Detection of Threats using Machine Learning

Harshika Srivastava
016019120
Computer Engineering
San Jose State University

Teja Sree Goli
016040986
Computer Engineering
San Jose State University

Rutik Sanjay Sangle
016007589
Computer Engineering
San Jose State University

Abstract

With the increasing number of systems connecting to the internet, safeguarding networks against various cyber threats has become more critical than ever. The cybersecurity landscape is evolving rapidly, prompting companies to invest millions in advanced technologies to protect their businesses and fortify defenses against potential attacks. To address these challenges, leveraging Artificial Intelligence (AI) and Machine Learning (ML) technology is crucial for automating the detection and response to cyber threats. This project aims to develop a machine-learning model for a network intrusion detection system. The primary goal is to provide a binary classification of results by identifying whether the input network log is an attack or benign. Organizations can benefit from these technologies by automating the detection of outlier patterns in networks and flagging systems that do not comply with organizational standards. This proactive approach enhances the overall cybersecurity posture and helps prevent potential threats before they can cause harm.

Keywords*: *Cyber Security, Machine Learning, Network Intrusion Detection, Attack Classification, Outlier Patterns

I. INTRODUCTION

It is more important than ever to keep our computer systems safe from attackers in the connected world of today. The technologies for protecting organizations and individuals against cyberattacks are evolving rapidly. With the advances in the field of Artificial Intelligence and machine learning, there are already several examples of implementations available widely across the internet. However, companies are still working towards leveraging these technologies in the right way to their advantage. Utilizing AI and ML for cybersecurity requires an understanding of the many forms of cyberattacks and their implications. A few of the most common attacks that organizations encounter are malware attacks that compromise sensitive data and disrupt business operations, phishing attacks leading to unauthorized access to systems, identity theft, Denial-of-service (DOS) attacks resulting in service disruptions and financial losses, SQL-Injection attacks leading to unauthorized data access and creation of additional vulnerabilities entries.

Considering the constantly shifting landscape of cyber threats, cybersecurity solutions must incorporate AI and ML. By learning from historical data and identifying developing patterns, these technologies allow a shift in threat detection from reactive to proactive. Anomaly detection and pattern recognition enable early detection of anomalous activity, while flexible security mechanisms provide resistance against dynamic cyber threats. The Machine Learning algorithm's ability to identify patterns and flag anomalies enables its application to email security and secure user authentications in various web applications. Machine Learning and Artificial Intelligence technologies provide advanced techniques for monitoring networks, identifying anomalies, and responsive measures to mitigate potential cyber-attacks by offering network security. The technology's ability to take large amounts of input aids in real-time implementations.

This paper focuses on various web-based attacks on networks and the development of a machine-learning model to classify these attacks as malignant or

benign. The web attacks considered fall into 3 different categories:

1. **Cross-site scripting (Brute Force-XSS):** These attacks occur when data enters a web application through an untrusted source, most frequently a web request or the data is included in dynamic content that is sent to a web user without being validated for malicious content [5].

2. **SQL-Injection (SQL-Injection):** This consists of the insertion of a SQL query via the input data from the client to the application. This attack allows attackers to spoof identity, tamper with existing data, cause repudiation issues, or make data unavailable to authorized users [5].

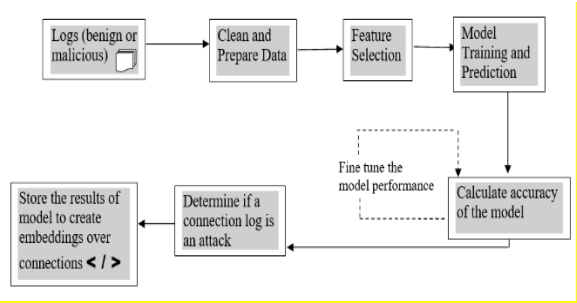
3. **Brute force attempts on passwords (Brute Force-Web):** These attacks are often used for attacking authentication and discovering hidden content/pages within a web application [5].

II. MACHINE LEARNING IN CYBERSECURITY

Artificial intelligence (AI) solutions can recognize shadow data, keep monitoring for anomalies in data access, and notify an organization's cybersecurity team of potential threats from anyone gaining access to sensitive data. This can save a significant amount of time by quickly identifying and resolving issues as they arise. Machine learning (ML) is a popular technology used to provide data to the Intrusion Detection System (IDS) in order to detect malicious network traffic. The detection performance of ML models is fundamentally dependent on the quality of the dataset used to train the model. This project presents a detection framework for IDS to detect network traffic irregularities using a machine learning model.

