulnerability Scanning Tools	Enhanced detection with tool variety, improved scan accuracy with custom policies, broader vulnerability capture with varied timing.	Missed specific vulnerabilities, complex and time-consuming policy setup, operational disruption with varied timing.	High	High, with setup challenges

2.2 Placing Original Work in Context

The proposed Ethical Hacking System is designed to provide a comprehensive solution that addresses the restrictions of current gear at the same time as leveraging their strengths. This challenge sticks out with the aid of integrating a big range of ethical hacking equipment right into a single, person-friendly platform, followed via an in-depth academic factor. Here's an in-depth breakdown of its context:

Integration of Tools:

Current State: Existing ethical hacking equipment are often scattered throughout diverse platforms, requiring customers to replace between more than one packages for unique tasks, including community scanning, vulnerability assessment, and exploitation.

Proposed Solution: The new device will consolidate these various gears right into a unmarried platform. This integration aims to streamline workflows, reduce the time spent on assignment switching, and enhance general productiveness. Users might be capable of perform complete penetration exams without having to juggle more than one software programs.

Interface That's Easy to Use:

Present Situation: Since many ethical hacking tools are made with advanced users in mind, novices may find it challenging to traverse their complicated interfaces. This intricacy may lead to a challenging learning curve and possible tool misuse.

The system will have an easy-to-use interface that is suitable for users of all experience levels, according to the suggested solution. The functioning of each tool will be explained to users through straightforward navigation and clear, contextual support. Beginners will find it easier to enter the market thanks to this emphasis on user experience, while seasoned users will find it more efficient. (Linux Base)

All-inclusive Educational Materials:

Current Situation: Practical tool usage and ethical hacking education resources frequently diverge. Users could discover theoretical information lacking sufficient application in real-world situations, creating a gap between learning and doing.

Proposed Solution: A comprehensive educational portion with in-depth lessons, comprehensive manuals, and practical laboratories will be part of the proposed system. These materials will offer relevant, hands-on training that is directly connected to the integrated tools. In order to promote appropriate use, the instructional component will also address ethical issues and legal requirements.

Security and Ethical Considerations:

Current State: Ethical hacking tools must be used responsibly to prevent misuse. Some current tools lack sufficient security measures to ensure ethical use, leading to potential legal and ethical issues.

In particular, I will not take any steps to include a database in the educational website where I can download this toolset. The reason is that Open-Source Linux Projects do not collect anyone's data. Also, no website is 100% secure. If this website is hacked for some reason, and if anyone enters the database here, the risk of all the information of everyone who downloaded this toolset being leaked is 100%.

Especially in today's world, such tools have not invaded any kind of market. Cybersecurity companies often create their own tools and never give them out. Some tools are taken from outside and updated as they need.

If, for some reason, this toolset is obtained by a person working at XYZ Company, the hacker who hacked this site can access the database and obtain that person's information, and there is a possibility that both that person and that organization will face serious danger. There is definitely a good side and a bad side to everything. A knife can cut vegetables and fruits, and it can also harm a person. This is also something similar. It has a good side and a bad side.

2.3 Major Issues Surrounding the Topic

The important troubles surrounding the topic of moral hacking gear and structures encompass the fragmentation of equipment, person interface complexity, instructional gaps, and protection and moral worries. The fragmentation of equipment forces customers to interchange between more than one packages, that can disrupt workflow and cause inefficiencies. Many of those gear is not user-pleasant, mainly for beginners, ensuing in a steep getting to know curve and capability misuse. Furthermore, current academic resources regularly do not align with sensible device utilization, leaving users without good enough training to correctly appoint those tools. Finally, tools need to be designed with robust security measures and ethical concerns to prevent misuse and make sure compliance with legal standards.

2.4 Relationships and Gaps in Existing Literature

The project's literature research identified a sizable gap in the current inventory of instructional materials and ethical hacking tools. Although there are a number of tools that are excellent in particular domains, including vulnerability assessment, exploitation, or network scanning, there are notably few integrated solutions that pair these useful tools with extensive training materials. Because of this lack of connection, users frequently have to switch between several programs, which makes workflow disjointed and ineffective.

Furthermore, there is currently no user-friendly platform available on the market that adequately handles ethical hacking education as well as tool usage. Users, especially those who are new to the

topic of ethical hacking, have a challenging learning curve due to the intricacy of these tools and the fragmented nature of available teaching materials. This piecemeal and inconsistent approach reduces the efficacy of ethical hacking techniques and impairs user competency.

The gap that has been found highlights the urgent need for an integrated strategy that combines instructional materials and useful tools into a single, user-friendly platform. An all-inclusive system like this would improve user proficiency, optimize workflows, and offer a smooth setting for carrying out ethical hacking operations. By addressing these shortcomings, the suggested system seeks to close a significant gap in the market by providing a comprehensive solution that meets the requirements of both inexperienced and seasoned users.

In addition to incorporating a wide range of ethical hacking tools, the proposed system will also include extensive teaching materials that are directly related to the use of the tools. Through this connection, users will be able to effectively use their abilities in real-world circumstances by bridging the gap between theoretical understanding and actual application. While the system's sophisticated features and operations will meet the needs of seasoned users, its user-friendly interface will help decrease the entrance barrier for newcomers.

The suggested system aims to transform ethical hacking by offering a centralized, all-inclusive, and intuitive platform that improves workflow efficiency, encourages ongoing learning, and equips users to become skilled ethical hackers.

2.5 Resolving Conflicts and Identifying Contributions

Because of their restricted scope or lack of real-world application, earlier research in the topic of ethical hacking frequently has conflicting findings. Instead of offering a comprehensive and integrated solution, a lot of the current tools and instructional materials frequently focus primarily on one or more specific facets of ethical hacking, including network scanning, vulnerability analysis, or exploit building. This narrow focus leads to fragmented toolsets, disjointed workflows, and difficulties converting theoretical knowledge into actual skills.

The suggested approach aims to address these issues by providing a comprehensive answer that combines extensive instructional material with a wide variety of ethical hacking tools. This integration is essential because it fills in the gaps between theoretical knowledge and real-world application, giving users a smooth and efficient environment in which to engage in ethical hacking activities. Through the resolution of existing systems' shortcomings and utilization of their advantages, this project seeks to build a completer and more reliable platform for ethical hackers.

The suggested system's ability to seamlessly integrate instructional materials with practical tools is one of its main advantages. Users will have access to a comprehensive toolbox that offers comprehensive instructional resources directly related to tool usage, in addition to making ethical hacking activities easier. By integrating the tools, this integration guarantees that users will not

only comprehend the theoretical principles of ethical hacking but will also have practical experience putting these principles into practice.

Furthermore, the system differs from other solutions in that it places a strong emphasis on encouraging moral behavior in cybersecurity. The system seeks to foster a responsible and ethical hacking culture within the cybersecurity industry by integrating ethical rules, legal standards, and best practices into the instructional content. In order to ensure that ethical hacking is carried out in a way that is both legal and moral, this proactive approach to ethics is essential for creating a more secure digital ecosystem.

The new technology is also intended to greatly improve the user experience. Particularly for inexperienced users, the user-friendly layout, simple navigation, and interactive learning modules foster a supportive learning environment. Expert practitioners' needs are met by sophisticated features and functions, which enable the system to be customized for a broad range of use cases and skill levels.

In conclusion, the suggested method resolves issues raised in earlier research by offering an all-encompassing, user-focused, and integrated approach to ethical hacking. The system makes a substantial contribution to the area by supporting ethical practices in cybersecurity, increasing educational outcomes, and improving user experience through the combination of different technologies, educational content, and ethical guidelines.

2.6 Further Research Directions

The Integrated Ethical Hacking System's continued relevance, efficacy, and evolution depend heavily on the routes future research takes. Future research should concentrate on the following important areas to improve the system's capabilities and handle new cybersecurity challenges:

1. Regularly adding new technologies to the toolkit

Significance: The swift progression of cybersecurity technology demands frequent modifications to the toolkit in order to remain up to speed with new threats and weaknesses.

Areas of Focus:

Including New Tools: Examine how to combine state-of-the-art methods and technologies for threat intelligence collection, penetration testing, and network scanning.

Enhanced Automation: To save manual involvement and increase efficiency, develop automated procedures for patch management, security configuration, and vulnerability assessment.

Cloud Security Integration: To handle security issues related to cloud environments, like data protection, access control, and compliance, include cloud security technologies and procedures.

2. Improving the Educational Framework: In order for ethical hacking practitioners to stay current with emerging risks, methods, and best practices, they must engage in ongoing learning.

Key Topics: Advanced Instructional Materials: Provide sophisticated training materials that address subjects including malware analysis, incident response, and sophisticated exploitation tactics.

Certification Programs: Provide ethical hacking skills and knowledge validation through certification programs, which also offer prospects for job progression and recognition.

Real-world scenarios, simulations, and interactive challenges can be added to hands-on practical laboratories to enhance learning and skill application.

3. Ensuring Sturdy Security Protocols

Significance: Sturdy security measures are essential for guarding against hostile assaults, illegal access, and data breaches on the system.

Areas of Focus:

Granular access control techniques should be implemented to limit privileges according to user roles and responsibilities.

Data Encryption: To guarantee confidentiality and integrity, use robust encryption solutions for both data in transit and data at rest.

Intrusion Detection and Prevention: To identify and address questionable activity and possible security problems, implement intrusion detection and prevention systems (IDPS).

Examining the Integration of Innovative Tools 4. Importance: The system's capabilities are increased by the integration of innovative tools, making ethical hacking techniques more thorough and successful.

Areas of Focus:

IoT Security Devices: Investigate how to incorporate IoT security devices and protocols to counter new risks that are appearing in IoT ecosystems, like network security, device vulnerabilities, and privacy concerns.

Blockchain Security: Examine methods and tools for safeguarding bitcoin transactions, auditing smart contracts, and reducing risks associated with blockchain technology.

AI and Machine Learning: To improve proactive security measures, use AI and machine learning algorithms for behavior analysis, anomaly identification, and threat modeling.

5. Real-time threat detection and automated updates

Importance: The system's agility, responsiveness, and resilience against changing cyber threats are improved by automated updates and real-time threat detection capabilities.

Focus Areas: Automated Patch Management: To guarantee prompt mitigation of vulnerabilities that are known, develop automated procedures for patch management, vulnerability screening, and software updates.

Threat Intelligence Integration: For proactive incident response and real-time threat detection, integrate threat correlation algorithms, threat hunting capabilities, and threat intelligence feeds.

Behavioral Analytics: Use anomaly detection and behavioral analytics methods to quickly spot odd trends, questionable activity, and possible security breaches.

The Integrated Ethical Hacking System can continue to be relevant, efficient, and competitive in the face of the ever-changing cybersecurity ecosystem by concentrating on these future research objectives. Ongoing development work will guarantee that the system stays up to date, flexible, and able to successfully handle novel kinds of cyberthreats.

In summary, the Integrated Ethical Hacking System offers a comprehensive, user-friendly platform that combines useful tools with instructional resources, marking a significant leap in the field of cybersecurity. By bridging the gap between theoretical knowledge and actual application, the system's holistic approach improves user competence and encourages ethical cybersecurity practices.

Future research and development priorities include adding new technologies to the toolkit on a regular basis, improving the educational framework with cutting-edge training courses and certification programs, making sure that access control and data encryption are strong security measures, investigating the integration of cutting-edge tools like blockchain security and Internet of Things security, and putting automated updates and real-time threat detection capabilities in place.

The Integrated Ethical Hacking System will continue to be relevant, efficient, and resilient in tackling changing cybersecurity issues by concentrating on these future research avenues. The continuous development work will strengthen the security of the digital ecosystem and enable ethical hackers to efficiently fend off new attacks.

CHAPTER 3 – ANALYSIS

3.1 Introduction

The Ethical Hacking System, featuring web and Linux-oriented toolkits, meets essential demands of security specialists, ethical hackers and training institutes by offering an all-encompassing toolkit for any and all penetration testing and system security assessment. This part of the study seeks to present an evaluation of the toolkit’s components, design, usability, and security. Each of the tools has its purpose in ethical hacking and cybersecurity, and therefore the user is capable of interacting with real issues of security.

3.2 Overview of Tools and Functionalities

The Linux-based toolkit comprises a variety of tools ranging from basic network and web reconnaissance to advanced intrusion and phishing tactics. The tools are categorized based on their purpose and functionality as follows:

3.2.1 Network and Website Scanning Tools

- **Scan Website:** This tool performs a detailed scan of the target website, collecting information about its structure, potential vulnerabilities, and security configurations. This foundational tool is essential for identifying points of interest for further analysis.
- **Resolve Hostname to IP Address:** Converts website domain names into IP addresses, aiding in identifying server locations and further network mapping.
- **Whois Lookup:** Retrieves registration and ownership information of domain names, which is critical in understanding the administrative boundaries and contact points of a target website.

3.2.2 Denial of Service (DoS) and Network Disruption

- **Perform DoS Attack:** This tool allows users to simulate a Denial of Service attack for educational purposes, demonstrating how flooding a server can render it unavailable. Due to its potential impact, this tool is restricted to secure, sandboxed environments.
- **Change MAC Address:** Spoofing the Media Access Control (MAC) address can be useful in evading network monitoring and testing certain network restrictions. This tool simulates MAC address modification, providing a deeper understanding of network tracking and identification mechanisms.

3.2.3 Wireless Network and Authentication Tools

- **Find Wifi Password:** This tool explores vulnerabilities in local Wi-Fi networks, helping users identify weak access points and understand Wi-Fi security protocols.
- **HTTP Authentication:** This tool attempts to access areas of websites protected by HTTP Basic Authentication, simulating attacks that seek unauthorized entry into secured sections of a web application.

3.2.4 Reconnaissance and Information Gathering

- **Find Instagram Profile:** Automates profile lookups for social media reconnaissance, demonstrating how publicly available information can be gathered and analyzed.
- **Cookies Information:** This tool extracts cookie data from websites, aiding in understanding how session management and user tracking work online. It is especially useful for testing cookie-related vulnerabilities, such as session hijacking.

- **Scan Website Admin Page:** This tool scans websites for hidden or default admin page entries, which could serve as entry points for unauthorized access if left unprotected.

3.2.5 Phishing and Credential Collection

- **Goldphish Phishing Server:** This server simulates phishing attacks, providing a realistic environment for training users on spotting and avoiding phishing attempts.
- **Cam Phishing:** This tool is designed to demonstrate the dangers of malicious links by attempting to simulate remote access to webcam feeds in a controlled environment.

3.2.6 Payloads, Malware, and Remote Control

- **Payload Creating:** Enables users to generate custom payloads that can be deployed to test endpoint security defenses.
- **Remote Connecting**: Provides the functionality to connect remotely to devices for educational purposes, demonstrating the risks associated with unauthorized remote access.
- **Key Logger:** Simulates keystroke logging to illustrate how attackers capture sensitive information. This tool is restricted to educational setups and secure environments to prevent misuse.

3.2.7 Surveillance and CCTV Access

- **CCTV Hacking**: This tool tests the security of IP-based camera feeds and helps users understand how insecure configurations can expose visual data.

3.3 Feasibility Analysis

To evaluate the practicality of the Ethical Hacking System, feasibility has been assessed in terms of technical, operational, and economic dimensions.

3.3.1 Technical Feasibility

The toolkit was designed using a Linux-based infrastructure, which is ideal for cybersecurity applications due to Linux's robust security, open-source support, and compatibility with various penetration testing tools. Leveraging Python and shell scripting, each tool integrates efficiently into the Linux environment, ensuring smooth functionality and interoperability.

The development of the website user interface was directed at ensuring accessibility and responsiveness, thus utilizing cross-device compatible frameworks. The very design of the system was focused on modularity, which allows conjoining various tools and extensions with ease thus making the system easily extensible and making its future guaranteed.

3.3.2 Operational Feasibility

The system's design focuses on providing a user-friendly experience, allowing users of varying expertise to navigate and utilize each tool without extensive training. Comprehensive

documentation, instructional guides, and usage examples are provided for each tool, enhancing usability.

Operational feasibility is also achieved through the ethical framework built into the system. By embedding guidelines and best practices on responsible usage and legal compliance, the toolkit ensures users are well-informed about the ethical boundaries of ethical hacking.

3.3.3 Economic Feasibility

The Ethical Hacking System was developed with an economically sustainable approach, relying primarily on open-source technologies, minimizing licensing costs, and allowing for community-driven enhancements. The choice of Linux and Python as core components reduces dependency on costly proprietary software, which aligns well with educational and small-business budget constraints.

3.3 Security and Ethical Implications

Given the power and sensitivity of these tools, it is crucial to emphasize that the Ethical Hacking System is intended solely for educational purposes in controlled environments. Each tool is designed with built-in safeguards to prevent accidental or malicious misuse. Additionally, user access is logged and monitored to ensure adherence to ethical standards. The educational platform also includes comprehensive documentation on ethical hacking best practices, legal implications, and guidance on responsible use.

3.4 Fact-Finding Techniques

To gather comprehensive data, various fact-finding techniques were employed. These techniques ensured a thorough understanding of user requirements and current challenges in the field of ethical hacking.

Interviews:

Conducting interviews with cybersecurity professionals, penetration testers, and IT managers to gain qualitative insights into their experiences, challenges, and expectations for the new system.

Document Reviews:

Reviewing existing documentation on ethical hacking practices, tool usage, and educational resources. This provided a contextual understanding of current standards and best practices.

Questionnaire:

Distributing a detailed questionnaire to a broad audience of cybersecurity professionals. The questionnaire collected both qualitative and quantitative data on tools currently used, challenges faced, and preferences for new system features.

The combination of these techniques provided a comprehensive dataset that informed the requirement analysis.

3.5 Requirement Analysis

The requirement analysis synthesizes the data collected to identify the needs and expectations of potential users of the Integrated Ethical Hacking System. This analysis revealed the need for a unified toolkit, ease of use, comprehensive documentation, and integrated educational resources. The findings emphasized the importance of having a versatile system that can adapt to various user requirements while ensuring a seamless user experience.

3.6 Functional Requirements

The functional requirements define the core capabilities that the system must deliver to effectively meet user needs and support open-source accessibility:

Unified Toolkit: The system should provide a broad set of tools for penetration testing and ethical hacking within a cohesive, user-friendly interface. This integration will streamline workflows and enhance user efficiency.

Educational Resources: An educational section should be available, featuring tutorials, guides, hands-on labs, and best practices in ethical hacking. This supports continuous learning, skill enhancement, and the responsible use of the tools.

Web-Based Platform and Downloadable Toolkit: The system should be accessible both through a web-based platform and as a downloadable Linux toolkit, allowing users to select their preferred access method. This flexibility increases usability for diverse operating environments.

Open-Source Accessibility: As an open-source, Linux-based project licensed under the MIT License, the toolkit will be freely available for download without restrictions. This ensures transparency, community involvement, and ease of access, allowing users to adapt and improve the tools as needed.

License and User Rights: The project will operate under the MIT License, granting users the freedom to use, modify, and distribute the toolkit. This open-source approach aligns with the values of collaborative development while establishing rights and limitations for responsible use.

3.7 Non-Functional Requirements

Non-functional requirements define the quality attributes of the system, ensuring it meets performance standards and user expectations:

Reliability: The system should be highly reliable, with minimal downtime and robust fault tolerance to ensure consistent availability and performance.

Portability: The system must be compatible with various Linux distributions and accessible via multiple web browsers, ensuring broad usability.

Performance: The system should perform efficiently, with fast response times and minimal processing delays, even under heavy user loads.

Security: Robust security measures must be implemented to protect user data, ensure secure communications, and prevent unauthorized access. This includes encryption, authentication, and regular security audits.

Usability: The system should feature an intuitive and user-friendly interface to enhance user experience and minimize the learning curve. Clear navigation, comprehensive documentation, and responsive design are key aspects of usability.

3.8 Success Factors

Success factors are the prerequisites that must be met for the system to be used effectively. These include:

Instruction for Users:

Sufficient Training Materials: Providing thorough training materials, such as user manuals, tutorials, and video guides, to guarantee users comprehend the features of the system and know how to use them efficiently.

Webinars and Training Sessions: Holding webinars and training sessions to promote practical learning, respond to user inquiries, and offer advice on how to use the system for ethical hacking techniques.

Thorough Documentation: Producing thorough documentation that addresses every facet of the system, such as setup parameters, installation guidelines, troubleshooting procedures, and best practices for ethical hacking assignments.

Continual Updates:

Handling Emerging Threats: Making sure that updates, bug fixes, and security enhancements are applied to the system on a regular basis to combat evolving cyber threats and vulnerabilities.

Adding New Tools and Technologies: Adding new tools and technologies to the system will boost performance, increase functionality, and keep it compliant with industry best practices and standards.

Enhancements to Features: To provide value-added capabilities, system features are continuously improved in response to user input, technology developments, and shifting security requirements.

Participation in the Community:

Establishing a User Community: Bringing together a group of users to exchange knowledge, perspectives, and pointers on properly utilizing the system for ethical hacking.

Online Forums and User Groups: Creating online discussion boards, user groups, and forums where people can communicate, pose queries, and share information about the system and ethical hacking methods.

Organizing frequent community events, such meetups, hackathons, and webinars, is a great way to promote participation, support teamwork, and highlight the accomplishments and contributions of users.

The Integrated Ethical Hacking System may increase user acceptance, enhance user competency, keep up with the most recent security trends, and promote a thriving community of ethical hacking practitioners by concentrating on these success elements.

3.9 Selected Methodology

In creating the Integrated Ethical Hacking System, the Agile Development methodology has been selected. The open-source project for this system is executed by iteration and collaboration, which is the nature of Agile. Thus, it allows for enhancement on the project whenever the tools or the users' needs change. Agile promotes the need for re-planning and quick turnarounds within scheduled timeframes. This is very relevant in ethical hacking and cybersecurity which is an evolving field. Certain core activities such as sprints and iterative feedback loops make it possible for the system to be developed in a way that the users' expectations are satisfactorily and reliably achieved.

Simplified Development Stages

Needs Assessment

- Identify the needs and challenges specific to cybersecurity professionals, students, and penetration testers.
- Analyze existing ethical hacking tools and open-source educational resources to define gaps and opportunities for improvement.

Requirements Definition

- Specify the core requirements for the system's main components: the web-based platform, Linux toolkit, and educational modules.
- Outline key features, including tool functionalities, access flexibility, and ease of use, ensuring alignment with open-source principles.

Research and Planning

- Conduct research into recent trends in cybersecurity, ethical hacking tools, and educational frameworks.
- Establish a development roadmap, covering technology choices, project phases, and iterative milestones.

Web-Based Platform Development

- Implement a responsive and accessible platform for the toolset, with intuitive navigation and open-source access.
- Integrate simple user registration, focusing on accessibility while preserving an ethical and legal framework.
- Offer a direct download of the Linux toolkit, ensuring it's freely accessible as an open-source resource.

Linux Toolkit Development

- Build a downloadable Linux toolkit featuring a streamlined interface optimized for ethical hacking tasks.
- Include a range of essential tools, such as IP lookup, DoS attack capabilities, Wi-Fi password retrieval, and more.
- Ensure cross-distribution compatibility for various Linux environments.

Educational Section Development

- Develop educational resources covering topics in cybersecurity, forensics, penetration testing, and more.
- Create interactive tutorials, hands-on labs, and structured learning paths to support both beginners and advanced users.

Security and Legal Compliance

- Implement security measures for user registration, data integrity, and secure downloads, respecting the open-source framework.
- Perform penetration testing to ensure that vulnerabilities are addressed, promoting a safe user environment in line with legal standards.

Testing and Quality Assurance

- Conduct rigorous testing on all system components, including the web platform, toolkit, and educational resources.
- Address any issues related to usability, security, or performance to ensure high standards.

Deployment and Community Support

- Deploy the system on a secure, accessible platform, encouraging community contributions and open-source collaboration.
- Offer training resources and foster user engagement, inviting feedback to guide future development.

Continuous Improvement and Updates

- Actively seek user feedback and monitor system performance, planning for regular updates to introduce new tools and improve existing features.
- Keep the educational content and toolkit up to date to address new cybersecurity challenges and trends.

Documentation and Open-Source Licensing

- Develop clear documentation for each tool, method, and feature of the Ethical Hacking System.
- License the toolkit under the MIT License, supporting open-source collaboration and user freedom while protecting creator rights

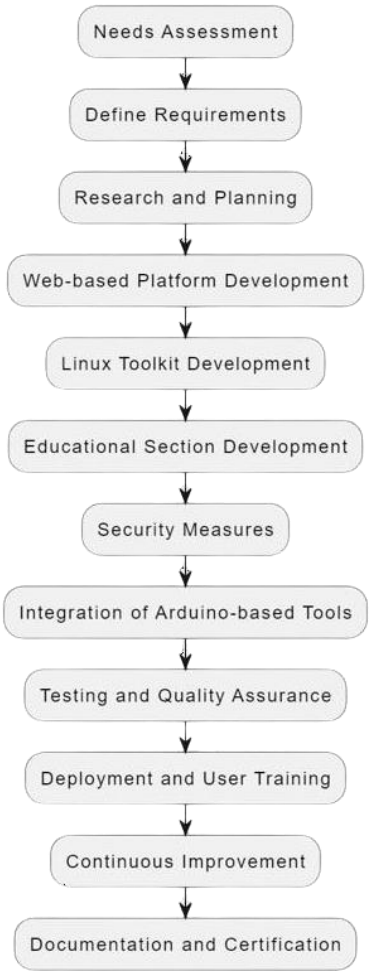

Figure 2 - Development Model

3.9.2 Questionnaire for Research on Integrated Ethical Hacking System

The questionnaire distributed to cybersecurity professionals included the following sections:

Section 1: Participant Information

- Name:
- Age:
- Gender:
- Occupation:
- Experience in Cybersecurity (in years):
- Educational Background in Cybersecurity:

Section 2: Current Practices and Challenges

1. What tools do you currently use for penetration testing and ethical hacking?
- Participants were asked to list all tools they use and describe their primary use cases. This information helps identify commonly used tools and their perceived effectiveness.

2. How would you rate the ease of use of these tools?
- (Scale of 1-5, where 1 is very difficult and 5 is very easy.) This question aimed to gauge the user-friendliness of current tools.

3. What challenges do you face with the current tools you use?
- Participants could describe issues such as lack of integration, user interface problems, or insufficient documentation.

4. Do you use multiple tools to conduct a single assessment? If yes, how many on average?
- This question explored the complexity of current workflows and the need for multiple tools.

5. Have you encountered any legal or compliance issues while performing ethical hacking?
- Participants were asked to describe any instances of legal or compliance issues they faced, highlighting the importance of legal awareness in ethical hacking.

Section 3: Preferences and Needs

6. What features do you believe are essential in an ethical hacking toolkit?
- Participants were asked to list features they consider important and explain why, helping to prioritize feature development.

7. How important is it for you to have an educational section integrated within the toolkit?
- (Scale of 1-5, where 1 is not important and 5 is very important.) This question gauged the value placed on integrated educational resources.

8. What topics or areas should be covered in the educational section?

- Participants could suggest topics such as tutorials on specific tools, best practices in ethical hacking, or legal guidelines.

9. Would you prefer a web-based platform, a downloadable toolkit, or both? Why?
- This question explored user preferences for platform accessibility.

10. How likely are you to use a new integrated ethical hacking system that includes both tools and educational resources?
- (Scale of 1-5, where 1 is very unlikely and 5 is very likely.) This helped assess the potential adoption of the proposed system.

Section 4: Feedback and Suggestions

11. What improvements would you suggest for current ethical hacking tools?
- Participants were asked for specific suggestions to improve existing tools.

12. Do you have any suggestions for the design and functionality of the proposed integrated ethical hacking system?
- This question invited feedback on the proposed system's design and functionality.

13. How do you think an integrated ethical hacking system could benefit your work or studies in cybersecurity?
- Participants could describe potential benefits, helping to align the system with user needs.

14. Would you be willing to participate in a beta testing phase for the new system?
- If yes, participants could provide their contact information, helping to identify potential beta testers.

The insights gained from this questionnaire informed the requirement analysis and helped shape the design and development of the Integrated Ethical Hacking System. The data collected provided valuable feedback on user preferences, current challenges, and essential features, ensuring the final product meets the needs of cybersecurity professionals effectively.

CHAPTER 4 – DESIGN

4.1 Introduction

This chapter describes the design of the Integrated Ethical Hacking System, detailing the system architecture, components, user interface, and design principles. The design aims to deliver a streamlined user experience for cybersecurity professionals, students, and enthusiasts by integrating a diverse toolset,

educational resources, and open-source flexibility. By aligning with Agile principles, the design approach supports iterative development, modularity, and ease of expansion.

4.2 System Architecture

The system architecture is built around two primary components: the Web-Based Platform and the Linux Toolkit. While these components operate independently, they integrate seamlessly to deliver a unified user experience. MySQL, running on XAMPP, serves as the local database to handle structured data needs, providing flexibility for those who require basic data management.

4.2.1 Web-Based Platform Architecture

Frontend: Developed using HTML5, CSS3, and JavaScript for responsive design, the frontend allows users to easily access tools, download options, and educational resources. The interface is tailored for usability, even on different devices.

Backend: Built using a server-side framework (e.g., PHP with XAMPP), the backend handles user sessions, processes requests, and manages interactions with the MySQL database.

MySQL Database: The database is implemented via XAMPP to enable local storage and management of essential data. MySQL can handle:

4.2.2 Linux Toolkit Architecture

The Linux Toolkit, a collection of command-line tools, is organized into categories to streamline user interactions in a Linux environment.

Tool Organization: Tools are structured into categories like Network Scanning, Attack Simulation, Information Gathering, and Phishing Tools.

Scripts and Commands: Scripts are built using Bash or Python, designed to be compatible with multiple Linux distributions. Examples include DoS Attack Simulation, MAC Address Spoofing, and Website Admin Page Scanning.

4.3 User Interface Design

The user interface is organized for ease of use, catering to both novice and experienced users.

4.3.1 Web-Based Platform UI

Homepage: The homepage introduces the project, emphasizing its ethical focus and open-source MIT licensing.

Tool Access Interface: Organized in a dashboard format, each tool’s page contains an overview, usage guide, and examples.

Educational Section UI: The educational section categorizes content by difficulty, allowing users to follow a structured path. Tutorials, video content, and labs provide an interactive learning experience.

4.3 User Interface Design

The user interface is organized for ease of use, catering to both novice and experienced users.

4.3.1 Web-Based Platform UI

Homepage: The homepage introduces the project, emphasizing its ethical focus and open-source MIT licensing.

Tool Access Interface: Organized in a dashboard format, each tool's page contains an overview, usage guide, and examples.

Educational Section UI: The educational section categorizes content by difficulty, allowing users to follow a structured path. Tutorials, video content, and labs provide an interactive learning experience.

Download area: The Linux toolkit is easily downloadable by anyone.

