masonadvisory

WHITE PAPER

August 2018

# SECDEVOPS – SHIFTING SECURITY TO THE LEFT

**Organisations adopting DevOps can deliver security at speed – they just need to rethink their security model**

**DevOps has brought quicker, more user-focused development. This can seem incompatible with the checks and constraints of security. But with this agile approach, it is more important than ever that security is 'shifted to the left' – meaning it should be included from the very start of the process.**

## Introduction

DevOps is a cultural shift that requires close collaboration between those who develop, deliver and manage applications – ensuring development is aligned to IT operations rather than acting in isolation. Proponents of the DevOps mindset have been able to capitalise on their time to market, and create interactions that truly address consumer requirements.

The areas of continuous integration (CI) – allowing code to be created, committed and tested – and continuous deployment (CD) – moving code from testing to production – have been key to driving DevOps, but often organisations have failed to fully embrace this culture due to other functions such as security, legal and architecture remaining siloed, stuck in a waterfall approach and unable to be scaled. The maturity of a DevOps culture can be measured by assessing how far security has 'shifted to the left' to create something more resembling SecDevOps, where security is no longer an afterthought, but is one of the earliest considerations.

### Key challenges

- Security tooling has not evolved at the same pace as development tooling.
- Security is often seen as a handbrake to the speed offered through a DevOps culture.
- Compliance teams still want to operate in a gatepost fashion ensuring they have the final word on product suitability.

## Security at speed

One would be forgiven for thinking that security and DevOps are incompatible. Start-ups live by the mentality of 'fail fast', but failing fast at security can be the death knell of a business no matter how mature it may be. Shifting security to the left and truly integrating security using a SecDevOps pipeline can ensure products are deployed with adequate protection and provide a way to quickly address any flaws that may present themselves. While the speed of development may scare a traditional security architect, DevOps opens the door to allow security architects and engineers to engage early in the design process to ensure that security is baked into a product or software from the beginning.

The CI/CD pipeline is the backbone of DevOps: for security to be embraced effectively it needs to fit seamlessly into this pipeline creating a process of continuous security. Many security vendors are now releasing tools to assist with security automation which integrate directly into DevOps tools.

## The CI/CD pipeline

While there is no generic CI/CD pipeline, the majority will follow a basic flow, as shown in Figure 1.

- Developer checks out code from source control.
- Developer collects open-source software (OSS) from a repository manager (which may in turn collect from a public repository).
- Source code is tweaked/added to before pushing back to source control.

- CI events are triggers which download new dependencies and conduct unit and integration testing. Feedback is pushed to the developer regarding pass/fail status.
- CD tool then collects the application and runs it through development environment, quality assurance, and user acceptance testing (UAT) before pushing to production.

In addition to this pipeline, numerous cloud-based productivity tools are used for automated tracking and messaging, and the underlying cloud-based infrastructure is exposed to security concerns.

To succeed in bringing security to the left, security professionals must align security at numerous points throughout the CI/CD pipeline in a collaborative and unobtrusive way that maintains the automation and speed of DevOps. The importance of integrating security into this infrastructure cannot be overlooked.

Automating security testing can significantly reduce the chance of basic vulnerabilities being introduced into software which, in turn, reduces organisational risk. Automation can also reduce the requirements for security professionals to be constantly engaged in the development pipeline, allowing them to focus on other areas.

