Detection of Malware Trojans in Software using Machine Learning

Prof. Aravinda Thejas Chandra[1], Ms. Sindhu R[2], Ms. Spoorthi H[3], Ms. Prerana R P[4], Ms. V Bhavana[5]

Associate Professor, Department of Information Science and Engineering[1]
Students, Department of Information Science and Engineering[2,3,4,5]
S J C Institute of Technology, Chikballapur, India
thejaschandra@gmail.com, sindhu.17r@gmail.com, spoorthigowdah007@gmail.com
ppreranarp@gmail.com, bhavnamurthy227@gmail.com

Abstract*:* *As the prevalence of malicious software, particularly trojans, continues to pose a significant threat to the integrity and security of computer systems, the need for effective detection mechanisms becomes paramount. This research presents a comprehensive approach to the detection of malware trojans in software, leveraging advanced techniques from machine learning, behavior analysis, and signature-based methods. The proposed system employs a hybrid model that combines the strengths of static and dynamic analysis to enhance detection accuracy. Static analysis focuses on examining code structures and identifying potential indicators of trojans presence, while dynamic analysis observes the software's behavior during execution to uncover malicious activities that may evade static analysis. Machine learning algorithms play a crucial role in training the detection system to recognize patterns indicative of trojans behavior. The model is trained on a diverse dataset of both benign and malicious software samples, enabling it to adapt and evolve to emerging threats. Feature extraction techniques are applied to capture the essential characteristics of trojans, contributing to the model's ability to generalize effectively. Furthermore, the system incorporates a signature-based approach, utilizing known patterns and signatures of trojans to quickly identify and mitigate known threats. Regular updates of signature databases ensure the system remains current and capable of detecting the latest trojan variants. To evaluate the effectiveness of the proposed approach, extensive testing is conducted on a variety of software samples, including both well-established trojans and newly emerging threats. The results demonstrate the system's robustness and efficiency in detecting trojan activity, with a low false positive rate. In conclusion, the presented research provides a holistic and adaptive solution for the detection of malware trojans in software. By combining static and dynamic analysis with machine learning and signature-based methods, the proposed system offers a versatile defense against the evolving landscape of trojan threats, contributing to the overall cybersecurity resilience of computer systems..*

Keywords: Trojans, Malware, cybersecurity, static analysis, dynamic analysis , signature based approach, Random forest, machine learning algorithms, efficiency, resilence

I. INTRODUCTION

In the world of anti-malware technology, they use a combination of methods to detect malicious software. These methods involve few steps: first, they gather information about the software, and second, they decide if it's malware or not. Feature Extraction: To do this, they gather specific details about the software. There is a way to do it: Static Method: They look at the code of the software without running it. It's like inspecting a book's cover without reading it. Classification: After gathering these details, they use computer programs machine learning algorithms to decide if the software is harmful or not based on the information collected.

By incorporating measures of both electrical power usage and network traffic, this system offers a more comprehensive perspective on software behavior. It's fascinating how they found that focusing on the power usage from specific components, like the +12V CPU rails, was particularly effective. And the use of machine learning techniques like Random Forest to analyze these features is a smart choice, given its success in identifying malware.

DOI: 10.48175/IJARSCT-18083

The idea of combining power-based and network traffic-based features to achieve the best results makes a lot of sense. It's all about looking at multiple dimensions of software behavior to gain a deeper understanding of its nature. Plus, the fact that they were able to identify the smallest set of features necessary for malware detection highlights the efficiency and precision of their approach.

Overall, this sounds like a promising advancement in the ongoing battle against malware. It's always exciting to see new methods emerge that push the boundaries of what's possible in cybersecurity.

Utilizing datasets from Kaggle pertaining to cybersecurity, various machine learning algorithms including decision trees and Random Forest were employed to detect trojans. Each algorithm's efficiency was rigorously assessed, culminating in a recommendation to end users advocating for the adoption of the most effective algorithm, Random Forest, for trojan identification. Random Forest excels due to its ensemble nature, which enables it to handle diverse data types effectively, mitigate overfitting, and discern intricate patterns within the dataset. By leveraging Random Forest's capabilities and providing evidence-based guidance, this approach enhances the accuracy and reliability of trojan detection systems. Such data-driven methodologies not only bolster cybersecurity efforts but also empower end users with actionable insights derived from empirical analyses, ensuring a proactive stance against evolving cyber threats.

