DevSecOps
An Implementation Strategy
With a Focus on Cultural Implications

6th Annual COV Information Security Conference
Richmond, Virginia
April 12, 2019
Presenters

Eddie McAndrew
COO
AIS Network
(804) 239-5185
Email:
eddie.mcandrew@aisn.net

Barry Davis
CISSO
Virginia Dept. of Social Services
(804) 726-7153
Email:
barry.davis@dss.virginia.gov
Agenda

• Introduction
• DevOps
• DevSecOps
• DevSecOps Culture
• DevSecOps Process
• DevSecOps Tools
• Summary
• Q&A
DevOps (development and operations) is an enterprise software development approach that leverages agile relationship between development and IT operations. The objective is to drive innovation through high velocity delivery of business applications.
What Is DevOps?

- Tools and practices employed to drive high velocity deployment of applications
- Key component of value proposition behind going to the cloud
- Drives Continuous Integration/Continuous Deployment (CI/CD)
- Intended to drive innovation/continuous learning, high-quality applications through flexibility and enhanced competitiveness
Key Elements

- Defining and managing system configuration through code that can be versioned and tested in advance, to increase the speed of building systems and offering efficiencies at scale.

- Using Continuous Integration and test automation to build pipelines from development to test and then to production.

- Creating feedback loops from production back to engineering, collecting metrics and making them visible to everyone to understand how the system is actually used, and using this data to learn and improve.
Agile
- Own set of objectives
- Different methods to achieve goals
- An enabler of DevOps
- Small teams deliver high quality code

DevOps
- Broad concepts
- Have implementations
- Rely on deep communication between software Dev and IT Ops groups
- Rely on automated deployment

- Own set of objectives
- Different methods to achieve goals
- Not contingent upon specific dev discipline
- Principles compatible with Agile
- Considered logical continuation of Agile

Ref: The DevOps Handbook
DevSecOps the Why & the What...

- **Faster** deployment, **rapid** and **continuous** updates and rollout lead to **what**?
  - More potential vulnerabilities
  - Greater potential risk
    - So to drive speed, flexibility & innovation securely -> **DevSecOps**
- DevSecOps – Bridging Agility & Security
- DevSecOps consists of the tools, frameworks and principles for adapting to a high velocity environment
  - Driving *enabled* innovation, flexibility and competitiveness *securely*...

“regardless of the software development and lifecycle management approach, security needs to be built into the software, not bolted on after the fact.”
Hurdles to Using DevOps in Regulated Situations

<table>
<thead>
<tr>
<th>Hurdle</th>
<th>Percentage of Respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DevOps team collaborating with information security</td>
<td>59%</td>
</tr>
<tr>
<td>Automated testing</td>
<td>47%</td>
</tr>
<tr>
<td>Automated release/deployment</td>
<td>47%</td>
</tr>
<tr>
<td>Automating the workflow</td>
<td>45%</td>
</tr>
<tr>
<td>Automation of manual steps</td>
<td>40%</td>
</tr>
<tr>
<td>Modifying architectures</td>
<td>38%</td>
</tr>
<tr>
<td>DevOps team collaborating with compliance team</td>
<td>37%</td>
</tr>
<tr>
<td>Revising the production infrastructure change models</td>
<td>26%</td>
</tr>
<tr>
<td>Education of stakeholders about how DevOps can manage risk</td>
<td>28%</td>
</tr>
<tr>
<td>Tiering systems/services to better balance speed and risk</td>
<td>26%</td>
</tr>
<tr>
<td>DevOps team collaborating with internal audit team</td>
<td>22%</td>
</tr>
<tr>
<td>Rebalanced controls to mitigate risk via detective and corrective means</td>
<td>15%</td>
</tr>
<tr>
<td>Other</td>
<td>4%</td>
</tr>
<tr>
<td>We could not overcome the regulatory hurdles and abandoned that project</td>
<td>1%</td>
</tr>
<tr>
<td>We did not face any regulatory hurdles</td>
<td>8%</td>
</tr>
</tbody>
</table>

Collaborating with legal (1%), collaborating with product engineering and regulatory groups (1%), not sure (2%)

Base: n = 78 Gartner Research Circle Members who use DevOps approach and comply with regulations and/or obligations.