4.3.1.1 Mock Screen of the System

01). Login Screen

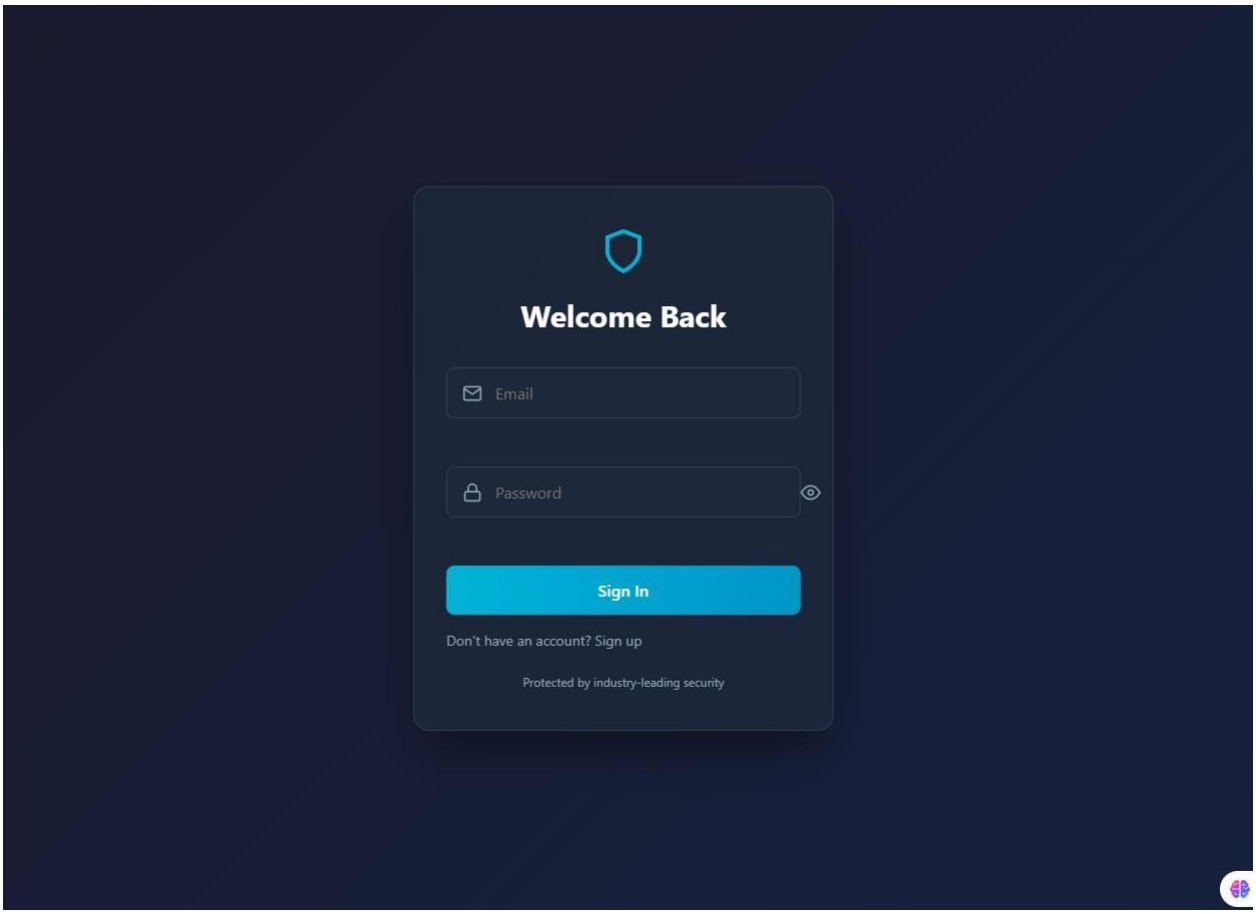

Figure 3 - Login Page

02). Register Form

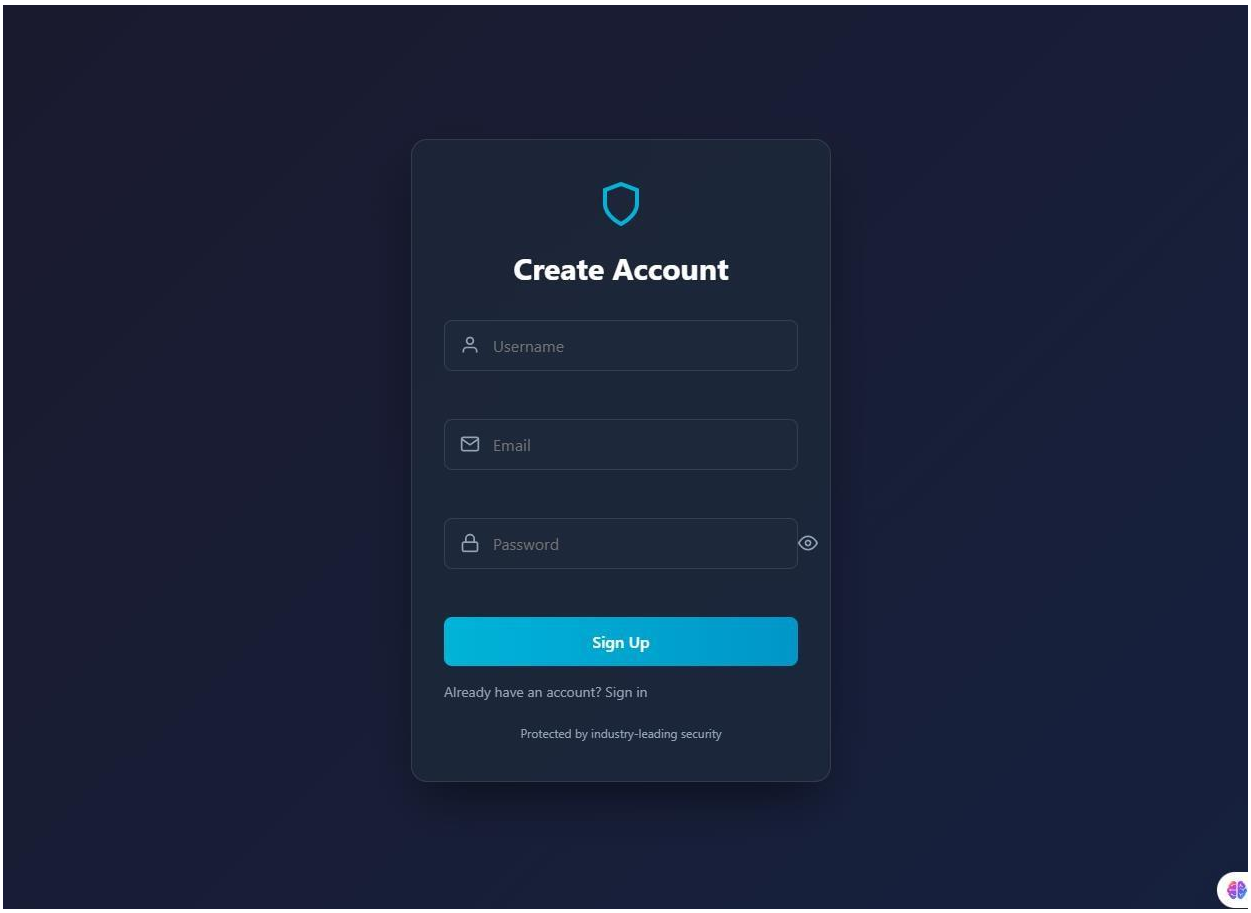

Figure 4 - Register Page

03). Home Page

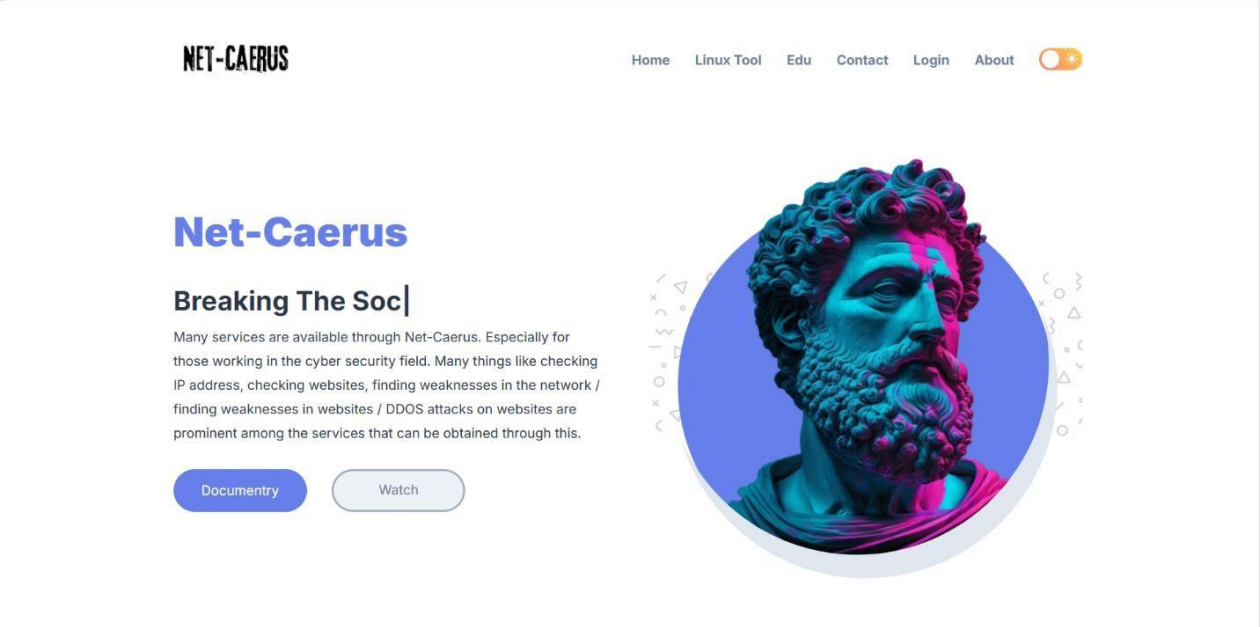

Figure 5 - Home Page

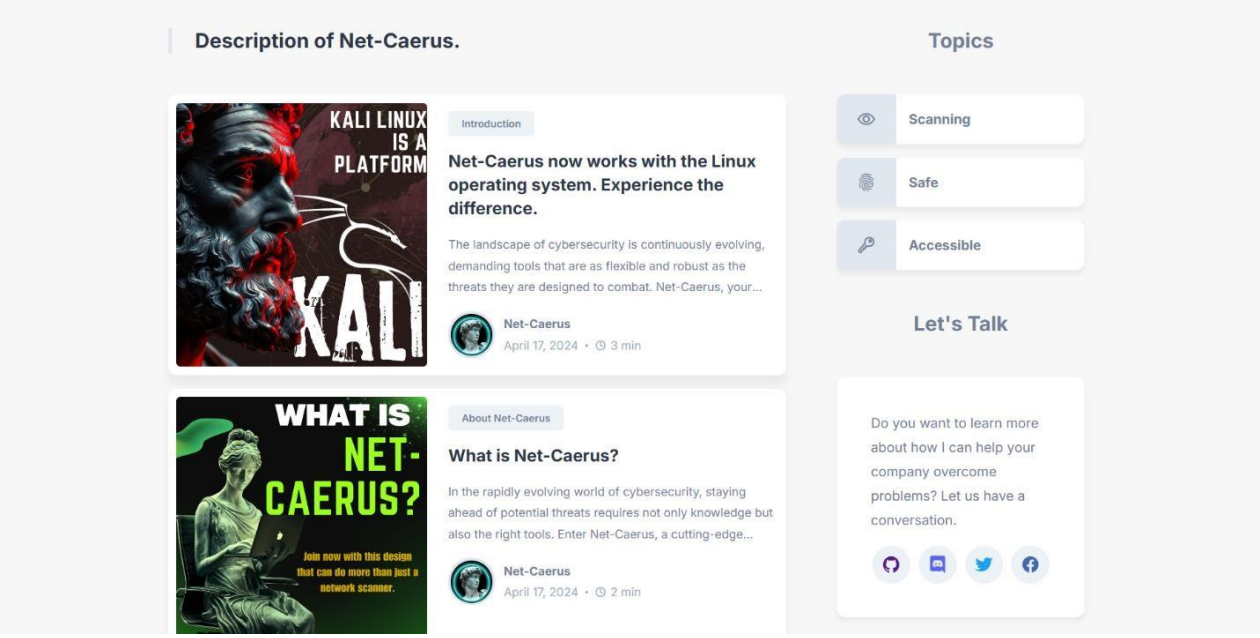

Figure 6 - Home Page Down

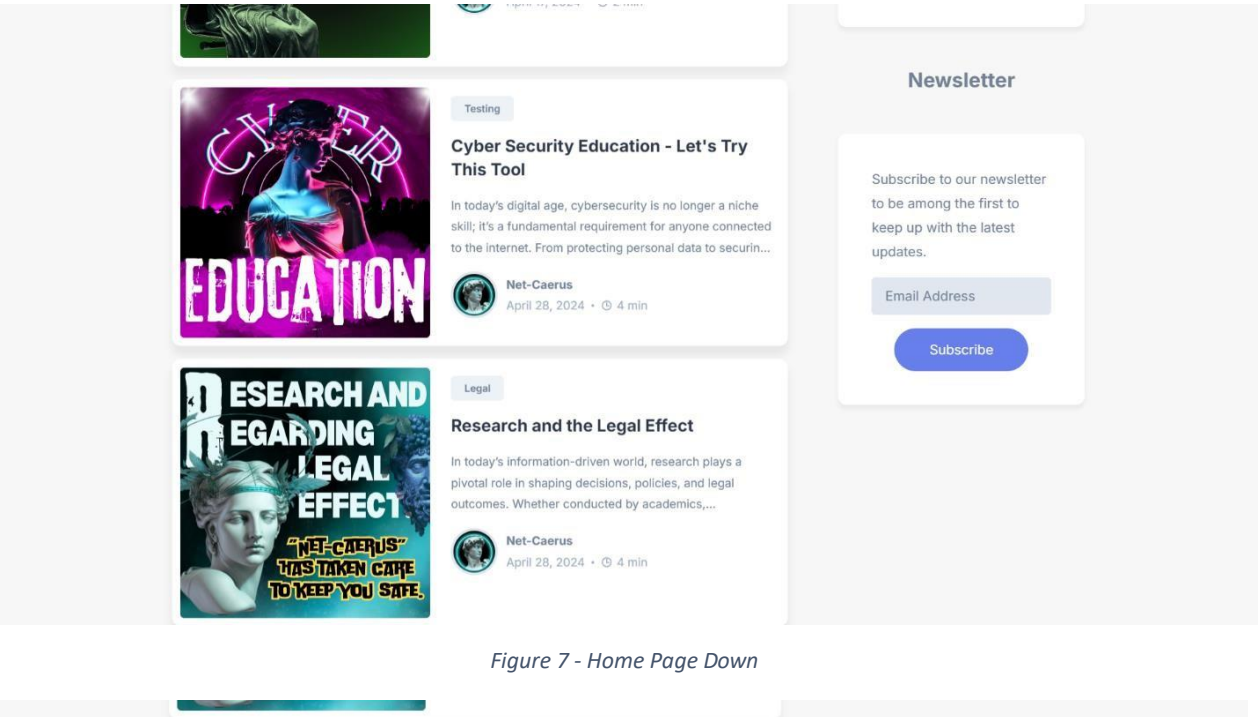

Figure 7 - Home Page Down

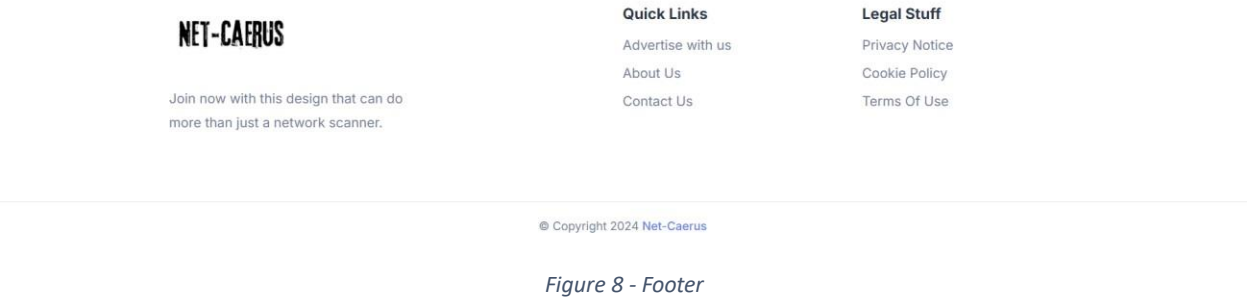

Figure 8 - Footer

Dark Mood

Figure 9 - Dark Mood

04). Edu Page

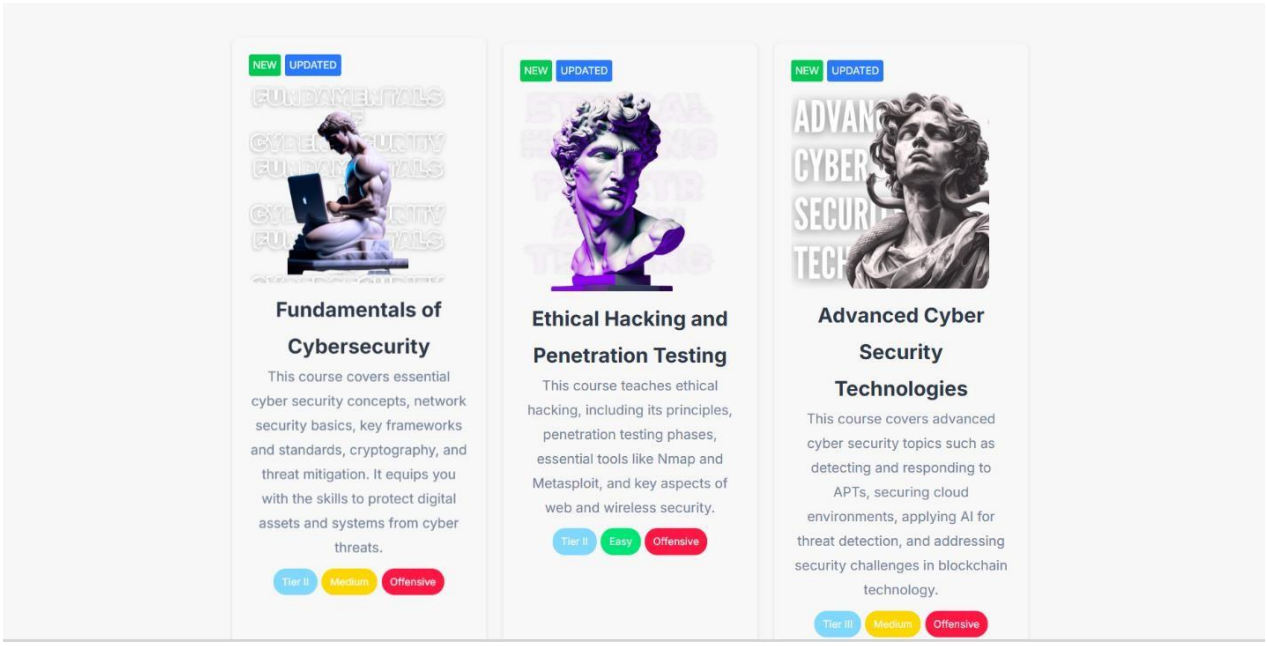

Figure 10 - Education Page

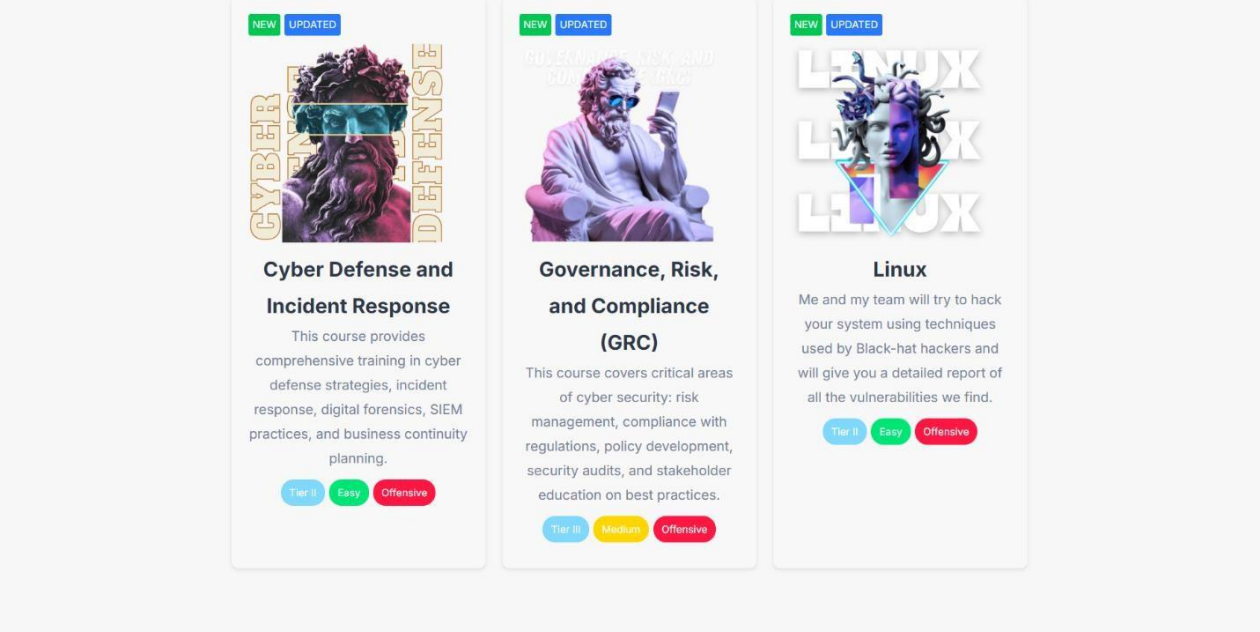

Figure 11 - Education Page

05). About Page

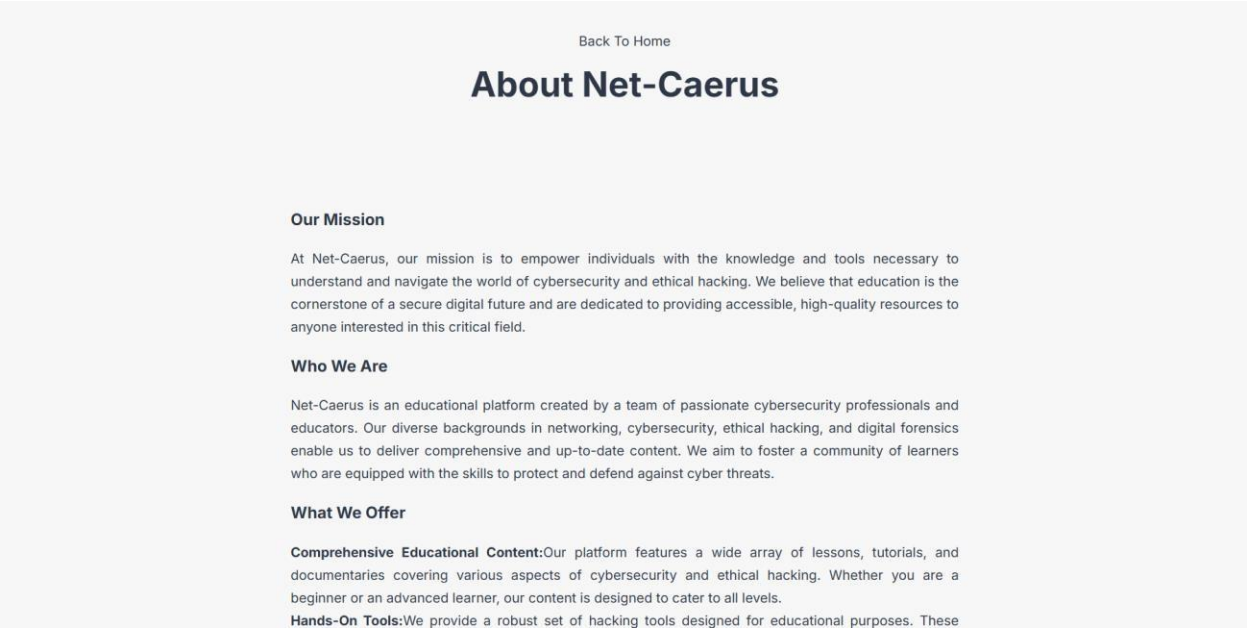

Figure 12 - About Page

06). Lesson Page

Figure 13 - Lesson

07). Lesson Video Page

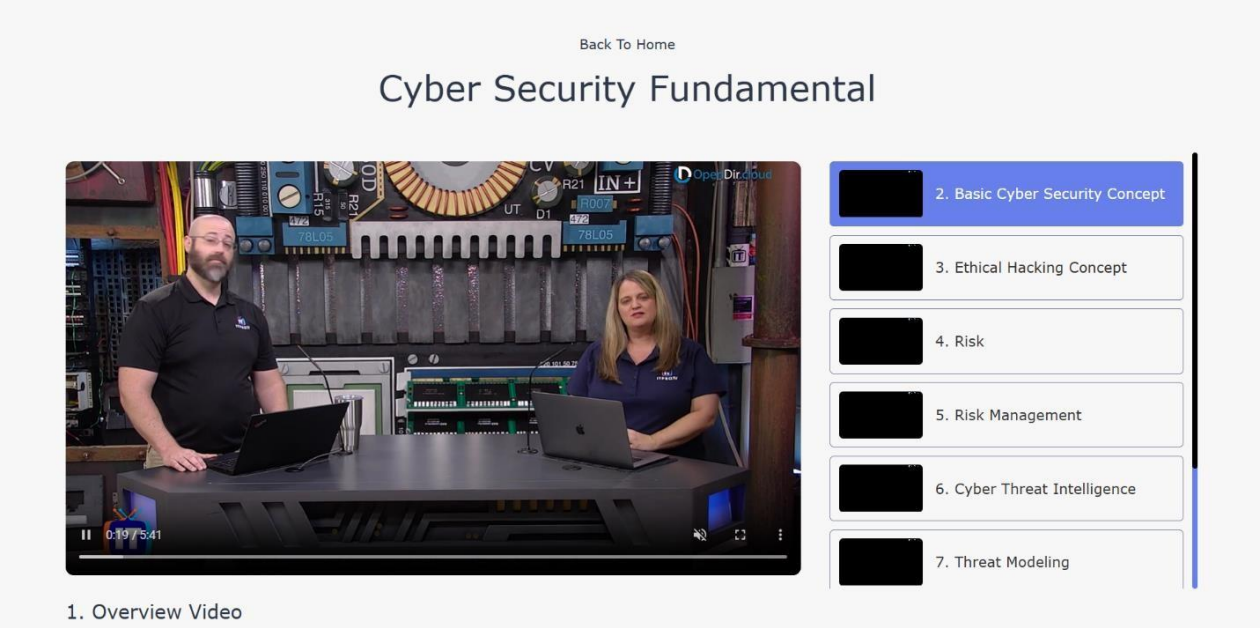

Figure 14 - Lesson Video Page

08). Lesson Pdf Downloadable page

Figure 15 - PDF downloader

09). Contact Page

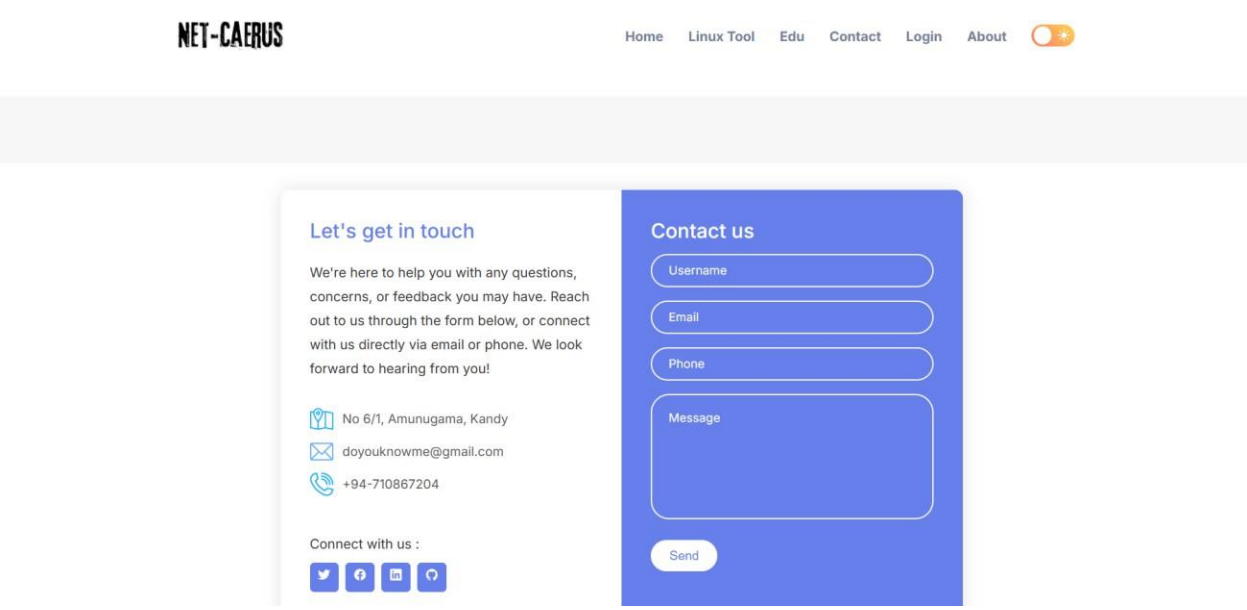

Figure 16 - Contact

4.3.2 Linux Toolkit UI

CLI Menu: A simple command-line menu categorizes tools, allowing users to navigate and launch tools efficiently.

Help options: There is a handy document for each tool.

4.3.2.1) Linux tools 1st Section

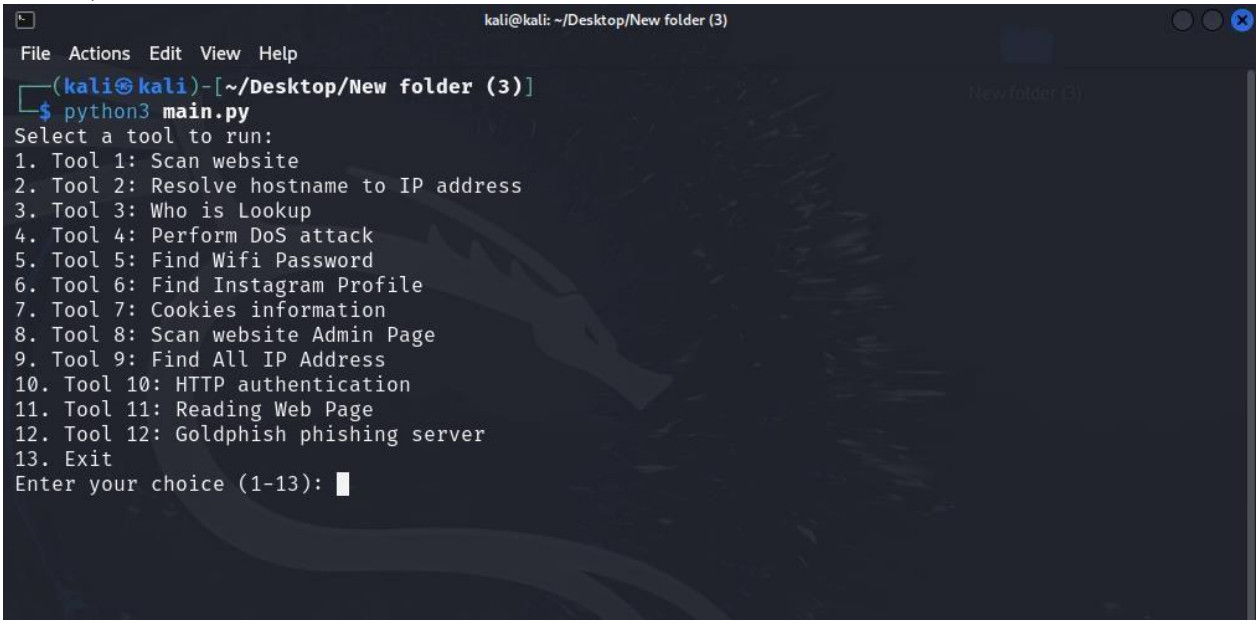

Figure 17 - Main Menu Tools

4.4 Security Design

Security is prioritized to ensure the system's open-source tools are used responsibly.

Data Encryption: All interactions between the web platform and MySQL are encrypted.

MIT Licensing: Clear terms outline user rights and responsibilities, ensuring ethical use.

4.5 Design Considerations for Open-Source Development

As an open-source project, the system is designed for flexibility, community engagement, and ease of modification:

Modular Codebase: Each tool is designed as an independent module, allowing users to customize the toolkit easily.

Community Feedback: Users can contribute feedback or modifications, supporting continuous improvement.

Version Control and Licensing: The project uses version control for tracking changes and adheres to MIT licensing, making it transparent and accessible.

4.6 System Design Process

The system design process involves creating various diagrams and models to represent the system's architecture, functionality, and interactions.

4.6.1 Use case Diagram

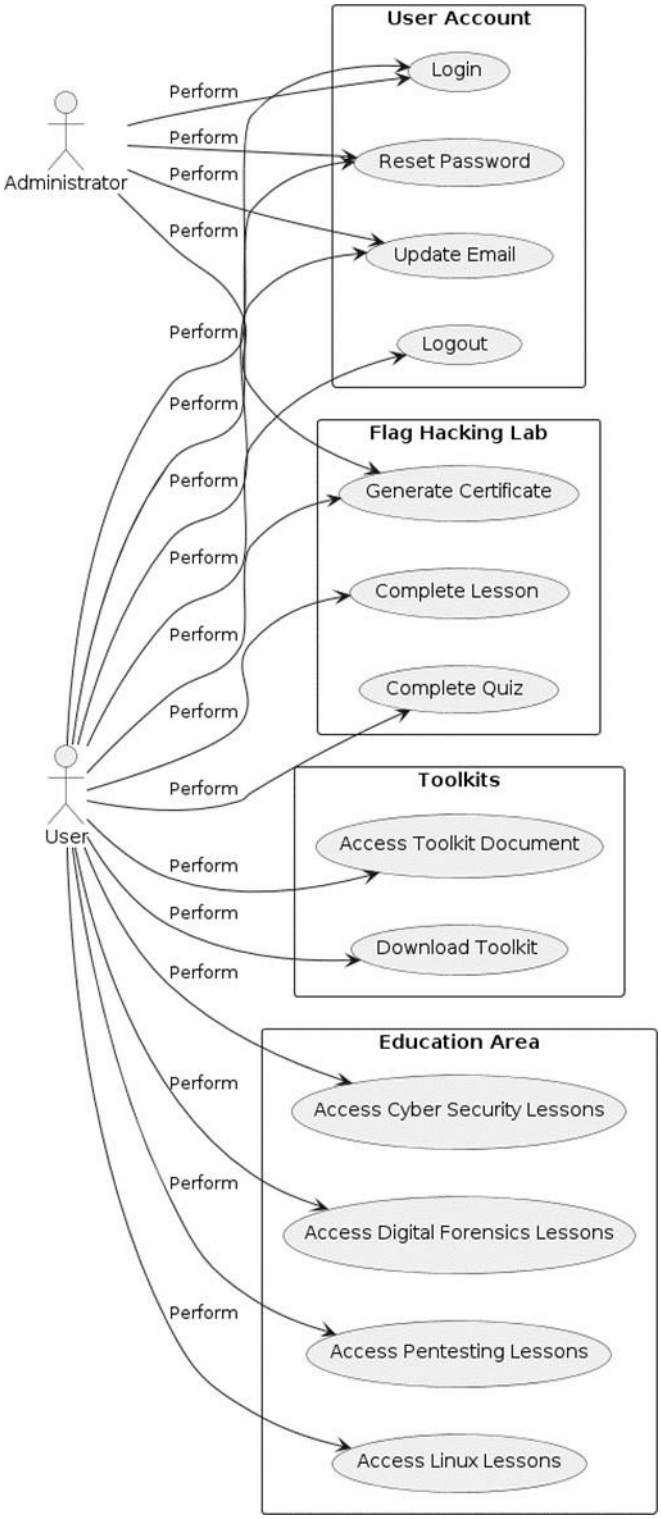

Figure 18 - Use case Diagram

4.6.2 Class Diagrams

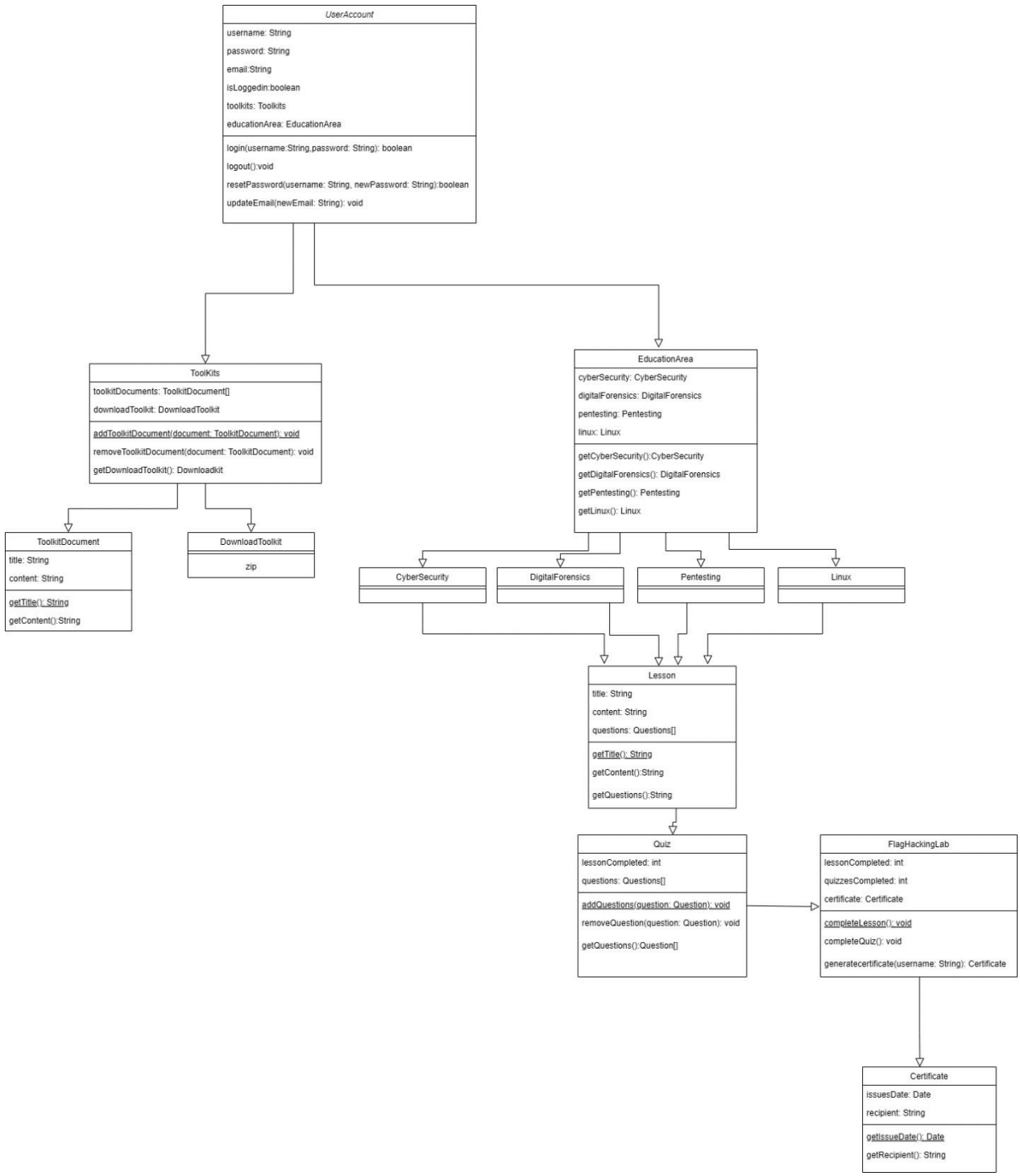

Figure 19 - Class Diagram

4.6.3 Sequence Diagrams

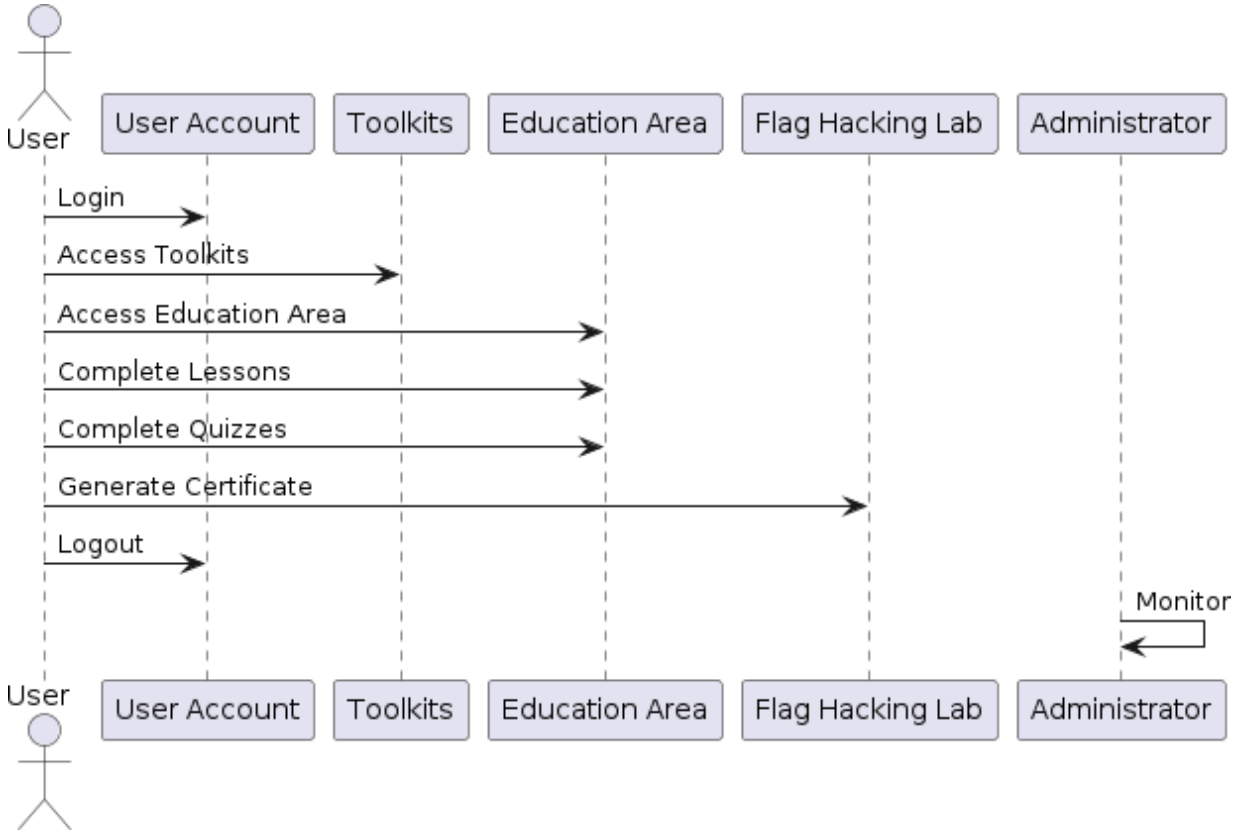

Figure 20 - Sequence Diagram

4.6.4 Activity Diagrams

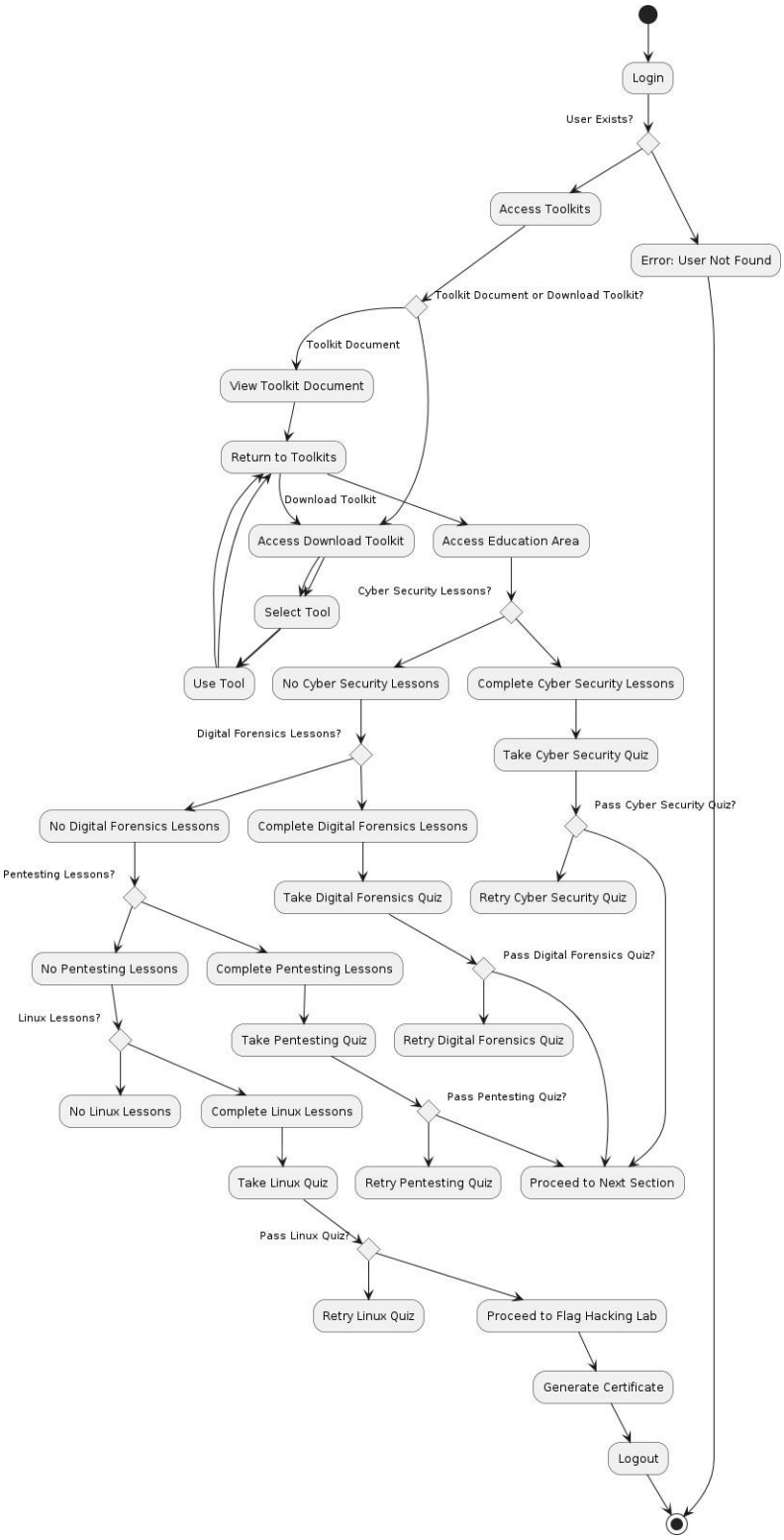

Figure 21 - Activity Diagram

4.6 Database Design

The database design supports essential functions like logging tool usage, tracking downloads, and recording educational progress if needed. While the system is primarily designed as an open-source toolset, the inclusion of XAMPP with MySQL allows for optional database functionality, enabling users to store data locally for metrics, user tracking, or performance logs.

