

APPLICATION SECURITY REVIEW

Aquarius Pro Security Assessment

Source code, configuration & dependency review

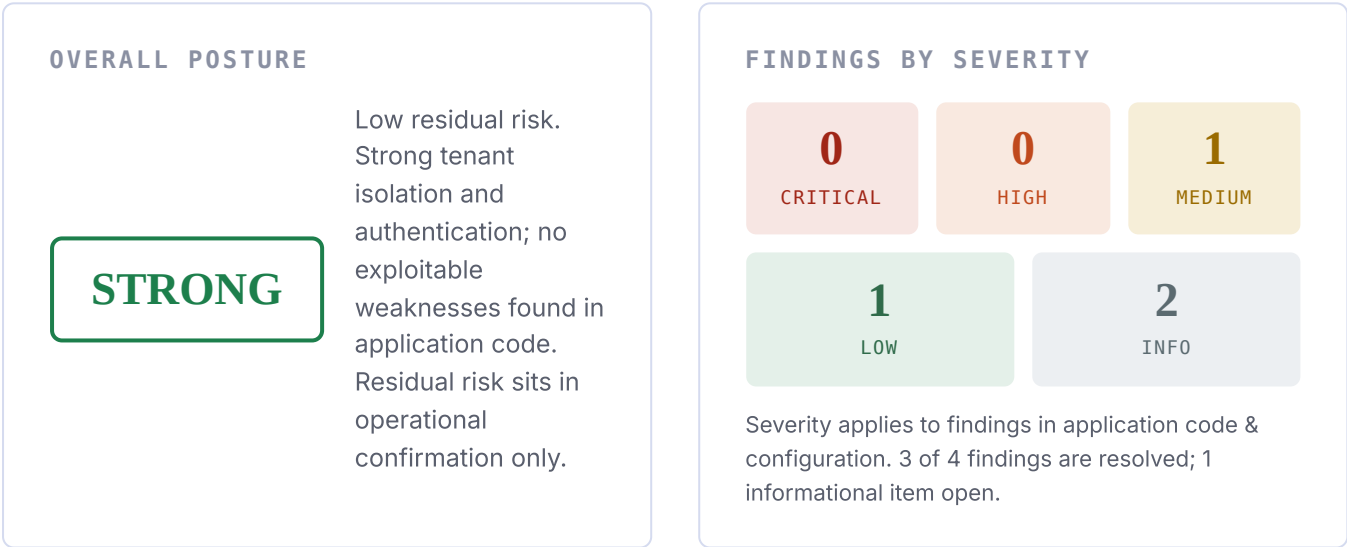
app.acss.co.uk

CONFIDENTIAL — INTERNAL

01 Executive summary

This report presents the results of an application security review of Aquarius Pro (app.acss.co.uk), the cloud-hosted SaaS platform ACSS provides to the fish and seafood processing industry. It was carried out as part of ACSS's monthly internal security screening of the platform. The review covered the application source code, its configuration and hosting, and its third-party dependencies. It is a code-and-configuration assessment, not an independent penetration test.

The application demonstrates mature, deliberate security engineering. The review identified **no injection, authentication or access-control weaknesses in the application code**. The single substantive finding — a backlog of known-vulnerable third-party dependencies — has since been remediated in full, and preventive controls added. One low-priority confirmation item remains open (F4, MFA enforcement).



Remediation status. All 27 dependency issues flagged during the review (1 critical, 10 high, remainder moderate/low) have been updated to fixed versions, a build-time vulnerability gate now blocks recurrence, and automated data-separation tests guard tenant isolation on every build. Operational controls — backups, encryption, data residency, secrets and incident response — have been confirmed.

02 Scope & methodology

In scope: the Aquarius Pro web application (React front end and Node.js/Express API), the PostgreSQL data layer and its multi-tenant isolation model, and the authentication, authorisation, input-handling, secrets, transport-security and dependency posture.

Out of scope: live penetration testing / active exploitation; customer-side devices, networks and packing-station hardware; and the physical/platform security of Google Cloud, covered by Google's own certifications (ISO 27001, SOC 2).

Approach:

- Architecture & hosting review — deployment model and platform controls.
- Source-code security review against the OWASP Top 10 and ASVS themes: authentication, session management, access control and tenant isolation, injection, input handling, data protection, secrets and configuration.
- Configuration review — transport security, HTTP security headers, CORS and CSP.
- Dependency & known-vulnerability check (`npm audit`, production dependencies of both roots), followed by remediation and re-scan.

Method note. A manual and tool-assisted review of the code and configuration. It reasons about the code as written; it does not prove runtime behaviour and is not a substitute for dynamic testing against the running system.

03 Findings register

No Critical or High findings were identified in the application code itself. F1 concerned third-party library versions and has been remediated.

F1 Known-vulnerable dependencies

MEDIUM

RESOLVED

OBSERVATION

At review time the third-party dependencies carried 27 known-vulnerable versions (1 critical, 10 high, the remainder moderate/low). Most were transitive denial-of-service issues. Two direct, in-use libraries stood out: `nodemailer` (CRLF header injection — the application sends invoices and notifications by email) and `sharp/libheif` (image processing for logos, product images and complaint photos).

IMPACT

Predominantly availability (crafted input able to slow or crash a process). The two direct libraries raised the possibility of header injection in outbound mail and image-parser flaws on upload.

REMEDIATION — COMPLETED

All flagged components updated to fixed versions — `nodemailer`, `sharp`, the live-updates library (`ws`), `react-router`, the web-server and cloud libraries, and the content sanitiser. Zero known vulnerabilities remain. A CI vulnerability gate now blocks any build carrying a high or critical component from release.