## Introducing security – pre-build

*Securing the environment*

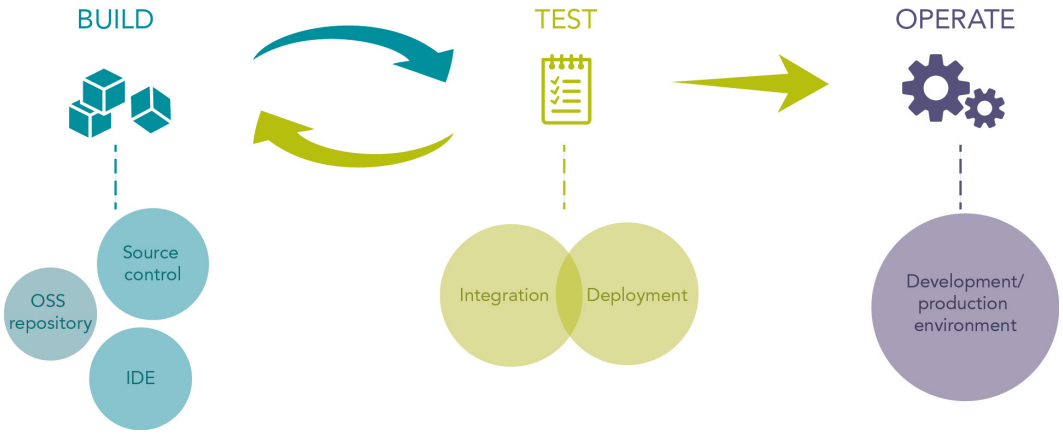
Building a secure application on top of an insecure platform is akin to building Fort Knox on sand: no matter how well designed, there is no escaping the foundations. Most DevOps applications are now hosted in cloud environments, which present many additional risks, but also significant benefits. From the outset it is key to make sure that your supporting infrastructure is correctly configured.

These configurations extend beyond simply locking down ports on a virtual private cloud. They require organisations to consider every element of the supporting environment, from the set-up of the code repository, container stores and CI tools, to the information transmitted to productivity tools.

When creating your environment, the name of the game is automation. The more automation that can be given to the toolsets, the faster the time is to production. This means granting access and privileges for third-party applications to pull code, read from error logs and push code to live environments. In order to do this effectively and in a secure manner, organisations should consider the identity and access management (IAM) profiles that each application requires. It is important to use a true invocation of least privilege, i.e. only allowing access to the minimum number of entities a service needs to conduct its role, as access to one of these third-party services could be catastrophic for down-line production.

Furthermore, the majority of these services still have an administration or root account associated with them, and to ensure they are as secure through the side door as they are through the front, businesses need to continue to leverage two-factor authentication where supported.

**Figure 1: The basic flow for a CI/CD pipeline [Mason Advisory, 2018]**



### *Rapid risk assessments*

Traditional risk assessments are large-scale exercises conducted on a yearly basis that seek to identify risks across a large monolithic application or organisation. In an environment where there are potentially hundreds of code-commits a day this simply does not work. In a SecDevOps environment, businesses need to use rapid risk assessments to introduce an element of risk-based threat modelling.

Breaking down risk assessments into modular rapid exercises allows businesses to address risks as they arise during the planning for a sprint or new microservice. Having developers demonstrate the proposed architecture and discuss possible security concerns will help to develop proactive thought processes in non-security-focused individuals. For example, the Mozilla Rapid Risk Assessment allows you to conduct risk assessments against specific sprint objectives or modular functions in an Agile fashion. It can be conducted in as little as 20 minutes and can then feed into a wider risk assessment framework.

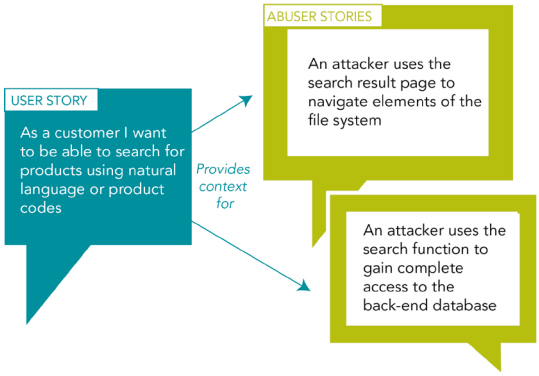
### *Selecting a standard*

The Open Web Application Security Project (OWASP) Application Security Verification Standard (ASVS) provides a basis for testing the technical security of Web applications and gives a detailed list of requirements for secure development. The simple standard gives organisations a yardstick to measure their development against three different verification levels (opportunistic, standard and advanced) which introduces security considerations and goalposts from the start of a development process.

