جامعة إربد الأهلية
IRBID NATIONAL UNIVERSITY

From Basics to Breach: A Step-by-Step Virtual Environment for Cybersecurity Education

By:
Mohamad Alsharou
Hamza AL-khazali
Mahmoud AL-bokhari

Supervisor:
Motasem Abu-Dawas
Mahmoud Aljamal

Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor's in Computer Science \Faculty of Science and Information Technology IT at Irbid National University

Irbid – Jordan
14/1/2025

Contents

List of Figures

List of Tables

Chapter 1

Abstract

Abstract

This project presents Hack Lab, a custom-designed educational environment developed to help beginners learn penetration testing through hands-on practice. Recognizing the gap between theoretical cybersecurity education and practical skill development, the project introduces three themed virtual machines, each simulating real-world network configurations and vulnerabilities using common protocols such as SSH, FTP, SMTP, Apache, and MySQL. Each machine was built from scratch using Kali Linux tools and configured with a unique scenario involving web enumeration, password cracking, service misconfigurations, and privilege escalation.

Throughout development, significant attention was given to balancing technical challenge with accessibility. Tools such as Nmap, Hydra, Hashcat, Gobuster, and Metasploit were carefully integrated into the scenarios to reflect actual penetration testing workflows while remaining beginner-friendly. All virtual machines were tested thoroughly and distributed via Google Drive, with full walkthrough videos hosted on YouTube to ensure open access and independent learning.

Rather than relying on passive theory, Hack Lab provides students with a structured and engaging platform to practice offensive security techniques in a controlled environment. The goal was not only to teach tools and commands, but to help learners think like ethical hackers. This abstract reflects the practical design choices, the challenges faced during development, and the educational value the lab aims to deliver.

Chapter 2

Introduction

2.1 Background and Motivation

In recent years, the digital world has witnessed an alarming rise in cyberattacks targeting individuals, corporations, and governments. As organizations become increasingly dependent on interconnected systems and online services, the potential for exploitation and data breaches grows exponentially. This evolving threat landscape has created a significant demand for cybersecurity professionals who possess not only theoretical knowledge but also practical skills in detecting and mitigating vulnerabilities. Ethical hacking, or penetration testing, has emerged as a critical discipline to evaluate and strengthen digital defenses before malicious actors can exploit them.

Despite the growing demand for cybersecurity expertise, many educational pathways remain heavily focused on theory, offering limited opportunities for hands-on learning. Most academic programs introduce students to the concepts of information security, cryptography, and network defense, but often neglect the experiential components that are vital for understanding real-world threats. Additionally, many online courses and CTF platforms assume a baseline of technical experience, which can be discouraging for beginners who are still developing foundational skills.

One of the main barriers facing aspiring ethical hackers is the lack of beginner-friendly environments that allow them to practice safely and progressively. Existing platforms are frequently fragmented, either offering narrowly focused exercises or overwhelming learners with complex tools and setups. Furthermore, many of these platforms are paywalled or difficult to navigate without prior experience, leaving many students without an accessible path to practical engagement with cybersecurity tools and techniques.

This project was motivated by the need to fill this gap by creating a comprehensive and immersive learning experience specifically tailored to beginners. The idea was to build a virtual lab environment where learners can engage with realistic scenarios, gain exposure to commonly used tools, and gradually build their skills through a structured,

step-by-step process. Each virtual machine within the lab presents a themed challenge that mimics real-world vulnerabilities, guiding learners through the entire attack chain—from reconnaissance to privilege escalation—using widely recognized cybersecurity methodologies.

By focusing on accessibility, educational value, and technical realism, this project aims to democratize cybersecurity education. It removes barriers such as high costs, steep learning curves, and fragmented resources by offering a unified platform that includes pre-configured machines, guided tutorials, and video walkthroughs. The goal is not only to teach students how to exploit systems ethically but also to help them think like security professionals—analyzing systems critically, recognizing potential threats, and understanding the mindset behind common attack vectors.

2.2 Problem Statement

Despite the growing global demand for cybersecurity professionals, there is a noticeable lack of practical and beginner-friendly environments for learning ethical hacking and penetration testing. This project addresses that gap by examining the key barriers that beginners face when entering the field.

- **Most existing platforms are too advanced for newcomers:** Many cybersecurity learning platforms are designed for intermediate or advanced users, assuming prior knowledge of Linux commands, networking, and tools leaving beginners overwhelmed, confused, and discouraged before they can build foundational skills.
- **Resources are often behind paywalls or require complicated setup:** Even when quality content exists, it's often locked behind paywalls or requires technically challenging setups (like VMs and tool installations), creating a barrier that stops beginners from reaching hands-on practice.
- **Educational content is fragmented across different tools and websites:** Learning ethical hacking often involves jumping between unstructured resources like YouTube, GitHub, and blogs, making it hard for beginners to know where to start or how tools work together in real scenarios.
- **Beginners face steep learning curves due to tool complexity:** Tools like Nmap, Hydra, Burp Suite, and Metasploit are powerful but complex without clear, simplified guidance, beginners struggle to understand their usage, leading to slow progress and reduced confidence.
- **There is a lack of guided, hands-on scenarios for real practice:** Most platforms fail to offer structured, practical environments where learners can apply theory step

by step, which limits the connection between knowledge and real-world cybersecurity skills.

- **This leads to discouragement and limited skill development:** These combined challenges technical barriers, disorganized resources, and lack of practice often cause learners to give up early, contributing to the ongoing shortage of qualified cybersecurity professionals.

2.3 Project Objectives

This project aims to design and implement three custom virtual machines that simulate real-world protocols and services, including Apache, SSH, FTP, SMTP, and MySQL. Each machine is designed with a specific theme and escalating difficulty, providing users with realistic Capture The Flag (CTF) scenarios. The goal is to offer an interactive, hands-on environment that allows beginners to apply cybersecurity concepts in a controlled, challenge-based format.

Another core objective is to teach essential penetration testing tools such as Nmap, Hydra, Gobuster, and Metasploit. These tools will be introduced through practical examples within each virtual machine. Learners will follow guided tutorials that demonstrate real-world attack vectors, from network scanning to brute-force login and vulnerability exploitation. By learning in context, users gain not only technical skills but also the intuition to select and apply the right tools for different security assessments.

The project is committed to open-access learning. All resources including the virtual machines, instructional documentation, and detailed video tutorials will be freely distributed through platforms like YouTube and Google Drive. This ensures that students and self-learners from any background can access high-quality, structured cybersecurity education without facing financial or technical barriers. The ultimate goal is to make ethical hacking education more inclusive, practical, and engaging for aspiring cybersecurity professionals.

Table 1: Main and Sub-Objectives of the Project

Main Objective	Sub-Objectives
Build a practical lab environment	Develop 3 themed VMs, Integrate real protocols (SSH, FTP, etc.)
Teach ethical hacking tools	Demonstrate tool usage in real scenarios, Publish video walkthroughs
Provide free and open access	Host VM images on Google Drive, Upload videos to YouTube

Table 1 outlines the primary objectives of the Hack Lab project alongside their corresponding sub-objectives. The overarching goal was to design a beginner-focused lab environment that simulates real-world cybersecurity challenges. To fulfill this vision, three themed virtual machines were developed, each integrating widely-used protocols such as SSH, FTP, and MySQL. The project also aimed to introduce ethical hacking tools like Nmap, Hydra, and Gobuster within practical, hands-on scenarios supported by guided video walkthroughs. Finally, the commitment to open-access delivery was achieved by hosting the virtual machines and tutorial content on platforms like Google Drive and YouTube, removing financial and technical barriers for learners around the world.

Table 2: Protocols and Services Used in Each Virtual Machine

Machine Name	Protocols Included	Main Educational Focus
Mistry Mail	Apache, SSH, SMTP	Email exploration, password cracking, root escalation
Over Power	Apache, SSH, FTP	Brute-force attacks, FTP enumeration, service scanning
Treasure	Apache, SSH, MySQL	Database analysis, base64 decoding, SSH login chaining

Table 2 presents the set of protocols implemented in each virtual machine, highlighting the core educational focus behind their inclusion. Mistry Mail combines Apache, SSH, and SMTP to simulate a scenario involving internal communication leaks, hashed credentials, and root privilege escalation. Over Power incorporates Apache, SSH, and FTP, emphasizing service enumeration, brute-force access, and vulnerable file-sharing setups. Treasure includes Apache, SSH, and MySQL, guiding learners through multi-step attacks

involving database queries, base64 decoding, and chained SSH authentication. This deliberate variation across machines ensures that each scenario introduces a different technical focus, gradually building the learner's penetration testing capabilities.

Table 3: Tools Covered and Skills Developed

Tool	Primary Use	Skills Developed
Nmap	Network scanning	Port discovery, service enumeration
Hydra	Brute-force login	Credential attacks on SSH/FTP
Gobuster / Dirb	Directory brute-forcing	Hidden path discovery, web structure analysis
Metasploit	Exploitation framework	Exploiting vulnerabilities, reverse shell generation
Netdiscover	Network reconnaissance	Identifying live hosts and MAC addresses in LAN
Hashcat	Password hash cracking	Cracking MD5/SHA1 hashes using GPU acceleration
John the Ripper	Legacy password cracking	Wordlist attacks and hash analysis for password recovery
Wget	File downloading	Retrieving files from web servers, automated file grabbing

Table 3 provides an overview of the core tools covered throughout the Hack Lab project and the primary cybersecurity skills they are intended to develop. These tools were selected not only for their popularity in real-world penetration testing workflows but also for their suitability in beginner-level education. Nmap introduces learners to port scanning and basic network reconnaissance. Hydra helps demonstrate how brute-force attacks are conducted against services like SSH and FTP. Gobuster and Dirb allow students to practice discovering hidden web directories and understanding server structure. Metasploit is included to give exposure to automated exploitation techniques and reverse shell deployment. Other tools such as Netdiscover and Wget contribute to network mapping and file retrieval skills respectively. Additionally, password cracking is reinforced through tools like Hashcat and John the Ripper, which are capable of attacking MD5 and SHA1 hashes using dictionary-based methods or GPU acceleration. Overall, the toolset forms a

foundational part of the curriculum, guiding learners thr

2.4 Scope of the Project

This project is focused on providing a foundational, beginner-level penetration testing environment using three custom virtual machines. Each machine includes real-world protocols such as Apache, SSH, FTP, SMTP, and MySQL. The scope covers core security concepts like reconnaissance, enumeration, brute-force attacks, web directory discovery, password cracking, and basic privilege escalation. Learners are introduced to essential tools including Nmap, Hydra, Gobuster, Metasploit, Netdiscover, Hashcat, John the Ripper, and Wget. The project is designed to be fully self-contained, requiring no external infrastructure beyond a virtualization platform like VirtualBox or VMware.

However, the project deliberately excludes advanced cybersecurity topics that fall outside the beginner scope. These include deep binary exploitation, buffer overflows, advanced reverse engineering, malware analysis, complex web application vulnerabilities like server-side request forgery (SSRF) or race conditions, and real-world threat emulation. The lab does not simulate enterprise networks, intrusion detection systems (IDS), or active defense mechanisms. Its aim is strictly educational to build a strong foundational skill set without overwhelming the learner with highly complex attack vectors or enterprise grade setups.

Table 4: Scope of the Project – Included vs. Excluded Topics

Included in the Project	**Not Included in the Project**
Basic protocols: Apache, SSH, FTP, MySQL, SMTP	Advanced services: LDAP, Kerberos, Active Directory
Essential tools: Nmap, Hydra, Gobuster, Metasploit, Netdiscover, Hashcat, John, Wget	Complex tools: Burp Suite Pro, Wireshark deep packet analysis
Beginner-level CTF challenges	Real-world red team vs. blue team simulation
Basic privilege escalation (Linux-based)	Kernel exploits, rootkit development, memory corruption
Offline learning with pre-configured VMs	Cloud or hybrid infrastructure simulation

Table 2.1 presents a comparative analysis of the core network protocols integrated into the Hack Lab environment. Each protocol was selected to simulate realistic services that

are commonly encountered in ethical hacking scenarios. Apache, for instance, serves as the primary web server in all machines and allows for exercises in directory enumeration, reverse shell uploads, and source code inspection. SSH enables secure remote access and introduces learners to privilege escalation techniques through the exploitation of weak login credentials. FTP is used to simulate legacy file-sharing systems and demonstrates vulnerabilities such as anonymous access and exposed login information. The inclusion of MySQL supports database-based exploitation and decoding tasks, while SMTP models internal communication systems with mailbox information leakage. Collectively, these protocols provide a hands-on learning platform that reinforces practical skills while exposing students to common misconfigurations and attack vectors found in real-world infrastructures.