II. LITERATURE REVIEW

Here's a literature survey highlighting key works and approaches :

“Trojan Detection System Using Machine Learning Approach”,[1] by Mohd Faizal, Izham Jaya, Zahian Ismail, Ahmad Firdaus (Indonesian Journal of Information Systems).This survey provides an overview of a different type of machine learning classification algorithm is used, and the results are evaluated in terms of performance.

“Detecting Trojan Horses Based On System Behaviour Using Machine Learning Method” [2]by Yu-Feng Liu, Li-Wei Zhang, Jain Liang, Sheng Qu, Zhi-Qiang Nil. This work presentsgathering samples of Trojan horse malware from real network environments and classify them using a scanner. We use WMI (Windows Management Instrumentation) tools to do this.

“Trojan Traffic Detection Based On Machine Learning”[3] byMa Zhongrui, Huang Yuanyuan, Lu Jiazhong. Aims to use features extracted from network traffic data to identify and detect Trojan traffic accurately. It appears that CICF low meter is a tool used for this feature extraction process, and the extracted features are aligned with characteristics observed in Trojan network behavior.

“Malware Detection Using Deep Learning Algorithms”, Aurum Journal Of Engineering Systems And Architecture ”[4] by Mohammed Altaiy, Incilay Yildiz,Bahadir Ucan.The aim of this study is to benefit from deep learning algorithms in the classification of malware. It is to determine the most effective classification algorithm by comparing the performances.

“Malware Detection Techniques: A Survey**"[5]** by YamjalaSupriya, Dammu Sowjanya, Gautam Kumar, Deepali Yadav, Devarakonda lakshmi kameshwari. This study involves the methodology of the paper likely involves a systematic and structured approach to surveying and synthesizing existing knowledge on malware detection techniques, with the aim of providing valuable insights to researchers, practitioners, and policymakers in the cybersecurity domain.

“Malware Detection”[6]by Subhadeep Chakraborty. This worklikely involves collecting malware datasets, extracting features, training machine learning models, and evaluating their performance. The study aims to identify effective techniques for detecting malware, with implications for cybersecurity.

"Malware Detection Using Ensemble Machine Learning Algorithms"[7] by B. M. Lakshmi and V. Vijayakumar. Ensemble methods like Random Forest or Gradient Boosting are employed to combine multiple base learners for improved malware detection accuracy. The dataset is likely split into training and testing sets for model evaluation, with metrics such as accuracy, precision, recall, and F1-score used to assess performance. Hyperparameter tuning and cross-validation techniques may also be applied to optimize model performance.

"Feature Selection and Machine Learning Techniques for Malware Classification"[8] by Fatima Abu Salem and Azzam Mourad . This involves Machine learning algorithms such as decision trees or support vector machines are trained on the selected features. Evaluation is likely performed using metrics like accuracy, precision, recall, and F1-score to assess classification performance.

DOI: 10.48175/IJARSCT-18083

"Deep Learning-Based Malware Detection"[9] by Rajesh Saini and Sushil Chauhan.The paper likely discusses the utilization of deep neural networks to analyze malware behavior and extract features automatically. It may involve training convolutional neural networks (CNNs) or recurrent neural networks (RNNs) on malware datasets and evaluating their performance in terms of accuracy and efficiency.

"A Survey on Malware Detection Using Machine Learning Techniques: Classification, Features and Datasets" [10]by Raja Vikramaditya Singh Chandel and B. B. Gupta (2019).

These works collectively contribute to our understanding of efficient Trojan detection techniques by using machine learning algorithms.

