KEYFACTOR
SECURE EVERY DIGITAL IDENTITY

IN PARTNERSHIP WITH
DevOps.com
Where the world meets DevOps

EBOOK

How to Enable DevSecOps with Certificate Lifecycle Automation

A GUIDE TO SECURING DEVOPS AND THE CI/CD PIPELINE

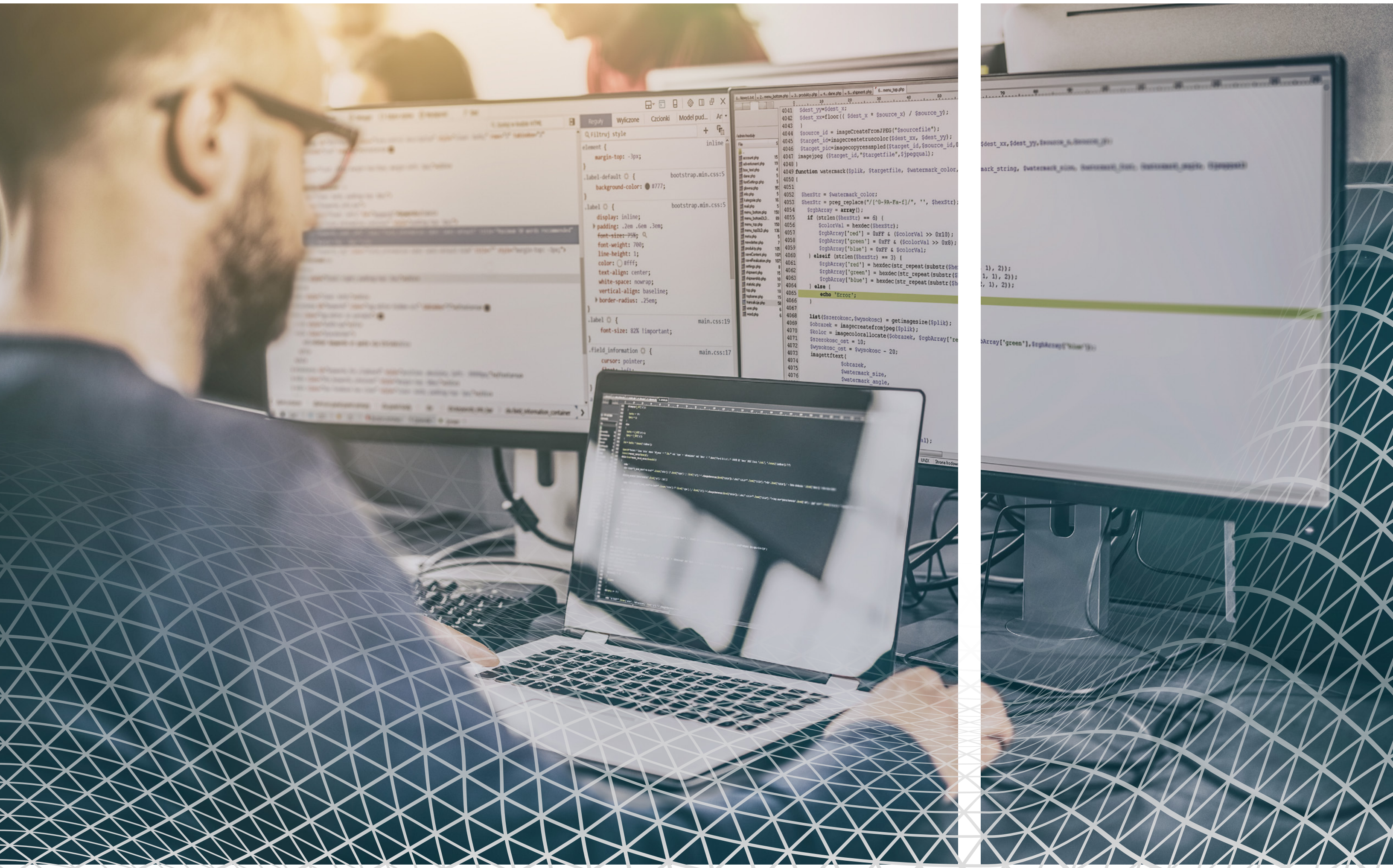


Table of Contents

Introduction

DevOps is everywhere. Not only have more than 50 percent of enterprises adopted DevOps[1], but as a cultural paradigm, it represents perhaps the greatest shift to occur within the technology ecosystem during the last decade. By enabling faster application delivery and greater agility, DevOps has allowed organizations large and small to double down on their cloud computing, application modernization and digital transformation strategies.

