Trellix Product Security Practices

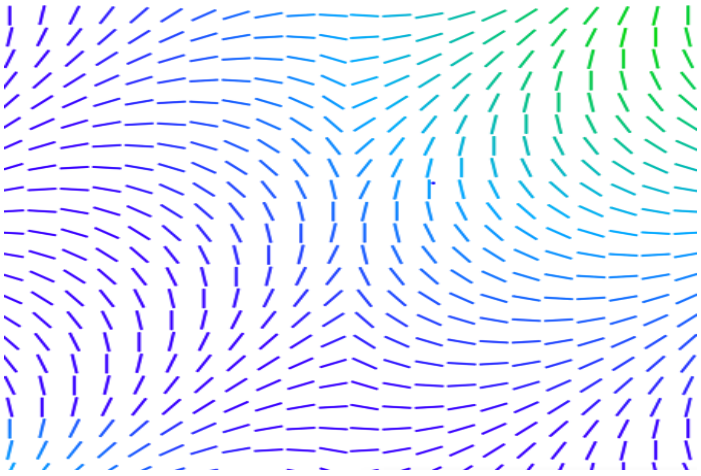


Table of Contents

Importance of Security

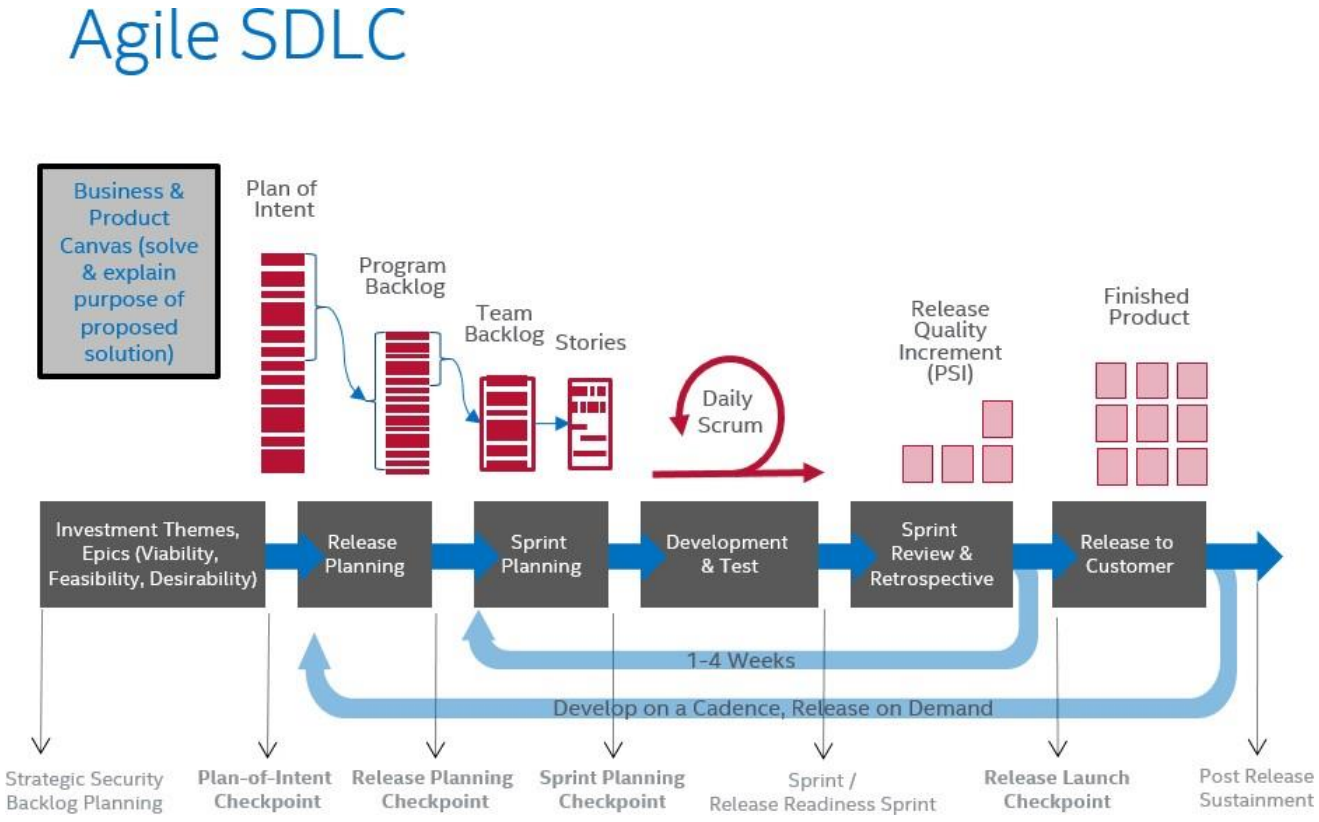
At Trellix (the "Company," "We," "Us," "Our"), we take product security very seriously. Our practices include designing for both security and privacy, in product software, IT applications, and cloud services. We have rigorous software security policies and processes designed to proactively find and remove software security defects such as security vulnerabilities. We understand that our products, IT applications, and cloud services must not only fulfill the stated function to help protect our customers, the Company software itself must also aim to protect itself from vulnerabilities and attackers. We strive to build software that demonstrates resilience against attacks.

