SISL-IT-POL-Secure Application Development Policy

Version No: V 1.0

INTERNAL DOCUMENT

OCTOBER 2025

Document Control

Document Name	SISL-IT-POL-Secure Application Development Policy
Abstract	This document describes application security at Share India Group
Security Classification	Internal
Location	Share India Group– Delhi

Authorization		
Document Owner	Reviewed by	Authorized by
IT Team	Head – IT	Head – IT

Amendment Log				
Version	**Modification Date DD MMM YYYY**	**Section**	**A/M/D**	**Brief description of change**
1.0	30th October 2025	Initial Version	A	Final

Distribution list
Designated Officer (DO)
Information Security Steering Committee (ISSC)
ISMS Core Team
Auditors (Internal & External)
All users at Share India Group

Table of Content

1 Introduction

Share India Securities Limited (SISL) develops its own software as per their business requirements. An insecurely developed application may lead to data compromise and hence business loss. SISL shall ensure that its applications are developed and tested based on OWASP development and testing guidelines to ensure that the security related vulnerabilities are minimized.

Given the changing needs of the business, the corresponding business applications have also increasingly become more complex. Considering the importance and criticality of these applications, rigorous testing is of paramount importance. It is important to ensure that the developed application meets the business requirements and helps the end user in performing their tasks efficiently. Involving the business users as part of the testing procedure provides the requisite depth and perspective to ensure that the developed application satisfies the requirements. User Acceptance testing is performed to ensure that the business functionality is implemented correct, and all the requisite use cases are satisfied.

2 Policy Statement

Applications developed and deployed in SISL's infrastructure shall have controls to secure the input, processing, storage and output of data. It shall further be ensured that the applications shall be subject to a user acceptance test prior to its deployment in the production environment.

It shall be ensured that the user acceptance test must be performed by the business users to ensure that the business requirements and use cases are met. The security testing shall be performed by the IT team to ensure that all the security controls have been implemented.

Access to the application shall be restricted to authorized persons and access provided on the principle of least privileges.

3 Scope

This policy applies to

- All applications developed by SISL's and their outsourced vendors.
- All applications deployed in SISL.

4 Roles and Responsibilities

Sr. No.	Role	Responsibility
1.	Product Head	Ensure that this policy is effectively implemented
2.	Information Security Manager (ISM)	Enforce the policy
3.	Application Team	Implement and adhere to the policy and abide by it

5 Standards and Guidelines

5.1 Secure Application Development

5.1.1 Software Requirement Document

Before any software development is undertaken, a well-defined software requirement analysis document shall be prepared and approved by the stakeholders. This document shall be referenced throughout the development of the lifecycle.

The information security-related requirements shall be included in the requirements for new information systems or enhancements to existing information systems. Principles for engineering secure systems shall be established, documented, maintained and applied to any information system implementation efforts. The application development shall be in compliance with industry best practice guidelines like OWASP.

The application shall be designed and developed such that Information involved in application services passing over public networks shall be protected from fraudulent activity, contract dispute and unauthorized disclosure and modification.

Furthermore, Information involved in application service transactions shall be protected to prevent incomplete transmission, misrouting, unauthorized message alteration, unauthorized disclosure, unauthorized message duplication or replay.

5.1.2 Secure Development environment

The development environments shall be strictly controlled. System administrators shall be responsible for the security of the development environments. They shall ensure that all proposed system changes are reviewed before deployment into production so that they do not compromise the security of either the system or the operating environment.

5.1.3 Secure Coding

Secure coding principles shall be applied to software development. To ensure software is written securely thereby reducing the number of potential information security vulnerabilities in the software.

SISL shall establish organization-wide processes to provide good governance for secure coding. A minimum secure baseline shall be established and applied. Additionally, such processes and governance shall be extended to cover software components from third parties including open-source software.

