TRAINING GUIDE

Configuration, Setup & Use

A hands-on course for new Users & Administrators

Version 1.1 · Labs · Objectives · Knowledge Checks

How to Use This Guide

This guide is organized into **modules**. Each opens with learning objectives, walks you through the concepts, includes a hands-on lab, and ends with a knowledge check. Modules are tagged by audience:

Tag	Audience
ALL USERS	Everyone — orientation and day-to-day use.
ADMINISTRATOR	Admins — deployment, configuration, governance.
ANALYST	1st/2nd/3rd Line analysts — operational workflows.

Curriculum

#	Module	Audience
0	Orientation & First Login	All
1	Deployment & Initial Setup	Admin
2	Configuring the Platform (AI, MFA, Users)	Admin
3	Integrations & the Credential Vault	Admin
4	Multi-Tenancy & Provisioning Orgs	Admin
5	Building Your First Playbook	Analyst
6	Configuring Incoming Webhooks	Analyst
7	Working the Triage Queue	Analyst
8	Case Management & Reporting	Analyst
9	Security Best Practices	All

Module 0 · Orientation & First Login

ALL USERS

Learning objectives

- ✓ Understand what Synapse IronNode does and its core building blocks.
- ✓ Log in and navigate the main areas of the application.
- ✓ Complete (or recognise) the first-run setup.

What is Synapse IronNode?

A self-hosted SOAR (Security Orchestration, Automation & Response) platform. It receives security events, enriches them, lets an embedded Claude AI analyst triage them, and runs analyst-approved remediations — all recorded in an immutable case ledger.

The Main Areas

Area	What it's for
Dashboard	SOC triad (Alerts / Incidents / Events), recent incidents, AI co-pilot status.
Playbooks	Build and manage visual automation workflows.
Triage Queue	Review AI analysis and approve remediations awaiting a human.
Incidents	Full incident history with search and filters.
Settings	Profile, MFA, AI co-pilot, integrations, users & audit log.
Organizations	(Platform Admin) Provision and manage tenants.

Hands-on Lab — Log in for the first time

1. Open **http://<your-host>:3000**.
2. If this is a brand-new deployment, you'll see **First Launch Detected** — see Module 1.
3. Otherwise enter your email and password. If prompted, enter your 6-digit MFA code.
4. Confirm you land on the **Operations Dashboard** and can see the sidebar navigation.

Knowledge check — you should now be able to:

- ■ Describe the purpose of the platform in one sentence.
- ■ Navigate to the Dashboard, Playbooks, Triage Queue and Settings.

Module 1 · Deployment & Initial Setup

ADMINISTRATOR

Learning objectives

- ✓ Configure environment secrets and start the stack with Docker.
- ✓ Complete first-run setup to create the organization and Platform Admin.
- ✓ Verify the platform is healthy.

Prerequisites

- Docker Desktop / Docker Engine + Compose.
- An Anthropic API key (optional now; can be added later in Settings).

Hands-on Lab — Deploy the stack

1. Copy the env template: `Copy-Item .env.example .env`
2. Generate secrets and fill them into `.env`:
3. `openssl rand -base64 64` → JWT_SECRET and JWT_REFRESH_SECRET (run twice).
4. `openssl rand -hex 32` → VAULT_ENCRYPTION_KEY (32-byte).
5. Place your logo at `public/logo.png` (and logo.jpg).
6. Start everything: `docker compose up --build -d`
7. Watch the logs until you see `[entrypoint] Starting Synapse IronNode…` — migrations run automatically on start.

First-Run Setup

Hands-on Lab — Create your organization & admin

1. Browse to **http://<host>:3000**; the login screen shows **First Launch Detected**.
2. Enter your **Organization name**, your name, email, and a strong password (12+ chars).
3. Click **Initialize Synapse IronNode**. This atomically creates the tenant and your **Platform Admin** account.
4. You are signed straight into the Dashboard.

Tip: If you ever see an “Initialization Error” screen, the database isn't reachable yet — wait for the db container to become healthy and click Retry. Setup stays available until a real account exists.

Knowledge check — you should now be able to:

- ■ Generate the three required secrets and start the stack.
- ■ Complete first-run setup and explain why no default credentials exist.

Module 2 · Configuring the Platform

ADMINISTRATOR

Learning objectives

- ✓ Enable and configure the AI co-pilot.
- ✓ Create users and assign SOC roles.
- ✓ Enforce MFA org-wide and per user.

