

DevSecOps - Overview

Data & Reference Collection By

P VinothKumar


01



MISSION: DEVSECOPS

As organizations look to fully harness the power of digital tools and technologies, developers are having to rewire the way they work, along with underlying workflows. One approach that's gained enormous popularity is DevOps. It revolves around faster, better collaboration and communication for teams building, testing, and releasing software. When it's used effectively, DevOps leads to remarkable agility, flexibility, and quality, as well as cost efficiencies.

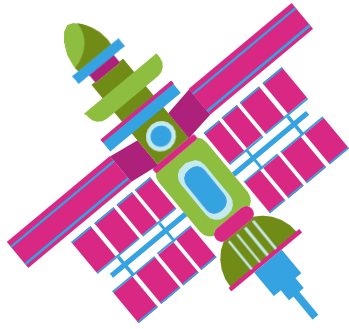
But DevOps also creates new and sometimes formidable challenges. Developers want to innovate and move software and products to market faster, while operations staff prefer to keep things stable and are rewarded for up-time reliability. These two contradictory approaches can make it difficult to align the two groups, and they also make it challenging to address security concerns. That's a serious issue because application-layer attacks were the leading cause of data breaches in 2016.¹



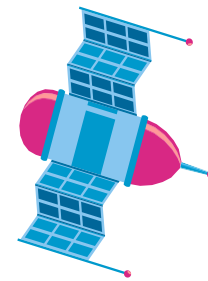
**Application-layer attacks
were the leading cause
of data breaches in 2016.**

-Verizon Data Breach
Investigations Report





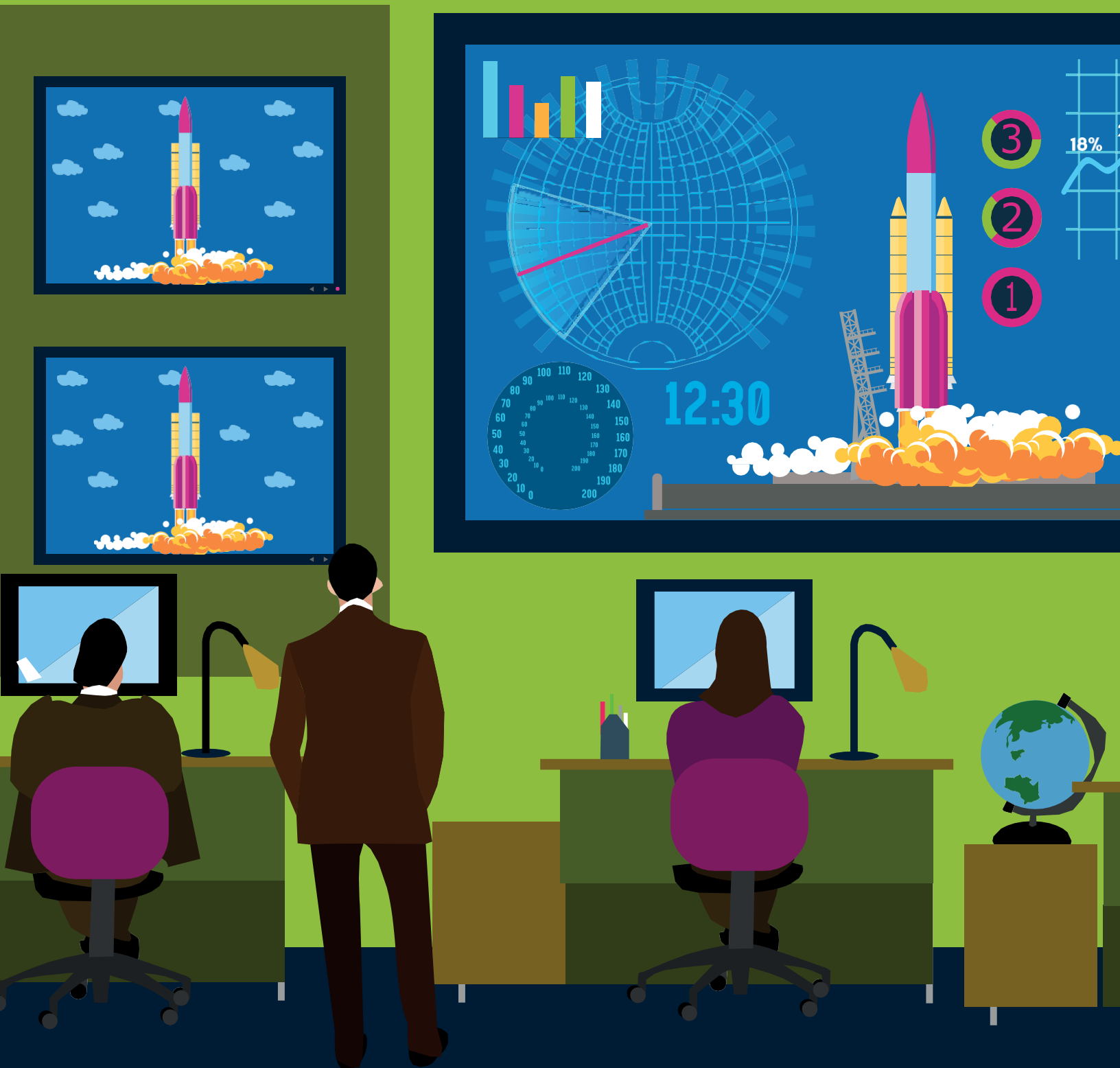
As security becomes ever more mission-critical for development organizations, there's a new trend on the rise — DevSecOps. This guide can help you get ready to launch on your journey to the DevSecOps galaxy. It shows you the strategies and technologies you need to build security into every stage of the DevOps process, so you can produce higher quality code faster and more securely, while reducing costs and meeting compliance. In a best-case scenario, DevSecOps will afford you a distinct competitive advantage.



