Information Technology Executive Council

ITEC 7050-P

1.0 TITLE: Secure System Development Policy

2.0 PURPOSE: The purpose of this policy is to establish common expectations Entities will follow to reduce the number of vulnerabilities in released software, mitigate the potential impact of the exploitation of undetected or unaddressed vulnerabilities, and address the root causes of vulnerabilities to prevent future recurrences.

3.0 SCOPE: This policy applies to all software or code developed, maintained by the Entity for systems that are categorized as Restricted-Use Information Systems or External Facing. This includes software and code developed in-house or by third-party vendors.

4.0 ORGANIZATIONS AFFECTED: This policy applies to all State of Kansas Executive branches, boards, commissions, departments, divisions, Entities, and third parties involved in processing, transmitting, or providing business capabilities on behalf of Kansas state government, hereafter referred to as Entities.

5.0 REFERENCES:

5.1 National Institute for Standards and Technology (NIST) Cyber Security Framework (CSF) 2.0

5.2 NIST Special Publication 800-53 Revision 5

5.3 NIST SP 800-218 Secure Software Development Framework Version 1.1

6.0 DEFINITIONS:

6.1 Dynamic Application Security Testing (DAST): Security testing used to identify vulnerabilities in a running application by analyzing its behavior, responses, and interactions in real time.

6.2 Software Bill of Materials (SBOM): A formal record containing the details and supply chain relationships of various components used in building software.

6.3 Software Composition Analysis: A process or set of tools used to inspect software components and their dependencies to identify security and compliance concerns.

6.4 Static Application Security Testing (SAST): Security testing that analyzes an application's source code, bytecode, or binaries for vulnerabilities without executing the application.

7.0 POLICY:

This policy is the principal governing authority for Secure System Development by all Entities. While individual Entities retain the right to impose supplemental restrictions through their Entities-specific policies, such policies must not contradict the provisions outlined in this policy.

Entities must:

Roles and Responsibilities

7.1 Identify and document individuals with information security roles and responsibilities throughout the system development life cycle (SDLC).

7.2 Provide role-based training for all personnel with responsibilities that contribute to secure development.

Planning

7.3 Identify and document security requirements as part of the requirements gathering process.

7.4 Communicate security requirements to developers and third-party software providers.

7.5 Require third parties to attest that their software complies with all ITEC and Entity security requirements.

7.6 Track and maintain the software's security requirements, risks, and design decisions.

7.7 Perform risk modeling – such as threat modeling, attack modeling, or attack surface mapping – to help assess the security risk for the purposed system.

Development

7.8 Adopt and enforce secure coding standards based on relevant standards (e.g., OWASP Secure Coding Practices, Microsoft Secure Development Lifecycle, CERT Secure Coding Standards, etc.).

7.9 Confirm security requirements are included within the development process.

7.10 Protect and separate preproduction (development, test, and integration) environments commensurate with risk throughout the system development life cycle.

7.11 Store all forms of code – including source code, executable code, and configuration-as-code – based on the principle of least privilege.

7.12 Use code repository version controls to track individual accountability for all source code changes.

7.13 Approve, document, and control the use of live Restricted-Use Information, in pre-production environments.

7.14 Integrate Software Composition Analysis (SCA) tools into the development lifecycle to identify and manage risks associated with third-party and open-source components.

7.15 Generate, maintain, and validate Software Bill of Materials (SBOM) for system developed in-house or by third-party providers that inventory items list in Appendix A.

7.15.1 Only approved third-party components and libraries, documented in an SBOM, may be used

7.15.2 Deprecated or end-of-life components must be replaced with supported versions.

Testing

7.16 Conduct Static Application Security Testing (SAST) or peer code reviews to identify vulnerabilities, coding errors, and adherence to security requirements.

7.17 Conduct Dynamic Application Security Testing (DAST) prior to deployment to the production environment.

7.18 Track and address security findings to include categorization by severity, establishment of timelines for resolution based on the severity and potential impact of the identified vulnerabilities.

Deployment

7.19 Where applicable, enforce controlled ~~through~~ deployments through continuous integration and continuous deployment (CI/CD) pipelines.

7.20 Require security review and approval for production releases.

7.21 Validate all configurations against security requirement, applicable benchmarks, and secure coding standards during deployment.

7.22 ADD – Separation of duties, person writing code should not be deploying the code …

Maintenance

7.23 Regularly check whether there are publicly known vulnerabilities in the software modules and services that vendors have not yet fixed.

7.24 Ensure software components are actively maintained and have not reached end of life (EOL).

7.25 ADD – Separation of duties, person maintaining/changing should not be deploying the code … reference or review change control policy to see if there is any overlap for this control.

8.0 RESPONSIBILITIES:

8.1 Heads of entities are responsible for establishing procedures for their organization's compliance with the requirements of this policy.

8.2 The Chief Information Security Officer, Executive Branch, is responsible for the maintenance of this policy.

9.0 CANCELLATION: Previous versions of this policy

Appendix A.

Software Bill of Materials (SBOM) Minimum Requirements

Category	**Requirement**	**Examples/Details**
Component Details	Component Name	log4j-core
	Version Number	2.17.1
	Component Type	Library, framework, API, container image
Dependency Information	Direct Dependencies	Components directly integrated into the application
Licenses	License Information	GPL, MIT, Apache 2.0
	License Compliance Status	Identification of license conflicts or violations
Source Information	Repository URL	https://github.com/apache/logging-log4j2
	Download Location	Package manager registry or vendor source
Supplier Information	Supplier Name	Refers to the originator or manufacturer of the software component (i.e., vendor or organization responsible for the component).
	Contact Information	Optional contact details for the supplier
Metadata	SBOM Version	Version of the SBOM format (e.g., SPDX, CycloneDX, SWID)
	Author Information	Name or organization generating the SBOM
	Creation Timestamp	Date and time the SBOM was generated