4.6.1 Entity Relationship Diagrams (ERD)

ERDs illustrate the data entities, their attributes, and the relationships between them. These diagrams help in understanding the data structure and organization within the system.

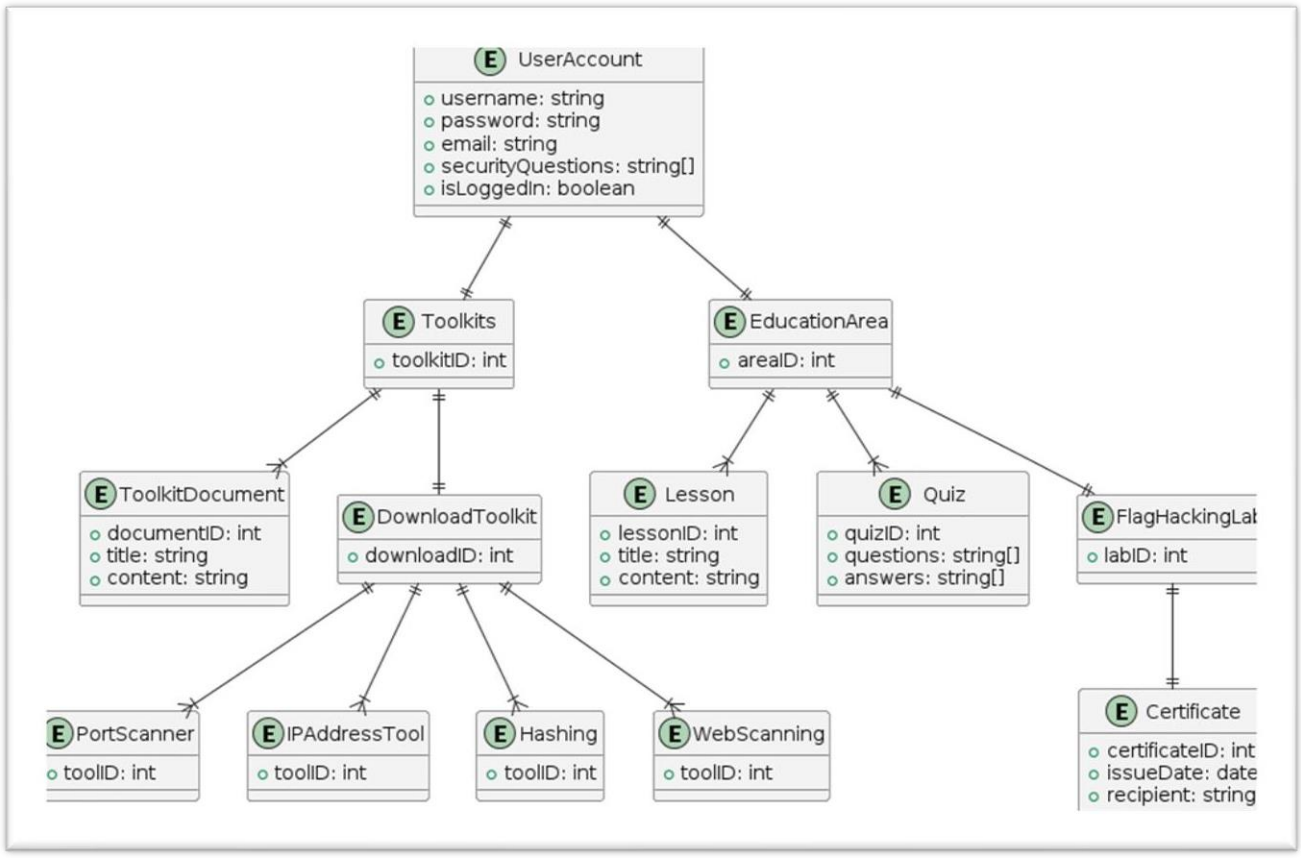

Figure 22 - ERD

CHAPTER 05 – SYSTEM DEVELOPMENT

Chapter 5 provides a detailed overview of the development process for the Integrated Ethical Hacking System. This includes the tools, technologies, and methods used to create the system, as well as specific implementation details for each component.

5.1 Overview of Development Environment

The Integrated Ethical Hacking System is developed as both a web-based platform and a Linux toolkit to accommodate diverse user needs. This chapter covers the core elements of the development environment and tools that contributed to building a seamless, effective system.

5.1.1 Software and Tools Used

Programming Languages:

- Python: Used as the primary language for scripting most of the hacking tools due to its extensive libraries in cybersecurity and networking (e.g., scapy, requests, socket).
- HTML/CSS/JavaScript: Employed for the front-end design of the web-based platform, ensuring a user-friendly interface.
- Ruby and Bash were used for payload development.
- C language was also used for Backdoor Access/Remote Access and Flag development.

Database:

- MySQL on XAMPP: Utilized for optional data storage and to log activities and user progress. It provides a relational structure to manage data effectively.

Platform:

- Linux (Ubuntu 20.04): Selected as the primary OS for developing and testing the toolkit due to Linux's compatibility with cybersecurity tools and scripts.
- Kali Linux
- Parrot OS
- Metasploitable 2.0
- Windows XP
- Windows 7
- Windows 10 *64
- Windows 10 *84

Version Control:

- Git: Employed to manage source code versions, track changes, and collaborate effectively throughout the Agile development process.

5.2 Development Methodology

The Agile methodology was chosen to develop the Ethical Hacking System due to its flexibility and adaptability, which align with cybersecurity's evolving nature. Key Agile practices included:

Sprints: Short development cycles focused on achieving small, incremental improvements.

Daily Stand-ups: Brief meetings to review progress, address challenges, and set daily goals.

5.3 Core Development Phases

The system development followed a structured phase-based approach to ensure systematic completion of key components.

5.3.1 Requirement Analysis

This phase involved gathering and analyzing the requirements for both the toolkit and web platform, based on user needs. The requirements analysis emphasized:

Tool Functionality: Understanding what tools are essential for ethical hacking and penetration testing.

User Experience: Ensuring that both beginners and advanced users could easily navigate and utilize the tools.

Security: Designing robust security features to protect data and limit access as needed.

5.3.2 Implementation

During this phase, each module and feature of the Ethical Hacking System was developed according to the specified requirements.

Tool Development

- Python scripts were developed to perform essential ethical hacking functions, including IP scanning, DoS simulation, MAC address manipulation, and more.
- Testing was conducted on various Linux distributions to ensure compatibility.

Web Platform Development

- Flask was used to build the backend, enabling smooth communication between the server and user interface.
- The front end was built with HTML, CSS, and JavaScript (including Bootstrap) for a responsive and visually appealing design.

Database Integration

- MySQL tables were created within XAMPP for optional logging and user activity tracking.
- Secure access controls were implemented to ensure that only authorized users could view or modify sensitive data.

5.3.3 Testing and Quality Assurance

Rigorous testing was performed to ensure each component worked as intended:

- Functional Testing: Each tool and web feature was tested to confirm it met the specified requirements.
- Compatibility Testing: The toolkit was tested on different Linux distributions to ensure broad compatibility.
- Performance Testing: The platform was tested for efficiency, particularly under high usage scenarios.
- Security Testing: Penetration tests were conducted on the system itself to identify and rectify any vulnerabilities.

5.4 Challenges Faced and Solutions

I faced several challenges and solved them as follows:

Challenge: Integrating multiple tools into a cohesive Linux toolkit.

Solution: Modular development ensured each tool was implemented as a standalone module, allowing flexible integration.

Challenge: Ensuring compatibility across Linux distributions.

Solution: Testing and refining the code for cross-distribution compatibility, with adjustments for package dependencies.

5.5 Code Samples and Key Snippets

Here are some simplified code samples from the toolkit, illustrating core functionalities:

Main.py

```
# main.py

from tools import findIP_main, dos_main, scan_port_main, who_main,
display_wifi_passwords, get_instagram_profile_info, get_cookies_from_url,
scan_website, get_ip_addresses, fetch_authenticated_url, fetch_web_page,
run_goldphish
from tools.admin import website

def display_menu():
```

```
    print("Select a tool to run:")
    print("1. Tool 1: Scan website")
    print("2. Tool 2: Resolve hostname to IP address")
    print("3. Tool 3: Who is Lookup")
    print("4. Tool 4: Perform DoS attack")
    print("5. Tool 5: Find Wifi Password")
    print("6. Tool 6: Find Instagram Profile")
    print("7. Tool 7: Cookies information")
    print("8. Tool 8: Scan website Admin Page")
    print("9. Tool 9: Find All IP Address")
    print("10. Tool 10: HTTP authentication")
    print("11. Tool 11: Reading Web Page")
    print("12. Tool 12: Goldphish phishing server")
    print("13. Exit")

def get_user_choice():
    return input("Enter your choice (1-13): ")

def main():
    while True:
        display_menu()
        choice = get_user_choice()

        if choice == '1':
            scan_port_main()
        elif choice == '2':
            findIP_main()
        elif choice == '3':
            who_main()
        elif choice == '4':
            dos_main()
        elif choice == '5':
            display_wifi_passwords()
        elif choice == '6':
            get_instagram_profile_info()
        elif choice == '7':
            get_cookies_from_url()
        elif choice == '8':
            scan_website(website)
        elif choice == '9':
            hostname = input("Enter the Hostname: ")
            get_ip_addresses(hostname)
        elif choice == '10':
            url = input("Enter the URL: ")
            username = input("Enter your username: ")
            password = input("Enter your password: ")
            fetch_authenticated_url(url, username, password)
        elif choice == '11':
            url = input("Enter the URL of the web page: ")
            fetch_web_page(url)
        elif choice == '12':
            run_goldphish()
        elif choice == '13':
            break
        else:
            print("Invalid choice. Please try again.")
```

```
        back_to_menu = input("Would you like to go back to the menu?
(yes/no): ")
        if back_to_menu.lower() != 'yes':
            break

if __name__== "__main__":
    main()
```

CHAPTER 06 – TESTING AND EVALUATION

6.1 Introduction

This chapter outlines the testing and evaluation process for the Integrated Ethical Hacking System. Thorough testing and evaluation were essential to ensure the system met functional requirements, performed reliably, and provided a secure and intuitive user experience. This chapter discusses the methods used to verify the system's effectiveness, details on testing strategies, and feedback incorporated from real-world users.

6.2 Testing Procedure

The testing procedure for the Integrated Ethical Hacking System was designed to ensure comprehensive assessment across various aspects of the system, from functionality to security. This involved establishing testing protocols, defining testing techniques, and setting specific test cases to address each feature of the system.

6.2.1 Techniques of Testing

- Various testing techniques were employed to validate the system's integrity, security, performance, and usability:
- Manual Testing: For functional testing, each tool was manually tested to ensure expected output and system behavior.
- Automated Testing: Scripts were used to automate repetitive testing tasks, such as input validation and error handling for various tools.
- Penetration Testing: Performed on the web-based platform to detect vulnerabilities, identify security flaws, and verify secure download of the Linux toolkit.

6.2.2 Types of Testing

Several types of testing were employed to verify the robustness, functionality, and security of the system:

- Unit Testing: Each individual tool (e.g., IP scanner, port scanner, DoS tool) was tested independently to confirm functionality.
- Integration Testing: Tools were tested together to ensure they interact correctly within the integrated toolkit.
- System Testing: The entire toolkit, including the web platform and Linux-based tools, was tested as a whole to evaluate performance in a real-world scenario.
- User Acceptance Testing (UAT): End-users (e.g., cybersecurity students and professionals) tested the system to ensure it met user expectations and was easy to navigate.
- Regression Testing: After each update or bug fix, previously tested tools and features were retested to ensure no new issues were introduced.
- Compatibility Testing: The system was tested across multiple Linux distributions and major web browsers to confirm broad compatibility.

6.3 Test Plan and Test Cases

The test plan for the integrated ethical hacker system was structured to cover all essential functions and ensure that each component met its intended purpose. Test cases were created for each tool and feature, specifying the inputs, expected outputs, and success criteria.

6.3.1 Test Cases

The following are some of the key test cases created to evaluate specific tools and functions of the system:

Let's first look at the Login Module test and focus on the tool.

Table 2 Test cases

Test Case ID		1		
Tested Component		Login Activity for Guest user		
Module Name		User Module		
Tested Area		Login Functionality		
Testing the login module by entering the Email and the password.				
No	High Level test steps	Expected Result	Actual Result	Status

1	User enter invalid user name and/or password	Display error message	pending	pending
2	User enter empty username	Display error message	pending	pending
3	User enter empty password	Display error message	pending	pending

Test Case 2- Registration Module

Table 3 Test cases 2

Test Case ID		1		
Tested Component		Registration Activity for Guest user		
Module Name		User Module		
Tested Area		Registration Functionality		
Testing the Registration module.				
No	High Level test steps	Expected Result	Actual Result	Status
1	Create new account by adding required fields and hit on Register button	Show successfully registered message	pending	Pass
2	User enter empty fields	Display error message	pending	Pass
3	User enter invalid input to the fields	Display error message	pending	pass

Test Case 1: IP Address Scanner

Objective: Verify that the tool successfully scans and returns the correct IP address.

Input: Domain name (e.g., example.com)

Expected Output: IP address of the specified domain.

Success Criteria: The IP address matches the expected IP address of the domain.

Test Case 2: Port Scanner

Objective: Ensure the port scanner correctly identifies open and closed ports on a specified IP address.

Input: IP address (e.g., 192.168.1.1)

Expected Output: List of open ports on the target IP address.

Success Criteria: The output accurately reflects the open ports on the specified IP.

A screenshot is below.

6.3 Test Data and Test Results

Test Data was carefully curated to simulate realistic inputs and scenarios for the Ethical Hacking System. Test data includes URLs, IP addresses, port ranges, and sample credentials where necessary, allowing a thorough assessment of each tool's capabilities.

Sample Test Data:

URL for scanning: example.com, testsite.org

IP Addresses: 192.168.1.1, 10.0.0.1

MAC Addresses: 00:14:22:01:23:45, 00:25:96:FF:FE:12

User Inputs: Example login credentials, sample keystrokes for keylogger, arbitrary port numbers (e.g., 80, 443, 8080)

Test Results:

Table 4 Test result

Test Case ID	Expected Result	Actual Result	Pass/Fail
TC-01	Lists all open ports	All open ports listed accurately	Pass
TC-02	Initiates DoS traffic correctly	DoS attempt simulated as expected	Pass
TC-03	Changes MAC address successfully	MAC address changed without issues	Pass
TC-04	Phishing server accessible on designated IP	Server operational and page rendered	Pass
TC-05	Admin page scanner locates hidden pages	Common admin pages identified	Pass

6.4 Acceptance Testing

Acceptance testing was conducted to validate the system for public release as an open source toolkit, and to confirm that it met user expectations and met all stated requirements.

Acceptance Testing Process:

User Acceptance Criteria: Defined based on functional and non-functional requirements to ensure that the ethical hacking system aligns with the intended objectives.

Test Cases: Test cases were developed by cybersecurity students, professionals, and enthusiasts to simulate real-world usage.

Stakeholder Review: Feedback from early adopters, including university students and cybersecurity educators, was collected to identify any usability or functionality issues.

Final Validation: After resolving the identified issues, the system was retested to confirm that user requirements were met.

CHAPTER 07 – CONCLUSION, FUTUREWORKS AND CHALLENGES.

7.1 Conclusion

The development of the Integrated Ethical Hacking System represents an important step in making cybersecurity tools more accessible, efficient, and user-friendly for cybersecurity professionals,

students, and enthusiasts. The system integrates multiple tools into a unified platform, including a suite of web-based and Linux-based tools to facilitate a variety of penetration testing and ethical hacking activities. The open-source nature of this project ensures unlimited access to users, promotes community collaboration, and ensures continued development in the field. Each tool in the toolkit, from IP scanning to phishing simulation, is designed to meet real-world testing needs, supporting hands-on learning and skill development.

This project successfully meets its objectives of delivering an open-source ethical hacking toolkit that is flexible, comprehensive, and tailored to the needs of a diverse audience. The inclusion of educational resources further enhances its value, offering a structured way for users to improve their knowledge of cybersecurity principles and techniques.

7.2 Challenges

Throughout the project, several challenges were encountered, and there were also areas where future development could present difficulties:

Security and ethical concerns: One of the main challenges is ensuring that the toolkit is used responsibly and ethically. As these tools can be used for both secure and malicious purposes, maintaining a clear ethical framework and compliance with legal standards is essential. Regular monitoring and community engagement are required to promote responsible use.

Ongoing maintenance and updates: Cyber threats are constantly evolving, requiring frequent updates to the toolkit. The open-source nature of the project helps address this by allowing contributions from the cybersecurity community, but it still requires consistent monitoring and development to remain effective.

Cross-platform compatibility: Ensuring that all tools work seamlessly across different Linux distributions and maintaining a stable web-based platform presents an ongoing technical challenge. The toolkit needs to be tested and updated for compatibility with new system versions and dependencies to avoid potential issues.

User data privacy: Although the system does not rely heavily on a database due to its open source architecture, any expansion that includes user data (e.g. in personalized educational modules) will require strict data privacy measures, especially if integrated into web-based features.

Resource management for educational content: Maintaining high-quality educational resources that stay up to date with industry trends is resource-intensive. Creating and updating tutorials, labs, and quizzes in a rapidly changing field like cybersecurity requires a dedicated team or community contribution.

Legal Restrictions: In certain regions, the distribution or use of ethical hacking tools may be subject to legal restrictions, which may limit accessibility and require compliance efforts for wider distribution.

7.2 Future Works

the Integrated Ethical Hacking System can expand its toolset and educational resources to include capabilities for drone security and hacking. With drones becoming increasingly prevalent across industries like logistics, agriculture, surveillance, and more, the potential for cybersecurity vulnerabilities within drone systems has grown substantially. Integrating drone-related tools and training would position the toolkit to meet emerging security needs in this area.

Drone Security and Hacking Module: Adding a specialized module for drone security would enable users to explore and learn about the cybersecurity aspects unique to drones, including communication protocols, GPS spoofing, signal jamming, and unauthorized access vulnerabilities.

Enhanced Toolset: Expanding the toolkit to include additional tools like network traffic analyzers, protocol-specific scanners, and more sophisticated vulnerability assessment tools would add depth to the system.

Artificial Intelligence for Threat Detection: AI and machine learning techniques could be integrated into the toolkit for smarter analysis and detection capabilities. For instance, an AI-driven tool that analyzes network patterns to identify abnormal behavior could be a valuable addition.

Multi-language Support: Making the platform available in multiple languages would enhance its accessibility to a global audience, helping people from different regions and backgrounds engage with the platform more effectively.

References

[1] A. Chowdhary, "Autonomous Security Analysis and Pentesting (ASAP)," *ResearchGate*, Aug. 2020. https://www.researchgate.net/publication/343537449_Autonomous_Security_Analysis_and_Pentesting_ASAP (accessed Feb. 01, 2024).

[2] Y. Dong, Z. Li, Y. Tian, C. Sun, M. W. Godfrey, and M. Nagappan, "Bash in the Wild: Language Usage, Code Smells, and Bugs," *ResearchGate*, Apr. 2022. https://www.researchgate.net/publication/366820748_Bash_in_the_Wild_Language_Usage_Code_Smells_and_Bugs (accessed Feb. 01, 2024).

[3] M. Salim, "Analisis Ancaman Phishing Backdoor Remote Access Trojan (BRAT)," *ResearchGate*, Jul. 2023.https://www.researchgate.net/publication/373392308_Analisis_Ancaman_Phishing_Backdoor_Remote_Access_Trojan_BRAT (accessed Feb. 01, 2024).

[4] B. A Patel, "(PDF) Role of Ethical Hacking in System," *ResearchGate*, Jan. 2016. https://www.researchgate.net/publication/325102785_Role_of_Ethical_Hacking_in_System (accessed Feb. 01, 2024).

[5] P. Cisar and R. Pinter, "Journal of Applied Technical and Educational Sciences jATES Some ethical hacking possibilities in Kali Linux environment," *Journal of Applied Technical and Educational Sciences jATES* , vol. 9, no. 4, pp. 129–149, 2019, doi: https://doi.org/10.24368/jates.v9i4.139.

[6] M. N. M. Najath, D. Herath, and A. Rajapakse, "Design and Testing of an Arduino-based Network JammerDevice,"*ResearchGate*,Dec.2022.https://www.researchgate.net/publication/366611575_Design_and_Testing_of_an_Arduino-based_Network_Jammer_Device (accessed Feb. 03, 2024).

[7] N. I. Daud, K. A. Abu Bakar, and M. S. Md Hasan, "A case study on web application vulnerability scanning tools," *2014 Science and Information Conference*, Aug. 2014, doi: https://doi.org/10.1109/sai.2014.6918247.

[8] trustedsec,"trustedsec/social-engineer-toolkit," *GitHub*, Aug. 30, 2020. https://github.com/trustedsec/social-engineer-toolkit (accessed Feb. 2024).

[9] Kali, "Our Most Advanced Penetration Testing Distribution, Ever.," *Kali.org*, Dec. 04, 2018. https://www.kali.org/

[10] M. Z. Trujillo, L. Hébert-Dufresne, and J. Bagrow, "The penumbra of open source: projects outside of centralized platforms are longer maintained, more academic and more collaborative," *EPJ Data Science*, vol. 11, no. 1, May 2022, doi: https://doi.org/10.1140/epjds/s13688-022-00345-7.

[11] skavngr, "skavngr/rapidscan," *GitHub*, Jun. 01, 2021. https://github.com/skavngr/rapidscan

[12] A. Hennig, L. Schulte, S. Herbold, O. Kulyk, and P. Mayer, "Understanding issues related to personal data and data protection in open source projects on GitHub," *ar5iv*, 2023. https://ar5iv.labs.arxiv.org/html/2304.06367v1 (accessed Nov. 03, 2024).

[13] S. Spiekermann and L. F. Cranor, "Engineering Privacy," *IEEE Transactions on Software Engineering*, vol. 35, no. 1, pp. 67–82, Jan. 2009, doi: https://doi.org/10.1109/TSE.2008.88.

[14] "Insecam - World biggest online cameras directory," *Insecam.org*, 2022. http://www.insecam.org/

APPENDIX A - SYSTEM DOCUMENTATION

Introduction

The Net-Caerus Tool Set represents a cutting-edge, integrated suite of cybersecurity tools designed to support ethical hacking, penetration testing, and digital forensics. Developed as a unified system, Net-Caerus provides a centralized toolkit for cybersecurity professionals, researchers, and students, helping them efficiently manage, test, and secure digital infrastructures. This documentation offers an extensive guide to understanding each feature within the toolkit, including how to install, configure, and use it effectively across various scenarios. The goal of Net-Caerus is to streamline ethical hacking workflows by consolidating essential tools into one system, which is deployable as both a web-based platform and a downloadable Linux toolkit.

This toolkit includes a wide range of utilities—from information gathering and network analysis tools to more advanced functions, like payload generation and remote access capabilities. Users can achieve thorough security assessments, perform controlled simulations, and analyze potential vulnerabilities systematically, all from within a single, cohesive interface. Additionally, each tool within Net-Caerus is designed to prioritize user-friendliness while ensuring adherence to ethical and legal standards in cybersecurity.

Purpose

The purpose of Net-Caerus is to serve as a comprehensive, all-in-one ethical hacking and cybersecurity assessment toolkit, designed to streamline the process of identifying, analyzing, and mitigating security vulnerabilities across systems, networks, and applications. It addresses the fragmentation in existing cybersecurity tools by providing a unified platform that brings together essential functionalities in one place, improving efficiency and usability for cybersecurity professionals and ethical hackers.

Net-Caerus enables users to simulate cyber-attacks, assess system defenses, and conduct penetration testing using a variety of tools tailored to different aspects of vulnerability assessment—such as network scanning, DoS testing, phishing simulations, and payload creation. This integration saves users the time and complexity of switching between multiple tools and platforms, making it easier to conduct thorough, cohesive security audits.

Additionally, Net-Caerus includes minimal but practical educational resources, offering users a foundational understanding of responsible tool usage and ethical hacking best practices. While the toolkit emphasizes practical functionality, it is designed to foster ethical and legal use, promoting responsible cybersecurity practices. By equipping cybersecurity professionals with a centralized, user-friendly toolkit, Net-Caerus contributes to the enhancement of cybersecurity defenses in an era of rising digital threats.

Document Conventions for Net-Caerus System Documentation

To ensure clarity, consistency, and ease of understanding, this document follows a set of conventions regarding terminology, formatting, and symbol usage. Below are the conventions applied throughout this system documentation for Net-Caerus:

Terminology and Nomenclature

Net-Caerus: Refers specifically to the ethical hacking toolkit that consolidates various cybersecurity tools into a single platform.

User: Represents cybersecurity professionals, ethical hackers, and penetration testers utilizing Net-Caerus for security assessments.

System: May refer either to the Net-Caerus application or the target system under assessment, as context dictates.

Tool/Feature/Module: Specific functions or components within the Net-Caerus toolkit.

Text Formatting

Bold Text: Used for emphasizing key terms, tool names, or section titles within descriptions (e.g., **Payload Generator, Port Scanner**).

Italics: Used for file names, path directories, or technical terms when first introduced (e.g., *hostname resolution, DoS attack*).

Code Blocks: Displayed in a monospaced font to denote commands, code snippets, or file paths (e.g., ping <IP address>, /etc/net-caerus/tools/).

Numbered and Bulleted Lists

Numbered Lists: Used to present steps in a sequential procedure or multi-step process for using specific tools.

Bulleted Lists: Used to list non-sequential features, options, or requirements.

Cross-Referencing and Hyperlinks

Cross-references within the document direct users to other sections for further information, typically denoted in parentheses (e.g., see 3.2 Tool Setup).

Hyperlinks (if used within digital versions) will appear in blue and underlined, allowing for easy navigation between sections.

Command Syntax

Command-line instructions follow syntax conventions to clearly indicate where user input is required:

<parameter>: Placeholder for user-provided values, such as IP addresses or file names.

[optional]: Optional arguments or parameters for commands or scripts.

Project Scope of Net-Caerus

The Net-Caerus project aims to deliver an integrated, comprehensive toolkit specifically designed to streamline the ethical hacking and penetration testing process. It addresses the critical need for an accessible, all-encompassing suite of tools that cater to the demands of cybersecurity professionals, penetration testers, and ethical hackers. The scope of the project is defined as follows:

Toolset Development and Integration

Consolidated Toolkit: Net-Caerus will unify a diverse range of cybersecurity tools within a single platform, removing the need for professionals to switch between multiple applications.

- Core Functionalities: The toolkit will cover essential areas of ethical hacking, including:
- Network scanning and enumeration
- Vulnerability assessment and exploitation
- Social engineering simulation and phishing toolkits
- Denial-of-Service (DoS) attack simulations
- MAC address manipulation

Innovative Tools: In addition to standard functions, Net-Caerus will introduce unique tools, such as a payload generator and an Arduino-powered Wi-Fi jammer. This innovation aims to expand the toolkit's usability beyond conventional platforms.

User-Friendly Interface: Tools will be accessible through a streamlined, intuitive interface designed for efficient navigation, minimizing setup time and maximizing ease of use.

Linux Toolkit Availability

Downloadable Linux Version: Recognizing that many ethical hackers work within Linux environments, Net-Caerus will be available as a downloadable toolkit for Linux, optimized for seamless functionality across various distributions.

Command-Line Interface (CLI): The toolkit will include a robust CLI option, allowing users to operate tools through command-line commands, increasing efficiency and flexibility in different usage scenarios.

Web-Based Platform (Limited Scope)

Accessible Resources: While the primary focus is on the Linux toolkit, the project will also include a minimal web-based interface for basic tools. This online version provides users with quick access to essential functions, such as IP scanning and basic reconnaissance tools, from any device with internet access.

Educational Resources (Minimal Scope): Limited educational content will be incorporated to aid users in understanding the practical application of each tool within the toolkit. Unlike typical online platforms, this resource will focus more on tool usage guidance than on general cybersecurity education.

Specific Tools within the Scope

Scanning and Enumeration: Net-Caerus will include tools for scanning and gathering information on networks and web applications, such as website scanners, hostname resolution, and IP address lookup.

Social Engineering Tools: Features like Goldphish and CamPhishing will enable users to simulate phishing and social engineering attacks responsibly.

Payload Creation and Exploitation Tools: Includes features like payload generation for remote access, keyloggers, and remote connection utilities.

Hardware Integration: Specialized tools like the Arduino-powered Wi-Fi jammer will be incorporated to address physical security aspects.

Security and Compliance

Ethical Use and Access Controls: Since Net-Caerus is an ethical hacking toolkit, it will include usage guidelines to encourage responsible and lawful testing. Certain features may be restricted or require explicit user acknowledgment of ethical guidelines to prevent misuse.

Regular Updates and Maintenance: The toolkit will be designed for future updates, with mechanisms for incorporating new features and security patches as cyber threats evolve.

Project Limitations

Legal and Ethical Boundaries: Tools that could be potentially misused, like remote access and DoS functionalities, will include warnings and disclaimers to emphasize ethical usage.

Platform Constraints: While the toolkit will support most Linux distributions, compatibility with other operating systems, such as Windows and macOS, is outside the current project scope.

Educational Component: Minimal educational resources will be included, mainly focused on tool instructions. Comprehensive training or certification is outside the scope of this project.

Expected Deliverables

Net-Caerus Linux Toolkit: A complete, downloadable toolkit tailored for Linux environments, incorporating all core functionalities and tools.

Web-Based Platform: An accessible online platform with basic tools and limited educational resources for quick, convenient access.

Documentation: Detailed system documentation for tool use, setup, and ethical guidelines.

Security and Usage Guidelines: Comprehensive guidelines to ensure ethical, responsible use of Net-Caerus.

Net-Caerus Full Description

Net-Caerus is a unified ethical hacking system designed to equip cybersecurity professionals with a comprehensive toolkit that addresses the growing complexity and sophistication of cyber threats. As organizations increasingly rely on digital infrastructure, the need for effective ethical hacking tools has never been more critical. This project addresses the fragmented nature of existing cybersecurity resources by offering a unified platform that simplifies the penetration testing process, increases productivity, and promotes responsible hacking practices.

Key Features and Components

Integrated Toolset:
Net-Caerus consolidates a wide array of ethical hacking tools into a single, user-friendly platform. This toolset includes functionalities for scanning networks, performing vulnerability assessments, social engineering simulations, and executing denial-of-service (DoS) attacks.

Tools are designed to operate seamlessly, allowing users to perform comprehensive security assessments without the need to switch between disparate applications.

Dual Delivery Method:
The system will be available both as a downloadable Linux toolkit and a web-based platform. The Linux toolkit is optimized for various distributions, ensuring compatibility and functionality for users who primarily operate in Linux environments.

The web-based version will provide quick access to essential tools and functions, enabling users to perform tasks on the go without the need for installation.

User-Centric Design:
The interface of Net-Caerus is designed with user experience in mind, ensuring ease of navigation and operation. This focus on user-friendliness aims to reduce the learning curve associated with ethical hacking tools, making them accessible to both novice and experienced users.

Detailed documentation and minimal educational resources will be provided to support users in effectively utilizing the tools.

Comprehensive Tool Functionality:
The toolset will cover a broad range of functionalities, including but not limited to:

Network Scanning: Tools for resolving hostnames to IP addresses, scanning for open ports, and identifying connected devices.

Web Application Testing: Capabilities for scanning web applications, reading content, and gathering information from web pages.

Social Engineering: Tools for simulating phishing attacks and social engineering scenarios, including Goldphish and Cam Phishing.

Denial-of-Service (DoS) Tools: Functionality to simulate DoS attacks for stress-testing systems and networks.

Payload Creation: A dedicated tool for generating payloads that facilitate remote connections and exploit vulnerabilities.

Hardware Integration: Innovative tools such as an Arduino-powered Wi-Fi jammer will be included to address unique cybersecurity scenarios.

Ethical and Responsible Use:
Emphasis on ethical hacking practices is a cornerstone of the Net-Caerus project. The toolkit will incorporate guidelines and warnings to encourage responsible usage of its functionalities.

Users will be required to acknowledge ethical considerations before accessing certain powerful tools, promoting a culture of responsible hacking within the cybersecurity community.

Scalability and Future Updates:
The architecture of Net-Caerus is designed to be scalable, allowing for the integration of new tools and features as they are developed. Regular updates will be essential to address emerging cybersecurity threats and enhance tool functionalities.

The project will also consider user feedback for continuous improvement and expansion of tool capabilities.

Target Audience
The primary audience for Net-Caerus includes:

Cybersecurity Professionals: Individuals working in cybersecurity roles, including penetration testers, security analysts, and ethical hackers, who require a comprehensive set of tools for their assessments.

Educational Institutions: Schools, colleges, and training organizations that teach ethical hacking and cybersecurity practices, benefiting from the integrated toolset for practical learning.

Organizations: Businesses and government agencies seeking to enhance their cybersecurity posture through efficient vulnerability assessments and penetration testing.

Product Perspective of Net-Caerus

Net-Caerus is positioned as an innovative solution in the rapidly evolving landscape of cybersecurity tools, specifically targeting the ethical hacking and penetration testing domains. The following outlines the product's perspective regarding its relationship to existing tools, its unique features, and the overall environment in which it operates.

1. Relation to Existing Tools
Integration and Unification: Net-Caerus addresses the existing fragmentation in the cybersecurity tool landscape. Unlike current solutions that often consist of disparate tools with limited integration, Net-Caerus consolidates multiple functionalities into a single platform. This integration streamlines workflows and enhances the efficiency of penetration testing processes, allowing users to switch seamlessly between tools without losing focus or productivity.

Versatile Toolkit: The toolset encompasses a wide range of functionalities, including network scanning, web application testing, social engineering, and DoS simulation. While many existing tools focus on specific niches, Net-Caerus aims to provide a holistic solution that caters to diverse penetration testing needs. This versatility makes it suitable for a variety of users, from novices to seasoned professionals.

User-Friendly Interface: Many existing ethical hacking tools suffer from steep learning curves due to their complex interfaces. Net-Caerus prioritizes user experience by offering an intuitive and straightforward interface, allowing users to navigate the toolkit efficiently. This focus on usability empowers users to maximize the toolkit's potential, regardless of their technical background.

2. Operating Environment

Technical Environment: Net-Caerus will operate within various technical environments, including Linux distributions and web browsers. This dual operating environment ensures accessibility for a wide range of users, regardless of their system preferences.

Regulatory and Compliance Considerations: The development of Net-Caerus takes into account the legal and ethical considerations surrounding the use of ethical hacking tools. Compliance with cybersecurity regulations and adherence to ethical standards will be integral to the product's development and deployment.

Evolving Threat Landscape: As cyber threats continue to evolve in complexity and frequency, Net-Caerus will be designed with scalability in mind. The toolkit will be regularly updated to incorporate new tools and features, ensuring that it remains relevant and effective in combating emerging cybersecurity threats.

Interview Questions to Requirement Gathering.

The questionnaire distributed to cybersecurity professionals included the following sections:

Section 1: Participant Information

- Name:
- Age:
- Gender:
- Occupation:
- Experience in Cybersecurity (in years):
- Educational Background in Cybersecurity:

Section 2: Current Practices and Challenges

15. What tools do you currently use for penetration testing and ethical hacking?
- Participants were asked to list all tools they use and describe their primary use cases. This information helps identify commonly used tools and their perceived effectiveness.