III. METHODOLOGY

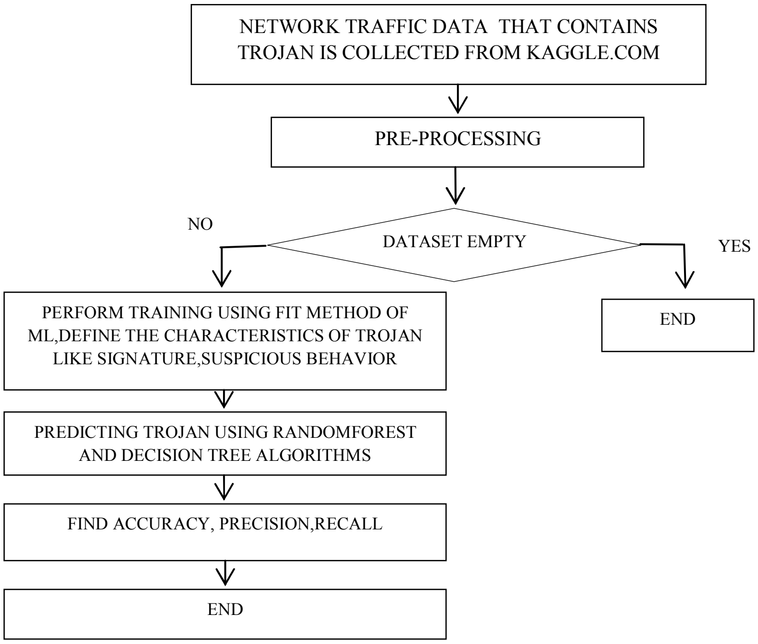

NETWORK TRAFFIC DATA THAT CONTAINS TROJAN IS COLLECTED FROM KAGGLE.COM
PRE-PROCESSING
DATASET EMPTY
NO
YES
END
PERFORM TRAINING USING FIT METHOD OF ML,DEFINE THE CHARACTERISTICS OF TROJAN LIKE SIGNATURE,SUSPICIOUS BEHAVIOR
PREDICTING TROJAN USING RANDOMFOREST AND DECISION TREE ALGORITHMS
FIND ACCURACY, PRECISION,RECALL
END

Fig 1: Flowchart

The dataset "Trojan detection.csv," sourced from Kaggle.com, contains network packets categorized as malicious trojans, offering insights into trojan detection through network traffic analysis. Pre-processing is essential to transform the raw data into a readable format by eliminating null rows/columns. Subsequently, the filtered data, encompassing various data types beyond integers, doubles, or floats, undergoes conversion to integers for further analysis.

Defining the characteristics and behaviors of malware trojans, including their signatures and typical behaviors, enables the implementation of Signature-based Detection. This method entails comparing software code with a database of known malware signatures. Detection relies on identifying matches, indicating the presence of trojans or other malware. However, this approach's effectiveness is constrained by its reliance on known threats, rendering it less effective against novel or zero-day attacks.

For predicting trojan malware, selecting suitable machine learning algorithms is crucial. Commonly employed algorithms include Random Forest, Decision Tree, and variants like Random Forest Plus. Among these, the Random Forest algorithm stands out for its effectiveness in identifying malware. Random Forest operates akin to assembling a diverse team of advisors, each offering a unique perspective. By aggregating their insights, more reliable decisions are made, making it an optimal choice for tasks requiring high accuracy, such as malware detection.

DOI: 10.48175/IJARSCT-18083

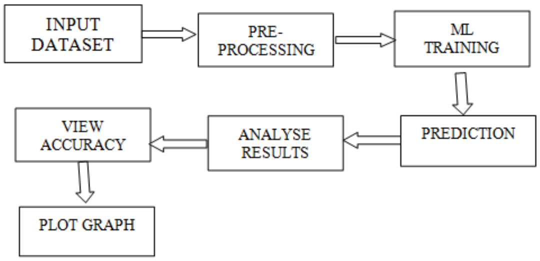
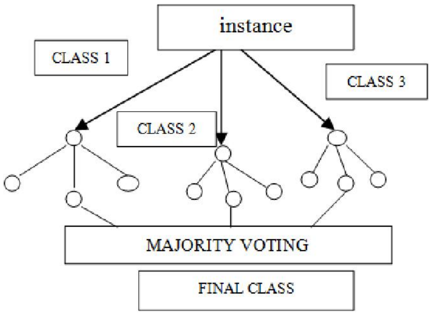


