TRIANZ℠

SECURE SDLC PROCEDURE

TRIANZ INTERNAL

trianz.com

Statement of Confidentiality

The information contained in this document is internal to Trianz. It shall not be disclosed, duplicated, or used for any purpose other than that stated herein, in whole or in part, without the express written consent of Trianz.

Information Classification

☐	Public
☒	Internal
☐	Confidential
☐	Restricted

Table of Contents

Glossary

Word/Abbreviation	Description
FEDRAMP	Federal Risk and Authorization Management Program
HIPAA	Health Insurance Portability and Accountability Act
PCIDSS	Payment Card Industry Data Security Standards
CSSLP	Certified Secure Software Lifecycle Professional
ISO	International Organization for Standardization
SOC	Security Operations Center
MFA	Multi Factor Authentication
IP	Internet Protocol
AD	Active Directory
OWASP	Open Web Application Security Project

1. Introduction

Secure SDLC is a systematic approach and structures concepts to include security at every phase of software Development Lifecycle (SDLC). It ensures the security assurance of specific activities including architecture analysis, code review, and penetration testing, all of which are integral aspects of the development effort.

1.1 Objective(s)

The secure software development life cycle is a step-by-step procedure to develop software with several objectives, including:

- Awareness of security considerations by stakeholders.
- Early detection of flaws in the system.
- Optimizing the design, deployment, and maintenance of said software.
- Cost reduction as a result of early detection and resolution of issues.
- Overall reduction of intrinsic business risks for the organization.

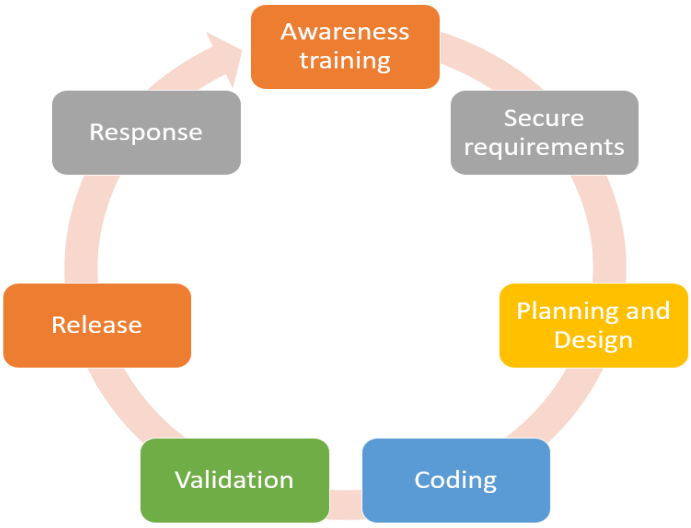
1.2 Scope

The scope includes all internal and applicable vendor applications that are owned, operated, maintained, and controlled by Trianz by Internally and Externally either on premises or on cloud.

2. Secure SDLC Lifecycle

Secure software development includes integrating security in different phases of the software development lifecycle such as requirements, design, implementation, testing and decommissioning. It includes seven phases starting from providing awareness training, secure requirements, planning and design, secure implementation and coding, validation and security review. The Secure SDLC response applied to all project methodology followed, waterfall , agile or Devops.

The way security is embedded within each phase is as illustrated in the below diagram:



2.1 Security Training:

- All Trianz employees receive internal security training. Individuals in technical roles (Architect, developers, and program managers) who are directly involved with the development of software programs must attend at least one unique security training class annually.
- Project team members are strongly encouraged to seek additional security and privacy education that is appropriate to their needs or products.

2.2 Requirements:

- Ensure security requirements must cover both functional security and Non-Functional Requirements are captured from the contractual obligation tracker.
- Ensure security requirements produce:
    - misuse cases that depict non-malicious (intentional or unintentional) and
    - Abuse cases that depict malicious (Intentional) threat based on inaccurate risk assessment/attack models.

- Ensure all security requirements are captured with respect to CIAP (confidentiality, integrity, availability, and privacy) and AAA (authentication, authorization, and accountability)
- Ensure tracking and documenting attack surfaces and any changes made to them during the development process.
- Threat modeling is carried out by which threats to the applications are discovered so they can be mitigated during implementation phase.
- The security & Privacy requirements need to be complied with legal and industry compliance requirements as ISO 27001, ISO 27701, ISO 27017/18, SOC, PCI DSS, HIPAA etc.

2.3 Planning and Design:

- Ensure Performing a Risk Analysis, or a Threat Modelling in order to ensure that all the security requirements and specifications are enough to protect against the threats identified in the threat or risk models.
- Ensure following appropriate secure-design principles as per the technology in use.
- Ensure developing security blueprint and select the technologies needed to support security blueprint.
- At this phase, a Security Testing Plan shall be developed based on the threats identified during Threat Modelling or Risk Analysis exercise
- Ensure examining legal issues.
- Ensure planning incident response section and business response to disasters.
- Check for vulnerabilities in any feature of the application.
- Ensure the Controls and remediation against the OWASP top 10 vulnerabilities, CWE/SANS top 25 programming errors that would be applicable as per technologies are in place.