2.1 · Enable the AI Co-Pilot

Hands-on Lab — Turn on Claude triage

1. Go to **Settings** → **AI Co-Pilot**.
2. Paste your **Anthropic API key** and click **Save Key** (stored AES-256-GCM encrypted).
3. Flip the **AI co-pilot** toggle to **On**.
4. Confirm the panel pill and the **Dashboard** chip turn green / **Active**.

2.2 · Create Users & Assign Roles

Roles inherit cumulatively. Assign the least privilege needed:

Role	Can do
1st Line Analyst	Triage; approve LOW-risk actions.
2nd Line Analyst	+ approve HIGH-risk; build playbooks.
3rd Line Analyst	+ manage integrations, vault & AI key.
Administrator	+ manage users, force MFA, view audit log.
Platform Admin	+ provision organizations (cross-tenant).

Hands-on Lab — Add an analyst

1. Open **Settings** → **Users & Access** → **Add User**.
2. Enter their name, email, a temporary password and select a role (e.g. **1st Line Analyst**).
3. (Optional) tick **Require this user to set up MFA**.
4. Click **Create User** and confirm they appear in the list.

2.3 · Enforce MFA

- **Org-wide:** toggle **Enforce MFA organization-wide** — every user must enrol on next login.
- **Per-user:** click **Force MFA** on an individual's row.
- Required users are sent to a QR-code enrolment screen at their next login.

Knowledge check — you should now be able to:

- Enable the AI co-pilot and confirm Active status.
- Create a user with an appropriate least-privilege role.
- Enforce MFA both org-wide and for a single user.

Module 3 · Integrations & the Credential Vault

ADMINISTRATOR

Learning objectives

- ✓ Register an outgoing API integration playbooks can call.
- ✓ Understand how credentials are encrypted and injected.

Integrations are the outgoing APIs used by playbook Enrich/Execute nodes. They are schema-driven: define the base URL and endpoints once, and the runner injects credentials and templated values at runtime.

Hands-on Lab — Create a threat-intel integration

1. Go to **Settings** → **API Integrations** → **New API**.
2. Name it (e.g. **Threat Intel**) and set the **Base URL** (e.g. https://api.example.com/v3).
3. Define **Endpoints** as JSON, using {{field}} for payload values:

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

Hands-on Lab — …continued

1. Enter an **API Key** (Bearer) and/or **API Token** (X-API-Token).
2. Click **Create Integration**. The secret is encrypted into the vault — it is never shown again.
3. In a playbook, select this integration + the endpoint key (the action) on an Enrich/Execute node.

Tip: Secrets are AES-256-GCM encrypted per tenant and only decrypted inside the runner at request time — they never reach the browser or appear in playbook definitions or exports.

Knowledge check — you should now be able to:

- ■ Register an integration with at least one endpoint and a credential.
- ■ Explain where the credential is stored and when it is used.

Module 4 · Multi-Tenancy & Provisioning Orgs

ADMINISTRATOR

Learning objectives

- ✓ Understand tenant isolation.
- ✓ Provision a new organization (MSSP scenario).

Every record belongs to an organization (tenant). Isolation is enforced both in the application and by PostgreSQL Row-Level Security, so tenants can never see each other's data. A **Platform Admin** can run many client tenants from one deployment.

Hands-on Lab — Provision a client organization

1. As a Platform Admin, open **Organizations** → **New Organization**.
2. Enter the org **name**, a URL-safe **slug**, and a webhook rate limit (per minute).
3. Provide the new org's **Administrator** (name, email, password).
4. Click **Create Tenant & Org Admin**. The new admin can now log in and manage only their tenant.

Tip: Each tenant gets its own webhook slug and its own rate limit, so one client's event burst can never affect another.

Knowledge check — you should now be able to:

- ■ Explain the two layers that enforce tenant isolation.
- ■ Provision a new organization with its own administrator.

Module 5 · Building Your First Playbook

ANALYST

Learning objectives

- ✓ Use the visual builder to assemble a workflow.
- ✓ Add, connect, copy and delete nodes.
- ✓ Save a playbook (2nd Line Analyst or above).

The flow you'll build: **Trigger** → **Enrich** → **AI Triage** → **Human Approval** → **Execute**.

Hands-on Lab — Assemble a playbook

1. Open **Playbooks** → **New Playbook**. A **Webhook Trigger** node is placed for you.
2. Use the toolbar to add **Enrich**, **AI Triage**, **Human Approval** and **Execute** nodes.
3. Drag from a node's right handle to another's left handle to connect them.
4. Select a node and press **Ctrl+D** to copy it, or **Del** to remove it (connected edges are cleaned up).
5. On the Enrich/Execute nodes, choose an integration + endpoint (from Module 3).
6. Give the playbook a name and click **Save Playbook**.
7. Confirm it now appears as a card in the Playbooks list with a webhook URL.

