Team Autopenters

AUTOPENT

Automated Penetration Testing Framework

TEAM INTRODUCTION

01

Name:
Himanshu Gaur
Reg. No.:
23BCY10127
Branch:
Cybersecurity (BCY)

02

Name:
Tanya Bharti
Reg. No.:
23BCE11555
Branch:
Core (BCE)

03

Name:
Abhinav Mehra
Reg. No.:
23BCY10015
Branch:
Cybersecurity (BCY)

04

Name:
Rananjay Singh Chauhan
Reg. No.:
23BAI10080
Branch:
AI/ML (BAI)

INTRODUCTION

- This AI-powered, fully automated penetration testing framework enhances the **speed, accuracy, and scalability** of security assessments. It modernizes the ethical hacking workflow by overcoming the limitations of slow, manual, and disconnected traditional methods, creating a single, intelligent system for comprehensive security analysis.

- It seamlessly combines industry-standard tools like **Nmap, Nikto, and SQLMap** into one cohesive platform. This is managed through an intuitive Graphical User Interface (GUI) built with **Pyqt5 & Flask,** providing a simple dashboard for launching tests and reviewing results with ease.

- Leveraging AI, the system generates detailed security reports in **PDF** and text formats. These reports detail identified vulnerabilities, explain their threats, provide base-level remedies, and intelligently prioritize all findings by severity, offering clear, actionable insights for security teams.

PROJECT OBJECTIVES

01

It seamlessly combines tools like **Nmap and Metasploit** into a single framework for **efficient, structured, and multi-layered** security assessments.

02

Local AI generates **structured reports** detailing vulnerabilities, exploitation steps, and remediation recommendations for security teams to take action.

03

It automates the entire pentesting workflow, leveraging AI and scripts for rapid **vulnerability identification,** making assessments faster and more scalable.

04

It automates **credential dumping, privilege escalation exploits, and persistence mechanisms** for a thorough security evaluation beyond just basic exploitation.

PROBLEM STATEMENT

LACK OF INTEGRATION

No centralized framework exists that seamlessly integrates all these tools into an automated workflow, reducing testing efficiency, increasing the risk of human error and oversight.

MANUAL EFFORT AND TIME CONSTRAINTS

Penetration testing typically requires highly skilled cybersecurity professionals to manually analyze network security, perform vulnerability assessments, and attempt exploitations.

POORLY STRUCTURED REPORTING MECHANISMS

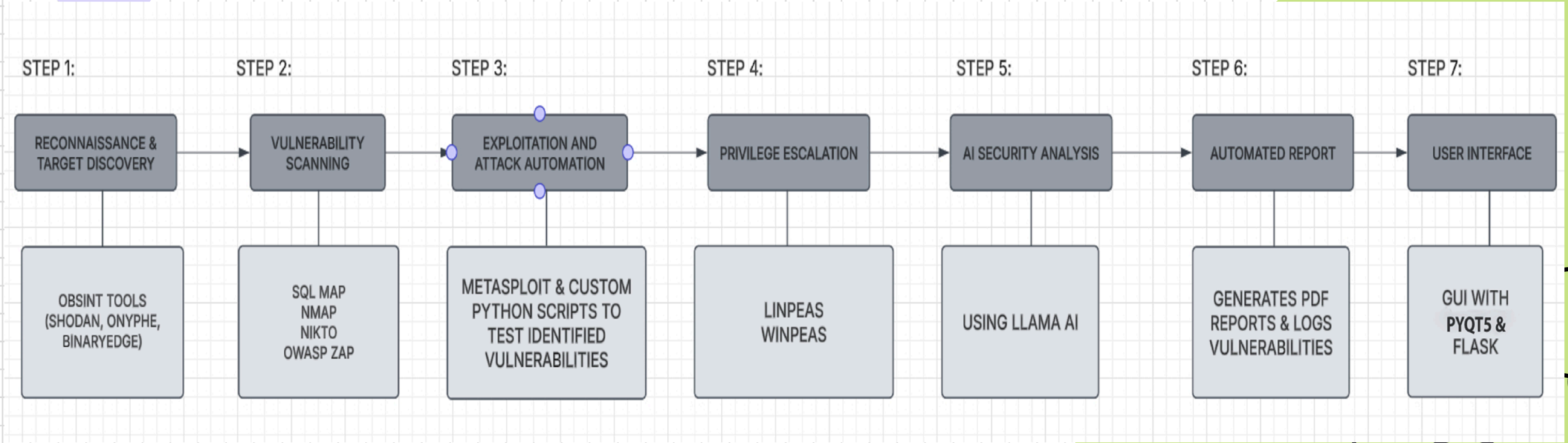
Reports often lack a uniform structure. Some pentesting tools report vulnerabilities but fail to provide detailed mitigation steps, leaving organizations struggling to fix issues.

