North America II | 2018

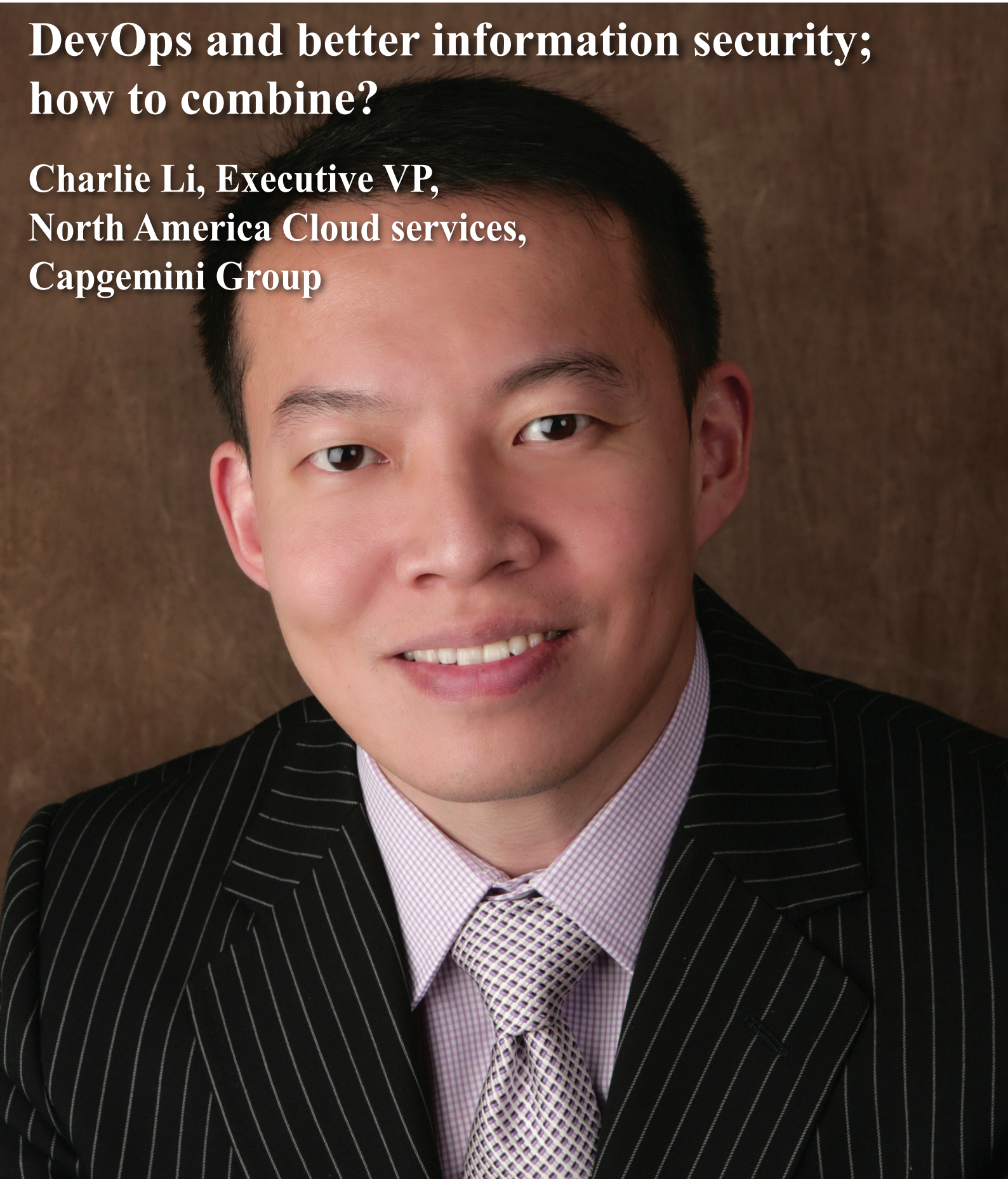
Connect-World

The Magazine that Provides Thought Leadership for ICT Decision Makers

www.connect-world.com

DevOps and better information security; how to combine?

Charlie Li, Executive VP, North America Cloud services, Capgemini Group



CONTENTS

All articles are available for download at www.connect-world.com

DevOps and Security

DevOps Revolution

Outage-Proof Systems

Cyber Attacks

How to address the DevOps information security challenge

by Yuri Gubin, Vice President, Cloud Solutions, DataArt

Problems that arise from rapid growth and disconnected workflows within security functions can be addressed by implementing a DevSecOps model. This model embraces the principle that everyone in the development lifecycle is responsible for security. DevSecOps complements the fundamental concept of DevOps by applying automation to core controls and processes early in the workflow that reduces the chance of the types of misadministration and mistakes which often lead to security issues down the line.

Yuri Gubin is Vice President, Cloud Solutions at DataArt.

Solutions architect with more than 13 years professional experience across financial services, healthcare, travel and IoT industries. Passionate about technology and latest tech innovations; certified AWS Certified Solutions Architect. Skilled in designing solutions through the best use of Big Data, IoT, AI, Cloud and other technologies.

Strong team leader with a MS degree in Computer Science.

The DevOps culture revolutionised the process of software development. With its emphasis on the value of automation, which it applies to testing, integration, deployment and infrastructure management, the approach enabled the development of many new tools and technologies in areas that never existed before. Rather than waiting for a schedule release, a new feature can be deployed the day that development is finished – making it to market faster and ahead of competitors not using a DevOps approach.

