**Course:** DSA – Semester 3

**Deliverable Title:** Core Data Structures Implementation and Behavioral Anomaly Detection Foundation

**Team Members:** Ali & Fatima

**Date:** October 19, 2025

## 1. Problem Statement

Modern digital systems increasingly struggle to detect insider-driven cyber incidents—cases where authorized users or automated processes perform actions that appear legitimate but are actually malicious.  
  
In a simulated case, an internal operator initiated a large-scale data exfiltration attempt, exporting several gigabytes of confidential logs. Traditional monitoring tools failed to recognize this as a threat because all operations were performed within valid access permissions.  
  
This scenario highlights the growing importance of Behavioral Anomaly Detection (BAD)—the ability to recognize patterns that deviate from normal behavior. Our project focuses on developing the core data structures and monitoring utilities that will serve as the technical foundation for such systems. These tools will enable future modules to detect anomalies in real-time, such as irregular memory usage, repetitive I/O operations, or abnormal command sequences.

## 2. Motivation and Necessity for Behavioral Anomaly Detection (BAD)

The primary motivation behind BAD is to identify deviations from expected behavior that may indicate a security threat or operational anomaly. Anomaly detection can be described as “the process of finding patterns in data that do not conform to expected behavior.”

## 2.1. Motivation in Cybersecurity and Threat Detection

In cybersecurity, BAD—often implemented as User and Entity Behavior Analytics (UEBA)—is critical for combating intelligent, persistent, and stealthy threats.  
  
• Combating Advanced Persistent Threats (APT): APTs are long-term, organized attacks that remain hidden within networks. BAD enables early identification of unusual behavior indicative of such attacks.  
• Proactive Defense: Detecting behavioral anomalies early helps prevent larger breaches, such as data leaks or system compromises.  
• User Irregularity Analysis: BAD systems establish a baseline for normal user activity and flag deviations (e.g., a user who typically downloads 20 MB daily suddenly downloads 4 GB).  
• Early Threat Indicators: Monitoring early behavioral shifts provides valuable warning signs of potential attacks, allowing proactive response.

## 2.2. Motivation in Military and Situational Awareness

In defense systems, BAD enhances situational awareness by recognizing behavioral deviations that might signal deception or strategic change.  
  
• Indicators of Critical Change: Sudden shifts in entity behavior may indicate outdated predictive models or environmental changes.  
• Alerting Decision-Makers: Detecting when and where behavioral anomalies occur supports timely human or automated intervention.  
• Counteracting Deception: Entities may behave normally to avoid suspicion, or abnormally to create confusion. BAD helps distinguish between intentional deception and genuine anomalies.

## 2.3. Necessity for Robust Detection Methods

Real-world anomaly detection requires algorithms that can handle complex, uncertain, and dynamic environments.  
  
• Data Scarcity: Anomalous data is rare, and distinguishing it from normal variations is difficult.  
• Environmental Adaptability: Detection systems must adapt to evolving baselines and changing user behavior.  
• Temporal Awareness: Behavioral data often changes over time, demanding methods that can track and evaluate temporal dependencies accurately.

## 3. Executive Summary

Deliverable 2 centered on building and integrating core data structures and supporting utilities that form the foundation of our system monitoring and anomaly detection framework.  
  
The project was divided into two modules—Ali’s and Fatima’s—each addressing key areas of data management and system monitoring. All modules were compiled and tested successfully through a unified main.cpp file, confirming stable integration and functionality.

## 4. Technical Contributions and Data Structures

The deliverable was split into two major modules, focusing on linked list-based and array-based structures respectively.

## 5. Simulated Cyber Incident: Data Exfiltration Attempt

A simulated cyber-incident was conducted to evaluate the project’s anomaly detection capabilities. A background process attempted to export approximately 4 GB of internal logs, despite the normal daily data transfer being under 50 MB.  
  
The system successfully logged the anomaly, halted further write operations, and issued an alert to the administrator, confirming the early detection and containment capability.

## 6. Integration and Validation

Both modules were integrated through a unified build setup, ensuring smooth compilation and linkage. Functionalities such as memory tracking, queue integrity, and file persistence were verified.

## 7. Conclusion and Future Work

Deliverable 2 was successfully completed, providing a stable and extensible foundation for future anomaly detection work. Deliverable 3 will focus on automating anomaly recognition using behavioral scoring models and implementing advanced data structures such as Trees (BST/AVL) and Hash Tables for long-term behavior profiling.