We also understand that our customers may, from time to time, wish to review our software security practices so that they may make their own risk-based decisions on how best to use our products and to fulfill any due diligence responsibilities they may have.

Specific policies and practices can vary by product. The summary of practices described in this statementapplies to all Company branded products as well as customer facing IT and Web applications.

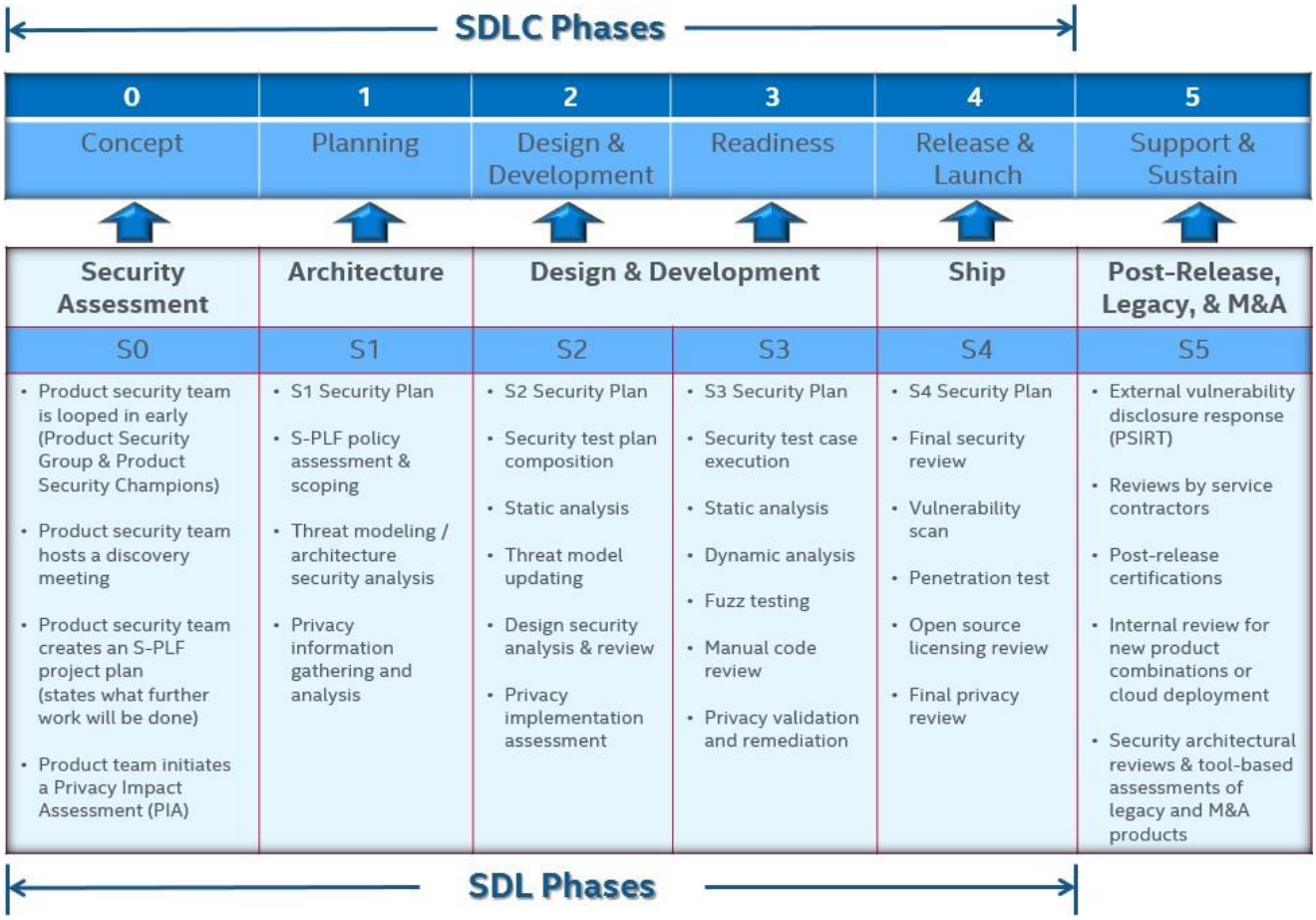
Software Development Lifecycle (SDLC)