16. How would you rate the ease of use of these tools?
- (Scale of 1-5, where 1 is very difficult and 5 is very easy.) This question aimed to gauge the user-friendliness of current tools.

17. What challenges do you face with the current tools you use?
- Participants could describe issues such as lack of integration, user interface problems, or insufficient documentation.

18. Do you use multiple tools to conduct a single assessment? If yes, how many on average?
- This question explored the complexity of current workflows and the need for multiple tools.

19. Have you encountered any legal or compliance issues while performing ethical hacking?
- Participants were asked to describe any instances of legal or compliance issues they faced, highlighting the importance of legal awareness in ethical hacking.

Section 3: Preferences and Needs

20. What features do you believe are essential in an ethical hacking toolkit?
- Participants were asked to list features they consider important and explain why, helping to prioritize feature development.

21. How important is it for you to have an educational section integrated within the toolkit?
- (Scale of 1-5, where 1 is not important and 5 is very important.) This question gauged the value placed on integrated educational resources.

22. What topics or areas should be covered in the educational section?
- Participants could suggest topics such as tutorials on specific tools, best practices in ethical hacking, or legal guidelines.

23. Would you prefer a web-based platform, a downloadable toolkit, or both? Why?
- This question explored user preferences for platform accessibility.

24. How likely are you to use a new integrated ethical hacking system that includes both tools and educational resources?
- (Scale of 1-5, where 1 is very unlikely and 5 is very likely.) This helped assess the potential adoption of the proposed system.

Section 4: Feedback and Suggestions

25. What improvements would you suggest for current ethical hacking tools?
- Participants were asked for specific suggestions to improve existing tools.

26. Do you have any suggestions for the design and functionality of the proposed integrated ethical hacking system?
- This question invited feedback on the proposed system's design and functionality.

27. How do you think an integrated ethical hacking system could benefit your work or studies in cybersecurity?

- Participants could describe potential benefits, helping to align the system with user needs.

28. Would you be willing to participate in a beta testing phase for the new system?
- If yes, participants could provide their contact information, helping to identify potential beta testers.

The insights gained from this questionnaire informed the requirement analysis and helped shape the design and development of the Integrated Ethical Hacking System. The data collected provided valuable feedback on user preferences, current challenges, and essential features, ensuring the final product meets the needs of cybersecurity professionals effectively.

APPENDIX B - DESIGN DOCUMENTATION

Use case Diagram

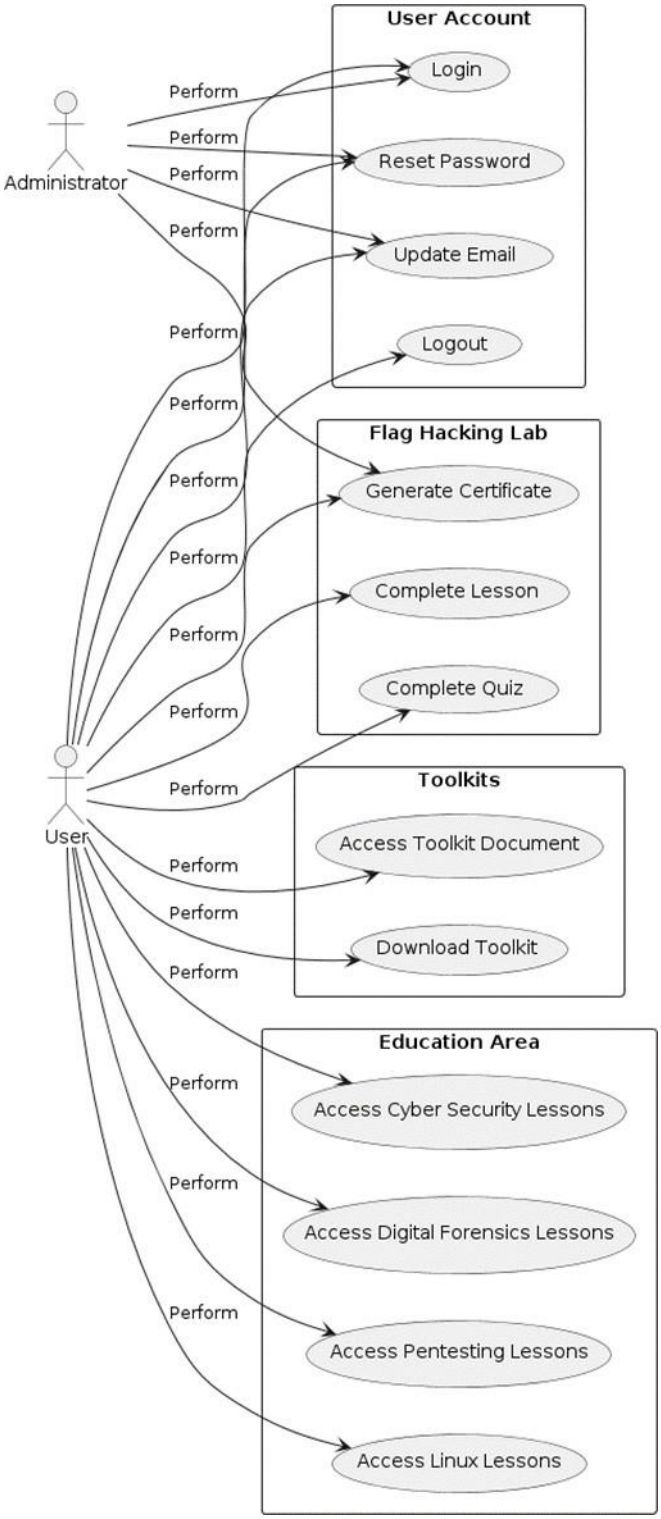

Figure 23 - use case diagram

Class Diagram

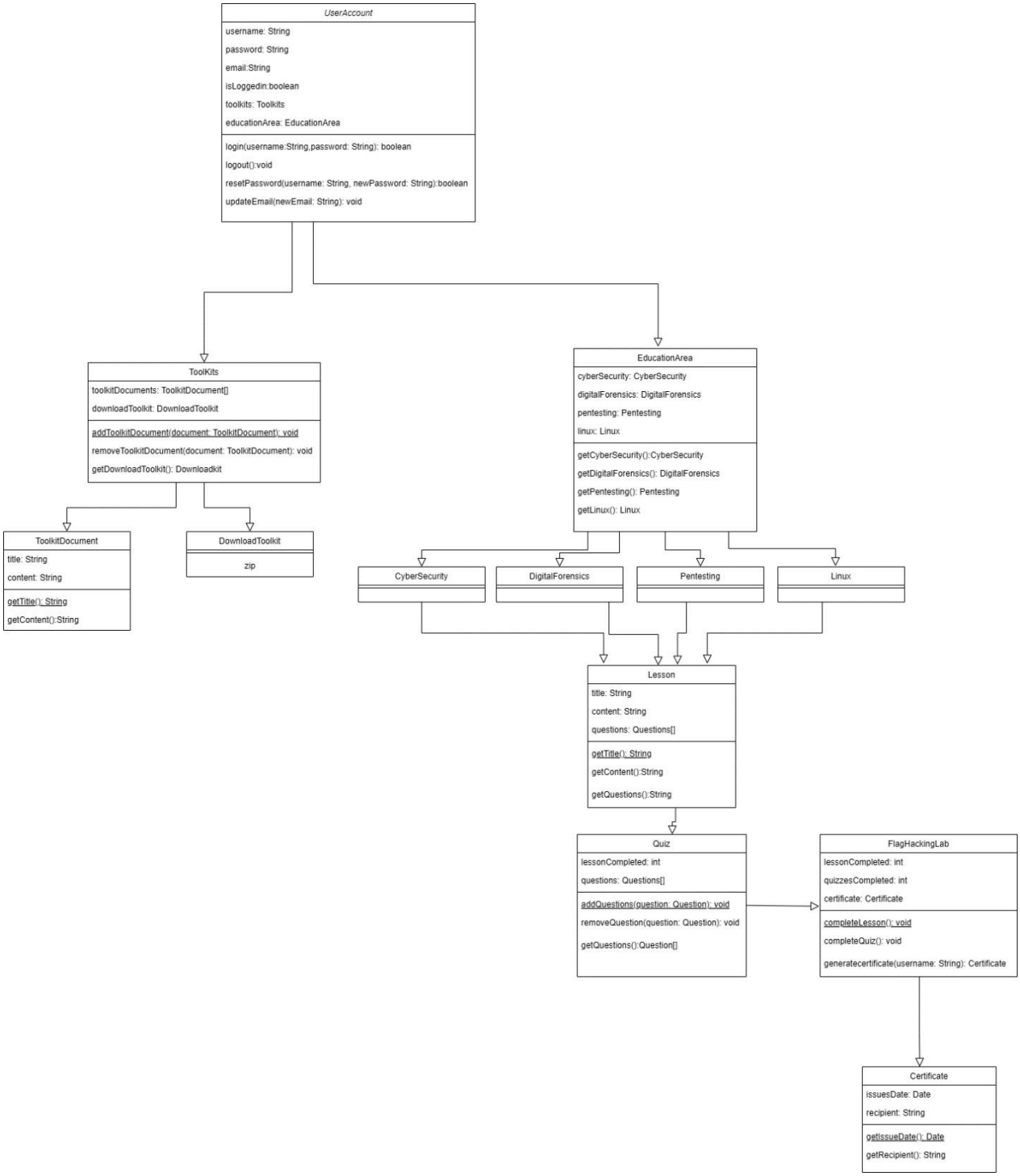

Figure 24 - Class Diagram

Sequence Diagrams

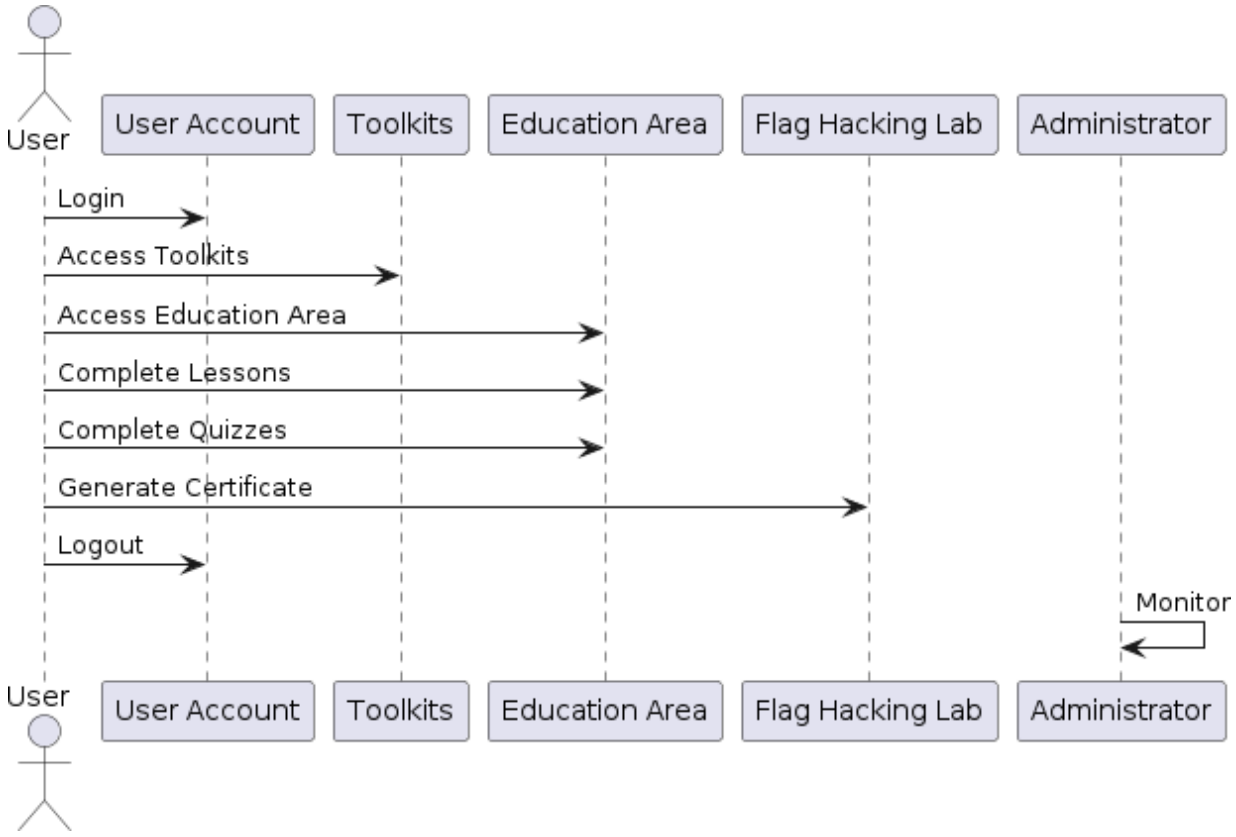

Figure 25 - Sequence Diagram

Activity Diagrams

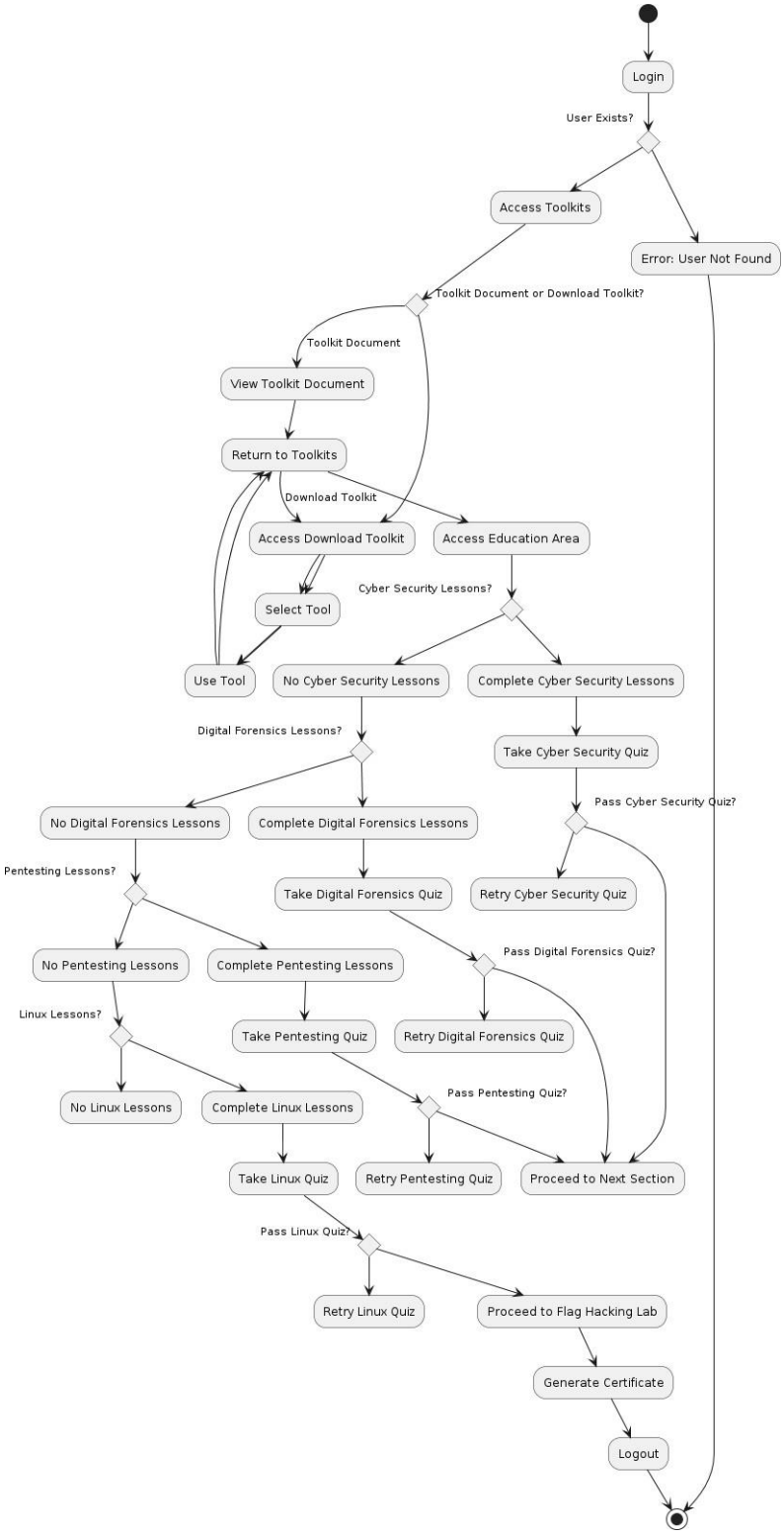

Figure 26- Activity Diagrams

APPENDIX C – USER DOCUMENTATION

User Interface Design

The user interface is organized for ease of use, catering to both novice and experienced users.

Web-Based Platform UI

Homepage: The homepage introduces the project, emphasizing its ethical focus and open-source MIT licensing.

Tool Access Interface: Organized in a dashboard format, each tool's page contains an overview, usage guide, and examples.

Educational Section UI: The educational section categorizes content by difficulty, allowing users to follow a structured path. Tutorials, video content, and labs provide an interactive learning experience.

1. Login Page
2. Register Page
3. Home Page
4. Edu page
5. Contact
6. About

Login Page

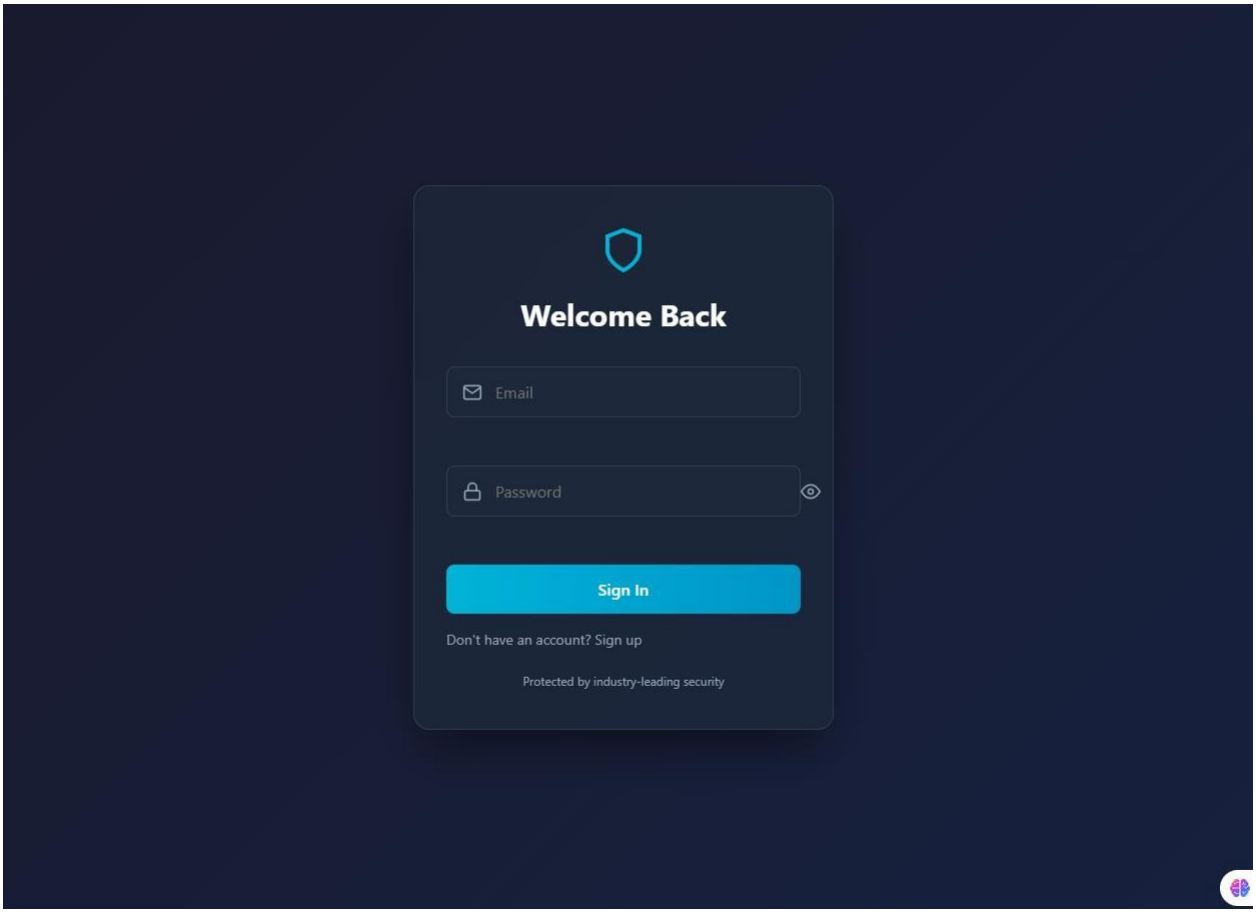

Figure 27 - login page

Register Page

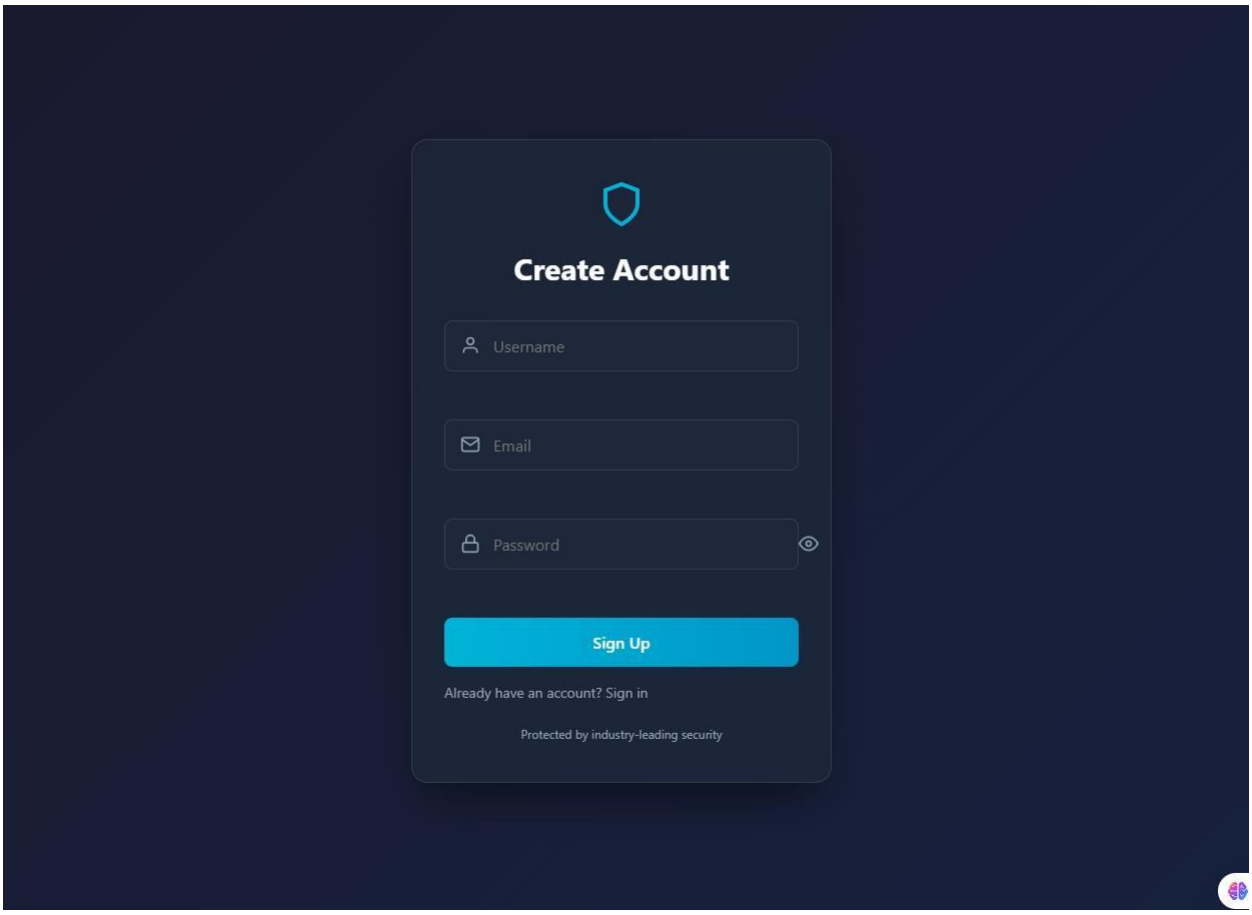

Figure 28 - Register Page

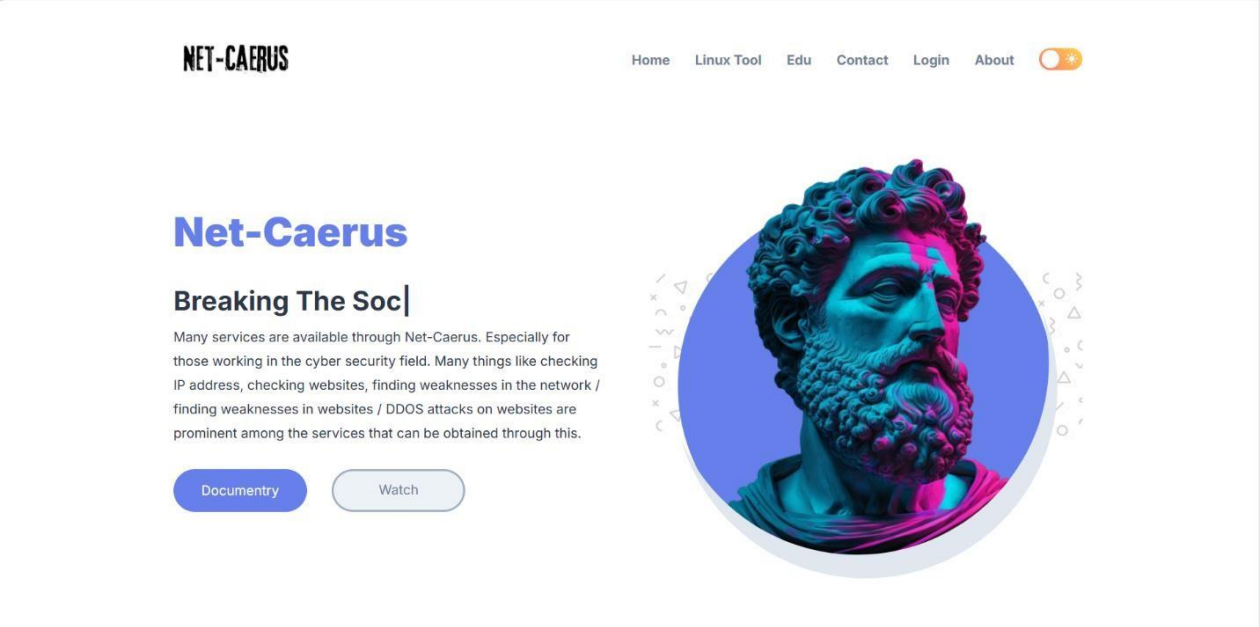

Figure 29 - Home Page

Home Page

Dark Mood

Figure 30 Home Page dark Mood

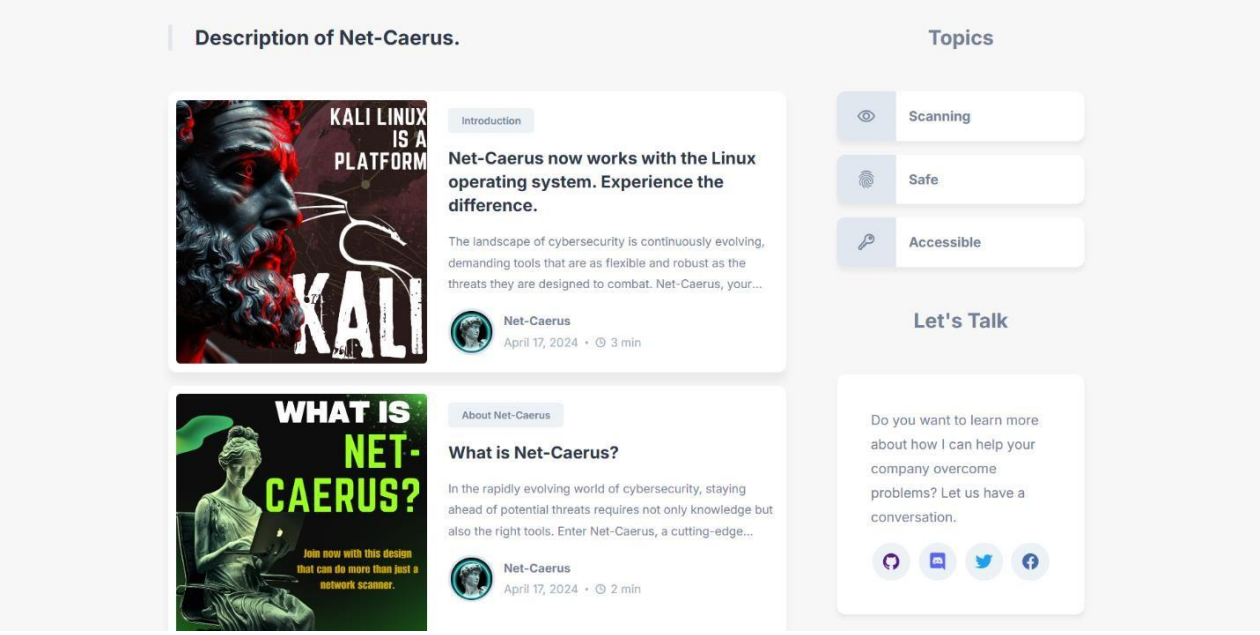

Figure 31 - Home Page Down

Home Page Down

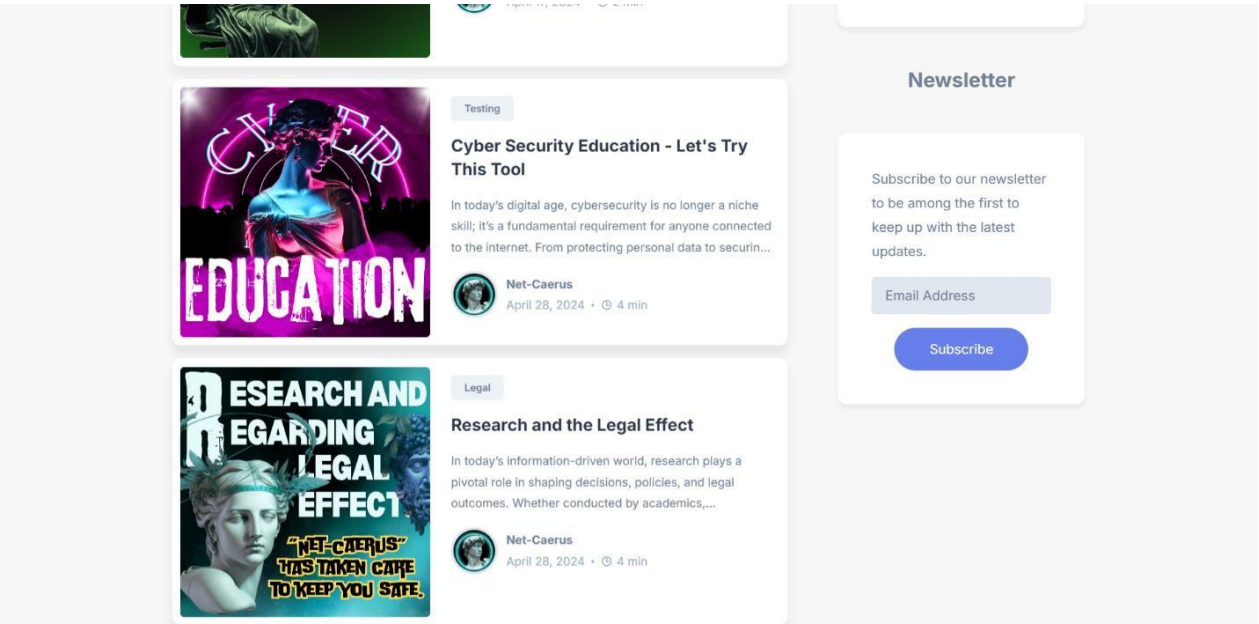

Figure 32 Home Page Down

Home Page Down

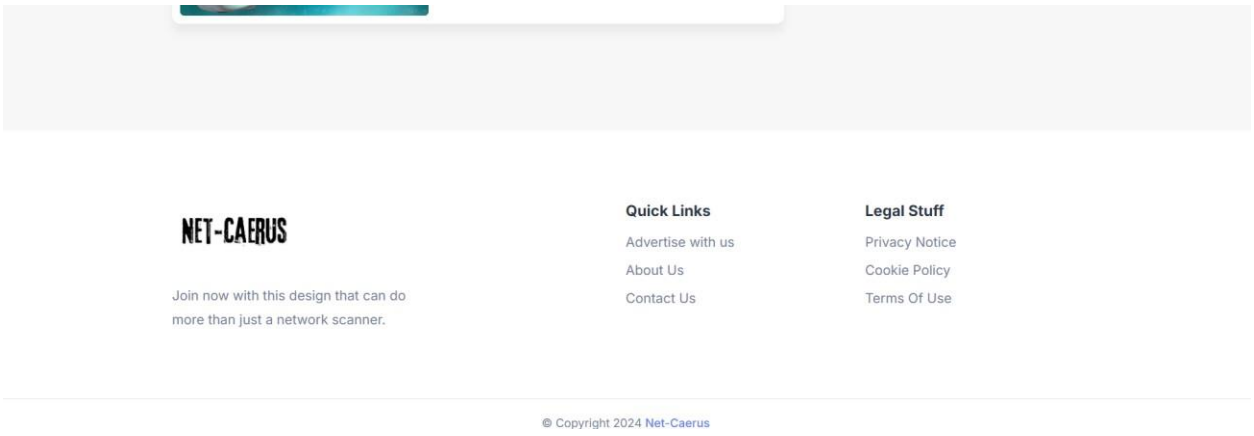

Figure 33 - footer

Footer

Education Page (Lesson)

