**DEVELOPMENT OF A WEB-BASED PENETRATION TESTING FRAMEWORK (PEN-T) FOR SECURITY ASSESSMENTS**

**OLARINDE OYEWOLE TIMOTHY**

**125/21/1/0029**

**FINAL YEAR PROJECT SUBMITTED IN THE DEPARTMENT OF COMPUTER SCIENCES**

**FACULTY OF NATURAL AND APPLIED SCIENCES**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE (B. Sc.) HONS IN CYBER SECURITY**

**ABIOLA AJIMOBI TECHNICAL UNIVERSITY**

**IBADAN**

**NIGERIA**

**MARCH, 2025**

# CERTIFICATION

This is to certify that the Final Year Project titled, **DEVELOPMENT OF A WEB-BASED PENETRATION TESTING FRAMEWORK FOR SECURITY ASSESSMENTS**, prepared and submitted by **OLARINDE, OYEWOLE TIMOTHY** with Matriculation Number **125/21/1/0029** in the Department of Computer Sciences, in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE HONS (Cyber Security)** is hereby accepted.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Signature and Date)

**Mr. Akano**

**Supervisor**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Signature and Date)

**Akinsola J.E.T Ph.D.**

**Head of Department, Computer Sciences**

# DEDICATION

This project is dedicated to the Almighty God, whose unfailing grace and direction have provided sustenance throughout my years in this great institution. I also wish to appreciate my parents, Mr. and Mrs. Olarinde, from the depth of my heart for their efforts in various areas prayerful, financial, emotional, moral, and logistical without which this success would have remained an illusion.

# ACKNOWLEDGMENT

My deepest gratitude goes to Almighty God for His unwavering grace, guidance, and strength throughout this phase of learning and growth. I am profoundly grateful to my parents, Mr. and Mrs. Olarinde, and my entire family for their relentless prayers, support, and encouragement spiritually, financially, and emotionally.

I sincerely appreciate the **Josephus Foundation** for their generous scholarship, which enabled my enrollment at **Abiola Ajimobi University, Ibadan**. My heartfelt gratitude extends to my supervisor, **Mr. Akano**, for his invaluable guidance, instructions, and patience throughout this journey.

I also wish to express my appreciation to my lecturers, **Professor Okesola J. O., Professor Olajubu E. O., Dr. Akinsola J. E. T., Mr. Awoseyi A. A., Dr. Abimbola R. O., Dr. Akinyemi, Mr. Akano O. A., Mrs. Olayinka T. O., and Mr. Fafowora B. O.**, for their dedication and impactful teachings that shaped my academic experience.

A special acknowledgment goes to my **wonderful friends and coursemates** for their unwavering support and camaraderie throughout this program. I am particularly grateful to **Akandeolu Erimi, Akanbi Ooreoluwa, Olatunji Goodness, and Olaniyi Mojubaoluwa** your friendship has been a source of inspiration and strength.

May the Almighty bless, guide, and reward you all abundantly.

# ABSTRACT

Standard approaches to penetration testing usually require manual installation of various security testing tools, which can be time-consuming, inconsistent, and hard to optimize for security experts as well as for enthusiasts. Modern systems lack the provision of single, web-based platform that unifies the different steps of penetration testing, including reconnaissance, scanning, exploitation, and reporting. A web-based penetration testing framework that simplifies and streamlines the security evaluation processes to provide a methodically organized, interactive, and efficient testing platform is proposed in the current research.

The project focuses on creating a modular platform for penetration testing with inspiration drawn from the OSINT Framework, especially with regard to ethical hacking and vulnerability analysis. It is built as a web app accompanied with an additional command-line interface (CLI) for better user experience. Python is the language of the backend using security focused libraries like Scapy, Nmap, Requests, and the Metasploit RPC API. However, the front-end uses React.js. It is built following the Open Web Application Security Project (OWASP) methodology with vulnerability analysis and simulation as per a systematic method.

The major features include automated discovery, network and web-based scanning, penetration testing, and comprehensive reporting with data visualization capabilities. Both SQLite and MySQL are supported as the database behind the application for effective storage and retrieval of logs and reports during testing. Role-based authentication mechanisms are implemented in the web application for security against unauthorized access, while the framework is cloud-based for remote access and collaboration purposes among cyber security experts.

