**Enhanced Network Intrusion Detection with Threat Intelligence Integration and Hybrid Models**

**Team members.**

Abett Reddy cheruku

Karthikeya Nagisetty

Hemanjan Reddy Pundla

Venkata Karthik Reddy Banka

**Abstract**

With exponential growth in the size of computer networks and developed applications, the significant increasing of the potential damage that can be caused by launching attacks is becoming obvious. Meanwhile, Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) are one of the most important defense tools against the sophisticated and ever-growing network attacks. Due to the lack of adequate dataset, anomaly-based approaches in intrusion detection systems are suffering from accurate deployment, analysis and evaluation. There exist a number of such datasets such as DARPA98, KDD99, ISC2012, and ADFA13 that have been used by the researchers to evaluate the performance of their proposed intrusion detection and intrusion prevention approaches. Based on our study over eleven available datasets since 1998, many such datasets are out of date and unreliable to use. Some of these datasets suffer from lack of traffic diversity and volumes, some of them do not cover the variety of attacks, while others anonymized packet information and payload which cannot reflect the current trends, or they lack feature set and metadata. This paper produces a reliable dataset that contains benign and seven common attack network flows, which meets real world criteria and is publicly avaliable. Consequently, the paper evaluates the performance of a comprehensive set of network traffic features and machine learning algorithms to indicate the best set of features for detecting the certain attack categories.

**Introduction**

defense process by aiming security administrators in forewarning them about malicious behaviors such as intrusions, attacks, and malware. Having IDS is a mandatory line of defense for protecting critical networks against these ever-increasing issues of intrusive activities. So, research on IDS domain has flourished over the years to propose the better IDS systems. However, many researchers struggle to find comprehensive and valid datasets to test and evaluate their proposed techniques (Koch et al., 2017) and having a suitable dataset is a significant challenge itself (Nehinbe, 2011).

**Goals and Objectives**

Motivation: The outdated network intrusion detection systems (NIDS) need to be updated according to the growing professionalism of cyber threats. The necessity for an improved NIDS that can adjust to new threats is addressed by this project.

Significance: The project holds great importance as it has the ability to greatly enhance network security and resilience against a wide variety of cyber threats.

Objectives: To develop a hybrid intrusion detection system integrating threat intelligence feeds, thereby enhancing the detection accuracy and response to both known and emerging threats.

Features: The system features real-time threat intelligence integration, a combination of signature-based and anomaly-based detection methods, and a dynamic update mechanism to adapt to new threats.

**Problem Statement**

This project's main goal is to improve network intrusion detection systems' accuracy and efficiency. The following goals are explored by the project's structure: Is it possible for hybrid models to enhance real-time threat recognition, both known and unknown? In what ways can network intrusion detection be enhanced by the effective integration of threat intelligence feeds? In what ways do these technological advancements affect network security in practise, and how can organisations adjust to strengthen their security postures in response to these changes?

**Dataset and Attributes**

A comprehensive dataset is critical for training and testing the proposed hybrid models for intrusion detection. The dataset was created by amalgamating data from diverse sources, with an ample quantity to facilitate effective training and testing. Key attributes include:

Duration: Length of the network connection.

Protocol Type: The network protocol in use, such as TCP or UDP.

Service: The network service related to the connection, like HTTP or FTP.

Flag: The status of the connection, such as open or closed.

Count: The number of connections to the same host within a specific timeframe.

Srv\_Count: The number of connections to the same service within a specified timeframe.

Label: A categorical attribute indicating whether the connection is normal or an intrusion.

These attributes are pivotal for the detection models to discern between normal activities and potential threats.

Here is the list of columns:  
A screenshot of a computer

Description automatically generated

**Methodology**

The project's methodology is divided into multiple phases, starting with data collecting and preprocessing to guarantee privacy and quality of the data. The data must be encoded, standardized , and normalized in order for the machine learning algorithms to use it at the preprocessing stage.

Through the use of exploratory data analysis (EDA), it is possible to identify underlying patterns and potential indicators of network breaches, leading to a deeper understanding of the data. Feature selection is used to choose the most important features for the detection models. Next, a range of machine learning models are examined, encompassing both traditional signature-based models and contemporary anomaly-based models. A hybrid technique is produced by combining the best aspects of both model types.

Model training involves dividing the dataset into training and testing sets, followed by the optimization of hyperparameters to enhance performance. Model evaluation is performed using metrics such as accuracy, precision, recall, and F1-score, supplemented by cross-validation techniques to ascertain the models' robustness.

