White Paper

Security
Product Assurance

Intel Security Development Lifecycle

The Intel SDL applies privacy and security practices across software, firmware and hardware to develop and maintain more secure products.

Authors

Vernetta Dorsey
Product Security Practices Engineer

Camille Morhardt
Director of Communications

Overview

Developing products with a security mindset is an important industry practice that reduces mitigation costs and improves product resiliency. The security development lifecycle (SDL) is a set of processes that implement security principles and privacy tenets into product development. These processes incorporate security minded engineering and testing at the onset of product development when it is more effective and efficient to employ. SDL is part of Intel's comprehensive product security assurance approach. Other aspects of Intel's approach include training, conferences, the Product Security Incident Response Team (PSIRT), the Bug Bounty program, research (offensive and defensive), and industry collaborations.

While SDL is most common in software development, Intel has been applying these principles across software, firmware and hardware development since at least 2009. Intel's Security and Product Assurance organization was formed in 2018, furthering the implementation of security best practices like SDL. The methodology and adoption of SDL continue to evolve.

Table of Contents

Intel's SDL Process

The physical nature of hardware security offers a unique set of challenges and opportunities. One challenge is that hardware generally has longer development cycles and support lifetimes than software. The opportunity is to impose security objectives early in the product definition before doing so becomes increasingly costly to change. Due to the length of time for hardware development and manufacturing, architects must attempt to anticipate new usage models and potential threats years in advance. Unlike software, though, hardware offers opportunities to improve the robustness of product security beyond what is possible in code. For example, trust boundaries can be enforced through physical separation of trusted and untrusted memory regions.

Moving up the stack, Intel SDL is applied to BIOS, drivers, open source software that Intel maintains, and myriad products across the company. Although every product differs, the same mindset and methodology are applied where relevant, and broadly enforced. This is one of many investments in security that underpin Intel's Security First Pledge and ongoing commitment to delivering high quality, more secure products.

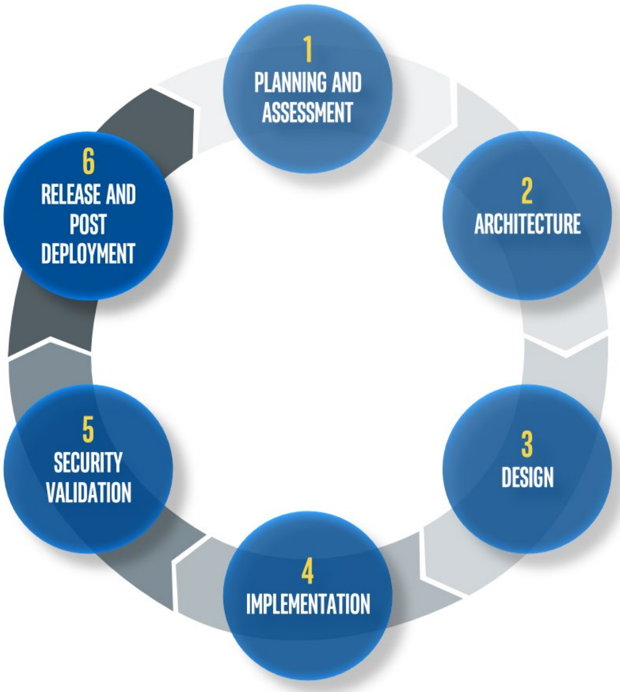

Figure 1. The six phases of Intel's SDL process

The 6 Phases of Intel SDL

Planning and assessment: This first step in the SDL process helps the project team identify what tasks and activities will be needed throughout a project's development lifecycle. These tasks are specifically assigned to individual projects and are tailored to the project's expected security and privacy risks by security experts assisting product architects.

Architecture: Architects and developers collaborate to define security objectives and use them to build an appropriate threat model. Following industry best practices for secure design principles, the team executes the reviews laid out in the planning and assessment phase, documenting all work. If appropriate, the team will complete a series of architectural reviews during this timeframe, including cryptography and more.

Design: Engineers perform design level security and privacy analysis based on the security objectives, threats and requirements identified in the previous phase. The team translates these items and updates security documentation. They define the security and privacy validation strategy in a manner intended to ensure that sufficient resources will be available to cover all requirements during the validation phase.

Implementation: The team works to ensure that the product implementation addresses the threat models defined in the architecture phase. Engineers perform secure code reviews and static code analysis, establish formal verification for any mitigations applied for potential vulnerabilities, and check that the architecture and design of the product is performing as intended. The team accounts for any necessary updates to the SDL and formal security validation plan to execute in the next phase.

Security tools are essential to identifying potential and known vulnerabilities, and doing so at scale. That's why Intel maintains a dedicated team for assessing and implementing security tools across the enterprise. There are also engineers developing custom fuzzing tools, among others, across the company for internal, academic, and customer use.

Security validation: When done well, validation monitors known vulnerabilities and emerging threats. It can even help anticipate potential threats. Ultimately, validation results affect "ship/no ship" decisions based on whether the product requirements have been met and required plans have been executed. Validation teams use many different forms of testing, including but not limited to, penetration testing (external if required) and fuzzing.

Release and post deployment: During this final phase, teams execute another round of testing and verify that previous issues were resolved. Teams scan for malware and for known vulnerabilities in third-party components and IP, updating them if appropriate. Finally, teams put a plan in place for product support and survivability. This includes vulnerability management over the lifespan of the product and triage/mitigation in partnership with Intel's Product Security Incident Response Team (PSIRT).

Conclusion

The security development lifecycle is important to building consistent quality, security and privacy across Intel products. However, it is only part of the total investment we make in security, which includes red teams, Bug Bounty, hackathons, tool development, academic research programs, and more. Across all these efforts, we are committed to continuous improvements. This means embracing new learnings by embedding them in trainings and institutionalizing them within SDL processes and tools.

Once a product has been released, we encourage users to keep their systems up to date. Our products are constantly being evaluated, and improvements can be made at any time to help improve security or deploy mitigations for potential vulnerabilities. Often, these updates are delivered to the end user via microcode pushed out from the operating system vendor. Security doesn't end at development. That's why we work with our technology partners to create more secure solutions and maintain them via regular updates.

Notice & Disclaimers

Intel provides these materials as-is, with no express or implied warranties.
All products, dates, and figures specified are preliminary, based on current expectations, and are subject to change without notice.
Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.
Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No product or component can be absolutely secure. Check with your system manufacturer or retailer or learn more at http://intel.com.
Some results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.