All of software is developed using the Agile or Continuous Integration / Continuous Delivery (**CI/CD**) methodology. These agile and CI/CD practices are referred to as the Agile Software Development Lifecycle (**SDLC**). The Waterfall methodology is no longer used here. We believe the SDLC is referred to internally as the Product Lifecycle Framework (**PLF**) v2.



Development Methodologies

The chart below was developed for a traditional Waterfall SDLC. This chart has been adapted and redefined for Our Agile SDL, which includes CI/CD. Security and privacy tasks are integrated into Our SDL as a seamless, holistic process designed to produce software that has appropriate securityand privacy built into it.



While the following description may appear to apply only to Waterfall development, the same set of security tasks are performed across the iterations of Agile just as they are performed in discrete phases during Waterfall. For CI/CD, SDL activities are determined by certain triggers which are set by milestones, events, and time intervals. We encourage full engagement by software security architects and engineers within Agile sprints to ensure that security and privacy are integral parts of the Agile process.

Security Development Lifecycle (SDL)

In line with IT and application development industry standards such as ISO/IEC 27001, 27002, and 27034,BSIMM, and SAFECode, the Company software development has processes designed to adhere to a Security Development Lifecycle (**SDL**).

The Company SDL covers the technical, operational, and enterprise aspects of building secure software. The SDL technical activities defined for each product, IT application, or cloud services release is the focus of this document.

Technical SDL Activities (Engineering)

SDL.T1	Security Definition of Done (DoD)	(security **To Do** list before shipping)
SDL.T2	Security Architecture & Design Reviews	
SDL.T3	Threat Modeling	
SDL.T4	Privacy & Data Protection Review	
SDL.T5	Secure Coding Standards	(includes cryptography)
SDL.T6	Manual Code Review	
SDL.T7	Open Source & 3rd Party Libraries	
SDL.T8	Vendor Management	(includes software legal compliance)
SDL.T9	Static Security Testing (SAST)	
SDL.T10	Interactive Security Testing (IAST)	
SDL.T11	Dynamic Security Testing (DAST)	(includes Web Application scanning)
SDL.T12	Fuzz Testing	
SDL.T13	Vulnerability Scan	
SDL.T14	Penetration Testing	
SDL.T15	Security Testing & Validation	
SDL.T16	Operating Environment	(includes public cloud services)

Not all the sixteen (16) technical SDL activities are mandatory for each product release. Some are conditionally required. The SDL.T1 Security Definition of Done (DoD) lists which activities are required for each releaseand is owned by the SSAs. Several activities are mandatory no matter what, such as the Security DoD (**T1**), Privacy Review (**T4**), Manual Code Review (**T6**), and SAST (**T9**).