Table 4 delineates the pedagogical boundaries of the Hack Lab project by contrasting the included topics with those that were intentionally excluded. The project's focus was to provide a clear, structured foundation in ethical hacking without overwhelming learners with overly complex or advanced content. Included in the project are essential network protocols such as Apache, SSH, FTP, MySQL, and SMTP, as well as widely-used penetration testing tools like Nmap, Hydra, Gobuster, Metasploit, and others. These components support beginner-level Capture The Flag (CTF) challenges that emphasize skills such as enumeration, brute-force login, file retrieval, and basic privilege escalation.

Conversely, advanced services such as LDAP, Kerberos, and Active Directory were excluded, along with complex tools like Burp Suite Pro and Wireshark deep packet inspection. High-level techniques including kernel exploits, memory corruption, and red team vs. blue team simulations were also left out of scope. The decision to keep the lab offline and self-contained, using pre-configured virtual machines rather than cloud-based setups, further reinforces its accessibility and simplicity. This clearly defined scope ensures that the learning path remains focused, manageable, and aligned with the needs of cybersecurity beginners.

2.5 Educational Delivery and Distribution

Table 2.1: Comparison Table of Integrated Protocols

Protocol	**Purpose in Lab**	**Real-World Use Case**	**Key Vulnerabilities Simulated**	**Skills Developed**
Apache	Hosts web files and hidden directories	Web applications and sites	Directory traversal, exposed source code	Web enumeration, reverse shell uploads
SSH	Secure remote access	Server administration	Weak login credentials, lack of 2FA	Remote access, privilege escalation
FTP	File hosting and retrieval	Legacy file-sharing systems	Anonymous access, exposed credentials	File discovery, leak exploitation
MySQL	Credential storage & data simulation	Web/database backends	Poor encryption, misconfigurations	Database enumeration, Base64 decoding
SMTP	Simulated email system	Internal communications	Info leakage in mailboxes	Local info gathering, lateral movement

The Hack Lab project incorporates five commonly used network protocols **Apache (HTTP), SSH, FTP, MySQL, and SMTP** to simulate realistic and diverse attack surfaces. These protocols were intentionally selected because they reflect services frequently encountered in real-world infrastructures, especially in enterprise and legacy systems. By embedding them across the lab's three machines, learners are exposed to a range of vulnerabilities associated with each protocol and are challenged to apply appropriate techniques to exploit them ethically.

Apache, the most widely used web server, forms the public-facing entry point for most of the machines. It is used to host websites and hidden directories that include vulnerable elements such as backup files, login pages, or embedded comments in source code. These challenges help students practice web enumeration, learn to read HTML and robots.txt files, and even exploit misconfigured file uploads to gain reverse shell access. Apache sets the stage for initial reconnaissance and is a critical protocol in real-life web

application penetration testing.

SSH is integrated into the lab to simulate remote access scenarios, mimicking how system administrators manage Linux servers (Stöcklin, 2022). In this project, SSH login credentials are hidden within web pages, encoded files, or databases, challenging learners to brute-force access or crack hashes. Once inside, students must navigate the system, search for privilege escalation paths, and ultimately obtain root access. This reinforces the concepts of post-exploitation and privilege escalation, which are vital in any complete penetration testing workflow.

FTP is included to demonstrate the dangers of outdated or poorly secured file-sharing services. In Hack Lab, FTP servers may allow anonymous login, expose sensitive files like ‘creds.txt‘, or include misleading configuration notes. Learners use command-line tools to connect, download files, and uncover further clues that may lead to user credentials or deeper access. This helps students understand data leakage risks and teaches the importance of validating permissions on public file services.

MySQL and SMTP represent back-end and internal communication vulnerabilities. With MySQL, students explore databases containing encrypted credentials, dummy user data, and Base64-encoded usernames. This teaches them how to query, decode, and interpret database content a critical skill in data-focused security assessments. SMTP, on the other hand, simulates internal email servers where attackers might extract credentials or system notes from local mailboxes after gaining shell access. This protocol emphasizes the role of information leakage and lateral movement, showing that even “non-critical” services can reveal vital insights in an attack chain.

Table 2.2: Comparison Table of Core Penetration Testing Tools

Tool	**Purpose in Lab**	**Primary Use Case**	**Skill Developed**	**Machine Used In**
Nmap	Port and service scanning	Network reconnaissance	Enumeration and mapping of open ports	All Machines
Hydra	Brute-force login attempts	Cracking SSH/FTP credentials	Authentication attacks	Over Power, Treasure
Hashcat	Password hash cracking	Cracking SHA1/MD5 hashes	Cracking encrypted credentials	Mistry Mail, Treasure
John the Ripper	Alternative hash cracking	Legacy password analysis	Wordlist brute-force	Mistry Mail
Gobuster / Dirb	Web directory enumeration	Discovering hidden web files/folders	Web reconnaissance	All Machines
Netcat	Reverse shell handler	Reverse connection management	Post-exploitation control	Over Power, Treasure
Wget	File downloading via HTTP/FTP	Pulling target files from servers	File retrieval & analysis	All Machines

The Hack Lab project integrates a carefully selected set of core penetration testing tools that reflect the real-world methodologies used by cybersecurity professionals. These tools were chosen for their simplicity, power, and relevance to the types of vulnerabilities embedded in the virtual machines. Rather than overwhelming beginners with advanced frameworks, the lab introduces tools that are both practical and foundational ensuring learners develop confidence as they progress. Each tool plays a critical role in the exploitation workflow, from initial reconnaissance to post-exploitation tasks.

Nmap is used to discover open ports and services during the enumeration phase, while Hydra assists in brute-forcing login credentials for services like SSH and FTP. Tools like Gobuster and Dirb are essential for locating hidden directories in web servers, and Wget enables the direct download of exposed files. In terms of credential attacks, Hashcat and John the Ripper help students learn how to crack SHA1 or MD5 hashes found in system files or databases. Netcat, on the other hand, introduces learners to reverse shell sessions helping them understand post-exploitation access and system control. Each machine in

the lab was designed with challenges that highlight when and how to use these tools effectively.

The project integrates two major delivery platforms:

YouTube

- Provides full exploitation walkthroughs with clear narration and visual demonstration.
- Helps beginners follow real attack chains without prior experience.
- Enables self-paced, repeatable learning.

Google Drive

- Hosts all OVA/ISO virtual machine files.
- Includes setup instructions and folder organization by machine.
- Ensures free, global access without technical or financial barriers.

Table 2.3: Educational Platforms Comparison Table

Platform	**Purpose**	**Benefits to Learners**	**Type of Content Provided**
YouTube	Educational walkthrough delivery	Visual, interactive learning; global reach; beginner-friendly	Full video tutorials for each machine
Google Drive	Machine file distribution	Easy downloads; organized folders; open sharing	OVA/ISO virtual machine files + setup guides

Table 2.3 compares the two primary platforms used for educational content delivery in the Hack Lab project. YouTube was selected as the medium for video-based walkthroughs due to its accessibility, visual clarity, and interactive learning features. It enables learners to pause, rewind, and follow along with real-time demonstrations, making it especially beneficial for beginners. In contrast, Google Drive serves as the main platform for distributing the virtual machine files and supplementary materials. By organizing files into clearly labeled folders and allowing open access, it simplifies the setup process for users across different systems. Together, these platforms ensure that both the instructional and technical components of Hack Lab are delivered in an efficient, user-friendly, and globally accessible manner.

2.5.1 Research Contribution

This project offers a novel contribution to cybersecurity education by addressing key gaps in accessibility, practicality, and instructional design for beginners. Unlike traditional platforms that rely heavily on theory or assume advanced knowledge, the Hack Lab initiative delivers an end-to-end learning experience built on real-world simulations, guided tutorials, and free access. The following points highlight the main contributions achieved through this work.

- **Development of a Beginner-Centric Penetration Testing Lab:** Designed and implemented three themed VMs with real-world services (Apache, SSH, FTP, MySQL, SMTP) to offer a structured, beginner-friendly penetration testing environment.
- **Integration of Open-Access Learning with Guided Multimedia Support:** Distributed all learning materials freely via YouTube and Google Drive, removing cost and setup barriers for global self-learners.
- **Scenario-Based Instructional Design for Applied Skill Development:** Integrated CTF-style challenges and essential tools (Nmap, Hydra, Metasploit, etc.) to teach full attack chains in a realistic, hands-on format.

2.6 Organization of the Paper

This project is organized into six chapters:

- **Chapter 1** introduces the research topic, outlines the motivation, problem statement, objectives, and scope of the project.
- **Chapter 2** provides a comprehensive review of existing literature and platforms related to cybersecurity education and practical penetration testing environments.
- **Chapter 3** details the methodology used to design and implement the Hack Lab, including the planning, tool selection, and scenario development.
- **Chapter 4** offers a technical overview of the services, tools, and protocols integrated into the virtual machines.
- **Chapter 5** presents the results and outcomes of the project, including the effectiveness of the machines, tool performance, and user feedback.
- **Chapter 6** concludes the thesis with a summary of findings and recommendations for future improvements or expansions.

Chapter 3

Literature Review

(Piantadosi et al., 2023) Their study investigates how vulnerabilities are identified, managed, and resolved in two of the most widely-used open-source web servers: Apache HTTP Server and Apache Tomcat. The authors conducted a comprehensive analysis of 337 real-world vulnerabilities, reviewing the time required for fixes, the types of issues encountered, and the engineering practices used in responding to security reports. The research highlights the real-world challenges of maintaining secure open-source infrastructure and the importance of timely patching and community involvement. This work is directly relevant to environments like Hack Lab, where learners interact with intentionally vulnerable configurations. The findings help justify the pedagogical use of Apache-based vulnerabilities as real-world examples in hands-on cybersecurity training.

(Bishop et al., 2021) Their study presents an analysis of the types of cybersecurity skills and knowledge that participants develop through Capture the Flag (CTF) competitions. It categorizes CTF challenges into domains such as cryptography, reverse engineering, and web security, and maps these challenges to learning objectives found in academic curricula. The authors argue that CTFs represent a form of experiential learning that combines theoretical knowledge with hands-on application. This approach encourages active learning, enhances critical thinking, and better prepares learners for real-world cybersecurity tasks. In the context of Hack Lab, this study supports the use of CTF-style machines for beginner training, confirming that such challenges are effective for teaching complex topics in an accessible way.

(Goodall et al., nd) Their study explores how CTF competitions facilitate gamified learning in cybersecurity education. Using the Information Search Process as a theoretical framework, the authors examined how students interact with CTF challenges, including their problem-solving behavior, research strategies, and learning motivation. The findings show that CTF environments promote engagement, persistence, and knowledge retention. Learners are motivated by the challenge-based format, and they build both individual and

collaborative skills while solving practical security tasks. This aligns closely with Hack Lab's approach, which uses CTF-style problems to simulate real-life hacking scenarios in a motivational structure.

(Nasir and Islam, 2023) Their study examines security weaknesses associated with the use of SSH in wireless network environments. It highlights common configuration errors, poor key management practices, and outdated encryption algorithms that make SSH vulnerable to brute-force attacks and man-in-the-middle threats. The study also proposes several mitigations, including using key-based authentication, stronger encryption, and secure tunneling methods. It emphasizes the need for administrators and learners alike to understand the limitations of security tools, especially when deployed in insecure environments. For Hack Lab, where SSH is a primary access method in each virtual machine, this research supports the inclusion of SSH-focused challenges.

(Arce et al., 2010) Their study analyzes weaknesses in the MySQL authentication protocol and demonstrates how an attacker could compromise user credentials with minimal access. The paper models various attack scenarios and highlights cryptographic flaws that allow for password recovery and session hijacking. The research provides insight into how seemingly secure login mechanisms can be subverted when not properly implemented or updated. It also outlines recommendations for more secure configurations and future protocol designs. This paper is directly relevant to Hack Lab's MySQL-based virtual machine, which teaches learners how to exploit and harden database systems.

(Ali and Noor, 2022) Their study compares traditional lecture-based learning with practical, challenge-driven formats such as CTFs. The authors argue that CTF environments provide a more engaging and effective platform for developing penetration testing skills. Through case studies and interviews, the paper demonstrates that students who train using CTFs are better able to understand complex security issues, retain technical knowledge longer, and apply their skills in real-world situations. Hack Lab adopts this exact educational approach by combining theory with practical scenarios, making this study a direct academic support for its learning methodology.

