



SYNAPSE
SONAR

User & Administrator Guide

Operating and administering Synapse Sonar

1. Introduction

Synapse Sonar is a self-hosted, agentless **Dynamic Internal Attack Surface & Dependency Mapper**. It builds a live map of your internal network from observed traffic flows, overlays vulnerability data to highlight real-world risk, and enforces Zero-Trust segmentation by detecting architectural drift.

This guide covers day-to-day use and administration of the web application. For deploying the data collectors, see the companion **Agent Setup & Configuration Guide**.

Key capabilities

- Interactive force-directed topology map with hover detail and a deep-dive sidebar.
- Vulnerability overlay (CVE badges, CVSS) driven by the NetscanXi feed.
- Blast-radius simulation: highlight every asset reachable within two hops of a compromised node.
- Zero-Trust drift & rogue/shadow-IT detection on ingested flows.
- Multi-tenant isolation, role-based access control, optional TOTP MFA.

2. Architecture at a glance

Synapse Sonar runs as two containers orchestrated by Docker Compose:

- **Application** — the Next.js web app and API (default port 3000).
- **Database** — PostgreSQL, storing all tenant data (JSONB for dynamic asset metadata).

Network visibility is provided by **collectors** you deploy separately (host agents or a passive sensor) that push observed flows to the application's ingestion API. The application never scans the network itself.

Isolation. All data is strictly scoped to a tenant. Every query is filtered by the signed-in user's tenant; there is no cross-tenant read path.

3. First-run setup

On a fresh installation the database has no users, so any visit redirects to the **setup** screen. Create the first administrator with a username and password (an optional display name). A default organization is created automatically — you can rename it later under User Administration.

```
docker compose up -d --build      # start the stack (auto-applies migrations)
# then open http://<host>:3000    ->  redirected to /setup
```

Security. Setup is one-shot: once the first account exists, the setup endpoint refuses further use. There is no built-in default/backdoor account.

4. Signing in

Log in with your **username and password**. There is no organization field — your tenant is derived from your account. If multi-factor authentication is enabled on your account, you are prompted for a 6-digit code; accounts without MFA are never asked for one.

5. Roles & access control

Synapse Sonar has three roles. Permissions are enforced on every API route.

Capability	Read-Only Viewer	Security Analyst	Tenant Admin
View topology & assets	Yes	Yes	Yes
Run blast-radius simulation	—	Yes	Yes
Edit assets / segmentation rules	—	Yes	Yes
View integrations	Yes	Yes	Yes
Configure integrations & keys	—	—	Yes
Manage users & org settings	—	—	Yes

6. The topology map

The map is the home screen. Nodes are assets; edges are observed connections with directional flow animation whose thickness scales with traffic volume.

Interactions

- **Pan / zoom / drag** — click-drag the canvas to pan, scroll to zoom, drag a node to reposition.
- **Hover** — a lightweight tooltip shows hostname, internal IP, criticality tier and a risk badge (e.g. ■ 3 Critical CVEs). It disappears the moment you move away.
- **Click** — opens the deep-dive sidebar with full metadata, all open ports, and the vulnerability list (when NetscanXi is enabled).

Blast-radius simulation

In the deep-dive sidebar, **Simulate Blast Radius** treats the selected node as compromised and highlights every downstream asset reachable within two hops in neon violet, so you can see the lateral-movement path. Requires the Security Analyst role or higher.

Map legend

- **Cyan** edges — data flow.
- **Red** ring — node has critical CVEs.
- **Rose** — rogue / shadow-IT asset (not in known inventory).
- **Violet** — blast-radius path.

7. Integrations

Open **Integrations** (Tenant Admin). Each integration has a master on/off toggle that takes effect immediately.

NetscanXi — vulnerability feed

When enabled, the card shows the **Endpoint URL** and an **API key (Bearer token)** to configure in your NetscanXi scanner so it pushes scan results to Synapse Sonar. While enabled, vulnerability indicators appear on the map and in the sidebar; when disabled, the feed is rejected and indicators are hidden.

Synapse IronNode — SOAR alerting

When enabled, reveals a **Webhook Receiver URL** and a **Secret Signing Key**. Segmentation violations and rogue-asset detections are dispatched to that webhook as signed JSON (HMAC-SHA256). When disabled, the alerting pipeline pauses.

Security. Integration secrets are encrypted at rest with AES-256-GCM. The UI only ever displays masked values; ingest keys are stored only as a SHA-256 hash and the plaintext is shown exactly once at generation.

8. User administration

Tenant Admins manage members under **User Administration**:

- **Organization name** — edit the display name shown across the app and on invitations.
- **Add user** — enter a username and role; an invite link is generated for them to set their own password (no email service is required in the MVP — copy the link).
- **Roles & status** — change a member's role inline, or disable/enable an account.
- **Require MFA for all members** — an org-wide policy (see next section).

9. Multi-factor authentication (MFA)

MFA is optional and TOTP-based (any authenticator app).

- **Self-enrolment** — each user enables MFA on **Profile & Security**: scan the QR code, confirm a 6-digit code. From then on, sign-in requires the code.
- **Org-wide enforcement** — a Tenant Admin can turn on **Require MFA for all members**. Any member without MFA is then sent to a mandatory enrolment screen before they can use the app.

Note. Enabling the org-wide policy also applies to the admin who set it — you cannot require MFA of others while exempting yourself.

10. Security & data handling

- Passwords are hashed with bcrypt; TOTP secrets and integration secrets are encrypted (AES-256-GCM).
- Collector / ingest keys are stored as SHA-256 hashes and matched by hash.
- Tenant isolation is enforced on every query by the authenticated tenant id.
- IronNode webhook payloads are signed with HMAC-SHA256 (header X-Sonar-Signature).
- Set strong NEXTAUTH_SECRET and INTEGRATION_ENCRYPTION_KEY values in the environment.

11. Troubleshooting

Symptom	Likely cause & fix
Redirected to /setup	No users exist yet — create the first admin.
“Invalid username or password”	Check the username (not email). If upgraded from an older build, your username may be your former email address.
Asked for an MFA code unexpectedly	MFA is enabled on that account, or the org-wide policy is on.

Symptom	Likely cause & fix
No vulnerabilities on the map	Enable the NetscanXi integration; data is hidden while it is off.
No nodes appear	No collector is sending flows yet — deploy an agent or sensor.

12. API reference (summary)

Endpoint	Purpose
POST /api/ingest/flow-logs	Ingest observed flows (assets + connections).
POST /api/ingest/vulnerabilities	Ingest NetscanXi findings (CVEs).
GET /api/graph	Tenant topology for the map.
GET /api/assets/:id/blast-radius	Downstream reachable assets (2 hops).
GET/PATCH /api/integrations	Read / configure integrations.
GET/POST/PATCH /api/users	List, add, update users.

Ingestion endpoints authenticate with a Bearer ingest key; all others require an authenticated session.