The framework was tested through the running of simulated attack tests with the aim of demonstrating its effectiveness in detecting weaknesses, reporting, and performing penetration testing protocols. Key performance factors show that the framework reduces manual work up to 60%, improves scan quality, and assures secure user access with the use of encryption and authentication protocols.

This research contributes to cybersecurity by providing an interactive, scalable, and modular penetration testing framework that can be used by both beginners and experienced professionals. Future enhancements may include AI-driven vulnerability detection, IoT security testing, and integration with third-party security tools for extended functionality.

**Keywords**: Penetration Testing, Cybersecurity, Web Application Security, OWASP, Ethical Hacking, Python, Automation, Security Framework.

**Word Count:** 340

TABLE OF CONTENTS

[**CERTIFICATION** i](#_Toc193298054)

[**DEDICATION** ii](#_Toc193298055)

[**ACKNOWLEDGMENT** iii](#_Toc193298056)

[**ABSTRACT** iv](#_Toc193298057)

[**TABLE OF FIGURES** vii](#_Toc193298058)

[**ABBREVIATIONS** viii](#_Toc193298059)

[**CHAPTER ONE** 1](#_Toc193298060)

[**INTRODUCTION** 1](#_Toc193298061)

[**1.1** **Background to the Study** 1](#_Toc193298062)

[**1.2** **Statement of the Problem** 2](#_Toc193298063)

[**1.3** **Aim and Objectives of the Study** 3](#_Toc193298064)

[**1.4** **Methodology** 5](#_Toc193298065)

[**1.4.1 Develop an Automated Reconnaissance and Vulnerability Scanning Module** 5](#_Toc193298066)

[**1.4.2 Design a Secure and Scalable Database for Logging and Reporting** 6](#_Toc193298067)

[**1.4.3 Develop an Interactive Web-Based Penetration Testing Interface** 6](#_Toc193298068)

[**1.4.4 Implement and Test the Developed Web-Based Application** 7](#_Toc193298069)

[**1.5** **Scope of the Study** 8](#_Toc193298070)

[**1.6** **Significance of the Study** 9](#_Toc193298071)

[**1.7** **Justification for the Study** 9](#_Toc193298072)

[**1.8** **Motivation for the study** 11](#_Toc193298073)

[**1.9** **Organization of the Project** 12](#_Toc193298074)

[**1.10** **Operational Definition of Terms** 13](#_Toc193298075)

[**CHAPTER TWO** 16](#_Toc193298076)

[**REVIEW OF LITERATURE** 16](#_Toc193298077)

[**2.0** **Introduction** 16](#_Toc193298078)

[**2.1** **Overview of Penetration Testing** 16](#_Toc193298079)

[**2.1.1** **Definition and Importance of Penetration Testing** 16](#_Toc193298080)

[**2.1.2** **Phases of Penetration Testing** 17](#_Toc193298081)

[**2.1.3** **Types of Penetration Testing** 18](#_Toc193298082)

# TABLE OF FIGURES

[**Figure 2.1 PHASES OF PENETRATION TESTING** 18](#_Toc193298049)

# ABBREVIATIONS

**API** Application Programming Interface  
**ATT&CK**  Adversarial Tactics, Techniques, and Common Knowledge  
**AWS**  Amazon Web Services  
**CLI**  Command Line Interface  
**CTF** Capture The Flag (Cybersecurity competition)  
**D3.JS** Data-Driven Documents (JavaScript Library for Visualization)  
**DBMS** Database Management System  
**DRF** Django Rest Framework  
**FQDN** Fully Qualified Domain Name  
**GUI** Graphical User Interface  
**HTTP** Hypertext Transfer Protocol  
**HTTPS** Hypertext Transfer Protocol Secure  
**IoT** Internet of Things  
**IP** Internet Protocol  
**JSON** JavaScript Object Notation  
**JWT** JSON Web Token  
**MSF** Metasploit Framework  
**NIST** National Institute of Standards and Technology  
**NMAP** Network Mapper (Port Scanning Tool)  
**ORM** Object-Relational Mapping  
**OS** Operating System  
**OSINT** Open-Source Intelligence  
**OWASP** Open Web Application Security Project  
**PEN-T** Web-Based Penetration Testing Framework  
**RBAC** Role-Based Access Control  
**RPC**  Remote Procedure Call  
**SQL** Structured Query Language  
**SVM** Support Vector Machine (Machine Learning Model)  
**UAT** User Acceptance Testing  
**URL** Uniform Resource Locator  
**VM** Virtual Machine  
**VPN** Virtual Private Network  
**XSS** Cross-Site Scripting

