SecurityCompass

How SD Elements Enables DevSecOps

FEBRUARY 2019

Table of Contents

Intro

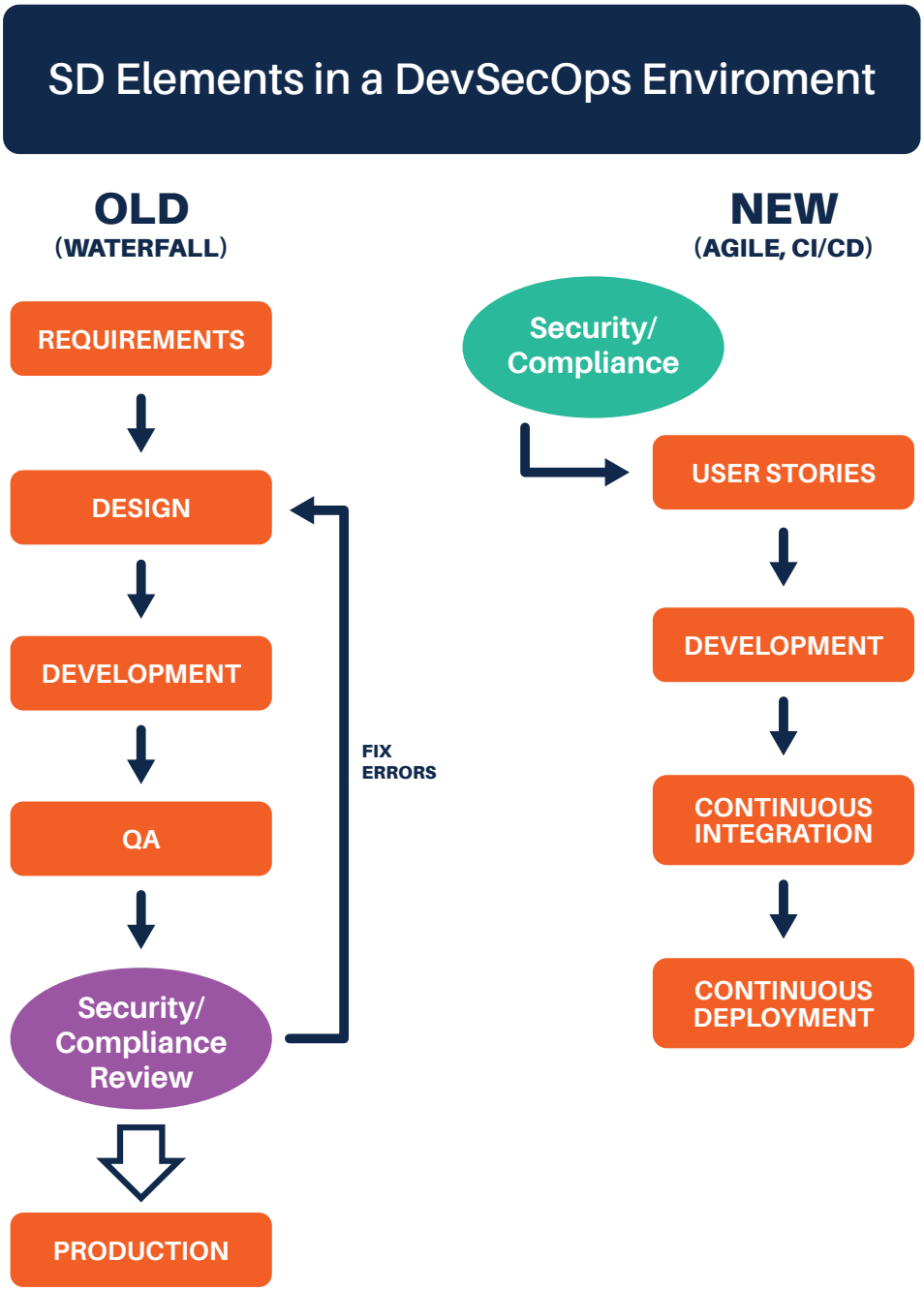
In the past, when Software Engineering teams wanted to build an application, the security, privacy, compliance, and other "non-functional" teams would define the steps needed in order to advance the application to production. In the new paradigm, the non-functional teams bring their expertise to the development and insert themselves into the existing workflow. The main idea is that security and privacy must enable development, not block it.

