Software Composition Analysis (SCA) Policy

AI Security Policy Framework

Our current software supply chain contains 33 known vulnerable dependencies, including 16 high-severity findings. Immediate remediation and continuous monitoring are required to reduce exposure and comply with regulatory and contractual obligations.

Generated: November 13, 2025

Software Composition Analysis (SCA) Policy

Policy Metadata

Policy ID	POL-SCA-2025-001
Status	Active
Created Date	2025-11-05T00:00:00Z
Last Updated	2025-11-05T00:00:00Z
Author	Security Policy Generator

Executive Summary

Our current software supply chain contains 33 known vulnerable dependencies, including 16 high-severity findings. Immediate remediation and continuous monitoring are required to reduce exposure and comply with regulatory and contractual obligations.

Policy Statement

Purpose

To manage third-party dependency security risks and ensure supply-chain integrity for all software assets.

Description

This policy establishes mandatory requirements for automated dependency scanning, vulnerability triage, remediation, and the creation and maintenance of a Software Bill of Materials (SBOM).

Applicability

All applications, services, containers, and infrastructure-as-code artifacts that incorporate third-party libraries, packages, or open-source components.

Enforcement

Automated SCA scans must run on every code commit and build pipeline; builds containing high-severity (CVSS ≥7.0) vulnerable dependencies are blocked until remediation or approved exception.

Exceptions

Critical business-critical dependencies with no viable alternatives may be approved by the Change Advisory Board (CAB) provided compensating controls such as runtime monitoring and additional penetration testing are implemented.

Objectives

1. Establish a definitive, version-controlled inventory of all third-party components.
2. Detect and remediate high-severity vulnerabilities within 7 days of discovery.
3. Maintain an up-to-date SBOM and ensure continuous compliance with licensing and security policies.

Risk Assessment

Risk Metric	Value
Overall Risk Level	High
Critical Risks	0
High Risks	16
Medium Risks	14
Low Risks	3
Likelihood	High

Security Controls

SC-SCA-001: Dependency Inventory Management

Maintain a centralized, version-controlled Software Bill of Materials (SBOM) for every application repository.

- Integrate sbom-generation tools (e.g., CycloneDX, Syft) into CI pipelines.
- Store SBOM files in a read-only artifact repository linked to the source code.
- Review and reconcile SBOM against approved vendor list quarterly.

SC-SCA-002: Automated Vulnerability Scanning

Run SCA scans on every pull request and nightly builds; block merges when high-severity findings are present.

- Configure SCA tool (e.g., Snyk, Dependabot, OWASP Dependency-Check) to fail builds on CVSS ≥7.0.
- Publish scan results to a centralized dashboard for visibility.
- Enforce policy as code using GitHub Actions or Azure Pipelines gate.

SC-SCA-003: Continuous Threat Intelligence Integration

Subscribe to CVE feeds and vulnerability databases to automatically update scanning signatures.

- Enable daily sync with NVD, GitHub Advisory Database, and vendor security bulletins.
- Map incoming CVEs to affected components in the SBOM.
- Trigger alerts in the Security Operations Center (SOC) when new high-severity CVEs are identified.

SC-SCA-004: Vulnerability Remediation Workflow

Standardize a ticket-based process for triaging, fixing, and verifying vulnerable dependencies.

- Create a remediation ticket automatically for each high-severity finding.
- Assign tickets to the owning development team with a 7-day SLA.
- Require peer-review and automated regression testing before merge.

SC-SCA-005: Rollback and Patch Deployment

Ensure rapid rollback or patch deployment mechanisms are in place for vulnerable components.

- Maintain version-controlled rollback scripts in the repository.
- Test rollback procedures in a staging environment quarterly.
- Document recovery steps in the incident response playbook.

Remediation Actions

P0: Patch all high-severity vulnerable dependencies

Owner: DevOps Team | Timeline: Immediate (0-7 days)
Affected Assets:

- lodash@4.17.20
- express@4.16.0
- moment@2.24.0
- axios@0.19.2
- react@16.8.6

Success Criteria: All high-severity CVEs resolved; no build failures due to blocked dependencies.

P1: Upgrade medium-severity dependencies with known exploits

Owner: Application Development Leads | Timeline: Within 30 days
Affected Assets:

- chalk@2.4.2
- debug@3.2.7
- xml2js@0.4.19

Success Criteria: Medium-severity CVEs reduced by 80%; regression tests pass.

P2: Review low-severity and license-risk dependencies

Owner: Legal & Compliance | Timeline: Within 60 days
Affected Assets:

- left-pad@1.3.0
- underscore@1.9.1

Success Criteria: All license conflicts resolved; low-severity CVEs documented with risk acceptance.

Compliance Mapping

Framework	Controls
NIST CSF	• ID.AM-2 • PR.IP-12 • DE.CM-8 • RS.RP-1 • RC.RP-1
ISO 27001	• A.15.1.1 • A.15.1.2 • A.12.6.1 • A.12.4.1 • A.16.1.2 • A.17.1.2

Monitoring Requirements

- Daily automated SCA scans on all repositories.
- Continuous CVE feed ingestion and correlation with SBOM.
- Weekly license compliance reports.
- Monthly KPI dashboard (mean time to remediate, scan pass rate).

Review Schedule

Monthly dependency review meetings; quarterly formal audit of SCA process and compliance.