Security Audit Report

This report outlines the security measures implemented in the Flask - React application, identifies potential vulnerabilities, and describes the steps taken to mitigate these vulnerabilities.

The application consists of a user portal with authentication controlled via Firebase, a Flask-based WhatsApp bot that performs secure chat with encryption, and validates message signatures before responding. User information is stored securely on Firebase.

Security Measures Implemented

1. Authentication via Firebase:

- OAuth 2.0: Utilized Firebase Authentication for secure user login, employing OAuth 2.0 to handle authentication flows.
- JWT Tokens: Firebase issues JWT (JSON Web Tokens) for maintaining user sessions securely.
- Multi-Factor Authentication (MFA): Allows optional MFA to enhance security for sensitive accounts.

2. Secure Communication (Via Whatsapp Bot):

- HTTPS: Both the React frontend and Flask backend are served over HTTPS to ensure encrypted data transmission.
- End-to-End Encryption: Implemented for messages between the WhatsApp bot and users. Uses strong encryption algorithms (e.g., AES-256).
- Signature Validation: Messages are signed and validated using cryptographic signatures to ensure authenticity and integrity.

3. Data Security:

- Firebase Security Rules: Configured to restrict access to user data based on authentication status and roles.
- Data Encryption: All sensitive data stored on Firebase is encrypted.
- Environment Variables: Secrets such as API keys and database credentials are stored securely using environment variables.

Performed Error Logging: Configured detailed error logging to monitor suspicious activities.

Potential Vulnerabilities and their Mitigations

1. Potential Vulnerability: Authentication Bypass

Mitigation: Ensured robust implementation of Firebase Authentication with strict validation of JWT tokens on the backend.

2. Potential Vulnerability: Man-in-the-Middle (MitM) Attacks

Mitigation: Enforced HTTPS for all communications. Utilized HSTS (HTTP Strict Transport Security) to ensure browsers only connect over HTTPS.

The Flask - React application demonstrates comprehensive measures in place to protect against various threats. Future security audits should focus on emerging threats and evolving best practices to ensure ongoing protection.