PROPOSED METHODOLOGY

The explanation how the model will go through and what steps will be running in the background.

- Step 1: Reconnaissance
- Step 2: Vulnerability Scanning
- Step 3: Exploitation
- Step 4:Privilege escalation
- Step 5: Source Code Analysis
- Step 6: AI-Powered Security Analysis
- Step 7: User Interface and User experience

WORKFLOW OF THE PROJECT


STEP 1:
RECONNAISSANCE & TARGET DISCOVERY
OBSINT TOOLS (SHODAN, ONYPHE, BINARYEDGE)
STEP 2:
VULNERABILITY SCANNING
SQL MAP
NMAP
NIKTO
OWASP ZAP
STEP 3:
EXPLOITATION AND ATTACK AUTOMATION
METASPLOIT & CUSTOM PYTHON SCRIPTS TO TEST IDENTIFIED VULNERABILITIES
STEP 4:
PRIVILEGE ESCALATION
LINPEAS
WINPEAS
STEP 5:
AI SECURITY ANALYSIS
USING LLAMA AI
STEP 6:
AUTOMATED REPORT
GENERATES PDF REPORTS & LOGS VULNERABILITIES
STEP 7:
USER INTERFACE
GUI WITH PYQT5 & FLASK

RECONNAISSANCE

It involves collecting as much information as possible about the target system to **identify potential attack vectors** before launching any exploits. It extracts **DNS records, IP addresses,** and exposed services for attack surface mapping.

SHODAN

Vulnerability assessments, attack surface mapping, and OSINT (Open Source Intelligence) by finding misconfigured and vulnerable devices on the internet

BINARYEDGE

BinaryEdge provides IP scanning data and device fingerprinting. It is best for mass scanning which helps in detecting vulnerability in data and also helpful for botnet and malware hunting.

ONYPHE

Automating threat intelligence gathering, streamlining incident response, detecting network vulnerabilities, analyzing domain and IP reputation, and discovering an organization's hidden infrastructure

VULNERABILITY SCANNING

Automates detecting vulnerabilities in web applications, network, and databases.

CORE SCANNING COMPONENTS: NIKTO & SQLMAP

- **NIKTO:** Scans for outdated software, security misconfigurations, and exposed sensitive files. It's fast with minimal false positives, enabling early detection of security loopholes before manual testing.
- **SQLMap:** Automates SQL injection testing and database enumeration. It supports multiple databases and bypasses security to uncover data leaks, unauthorized access, and misconfigurations.

COMPREHENSIVE VULNERABILITY SCANNING

- **Nmap:** Features port scanning, service fingerprinting, and vulnerability detection to identify weak services, misconfigurations, and open ports. It can also detect the presence of firewalls and intrusion detection systems (IDS).
- **OWASP Top 10:** A custom Python script scans for the ten most critical web application security risks, providing robust coverage against common, high-impact vulnerabilities.

EXPLOITATION

This helps to determine the actual risk of vulnerabilities by simulating real-world cyberattacks.

