NetraScale's RiskAct SaaS Supply Chain Checklist
fast actions to reduce risk

About This Checklist

The NetraScale SaaS Supply Chain Checklist is designed to help organizations quickly evaluate and strengthen vendor security and compliance practices. While not a one-to-one mapping, the checklist draws directly from **ISO 27001, SOC 2, and NIST Cybersecurity Framework (CSF)** — three of the most widely adopted standards in regulated industries.

By consolidating overlapping requirements, this checklist provides a **practical, actionable baseline** for SaaS providers and their partners. It enables security, risk, and compliance teams to:

- Align with recognized global standards.
- Prepare for audits and assessments.
- Simplify vendor risk management.
- Strengthen overall cyber resilience across the supply chain.

This makes it a flexible tool for both **CISOs and compliance teams** looking to save time while staying audit and board ready.

Inventory and Visibility

- ❑ Inventory all SaaS apps and integrations. **Why**: you cannot protect what you do not know. **Action**: create or export a list from your SSO or IT asset tool.
- ❑ Map which apps have access to critical data. **Why**: not every app needs the same level of trust. **Action**: tag apps by data sensitivity.
- ❑ Log every publisher and package used in builds. **Why**: open-source components flow into production quickly. **Action**: capture package names, versions, and publisher accounts.

Access and Authentication

- ❑ Enforce multi-factor authentication on publisher and admin accounts. **Why**: tokens and credentials are common targets. **Action**: turn on MFA for all service and publisher accounts.
- ❑ Remove long-lived credentials from CI/CD. **Why**: build agents are high value targets. **Action**: use ephemeral credentials for builds.
- ❑ Limit who can publish or approve package changes. **Why**: smaller blast radius. **Action**: enforce least privilege and approval workflows.

Dependency and Package Controls

- ❑ Pin dependency versions in package manifests. **Why**: prevents silent upgrades to malicious releases. **Action**: update manifests and lockfiles.
- ❑ Verify package checksums or signatures where possible. **Why**: integrity matters. **Action**: add checksum verification to CI.
- ❑ Block or quarantine packages from unverified publishers. **Why**: stop risky code before it runs. **Action**: add allowlists and deny lists.

CI/CD and Build Hygiene

- ❑ Run builds in isolated environments. **Why**: reduces side effects if malicious code executes. **Action**: use ephemeral containers or dedicated runners.
- ❑ Do not store secrets in build logs or environment variables. **Why**: attackers harvest them. **Action**: move secrets to vaults and ephemeral fetch.
- ❑ Scan build outputs for suspicious bundle.js or embedded scripts. **Why**: supply chain payloads often hide in bundles. **Action:** integrate SCA and static analysis.

Runtime Monitoring and Detection

- ❑ Monitor for unusual outbound connections from build or runtime environments. **Why**: C2 and credential exfiltration are key signs. **Action**: add network egress rules and alerts.
- ❑ Watch for unexpected process activity tied to package execution. **Why**: persistence shows compromise. **Action**: enable runtime behavior monitoring.

Vendor and Publisher Management

- ❑ Validate publisher identity and account security for any package you consume. **Why**: compromised publisher accounts are a frequent vector. **Action**: require MFA and verified contacts for publishers.
- ❑ Limit direct integrations with third-party marketing tools or CRMs. **Why**: these often hold tokens. **Action**: segment and isolate integrations.

Incident Readiness and Response

- ❑ Have a supply chain playbook and run a tabletop exercise. **Why**: speed matters during contamination. **Action**: run a scenario that includes token theft.
- ❑ Rotate credentials quickly after any suspected exposure. **Why**: tokens are immediate risk. **Action**: automate rotation procedures where possible.
- ❑ Preserve build artifacts and logs for forensics. **Why**: investigation needs context. **Action**: enable secure, write-once logging retention.

Governance, Reporting and Compliance

- ❑ Assign ownership for SaaS and package risk. **Why**: accountability reduces drift. **Action**: name an owner for each critical app.
- ❑ Add supply chain risk metrics to leadership reports. **Why**: board-level visibility drives investment. **Action**: track time-to-detect and time-to-rotate metrics.
- ❑ Conduct quarterly third-party risk reviews. **Why**: vendor risk changes fast. **Action**: require attestation and evidence from critical vendors.

Start with visibility, then reduce exposure. Repeat often.