SecuBot: AI-Powered Cybersecurity Automation

Synopsis submitted to

Shri Ramdeobaba College Of Engineering & Management , Nagpur

In partial fulfillment of requirement for the award of the degree of

Bachelor of Technology (B.Tech)

In

COMPUTER SCIENECE AND ENGINEERING

(Cyber Security)

By

Aditya Rameshwar Bahe	(28)
Bhushan Govinda Madankar	(39)
Shreyash kumar virendra kumar	(60)
Sayed Jafar Hussain Naqvi	(62)
Sarvesh Rajendra Mishra	(63)

Guide

Prof. Harshala Shingne

RCOEM

Shri Ramdeobaba College of Engineering and Management, Nagpur

Department of Computer Science And Engineering – Cyber Security

Shri Ramdeobaba College of Engineering & Management , Nagpur 440 013

(An Autonomous Institute to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur)

December 2024

I N D E X

PROBLEM STATEMENT

Modern cybersecurity tools have complex commands that users must memorize and execute manually, leading to inefficiencies and errors. AI solutions provide guidance but lack execution and integration. An AI-driven agent is needed to suggest, execute, and automate commands across multiple tools.

PROBLEM DESCRIPTION :-

Managing modern cybersecurity tools can feel like navigating a maze of complex commands. Security professionals must memorize intricate syntax, manually execute commands, and interpret raw outputs—all while ensuring systems remain secure. This process is not only time-consuming but also prone to errors, making it especially difficult for non-experts.

Imagine a security analyst who needs to scan a network for vulnerabilities. They must remember the exact syntax for tools like Nmap, carefully execute the command, and then sift through large amounts of output data to identify potential threats. A small syntax mistake or misinterpretation of results could lead to missed vulnerabilities or misconfigurations, putting the system at risk.

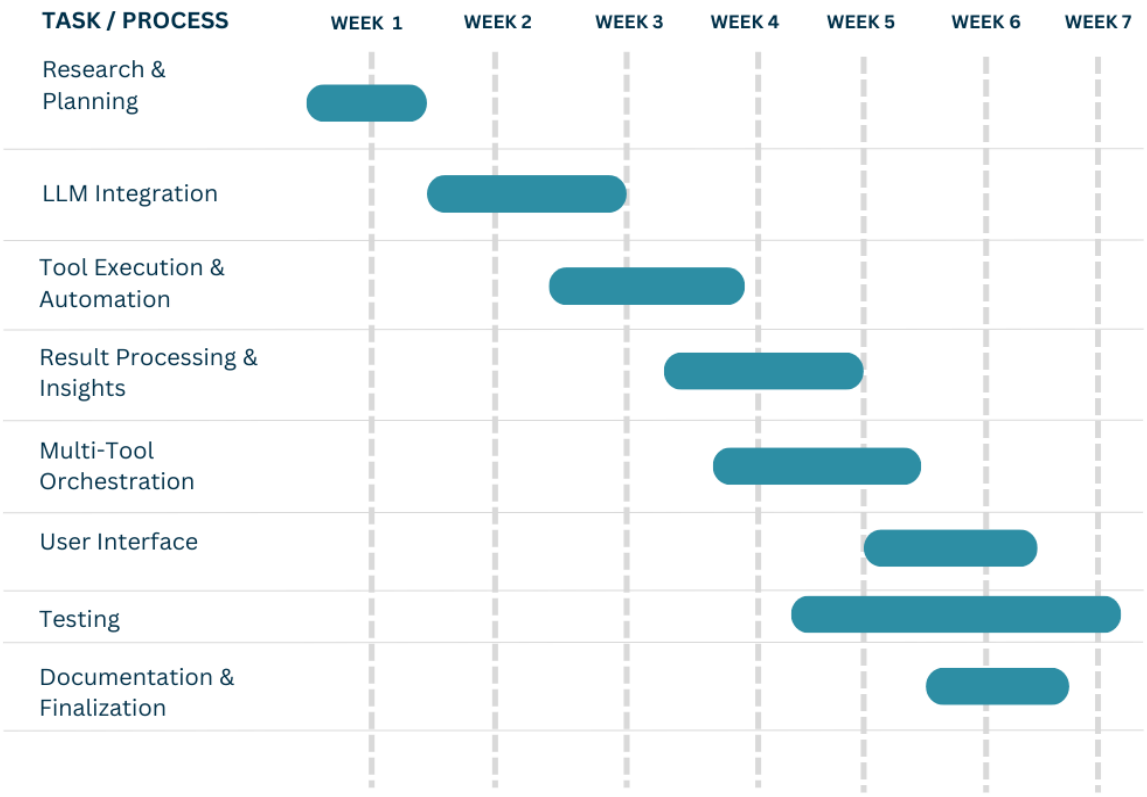
While AI-powered tools offer textual guidance, they lack the ability to directly execute commands, integrate multiple tools, or provide intelligent insights. As a result, security teams face fragmented workflows, inconsistent automation, and an overwhelming cognitive load.

There is a pressing need for an AI-driven agent that not only suggests the right commands but also executes them, interprets the results, and automates workflows. This would streamline cybersecurity operations, reduce errors, and make advanced security tools more accessible to all users.

PROJECT OBJECTIVES :-

- Develop an AI-driven system that understands user queries and converts them into precise tool-specific commands across cybersecurity.
- No more manual execution, the AI will automatically run commands, integrating with tools like Nmap, OpenVAS, Wireshark and git.
- Automate complex workflows—scan a network with Nmap, analyze traffic using Wireshark, and assess vulnerabilities with OpenVAS.
- Forget command-line complexity just describe your task in plain English, and the AI handles the rest, making security and system management accessible to all.
- The AI will not only execute commands but also interpret outputs, detect anomalies, suggest fixes, and prevent misconfigurations.
- This AI-powered agent will transform cybersecurity and IT operations, reducing errors, saving time, and making advanced tools easier to use.