The non-functional teams have a new role to play in this altered paradigm. Developers can now go to non-functional team members and say, "Here is an automated, scalable, self-service solution — if you want us to know about security and privacy, then please communicate with us through here."

This process starts by having development teams profile their application in SD Elements. From this point, security, privacy, and compliance controls are automatically identified and synced into the development team's native ALM system (e.g., Jira).

*See paradigm diagram on next page

Here, we will review SD Elements' role in multiple aspects of a DevSecOps environment.



PARADIGM DIAGRAM

Automates Threat Modeling

Threat modeling is a very high-value activity, but real threat modeling requires significant time and experienced personnel. In DevSecOps, a real threat model has to be used sparingly.

With SD Elements, threat modeling can be significantly accelerated. Identifying the potential security weaknesses and privacy requirements can be partially automated. The SD Elements profiling survey and knowledge base can be extended to include non-functional requirements from:

1. Business Context (e.g. Application is an online banking application, internet-facing, allows money transfer)
2. Technical Context (e.g. web application, uses REST APIs, uses Java, SQL DB, etc.)
3. Compliance Context (e.g. GLBA, PCI, SOX, GDPR, etc.)

This still leaves a gap in "threats" that relate to component interaction, business logic, and other areas which should be addressed in a streamlined risk analysis process.

De-duplicates Controls

In the old system, developers were often presented with "security-centric" or "compliance-centric" controls that needed to be implemented. Now, development teams can profile an application in SD Elements and be presented with a single, consolidated list of controls. This way, if an auditor wants a demonstration of which controls were in scope, who implemented them and when, who validated them and when, and what "notes" were attached to those activities, then they can just generate a report without having to interrogate software engineers. This can be used to demonstrate due diligence in the event of a security breach.


Reporting
Business Units
Library
Manage
DASHBOARD
GLOBAL REPORTS
My Recent Activity
Thursday, Nov. 17, 2016
Tuesday, Nov. 15, 2016
Rhyzgen Projects In-Progre...
41
Hailstorm Progress v5.15
Total: 101 of 160
Ealith Incomplete Tasks
62
Training Report 2016
Email
Completed
True
True
False

Reduces the Noise from DAST, SAST, and Other Automated Security Testing Tools

Automated security testing plays a major role in Continuous Integration and Deployment processes. It's a necessary activity that enables developers to write, test, and publish code within minutes. Unfortunately, this kind of testing is necessary but not sufficient, as it suffers from a significant problem of false negatives. When you tune a scanner to run fast and have few false positives, you often tune out the warnings. You know the old saying, "where there is smoke, there is fire"? Well, in SAST and DAST, you can tune out the smoke in order to improve your signal-to-noise ratio, but then you'll miss some cases where the smoke would have led you to a fire. So, you need to bolster your SAST and DAST pipeline with supporting activities.

Identifies and Tracks Controls as Early as Possible

By starting with the controls, you can identify the baseline of security and privacy mechanisms which should be implemented. There is a significant advantage here - SD Elements has a button you can click that says, "Show me all of the controls which my SAST/DAST pipeline didn't test for." Hence, SD Elements can significantly reduce your risk from false negatives.

Supplements Penetration Testing with Automation

Penetration Testing (with real humans) is still immensely valuable in DevSecOps shops. Real humans may not be able to test as frequently as your developers can deploy, but they will still find real vulnerabilities, even in an application that has been scanned several times per day with SAST and DAST because those tools have significant blind spots and false negatives, as discussed earlier.