# CHAPTER ONE

## INTRODUCTION

### Background to the Study

In the rapidly evolving digital age, cybersecurity has become an essential aspect of protecting information systems, networks, and applications from threats and vulnerabilities. As cyberattacks grow in sophistication and scale, organizations require robust tools and frameworks that support regular security assessments and proactive defense. One of the primary ways to achieve this is through penetration testing, a simulated cyberattack designed to evaluate the security posture of a system.

Traditional penetration testing methods often rely on multiple disconnected tools and manual procedures, making the process time-consuming, inconsistent, and sometimes inefficient for real-world use. Many available tools are either overly complex for beginners or lack flexibility for professionals to tailor assessments to specific organizational contexts (Kim & Solomon, 2023). In addition, penetration testing is often limited by a lack of automation, poor integration between phases (reconnaissance, scanning, exploitation, reporting), and weak documentation or visualization capabilities (Stuttard & Pinto, 2020).

To address these issues, there is a growing need for a unified penetration testing framework that is web-based, user-friendly, and integrates the core phases of testing into one platform. Such a solution can enable cybersecurity professionals to perform efficient, repeatable, and well-documented tests while following recognized standards like OWASP Testing Guide (OWASP, 2021), NIST SP 800-115 (NIST, 2021), and the MITRE ATT\&CK framework (MITRE, 2022).

This study aims to develop a Web-Based Penetration Testing Framework (PEN-T) that automates critical testing stages while implementing role-based access control (RBAC), secure reporting, and real-time scan visualization. By streamlining penetration testing operations, the framework supports both novice learners and experienced professionals in conducting assessments with greater confidence, security, and accuracy.

### Statement of the Problem

Despite the availability of various penetration testing tools, cybersecurity professionals often face challenges due to the fragmentation and complexity of these tools. Most existing solutions are either overly specialized, difficult to integrate, or lack user-friendly interfaces making them inaccessible to beginners and time-consuming for experts. Additionally, many tools do not provide seamless transitions between the core stages of testing; reconnaissance, scanning, exploitation, and reporting leading to inefficiencies and inconsistent results.

Furthermore, many penetration testing processes lack proper automation and structured reporting, which increases the likelihood of missed vulnerabilities and delays in remediation. The absence of centralized control, role-based access, and visual analytics further limits the efficiency and effectiveness of penetration assessments in both academic and organizational environments.

To address these limitations, there is a need for a comprehensive, automated, and web-based penetration testing framework that streamlines the testing lifecycle, ensures secure access based on user roles, and enhances decision-making through detailed reports and visualizations.

### Aim and Objectives of the Study

**The aim of this study is to design and implement a web-based penetration testing framework that automates and integrates key testing phases—reconnaissance, scanning, exploitation, and reporting within a secure, user-friendly, and role-based platform. The specific objectives of the study are to:**

1. **Develop an automated module for reconnaissance and vulnerability scanning using integrated open-source tools.**
2. **Design a secure and scalable relational database to store user data, scan logs, and test results with role-based access control (RBAC).**
3. **Implement a web-based interface using React.js and Django to support real-time test execution, result visualization, and report generation.**
4. **Conduct functional, performance, and user acceptance testing to evaluate the usability, security, and effectiveness of the developed framework.**

### Methodology

This section outlines the development techniques, processes, tools, and implementation strategies adopted in the creation of the Web-Based Penetration Testing Framework (PEN-T). The methodology combines both structured and iterative software engineering approaches to ensure a secure, efficient, and scalable system. It draws from globally recognized standards such as the OWASP Testing Guide, NIST SP 800-115, and MITRE ATT&CK Framework, integrating reconnaissance, scanning, exploitation, and reporting into a unified web-based interface.

#### ****1.4.1 Development Strategy****

The framework was developed using the Design Science Research (DSR) methodology. This approach emphasizes the construction and evaluation of artefacts designed to solve identified problems. DSR was chosen for its suitability in security tool development where iterative testing and improvement are required. Agile development principles were also applied to allow for modular construction and feedback-based refinement at each stage.

