USER GUIDE

Security Orchestration, Automation & Response

Zero-Trust Identity · Visual SOAR · Embedded AI Co-Pilot

Version 1.1 · Multi-Tenant · RBAC · Self-Hosted Edition

Welcome to Synapse IronNode

Synapse IronNode is a lightweight, self-hosted SOAR platform that unifies a visual automation engine, a strict Zero-Trust identity layer, an embedded Claude AI analyst, and an immutable case-management ledger. This guide walks you through every workflow, from first launch to exporting a signed incident audit report.

What's inside

1 · Getting Started

First Launch — Organization & Platform Admin

On its very first launch, Synapse IronNode detects that **no users exist** and the login screen transforms into a one-time **Initial Setup** form. No default credentials are ever shipped. This single atomic step creates your first **Organization (tenant)** and its owner account.

- Open **http://localhost:3000** in your browser.
- A "First Launch Detected" banner appears.
- Enter your **Organization name**, plus your name, email, and a strong password (12+ characters).
- Click **Initialize Synapse IronNode**. This creates the tenant and a **Platform Admin** — the MSSP/owner account that can later provision additional organizations.
- You are signed straight into the Operations Dashboard.

Setup is keyed off user existence and runs as a single transaction, so a half-completed attempt can never lock you out — it stays available until a real account exists.

Subsequent Logins

Later visits show the **Zero-Trust login portal**. If MFA is enabled on your account you are prompted for a 6-digit TOTP code. If your administrator has **required** MFA but you have not enrolled yet, login redirects you to a

one-time **MFA enrolment** screen (scan a QR code) before access is granted.

2 · Roles & Access (RBAC)

Access is governed by a SOC-aligned role hierarchy. Each tier inherits the capabilities of those below it. Permissions are enforced server-side on every request.

Role	Capabilities (cumulative)
1st Line Analyst	View dashboard & incidents; work the triage queue; approve LOW-risk remediations.
2nd Line Analyst	+ Approve HIGH-risk remediations; build and edit playbooks.
3rd Line Analyst	+ Create / manage API integrations and the credential vault.
Administrator	+ Manage users; force MFA; view the audit log.
Platform Admin	+ Provision and manage organizations across tenants (MSSP operator). The first account.

Risk-gated approvals: any 1st Line Analyst may approve low-risk actions, but approving an action the AI flagged **HIGH risk** requires a 2nd Line Analyst or above — enforced at execution time.

Step-Up MFA: privileged or destructive actions can demand fresh TOTP verification even within an active session.

3 · Multi-Tenancy

Every record belongs to an **Organization (tenant)**. Isolation is enforced in depth: the application scopes every query by tenant, AND the database independently enforces it through PostgreSQL **Row-Level Security**. One tenant can never see or affect another's data.

- Playbooks, integrations, vault secrets, incidents and audit logs are all per-tenant.
- A **Platform Admin** can provision new organizations from the **Organizations** screen, each with its own Administrator — ideal for MSSP and multi-business-unit deployments.
- Each organization has its own webhook slug and its own per-tenant rate limit (see §6).

4 · The Operations Dashboard

The dashboard is your real-time security posture at a glance. It opens with the **SOC triad** — three headline cards — followed by detailed status tiles and a live incident table.

The SOC Triad Cards

Card	Shows	Click-through
Alerts	Open & in-flight items (triggered, enriching, awaiting triage, executing).	Triage queue
Incidents	Every case tracked by the platform, across all statuses.	Incidents list
Events	Total ledger entries logged across all incidents — the raw event volume.	Incidents list

Status Tiles & Incident Table

- **Awaiting Triage**, **Total Incidents**, **Critical (≥8)**, and **Closed** tiles give a quick breakdown.

- The **Recent Incidents** table lists each case with its ID, playbook, status badge, risk score, and timestamp.
- A shortcut banner appears whenever items await triage, linking straight to the queue.

AI Co-Pilot Status Indicator

The dashboard header shows a live **AI Co-Pilot** chip — a green “Active” dot when the embedded Claude analyst is enabled and configured, or a red “Inactive” dot when it is off or has no API key. Click the chip to jump to its configuration. See §7 for how to enable it.

5 · Building Playbooks

Playbooks are visual, node-based automation workflows (2nd Line Analyst or above). Open **Playbooks** → **New Playbook** to launch the drag-and-connect canvas. Wire nodes left-to-right: **Trigger** → **Enrich** → **AI Triage** → **Human Approval** → **Execute**.

Node Types

Node	Purpose
Webhook Trigger	Entry point. Fires the playbook when an external system POSTs an event (see §6).
Enrich	Calls an outgoing API integration to add context (IP reputation, hash analysis, geo).
AI Triage	Sends the enriched payload to the Claude AI analyst. The runner suspends here.
Human Approval	Routes the incident to the triage queue for analyst sign-off.
Execute Action	Runs an approved remediation via an integration (block IP, create ticket).
Condition	Branches the flow based on a field value or threshold.