- For powerful and efficient **automated exploitation**, our framework uniquely integrates the flexibility of **Python sockets** with the robust capabilities of the renowned **Metasploit framework** . This synergy solves a common challenge in pentesting automation: bridging user-friendly inputs with complex backend tools. It all begins with a custom Python script that simplifies the process for the user. This script leverages sockets to transparently convert a target **URL into an IP address**, a critical step that prepares the target for the exploitation engine.
- Once the IP is obtained, it is passed directly to Metasploit, which acts as the heavy lifter, automating the intricate phases of **vulnerability exploitation and payload delivery.** The crucial link enabling this is an **RPC server**, which allows our Python script to remotely command Metasploit's actions. This turns our script into an intelligent controller for Metasploit, enabling it to launch exploits, deliver payloads to gain control, and set the stage for post-exploitation tasks. The result is a highly efficient and **seamless workflow** that combines custom logic with a world-class exploit library, achieving comprehensive penetration testing with minimal manual intervention.

PRIVILEGE ESCALATION

Privilege escalation is the process of gaining higher-level access to a system or network, usually moving from a low-privileged user to an administrator or root user. Attackers exploit security misconfigurations, vulnerabilities, or weak permissions to achieve this.

WINPEAS (WINDOWS PRIVILEGE ESCALATION AUTOMATION SCRIPT)

- It detects weak file permissions, unquoted service paths, and passwords stored in plaintext.
- It identifies Windows vulnerabilities that allow privilege escalation and finds misconfigured services, registry settings, and scheduled tasks.

LINPEAS (LINUX PRIVILEGE ESCALATION AUTOMATION SCRIPT)

- It detects sudo misconfigurations, SUID binaries, and writable files and finds kernel exploits that allow privilege escalation.
- It identifies hardcoded credentials and environment variables with sensitive data.

SOURCE CODE ANALYSIS

Source code analysis is the automated process of examining an application's source code to identify and remediate **security vulnerabilities, logic flaws, and coding errors** before they can be exploited by attackers.

Our framework utilizes **Haiku, an AI-powered static code analysis tool** designed to help developers find security flaws and coding mistakes across multiple programming languages, including **Java, Python, and JavaScript.**

We leverage the next-generation **Claude 3.5 Haiku model,** the fastest and most advanced version available. It offers significant improvements across all skill sets, providing more **efficient analysis and greater detail on identified vulnerabilities** compared to its predecessors.

A critical feature is its ability to provide actionable **fix recommendations** for discovered vulnerabilities. This capability is crucial for automating secure code reviews and integrating security seamlessly into **DevSecOps pipelines**, helping developers build more secure applications from the start.

AI-POWERED SECURITY ANALYSIS

The report generation process is handled by the generate_security_report() function in our reporting.py file. The step by-step working of this function:

1. First, in **Data Aggregation**, outputs from reconnaissance, scanning, and exploitation are consolidated into a structured format.
2. Next, during **AI-Powered Analysis**, this data is fed to the **LLaMA model.** The AI analyzes findings, correlates vulnerabilities, and generates a summary with prioritized **remediation recommendations.**
3. Then, a structured **HTML Report** is generated, embedding both the raw technical data and the AI's narrative into a clean, human-readable format.
4. Finally, for professional presentation and portability, the **pdfkit** library is used to convert the HTML file into the final **security_report.pdf**, ready for sharing and review.

