Devolutions' Secure Software Development Practices

Devolutions

DEVOLUTIONS' SECURITY PROGRAM INCLUDES A FORMAL SECURE SOFTWARE DEVELOPMENT POLICY

Devolutions' security program includes a formal secure software development policy, which governs all security aspects of the organization's software development practices. Our approach is based on the Secure Software Development Framework (SSDF), which was developed by the National Institute of Standards and Technology (NIST). SSDF defines a set of security standards and practices for integration in the software development life cycle (SDLC).

This document describes the security practices that are integrated in our SDLC, focusing on seven core aspects:

1. **Roles and Responsibilities**
2. **Expertise and Training**
3. **Control Environment**
4. **Technologies**
5. **Internal and Third-Party Activities**
6. **Privacy by Design**
7. **Reporting and Fixing Vulnerabilities**

1. Roles and Responsibilities

Delivering secure and reliable software can only be achieved through a solid plan for secure development practices. This plan is one of the highest priorities in our organization, and it is governed under the leadership and responsibility of the Executive Committee; the Chief Security Officer; the Director of Legal Affairs, Risks and Privacy; and the Secure Software Development Practice Lead. Each role provides valuable input that contributes to the overall success of the program. Here is an overview of their respective mandates:

Role	Mandate
Executive Committee	The Executive Committee is committed to providing secure and reliable software to our customers. Its members provide all necessary support and resources to effectively run and govern the program.
Chief Security Officer (CSO)	The Chief Security Officer is responsible for designing, implementing and monitoring security policies, procedures and controls, which are collectively required to achieve the organization's goals. The CSO oversees all security initiatives, reports to the Executive Committee, and provides required support to the Secure Development Practice Lead in carrying out a secure SDLC program.

Director of Legal Affairs, Risks and Privacy	The Director of Legal Affairs, Risk and Privacy is responsible for implementing policies, procedures and controls regarding the management and the protection of personally identifiable information (PII), in compliance with all applicable laws and regulations. The Director reports to the Executive Committee.
Secure Software Development Practice Lead	The Secure Software Development Practice Lead collaborates with the CSO and oversees establishing and improving the SDLC program, as well as facilitating its adoption across the organization.

2. Expertise and Training

A successful SDLC program requires appropriate expertise and skills. Our organization invests in annual training and skills development programs. Specifically:

- Our Security Team is comprised of professionals who hold trusted and globally-recognized certifications, which are bestowed by leading organizations such as GIAC, (ISC)2 and Offensive Security. Members of the Security Team are encouraged to attend security-related events, such as conferences and workshops worldwide, and we sponsor their participation.
- Our Developer Team regularly attends mandatory in-house trainings that cover a wide range of security topics, such as common vulnerability types and how to avoid them, general security guidance and principles, use of cryptography, etc. Training is provided in various formats, such as online interactive courses, live sessions, and "lunch and learn" sessions. Attendance and completion of all training is monitored by our organization, in order to ensure that all employees comply with expectations and requirements.

3. Control Environment

The control environment is vitally important, as it directly impacts the success of the secure SDLC program, as well as the ongoing assurance of quality and security. Below are some of the controls that we currently have in place:

- The environments for development, staging and production are kept separate. This isolation is enforced to ensure that untested and unauthorized code does not reach production. Transition of code between

environments is limited to the Operations Team.

- Our change control process requires code to be peer-reviewed, tested and approved before being deployed to production.
- Data used in development and staging environments is artificial or anonymized, and does not contain any production data.
- Code integrity is provided by source control technologies, and all changes must go through the code review process before it is merged with the code base.
- All access and actions performed in the production environment are monitored and periodically audited.

4. Technologies

The selection and implementation of technology are a major factor in software quality, and this helps us prevent and address threats. Our organization only selects technologies that are known to provide security, stability and reliability. Specifically:

- We endorse the use of programming platforms that provide memory safe operations and enforce security over traditional C/C++. Examples of software in our environment that meet this standard are Managed C#, Rust and Typescript.
- We enforce compilation security features such as GS, SafeSEH, NX and ASLR, in order to harden applications against vulnerabilities.
- We assure product integrity with code signing technology for all publicly-released software.

