DEVOPS: THE SECURITY GAP

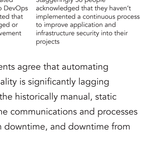
The evolution of cloud computing, SaaS and mobile apps has helped accelerate the transformation of how software is developed and released. It has highlighted the requirement for leaner, more agile ways of working collaboratively across all key teams in the development lifecycle to release competitive, stable products and software release updates on a shorter, more frequent timescale. The DevOps paradigm has done just that: breaking down operational & communication silos between Developers and Operations to establish a shared culture of trust, and automating infrastructure and workflows to create a continuous delivery model where new features are rolled into live software as they are created. But whilst organizations are embracing DevOps to realize compelling business benefits, security and compliance monitoring practices have not kept up and often represent the single largest remaining hurdle to continuous delivery.

TRADITIONAL SECURITY IN DEVOPS IS PROBLEMATIC BECAUSE:

- *IT IS BOLTED ON AT THE END*
- *SECURITY TOOLS ARE NOT AUTOMATED*
- *IT INCLUDES MANUAL PROCESSES*
- *CONTINUOUS DEPLOYMENT STALLS WITHOUT SECURITY AUTOMATION*
- *IT REQUIRES LONG CYCLE TIMES*

SO CAN YOU TAKE A DEVOPS APPROACH TO SECURITY?

Automated software delivery, through short, predictable release times is adding huge value to software developer innovations, but how can security be automated into that process – to ensure secure software code is developed, tested, monitored and released in a continual delivery cycle?



To understand more about the current role of Security in software development Alert Logic conducted a survey asking 73 DevOps practitioners their views on how well the automated development process is working for them, the challenges they are facing, and when/how they are bringing security into that automated workflow (if at all):

KEY FINDINGS

HAD GOOD COMMUNICATIONS BETWEEN TEAMS

Communications and the sharing of common toolsets between the Development and Operations teams were considered to be working well

COMMUNICATIONS WERE BAD OUTSIDE THE TEAM

Communications and toolsets outside of the DevOps teams, with security and compliance were rated as poor, or required significant improvement

BELIEVE INTEGRATION WAS POORLY MANAGED

Of those that have integrated security infrastructure into DevOps process 52 people indicated that it had been poorly managed or needed significant improvement

ADMIT NOT IMPLEMENTING CONTINUOUS PROCESS

Staggeringly 56 people acknowledged that they haven't implemented a continuous process to improve application and infrastructure security into their projects

And this is where the data becomes really enlightening. All the respondents agree that automating security into the development lifecycle is the right thing to do, but the reality is significantly lagging behind. The challenge is that the agile nature of DevOps is at odds with the historically manual, static nature of information security. Security is often siloed and breaks down the communications and processes across development lifecycle – causing the vast majority of critical system downtime, and downtime from security breaches.

SECURITY SHOULD BE INTEGRATED

Key finding: Big disconnect between where respondents believe security should be automated ( ), and where in reality they actually DO automate it ( ):

	% FOR SHOULD AUTOMATE	% WHO DO AUTOMATE
PLANNING	100%	40%
CODE DEVELOPMENT	75%	31%
TESTING	67%	59%
PRODUCT RELEASE	88%	40%
OPERATIONAL REVIEW	92%	16%
MONITORING	100%	23%

WHICH SECURITIES SHOULD BE AUTOMATED?

Key Finding: Big disconnect between which security technologies should be automated into DevOps, and which ones actually have been:

- Network Firewall (83%)
- IDS/IPS (68%)
- Web Application Firewall (62%)
- Network Scanners (45%)
- Log Management (30%)

Of the core security technologies that they believed should be automated in the development lifecycle log management came out as the most important – at 86%. However log management fell far short of the reality – with the lowest number of people actually implementing it – just 30% of respondents

23%

The majority of respondents surveyed cited a lack of people and relevant skills as the biggest barrier to implementing a continuous security model—in fact, only 23% of respondents believe they have the right people with the right resources.

"CONTINUOUS DEPLOYMENT STALLS WITHOUT SECURITY AUTOMATION."

CLOSING THE SECURITY GAP

Development, operations and security are fundamentally intertwined and dependent on each other. The evolution of DevOps should now be extended to embrace Security – providing speed and agility to securing critical applications, assets and services in a more predictable, auditable and secure way.

KEY TAKEAWAYS:

- Security should be involved in the planning stage and early development, to harden the software as much as possible.
- Security teams must standardize secure configuration settings for faster deployments, and continually model potential security threats and vulnerabilities, and test for them.
- Test results should be fed back into the development teams to ensure that software is continually developed to proactively mitigate security threats – minimizing security breaches and all the financial, operational and reputational damage that a breach will cause
- The threat landscape is constantly changing: continuous real-time monitoring is key
- Move to 'security as code' – embedding security into scripts to automate processes that can be executed in a repeatable and predictable way
- Conduct security validation throughout the development lifecycle

Dev'Sec'Ops will be fully realized when organizations stop adding it onto the end of the development lifecycle and start integrating it in so that it becomes a seamless part of the secure continuous delivery lifecycle.

ALERT LOGIC®
Security. Compliance. Cloud.

ALERTLOGIC.COM / U.S. 877.484.8383 / U.K. +44 (0) 203 011 5533