Network Anomaly Detection

Problem Statement

In the realm of cybersecurity, network anomaly detection is a critical task that involves identifying unusual patterns or behaviors that deviate from the norm within network traffic. These anomalies could signify a range of security threats, from compromised devices and malware infections to large-scale cyber-attacks like DDoS (Distributed Denial of Service). The challenge lies in accurately detecting these anomalies in real-time, amidst the vast and continuous streams of network data, which are often noisy and heterogeneous.
The traditional methods of network anomaly detection often rely on predefined rules or signatures based on known attack patterns. However, these methods fall short in detecting new or evolving threats that do not match the existing signatures. Furthermore, as network environments grow in complexity, maintaining and updating these rules becomes increasingly cumbersome and less effective.

Need for Network Anomaly Detection

The need for robust network anomaly detection systems is driven by several key factors:

- Evolving Security Threats: Cyber threats are constantly evolving, with attackers finding new ways to bypass traditional security measures. An effective anomaly detection system must adapt to new threats dynamically.
- Increasing Network Complexity: Modern networks encompass a wide range of devices and applications, many of which are interconnected across multiple platforms. This complexity makes it difficult to establish a baseline of "normal" behavior and identify deviations using traditional methods.
- Operational Continuity: Network anomalies can lead to significant disruptions in business operations and services. Detecting and addressing these anomalies promptly ensures that network services remain reliable and available.
- Regulatory Compliance: Many industries face stringent regulatory requirements for data security and privacy. Anomaly detection helps organizations comply with these regulations by providing tools to detect and mitigate potential security breaches.

Dataset:
https://drive.google.com/file/d/1AlZak8gC27ntWFR0-ZJ0tMxVWFac-XPf/view?usp=drive_link

Data description

Basic Connection Features

1. Duration: Length of time duration of the connection.
2. Protocol_type: Protocol used in the connection.
3. Service: Destination network service used.

4. Flag: Status of the connection (Normal or Error).
5. Src_bytes: Number of data bytes transferred from source to destination in a single connection.
6. Dst_bytes: Number of data bytes transferred from destination to source in a single connection.
7. Land: Indicator if source and destination IP addresses and port numbers are equal (1 if equal, 0 otherwise).
8. Wrong_fragment: Total number of wrong fragments in this connection.
9. Urgent: Number of urgent packets in this connection. Urgent packets are packets with the urgent bit activated.

Content-Related Features

10. Hot: Number of 'hot' indicators in the content, such as entering a system directory, creating programs, and executing programs.
11. Num_failed_logins: Count of failed login attempts.
12. Logged_in: Login status (1 if successfully logged in, 0 otherwise).
13. Num_compromised: Number of 'compromised' conditions.
14. Root_shell: Indicator if root shell is obtained (1 if yes, 0 otherwise).
15. Su_attempted: Indicator if 'su root' command is attempted or used (1 if yes, 0 otherwise).
16. Num_root: Number of 'root' accesses or operations performed as root in the connection.
17. Num_file_creations: Number of file creation operations in the connection.
18. Num_shells: Number of shell prompts.
19. Num_access_files: Number of operations on access control files.
20. Num_outbound_cmds: Number of outbound commands in an ftp session.
21. Is_hot_login: Indicator if the login belongs to the 'hot' list, i.e., root or admin (1 if yes, 0 otherwise).
22. Is_guest_login: Indicator if the login is a 'guest' login (1 if yes, 0 otherwise).

Time-Related Traffic Features

23. Count: Number of connections to the same destination host as the current connection in the past two seconds.
24. Srv_count: Number of connections to the same service as the current connection in the past two seconds.
25. Serror_rate: Percentage of connections that have activated the flag s0, s1, s2, or s3, among the connections aggregated in count.
26. Srv_serror_rate: Percentage of connections that have activated the flag s0, s1, s2, or s3, among the connections aggregated in srv_count.
27. Rerror_rate: Percentage of connections that have activated the flag REJ, among the connections aggregated in count.
28. Srv_rerror_rate: Percentage of connections that have activated the flag REJ, among the connections aggregated in srv_count.
29. Same_srv_rate: Percentage of connections that were to the same service, among the connections aggregated in count.