#### ****1.4.2**** Reconnaissance and Vulnerability Scanning Module

This module automates information gathering and vulnerability detection to save time and improve efficiency.

1. Reconnaissance: The framework performs passive reconnaissance using OSINT (Open-Source Intelligence) techniques. Tools such as Shodan, WHOIS Lookup, and custom Python-based subdomain enumeration scripts are integrated to collect publicly available data about target systems.
2. Port and Service Scanning: Active scanning is performed using Nmap, which handles TCP/UDP port detection, service fingerprinting, and OS detection. Automation is achieved by integrating python-nmap with the backend logic.
3. Web and Network Vulnerability Scanning: Tools such as OWASP ZAP, Nikto, and OpenVAS are incorporated to scan for common vulnerabilities like SQL injection, Cross-Site Scripting (XSS), misconfigured services, and outdated software.
4. Implementation Approach: Python was chosen for scripting and automation, allowing dynamic integration with various scanners and providing support for both manual and automated execution of reconnaissance tasks.

### ****1.4.3**** Database Design and Security

A structured and secure relational database underpins the data management layer of the PEN-T framework.

#### 1.4.3.1 Database Technology

MySQL is used for structured storage of user data, vulnerability scan logs, and exploitation records. SQLite may be used during local development for convenience.

#### Security Measures

1. Passwords are hashed using bcrypt to prevent unauthorized access.
2. Role-Based Access Control (RBAC) is implemented with four roles: Admin, Pentester, Viewer/Analyst, and Guest, restricting functionality based on privileges.
3. The database is protected with access controls and encrypted communication channels.

#### 1.4.3.3 Stored Data Includes

1. User credentials and activity logs
2. Test configurations and scan results
3. Exploit records and timestamped reports

### ****1.4.4**** Web-Based Interface Development

A responsive web application provides a user-friendly interface for conducting penetration testing activities.

1. Frontend Development: The frontend is built using React.js, ensuring a modern, interactive, and mobile-responsive design.
2. Backend Development: The backend is developed using the Django framework in Python. It handles authentication, session management, database interaction, and RESTful API logic.
3. Core Features Include:
   * + - 1. An interactive dashboard for executing modules (Reconnaissance, Scanning, Exploitation, Reporting)
         2. Real-time feedback using WebSockets for test progress and vulnerability detection updates
         3. Graphical reports generated using Matplotlib and D3.js, including attack vectors, severity levels, and success rates
         4. JWT (JSON Web Tokens) for secure authentication and session management

### Integration with Exploitation Tools

To support active exploitation of identified vulnerabilities:

1. The Metasploit RPC API is integrated to allow safe and controlled execution of exploits directly from the PEN-T interface.
2. This integration provides automated payload delivery, session management, and real-time result tracking.

### Testing and Evaluation

The system undergoes multiple layers of validation to ensure reliability, performance, and security.

1. Unit Testing: The Pytest framework is used to test individual components such as the scanning module, authentication system, and reporting module.
2. Load and Performance Testing: Locust is used to simulate multiple users interacting with the platform concurrently, testing its resilience under load.
3. User Acceptance Testing (UAT): The application is tested by cybersecurity students and professionals who provide feedback on usability and effectiveness, leading to further iterations.

### Scope of the Study

This study is focused on the design and implementation of a web-based penetration testing framework that automates core aspects of security testing specifically reconnaissance, scanning, exploitation, and reporting. The scope encompasses both the backend and frontend development of the system, as well as database design, user role management, integration with existing tools, and testing processes.

The framework is intended for use in educational institutions, cybersecurity training environments, and small to mid-sized organizations that require structured and repeatable security assessments. It will provide role-based access for different users such as Administrators, Pentesters, Analysts, and Guests, thereby ensuring flexibility and controlled access.

Functionally, the study covers:

1. Automated Reconnaissance: Integration of open-source intelligence (OSINT) tools for passive information gathering.
2. Port and Vulnerability Scanning: Leveraging tools such as Nmap, OWASP ZAP, and OpenVAS.
3. Exploit Execution: Safe integration with the Metasploit RPC API for controlled exploitation.
4. Reporting and Visualization: Graphical representation of scan results using tools like Matplotlib and D3.js.

The implementation will be done using React.js for the frontend and Django for the backend, with MySQL/SQLite as the database. The deployment environment includes local servers (Windows and Ubuntu), with optional cloud deployment considered for scalability.