Yet despite all of the momentum behind DevOps, as it continues to transform the way organizations leverage technology, there remains one critical area where processes and tooling have not fully caught up with broader DevOps workflows. That area is security. For many companies, enabling a level of security automation that keeps pace with DevOps-driven application delivery and deployment processes has proven challenging.

Addressing this shortcoming is critical for companies wishing to capitalize fully on DevOps. Without ensuring that security is able to keep pace with fast-moving and automated development pipelines, organizations cannot achieve the full benefits that DevOps offers. Nor can they thrive in highly distributed, cloud-based environments that lack rigid boundaries between trusted and untrusted infrastructure, and where “castle and moat” security models, therefore, no longer work.

With these challenges in mind, this eBook explores strategies for addressing the DevOps security dilemma. It focuses in particular on the role of certificate management, a critical yet easy-to-overlook component of effective DevOps security. But it touches as well on the broader context of securing continuous delivery pipelines and embracing best practices associated with the DevSecOps movement.

Ultimately, the following chapters aim to empower modern enterprises to implement a certificate management strategy that enables them to make certificate delivery as seamless and automated as the rest of the DevOps pipeline; to sign code quickly and efficiently in order to defend against abuse; and to enable certificates as a service that can be easily accessed with as much agility as any other DevOps-friendly cloud service. By extension, organizations should be able to build DevOps-based software delivery processes that are faster, less prone to bugs and less dependent on manual oversight.

> “Firms that are serious about improving their security posture should be investing more in automation."[2]
>
> 2019 STATE OF DEVOPS REPORT

[1] https://go.forrester.com/blogs/2018-the-year-of-enterprise-devops/
[2] https://puppet.com/resources/report/state-of-devops-report/

DevOps Defined

Before diving into the details of certificate management, let's first define what we mean when we refer to DevOps. A holistic understanding of DevOps is critical for identifying the ways in which proper certificate management helps to reinforce DevOps processes and best practices.

There is no specific set of tools or technologies required to "do DevOps"; it can be implemented using a variety of different solutions. However, all DevOps workflows are oriented around four central tenets:

COLLABORATION

Doing DevOps successfully means achieving constant collaboration between all stakeholders in the software delivery process. This means that developers, IT engineers, software test professionals and others must be able to communicate seamlessly, as well as coordinate their activities around central goals. Tools can help facilitate this. But ultimately, DevOps collaboration is a cultural paradigm.

AUTOMATION

Automating processes such as code integration, software testing and application deployment is a key component of a DevOps workflow. In particular, "purposeful" automation – meaning automation designed to facilitate specific goals, such as increases in software delivery velocity and improvement in software quality – is essential to effective DevOps.

CI/CD

Continuous Integration/Continuous Delivery, or CI/CD, refers to the set of processes that enable code to be written, integrated, tested and deployed on a rapid basis. Collaboration and automation are prerequisites for successful CI/CD.

TECHNOLOGY

Although, again, there are no specific technologies required for DevOps, a variety of modern tools are commonly used to reinforce the processes and cultural goals associated with DevOps. Containers, serverless functions, microservices architectures and cloud-based development and deployment tools are common examples.

When we talk of DevOps in this eBook, we are referring to software delivery practices oriented around the concepts and tools described above.

DevOps vs. DevSecOps

Although security remains an under-addressed facet of DevOps, it has by no means been entirely absent from DevOps conversations. On the contrary, the intersection of security and DevOps has spawned an entirely new offshoot, now called DevSecOps.

DevSecOps integrates security into the mix by prioritizing processes that make security as seamless and automated as the rest of the DevOps workflow. Like DevOps, DevSecOps doesn't require any specific tools; it can be implemented in multiple ways. At its core, however, are the principles of collaboration between stakeholders around security; the automation of processes that promote security; the integration of security into the CI/CD pipeline; and the use of DevOps-friendly technologies that promote strong security by, for example, enabling rapid response to security issues and minimizing attack surfaces.

DEVSECOPS: CHALLENGES AND TRADEOFFS

DevSecOps is a powerful concept to embrace. But to date, many organizations that have attempted to bake security into their DevOps strategies have tended to run into roadblocks, for several reasons.

One is the inherent tension between security and velocity within DevOps-centered software delivery. Solid software security tends to take time, and time-consuming processes are at odds with the goal of fast, continuous software delivery. Squaring this circle requires finding tools and processes that enable rapid, automated security; but these have proven elusive for many companies.

