Information Security Policy & Procedures

Effective date: February 2026 **Owner:** Crumbs Money Engineering **Review cadence:** Annually or after any security incident

---

1. Purpose

This document defines the information security policies and procedures Crumbs Money follows to identify, mitigate, and monitor risks related to the handling of user financial data. Crumbs Money is a read-only personal finance dashboard that accesses bank, credit, and investment data through Plaid. We never initiate transfers or move funds.

---

2. Scope

This policy applies to:

- All application infrastructure (frontend, backend, database, hosting)
- All third-party integrations (Plaid, Firebase, Railway)
- All team members with access to production systems
- All user data including authentication credentials, Plaid access tokens, and financial data

---

3. Data classification

Classification	Examples	Handling
Critical	Plaid access tokens, Firebase service account keys, database credentials	Encrypted at rest, never committed to source control, access restricted to production environment only
Sensitive	Transaction data, account balances, account names, user identity (Firebase UID)	Stored in encrypted database, scoped per user, accessed only through authenticated API calls
Internal	Application logs, error messages, infrastructure configuration	Restricted to authorized team members, scrubbed of sensitive data before logging
Public	Marketing content, open-source dependencies	No restrictions

---

4. Authentication & access control

4.1 User authentication

- Users authenticate via **Google Sign-In** through Firebase Authentication.
- The backend verifies Firebase ID tokens on every API request using Firebase Admin SDK.
- No passwords are stored by Crumbs Money — authentication is fully delegated to Google/Firebase.
- Sessions are managed by Firebase; ID tokens are short-lived and refreshed automatically.

4.2 Data isolation

- **Identity comes from the verified auth token only**, never from URL parameters, request body, or query strings.
- Every database query and Plaid API call is scoped by the authenticated user's Firebase UID (`req.uid`).
- There is no admin endpoint or cross-user data access path in the application.

4.3 Infrastructure access

- Production database credentials are stored as environment variables on the hosting platform (Railway), not in source code.
- Firebase service account keys are stored as files on the server, excluded from version control via `.gitignore` .
- Access to production infrastructure (Railway, Firebase Console, Plaid Dashboard) is restricted to authorized team members with individual accounts.

---

5. Data storage & encryption

5.1 Database

- User data is stored in a **PostgreSQL** database hosted on Railway.
- Railway Postgres uses **encryption at rest** (AES-256) and **encryption in transit** (TLS).
- Plaid access tokens are stored in the `plaid_items` table. These tokens grant read-only access to a user's financial data and are scoped to the products requested (Transactions, Investments).

5.2 What we store

Data	Stored?	Retention
Plaid access tokens	Yes	Until user disconnects the connection
Transaction history	Yes	Until user disconnects the connection
Account balances	No	Fetched live from Plaid on each request
Investment holdings	No	Fetched live from Plaid on each request
User passwords	No	Authentication delegated to Google/Firebase
Bank credentials	No	Handled entirely by Plaid Link; never touch our servers

5.3 Data deletion

- When a user disconnects a connection, the associated `plaid_items` row and all related `transactions` rows are deleted from the database.
- The Plaid access token is revoked via Plaid's `/item/remove` API.
- No financial data is retained after disconnection.

---

6. Third-party security

6.1 Plaid

- Bank credentials are entered exclusively in **Plaid Link**, a Plaid-hosted UI. User banking credentials never pass through or are stored on Crumbs Money servers.
- Crumbs Money receives only an `access_token` (opaque string) and uses it for read-only API calls.

- Plaid is SOC 2 Type II certified and undergoes regular third-party security audits.
- We request only the minimum Plaid products needed: **Transactions** and **Investments**.

6.2 Firebase

- Firebase Authentication handles all user identity management.
- Firebase is SOC 2 and ISO 27001 certified as part of Google Cloud.
- Firebase Admin SDK is used server-side to verify ID tokens; the service account key is stored securely and never exposed to the client.

6.3 Railway

- Application hosting and database are on Railway, which provides TLS encryption, network isolation, and encrypted storage.
- Environment variables (secrets) are managed through Railway's dashboard and are not accessible in application logs.

---

7. Network security

- All client-to-server communication is over **HTTPS/TLS**.
- The backend enforces **CORS** restrictions, only accepting requests from the configured frontend origin.
- API endpoints require a valid Firebase ID token in the `Authorization` header; unauthenticated requests receive a 401 response.
- No public endpoints expose user data; the only unauthenticated endpoint is `/health` (returns `{ ok: true }`).

---

8. Secrets management

- **No secrets are committed to source control.** The `.gitignore` file explicitly excludes:
  - `server/.env`
  - `server/firebase-service-account.json`
  - `**/firebase-service-account*.json`
- Production secrets (Plaid keys, database URL, Firebase credentials) are stored as environment variables on Railway.
- Plaid API keys and Firebase service account keys are rotated if a compromise is suspected.

---

9. Logging & monitoring

- Server-side errors are logged to stdout (captured by Railway's logging infrastructure).
- Plaid API errors (including item-level errors like `ITEM_LOGIN_REQUIRED`) are logged with the item ID but **without** access tokens or user financial data.
- No sensitive data (access tokens, balances, transaction details) is included in application logs.
- Railway provides infrastructure monitoring and alerting for uptime and resource usage.

---

10. Incident response

10.1 Identification

- Monitor application logs and Plaid webhook errors for anomalies.
- Users can report issues through the application interface.

10.2 Response procedure

1. **Contain** — Immediately revoke compromised credentials (rotate Plaid keys, database passwords, Firebase service account).
2. **Assess** — Determine scope of exposure (which users, which data, what time window).
3. **Remediate** — Patch the vulnerability, deploy fix, revoke affected Plaid access tokens via `/item/remove`.
4. **Notify** — Inform affected users within 72 hours. Notify Plaid's security team if access tokens were compromised.
5. **Document** — Record the incident, root cause, remediation steps, and preventive measures.

10.3 Plaid-specific

- If Plaid access tokens are suspected to be compromised, call `/item/remove` for all affected items and prompt users to re-link.
- Contact Plaid's security team at security@plaid.com.

---

11. Risk assessment

Risk	Likelihood	Impact	Mitigation
Database breach exposing access tokens	Low	High	Encrypted at rest (Railway), tokens scoped to read-only, revocable via Plaid API
Firebase service account key leak	Low	High	Excluded from source control, stored only on production server, rotatable
Plaid API key compromise	Low	Medium	Stored only in environment variables, rotatable from Plaid dashboard
Cross-user data access	Very Low	High	All queries scoped by verified Firebase UID from auth token, no client-supplied user ID
Man-in-the-middle attack	Very Low	High	All communication over TLS, CORS restrictions enforced
Unauthorized API access	Low	Medium	Every endpoint requires valid Firebase ID token, 401 on failure

---

12. Secure development practices

- Dependencies are managed via `package.json` with specific version ranges.
- No user input is used in raw SQL queries; all database queries use parameterized statements (`$1` , `$2` , etc.) to prevent SQL injection.
- The frontend does not store access tokens or sensitive data in localStorage or cookies.
- Environment-specific configuration (API URLs, Plaid environment) is managed through environment variables, not hardcoded.

---

13. Compliance & review

- This policy is reviewed annually and updated after any security incident or significant architecture change.

- All team members with production access are expected to be familiar with this policy.
- Changes to security-relevant infrastructure (database, authentication, third-party integrations) require review against this policy before deployment.

Last updated: February 2026