PHISHING EMAIL DETECTION

INTRODUCTION

- In today's digital age, phishing emails pose a significant threat to cybersecurity, as attackers continually develop more sophisticated methods to deceive users.
- This project aims to build a robust phishing email detection system that leverages Natural Language Processing (NLP) and Machine Learning (ML) techniques.
- The system not only analyses the content of emails to detect phishing attempts but also scans email attachments with **ClamAV** and checks embedded links using the **VirusTotal API** for malware.
- This comprehensive approach enhances detection accuracy and helps safeguard sensitive information from cyber threats.

LITERATURE REVIEW

Project Overview

- Project focuses on developing a phishing detection system using machine learning techniques.

- It also aims to enhance detection accuracy by checking email attachments with ClamAV and analyzing links with the VirusTotal API for malware detection.

- Additionally, it involves comparing input emails against a phishing email dataset and integrating this functionality with Django for a comprehensive web-based solution.

COMPARISON WITH EXISTING RESEARCH

[ScienceDirect (5th International Conference on AI in Computational Linguistics) - Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey]

- Phishing emails are a significant cybersecurity threat, with a rising number of attacks targeting users to steal sensitive information.
- This literature survey explores the use of Natural Language Processing (NLP) and Machine Learning (ML) techniques in detecting phishing emails.
- While NLP and ML are crucial in identifying such emails, challenges in deep semantics analysis persist.

COMPARISON WITH EXISTING RESEARCH

[ScienceDirect (5th International Conference on AI in Computational Linguistics) - Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey]

- The survey underscores the importance of combining NLP and ML for effective detection, highlighting the need for more advanced solutions to combat evolving phishing techniques.

- Current research focuses on ML and NLP techniques for phishing detection, emphasizing feature selection and classification.

COMPARISON WITH EXISTING RESEARCH

[ScienceDirect (5th International Conference on AI in Computational Linguistics) - Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey]

Project Comparison

- Project aligns with these findings by employing ML techniques for phishing detection.
- Additionally, we use of ClamAV for attachment scanning and VirusTotal for link analysis complements the ML approach by integrating multiple layers of security.
- This multifaceted approach could address some of the challenges highlighted in the literature, such as the limitations in deep semantic analysis.

COMPARISON WITH EXISTING RESEARCH

[ScienceDirect (5th International Conference on AI in Computational Linguistics) - Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey]

Significance of Combining NLP and ML

- **Existing Research:** The literature emphasizes the importance of combining NLP and ML techniques for effective phishing detection. This approach helps in addressing the evolving nature of phishing attacks by leveraging both linguistic and statistical insights.

- **Our Project:** Our project's integration of NLP with ML, along with additional checks via ClamAV and VirusTotal, reflects the literature's emphasis on a multifaceted approach. This combination aligns with the current research focus and can potentially provide a more robust solution for phishing detection.

ISSUES TO BE ADDRESSED

- **Evolving Phishing Techniques:** Ensure the system can adapt to new and emerging phishing tactics, which may involve sophisticated social engineering or novel linguistic tricks.

- **Integration with Existing Tools:** Seamlessly integrate with tools like ClamAV and VirusTotal to enhance overall detection capabilities, ensuring compatibility and minimal false positives/negatives.

- **Real-Time Detection:** Develop a system that can analyze and respond to incoming emails in real-time, providing immediate feedback to users and reducing the window of vulnerability.

ISSUES TO BE ADDRESSED

- **False Positive/Negative Rates:** Minimize false positives (legitimate emails flagged as phishing) and false negatives (phishing emails passing as legitimate), maintaining high detection accuracy.

- **User-Friendly Interface:** Design a clear and intuitive interface that allows users to easily understand the results of the phishing detection, including why an email was flagged.

- **Continuous Learning and Updates:** Implement mechanisms for the system to learn from new phishing attempts, updating its models and detection strategies to stay effective against evolving threats.

MODULE DESCRIPTION

EMAIL PREPROCESSING MODULE

- **Functionality:** Extracts key components from incoming emails, such as the sender address, subject, body text, attachments, and embedded links.

- **Technologies Used:** Python libraries for email parsing (e.g., email and imaplib).

MODULE DESCRIPTION

MACHINE LEARNING MODULE

- **Functionality:** Trains and applies machine learning models to classify emails as phishing or legitimate based on extracted features.

- **Technologies Used:** Nltk ,pandas ,sklearn

MODULE DESCRIPTION

CLAMAV INTEGRATION MODULE

- **Functionality:** Scans email attachments for known malware and viruses using ClamAV.
- Flags emails with suspicious attachments and provides detailed reports on any detected threats.

- **Technologies Used:** pyclamd or ClamAV API for integration.

MODULE DESCRIPTION

VIRUSTOTAL API INTEGRATION MODULE

- **Functionality:** Analyzes URLs within the email body using the VirusTotal API to check for known malicious links.
- Flags emails containing suspicious or dangerous links, adding this information to the overall email risk assessment.

- **Technologies Used:** VirusTotal API, with requests handled via requests or similar Python HTTP libraries.

PROPOSED METHODOLOGY

DATA COLLECTION

- **Phishing Email Dataset:** Gather a comprehensive dataset of phishing and legitimate emails from publicly available sources and internal collections.

- Ensure the dataset is diverse in terms of email content, structure, language, and tactics used by attackers.

PROPOSED METHODOLOGY

PHISHING DATASET COMPARISON

- **Pattern Matching:** Compare incoming emails against a dataset of known phishing emails to identify common patterns and semantic similarities.

- Use these comparisons to refine the detection model and improve its ability to identify subtle phishing attempts.

PROPOSED METHODOLOGY

INTEGRATION OF ADDITIONAL DETECTION TECHNIQUES

- **ClamAV for Attachment Analysis:** Integrate ClamAV to scan email attachments for known malware and viruses, adding an extra layer of security.
- Incorporate the results into the overall phishing detection score.
- **VirusTotal for URL Analysis:** Use the VirusTotal API to analyze URLs in emails, identifying any links associated with known phishing or malicious websites.
- Factor these results into the final decision-making process.

PROPOSED METHODOLOGY

SYSTEM INTEGRATION AND WEB INTERFACE

- **Django Integration:** Develop a Django-based web interface where users can submit emails for analysis, view detection results, and receive detailed reports.

- Ensure that the system processes emails in real-time, providing immediate feedback and minimizing the time window in which a phishing attack could succeed.

REFERENCE

- **Kaggle Dataset:** https://www.kaggle.com/datasets
- **ClamAV Documentation:** https://docs.clamav.net
- **VirusTotal API Documentation:** https://developers.virustotal.com
- **Python Libraries: NLTK:** Natural Language Toolkit - https://www.nltk.org , scikit-learn: Machine Learning in Python - https://scikit-learn.org