5. Internal and Third-Party Activities

Our organization continually assesses the security of produced software through manual and automated activities. The methodology and approach are based on documented and widely recognized standards and guidelines, which are published by OWASP and NIST. Additionally, the Director of Legal Affairs, Risks and Privacy conducts activities related to the protection of privacy. An overview of key activities is as follows:

Activity	**Description**
Threat Modeling	This activity involves brainstorming and analyzing potential threat scenarios to evaluate their respective impact. The goal is to ensure that adequate controls are present, in order to prevent or mitigate the identified risks as soon as possible in the SDLC. The chief benefit of this activity is to help identify structural issues in the design of an application, which greatly reduces the cost of performing changes and the likelihood of vulnerability exposure.

Security Code Review	This activity involves the Security Team using manual and automated techniques to identify potential vulnerabilities directly in the source code. Code reviews are performed to ensure that security controls (e.g. authentication, authorization, encoding and filtering, logging, etc.) have been properly implemented to identify and prevent vulnerabilities from reaching production.
Static Analysis	This activity uses an automated code analysis tool that inspects source code without executing it. Security issues can be identified by tracing inputs from sources to sinks, while also identifying code paths where user data validation and filtering is missing. The chief benefit of this activity is that it can be implemented directly in the integrated development environment (IDE) and continuous integration (CI) toolchain, in order to prevent exposure of vulnerabilities as developers are coding and building their code. This ultimately limits vulnerabilities from being leaked in releases.
Fuzzing	This activity is a testing method that evaluates the robustness of a program when exposed to invalid or malformed data. Combined with program coverage information, fuzzing can be highly effective at identifying improper memory handling issues in low-level languages such as C and C++.
Penetration Testing	This activity is performed by the Security Team and third-party auditors to simulate targeted attacks on software. The goal is to validate the effectiveness or absence of security controls in software by identifying vulnerabilities and attempting to exploit them. Penetration Testing is seen as the last line of defense against allowing vulnerabilities to reach production.
Compliance	Third-party audits are performed by trusted partners periodically to assess our organization's commitment to compliance programs, such as SOC2 and GDPR. Various processes, policies, standards and technical controls that are designed and implemented in software and services are validated by reputable auditors, which provides transparency of security practices. Combined with the other activities, this provides a high level of trust and assurance for our customers and partners.

6. Privacy by Design

Under the supervision of the Director of Legal Affairs, Risks and Privacy, the integration of Privacy by Design in our development process starts at the top through our Executive Committee's commitment to security, which is documented in the Devolutions Information Security Policy. Collection of personal information is carried out only when:

- There is a clearly defined business need.
- The data is exchanged and stored securely.
- Access is limited on a need-to-know basis.

Our organization conforms to the requirements of the EU General Data Protection Regulation (GDPR) and the Personal Information Protection and Electronic Documents Act (PIPEDA). By applying appropriate privacy principles and security measures from the earliest stages ("left shift") of the SDLC, the personally identifiable information (PII) attack surface is significantly reduced.

7. Reporting and Fixing Vulnerabilities

Having a defined vulnerability reporting and remediation process is critical to the success of our organization's secure SDLC program. This commitment reduces opportunity for attackers to exploit a security issue that could impact our organization, our customers and our partners by enabling us to rapidly produce and implement security patches.

Users and security researchers are encouraged to report security issues to security@devolutions.net whenever an issue has an impact on confidentiality, integrity or availability. To speed up the resolution process, the following information should be included in the report:

- Proof-of-concept code and/or relevant screenshots to help us confirm and reproduce findings.
- An explanation of how the issue may affect the organization and/or customers when exploited.
- A proposed fix (if possible and when applicable).

Upon receiving a report, our Security Team will:

- Reproduce and confirm the vulnerability as described in the report.
- Establish a severity score according to CVSS 3.0.
- Consider the recommendations from the report and build an action plan with relevant teams.
- Maintain communication with the user or security researcher until the case is resolved.