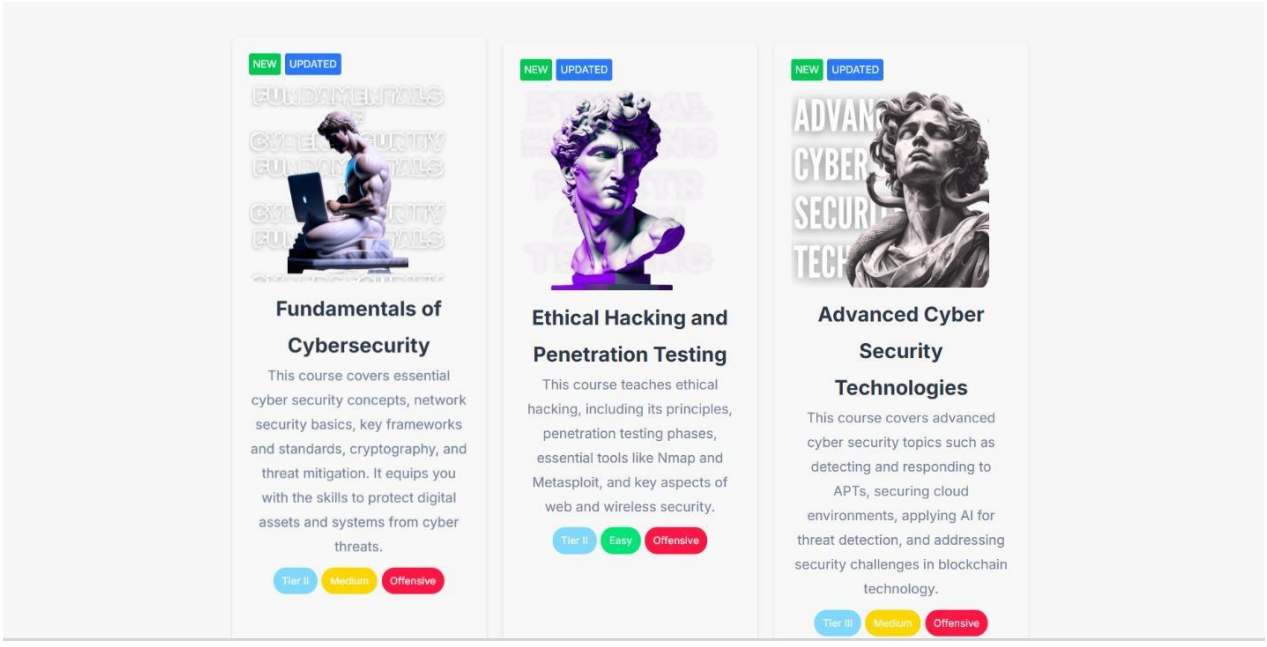

Figure 34 - Edu page

Education page 2

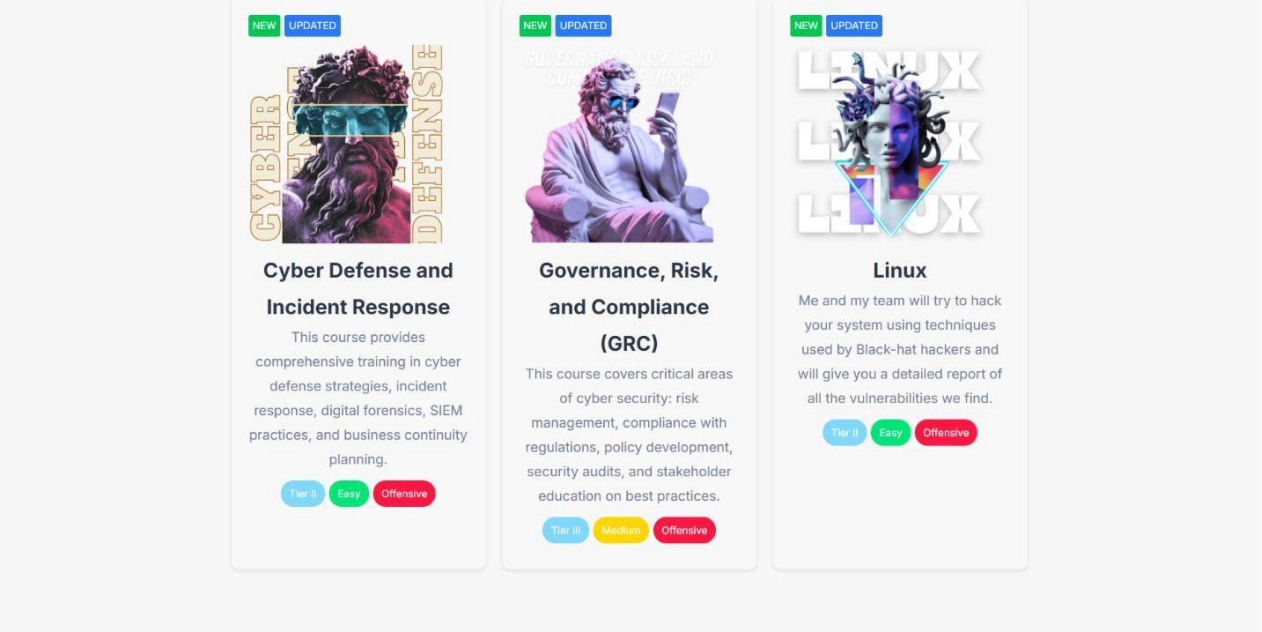

Figure 35 - Edu page 2

Video Lesson Page

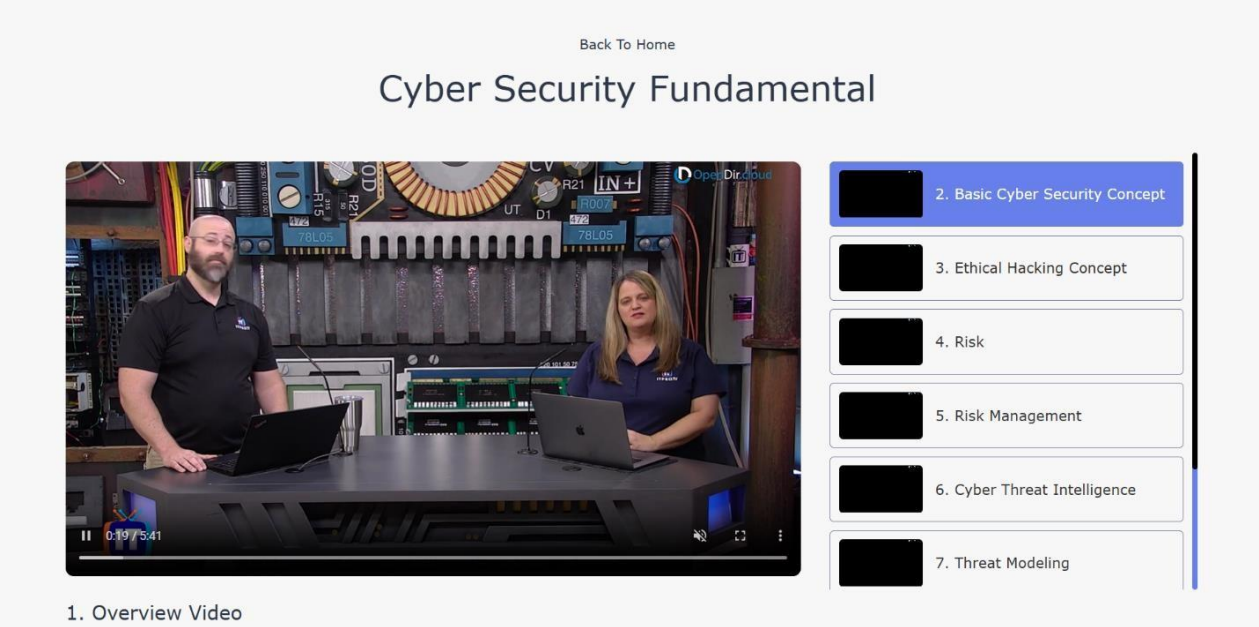

Figure 36 - video Page

PDF Downloadable Page

Figure 37 - Downloadable PDF

Contact Page

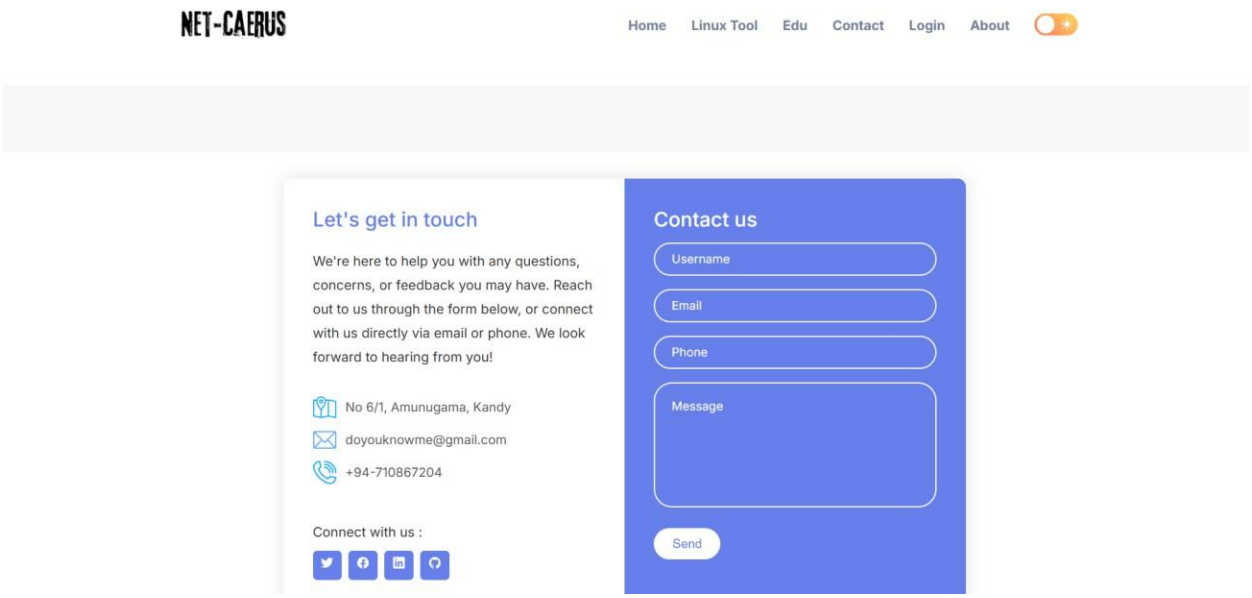

Figure 38 - Contact Page

About Page

Figure 39 About page

Lesson Page Readble

Figure 40 - Lesson Page

APPENDIX D – TEST CASES

Test Case 1- Login Module

Table 5 Login Module test case

Test Case ID	1
Tested Component	Login Activity for Guest user
Module Name	User Module
Tested Area	Login Functionality

Testing the login module by entering the Email and the password.				
No	High Level test steps	Expected Result	Actual Result	Status
1	User enter invalid user name and/or password	Display error message	pending	pending
2	User enter empty username	Display error message	pending	pending
3	User enter empty password	Display error message	pending	pending

Test Case 2- Registration Module

Table 6 Resgistraion module

Test Case ID	1			
Tested Component	Registration Activity for Guest user			
Module Name	User Module			
Tested Area	Registration Functionality			
Testing the Registration module.				
No	High Level test steps	Expected Result	Actual Result	Status
1	Create new account by adding required fields and hit on Register button	Show successfully registered message	pending	Pass
2	User enter empty fields	Display error message	pending	Pass
3	User enter invalid input to the fields	Display error message	pending	pass

Test Data and Test Results

Table 7 test results

Test Case ID	Expected Result	Actual Result	Pass/Fail
TC-01	Lists all open ports	All open ports listed accurately	Pass
TC-02	Initiates DoS traffic correctly	DoS attempt simulated as expected	Pass
TC-03	Changes MAC address successfully	MAC address changed without issues	Pass
TC-04	Phishing server accessible on designated IP	Server operational and page rendered	Pass
TC-05	Admin page scanner locates hidden pages	Common admin pages identified	Pass

APPENDIX E – CODE LISTNING (MAJOR CODE SEGMENT)

Development project File

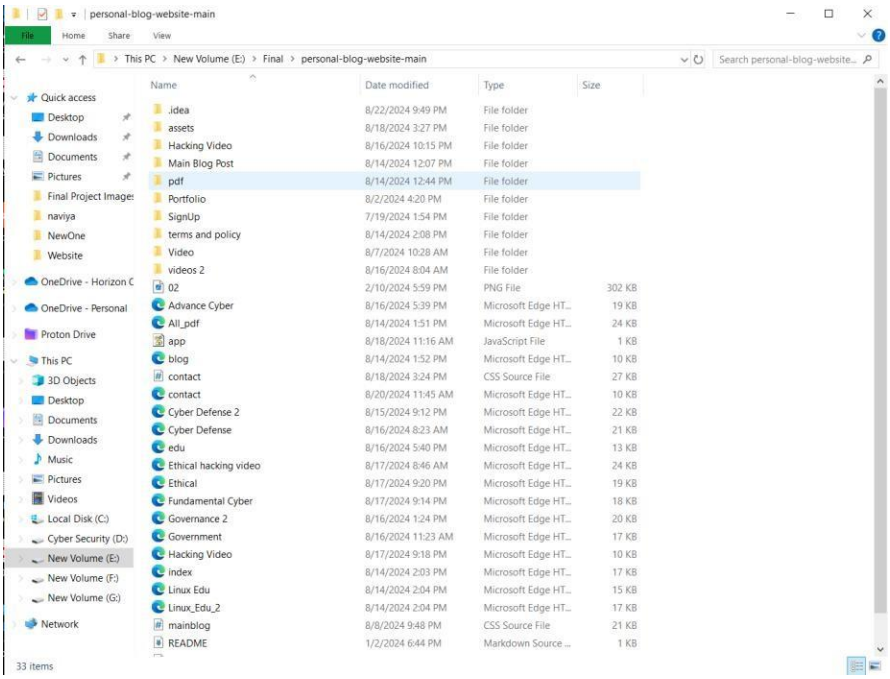

Figure 41 - Project File

Linux Project File

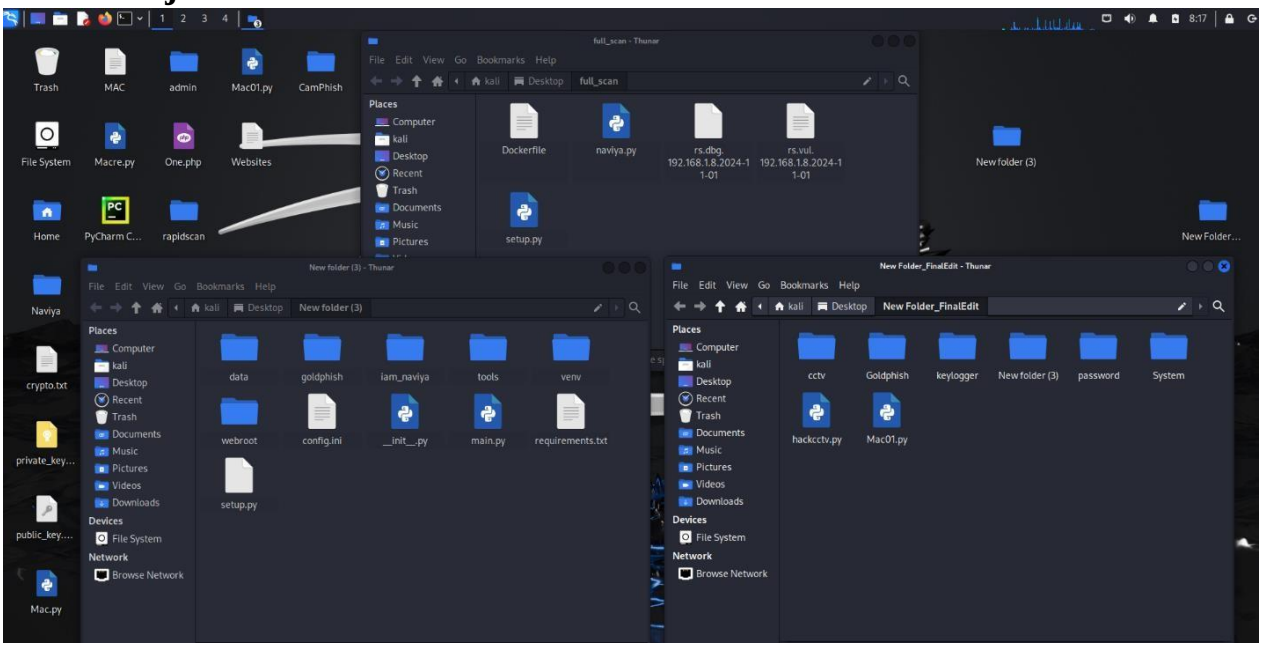

Figure 42 Linux File

Main coding Part Python Tool Kit

```
# main.py

from tools import findIP_main, dos_main, scan_port_main, who_main,
display_wifi_passwords, get_instagram_profile_info, get_cookies_from_url,
scan_website, get_ip_addresses, fetch_authenticated_url, fetch_web_page,
run_goldphish
from tools.admin import website

def display_menu():
    print("Select a tool to run:")
    print("1. Tool 1: Scan website")
    print("2. Tool 2: Resolve hostname to IP address")
    print("3. Tool 3: Who is Lookup")
    print("4. Tool 4: Perform DoS attack")
    print("5. Tool 5: Find Wifi Password")
    print("6. Tool 6: Find Instagram Profile")
    print("7. Tool 7: Cookies information")
    print("8. Tool 8: Scan website Admin Page")
    print("9. Tool 9: Find All IP Address")
    print("10. Tool 10: HTTP authentication")
    print("11. Tool 11: Reading Web Page")
    print("12. Tool 12: Goldphish phishing server")
    print("13. Exit")
```

```
def get_user_choice():
    return input("Enter your choice (1-13): ")

def main():
    while True:
        display_menu()
        choice = get_user_choice()

        if choice == '1':
            scan_port_main()
        elif choice == '2':
            findIP_main()
        elif choice == '3':
            who_main()
        elif choice == '4':
            dos_main()
        elif choice == '5':
            display_wifi_passwords()
        elif choice == '6':
            get_instagram_profile_info()
        elif choice == '7':
            get_cookies_from_url()
        elif choice == '8':
            scan_website(website)
        elif choice == '9':
            hostname = input("Enter the Hostname: ")
            get_ip_addresses(hostname)
        elif choice == '10':
            url = input("Enter the URL: ")
            username = input("Enter your username: ")
            password = input("Enter your password: ")
            fetch_authenticated_url(url, username, password)
        elif choice == '11':
            url = input("Enter the URL of the web page: ")
            fetch_web_page(url)
        elif choice == '12':
            run_goldphish()
        elif choice == '13':
            break
        else:
            print("Invalid choice. Please try again.")

        back_to_menu = input("Would you like to go back to the menu?
(yes/no): ")
        if back_to_menu.lower() != 'yes':
            break

if __name__ == "__main__":
    main()
```

About how the Net-Caerus Tool works.

Below is how to run the Zip file after downloading the toolset from the website.

Figure 43 main.py

First, this toolset is divided into several parts. To run the first set, you need to run this Python command.

Python3 main.py

Since this runs on Linux, the user must have basic knowledge of Linux.

The toolset in the first phase of this toolset appears as shown below when run. It consists of about 14 primary and medium-sized tools.

Figure 44 menu

Let's focus on the tools first. All of these tools are tools created in the Python language.

After giving the Python3 main.py command, the tool set is run very easily in the first instance. Now you are given the opportunity to choose a tool of your choice in the second instance. You can use any tool of your choice from 1-14 for your use. (According to your need)

" Enter your choice (1-14): " You can enter the number corresponding to the tool of your choice.

Let's focus on these tools one by one.

The first 14 tools are categorized as follows.

Table 8 Linux Menu

Select a tool to run: 1. Tool 1: Scan website 2. Tool 2: Resolve hostname to IP address 3. Tool 3: Who is Lookup 4. Tool 4: Perform DoS attack 5. Tool 5: Find Wifi Password 6. Tool 6: Find Instagram Profile 7. Tool 7: Cookies information 8. Tool 8: Scan website Admin Page 9. Tool 9: Find All IP Address 10. Tool 10: HTTP authentication 11. Tool 11: Reading Web Page 12. Tool 12: Goldphish phishing server 13. Tool 13: Change MAC address 14. Exit

To understand this simply, let's use all these tools one by one.

Let's focus on the first tool.

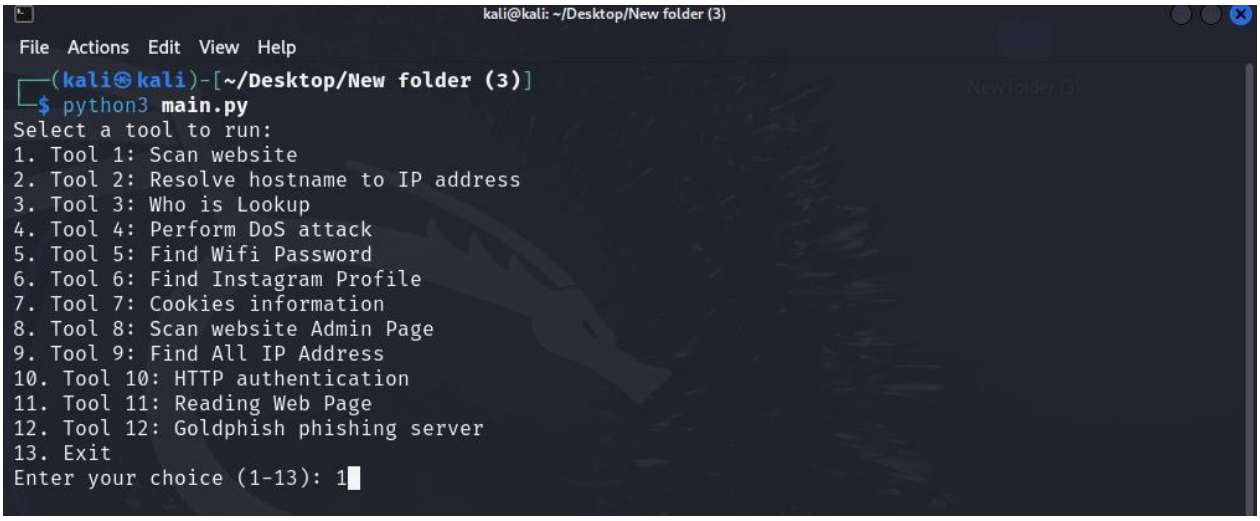

Figure 45 scan website number 1

To run this first tool, you need to issue command number **1** as shown in the screenshot. After entering that command, you must press the **'Enter Key'** on the keyboard.

When you enter it, you will see the bar below.

In that column, you will see a message asking you to enter the website you want to scan.

In the example below, I have used the website https://google-gruyere.appspot.com. Scanning the site shows that both **Port 21** and **Port 80** are open. Pay close attention to the screenshot below, and let's take another example to understand this better.

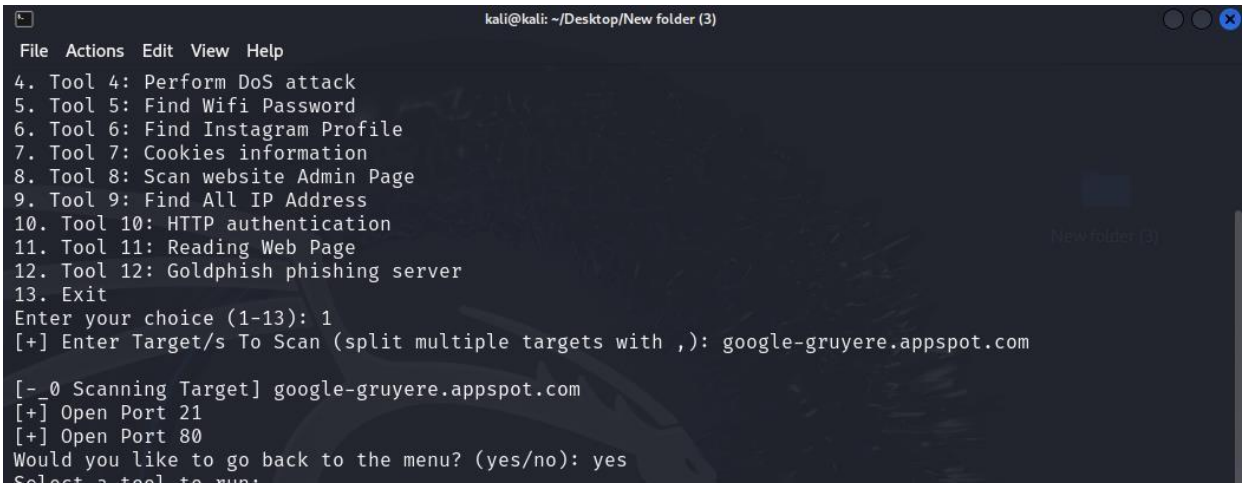

Figure 46 scan result

This is the first example, and we will now move on to the second example.

Look at the example below. For that example, I used a different website, namely https://redlogicx.com.

When scanning the **Redlogicx** website, it clearly shows that **Ports 21, 25, 53, and 80 are open**.

```
12. Tool 12: Goldphish phishing server
13. Exit
Enter your choice (1-13): 1
[+] Enter Target/s To Scan (split multiple targets with ,): redlogicx.com

[-_0 Scanning Target] redlogicx.com
[+] Open Port 21
[+] Open Port 25
[+] Open Port 53
[+] Open Port 80
Would you like to go back to the menu? (yes/no):
```

Figure 47 2nd website scan

Let's take another example of this. Let's use the IP address of a server. This is the IP address of my **Metasploitable Machine**. Pay attention to the screenshot below.

```
5. Tool 5: Find Wifi Password
6. Tool 6: Find Instagram Profile
7. Tool 7: Cookies information
8. Tool 8: Scan website Admin Page
9. Tool 9: Find All IP Address
10. Tool 10: HTTP authentication
11. Tool 11: Reading Web Page
12. Tool 12: Goldphish phishing server
13. Exit
Enter your choice (1-13): 1
[+] Enter Target/s To Scan (split multiple targets with ,): 192.168.161.88

[-_0 Scanning Target] 192.168.161.88
[+] Open Port 21: 220 (vsFTPd 2.3.4)
[+] Open Port 22: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
[+] Open Port 23
[+] Open Port 25: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[+] Open Port 53
[+] Open Port 80
Would you like to go back to the menu? (yes/no): yes
```

Figure 48 another website scan

For that, I have used the IP Address 192.168.161.88. That is the IP address of the Metasploitable Machine. So pay attention to that. You can clearly see that several ports are fully identified.

These are **ports 21, 22, 23, 25, 53 and 80** respectively. In particular, this scan found **vsftpd 2.3.4** (very secure FTP daemon) FTP server for Linux and Unix-like systems. Also, **SSH-2.0-OpenSSH_4.7p1 debian-**8ubuntu1 and this has been identified as a **Metasploitable.**

Something very important has been noted here. That is, in this scan we were able to find more information than before, and it seems that something special has been revealed among this information. That is, the revelation about **VSFTPD v2.3.4**.

If we talk about this briefly, this is a **VSFTPD v2.3.4 Backdoor Command Execution.**

This module exploits a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011 according to the most recent information available. This backdoor was removed on July 3rd 2011.

We can keep in mind that scanning something like this with this tool is a big advantage. Especially since such tools are common in today's world, many cyber security experts are of the opinion that some tools do not provide such details.

Resolve hostname to IP address

Now let's focus on the second tool. All you need to do here is enter the **number 2** in the command **"Enter your choice (1-14):".** You will be prompted to enter the host name as shown below. You can enter the website of your choice in it. What it actually does is resolve hostname to IP address.

Figure 49 2nd Tool

The website I have provided here is https://hackthissite.org/. After providing it, the final result here is the **IP address 137.74.187.102,** which is easily found.

Let's look at another example. I drew attention to my campus' website. Let's take a look at the details below. https://horizoncampus.edu.lk

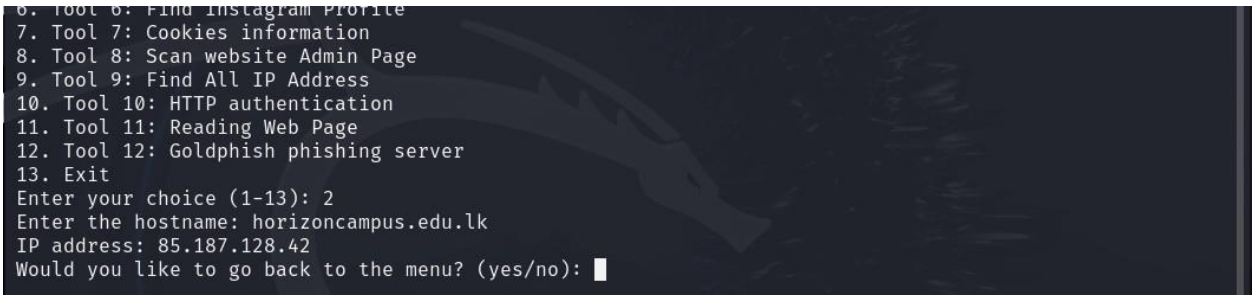

Figure 50 Campus website

After providing it, the **IP address came up as 85.187.128.42.**

In short, this script allows users to input a hostname and attempts to display its corresponding IP address, handling errors gracefully if the hostname can't be found.

Who is Lookup

Let's focus on the 3rd tool.

As before, after using the 2nd tool, when you give **"Yes"** to the command **"Would you like to go back to the menu? (yes/no): ",** you will go back to the main menu. Then you can give the **number 3** and go to the 3rd tool.

After doing so, you will see a prompt saying "Enter the domain name :". Once this appears, you can get details from this tool by providing the relevant website. For example, take a look at the screenshot below.

Figure 51 Result

The website I have provided is **redlogicx.com**. Once I have provided it, pay close attention to the details provided about it.

By paying attention to the information provided, a large amount of data could be collected. That is,

this **WHOIS LookUp** output provides essential information, particularly valuable in domain management, cybersecurity, and legal contexts. Here's why each section of the data is significant:

1. **Domain Information:**
   - **Domain Name**: Identifies the specific domain being queried, in this case, REDLOGICX.COM.
   - **Registrar:** Indicates the company responsible for managing the domain registration (Porkbun LLC). This information can be used to contact the registrar for administrative or technical issues related to the domain.

2. **Registration and Expiration Dates:**
   - **Creation Date:** Marks when the domain was initially registered (2024-02-29), giving insights into the domain's age. Newly registered domains may sometimes be associated with temporary or suspicious websites, while older domains are often more established.
   - **Expiration Date:** Indicates when the current registration expires (2025-02-28). Knowing the expiration date helps domain managers ensure timely renewals and can also be useful if someone else is interested in acquiring the domain once it expires.

3. **Name Servers:**
   - **Name Servers:** These are servers that translate the domain name into an IP address. They are critical for the domain's web accessibility. Here, the domain uses multiple nameservers provided by hostingoven.com and hostingoven.net, which suggests it has redundancy for improved stability.

4. **Domain Status:**
   - **Status Codes:** These indicate the current operational status and restrictions of the domain. Here, the status includes *clientDeleteProhibited* and *clientTransferProhibited*, meaning that the domain is protected from unauthorized deletion or transfer. These restrictions are essential for domain security, helping prevent hijacking.

5. **Contact and Privacy Information:**
   - **Registrant Info (Name, Org, Address):** Indicates that the domain owner is using a privacy service (WhoisLookUp Privacy by Private by Design, LLC). Many registrars offer this to shield registrant details from the public, enhancing user privacy.
   - **Contact Email:** Often used for administrative queries or to report abuse. Here, abuse@porkbun.com is likely a general contact for reporting misuse or suspicious activity involving this domain.

6. **DNSSEC Status:**
   - **DNSSEC:** A cryptographic protocol that ensures DNS responses haven't been tampered with. Here, it's unsigned, meaning it lacks this security layer, which may make it more vulnerable to certain attacks, though this depends on the site's security needs.

In summary, this WHOIS-LookUp data provides critical insight into the registration, ownership, and security of a domain. It is particularly useful for managing domain lifecycles, protecting brand integrity, and conducting security investigations or due diligence.

Let's move on to the 4th tool now.

Perform DoS attack

Let's first explain what a **DoS attack** is. A **denial of service (DoS)** attack is a cyberattack that aims to make a server, network, or service unavailable to legitimate users by overwhelming it with illegitimate requests or data streams. The goal is to exhaust the target's resources, causing it to slow down, crash, or otherwise become unreachable.

The largest DDoS attack was against Google.

*This attack targeted a **Google Cloud Armor** user with HTTPS on June 1, 2020. The attack peaked at 46 million requests per second (rps) and lasted for 69 minutes. The attack was 76% larger than the previous record and was similar to receiving all of the daily requests to Wikipedia in 10 seconds.*

Now let's focus on our 4th tool, "**Perform DoS attack**". This is a really damaging and powerful tool. Like all the other tools, this one is also written in Python. As before, you can run this tool by pressing the number 4. Pay attention to the screenshot.

The website I took as the first example is the website of a powerful political party in Sri Lanka, **slpp.org**. So, below is how that website faced the DoS attack. Pay attention to the screenshot below. (All activities here are for educational purposes only.)

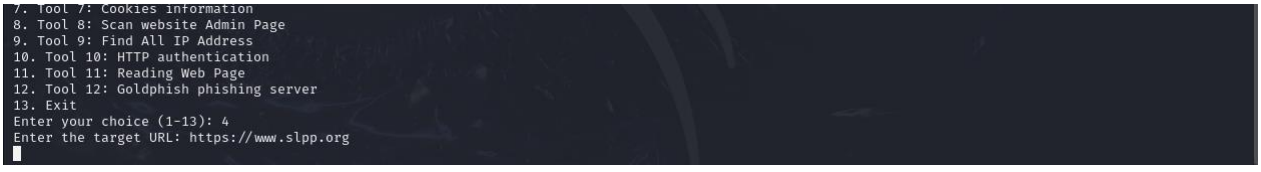

Figure 52 4th tool

When this attack was launched, the tool returned a **503 loop**, indicating that the DoS attack was temporarily successful.

When the attack is successful, we cannot see the website. This is a screenshot of it.

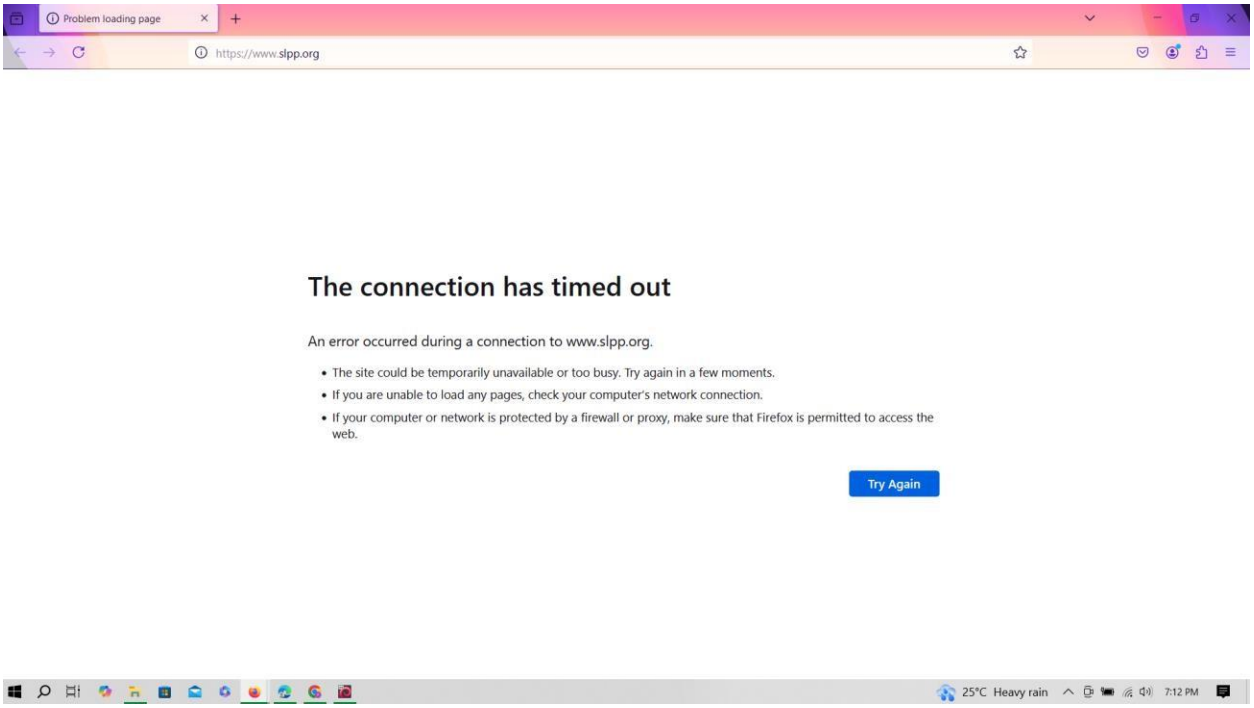

Figure 53 DoS Result

This screenshot shows how the slpp.org website went down for a few minutes. What's really happening here is that the tool is sending a large number of requests and the website is unable to handle them, causing it to temporarily stop.

If you see a continuous 503 Service Unavailable response during a DoS (Denial of Service) attempt, it indicates that the attack is having a significant impact on the server. Here's what it means:

What Continuous 503 Status Codes Mean

A 503 Service Unavailable means that the server is temporarily unable to handle the request. This often happens when the server is overwhelmed by too many requests and cannot process new ones due to exhausted resources such as CPU, memory, or network bandwidth.

When the 503 status repeats in a loop, it usually suggests that the server is constantly under heavy load and cannot recover between requests.

Indicators of DoS Attack Success

Increased server downtime: Continuous 503 responses suggest that the server cannot handle the additional load for legitimate users and may be effectively down.

Resource depletion: The server may have run out of resources and cannot respond properly to requests until it is manually reset or the attack subsides.

Let's talk more about how a DoS attack can be successful.

Expected Output in a Successful DoS Attack

Response Codes:

- Error Codes (**e.g., 500, 502, 503, or 504**): If the server begins to overload, you may start to see error response codes indicating that the server is under strain. For example:
- **500** Internal Server Error: The server may be unable to process requests due to resource exhaustion.
- **504** Gateway Timeout: If there is a delay in the server's responses, it might result in a timeout.

Connection Failures:

- The script's retry mechanism may kick in if it cannot establish a connection, potentially leading to repeated "Connection error" messages as the server struggles to respond.
- After several retries, if the server completely fails, the script might output continuous "Connection error" messages.

Server Unresponsiveness:

- In a fully successful DoS scenario, the server might become unresponsive and stop replying to requests altogether. This would cause an indefinite loop of "Connection error" messages, or you might see a mixture of HTTP errors (**503, 504**) and retries.

Rate-Limiting Response Codes (Possible):

- Some servers have rate-limiting or DoS protection mechanisms that might respond with HTTP **429** (Too Many Requests). If the target URL has such protections, you might start seeing repeated 429 errors in the output.

Let's focus on one more example.

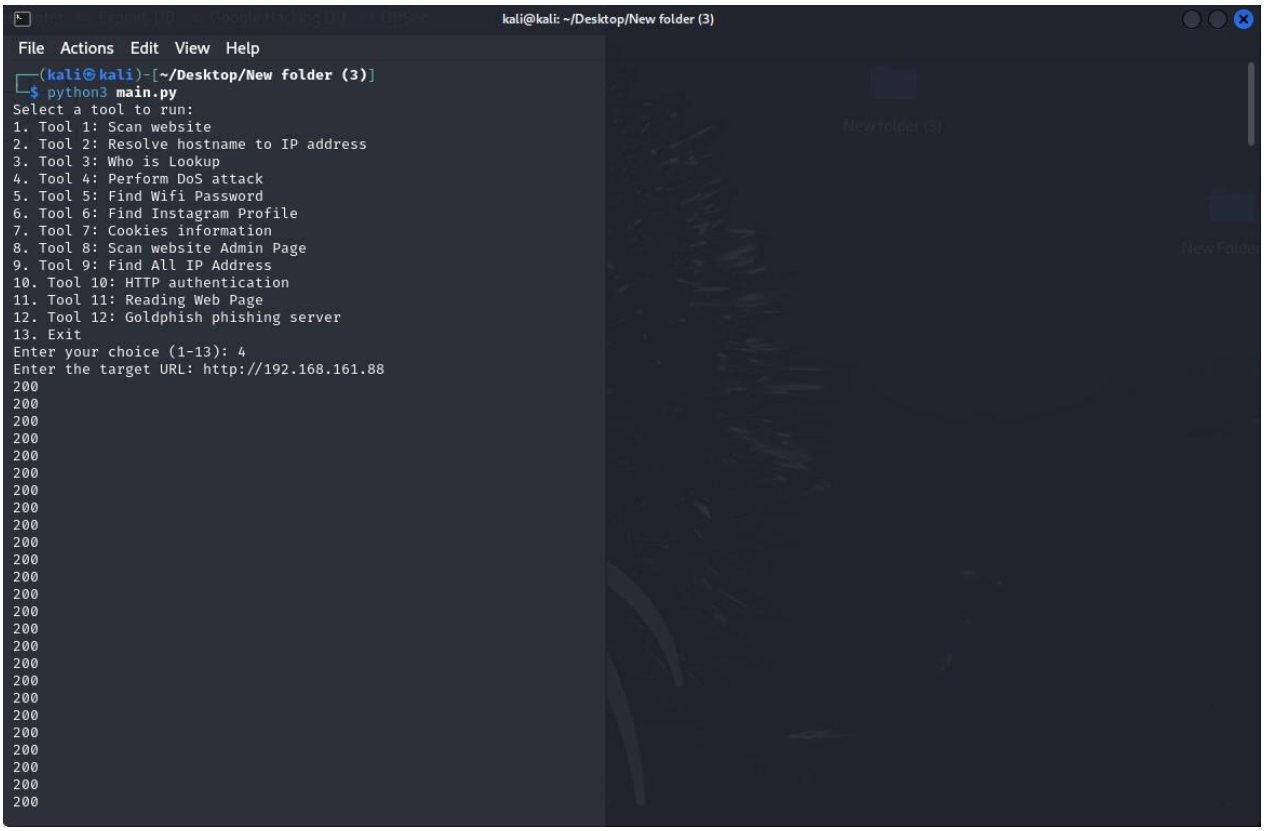

Figure 54 dos attack

This tool sets up a script to send a large number of POST requests to a specific URL, which can be considered a denial of service (DoS) attack. Below is a breakdown of how it works and the output to expect.

