AegisSecure: AI-Powered Threat Detection App

Part of the Requirements Not Completed & Challenges Faced

Group Number: 35
Mevada Soham Meghalkumar [202301484]
Gohil Suryadeepsinh Hardevsinh [202301463]
Rana Neelabh Vijaykumar [202301476]
Hrithik B Patel [202301441]
Vadsmiya Pransu Pradipkumar [202301445]
Dhruv Jigneshkumar Patel [202301095]
Bhagiya Jenish Rameshbhai [202301480]
Akshat Bhatt [202301460]
Vrajkumar Makwana [202301436]
Chavda Mihirsinh Labhubhai [202301479]

November 25, 2025

IT314 - Software Engineering
Prof: Saurabh Tiwari
Mentor: Shyam Patel

Contents

1 Part of the Requirements Not Completed/Implemented

The following features, originally outlined in our requirements document, were either partially implemented or de-scoped from the final submission due to technical constraints.

- **Multilingual Analysis (FR-010):**
  Status: Partially Implemented.
  The feature has been implemented, but the performance is not on par with the primary English detection model. It is currently functioning as a Beta feature with variable accuracy.
- **User Feedback Loop (FR-006):**
  Status: Not Implemented.
  The functionality for users to correct or flag results (e.g., "Report False Positive") was not developed.
- **Push Notifications (FR-007):**
  Status: Not Implemented.
  Real-time push alerts for detected threats were not implemented due to configuration conflicts between the Telephony API and Android's notification requirements.
- **Settings & Account Preferences (FR-009):**
  Status: Partially Implemented.
  A settings page exists, but its functionality is limited. It currently serves only to toggle the "Automatically Fetch SMS" option. Other preferences (such as language selection or notification controls) were not implemented.
- **Multilingual App UI (FR-012):**
  Status: Not Implemented.
  The User Interface remains English-only; localization for other languages was not implemented.

2 Challenges Faced

Throughout the development lifecycle of AegisSecure, the team encountered several significant technical hurdles. The following list outlines the major challenges we faced during the entire process of building the project, detailing the complex issues we solved and the architectural decisions we made to overcome them.

1. **SMS Read Permissions:**
   Access to SMS data was denied on Android 11+ due to strict permission changes. We had to undergo the Google Play Protect review process and request restricted permissions to resolve this.
2. **Google Pub/Sub Duplication:**
   Google Pub/Sub sends multiple notifications for the same email, which was initially unknown to us. This led to multiple entries being passed to the model, causing heavy token usage and delays in output generation. We changed the flow: Pub/Sub now notifies the backend, which first inserts the message into the database to ensure uniqueness. Then, an asynchronous worker polls the database and processes only those messages not yet analyzed.
3. **WebSocket Redundancy:**
   Due to Pub/Sub duplications and unidentified extra WebSocket connections, 6-7 requests were sometimes hitting the model for a single message instead of one. We tried to enforce unique WebSocket connections but duplicate sockets still formed. We changed the flow to rely on the database and the asynchronous worker (as described above), removing WebSockets completely.

4. **Notifications Configuration:**
   We faced issues integrating the older version of the Telephony API with the new configuration requirements for Android notifications. We were unable to solve these compatibility issues within the timeline.