Operational SDL Activities (InfoSec)

- SDL.O1 Program
- SDL.O2 Security Development Lifecycle (SDL)
- SDL.O3 Vulnerability Response (PSIRT/ASIRT)
- SDL.O4 People & Resources
- SDL.O5 Tools & Services
- SDL.O6 Policy & Compliance
- SDL.O7 Security Training
- SDL.O8 Metrics & Reporting
- SDL.O9 Maturity Models

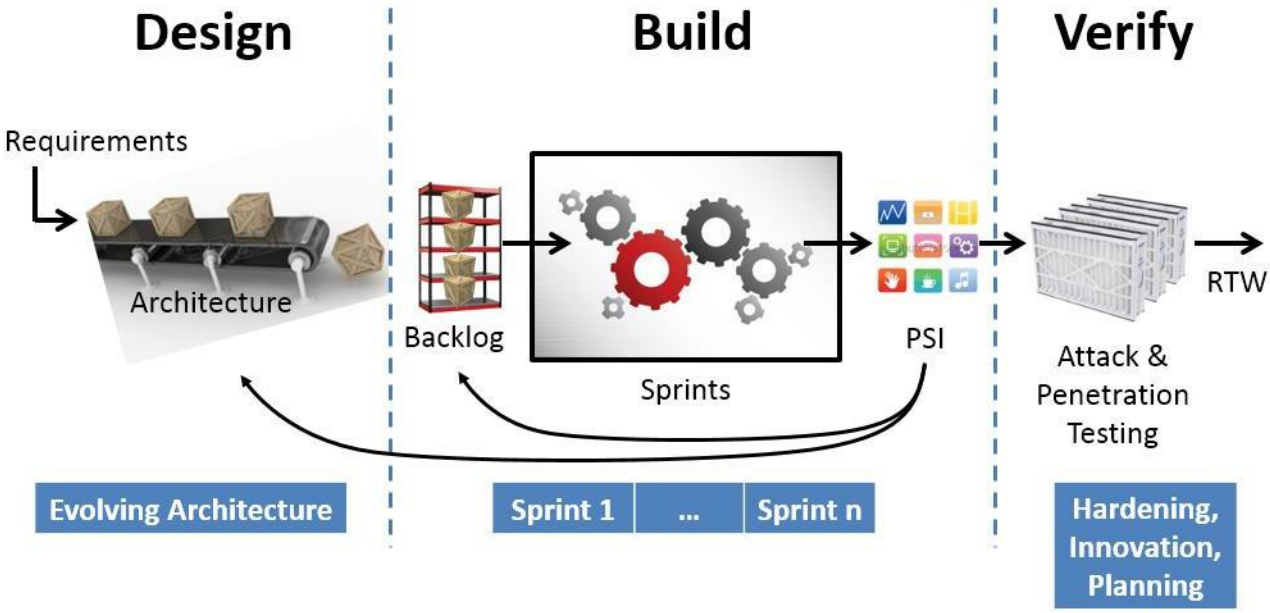
Enterprise SDL Activities (IT)

- SDL.E1 Vulnerability Management
- SDL.E2 Risk Management
- SDL.E3 Asset Management
- SDL.E4 Remediation Management
- SDL.E5 Exception Management
- SDL.E6 Security Monitoring
- SDL.E7 Certifications

The following paragraphs describe, at an elevated level, the Company SDL process.

SDL.O2 High-Level SDL

For a new product, the security process typically begins at project initiation. A seasoned Company security architect or Software Security Architect (**SSA**) or Engineer (**SSE**) assesses a proposal for its security implications. The output of this engagement is any additional security features that will be added to software self-protection so that the software can be deployed by the different security postures of Our customers.



SDL.T2.1 Security Architecture Review

Any project that involves a change to the architecture of the product is required to go through a security architecture and design review. The proposed architectural and design changes are analyzed for security requirements, as well as analyzed within the whole of the architecture of the software for each change's security implications. An architecture review may be a discrete event, may be accomplished iteratively asthe architecture progresses (**Agile**), or may be updated continuously (**CI/CD**).

