SEVEN TIPS TO EVALUATE AND CHOOSE THE RIGHT DEVSECOPS SOLUTIONS

JFrog

SEVEN TIPS TO EVALUATE AND CHOOSE THE RIGHT DEVSECOPS SOLUTIONS

Demand for DevSecOps products has been growing strongly, as more companies realize the importance of integrating security into their DevOps pipelines. However, IT and DevOps pros who dive into the DevSecOps market looking for options quickly realize that the number of DevSecOps tools and frameworks is vast and confusing. This overabundance of choices often leaves them with decision fatigue and analysis paralysis, as they try to understand which security solutions to pick and how to integrate them into their software development pipeline.

But why is DevSecOps becoming such a focus in the first place? To keep up with the pace of innovation, developers have exponentially increased their usage of open source software (OSS), making it now widespread in application development pipelines. As more and more source code comes from the "outside," the need to corral and understand its contents is now mission-critical.

In this ebook, we'll take a look at the types of tools and technologies that should be most successful in mitigating vulnerabilities that are possibly contained in OSS. Then we'll share tips that'll help you separate the wheat from the chaff and make better and more informed decisions when evaluating the many different options available in the market, specifically in the area of software composition analysis (SCA.)

MODERN DEVELOPMENT REALITIES

The typical application today utilizes as much as 90% OSS components, taken from publicly-available open source libraries. This trend is driving up the number of vulnerabilities present in applications — and consequently of exploits and breaches. Companies are reacting by adding more security checks, integrated into their DevOps pipelines.

But what types of tools do security pros and developers really need to ensure the safety and stability of their production software? In fairness, there are several broad categories of DevOps security tools that address different areas of the software development lifecycle (SDLC):

- Code analysis (static and dynamic)
- Software composition analysis (for 3rd party OSS)
- Run-time security analysis (including containers)

Ideally teams should aim to adopt all of these areas for complete SDLC security, but for this blog, we'll focus on Software Composition Analysis, which specifically targets mitigation of vulnerabilities and license compliance violations in OSS components and binaries.

SEVEN MUSTS FOR DEVSECOPS

Here are 7 things you need to ensure as you select your DevSecOps tools:

1. Demand tools that can manage and understand all artifacts natively
Before teams even get to the task of identifying which OSS components have vulnerabilities, they first need a universal DevOps platform that can, as a basic requirement, manage all artifacts and binaries in a central place, regardless of their type and technology. The DevOps platform needs to know which artifacts are used, consumed or created and what their dependencies are.

2. Grab the best fuel
The most effective solutions will require the power of a world class vulnerability intelligence source like VulnDB, to make sure it has the most up-to-date vulnerability knowledge. The best cars in the world are nothing if they don't have great fuel to propel them.

3. Insist on visibility and impact analysis
The DevSecOps "winners" will not only be able to understand which OSS libraries and components your binaries use, but also how to unpack and scan them to see into all of the underlying layers and dependencies – even those packaged in Docker images and zip files. A solution that can understand an organization's artifact and dependency structure can provide visibility and determine the impact of any vulnerability or license violation discovered anywhere in a software ecosystem.

4. Require support for containers and cloud native frameworks
Solutions should support container-based release frameworks, which are fast becoming the de facto standard for cloud native deployments. Deep, recursive understanding of container technology and the ability to dive deeply into each layer will ensure vulnerabilities can't hide. Unfortunately, some scanning tools don't support containers, or don't understand enough about all of their different layers and transitive dependencies.

5. Automate governance

Table stakes in this space are the ability to automate governance in cooperation with a company's security office. A governing system must be able to automatically enforce company policies, and to take action accordingly without intervention. Key features should include:

- Notification of security or compliance violations via different channels, such as email, instant messages, and Jira
- Blocking of downloads
- Failing of builds that depend on vulnerable components
- Prevention of the deployment of vulnerable release bundles

6. Go broad across the pipeline

Differentiators in DevSecOps will be solutions that know how to take this exhaustive data and connect it to the security scans of all the binaries across repos, builds and containers. A platform that can stretch across the whole SDLC and continuously detect and monitor for vulnerabilities and compliance violations, even after production deployment, will stand out from the crowd.

7. Go hybrid

Even if you're not maintaining a hybrid infrastructure yet, you will. Selecting tools and solutions now that support your ongoing cloud journey and hybridization of your infrastructure will ensure you have consistency and standards across your DevSecOps pipelines wherever they may live.

Closing Thoughts

DevSecOps is no longer a wish list item for a CIO. It is now a must-do IT strategy which needs to be an integral part of any SDLC. Even when an organization has chosen the appropriate DevSecOps solutions, leaders need to make sure that they implement a sound DevSecOps process across teams. This includes the need to continually educate developers and DevOps practitioners on application security best practices. Developers outnumber security professionals by 250:1, so distributing security knowledge across development teams is essential to closing the vulnerabilities gap.

Choosing a DevSecOps platform that can manage repositories, binaries, CI/CD automation, and OSS component analysis, and that supports containerized release frameworks can seem a daunting task. Further, supporting your on-prem, cloud, multi-cloud and hybrid deployments is an additional complication. But a checklist of what to demand in a solution is a great place to start. We hope these seven tips will give you a solid foundation for asking vendors the right questions, cutting through the noise in the market, and making informed decisions.

ABOUT JFROG

JFrog is on a mission to enable continuous updates through liquid software that helps empower developers to deliver high-quality applications that securely flow to end-users with zero downtime. Our solutions meet your business model needs and support on-premises, cloud, multi-cloud and hybrid deployments.

JFrog solutions are used by more than 5,800 customers—from startups to large enterprises— who depend on JFrog to manage their binaries for their mission-critical applications. This includes more than 75% of the Fortune 100 companies such as Amazon, Facebook, Google, Netflix, Uber, VMware, and Spotify who put their trust in JFrog.

TRANSFORM YOUR DEVOPS INTO A FULLY AUTOMATED AND INTEGRATED DEVSECOPS PLATFORM

Webinar: DevSecOps Best Practices with the JFrog Platform
Webinar: DevSecOps for Kubernetes-based Applications
Get Started with the JFrog Platform For FREE