SISL shall monitor real world threats and give up-to-date advice and information on software vulnerabilities to guide SISL's secure coding principles through continual improvement and learning. This can help with ensuring effective secure coding practices are implemented to combat the fast-changing threat landscape. (e.g. use of OWASP top 10)

5.1.3.1 Planning before coding

Secure coding principles shall be used both for new developments and in reuse scenarios. These principles shall be applied to development activities both within SISL and for products and services supplied by SISL to others. Planning and prerequisites before coding shall include:

- SISL-specific expectations and approved principles for secure coding to be used for both In-house and outsourced code developments.
- Common and historical coding practices and defects that lead to information security vulnerabilities.
- Configuring development tools, such as integrated development environments (IDE), to help enforce the creation of secure code.
- Following guidance issued by the providers of development tools and execution environments as applicable.
- Maintenance and use of updated development tools (e.g. compilers).
- Qualification of developers in writing secure code.
- Secure design and architecture.
- Secure coding standards and where relevant mandating their use.
- Use of controlled environments for development.

5.1.3.2 During coding

Considerations during coding shall include:

- Secure coding practices specific to the programming languages and techniques being used;
- Using secure programming techniques, such as pair programming, refactoring, peer review, security iterations and test-driven development.
- Using structured programming techniques.

- Documenting code and removing programming defects, which can allow information security vulnerabilities to be exploited.
- Prohibiting the use of insecure design techniques (e.g. the use of hard-coded passwords, unapproved code samples and unauthenticated web services).
- Testing shall be conducted during and after development. SISL may also deploy static application security testing (SAST) processes that can identify security vulnerabilities in source code.
- Before software is made operational, the following shall be evaluated:
    - Attack surface and the principle of least privilege.
    - Conducting an analysis of the most common programming errors and documenting that these have been mitigated.

5.1.3.3 Review and maintenance

After code has been made operational:

- Updates shall be securely packaged and deployed. (Using Installer)
- Reported information security vulnerabilities shall be handled;
- Errors and suspected attacks should be logged, and logs regularly reviewed to make adjustments to the code as necessary; (Investigate recurring errors or suspicious activities / during VAPT)
- Source code shall be protected against unauthorized access and tampering. (user access reviews)

If using external tools and libraries, SISL shall consider:

- Ensuring that external libraries are managed (e.g. by maintaining an inventory of libraries used and their versions) and regularly updated with release cycles.
- Selection, authorization and reuse of well-vetted components, particularly authentication and cryptographic components.
- The license, security and history of external components.
- Ensuring that software is maintainable, tracked and originates from proven, reputable sources.
- Sufficiently long-term availability of development resources and artefacts.

Where a software package needs to be modified SISL shall consider the following points:

- The risk of built-in controls and integrity processes being compromised.
- Whether to obtain the consent of the vendor.
- The possibility of obtaining the required changes from the vendor as standard program updates.
- The impacts if SISL becomes responsible for the future maintenance of the software as a
- result of changes.
- Compatibility with other software in use.

5.1.3.4 Other information

A guiding principle is to ensure security-relevant code is invoked when necessary and is tamper resistant. Programs installed from compiled binary code also have these properties but only for data held within the application. For interpreted languages, the concept only works when the code is executed on a server that is otherwise inaccessible by the users and processes that use it, and that its data is held in a similarly protected database. For example, the interpreted code can be run on a cloud service where access to the code itself requires administrator privileges. Such administrator access shall be protected by security mechanisms such as just-in-time administration principles and strong authentication. If the application owner can access scripts by direct remote access to the server, so in principle can an attacker. Webservers shall be configured to prevent directory browsing in such cases. Application code is best designed on the assumption that it is always subject to attack, through error or malicious action. In addition, critical applications can be designed to be tolerant of internal faults. For example, the output from a complex algorithm can be checked to ensure that it lies within safe bounds before the data is used in an application such as a safety or financial critical application. The code that performs the boundary checks is simple and therefore it is much easier to prove correctness.

