The Secure Software Development Lifecycle at SAP

Table of Contents

3 Security Activities in SAP Software Development
3 Security Training
3 Security Risk Assessment
5 Security Planning
6 Secure Development
7 Security Testing
8 Security Validation
8 Security Response
9 Managing Security and Building Trust
For software development projects, we at SAP have implemented a secure software development lifecycle (secure SDL), providing a framework for training, tools, and processes. As security is in the vital interest of anyone who is using SAP® products to run critical business processes and to store and process sensitive data, secure products are a prerequisite for secure operations.

Following the secure SDL is an undisputable requirement for all product teams at SAP, whether the products are provided for on-premise use or in the cloud. This document gives an overview of the secure SDL. We emphasize the provisioning stages of preparation, realization, and transition, as well as the operation management stages of utilization and maintenance, as defined in the ISO/IEC 27034-1 standard, “Information technology – Security techniques – Application security,” part 1. In this sense, the secure SDL covers the processes describing how security is integrated into software creation and maintenance.

Security is a primary concern for any global company, and as such, your company expects solid and secure products and cloud offerings that you can rely on for your businesses. Consequently for SAP, we have to address security in all phases of the software development lifecycle for security to be effective (see Figure 1). We use a well-selected combination of methodologies, guidelines, processes, and tools to master this complex topic. We properly enforce and continuously improve those guidelines and tools as technology advances and the environment and threat landscape keep changing.

These activities are largely embedded in our software development lifecycle (which is certified to ISO 9001:2008) and in other corporate processes, such as HR (education and learning), product support, and cloud operations.

**Figure 1: Security Development Phases in the Secure Software Development Lifecycle**

<table>
<thead>
<tr>
<th>Start of standard development</th>
<th>Release decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
<td>Transition</td>
</tr>
<tr>
<td>Development</td>
<td>Utilization</td>
</tr>
</tbody>
</table>

- Security research
- Security training
- Security risk assessment
- Security planning
- Secure development
- Security testing
- Security validation
- Security response
Security Activities in SAP Software Development

SECURITY TRAINING
Security is a cultural and organizational matter for a global company like SAP where all employees need to be aware of and embrace security needs. For SAP as a provider of market-leading enterprise application software, security awareness and regular role-specific trainings are mandatory for all roles contributing to the creation and maintenance of our software products.

For product managers and development-supporting roles, this means you need knowledge and awareness about threats, common vulnerabilities, and attack patterns. Knowing how to apply methods for threat modeling and security risk assessment helps these workers derive and decide about the security needs of applications and plan corresponding requirements and application security controls. For architects and developers, the security trainings provide knowledge about how to design for security and write secure code. Developers and quality assurance engineers learn about appropriate security test methods and tools.

In addition, SAP runs a dedicated training curriculum for developing the role of security experts. During this training, the participants acquire or extend the necessary topic-matter knowledge and skills to support their teams during product creation and operation.

SECURITY RISK ASSESSMENT
SAP follows a risk-based approach to efficiently achieve security within economic boundaries, taking time and cost of product provisioning and operations into consideration. This risk-based approach facilitates targeted security investments addressing identified risks in the context of a particular SAP solution. At the beginning of a new software development cycle, product teams first conduct a security risk assessment, during which they analyze and evaluate identified risks. The security activities that the teams plan and execute later in the development lifecycle follow the results and decisions from the security risk assessment.

To perform a proper security risk assessment, product teams have to precisely know the assets that are managed by the product and create transparency for them and for the context in which the assessment takes place. Such assets can be not only data but also business processes as described by application specifications.

It requires expertise to identify security risks from potential threats, assess these risks, and make decisions on how to treat each risk. At SAP, product teams benefit from applying corresponding methodologies that have evolved over years within the company, with threat modeling being the most effective. Developers use threat modeling in two different variants at SAP.

Following the secure SDL is an undisputable requirement for all product teams at SAP, whether the products are provided for on-premise use or in the cloud.