### *Abuser stories and misuse cases*

During sprint planning, developers are used to the concept of user stories and 'definitions of done', but these stories or use cases are based on the premise that the user will interact with an application in the desired fashion. By introducing security considerations in the form of 'abuser stories' and 'misuse cases' (see Figure 2), the non-functional requirements and technical constraints can be considered and scored.

**Figure 2: User requirements can be exploited [Mason Advisory, 2018]**




## Securing the build

To secure the CI/CD pipeline, there are numerous tools coming to market, but security architects must work across people, process and technologies to ensure a cultural shift takes place in support of these technologies, using the approaches outlined below.

**Integrated development environment (IDE).** The IDE should be the first place that security is introduced: using plug-ins that run in the background of the IDE application allows real-time detection and alerting of coding issues. Empowering developers prior to committing code limits the need for bad code to pass through any additional integration testing, while also educating them on potential misconfigurations and helping them to develop better code in future.

**Blocking bad code.** It has been a long time since software was truly 'written': code is imported into development environments from public repositories and re-used across numerous applications. While this increases the speed of development, it means that many applications share core components. If a vulnerability is discovered in one component it can quickly affect many production applications. Security solutions can act as the gatekeeper, ensuring that dependencies and imported code are checked in case they are malicious or are suffering from known vulnerabilities, and triggering alerts if they are incorporated into an application.

**Code coverage and static application security testing (SAST).** Code coverage testing can be used to show how much of the source code of

a programme is executed at run time; the higher the coverage, the less likely there are to be undetected software vulnerabilities. Although an element of SAST will have been conducted within the IDE, using SAST at build can ensure that code (including dependencies) can be checked again in an automated fashion throughout the pipeline.

**Behaviour-driven development (BDD) testing.** BDD is a security testing framework which uses natural language to allow the definition of security-as-code. Security tests can be run in the same way as unit or integration tests and can therefore be implemented within the CI/CD pipeline. Testing security on a unit basis allows development teams to 'fail fast', and the use of natural language means that the tests can be easily understood and reported on to all stakeholders.

**Knowing your sources.** As a delivery squad grows so does the risk. In order to ensure that malicious code does not enter the pipeline from an unknown source, it is important to assure the integrity of any commits. By setting a policy of only using git commits that have been signed by developers with a PGP key, and checking for this PGP key at build time, organisations can secure the very start of the code cycle.

## Securing runtime

Dynamic Application Security Testing (DAST) is concerned with detection of security vulnerabilities at runtime. Using a runtime tool such as OWASP Zed Application Proxy (ZAP), businesses can insert runtime testing into the CI/CD pipeline. Moving from white-box to black-box testing has advantages and disadvantages. Black-box testers traditionally only support in-band testing as they don't have access to the application source code, but some scanners can bridge this gap to consider out-of-band vulnerabilities as well.

In addition, most applications developed using SecDevOps use container technologies such as Docker for deploying microservices. This again changes the security landscape, and the gains made from micro-segmentation of an application can be lost by the reliance upon pre-built images. Using container security technology can assure base images, enforce runtime policies and lock down container network connectivity.

## Recommendations

- **Build the culture, empower the team**. By becoming one of the team and empowering them to make secure decisions, you can reduce the traditional fractured approach that security normally takes in a waterfall organisation.
- **Leverage the CI/CD pipeline to ensure multi-layer security**. The wealth of automated security tools now entering the market means this can truly be brought to bear.
- **Understand the security risks** to your application and apply appropriate, scalable and automated controls to bring this risk in line with your organisation's appetite – but don't slow down production.
- **Adopt a mindset of secure production** and address your security concerns through future sprints.

**Contact us**

Mason Advisory is an IT consultancy that works with clients to solve complex business challenges through intelligent use of IT resrouces. If you would like to discuss the issues raised in this white paper, please email contact@masonadvisory.com or call +44 333 301 0093.

**www.masonadvisory.com**