Some web applications are susceptible to a variety of vulnerabilities that are introduced by poor design and coding, such as database injection and cross-site scripting attacks. In these attacks, requests can be manipulated to abuse the webserver functionality.

5.1.4 Application Controls Validation

Poor access control leads to unauthorized data disclosure. Access Control shall be in compliance with SISL's access control policy.

SISL shall ensure that the access controls in the applications deployed in-house cannot be bypassed or compromised.

5.1.4.1 Input Data validation

Poor or non-existent input validation leads to vulnerabilities in the applications which can be exploited such as SQL injection or OS command injection. Attackers can gain unauthorized access to information by embedding OS commands or database related SQL commands in the input data.

Data input to the application systems shall be validated to ensure that it is correct and appropriate. Checks shall be applied to the input of business transactions, standing data and parameters tables. The following checks shall be applied:

- Out-of-range values
- Invalid characters in data field
- Missing or incomplete data
- Unauthorized or inconsistent control data

All the projects involving data inputs from external interfaces such as User Interface shall include a check for input data validation. This shall be verified through testing.

5.1.4.2 Processing Validation

Data that has been correctly entered can be corrupted by processing errors or through deliberate acts. Validation checks shall be incorporated into the system to detect such corruption. The application design shall ensure that restrictions are implemented to minimize the risk of processing failures leading to loss of integrity. The checks that are incorporated shall include the following:

- Checks to ensure that programs are run in the correct order
- Terminate in case of a failure, and
- Further processing is halted until the problem is resolved in case it is found critical.

Projects shall address this point, as per project specific requirements and design.

5.1.4.3 Output Data Validation

Data output from any application system shall be validated to ensure that the processing of stored information is correct and appropriate. Typically, all the applications created within SISL shall be well tested, and the output validation required shall include:

- Plausibility checks to test whether the output data is reasonable.
- Procedures for responding to output validation tests
- Well defined responsibilities of all personnel involved in the data output process

All the projects involving data output to external interfaces such as user Interface shall include a check for output data validation. This shall be verified through testing. The test plans for the developed application shall include test cases validating output data as applicable.

5.1.5 Software Testing

The application shall be tested exhaustively for acceptable security, performance and user acceptance criteria.

A separate test environment shall be available independent of the development and production environment. The test environment shall emulate the production environment to ensure the correctness of the test results.

Each test shall be accompanied by test scenarios, test cases and their test results. Access to the test data shall be controlled and protected appropriately.

5.1.5.1 Security Testing

Application owner shall ensure that the application is tested for security controls before being deployed in the production environment. The application owner shall analyze the modules, menu options and parameters in the application for compliance to the security requirements of SISL, e.g. password policy enforcement, access control, audit logs etc.

- Security testing of the application shall be performed by the team to ensure that all security requirements for the application are fulfilled.
- OWASP Testing guidelines shall be used as a reference for web application testing.
- All the identified vulnerabilities shall be closed before porting the application to the production environment.
- In case of an emergency where all the vulnerabilities cannot be addressed, at least all the critical, high and medium vulnerabilities shall be addressed and closed.
- Security sign off shall be taken from the DO before the application is moved to the production environment.

5.1.5.1.1 Security Testing Frequency

SISL shall conduct application vulnerability testing of the developed / installed applications

- At least ones in a year
- Whenever a new release of their application has been released.
- Access to system audit tools shall be restricted.

5.1.5.1.2 Audit tools

Access to the audit tools shall be restricted. Audit tools shall be maintained separately from operational tools.

5.1.5.2 Performance Testing

Application owner shall ensure that the application is tested for peak load conditions before deployment. Multi-user application shall be tested in an environment that simulates real life conditions.

5.1.5.3 User Acceptance Testing

Acceptance testing programs and related criteria shall be established for new information systems, upgrades and new versions.

