International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 10 | October - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

DETECTION OF PHISHING ATTACKS USING TLS TRAFFIC

Mr. S. Nitin Reddy
Department of Computer Science and Engineering
GMRIT
Rajam, India
20341a05h4@gmrit.edu.in

Dr. V. Srinadh
Sr. Associate Professor
Department of Computer Science and Engineering
GMRIT
Rajam, India
srinadh.v@gmrit.edu.in

Mr. P. Kannam Naidu
Department of Computer Science and Engineering
GMRIT
Rajam, India
20341a05e8@gmrit.edu.in

Mr. P. Charan
Department of Computer Science and Engineering
GMRIT
Rajam, India
20341a05f1@gmrit.edu.in

Mr. Sk. Musharaf
Department of Computer Science and Engineering
GMRIT
Rajam, India
20341a05h0@gmrit.edu.in

Mr. R Karteek
Department of Computer Science and Engineering
GMRIT
Rajam, India
20341a05f2@gmrit.edu.in

Mr.Dhushyanth
Department of Computer Science and Engineering
GMRIT
Rajam, India
21345a0516@gmrit.edu.in

***Abstract*--** Phishing attacks trick people into disclosing personal information like passwords and credit card numbers with the intention of stealing or corrupting sensitive data. Some of the existing phishing detection techniques like Content-Based, URL-Based, List-Based may not provide good security, because the attacker using the latest technologies to steal data. The Transport Layer is responsible for ensuring reliable and efficient data transfer between end systems (computers or devices) over a network. TLS is a cryptographic protocol used to secure communication over a computer network. In this work, TLS 1.2 and TLS 1.3 features will be used to train machine learning algorithms to increase the performance metrics. Phishing and Legitimate URLs are used to generate TLS features using the cyber security tools like Selenium and Wireshark. These TLS features will help to detect phishing URLs efficiently. This work includes machine learning algorithms like XGBOOST, Random Forest and Multi-Naïve Bayes will be used.

Keywords: Phishing, Legitimate URL's, Light GBM, TLS 1.2, TLS 1.3, URL, Random Forest.

1. INTRODUCTION

Phishing is a method of online fraud where users are being targeted to gain access to the computer systems and private information for personal gains or benefits by the attackers. Phishing is a method of online fraud where users are being targeted to gain access to the computer systems and private information for personal gains or benefits by the attackers. The personal information of the users is being robbed by the attackers using this phishing attack. It is a technique in which the attackers pose themselves as legitimate entities to gain the users sensitive information. With the increase in number of the internet users over the years, cybercrime is increasing at a high rate. Over the last few years, phishing is one of the most used attacks and it is used to attack in many ways on targeted users. With the advancements in cyber-security over the last few years, many techniques evolved to detect phishing-related activities. Http scheme, Black list, White list are some old techniques for the detection of phishing websites. These old techniques are based on some rules and these rules are interpretable, and the logic of the rules is limited, so the detection methods based on the rules can be easily cracked and used by attackers. As technology continues to advance, vulnerabilities and traditional exploitation methods, such as phishing links, are becoming more easily exploited. To address these challenges, the use of machine learning algorithms is emerging as a powerful solution.

In this project, the analyse TLS (Transport Layer Security) traffic by capturing the necessary attributes using the Wireshark tool. The process begins with loading a URL in a web browser, followed by the activation of the Wireshark tool to monitor network traffic. Specifically, we focus on identifying TLS packets within the network traffic data and extract specific attributes from these TLS packets. Subsequently, creation a pcap file containing only the selected packets using the Wireshark tool.

Subsequently, the pcap file is transformed into an XML file containing only the essential attributes. These chosen attributes are then inputted into our most potent machine learning models, namely XG Boost, Logistic Regression, and Random Forest. Here's a concise overview of these models: These machine learning models play a pivotal role in analysing the XML data and producing valuable insights or predictions based on the provided attributes.

In this project, creation of a dataset by combining legitimate and phishing links from various online sources. The attributes necessary for analysing these URLs were obtained using the Wireshark tool, focusing on capturing TLS versions 1.2 and 1.3 packets, with particular attention to the "Server Hello" and "Client Hello" packets. It's essential to emphasize that this process of attribute extraction was carried out manually for all the collected URLs.