**Implementation**

The practical application of the proposed system was executed in a systematic manner, ensuring that the theoretical models were accurately represented in a real-world environment. The development environment was set up using Python as the primary programming language due to its rich ecosystem of data science libraries. Key libraries such as NumPy for numerical computations, pandas for data manipulation, matplotlib and seaborn for visualization, and scikit-learn for machine learning were utilized.

Implementing data pipelines for feature extraction, preprocessing, and integration with real-time threat intelligence feeds were all part of the coding process. Modularity was given priority during the coding of the models to provide simple upgrades and maintenance. The core of the implementation was the hybrid model, which merged anomaly detection with signature-based identification through the use of techniques including decision trees, support vector machines, and neural networks.

Addressing challenges such as overfitting required meticulous tuning of the models and adopting techniques like k-fold cross-validation. Computational constraints were managed by optimizing algorithms and selecting efficient data structures. Data imbalance, a common issue in intrusion detection datasets, was mitigated using sampling techniques and cost-sensitive learning.

The deployment of the intrusion detection system involved integrating the machine learning models with a simulation of network traffic. Continuous monitoring was set up to evaluate the system’s performance, with a focus on the system's responsiveness to real-time data and its ability to adapt to evolving threats based on the threat intelligence inputs.

The system was also equipped with an interface for security analysts to review alerts and update the threat database, ensuring that the system remained dynamic and responsive to new threats. This hands-on approach to threat detection and system evolution paves the way for future enhancements and continuous improvement of network security measures.

**Threat Intelligence Integration**

A central aspect of the project was the incorporation of real-time threat intelligence to enhance the predictive power of the intrusion detection system. Threat intelligence feeds from AlienVault OTX and other reputable sources were integrated, providing up-to-date information on the latest threat landscape.

**The integration process involved:**

Establishing secure and reliable channels for receiving threat intelligence data.

Automating the ingestion and parsing of data feeds to extract relevant information.

Mapping the threat data to the network traffic features to enable proactive detection.

Developing a system for continuously updating the model's knowledge base with new threat intelligence.

This integration allows the intrusion detection system to not only rely on historical data but also to utilize current global cyber threat information to identify and respond to new and sophisticated attacks promptly. The system was designed to dynamically update its detection patterns and indicators of compromise (IOCs) based on this intelligence, thus staying ahead of attackers' evolving tactics.

Furthermore, the system leverages this intelligence to perform heuristic analysis and context-aware detection, considering factors such as the source and reputation of network traffic, which significantly improves its ability to detect advanced persistent threats (APTs) and zero-day attacks. The collaboration between the hybrid detection models and threat intelligence feeds culminates in a more resilient and adaptive network intrusion detection system, leading to a marked improvement in security efficacy.

Attacker1: 192.168.1.2

Attacker2: 192.168.1.3

Attacker3: 192.168.1.4

Victim: 192.168.1.5

Routing:

Configure static routes on routers if needed to ensure proper communication between devices.

Attacker PCs:

Configure attacker PCs to simulate malicious traffic. Use tools such as Metasploit, Wireshark, or other penetration testing tools. Simulate attacks like Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS.

Victim PC:

Configure the victim PC with security measures like firewalls, antivirus, and intrusion detection/prevention systems.

Monitoring:

Use Wireshark or Packet Tracer's built-in simulation mode to monitor network traffic. Analyze packets to detect malicious activity.

Security Measures:

Implement security measures such as ACLs (Access Control Lists), firewalls, and intrusion prevention systems to mitigate the simulated attacks.

Building a network simulation for a scenario like this involves creating a topology in Cisco Packet Tracer and configuring devices to simulate the described attack. Below is a step-by-step guide on how you can set up such a network:

Topology Design:

Create a network topology in Cisco Packet Tracer. Place routers, switches, and PCs on the workspace. Design the network to represent a simple scenario where attackers are trying to send malicious traffic to a victim.

Example Topology:

Attacker1 ----\

Attacker2 ----- Switch ----- Victim

Attacker3 ----/

Device Configuration:

Configure IP addresses for each device using the CLI or GUI of each device.

For example (assuming Class C IP addresses):

Attacker1: 192.168.1.2

Attacker2: 192.168.1.3

Attacker3: 192.168.1.4

Victim: 192.168.1.5

Routing:

Configure static routes on routers if needed to ensure proper communication between devices.

Attacker PCs:

Configure attacker PCs to simulate malicious traffic. Use tools such as Metasploit, Wireshark, or other penetration testing tools. Simulate attacks like Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS.

Victim PC:

Configure the victim PC with security measures like firewalls, antivirus, and intrusion detection/prevention systems.