2.4 Secure Code:

- Ensuring that code is developed securely and implementing the security controls identified during the design phase.
- Ensure following OWASP top 10 secure coding standards/guidelines and perform code review to all source code. It is essential to review the code and identify bugs through secure code review software such as SonarQube.
- Use the latest compiler and supporting tool versions.
- secure software is to minimize the number of unintentional code-level security vulnerabilities. This can be achieved by
    - Defining coding standards,
    - Selecting the most appropriate (and safe) languages, frameworks, and libraries,

2.5 Validation:

- Ensure that security criteria are included in every test case. Use the misuse/abuse scenarios as the basis for defining the test case strategy.
- Use the risk analysis to prioritize the tests to be performed, testing the highest risk/highest consequence items first.
- Ensure performing Fuzz testing. Fuzz tests all file formats and parser per format, each parser must be tested separately.
- Ensure testing includes DAST (Dynamic Analysis Security Testing) tools like Burp suite, Aqua Enforcer tool to test your application while in runtime. These tools can detect errors associated with user authentication, authorization, SQL injection, and API-related endpoints.
- Ensure performing vulnerability assessment, infrastructure assessment and penetration test (for all public facing applications) on the application. These

tools can detect errors associated with (Weak Credentials, Injection Attacks, Buffer Overflows, Cross-site Scripting, etc.) and Pentest has been used to test networks, host operating system configurations, and patch levels.

- Reevaluate the attack surface of the software during the testing.

2.6 Review and Release

- Ensure performing security code reviews for all high-priority code identified.
- The code review can be either manual or automated using technologies such as static application security testing (SAST).
- All threat models have been reevaluated and updated (or, for very old components or products, threat models have been created for all components).
- The attack surface has been reanalyzed, and the appropriateness of the default attack surface has been confirmed.
- Ensure before releasing the application, security review and employ security analysis tools to perform thorough penetration testing and vulnerability scanning.
- After completing all the necessary tests in runtime, send a secure build to production for final deployment.
- Ensure access restrictions are compliant as per below controls:
- Restrict the access to the application is only via like user identity specific IPs and block the application accessing from public IPs.
- Application is authenticated via Azure SSO or AD integration with MFA enabled.
- Deploy Web Application Firewall and harden the rules to protection mode.
- Ensure the application is being monitored for any security incidents and alerted.

- Ensure the security incidents are logged and incident management procedure is followed.

3. Roles & Responsibilities

Roles	Responsibilities	External/Internal
Internal Apps Team	To conduct the necessary scans and provide the reports To provide all the Internal and external application details related to design, implementation of the Trianz Applications, Client Applications (wherever applicable) Remediate the identified vulnerabilities as per the Assurance Report	Internal
IS Operations	Perform Static Analysis, Dynamic Code Analysis, Vulnerability scan on Application and provide reports	Internal
InfoSec Team	Conduct the end-to-end application security assessment Review the VA scan and Code Review report Track the closure report	Internal
CIO/CISO	Review and approve the final assurance report.	Internal
Security Engineer	Perform security risk assessment and mitigating those risks	Internal
Developer	Ensure following secure coding standards/guidelines while coding	Internal

Project Manager/Product Manager	Define the strategy for the appropriate security standards of all software applications handling sensitive information that are accessible from the organization., as well as to monitor, establish and enforce remediation timelines and sanctions for non-compliant systems.	Internal

4. Prerequisites

4.1 4.1 Training

CSSLP Training

4.2 4.2 Reference Policy, Process, Procedure, Templates, Checklist

Document Name
Secure SDLC Policy
Access control Policy
Dev Sec Ops Procedure
Vulnerability Assessment and Penetration Testing Procedure
Information Security Assessment Checklist
Cloud Security Privacy Policy
Patch Management Policy
Incident management procedure
Risk and Opportunity Management Procedure

5. Standards Addressed

Standards Covered
ISO 27001:2013-ISMS (Information Security Management System)
ISO 27017: 2015- Cloud Security
ISO 27701:2019-PIMS (Privacy Information Management System)

6. ISO Control Mapping(s)

Category of Control	ISO 27001:2022 Control	Document Name as per ISO 27001:2022
Technological	8.25 Secure development life cycle Control Rules for the secure development of software and systems shall be established and applied.	Secure SDLC Procedure
Technical Control	8.29 Security testing in development and acceptance	

Document Control

Owner:	CISO	**Release ID:**	SEC_SDLC_PROC-0162

For Trianz Process Improvement Group (TPIG) Purpose Only

Version History

Ver. No.	Date	Author	Reviewer	Approver	Reason for Change	Change Description
0.0	30-July-21	Divya G	Karthik	Phani	Initial Version	None
1.0	30-July-21	Divya G	Karthik	Phani	Request for Baseline	Baselined
1.0	03-Jan-22	Divya G	Karthik N		Annual Review	Reviewed and no changes
1.1	14-Mar-22	Kruti	Kartik		For Review	The scope has been extended to products and services
2.0	18-Mar-22	Kruti	Siva N	Siva N	For Approval	Approved and baselined
2.0	10-Mar-23	Beniyel S and Asha Veeramallu	Balu N		For review	No change
2.0	10-Mar-23	Beniyel S	Balu N	Srikanth M	For Approval	Approved and Baselined

2.1	12-May-23	Asha Veeramallu	Vijaya		For Review	Migrated to New Template
3.0	12-May-23	Asha Veeramallu	Vijaya	Srikanth M	For Approval	Approved and Baselined
3.1	11-Feb-24	Shalini	Vijaya	Srikanth M	For Review	Maoped to new ISO 27k 2022 Controls
4.0	23-Feb-24	Shalini	Vijaya	Srikanth M	For Approval	Approved and Baselined
4.1	28-May-25	Kruti	Vijaya		For Review	Migrated to New Template
5.0	29-May-25	Kruti	Vijaya	Srikanth M	For Approval	Approved and Baselined

TRIANZ℠

Contact Information

Name

Email

Phone

Thank You

infosec@trianz.com