SAAS SECURITY CHECKLIST

Step 1

Map the chaos, then pick your battles

- [ ] Run an **initial discovery** of your SaaS stack.

  Why: Create your baseline - you can't prioritize or protect what you don't know exists.

- [ ] Classify systems by **data type** (customer, employee/HR, business/financial; candidates/marketing adjacent).

  Why: priorities follow data impact, not vendor logos.

- [ ] Tag the **core apps** that hold sensitive data

  Why: email/docs & shared drives, code repos, CRM, and HR are where most real risk live.

- [ ] Run an org-wide review of third-party OAuth apps (start with your email/docs suite).

  Why: finds the silent "signed in with X" connections you forgot existed.

Step 2

Enforce onboarding, off-boarding, and access control

- [ ] **Onboard on rails:** default roles + SSO/MFA for core apps; no side-door invites.

  Why: ad-hoc access is painful to revoke later.

- [ ] **Off-board with zero leftovers:** disable accounts, revoke access.

  Why: lingering access is the easiest breach.

- [ ] **Close the loop when users leave**

  Why: disabling accounts isn't enough. Ensure seats are cleaned up and no stray access lingers

- [ ] **Lightweight access reviews** on top apps (admins, exporters, owners).

  Why: roles drift; reviews catch it.

Step 3

Trace what's connected to what (advanced)

- [ ] Export the current OAuth roster again

  Why: ad-hoc access is painful to revoke later.

- [ ] Flag connections touching core apps.

  Why: not all integrations are risky — start with those near sensitive data.

- [ ] (Advanced) Sketch **app→ app chains** where data hops between systems; park them for a focused pass.

  Why: app-to-app chains expand your blast radius.

Step 4

Schedule monthly & quarterly checks

- [ ] **Monthly:** review **new OAuth connections** to your core systems.

  Why: approvals pile up quietly; keep the noise down.

- [ ] **Quarterly:** review **active users & permissions** across your top five apps.

  Why: "temporary" access lingers.

- [ ] **Quarterly:** scan for **public or widely shared links** in document storage.

  Why: lowest-effort data leaks.

- [ ] **Quarterly:** run an **AI-assisted sweep** for risky patterns (see Step 5).

  Why: catch what you'd miss by hand.

Step 5

Use AI where it actually helps

- [ ] Connect your core data sources to Sola
- [ ] Write plain-language prompts to get answers and action items
  - Find files in our documents storage that are publicly accessible or overshared
  - List all integrations, OAuth apps, and tokens with repo-wide or org-wide access
  - Identify integrations with high-impact scopes unused in the last 60–90 days
  - Inventory non-human identities with elevated permissions
- [ ] Act on the AI's answers to fix what matters
- [ ] Circle back on a regular cadence

sola