

Security Assessment and Compliance Readiness Report

Multi-tenant SaaS platform with voice and chat AI features

System	Multi-tenant web application (Next.js), PostgreSQL with row-level security, voice and chat AI agents, telephony, payments and CRM integrations
Assessment window	October 5 to 6, 2026
Assessor	Wardkey (Meridal Group LLC). Client reports name the lead tester with certification and the engineer who wrote each fix; removed from this sample.
Standards	OWASP ASVS 4.0 Level 2, OWASP Top 10 for LLM Applications 2025, SOC 2 Trust Services Criteria, ISO/IEC 27001:2022 Annex A
Version	1.0 (redacted sample)

This sample keeps the structure, method and four representative findings of a real engagement. Names, identifiers, phone numbers, internal paths and infrastructure details have been removed or generalized. No secret values or customer data appear in this document.

1. Executive summary

What was assessed. The production service: the multi-tenant web application (about 140 API route files), the database (54 tables, 21 functions, 6 storage buckets), the voice AI worker and its tools, telephony configuration, OAuth integrations with 13 providers, third-party AI vendors, and the code delivery pipeline.

How. Code review of every route and database policy, followed by live testing against production with throwaway test organizations created for the assessment and deleted afterwards. Each finding below was reproduced with a real request before being reported.

54 / 54

tables where tenant isolation held in a live two-organization test

0

critical findings

105

findings in total, each with evidence, exploit scenario and fix

22 → 0

known-vulnerable dependency advisories after remediation

Headline results.

- **Tenant isolation held.** With a second organization's session and with the anonymous key, no row in any of the 54 tables could be read, inserted, updated or deleted. Against the application routes, 19 requests carrying another organization's ids and 46 anonymous requests were all refused.
- **The serious issues sat around the policies, not in them:** business logic that let an outsider generate billable events, an AI tool that could be steered into placing calls to numbers the caller chose, and third-party OAuth credentials stored in plain text.
- **Both externally exploitable High findings were fixed and verified within the assessment window.** All 49 stored integration credentials were encrypted at rest with AES-256-GCM.

Severity	Findings
HIGH	13
MEDIUM	39
LOW	39
INFO	14

2. Scope and method

Workstream	Coverage
Tenant routes	68 handlers, 32 AI assistant actions, 12 builder tools; every request replayed as another organization and as an anonymous user
Database	54 tables, 21 functions, 6 storage buckets, auth settings; live two-organization test of every table and operation

Machine and integration endpoints	103 handlers: webhooks (signature and replay), scheduled jobs, OAuth flows for 13 providers, telephony configuration
Secrets, supply chain, change	Secret scan of the full git history, production bundle scan, dependency audit, repository and deployment settings
Voice, telephony and AI	AI tool permissions and tenant scoping, prompt injection (direct and through stored content), toll fraud, recording consent, personal data in logs, AI vendor data terms
Web and data	Security headers, TLS, cookies, CSP, XSS, CSRF, file uploads, error leakage, encryption at rest, retention and deletion

Rules of engagement. Production configuration was read only. Dynamic tests used dedicated test organizations only. No real calls, texts, purchases or payments were made and no load tests were run. Database changes were delivered as reviewed SQL for the system owner to apply. Secret values were compared by hash and never written into documents.

Severity	Meaning
CRITICAL	Exploitable now; cross-tenant data or money at stake
HIGH	Exploitable with a precondition, or a control a SOC 2 auditor would fail
MEDIUM LOW INFO	Descending impact and likelihood

3. Selected findings

Four of the 105 findings, shown as they appear in the full report. Evidence has been generalized; in the full report each finding cites the exact file and line, the request and the response.

HIGH F-01 Any script can bill a customer for chat "resolutions"

Standards: SOC 2 CC6.1, CC6.6, PI1.2; ISO 27001 A.8.3, A.8.26; OWASP ASVS V11.1 (business logic anti-automation)

Evidence: The website chat widget's feedback endpoint required no credential. Its only checks were ids that the caller itself created through the public message endpoint, and the `Origin` header, which a browser cannot forge but any HTTP client can. A positive vote created a billable resolution that was reported to the payment provider as overage.

```
POST /api/chat/widgets/<widget>/feedback  Origin: https://attacker.example  -> 403
POST /api/chat/widgets/<widget>/feedback  Origin: https://customer.example  -> 200
{"resolved":true}
billable resolutions for the customer: +1 (no browser involved)
```

Exploit scenario: A competitor reads the widget id from the customer's public embed code and loops "open conversation, vote helpful". Within the existing rate limits that is about 3,000 billable events a day per widget, roughly \$1,350 a day of overage, while the same loop exhausts the widget's message limit so real visitors are refused. The review also found that every idle chat was being billed.

Fix: Bill only on real engagement (two visitor messages, a model-written reply, 30 seconds), once per visitor and network per widget a day, with a daily cap per widget and an alert on unusual spikes.

