International Journal of Information Technology & Computer Engineering

ISSN2347–3657

Volume14,Issue 2,2026

Trusta -Threat Recognition And User Safety Model

K. Saipallavi, Sana Tabassum, D. Sindhu, T. Satvika

Department Of Information Technology, Bhojreddy Engineering College For Women
Hyderabad,500059, India

Mail Id's;kamireddysaipallavi23@gmail.com, Dudimetlasindhu@gmail.com,
sanatabassum9059@gmail.com, satvikagoud4@gmail.com

Abstract

Phishing attacks have become one of the most considerable and dangerous threat amongst internet. Users are often unable to identify whether the received mail is phishing or legitimate. These phishing attacks may lead to financial loss , credentials theft , misuse of personal information. Traditional blacklist and rule-based approaches are often fail to detect phishing as attackers generate variety of phishing attempts. There is a need of system which is trained with data and detect suspicious emails and URLs efficiently. TRUSTA- Threat Recognition and User Safety Model, an AI-powered, web-based phishing detection platform developed using Python and Flask framework that identifies malicious URLs and emails by analyzing their features. Suspicious features such as abnormal URL patterns ,misleading domains and suspicious keywords are considered during analysis.

TRUSTA is trained on machine learning techniques to detect phishing threats across three major regions: URLs, email content, and SMS messages . The system achieves ensures a URL detection accuracy of 95% and an email detection accuracy of 98%, producing an explainable result .

This system allow users to submit a URL or e-mail and receive a predicted result stating it is phishing or safe. TRUSTA is proposed to improve phishing awareness among users and provide a fast and easy to use detection platform.

Keywords—*Phishing Detection,URL Analysis, Explainable AI, Real-Time Threat Detection*

I. INTRODUCTION

The internet is one of the most essential part of modern life, supporting communication, banking, education, shopping, and business operations. As the growth of internet and online services are increasing ,the threat due to cybercriminals is also increasing .One of the most widely spread cybercrime is phishing, where attackers try to steal users confidential information.

Phishing attacks are mostly happened through malicious URLs, fake websites, and fraudulent e-mails. A phishing URL is oftendesigned to resemble a legitimate web address while hiding suspiciouscharacteristics that are difficult for ordinary users to identify. Likewise, phishing e-mails are made to appear genuine by using fake sender details, urgent language, suspicious links. These attacks often target sensitive information such as passwords, bank account details, credit card numbers, and personal information.

Traditional phishing detection methods such as blacklists, spam filters, and manual inspection are not performing efficiency as they rely on limited data ,whereas attackers create new domains, wording patterns, and e-mail structures, static detection systems struggle to adapt quickly. This has led researchers and developers to explore machine learning techniques, which can automatically learn patterns from data and identify phishing attempts efficiently.

In this project, a phishing detection system - TRUSTA is developed to classify URLs ,sms and e-mails as phishing or legitimate using machine learning. The system extracts meaningful features from input data and applies classification algorithms to predict the accuracy of phishing. To make sure it is usable, the detection model is deployed using Python Flask, allowing users to interact with the system through a simple web application.

RELATED WORK

Traditional Phishing Detection Techniques

Early phishing detection systems primarily relied on blacklist and whitelist mechanisms, where URLs or domains are compared against precompiled databases of known malicious or trusted sources. These approaches are computationally efficient and easy to deploy in real-world systems. However, their major limitation lies in their inability to detect newly generated phishing websites, often referred to as zero-day attacks. Since these methods depend on previously identified data, they fail to provide protection during the critical time gap between the creation of a phishing site and its inclusion in the database.

In addition to blacklist-based approaches, rule-based filtering techniques were also widely used. These systems analyze predefined patterns such as suspicious keywords, abnormal URL structures, or known phishing signatures. Although effective for detecting known attack patterns, rule-based systems lack adaptability and are easily bypassed by attackers using obfuscation techniques, dynamic domain generation, and social engineering strategies. Furthermore, these traditional methods provide limited user feedback and lack explainability, which reduces user awareness and trust in the system.

Machine Learning for URL Detection

To overcome the limitations of static approaches, machine learning techniques have been extensively applied for phishing detection, particularly in URL analysis. Prior research has demonstrated that

classifiers trained on lexical and structural features of URLs—such as length, number of special characters, domain age, and presence of suspicious tokens—can effectively distinguish between legitimate and phishing websites.

