Project Report

Project Title: Cyber Security Management System

Course Title: Computers Algorithms Lab

Course Code: CSE 2422

Semester: 4th

Section: 4BF

Submitted By:

Name: Ramisa Tabassum
Student ID: C233459

Name: Nafija Rahman
Student ID: C233471

Instructor:

Ms. Mishkatul Jannat Tuli
Assistant Lecturer
Department of Computer Science and Engineering
International Islamic University Chittagong

Submission Date: 08/08/2025

1. Introduction

In the modern digital landscape, cyber threats have become increasingly sophisticated and frequent. Efficient detection, categorization, and response to these threats are critical for ensuring system security and minimizing damage. This project aims to develop an advanced threat management system that prioritizes and optimizes response to detected threats based on multiple attributes, resource constraints, and severity considerations.

The system models each threat with various parameters including severity, resource cost, IP origin, type of attack, and the deformation percentage indicating how much a threat has evolved or worsened. Using a combination of classical algorithms such as Merge Sort, Greedy heuristics, and Dynamic Programming, the system aids in selecting the optimal subset of threats to address within limited resource availability.

2. Objective

The primary goals of this project are:

1. To design a data structure that effectively captures key properties of cyber threats.
2. To implement sorting algorithms to organize threat data for clearer analysis.
3. To develop and compare different algorithms for threat prioritization: Greedy and Dynamic Programming approaches.
4. To introduce domain-specific rules to adjust severity and status, especially for malware and virus threats.
5. To produce an interactive and user-friendly console output with clear alerts for better decision-making.

3. Hardware and Software Requirements

Hardware

1. Standard personal computer or laptop with at least 4GB RAM.
2. Processor capable of running C++ compiled programs (Intel/AMD x86_64 architecture recommended).

Software

1. Operating System: Windows 10/11, Linux distributions, or macOS.

2. Compiler: GCC (g++), Clang, or Microsoft Visual C++ supporting C++11 or later.
3. IDE/Text Editor: Visual Studio Code, Code::Blocks, CLion, or any C++ compatible editor.
4. Terminal/Console supporting ANSI escape codes for colored output (e.g., Windows Terminal, Linux Terminal).