Editing the Canvas

- Click a node to select it; **shift-click** or drag a box to select several.
- **Copy** (Ctrl+D) duplicates the selection, preserving edges between copied nodes.
- **Delete** (Del / Backspace) removes the selection and any connected edges.
- Name the playbook and click **Save Playbook** — it then appears in the Playbooks list with its webhook URL.

6 · Webhooks — Incoming & Outgoing

Synapse IronNode uses webhooks in **both directions**. An **incoming** webhook lets an external system trigger a playbook; an **outgoing** integration lets a playbook call an external API to enrich an event or execute a remediation.

6.1 · Incoming Webhooks (Triggers)

When you save a playbook, the platform mints a unique, unguessable **webhook path** for it. The full trigger URL is **tenant-aware**:

```
POST http://<host>:3000/api/webhook/<tenant-slug>/<webhook-path>
```

Setting one up

- Build and **Save** a playbook whose first node is a **Webhook Trigger**.
- On the **Playbooks** list, each card shows its trigger URL with a one-click **copy** button. (The base pattern is also shown in **Settings** → **Webhook Endpoint**.)
- Configure your detection source (SIEM, EDR, firewall, monitoring tool) to POST a JSON event to that URL.
- Send a test event and confirm a new incident appears on the Dashboard.

Example trigger

```
curl -X POST \
http://localhost:3000/api/webhook/demo/a1b2c3d4-… \
-H "Content-Type: application/json" \
-d '{"src_ip":"1.2.3.4","event":"brute_force","attempts":847}'
```

Behaviour & protection

- Returns **202 Accepted** immediately and runs the playbook asynchronously — the caller never blocks.
- The **tenant-slug** scopes the trigger to your organization; the webhook-path is an unguessable UUID.
- Each organization has its own **rate limit** (requests/minute). Exceeding it returns **429** with X-RateLimit-* and Retry-After headers, so one tenant's burst can't affect another.
- Posting the JSON body fields your playbook references (e.g. src_ip) lets enrichment and actions template them.

Configuring a Playbook Webhook — Step by Step

This worked example wires an external alert source to a playbook end-to-end.

Step	Action
1 · Create	Go to **Playbooks** → **New Playbook**. The canvas opens with a **Webhook Trigger** node as the entry point.
2 · Design	From the trigger, wire **Enrich** → **AI Triage** → **Human Approval** → **Execute**. Decide which payload fields you'll reference (e.g. src_ip, file_hash) — these come from the JSON your source will POST.
3 · Name & Save	Enter a name and click **Save Playbook**. The platform generates the unique **webhook-path** for this playbook and returns to the Playbooks list.
4 · Copy the URL	On the playbook's card, click the **copy** icon to grab the full tenant-aware trigger URL: `/api/webhook/<tenant-slug>/<webhook-path>`.
5 · Configure source	In your SIEM / EDR / firewall / monitoring tool, add an outbound webhook / alert action that POSTs a JSON body to that URL with `Content-Type: application/json`.
6 · Match fields	Ensure the JSON keys match the placeholders your Enrich/Execute nodes use (e.g. send "src_ip" if a node references {{src_ip}}).
7 · Test	Fire a test event (or the curl above). A new incident appears on the **Dashboard**; after AI triage it lands in the **Triage Queue** for approval.
8 · Enable / disable	A saved playbook is enabled by default; its card shows an **ENABLED** badge. A disabled playbook's webhook returns **404**.

Tip: rotating a webhook is as simple as saving a new playbook (new path) and repointing your source; disable the old one to revoke its URL immediately.

6.2 · Outgoing Webhooks (API Integrations)

Outgoing integrations are the APIs your playbooks call from **Enrich** and **Execute** nodes. They are **schema-driven**: you describe the API once, and the runner injects credentials and templated values at runtime. Create them in **Settings** → **API Integrations** (3rd Line Analyst or above).

Setting one up

- Open **Settings** → **API Integrations** → **New API**.
- Enter a **Name**, the API's **Base URL**, and one or more **Endpoints** as JSON.
- Provide an **API Key** (sent as a Bearer token) and/or an **API Token** (sent as X-API-Token). These are encrypted into the vault and never returned to the browser.
- In a playbook's Enrich/Execute node, select the integration and the endpoint key (the action) to call.

Endpoints schema

```
{
"ip_lookup": { "method": "GET", "path": "/ip/{{src_ip}}" },
"block_ip": { "method": "POST", "path": "/block",
"bodyTemplate": { "ip": "{{src_ip}}" } }
}
```

{{field}} placeholders are interpolated from the live event payload (and prior enrichment) when the node runs.

The Credential Vault

- Secrets are **never stored in plaintext** — encrypted with **AES-256-GCM** (unique IV + auth tag per record), scoped to your tenant.
- The runner decrypts and injects the credential into the outbound request headers at execution time only.
- Secrets never reach the browser or appear in playbook definitions or exports.