I directed the DOS attack to the Metasploitable Server and the result was this: The attack was launched by entering **https://192.168.161.88** as the URL.

It's important to talk about this tool in depth, so pay close attention to the following.

If your result is 200, this is the reason. (Pay attention to my screenshot above.)

If the script receives **200 OK** responses repeatedly in a loop, it indicates that the server is successfully processing each request and responding with a status that signifies "success." Here's what this could mean for the effectiveness of the DoS attack:

The server is still responding: Since the server can repeatedly return 200 OKs, it suggests that the server is not yet overwhelmed. A successful DoS attack will typically prevent the server from responding to requests altogether, causing timeouts or connection failures instead of 200 responses.

Increased server load: Even if the server is still able to respond with 200 OKs, the constant influx of requests can strain its resources. Depending on the server's infrastructure and resiliency (such as load balancing and rate limiting), this can lead to slow response times or even eventual failure if the load becomes unmanageable.

Potential partial success: In some cases, a high load may not immediately take down a server, but its performance may degrade, impacting legitimate users by slowing down services or reducing availability. This can be considered a partial success, as the server's ability to handle normal traffic may be affected.

Let's look at another example.

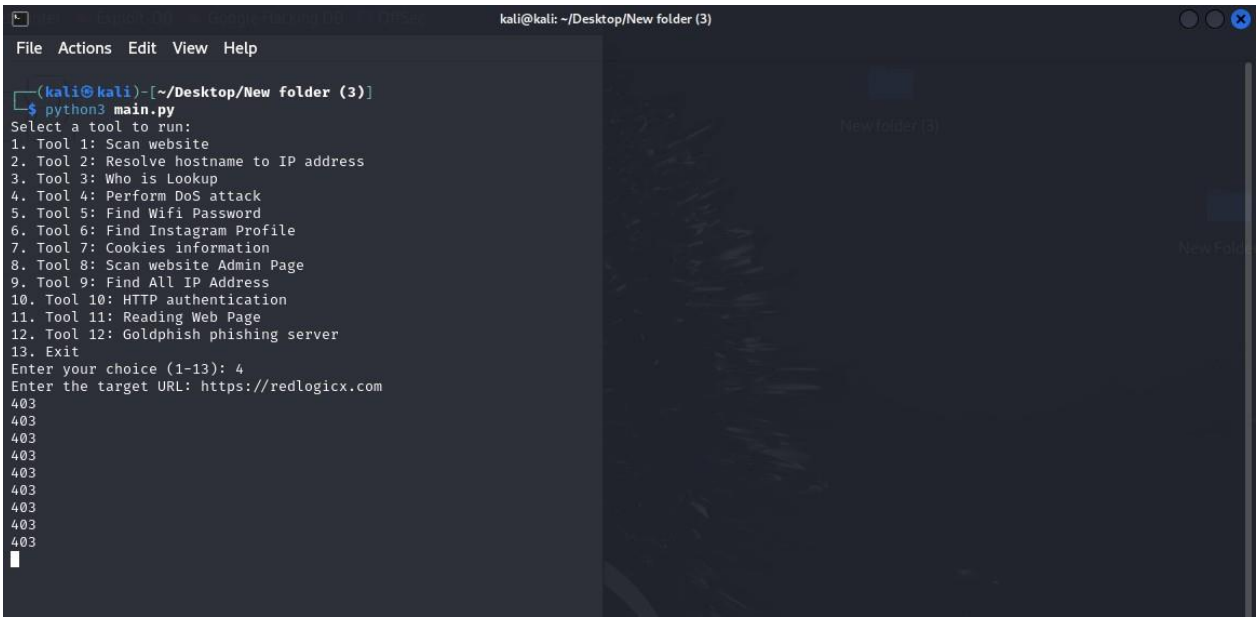

Figure 55 Another Site

For this example, I used the website **redlogicx.com** and it responds like a **403 loop**. This means,

Access Denied by the Server: The **403 Forbidden status** code typically means the server has rules or protections in place (such as a firewall, IP block, or rate-limiting) that prevent access to the requested resource. The server is functioning as intended by blocking the requests, which means it's not overwhelmed or taken down by the attack.

DoS Mitigation in Place: The server's ability to return 403 responses repeatedly suggests it may have DoS or intrusion detection mechanisms in place to mitigate this type of attack. These protections detect and block suspicious traffic patterns, preventing unauthorized access and reducing server load.

Attack Not Successful: Since the server is blocking the requests and still capable of responding, the DoS attack is not successful. The 403 responses show that the server's defenses are working effectively to reject potentially harmful traffic without compromising its availability.

In today's world, DoS/DDoS attacks have become an unethical and illegal business. The example below is the price charged by a certain illegal organization for DDoS attacks.

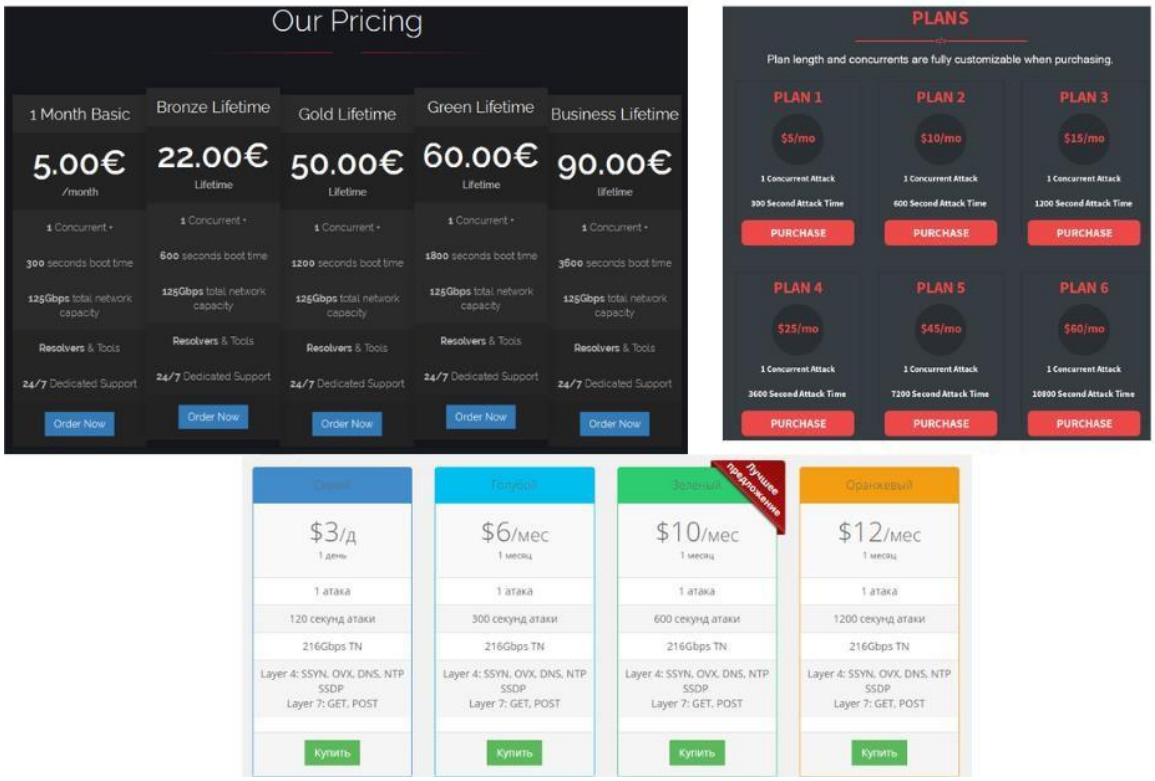

Figure 56 Dos Price

We create and distribute these tools completely free of charge to equip the world with all the knowledge. I know very well that all the tools I have created have an impact on both good and bad sides. But if we want to face the bad, we must definitely know the good side. So this is my effort to show the good side. Good and bad depend on the person using it, just like a knife can cut fruits and vegetables and can also harm a person with it. Using this for educational purposes only is a matter of being optimistic and happy about your ability and discipline.

Now let's focus on our next tool.

Find WIFI Password

Our next tool is a python-based tool that can find WIFI passwords. All of these tools are tools created based on Linux.

The This tool is described below. Please refer to the screenshot. To run it, type command number 5 in the terminal as before.

Figure 57 WiFi Password checker

This is a very simple tool that tries to show all your WIFI passwords in a single table. (The passwords is not shown in this screenshot because I have never connected WIFI to my Linux machine.)

If I show this as an example, it will work like this and give the result.

Table 9 Wifi Password List

SSID	Password	Active
Area_51	Alien#1998	yes
Network1	FlyCanWithMe24	
Network2	Cannot retrieve password	
Network3	No WPA security	

Key Points and Limitations

WPA Security Limitation: Only networks with WPA security will have an attempt to retrieve the password.

Active Network Limitation: *nmcli* can only retrieve the password of the currently active network (i.e., the one the device is currently connected to).

Permissions: Accessing stored Wi-Fi passwords typically requires administrative (root) privileges. Without proper permissions, the script may not retrieve passwords, resulting in "Cannot retrieve password" for secured networks.

This script could be used for legitimate purposes, such as retrieving Wi-Fi credentials for networks you own or have authorized access to. Unauthorized attempts to access or retrieve network credentials can be illegal and unethical.

Let's move on to the next tool now.

Find Instagram Profile

This script uses the **Instaloader** library to retrieve and display information from an Instagram profile, such as the number of posts, followers, followers, and bio, and downloads the profile picture. Especially with this tool, we can use this tool as a social engineering method to collect data in the cyber security field.

This can be used as a social engineering method in this way.

Used for Profiling: Collecting data (like bios, followers, and photos) to build detailed personal profiles, potentially using this information to tailor persuasive or manipulative messages.

Pretending Familiarity: Information gathered might be used to craft convincing messages, impersonating someone familiar or exploiting knowledge of their interests.

Gaining Trust: By appearing informed about the target's interests or network, attackers might use the data to gain trust and push for specific actions.

Phishing or Impersonation: In a more malicious approach, someone might use downloaded images and public info to create fake accounts or support phishing schemes.

Now let's focus on this tool.

Figure 58 instagram checker

When I type the command number 6 in the terminal, **"Find Instagram Profile"** is run. Then I can enter the name of the person I want in the **"Enter User Name"** field. After a while, the person with that name is searched on Instagram, all the details of that person are displayed in the terminal, and a file containing all the records of that person is downloaded to our Linux Kali Machine. The above screenshot shows the person's details section displayed in the terminal. I used my own name for the example.

The yellow circle below shows the folder file containing the details of the person being downloaded. It is established in the name of that person. (Example iam_naviya)

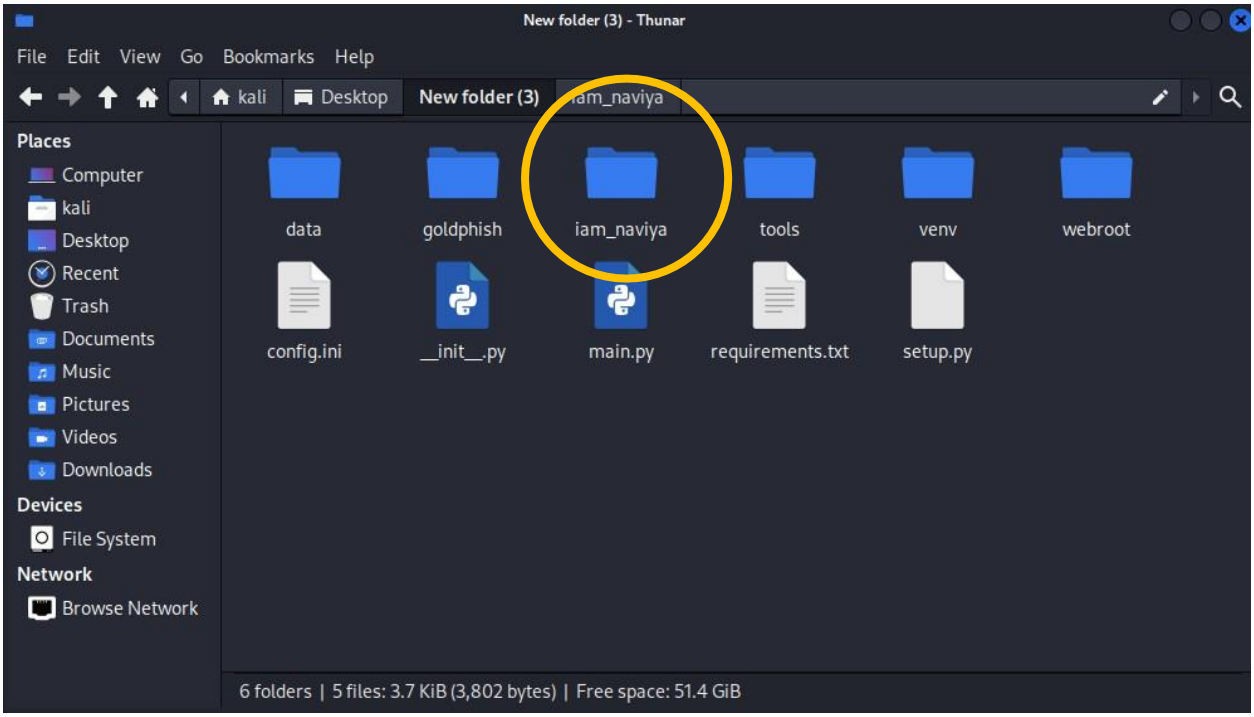

Figure 59 Downlodable File

Inside the folder file, 2 files containing that person's profile photo and details will be installed.

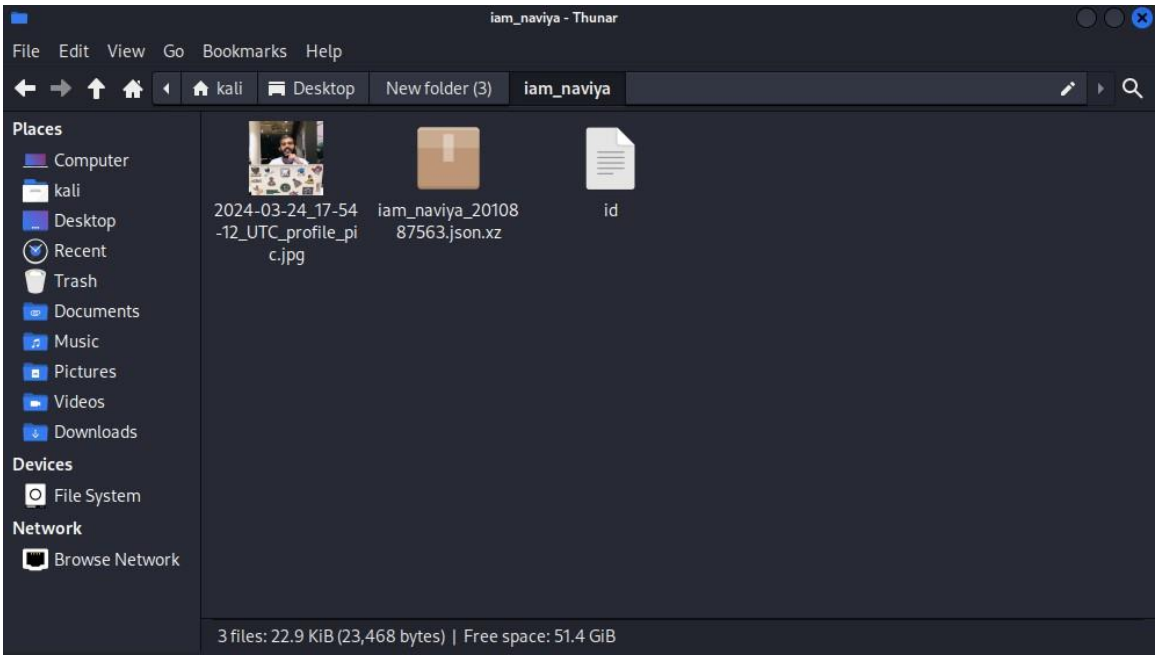

Figure 60

Specifically, this file contains the following:

- The script then prints the details about the profile:
- Username (profile.username)
- Number of posts (profile.mediacount)
- Number of followers (profile.followers)
- Number of followings (profile.followees)
- Biography (profile.biography)

Important Note

Authentication: Some Instagram data may require login authentication for access. If Instaloader prompts for login, it will need valid Instagram credentials.

Ethical Use: This code should only be used for profiles you have permission to access, as unauthorized access or data scraping is against Instagram's terms of service.

It is very important to use everything for educational purposes only.

Now let's move on to the next tool.

Cookies information

To run the "Cookies information" tool, you must first enter the number 7 on the terminal.

Pay attention to the screenshot below.

Figure 61 cookies info

After giving the command number 7, you will see an output as "**Enter The URL**". You need to give it a website of your choice.

The URL I have provided here is https://hackthissite.org.

The output when given is

Cookie: ***HackThisSite***

Value: ***esf5ofeue2d2b0ec0pdhk687m6.***

The output Cookie: ***HackThisSite*** with Value: ***esf5ofeue2d2b0ec0pdhk687m6*** suggests that the website has set a cookie that might be used to track session or user information.

Potential Importance of This Cookie

Session Management:

The cookie may contain a session identifier, which is a unique token associated with your browsing session. Websites use session cookies to remember users as they navigate between pages, store temporary preferences, or keep users logged in.

If this is a session cookie, it's important because anyone with access to it could potentially impersonate the session, depending on the site's security and the protection it places around cookies (like encryption, HTTP-only, and Secure flags).

Authentication or Authorization:

Cookies like this can sometimes store authentication tokens or identifiers that grant access to certain parts of a website or to specific user data. This could be important if the cookie is linked to permissions within the site.

Analyzing cookies in security testing can reveal whether sensitive authentication information is being sent or stored securely.

Tracking and Personalization:

Sometimes, cookies store unique user IDs or preferences, which help the website personalize the experience or track usage behavior. This cookie might be used by the site to track your interactions anonymously or as part of a testing mechanism to differentiate between users.

Vulnerability Assessment:

If the cookie value can be easily guessed or modified, it could reveal security vulnerabilities. For example, if there's no server-side verification of the cookie, an attacker could try manipulating the cookie value to gain unauthorized access or impersonate other users.

Testing for vulnerabilities like cross-site scripting (XSS) or cross-site request forgery (CSRF) can also involve cookies, as these attacks may exploit improperly handled cookies.

Let's look at some more examples.

Figure 62 another site check

This is a URL for one of the main political parties in Sri Lanka. It is listed as **PHPSESSID**.

The **PHPSESSID cookie** is commonly used by PHP-based websites to track user sessions. Here's a look at the implications of finding this session cookie and what it might mean to a security professional analyzing it.

Significance of PHPSESSID

Session Management:

The PHPSESSID cookie is usually set when a PHP application initializes a new session. This session ID is a unique identifier that links your browser session to the server, allowing the server to remember information (like login status or user preferences) across page loads.

If this is a session token, then the server associates it with a specific session on the backend. This could mean that whoever possesses this cookie might gain access to the session associated with it if proper security controls aren't in place.

Session Hijacking Risks:

If this cookie is accessible to an unauthorized person, it could potentially lead to session hijacking. In session hijacking, an attacker can impersonate the legitimate user if the server does not have additional verification mechanisms, like IP binding or multi-factor authentication.

By copying this cookie and adding it to a browser, an attacker might gain unauthorized access if the website relies solely on this cookie for authentication.

Testing for Security Flags:

The PHPSESSID cookie should ideally be secured with the HttpOnly and Secure flags:

HttpOnly: This prevents the cookie from being accessible to JavaScript, reducing the risk of theft through cross-site scripting (XSS).

Secure: This ensures that the cookie is only transmitted over HTTPS, protecting it from interception on insecure networks.

Without these flags, an attacker could exploit this cookie more easily.

Session Expiration and Security:

In a secure environment, session cookies like PHPSESSID should have limited lifespans. They should expire when the user logs out or after a period of inactivity to limit their usefulness if intercepted.

You could check if the session persists across logins or remains valid for an extended period, which would indicate weaker session security.

Finally, The PHPSESSID cookie links a session to the user on a PHP-based site. If the cookie is unsecured or improperly managed, it can be vulnerable to hijacking or unauthorized access. Ethical hackers use such

cookies in authorized tests to assess the security of session management systems and recommend improvements.

Let's look at another example.

Figure 63 another test

The URL I included here is the website address of the strongest and oldest political party in Sri Lanka. https://unp.lk. The **ci_session** cookie is typically used by web applications built with the CodeIgniter framework to manage sessions.

Like the **PHPSESSID** cookie in PHP, the **ci_session** cookie in CodeIgniter is usually used to manage user sessions by storing a unique session identifier.

This unique identifier links a user's browser to session data on the server, such as login status, preferences, or activity history. If an attacker obtains this session cookie, they could potentially impersonate the user, depending on how the server handles session verification.

Important Note on Ethical Usage

Testing live websites for session vulnerabilities, including intercepting or analyzing cookies, should only be done with explicit permission. Unauthorized testing is illegal and unethical, as it violates the privacy and security of the users and the organization.

Now let's move on to our next tool.

Scan website Admin Page

This Python script is a basic "**Scan website Admin Page**" tool often used in security testing. Its main purpose is to scan a website for hidden or administrative pages (like /admin, /login, etc.) by appending paths from a wordlist to the URL and testing each for a valid response. This kind of tool can be helpful in web security to assess the exposure of admin panels.

To run this tool, type command number 8 in the Terminal.

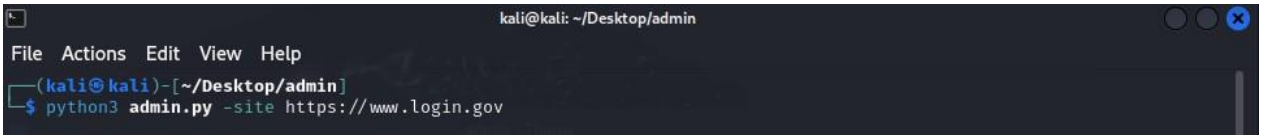

Figure 64 find web admin

Here you need to enter the desired website.

Figure 65 Result

This will then be scanned from all the login details in the wordlist and the correct one will be shown to you to log in. In this example, only **admin and admin/** are successful.

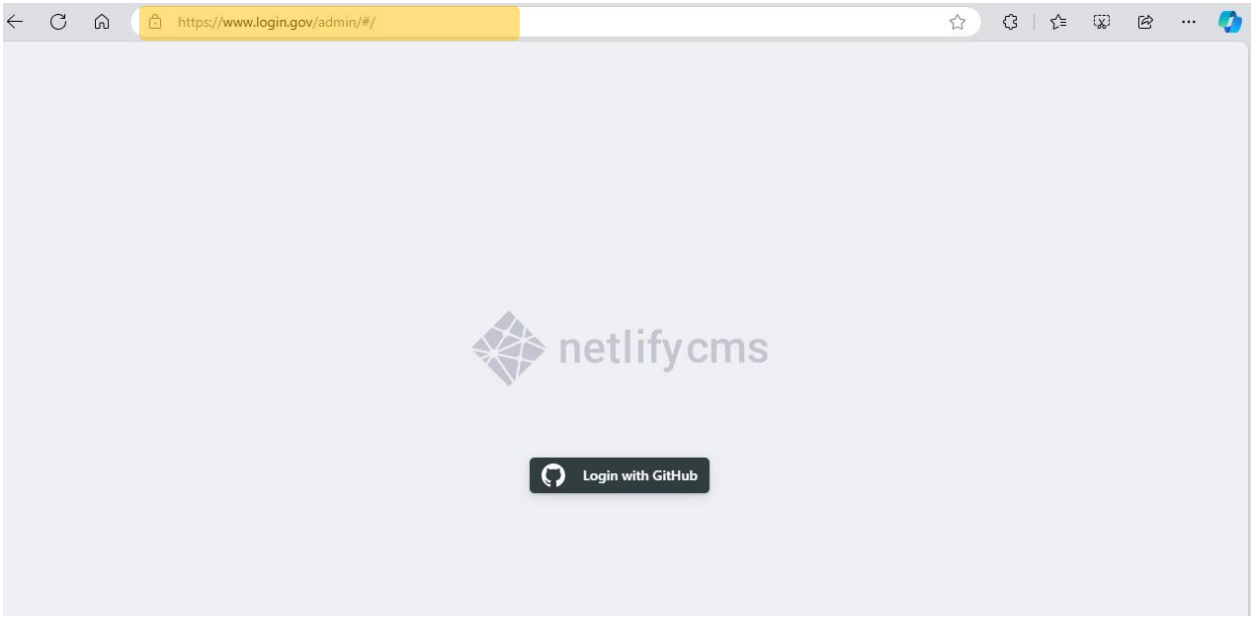

Figure 66 admin panel

Copy that **"admin"** and add it to the previous URL link, and when browsing, it will appear like this. Look at the yellow bar.

This admin panel finder script offers several advantages, particularly in the context of web security, ethical hacking, and penetration testing.

1. **Efficient Scanning with Automation**

- By automating the process of checking for potential admin pages, the script can scan through hundreds or thousands of URLs quickly, saving time compared to manual searches.
- The use of a queue and multithreading allows multiple URLs to be checked simultaneously, which speeds up the process and makes it efficient.

2. **Proxy Support for Anonymity and Flexibility**

- Proxy support enables users to mask their IP address, which is useful for anonymous scanning or bypassing IP-based restrictions.
- Proxy usage also allows testers to simulate requests from different geographic locations, testing the site's responses under varying conditions.

3. **Customization with Wordlists and Delays**

- The script allows users to specify a custom wordlist, enabling tailored scans for different types of sites (e.g., e-commerce, CMS-based sites) where admin paths might differ.
- Adding a time delay between requests reduces the likelihood of triggering rate limits or being detected, which is beneficial for remaining undetected during authorized tests.

4. **Error Handling for Robustness**

- With built-in error handling, the script can gracefully handle issues such as connection errors, missing URLs, or schema errors (like forgetting http:// or https://), making it more robust in various network conditions.

5. **Educational Value for Security Enthusiasts**

- The script serves as a learning tool for beginners in cybersecurity, illustrating concepts like HTTP requests, threading, proxies, and handling command-line arguments.
- It provides insight into how malicious actors might try to identify hidden admin pages, giving security teams a better understanding of potential vulnerabilities.

6. **Support for Multiple Target Websites**

- Users can specify multiple sites to scan in one go, which can be helpful in penetration testing, where an organization might own multiple domains, all needing assessment.
- This feature allows security teams to apply the same level of scrutiny across their entire web portfolio.

7. **Applicable in Web Application Security Testing**

- The tool can be helpful for ethical hackers, penetration testers, and security analysts during assessments to ensure no critical, hidden pages are accessible to unauthorized users.
- It can be integrated as part of a larger security testing process to identify potential entry points for more detailed examination.

Let's now focus on the next tool.

Find All IP Address

As before, command **number 9** should be used to execute this.

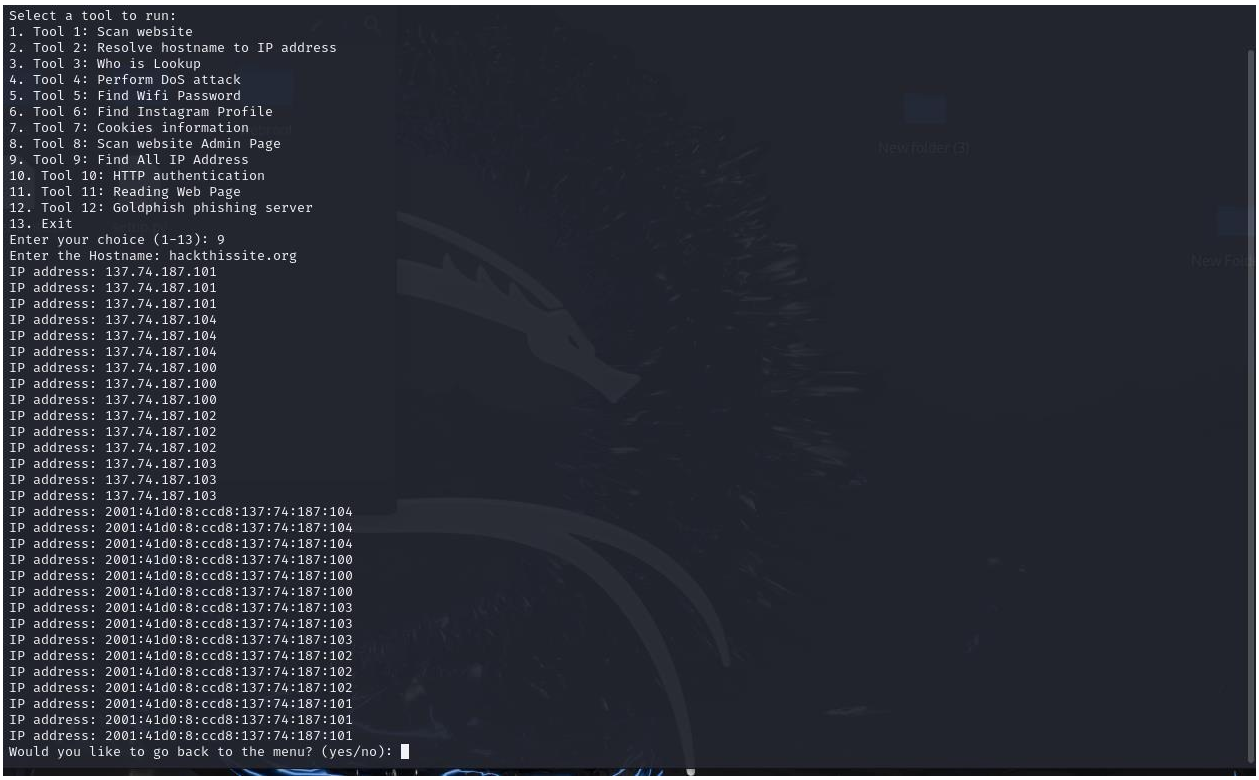

Figure 67 Ip address

This Find All IP Address tool is designed to find and display all IP addresses associated with a given hostname, such as **hackthissite.org**. I have also used Python's socket library in the script to get a list of IP addresses for a specific hostname.

If you look at the example, the number of IP addresses that the **URL hackthissite.org** has is visible in the screenshot.

Useful in Networking: This script helps retrieve multiple IP addresses associated with a hostname, which is valuable for troubleshooting network issues, understanding DNS configurations, and analyzing content delivery networks (CDNs).

Simple and Direct: It provides straightforward access to IP addresses without needing complex configurations or tools, making it handy for quick lookups.

This IP address lookup script is advantageous in the cybersecurity field for several key reasons, particularly in the areas of network reconnaissance, threat analysis, and vulnerability management.

1. Reconnaissance and Footprinting

- Understanding Infrastructure: During the initial stages of a cybersecurity assessment or penetration test, gathering information about a target's infrastructure is crucial. This script helps security analysts discover all IP addresses associated with a hostname, which might reveal multiple servers, load balancers, or services.
- Mapping Attack Surface: Knowing all IP addresses related to a hostname helps identify the scope of potential attack vectors. For example, some IP addresses may expose web servers, while others

might expose other types of services like databases or APIs, each of which could have unique vulnerabilities.

2. Detecting Potential Security Risks with Multiple IP Addresses

- Identifying Misconfigurations: When a domain resolves to multiple IPs, sometimes different servers may have inconsistent configurations. This can help a security analyst spot servers that might be misconfigured, outdated, or inadvertently exposing sensitive information.
- Bypassing Firewalls or Filters: If certain IPs are firewalled or inaccessible due to location restrictions, knowing multiple IPs could reveal alternative points of entry that might be less protected or overlooked in security measures.

3. Protection Against DNS Spoofing and Cache Poisoning

- Cross-Checking Valid IPs: Attackers can manipulate DNS records (via spoofing or cache poisoning) to redirect traffic to malicious IP addresses. By repeatedly checking known IPs for a domain, security teams can verify whether the DNS records have been tampered with, helping detect these types of attacks.
- Verifying Legitimate IPs for Critical Domains: Organizations often need to know the legitimate IP addresses for their own domains. Using this script, they can verify the IPs they expect to see, ensuring that no suspicious addresses have been introduced.

4. DNS-Based Threat Intelligence

- Identifying Malicious IPs Associated with Suspicious Domains: Cybersecurity teams can use this tool to investigate suspicious domains and check if any of their IPs appear in threat intelligence databases. For instance, if a domain has multiple IP addresses, this tool helps identify which ones might already be flagged as malicious.
- Detecting Potential Indicators of Compromise (IoCs): Cybersecurity professionals can use the IP data gathered from suspicious domains to feed into monitoring systems. These indicators can then be tracked for any attempts to connect, helping detect possible security incidents early.

5. Enhanced Response in Incident Management

- Network Blocking: When an organization detects malicious activity from a domain, knowing all IP addresses associated with it allows them to apply blocking rules to the entire range, rather than blocking just one IP and missing others.
- Tracking Attack Vectors: For domains involved in attacks, knowing all associated IPs helps cybersecurity teams track where attacks might be originating from, improving understanding of an attack's structure and its geographical sources.

Let's move on to the next tool now.

HTTP authentication

To run this tool, type the number 10 in the terminal.

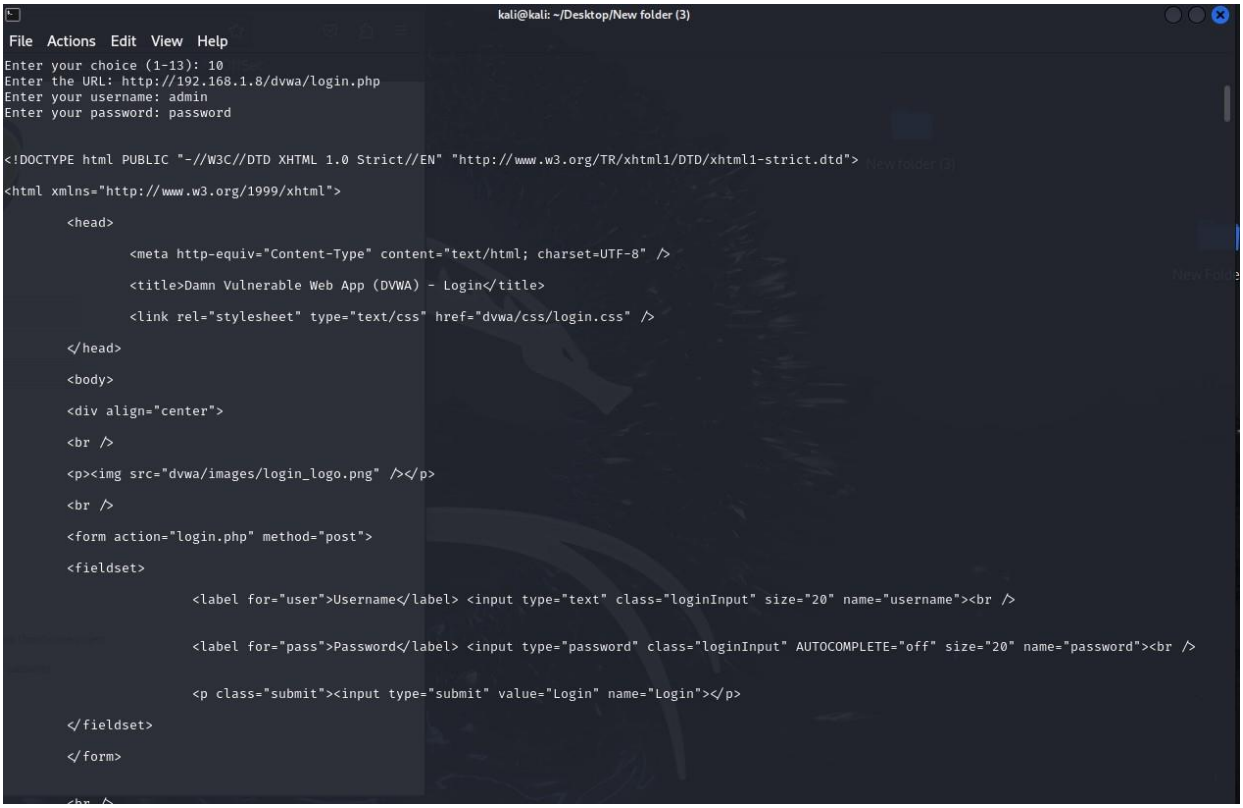

Figure 68 Http auth

Then you need to enter the URL of the output. In this example, I have entered the IP of the **dvwa**. You must enter your username and password to verify it.

This script is a Python tool that retrieves content from a web page that requires basic HTTP authentication. The script uses the urllib library to handle the HTTP request and encrypts the username and password in the format required for basic authentication.