While the system is designed for scalability, this version of the study does not include advanced AI-based vulnerability classification, mobile app support, or real-world live environment testing beyond internal lab simulations. Future improvements may include machine learning integration, real-time collaboration, and cloud-native orchestration.

### Significance of the Study

This study is significant as it addresses the growing need for structured, automated, and accessible penetration testing tools tailored for both professional use and cybersecurity education. As cyber threats continue to evolve in complexity and volume, traditional manual approaches to penetration testing are often time-consuming, error-prone, and limited in scalability.

The development of the PEN-T Framework offers a robust and user-friendly solution that integrates essential stages of penetration testing; reconnaissance, scanning, exploitation, and reporting into a single web-based application. This eliminates the need for switching between multiple tools and enables security professionals and learners to conduct comprehensive security assessments with greater speed and accuracy.

By incorporating role-based access control (RBAC), the system ensures secure usage for various user groups, including administrators, ethical hackers, analysts, and guests. Its structured reporting and visualization features also improve clarity and communication of results, which are critical for remediation planning and executive decision-making.

Furthermore, this study supports cybersecurity education by providing a learning platform that reflects real-world scenarios and testing workflows. Educational institutions, cybersecurity trainers, and learners can benefit from a hands-on tool that mirrors professional penetration testing standards, such as OWASP, NIST SP 800-115, and the MITRE ATT&CK framework.

Ultimately, the PEN-T Framework enhances security awareness, promotes best practices in ethical hacking, and contributes to the global effort in reducing system vulnerabilities through structured assessments and responsible disclosure.

### 1.7 Justification for the Study

The justification for this study stems from the increasing demand for efficient, centralized, and user-friendly penetration testing solutions in both academic and professional environments. Traditional penetration testing often requires extensive expertise and the use of disjointed tools, which can complicate the assessment process, reduce productivity, and limit learning opportunities for aspiring cybersecurity professionals.

Studies such as McKinnel et al. (2019) and Koroniotis et al. (2021) highlight the challenges in scaling manual security testing approaches, especially when multiple tools must be coordinated manually. Similarly, the NIST SP 800-115 framework underscores the importance of structured and repeatable security testing processes, a need that the PEN-T Framework seeks to fulfill.

By integrating essential testing phases such as reconnaissance, scanning, exploitation, and reporting into a cohesive web-based platform, this research not only simplifies the penetration testing lifecycle but also promotes consistency and repeatability. The application of Role-Based Access Control (RBAC) ensures that varying levels of users can interact with the platform securely, enhancing the framework’s flexibility and adoption across different use cases.

The project is also justified by its relevance to cybersecurity education. With a growing demand for hands-on cybersecurity training in academic institutions, this study provides a real-world simulation environment aligned with recognized standards like OWASP Top 10, MITRE ATT\&CK, and SANS security guidelines. This ensures that both learners and professionals can improve their practical skills while aligning with industry expectations.

In addition, the modular nature of the framework encourages extensibility and community-driven contributions, allowing for future improvements and collaboration, particularly important in open-source software initiatives.

### 1.8 Motivation for the study

The motivation for this study arises from the increasing cybersecurity threats targeting web applications and networked systems, coupled with the complexity and inefficiency of existing penetration testing tools. As cyberattacks grow in sophistication, organizations and cybersecurity professionals require streamlined, automated tools that not only enhance detection capabilities but also reduce human error and improve response time.

A 2023 report by the SANS Institute emphasizes the evolving nature of attack vectors and highlights the need for more structured, automated security testing to meet modern threat challenges. Manual testing methods often result in inconsistencies, missed vulnerabilities, and steep learning curves for beginners. This project aims to address those gaps by developing a unified, web-based penetration testing framework PEN-T that automates the core phases of testing while remaining intuitive and extensible.

Furthermore, as a final year cybersecurity student with practical exposure in security operations and ethical hacking tools (such as Metasploit, Nmap, and Burp Suite), the desire to build a real-world solution that aligns with industry methodologies like OWASP Testing Guide, NIST SP 800-115, and MITRE ATT&CK became a natural next step. The framework is envisioned to be beneficial not just to professionals, but also to students, educators, and enthusiasts seeking a structured platform for learning and simulating real-world attack scenarios.

