INSERT LOGO

SECURE DEVELOPMENT POLICY

Commented [A1]: This policy is deliberately not very verbose. It has been rationalised in terms of words and sentences to make it as easy to read as possible.

Classification: **Public**

This document may be shared with interested parties outside of [Company Name]

Version	Approved By	Owner	Date Last Updated	Review Frequency	Next Review	Comments

Contents

Objective

This Secure Development Policy aims to establish a comprehensive framework for integrating security into the software development lifecycle.

This policy ensures that all software development activities within the organisation prioritise security to protect our systems, data, and users from potential threats.

Scope

This policy applies to all employees, contractors, and third-party vendors involved in the software development process, including planning, design, coding, testing, deployment, and maintenance.

Policy

1. Secure Development is Everyone's Responsibility

- All team members must understand and embrace their role in ensuring the security of the software they develop.
- Security training and awareness programs will be provided to all employees involved in development activities.

2. Continuous Security Knowledge Enhancement

- Developers must stay updated with security trends, threats, and best practices.
- Regular training sessions and access to security resources will be provided to ensure continuous learning.

3. Production of Clean & Maintainable Code

- Code must be written following established coding standards and best practices to ensure it is clean, maintainable, and secure.
- Code reviews and pair programming practices should be implemented to identify and mitigate security vulnerabilities early in development.

4. Secure Development Environment

- The development environment must be secured to prevent unauthorised access and potential security breaches.
- Secure communication channels must be used, and access to development resources should be controlled and monitored.

5. Protection of Code Repositories

- Version control systems must be used to manage code securely.
- Access controls, code signing, and regular audits must be implemented to protect code repositories from unauthorised access and modifications.

6. Secure Build and Deployment Pipeline

- Automated build and deployment processes must incorporate security checks at every stage.
- Tools for static code analysis, dependency checking, and other security measures must be integrated into the pipeline.

7. Continuous Security Testing

- Regular security testing must be conducted, including static and dynamic analysis, penetration testing, and vulnerability scanning.
- Security testing should be ongoing throughout the development lifecycle to promptly identify and address vulnerabilities.

8. Planning for Security Flaws

- A proactive approach must be taken to anticipate and plan for potential security flaws.
- A robust incident response plan must be established and regularly tested to ensure quick and effective remediation of security breaches.

9. Data Masking Techniques

To limit the exposure of sensitive data, including PII, through the use of data masking techniques.

Techniques

- **Substitution**: Replace sensitive data with fictitious but realistic values.
- **Shuffling**: Randomly reorder data within the same dataset to mask original values.
- **Redaction**: Remove or obscure parts of data fields to prevent exposure of sensitive information.
- **Encryption**: Encrypt sensitive data and ensure proper management of cryptographic keys.

• Ensure data masking techniques are integrated into the secure development lifecycle, particularly during data processing, storage, and transmission stages.

Implementation and Compliance

Establishing a Security-Friendly Culture

- Encourage open communication about security issues and integrate security objectives into overall development goals.

- Foster a culture where security is prioritised, and every team member feels responsible for maintaining it.

Periodic Re-evaluation and Updates

- Continuously assess and update security practices to adapt to new threats and technological advancements.
- Regular reviews of this policy and related procedures must be conducted to ensure they remain effective and relevant.

Handling Sensitive Data

- Additional measures for projects handling highly sensitive information must be taken, including seeking specialised advice and implementing stricter security controls.

Roles and Responsibilities

Development Team

- Follow secure coding practices and participate in security training programs.
- Conducted code reviews and integrated security checks into the development process.

Security Team

- Provide security training and resources to development teams.
- Conduct regular security assessments and audits of development practices.

Management

- Support and enforce the secure development policy.
- Allocate resources and tools for security training and continuous improvement of security practices.

Exceptions

Any exceptions to this policy must be approved by the Chief Information Security Officer (CISO) or an authorised delegate. Exceptions will be granted only in cases where a compelling business justification and appropriate mitigating controls are in place.

Enforcement

Failure to comply with this policy may result in disciplinary action, including termination of employment or contract. Regular audits will be conducted to ensure compliance with this policy.

Review and Revision

This policy will be reviewed annually and updated as necessary to address emerging threats, changes in technology, and improvements in security practices.