Homericly — Information Security Policy

Live online: https://homericly.com/security — always the canonical version.

Owner: Founder / Engineering Lead — roofing@homericly.com **Monitored security channel:** security@homericly.com (group address, always watched) **Last reviewed:** 2026-06-05 **Review cadence:** Quarterly, or on any incident.

This policy describes how Homericly identifies, mitigates, and monitors information security risks across the platform. It is the operational record of the practices already implemented in this codebase, mapped to the controls a third-party reviewer (Plaid SPQ, future SOC 2 readiness) would expect to see.

1. Scope

Applies to all Homericly systems that process customer or contractor data: the Next.js application (this repo), the OAuth/account-state broker (oauth.homericly.com), the render service (render.homericly.com), Supabase Postgres, AWS SES, GitHub repositories, and Vercel deployments.

2. Governance

- **Single accountable owner.** The founder (roofing@homericly.com) is the

named security contact and decision-maker. The monitored group alias `security@homericly.com` routes to the same inbox and is the address auditors and reporters should use.

- **Policy set.** This document plus its siblings — Access Control,

Data Retention & Deletion. All under `docs/policies/`.

- **Review.** Each policy is re-read every quarter and on any incident.

Material changes are recorded in git history.

3. Asset and data inventory

- **Production data store:** Supabase Postgres (single region, EBS-encrypted).
- **File storage:** Vercel Blob (encrypted at rest).
- **Secrets:** Vercel environment variables (project-scoped, encrypted at

rest) and an `.env.local` excluded from git.

- **Code:** GitHub (`Homericly-Roofing/roofing-prototype`).
- **Third-party processors:** Stripe (payments), Plaid (bank data),

AWS SES (email), Telnyx (SMS), Composio (tool broker), Anthropic + Groq + OpenAI (AI inference).

4. Risk identification and mitigation

- **Authentication:** Next-Auth + Better-Auth with Google SSO for staff;

email magic-link or Google SSO for portal users. MFA enforced on all admin accounts (Vercel, Supabase, GitHub, AWS, Stripe, Plaid).

- **Authorization:** RBAC matrix at `lib/auth/roles.ts`; every server

action calls `requireOps(<capability>)` or `requireStaff(<capability>)`.

- **Tenant isolation:** Postgres Row-Level Security keyed on

`current_setting('app.current_org')`. Cross-tenant reads / writes are not architecturally possible from a request-scoped DB session.

- **Transport encryption:** TLS 1.2+ on every edge (Vercel, Supabase

pooler, SES STARTTLS). HTTP requests redirect to HTTPS.

- **At-rest encryption:** AES-256 on Supabase (RDS+EBS), Vercel Blob,

GitHub. No production data is written to local disk.

- **Secrets management:** All keys in Vercel env (production) or

`.env.local` (development). `.env.local` is in `.gitignore`. No secrets are ever logged.

- **Audit trail:** Every privileged action calls `record()` from

`lib/audit/index.ts`; appended to an immutable `audit_log` table.

5. Operations security

- **Patch management:** GitHub Dependabot (see `.github/dependabot.yml`)

opens PRs for npm + GitHub Actions updates weekly. SLA: critical and high-severity advisories patched within 7 calendar days; medium within 30 days. End-of-life runtimes are upgraded on the LTS schedule.

- **Vulnerability scanning:** Dependabot security alerts run continuously

on the production codebase. `npm audit` is part of the build pipeline.

- **Deploy review:** Changes ship through GitHub → Vercel production via

the linked deploy hook. Production deploys are visible to all team members in the Vercel dashboard.

- **Backups:** Supabase point-in-time recovery (24h window standard,

longer on Pro). Restoration drills are run on schema-changing batches.

6. Vendor / third-party management

- Each third party is enabled only where its capability is required and

reviewed annually for continued necessity.

- Each integration has a documented rollback path (see

`docs/build/INTEGRATION_ACTIVATION.md`).

- API keys are scoped to the least-privileged role offered by the vendor.

7. Incident response

- **Detection:** Sentry (when configured) for runtime errors; Vercel

Function logs; Supabase audit log.

- **Communication:** Email to `security@homericly.com` (monitored group

address) or `roofing@homericly.com` (owner) is the always-watched channel. For active incidents the founder is on-call.

- **Containment:** Provider-side credentials can be rotated within

minutes (Vercel env rotate + redeploy). Database-level revocation via Supabase pause/keys.

- **Postmortem:** Every customer-affecting incident is recorded in a

postmortem committed under `docs/incidents/`. No-blame, root-cause, what-changed.

8. Data privacy

Tracked under the separate Data Retention & Deletion Policy and the publicly published Privacy Policy at homericly.com/privacy-policy.

9. Compliance posture

- **Plaid:** This policy maps to the Plaid Security Practices

Questionnaire (SPQ).

- **GDPR / CCPA:** Right-to-deletion is operationalized in the Data

Retention & Deletion Policy.

- **SOC 2:** Not certified. The control set documented here is the

baseline a future SOC 2 audit would build on.

10. Acknowledgement

Anyone with production access has read this document and agrees to follow it. Changes to the team trigger a re-acknowledgement.