Ultimately, the motivation is driven by a desire to contribute to the field of cybersecurity by building a secure, accessible, and customizable platform that bridges the gap between fragmented penetration testing tools and integrated, role-based, open-source solutions.

### 1.9 Organization of the Project

This project report is structured into five well-organized chapters to present a comprehensive and logical flow of the research and development process of the Web-Based Penetration Testing Framework (PEN-T). Each chapter contributes to building an understanding of the study and its implementation.

Chapter One lays the foundation of the study. It provides the background to the study, clearly defines the statement of the problem, the aim and four specific objectives of the research, outlines the methodology employed, and specifies the scope, significance, justification, and motivation for the research. This chapter serves to orient the reader on the purpose and importance of the study.

Chapter Two is devoted to a comprehensive literature review. It examines existing theories, methodologies, and tools related to penetration testing, web application security, and vulnerability assessment. This chapter highlights recent advancements, summarizes related works, and identifies existing gaps in prior studies which this research seeks to address.

Chapter Three focuses on the materials and methods used in building the PEN-T Framework. It discusses the design approach, tools and technologies (such as Django, React.js, Nmap, OWASP ZAP, etc.), and the step-by-step implementation strategy adopted in developing the framework. It also includes system architecture, workflow diagrams, and security design principles.

Chapter Four presents the implementation, testing, and evaluation of the developed web-based application. It shows how each module functions, how test cases were designed, the performance of the system under various scenarios, and the outcomes of unit, functional, penetration, and load testing.

Chapter Five concludes the study. It summarizes the findings of the research, reflects on the objectives achieved, and provides conclusions, recommendations, and suggestions for future improvements or research directions.

### 1.10 Operational Definition of Terms

To ensure clarity and avoid ambiguity, the following key terms used in this research are defined as they apply to the study:

1. Penetration Testing (Pentesting): A simulated cyberattack performed to evaluate the security of an information system by exploiting vulnerabilities in applications, networks, or infrastructure under controlled conditions.
2. Vulnerability Assessment: The process of identifying, classifying, and prioritizing security weaknesses or flaws in a system before they are exploited by malicious actors.
3. Reconnaissance: The initial phase in penetration testing where information is passively or actively gathered about a target system or organization to identify potential entry points.
4. Exploit: A piece of code, technique, or software that takes advantage of a vulnerability to compromise the security of a system or application.
5. OWASP: The Open Worldwide Application Security Project, a nonprofit foundation that provides widely accepted best practices and tools for improving the security of web applications.
6. NIST SP 800-115: A technical guide developed by the \*\*National Institute of Standards and Technology\*\* that provides a structured framework for planning and conducting information security testing and assessments.
7. MITRE ATT&CK Framework: A globally accessible knowledge base of adversary tactics and techniques based on real-world observations used to assess security posture and guide penetration testing simulations.
8. Role-Based Access Control (RBAC): A method of restricting system access based on predefined roles. In this study, RBAC ensures that users (Admin, Pentester, Viewer/Analyst, and Guest) only have access to functions necessary for their role.
9. Web Application: A software application that is accessed through a web browser and interacts with a server over the internet using web technologies such as HTML, CSS, JavaScript, and backend frameworks like Django.
10. Open Source Intelligence (OSINT): The process of collecting publicly available data from online sources (such as WHOIS, DNS records, and websites) to aid in reconnaissance and information gathering during penetration testing.
11. Exploit Development: The practice of creating custom code or scripts to take advantage of known or unknown software vulnerabilities in a controlled testing environment.
12. JSON Web Token (JWT): A compact, URL-safe token used for securely transmitting information between parties, often used for authentication in modern web applications.
13. React.js: A JavaScript library for building interactive user interfaces, particularly single-page web applications with dynamic data rendering.
14. Django: A high-level Python web framework that promotes rapid development and clean, pragmatic design, used for building the backend of the PEN-T framework.

# CHAPTER TWO

## REVIEW OF LITERATURE

### 2.0 Introduction

This chapter presents a comprehensive review of existing literature related to penetration testing, cybersecurity frameworks, methodologies, and tools. The review aims to provide foundational knowledge and highlight current trends, standards, and practices that influence the development of web-based penetration testing frameworks.

