**AI-Based Threat Detection System**

**Executive Summary**

This project developed an AI-based threat detection system for Microsoft 365 environments using audit log analysis. The solution leverages machine learning (Random Forest Classifier) to identify suspicious activities and potential security threats from user behavior patterns. The system processes Microsoft 365 audit logs through a comprehensive data collection, preprocessing, and analysis pipeline to predict threats based on key indicators like failed logins, suspicious operations, and anomalous geolocations.

**1. Project Overview**

Objective

To create an automated threat detection system that:

* Analyzes Microsoft 365 audit logs
* Identifies patterns indicative of malicious activity
* Predicts security threats in real-time

Methodology

1. **Data Collection**: Simulated user activities and captured audit logs
2. **Preprocessing**: Cleaned and transformed raw log data
3. **Feature Engineering**: Created meaningful threat indicators
4. **Model Development**: Built and trained a Random Forest classifier
5. **Threat Prediction**: Deployed model to classify activities as threats/non-threats

**2. Data Collection Process**

Microsoft 365 Audit Log Collection

The data collection followed a structured 5-phase process:

1. **Environment Setup**
   * Activated Microsoft 365 Business Premium Trial
   * Created 8 test users (1 admin + 7 standard users)
2. **Audit Logging Configuration**
   * Enabled audit logging via Microsoft Purview Compliance Portal
   * Activated "Start recording user and admin activity"
3. **User Activity Simulation**
   * Performed normal activities (file operations, emails)
   * Executed suspicious activities (failed logins, VPN access)
   * Waited for log generation
4. **Log Retrieval**
   * Searched audit logs with filters (date range, users)
   * Exported logs in CSV format
5. **Data Preparation**
   * Cleaned and formatted logs using Excel Power Query
   * Saved final dataset for analysis

**3. Data Preprocessing**

Key Preprocessing Steps

1. **Initial Exploration**
   * Analyzed data structure, null values, and basic statistics
   * Identified key columns and data distributions
2. **Missing Value Treatment**
   * Filled missing LogonError values with "Success"
   * Handled missing ApplicationDisplayName based on ObjectId
3. **Feature Engineering**
   * Created time-based features:
     + HourOfDay
     + DayOfWeek
     + IsWeekend
   * Developed custom risk indicators
   * Added threat labels based on activity patterns
4. **Data Cleaning**
   * Dropped irrelevant columns
   * Addressed blank entries in ResultStatus
   * Standardized categorical values
5. **Conditional Updates**
   * Marked suspicious IPs/geolocations for failed login patterns
   * Updated Geolocation to "IND" for successful logins

**4. Exploratory Data Analysis**

Univariate Analysis Highlights

1. **Record Types**
   * Type 6 (∼250) and 15 (∼160) dominated the dataset
   * Rare types (295, 25, 3) formed a long tail
2. **Applications**
   * AzureActiveDirectory (∼155) was most common
   * SecurityComplianceCenter showed high failure rates
3. **Logon Errors**
   * Success (∼550+) was predominant
   * Critical errors: InvalidUserNameOrPassword, IdsLocked
4. **Temporal Patterns**
   * Peak activity at 10 AM and 3 PM
   * 15:1 weekday-to-weekend activity ratio

Bivariate Analysis Highlights

1. **Operation vs Login Failure**
   * UserLoginFailed showed expected high failure rates
   * Suspicious patterns in MailItemsAccessed with failures
2. **ResultStatus**
   * All unknown statuses correlated with login failures
   * Strong indicator for threat detection
3. **Geolocation**
   * India (IND) showed both successful and failed activities
   * All China (CHN) activities were successful
4. **Workload Analysis**
   * SecurityComplianceCenter had highest failure rates
   * OneDrive showed trusted patterns with low failures

**5. Machine Learning Implementation**

Model Selection

* **Algorithm**: Random Forest Classifier
  + Handles non-linear relationships well
  + Provides feature importance metrics
  + Robust to outliers and noise

Feature Selection

Used **Feature Importance** to select key predictors:

* Operation\_FileAccessed
* Operation\_FileModified
* Operation\_FileUploaded
* Operation\_FileDownloaded
* Operation\_MoveToDeletedItems
* IsRiskyHour
* Operation\_UserLoginFailed
* GeoLocation\_IND
* ClientIP\_203.28.245.168
* ClientIP\_192.168.1.100
* ResultStatus\_unknown

Target Variable

* IsThreat: Binary (Yes/No) threat indicator

**6. Threat Detection System**

Input Parameters

Users can input the following activity features for threat assessment:

1. File operations (access, modification, upload/download)
2. Login failures
3. Suspicious timings (IsRiskyHour)
4. Geolocation (India)
5. Client IP addresses (specific risky IPs)
6. Unknown result statuses

Output

* Binary threat prediction (Yes/No)
* Probability score for threat likelihood

Key Threat Indicators

1. Multiple failed login attempts
2. File operations during risky hours
3. Activities from suspicious IPs
4. Unknown result statuses
5. SecurityComplianceCenter access failures

**7. Conclusion and Recommendations**

Key Findings

1. **Data Patterns**
   * Most threats manifest through login failures and unknown statuses
   * SecurityComplianceCenter is a high-risk workload
   * Time-based patterns are strong threat indicators
2. **Model Insights**
   * Random Forest effectively captures complex threat patterns
   * Feature importance aligns with security best practices

Recommendations

1. **Monitoring Focus**
   * Prioritize SecurityComplianceCenter activities
   * Monitor all unknown result status events
   * Track India-based activities more closely
2. **System Improvements**
   * Add real-time monitoring capabilities
   * Incorporate more threat indicators over time
   * Implement alerting system for high-risk predictions
3. **Future Enhancements**
   * Include more geolocation data
   * Add multi-factor authentication events
   * Incorporate user behavior analytics

**8. Final Assessment**

This AI-based threat detection system provides a robust framework for identifying potential security threats in Microsoft 365 environments. By combining comprehensive data collection, thorough preprocessing, and machine learning, the solution offers actionable insights into suspicious activities. The Random Forest model demonstrates strong predictive capability, particularly in detecting login-based threats and anomalous file operations. This system can significantly enhance organizational security posture by enabling proactive threat detection and response.