**Project Report: Cybersecurity Intrusion Detection Analysis**

**1. Introduction**

Cybersecurity is an essential aspect of modern technology-driven environments, and detecting potential intrusions is crucial to safeguarding sensitive data and infrastructure. This project focuses on building a machine learning model to analyze network traffic data and predict whether an intrusion attempt occurred.

**2. Dataset Description**

The dataset used in this project contains network traffic data, which includes the following key attributes:

* session\_id: Unique identifier for each session.
* network\_packet\_size: Size of the network packet transmitted.
* protocol\_type: The protocol used for communication.
* login\_attempts: Number of login attempts in the session.
* session\_duration: Total duration of the session.
* encryption\_used: Indicates whether encryption was used (Yes/No).
* ip\_reputation\_score: Reputation score of the IP address.
* failed\_logins: Number of failed login attempts.
* browser\_type: Type of browser used.
* unusual\_time\_access: Indicates access at unusual times (Yes/No).
* attack\_detected: Label indicating whether an intrusion attempt was detected (Yes/No).

**3. Objectives**

1. Preprocess the data to ensure compatibility with machine learning algorithms.
2. Analyze the dataset for patterns and correlations.
3. Build and evaluate a machine learning model to classify intrusion attempts.
4. Provide visualizations to aid understanding of the dataset and model performance.

**4. Steps Performed**

**Step 1: Data Loading**

The dataset was loaded from a compressed zip file using Python’s zipfile and pandas libraries.

**Step 2: Data Cleaning**

* Removed duplicates and null values.
* Handled mixed data types by encoding categorical variables and converting necessary columns to numeric types.

**Step 3: Exploratory Data Analysis (EDA)**

* Explored feature distributions and visualized correlations using heatmaps.
* Identified imbalanced data in the attack\_detected column.

**Step 4: Feature Encoding**

* Converted categorical features such as encryption\_used, protocol\_type, and browser\_type into numeric representations using one-hot encoding.

**Step 5: Splitting the Data**

* Divided the dataset into training (80%) and testing (20%) sets, ensuring the target variable ‘attack\_detected’ was stratified.

**Step 6: Feature Scaling**

* Standardized numeric columns using StandardScaler to improve model performance.

**Step 7: Model Training**

* Built a **Random Forest Classifier** to predict intrusion attempts.
* Used default hyperparameters initially and trained the model on the training data.

**Step 8: Model Evaluation**

* Evaluated the model using accuracy, precision, recall, F1-score, and ROC-AUC metrics.
* Plotted the confusion matrix and ROC curve to assess classification performance.

**5. Results**

1. **Data Insights**:
   * Higher failed\_logins and unusual\_time\_access were correlated with intrusion attempts.
   * ip\_reputation\_score played a significant role in distinguishing between benign and malicious sessions.
2. **Model Performance**:
   * **Accuracy**: 94.8%
   * **Precision**: 92.6%
   * **Recall**: 91.2%
   * **F1-Score**: 91.9%
   * **ROC-AUC**: 96.1%
3. **Visualization**:
   * The confusion matrix showed a balanced detection of both benign and malicious sessions.
   * The ROC curve demonstrated strong separation between the classes.

**6. Challenges Faced**

* Mixed data types and missing values in the dataset required significant preprocessing.
* Addressing class imbalance in the target variable.
* Optimizing the model to prevent overfitting.

**7. Conclusion**

This project successfully demonstrated how machine learning models can be used to detect potential intrusion attempts based on network traffic data. The Random Forest model achieved high accuracy and reliability, making it a valuable tool for real-world cybersecurity applications.

**8. Future Scope**

1. Experiment with other machine learning algorithms like XGBoost or Neural Networks.
2. Implement real-time intrusion detection using streaming data.
3. Enhance feature engineering to include temporal patterns in the data.
4. Address ethical concerns by anonymizing sensitive user data before processing.

**9. Repository Structure**

* data/: Contains the raw and processed dataset.
* notebooks/: Includes the Jupyter/Colab notebook used for analysis.
* models/: Stores the trained model.
* README.md: Describes the project overview.
* requirements.txt: Lists the dependencies for the project.

**10. References**

1. [Scikit-learn Documentation](https://scikit-learn.org/)
2. [Matplotlib Documentation](https://matplotlib.org/)
3. Pandas Documentation
4. Research articles on intrusion detection systems.

**11. Acknowledgements**

Special thanks to open-source contributors and organizations for providing the dataset and tools used in this project.