USER INTERFACE AND USER EXPERIENCE

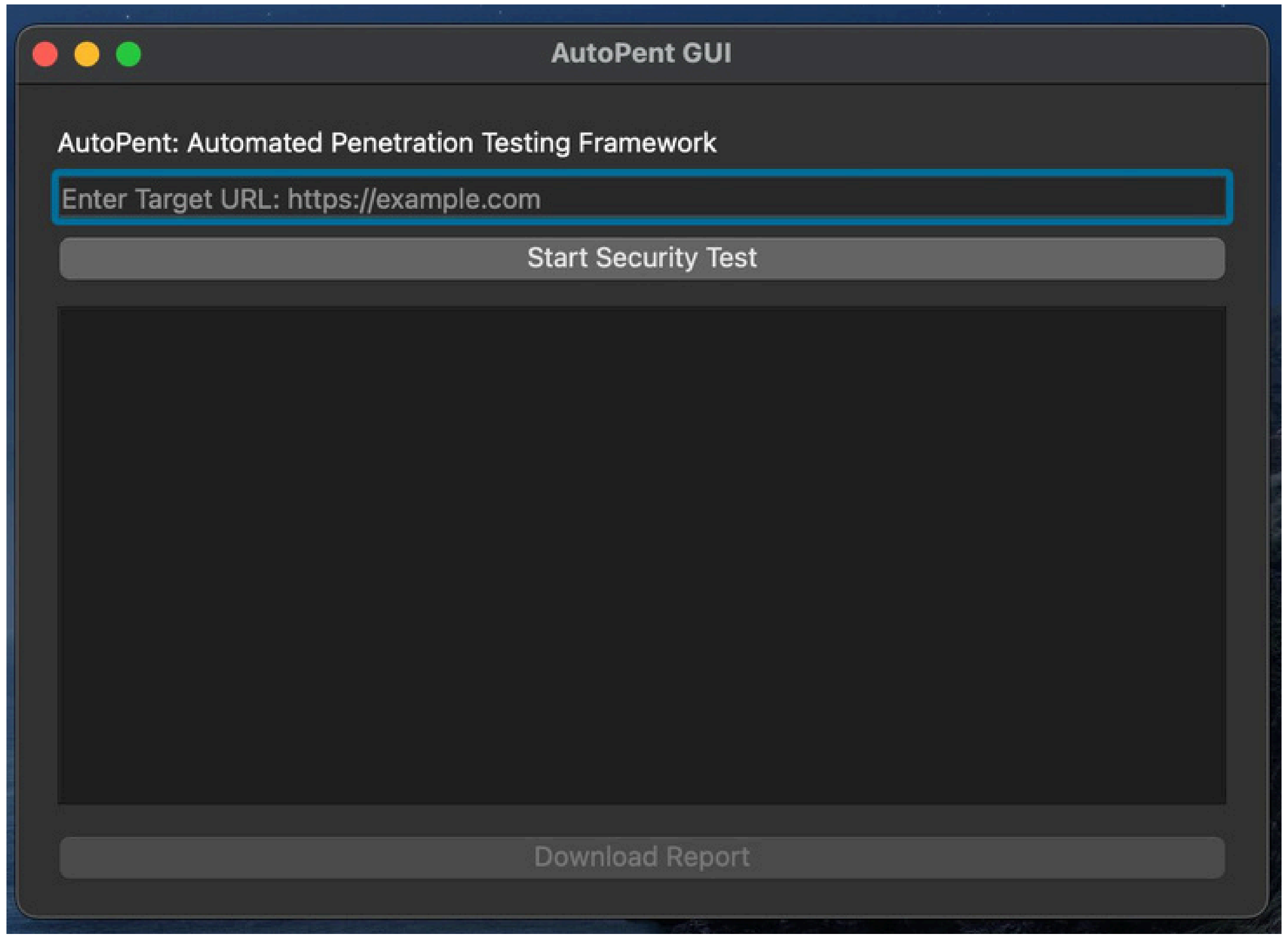
PYQT5

- Cross-Platform & Feature-Rich, as it can Build GUI applications for Windows, macOS, and Linux with 600+ built-in widgets.
- Event-Driven Programming it Uses Signals & Slots for seamless event handling.
- Drag & Drop UI with Qt Designer It can also Create UIs visually and convert .ui files into Python code.

FLASK

- It handles HTTP Requests as it connects the front-end UI to backend modules.
- It supports API Calls which enables us remotely execute pentesting tasks rather than manually performing them.
- It integrates with Databases & Logging Systems and stores pentesting reports which provides all time access to the report as it is stored in the local memory.

GUI DESIGN AND UI OVERVIEW


AutoPent GUI
AutoPent: Automated Penetration Testing Framework
Enter Target URL: https://example.com
Start Security Test
Download Report

THE GUI OF THE PROJECT APPLICATION

1. Built using PyQT5 and Flask.
2. Seamlessly integrates all of our features and allows for an intuitive user interaction
3. Real-time Summary report generated aftercompletion of testing

THANK YOU

Team Autopenters