MISSION TIMELINE



01 | COUNTDOWN DevOps Security Challenges



At the center of a successful DevOps initiative is a simple but often overlooked concept: Because developers drive the software agenda, their participation is crucial for achieving a more secure framework. Yet simply acknowledging this fact won't get the job done. As a developer, you need to position yourself at the center of an application security strategy, and DevSecOps represents the natural evolution of the concept.

You'll face challenges in three key areas when making the move to DevSecOps:



PROCESS

Migrating successfully to DevOps and DevSecOps requires significant changes in governance models, workflows, and processes. A starting point is to recognize that not all teams are created equal. You need mechanisms in place to bridge development groups using robust communication and collaboration tools, reporting systems, and metrics.



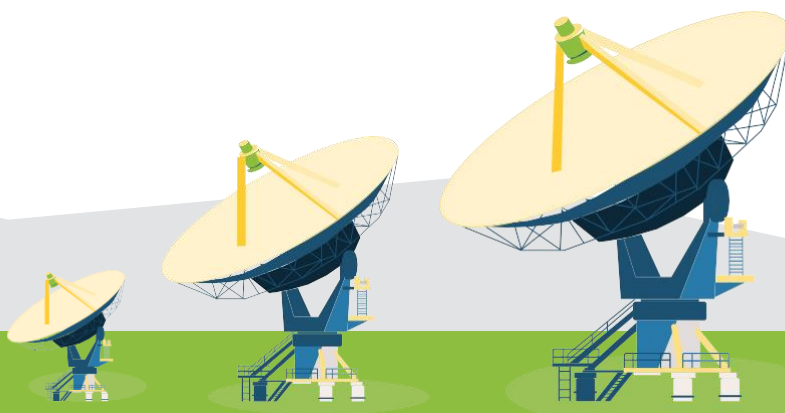
TECHNOLOGY

Software development is a complex endeavor that requires multiple teams, numerous languages and repositories, a complex web of open source code libraries, and sophisticated practices for continuous delivery of new features. Tackling vulnerabilities manually or even in a semi-automated fashion can lead to inconsistent results and gaps in coverage.



CULTURE

Even the best governance framework and leading-edge security tools can't protect an enterprise if the company culture doesn't support it. That's why buy-in and behavioral change is imperative. Ultimately, development teams must change the way they've traditionally worked and interacted with operations, and join forces to deliver value downstream. The security teams, however, face the biggest adjustment. Security people need to abandon the mindset of check-box compliance, or else get left behind as DevOps takes off.



Why DevOps and Security Are Often in Conflict

Today, nobody disputes the need to move fast and deploy code quickly. An Agile framework underpins success in the digital age. But rapid innovation can conflict with stability and security. Without security, DevOps merely introduces vulnerabilities into software faster.