Another DevSecOps challenge is expertise. Your typical developer or DevOps engineer is not a security expert. As a result, achieving collaboration and automation around security is difficult for many of the stakeholders in the software delivery process, simply because security is not their specialty.

Finally, many of the technologies that facilitate DevOps have security tradeoffs. For example, containers make software deployment agile and faster, but they also introduce more potential security risks because they do not isolate applications[3] from each other and from the host system as strictly as older technologies (like virtual machines) do. Similarly, serverless functions, which help DevOps teams run compute-intensive code quickly and cost-efficiently, can create unique risks associated with DDoS attacks[4] and other types of breaches.

Code signing, too, is a challenge. Distributed development teams must be able to sign code quickly, without disruption. This fast pace means that it can be easy to lose track of signing keys, or to leave them in unsecured locations (such as developer workstations or build servers). This is a lesson that vendors such as D-Link[5] and ASUS[6] have learned in recent years, when unsecured keys gave hackers an opening into private systems.

The bottom line: Achieving DevSecOps by keeping security in step with fast-moving DevOps pipelines has become a widespread goal for many DevOps organizations. But achieving DevSecOps in practice is challenging because, in certain key respects, security is at odds with the central goals and practices of DevOps.

[3] https://cloud.google.com/blog/products/gcp/exploring-container-security-isolation-at-different-layers-of-the-kubernetes-stack
[4] https://thenewstack.io/zombie-toasters-eat-startup/
[5] https://threatpost.com/d-link-accidentally-leaks-private-code-signing-keys/114727/
[6] https://techcrunch.com/2019/03/25/asus-update-backdoor/

How Digital Certificates Enable Secure DevOps Automation

How can companies more effectively square their DevOps practices with DevSecOps goals? A key part of the answer — although it is a consideration that is easy to overlook — lies in digital certificates.

To be sure, digital certificates are likely not the first topic that comes to many DevOps engineers' minds when they think about DevSecOps or security. Like usernames and passwords, certificates, which provide a secure identity so that users, devices, and applications can securely authenticate and connect with one another, may seem like a mundane and relatively minor component of DevOps workflows.

Yet in reality, certificates play a much more prominent role in DevOps than many engineers realize. Consider the following ways in which certificates reinforce DevOps practices:

INFRASTRUCTURE

Provisioning infrastructure automatically and quickly is essential for DevOps. For many DevOps teams, this means relying on Infrastructure-as-a-Service (IaaS) and Infrastructure-as-Code (IaC) solutions, in which infrastructure management and configuration are outsourced to third-party organizations and tools. Under these conditions, being able to verify the identity of the teams responsible for setting up and maintaining infrastructure is critical. Certificates are an effective way to do this.

PIPELINE

As noted above, a CI/CD pipeline that allows code to move automatically from development to testing to deployment is a critical component of a healthy DevOps workflow. A typical CI/CD pipeline involves many moving parts, such as CI servers, test servers and pre- and post-deployment containers. Certificates can help keep track of these various components and ensure that they are authentic, making CI/CD pipelines not only more secure, but also easier to manage.

CODE

DevOps workflows also tend to involve many discrete units of code. Not only is code written in different units, but many CI/CD pipelines include multiple "branches," with each branch dedicated to a different application version or set of features. Keeping track of all of these code components is difficult, as is ensuring all DevOps stakeholders that the code is authentic and secure. Certificates help to solve both of these challenges by allowing DevOps teams to sign code commits securely and automatically.

MICROSERVICES INTEGRATION

Certificates play a critical role in tying together the microservices that comprise modern DevOps apps. SSL/TLS certificates are a core part of applications themselves, and platforms like Kubernetes use certificates internally to separate nodes within clusters.

Respectively, certificates solve the security conundrums and uncertainties that arise within the typical DevOps workflow. By extension, they bridge the gap separating DevOps from DevSecOps.

AUTOMATED CERTIFICATE MANAGEMENT AND DEVSECOPS

Indeed, by automating certificate management, DevOps organizations can achieve the security automation and integration that is critical to DevSecOps. With certificate automation — specifically, by deploying Public Key Infrastructure (PKI)-as-a-Service — enterprises achieve the following DevSecOps goals:

VISIBILITY

Automated certificate management allows stakeholders at all stages of the DevOps workflow and CI/CD pipeline to discover and test keys and certificates for any software components they are working with. In this way, PKI services provide a central resource and single source of truth that allows engineers to keep track of code they are writing and infrastructure they are working with, whether it is during the development, testing or production phase of the pipeline.