F2 `react-router` CSRF advisory (frontend)

LOW

RESOLVED

OBSERVATION

The `react-router` advisory is rated high by npm, but it applies to React Server Components mode. Aquarius Pro uses `react-router-dom` as a client-side SPA router, so real-world exposure was limited.

REMEDIATION — COMPLETED

Updated to the fixed version ($\geq 7.18.2$) as part of the dependency sweep.

F3 No dependency-audit gate in CI

INFORMATIONAL

RESOLVED

OBSERVATION

Automated dependency-update PRs (Dependabot) were configured, but nothing blocked a merge that reintroduced a known-high vulnerability — which is how the F1 backlog accrued.

REMEDIATION — COMPLETED

A build-time vulnerable-software gate was added: a build containing a component with a known high or critical issue cannot be released.

F4 Confirm multi-factor authentication policy

INFORMATIONAL

OPEN

OBSERVATION

MFA is available through Firebase Authentication, but enforcement is a console-level setting not visible in the codebase. Customers increasingly ask whether MFA is enforced for privileged accounts.

RECOMMENDATION — OPEN

Confirm (and, if not already, enforce) MFA for staff and admin accounts, and state the position for customer users. This is the only outstanding item in this report.

04 Controls assessment

Result of the review and the remediation that followed. One item (MFA enforcement) remains for confirmation.

CONTROL DOMAIN	ASSESSMENT	STATUS
Transport security	HTTPS throughout; HSTS preload (2-year, includeSubDomains); database accepts encrypted connections only.	● Implemented
Authentication	Firebase ID-token verification; bcrypt (cost 12); constant-time compare with dummy-hash to prevent user enumeration; fresh-auth window for sensitive actions.	● Implemented
Multi-factor auth	Available via Firebase; enforcement for staff/admin to be confirmed.	● Confirm
Session management	Short-lived Firebase tokens; 8-hour idle logout; server-side session-disconnect check; re-auth on stale sessions.	● Implemented
Access control & tenant isolation	PostgreSQL Row-Level Security in FORCE mode; tenant pinned per query/transaction; boot refuses to serve if the DB role can bypass RLS. Now also guarded by automated data-separation tests on every build.	● Strong
Injection & input handling	Parameterised queries throughout; SQL identifiers from server-side whitelists, never request keys; per-field validation; per-route body-size caps.	● Implemented
Data protection	Encrypted in transit (TLS) and at rest (Google-managed keys) across databases and file storage; least-privilege DB role.	● Confirmed
Secrets management	No secrets in source; service credentials held in Google Secret Manager.	● Confirmed
HTTP security headers	Helmet with tight CSP (default-src 'self', object-src 'none', frame-ancestors 'none'), CSP violation reporting, x-powered-by disabled.	● Strong
Dependencies	All 27 flagged issues on fixed versions; zero known vulnerabilities remaining; build-time scan blocks new high/critical issues.	● Remediated
Logging & monitoring	Structured logging; request/audit-log middleware; row-level change-log trigger; error/observability capture; CSP + client-error ingest.	● Implemented
Backups & recovery	Automated daily backups retained 7 days, plus point-in-time recovery across the window; deletion protection on.	● Confirmed
Infrastructure & IAM	Least-privilege DB role enforced at boot; Cloud Run; London region; quarterly access review of partner keys, device tokens and admin access (first completed 25 Sep 2026).	● Confirmed

CONTROL DOMAIN	ASSESSMENT	STATUS
Secure development	Pre-commit hooks; zero-warning lint; type-check; unit + isolated test suites; CI build/lint/test + vulnerability gate before deploy.	● Strong
Incident response	Written process covering detection, containment, recovery and notification, aligned to UK GDPR timelines.	● Confirmed
Data rights / GDPR	Hard-delete tenant purge; data-protection pack and template DPA available on request.	● Confirmed

05 System & hosting overview

Application	React single-page front end (Vite) and a Node.js/Express REST API — versioned /api/v1 surface plus a separate partner API.
Data store	PostgreSQL (Google Cloud SQL), via a least-privilege role verified at boot to be non-superuser and non-BYPASSRLS.
Hosting	Google Cloud — Cloud Run behind Firebase Hosting; London region (europe-west2); backups retained in the EU.
Authentication	Firebase Authentication (Google and email/password).
Multi-tenancy	One application and database shared across ~16 tenants, isolated per-row by tenant_id under Row-Level Security.
Sub-processors	Google Cloud (hosting/DB), Firebase (auth), and — by tenant configuration — Anthropic (AI), PostHog (analytics) and email/SFTP integrations. Confirm the full list for the data-processing register.

06 Additional hardening completed

Alongside the dependency remediation, two data-handling paths were tightened:

- **Call recordings** (phoned-in orders) now play through short-lived signed links, valid for 15 minutes and scoped to the owning tenant; recording URLs are no longer written to server logs.
- **Support-ticket attachments** (screenshots and documents) are served through per-viewer signed links, restricted to that tenant's own tickets.

07 Appendix

A. Document control

VERSION	DATE	STATUS	NOTES
0.9	25 Sep 2026	Draft	Code review complete; operational items pending confirmation.
1.0	25 Sep 2026	Final	Findings remediated; operational controls confirmed. F4 (MFA) open.

B. Severity scale

Critical / High	Directly exploitable or high-impact; remediate first.
Medium	Meaningful risk requiring remediation on a planned basis.
Low / Informational	Limited exposure or good-practice hardening.

C. Standards referenced

OWASP Top 10	Most critical web application security risks.
OWASP ASVS	Application Security Verification Standard.
CVSS	Common Vulnerability Scoring System, used for severity ratings.