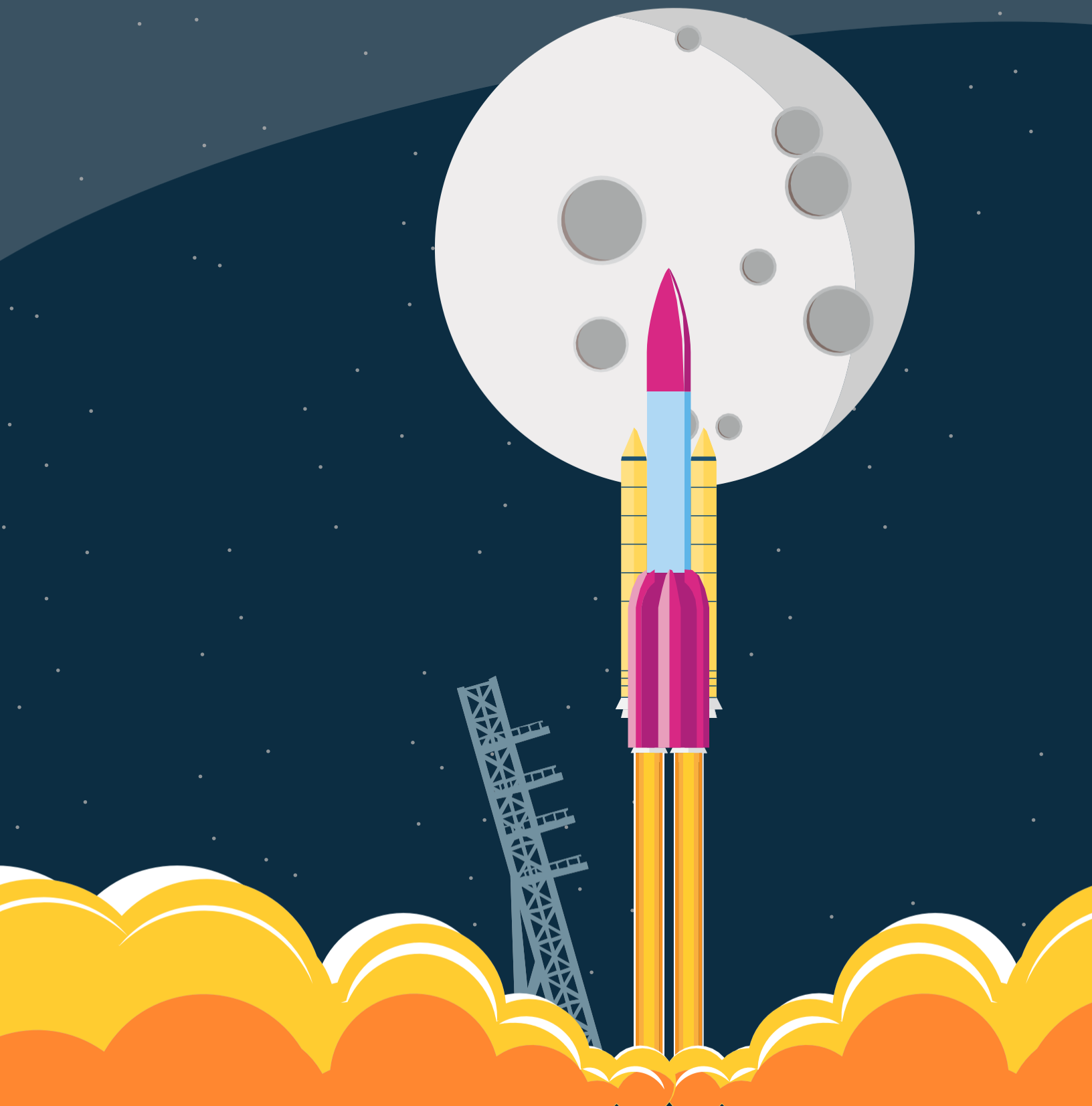
To resolve these conflicts, you need to close gaps in the feedback loop. Too often, connecting and intersecting loops — often between different teams, but even within teams — aren't optimized or integrated effectively. The result? Gaps, glitches, and breakdowns in code development, which result in slower delivery schedules and serious vulnerabilities that create heightened security risks.

In many organizations, the underlying problem is that security isn't addressed until the end of the software development lifecycle. Essentially, the organization reverts to a Waterfall methodology at the middle or end of a DevOps process. Not surprisingly, these two models collide.

The result is that developers often find creative ways to work around security controls that slow them down or create more work. In the quest to move quickly, developers inadvertently create new and bigger problems. In the next chapter, we'll examine best practices for implementing a DevSecOps culture with supporting technology and processes.



02 | LAUNCH Best Practices for DevSecOps



The goal of DevSecOps is to introduce a framework that builds a bridge between fast and secure software development. What does a DevSecOps best-practice model look like? How can you adopt this best-practice approach? Let's examine how culture, technology, and process influence and support a DevSecOps initiative.

Culture Change



Develop a culture of openness and ongoing learning

Within the DevSecOps universe, trust and cooperation are everything. Otherwise, security typically becomes reactive and subpar. No less important: ongoing training and learning. Raising developers' security IQ level pays enormous dividends.