PROPOSED PLAN OF WORK :-



METHODOLOGY :-

1. Understanding the Problem

Before building the AI-powered automation agent, we analyze existing cybersecurity tools. What makes them complex? Why do users struggle? This phase helps identify the key challenges and define the best approach to solving them.

2. Choosing the Right Tools & AI Model

- We select a fine-tuned Large Language Model (LLM) that understands security queries.

- Identify integration tools like Nmap, OpenVAS, Metasploit.
- Define the system architecture—ensuring scalability, security, and efficiency.

3. Automating Tool Execution & Response Processing

- The AI executes security commands dynamically.
- Results are retrieved, parsed, and formatted into actionable insights instead of raw outputs.
- Example: A user asks, *"Scan the network for vulnerabilities"* → AI runs an Nmap scan, analyzes with OpenVAS, and reports findings.

4. Multi-Tool Orchestration & Workflow Automation

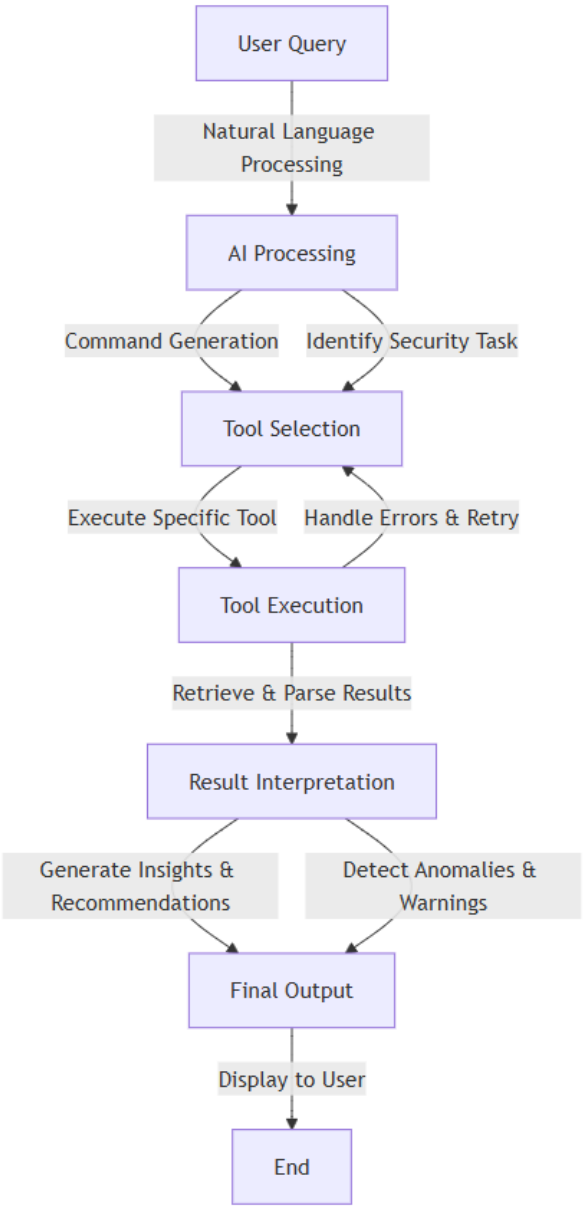
- The agent connects multiple tools.
- Example workflow:
    1. Scan network with Nmap
    2. Analyze traffic with Wireshark
    3. Identify vulnerabilities using OpenVAS

5. Building an Interactive UI

- A web-based chatbot interface makes interaction intuitive.
- Users type natural language queries instead of memorizing complex commands.
- Professionals and beginners alike can leverage security tools.

7. Testing & Optimization

- Unit tests ensure LLM-generated commands are correct.
- Security testing prevents unauthorized or risky command execution.
- Performance optimizations ensure real-time response without lag.



TECHNOLOGY :-

- *Python (Backend, AI) – Used for AI integration, automation, and tool execution.*
- *JavaScript (Frontend, React.js) – Enables a dynamic and interactive UI.*
- *LLMs (OpenAI GPT, Llama 3, Hugging Face) – Powers natural language understanding and command generation.*
- *Tool Integration (Nmap SDK, Metasploit SDK, etc.) – Automates cybersecurity and other tool executions.*
- *FastAPI / Flask (Backend) – Ensures fast, scalable API handling for AI interactions and tool execution.*

FUNTIONAL SPECIFICATION :-

- Accepts natural language queries via web UI.
- LLM Integration ,Generates and executes tool commands intelligently.
- Automates tools like Nmap, Metasploit, OpenVAS.
- Uses Web Sockets for live execution updates.
- Allows adding new tools via SDKs/APIs.

PROJECT SCOPE :-

- It will integrate with tools like Nmap, OpenVAS, Metasploit, Git, eliminating the need for manual execution.
- Users can simply describe their tasks in plain language, making advanced operations accessible to beginners and professionals alike .

Roll No.	Name Of Students	Name Of Guide
28	Aditya Rameshwar Bahe	*Prof. Harshala Shingne*
39	Bhushan Govinda Madankar	
60	Shreyash kumar virendra kumar	
62	Sayed Jafar Hussain Naqvi	
63	Sarvesh Rajendra Mishra	

Approved By :-