(Al-Shaer and Hamed, 2020) Their study focuses on common SMTP protocol vulnerabilities and their implications for email security. The authors discuss risks such as spoofing, open relay attacks, and the lack of encryption in standard SMTP implementations. It also reviews the effectiveness of modern defenses like SPF, DKIM, and DMARC, and provides guidance on hardening mail servers. These recommendations are positioned in the context of both enterprise systems and training environments. Hack Lab includes SMTP in one of its virtual machines to simulate these real-world risks, making this re-

search an ideal reference.

(Asif and Nwankwo, 2021) Their study evaluates FTP's relevance and risks in modern networks. It outlines classic weaknesses in the protocol, including cleartext password transmission and anonymous access, and proposes updated configurations using secure variants like FTPS and SFTP. The authors suggest architectural and configuration-level improvements for organizations that still rely on FTP. They also identify training gaps in handling legacy systems. For Hack Lab, which includes an FTP-based challenge, their study strengthens the rationale for teaching FTP security and offering hands-on practice with misconfigured file transfer services.

(Gupta and Sharma,) Their study explains why Linux is widely adopted in cybersecurity training and penetration testing. It highlights Linux's open-source nature, access to security tools, system transparency, and flexibility in creating custom lab environments. The paper compares Linux with Windows, showing how Linux provides more control over the system and better access to low-level components required in security tasks. It also emphasizes the role of distributions like Kali Linux in ethical hacking education. Hack Lab is built entirely on Linux, and this paper validates that choice by outlining its pedagogical and technical strengths.

Chapter 4

Methodology

4.1 Methodology Introduction

This section presents the methodological framework adopted for the implementation of the "Hack Lab" project, a scenario-based educational platform designed to teach penetration testing fundamentals to absolute beginners. The project's primary aim is to provide learners with a hands-on, practical experience in ethical hacking by interacting with a controlled, virtualized environment. The methodology is rooted in instructional design principles and simulates real-world attack paths using simplified, story-driven challenges. The approach combines pedagogical clarity with technical depth to ensure participants can build a strong foundation and gradually scale their knowledge.

The project was developed on a Kali Linux-based attacker machine, targeting three custom-built Ubuntu Server virtual machines, each of which represents a distinct penetration testing scenario. Each target machine incorporates essential network services and protocols namely SSH, FTP, MySQL, SMTP, and Apache (HTTP) to emulate real vulnerabilities encountered in the field. These protocols were selected based on their relevance to commonly exploited services in modern cybersecurity contexts. The virtual machines are structured to cover key ethical hacking concepts, including reconnaissance, enumeration, decryption, brute-force attacks, reverse shell access, and privilege escalation.

The methodology is divided into four major phases: planning, environment construction, penetration testing, and documentation. The planning phase focused on identifying the most critical beginner-level tools and commands that every penetration tester must know. This includes selecting practical techniques for password discovery, encryption analysis, and file inspection. In the construction phase, scenarios were implemented through user-focused narratives that promote active learning and intuitive discovery of system weaknesses. The testing phase ensured the functionality and clarity of all attack

paths by conducting full penetration tests on each machine. Finally, every stage was carefully documented using step-by-step screenshots to support learners during their practice and review.

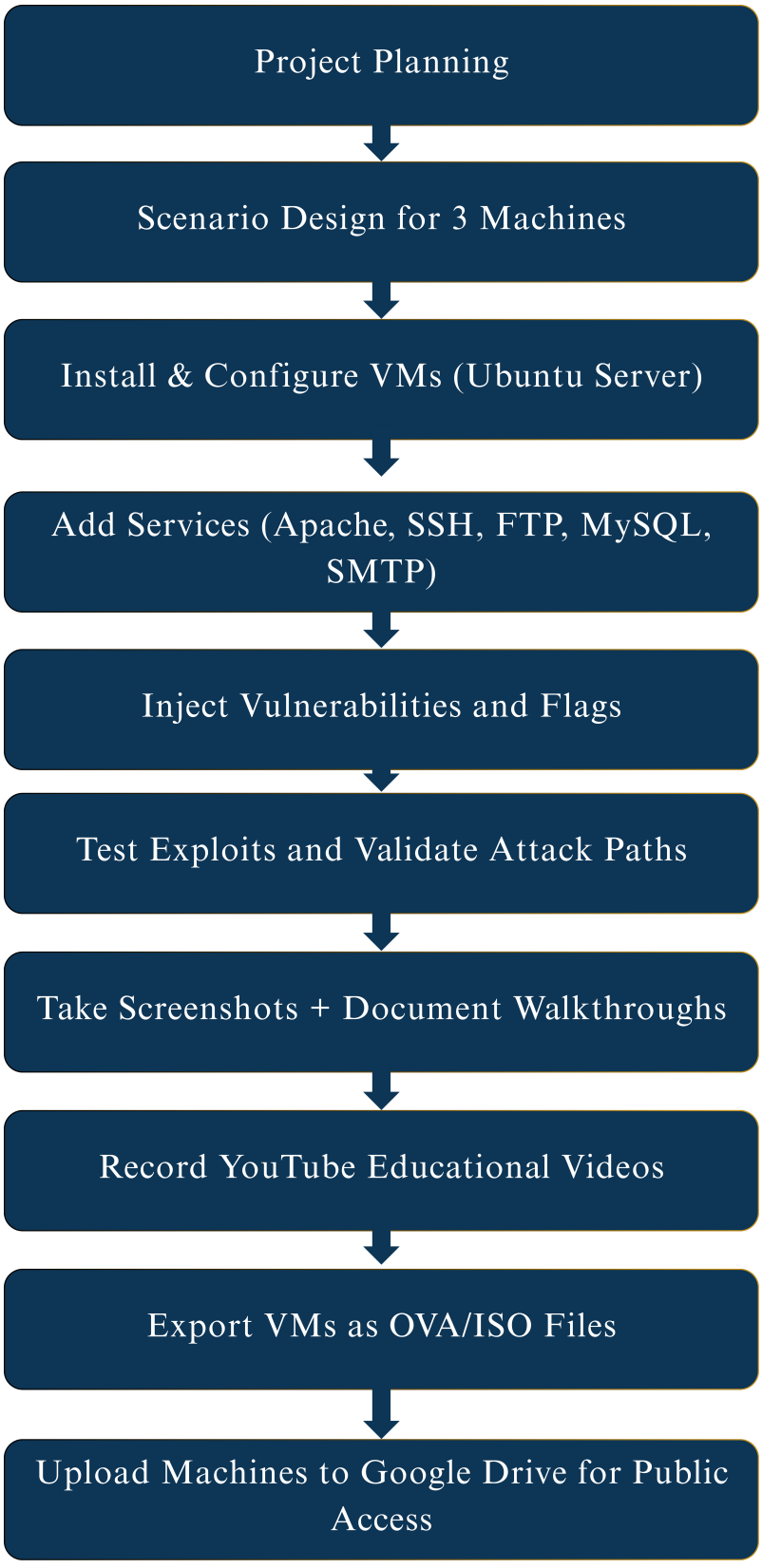

Figure 4.1: Overview of the Hack Lab Project Workflow

Figure 1 presents the complete workflow of the *Hack Lab* project, emphasizing the methodological flow from concept to public release. The process begins with a dedicated

planning phase, where educational goals are identified, and the appropriate tools, commands, and techniques are selected to suit cybersecurity beginners. Following that, three distinct Capture The Flag (CTF)-style scenarios were designed, each encapsulated within a separate virtual machine. These scenarios were carefully crafted to reflect real-world penetration testing situations while maintaining an accessible difficulty level for learners.

Once the scenarios were finalized, Ubuntu Server-based virtual machines were installed and configured manually, simulating real operating environments. Core services and protocols namely Apache (HTTP), SSH, FTP, MySQL, and SMTP were then deployed based on the requirements of each machine's scenario. Vulnerabilities and challenge flags were deliberately injected to guide learners through various techniques such as brute-force attacks, web enumeration, encryption cracking, and privilege escalation.

After constructing the machines, a complete penetration test was conducted on each one to validate the logic and integrity of the attack paths. Screenshots were taken at every critical stage to support learners with visual guidance. Subsequently, high-quality educational videos were recorded and published on YouTube to explain each exploitation step. Finally, the machines were exported as OVA/ISO files and uploaded to Google Drive, allowing learners around the world to access, download, and replicate the full learning experience.

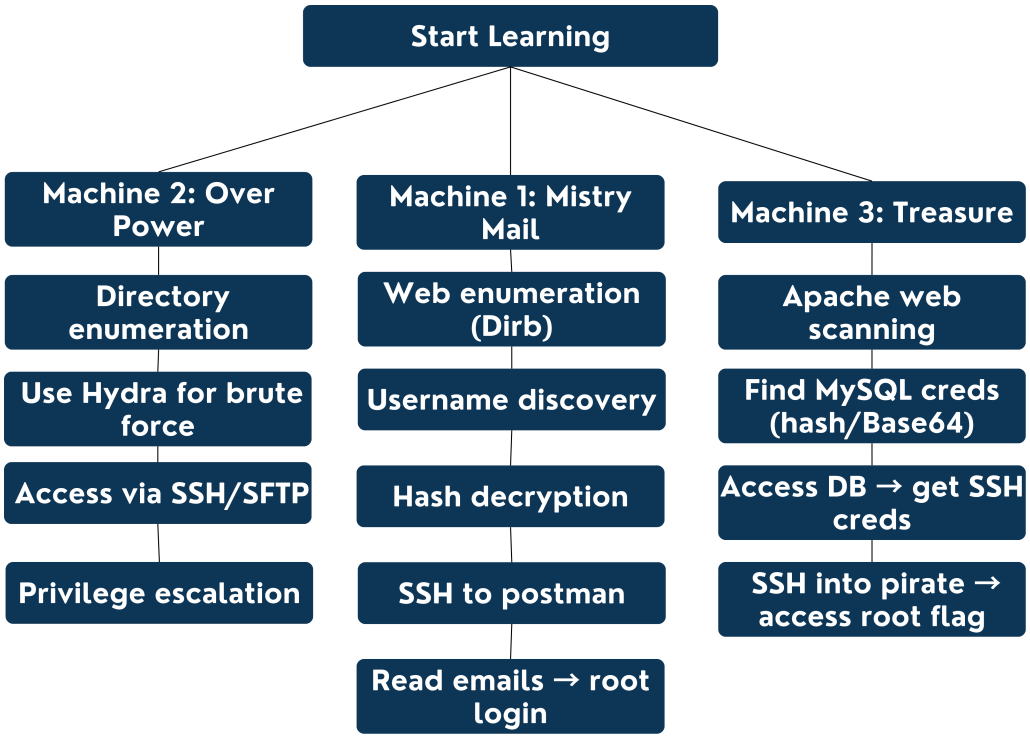

Figure 4.2: Overview of the Hack Lab Project Workflow

Figure 2 – Machine-Based Learning Path

The second diagram demonstrates the structured learning path provided by the three virtual machines that constitute the Hack Lab environment. These machines are ordered in a logical progression of difficulty, each focusing on different services and penetration testing skills. The visual layout highlights how learners begin with foundational enumeration tasks and gradually move toward more advanced exploitation and privilege escalation techniques.

Machine 1 – Mistry Mail introduces basic web enumeration using tools such as Dirb. Learners uncover usernames through source code inspection and hidden files, decrypt password hashes, and gain access to the system via SSH using a mail user account. This machine teaches core skills in information gathering, password cracking, and basic access control evaluation.

Machine 2 – Over Power increases complexity by challenging learners to perform directory enumeration and use Hydra for brute-force attacks. Once the correct credentials are obtained, access is granted via SSH or SFTP. The final goal is to escalate privileges and obtain root access. This machine reinforces authentication attack strategies and local exploitation techniques.

Machine 3 – Treasure integrates Apache scanning, cryptographic analysis, MySQL database queries, and Base64 decoding. Learners extract MySQL credentials from exposed files, gain access to the database, retrieve SSH login credentials, and eventually access the system as a root-level user. This scenario represents a more complete and realistic penetration testing flow.

Collectively, these machines offer a step-by-step, story-driven, hands-on journey that mirrors real-world ethical hacking practices in a safe and guided environment.