2. LITERATURE SURVEY

[1] This paper proposed a learning-based phishing detection technique from transport layer security (TLS) 1.2 and TLS 1.3 traffic. The proposed model detects phishing URLs at the transport layer and classifies them as legitimate or phishing with high accuracy. The one limitation of the proposed model, which is that it may not detect phishing if the traffic is not HTTPS. This is because the features used for phishing detection are based on TLS 1.2 and TLS 1.3 traffic. It will perform less when the traffic is not HTTPs. The proposed model achieves high accuracy when compared to the models which were trained by using existing features in URL in detecting phishing. It produced an accuracy of 93.63% for Random Forest (RF), 95.07% for XGBoost (XGB), and the highest accuracy of 95.40% for Light GBM (LGBM).

[2] The development of an ensemble machine learning-based model to detect phishing emails using a proposed Remove-Replace Feature Selection Technique (RRFST). This paper results indicate that the proposed RRFST produces remarkable performance with 99.27% accuracy using an ensemble of C4.5 and CART with only 11 features. The proposed ensemble model with the Remove-Replace Feature Selection Technique (RRFST) achieves a remarkable accuracy of 99.27% in detecting phishing emails. The paper contributes to the development of an efficient anti-phishing tool by incorporating feature selection in the model development process. The performance of the model was evaluated using a specific dataset and it may not generalize well to other datasets and does not discuss the computational complexity of the proposed RRFST algorithm and its scalability to larger datasets.

[3] The proposed machine learning-based approach for phishing website detection is faster, simpler, and more interpretable than current black box methods. The efficiency, accuracy, and execution time of the final model are evaluated and reported in the paper. This paper does not provide a detailed comparison of the proposed machine learning-based approach with other existing phishing detection methods, such as content-based, blacklisting, or rule-based approaches. This paper includes measures such as efficiency, accuracy, and model execution time will be evaluated against the final model.

[4] The proposed system achieves an accuracy of 98.5%, a false positive rate of 0.5%, and a false negative rate of 1%. These results indicate that the proposed system is effective in detecting phishing web pages with high accuracy and low false positive rate. The paper proposes a machine learning-based URL detection system to identify phishing web pages by combining the URL of the web page and the URL of the web page source code as features. The system aims to provide high accuracy and low false positive rate detection results for unknown phishing pages.

[5] The paper proposes a model for detecting phishing websites using a combination of tiny-Bert feature extraction and Stacking algorithm-based classification. This model achieves high accuracy rates without requiring manual feature extraction. The paper reports the performance of the proposed phishing website detection model based on tiny-Bert Stacking using a dataset of real phishing websites. The performance matrix of the model is as follows, Accuracy rate: up to 99.14%, Recall rate: up to 99.13%.The model uses a combination of tiny-Bert feature extraction and Stacking algorithm-based classification, which allows for the learning of semantic and long-range dependent features in URLs.

[6] The framework is based on machine learning and uses only URL features for classification, without the need to visit the webpage or use any third-party services. The framework uses a limited number of features (30) for classification, which reduces the time required for feature extraction. Detecting phishing URLs based on lexical features extracted from the URL string. The proposed framework achieves high accuracy on benchmark datasets and outperforms existing approaches. Random Forest classifier achieved the highest accuracy on all datasets. It achieved an accuracy of 96.25% and 94.65% on the Kaggle datasets and an accuracy of 92.2%, 91.63%, 94.80%, and 96.85% on benchmark datasets.

[7] The paper proposes a new approach called PhishDet, which combines Long-term Recurrent Convolutional Network and Graph Convolutional Network using URL and HTML features to detect phishing websites. The paper use advanced classification metrics like accuracy, precision, recall, and f1 score to understand the model's performance better. PhishDet recorded a 96.42% detection accuracy, with a 0.036 false-negative rate, and achieved a 99.53% f1-score with a public benchmark dataset. The objective of the paper is to propose a new way of detecting phishing websites using Long-term Recurrent Convolutional Network and Graph Convolutional Network with URL and HTML features.