Monitoring:

Use Wireshark or Packet Tracer's built-in simulation mode to monitor network traffic. Analyze packets to detect malicious activity.

**Deployment and Recommendations**

The deployment phase marked the transition of the project from a controlled development setting to a live network environment. The intrusion detection system was deployed on a test network to monitor traffic and identify potential threats. This live testing phase allowed the team to fine-tune the system in a real-world context and make necessary adjustments.

Here is the packet routing structure:

A computer screen shot of a diagram

Description automatically generated

**Recommendations for organizations looking to adopt this system include:**

Ensuring network infrastructure is prepared for integration with the new system.

Training IT personnel to understand the functionality and output of the system.

Establishing protocols for responding to detected threats.

Regularly updating the threat intelligence database to maintain system efficacy.

Additionally, it is recommended that organizations consider the broader implications of integrating such a system, including potential privacy concerns and the need for a robust cybersecurity policy that addresses the ethical use of threat intelligence and intrusion detection technologies.

**EDA :**

Below are some of the EDA screenshots.

Description of the dataset:

A screenshot of a computer

Description automatically generated

**Snippet for generating plot**

**A screen shot of a computer

Description automatically generated**

**A graph of a number of people

Description automatically generated**

**A graph of a number of blue rectangular objects

Description automatically generated with medium confidence**

**A chart with text overlay

Description automatically generated**

**Correlation matrix:**

**A screenshot of a computer

Description automatically generated**

**Anova test:**

**A screenshot of a computer

Description automatically generated**

**Results**

The results of the project underscore the effectiveness of the hybrid intrusion detection system enhanced with threat intelligence. The signature-based models demonstrated high accuracy in detecting known threats, while the anomaly-based models were adept at identifying novel attack patterns. The hybrid models leveraged the strengths of both approaches, resulting in a balanced and versatile detection system.

Quantitative results indicated a significant improvement in the system’s ability to detect a range of intrusions, with precision and recall metrics exceeding benchmark standards. The integration of threat intelligence further refined the detection capabilities, enabling the system to adapt to the ever-changing threat landscape swiftly.

**Discussion**

The discussion of results revealed that the integration of threat intelligence with hybrid detection models offers a promising approach to network intrusion detection. This project's approach addressed the limitations of traditional NIDS by enhancing their adaptability and responsiveness to emerging threats.

The discussion also explores the potential for such systems to evolve using machine learning techniques, highlighting the importance of continuous learning and adaptation in the field of cybersecurity. Future research could focus on the incorporation of more advanced AI techniques, such as deep learning and reinforcement learning, to further enhance detection accuracy and reduce false positives.

**Conclusion**

Having a reliable, publicly available IDS evaluation datasets is one of the fundamental concerns of researchers and producers in this domain. In this paper, we have monitored the state-of-the-art in the IDS dataset generation and evaluation by analyzing the eleven publicly available datasets since 1998 which are limited because of the lack of the traffic diversity and volumes, anonymized packet information and payload, constraints on the variety of attacks, and lack of the feature set and metadata. Then we generate a new IDS dataset includes seven common updated family of attacks that met real worlds criteria and is publicly available (http://www.unb.ca/cic/datasets/IDS2017.html). On the evaluate section, we fist extract the 80 traffic features from the dataset and clarify the best short feature set to detect each attack family using RandomForestRegressor algorithm. Afterwards, we examine the performance and accuracy of the selected features with seven common machine learning algorithms. Finally, we compare the quality of the generated dataset by searching for common mistakes and criticisms of other synthetically created datasets, based on the 11 criteria of the last proposed dataset evaluation framework with other publicly available datastes since 1998till 2016. In the future, we sould like to increase number of PCs as well as conducting more up to date attacks.

**Work completed:-**

* We are done with data processing ,EDA, Threat intelligence integration.
* Visualizations are done.
* We are done with data collection and preprocessing with the help of Abett
* We are done with model selection and training with the help of hemanjan.
* We are done with threat intellience with the help of karthikeya
* We are done with EDA with the help of karthik

**Work to be done:**

* Still enhancing EDA and models
* We will be working on improving efficiency
* We will be trying to bringing down the time and space complexity of code actively.
* We will be working on enhancing models in second icrement
* We will be also working on enhancing EDA.

**References**

Cui, Jin; Wang, Zhijian; Wu, Yulei. "Hybrid Intrusion Detection with Weighted Signature-Based and Anomaly-Based Intrusion Detection Mechanisms." IEEE Transactions on Information Forensics and Security, 2017.

**Github Link**: https://github.com/VenkatakarthikReddyB/Final\_project.git