🔐 Security Configuration Report

Chat Application Database (PostgreSQL)

Ensuring Secure Access, Role Management, and Full Auditability

1. Introduction

This document outlines a robust database security framework for a chat application built using PostgreSQL. The application supports:

• One-on-one and group chats.

• Multimedia messaging (text, images, video, files, and voice notes).

• Voice and video calls.

• User statuses (like Stories).

• Device tracking, notification delivery, and custom user settings.

To secure data access and ensure traceability, this plan uses:

• Role-Based Access Control (RBAC).

• Granular table and column privileges.

• Audit trails via triggers for all major entities.

• Daily reporting system for monitoring activity.

2. Core Tables and Functions

|  |  |
| --- | --- |
| Table | Description |
| Users | Stores user profiles, contact lists, and account metadata. |
| Chat | Represents chat sessions (personal or group-based). |
| Messages | Stores text, media, or voice messages within chats. |
| Group | Contains metadata for group chats including membership and admin lists. |
| Call | Records voice and video call sessions, participants, and durations. |
| Status | Temporary posts visible to selected contacts, with media attachments. |
| Media | Stores metadata and links for uploaded files shared in messages/statuses. |
| Notifications | Alerts users about messages, calls, or events. |
| Device | Tracks login devices, types, and authentication status. |
| Settings | Saves interface and behavior preferences for users. |
| Privacy Settings | Controls visibility of profile elements like last seen or photo. |
| Audit\_logs | Stores tracked changes across major tables. |
| Daily\_reports | Summarizes all audit activity by date, for review or alerts. |

3. Defined Database Roles

|  |  |
| --- | --- |
| Roles | Description |
| Chat\_admin | Superuser role for full access, maintenance, and moderation. |
| Chat\_user | Regular authenticated user with access only to their personal data. |
| Chat\_auditor | Security reviewer who reads logs and reports but cannot change data. |

4. Access Privilege Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Chat\_admin** | **Chat\_user** | **Chat\_auditor** |
| User | All | SELECT/UPDATE (own data only) |  |
| Chat | All | SELECT, INSERT, DELETE (if participant |  |
| Message | All | SELECT, INSERT, DELETE (own, in allowed chat) |  |
| Group | All | SELECT, UPDATE (if admin), DELETE (if creator) |  |
| Call | All | SELECT/INSERT (own calls) |  |
| Status | All | INSERT/SELECT (from contacts), DELETE (own) |  |
| Media | All | INSERT/SELECT (own), DELETE (own) |  |
| Notification | All | SELECT (own notifications) |  |
| Device | All | SELECT/INSERT (own devices) |  |
| Settings | All | SELECT/UPDATE (own settings) |  |
| PrivacySettings | All | SELECT/UPDATE (own settings) |  |
| Audit\_logs | SELECT |  | Can Select on all via Logs |
| Daily\_report | SELECT |  |  |

5. Auditing & Tracing Changes

🔸 audit\_log Table

Used for tracking inserts, updates, and deletions across all key tables.

Columns include:

• audit\_id (UUID)

• table\_name (e.g., Message, Group, Call)

• operation (INSERT / UPDATE / DELETE)

• old\_data (JSON, for updates/deletes)

• new\_data (JSON, for inserts/updates)

• changed\_by (UserID)

• changed\_at (timestamp)

🔸 Trigger-Based Auditing

Each of these tables has an AFTER INSERT OR UPDATE OR DELETE trigger:

• User

• Chat

• Message

• Group

• Call

• Status

• Media

• Notification

• Device

• Settings

• PrivacySettings

All triggers log changes into audit\_log.

6. Daily Reporting

🔹 daily\_reports Table

Summarizes daily audit activity to detect unusual behavior or volume spikes.

Key fields:

• report\_id (UUID)

• summary\_date

• total\_changes

• table\_activity\_summary (JSON showing per-table change counts)

• most\_active\_users (JSON array)

• flagged\_events (suspicious patterns, e.g., >1000 deletes)

🔹 Report Generator

• Scheduled job (e.g., via pg\_cron or external scheduler).

• Aggregates all audit\_log entries from past 24h.

• Populates the daily\_reports table.

• Alerts chat\_admin if anomalies are detected.

7. Additional Security Measures

|  |  |
| --- | --- |
| **Category Applied** | **Controls** |
| Authentication | Devices tracked via the Device table with timestamp & status |
| Authorization | Fine-grained roles/permissions for users vs. auditors vs. admins |
| Data Integrity | Enforced with foreign keys and constraints between all related tables |
| Audit Compliance | Full trace of who changed what, when, and how |
| Privacy | User-controlled visibility via PrivacySettings |
| Media Protection | Access to Media rows based on ownership or linked Message visibility |

8. Conclusion

This security and audit framework for the chat app ensures:

* Strong access controls: Only authorized users can access or modify data relevant to them.
* Complete auditability: All user actions are recorded, reviewable, and reportable.
* Real-time and historical traceability: Via audit\_log and daily reports.
* User data protection and privacy: Through granular privacy settings and device-level tracking.

Together, these measures help protect the chat platform from misuse, ensure compliance with best practices, and support a transparent and secure communication environment.