Knowledge check — you should now be able to:

- ■ Build a 5-node playbook and connect the nodes in order.
- ■ Copy and delete a node using keyboard shortcuts.
- ■ Save the playbook and locate its card in the list.

Module 6 · Configuring Incoming Webhooks

ANALYST

Learning objectives

- ✓ Locate a playbook's tenant-aware webhook URL.
- ✓ Point an external source at it and test the trigger.

Webhook URLs are tenant-aware:

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

Hands-on Lab — Wire up a trigger

1. On the **Playbooks** list, click the **copy** icon on your playbook's card to grab its full URL.
2. In your SIEM / EDR / firewall, add an outbound webhook that POSTs JSON to that URL with `Content-Type: application/json`.
3. Ensure the JSON keys match your playbook's {{placeholders}} (e.g. send "src_ip").
4. Send a test event, or use curl:

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

- A **202 Accepted** response means the run started; a new incident appears on the Dashboard.
- Disabling a playbook makes its webhook return **404** — an instant kill-switch.
- Exceeding the tenant rate limit returns **429** with a Retry-After header.

Knowledge check — you should now be able to:

- ■ Retrieve a playbook's webhook URL and identify the tenant-slug and path.
- ■ Trigger the playbook with a test event and see the incident created.

Module 7 · Working the Triage Queue

ANALYST

Learning objectives

- ✓ Review an incident's AI analysis.
- ✓ Approve or reject individual remediations.
- ✓ Understand role gating on high-risk actions.

When a playbook reaches the AI Triage node it suspends and the incident appears in the **Triage Queue**. Open it to see three accordions: **Raw Trigger Data**, **AI Analysis & Guidance**, and **Proposed Remediations**.

Hands-on Lab — Triage an incident

1. Open **Triage Queue** and select an incident awaiting review.
2. Expand **AI Analysis** to read the risk score, severity and analyst guidance.
3. In **Proposed Remediations**, toggle on the actions you approve; leave the rest off.
4. Click **Execute** — the count reflects your selection. Only approved actions run.
5. (Optional) Export the incident as JSON or PDF from the triage view.

Tip: Approving an action the AI marked HIGH risk requires a 2nd Line Analyst or above. 1st Line can approve low-risk actions and escalate the rest.

Knowledge check — you should now be able to:

- ■ Interpret an AI analysis and its proposed actions.
- ■ Approve a subset of actions and execute them.
- ■ State who may approve high-risk actions.

Module 8 · Case Management & Reporting

ANALYST

Learning objectives

- ✓ Understand the immutable incident ledger.
- ✓ Export incidents for records and stakeholders.

Each incident is a single append-only JSON ledger capturing its whole lifecycle — nothing is overwritten, producing a tamper-evident trail.

Phase	Recorded
TRIGGER	Original inbound event.
ENRICHMENT	Integration results.
AI_ANALYSIS	Claude's structured assessment.
HUMAN_APPROVAL	Who approved which actions.
EXECUTION	Result of each remediation.

Hands-on Lab — Export a case

1. Open any incident from **Incidents** or the Triage Queue.
2. Click **Export JSON** for the raw ledger (SIEM/archival).
3. Click **Export PDF** for a branded **SOAR Incident Audit Report** to share with stakeholders.

Knowledge check — you should now be able to:

- ■ List the five ledger phases in order.
- ■ Produce both a JSON and a PDF export of an incident.

Module 9 · Security Best Practices

ALL USERS

Learning objectives

- ✓ Apply day-to-day security hygiene.
- ✓ Know the platform's built-in protections.

- **Enrol MFA** immediately (Settings → Security) even if not yet enforced.
- **Least privilege:** assign the lowest analyst tier that lets someone do their job.
- **Rotate webhooks:** to revoke a trigger, disable the playbook (instant 404) or save a new one for a fresh path.
- **Protect secrets:** never paste API keys into playbook fields — use the encrypted vault (Integrations / AI Co-Pilot).
- **Review the audit log** (Administrators) regularly for unexpected logins or changes.
- **Keep .env private:** JWT secrets and the vault key must never be committed or shared.

Built-in protections

- 15-minute access tokens; revocable refresh tokens.
- Tenant isolation by app-level scoping AND PostgreSQL RLS.
- Per-tenant webhook rate limiting; append-only audit logging.

Further reading: USER_GUIDE.pdf (reference), TECHNICAL_OVERVIEW.pdf (architecture), SETUP.md (environment).

— End of Training Guide —