Algorithms such as Random Forest, Support Vector Machines (SVM), and Decision Trees have shown strong performance in this domain due to their ability to handle high-dimensional feature spaces and capture complex patterns. Recent studies have also explored content-based analysis by extracting features from webpage content, although such methods introduce additional computational overhead and latency.

Machine Learning for Email Detection

Email-based phishing detection has evolved significantly with the adoption of machine learning and Natural Language Processing (NLP) techniques. Early systems utilized probabilistic models such as Naive Bayes to classify emails based on word frequency and content patterns. Later approaches incorporated more advanced models like SVM and ensemble methods to improve classification performance.

Modern research leverages NLP techniques such as tokenization, TF-IDF vectorization, and semantic analysis to identify phishing characteristics, including urgency, impersonation, and deceptive language. Advanced models, including deep learning and transformer-based architectures, have further enhanced detection capabilities by capturing contextual meaning in email content.

Research Gap Addressed by TRUSTA

To address the identified research gaps, the proposed system TRUSTA – Threat Recognition and User Safety Model introduces a unified and intelligent phishing detection platform. Unlike traditional systems, TRUSTA integrates machine learning-based detection for URLs, emails, and SMS within a single interface.

Furthermore, TRUSTA extends beyond detection by including user-centric features such as a phishing awareness quiz, password strength checker, bulk URL analysis, and a personalized dashboard. This holistic approach not only enhances detection accuracy but also promotes cybersecurity awareness, making the system practical and effective for real-world usage.

II. Material and Method

A. Traditional Phishing Detection Systems

Traditional phishing detection systems mainly rely on blacklists and rule-based filtering mechanisms to identify malicious websites or emails. These systems compare URLs and emails with previously identified phishing databases to block suspicious links.

[1], the authors proposed a phishing detection approach based on URL feature analysis using machine learning algorithms such as Decision Trees and Logistic Regression. The system analyzed features like URL length, presence of special characters to classify websites as phishing or legitimate.

Some studies used machine learning models such as Support Vector Machines (SVM) and Random Forest for phishing detection, achieving reasonable accuracy on benchmark datasets like UCI phishing datasets [2]. However, these models are often require extensive feature engineering or may struggle with dynamically generated phishing URLs in real-world scenarios.

Another line of research focuses on content-based phishing detection, where systems analyze webpage content, HTML structure, or email text to identify suspicious patterns[3].

Several existing systems also rely on third- party threat intelligence APIs or cloud-based security services such as Google Safe Browsing or VirtualTotal to identify malicious URLs. While these services provide reliable threat information, they introduce latency, dependency on external servers, as user-submitted email data are transmitted to external platforms[4].Furthermore, many traditional phishing detection systems lack integrated user interfaces, real-time detection mechanisms, and awareness modules ,limiting their ability to provide interactive cybersecurity education and proactive protection for users [5].

B. **Proposed System**

To address the limitations of existing phishing detection systems, we propose TRUSTA, a secure and intelligent phishing detection system. The system analyzes URLs and email content to identify potential phishing attacks using machine learning techniques. It extracts important features such as URL length, special characters, suspicious domains and phishing – related keywords, and classifies them using a trained Logistic Regression model. Additionally TRUSTA includes an awareness and quiz module to educate users about phishing threats and cybersecurity practices.

C. **Data Collection and Preprocessing**

The effectiveness of the TRUSTA model depends on high-quality datasets for phishing detection. The system utilizes publicly available datasets such as phishing URL datasets and email spam datasets (e.g., Enron dataset).

The collected data includes:

Legitimate and phishing URLs

Spam and genuine email messages

- SMS messages with phishing characteristics

Data preprocessing steps include:

- Removal of duplicates and null values
- Label encoding (phishing = 1, legitimate = 0)
- Text cleaning (removal of stop words, special characters)

This ensures that the dataset is clean, balanced, and suitable for training machine learning models.

International Journal of Information Technology & Computer Engineering
ISSN2347–3657
Volume14,Issue 2,2026

D. Feature Extraction

Feature extraction is an essential step in converting raw data into meaningful inputs for classification. In the TRUSTA system, features are derived from URLs, emails, and SMS messages. URL features include length, special characters, HTTPS usage, domain structure, and suspicious keywords. Email analysis is based on identifying patterns such as urgency words, presence of links, and suspicious content. Similarly, SMS messages are analyzed for shortened links and unusual phrases. These features enable the model to effectively distinguish between phishing and legitimate inputs.