Key industry-recognized methodologies such as the OWASP Testing Guide, NIST SP 800-115, and the MITRE ATT\&CK framework are examined for their structured approaches to vulnerability assessment and threat modeling. Additionally, widely adopted penetration testing tools and technologies are reviewed to identify strengths, limitations, and areas for integration into the proposed system.

The chapter also explores recent advancements in the field, such as the integration of automation and artificial intelligence in penetration testing. Prior academic and industrial works are critically analyzed, and the existing research gaps are identified. This forms the basis for justifying the development of a new, modular, and automated penetration testing framework tailored to modern security challenges.

### 2.1 Overview of Penetration Testing

Penetration testing, also known as ethical hacking, is a proactive approach to identifying and addressing security vulnerabilities in computer systems, networks, or web applications by simulating real-world cyberattacks (Kim & Solomon, 2023). The primary goal is to uncover exploitable weaknesses before malicious actors do, thereby strengthening the overall security posture of an organization.

A typical penetration test involves a series of phases such as reconnaissance, scanning, exploitation, and reporting. Each phase targets different aspects of the system, aiming to uncover misconfigurations, insecure services, outdated software, and logic flaws. According to the National Institute of Standards and Technology (NIST), penetration testing is a critical component of a robust information security program, serving both technical and compliance purposes (NIST, 2021).

Penetration testing can be classified into various types based on the scope and knowledge available to the tester:

1. Black-box testing: Testers have no prior knowledge of the system, simulating an external attacker.
2. White-box testing: Testers have full knowledge of the internal architecture and source code.
3. Gray-box testing: Testers have partial knowledge, combining aspects of both black-box and white-box testing.

Organizations frequently conduct penetration tests to assess the effectiveness of their security controls, ensure compliance with industry standards like ISO 27001, PCI DSS, and HIPAA, and to identify unknown vulnerabilities that automated scanners may miss (Stuttard & Pinto, 2020).

Over time, penetration testing has evolved from manual assessments to automated and framework-driven approaches, using tools like Metasploit, Nmap, Burp Suite, Nikto, and OWASP ZAP. Despite these advancements, many tests remain fragmented, requiring coordination across multiple platforms. This fragmentation creates challenges in usability, report generation, and result aggregation necessitating unified frameworks that streamline the testing process (Koroniotis et al., 2021).

As cybersecurity threats continue to grow in complexity, the importance of systematic and repeatable penetration testing methodologies becomes increasingly vital.

### **2.1.1 Definition and Importance of Penetration Testing**

Penetration testing is a structured process that involves **identifying, exploiting, and reporting security vulnerabilities** in IT infrastructures. It is an essential component of cybersecurity because:

1. It helps organizations **identify and remediate security weaknesses** before attackers exploit them.
2. It provides **real-world insights** into an organization’s security posture.
3. It ensures compliance with security **frameworks and regulations** such as **OWASP, NIST, and ISO 27001**.
4. It helps assess the effectiveness of **firewalls, intrusion detection systems, and security controls** (Vacca, 2021).

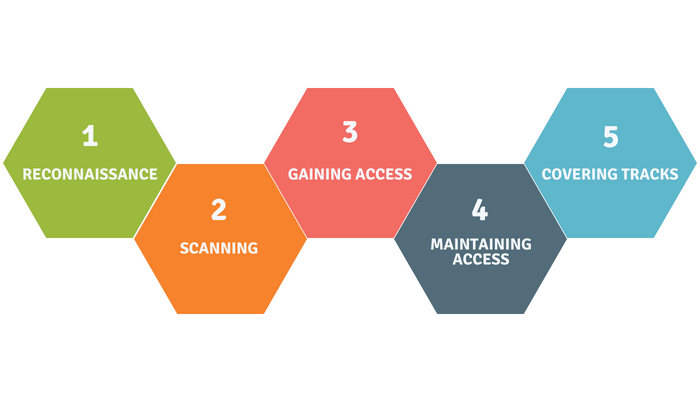
According to a **2023 IBM Security Report**, organizations that conducted regular penetration testing reduced their **data breach detection time by 37%** compared to those that relied solely on traditional security monitoring methods. This emphasizes the **importance of penetration testing in proactive security management.**

### **2.1.2 Phases of Penetration Testing**

Penetration testing follows a structured methodology, commonly divided into five key phases (Stuttard & Pinto, 2020):

