DevSecOps: A Secured Approach

-Aman Chhabra

Introduction

DevSecOps came out of the integration of development, security and operations that all together is a wide category, but generally it is the inclusion of security across all parts of the Software Development Life Cycle (SDLC), rather than having it as a gating function at the end or just a layer that is added during deployment cycle. Depending on the part of the stack, this can have different process and tooling consequences.

DevSecOps aims on the concept of *'***everyone is accountable for security'** with the goal to embed security parameters in each and every part of the Software Development Life Cycle (SDLC). It is about trying to automate core security tasks by embedding security controls and processes early in the DevOps workflow.

DevSecOps Appraoch:

DevSecOps involves the creation of 'Security as Code' methodology with the ongoing and flexible collaboration between the release engineers and security teams. DevSecOps approach, like DevOps itself, is focused on creating new solutions for complex software development processes within an agile framework and maintaining the security precautions from day one of the product life cycle.

Implementing the DevSecOps approach is a natural and necessary response to the bottleneck effect of older security models on the modern continuous delivery pipeline. The goal is to bridge traditional gaps between IT and security while ensuring fast, safe delivery of code. Silo thinking is replaced by increased communication and shared responsibility of security tasks during all phases of the delivery process.

Why we need DevSecOps?

The Information Technology era has undergone exponential changes over the past decade. The shift to agile cloud computing platforms, shared storage and data, and dynamic applications has brought huge benefits to organizations looking to thrive and grow through the use of advanced applications and services.

With business demand for DevOps, Agile and Public Cloud Services, traditional security processes have become a major roadblock targeted for elimination. And sadly, sometimes these are the easiest to bypass all together. Traditional security operates from the position that once a system has been designed, its security defects can then be determined by security staff and corrected by business operators before the system is released. This allows for a limited supply of skills in security to be applied to outcomes and avoids the need to increase security context within the larger system. But a process designed this way only works where the pace of business activities is waterfall and is agreed by all parties. Unfortunately, the belief that security must operate this way is flawed with the introduction of iteration and has since created inherent risks within the system because of this decision in the organizations.

What can be achieved with DevSecOps?

Considering the DevSecOps approach, two seemingly opposing goals —"speed of delivery" and "secure code"—are merged into one efficient process. In alignment with lean practices in agile, security testing is done in iterations without slowing down delivery cycles. Critical security issues are dealt with as they become apparent, not after a threat or compromise has occurred.

Security protocols that are baked into the development process rather than added as a "layer on top" allows DevOps and security professionals to harness the power of agile methodologies—together as a team—without short circuiting the goal of creating secure code.

How DevSecOps is beneficial?

A 2017 EMA report found the top two benefits of security operations (SecOps): better ROI in existing security infrastructure and improved operational efficiencies across security and the rest of IT.

Another top benefit identified in the study was the ability to make full use of cloud services. For example, organizations running services in the Amazon Web Services (AWS) cloud reap the benefits of increased preventive and detective security controls within the continuous integration and deployment model of AWS. As more organizations rely on cloud applications to keep operations up and running, security efforts independent of those performed by AWS are crucial to prevent costly downtimes.

The safety measures inherent in DevSecOps have many other advantages. These include:

-An ability to respond to change and needs rapidly

-Better collaboration and communication among teams

-Greater speed and agility for security teams

-Early identification of vulnerabilities in code

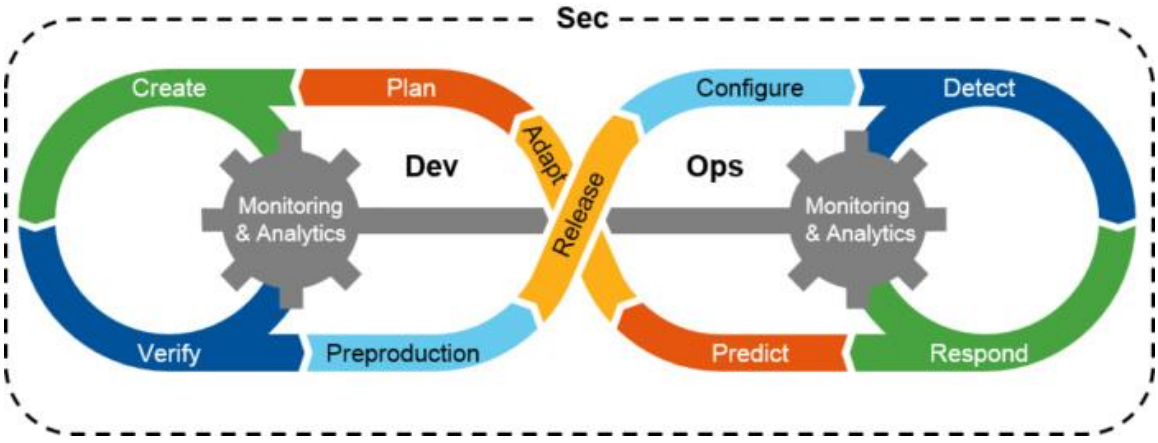
-Team member assets are freed to work on high-value work

-More opportunities for automated builds and quality assurance testing

Benefits on a wider picture:

More automation from the start reduces the chance of misadministration and mistakes, which often leads to downtime or attacks. This automation also reduces the need for security architects to manually configure security consoles.