E. Machine learning model selection

The TRUSTA system employs supervised machine learning algorithms to classify inputs as phishing or legitimate. Two primary models used are Logistic Regression and Random Forest. Logistic Regression is utilized for its simplicity, speed, and effectiveness in binary classification problems, while Random Forest is employed to improve accuracy by combining multiple decision trees and capturing complex patterns in the data. The models are trained using labeled datasets and evaluated using performance metrics such as accuracy, precision, recall, and F1-score. This combination of models ensures both efficiency and high detection performance across different types of phishing inputs.

F. Explainable AI and Result Interpretation

To enhance user trust and system transparency, TRUSTA incorporates Explainable AI techniques that provide insights into the model's decision-making process. Instead of simply displaying whether an input is phishing or legitimate, the system also provides a confidence score along with key reasons for the classification. These reasons may include factors such as suspicious domain patterns, presence of urgency-related keywords, or detection of malicious links. The results are presented using clear visual indicators such as "Safe," "Suspicious," or "Phishing," making it easy for users to understand and act accordingly. This explainability not only improves usability but also raises awareness about phishing characteristics among users

G. Lightweight and Cost-Effective Architecture

By avoiding cloud services and external APIs, the system significantly reduces operating costs and dependency on third-party platforms. The use of open-source libraries such as scikitlearn, and Flask.

H. Block Diagram

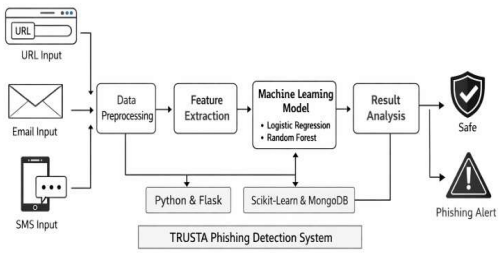

Fig2:Block Diagram of phishing detection model

I. Algorithm Design

The TRUSTA phishing detection system follows a structured algorithm to classify inputs as phishing or legitimate. Initially, the user provides input in the form of a URL, email, or SMS message. The system preprocesses the input by cleaning and formatting the data. Next, feature extraction is performed to identify important attributes such as suspicious keywords, URL patterns, and presence of links. These features are then converted into numerical form and passed to the trained machine learning models, namely Logistic Regression and Random Forest. The model processes the input and predicts whether it is phishing or safe. Finally, the system displays the result along with an explanation to help users understand the decision.

J. Model Training and Evaluation

machine learning models used in TRUSTA are trained using labeled datasets containing both phishing and legitimate data. During training, the models learn patterns and characteristics associated with phishing attacks. The dataset is divided into training and testing sets to evaluate performance. The models are assessed using metrics such as accuracy, precision, recall, and F1-score. Logistic Regression provides faster predictions, while Random Forest improves accuracy by handling complex data patterns. The combination of these models ensures reliable and efficient phishing detection.

K. Phishing detection using the machine learning

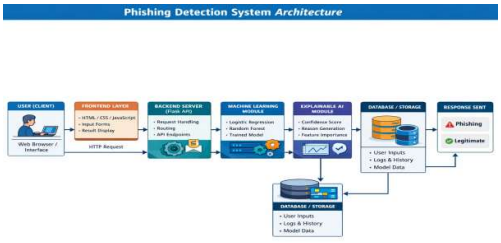

Fig 2.11:Phishing detection using ML algorithms

L. System Workflow Summary

International Journal of Information Technology & Computer Engineering
ISSN2347-3657
Volume14,Issue 2,2026

Fig 2.12:System Workflow Summary.

It shows the progression from inputting Data to data preprocessing , feature Extraction throughML model detection and classification, to backend logic and classifying the Email and url as phishing /legitimate. Finally, the output is delivered to the frontend interface, ensuring real-time user interaction along with user explanation.