4.2 Project Planning and Role of Each Component

The planning phase of the *Hack Lab* project was highly intentional and aligned with our goal of creating a comprehensive, beginner-friendly cybersecurity training platform. This phase involved a series of strategic decisions related to the tools, platforms, protocols, and delivery mechanisms used throughout the project. Every element was selected not arbitrarily, but based on its direct contribution to both the technical and educational value of the final product. Below is a detailed explanation of each key decision and its role within the overall structure of the project.

4.2.1 1. Selection of Educational Tools and Core Commands

One of the earliest planning steps involved curating a list of essential penetration testing tools and Linux commands that beginners must learn. We prioritized tools that are widely used in professional environments and represent fundamental ethical hacking techniques. Among them were:

- **Nmap** for network scanning and reconnaissance.
- **Hydra** for brute-force attacks against login services.
- **Dirb** and **Gobuster** for directory enumeration.
- **Hashcat** and **John the Ripper** for password cracking.
- **Netcat** for reverse shell handling and basic networking.

These tools were chosen not just for their popularity, but because they represent the foundational knowledge upon which more advanced skills can be built. By learning how and when to use these tools, students begin to develop a mindset consistent with real-world penetration testing workflows.

4.2.2 2. Choice of Kali Linux as the Attacker Environment

Kali Linux was selected as the attacker's operating system due to its status as the industry-standard penetration testing distribution. Its pre-installed suite of over 600 cybersecurity tools allows learners to get started immediately without the need to install or configure additional software. This decision significantly lowered the barrier to entry and allowed students to focus on skill development rather than system setup. Moreover, it provided exposure to a platform used by professionals, thus enhancing the project's realism and relevance.

4.2.3 3. Inclusion of Five Real-World Protocols

The selected protocols Apache (HTTP), SSH, FTP, MySQL, and SMTP were chosen based on their frequent presence in corporate and enterprise environments. Each protocol served a unique instructional purpose:

- **Apache (HTTP)**: Used to create web-based challenges including hidden directories, source code analysis, and vulnerable file uploads.
- **SSH**: Allowed learners to simulate real remote access attacks and served as a stepping stone for privilege escalation.

- **FTP**: Offered experience with misconfigured file servers, which are common in many legacy systems.
- **MySQL**: Used to teach database exploitation, credential leaks, and SQL-based enumeration.
- **SMTP**: Simulated internal email systems, allowing students to explore credential exposure and message harvesting.

These services are widely deployed across networks, making them highly relevant to any cybersecurity role. Their inclusion ensured that learners were exposed to real attack surfaces found in organizations.

4.2.4 4. Scenario-Driven Machine Design

Three virtual machines were planned with increasing difficulty: *Mistry Mail*, *Over Power*, and *Treasure*. Each was built around a unique theme and included progressive challenges. The planning process included:

- Designing realistic, yet manageable exploitation paths.
- Mapping flags to specific skills (e.g., brute-force, enumeration, decryption).
- Creating story-based environments that engage learners and simulate real operations.

This design approach helped students progress in a structured manner, reinforcing skills through repetition and increasing complexity.

4.2.5 5. Flag and Vulnerability Placement

Flag placement was intentional and tied to learning objectives. Some flags required understanding of hash cracking, while others were located in hidden directories or encrypted files. This forced learners to apply multiple techniques and enhanced their problem-solving abilities.

Deliberate vulnerability planning ensured that each machine mimicked real-world systems while remaining safe and solvable. For example, weak login credentials were included to facilitate Hydra attacks, and exposed configuration files were used to simulate poor security practices.

4.2.6 6. YouTube for Educational Delivery

YouTube was chosen as the primary delivery platform for walkthrough videos due to its accessibility, scalability, and global reach. Publishing the content publicly ensured that:

- Learners from around the world could benefit without needing specialized access.
- Videos could include visual demonstrations, voiceover explanations, and terminal walkthroughs.
- Students could revisit the material at their own pace, enhancing retention.

This method allowed us to bridge the gap between theoretical concepts and practical application, making the learning experience dynamic and engaging.

7. Google Drive for Machine Distribution

To share the virtual machines with users worldwide, we chose Google Drive as the hosting platform for OVA and ISO files. This decision contributed to the project by:

- Providing a free, stable, and easily accessible hosting solution.
- Allowing bulk uploads and structured folder organization.
- Enabling instant updates and version control for distributed files.

Unlike proprietary or complex platforms, Google Drive ensured a frictionless download experience, making it ideal for beginner users.

Conclusion

Every component in the planning phase was selected and integrated based on its educational value, technical relevance, and ease of use. From choosing protocols that are prevalent in real environments to designing thematic machines and selecting global delivery platforms like YouTube and Google Drive, the planning ensured that the *Hack Lab* project was not only technically sound, but also highly accessible and educationally impactful. This deliberate and structured planning process laid the foundation for a complete, story-driven penetration testing experience tailored specifically for cybersecurity beginners.

4.3 services and protocols

4.3.0.1 Apache & HTTP

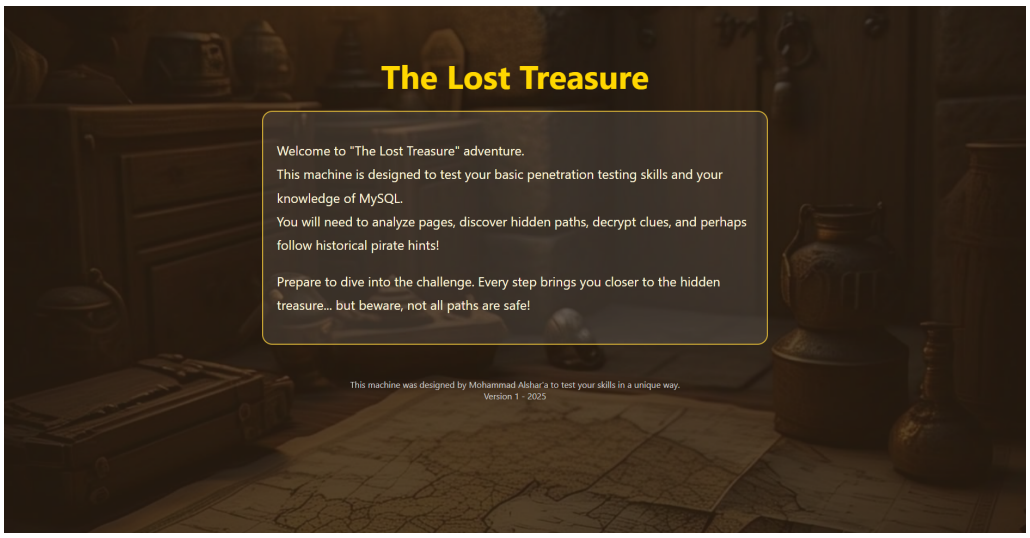

Figure 4.3: Apache service providing the HTTP interface for web-based challenges.

The Apache web server, combined with the HTTP protocol, played a foundational role in the construction of all virtual machines within the Hack Lab project. Its inclusion allowed the development of realistic web-facing services, enabling learners to engage in multiple types of reconnaissance, enumeration, and exploitation activities. Apache served as the front-facing layer of each machine, exposing directories, files, and web applications that simulate real-life vulnerable websites.

From a technical standpoint, Apache enabled us to build static and dynamic content that included login forms, hidden admin panels, and vulnerable file upload scripts. By placing sensitive content in specific web directories, we challenged learners to use tools like Dirb and Gobuster to discover resources not listed in the navigation. In several scenarios, the index page was purposefully vague or misleading, requiring students to inspect the source code, headers, and other elements for clues.

Apache also supported simulated content management systems and poorly configured file permission setups. This allowed learners to practice reading ‘robots.txt‘, analyzing backup files like ‘backup.zip‘, and identifying insecure file types left in publicly accessible directories. In one machine, the web server exposed a hashed password inside the source code of an HTML page. In another, Apache was misconfigured to allow the upload and execution of reverse shells.

From an educational perspective, using Apache introduced students to the idea that a web server is often the initial attack surface in penetration testing. It taught them how to approach web applications from an attacker’s mindset and how to leverage even the smallest exposure (like a comment in the HTML) into actionable insight. It also intro-

duced basic principles of web security such as directory traversal, lack of authentication, and unsafe file handling.

In terms of machine design, Apache shaped the structure and interaction of each VM. It provided an intuitive and user-friendly entry point for beginners, easing them into the challenge before transitioning them into more advanced techniques like SSH access or database exploitation. The visibility and interactivity of the HTTP interface made it easier to guide learners visually and conceptually.

Overall, Apache & HTTP were critical not only in simulating real-world environments, but also in reinforcing essential security principles. Their flexibility, ease of configuration, and realism made them ideal for a beginner-focused penetration testing lab. The inclusion of this service set the stage for most enumeration activities and served as the learners' first contact with each machine's internal design.

4.3.0.2 SSH

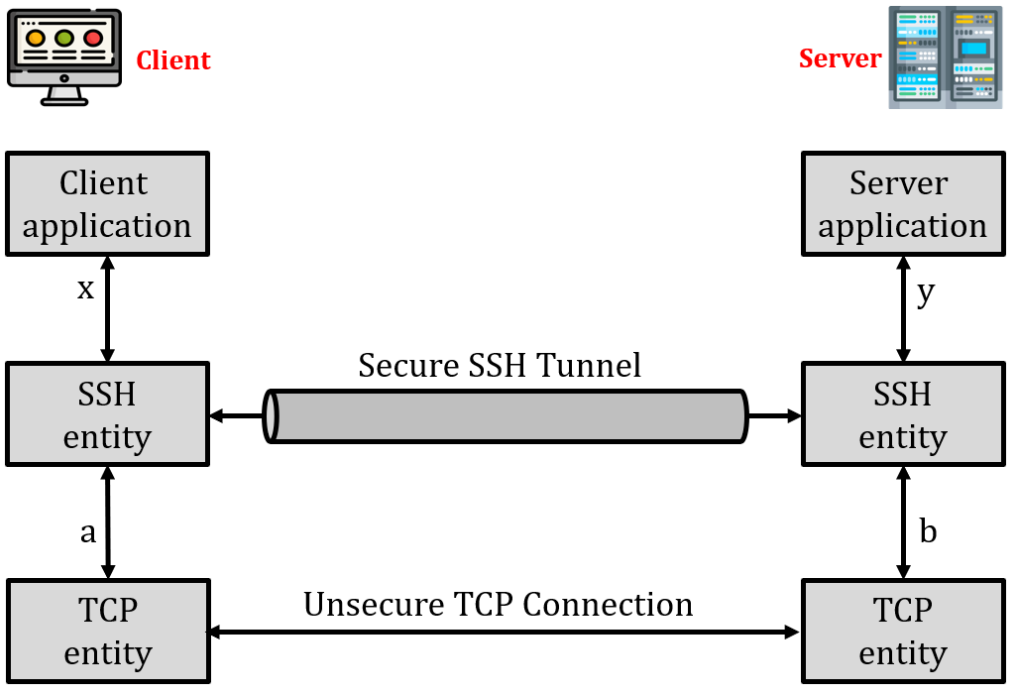

Figure 4.4: SSH service used for secure remote access and privilege escalation.

The Secure Shell (SSH) protocol was a critical element in the Hack Lab project, providing learners with experience in remote system access and command-line navigation. SSH served as the second stage in most exploitation paths, allowing students to pivot from web-based enumeration into deeper system-level interaction.

In our virtual machines, SSH was configured to accept login attempts with specific credentials that learners had to discover through reconnaissance. These credentials were often hidden within web files, hashed in user data, or embedded in base64/MD5 encoded

content. Once inside the system, students practiced navigating Linux file structures, reading user-owned files, and preparing for privilege escalation to root access.

SSH also introduced learners to common misconfigurations and attack vectors, such as weak passwords, reusable usernames, and lack of two-factor authentication. More advanced tasks included transferring files, checking running services, and analyzing system logs.

Through SSH, learners developed core Linux interaction skills while understanding the risks of exposed remote access. It added depth and realism to the lab, transforming the exercise from a simple CTF into a real-world simulation.