7 · Incident Triage (Human-in-the-Loop)

Configuring the AI Co-Pilot

The AI Triage node uses Claude to score and analyze each enriched event. The co-pilot is **per-organization** and is **off by default** — an administrator (3rd Line Analyst or above) configures it under **Settings** → **AI Co-Pilot**.

Step	Action
1 · Key	Paste your **Anthropic API key** and click **Save Key**. It is encrypted (AES-256-GCM) into the tenant vault and never shown again.
2 · Toggle	Switch the **AI co-pilot** toggle to **On**. The panel pill and the dashboard chip turn green / Active.
3 · Verify	Trigger a playbook with an AI Triage node — the incident's **AI Analysis** accordion now shows the risk score, guidance and proposed actions.

When disabled: incidents skip AI scoring and route straight to manual triage with an "AI co-pilot disabled — manual review required" note, so playbooks keep working. The key falls back to the `ANTHROPIC_API_KEY` environment variable if one was provided at deploy time, but the bot still only runs once the toggle is On.

The Triage Workflow

When a playbook reaches the AI Triage node, the runner serializes its state to Redis, generates a callback token, and **suspends execution** — preventing HTTP timeouts on long investigations. The incident surfaces in the **Triage Queue** for human review.

The Progressive-Disclosure Triage Dashboard

Each incident opens as three vertical accordions, ordered by how often you need them:

Accordion	State	Contents
1 · Raw Trigger Data	Collapsed	The original event payload as formatted JSON.
2 · AI Analysis & Guidance	Collapsed	Risk score, severity, executive summary, analyst guidance & IOCs.
3 · Proposed Remediations	Expanded	Independent action toggles — approve or reject each one.

Granular Approvals & Execution

- Toggle each proposed action independently — approve the safe ones, reject the rest.
- The execute button shows a live count of selected actions.
- Approving any action the AI marked **HIGH risk** requires a **2nd Line Analyst or above**; 1st Line may approve low-risk actions.
- On execution, the runner resumes from the suspended state and dispatches only the approved actions. Both raw JSON and a branded PDF can be exported from the triage view.

8 · Users, MFA & Audit

Administrators manage their organization's users, MFA policy, and audit trail from **Settings**. These sections are hidden from analysts.

Managing Users

- **Settings** → **Users & Access** lists every user with their role and MFA status.
- **Add User** creates an account with a name, email, password and role. You cannot grant a role above your own.
- Change a user's role inline; remove a user with the trash icon (you cannot delete yourself or a higher-ranked user).

Forcing MFA

MFA can be required two ways, and both force enrolment on the user's next login:

Method	Effect
Organization-wide	Toggle **Enforce MFA organization-wide** — every user in the tenant must enrol TOTP before access.
Per-user	The **Force MFA** button on a user row requires just that individual to enrol.

A required user is sent to a one-time enrolment screen at login: they scan a QR code with an authenticator app and confirm a 6-digit code. Users may also enrol voluntarily from **Settings** → **Security** → **Set up MFA now**.

The Audit Log

Settings → **Audit Log** shows a paginated, tamper-evident record of security-relevant events — logins, MFA enrolment, triage approvals, user and organization changes — each with the actor, timestamp and detail.

9 · Case Management & Export

Every incident maintains a single, evolving **immutable JSON ledger**. As the case progresses, each phase is appended — never overwritten — producing a complete, tamper-evident audit trail.

The Unified Ledger Lifecycle

Phase	Recorded
TRIGGER	The original inbound event payload.
ENRICHMENT	Results from each enrichment integration.
AI_ANALYSIS	The Claude analyst's full structured assessment.
HUMAN_APPROVAL	Which analyst approved which actions, with timestamp.
EXECUTION	The result of each dispatched remediation.

Exporting an Incident

- **Export JSON** — downloads the raw ledger for SIEM ingestion or archival.
- **Export PDF** — generates a branded *SOAR Incident Audit Report* with the risk summary, full timeline, and analyst guidance, ready to share with stakeholders.

10 · Security Model

- **Zero-Trust:** every login evaluates context — user role, network origin, and MFA status.
- **Short-lived sessions:** access tokens (JWT) expire after 15 minutes; secure HttpOnly refresh tokens last 7 days and can be revoked.
- **RBAC:** capabilities are enforced server-side on every request; high-risk approvals are role-gated.
- **Enforceable MFA:** TOTP enrolment can be required org-wide or per user; step-up re-verification protects privileged actions.
- **Tenant isolation in depth:** application-level scoping AND PostgreSQL Row-Level Security confine every row to its organization.
- **Encrypted vault:** all integration credentials are stored under AES-256-GCM, per tenant.
- **Full audit logging:** logins, triage decisions, and user/org changes are written to an append-only audit log.
- **Network isolation & rate limiting:** only the app port is exposed; per-tenant webhook rate limits prevent cross-tenant interference.

For installation, environment configuration, and Docker orchestration, see **SETUP.md** in the project root.

— End of User Guide —