Establish strong feedback loops

The term "ChatOps" is rapidly moving into the developer's lexicon. Chat applications, such as Slack and HipChat, help teams collaborate faster and more effectively. At a higher level, some of these interactions can be automated using chat bots, and teams can begin doing away with legacy systems that hold them back, including email.



Create security champions

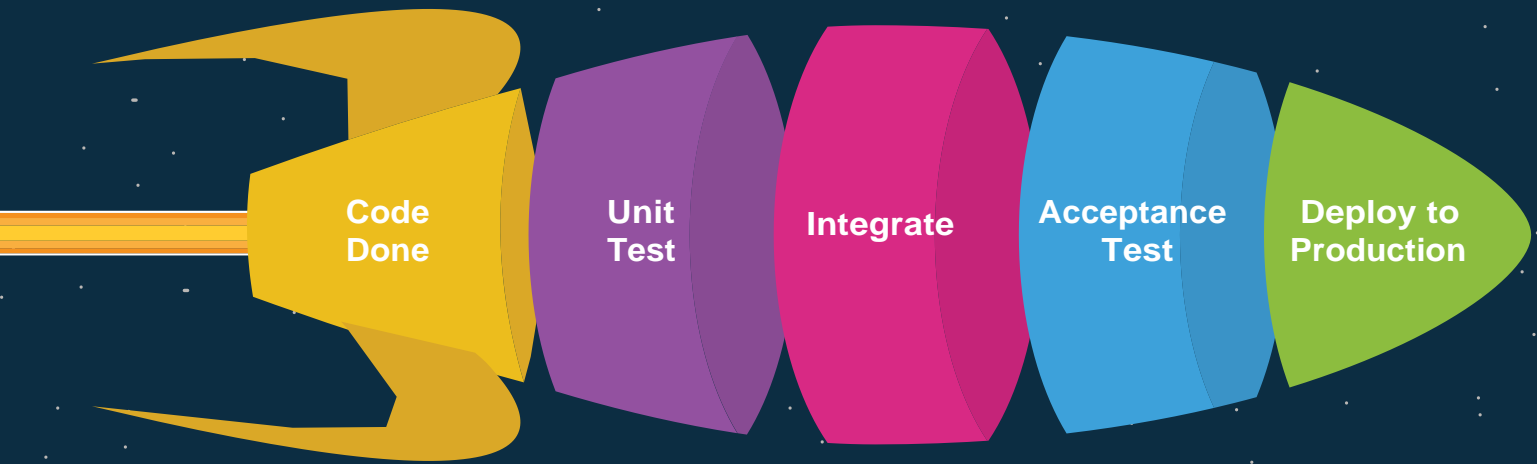
The lack of highly qualified security professionals can make the transition from DevOps to DevSecOps difficult. Savvy organizations identify individuals that understand security within both the Dev and the Ops groups. These individuals serve as champions and continually infuse teams with their knowledge and enthusiasm.



Bolster team autonomy

Successful DevSecOps leaders empower their teams and give them the authority to determine many of their own processes and tools based on their needs. Teams should define their own culture as well. Distributed decision-making forces groups to assume greater responsibility and work in ways that fit their situation.

CONTINUOUS DEPLOYMENT



Technology Transformation



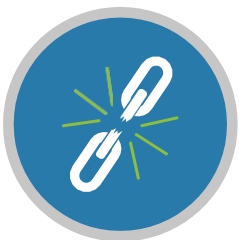
Automate security

The ability to automate security testing through scripting, static and dynamic analysis, composition analysis, and integration of testing within existing tools and processes goes a long way toward identifying flaws early in the lifecycle and speeding up the delivery of secure code.



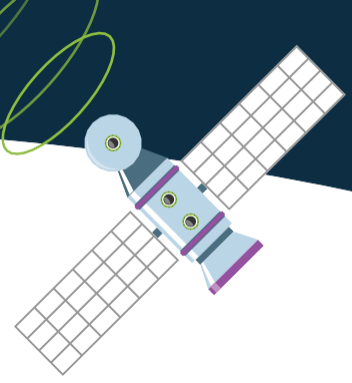
Detect security flaws early

DevSecOps assumes that it's wise to "fail" at the developer's desktop rather than on the customer's laptop or smartphone. Finding code vulnerabilities early requires IDE plugins that deliver instant insights and remediation guidance as problems are introduced.



Break the build

Introducing a security gate in a DevOps build process means that tools can block a release. As a result, they must be configured properly. You also must define and document the exception process because there are essentially two options: Go back and fix the problem — potentially delaying the release — or accept the risk and push out the release. Don't wait to document the exception process until the first time you need it.