Fig 2: Data flow Diagram

3.1 RANDOM FOREST SIMPLIFIED

Random Forest is a powerful and versatile algorithm that leverages the collective strength of multiple Decision Trees. It is like gathering a team of diverse advisors, each with a unique perspective. By considering their opinions collectively, you make more reliable decisions, making it a great choice for tasks like finding malware where accuracy is crucial. Random Forest is a powerful and versatile algorithm that leverages the collective strength of multiple Decision Trees. It is like gathering a team of diverse advisors, each with a unique perspective. By considering their opinions collectively, you make more reliable decisions, making it a great choice for tasks like finding malware where accuracy is crucial.



IV. MODULES

In proposed system there are mainly three modules:

- Pre-Processing
- Encoding and Training
- Prediction

Pseudocode:

Procedure for Pre-Processing

```
Input: Dataset
Output: Filtered Dataset
Begin
Step 1: Read Network traffic data because trojan can be better understood and realized by monitoring the network and capturing the network packet.
Step 2: Trojan detection.csv file is read using pandas library.
Step 3: Pre-processing in ML is a data mining technique that tranforms raw data into readable format by removing null values.
Step 4: Return filtered Dataset.
Step 5: End
```

DOI: 10.48175/IJARSCT-18083

Procedure for Encoding And training

```
Input: Pre-Processed Dataset
Output: Encoded Dataset
Begin
Step 1:Define the characteristics and behaviors of the malware trojan
Define trojan_signature = "unique_string_or_pattern"
Define suspicious_behavior = "malicious_activity_pattern"
Step 2: Convert text features,float pre processed datasets to integers
Function encode_malware(input_file):
Read input_file
Apply encoding algorithm (e.g., XOR, encryption) to detect malware code
Step 3: Combine encoded features
Step 4: End
```

Procedure for Detection

Load trained models:

a. Decision Tree b. Random Forest

Predict if the software is malware using Decision Tree and Random forest. Change parameters like the number of trees.

Record the performance (e.g., accuracy) of each model variation.

Plot a graph with the parameter values on the x-axis and the corresponding performance metric on the y-axis.

Compare the performance of decision tree and random forest models at different parameter values.

```
    dt_predictions = decision_tree_model.predict(X_test)
    rf_predictions = random_forest_model.predict(X_test)
```

By following these steps, you can build, evaluate, and compare decision tree and random forest models for predicting malware trojans and visualize their efficiency through a graph.

V. OBJECTIVES OF THE PROPOSED WORK

Early Prediction: By implementing ML algorithms at the firewall level, the system can predict the presence of malware before it infiltrates the network or compromises computing devices. This early detection enables proactive measures to be taken to mitigate potential threats, preventing or minimizing damage to the system.

- **Reduction of Malware Attacks**: By accurately predicting malware presence, the system can preemptively block or quarantine malicious entities before they can execute harmful activities. This proactive approach helps reduce the frequency and severity of malware attacks, safeguarding the integrity and security of the network and computing devices.
- **Mitigation of Losses**: Malware attacks often result in significant losses, including data breaches, financial losses, and damage to reputation. By detecting malware before it can cause harm, the project aims to mitigate these losses by preventing unauthorized access, data theft, system downtime, and other adverse consequences associated with malware infections.
- **Enhanced Security**: Implementing ML-based malware detection enhances the overall security posture of the system by providing real-time threat intelligence and adaptive defenses. By continuously analyzing network traffic and identifying patterns indicative of malicious behavior, the system can adapt and evolve to counter emerging threats effectively.
- **Resource Optimization**: Early detection of malware at the firewall level helps optimize resource utilization by preventing unnecessary processing of malicious content and conserving computing resources. By filtering out potential threats before they reach internal systems, the project reduces the burden on endpoint devices and network infrastructure, improving overall system performance and efficiency.