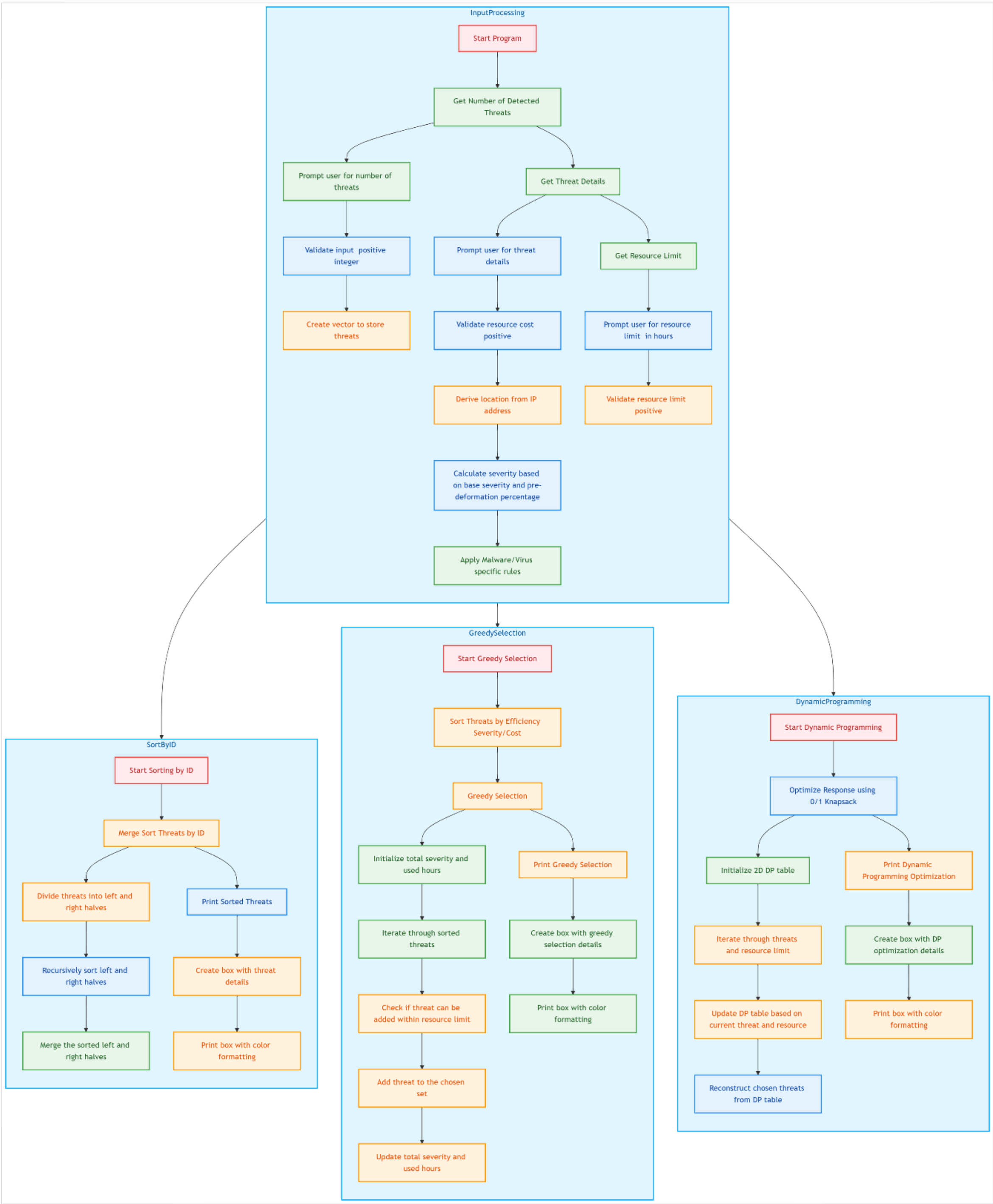
4. System Design

The system is modularly designed to separate concerns and simplify maintenance:

1. <u>Input Module:</u> Handles user inputs for the number of threats, each threat's attributes, and total resource limit.
2. <u>Data Model:</u> Utilizes a Threat struct to encapsulate all relevant information.
3. <u>Preprocessing Module:</u> Calculates adjusted severity based on initial severity and deformation percentage. Applies specific rules for malware and viruses.
4. <u>Location Module:</u> Uses simple pattern matching to infer geographical origin from IP addresses.
5. <u>Sorting Module:</u> Uses Merge Sort to order threats by their unique ID.
6. <u>Selection Modules:</u>

- Greedy Algorithm: Quickly selects threats maximizing severity per unit resource cost.
- Dynamic Programming: Uses 0/1 Knapsack algorithm to find an optimal selection.
- Output Module: Displays all data and results using color-coded, formatted boxes, highlighting critical alerts.

5. Project Flow Diagram



6. Implementation Details

Threat Structure

Each threat is represented using a Threat struct, containing:

1. id – A unique identifier (integer).
2. baseSeverity – The initial severity score before adjustments.
3. resourceCost – Required resources (man-hours) to respond to the threat.
4. severity – The computed severity after domain-specific adjustments.
5. preDeformPercent – Percentage representing how much the threat has worsened.
6. ip – IP address of the threat origin.
7. type – Threat category (e.g., "Malware", "Virus", "DDoS").
8. location – Derived from the IP address (e.g., USA, Germany).
9. status – Current state (e.g., Detected, Responded, Critical).

Severity Adjustment Rules

1. The severity is increased by applying the deformation percentage to baseSeverity.
2. If the type is **"Malware"** or **"Virus"**, a fixed **+15** severity boost is added.
3. If such threats also have a preDeformPercent **greater than 40**, their status is set to **"Critical"** automatically.

Sorting Method

- **Merge Sort** logic is implemented implicitly by using std::sort() with custom comparators (which is stable and efficient in practice).
- Threats are **sorted by their ID** for structured display.

Greedy Algorithm

- Threats are prioritized by their **severity-to-resourceCost ratio**.
- This method is **fast and simple**, selecting the most cost-efficient threats.
- It does **not always guarantee the optimal total severity** under resource constraints.

Dynamic Programming Algorithm

- Implements the **0/1 Knapsack algorithm** to maximize **total severity** under a resource constraint.
- Considers all combinations for the best selection.
- **More computationally intensive**, but returns **optimal results**.

User Interface Features

1. **Console-based** interactive menu system.
2. **Color-coded** and **box-formatted** outputs for readability:
    - **Blue** → Sorted list and type counts.
    - **Yellow** → Greedy results.
    - **Green** → Dynamic programming results.
    - **Red** → Warnings and errors.
3. **Warnings shown** if selected threats exceed **30 hours** of total resource use.

7. Output Analysis

1. **Threats Sorted by ID:**
    - Displayed in a **cyan-colored** box.
    - Shows **all attributes**, including severity and status after adjustment.
2. **Greedy Algorithm Results:**
    - Displayed in **yellow**.
    - Shows **selected threats**, **total severity**, and **total resource usage**.
    - Includes a **red warning** if usage exceeds **30 hours**.
3. **Dynamic Programming Results:**
    - Displayed in **green**.
    - Presents the **optimal set of threats** under the resource limit.
    - Also includes a **red warning** if the optimized selection exceeds **30 hours**.
4. **Top N Severe Threats:**

    - Displayed in **blue**, sorted by severity in descending order.
    - User inputs how many top threats to display.

