Internal Policies and Procedures of the Utah State Board of Education	
Policy #	05-14
Subject:	Software Development Secure Practices Policy
Date Approved	February 21, 2024
Policy Owner's Title	Chief Information Security Officer
Policy Officer's Title	Deputy Superintendent of Operations
References: NIST Special Publication 800-218 USBE Data Governance Plan OWASP Developer Guide SAFECode Fundamental Practices for Secure Software Development, Third Edition -Center for Internet Security (CIS) Critical Security Controls – Control 16	

1) Purpose and Scope

a) This policy sets forth a minimum set of requirements for all software/application development within the Utah State Board of Education (USBE).

  i) This policy applies to all application developed fully or partially by USBE developers, applications modified by USBE Developers, and Commercial off-the-shelf (COTS) application.

2) Policy.

a) A Secure Application Development Process must be established and maintained.

  i) This process should address items such as: secure application design standards, secure coding practices, developer training, vulnerability management, security of third-party code, and application security testing procedures.

  ii) Documentation should be reviewed and updated annually, or when significant enterprise changes occur that could impact this Process.

b) A process must be established and maintained for accepting and addressing Software Vulnerabilities.

  i) The process should include: a vulnerability handling policy that identifies reporting process; responsible party for handling vulnerability reports; and a process for intake, assignment, remediation, and remediation testing.

  ii) As part of the process, a vulnerability tracking system that includes severity ratings, and metrics for measuring timing for identification, analysis, and remediation of vulnerabilities should be used.

  iii) Documentation should be reviewed and updated annually, or when significant enterprise changes occur that could impact this Process.

c) Root Cause Analysis should be performed on Security Vulnerabilities

d) An inventory of third-party components used in development must be established and updated monthly.

  i) This inventory is to include any risks that each third-party component could pose.

e) Third-party software components must be up to date and from trusted sources.

  i) When possible, choose established and proven frameworks and libraries that provide adequate security.

    ii) Components should be acquired from trusted sources or the software should be evaluated for vulnerabilities before use.

f) A severity rating system and process for application vulnerabilities should be established that facilitates prioritizing the order in which discovered vulnerabilities are fixed.
    i) This process includes setting a minimum level of security acceptability for releasing code or applications.
    ii) This system and process should be reviewed and updated annually.

g) Use standard, industry-recommended hardening configuration templates for application infrastructure components.
    i) This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components.
    ii) In-house developed software must not weaken configuration hardening of the USBE network.

h) Separate environments for production and non-production systems must be maintained.

i) Software development personnel should receive training in secure code writing for their specific development environments and responsibilities.
    i) Training can include general security principles and application security standard practices.
    ii) Training should be conducted at least annually and designed in a way to promote security within the development team and build a culture of security among the developers.

j) Secure design principles should be applied to application architectures.
    i) Secure Design Principles Include:
        (1) The concept of least privilege and enforcing mediation to validate every operation that the user makes.
        (2) Promoting the concept of "never trust user input".
        (3) Minimizing the application infrastructure attack surface.
            (a) Includes but not limited to turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.

k) Vetted modules or services for application security components should be leveraged to reduce developers' workload and minimize the likelihood of design or implementation errors.
    i) Examples include: identity management, encryption, auditing and logging, and mechanisms to create and maintain secure audit logs.

l) Static and dynamic analysis tools should be applied within the application life cycle to verify that secure coding practices are being followed.

m) All Storage and transmission of sensitive data like personally identifiable information (PII) must follow all USBE privacy rules and policies.
    i) Each Program Manager should maintain a list of the systems they manage and their approval status for sensitive data interactions.

n) Application development leadership are required to create additional application development policies, procedures, and best practices be developed following this policy, National Institute of Standards and Technology (NIST) Special Publication 800-218, and other industry best practices such as Open Worldwide Application Security Project (OWASP), SAFECode, etc.

i) These should be created with input from other stakeholders such as the Chief Information Security Officer, the Data Privacy Office, and Information Technology (IT) Operations.

3) Change History

Date	Version	Author	Changes Made / Section(s)
April 11, 2023	0.1.0	Patrick Hawkins	Initial Draft
December 8, 2023	0.1.1	Patrick Hawkins	Draft review and update