Cisco Secure Development Lifecycle (CSDL)

Our process designed to mitigate the risk of vulnerabilities and increase resiliency of Cisco products.

Contents

CSDL at a Glance

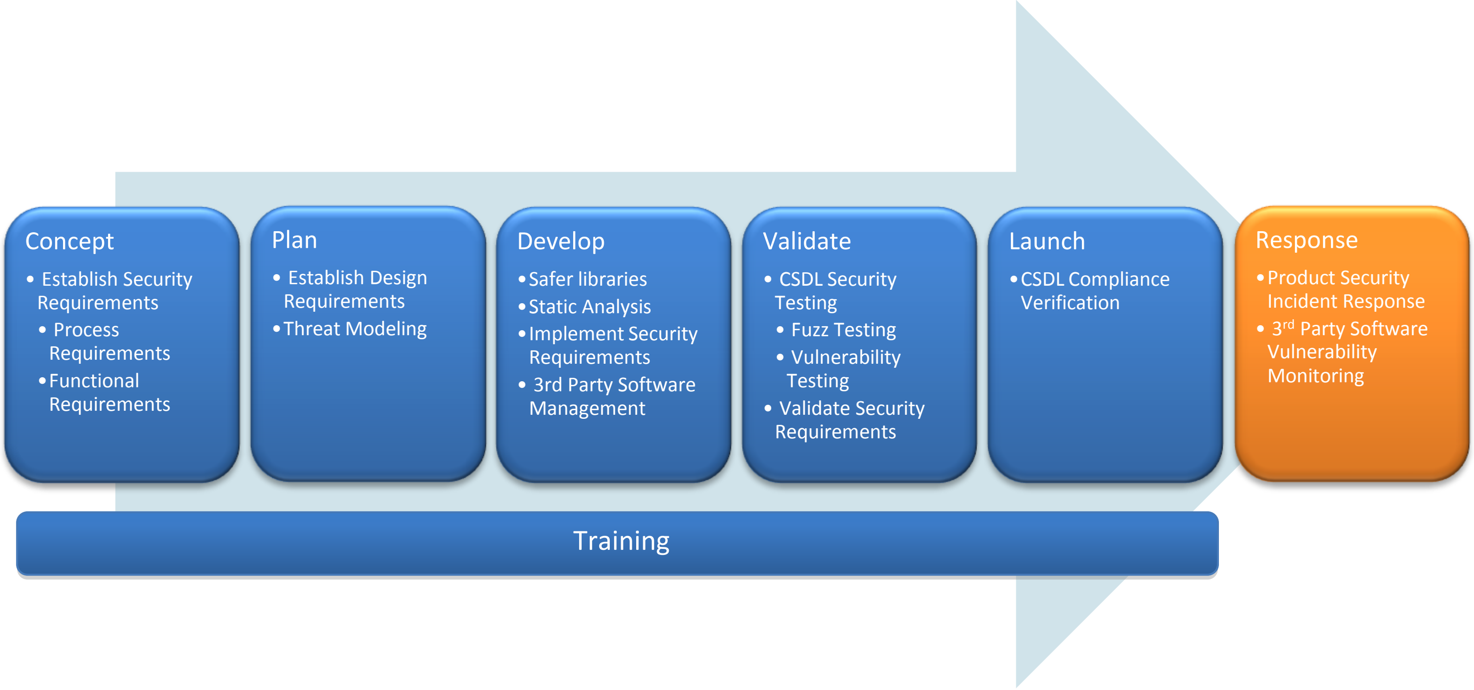
- Industry-leading secure software development best practices, processes, and tools
- Focused on making security an inherent part of the development DNA
- Built on the best practices and industry collaboration
- Evolves and improves through evaluating the changing threat landscape and ongoing community engagement

Overview

Cisco’s products and services range from helping customers create borderless networks with collaboration to data centers enabling virtualization. Our products work in almost every market segment from governments and large enterprises to small businesses and homes. As an industry leader, Cisco is expected to deliver secure and resilient products that can withstand attack. Our customers not only look to us to ensure their networks are more safe and secure, but they expect product security to be seamlessly integrated into all of our products. In order to achieve this, we’ve integrated security best practices into our product architecture, design and development processes so that product security becomes part of our DNA and corporate culture.

This document describes the **Cisco Secure Development Lifecycle (CSDL)**, our process designed to mitigate the risk of vulnerabilities and increase resiliency of Cisco products.

CSDL is applicable to all programming languages, operating systems, and application development efforts. The requirements defined are both agnostic and complimentary to any development methodology including standard waterfall, agile, or any other type of defined process. However, for different development methodologies there are distinct implementations. CSDL is divided into activities called elements. For example, Threat Modeling, Static Analysis, Identity Assurance, and Security Testing are some of the elements of CSDL. Inside Cisco, CSDL elements are used as an overlay to our standard development lifecycle. Augmenting each phase to include processes, tools or deliverables that are targeted towards product security and resiliency.