30. Diff_srv_rate: Percentage of connections that were to different services, among the connections aggregated in count.
31. Srv_diff_host_rate: Percentage of connections that were to different destination machines, among the connections aggregated in srv_count.

Host-Based Traffic Features

32. Dst_host_count: Number of connections having the same destination host IP address.
33. Dst_host_srv_count: Number of connections having the same port number.
34. Dst_host_same_srv_rate: Percentage of connections that were to the same service, among the connections aggregated in dst_host_count.
35. Dst_host_diff_srv_rate: Percentage of connections that were to different services, among the connections aggregated in dst_host_count.
36. Dst_host_same_src_port_rate: Percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count.
37. Dst_host_srv_diff_host_rate: Percentage of connections that were to different destination machines, among the connections aggregated in dst_host_srv_count.
38. Dst_host_serror_rate: Percentage of connections that have activated the flag s0, s1, s2, or s3, among the connections aggregated in dst_host_count.
39. Dst_host_srv_serror_rate: Percentage of connections that have activated the flag s0, s1, s2, or s3, among the connections aggregated in dst_host_srv_count.
40. Dst_host_rerror_rate: Percentage of connections that have activated the flag REJ, among the connections aggregated in dst_host_count.
41. Dst_host_srv_rerror_rate: Percentage of connections that have activated the flag REJ, among the connections aggregated in dst_host_srv_count.

Block 1: Tableau Visualisations

The primary goal of creating a Tableau dashboard for Network Anomaly Detection is to enable cybersecurity professionals and network administrators to visually monitor, analyze, and detect anomalies in network traffic effectively. The dashboard aims to simplify the complexity of network data analytics, providing clear and actionable insights that can be used to enhance network security measures.

Suggestions for Dashboard Components

1. Traffic Volume Over Time:
    - Display line graphs or area charts showing inbound and outbound traffic over time. This helps in identifying spikes or drops that may indicate anomalies.
2. Anomaly Detection Metrics:

    - Use statistical charts to highlight metrics such as number of wrong fragments (Wrong_fragment), urgent packets (Urgent), and failed login attempts (Num_failed_logins), which are indicative of potential security issues.
3. Protocol and Service Analysis:
    - Pie charts or bar graphs showing distributions of Protocol_type and Service used, helping to pinpoint which protocols or services are most associated with anomalies.
4. Connection Status Overview:
    - A heatmap or bar chart for the Flag status indicating normal or error states in connections, providing a quick overview of connection health across the network.
5. Geographical Insights:
    - If IP geolocation data can be integrated, a map showing the geographical distribution of network requests could be useful for identifying anomaly sources by region.
6. Host-based Traffic Features:
    - Visualize Dst_host_count, Dst_host_srv_count, and other host-based features to assess which hosts are most frequently targeted or exhibit unusual behavior.
7. Interactive Filters:
    - Allow users to filter data based on time periods, Protocol_type, Service, and other key metrics to drill down into specific areas of interest.
8. Correlation Analysis:
    - Scatter plots or correlation matrices showing relationships between different numerical features like Src_bytes, Dst_bytes, and Duration to identify patterns that typically represent anomalous activities.
9. Alerts and Anomalies Log:
    - Implement a section where the most recent detected anomalies are listed along with their details for quick review and action.
10. Predictive Insights:
    - If predictive modeling is integrated, provide forecasts of potential future anomalies based on historical trends and current activities.

Block 2: EDA and Hypothesis testing

Exploratory Data Analysis (EDA) and hypothesis testing are essential components of the data science workflow, particularly in the field of network security. EDA helps uncover underlying patterns, relationships, and potential anomalies in the dataset, while hypothesis testing enables validating theories or assumptions statistically. These methodologies are pivotal in understanding network behaviors and enhancing the accuracy of anomaly detection systems.