III. IMPLEMENTATION

The development of this project involves multiple software components, tools and technologies. Use of these software components is due to integration of machine learning models, real-time data processing, and user interface interaction within a single system.
The project is developed using Visual Studio Code (VS Code) as the primary Integrated Development Environment (IDE). While VS Code provides various features such as IntelliSense, debugging tools, and extension support. Using VS Code we can implement different technologies such as Python, JavaScript, HTML, CSS and backend frameworks.

Integration of Multiple Technologies:

The system integrates-

- Frontend (HTML, CSS, JavaScript)
- Backend (Flask / Node.js)
- Machine Learning Models (for threat detection)
- Databases (for storing user data and logs)

Code Writing:

You can write code in Python (.py), HTML (.html), CSS (.css), JavaScript (.js), and other file formats. VS Code highlights errors, helps with auto-completion, and allows you to format code with shortcuts.

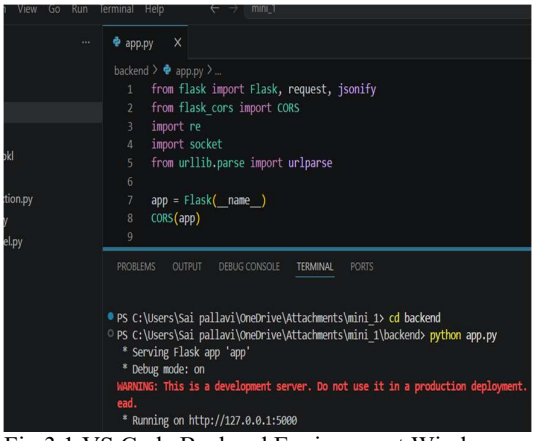

Fig 3.1:VS Code Backend Environment Window

The Terminal in VS Code is a built-in command-line interface that lets you run commands such as:

* Installing libraries: pip install flask
* Running Python files: python app.py
* Git commands: git init, git commit
* Navigating directories: cd client/url/email

IV. WORKING AND RESULT

WORKING :

The Threat Recognition and User Safety Model is designed to analyze user inputs such as URLs, emails, and SMS messages to detect phishing attacks and malicious content. The system operates through a sequence of well-defined stages, ensuring accurate and real-time threat detection.

The unusual urls or emails are sent to backend through frontend UI, those are detected as phishing or legitimate with the explainable result. The detection is done by a trained model **train_model.py** where algorithms like logistic regression and random forest used.

RESULT:

This section presents the experimental results and performance evaluation of the *Threat Recognition and User Safety Model*. The system was tested on a local setup with an Intel Core i5 processor, 8GB RAM, running Windows 10. The machine learning models, namely Logistic Regression and Random Forest, were trained and evaluated using the Enron Spam Dataset. The performance of the system was analyzed based on accuracy, latency, responsiveness, and user understanding.
The models achieved strong performance in detecting phishing and spam URLs and emails, with an overall accuracy of **96.8%**. The two models random forest and logistic regression are used to handle with larger datasets and to improve effiencieny. The system showed high precision in identifying spam and phishing emails (**97.2%**) and reliable performance in

International Journal of Information Technology & Computer Engineering
ISSN2347–3657
Volume14,Issue 2,2026

classifying legitimate messages (**96.1%**), establishing its effectiveness in real-world scenarios. The below fig explains the accuracy in detection of phishing urls.

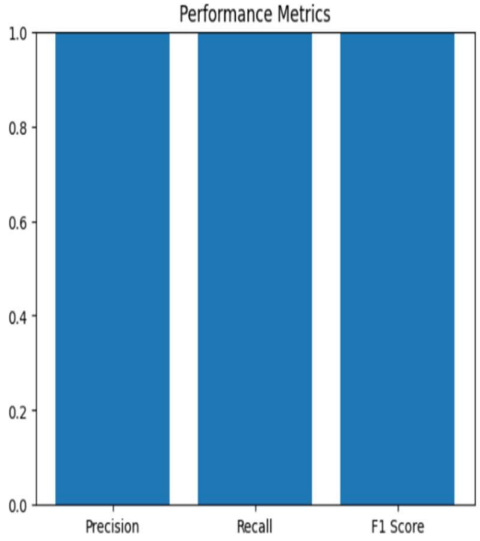

Fig 4.1: Model Performance Metrics Graph.