SDL.T2.2 Security Design Review

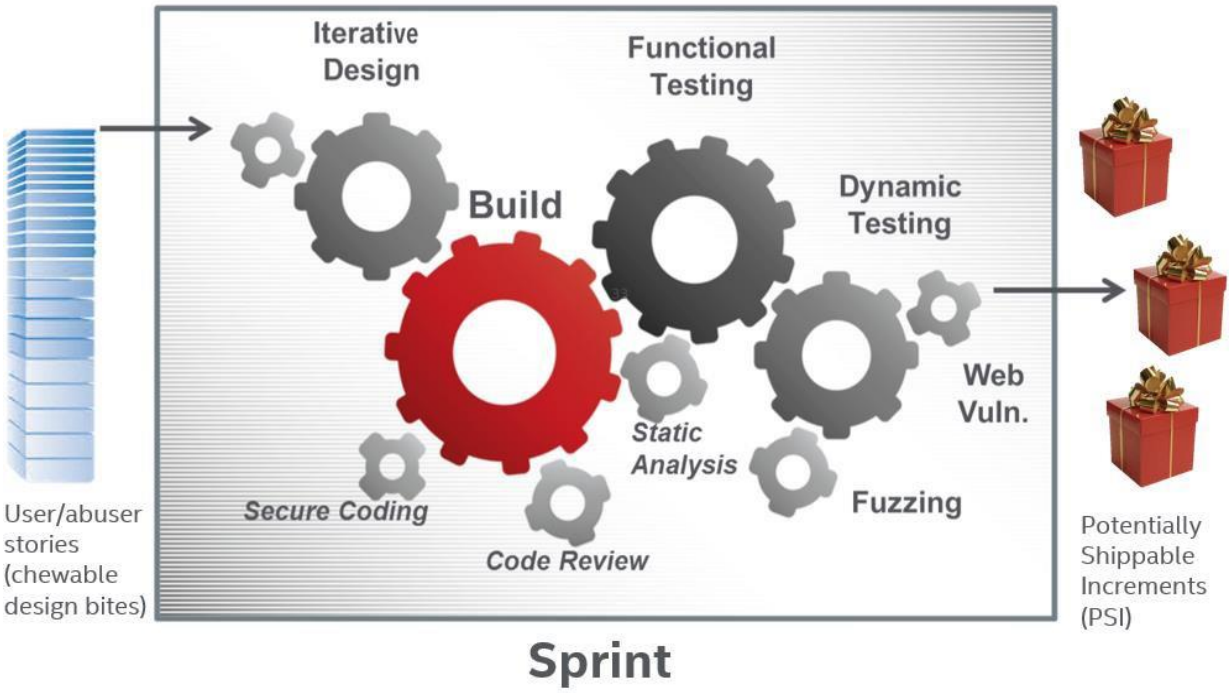
The SDL requires that designs that contain security features or effects are reviewed to make sure that security requirements are built correctly. The SSA signs off when the design meets expectations. All functional items, including security design elements, are included in the thorough functional test plan. Like architectural reviews, a design review may be a discrete event or may be accomplished iteratively when design work occurs (Agile or CI/CD).

SDL.T3 Threat Modeling

A threat model is created or updated. The output of this analysis will typically be the security requirementsthat must be folded into the design that will be implemented.

SDL.T4 Privacy and Data Protection Review

In tandem with architecture and design reviews, privacy and data protection reviews are conducted. A Privacy Impact Assessment (**PIA**) is performed to determine if any additional privacy activities are requiredto protect personal data. Privacy reviews cover the whole lifecycle of personal data and often extend beyond the product collecting the data and include backend systems and infrastructure.



SDL.O7 Security Training

We foster industry standard secure coding practices. To that end, our Learning Management System (**LMS**) contains many courses on building software securely. Some are home-grown from internal subject matter experts, while others are purchased from third-party vendors. Developers are expected to pursue ongoing developer education. Self-training is encouraged.

SDL.O4 Software Security Architects

Software Security Architects (**SSA**) and Software Security Engineers (**SSE**) are assigned to each productline and IT application. Our 120+ SSAs and SSEs perform the SDL activities and help to confirm that every part of the software security process is applied appropriately.