The first variant – product-level threat modeling – applies threat modeling to the full product scope and architecture, comprising all parts and components, including self-developed but also open-source, third-party, freeware, outsourced, and acquired components. It is an “in-breadth” approach to quickly get an overview about the main threats applicable to the product and the security risks associated with these threats. Product teams typically perform this in-breadth threat modeling before starting development of a new product or when planning major revisions of an existing product.

The second variant of threat modeling – scenario-level threat modeling – is closer to the traditionally known threat-modeling approach, and product teams apply this variant for “in-depth” analysis of a particular product’s components and supported scenarios. Typically, the in-depth threat modeling of selected critical scenarios often is an outcome of the in-breadth threat-modeling approach.

If a product stores or processes personal data, developers also conduct a privacy-impact assessment for the affected components. This method focuses on regulatory risks associated with data privacy.

After performing a security risk assessment, a product team makes decisions about which risks it will mitigate and manage further. This happens during the security planning phase. Figure 2 provides a comprehensive overview of security risk management phases.
Conducting security risk assessments is a mandatory process for all standard SAP software products following the risk-based secure SDL, independent of the supported deployment models.

SECURITY PLANNING

On the basis of the results of the security risk assessment, the product team derives the security requirements applicable to the product to mitigate the risks. For each applicable requirement, the team defines a suitable security control, which consists of a security activity, a verification measurement, and the time to apply it (see Figure 3). The product’s security plan encompasses all security controls that the product team decides to complete.

We can group security controls mainly into two categories. First, security functions are those functions that the product team implements in order to enforce security inside the software or that are used from an underlying application platform. Examples include:

- Authentication and authorization functions enforcing access control
- Data encryption during transfer and at rest
- Integrity protection and message authentication codes
- Secure session management and request forgery and click-jacking protection
- Logging of security events and data access

Figure 3: Threats, Risks, Requirements, and Controls
A second set of possible security controls is defined by what a product team decides to do to prevent vulnerabilities in the product and to achieve **secure functions**. For example, the development teams make sure that all input is appropriately validated, that memory cannot be corrupted, that output is appropriately encoded, or that privileges cannot be escalated in case of errors.

Each security control includes one or more verification measures. Threat modeling for individual components and scenarios helps verify that no significant threats have been overlooked in the security risk assessment. Examples of verification measures include:

- **Architecture and code reviews** that help to verify that the product team has put the security controls at the right places and implemented them correctly
- **Static code analysis** that can identify paths in the code where nonvalidated input finds a way to output, can be injected into code or database queries, or can cause memory corruptions
- **Dynamic security testing** that can reveal unprotected access paths, indirect object references, or unforeseen error situations leading to privilege escalations
- **Penetration tests** that can affirm the expected security status or uncover additional attack paths

Product teams find a supporting library of security requirements and security controls within SAP’s internal “Product Standard Security.” This library contains a large set of security requirements that help product teams mitigate security risks, as well as find and select suitable solutions and appropriate ways to verify them. Collected and maintained over years, the library builds on SAP’s experience as a provider of enterprise and cloud applications but also incorporates content from valuable public sources, such as OWASP, SANS, CWE, and others.

Creating and maintaining a security plan is a mandatory task for all standard software products from SAP. Additionally, product teams plan for security response, which is the process to handle security vulnerabilities reported by external sources as soon as the product has been released.

**SECURE DEVELOPMENT**

During the development phase, product teams design and implement a product’s specified functionality and nonfunctional qualities. The teams apply principles of secure design, such as “fail securely” in case of errors, “secure by default,” “never assume trust,” “least privilege,” and “check authorization close to the resource.” They use secure programming techniques, corresponding to the necessary verification measures.
libraries, and tools that help avoid security flaws during implementation. The goal is to implement secure functions for the whole product, including its application functions and its security functions, that do what they are supposed to do but do not contain vulnerabilities that can be attacked.