Architecture

The diagram below shows the detailed architecture of the proposed machine learning project for an intrusion detection system. Each block represents each step taken during the implementation.



Logs: The process initiates with the collection of logs. The dataset utilized for this project encompasses two types of network traffic: benign and malicious. The original dataset is sourced from publicly available data, comprising more than 100k rows and featuring over 80 available attributes. It encompasses diverse attack scenarios, including brute force, Botnet, DDoS, Web attacks, and network infiltration [3].

Cleaning and Preparing Data: After collecting the initial log data, the next step is to clean and prepare it. This means going through the data to get rid of any irrelevant features, dealing with missing or incomplete data, and changing the log entries into a format that can be easily used for machine learning.

Feature Selection: After cleaning the data, an analysis of features is conducted to determine their suitability for integration into the machine learning model. Features represent individual measurable properties or characteristics of the observed phenomena. The selection of appropriate features is crucial for effectively training the model.

Model Training and Prediction: Utilizing the chosen features, the model undergoes training using a dataset with known outcomes. Subsequently, the trained model is capable of making predictions or decisions concerning new, unseen data, thereby assessing whether a new log entry is benign or malicious. The selection of appropriate features is crucial for the effective training of the model.

Fine-tune Model Performance: Following the model training, the performance of the model is fine-tuned. This process may include adjusting the model's parameters, incorporating additional training data, or applying techniques such as cross-validation to enhance its accuracy.

Calculate Model Accuracy: The model's accuracy is determined to assess its performance. This

generally involves comparing the model's predictions to a dataset with known outcomes and calculating the percentage of accurate predictions.

Determine if a connection log is an attack: As new log data comes in, the model determines if a connection log is an attack based on what it has learned.

Store the results of the model to create embeddings over connections: The model's predictions, indicating whether a log is benign or malicious, are stored. These results can be utilized to create embeddings, which are vector representations of the logs. This process helps in visualizing and comprehending the connections within the data.

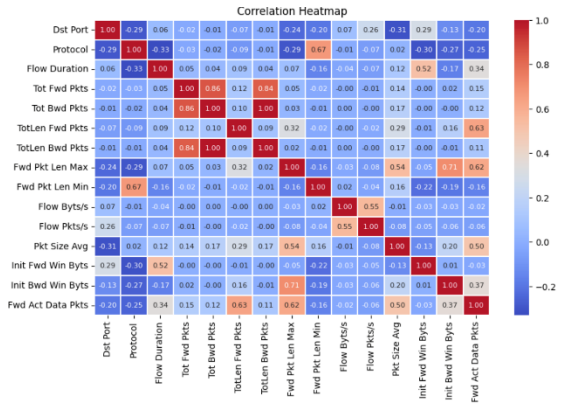
Data Preparation: The performance of any machine learning model is highly dependent on the training data and hence, it is crucial to make sure the data we have is cleaned. Checking for consistent data types and null values in the columns are the data cleaning steps to be performed to ensure proper training data. For large datasets, a Python script is used to clean the data which identifies and eliminates rows based on the defined conditions.

Feature Engineering

We analyzed a dataset representing various network traffic types. The key features examined include:

- **Flow Duration**: Indicative of the time span of network flow.
- **Total Forward and Backward Packets:** Reflects the count of data packets sent in both directions.
- **Total Length of Forward and Backward Packets:** Provides insights into the data volume.
- **TCP Flag Counts:** Includes flags like PSH, SYN, ACK, which are crucial for understanding TCP connection states.

A set of 15 important numerical features is considered to visualize the data using a correlation matrix. The resultant matrix is as shown below:



A few observations from the correlation heatmap of the features are as below:

1. The most correlated features are the count of forward packets and backward packets. This correlation arises from the common characteristic of legitimate network traffic, which typically exhibits an equal number of forward and backward packets. Conversely, malicious traffic often displays an imbalance between the number of forward and backward packets.
2. Another closely correlated feature is the flow duration. Legitimate network flows usually have a short duration, whereas malicious flows tend to be more prolonged.
3. The total length of forward packets and backward packets exhibits a high correlation, given that the combined length of packets is generally proportional to the flow duration.
4. There is a negative correlation between the flow bytes per second feature and the packet size average feature. This is attributed to the fact that larger packets typically result in a lower flow bytes per second rate compared to smaller packets.