Suggestions for EDA

1. Distribution of Each Feature:
2. Correlation Analysis:
3. Outlier Detection:
4. Time Series Analysis:
5. Feature Engineering:
6. Missing Values Analysis:

Possible Hypotheses to Test

1. Network Traffic Volume and Anomalies:
    - Hypothesis: Network connections with unusually high or low traffic volume (bytes transferred) are more likely to be anomalous.
    - Tests: Use t-tests or ANOVA to compare the means of Src_bytes and Dst_bytes in normal versus anomalous connections.
2. Impact of Protocol Type on Anomaly Detection:
    - Hypothesis: Certain protocols are more frequently associated with network anomalies.
    - Tests: Chi-square test to determine if the distribution of Protocol_type differs significantly in normal and anomalous connections.
3. Role of Service in Network Security:
    - Hypothesis: Specific services are targets of network anomalies more often than others.
    - Tests: Chi-square test to compare the frequency of services in normal versus anomaly-flagged connections.
4. Connection Status and Anomalies:
    - Hypothesis: Error flags in the Flag feature are significantly associated with anomalies.
    - Tests: Use logistic regression to assess the impact of connection status on the likelihood of an anomaly.
5. Influence of Urgent Packets:
    - Hypothesis: Connections that include urgent packets are more likely to be anomalous.
    - Tests: Logistic regression to evaluate whether the presence of Urgent packets increases the odds of an anomaly.

Block 3: ML Modeling

1. Data Processing

- Data Cleaning: Handle missing values, remove duplicates, and correct errors in the dataset.

- Feature Engineering: Develop new features that could enhance model performance, such as aggregating data over specific time windows or creating interaction terms.
- Data Transformation: Scale and normalize data, especially for algorithms that are sensitive to the scale of input features, like neural networks and distance-based algorithms.
- Train-Test Split: Divide the data into training and testing sets to ensure the model can be evaluated on unseen data.

2. Model Selection

- Supervised Learning Models: on attack column,
    - Classification Models: Logistic regression, decision trees, random forests, support vector machines, and neural networks.
    - Ensemble Techniques: Boosting, bagging, and stacking to improve prediction accuracy and reduce overfitting.
- Unsupervised Learning Models: When labels are not available,
    - Clustering Models: K-means, DBSCAN, or hierarchical clustering to identify unusual patterns or groups.
    - Dimensionality Reduction: PCA or t-SNE for anomaly detection in a reduced dimensional space.

3. Model Evaluation and Validation

- Cross-Validation: Use techniques like k-fold cross-validation to assess model performance across different subsets of the dataset.
- Performance Metrics:
    - For supervised models: Accuracy, Precision, Recall, F1-score, and ROC-AUC.
    - For unsupervised models: Silhouette score, Davies-Bouldin index, or reconstruction error.
- Confusion Matrix: Analyze the true positives, true negatives, false positives, and false negatives to understand the model's performance in different scenarios.
- Adjustment and Tuning: Refine models based on evaluation results, adjusting hyperparameters and experimenting with different model configurations.

Block 4 : Deployment

Purpose of Deployment via Flask API

- Operational Integration: Seamlessly integrate the machine learning model with operational systems for real-time anomaly detection.
- Accessibility: Make the model accessible from various client applications, including web interfaces, network monitoring tools, or even mobile apps.
- Sohowcasing your Project to recruites

Steps for Deploying a Machine Learning Model via Flask API

1. Flask API Setup

- Environment Setup: Create a Python virtual environment and install Flask along with other necessary libraries like numpy, pandas, scikit-learn, or any specific libraries used in the model.
- App Structure: Set up a basic Flask application structure with routes to handle requests for making predictions.

2. Model Integration

- Model Serialization: Serialize the trained model using libraries like pickle or joblib to save the model to a file that can be loaded into the Flask application.
- Load Model: In the Flask app, load the serialized model into memory so it can handle incoming prediction requests efficiently.

3. API Endpoints

- Define Routes: Create a route that accepts POST requests where users can submit their network data in JSON format.
- Data Preprocessing: Ensure the incoming data is appropriately preprocessed to match the format and structure the model expects (e.g., feature scaling, encoding).
- Prediction: Use the model to predict anomalies based on the processed input data and return the prediction results.

Note:

- The suggestions/Ideas provided above are intended to assist you. The primary aim is to offer guidance on what aspects can be analyzed. If no valuable insights can be derived from a particular analysis, feel free to skip it.