Don't accept high false-positive rates

Achieving an effective “break the build” approach requires technology that can deliver valid findings via reports and dashboards, creating operational visibility. Keeping false positives low allows development teams to trust that security tools won't create additional work for them — otherwise, they'll start distrusting and working around them.



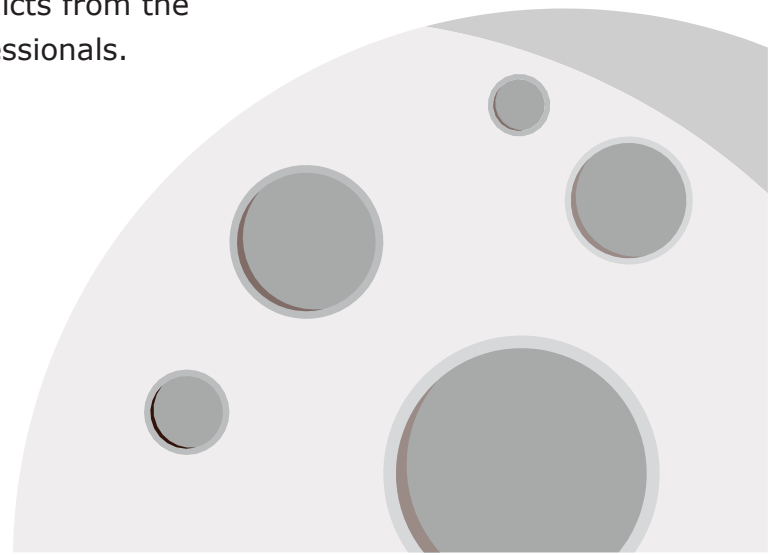
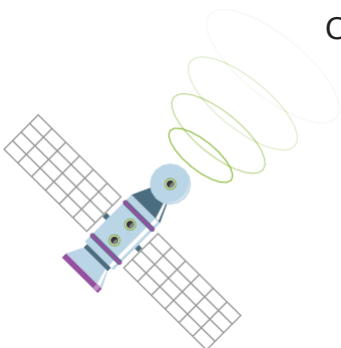
Use composition analysis

Composition analysis tools can scan entire applications and open source components to ensure development teams aren't inadvertently including code with known vulnerabilities. In addition, composition analysis allows you to build an inventory of the components you're using, so it's easier to locate and update them when a vulnerability is disclosed. The March 2017 disclosure of a critical vulnerability in Apache Struts 2 left many organizations scrambling, as attackers began exploiting Struts 2 almost immediately.²



Emphasize orchestration

Today, it's possible to spin up computing power through the cloud, grab code from online libraries, and use automated tools to speed software development. As almost everything, including infrastructure, becomes code, finding and eliminating vulnerabilities is mission critical. Recognize that all systems are prone to bugs and errors. You need to “orchestrate” code and systems during rapid spin-ups and shut-downs. Equally important: Empower teams to act without waiting for edicts from the CISO or other security professionals.



Process Optimization



Adopt a holistic model

DevOps guru Gene Kim says there are three key factors for DevOps success: delivering value to customers (internally and externally); creating feedback loops that lead to process improvement; and molding a culture that encourages people to take risks, learn, and move forward.³ High-performing organizations encourage team autonomy, high levels of communication, responsibility, and accountability — and they offer ongoing learning. These processes contribute to better and more secure code.



Perform regular code reviews

It's important to maintain a big picture view of DevSecOps. While the organization's objective is to move faster, spending additional time identifying high-risk code usually delivers a high ROI. In addition, code reviews boost developer accountability, promote transparency, and reduce the risk of deploying bad code to production.



Experiment with security exercises

Security exercises like Capture the Flag (CTF) and Red Team/Blue Team competitions aren't just fun and exciting, they expose vulnerabilities and help build a culture of awareness. CTFs can involve answering questions, or revolve around hacking or puzzle solving for bragging rights or prizes. Red Team/Blue Team competitions pit an attacking team (Red Team) against a defending team (Blue Team). These competitions give both teams new perspectives on security and encourage cross-pollination of learnings and best practices.