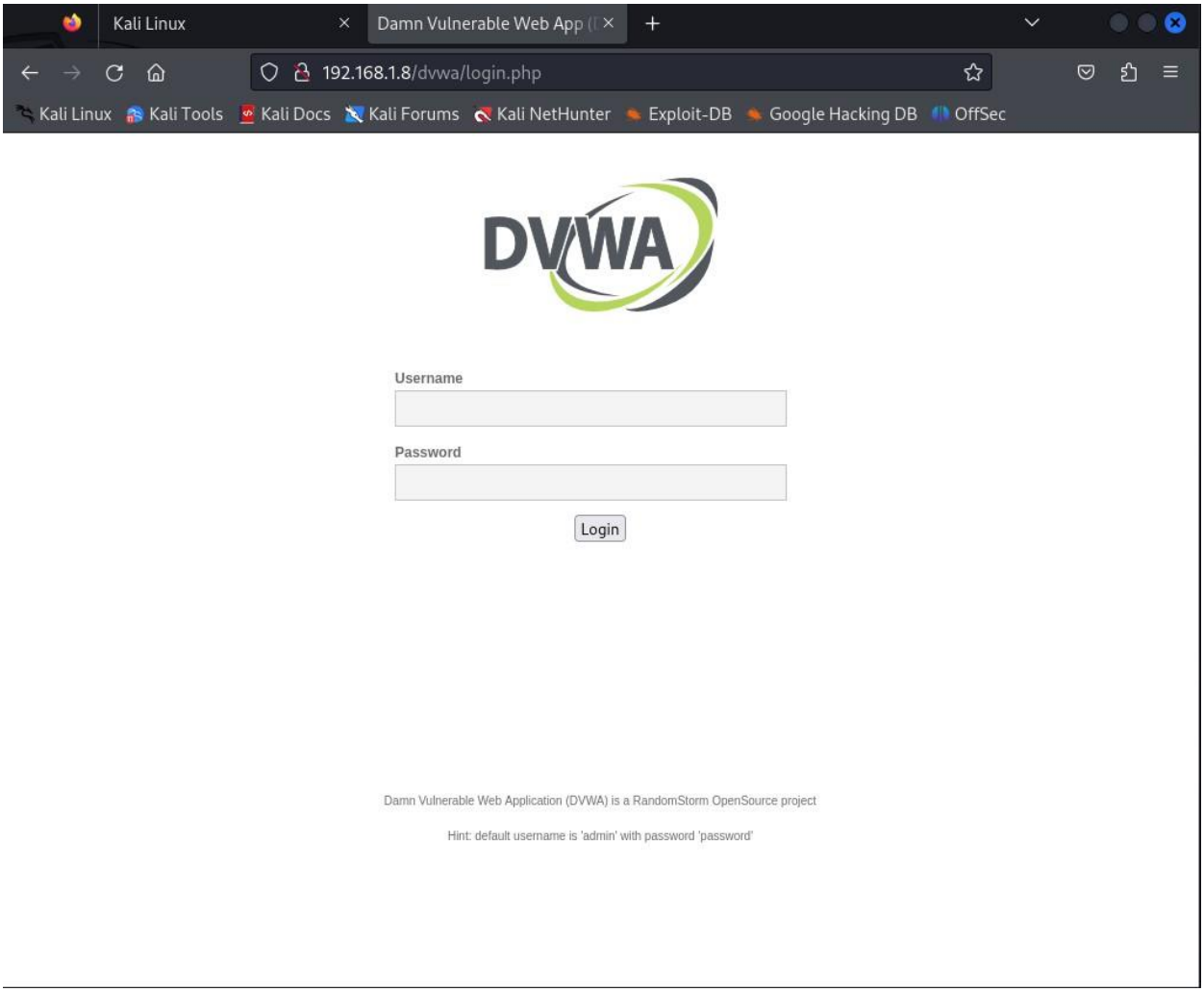

Figure 69 result site

I used this website as an example. Look closely, the complete coding diagram is shown in the first terminal.

Also pay attention to the example below.

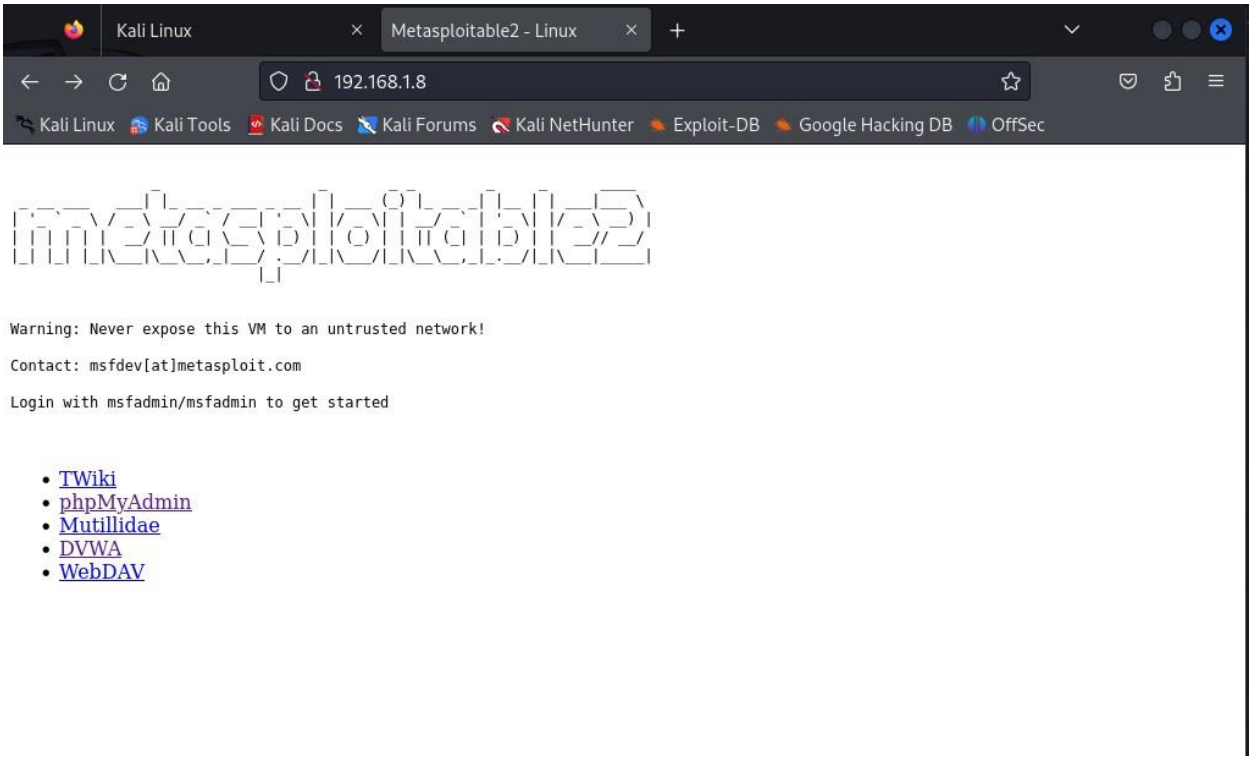

Figure 70 Another testing site

The server of metasploitable 2

Figure 71 Result

Let's visit a large website to focus more closely. Look at the example below.

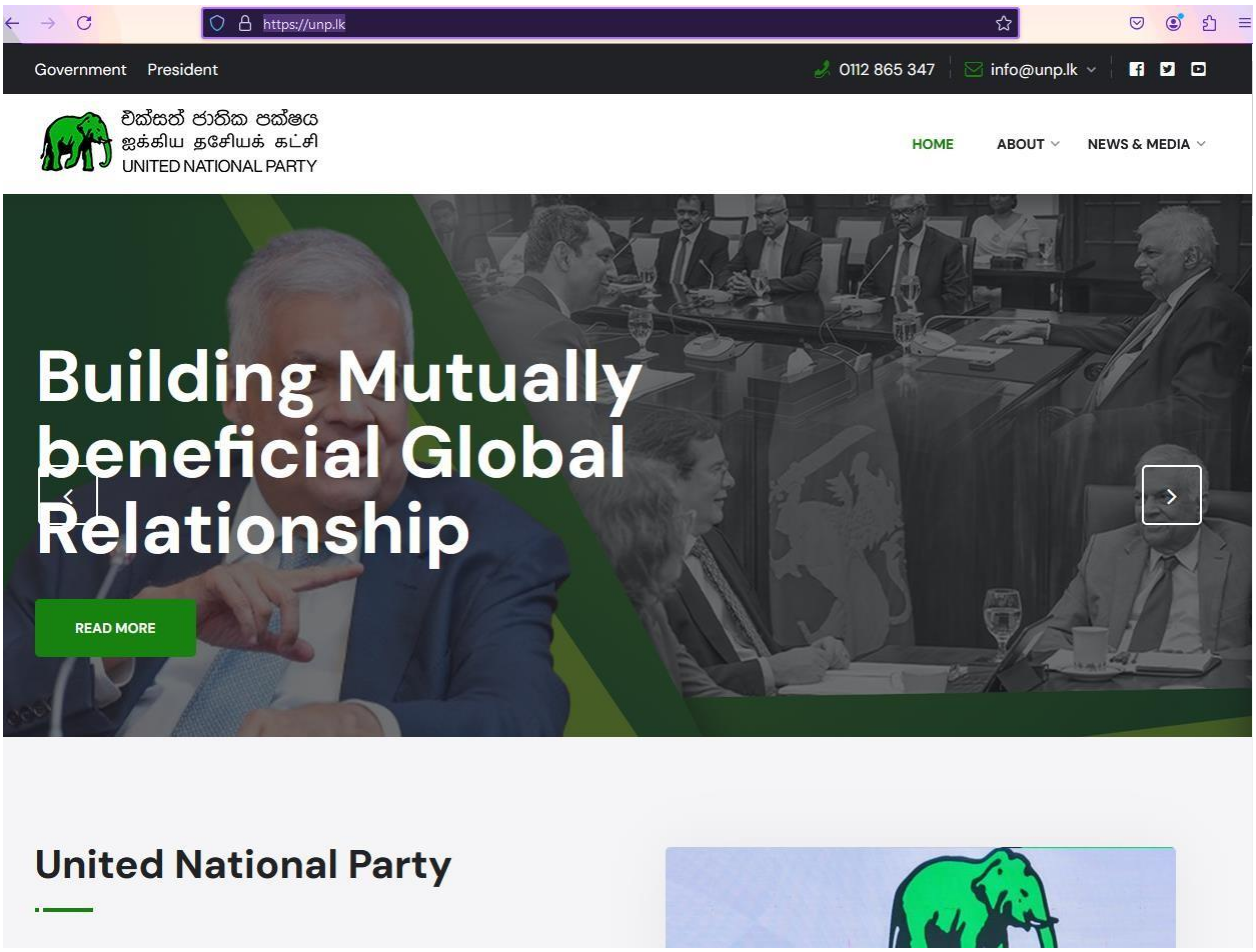

Figure 72 Another Site

Let's focus on the website of one of the strongest and oldest political parties in Sri Lanka. https://unp.lk

We can see the content of this page at a very large size. Look at the example below.

```
Enter the URL: https://unp.lk
Enter your username: admin
Enter your password: password
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0">
    <meta name="description" content="The United National Party often abbreviated as UNP, is a political party in Sri Lanka, currently headed by the Presi
dent Hon. Ranil Wickremesinghe. The UNP is considered as a center-right party, optin"/>
    <meta name="keywords" content="United,National,Party,-,Building,a,Stronger,Economy"/>
    <meta property="og:title" content="United National Party - Building a Stronger Economy"/>
    <meta property="og:description" content="The United National Party often abbreviated as UNP, is a political party in Sri Lanka, currently headed by th
e President Hon. Ranil Wickremesinghe. The UNP is considered as a center-right party, optin"/>
    <meta property="og:url" content="https://unp.lk"/>
    <meta property="og:type" content="website"/>
    <meta property="og:image" content="https://unp.lk/assets/main/images/site-overview.png"/>
    <title>United National Party - Building a Stronger Economy</title>

    <!-- Fav Icon -->
    <link rel="icon" href="assets/main/images/favicon.png" type="image/x-icon">

    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css2?family=DM+Sans:ital,wght@0,400;0,500;0,700;1,400;1,500;1,700&amp;display=swap" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css2?family=Merriweather+Sans:ital,wght@0,300;0,400;0,500;0,600;0,700;0,800;1,300;1,400;1,500;1,600;1,700;1,8
00&amp;display=swap" rel="stylesheet">

    <!-- Stylesheets -->
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/font-awesome-all.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/font-awesome-all.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/flaticon.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/owl.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/bootstrap.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/jquery.fancybox.min.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/animate.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/nice-select.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/color.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/style.css" />
    <link rel="stylesheet" href="https://unp.lk/assets/main/css/responsive.css" />

<!-- Global site tag (gtag.js) - Google Analytics -->
    <script async src="https://www.googletagmanager.com/gtag/js?id=G-ZDL9T62ZYC"></script>
    <script>
      window.dataLayer = window.dataLayer || [];
      function gtag(){dataLayer.push(arguments);}
      gtag('js', new Date());

      gtag('config', 'G-ZDL9T62ZYC');
    </script>
</head>
```

Figure 73 Result

The content is displayed properly in the terminal.

If we talk more about this tool,

Accessing Restricted Resources: This script can help automate access to restricted areas during penetration testing when provided with valid credentials.

Testing Credentials: It's helpful in validating that a server's Basic Authentication is working correctly and handling errors properly.

Reconnaissance: In a controlled environment, this tool can verify access configurations or perform enumerations on web directories that require authentication.

Let's now move on to exploring the next tool.

Reading Web Page

To implement this, number 11 must be executed in the terminal.

I created this script to retrieve and **display the HTML content of a specific web page**. It was created using Python's **urllib** library to make an HTTP request to the given URL and handle various types of errors that may occur in the process.

I used the **unp.lk** website for this example.

Figure 74 Result

Purpose and Advantages

Web Content Retrieval: This script can fetch HTML content from a given URL, which is useful for tasks like web scraping or content verification.

Error Handling for Stability: Built-in error handling ensures that the script handles different scenarios gracefully, making it resilient against invalid URLs, connection issues, and HTTP errors.

Automated Testing: It's helpful in testing URL availability or verifying web application responses without needing to visit the URL manually.

Controlled User-Agent: Setting a custom User-Agent allows it to bypass some restrictions that prevent automated access, making it more reliable for accessing certain websites.

In cybersecurity, this script could be used for URL enumeration, link validation, and gathering HTML content for deeper analysis, particularly in controlled environments.

Since **"HTTP authentication"** and **"Reading Web Pages"** seem to be the same, a brief explanation is provided below.

Purpose

HTTP authentication

- Specifically designed for accessing web pages that require Basic Authentication.
- This is often useful in scenarios where web resources are restricted and require a username and password for access.

Reading Web Pages

- Meant for general web page fetching without requiring authentication.
- The focus here is on retrieving public HTML content and handling errors related to HTTP requests.

Authentication Handling

HTTP authentication

- Handles HTTP Basic Authentication by encoding the username and password in a Base64 format and including it in the Authorization header.
- This makes it useful for accessing restricted content that requires user credentials.

Reading Web Pages

- Does not include authentication. Instead, it simply sends a request to the specified URL, suitable for fetching publicly available content.

Error Handling

HTTP authentication

- Catches HTTP errors, particularly related to failed authentication attempts (like 401 Unauthorized), and other common exceptions.

Reading Web Pages

- Includes additional error handling with URLError to handle malformed URLs and incorrect URL schemes, along with HTTP error handling.

Applications in Cybersecurity

HTTP authentication

- Useful for penetration testing and enumeration of protected resources when valid credentials are known. It's valuable for testing access controls and authentication robustness.

Reading Web Pages

- Useful for publicly accessible URL enumeration and content validation. It can be a quick way to verify if a URL is up, perform reconnaissance on open web pages, and gather content for analysis.

Let's now focus on the next tool.

Goldphish phishing server

To run this tool, you must first enter the number 12 in the terminal.

Goldphish is an http phishing server that clones a website, modifies it, and captures the POST request to potentially obtain credentials from it.

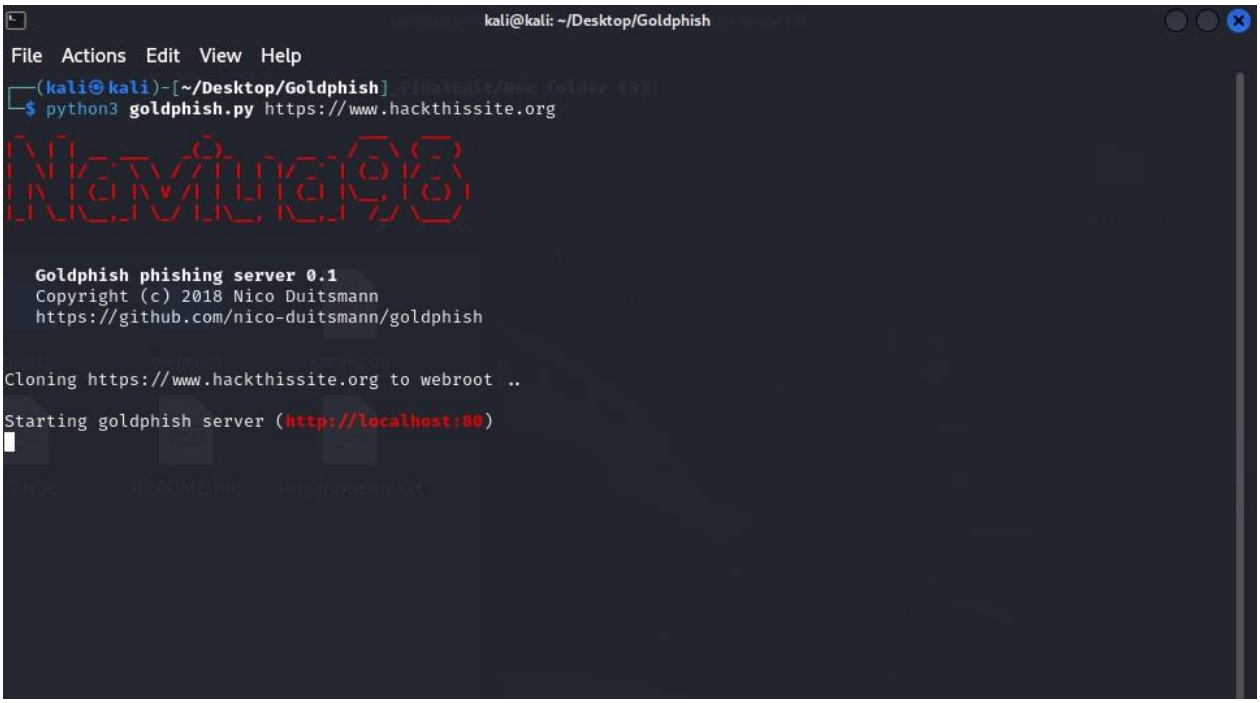

Figure 75 clone site

When we give this tool the website we want to clone, it clones it and installs it on our machine in a few minutes. This is especially easy to use for things like social engineering/phishing.

The screenshot above shows how the clone is started.

When the clone site is complete, all files will be installed in the webroot folder.

Look at the yellow circle in the example below.

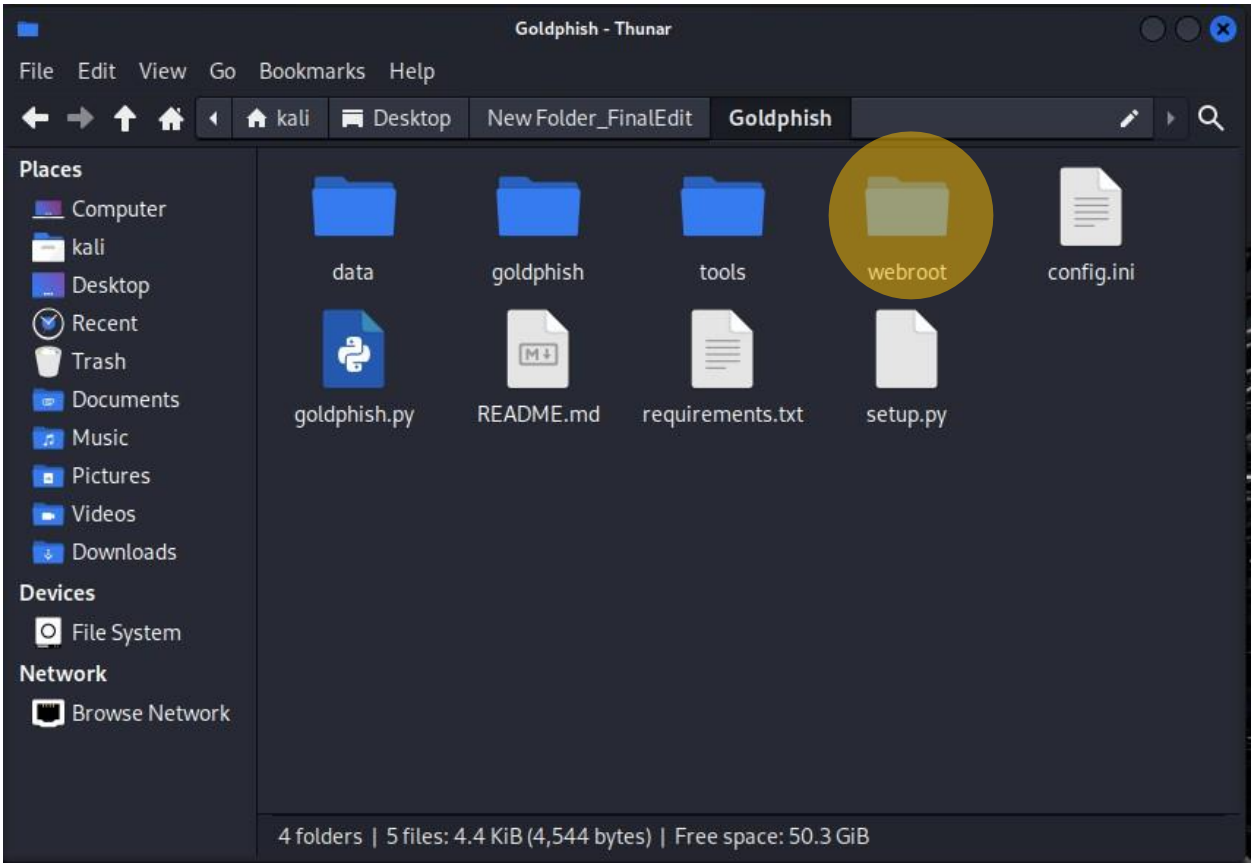

Figure 76 Download clone site

All html, css, js, php files, as well as videos and images on the website, are cloned and installed under this.

See the example below.

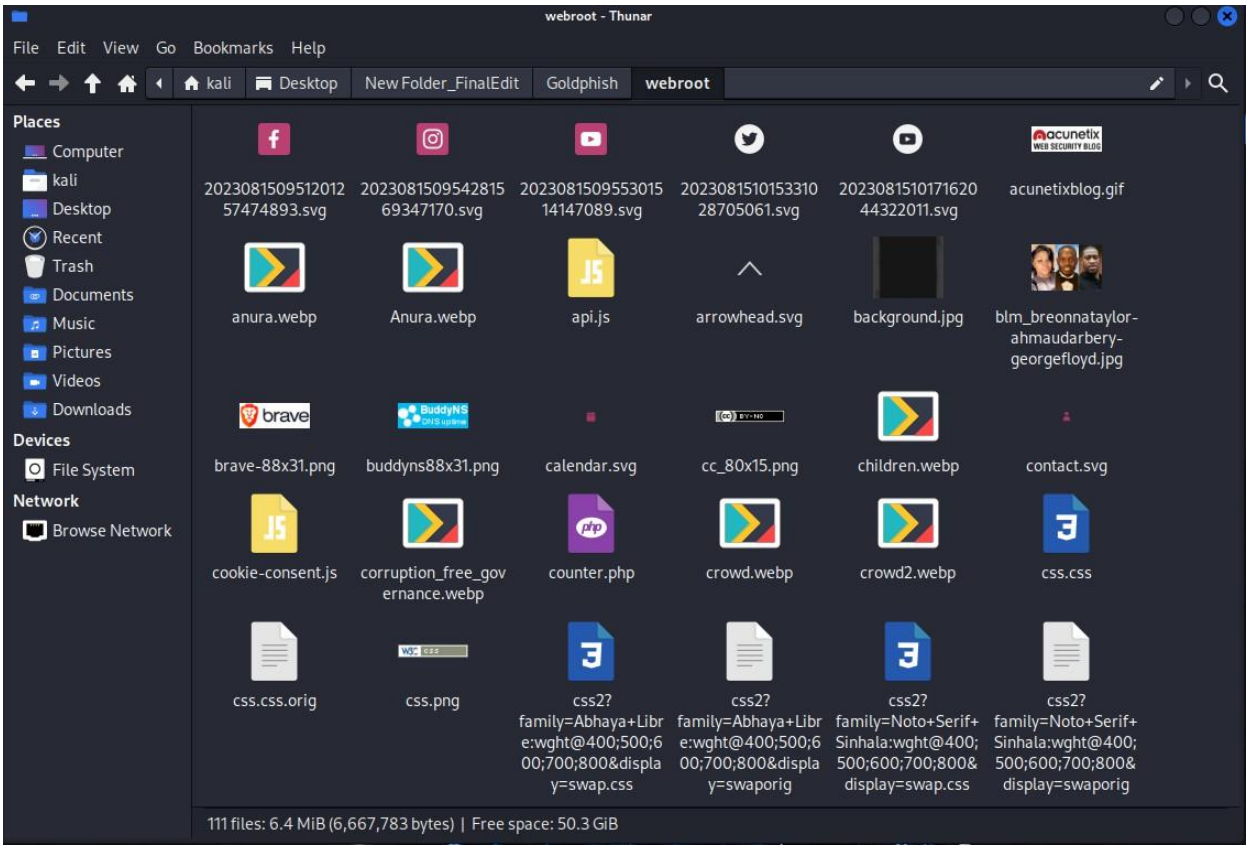

Figure 77 Downloadble files

This is what the cloned site looks like when opened. It is in good condition without any damage.

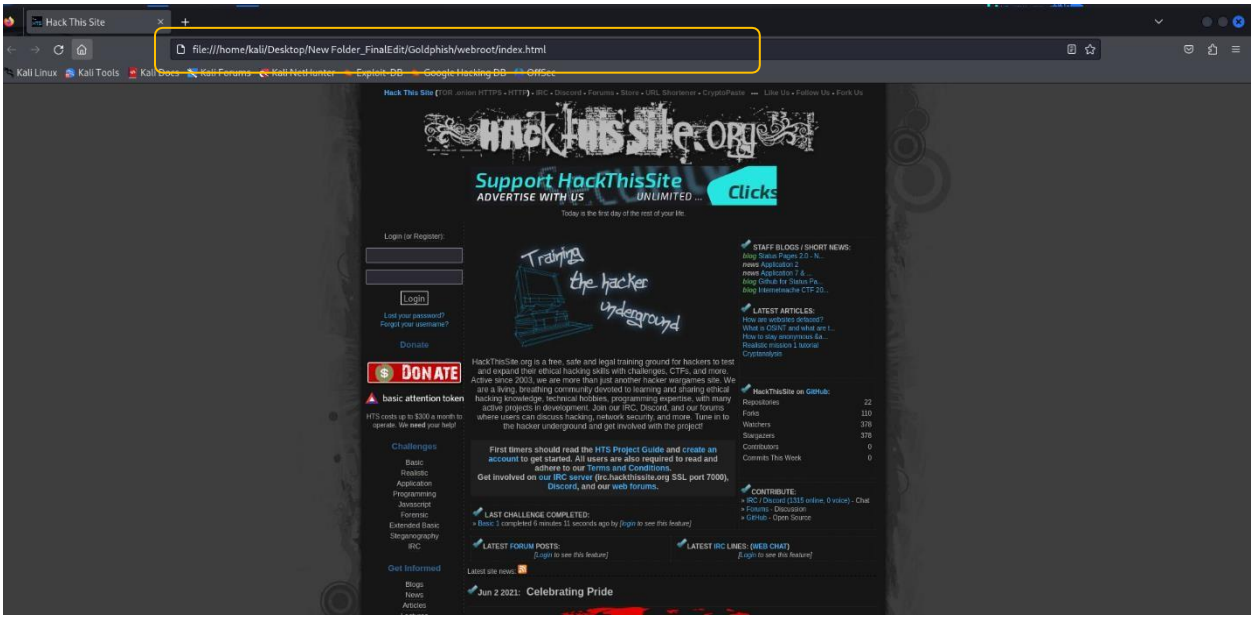

Figure 78 Clone site

Look at the yellow circle. The file is shown on the local server.

Now let's focus on the next tool.

Change MAC address

This script is a Python tool designed to change the MAC address of a network interface (in this case, eth0). It uses a series of shell commands to bring the network interface down, change its MAC address, and bring it back up. Once these steps are complete, I have set up a progress bar to show that the process is in progress, simulating a long process. This is actually a Linux command-line tool that was developed using "subprocess."

Below are examples of how to do this. Pay attention to them.

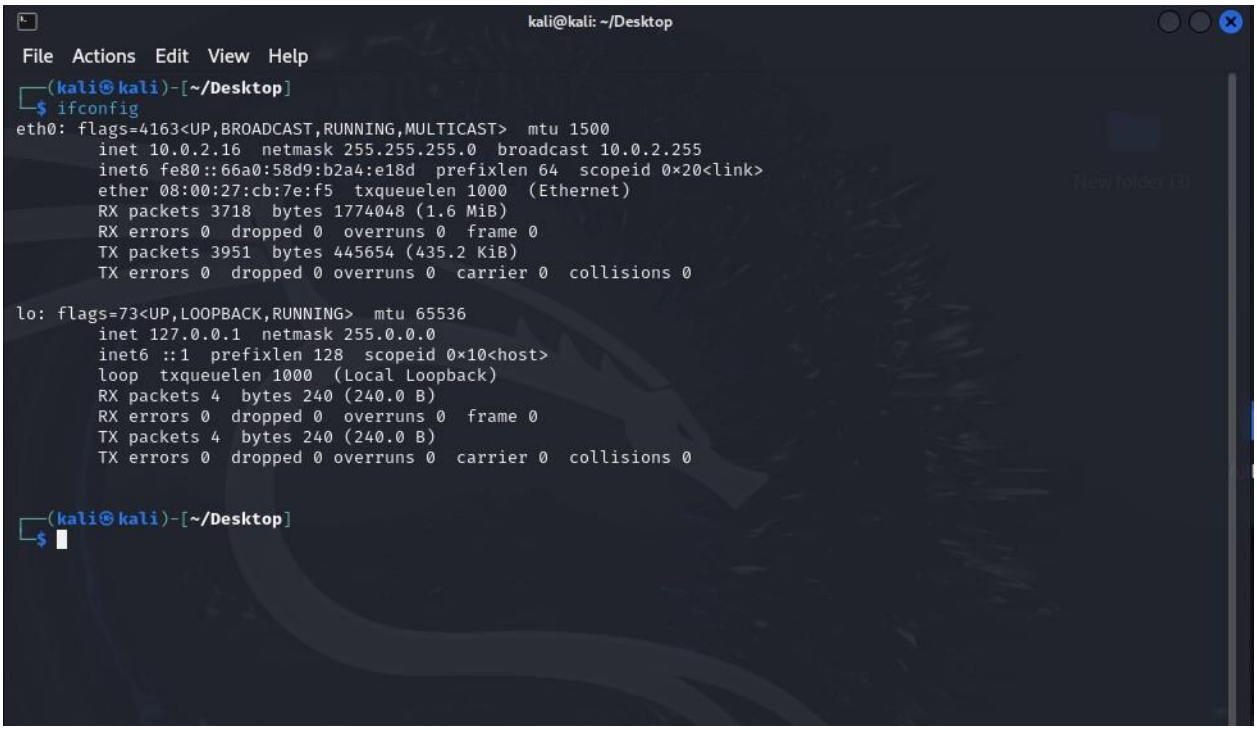

Figure 79 MAC Changer

Before changing the MAC address, pay attention to the actual MAC address, which can be done using the **ifconfig** command.

In my screenshot, the MAC address is **08:00:27:cb:7e:f5**. Now let's change this logically.

Figure 80 proccessing part

You can easily change the MAC address using my "**Change MAC address**" tool by following these steps. That is, by running the **Python3 Mac01.py** file, the Auto MAC address will be changed.

Let's type the **ifconfig** command again in the terminal and see if the MAC address has changed.

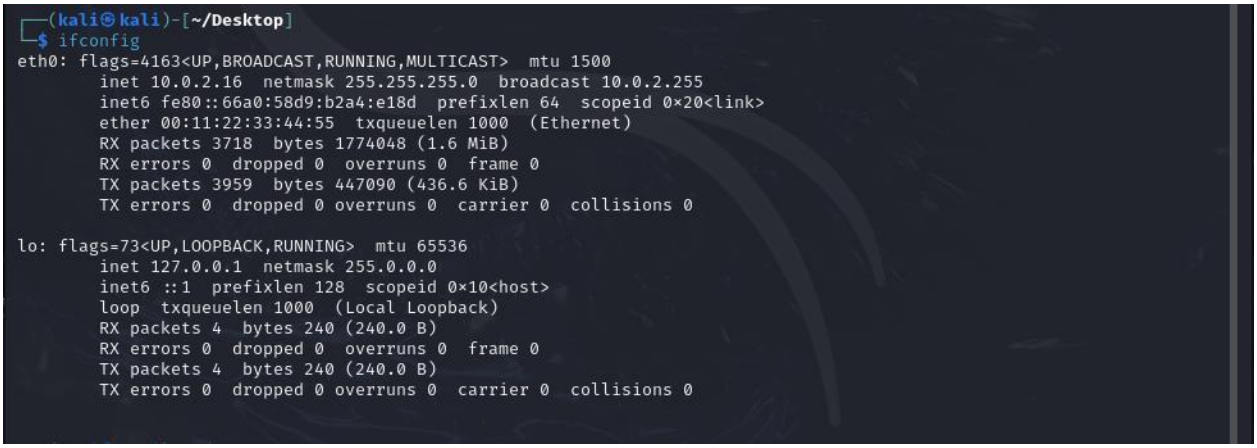

Figure 81 Changed MAC

Our MAC address has changed as seen in the terminal. Previously, it was **"08:00:27:cb:7e:f5"** and now it is **"00:11:22:33:44:55".** We have changed it.

Purpose of the Tool

The primary purpose of this script is to programmatically change the MAC address of a network interface. This type of tool is often used in cybersecurity and networking for purposes such as:

Network Anonymization: Changing the MAC address helps mask device identity on a network, which is useful in penetration testing.

Network Security Testing: Testing whether a network restricts access based on MAC addresses.

Device Emulation: Emulating other devices by temporarily adopting their MAC addresses.

Now let's focus on the next toolset. This is a second type of toolset. The first tool here is the cctv hacking python tool. Let's focus on that now.

CCTV Hacking

This tool has the ability to access any CCTV in the world. It was created using Python. The code was written by scraping a website (insecam.org) for open, unprotected **CCTV** camera feeds from various countries.

The insecam.org website is a very dangerous website and was first reported in 2014. Legal action was taken saying that this is a website that damages the privacy of all countries and people. Although it was popular in the **darkweb** during 2012/2016, it has now resurfaced, but various people on **Reddit** have commented that not everyone has access.

Currently, the CCTV hacking tools found on Github are based on websites called Shodan, ZoomEye, Censys. But since they operate very ethically and according to the rules, CCTV hacking tools cannot be classified as powerful tools. But this is not the case. This has developed very strongly due to the insecam website.

If we pay attention to other sites,

Shodan: Shodan is a search engine for internet-connected devices, and it indexes everything from servers to webcams to IoT devices. While it's widely used in cybersecurity to find and analyze devices, it requires some technical knowledge. Unlike Insecam, Shodan doesn't present video feeds directly; it indexes IP addresses of devices with open ports and weak security settings. Accessing these devices without permission is, however, illegal.

ZoomEye: Similar to Shodan, ZoomEye is a Chinese-based search engine that scans for internet-connected devices and can find IP cameras, among many other types of devices. It provides information for cybersecurity professionals, but it's also possible to encounter exposed, unsecured cameras through the search. ZoomEye requires registration, and some features are behind a paywall.

Censys: Censys is another platform used in cybersecurity research to scan and index internet-connected devices. It is a robust tool for understanding the global exposure of various devices and services. Although Censys itself does not promote unsecured camera feeds, it can be used to locate vulnerable devices if a user knows what to look for.

Before implementing this tool, we should also look at the legal side of it.

Legal and Ethical Implications

While these tools and sites have legitimate uses in cybersecurity for identifying security risks and vulnerabilities, accessing private or unsecured cameras without authorization is almost always illegal. This practice can be a violation of the Computer Fraud and Abuse Act (CFAA) in the United States, or equivalent laws in other countries, as well as a breach of privacy laws.

First, run the tool as before using the command "**python3 cctvhack.py**".

Then the tool will appear to you like this.

