IBM Innovation Center for Education

PHEMESOFT Software That Matters

TRUSTED DELIVERY PARTNER for IBM ICE

THE YENEPOYA INSTITUTE OF ARTS SCIENCE
COMMERCE AND MANAGEMENT
(a constituent unit of Yenepoya Deemed to be University)

PHISHING EMAIL DETECTION SYSTEM (PEDS)

PROJECT SYNOPSIS

MASTER OF COMPUTER SCIENCE AND APPLICATIONS

SUBMITTED BY :

K Samruddhi Shenoy 24MCA211

Ramya Shetty 24MCA218

Gouthami

GUIDED BY:

MR. SHASHANK

IBM Innovation Center for Education

PHEMESOFT Software That Matters
TRUSTED DELIVERY PARTNER for IBM ICE

TITLE PAGE

1. Name of the student: K Samruddhi Shenoy
2. Class Roll No. 24MCA211
3. Campus ID: 37931
4. Present official Address: YIASCM Blamatta, Mangalore 575002
5. Email: 37931@yenepoya.edu.in
6. Phone No. +91 6363269950
7. Branch: Computer Science
8. Batch: 2024-2026
9. Proposed Topic: Intrusion Detection System (IDS)

TABLE OF CONTENTS

Cover page - 1

Title - 2

Content - 3

1.1 Introduction - 4

1.2 Key Features

1.3 Technology Stack

1.4 Specialized Field: Cybersecurity and Ethical Hacking

2.1 Methodology- 5

2.2 Requirement Analysis and Tool Selection

2.3 System Architecture and Design

2.4 Frontend Development (Tkinter)

2.5 Backend Integration

2.6 Final Testing and Documentation

3.1 Facilities required for proposed work - 6

3.2 Development Environment

3.3 Detection & Mitigation Tools

3.4 Testing and Deployment

3.5 Reporting Tools

1.1 Introduction

The Phishing Email Detection System (PEDS) is designed to identify and classify emails as phishing or legitimate, enhancing email security for users and organizations. By analyzing email content, headers, and attachments, the system detects potential phishing attempts in real-time. Utilizing advanced machine learning techniques, including natural language processing and classification algorithms, PEDS aims to mitigate the risks associated with phishing threats, thereby ensuring a safer communication environment.

1.2 Key Features

- Real-time Email Analysis – Continuously scans incoming emails for phishing indicators.
- Content Classification – Utilizes machine learning to classify emails as phishing or legitimate based on learned patterns.
- User Alerts – Notifies users of detected phishing attempts with clear and actionable reports.
- Custom Detection Rules – Implements specific rules for identifying common phishing tactics and patterns.
- Logging & Reporting – Stores phishing detection events and email logs for auditing and analysis.
- Machine Learning Integration – Improves detection accuracy by learning from new phishing techniques over time.
- User-Friendly Interface – Provides an intuitive GUI for managing phishing email detections and reviewing reports.

1.3 Technology Stack

- **Frontend:**

  Tkinter: Provides the graphical user interface allowing users to view detected phishing emails, logs, and configure system settings.

- **Backend:**

  Scikit-learn: Implements machine learning algorithms responsible for email classification.

  Pandas: Manages data loading and preprocessing for machine learning.

  Natural Language Toolkit (NLTK): Supports text processing and feature extraction from email content.

Flask: Serves as the web framework to enable backend API communication if required.

SQLite: Stores logs of detected phishing emails and user interactions in a lightweight database.

1.4 Specialized Field: Cybersecurity and Email Security

This project falls under cybersecurity with a focus on email security. The Phishing Email Detection System addresses the growing threat of phishing attacks by equipping organizations and users with automated tools to identify and mitigate deceptive emails before they cause harm. This helps protect sensitive data, prevent financial loss, and maintain trust in digital communications.

2.1 Methodology

The development of the Phishing Email Detection System follows a systematic approach to ensure accurate and timely phishing email detection.

2.2 Requirement Analysis & Tool Selection

- Identify core functionalities such as email content scanning, machine learning-based classification, logging, and alerting.
- Select and integrate tools including Scikit-learn, NLTK, and Flask for processing, classification, and deployment.
- Ensure platform compatibility for broad usability across operating environments.

2.3 System Architecture and Design

- Architect a modular system separating the frontend interface, backend processing, and database storage.
- Design interaction flows between the user interface and the backend classification module for real-time detection.

2.4 Frontend Development (Tkinter)

- Create an interactive dashboard that displays detected phishing emails, notifications, and system logs.
- Allow users to customize detection sensitivities and manage reports.

2.5 Backend Integration

- Implement email text preprocessing and feature extraction using NLTK and Pandas.