FEEDBACK LOOPS





Measure and benchmark performance

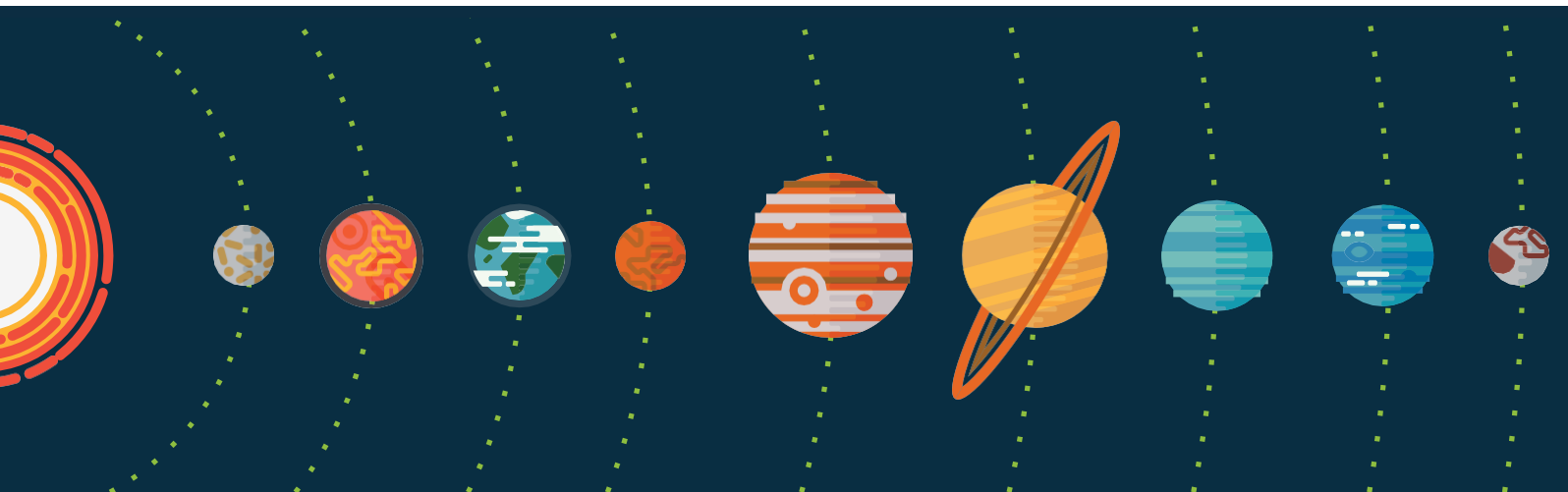
Orchestration doesn't happen in a vacuum. Shorter feedback loops don't occur on their own. A common denominator among DevSecOps organizations is the use of metrics and telemetry. Metrics and measurement tools help teams stay in sync because there's constant and immediate feedback.

Implement mechanisms and procedures for dealing with problems

For example, a group must know how to roll back to a previous release or version if a serious problem occurs. Use canary releases for a limited set of customers to spot problems early, and manage and control the use of binary repositories, third-party code, and open source components. Other important issues include: code grooming guidelines, threat modeling methods, and policies for developers to escalate issues to the security team.

Eliminate surprises for developers and the line of business

When an enterprise promotes the right culture, embeds the right technology, and develops robust processes tied to metrics and key performance indicators (KPIs), the result is a DevSecOps framework that reduces problems and minimizes emergencies. This, in turn, allows teams to focus even more strategically on writing high-quality, secure code.



03 | FLIGHT Model Organizations



Only about 22 percent of organizations have made the switch to DevOps, according to Puppet's *2016 State of DevOps Report*.⁴ Even among those organizations, DevOps is not uniformly used across teams and products. However, there are some examples of organizations that have successfully adopted DevOps and are on their way to DevSecOps. They're demonstrating that a highly-focused approach results in net gains for development teams, the enterprise, partners, and customers.



CAPITAL ONE

The financial services giant, which boasts upwards of 70 million credit card accounts, has fully embraced putting developers in charge of their code. DevOps is at the center of the firm's move into a digital business framework. Capital One CIO Rob Alexander has stated that DevOps helps reduce bureaucracy and technical debt. Capital One moved from a Waterfall approach to a continuous deployment environment that relies heavily on containers, microservices, and cloud technology.⁵ This has helped the company move faster and strengthen its IT and development teams' core competencies.



ETSY

The online merchant — which sells a variety of clothing, accessories, crafts, and other items — had struggled with slow site updates and buggy code that sometimes crashed the site and brought e-commerce to a halt. Rather than risk losing customers and watching the business wither, Etsy tapped a new management team to migrate from Waterfall to Agile. This eventually led to bi-weekly full-site deployments. Now, with a highly-automated development pipeline and continuous delivery, Etsy has reportedly achieved up to 60 deployments a day with near zero interruption.⁶