As Gartner details, this can lead to security functions like identity and access management (IAM), firewalling, and vulnerability scanning being enabled programmatically throughout the DevOps lifecycle, leaving security teams free to set policies. The analyst firm predicts that DevSecOps -- which is slightly different from SecDevOps -- will be embedded into 80 percent of rapid development teams by 2021.



“It should be about getting security back in to the lifecycle, or as it has been described: ‘shifting security left,’” says Daniel Cuthbert, global head of cyber security research at Santander. “Security is seen as the traditional firewall to innovation and often has a negative connotation. With shifting security left, it’s about helping build stuff that’s innovative and also secure. If we get this right, we can start to reverse this image currently faced by all in security.”

Summing up all the benefits and pros, DevSecOps is a healthy model that assumes everyone is responsible for security. It’s not conceivable that one team would roll out an application and then hand it on to another team to worry about security. The two have to be determined and implemented together. This has led to the creation of tools and techniques that are aimed at providing improved security at various stages of the DevOps chain.

How to start with DevSecOps?

Merging security with DevOps to deliver DevSecOps requires new mindsets, processes, and tools. Security and risk management leaders need to adhere to the collaborative, agile nature of DevOps to be seamless and transparent in the development process, making security as silent and seamless as possible.

If you're operating on the cloud, the most important attack vector is misconfiguration of cloud services. According to Gartner, more than 80% of cloud breaches are due to misconfiguration. In the cloud, developers are creating and modifying infrastructure, so they're making decisions about configurations that can impact the security posture of cloud environments. This is a departure from the data center, where ops and security teams had more control over the configuration of IT resources like networks and firewalls. Security needs to get in front of this, while simultaneously not creating too much friction or too many limitations for the developers.

Scalability in the cloud requires embedding security controls on a larger scale. Continuous threat modeling and management of system builds is needed as technology-driven businesses evolve at a rapid pace.

One should start is with automated policy checking of existing environments. This can be done in hour or two, and that will give you an assessment of your current security posture. Use automation to stay up to date, as you’ll find cloud infrastructure frequently drifts out of policy due to manual maintenance and deployment updates.

Once your production security posture is in good shape, turn to making sure that no new problems are created.

Here are six important components of a DevSecOps approach:

Source Code Analysis – deliver code in small chunks so vulnerabilities can be identified quickly.

Compliance monitoring – be ready for an audit at any time.

Vulnerability assessment – identify new vulnerabilities with code analysis, then analyze how quickly they are being responded to and patched.

Change management – increase speed and efficiency by allowing anyone to submit changes, then determine whether the change is good or bad.

Security training – train software and IT engineers with guidelines for set routines.

Threat investigation – identify potential emerging threats with each code update and be able to respond quickly.

Integration in the workflow:

Organizations that want to unite IT operations, security teams and application developers need to integrate security into their DevOps pipelines. The objective is to make security a core component of the software development workflow, rather than retrofitting it later during the cycle.

Here are a few best practices that will make the DevSecOps process run smoothly:

Carry out threat modeling - Threat modeling exercises can help you to discover the vulnerabilities of your assets and plug any gaps in security controls.

Forcepoint's Dynamic Data Protection can help you to identify the riskiest events occurring across your infrastructure and to build the necessary protection into your DevSecOps workflows.

Automation is good - DevOps is all about speed of delivery, and this doesn't need to be compromised just because you are adding security to the mix.

By embedding automated security controls and tests early in the development cycle, you can ensure fast delivery of your applications.

Use DevSecOps for efficiency - You are only adding security to your workflows. By using tools that can scan code as you write it, you can find security issues early.

While there is still some consensus on what DevSecOps really means for business, it is plain to see its value in a world of rapid release cycles, evolving security threats and continuous integration.

Conclusion: Promising But A Long Way To Go

Some businesses are already seeing positive results as a result of combining development, security and operations teams, shortening feedback loops, reducing incidents and improving security through shared responsibility.

The mindset established by DevSecOps lends itself to a cooperative system whereby business operators are supplied with tools and processes that help with security decision making along with security staff that enable use and tuning for these tools. In this case, security engineers more closely align with the DevSecOps manifesto, which speaks to the value that a security practitioner must supply as well as the changes they must make to enable security value to be supplied to a larger ecosystem.

Secure Transformation:

DevSecOps as a security transformation further lends itself towards cooperation with other security changes as well. In other words, it doesn't matter if you believe that security needs to be added into Development or Operations or some other business process, you are right! Security needs to be added to all business processes and a dedicated team needs to be created to establish an understanding of the business, tooling to discover flaws, continuous testing, and science to forecast how to make decisions as a business operator. Further, for a full transformation to take place, DevSecOps requires Executive Management and the Board of Directors to be involved with information made available as a key indicator of how the business is operating and defending itself within an increasingly competitive low trust environment represented by today's economy.

Like actual highway building, DevSecOps is an ongoing practice with no real conclusion. But as you get into it, you'll seem more opportunities for automating security functions as well as the need to go back and revisit earlier decisions on the journey. Done well, DevSecOps unifies the teams rather than having them at tension with each other. Security is no longer a source of "no", but instead provides valuable tools to the developers and operators, working in concert with them. It's a lot more fun than operating a toll booth!