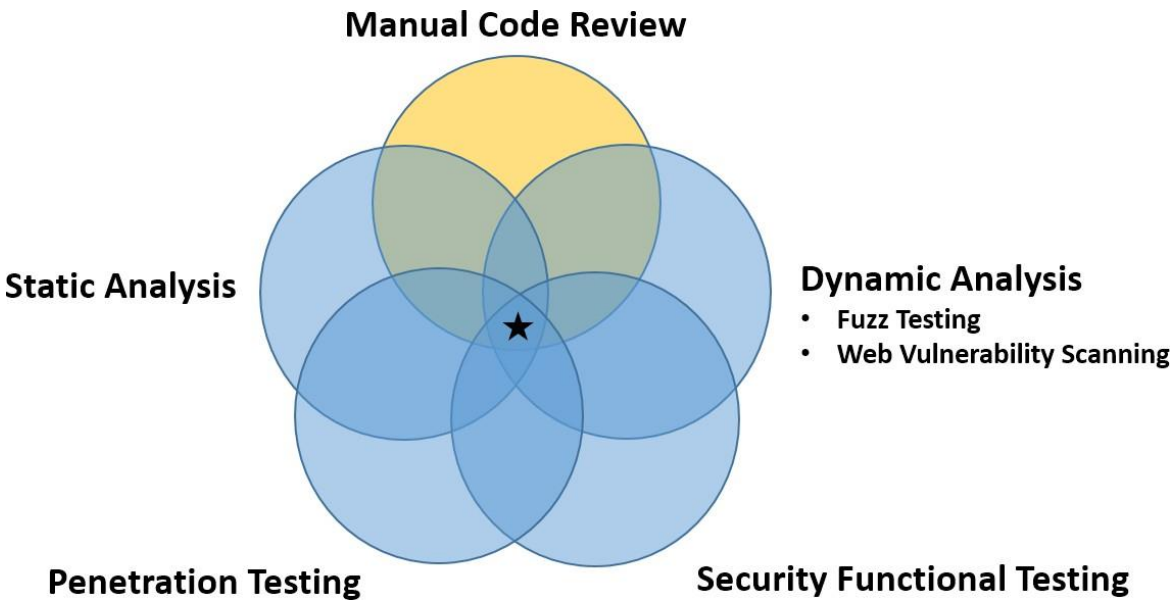
Software Security Architect/Engineer Qualifications

1. A minimum of 3-5 years software development experience
2. A passion for or background in software security
3. Approved by the BU Engineering VP/Sr. Director & SSA BU Lead
4. Dedicate a minimum of 20% of their time doing software security tasks
5. Time to be trained in software security, reviews, tools, and processes
6. Be collocated within each engineering team / BU
7. Must not only know how to develop (build) software but also know how to deconstruct it (take it apart) while "*thinking like a hacker*"

SDL.T6 "Trust and Verify"

Alongside each developer's responsibility to produce secure code, We have a "***trust and verify***" attitude. All new code must go through a manual code review. For non-sensitive and noncritical functions, this code may solely go through peer review. Critical and sensitive changes are also reviewed by staff with a sufficient level of expertise to assess critical changes.

Making use of overlapping complementary approaches, we employ several tools and automation to find security defects that may slip through manual code review. All code must be statically analyzed (unless no static analyzer exists for the language or environment). All web code is expected to undergo a web vulnerability scan. Other forms of input are routinely fuzz tested. Medium, high, and critical severity issues must be fixed before release. Low severity issues are prioritized then usually fixed or mitigated infuture patches and product releases.



SDL.O5 Complimentary Security Testing

Critical customer-premise releases may additionally be put through a third-party penetration analysis on a case-by-case basis before release. All hosted systems are routinely vulnerability scanned and penetration tested by either our Information Security (**InfoSec**) department or by a third-party engagement.

We believe that the preceding is a solid plan in line with industry standards and best practices. Since no computer system can be entirely secure, We do not claim that the SDL will prevent any particular issue or any collection of issues. We reevaluate and update our SDL policies and process on a regular basis.

SDL.O6 Policies

We believe that customer relations are best served through open, transparent dialog. We encourage customer engagement, including requests about our software security process.