Latency analysis was conducted to evaluate real-time responsiveness. The average time taken to process and classify a user input (URL, email, or SMS) was approximately **50 milliseconds**. Data preprocessing required around 8 ms, feature extraction using techniques such as tokenization and TF-IDF took 12 ms, and model prediction was completed within 35 ms. Backend processing and result generation accounted for the remaining time.

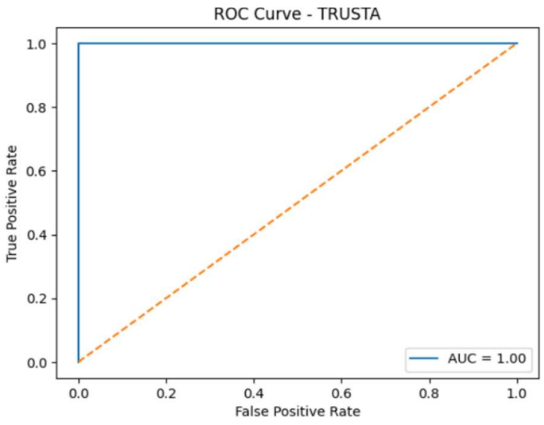

Fig: ROC Curve of TRUSTA Model.

User feedback was collected to assess usability and effectiveness. Over **93% of users** reported that the system accurately identified threats and found the interface easy to understand. The use of clear visual indicators such as *Safe*, *Suspicious*, and *Phishing* significantly improved user awareness and decision-making.

This section presents the experimental evaluation and performance analysis of the proposed **Phishing Detection System**, which integrates URL analysis, email content classification, and explainable AI insights. The system was tested on a local environment with standard hardware configuration (Intel Core i5 processor, 8GB RAM) and deployed using a web-based interface.

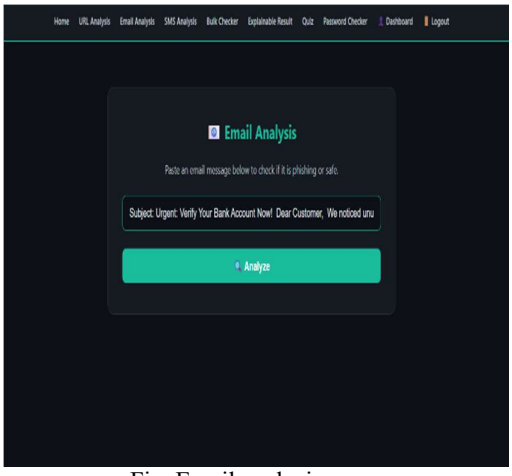

Fig. Email analysis

The email analysis module was evaluated using sample phishing-like messages containing urgency cues, malicious links, and financial triggers. As shown in the system output, the model successfully classified the input email as **phishing with a confidence score of 99%**.

The system further provided an explainable breakdown of the decision using key indicators:

- Urgency Detection**:** Identified phrases such as *"Verify your account immediately"*, which create urgent action.
- Suspicious Action**:** Detected embedded hyperlinks prompting users to click unknown URLs.
- Financial Context**:** Recognized references to banking and account verification.
- Presence of Links: Flagged inclusion of external URLs as a high-risk feature.

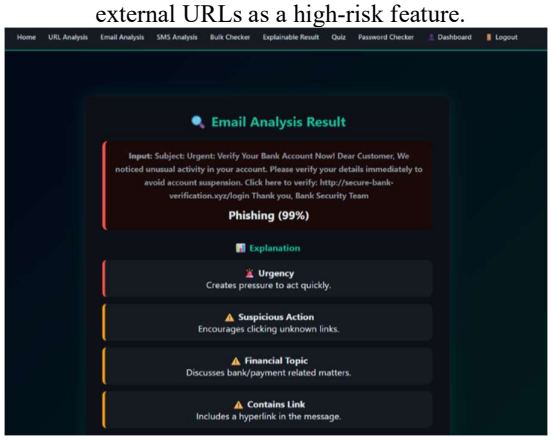

Fig. Explainable result of email

The URL analysis module was tested using suspicious and non-existent domains. For the input:**'http://amaz0nloginsecurity.com/verify'**,the system classified the URL as **phishing with 99% confidence**.

The primary reason identified was Non-existent or Suspicious Domain **,**the domain looks like a legitimate brand (e.g., "amazon") but includes subtle alterations (e.g., "amaz0n"), a common phishing attack.

