SAFETICA SECURE DEVELOPMENT LIFECYCLE

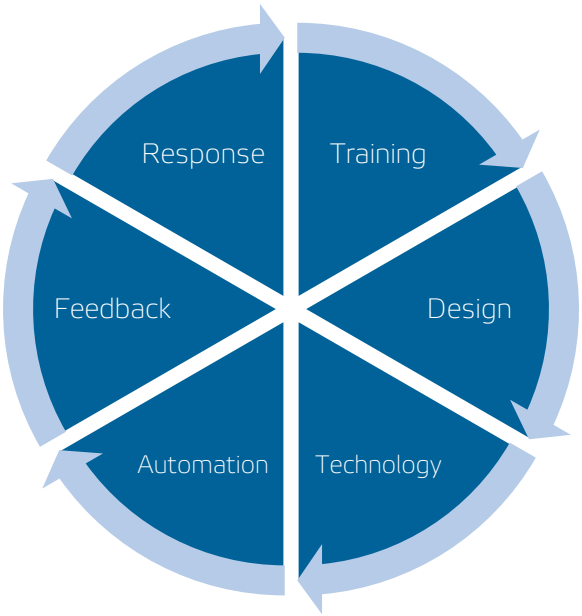
www.safetica.com

safetica®

Software development, especially security software development, is a complex problem. Customers' environment integrity and security is our top priority, so we in Safetica developed and strictly follow advanced security development process which is repeatable and measurable. It helps us to address the complexity of data loss prevention software engineering and to be and stay the trustful business partner.

The key points of our Safetica SDL are:

- Training
- Design with security focus
- Selection of technology partners
- Automated verification
- Feedback
- Quick response



TRAINING

Building a secure software and implementing it right into customers' environments begins with intensive security training and security information availability. To create a secure software design, to implement it and to deliver it right needs to be based on stable ground. We choose and hire professional and domain experts, but information market is evolving really fast, so we need to be evolving fast too.

1. Our developers are highly educated professionals and we lead them to watch technology and security trends and to spread this information into the rest of the company. We ensure the information flow between them not only on these technology and security trends but on all mistakes we made in the past and how to avoid them in the future.

2. All new developers have senior mentors who are ensuring fast learning curve in adoption of our processes and in knowledge and also to ensure the high standard of our development process.
3. The internal security study and internal information flow is the cornerstone of our security development and we have regular external security and technology trainings, especially for newly adopted technology and development trends.
4. Building the software is only the first step. We have strong focus on the deployment and implementation process in customers' environment. All our partners and delivery experts need to pass our training and receive appropriate certification. We arrange regular webinars and trainings for each new major version of our solution.

DESIGN WITH SECURITY FOCUS

Software development can provide top level of security only if all features and architecture is designed with security focus in the first place. We have regular trainings of the development team and we closely follow thoughtful development processes.

1. We lead our software developers and product architects to keep the security aspects in their mind. There are the synchronization and planning meetings as a control mechanism where the team presents their solution and one of the mandatory points is the security aspect. The rest of developers and architects provide a qualified feedback and an expert review.
2. When we design any solution, we follow internal rules as the security best practice:
    a. No security by obscurity!
    b. No deprecated technologies, encryption algorithms and hashes.
    c. Keep updated.
3. We keep continuous quality level by using agile development with small iterations. To ensure the level of quality, there is a fully tested version as result of the iteration.
4. Every source code change in the product must be reviewed by second developer. This process is enforced by our development ticketing system.
5. There are external domain expert review of our security architecture for the new product components.

SELECTION OF TECHNOLOGY PARTNERS

Security of any solution is equal to the security of its weakest point. Our solution adopts several 3rd party technologies and we have very strict rules about using them.

1. We co-operate only with stable partners with security focus and defined Security Development Lifecycle.
2. We require defined SLA for critical and security bug fixes.
3. Key security technologies need to be provided with source code for inspection purpose and to decrease dependency risk. For critical technologies we consider all technologies around encryption, technologies with network communication and technologies changing behavior of the 3rd party applications in the customers' environment.

4. For open source technology we choose only the worldwide-used components with strong community inspection.
5. All 3rd party technologies need to have regular updates. Every version of our solution consist only from latest stable versions of these components and they need to pass our semi-automated verification process.

AUTOMATED VERIFICATION

We are only humans and we all do mistakes. Even our solution is based on protection against human error. We are aware of it. So, we developed robust automated framework for verifying our software. It's based on multiple layers with quality and security focus. To avoid human error we even automate our release process and enforce passing critical automated verification on every released version.

1. Our development and building tools are set to do code analysis and to alert any issue.
2. Continuous Integration process runs whole set of Unit tests, Integration tests and End-To-End tests on every new build. Each new component and each fixed bug needs to have automated test if it's possible. We enforced this even in our development ticketing system as mandatory part of every change.
3. We have semi-automated release process to avoid human error during release. Release manager cannot skip any of important automated verifications.

FEEDBACK

For what we are doing and what we are trying to achieve in data security is a feedback as the most critical essence. We process and answer all partners' and customers' feedback. We also go out and ask for this feedback ourselves. We take seriously every concern and verify any reported possible security weakness in our product.

1. The key part of the feedback processing is our dedicated product management team connected with our well trained support and delivery department. They are co-operating directly with Quality Assurance and Development departments and reflect feedback into roadmap with regular releases or escalate it into security and quality hotfixes.
2. To reach the best possible technology feedback in the earliest phase we audit parts of our solution by university researches and domain experts.
3. The last but not least quality and security checkpoint before a public release is the early access Beta program. This program includes universities, regular companies, but even our company itself. Each new version needs to be run in those environments, the feedback needs to be collected and the implemented scenarios need to be approved.
4. Once the version is released we closely watch it's health together with customers who chose to be connected into our Customers' Technology Improvement Program (CTIP). CTIP sends anonymized telemetry information about unusual behavior of our

solution and helps us to detect and proactively solve potential issues and improve the environment compatibility.

5. We motivate our community experts to report all potential security issues by Security bug hunting program. Domain experts can join it, make penetration test on our solution and obtain reward for reported issues. We support and appreciate such activities.

QUICK RESPONSE

We optimize our release process to provide as fast as possible response relevant to an issue severity. For the most critical issues we react in hours thanks to stream updates of definitions. For the rest, we have processes for hotfixes, minor versions and regular major updates. The process of prioritization and potential escalation of the issues is driven by dedicated Quality Assurance manager.

1. Our priority is to keep customers' data and environments secured. We are trying to provide work around the issue as the fastest solution if it's possible. Then we run an escalation process, investigate the issue and create hotfix or minor version with permanent fix if it is possible to solve it.
    a. If it is not possible to solve it we report it in Known issues document released with each new version.
2. The quickest reaction can be done by our streaming definition updates. We strongly recommend not to turn these updates off. This type of update doesn't modify software components so there is a minimal risk of an environment interference. You can consider it as a antivirus definition, but in our case it consist of application- and web-categories, integration definitions into environments, and detection algorithms. For example, this type of update channel allows us to react on environment changes caused by 3rd party software changes, operation system updates or new software conflict detections.
3. If a software update is needed to solve an escalated issue then we provide hotfixes. After an hotfix validation we provide a minor version which consists of previous hotfixes and several small improvements. Customers and partners are notified about new minor versions by standard update channel integrated into our solution.
4. At least twice per year we release major updates. A major update consist of all hotfixes and improvements from minor versions and new features and enhancements following market trends and customers' feedback. We strongly recommend our customers to update within 6 months of a major release. We support two latest major versions or versions old one year. We cannot guarantee the security integrity and environment compatibility for older versions.