Modeling:

Models used in general

There are a wide variety of machine learning algorithms available to implement for network intrusion detection systems based on the problem statements. A few of the ML models that can be used are Decision Trees, Neural Networks, Support Vector Machines, K-means clustering, and others. The following section provides a brief description of the machine learning approach followed for this project.

Model description -

- **Model Type:**
  Sequential - This type of model is a linear stack of layers where each layer has exactly one input tensor and one output tensor.
- **Layers:**
  Dense (128 units): The first layer is a fully connected (dense) layer with 128 neurons. It takes an input of unspecified size (as denoted by None), which should match the number of features in your dataset. The layer has a weight parameter for each connection, resulting in 10,240 parameters (this suggests that the input size is 80, since 80 * 128 + 128 (bias terms) = 10240).

  Dense (64 units): The second layer is also a dense layer, now with 64 neurons. It takes inputs from the previous layer's 128 neurons, leading to 8,256 parameters (computed as 128 * 64 + 64 = 8256).

  Dense (1 unit): The final layer is a dense layer with a single neuron, which outputs the model's prediction. This could be a binary classification (attack vs. benign), and thus it has 65 parameters (64 * 1 + 1 = 65).
- **Output Shape:** The output shape of each layer indicates the dimensionality of the output space.
- **Parameters:**

1. Total parameters: 18,561 - The total number of trainable parameters in the model.
2. Trainable parameters: All 18,561 parameters are trainable, meaning they will be updated during training.
3. Non-trainable params: There are no non-trainable parameters in this model.

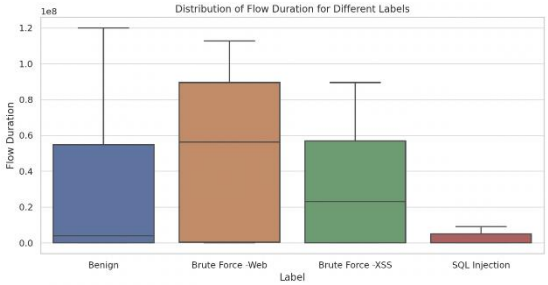
Implemented demo description

A pre-trained machine learning model is used to train the model on our dataset. The pre-trained model classifies the attacks into 4 different classes while the model in this project detects if there is an attack or not.

Observations:

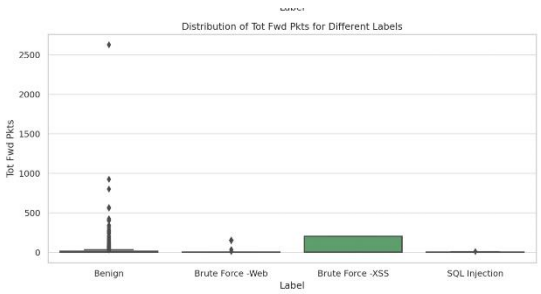
In order to build a machine learning model for prediction, we first analyzed the relevant features and tried to find the patterns in data to build a classifier. Our analysis revealed distinctive patterns for different attack types, particularly in terms of flow duration and packet counts. For instance, SQL Injection attacks typically exhibited shorter flow durations, while brute force-XSS attacks involved higher numbers of forward packets.
We have checked the distribution of "Flow Duration" across four categories of network traffic: Benign, Brute Force-Web, Brute Force-XSS, and SQL Injection. Flow Duration is likely to be measured in units of time, such as milliseconds, and represents how long a given network flow lasts.

Our observations from boxplot:



- Benign Traffic: Exhibited moderate median flow duration with a broad range, suggesting variability but within a specific range. This indicates a general consistency in benign network behavior, albeit with some natural variation.
- Brute Force-Web: Characterized by higher median flow durations and a wide range, reflecting the persistent and prolonged nature of these attacks. This could be attributed to the repeated attempts and extended interactions involved in brute force attacks.
- Brute Force-XSS: Similar to Brute Force-Web in terms of median flow duration, but with a slightly narrower range, indicating more consistent attack durations. This consistency might be a characteristic feature of brute force-XSS attacks.
- SQL Injection: Markedly different with a much shorter median flow duration and a tight interquartile range, suggesting quick, brief interactions typical of SQL Injection attacks.
These observations suggest that flow duration is a promising feature for differentiating SQL Injection from other types of traffic. However, the overlap in flow durations between benign traffic and Brute Force attacks indicates the need for additional features for accurate classification.