1. **Reconnaissance (Information Gathering)** – The attacker gathers intelligence about the target system using **OSINT (Open-Source Intelligence) techniques**, DNS lookups, and network scanning.
2. **Scanning & Enumeration** – Tools such as **Nmap and Nessus** are used to detect open ports, services, and vulnerabilities.
3. **Exploitation** – Security weaknesses are exploited using frameworks like **Metasploit** to gain unauthorized access.
4. **Post-Exploitation & Privilege Escalation** – Once access is gained, further attacks such as **data exfiltration, lateral movement, and persistence mechanisms** are tested.
5. **Reporting & Remediation** – The penetration tester documents **findings, risk levels, and recommendations**, helping organizations patch vulnerabilities effectively.

These phases ensure that penetration testing is conducted in a **systematic and ethical manner**, reducing risks associated with security testing.



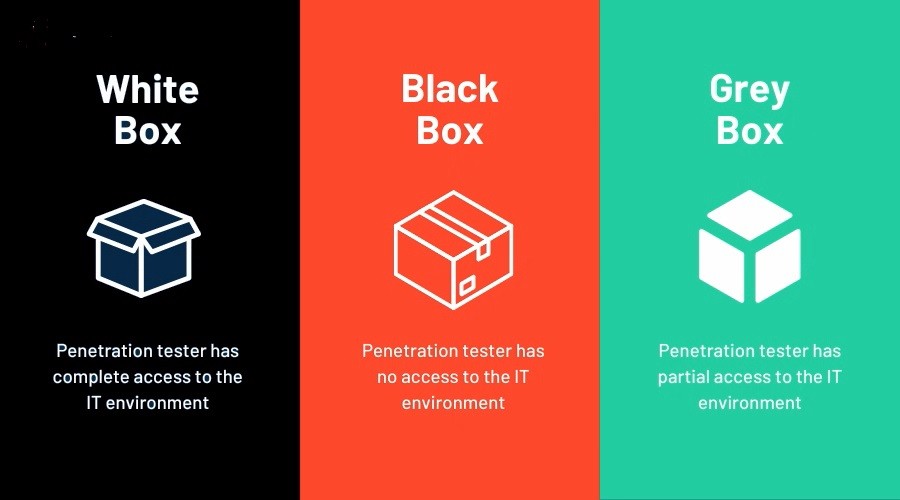
**Figure 2.1 PHASES OF PENETRATION TESTING**

**(**[*https://i0.wp.com/www.qatouch.com/wp-content/uploads/2018/08/phases-of-penetration-testing.png?w=700&ssl=1*](https://i0.wp.com/www.qatouch.com/wp-content/uploads/2018/08/phases-of-penetration-testing.png?w=700&ssl=1)**)**

### **2.1.3 Types of Penetration Testing**

Penetration testing can be categorized into different types based on the testing approach and target environment (Kim & Solomon, 2022):

1. **Black Box Testing** – The tester has **no prior knowledge** of the target system, simulating a **real-world cyberattack**.
2. **White Box Testing** – The tester has **full access** to the system’s internal architecture, allowing for a **detailed security assessment**.
3. **Gray Box Testing** – A hybrid approach where the tester has **limited knowledge** of the system, balancing **realistic attack scenarios and efficiency**.
4. **Web Application Penetration Testing** – Focuses on identifying **vulnerabilities in web applications**, such as **SQL Injection, XSS, and authentication flaws**.
5. **Network Penetration Testing** – Evaluates **network infrastructure security**, testing **firewalls, routers, and network services**.
6. **Social Engineering Testing** – Assesses human vulnerabilities by simulating **phishing attacks, impersonation, and psychological manipulation**.



**Figure 2.2 THE THREE MAIN TYPES OF PENETRATION TESTING**

#### **2.1.4 Automated vs. Manual Penetration Testing**

Penetration testing can be performed using **manual techniques or automated tools**:

* **Manual Testing** – Requires human expertise to manually identify security flaws, making it more thorough but **time-consuming and resource-intensive**.
* **Automated Testing** – Uses tools such as **Metasploit, Nmap, Burp Suite, and OpenVAS** to **speed up vulnerability detection and exploit testing**.

The **PEN-T Framework** aims to bridge the gap between **manual and automated penetration testing** by integrating multiple pentesting tools into a **single web-based solution** for efficiency and accessibility.