Status: **Fixed and verified in production**

HIGH F-02 Callers can make the voice AI place calls to any number they dictate

Standards: OWASP LLM06 (excessive agency), LLM01 (prompt injection); SOC 2 CC6.8, CC9.2; ISO 27001 A.5.14, A.8.12; TCPA exposure for artificial-voice calls without consent

Evidence: The voice agent's "schedule a callback" tool accepted any number the caller spoke ("a number the caller dictated wins"), and a scheduled job dialed every due callback with an AI greeting in the business's name. Nothing checked that the person being called was the person who called. In one live test call, three different dictated numbers were all accepted and queued for AI callbacks. A control test in the same run showed that a comparable tool (do-not-call) already enforced "only the line this call comes from", so the rule existed but had not been applied here.

Exploit scenario: An attacker calls a customer's line, or the public demo line, and asks to be "called back" on a list of other people's numbers. Each victim then receives an AI call in the business's name from its real number. The cost, the caller-ID reputation and the legal liability (up to \$500 to \$1,500 per unsolicited artificial-voice call) fall on the platform and its customer.

Fix: AI callbacks only to the line the call came from; a different dictated number becomes a task for a person. One AI callback per call, a per-organization daily ceiling, the tool removed from demo agents, and calling hours based on the callee's area code rather than the agent's time zone. The scenario was added to the regression suite.

Status: Fixed and verified in production

HIGH F-03 Third-party integration credentials stored in plain text and readable from the browser session

Standards: SOC 2 CC6.1, CC6.7, C1.1; ISO 27001 A.8.24, A.5.33; OWASP ASVS 6.4.1, 8.3.4

Evidence: OAuth refresh tokens, access tokens and API keys for calendar, CRM and messaging integrations were stored unencrypted. Row-level security correctly limited each row to its owner, but table-level read grants meant the owner's own browser session could read the long-lived tokens, and one settings page already loaded a refresh token into the browser just to display "connected".

encrypted values found: calendar 0/9, CRM and productivity tools 0/40, messaging 0/3
plaintext credential values across 14 tables: 52

Exploit scenario: Any cross-site scripting bug, a malicious browser extension or a shared computer would hand over a customer's calendar, CRM and messaging tokens, which keep working after the session ends. Anyone with database read access would get every customer's credentials in clear text.

Fix: Every credential sealed with AES-256-GCM on write and opened only on the server. The browser now reads connection status from a server route, never the token. Column-level grants remove token columns from browser sessions. Disconnect revokes the token at the provider, and PKCE and user-bound OAuth state were added.

Status: Fixed; all 49 stored values re-encrypted and verified

MEDIUM F-04 Account lookup treats % and _ as wildcards, so customer emails can be enumerated

Standards: SOC 2 CC6.1, P4.0; ISO 27001 A.5.34, A.8.3; OWASP ASVS V2.2.4, V5.3.4

Evidence: The public "does this account exist" endpoint matched the submitted email with a case-insensitive pattern operator instead of an exact match, so SQL wildcard characters in the input were honored.

```
{"email":"test-a-%@example-domain"}      -> {"exists":true}
{"email":"test-a-zz-nobody%@example-domain"} -> {"exists":false}
```

Exploit scenario: An unauthenticated attacker probes a@gmail.com , b@gmail.com and so on, then walks each hit one character at a time to recover customer email addresses, or tests %@competitor.com to learn whether a company is a customer. The per-address rate limit slows this down; proxy pools make it practical.

Fix: Exact, lower-cased match, plus a regression test that % and _ are treated literally.

Status: Fixed and verified in production

4. Backup and restore drill

The production database was dumped and restored end to end into an isolated environment, then compared object by object with the source.

Measure	Result
Tables compared	58, including authentication and storage metadata
Row count mismatches	0 of 11,562 rows
Schema objects restored	policies 65/65, functions 327/327, triggers 14/14, foreign keys 74/74, row-level security 54/54
Row-level security after restore	still scoped each organization to its own rows; the anonymous key read none
File storage	585 files (137.3 MB) copied and size-checked, 0 failures
Data restore time	about 10 minutes; full recovery estimated at 2 to 4 hours including keys and redeploy

5. Compliance readiness deliverables

Alongside the technical findings the engagement produced the documents an auditor asks for first:

- SOC 2 control matrix (38 criteria) and ISO 27001 Statement of Applicability (93 controls), each with current status and the evidence behind it
- Ten policies: information security, access control, change management, incident response, backup and disaster recovery, vendor management, data classification and retention, acceptable use, secure

development, risk management

- Risk register (28 risks), vendor register with each vendor's SOC 2, ISO, DPA and AI data terms, and data flow maps for ten flows
- A tabletop incident exercise (a leaked database key) with 11 follow-up actions
- A 90-day roadmap to SOC 2 Type I readiness

Lead tester name and certification

Signed on behalf of Wardkey (Meridal Group LLC)

Shown in full on client reports