- Train and deploy machine learning classification models with Scikit-learn.
- Integrate SQLite database for logging detection incidents and user actions.
- Build reporting capabilities to summarize detection results and trends.

2.6 Final Testing and Documentation

- Conduct rigorous testing on comprehensive email datasets to validate detection accuracy and minimize false positives.
- Optimize performance to ensure efficient real-time operations.
- Document system functionality, usage instructions, and maintenance guidelines.

3.1 Facilities Required for Proposed Work

Development requires appropriate software tools and hardware environments for coding, testing, and deployment.

3.2 Development Environment

- Python 3.x: The primary programming language for the entire system.
- Tkinter: For creating the graphical user interface.
- IDE (VSCode, PyCharm, or similar): For efficient development and debugging.

3.3 Detection & Mitigation Tools

- Scikit-learn: Machine learning framework for email classification.
- NLTK: Natural language processing toolkit for text feature extraction.
- Flask: Web framework for API and system integration.
- SQLite: Lightweight database for storing phishing detection logs.

3.4 Testing and Deployment

- Virtual environments for controlled testing of dependencies.
- Cross-platform testing on operating systems including Windows and Linux.

3.5 Reporting Tools

Python libraries such as ReportLab and WeasyPrint for generating detailed PDF and HTML detection reports.

System Architecture Overview

The Phishing Email Detection System (PEDS) is designed with a modular architecture that separates the frontend user interface from the backend processing components. This architecture ensures scalability, maintainability, and efficient real-time detection of phishing emails.

1. Frontend Layer

- **User Interface (UI)**:
    - Developed using **Tkinter**, this layer allows users to interact with the system, view detected phishing emails, manage logs, and configure detection settings.
    - **User Notifications**: Alerts users about detected phishing attempts through pop-ups or dashboard notifications.
    -

2. Backend Layer

- **Email Processing Module**:
    - **Content Analysis**: Utilizes **Scikit-learn** and **NLTK** for analyzing email content, headers, and attachments to identify phishing indicators.
    - **Feature Extraction**: Extracts relevant features from emails (e.g., keywords, links, sender information) for classification.

- **Machine Learning Module**:
    - **Classification Algorithms**: Implements various machine learning models (e.g., Naive Bayes, Random Forest) to classify emails as phishing or legitimate based on learned patterns.
    - **Training and Updating**: Continuously learns from new data to improve detection accuracy, allowing the model to adapt to evolving phishing tactics.

- **Database Module**:
    - **SQLite**: Stores logs of detected phishing attempts, user interactions, and historical data for analysis and reporting.

3. Integration Layer

- **API Layer**:
    - **Flask**: Provides a web framework for communication between the frontend and backend, allowing for real-time data exchange and processing of incoming emails.

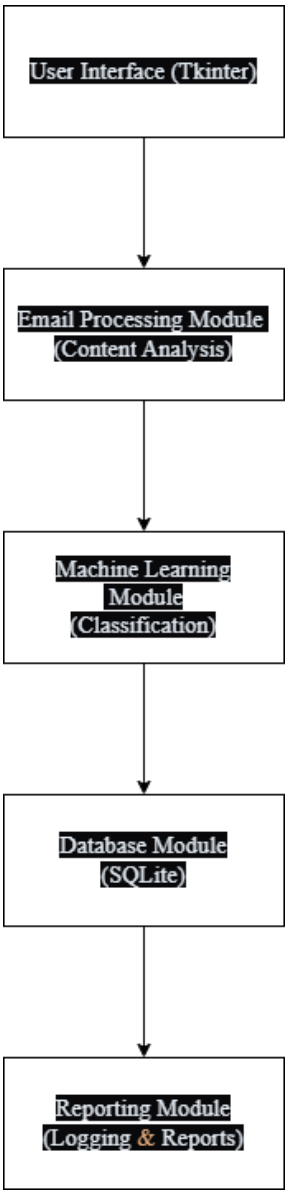
4. Reporting Module

- **Logging and Reporting**:
    - Generates detailed reports on detected phishing emails and user interactions, which can be exported in PDF or HTML formats for further analysis.

Data Flow

1. **Email Input**: Incoming emails are captured and sent to the Email Processing Module.
2. **Content Analysis**: The content of each email is analyzed for phishing indicators.
3. **Feature Extraction**: Relevant features are extracted and prepared for classification.
4. **Classification**: The Machine Learning Module classifies the email as phishing or legitimate.
5. **User Notification**: If phishing is detected, the user is notified through the UI.
6. **Logging**: Detected phishing attempts are logged in the database for future reference and reporting.

Diagram Representation



This architecture ensures that the Phishing Email Detection System operates efficiently, providing real-time analysis and robust security against phishing threats. If you need further details or specific components, feel free to ask!