[8] In this paper a new dataset create named Phishing Index Login URL (PILU-90K) to prove their statements and present a Logistic Regression model which, combined with Term Frequency -Inverse Document Frequency (TF-IDF) feature extraction, obtains 96.50% accuracy on the introduced login URL dataset. The paper discusses the advantages of using URL-based detection models for phishing detection. Some of the advantages mentioned in the paper are : Fast computation: URL-based detection models do not require loading websites, which makes the detection process faster. Content-based detection methods can also be less effective as they require loading the website, which can be time-consuming and language-dependent.

[9] The paper proposes a multidimensional feature phishing detection approach based on deep learning, which has several advantages: The proposed approach does not require any prior knowledge about phishing, making it more accessible and easier to use. The approach was tested on a dataset containing millions of phishing URLs and legitimate URLs, and the accuracy reached 98.99%, while the false positive rate was only 0.59%.The paper also conducts extensive experiments on a dataset to evaluate the performance of the proposed approach in terms of accuracy, false positive rate, and speed

[10] The paper aims to compare different studies detecting phishing attacks for each AI technique and examine the qualities and shortcomings of these methodologies. And provides a comprehensive set of current challenges of phishing attacks and future research directions in this domain. It mentions that several studies have reported achieving more than 95% accuracy using the Random Forest (RF) classification method. it mentions that several studies have reported achieving more than 95% accuracy using the Random Forest (RF) classification method. The disadvantage in this paper is the lack of a unified framework for evaluating and comparing different phishing detection techniques.

3. METHODOLOGY

In this work, TLS features were added to train the models. The features that present in TLS layer like number IPv4 and IPv6 packets, TTL(Time-to-Live) were considered which helped to improve the efficiency of the model in detecting phishing URLs. The features were collected from the specific packets such as “ServerHello” and “ClientHello” packets which were related to TLSv1.2 and TLSv1.3 layers. These features were collected using Wireshark tool.

3.1 Dataset

The dataset encompasses 4,000 phishing URLs and 4,000 legitimate URLs, all collected from online sources. Initially, URL features were extracted and employed for model training. To enhance the model's performance, additional features were incorporated, including counts of IPv4 and IPv6 packets, as well as Time-to-Live (TTL) values, gathered using the Wireshark tool. By integrating these network-related features into the training data, the model exhibited a substantial improvement in its ability to distinguish between phishing and legitimate URLs, ultimately producing superior results. This approach capitalizes on both URL and network-level attributes, significantly enhancing the model's phishing detection capabilities.

3.2 Data Preprocessing and Feature Extraction:

Data preprocessing techniques like Tokenization, Stemming were done to URLs. At first, the special characters, number, spaces were removed.It was done by using Tokenization. Then stemming was applied to get the root words. Later feature extraction was done by using Countvectorizer.

3.3 XGBoost:

XGBoost is an advanced gradient boosting technique for machine learning that combines multiple weaker models, typically decision trees, in an iterative manner to create a more accurate ensemble model. The base learners are decision trees, which base their decisions on feature values. An objective function, which varies based on the type of problem, is what the training process seeks to optimize. 'binary:logistic' or'multi:softmax' are frequently used for classification, while'reg:squarederror' is appropriate for regression. Decision trees are incrementally added during training, and the model outputs predictions depending on the existing ensemble. The model's updates are guided for greater accuracy by the gradient of the loss function with respect to the anticipated values.

