Internship Report

UNIFIED MENTOR
YOUR SKILL, SUCCESS & JOURNEY

Name : Prathyusha

Internship : Data Science Intern

Project Title : Cybersecurity: Suspicious Web Threat Interactions

Tools : Jupyter notebook

Domain : Data Analyst

Project Introduction:

This project aimed to analyze web traffic data to build a classification model that predicts whether a network interaction is suspicious or not. The dataset is composed of labeled records describing each web connection using features such as IP addresses, bytes transferred, response codes, protocols, and detection types.

Objective:

- Detect potentially malicious or suspicious behavior in web traffic logs.
- Use machine learning to classify network connections as either normal or suspicious.
- Help security teams prioritize threats based on model predictions.

Project Description:

The dataset simulates real-world traffic logs from a web server environment. Each log includes technical details such as source/destination IPs, the number of bytes exchanged, time of connection, and the associated security rule that flagged the interaction. The primary aim was to find hidden patterns in these attributes and develop models to automate the threat classification process.

Exploratory Data Analysis (EDA)

Key EDA Findings:

1. Missing Value Analysis:
   - Some columns had missing or null values, especially observation_name and rule_names.
   - Imputation strategies were applied (either fill with "Unknown" or remove rows).
2. Target Distribution:
   - The target class (e.g., "suspicious" vs "normal") was imbalanced. Suspicious records were less frequent, which made it important to use metrics like F1-score and recall during evaluation.
3. Bytes Analysis:

  - bytes_in and bytes_out showed right-skewed distributions, indicating occasional high-traffic flows. Log transformation was considered for normalization.
  - Outliers were present, particularly in attacks involving data exfiltration.

4. Country-Based Patterns:
   - Certain country codes (e.g., unusual or unknown locations) showed higher frequency of flagged traffic.
   - Some suspicious IPs repeatedly originated from the same geographic regions.
5. Protocol Analysis:
   - HTTP and HTTPS were the most common protocols.
   - Unusual protocol usage (e.g., uncommon ports) was often associated with suspicious labels.
6. Feature Correlation:
   - A heatmap was generated to study correlations between numerical features.
   - High correlation between certain port numbers and specific rule triggers.
7. Time Series Patterns:
   - Suspicious activities often spiked at certain hours, suggesting automated attack scripts or bot activities.

Sample Plots Used:

- Histograms of bytes_in, bytes_out
- Bar plots for protocol and country code distributions
- Box plots to compare normal vs suspicious traffic
- Heatmap of feature correlations
- Countplot of response codes (e.g., 200, 404, 500)
- Time-based line plots showing suspicious detection over time

Methodology & Implementation

Data Preprocessing:

- Encoded categorical variables like protocol, src_ip_country_code, rule_names
- Standardized numerical columns (bytes_in, bytes_out) using MinMaxScaler
- Split dataset into training and testing sets (e.g., 80:20 ratio)

Model Building:

- Baseline: Logistic Regression
- Advanced: Random Forest, XGBoost, Decision Trees
- Used GridSearchCV to optimize hyperparameters

Visualization Tools Used:

- ROC Curves for all models
- Precision-Recall curves
- Feature importance bar chart (from XGBoost and Random Forest)
- Distribution plots of bytes_in/bytes_out across classes
- Heatmaps to identify strong feature correlations

Data Preprocessing

- Cleaned and structured web traffic logs (handled missing values & duplicates)
- Detected and capped outliers using IQR method
- Normalized features using Min-Max Scaling for model compatibility

Feature Engineering

- Created derived features: session_duration, request_accel, and request frequency metrics
- Applied One-Hot Encoding to categorical variables like protocol and country code
- Integrated timestamp-based features to track temporal threat patterns

Model Development & Tuning

- Isolation Forest for unsupervised anomaly detection
- Random Forest and Logistic Regression for supervised classification
- Developed a 3-layer Neural Network for deep pattern recognition
- Used SMOTE to handle class imbalance and GridSearchCV for hyperparameter tuning
- Achieved consistent model performance with k-fold cross-validation

Model Evaluation Metrics

- Employed Confusion Matrix to track TP/FP/FN
- Used Precision, Recall, F1-Score to measure effectiveness
- Performed ROC-AUC Analysis (when multiple classes existed)
- Applied Cost-Benefit Analysis: penalized false negatives 5x higher than false positives

Result

- Achieved 92% precision, 88% recall in Isolation Forest
- Reduced false alarms by 30%, improving operational reliability
- Detected stealth attacks and periodic DDoS patterns
- Delivered interactive, actionable dashboards for cybersecurity teams

Key Insights:

- High correlation between environmental policy and visitor satisfaction
- Coastal regions scored high in sustainability due to green initiatives
- Outliers in air quality index affected overall scores