4.3.0.3 FTP

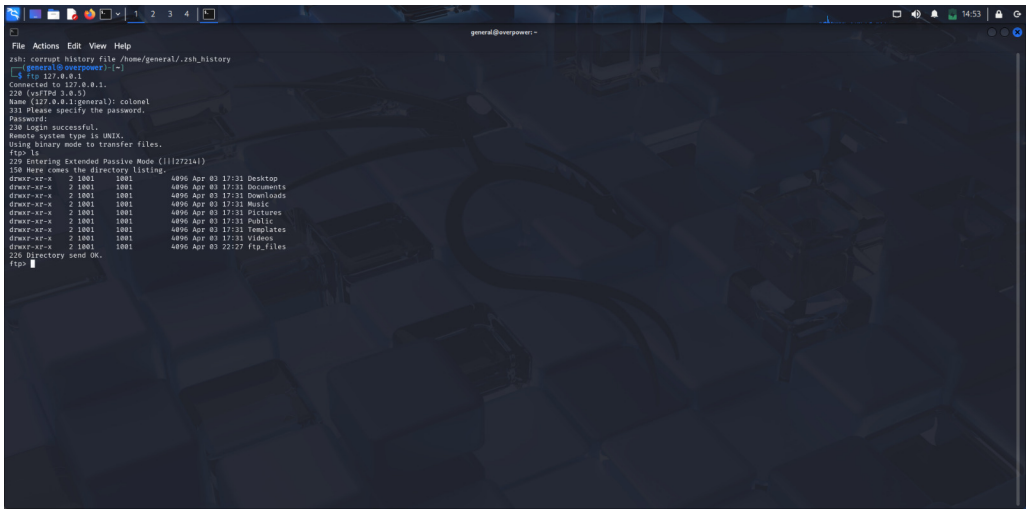

Figure 4.5: FTP service used to simulate unsecured file storage vulnerabilities.

The File Transfer Protocol (FTP) was implemented in the Hack Lab to expose students to common vulnerabilities found in legacy file-sharing systems. FTP is still present in many corporate networks, often misconfigured or lacking proper authentication. By including it in one of the machines, we enabled learners to understand how attackers exploit insecure file services to gain unauthorized access to sensitive data.

FTP served as an early-stage discovery point in the lab scenarios. In some machines, learners were able to connect to the FTP service anonymously or using default credentials that they uncovered through enumeration. Once connected, they explored directory structures, downloaded exposed files such as ‘creds.txt‘, and retrieved configuration files that hinted at deeper system vulnerabilities.

This protocol allowed students to practice using command-line FTP clients and tools like ‘ftp‘, ‘ncftp‘, and even ‘wget‘ for direct downloads. Files placed in the FTP directories often contained partial information, such as usernames, password hashes, or system notes, which students used to build a larger picture of the machine’s setup.

FTP also demonstrated the risks of leaving sensitive files in publicly accessible directories, especially when access controls are weak or nonexistent. From a pedagogical standpoint, it emphasized the importance of proper file permission settings, secure credential policies, and regular audits of network services.

By working with FTP, learners developed a clearer understanding of how something as seemingly harmless as a file server can become a gateway for deeper intrusion. It also helped reinforce the concept of chained exploitation using a minor leak (a file or password) to trigger a larger breach in the system.

4.3.0.4 SMTP

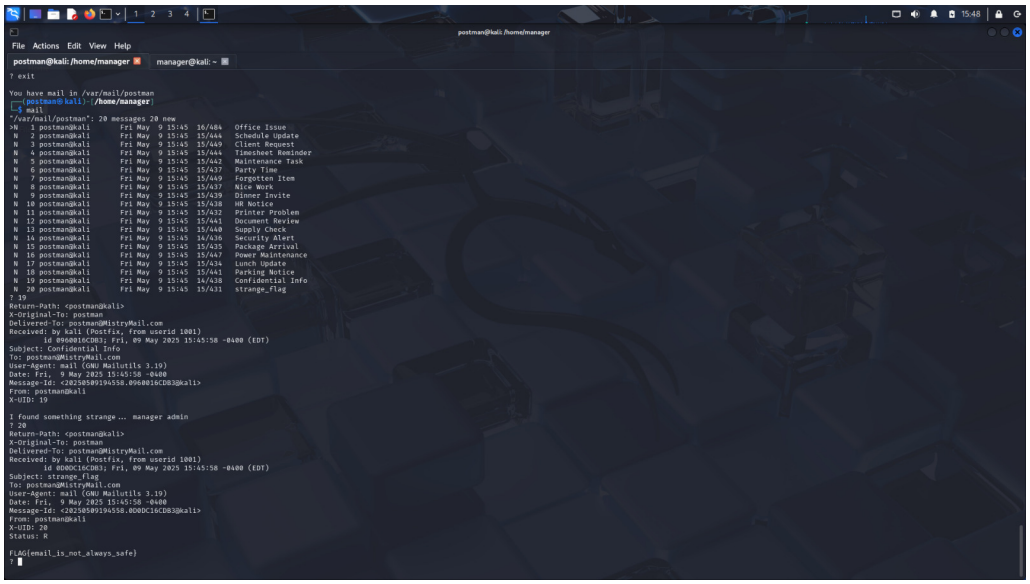

Figure 4.6: SMTP service used to simulate internal email communication and information leakage.

The Simple Mail Transfer Protocol (SMTP) was integrated into the Hack Lab project to simulate internal enterprise email systems and explore how poorly secured communications can expose sensitive data. This addition gave learners experience in identifying and extracting information through local services that are often overlooked.

In the lab environment, SMTP was configured to operate locally, allowing students to interact with stored or intercepted email messages after gaining access to a system. These messages often contained login credentials, partial hashes, usernames, or internal notes hinting at other weaknesses in the machine.

From a technical perspective, SMTP simulated common misconfigurations found in internal networks such as unencrypted messages, lack of authentication, and exposure of message content through accessible mail logs or mailboxes. By inspecting these messages using command-line tools or by browsing system directories, learners were able to uncover clues that would guide them toward privilege escalation or deeper system access.

This protocol also reinforced key learning outcomes related to information leakage, insider threat simulation, and lateral movement within a compromised environment. In particular, it illustrated how seemingly non-critical services can provide critical insights if improperly secured.

SMTP added depth and realism to the scenarios, and its use aligned with the objective of teaching learners to think like ethical hackers: always investigate all exposed services, no matter how irrelevant they may seem. It also introduced a soft layer of social engineering and contextual thinking, encouraging learners to connect textual information with system weaknesses.

4.3.0.5 MYSQL

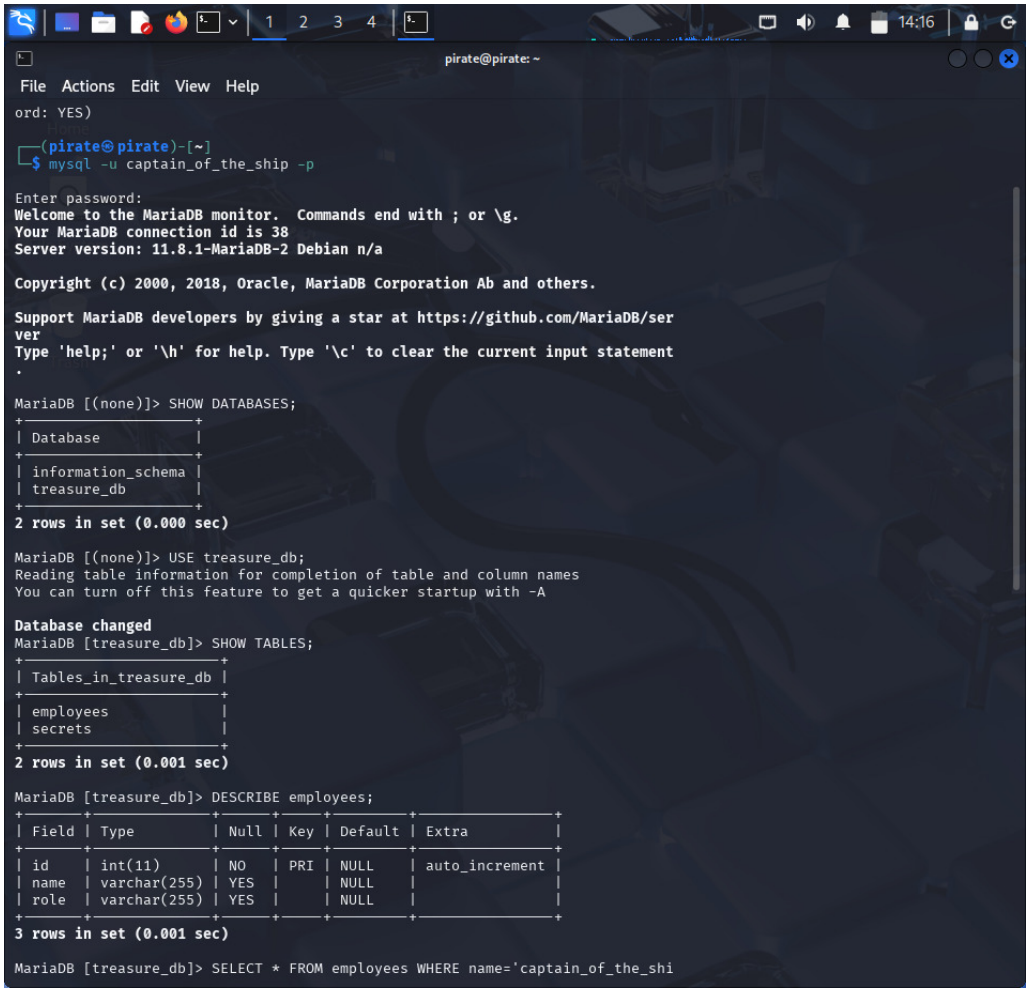

Figure 4.7: MySQL service used to simulate credential storage and database exploitation.

The MySQL database management system was introduced into the Hack Lab project to simulate common data storage vulnerabilities and to help students understand the risks associated with poorly secured or misconfigured databases. MySQL played a central role in one of the machines, providing a realistic simulation of data exposure through weak

credential storage and insufficient access controls.

In the lab scenario, learners had to first enumerate and decode credentials (e.g., through SHA1 or Base64) that were used to access the MySQL service. Once logged in, students were presented with a simulated employee database containing usernames, encrypted passwords, and even misleading or decoy entries. These data were intentionally crafted to encourage analytical thinking and careful inspection.

By engaging with MySQL, students were exposed to basic SQL commands and learned how to query tables, extract useful information, and detect patterns that may indicate privilege escalation opportunities. The database often included embedded clues such as Base64-encoded usernames or hints hidden in data fields that guided students to the next steps in the challenge.

MySQL also emphasized the importance of encryption and secure data management practices. Through hands-on interaction, learners saw how easily unencrypted or poorly encrypted data can be extracted and used to access deeper parts of a system.

The inclusion of MySQL added a new dimension to the Hack Lab experience by blending web-based and system-level exploitation with data-focused challenges. It provided a gateway between enumeration and system compromise, reinforcing the idea that databases are often the most valuable and vulnerable components in a target system.

4.3.0.6 SHA1 Encryption

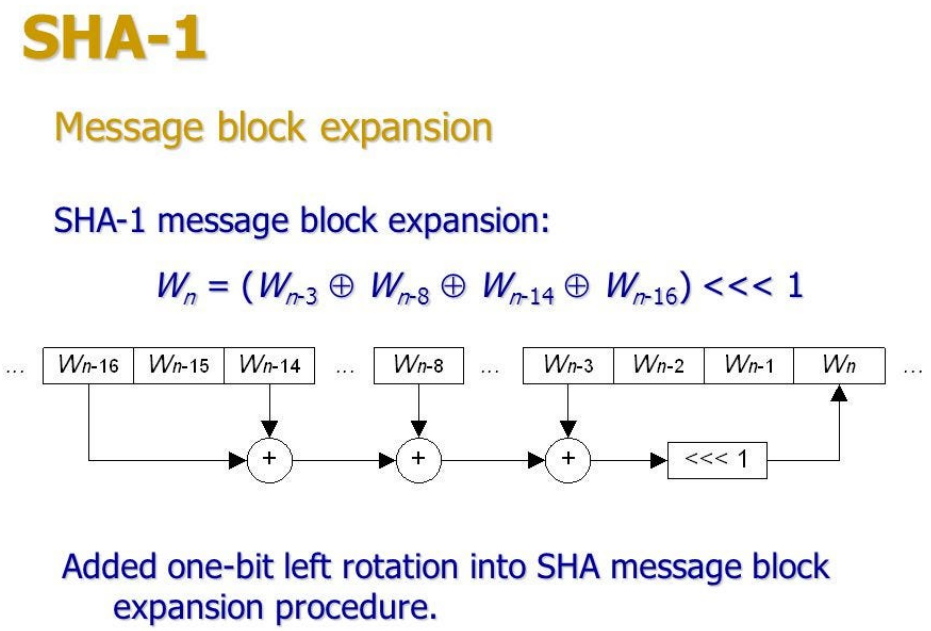

Figure 4.8: SHA1 encryption used to obscure credentials in the Hack Lab scenarios.