In this phase, the product team applies security controls as contained in the product’s security plan, such as the controls that are implemented in the software itself. In addition, developers perform design reviews, code reviews, additional threat modeling, and static-code analysis. It is crucial that the development teams follow the security plan, including plans for open-source, third-party, freeware, outsourced, and acquired components.

SECURITY TESTING
The product team performs further verifications of the implemented security controls by security testing, following the security test plan that the team has created as part of the product’s security plan. The secure SDL stipulates an approach for security testing that intelligently combines static and dynamic testing methods and tools.

Today, static application security testing (SAST) tools are available for almost all programming languages used at SAP. Whenever possible, the developers integrate these tools directly into their tool environment and use them daily. If this is not possible, the project team sets up daily or weekly runs of static-code analyzers and feeds the results back to the developers for immediate audit and analysis during the development phase. The runs enable audited results to be automatically carried over to subsequent source-code scans. The fact that such static analysis runs in an automated way enables developers to process vast amounts of code and to potentially find many issues of certain classes.

In addition, project teams plan and execute dynamic application security testing (DAST). Corresponding tools help developers and quality engineers dynamically traverse individual parts and scenarios supported by the product, observe the actual behavior of the application, and potentially identify further security deficiencies. These tools are particularly useful for testing the interaction and integration of components implemented in different languages or for including components that are available only as binaries.

The security test plan of a product typically contains a combination of SAST, DAST, and manual testing activities.
SECURITY VALIDATION
Product security validation at SAP is your advocate for holistic product security. Before we release and ship software to customers, a team from the SAP Global Security organization performs final validation of the software. This validation helps ensure that each standard SAP software product is up to the challenges of real-life deployment. The security validation team operates independent of the development teams and product-provisioning units.

Security validation checks the mandatory security report against the product team’s original security plan as well as against the product’s security risk assessment report. In addition, security validation checks the security response plan available for the product.

The security validation team also runs its own security tests. The amount and scope of these tests vary depending on the criticality and potential impact of security defects in the product. Testing can range from a pure process review to several days of active validation and penetration testing.

Running through the process of security validation is a mandatory step for all products developed within SAP’s software development lifecycle to successfully pass the “release decision” milestone.

SECURITY RESPONSE
We at SAP know that reducing the number of security vulnerabilities is key when developing secure products. However, even the best security assurance measures during development cannot guarantee complete absence of weaknesses or defects, in particular not against threats or insights arising after the release of a product. Consequently, this means that SAP has a vital security response process.

After the release of a product, or any extension or modification of it, the product team needs to be prepared for vulnerability reports received during use. In such a case, we must have contacts and technical skills available immediately to triage and investigate vulnerability reports and either confirm or reject the vulnerability. For a confirmed vulnerability, we must provide a security correction in time.
The product security response team at SAP helps ensure high-quality mitigation of the risk of security vulnerabilities in shipped SAP software. This comprises the following:

• Management of a responsible disclosure of vulnerabilities in SAP software reported by external sources such as security researchers and hackers
• Facilitation of the monthly “Security Patch Day”
• Crisis management for issues such as breaches involving SAP software

Vulnerability reports can be submitted by customers, partners, researchers, or anybody else through SAP’s online support tools or through PGP-encrypted e-mail. Required links are provided on SAP’s public Web site.

MANAGING SECURITY AND BUILDING TRUST

Teams from a variety of organizations within SAP help improve the security of SAP products using the secure SDL methodology. According to the Software Assurance Forum for Excellence in Code (SAFECode), “Software assurance is not achieved by a single practice, tool, or checklist; rather it is the result of a comprehensive secure software engineering process.”

5. “Principles for Software Assurance Assessment – A Framework for Examining the Secure Development Processes of Commercial Technology Providers,” 2015. SAFECode is a global, industry-led effort to identify and promote best practices for developing and delivering more secure and reliable software, hardware, and services.