

OUR OWN PRODUCTS

QuietWatcher

Uptime and security monitoring that runs on the user's own machine. The engine and the database ship inside the app, so the target, the response and the credential used to reach it never leave the device.

6

Platforms from one codebase

52

Tools in each agent server

1,706

Passing app tests at the last run

0

Secrets held in the database



THE PROBLEM WITH THE CATEGORY

Monitoring is sold on the assumption that you will hand over your systems list. Every hostname, every endpoint, every API key that reaches a private service, held on a vendor's servers so the vendor can run the check for you.

For a firm under an NDA, or one watching a client's infrastructure, that is a disclosure nobody asked for. QuietWatcher does the watching from the user's machine instead. A person can install it, add a monitor and get an alert with no account, no network round trip to us and no record of the target on our side. Cloud sync, team sharing and server-run checks exist for people who want them, and they are opt-in one monitor at a time.

It is our own product, about two and a half years of work, and nobody was paying us to cut corners on it.

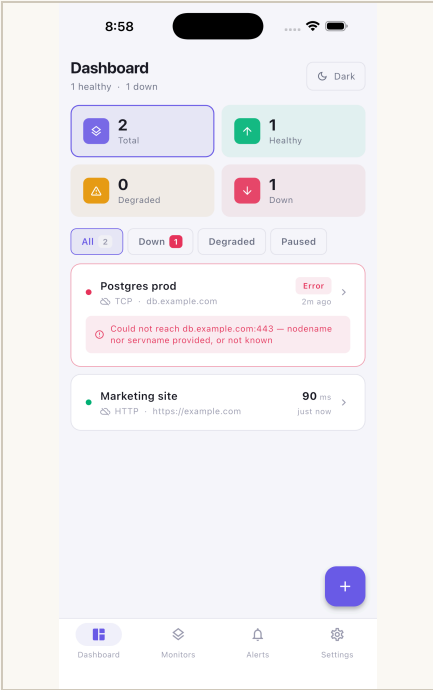
WHAT IT WATCHES

- **Six monitor types on a schedule.** HTTP, API, TCP, UDP, ping and process.
- **API monitors that assert something.** Keyword and JSON path assertions, expected status codes, custom headers, and a TLS verification flag that defaults to on.
- **Failures roll into alerts, alerts into incidents,** and an incident closes itself when the target recovers. Alert paths are local notification, email, SMS, webhook and Slack, with rules set per monitor.
- **A scan workbench** that discovers hosts on the local network, draws them on a map you can rearrange, and records per-host port scans with a timestamp and a detail report. The map exports as a PNG from the same painter that drew it on screen, so the file matches what you saw.
- **One letter grade for the fleet.** The Standing Watch is marked provisional until enough history backs it, and it never moves without naming the cause: B to A, api.example.com recovered. No streaks and no leaderboards.

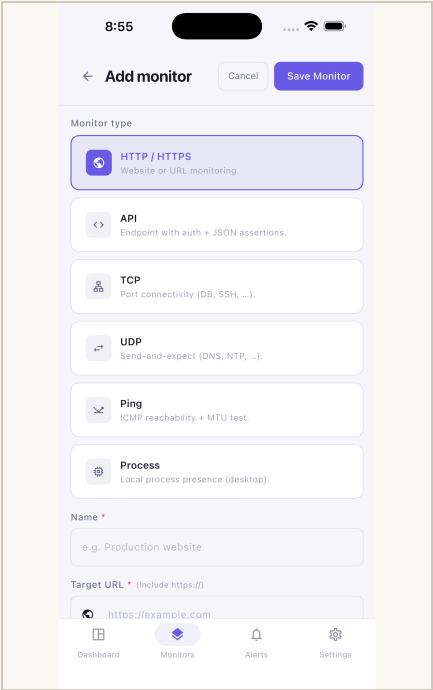
THE SECOND KIND OF WATCHING

The Security Check is one request to a site you already monitor, analyzed on your machine, looking for 20 issues the shipping engine actually runs: wallet-drainer injection, card skimmers, hidden miners, malicious redirects, pharma spam, certificate expiry, missing HSTS and CSP, clickjacking and MIME-sniffing gaps, insecure cookies, CORS misconfiguration, mixed content, insecure password forms and missing subresource integrity.

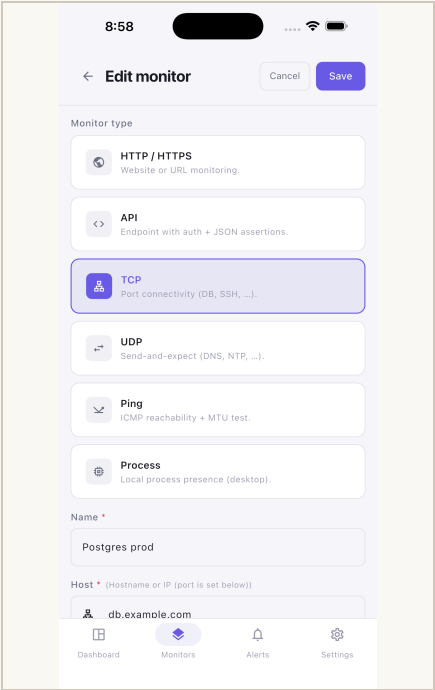
A test reads the analyzer source and fails the build if the shipping list and the engine ever disagree, because counting the bundled catalog instead would over-claim by a factor nobody would notice. Beside it, a content integrity scanner learns the external script origins a page normally loads plus a stable hash, then flags a new origin or a changed body against 16 bundled signatures.



DASHBOARD, ONE DOWN



ADDING A MONITOR



EDITING A TCP CHECK

HOW IT IS BUILT

One Flutter codebase compiled to macOS, Windows, Linux, iOS and Android, plus a Flutter web console. Hand-written Riverpod providers, go_router, an embedded ObjectBox store, and a three-layer split enforced in review: pure-Dart domain entities, a persistence layer with its own entities, and mappers between them. A domain entity never imports the database.

The control plane is one Django 5.1 project with DRF and PostgreSQL serving both the customer API and the operator surfaces. Identity is a separate service. The agent tool servers are written twice, once in Python and once in Node, kept name-for-name identical, and the drag-to-install bundle is gated at pack time against the server's own tool list.



WHERE THE SECRETS ARE NOT

A monitor row holds an opaque reference. The value lives in the platform keychain, is read only at check execution, and is deleted when the monitor or the credential is. Not storing a secret beats encrypting one, and that was the choice made every time the two competed.

Response snippets are scrubbed where they are produced, and keyword assertions run against the unscrubbed body so a redaction cannot cause a false failure. Biometric or device credential app lock is available on four platforms. Account deletion is in the app: credentials and response snippets purge at once, the rest is retained 365 days and then hard-deleted.

State today, stated the way we would want it stated to us. Version 1.14.0 is the last public release, tagged 18 July 2026. 1.15.0 is in TestFlight and on the Play internal track. Commerce is off and Stripe is in sandbox. No customers and no public store listing yet.

WHAT AN ACCESSIBILITY AUDIT FOUND, AND WHAT HAPPENED NEXT

A text color failing contrast at roughly 176 call sites, icon buttons under 44 points on three desktops, and fifteen layouts breaking at 200 percent text scale. All of it is now enforced by five test suites, alongside 184 Dart test files, 1,706 passing app tests and 1,178 on the control plane.

WHAT WE CAN START ON

A cross-platform product from one codebase across desktop, mobile and the browser. A monitoring or security capability inside something you already sell. Or a tool server so an AI agent can drive software you already own.

512-900-4871 · div42.com · roblesindustries.com