SHA1 (Secure Hash Algorithm 1) was one of the encryption techniques used in the Hack Lab project to simulate real-world credential protection and hashing challenges. Its inclusion in several machine scenarios introduced students to the concept of one-way hashing functions, their structure, and how they can be targeted during a penetration test.

In the lab, hashed passwords or usernames were placed in publicly accessible files or embedded within web page source code. Learners had to recognize the SHA1 format (typically a 40-character hexadecimal string), then use tools like Hashcat or online hash-cracking databases to try and recover the original values. This simulated the process of hash recognition, cracking, and applying recovered credentials in later exploitation stages.

Using SHA1 helped learners understand the limitations of outdated encryption techniques. It also taught them how attackers use wordlists, rainbow tables, and brute-force attacks to reverse weak hashes. By comparing SHA1 against stronger algorithms, students also gained insight into why modern systems are moving away from vulnerable hash types.

From an educational standpoint, SHA1 contributed significantly to the realism of the challenge. It created opportunities for cross-tool learning and reinforced the importance of strong password hashing practices in system design.

4.3.0.7 Base64 Encoding

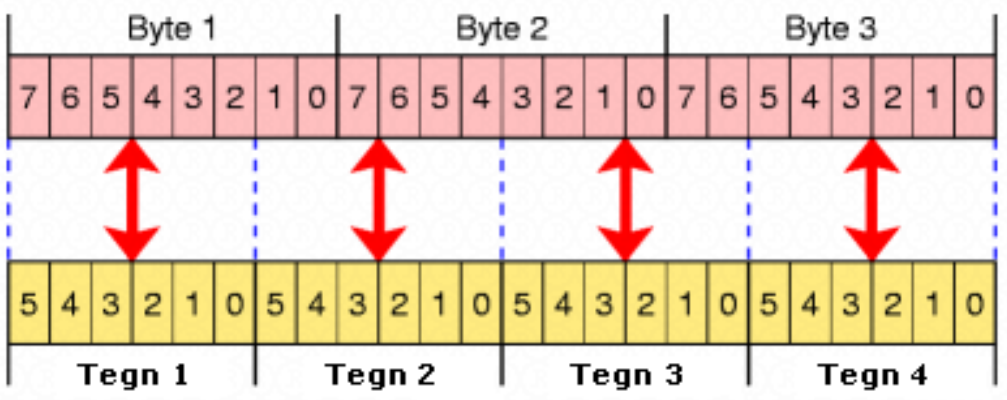

Figure 4.9: Base64 used to encode sensitive information within files and databases.

Base64 encoding was strategically used across various Hack Lab machines to obscure key pieces of information such as usernames, passwords, or instructional clues. While not an encryption method in itself, Base64 was included to teach students how basic encoding techniques can be used to mask information from casual inspection.

In some scenarios, learners encountered Base64 strings hidden inside HTML comments, FTP text files, or MySQL databases. They were required to recognize the pattern (typically ending with ‘==‘ or containing only alphanumeric characters and slashes) and use decoding tools like the Linux ‘base64‘ command, CyberChef, or online decoders.

Base64 served as a gateway to deeper challenge layers. It forced students to think critically about what data might be encoded and why it was obscured. In doing so, learners developed a sense of curiosity and pattern recognition, both essential in real-world cybersecurity operations.

The presence of Base64 also allowed instructors to hide layered clues within clues, adding depth to the lab without making it excessively difficult. It reinforced the idea that information is often present but requires the right mindset and tools to interpret.

4.3.0.8 MD5 Hashing

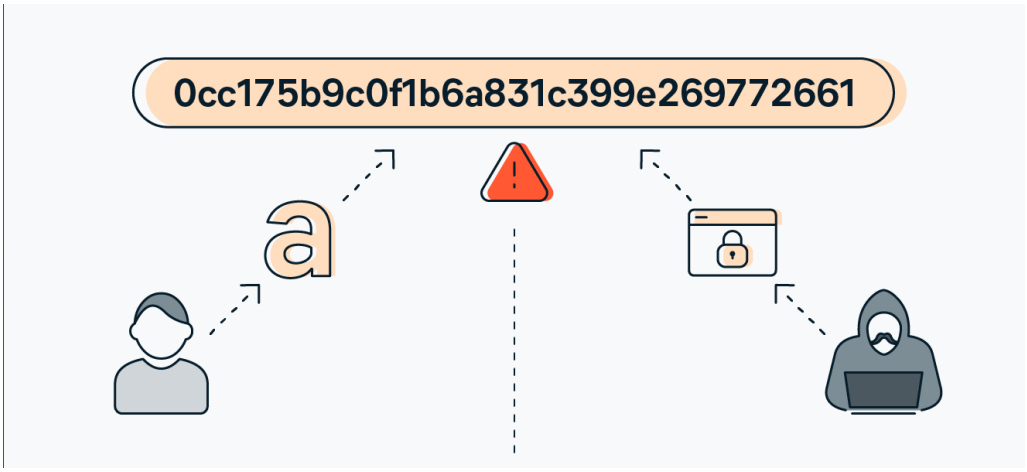

Figure 4.10: MD5 hashing used to simulate legacy password storage techniques.

MD5 (Message Digest Algorithm 5) was used in Hack Lab scenarios to mimic how older systems stored user credentials. Despite being widely considered insecure today due to vulnerabilities like hash collisions, MD5 is still commonly found in legacy applications and databases.

In the lab, MD5 hashes were embedded in public text files or exposed web pages. Students had to identify the hash format typically a 32-character hexadecimal string and use cracking tools or online hash databases to retrieve the plaintext values. This helped illustrate the process of converting encrypted data back into usable credentials.

MD5 was also used to highlight the difference between secure and insecure hashing algorithms. By comparing the ease of cracking MD5 hashes to more secure alternatives, learners gained a clear understanding of why modern cybersecurity practices discourage MD5.

Including MD5 in the lab environment gave students practical exposure to one of the most common hash types seen in capture-the-flag competitions, older databases, and forensic analysis tasks. It added realism and complexity without significantly increasing difficulty for beginners.

4.3.0.9 Kali Linux

Figure 4.11: Kali Linux as the primary attack environment used for testing and demonstration.

Kali Linux served as the primary operating system for conducting all penetration testing activities within the Hack Lab project. Its selection was based on its reputation as a professional-grade platform preloaded with hundreds of security tools. Using Kali Linux allowed us to simulate real-world ethical hacking scenarios in a consistent, efficient, and standardized environment.

From the planning phase, Kali Linux was chosen for its accessibility and its widespread use in both academic and industry settings. It significantly reduced the overhead for learners by eliminating the need to install tools manually. Instead, tools like Nmap, Hydra, Gobuster, Netcat, John the Ripper, Hashcat, and many others were readily available within the environment. This made it easier for students to focus on learning the techniques rather than worrying about setup or compatibility issues.

Throughout the lab, Kali Linux acted as the attacker's system, interacting with vulnerable virtual machines built specifically for the project. It enabled students to launch scans, brute-force logins, enumerate services, crack hashes, and deploy reverse shells. Students learned how to utilize terminal-based commands in a real Linux environment, developing skills in both navigation and scripting.

Another advantage of using Kali Linux was its visual familiarity. The interface (based on GNOME or XFCE) combined with terminal interaction offered an intuitive blend of usability and power. For learners new to Linux, this helped ease the transition and reduced frustration.

Pedagogically, Kali Linux contributed to a hands-on, immersive experience. By using the same tools employed by cybersecurity professionals, students were empowered to replicate real attack scenarios safely and ethically. This not only increased engage-

ment but also reinforced confidence as they saw how each tool connected to a specific vulnerability.

Overall, Kali Linux was more than just a platform it was the bridge that linked theoretical knowledge to technical execution. It transformed the Hack Lab from a static CTF into a dynamic, exploratory learning environment rooted in practical experience.

4.3.0.10 VirtualBox

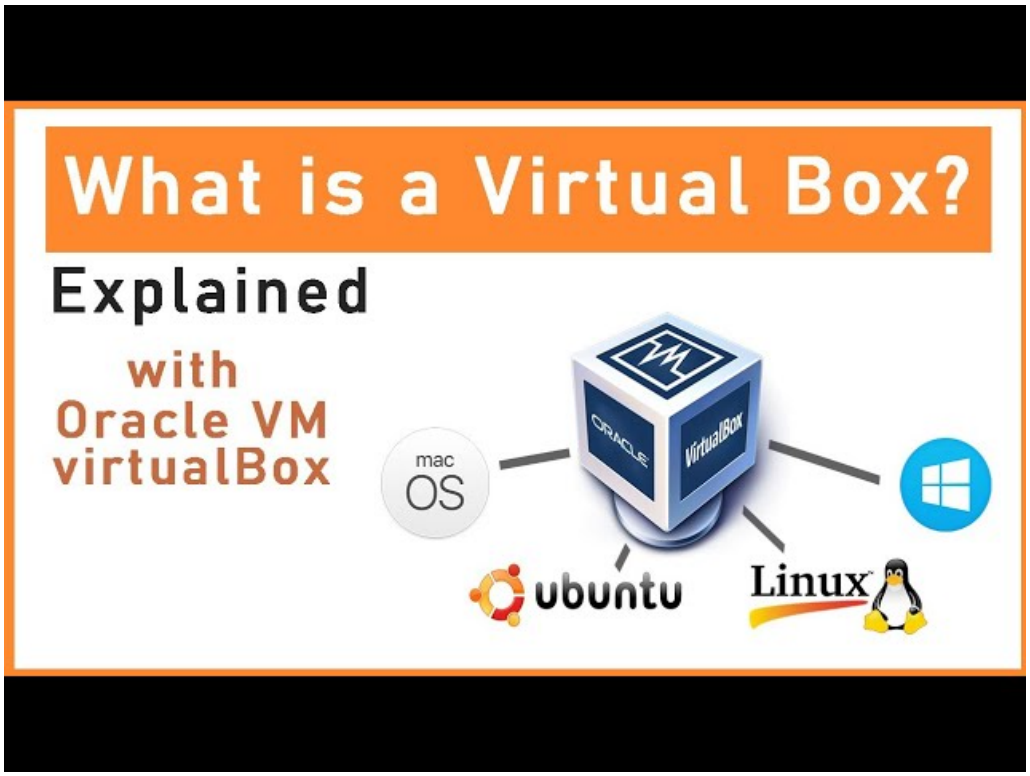

Figure 4.12: VirtualBox used to host and manage the Hack Lab virtual machines.

VirtualBox was used as the virtualization platform for creating, configuring, and deploying all virtual machines in the Hack Lab project. It served as the backbone infrastructure, allowing us to simulate isolated networked environments on a single host system. The choice of VirtualBox was driven by its open-source nature, cross-platform compatibility, and ease of use making it ideal for both development and student deployment.

Each of the three machines in the lab Mistry Mail, Over Power, and Treasure was built from a base Ubuntu Server image using VirtualBox. The tool allowed us to configure essential hardware settings (e.g., memory allocation, network adapter type) and create snapshots to easily revert to stable states during testing. This was crucial for debugging complex challenges and maintaining consistent environments for all learners.

VirtualBox's bridged and NAT networking modes were leveraged to simulate realistic interactions between the attacker (Kali Linux) and the target machines. This enabled full

penetration testing workflows, including port scanning, service enumeration, brute force attacks, and reverse shell sessions, without requiring access to external networks.

From a student perspective, VirtualBox offered a reliable and lightweight platform for running the lab locally. By exporting the machines as OVA files, we ensured learners could import them with minimal configuration. This improved accessibility and helped maintain uniformity across different setups.

Educationally, using VirtualBox allowed students to become familiar with virtualized infrastructure an essential concept in cybersecurity, ethical hacking, and system administration. It also exposed them to basic VM management tasks, such as allocating resources, starting/stopping VMs, and adjusting system configurations.

In summary, VirtualBox was a core enabler of the Hack Lab. It provided a stable, reproducible, and beginner-friendly environment for delivering complex cybersecurity scenarios safely and efficiently.

