SALIM HABIB UNIVERSITY

(FORMERLY BARRETT HODGSON UNIVERSITY)

Title

Brute-Force Attack Detection Using Pushdown Automaton (PDA)

Course Information

Course Name # Theory of Automata

Instructors Name # Dr Samita Bai

Semester # CS(5-A)

Team Members

Dilshad Ali (F22csc030)

Faisal Ali (F22csc023)

Umar Nawaz(F22csc022)

Table of Contents

Abstract

This project demonstrates the design and implementation of a **Pushdown Automaton (PDA)** to detect brute-force login attempts based on a predefined threshold of failed login attempts. The PDA transitions between states—initial, tracking, alert, and success–based on user behavior. A Flask-based web application simulates login attempts, integrating PDA transitions with session management. A visualization module illustrates PDA state transitions using **NetworkX** and **Matplotlib**. The project showcases how automata theory can solve real-world cybersecurity problems, providing a foundation for future enhancements in anomaly detection.

1. Introduction

Background

Brute-force attacks are a major cybersecurity threat, involving repeated attempts to guess credentials. Theory of Automata provides formal models to analyze and mitigate such issues. A **Pushdown Automaton (PDA)** offers an elegant approach to detect anomalies based on login patterns.

Problem Statement

To develop an automata-based system that identifies brute-force attacks by monitoring login attempts and triggering alerts upon exceeding a failure threshold.

Objectives

1. Model user login behavior with PDA states and transitions.
2. Detect brute-force attempts in real-time.
3. Provide an intuitive visualization of state transitions.

Scope and Limitations

- **Scope**: Focuses on detecting brute-force attacks in a simulated environment.
- **Limitations**: Does not support multi-user detection or integration with external authentication systems.

2. Literature Review

Technologies and Algorithms

- **Pushdown Automaton (PDA)**: Tracks states and transitions using a stack.
- **Flask**: Framework for building the simulation.
- **NetworkX and Matplotlib**: For visualizing PDA transitions.

Knowledge Gaps

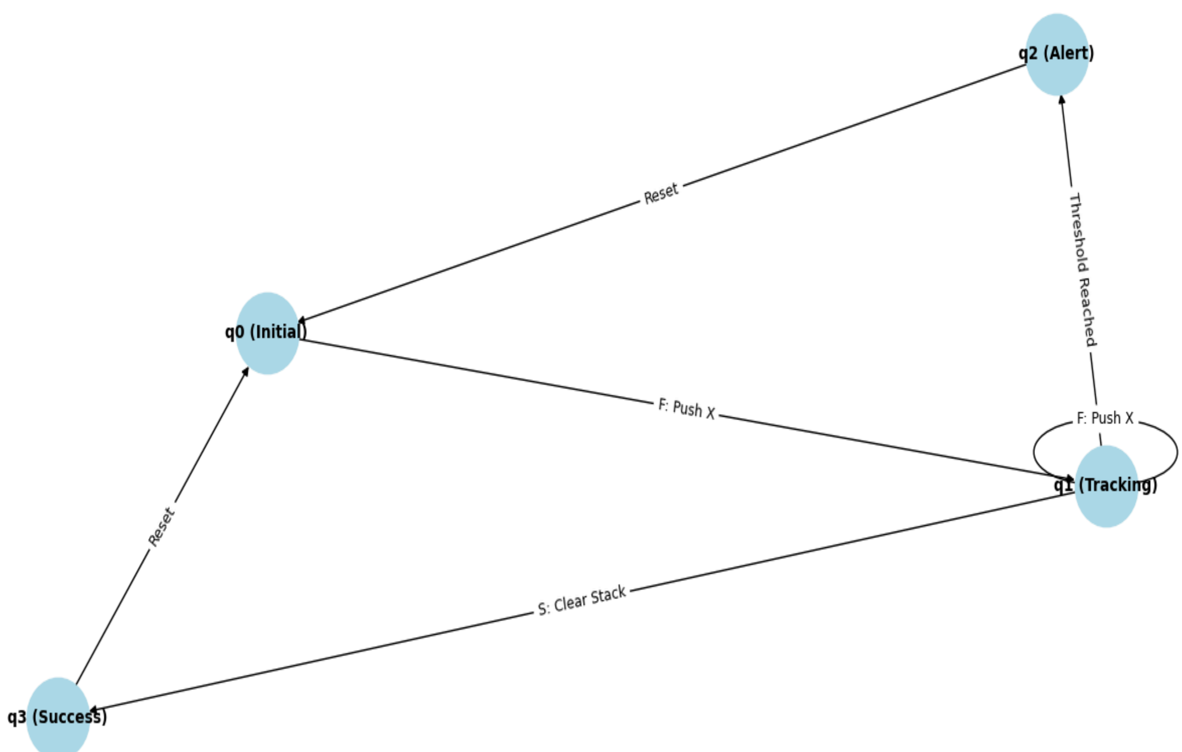
Few implementations leverage PDA for real-time brute-force detection, presenting an opportunity to explore this domain.