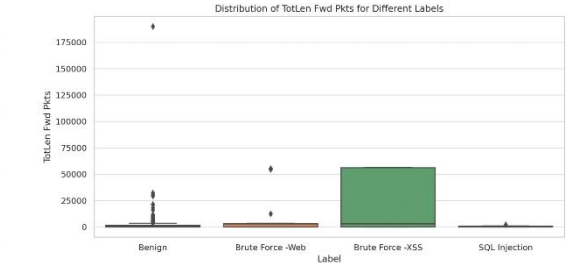
Total Forward Packets Analysis



- Benign Traffic: Showed a concentrated distribution with a low median and small interquartile range, with outliers suggesting occasional higher numbers of forward packets in benign flows.
- Brute Force-Web: Exhibited an extremely tight distribution, indicating a very consistent number of forward packets across events.
- Brute Force-XSS: Demonstrated a higher median and larger interquartile range compared to benign and Brute Force-Web attacks, suggesting higher variability and quantity in forward packets.
- SQL Injection: Similar to benign traffic in terms of the interquartile range but with outliers, indicating occasional variations.

These patterns reveal that the 'Total Forward Packets' feature could be crucial in distinguishing Brute Force-XSS attacks due to their higher median and variability.

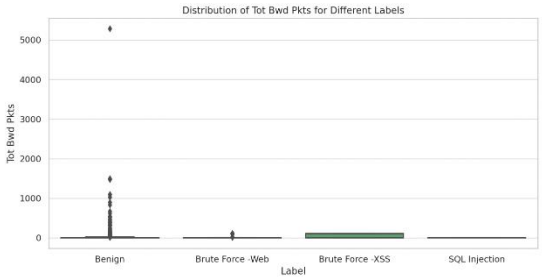
Total Length of Forward Packets Analysis



- Benign Traffic: Presented a tight distribution with a small interquartile range, indicating consistency in the total length of forward packets.
- Brute Force-Web: Almost negligible variation, suggesting very little difference in forward packet lengths across these attacks.
- Brute Force-XSS: Showed a larger median and range, indicating more data being sent in forward packets compared to other categories.
- SQL Injection: Featured the smallest interquartile range with a low median, suggesting uniformity in forward packet lengths.

This feature is significant for classifying traffic, especially in distinguishing brute force-XSS and SQL Injection attacks due to their distinct packet length behaviors.

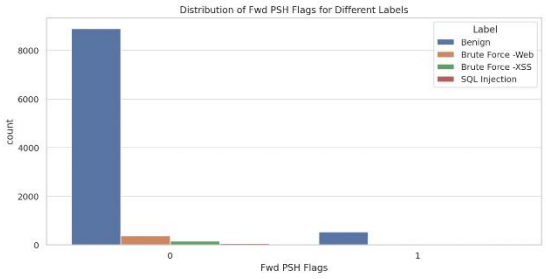
Total Backward Packets Analysis

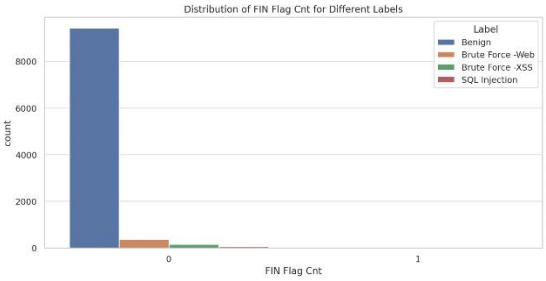


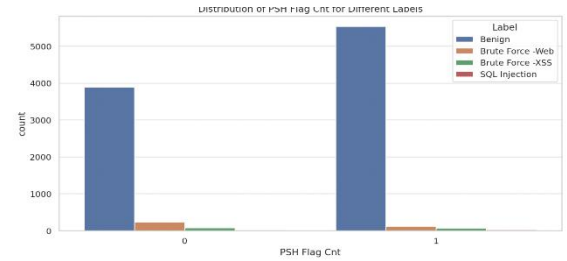
- Benign Traffic: Had a distribution concentrated near the lower end, with a small interquartile range but some extreme outliers.
- Brute Force-Web and Brute Force-XSS: Both showed low and consistent numbers of backward packets.
- SQL Injection: Displayed a similar pattern to Brute Force-Web and XSS, with very few outliers.

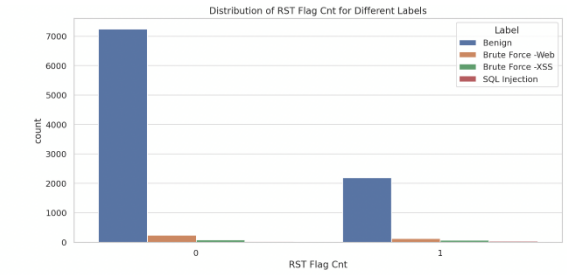
The 'Total Backward Packets' feature, while not highly distinctive on its own, contributes to a classification model when combined with other features, especially in identifying attack types with consistently low backward packets.