Concept
- Establish Security Requirements
  - Process Requirements
  - Functional Requirements

Plan
- Establish Design Requirements
- Threat Modeling

Develop
- Safer libraries
- Static Analysis
- Implement Security Requirements
- 3rd Party Software Management

Validate
- CSDL Security Testing
  - Fuzz Testing
  - Vulnerability Testing
- Validate Security Requirements

Launch
- CSDL Compliance Verification

Response
- Product Security Incident Response
- 3rd Party Software Vulnerability Monitoring

Training

Figure 1 CSDL 1.0 Elements

Given the myriad of options available to improve product security, the elements included in the CSDL share the following characteristics:

1) Pragmatic and highly actionable

2) Mature and applicable across multiple product families, operating systems and development environments

3) Preference for automated or infrastructure-level solutions to maximize ROI

4) Leverage or enhance product security specific aspects of existing development tools and process

While each CSDL element is designed to maximize protection against specific classes of vulnerabilities, no single CSDL element can ensure a product is “safe.” The combination of tools and processes introduced in all phases of the development lifecycle ensures defense-in-depth, and provides a holistic approach to product resiliency

CSDL is a living program that evolves over time. It includes resources such as secure coding guidelines, operating system hardening recommendations, penetration testing, esoteric testing tools etc. There are other resources available to development teams in addition to those defined as elements in CSDL. As these items mature, they may become standard elements in future releases of CSDL in response to changes in the threat landscape.

Training

In depth product security knowledge is not a pre-requisite for successful use of CSDL. Training, however, is an inherent part of CSDL. It is designed to provide guidance on product security aspects of specific tool/process requirements as they pertain to individual job roles or specific tasks. Each job function has a different task when comes to development and CSDL trainings is tailored to those different roles. For example, developers are offered training on Threat Modeling, Static Analysis, Secure Coding, and etc, while testers have different priorities.

In addition, Cisco holds an annual internal security conference aimed at raising security awareness within the company's development community, called SecCon. All of Cisco is invited; however, the Cisco development organization is the primary target audience. Product teams get to learn from and interact with security researchers directly. SecCon bridges the communications gap, bringing security expertise directly to teams developing Cisco products.

Security Advocates

How does one build good, secure development practices into the DNA of a company with many different business units, an incredibly diverse set of product lines, and employees distributed around the globe? A virtual community of sharp, knowledgeable people who understand network security and secure product development (and testing), and who can share and evangelize that knowledge with their peers, their colleagues, and their management is one of the important ways.

That virtual community is a reality at Cisco. The Security Advocates program has broad representation from the business units, representing diverse product lines ranging from small SOHO devices to core routers and switches to SaaS applications. As innovations in CSDL occur, security advocates are the ones who ensure it gets propagated to the product teams.

Concept Phase

The concept phase defines the full product from an end-user perspective, describing features and functionality necessary to create a useful, usable, and desirable product, and is documented in a statement of product requirements. During the concept phase, key product security requirements need to be included as this is the earliest opportunity in the development lifecycle to eliminate potential vulnerabilities, thus resulting in significant savings during the plan, develop, and validate phases.

Plan Phase

The plan phase specifies what will be designed and built (in response to requirements), and what resources are required to deliver the product. As the engineering team considers the overall system architecture, they must augment their design process to include security concepts such as reduction in attack surface, least privilege, intellectual property protection and defense-in-depth. Bolting on securing features during the implementation or validate phase is error prone and will significantly impact release schedule. CSDL provides tools and methodologies to assist teams with building products that are secure by design.

Development Phase

During the development phase, the product team begins to write the code and perform unit testing. Throughout this phase, a number of sophisticated tools and technologies are incorporated to detect and reduce security-related software vulnerabilities in the system. This starts with adhering to best practice coding guidelines, choosing secure versions of libraries used in the system, and running the latest static analysis tools to detect and eliminate security vulnerabilities from the source code. The next layer of defense is to enable the compiler to automatically provide protection when certain compile time conditions are met. Finally, certain run time defenses can be put in place to reduce the chances of successful exploits if the attacker were able to inject malicious code.

Validate Phase

The product is validated through integration, feature test, system and regression tests, Early Field Trial and Beta testing. As part of this process, the product security functionalities integrated in the prior phases need to be verified. In addition, a number of new activities are introduced to further validate the system prior to Launch and put in place the ability to continuously monitor threats post Launch.

Launch Phase

Compliance Verification examines security activities performed on software prior to release. The verification can include an examination of trusted product architecture evaluation, threat models, and CSDL tool outputs.

Response Phase

Once the product has been thoroughly validated, and passed the launch readiness review, it is launched officially. Even with multiple layers of defenses in CSDL, undoubtedly not all security vulnerabilities will be eliminated from the system. This is especially true as the threat landscape shifts over time. It is important teams have a strategy in place to report and respond to security vulnerabilities.

Summary

Experiences from CSDL deployments have shown it to be a highly effective process in detecting and preventing security vulnerabilities, and improving overall system quality. Given Cisco's leadership position across wide array of market segments, we are and always will remain a top target. CSDL allow us make security an inherit part of the development DNA, thus ensuring Cisco delivers secure and resilient products to our customers.