Security Overview

How we encrypt Notion tokens, isolate customer data, and harden the infrastructure that supports Notion Scan.

Last updated May 29, 2026

Overview

Notion Scan is a workspace intelligence platform operated by Arbr Labs LLC, doing business as Notion State. It connects to a customer's Notion workspace via an authorized internal integration, reads structural metadata about pages, databases, and users, and produces consultant-grade analysis of workspace health and adoption patterns.

This document explains what we read, what we store, how we protect it, and how we keep one customer's data isolated from every other customer's.

Security at a glance

Control	What it means
Token encryption	Notion integration tokens are encrypted at rest with AES-256-GCM and never logged in plaintext.
Customer isolation	Customer data is logically isolated using database-level access policies, role-based authorization, and scoped backend workflows.
Metadata only	We store titles and structural metadata. We never store page body content, comments, or non-title database entry values.
No ML training	We do not use customer workspace data to train machine learning models.
HTTPS-only	All traffic is TLS-protected and HSTS is enforced.
Hosted on SOC 2 infrastructure	Vercel (application), Supabase (database and authentication), and Railway (background workers) are SOC 2 Type II audited.

These are provider attestations for the infrastructure platforms we use. Notion State has not completed its own SOC 2 Type II or ISO 27001 certification.

Data collection

What we read and store

We use the Notion API to enumerate the structural elements of an authorized workspace. The complete set of data we read and persist:

Category	What we capture
Pages	Page ID, URL, parent reference, title, type, creation and last-edited timestamps, creator and last-editor user IDs.
Databases	Database ID, title, schema (property definitions and types), parent reference, last-edited timestamp.
Users	User ID, display name, user type (`person` or `bot`).
Relations	Page-to-page and database-to-database structural relationships derived from page parentage and schema properties.
Property statistics	For each database property, aggregate fill rates (what percentage of entries have a value for this property).
Activity metadata	Edit counts and contributor counts per database, derived from last-edited fields on entries.

What we never store

- **Page content and body text.** The Notion blocks endpoint is used for navigation and tree discovery (walking from a page to its children to find more pages and databases). We inspect block type, child page/database references, and whether a container block has children. Block text is not extracted, returned, logged, or stored.
- **Block-level data.** We do not store block records or block content. Only container relationships (page contains child page, page contains database) are traversed for discovery.
- **Non-title database entry values.** Notion represents database rows as pages, so we store each row's page title as page metadata. When computing per-property fill rates, non-title entry property values pass through application memory transiently and are immediately discarded. Non-title values are never stored, logged, or returned.
- **Comments.** We do not call the Notion comments API. Comment content, authors, and timestamps are never read or stored.
- **Notion workspace user email addresses.** The Notion API may return user email addresses depending on integration capabilities; we explicitly do not extract or store workspace user email addresses from Notion. Account and onboarding email

addresses that users provide directly to Notion Scan are handled as described in the Privacy Policy.

- **Files and attachments.** We do not download, open, or store file contents. For database file properties, we may count whether the property is filled as part of aggregate property utilization; file names, URLs, and attachments are not stored.

Data scope answers

For procurement questionnaires that ask about specific data scope:

- **Pages.** Metadata only: ID, URL, title, parent, type, timestamps, creator and last-editor user IDs. No content.
- **Databases.** Metadata and schema only: ID, title, property definitions and types, parent, timestamps. No non-title entry values stored.
- **Comments.** Not accessed. We do not call the Notion comments API.
- **Users.** Three fields per Notion workspace user: user ID, display name, type. No Notion workspace user emails, no profile photos, no workspace roles.
- **Guests.** Guest users that appear in the workspace user list are captured under the same three-field schema as members. We do not enumerate, distinguish, or store guest-specific metadata. The Notion API does not expose guest-versus-member status at the user object level, so our visibility is equivalent.
- **Integrations.** We do not enumerate or identify other workspace integrations. Other installed integrations may appear as bot users in the Notion user list. We capture them under the same three-field schema as any user and do not store integration-specific metadata.

A concrete example

When a customer authorizes Notion Scan and we begin a scan, here is what actually happens:

1. We enumerate the pages, databases, and data sources the integration can see.
2. We retrieve page and database metadata, including database schema.
3. We enumerate workspace users and extract three fields per user.

4. We count database entries, store each entry page's title as page metadata, and compute aggregate fill-rate statistics from transient non-title property reads.
5. We traverse child references to discover nested pages and databases. We use only the structural references needed for discovery; block text is not extracted, logged, or stored.

Throughout this process, we are interacting with structural metadata. The actual text in your pages, non-title values in your database rows, and the conversation in your comments are not what we are looking at.