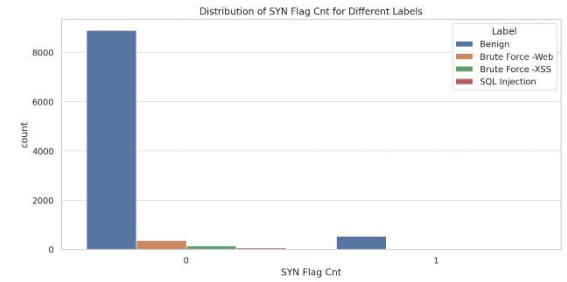
TCP Flag Count Analysis











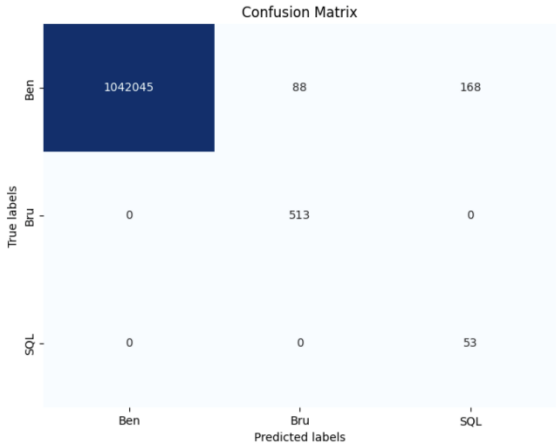
The distribution of the "PSH Flag Count" and other TCP flags like `FIN`, `SYN`, `ACK`, etc., across different traffic types revealed:
- Benign traffic often had the PSH flag set, indicating regular data transmission protocols favoring immediate delivery.
- Attack categories like Brute Force-Web, XSS, and SQL Injection frequently did not set the PSH flag.
- Other flags also showed varying patterns that could be useful in classification, such as certain flags being predominantly set in malicious traffic but rarely in benign.

These observations indicate that TCP flag counts, particularly the PSH flag, are useful features in a classifier for differentiating between benign and attack traffic. However, the presence of benign instances where the PSH flag is not set suggests the necessity to combine this feature with others for a robust classification model.

III. CONCLUSION

Results

The classifier showed promising results in differentiating between attack and benign traffic. Features like Flow Duration and Total Forward Packets were particularly effective in identifying specific attack types. The TCP Flag Counts, especially the PSH flag, emerged as a significant indicator of benign traffic. We achieved a model accuracy of 0.99.



Conclusion

The project successfully demonstrated how machine learning can be used to identify vulnerabilities in networks. The classifier's ability to distinguish between benign and malicious signals was noteworthy. Certain attack types could be identified quite well using features like Flow Duration and Total Forward Packets. One important predictor of benign traffic was found to be the TCP Flag Counts, particularly the PSH flag. These results highlight the potential of machine learning to improve intrusion detection systems on networks. In conclusion, it should be noted that the suggested machine learning model showed encouraging results and future work may concentrate on improving the model's performance and adaptability to changing cyber threats.

Future Work

Due to the growing sophistication of cyber threats, network security is becoming increasingly important. Using machine learning to detect network intrusions has shown to be quite successful, providing organizations all over the world with a potential defensive strategy. We have shown how machine learning may be used in real-world applications to improve network intrusion detection systems in our project. In particular, we built our model using a large dataset which brings assessment and

performance factors for system fitting into consideration.

In addition, to mitigate the difficulties associated with practical application, we support the integration of advanced machine learning techniques, particularly in the context of cybersecurity. Combining cloud computing resources with deep learning approaches seems to be a potent way to optimize our model's performance when it is implemented in operational networks. This integration highlights the critical role that machine learning plays in strengthening cybersecurity measures while also minimizing worries about dataset-driven performance difficulties. This method simplifies existing network security strategies and provides a strong basis for emerging research projects that aim to increase cybersecurity resilience against evolving attacks.

References:

1. Apply machine learning techniques to detect malicious network traffic in cloud computing | Journal of Big Data | Full Text (springeropen.com)
2. Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective | IEEE Conference Publication | IEEE Xplore
3. IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB
4. https://isyou.info/jisis/vol9/no4/jisis-2019-vol9-no4-01.pdf
5. https://owasp.org/www-community/attacks