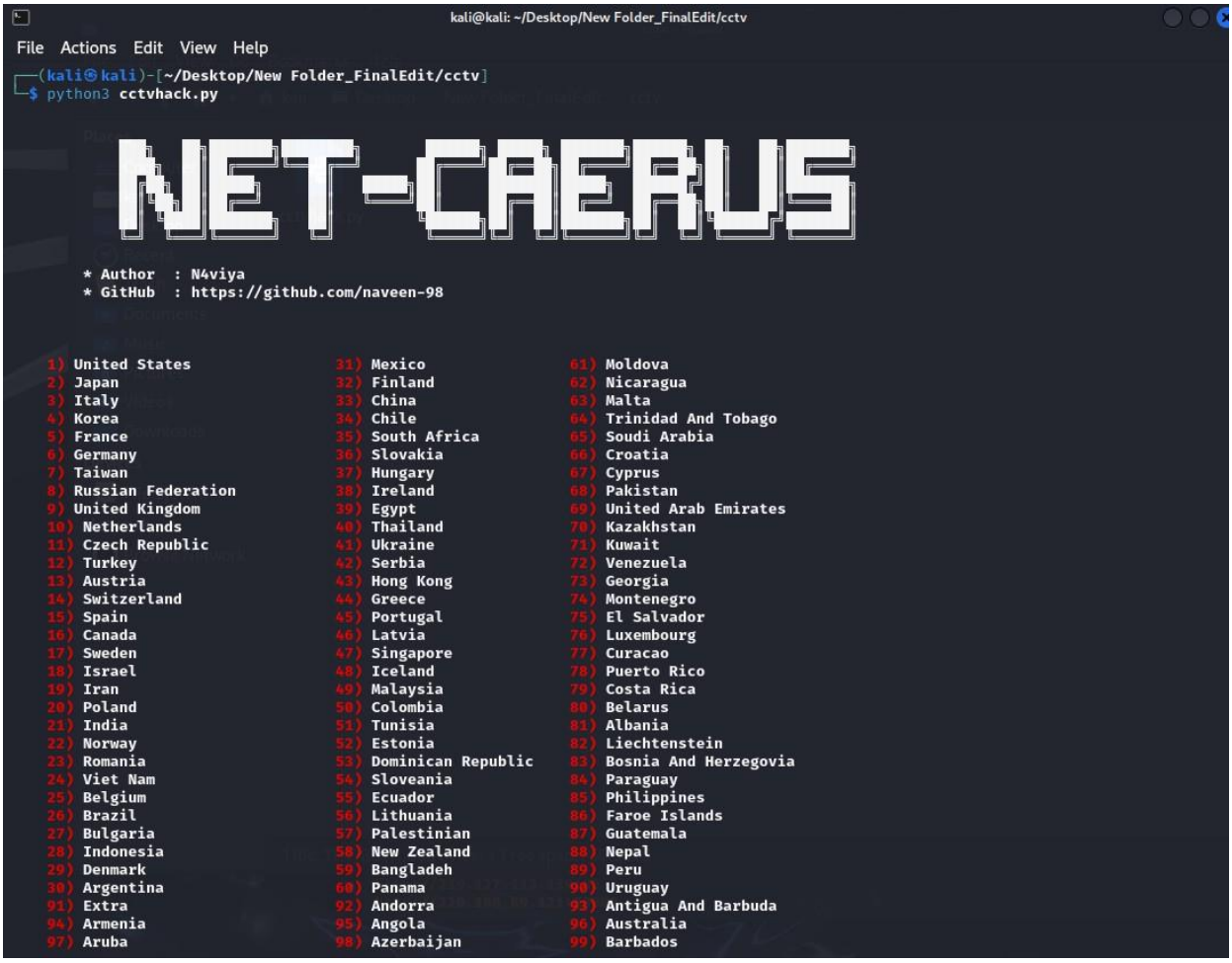

Figure 82 CCTV Hack

Now you can select a country of your choice and type in the corresponding number below.

Accordingly, if you select USA, you can enter the number 1. Then you will see the long IP address links appear in the terminal.

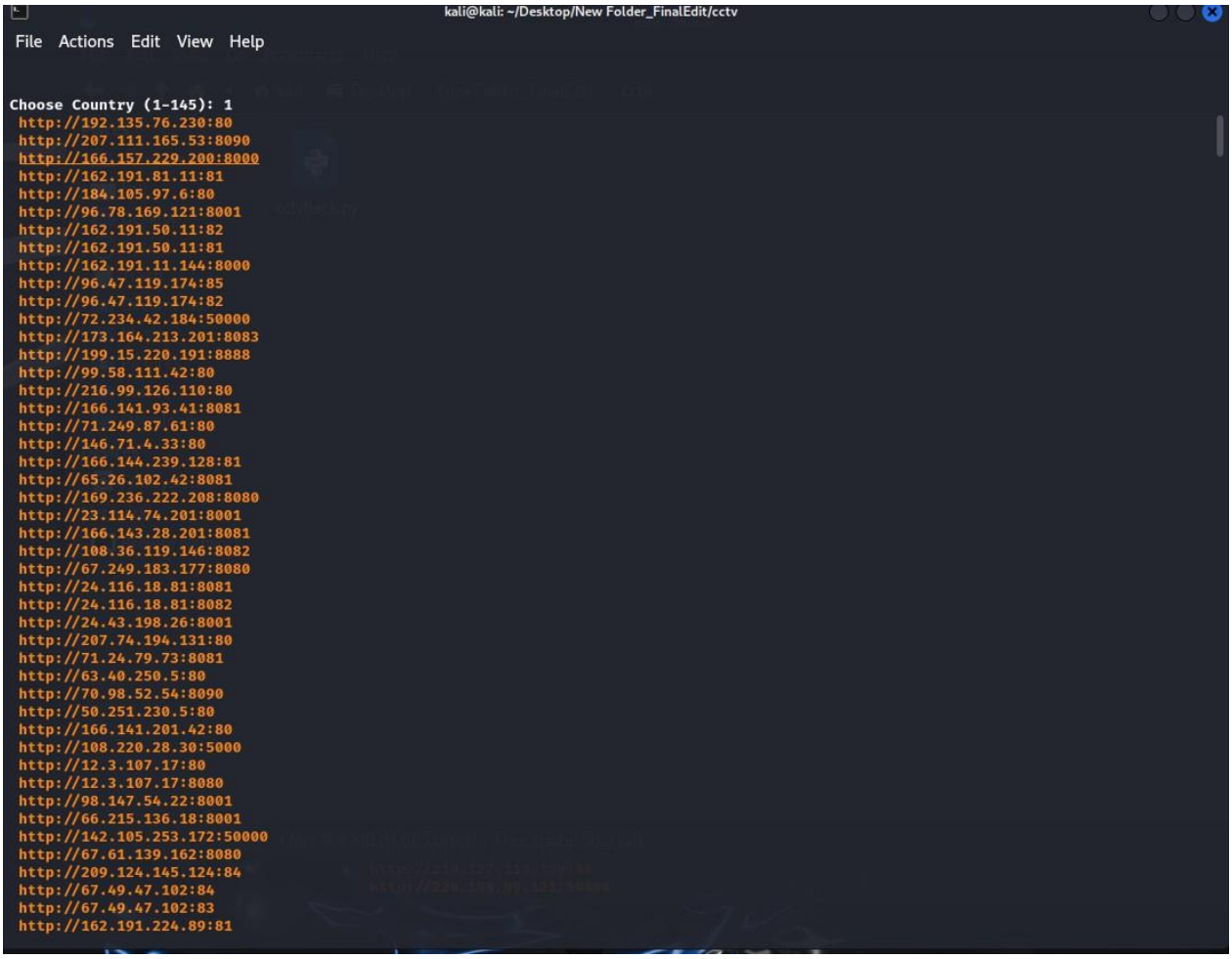

Figure 83 Public CCTV IPs

You can watch live USA CCTV cameras by opening any of the IP addresses above.

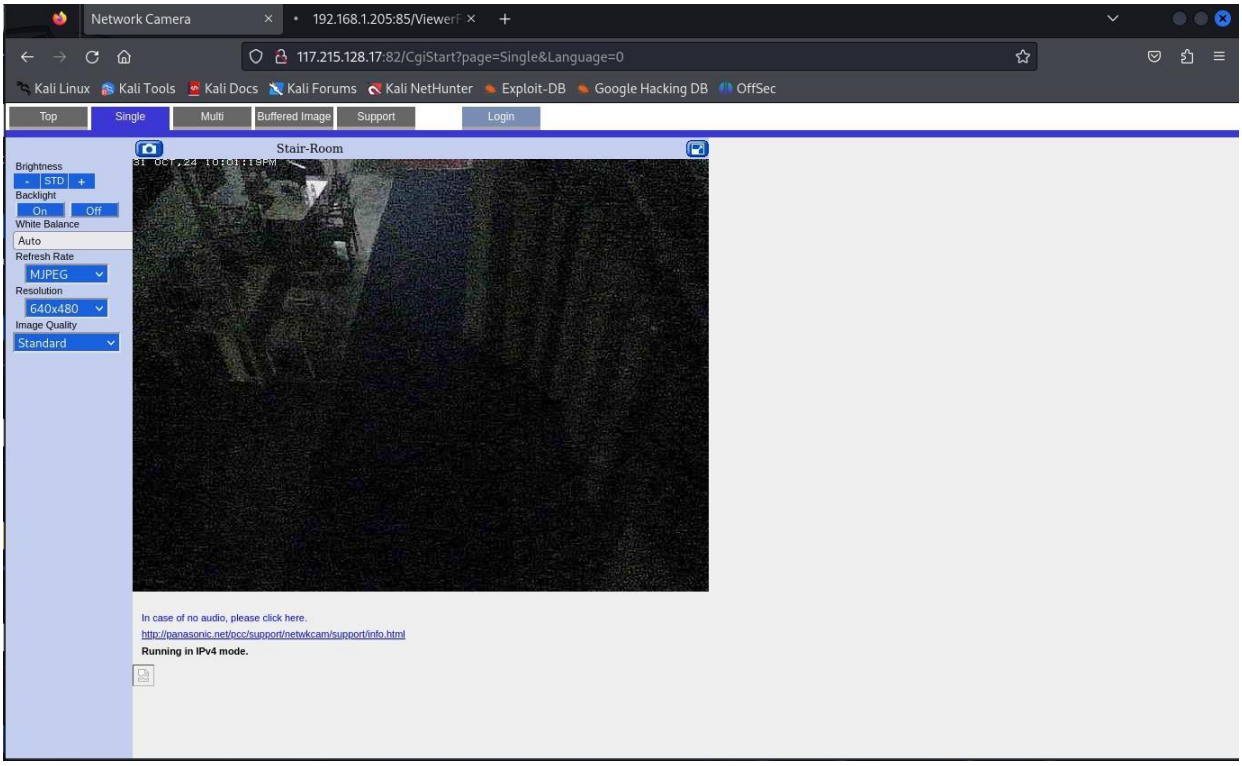

Figure 84 Live View

These are such CCTV live views.

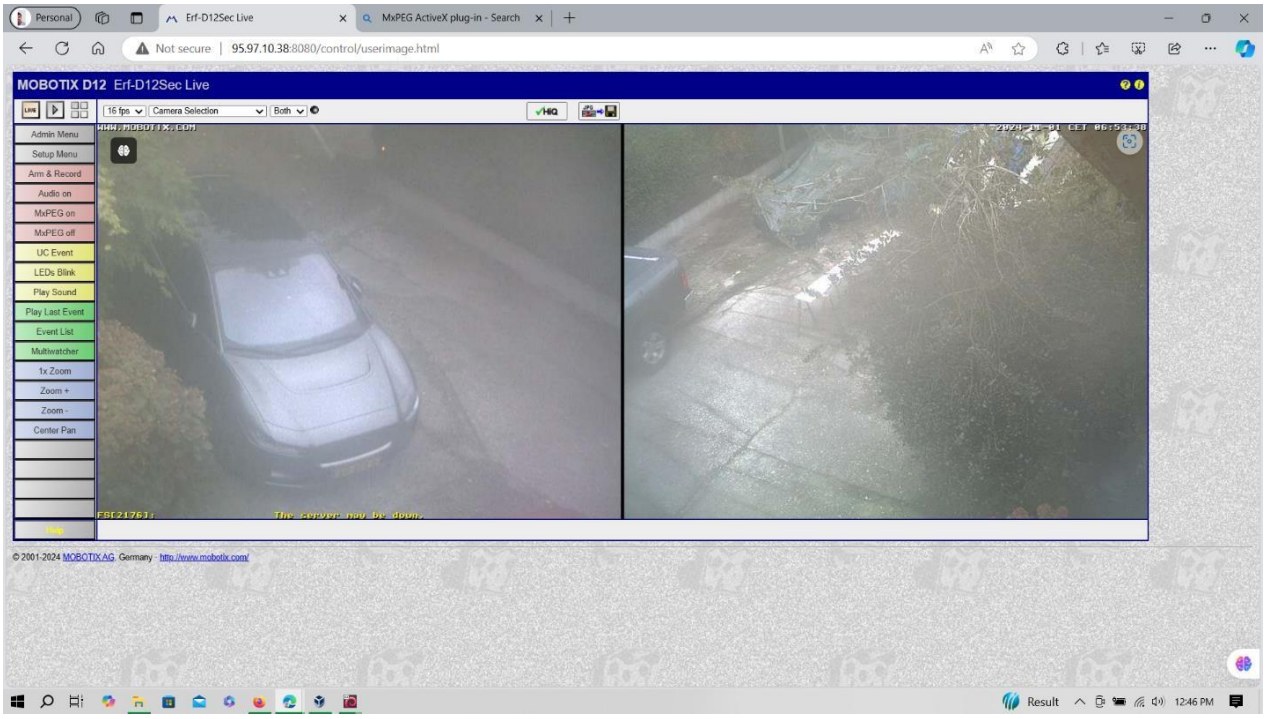

Figure 85 full control

At this point, I had complete control of the CCTV.

Let's move on to the next tool now.

Password Generator

This script generates a custom password wordlist based on a person's name and date of birth (DOB), useful for testing or research purposes, especially in cybersecurity, to understand potential password patterns.

This works just like a regular python program. It is run by issuing the command "**python3 password.py**".

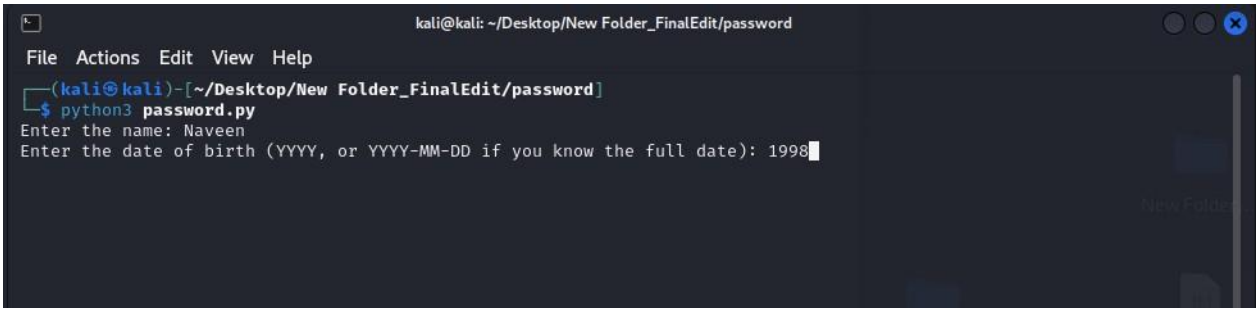

Figure 86 password Generate

This can actually be used as a social engineering tool. By providing the name and date of birth of the person being hacked, a wordlist of about **300,000** is generated. This is more practical and innovative than the old ones we know like **rockyou.txt** and **loveyou.txt**.

Figure 87 Result

The file above is a list of passwords that generated over 300,000.

These variations create a wordlist customized to the user's name and DOB patterns.

This is typically used for ethical testing and understanding password security, and misuse may lead to legal repercussions.

Let's focus on the next tool.

Remote Connect

This Python script was created by me as a simple tool to connect to a Metasploitable virtual machine via **SSH or FTP**. Metasploitable is a vulnerable machine commonly used in security testing and learning environments, allowing users to practice exploiting network services in a safe, controlled manner.

This also works like running a regular python tool. This command will run "**python3 remote_connect.py**"

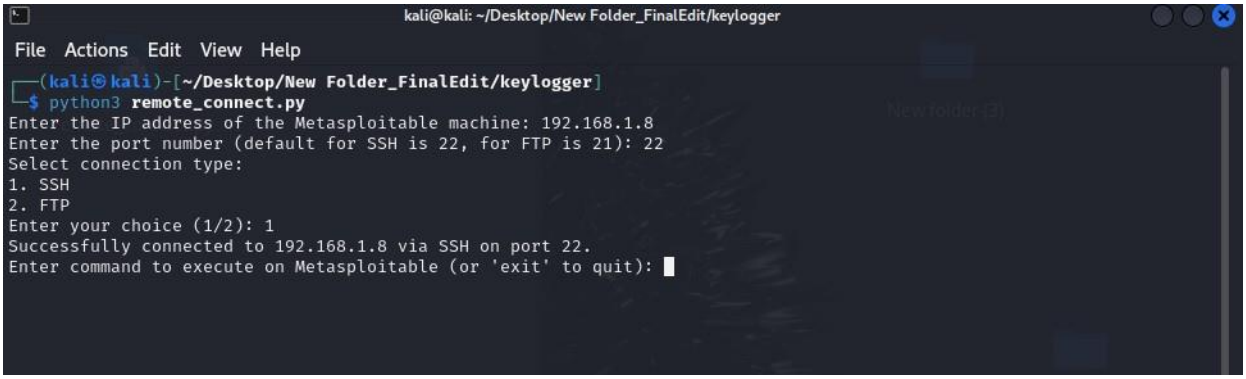

Figure 88 Remote Access

After issuing the command, you will be asked for the victim's IP address and when given, you should choose ftp or ssh, after which you will easily gain access to **Metasploitable.**

Look at the screenshot below. You are inside the Metasploitable machine.

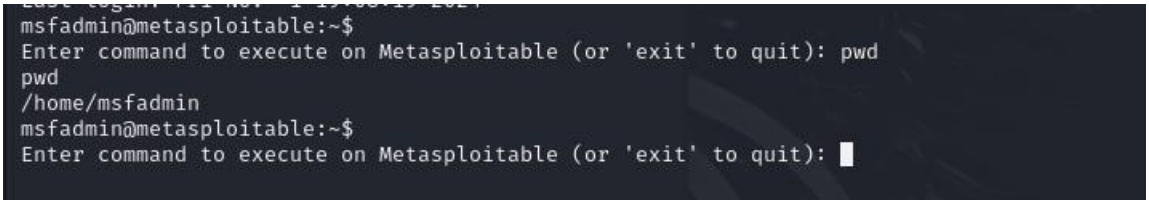

Figure 89 Hacked

pwd - print working directory = /home/msfadmin

I created this script because it is valuable for testing purposes in secure, controlled environments, such as understanding remote connections, automating shell commands, and transferring files. It could be developed to also be accessible on Linux systems. (RedHat/ Parrot Security)

Warning - Running this code on unauthorized machines or without permission could lead to legal consequences. This script is meant solely for ethical hacking or network testing within permitted environments.

Let's move on to the next tool,

Naviya_Multi-tool framework

In fact, this tool is a very powerful and robust tool. It is a new creation that combines all the scanning tools available on a Linux machine.

This tool is a tool that has been in the making for a long time. In fact, this tool is a very powerful and robust tool. This is a new creation made by combining all the scanning tools available on the Linux machine. For this, I added some new tools and I hope to implement it based on them. The creator of this is Shankar from India. I have renovated this again with his knowledge.

This tool is a multi-tool framework designed for security reconnaissance and vulnerability assessment. It integrates multiple tools and scans, making it easy for security analysts to quickly assess the security posture of a system.

Let's take a look at how this tool works.

Please read the action menu here before running the tool.

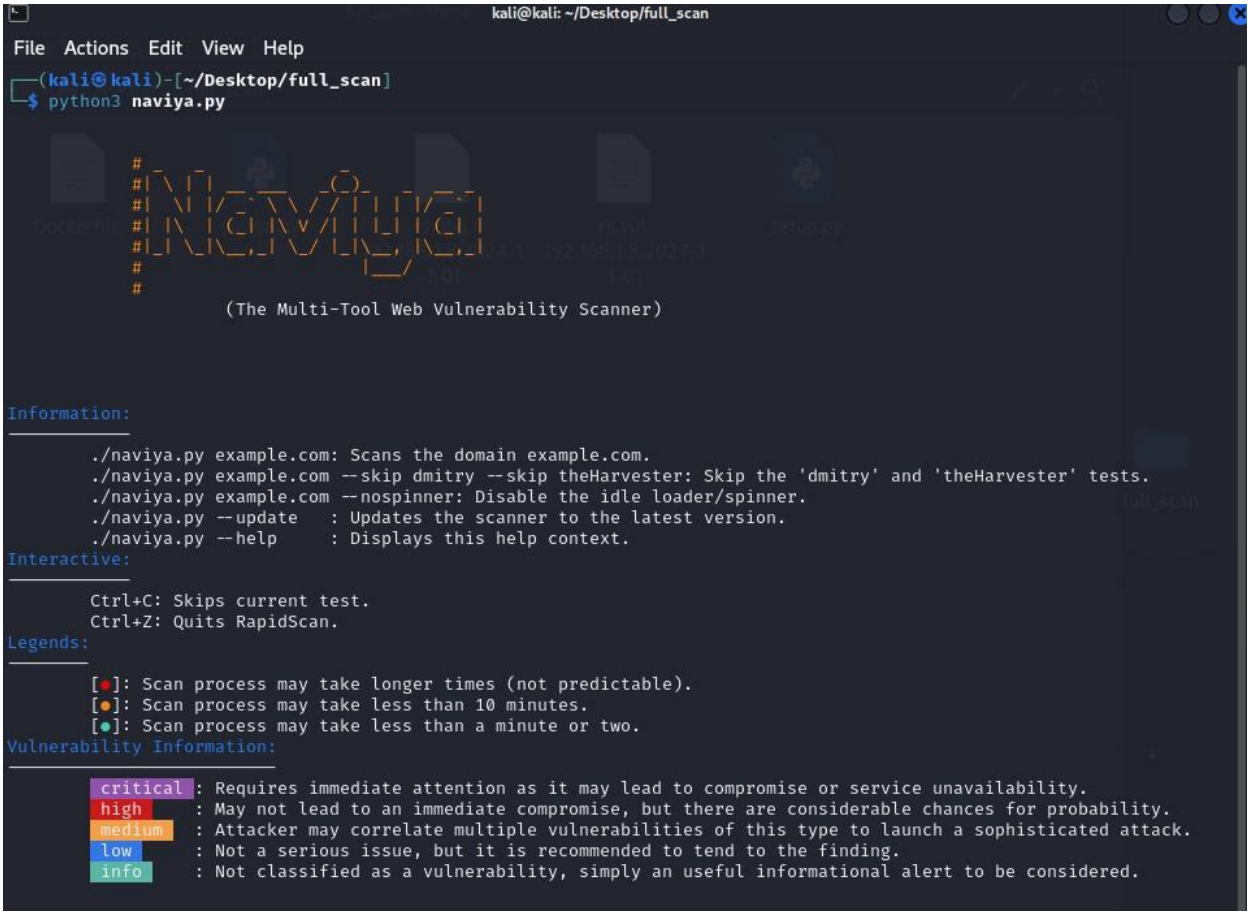

Figure 90 Web Scan Full

- *naviya.py example.com: Scans the domain example.com.*
- *naviya.py example.com --skip dmitry --skip theHarvester: Skip the 'dmitry' and 'theHarvester' tests.*
- *naviya.py example.com --nospinner: Disable the idle loader/spinner."*
- *naviya.py --update : Updates the scanner to the latest version."*
- *naviya.py --help : Displays this help context."*

Now you can scan the website as shown here.

Figure 91 Method

What I have currently obtained for this is the IP address of my Metasploitable machine.

“Python3 naviya.py 192.168.1.8”

Figure 92 proccessing

This scan tool uses all the scanning tools available on the Linux machine, so it will take some time. After a few minutes, we will finally get this output in the Terminal.

This also creates a large file containing separate data.

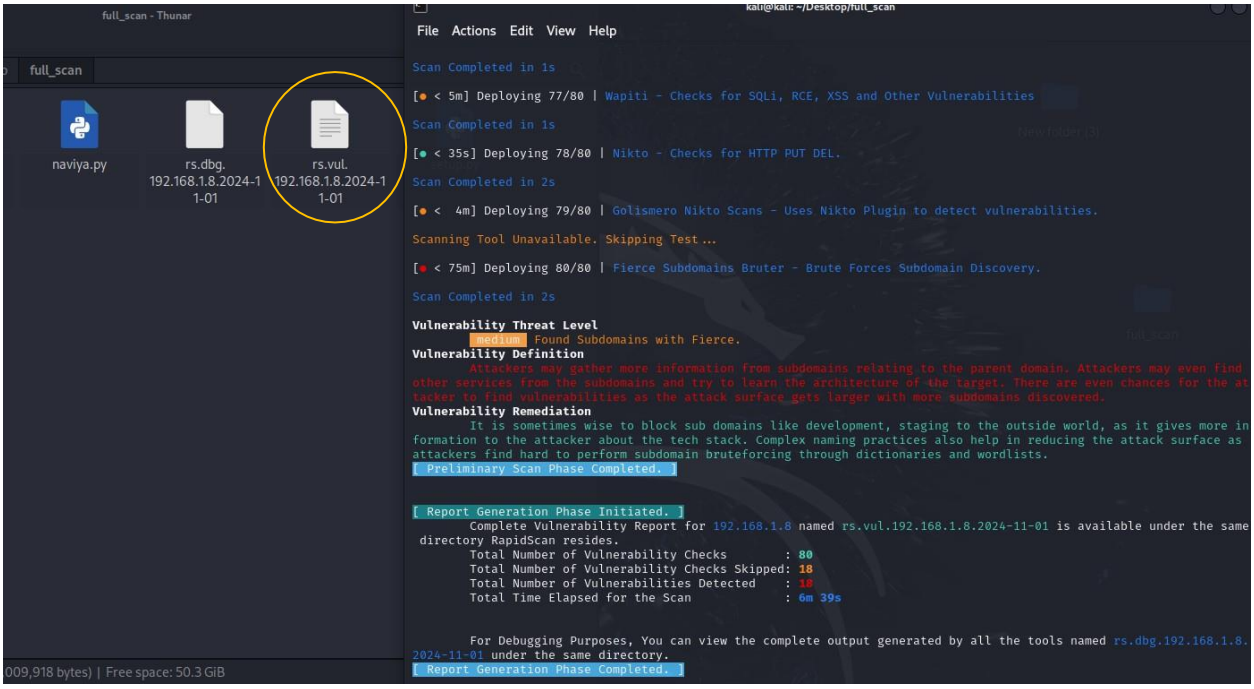

Figure 93 Result

The yellow circle is the data file that contains all the information.

This file is capable of scanning all the details of the scanned website using several different types of tools and providing accurate data.

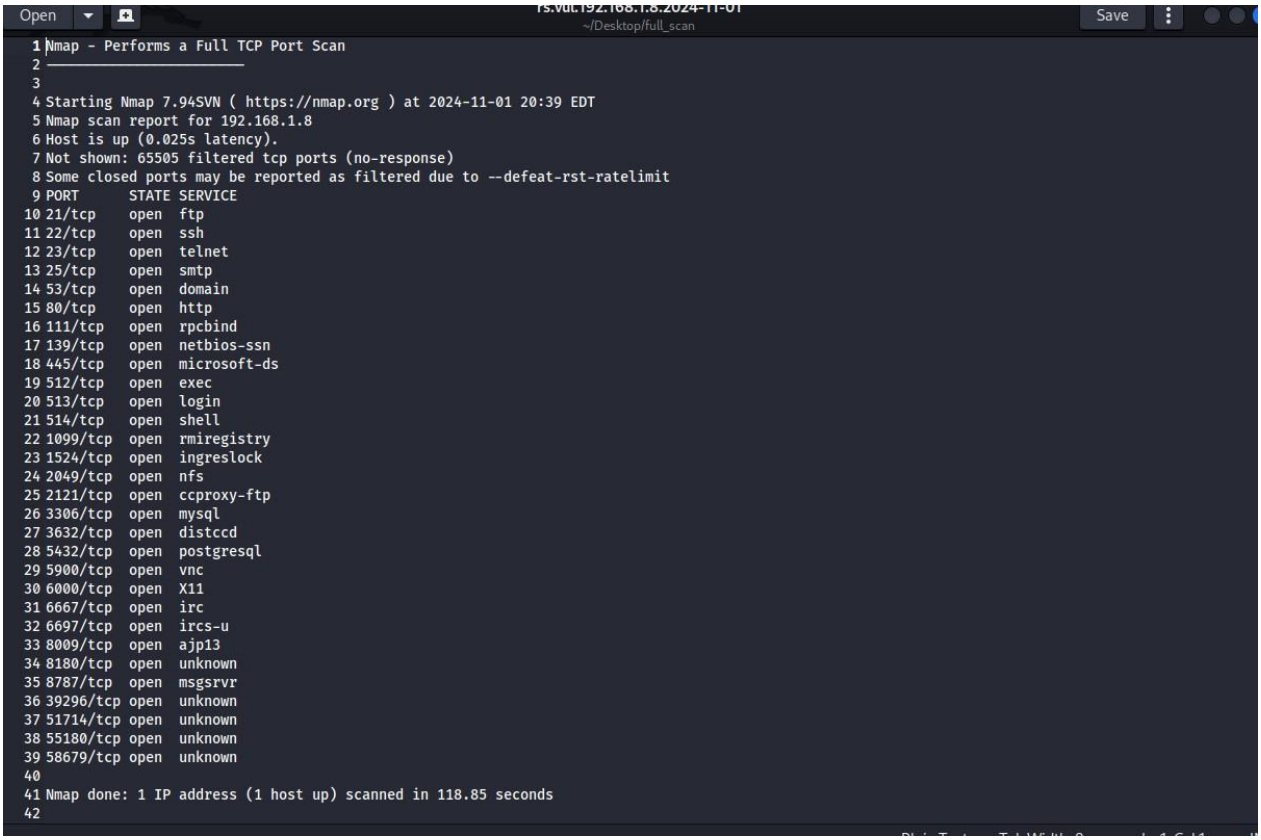

Figure 94 Result 1

This is the beginning of the file and it contains information about open ports and services.

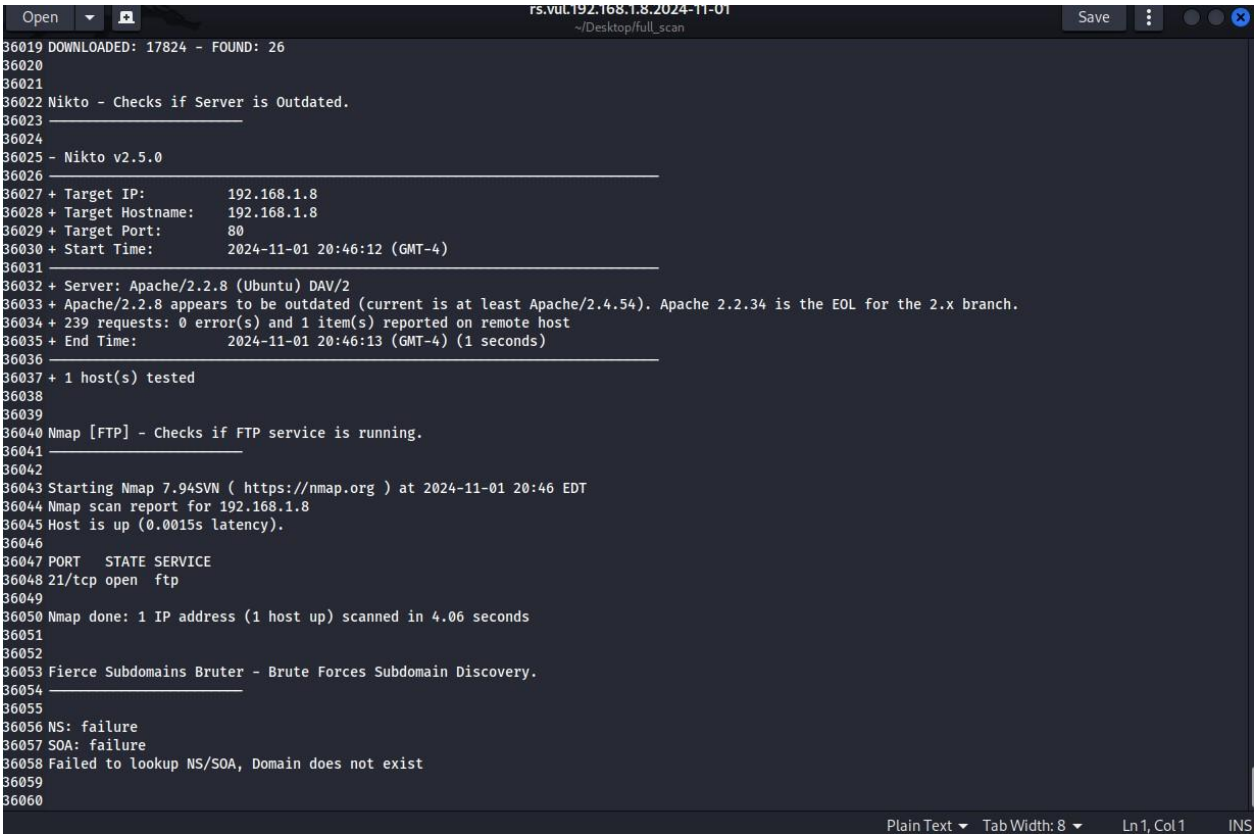

Figure 95 Big Result

This is the end of the file and also scanned the website very well. All the details are extended up to a value of 36060. This is very useful for Pentesting.

Key Features and Benefits

Multi-Tool Integration: This combines over 80 common security and intelligence gathering tools, automating processes that would otherwise require separate installations and commands. Examples include Nmap for port scanning, Nikto for web vulnerability scanning, and WhatWeb for web application fingerprinting.

Automation: By automating a series of checks, this provides a quick assessment of multiple aspects of system security, saving time and catching vulnerabilities that might otherwise be missed with manual, tool-based analysis.

Broad Coverage: This checks for network vulnerabilities, web application issues, service misconfigurations, and more. This makes it a well-rounded option for spying on multiple systems.

Ease of Use: With a single command, users can initiate multiple scans, making it accessible for people who don't want to learn each tool individually.

Limitations and Considerations

Accuracy and Overlap: Since it runs many tools in sequence, there may be overlapping results or redundant checks. An experienced analyst is usually required to interpret the results accurately.

Network Stress: Running multiple scans can increase network traffic and even trigger intrusion detection systems. This should be used responsibly, with explicit permission from the target systems.

Legal and Ethical Use: It is essential to use this only on systems that you own or have permission to assess, as unauthorized use may be illegal.

GLOSSARY

1. Ethical Hacking:

The practice of intentionally probing systems and networks to identify vulnerabilities that could be exploited by malicious hackers. Ethical hackers use the same tools and techniques as malicious actors but do so with permission and for the purpose of improving security.

2. **Penetration Testing:**

A method of evaluating the security of a system or network by simulating an attack from a malicious hacker. It involves using various tools and techniques to identify vulnerabilities and assess the effectiveness of security measures.

3. Vulnerability:

A weakness in a system, application, or network that can be exploited by attackers to gain unauthorized access or cause harm.

4. Toolset:

A collection of tools used for a specific purpose. In the context of Net-Caerus, the toolset comprises various software applications and utilities designed to facilitate ethical hacking and penetration testing.

5. Integrated Ethical Hacking System:

A comprehensive solution that consolidates various ethical hacking tools and resources into a single platform, providing users with an efficient and cohesive workflow for vulnerability assessment.

6. Web-Based Platform:

An online interface accessible via web browsers, allowing users to interact with the tools and features of the Net-Caerus system without needing to install software on their local machines.

7. Downloadable Linux Toolkit:

A package of ethical hacking tools designed specifically for Linux operating systems, which users can download and install on their local machines to conduct penetration tests.

8. User-Friendly Interface:

A design characteristic of software that makes it easy for users to navigate and operate the application without requiring extensive training or expertise.

9. Scanning Tool:

A software utility that examines systems or networks to identify vulnerabilities, open ports, and other security-related information.

10. Remote Access Tool:

A program that allows users to connect to and control a computer or device from a remote location, often used for legitimate purposes such as system administration.

11. Phishing:

A form of cyber-attack where attackers attempt to deceive individuals into providing sensitive information, such as usernames and passwords, by pretending to be a trustworthy source.

12. MAC Address:

A unique identifier assigned to network interfaces for communications on the physical network segment. It is used for network identification and access control.

13. DoS Attack (Denial of Service):

An attack that aims to make a computer, service, or network resource unavailable to its intended users by overwhelming it with traffic or exploiting vulnerabilities.

14. Cookies:

Small pieces of data stored by a web browser that contain information about user preferences and sessions, often used for session management and tracking.

15. CCTV Hacking:

The practice of gaining unauthorized access to closed-circuit television (CCTV) systems to view or manipulate camera feeds.

16. Payload:

The part of a malware program that performs the malicious action, such as deleting files, stealing data, or opening a backdoor for future access.

17. Keylogger:

A type of surveillance software that records keystrokes made by a user, often used to capture sensitive information like passwords and personal data.

18. Social Engineering:

A manipulation technique that exploits human psychology to gain confidential information or access to systems, rather than relying on technical hacking methods.

19. API (Application Programming Interface):

A set of protocols and tools for building software applications, allowing different software systems to communicate with each other.

20. GUI (Graphical User Interface):

A user interface that includes graphical elements, such as windows, icons, and buttons, enabling users to interact with the software visually.

21. Kali Linux

A Debian-based Linux distribution specifically tailored for penetration testing and security auditing. It includes a wide range of pre-installed tools for network and application testing, making it a popular choice among ethical hackers and security professionals.

22. Metasploitable

A deliberately vulnerable virtual machine designed for testing security tools and practicing penetration testing techniques. It serves as a target for ethical hackers to identify and exploit vulnerabilities in a controlled environment.

23. Parrot OS

A security-oriented operating system that provides a comprehensive suite of tools for penetration testing, digital forensics, and privacy protection. It is built on Debian and includes features aimed at security professionals and developers.

24. SSH (Secure Shell)

A cryptographic network protocol used for secure data communication, remote shell services, and executing commands on a remote computer securely. It provides a secure channel over an unsecured network in a client-server architecture.

25. SubProcess

A programming term referring to a secondary process created by a main process that allows for multitasking. In the context of scripting and programming, subprocesses can execute tasks simultaneously, enhancing the efficiency of operations.

26. Terminal

A text-based interface in Unix-like operating systems that allows users to interact with the system through command-line input and output. It enables users to execute commands, run scripts, and access system functions without a graphical user interface.

27. Unix

A powerful, multiuser operating system that has influenced many other operating systems, including Linux. Known for its stability, security features, and multitasking capabilities, Unix is widely used in server environments and for academic purposes.