4.3.0.11 YouTube

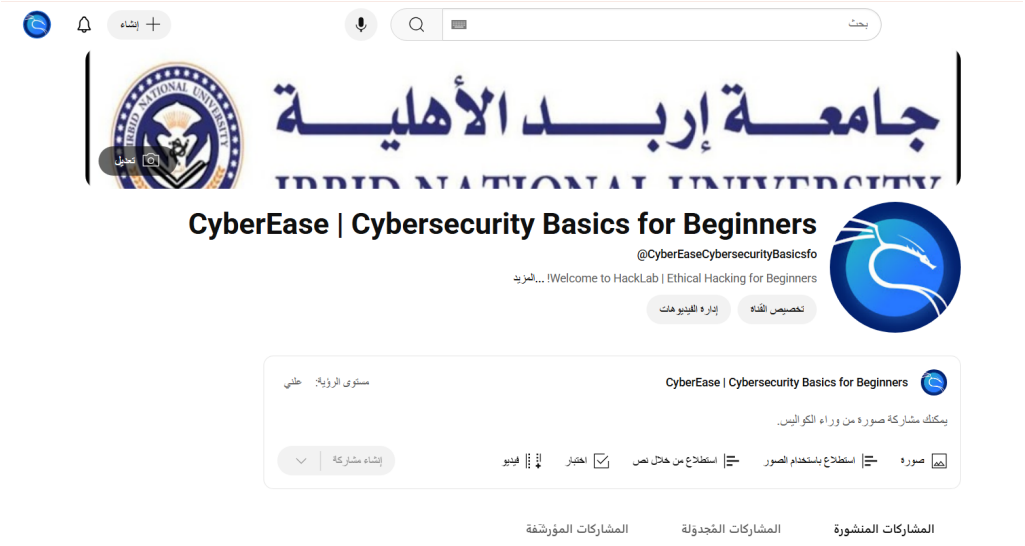

Figure 4.13: YouTube as the main platform for delivering educational walkthroughs.

YouTube played a pivotal role in the Hack Lab project by serving as the main channel for educational content delivery. Its global accessibility, ease of use, and support for high-quality video tutorials made it an ideal platform for explaining complex cybersecurity concepts to beginners. By integrating visual demonstrations with verbal explanations, YouTube helped transform technical material into engaging and understandable lessons.

Throughout the project, we produced a dedicated video for each virtual machine. These walkthroughs showcased the full attack chain from initial enumeration to privilege escalation while explaining every step and command in detail. The videos were designed with a beginner audience in mind, using clear narration, highlighted terminal commands,

and real-time problem solving. This approach helped bridge the gap between theoretical learning and practical execution.

In addition to full exploitation guides, short educational segments were also published to teach foundational Linux commands, penetration testing tools, and troubleshooting strategies. These videos enabled students to build confidence gradually, mastering basic concepts before tackling full machine challenges.

From a pedagogical perspective, YouTube allowed asynchronous learning, enabling students to study at their own pace and revisit material as needed. This flexibility proved especially beneficial for learners without access to formal cybersecurity training or structured lab environments.

Furthermore, YouTube's community features such as comments and likes provided feedback loops that helped us understand which concepts resonated most and where additional clarification was needed. This iterative refinement process ensured that the content evolved based on real learner needs.

In summary, YouTube was not just a content distribution platform, but an integral educational tool that expanded the reach and impact of the Hack Lab. It enabled dynamic, accessible, and scalable delivery of cybersecurity knowledge to learners across the globe.

4.3.0.12 Google Drive

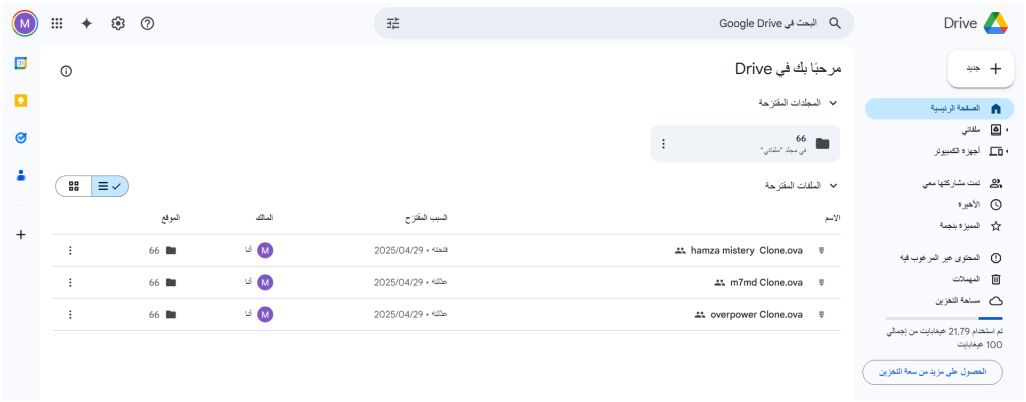

Figure 4.14: Google Drive used for distributing Hack Lab virtual machine files.

Google Drive was selected as the primary distribution platform for the virtual machines and educational materials created for the Hack Lab project. Its cloud-based architecture, user-friendly interface, and support for large file uploads made it the ideal solution for sharing content with a global audience, especially cybersecurity learners seeking practical experience.

The virtual machines, exported as OVA and ISO files, were uploaded to structured folders on Google Drive. Each folder was labeled clearly with the name of the corresponding machine (e.g., Mistry Mail, Over Power, Treasure) and contained the VM file,

a brief setup guide, and a direct link to the related YouTube walkthrough. This organized format made it easy for learners to download and start their lab environment within minutes.

Using Google Drive removed the technical and logistical barriers that often prevent students from accessing real-world learning tools. Unlike traditional distribution channels that may require paid subscriptions or specialized software, Google Drive offered free and secure access to all content. It also allowed us to update machine versions or fix errors instantly without needing to redistribute entire packages.

From an educational standpoint, Google Drive helped reinforce the concept of reproducible labs and consistent learning environments. By giving every student the same starting point, we ensured fairness and avoided discrepancies in configuration. This also simplified troubleshooting and made it easier to provide support to those facing setup issues.

Additionally, Google Drive's sharing features (public links, permission settings, and version history) allowed us to maintain control over content integrity while still keeping access open. Learners could revisit the drive anytime to re-download files, find updates, or share the lab with peers.

Overall, Google Drive contributed to the Hack Lab project by acting as a reliable, scalable, and accessible hub for lab distribution. It allowed us to extend the reach of our work beyond a classroom setting and into the hands of aspiring ethical hackers around the world.

Chapter 5

Comparison with Related works

The comparison between existing studies and the Hack Lab project highlights both alignment with established educational theories and key innovations introduced through this work. Collectively, the literature affirms the pedagogical, technical, and practical foundations of Hack Lab, while also illustrating how this project advances current approaches to cybersecurity education.

Several studies most notably those by bishop2021ctfskills, goodallGamifiedCTF, and ali2022ctf—consistently emphasize the value of **Capture the Flag (CTF)** environments as effective experiential learning tools. These works demonstrate that CTFs foster critical thinking, improve retention, and facilitate real-world application of cybersecurity skills. Hack Lab builds directly upon this evidence by embedding CTF-style scenarios into its three themed virtual machines. However, it further enhances this model by introducing **progressive difficulty**, **story-driven scenarios**, and **guided video walkthroughs**, offering a more structured and accessible learning pathway for beginners—something that most CTF platforms lack.

From a **technical relevance** standpoint, studies such as piantadosi2023apache and arce2010mysql validate Hack Lab's inclusion of real-world services like **Apache** and **MySQL**. Piantadosi's detailed examination of Apache vulnerabilities supports the educational use of this web server as a realistic attack surface, while Arce's analysis of MySQL authentication flaws aligns with the *Treasure* machine's database exploitation challenges. These integrations not only reflect real-world attack vectors but also ground the lab in contemporary security research, increasing its academic and technical credibility.

Protocol-specific research, including nasir2023ssh on **SSH**, alshaer2020smtp on **SMTP**, and asif2021ftp on **FTP**, further reinforce the educational validity of the services incorporated into Hack Lab. Each study identifies typical misconfigurations, legacy vulnerabilities, and the need for secure implementation practices. Hack Lab leverages these insights to design challenges that simulate weak or improperly configured protocols—thus training students to recognize and ethically exploit such flaws. This approach bridges the gap between theoretical understanding and practical skill, a goal repeatedly emphasized

across the reviewed literature.

A particularly distinct contribution of Hack Lab lies in its **integration of open-access platforms** like **YouTube** and **Google Drive** for resource delivery. While most of the related works focus on learning environments within institutional or competitive settings, Hack Lab democratizes cybersecurity education by removing both financial and technical barriers. Learners anywhere can access virtual machines, follow step-by-step video guides, and replicate real-world attacks without needing prior experience. This **accessibility-first approach** represents a novel enhancement that broadens participation and supports underrepresented or self-taught learners.

gupta2022linux's endorsement of **Linux as the ideal cybersecurity training platform** directly supports Hack Lab's use of **Kali Linux** as the attacker environment. The project's choice to simulate all attacks from a Kali-based system aligns with professional practice and ensures that students gain experience in the actual tools and workflows used in industry. Hack Lab does not merely replicate the findings of previous works it **synthesizes their strengths**, **addresses their gaps**, and **extends their educational reach**. By integrating real-world protocols, validated pedagogical models, and open-access delivery, the project stands as a practical, research-supported, and impactful solution for beginner-level cybersecurity training.

Table 5.1: Comparison Between Related Works and Hack Lab Project

Study	**Study Purpose**	**Educational Impact**	**Technical Relevance**	**Hack Lab Integration**
Piantadosi et al. (2023)	Analyze Apache/Tomcat vulnerability handling.	Shows the value of exposing learners to real-world software flaws.	Focus on open-source server security life cycle.	Apache-based challenges replicate known vulnerability patterns for learning.
Bishop et al. (2021)	Evaluate skills developed through CTFs.	Confirms experiential learning strengthens critical thinking.	Connects CTF challenges to academic curricula.	Hack Lab uses CTF structure aligned with real learning goals.

Continued on next page

Study	Study Purpose	Educational Impact	Technical Relevance	Hack Lab Integration
Goodall et al. (n.d.)	Study student motivation in CTF learning.	Shows CTFs enhance engagement and persistence.	Uses behavioral learning models for gamified learning.	Hack Lab motivates learners with story-driven, progressive CTFs.
Nasir and Islam (2023)	Examine SSH vulnerabilities in insecure networks.	Supports teaching secure remote access configuration.	Highlights risks like brute-force attacks and outdated encryption.	SSH attack paths simulate weak credentials and remote access issues.
Arce et al. (2010)	Analyze MySQL authentication protocol flaws.	Emphasizes the need for secure database access.	Demonstrates password recovery and session hijacking.	Treasure VM includes database exploitation and encoded credential recovery.
Ali and Noor (2022)	Compare challenge-based learning with lectures.	Demonstrates CTFs yield deeper understanding and longer retention.	Supports practice-driven training methods.	Hack Lab uses practical CTFs to replace passive theory.
Al-Shaer and Hamed (2020)	Analyze SMTP vulnerabilities and defenses.	Promotes awareness of internal data leakage threats.	Highlights open relays, spoofing, and insecure mail config.	Mistry Mail VM includes SMTP-based email leaks and user notes.
Asif and Nwankwo (2021)	Review FTP risks and secure alternatives.	Advocates for teaching legacy protocol weaknesses.	Cleartext logins and anonymous access emphasized.	Over Power VM includes FTP access and file enumeration challenges.

Continued on next page

Study	Study Purpose	Educational Impact	Technical Relevance	Hack Lab Integration
Gupta and Sharma (2022)	Justify Linux in cybersecurity education.	Supports Linux for hands-on security training.	Shows Linux enables better tool access and system control.	Hack Lab uses Kali Linux as standard attacker environment.

Chapter 6

Results and Discussion

6.1 Internal Execution and Technical Outcomes

The Hack Lab project was internally tested in a controlled environment to ensure that each virtual machine performed as designed and supported the intended learning objectives. These machines—**Mistry Mail**, **Over Power**, and **Treasure**—were constructed to simulate real-world services and security flaws across different stages of the penetration testing process. Each machine was deployed within VirtualBox, configured with appropriate network settings, and subjected to a complete exploitation process using a Kali Linux attacker system.

The tests confirmed that all essential services, including Apache, SSH, FTP, MySQL, and SMTP, functioned correctly within their scenarios. The logical flow of each attack path was preserved across restarts, and the machines remained stable throughout execution. At each stage, services responded predictably to scanning, enumeration, and exploitation, validating the structural integrity of each challenge.

6.2 Service Functionality and Stability Across Machines

To assess the integrity and scope of the simulated environments, each service integrated into the machines was evaluated in terms of operational status, exploitability, and relevance to the overall learning outcome. Table **??** provides a breakdown of the key services tested in each machine.

