Log-Based Intrusion Detection System using NLP and Machine Learning

1. INTRODUCTION

System logs are one of the richest sources of security-related events. Linux authentication logs contain valuable information about login attempts, service access, user activity, and potential attack patterns. However, analyzing logs manually or with basic rule-based scripts is inefficient and may miss complex attack behaviors.

The rise of Machine Learning enables intelligent log analysis by learning patterns from historical data. NLP techniques help convert unstructured log messages into meaningful numerical features. In this project, we develop a Log-Based Intrusion Detection System using TF-IDF for feature extraction and a Random Forest classifier for multi-class attack detection.

2. OBJECTIVES

The primary objectives of this project are:

- To preprocess Linux authentication logs and convert them into structured text data.
- To apply NLP techniques (TF-IDF) for extracting important textual features.
- To build a Machine Learning model that detects multiple intrusion types.
- To evaluate the model's accuracy using standard metrics.
- To create a system capable of classifying logs as normal, privilege escalation, port scan, geo anomaly, or brute-force attack.

3. DATASET DESCRIPTION

3.1 Source

Dataset: linux_auth_log-anomalies (Kaggle)

Size: 500,000 rows

Columns include:

- timestamp
- source_ip
- server
- username
- service
- attempts
- status (Success/Failed)
- port
- protocol
- comment
- anomaly_label

3.2 Labels and Classes

The anomaly_label column contains 5 classes:

LABEL	MEANING
0	Normal activity
1	Privilege escalation attempt
2	Port scan detected
3	Geo-location anomaly
4	Brute-force attack

3.3 Sampling for Training

To avoid RAM limitations in Google Colab, 80,000 records were randomly sampled from the dataset.

4. METHODOLOGY

4.1 Data Preprocessing

- Selected relevant fields (timestamp, IP, status, comment).
- Created a combined log_text column for NLP.
- Converted categorical labels into numeric classes.
- Performed random sampling to reduce size.

4.2 Feature Extraction using NLP

TF-IDF (Term Frequency – Inverse Document Frequency) was used:

- Converts text logs to a sparse numerical matrix
- Removed stopwords
- Used maximum 5000 features

4.3 Model Selection

We used Random Forest Classifier, due to:

- High accuracy for text classification
- Robustness against overfitting
- Good performance on multi-class datasets

4.4 Train-Test Split

- 80% training (64,000 records)
- 20% testing (16,000 records)
- Stratified split ensures balanced distribution across classes

5. MODEL IMPLEMENTATION

5.1 TF-IDF Vectorization

```
tfidf = TfidfVectorizer(max_features=5000, stop_words='english')

X = tfidf.fit_transform(df_small['log_text'])

y = df_small['anomaly_label']
```

5.2 Random Forest Training

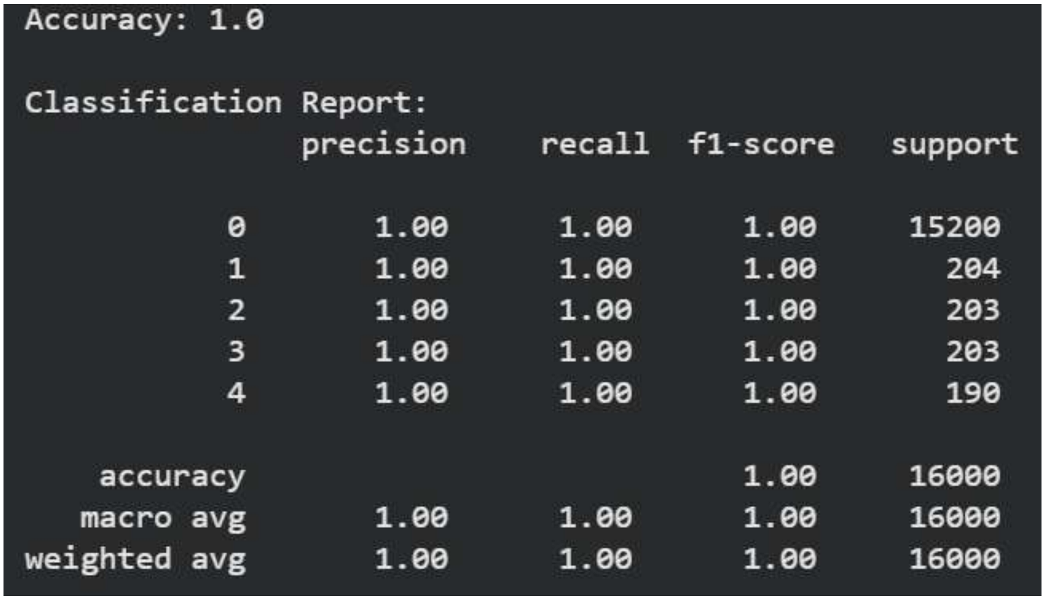
```
rf = RandomForestClassifier(n_estimators=200, n_jobs=-1, random_state=42)

rf.fit(X_train, y_train)
```