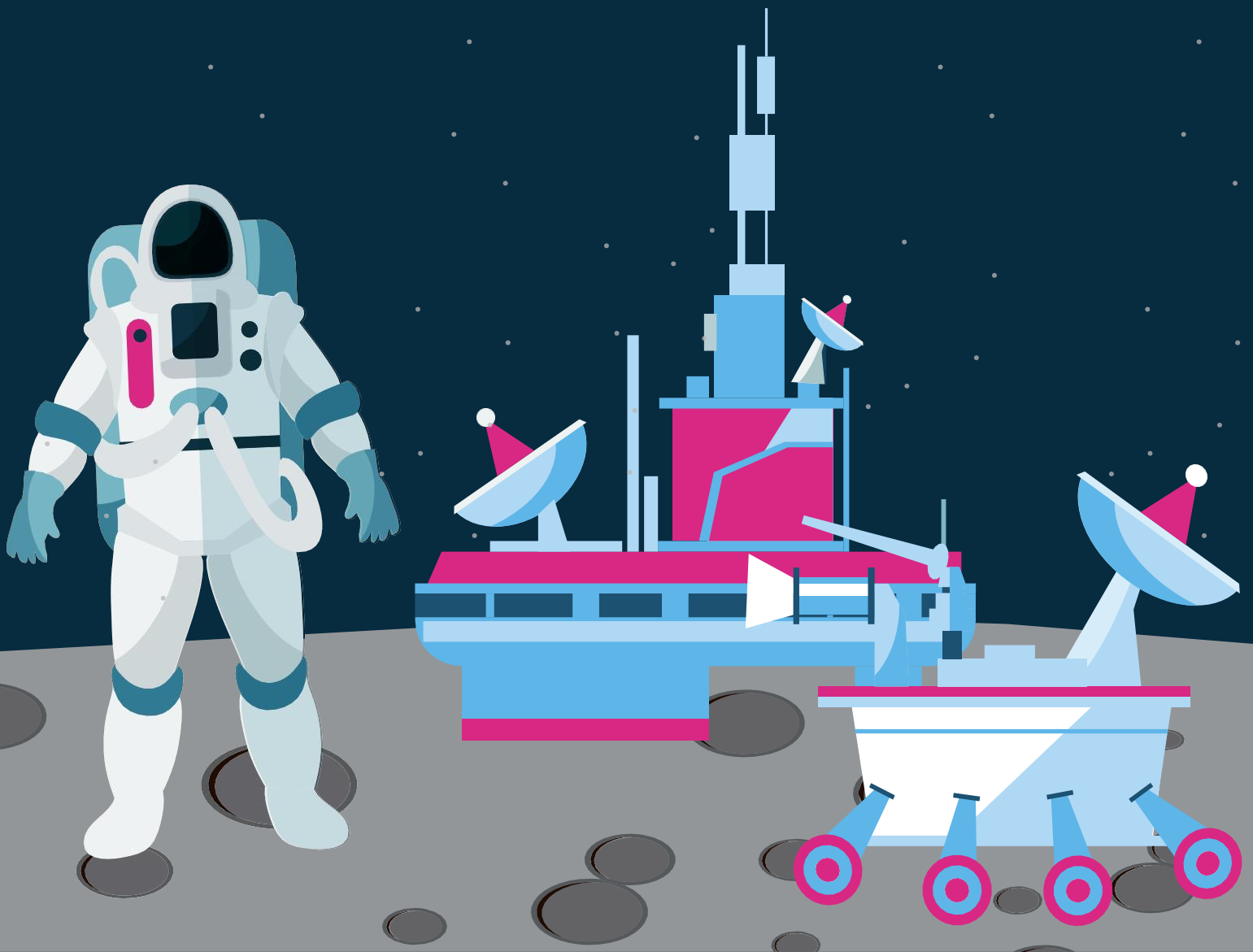




NETFLIX

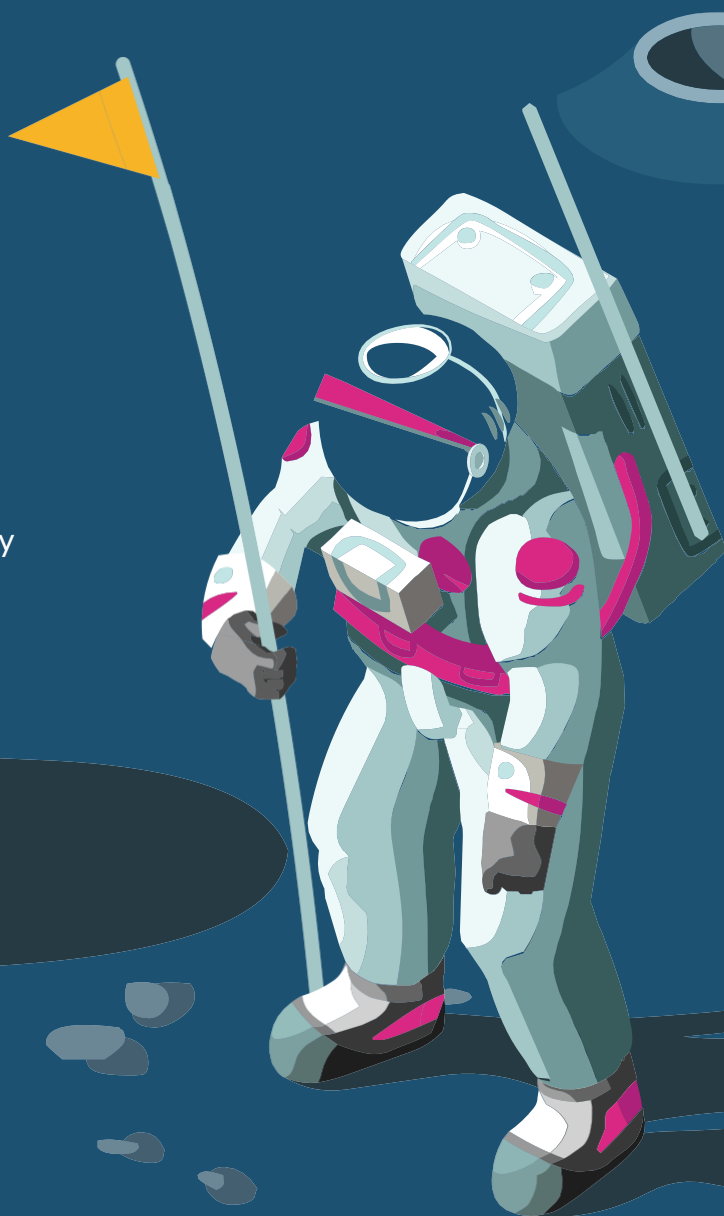
One of the better-known business success stories of the past decade is Netflix, which has transitioned from a company sending DVDs through the mail to a firm that streams video over the internet. The company relies on massive cloud operations, as well as applications that run on a variety of viewing devices, to create a seamless technology interface. DevOps is at the center of everything. The firm relies heavily on open source development, automated tools, a microservices-based infrastructure, and sophisticated testing methods to support projects.⁷ The result is faster deployment and higher-quality code.

04 | TOUCHDOWN Exploring Next Steps



As a developer or development manager, you have a unique opportunity to lead the transformation to DevSecOps. Below is a list of resources and tips that can help you launch your organization, and your career, to the next level.

- Join the Open Web Application Security Project (OWASP).¹⁰
- Volunteer to work on a project that interests you (such as a cross-functional “guild”).
- Become a security champion within your team, department, and organization.
- Get certified, including (ISC)² Certified Secure Software Lifecycle Professional (CSSLP).¹¹
- Read seminal books about Agile, DevOps, and DevSecOps, including “The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations,” by Gene Kim, et al.¹²
- Follow industry blogs and podcasts from leaders in the application security space, including.¹³
- Take advantage of online training, such as free security videos on YouTube, or eLearning offerings from AppSec companies.