Table 6.1: Service Integration Across Hack Lab Virtual Machines

Service / Protocol	**Mistry Mail**	**Over Power**	**Treasure**	**Primary Educational Focus**
Apache (HTTP)	Available	Available	Available	Web enumeration, file discovery, and initial reconnaissance
SSH	Available	Available	Available	Secure remote access and privilege escalation
FTP	Not Available	Available	Not Available	Legacy file sharing, weak authentication, and brute-force attack simulation
SMTP	Available	Not Available	Not Available	Internal mail leakage, information gathering post-exploitation
MySQL	Not Available	Not Available	Available	Database exploitation and Base64-encoded credential recovery

Table 6.1 presents a detailed overview of the services implemented across the three virtual machines in the Hack Lab environment. Apache and SSH were integrated into all scenarios as foundational components for web-based enumeration and remote access respectively. FTP was uniquely featured in the Over Power machine to simulate unsecured file sharing and credential exposure. SMTP was configured only in the Mistry Mail machine to introduce concepts related to email leakage and post-exploitation discovery. The Treasure machine included MySQL to simulate credential storage and database exploitation challenges. This structured distribution of services across machines allows each scenario

to emphasize distinct aspects of penetration testing, offering learners a comprehensive exposure to commonly encountered vulnerabilities in real-world systems.

6.3 Exploitation Path Completion and Flag Validation

Each machine was designed to contain a sequence of vulnerable components leading to a final flag, typically placed in the root user's home directory. Internal testing ensured that every step in the exploitation process could be completed using standard tools available in Kali Linux. Table 6.2 outlines the high-level execution flow and success criteria for each machine.

Table 6.2: Exploitation Flow Summary and Results

Machine	**Entry Point**	**Exploitation Chain**	**Root Access**	**All Flags**
Mistry Mail	Apache + Dirb	HTML Comments → Hash → SSH → Mail → Root	Yes	Yes
Over Power	FTP + Hydra	Anonymous Access → Credential Extraction → SSH → Root	Yes	Yes
Treasure	Apache + MySQL	Dirb → Base64 → DB Query → SSH → Root	Yes	Yes

Table 6.2 outlines the full exploitation paths tested in each virtual machine. All three environments were successfully compromised using a step-by-step logical sequence that began with initial enumeration and concluded with privilege escalation. Mistry Mail followed a path beginning with HTML comment inspection and hash cracking, followed by SSH access and local email discovery. Over Power emphasized brute-force login via Hydra and FTP misconfigurations, while Treasure incorporated multi-stage logic involving directory enumeration, Base64 decoding, MySQL credential extraction, and SSH chaining. The successful retrieval of root access and all embedded flags in each case confirms the robustness and completeness of the scenario design.

6.4 Tool Performance and Integration Validation

A core component of Hack Lab was the strategic integration of real-world tools in realistic scenarios. These tools were tested for operational compatibility, effectiveness in exploitation, and alignment with the designed vulnerabilities. Table 6.3 summarizes tool usage and performance across machines.

Table 6.3: Tool Usage Across Hack Lab Scenarios

Tool	**Usage Context**	**Machines Used In**	**Outcome**
Nmap	Port scanning and service discovery	All Machines	Accurate service enumeration
Hydra	Brute-force login	Over Power	Successfully cracked SSH/FTP login
Gobuster/Dirb	Directory enumeration	Mistry Mail, Treasure	Located hidden directories and files
Hashcat	Password hash cracking	Mistry Mail, Treasure	Cracked SHA1 and MD5 values
Netcat	Reverse shell management	Over Power, Treasure	Established post-exploitation shells
MySQL Client	Database enumeration	Treasure	Retrieved encoded credentials

Table 6.3 summarizes how key penetration testing tools were applied across the three machines. Each tool was selected for its relevance to the service configurations and the attack logic of the machine. Nmap was used universally for initial reconnaissance. Hydra was employed in Over Power for brute-forcing FTP and SSH credentials. Gobuster and Dirb were crucial in locating hidden directories within Mistry Mail and Treasure. Hashcat was used to recover credentials from SHA1 and MD5 hashes, while Netcat facilitated reverse shell sessions

6.5 Discussion and Interpretation

The results obtained from internal testing affirm the integrity and practical applicability of the Hack Lab environment. The machines were intentionally designed to simulate realistic service configurations and vulnerabilities commonly found in both legacy and modern infrastructure. The successful execution of full exploitation paths—from enumeration to privilege escalation—demonstrates that the lab provides a coherent and logically progressive learning experience.

A notable design outcome is the consistency in service behavior and tool response. Each protocol used adhered to its expected behavior under penetration testing, with no anomalies or unpredictable results. This ensures that learners interacting with the lab will experience reproducible and reliable behavior.

In terms of exploitation logic, each scenario guided the attacker through a distinct series of actions that reinforced specific skills. Mistry Mail emphasized basic enumeration and email inspection, Over Power highlighted brute-force attacks and FTP misconfigurations, while Treasure provided a complete attack chain involving encoding, database interaction, and lateral movement.

Furthermore, the use of multiple encoding and hashing schemes such as Base64, SHA1, and MD5 introduced an additional analytical layer without exceeding the expected difficulty level. Each encoded or hashed element was solvable using standard techniques, reinforcing the importance of using the correct tools and developing procedural thinking.

6.6 Conclusion

The internal results of the Hack Lab project confirm that the environment meets its technical and educational objectives. Each virtual machine performed reliably under testing conditions, and the services functioned according to the scenario design. The embedded challenges were solvable, logically ordered, and reinforced essential penetration testing concepts. The strategic use of real-world tools in realistic scenarios, combined with stable virtualization and clear flag progression, positions Hack Lab as a robust foundation for beginner-level cybersecurity training.

Bibliography

Adams, K. and Michaels, D. (2020). Red team vs. blue team simulations for beginner cybersecurity learners. *Cybersecurity Education Review*.

Ahmed, N. and Rahman, T. (2022). The effectiveness of youtube as a learning platform for cybersecurity tutorials. *Journal of Online Education*.

Al-Shaer, E. and Hamed, M. (2020). Smtp security vulnerabilities and their impact on email integrity. *IEEE Transactions on Network and Service Management*, 15(3):1043–1056.

Ali, J. A. and Noor, N. A. (2022). Intensifying practical based learning of penetration testing using ctf. ResearchGate.

Ali, M. and Omar, H. (2023). Facilitating cybersecurity learning via google drive-hosted labs. *Online Cybersecurity Education Journal*.

Alqahtani, A. and Alghamdi, S. (2021). Enhancing cybersecurity education through capture-the-flag competitions. *IEEE Access*, 9:122211–122223.

Arce, I., Kargieman, E., Richarte, G., Sarraute, C., and Waissbein, A. (2010). An attack on mysql's login protocol. *arXiv preprint arXiv:1006.2411*.

Asif, L. M. and Nwankwo, T. T. (2021). Ftp protocol analysis and security enhancements for modern infrastructure. *Journal of Network Security*, 7(2).

Barakat, N. and Sadiq, S. (2021). Gamification in cybersecurity education: Ctfs as a motivational tool. *Education and Information Technologies*, 26.

Barnes, T. and Hu, S. (2021). Analyzing youtube as a delivery platform for lab-based cybersecurity education. *Multimedia in Education*.

Bishop, M. J., Mulnix, C., and Rader, R. (2021). Cybersecurity knowledge and skills taught in capture the flag challenges. *arXiv preprint arXiv:2101.01421*.

Brown, E. and Thompson, G. (2021). Legacy password cracking with john the ripper: Tools and techniques. *Cyber Forensics Review*.

Cho, J. and Kim, T. (2020). Network reconnaissance using arp tools: A case study of netdiscover. *Journal of Network Engineering*.

Clark, J. and Lin, P. (2021). Penetration testing as a teaching strategy in cybersecurity programs. *Journal of Cyber Pedagogy*.

CTFd Team (2023). Ctfd: Open-source platform for ctf challenges.

Farouk, M. and Saleh, R. (2022). Ssh authentication risks and secure practices for learners. *Cyber Infrastructure Review*.

Fernandez, A. and Santos, D. (2021). Understanding database threats: A penetration testing perspective on mysql. *Database Security Review*.

GNU Project (2023). Gnu wget manual.

Gobuster Project (2022). Gobuster - directory brute-force tool.

Gomez, J. and Smith, T. (2020). Distribution of virtual machines through open cloud storage: A comparative study. *Cloud-Based Education Journal*.

Gomez, L. and Hernandez, F. (2023). Using virtual machines to simulate attack scenarios in cybersecurity training. *Cyber Lab Pedagogy*.

Goodall, J. R., Lutters, J. L., and Komlodi, A. (n.d.). Gamified cybersecurity learning in capture-the-flag competitions through the lens of the information search process. BYU CSRL.

Gupta, M. K. and Sharma, R. Why linux is the preferred environment for penetration testing education. *International Journal of Cybersecurity Education*, 10(1).

Hammoud, Y. and Sleiman, F. (2023). The role of bash scripting in managing penetration test environments. *Linux Security Pedagogy*.

Hassan, T. and Fawaz, M. (2023). Linux command-line mastery for ethical hacking labs. *Cybersecurity Training Series*.

Islam, M. and Khan, A. (2020). Smtp configuration flaws and their exploitation in lab environments. *Network Security Learning Journal*.

Jones, A. and Martin, L. (2021). Why offline labs are still effective in a cloud-first era. *Digital Education Research*.

Li, C. and Yang, Z. (2023). A structured approach to using metasploit in classroom labs. *Cybersecurity Instructional Methods*.

Liu, H. and Zhao, Y. (2023). Utilizing google drive for lab distribution in cybersecurity education. *International Journal of Digital Learning*.

Liu, Z. and Chen, M. (2023). Exploring smtp vulnerabilities in simulated environments. *Cybersecurity Simulation Journal*.

Malik, R. and Saleh, Y. (2022). Exploiting ftp misconfigurations in penetration testing labs. *Information Security Practice Journal*.

Musa, H. and Omer, A. (2021). Establishing a baseline skillset for entry-level penetration testers. *Cybersecurity Curriculum Journal*.

Nasir, A. A. and Islam, M. S. (2023). Understanding the limitations of secure shell (ssh) in wireless network security. *ResearchGate*.

Nguyen, V. and Tran, B. (2022). Learning outcomes from capture-the-flag activities in undergraduate courses. *Education in Computing*, 14.

Offensive Security (2023). Kali linux documentation.

Openwall Project (2022). John the ripper: Usage and tips.

Park, J. and Lee, H. (2021). Brute-force attacks and mitigations in ssh: A study using hydra. *Computer Security Journal*, 15(3).

Peterson, K. and Delgado, J. (2023). Building educational scenarios with simulated vulnerabilities. *Journal of Cyber Education Development*.

Piantadosi, V., Scalabrino, S., and Oliveto, R. (2023). Fixing of security vulnerabilities in open source projects: A case study of apache http server and apache tomcat. *IEEE Transactions on Software Engineering*, 49(4):1580–1598.

Rahman, M. and Mizan, M. (2020). Gamified learning in cybersecurity education: A survey and future directions. *Journal of Educational Technology*, 17(4).

Rapid7 Team (2022). Metasploit framework documentation.

Salem, R. and Zayed, A. (2022). A comparative study of hash cracking tools: Hashcat and john the ripper. *International Journal of Security Tools*, 8(1).

Security Research Group (2021). Netcat - the swiss army knife of networking.

Smith, R. and Allen, K. (2020). A practical guide to cracking hashes using hashcat. *International Journal of Cybersecurity Practice*.

Stöcklin, T. T. (2022). *Evaluating SSH for modern deployments*. PhD thesis, Thesis, Noroff University College.

Tarek, M. and Youssef, S. (2021). Directory enumeration strategies using dirb and gobuster. *Journal of Applied InfoSec*.

Tian, Y. and Zhang, J. (2020). Security analysis of legacy protocols: The case of ftp. *Journal of Network Protocols*.

Wang, L. and Xu, Y. (2022). An evaluation of port scanning tools: A case study of nmap. *International Journal of Cybersecurity*, 11(2).

Zhang, L. and Wei, J. (2021). Common misconfigurations in apache web servers and how to exploit them. *Web Application Security Review*.

Zhou, L. and Kang, M. (2021). Reverse shell exploitation in educational penetration testing environments. *Journal of Ethical Hacking*, 5(2).