There are some limitations as to what we may share. For instance, we never share our source code outside of our direct control. Also, we never make available the list of vulnerabilities that are found because of our own internal investigations or from any of our automated testing tools. After internally discovered vulnerabilities have been addressed in a hotfix, patch or new product release, all medium and high severity issues are documented in product release notes and in security advisories.

It is important to note that any scan of the Company production systems will be considered an attack. Response to perceived attack will be rapid and decisive. Please coordinate your needs with your account manager. Availability of test systems is subject to customer need, customer cost, and timing.

SDL.O5 Software Security Tools

Our engineering teams apply an appropriate combination of tools depending upon the target programming language, architecture, and the execution run-time. These tools are a combination of internally developed, vendor purchased, and open-source tools. We may provide a list of utilized tools upon request. For reference, we use many of the security tools listed in OWASP's Security Testing Tools.

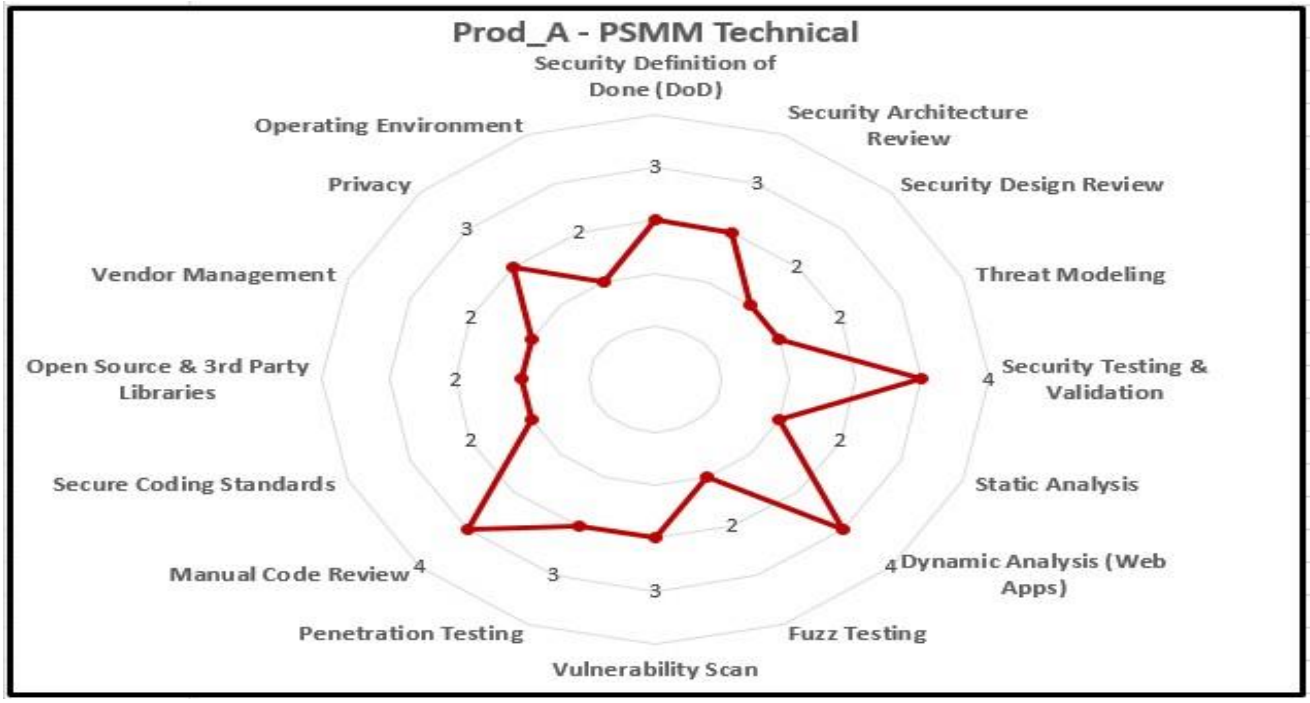
SDL.O9.1 Product Security Maturity Model

The SDL describes the "*what*" of software security. The Product Security Maturity Model describes the "*how well*" of software security.

