**Project Design Phase**

Exploring Cyber Security: Understanding Threats and Solutions in the Digital Age

Team Members:

1. Dhanraj Pawar
2. Kaushal Chougule
3. Athrav Pawar
4. Balaji Patil

Project Design Phase Summary

The Project Design Phase involves crafting a comprehensive framework for the project, covering system structure, workflows, and essential tools. This phase ensures that the project is systematically planned and prepared for implementation.

System Framework

The system framework defines the structural components of the proposed solution and consists of the following key elements:

1. User Interface (UI) – Frontend

Dashboard: A web-based platform that enables security analysts to review vulnerability assessment outcomes, track SIEM logs, and access incident reports.

Technologies: HTML, CSS, and JavaScript for interface development. A modern frontend library or framework, such as React or Angular, will be leveraged to enhance responsiveness and interactivity.

2. Server-Side Logic – Backend

Vulnerability Assessment: Seamless integration with Nessus for executing vulnerability evaluations on designated systems.

SIEM Connectivity: Incorporation of Splunk for log collection, monitoring, and detection of potential threats.

Technologies: Python for scripting, alongside a backend framework like Flask or Django for managing server-side operations.

3. Data Management – Database

Storage Solution: A MySQL database will store vulnerability assessment records, SIEM log data, and incident reports.

Tools: MySQL Workbench will be utilized for database administration and query enhancement.

4. System Integration

APIs: RESTful APIs will facilitate seamless communication between the frontend, backend, and database.

Data Format: JSON will be adopted for efficient data transmission between system components.

1. Database:
   1. Storage: A MySQL database will be used to store vulnerability scan results, SIEM logs, and incident reports.

○ Tools: MySQL Workbench will be used for database management and query optimization.

1. Integration:
   1. APIs: RESTful APIs will be used to connect the frontend, backend, and database.

○ Data Exchange: JSON will be used for data exchange between components.

# Workflows

The workflows define how the system will operate. Each workflow is designed to address a specific aspect of the project:

1. Vulnerability Assessment Workflow:
   1. Step 1: Perform a vulnerability scan using Nessus.

○ Step 2: Analyze the scan results and prioritize vulnerabilities based on severity (e.g., critical, high, medium, low).

○ Step 3: Generate a detailed scan report and share it with stakeholders for further action.

1. Threat Hunting Workflow:
   1. Step 1: Monitor Splunk logs for suspicious activity (e.g., unusual login attempts, unauthorized access).

○ Step 2: Investigate potential threats by analyzing log data and correlating events.

○ Step 3: Escalate confirmed threats to the incident response team and document findings in an incident report.

1. Incident Response Workflow:
   1. Step 1: Analyze phishing emails for indicators of compromise (IOCs) using custom Python scripts.

○ Step 2: Create an incident report that includes details of the attack, IOCs, and suggested remediation steps.

○ Step 3: Share the report with the incident response team and stakeholders.

1. Security Awareness Training Workflow:

* 1. Step 1: Develop training materials (e.g., presentations, handouts, quizzes) focused on phishing awareness, password security, and social engineering.

○ Step 2: Deliver the training to employees through workshops or online sessions.

○ Step 3: Collect feedback from participants and use it to improve future training programs.

○

# Tools and Technologies

The following tools and technologies will be used to implement the project:

1. Nessus: For vulnerability scanning and risk assessment.
2. Splunk: For SIEM and log monitoring.
3. Metasploit: For penetration testing and vulnerability exploitation.
4. Wireshark: For network traffic analysis.
5. Python: For scripting and automation (e.g., phishing email analysis, incident report generation).
6. MySQL: For storing and managing data (e.g., scan results, logs, incident reports).
7. React/Angular: For building the frontend dashboard.
8. Flask/Django: For backend development and API integration.

# Deliverables

1. System Architecture Diagram:
   1. A visual representation of the system architecture, showing how the frontend, backend, and database components interact.
2. Workflow Diagrams:
   1. Diagrams for each workflow (e.g., vulnerability assessment, threat hunting, incident response) to illustrate the steps involved.
3. Tool Setup Guide:
   1. A detailed guide for setting up and configuring the tools (e.g., Nessus, Splunk, MySQL).
4. Database Schema:
   1. A detailed schema for the MySQL database, including tables for scan results, logs, and incident reports.

# Next Steps

1. Project Planning Phase:
   1. Develop a sprint plan with tasks, timelines, and assigned team members.

○ Estimate story points and prioritize tasks.

1. Project Execution:
   1. Begin implementing the system based on the design.