However, the DevOps "way" developed so fast, that there was barely time to look back, review and examine retrospectively, asking whether lessons learned from the development of products and operations had been applied to the development of DevOps automation. Among other elements of the "agile" doctrine are automated testing of automation itself and – more importantly – security considerations and practices.

The expectation was that mistakes would happen and would be corrected immediately rather than after waiting for the next development cycle to play out. The theory being that moving fast and breaking things would lead to less failure and downtime. But herein lay a massive vulnerability, DevOps moved so quickly, it forgot the basics. Fortunately, this is not the end of the story.

Problems that arise from rapid growth and disconnected workflows within security functions can be addressed by implementing a DevSecOps model. This model embraces the principle that everyone in the development lifecycle is responsible for security. DevSecOps complements the fundamental concept of DevOps by applying automation to core controls and processes early in the workflow that reduces the chance of the types of misadministration and mistakes which often lead to security issues down the line.

Typically, there a six key security challenges that come up during the implementation process of DevOps. Knowing what these are allows each to be addressed to avoid security issues:

1. Issue: Culture. There is a perception that security will slow down the process of rapid DevOps automation due to increased scrutiny and time-consuming audits. The belief being that a stop for approval or an audit breaks the cycle and prevents improvements that arise from the positive effects of DevOps. This causes information security to sometimes be side-lined, a mere afterthought. And yet, a breach or external threat occurs, and developers can be quick to point the finger firmly at it being a security issue while security experts might put the blame on bad code.

Solution: Leaving security out of the picture is not sustainable and contrary to the view of some, security monitoring is not a barrier to innovation and rapid growth. It should be a goal of DevOps to automate security processes rather than being in conflict with it. Instead of automated tests to cover functionality, IT teams must adopt automated tools for auditing and automated security testing before production so that security is pushed into earlier stages of a release pipeline and back to planning.

2. Issue: Untouched cloud security, security of cloud accounts, keys and user access permissions. Besides the security of software and storage, it is not uncommon for the security of the Cloud environment itself – accounts, keys, permissions - to remain unmanaged.

Solution: Developers should always follow standard security guidelines and cover the cloud environment with standard policies, integrated with corporate identity management systems. In addition, monitoring can be improved by the separation of accounts and resource groups for different departments and environments. Such

models offer a clear understanding behind the allocation of resources, costs of environments and granularity of permissions, which in turn protect the cloud environment itself.

3. Issue: Dangerous 'wow' factor Sometimes teams are very keen to impress, and decision makers go for the brand new and shiny over sensible choices. Let's face it, when it comes to brilliant new technology, security can seem a little boring. Fundamental human psychology is not geared up to risk assessment and imagining possible scenarios that could happen.

Premature technology selection leads to disappointment and frustrating conclusions after it becomes evident that the technology lacks the desired functionality. More importantly, it becomes impossible to continue using the technology when it doesn't support appropriate security requirements and cannot be configured in the desired way. This avoidable mistake is very expensive and time consuming to replace one technology with another. Such replacements always significantly impact the development roadmap.

Solution: The 'Wow effect' features of certain tools should never be the sole, or even the primary, factor considered when selecting technology. Comprehensive prototyping and proper evaluation of technology must be actioned before onboarding and implementation of automation. Each prototype should address certain requirements or concerns. In addition, the security capabilities should be validated along with scalability, performance and functionality.

4. Issue: Technical debts and temporary solutions. With frighteningly common regularity, DevOps teams make the dangerous mistake of corner cutting - introducing quick workarounds during troubleshooting and fixing production issues temporarily on a case-by-case basis. As anyone who's ever put glue on an old chair joint or chewing gum in an engine's radiator knows – temporary solutions become permanent very fast.

Solution: Strongly enforce the rule and culture that security must not be violated, no matter what happens. It is of paramount concern not to expose applications and databases to public access by lifting firewall rules and hardcoding credentials, merely to make things work in "just this one instance". Make sure the seriousness of the inevitable ramifications down the line that would result are drilled into everyone.

5. Issue: DevOps goes too fast, and lacks appropriate design. When a system is developed in an ad hoc way, introducing security will be a slow and struggling process. DevOps automation implements infrastructure management, continuous integration, deployment and testing.

Solution: DevOps should be designed to include blueprints of network topologies, firewalls and cloud environment, artefacts and configuration management. The design should account for security objectives and disaster recovery strategy. Although 'over-architecting' can bring more risks than value, security rules and guidelines need to be clearly and carefully defined. Implementation of DevOps according to design needs to be tied back to security objectives on a continuous basis.

6. DevOps stops at infrastructure and continuous integration. Sometimes the development of applications still uses old techniques and this complicates security auditing. What should be an efficient automated process turns into maintaining a completely different set of guidelines for applications and for infrastructure.

Solution: Automation of security and security management will change the way applications are developed. Where imposed on a system, this solidifies security and enables the entire system to remain uncompromised.

In conclusion, not only do DevOps engineers need to respect security rules, but also security experts need to adopt DevOps culture and onboard automated tools to become part of the common development process. Developers are not security experts just as much as security experts are not developers. Security can be a part of the DevOps process without sacrificing agility. DevOps and DevSecOps are a shift in thinking and culture. No longer is security an afterthought, it is an integral part of the DevOps workflow and is incumbent on all parties involved to collaborate and share responsibility. If companies don't embrace this shift to DevSecOps, they will fall behind.