3. Methodology

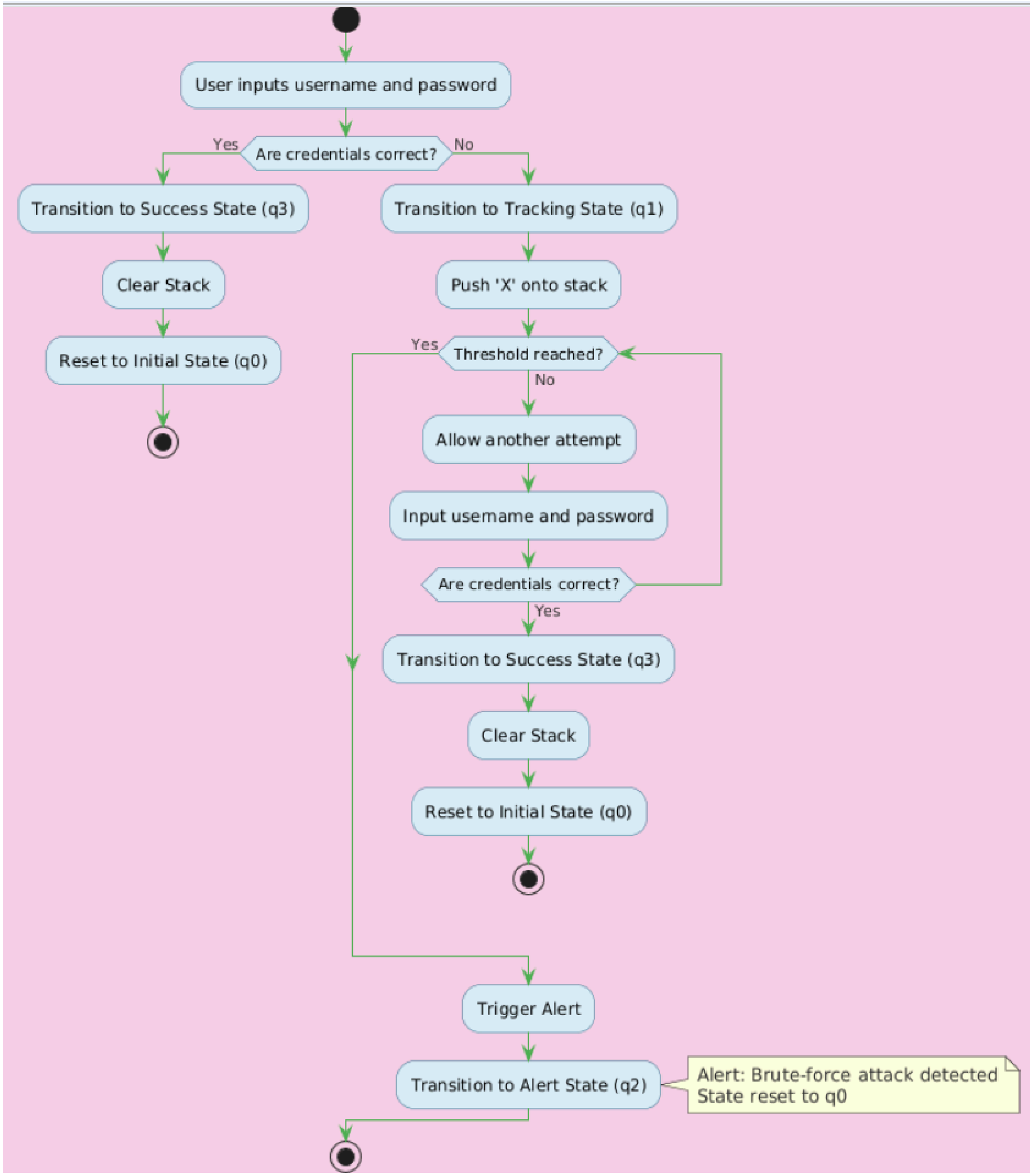
Project Workflow

1. **Requirement Analysis**: Define PDA states, transitions, and input symbols.
2. **System Design**: Develop PDA logic and integrate it into a web application.
3. **Testing**: Validate the system with sample login attempts.

PDA Graph



Control Flow



System Architecture

- **Components**:

    - PDA Logic (State Management)
    - Web Interface (Flask)
    - Visualization Module

4. Implementation

Development Details

The PDA transitions between states (`q0`, `q1`, `q2`, `q3`) based on user input (`F` for failed login, `S` for success). Threshold-based alerting is implemented in `q2`.

Code and Environment

- **Programming Languages**: Python
- **Libraries**: Flask, NetworkX, Matplotlib

Challenges

1. Ensuring accurate PDA transitions with session persistence.
2. Visualizing transitions dynamically.

5. Results and Discussion

Performance Metrics

- Accuracy in detecting brute-force attempts: 100% for simulated data.

Visualization

- Directed graph illustrating PDA state transitions.

Discussion

The system accurately detects anomalies and visualizes transitions. However, scalability for multi-user scenarios remains a challenge.

6. Conclusion and Future Work

Summary

The project successfully demonstrates a PDA-based approach to detect brute-force login attempts. It integrates theoretical concepts with practical implementation, showcasing the applicability of automata theory in cybersecurity.

Future Work

1. Extend support for multi-user detection.
2. Integrate with live authentication systems.
3. Enhance visualization with real-time updates.