VI. CONCLUSION

This project introduces a novel approach to malware detection by analyzing power usage and network traffic data. It demonstrates the effectiveness of Random Forest, particularly when considering power sources such as +12V CPU rails. Interestingly, solely examining power usage surpasses relying solely on network traffic for detection.

However, combining both yields the best results. The study emphasizes feature selection to enhance efficiency. Moreover, it provides practical guidance on efficiently selecting crucial features.

Additionally, the implementation of decision tree and random forest algorithms shows promise, with decision trees offering transparency and interpretability, while random forests provide enhanced accuracy and robustness through ensemble learning.

Overall, these techniques bolster software security against malicious threats by leveraging diverse data sources and selecting key features for precise detection.

REFERENCES

[1] Mohd Faizal, Izham Jaya, Zahian Ismail, Ahmad Firdaus "Trojan Detection System Using Machine Learning Approach", August 2022,Indonesian Journal of Information Systems.

[2] Yu-Feng Liu, Li-Wei Zhang, Jain Liang, Sheng Qu, Zhi-Qiang Ni "Detecting Trojan Horses Based On System Behaviour Using Machine Learning Method" , July 2021 , Proceedings of the Ninth International Conference on Machine Learning and Cybernetics.

[3] Ma Zhongrui, Huang Yuanyuan, Lu Jiazhong , "Trojan Traffic Detection Based On Machine Learning", 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing .

[4]Mohammed Altaiy, Incilay Yildiz,Bahadir Ucan ,"Trojan Traffic Detection Based On Machine Learning", Aurum Journal Of Engineering Systems And Architecture Volume 7, No 1, 2023 .

[5] Yamjala Supriya, Dammu Sowjanya, Gautam Kumar, Deepali Yadav, Devarakonda lakshmi kameshwari, Malware Detection Techniques: A Survey", sixth international Conference on parallel, distributed and grid computing, 2020.

[6] "A Survey on Malware Detection Using Machine Learning Techniques: Classification, Features and Datasets" by Raja Vikramaditya Singh Chandel and B. B. Gupta (2019).

[7]"Machine Learning Techniques for Android Malware Detection: A Survey" by Arati M. Dixit and Atul S. Auti (2018).

[8]"Machine Learning Based Android MalwareDetection: A Survey" by Z. Jin, W. Wei, and K. W. Hamlen (2018).

[9]"Android Malware Detection Using Machine Learning Techniques" by R. Kharkar, M. Goswami, and M. Bahirat (2017) .

[10]"Android Malware Detection Using Machine Learning Techniques" by S. V. Selvi and G. Hemalatha (2018).

[11]Thimbleby H, Anderson S, Cairns P. A Frameworkfor Modelling Trojans and Computer Virus Infection[M]. Computer Journal, 1998, 41(7): 444-458,1998 .

[12]Wang, R., Wang, W., Gong, X., Que, X., & Ma, J. (2010, April). A Real-Time Video Stream Key FrameIdentification Algorithm for QoS[M]. In Multimedia and Information Technology (MMIT), 2010 Second International Conference on (Vol. 1, pp. 115-118). IEEE, 2010 .

[13]Christodorescu M, Jha S, Seshia S A, et al. Semanticsaware malware detection[M]. In: Security and Privacy, Oakland:IEEE, 2005. 32-46,2005 .

[14] Wu Xianda. Remote control Trojan detection model based on abnormal network behavior [D]. Beijing University of Technology, 2018 .

[15]Yang Weijun, Zhang Shu, Hu Guangjun. Trojan detection method based on attack tree model[J]. Information Network Security, 2011(09): 170-172.

[16]Li Jianbin. Trojan detection technology based on traffic [D]. University of Electronic Science and Technology of China, 2014 .

[17] Wang Zhanhao. Research on Trojan Attackand Prevention Technology [J]. Shanghai Jiaotong University, 2007 .

DOI: 10.48175/IJARSCT-18083