The Six Pillars of DevSecOps:

Achieving Reflexive Security Through Integration of Security, Development, and Operations

SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

CSA cloud security alliance®

The permanent and official location for Cloud Security Alliance DevSecOps is
https://cloudsecurityalliance.org/group/DevSecOps/

Acknowledgements

Lead Authors:

John Martin
Setumadhav Kulkarni
Ronald Tse
Michael Roza
Sean Heide

Contributors:

David Lewis
Eric Gauthier
Lee Szilagyi

CSA Staff:

Sean Heide

Table of Contents

Introduction

Background

The widespread adoption of cloud computing presents an unprecedented security challenge. Organizations are confronted daily with headlines of breaches, compromises, and stolen sensitive data. These issues can arise from the effects of insecure applications, misconfiguration, problematic protocols, poor infrastructure architecture and lack of education and training.

With Digital Transformation firmly underway, software has rapidly risen as one of the top causes of business risk and exploitation. As a result of the rapid increase of the volume and pace of application development and delivery, the number and complexity of attacks on applications have also multiplied. The shortage of personnel with appropriate and adequate security skills and resources has become more acute than ever.

In other words, the ubiquity of computing and its complexity has exponentially grown the attack surface for software systems. Some have described this situation as "Moore's Revenge" [1].

Intricate interactions between multiple technology layers coupled with the globally interconnected and always-on nature of today's applications, are compounded by potential vulnerabilities lying dormant in systems, software, and hardware. These conditions have created a field ripe for picking by malicious parties across the world.

Vulnerabilities can occur anywhere – the underlying infrastructure, third party services, and the libraries that software products are built upon – and software development companies are not currently set up to find and resolve these quickly. For example, in a 2017 survey, over 80% of organizations reported that pre production vulnerability scanning is not performed [2].

The DevOps Impact

DevOps, the practice of applying developmental best practices such as collective collaboration to infrastructure operations, has been shown to positively impact efficiencies of development and operations teams today, especially in the cloud environment.

Given its strengthening adoption, it is necessary to consider its impact on information security itself and look to apply those practices to the arena of information security management.

The growing omnipresence of computing devices and computer power along modern trends like DevOps, Microservices and Open Source that accelerate deployment cycles continue to strain the ability to detect and identify exploitable flaws in a timely manner, leading to significant increases in overall security risk.

Cloud-native development paradigms like serverless computing, API-first applications, and microservices-based architectures have become mainstream choices with the rising prominence of DevOps as an enabler for these paradigms.

These new approaches to software development have evolved without security necessarily being a well thought out design principle. The only real option available to infuse security in a sustainable manner into these paradigms is to rely on the very concept that made these paradigms a reality - DevOps.

Putting security directly into DevOps substantially improves outcomes by integrating existing security into development and operational processes through modern integrated security paradigms, such as DevSecOps. For example, organizations that implement DevSecOps find that modern microservices-based applications have much better security outcomes. In addition, when security is built into a software development lifecycle, organizations find that even in the initial days of development, security-related performance can be controlled.

The goal is to reduce the complexity in the development and publication of software, ensure that only known and trusted components and services are used, empower software development teams with security resources (both automated and human) integrated directly into their development methods, use development environments that are well secured and monitored, and ultimately deliver the end-product per design and with only the features it was designed to have.

Scope

This document defines the six focus areas of DevSecOps critical to implementing and integrating DevSecOps into an organization.

The DevSecOps pillars provided in this document are meant to provide a holistic framework that blends the traditionally siloed operations: development, infrastructure operations, and information security, into a cohesive group that facilitates creation of secure software.

Normative references

Information Security Management through Reflexive Security, CSA

ISO/IEC 27000:2018, Information technology -- Security techniques -- Information security management systems -- Overview and vocabulary

Terms and definitions

For the purposes of this document, the terms and definitions given in Information Security Management through Reflexive Security, CSA, ISO 27000 apply.

DevSecOps for cloud-first challenges

DevSecOps is an approach that addresses the challenges of today’s interconnected, rapidly changing security environment with increasingly shortened infrastructure and product life cycles.

The practices of DevSecOps are created as a response to resolve issues that have risen from cloud-first software. It can be succinctly defined as the integration of continuous security principles, processes, and technologies into DevOps culture, practices, and workflows.

DevSecOps is also meant to provide a holistic framework that commingles the traditionally siloed operations, namely, development, infrastructure operations, and information security into a cohesive group that facilitates the development of secure software under controlled processes.

The CSA DevSecOps Working Group has defined the following six focus areas critical to integrating DevSecOps into an organization, in accordance with the six pillars described in the Reflexive Security framework.

Pillar 1: Collective Responsibility

One of the greatest challenges to embedding security in DevOps is changing the organization's mindset, its ideas, its customs and behaviors regarding software security.

With the introduction of Security into a DevOps organization, security must no longer be seen as someone else's responsibility. It must not be considered an afterthought to be addressed only when everyone else's work has been completed. Additionally, security cannot be seen as separate and distinct from business objectives. Lastly, security is not something ephemeral whose progress and contribution cannot be measured.