POLICY ENFORCEMENT

Being able to enforce security rules automatically and continuously at all stages of the CI/CD pipeline is critical for DevSecOps. With automated certificate management, DevOps teams can write policies that enforce certificate enrollment for any object within their workflows. This approach minimizes the risk of security oversights caused by human error, while also making it easy to enforce security policies consistently across the CI/CD pipeline.

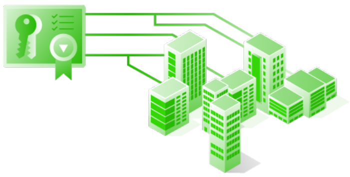
AUTOMATION AND INTEGRATION

Using protocols such as ACME (which stands for Automated Certificate Management Environment), DevOps teams can easily integrate virtually any technology (such as containers, container orchestrators, and secrets-management services) with their certificate enforcement strategy. In this way, integrating disparate DevOps resources together in an automated fashion becomes feasible, and the integration challenges associated with DevSecOps become possible to overcome.

For all of these reasons, digital certificates are critical to allow enterprises to put DevSecOps theories into practice. Automated certificate management certainly isn't the only strategy that helps to make DevOps more secure, but it is one vital component of a healthy DevSecOps workflow.

Meeting DevOps and Security Demands with Keyfactor

Keyfactor provides a complete platform that empowers DevOps and security teams to build and run applications more securely, without compromising productivity. It enables developers to focus on their core competencies, while the security team retains full visibility and control over keys and certificates across every stage of the DevOps process.



CERTIFICATE LIFECYCLE AUTOMATION

With certificate lifecycle automation, Keyfactor enables fast and secure delivery of certificates to any deployment point in the DevOps environment. Security teams can quickly discover, manage, and automate the lifecycle of keys and certificates, preventing the risk of a certificate-related breach or outage, and ensuring end-to-end policy enforcement.

SECURE CODE SIGNING

Keyfactor also provides a cloud-friendly code-signing solution that allows developers to sign any unit of code, at any stage in the CI/CD pipeline, from wherever they are – all while keeping keys protected in a secure vault or HSM. Security and compliance teams can track and monitor every code-signing action, and control access to keys to ensure that only the right developer signs the right code. This helps guarantee the authenticity of code and protects private keys to prevent misuse or theft.

CLOUD-HOSTED PKI AS-A-SERVICE

And because Keyfactor delivers a cloud-hosted PKI service, the platform's features can be accessed easily from anywhere. No matter where your DevOps tools and environments are hosted, and whether they are on-premise or in the cloud, they can be quickly and seamlessly integrated with Keyfactor's PKI service. This hosted PKI solution not only makes certificate management services available from anywhere, but also eliminates the need for DevOps teams to set up and maintain their own PKI infrastructure – a task that would distract from rapid, continuous software delivery.

Why Keyfactor?

✓ ORCHESTRATION

Use flexible tools to automate issuance, renewal and revocation of certificates — including agents, orchestrators, or ACME integration.

✓ COMPLETE PLATFORM

Keyfactor combines Certificate Lifecycle Automation and PKI as-a-Service into a single, cloud-delivered platform.

✓ MODULAR DESIGN

Built on a loosely coupled, modular architecture, making it ideal for dynamic and distributed cloud/multi-cloud environments.

✓ EXTENSIBILITY

An API-first approach makes it easy to incorporate certificates into the CI/CD pipeline, orchestration frameworks, and secrets management engines.

✓ EXPERTISE

We don't just develop software. We've spent more than 18 years working hands-on with enterprises to build, run, and operate their PKI.

✓ SCALABILITY

Because of its modular design, the Keyfactor platform can scale rapidly to DevOps demands — tested and proven in environments with more than 500M+ certificates.

The Bottom Line

Keyfactor is the only company that delivers a cloud PKI as-a-Service platform for certificate management, delivering end-to-end orchestration for every certificate in the enterprise. Learn more by requesting a Keyfactor demo.

Request a Demo ▶

DevOps.com

KEYFACTOR

Keyfactor empowers enterprises of all sizes to escape the exposure epidemic – when breaches, outages and failed audits from digital certificates and keys impact brand loyalty and the bottom line. Powered by the industry's only PKI as-a-service platform, IT and infosec teams can easily manage digital certificates and keys. And product teams can build IoT devices with crypto-agility and at massive scale.

CONTACT US

- www.keyfactor.com
- +1.216.785.2990