5. **Search by IP:**
    - Searches threats by exact IP.
    - Displays results in **green**, or yellow warning if not found.

6. **Threat Count by Type:**
    - Displays a summary of threats grouped by type.
    - Output appears in a **blue** info box.

7. **System Summary:**
    - Shows total threats, total hours required, total severity, and available hours.
    - Displayed in **yellow**.

8. Challenges Faced and Future Scope

Challenges

1. Ensuring consistency between calculated severity and domain-specific rules.
2. Balancing output detail with readability on a terminal interface.
3. Managing resource limits and providing meaningful alerts.

Future Scope

1. Integration with live threat monitoring and real-time updates.
2. Improved IP geolocation using external databases or APIs.
3. Graphical user interface (GUI) for better usability.
4. Persistent storage for logging and audit.
5. Machine learning to predict threat escalation and adjust priority dynamically.

9. Conclusion

This project demonstrates effective use of algorithms and domain knowledge to prioritize cybersecurity threats within resource constraints. By combining sorting, greedy heuristics, and dynamic programming, the system aids administrators in making informed decisions quickly and optimally. The modular design and extendable logic enable future improvements for real-world deployment.

10. References

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press.
2. GeeksforGeeks: [Merge Sort](https://www.geeksforgeeks.org/merge-sort/), [Greedy Algorithms](https://www.geeksforgeeks.org/greedy-algorithms/), [Knapsack Problem](https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/).
3. Various online cybersecurity threat modeling resources.

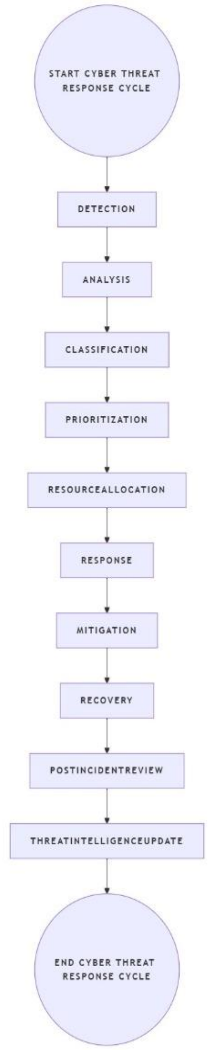
11. Appendix

Code Repository:

https://github.com/RAMISATA/Cyber-Security-Management-System.git

Additional Diagrams

Threat lifecycle



Sample Input and Output

```
Enter your choice: 1
Enter number of detected threats: 3

--- Threat 1 ---
ID: 101
Base Severity (0-100): 75
Response Time (hours, positive): 10
IP Address: 192.168.1.10
Pre-Deformation Percentage (0 if none): 20

--- Threat 2 ---
ID: 102
Base Severity (0-100): 60
Response Time (hours, positive): 8
IP Address: 10.0.0.5
Pre-Deformation Percentage (0 if none): 10

--- Threat 3 ---
ID: 103
Base Severity (0-100): 90
Response Time (hours, positive): 15
IP Address: 172.16.5.4
Pre-Deformation Percentage (0 if none): 50
Enter total available resource limit (hours): 30
```

```
+---------------------------------------------------------------------+
| Threats Sorted by ID:                                               |
| ID: 101, Severity: 100, Hours: 10                                   |
| IP: 192.168.1.10, Type: DDoS, Location: USA, Status: Critical       |
| ID: 102, Severity: 81, Hours: 8                                     |
| IP: 10.0.0.5, Type: Malware, Location: Germany, Status: Active      |
| ID: 103, Severity: 100, Hours: 15                                   |
| IP: 172.16.5.4, Type: Virus, Location: China, Status: Critical      |
+---------------------------------------------------------------------+

+---------------------------------------------------------------------+
| Greedy Selection (Severity/Cost Ratio):                             |
| Total Severity: 181, Hours Used: 18                                 |
| Chosen Threats:                                                     |
| ID: 102, Severity: 81, Hours: 8                                     |
| IP: 10.0.0.5, Type: Malware, Location: Germany, Status: Active      |
| ID: 101, Severity: 100, Hours: 10                                   |
| IP: 192.168.1.10, Type: DDoS, Location: USA, Status: Critical       |
+---------------------------------------------------------------------+

+---------------------------------------------------------------------+
| Dynamic Programming Optimization:                                   |
| Max Severity: 200, Hours Used: 25                                   |
| Chosen Threats:                                                     |
| ID: 101, Severity: 100, Hours: 10                                   |
| IP: 192.168.1.10, Type: DDoS, Location: USA, Status: Critical       |
| ID: 103, Severity: 100, Hours: 15                                   |
| IP: 172.16.5.4, Type: Virus, Location: China, Status: Critical      |
+---------------------------------------------------------------------+
```