At Security Compass, we are not alone in pen testing applications developed with modern DevSecOps best practices and on-boarded into Bug Bounty programs, still manifesting glaring holes with critical impacts. People like to pretend that this only applies to "business logic" vulnerabilities which don't lend themselves well to automation, but that isn't true. Empirically, we're finding problems in authentication, session management, templating engines, and more.

Gates Releases with Fast Automation

Free, open source continuous integration solutions (like Jenkins) and more feature-rich commercial solutions (like XL Release from XebiaLabs and Microsoft Azure Pipelines) allow you to control when a build is failed versus when it can be pushed to production. These tools routinely look at the output of test suites (like SAST and DAST), regression testing, and integration testing to make their release decisions.

Interestingly, these systems can also query SD Elements' "Risk Policy" to determine whether an application is suitable for promotion to production. In SD Elements, as you complete the "Project Profile," a Risk Policy can be assigned to the project. For example, an Internet-facing financial application would likely have a different Risk Policy than an internal productivity application. That Risk Policy is either "green" or "red," and it can be configured to look at different Implementation and Verification statuses. So for example, your critical applications would probably want at least the "High Priority" (weighted 7-10) items in SD Elements to be completed in order for the policy to be "green."

Using SD Elements in combination with continuous integration (the "CI" in "CI/CD") makes sense because SD Elements houses a single, consolidated set of security controls which has been de-duplicated and has mappings to security, privacy, and compliance teams. It also includes mappings from SAST and DAST output, so an up-to-date test status is available with broader coverage of issues than can be tested for with SAST/DAST alone (c.f. "changing role of SAST/DAST").

Operationalizes Cloud and Container Security Controls

The "Ops" in DevSecOps refers to the software engineering organization deploying code into production and being responsible for its "operation." Typically, this means that software engineers are interfacing with deployment-related technologies, like Amazon Web Services (AWS), Azure, or containers like Docker.

There are security best practices associated with deploying into these environments, and your organization may also want to create custom policies as well. SD Elements has partnered with the Center for Internet Security, and we are starting to incorporate their "CIS Benchmarks" into the SD Elements knowledgebase. For example, in the Project Profile, if you indicate your application is using AWS, SD Elements can provide hardening guidance for roughly two dozen AWS technologies.

Software Supply Chain and Dependency Analysis

In order to support more frequent releases, a best practice would be to use an automated Software Composition Analysis (SCA) tool that allows your software engineering team to centrally manage third-party dependencies. Sonatype and Black Duck have been the traditional market leader in the space, with SourceClear and Snyk acting as recent disruptors. OWASP Dependency Checker is a good free open source solution.

SD Elements indirectly integrates with these tools through ThreadFix, and there are tasks and Verification tasks in SD Elements that will flag as 'Incomplete' or 'Failed Verification' if a vulnerable or out-of-policy component is selected.

To learn more about how SD Elements works, please contact us.

SecurityCompass

Security Compass believes in a world where people can trust technology, and it is our mission to help customers proactively manage cybersecurity risk, without slowing down business. Our holistic program offerings are tailored to your organization's needs. Through advisory services, training, and SD Elements, our award-winning policy-to-execution platform, we set you up with all of the resources and tools your organization needs to develop secure software. Security Compass serves some of the world's largest enterprises, including 16 of the largest financial institutions in North America, South America, and Asia, as well as 4 of the largest tech companies in the world.

OFFICES

GLOBAL HEADQUARTERS
1 Yonge Street
Suite 1801
Toronto, Ontario
Canada M5E 1W7

TORONTO
390 Queens Quay W
2nd Floor
Toronto, Ontario
Canada M5V 3A6

CALIFORNIA
1001 Bayhill Drive
2nd Floor
San Bruno, California
USA 94066

NEW JERSEY
621 Shrewsbury Avenue
Suite 215
Shrewsbury, New Jersey
USA 07702

INDIA
#4.07
4th Floor, Statesman House
Barakhamba Road, New Delhi
India 110001

1.888.777.2211
info@securitycompass.com
www.securitycompass.com

@SECURITYCOMPASS
SECURITY COMPASS