Everyone is responsible for the security stance of the organization. The CSO (Cloud Security Officer) plays a leadership and shepherding role for information security within an organization, but each person has their own security responsibility and must be aware of their own contribution to the organization's security stance. Edge users and developers are not just "security-aware" but are the first line of defense.

This corresponds to the "Responsible collectively" pillar of the Reflexive Security framework.

Pillar 2: Collaboration and Integration

There is an enormous skill (knowledge) and talent (resources) gap in the software landscape across Development, Operations and Security. Without pan-organization collaboration around implementing security, success is going to be limited. Security can only be achieved only through collaboration, not confrontation. A security aware and collaborative culture is necessary for the members of all functional teams to report potential anomalies. The human factor is often the weakest link; in fact, or indeed, and remember that most security incidents are caused by simple human error.

This corresponds to the "Collaborate and integrate" pillar of the Reflexive Security framework.

Pillar 3: Pragmatic Implementation

DevSecOps provides a proliferation of point solutions. Organizations have a wide array of tools and solutions to choose from to implement application security within their software lifecycles. Since every software lifecycle is different in terms of structure, processes and overall maturity, there is no one-size-fits-all set of tools to implement DevSecOps. Organizations often end up procuring procure tools and point solutions that are hard to deploy, harder to operationalize and eventually do not provide actionable insights that can help mitigate the true security risks.

Organizations need to take a holistic view of the software lifecycle, their security needs, and the future state they want in order to choose platform solutions that offer a high degree of integrability.

By using a framework agnostic "Digital Security and Privacy Model" focused on Application Development to ensure safety, privacy and trust in the digital society, organizations will be able to approach security in DevOps in a pragmatic manner. This model will fulfill the unmet need of connecting all the stakeholders (development, operations, and security) in a manner such that security is built into applications and the software lifecycle that produces applications.

This corresponds to the "Pragmatic" pillar of the Reflexive Security framework.

Pillar 4: Bridging Compliance and Development

Risk-related requirements are difficult to translate into security requirements that can be easily measured over time. While security teams create requirements to support their risk-based methodology, compliance requirements are poorly translated to DevOps and product requirements. Conversely, it is not easy to obtain evidence that security requirements have been met even if technical controls are implemented.

Given the rapid evolution of software development paradigms and practices, compliance and agile software development are no longer aligned. The regulatory and compliance function is more interested in proof that there is a process in place than actually auditing each execution of that process. Most DevOps teams, on the other hand, believe that the proof is in the code, not in documenting the process.

The key to addressing this gap between compliance and development is to identify applicable controls, translating them to appropriate software measures and identifying inflection points within the software lifecycle where these controls can be automated and measured to improve the quality of risk mitigation and therefore compliance.

This corresponds to the "Align and bridge" pillar of the Reflexive Security framework.

Pillar 5: Automation

Some of the issues preventing software development practices from taking an idea to secure deployment quickly with minimal cost are manual and haphazard coding, testing, deployment and patching practices.

Without automated quality checks, manual coding can easily result in poor performing and insecure software that needs rework. In addition, manual and poorly-timed testing reduces the chance that vulnerabilities will be identified before deployment. Manual deployment and patching practices can result in insecure software from being released to production.

Automated security practices are the core of process efficiency because they can reduce manual processes, increasing efficiency and reducing rework. Software quality can be bettered by improving the thoroughness, timeliness and frequency of testing/feedback. Processes that can be automated should be automated, and those that can't should be automated as much as possible or be considered for elimination. Automated security checks may create new issues, such as build delays or failures, though these typically can be addressed by workflow improvements or semi-automated approaches. There is a saying in software development: if you do the same thing three times, it's time to program it, and this applies squarely with Reflexive Security.

This corresponds to the "Automate" pillar of the Reflexive Security framework.

Pillar 6: Measure, Monitor, Report and Action

The saying "you can't manage what you can't (or don't) measure" has never been more true than in the implementation and maintenance of DevSecOps. Typical DevSecOps initiatives can take anywhere from months to years to implement depending on scope and complexity. Without actionable metrics, progress cannot be measured and failures cannot be detected in a timely manner.

Some of the most critical metrics to monitor in a DevSecOps environment are deployment frequency, vulnerability patch time, percentage code automatically tested, and automated tests per application. The results during software development as well as post-delivery must be measured, monitored, reported and acted upon by the right people at the right time (continuously) for DevSecOps to succeed.

This corresponds to the "Measure and improve" pillar of the Reflexive Security framework.

Summary

The CSA DevSecOps Working Group concludes that the focus areas described in this document is able to address weaknesses in secure software development in the context of DevSecOps and will act as a building block for the future dynamic and creation of a properly implemented DevSecOps environment.

Each of the pillars will be addressed in depth in subsequent separate whitepapers.

References

[1] ‘Moore’s Revenge’ is upon us and will make the world weird, Mark Pesce, The Register.
https://www.theregister.co.uk/2018/06/04/moores_revenge/

[2] 2017 State of Application Security: Balancing Speed and Risk, Jim Bird, SANS. Available at:
https://www.sans.org/reading-room/whitepapers/analyst/2017-state-application-security-balancing-speed-risk-38100