- Join an online community such as Peerlyst,¹⁴ participate in an OWASP Slack channel, and consider joining professional groups.



YOUR JOURNEY IS JUST BEGINNING

The journey to DevSecOps presents enormous opportunities and challenges. Ultimately, you have to break down the barriers that block the three Cs of DevOps — communication, collaboration, and cooperation. Developers who help build a framework that supports DevSecOps are poised for a level of speed, innovation, and disruption that puts you and your organization at the forefront of the application economy. Don't panic! Embrace change, and you will be rewarded.

ABOUT ME:

VinothKumar P

DevOps Engineer at Cognizant



References :

1. Verizon Data Breach Investigations Report 2016, press release of June 21, 2016.
2. Zorabedian, John. "Don't Get Zapped by the Struts-Shock Vulnerability Affecting Apache Struts 2.
3. Kim, Gene. "The Three Ways: The Principles Underpinning DevOps," IT Revolution Press, 2016.
4. 2016 State of DevOps Report, Puppet, press release of June 22, 2016.
5. Boulton, Clint. "Capital One Shifts to DevOps to Keep Pace With Customers," CIO, October 25, 2016.
6. Dix, John. "How Etsy Makes DevOps Work," Network World, February 19, 2015.
7. Bergman, Gustav. "Serving 86 Million Users – DevOps the Netflix Way," Lean Magazine, December, 2016.
8. Evolution of DevSecops , Dzone.
9. Nielsen, Anne. "Introducing Automated AppSec Consultation Scheduling," January 27, 2017.
10. <https://www.owasp.org/>
11. <https://www.isc2.org/>
12. Kim, Gene, et al, "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations," IT Revolution Press, 2016.
13. Raghavan, Niru. "Top 20 Security Blogs".
14. <https://www.peerlyst.com>