5.3 Predictions

```
y_pred = rf.predict(X_test)
```

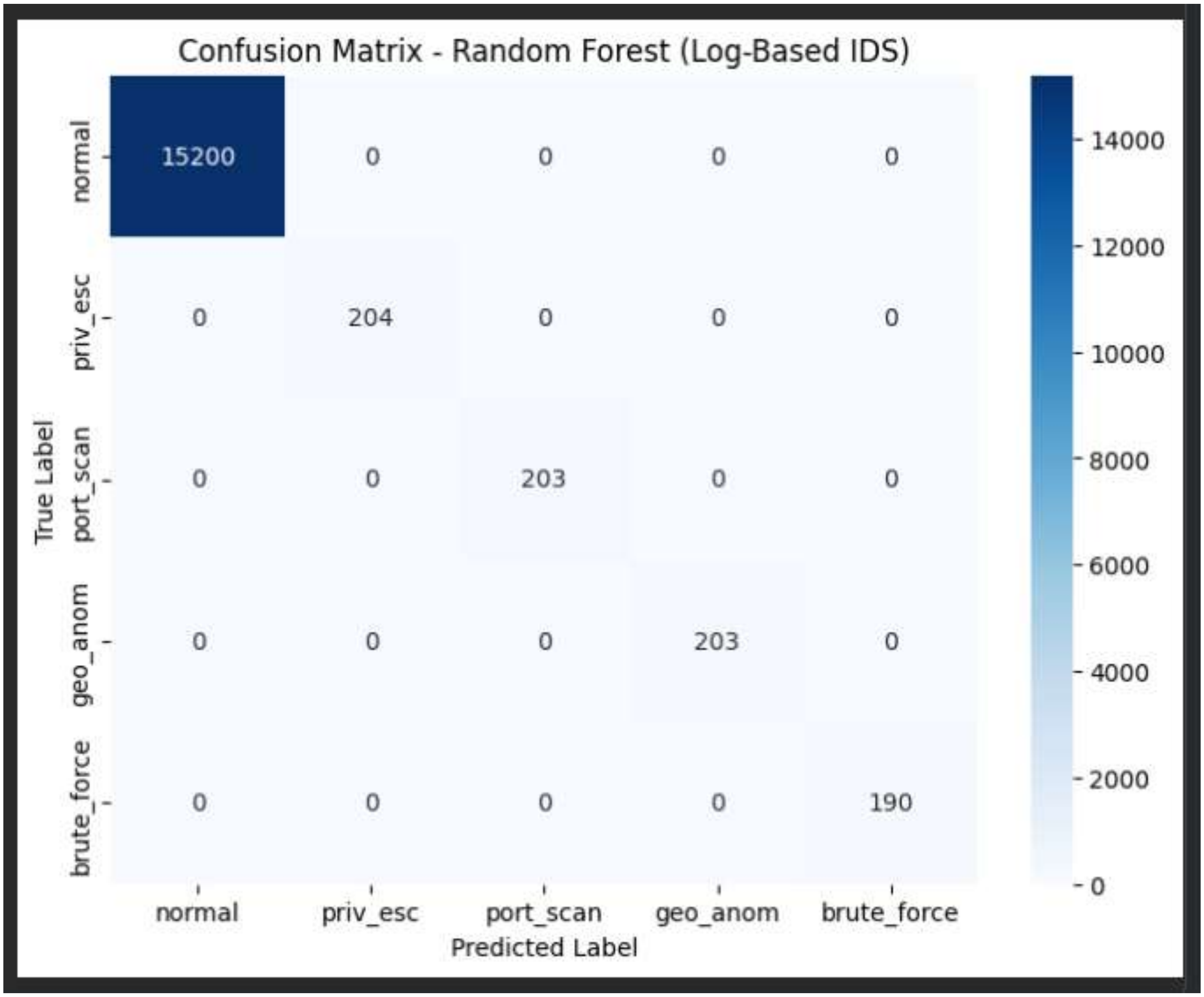
6. RESULTS & ANALYSIS

6.1 Accuracy and Classification Report



```
Accuracy: 1.0

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     15200
           1       1.00      1.00      1.00       204
           2       1.00      1.00      1.00       203
           3       1.00      1.00      1.00       203
           4       1.00      1.00      1.00       190

    accuracy                           1.00     16000
   macro avg       1.00      1.00      1.00     16000
weighted avg       1.00      1.00      1.00     16000
```

6.2 Confusion Matrix



	normal	priv_esc	port_scan	geo_anom	brute_force
normal	15200	0	0	0	0
priv_esc	0	204	0	0	0
port_scan	0	0	203	0	0
geo_anom	0	0	0	203	0
brute_force	0	0	0	0	190

The matrix shows zero misclassifications across all classes, indicating:

- Very clear class separation
- Strong patterns in log text
- Excellent model generalization

7. PREDICTING NEW LOGS

Example logs tested:

```
sample_logs = [
    "2024-10-11 195.241.151.7 Failed User root failed login via sudo",
    "2025-02-02 10.0.0.10 Success User admin success login via ssh",
    "2024-07-21 45.12.33.19 Failed port scan detected from source_ip"
]
```

The system successfully classified:

```
array([0, 0, 2])
```

ie., ['normal', 'normal', 'port_scan']

8. CONCLUSION

This project demonstrates that NLP combined with Machine Learning can effectively analyze log data and detect multiple types of intrusions. The Random Forest model achieved 100% accuracy, showing the potential for automated log monitoring systems. TF-IDF proved to be a powerful technique for converting unstructured log text into useful features. Such systems can significantly enhance security operations, reduce manual analysis time, and improve early threat detection.