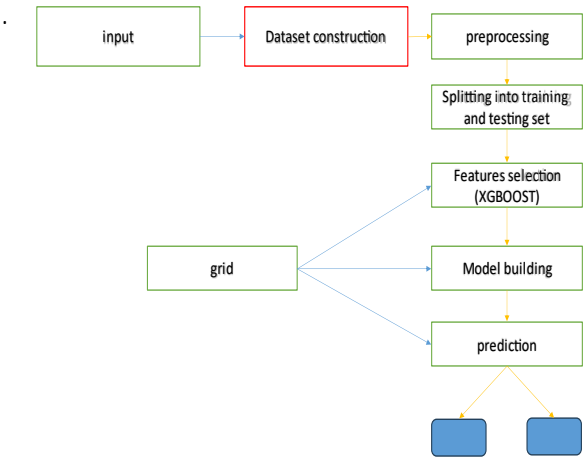

Figure: Architecture of XGBoost

3.4 Logistic Regression:

Logistic regression is a fundamental machine learning and statistical algorithm used for binary and sometimes multi-class classification. The preprocessed dataset is used to train a logistic regression model. It is perfect for binary classification, such as phishing detection, because it performs well at establishing a decision boundary that separates real from phishing traffic based on extracted attributes.On a different validation or test dataset, the model's performance is evaluated using measures like accuracy, precision, recall, and F1-score. The model's capacity to distinguish between authentic and phishing traffic is evaluated using this approach. The sigmoid function is used in logistic regression to convert expected values into probabilities between 0 and 1. The output is shaped into a "S" curve by this function.

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + \ \ldots + b_nx_n$$

3.5 Random Forest:

Random Forest is a versatile supervised machine learning algorithm used for both classification and regression tasks. It is preferred for managing complicated datasets, lowering overfitting, and determining the significance of features. The fundamental idea is to construct numerous decision trees during training and combine their forecasts for greater accuracy. Random Forest is built on decision trees, which are basic hierarchical models. To arrive to leaf nodes with final predictions, each tree splits data recursively based on input features. This method aids in feature selection by offering insights into feature relevance. The approach increases forecast accuracy, manages model complexity, and permits performance evaluation by combining the strengths of several trees. The ultimate output in classification is determined by the most prevalent class among each prediction made by the tree.

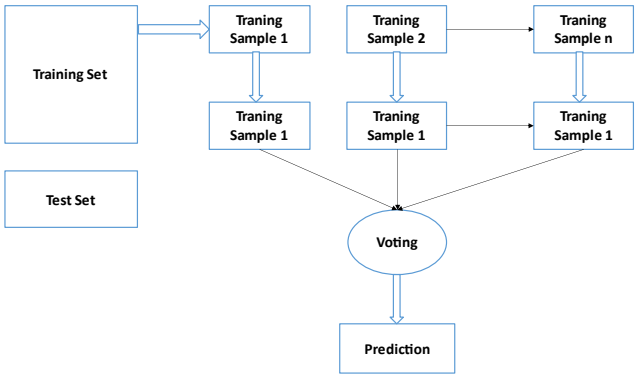

Figure: Architecture of Random Forest

4. Results & Discussions:

Our phishing detection model demonstrates promising performance, as evidenced by the evaluation metrics, including F1-score, support and recall scores. The model excels in several areas. It detects the phishing websites in a very good way to produce consistency.

The proposed model was trained by adding additional TLSv1.2 and TLSv1.3 features which helps to produce good results. These additional features helps to train model as it includes features like number of IPv4, IPv6 packets, number of intermediate nodes it travels between host and destination, etc. It was observed that by adding these features, we observed a good increase in improving the overall performance metrics like accuracy, F1-score, recall and support.

MODEL	URL Features	URL + TLS V-1.2, V-1.3 Features
Logistic Regression	94.85	96.00
Random Forest	95.00	96.10
XG Boost	93.08	95.25

Table: Accuracies produced by proposed models

Also, to evaluate a model some other metrics also need to be considered. Recall, Precision and F1-score were considered to evaluate. After adding the additional TLS features it was observed that the prediction of URLs produced good and consistent results.

Recall: Recall measures the ability of a model to correctly identify all relevant instances, which are the positive instances. It's also known as sensitivity or the true positive rate.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Precision: Precision measures the ability of a model to correctly identify only the relevant instances among those it classifies as positive. It quantifies how many of the positively classified instances were actually positive.

$$Precision = \frac{TruePositive}{True\ Positive + False\ Positive}$$

F1-score: The F1 Score is the harmonic mean of precision and recall. It provides a balance between these two metrics, allowing you to assess the model's overall performance.