5.1.5.3.1 Test Plans

- Testing strategy shall cover test preparation, test execution.
- Test scenarios shall be developed.
- Testing schedules and the resource requirement shall also be documented.
- Proper defect tracking and reporting strategy shall be defined clearly stating the severity of the defects found.

5.1.5.3.2 User Acceptance Test Case

- Client manages and maintains their UAT programme and provide feedback.

5.1.5.3.3 User Acceptance Testing Defect Logs

- All defects identified and reported during the user acceptance testing shall be documented.

- Severity level shall be marked against each defect finding.
- For each defect finding, proper description shall be given.

5.1.6 Application Documentation

All documents relevant to the application shall be created and saved.

- Application owners shall create secure configuration documents.
- Application owner shall ensure that detailed documentation is available as part of user / system manual.
- All setting mentioned in the configuration document shall be incorporated in the application documentation.
- Documents shall be adequately backed up.

5.1.7 Data Masking

SISL shall ensure data masking shall be used in accordance with SISL's topic-specific policy on access control and other related topic-specific policies, and business requirements, taking applicable legislation into consideration to limit the exposure of sensitive data including PII, and to comply with legal, statutory, regulatory and contractual requirements.

5.1.7.1 Guidelines for Data Masking

Where the protection of sensitive data (e.g. PII) is a concern, SISL shall consider hiding such data by using suitable encryption techniques which can hide PII, disguise the true identity of PII principals or other sensitive information.

Where essential, SISL shall make use of additional techniques for data masking like:

- Encryption (requiring authorized users to have a key).
- Replacing values with their hash

5.1.7.2 Enablers when implementing data masking techniques.

SISL shall ensure the following before considering implementation of data masking techniques;

- Not granting all users access to all data, therefore designing queries and masks in order to show only the minimum required data to the user; there are cases where some data shall not be visible to the user for some records out of a set of data; in this case, designing and implementing a mechanism for obfuscation of data (e.g. if a patient does not want hospital staff to be able to see all of their records, even in case of emergency, then the hospital staff are presented with partially obfuscated data and data can only be accessed by staff with specific roles if it contains useful information for appropriate treatment);
- When data are obfuscated, giving the PII principal the possibility to require that users cannot see if the data are obfuscated (obfuscation of the obfuscation; this is used in health

facilities, for example if the patient does not want personnel to see that sensitive information such as pregnancies or results of blood exams has been obfuscated);
- Any legal or regulatory requirements (e.g. requiring the masking of payment cards' information during processing or storage).

5.1.7.3 Considerations before implementing data masking techniques.

SISL shall consider the following when using data masking, pseudonymization or anonymization

- Criticality of data to be masked.
- Access controls to the processed data.
- Agreements or restrictions on usage of the processed data.
- Keeping track of providing and receiving the processed data.

5.2 Capacity Planning

Application owner shall identify the system requirements prior to the deployment of the application in the production environment. The current and future requirements shall be considered.

This shall include

- Hardware requirements like processor, hard disk capacity, etc.
- Software requirements like supporting OS and application software
- Bandwidth requirement

The capacity planning shall be in compliance with SISL's capacity management policy.

5.3 Change Management

Changes to systems within the development lifecycle shall be controlled by the use of formal change control procedures and shall be in compliance with SISL's Change Management Policy.

5.4 Incident Management

All incidents covering application security breach shall be logged and governed by SISL's Incident Management Policy

6 Reference

Ref: SISL-IT-PRO -Application Development and Maintenance Procedure

7 Policy Review Frequency

The policy shall be reviewed annually or when there is a major change within Share India Securities Limited environment.

8 Policy Exception

In case of any deviation against policy guidelines, Risk Acceptance form should be submitted to Designated Officer for approval.

9 Policy Violation Reporting Matrix

Any violation to the policy should be reported to Reporting Manager/Designated Officer

Level	Designation
Level 1	Employee's Reporting Manager
Level 2	Designated Officer
Level 3	MD & CEO