SNOWBE ONLINE Policy# SDLC-1

System Development Life Cycle Policy

Humberto Gonzalez

Version # 1.0

DATE: November 23, 2024

Table of Contents

Purpose

This policy outlines SnowBe Online's commitment to adopting and implementing a Secure Development Life Cycle Framework (SDLC Framework) based on NIST SP 800-160 r1 and the Secure Software Development Framework (SSDF) for secure, efficient, and compliant software development, deployment, and maintenance. SnowBe Online protects data confidentiality, integrity, and availability by implementing security procedures at every stage, as well as compliance with standards such as PCI DSS and NIST 800-53.

Scope

This policy applies to all system development activities carried out by SnowBe Online's internal teams and third-party vendors. It oversees the development, enhancement, and maintenance of software systems hosted on AWS or on-premises, as well as the integration of third-party apps or vendor tools and systems that handle client data, sales, and company activities.

Definitions

NIST SP 800-160 r1: Guidelines for using systems security engineering principles to design secure systems.

PCI DSS: The Payment Card Industry Data Security Standard controls secure credit card transaction processing.

SDLC Framework: An organized approach to building secure systems that ensures security is integrated at all stages.

SSDF: A framework to mitigate the risk of software vulnerabilities by integrating secure development techniques throughout the program lifecycle.

Roles & Responsibilities

IT Security Team:

- Define security requirements during the planning and design stages.
- Perform security testing, which includes vulnerability assessments and penetration testing.
- Approve deployment only after ensuring compliance with PCI DSS, SSDF, and NIST requirements.
- Conduct periodic audits of SDLC processes to ensure compliance and identify gaps.

System Development Team (Developers and Architects):

- Implement the SDLC framework and SSDF techniques in all projects.
- Integrate secure coding techniques, conduct peer code reviews, and maintain version control.
- Utilize tools such as static and dynamic code analysis platforms (e.g., Nessus for vulnerability scanning) and secure version control systems.

- Document security measures and ensure they are implemented during each SDLC step.

System Owners:

- Approve the project requirements, design documentation, and test findings.
- Align with corporate objectives, security requirements, and compliance standards.

Third-Party Vendors:

- Follow SnowBe Online's SDLC policies and SSDF recommendations.
- Provide examples of secure development and testing procedures.

Policy

1. Analysis:
    - Conduct a thorough study of the business, functional, and security requirements.
    - Determine potential hazards, compliance needs, and project objectives.

2. Planning:
    - Develop a detailed project plan, including resource allocation, timetables, and security checkpoints.
    - Incorporate SSDF techniques and PCI DSS compliance, and specific cloud-security controls (e.g., API restrictions, monitoring, and least privilege policies) into the project planning.

3. Design:
    - Develop a thorough system architecture with security mechanisms including encryption, authentication, and access controls.
    - Ensure that the design adheres to NIST SP 800-160 r1 principles and addresses identified risks.

4. Development:
    - Write and implement code using secure coding techniques (e.g., OWASP).
    - Identify vulnerabilities by using version control systems and conducting peer reviews.
    - Utilize automated tools for static and dynamic code analysis.
    - Document modifications and ensure traceability for all development operations.

5. Testing:
    - Perform thorough testing, including functional, integration, and security testing (e.g., vulnerability scans, penetration tests).
    - Use tools such as Nessus and SAST (Static Application Security Testing) tools to identify vulnerabilities.
    - Validate SSDF and PCI DSS compliance using User Acceptance Testing (UAT).
    - All vulnerabilities should be addressed and resolved prior to deployment.

6. Deployment:
    - Deploy systems in a controlled staging environment to ensure final validation.

- Pass security and functionality testing and obtain the necessary clearances for production release.
- Monitor the initial deployment for any performance or security issues.

7. Maintenance:
    - Apply patches, updates, and security upgrades on a regular basis.
    - Check systems for vulnerabilities, problems, and performance issues.
    - Store audit logs older than three months securely in the cloud.

8. Evaluation:
    - Conduct periodic audits of the SDLC procedures to detect gaps, ensure compliance with PCI DSS and NIST standards, and implement lessons learned into subsequent projects.
    - Evaluate the system's performance against security and business needs.
    - Collect user input and find opportunities for improvement.

9. Disposal:
    - Securely decommission systems by deleting or sanitizing sensitive information.
    - Document the disposal process to guarantee regulatory compliance.

Exceptions/Exemptions

In circumstances where regular security standards cannot be followed, exceptions or exemptions may be requested. To request an exception or exemption, the requester must send a formal written request to the IT department, outlining the precise security protocol that cannot be followed and the justification for the request. The request should include a reason for why the exception or exemption is required, the potential impact on security, and any proposed compensatory controls. The Chief Information Officer (CIO) or a designated senior IT executive has the power to approve or reject such requests. Approved exceptions or prohibitions will be provided for a brief period not exceeding one year unless otherwise justified. The requester must make certain that the exception or exemption is reviewed and re-evaluated before the end of the permitted term.

Enforcement

All SnowBe Online employees, contractors, and third-party entities must comply with this security policy. Failure to follow the policies and procedures outlined will result in disciplinary action, which may include verbal or written warnings for minor infractions, temporary suspension from duty for repeated or serious violations, and termination of employment for severe or repeated noncompliance. If noncompliance results in legal consequences or financial loss for SnowBe Online, the company has the right to take legal action. Contractors and third-party entities that violate this security strategy may face contract termination and legal action, depending on the severity of the breach. All disciplinary actions will be carried out under SnowBe Online policy and any applicable labor laws. Each individual must understand and follow security policies to protect the company's information assets.

Version History Table

Version #	Implementation Date	Document Owner	Approved By	Description
1.0	07/05/2024	Humberto Gonzalez	TBD	Initial SDLC Policy, outlining SnowBe Online's framework for secure system development.

Citations

Enforcement- Sample Detailed Security Policy (bowiestate.edu)

Exceptions/Exemptions- Sample Detailed Security Policy (bowiestate.edu)

NIST SP 800-160 r1- Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems

OWASP- Secure Coding Practices - Quick Reference Guide

PCI DSS- PCI_DSS-QRG-v4_0.pdf

SDLC Model- Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework (SSDF)

SDLC Template Sample- opm.gov/about-us/open-government/digital-government-strategy/fitara/opm-system-development-life-cycle-policy-and-standards/