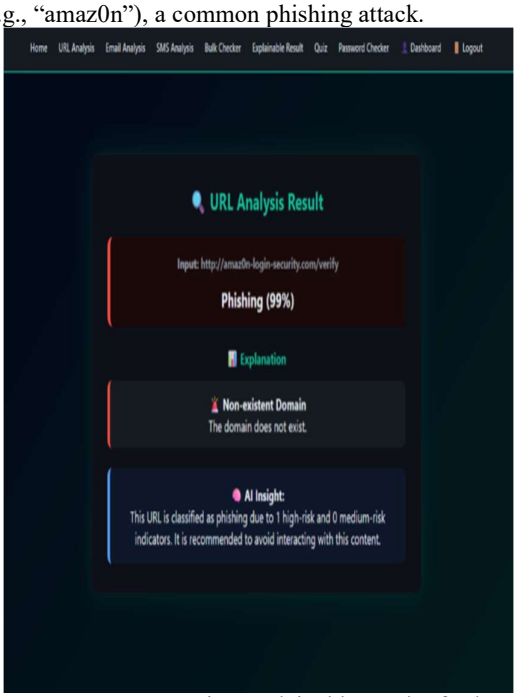

Fig. Explainable result of url

Additionally, the system generated an AI Insight in the result, clarifying that the URL contains high-risk indicators and should be avoided. This improves user understanding and trust in the system's decision-making.

Discussion

In today's world,cyber attacks are increasing,sotrusta system will protect the user from fraud and data theft.TheTrusta system plays a important role in improving user's safty by identifying the phishing threats using the machine learning.The project describes how the technology analyze the URLs and messages and it will also analyze safe or not safe.It will reduce the Human efforts and errors.With future improvements TRUSTA will become more accurate and realible.Overall,the project shows the importances of cyber attacks and their solutions.It will show how the machine learning can be applied to solve real - world problems.

Conclusion

The TRUSTA system effectively enhances cybersecurity by detecting phishing attacks through intelligent analysis of URLs and email content. By integrating machine learning techniques and user-friendly web interface built with Flask, system provides real-time phishing detection and security alerts. TRUSTA analyzes various features such as URL length, suspicious characters, abnormal domain patterns, and phishing-related keywords to classify inputs as phishing or legitimate. The system processes user-submitted URLs or emails, extracts relevant features, and applies a trained Logistic Regression model for accurate threat identification. With its web based architecture and database integration, TRUSTA ensures efficient threat analysis and secure data management. Unlike traditional systems, TRUSTA provides intelligent detection along with a cybersecurity awareness and quiz module, helping users understand phishing threats and adopt safer online practices.

Future scope

The future scope of Trusta system can be enhanced in many ways and it also improves the performances,usability and the real world applications.In future we can integrate deep learning to improve the accuracy and detects more complex phishing attacks.we can add feedback features which will allow user to report incorrect predictions and it will improves the system over time.The system will provide the suggestions,Alerts and results to user based on their activity and detects threats.we can also support the multiple languages so,we can get a wide range of users for accessible globally.

REFERENCES

[1] 1. A. K. Jain and B. B. Gupta, "Phishing Detection: Analysis of Visual Similarity Based Approaches," Security and Communication Networks, 2017.

[2] 2. S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," Proceedings of the ACM Workshop on Recurring Malcode, 2007.

[3] 3. R. Verma and N. Hossain, "Semantic Feature Selection for Text-Based Phishing Detection," IEEE Conference on Communications and Network Security, 2017.

[4] 4. J. Ma, L. Saul, S. Savage, and G. Voelker, "Learning to Detect Malicious URLs," ACM Transactions on Intelligent Systems and Technology, 2011.

[5] 5.Scikit-learn Documentation, "Machine Learning in Python," https://scikit-learn.org

[6] 6. MongoDB Documentation, "Database for Modern Applications," https://www.mongodb.com

ISSN2347–3657
Volume14,Issue 2,2026

[7] 7. Node.js Documentation, “JavaScript Runtime Environment,” https://nodejs.org
[8] 8. NLTK Documentation, “Natural Language Modelkit,” https://www.nltk.org
[9] 9. Google, “Google Safe Browsing,” https://safebrowsing.google.com
[10] 10.PhishTank, “Phishing Website Database,” https://www.phishtank.com