For each SDL activity, the PSMM describes 5 different levels from 0-4. These levels are:

Level 0	None	
Level 1	Minimal	[Initial]
Level 2	Good	[Basic]
Level 3	Better	[Acceptable]
Level 4	Best	[Mature]

With 16 technical activities and 9 operational activities, a perfect score is 100. The Company software development teams assess their products annually using the PSMM. This allows us to focus our efforts on what each particular product needs the most, while measuring the overall maturity of each product line,engineering BU, and the Company as a whole.



SDL.O3 Vulnerability Response

To manage vulnerabilities discovered in shipping Company products and live customer-facing applications, We have a Vulnerability Response Team. This team consists of both PSIRT and ASIRT. The Product Security Incident Response Team (**PSIRT**) responds to product vulnerabilities in shipping products. They work with the discoverer and engineering to develop and deliver a patch and accompanying security bulletin. The vulnerability's severity (**CVSS base-score**) and business risk factors determine our fix response time (**SLA**). Similar to PSIRT, the Application Security Incident

Response Team (**ASIRT**) responds to IT application and cloud services vulnerabilities in both externally and internally facing IT applications.

DISCLAIMER

No computer system can be entirely secure. **WE MAKE NO WARRANTY CONCERNING ANY MALFUNCTIONS OR OTHER ERRORS IN ITS HARDWARE PRODUCTS OR SOFTWARE PRODUCTS CAUSED BY VIRUSES, INFECTIONS, WORMS, OR SIMILAR MALICIOUS CODE NOT DEVELOPED OR INTRODUCED BY US. WE MAKE NO WARRANTY THAT ANY HARDWARE PRODUCTS OR SOFTWARE PRODUCTS WILL PROTECT AGAINST ALL POSSIBLE SECURITY THREATS, INCLUDING INTENTIONAL MISCONDUCT BY THIRD PARTIES. WE ARE NOT LIABLE FOR ANY DOWNTIME OR SERVICE INTERRUPTION, FOR ANY LOST OR STOLEN DATA OR SYSTEMS, OR FOR ANY OTHER DAMAGES ARISING OUT OF OR RELATING TO ANY SUCH ACTIONS OR INTRUSIONS.**

-Glossary follows this page-

GLOSSARY

Application Security Incident Response Team (**ASIRT**): Part of the Vulnerability Response team within the Company that responds to IT application and cloud services vulnerabilities in both externally and internally facing IT applications.

Continuous Integration / Continuous Delivery (CI/CD): A period for releasing software updates more frequently than agile as more products become cloud native.

Dynamic Analysis Security Testing (DAST): Run-time code review using automated tools.

General Data Protection Regulation (GDPR): The EU's privacy regulation effective 25 May 2018.

Interactive Analysis Security Testing (IAST): IAST is a form of application security testing that stems from a combination of dynamic application security testing (**DAST**) and runtime application self-protection (**RASP**) technologies.

Privacy Impact Assessment (PIA): A privacy review conducted on all products to determine if additional privacy activities are required before a product is released.

Product Lifecycle Framework (PLF): The Company SDLC (*defined below*).

Potentially Shippable Increment (PSI): An agile term that means that each unit produced from a series of Sprints has a quality of completion. A governance checkpoint determines each release. PSCs participate in release decisions. There is no mandate to release a PSI.

Product Security Incident Response Team (PSIRT): Part of the Vulnerability Response team within the Company that responds to product vulnerabilities in shipping products. https://www.trellix.com/en-us/assets/docs/legal/supplier-security-requirements.pdf

Product Security Maturity Model (PSMM): Measures how well each SDL activity is being performed.

Static Analysis Security Testing (SAST): Source code review using automated tools.

Security Development Lifecycle (SDL): A secure software development methodology that condenses the traditional waterfall methodology delivery cycles into weeks instead of month. Used by all Company software development teams. The security aspects of an SDLC.

Software Development Lifecycle (SDLC): Describes the processes, activities, and deliverables for developing, testing, and shipping software.

Software Security Architect (SSA): A senior security architect within the Company responsible for all security-related activities for a given product line.

Software Security Engineer (SSE): A security engineer within the Company responsible for all security-related activities for a given product line. SSEs are typically not as experienced as SSAs.

-End-