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Model	Precision	Recall	F1-score
Logistic Regression	0.95	0.96	0.96
Random Forest	0.96	0.96	0.97
XG Boost	0.94	0.95	0.95

Table: Comparison table for Performance metrics

5. CONCLUSION & FUTURE SCOPE:

In phishing URL detection, models employing Random Forest, Logistic Regression, and XGBoost were trained and tested using URL features. To get better results, some additional features in TLS layer were included. Some of them including counts of IPv4, IPv6, and Time-to-Live (TTL), were integrated to enhance model performance. These features altogether helped to increase the model efficiency and effectiveness to encounter the phishing URLs. The working model was developed by these additional features and it helped to produce better results.

Looking ahead, the future scope of this project involves identifying further features that can bolster the model's effectiveness. Exploration of deep learning techniques is a promising avenue for improvement, as they can autonomously extract valuable patterns from data. Furthermore, testing the model with a variety of other algorithms may yield superior results, as different models have distinct strengths. Continuous improvement, encompassing feature engineering, hyperparameter tuning, real-time detection, and integration with security systems, will be crucial in maintaining the model's relevance in the ever-evolving landscape of phishing threats. Regular updates, monitoring, and collaboration with cybersecurity experts are essential components of the project's future success.

6. REFERENCES:

[1] Kumar, M., Kondaiah, C., Pais, A. R., & Rao, R. S. (2023). Machine learning models for phishing detection from TLS traffic. *Cluster Computing*, 1-15.
[2] Jalil, S., Usman, M., & Fong, A. (2023). Highly accurate phishing URL detection based on machine learning. *Journal of Ambient Intelligence and Humanized Computing*, *14*(7), 9233-9251.
[3] Hota, H. S., Shrivas, A. K., & Hota, R. (2018). An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique. *Procedia computer science*, *132*, 900-907.
[4] Ghareeb, S., Mahyoub, M., & Mustafina, J. (2023, January). Analysis of Feature Selection and Phishing Website Classification Using Machine Learning. In *2023 15th International Conference on Developments in eSystems Engineering (DeSE)* (pp. 178-183). IEEE.
[5] Wu, C. Y., Kuo, C. C., & Yang, C. S. (2019, August). A phishing detection system based on machine learning. In *2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)* (pp. 28-32). IEEE.
[6] He, D., Lv, X., Zhu, S., Chan, S., & Choo, K. K. R. (2023). A Method for Detecting Phishing Websites Based on Tiny-Bert Stacking. *IEEE Internet of Things Journal*.
[7] Selvaraj, P., Burugari, V. K., Benadit, J., & Kanmani, P. (2022). Phishing attack detection using Machine Learning. *Measurement: Sensors*, *24*, 100476.
[8] Ariyadasa, S., Fernando, S., & Fernando, S. (2022). Combining long-term recurrent convolutional and graph convolutional networks to detect phishing sites using URL and HTML. IEEE Access, 10, 82355-82375.
[9] Sánchez-Paniagua, M., Fernández, E. F., Alegre, E., Al-Nabki, W., & Gonzalez-Castro, V. (2022). Phishing URL detection: A real-case scenario through login URLs. IEEE Access, 10, 42949-42960.
[10] Yang, P., Zhao, G., & Zeng, P. (2019). Phishing website detection based on multidimensional features driven by deep learning. IEEE access, 7, 15196-15209.
[11] Yun, X., Wang, Y., Zhang, Y., Zhao, C., & Zhao, Z. (2022). Encrypted TLS Traffic Classification on Cloud Platforms. IEEE/ACM Transactions on Networking, 31(1), 164-177.
[12] Basit, A., Zafar, M., Liu, X., Javed, A. R., Jalil, Z., & Kifayat, K. (2021). A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommunication Systems, 76, 139-154.
[13] Rao, R. S., Vaishnavi, T., & Pais, A. R. (2019). PhishDump: A multi-model ensemble based technique for the detection of phishing sites in mobile devices.
[14] Xiang, G., Hong, J., Rose, C. P., & Cranor, L. (2011). Cantina+ a feature-rich machine learning framework for detecting phishing web sites.
[15] Chen, Y. H., & Chen, J. L. (2019). Ai@ ntiphish—machine learning mechanisms for cyber-phishing attack. IEICE Transactions on Information and Systems, 102(5), 878-887.