Storage and encryption

Token encryption

Notion integration tokens are encrypted at rest using AES-256-GCM authenticated encryption. Decryption keys are held only by the application runtime.

Token safety controls

- Tokens are decrypted only in process memory when the application or scan worker needs Notion API access.
- During scans, the Railway-hosted worker holds the decrypted token in memory for the scan runtime and uses it only as the Notion API authorization credential.
- Plaintext tokens are never logged, never written to disk, and never included in requests to Anthropic, Slack, or other non-hosting subprocessors.
- A workspace administrator can revoke the integration in Notion at any time; the token becomes useless to us immediately.
- On customer deletion request, we delete the encrypted token from our database per the Data Retention Policy.

Database encryption

All customer data is stored in Supabase-managed Postgres on AWS infrastructure. The database is encrypted at rest using AES-256 (managed by AWS) and TLS-protected in transit between the application and the database.

Access control

Authentication

User authentication is handled by our auth provider using session cookies and automatic session refresh. Authentication is required for every protected route; unauthenticated requests are redirected to login or returned as 401 for API endpoints.

Role-based access

User access is scoped to specific customer workspaces through role-based access records. Users can access only the workspaces they are explicitly granted. A separate `admin` role exists for cross-client visibility and is granted only to authorized Notion State personnel.

Row-Level Security

Customer data is protected through database-level tenant access policies, role-based authorization, and scoped backend workflows. User-facing access is restricted to authorized customer workspaces. Privileged backend jobs and administrative endpoints are limited to the customer context they are operating on and are reviewed for access-control correctness. Full details in the Customer Isolation Policy.

Infrastructure security

Hosting

- **Application:** Vercel serverless functions and edge delivery, SOC 2 Type II. Serverless functions use the Vercel project region configuration; edge routing and middleware may run on Vercel network locations near the requester.
- **Database and authentication:** Supabase (managed Postgres on AWS), US-based, SOC 2 Type II.
- **Background workers:** Railway-hosted, US West (California), SOC 2 Type II; isolated worker processes with no public HTTP port and outbound-only network access.

Security headers

Every HTTP response includes the following headers:

Header	Value	Purpose
`Strict-Transport-Security`	`max-age=63072000; includeSubDomains`	Enforces HTTPS for two years across the domain and all subdomains.
`X-Frame-Options`	`DENY`	Prevents framing by other origins.
`X-Content-Type-Options`	`nosniff`	Prevents MIME-type sniffing.
`Referrer-Policy`	`strict-origin-when-cross-origin`	Limits referrer leakage.
`Permissions-Policy`	`camera=(), microphone=(), geolocation=()`	Disables sensitive browser APIs.
`Content-Security-Policy`	First-party script and style sources only (including framework-required inline scripts and styles), Supabase API connections, Loom video embeds for product demos, and no framing of our content by other origins.	

Rate limiting

Notion API calls are rate-limited per workspace and use retry/backoff handling for transient errors. This protects both our service and the customer's Notion workspace from accidental overload.

Worker isolation

The scan worker is a background process. It has no inbound HTTP port, opens only the outbound connections needed for scanning, persistence, and configured operational notifications, and is deployed separately from the user-facing application.

Subprocessors

We use a small set of third-party services to deliver Notion Scan. The complete list, including purpose, data processed, and location, is documented in Subprocessors.

Notion integration

Notion Scan uses an internal integration provided by the workspace administrator. The integration's permissions determine what we can see. We do not require workspace-wide admin access for scans; we read only what the integration is authorized to read.

Capability matrix

Capability	Used	Notes
Read content (search, pages, databases)	Yes	For structural enumeration and metadata.
Read users	Yes	Three fields per Notion workspace user; no workspace user email.
Read comments	No	Comments API is not called.
Update content	No	We never modify the workspace.
Read page content (blocks endpoint)	Used for navigation only	Block text is not extracted, logged, or stored; only child page/database references and traversal metadata are used.
Read non-title database entry values	Used transiently for fill-rate stats only	Non-title values are not stored, logged, or returned.
Insert content	No	Notion Scan is read-only.

Data offboarding

When a customer requests deletion, we delete customer data on the timeframes set by the Data Retention Policy. The workspace administrator can revoke the integration at any time, which immediately invalidates future Notion API access; deletion of stored data follows a confirmed customer deletion request.

About Notion State

Notion State is a Notion-focused consultancy that builds workspace intelligence tools. Notion Scan is one of those tools. We are not a Notion Labs subsidiary; we are an independent company operating under Notion's third-party integration program.

Contact

For security questions, vulnerability reports, or to schedule a security review call: security@notionscan.com.