Q65. Did your organization employ any of these strategies to overcome these and/or other hurdles specific to using DevOps in regulated situations?

Source: Gartner (October 2017)
Key Elements of DevSecOps

Culture

Process

Technologies
Traditional Security v. DevSecOps

**Traditional**

In the traditional view of security, operations and engineering must yield to avoid risk. A view might be that of:

- Development
- Security
- Operations

**DevSecOps**

Security must be communicated as a core value – and as a critical enabler.

Collaboration is key!
Communication Is Critical to the Cultural Change

Ref: ISC2 - DevSecOps – Integrating Security into DevOps
Changing behaviors & culture is fundamental to success
Gartner’s 5-Step Approach To Cultural Challenges

01 Gap Analysis
02 Gain Consensus
03 Small, Focused Pilot
04 Incremental Deploy With Feedback Loops
05 Continual Improvements Over Time

Gartner Highlights Five Key Steps to Delivering an Agile I&O Culture; Gartner, 20 April 2015 Press Release
Daily Touchpoints

Wikis, Blogs & Portals

Messaging Apps

Lunch & Learn
By 2018, 90 percent of infrastructure and operations organizations attempting to use DevOps without specifically addressing their cultural foundations will fail.
Security Champions Facilitate a Scalable DevSecOps Program

- Acting as the voice of Security
- Acting as an on-site advisors
- Anticipating potential design or implementation problems
- Deciding when to engage the security team
- Participating in code reviews and threat modeling
- Troubleshooting security bugs

AND MORE!
Cultural changes come in the form of integrating teams that historically have been disparate around a single vision. Technical changes come with automating as much of the development, deployment, and operational environment as possible to more rapidly deliver high-quality and highly secure code.
DevSecOps & Process

• Cultural change must be support by process change
• Security tools must be tightly integrated throughout the DevOps pipeline
• Processes must:
  • Incorporate continuous monitoring and remediation of security defects
  • Continuously test code throughout the life cycle
  • Incorporate automated testing
  • Support Test Driven Security (TDS)
    • https://freecontent.manning.com/where-security-meets-devops-test-driven-security/
  • Support continuous & open communications
• Continual learning & improvement is key

https://freecontent.manning.com/where-security-meets-devops-test-driven-security/
Apply security controls in application and infrastructure layers;

Test them continuously.
Secure Development as a Continuous Improvement Process

Source: Gartner (October 2017)
Gartner’s Ten Things to Get Right....

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Adapt your security testing tools and processes to the developers, not the other way around.</td>
</tr>
<tr>
<td>2</td>
<td>Quit trying to eliminate all vulnerabilities during development.</td>
</tr>
<tr>
<td>3</td>
<td>Focus first on identifying and removing the known critical vulnerabilities.</td>
</tr>
<tr>
<td>4</td>
<td>Don’t expect to use traditional dynamic or static app security testing without changes.</td>
</tr>
<tr>
<td>5</td>
<td>Train all developers on the basics of secure coding, but don’t expect them to become security experts.</td>
</tr>
<tr>
<td>6</td>
<td>Adopt a security champion model and implement a simple security requirements gathering tool.</td>
</tr>
<tr>
<td>7</td>
<td>Eliminate the use of known vulnerable components at the source.</td>
</tr>
<tr>
<td>8</td>
<td>Secure and apply operational discipline to automation scripts.</td>
</tr>
<tr>
<td>9</td>
<td>Implement strong version control on all code and components.</td>
</tr>
<tr>
<td>10</td>
<td>Adopt an immutable infrastructure mindset.</td>
</tr>
</tbody>
</table>
5 Principles for DevSecOps

• Automate security into the process
• Integrate to fail quickly
• No false alarms
• Build security champions
• Keep operational visibility
The Security Professional’s Role

- Enable developers to find and fix security-related code defects
- Govern the use of open source components
- Implement developer training on secure coding
- Manage and report on application security policy, KPIs and metrics
- Understand the requirements for security testing solutions in a DevSecOps environment
- Create developer security champions

DevSecOps Tools – The Third Leg of the Stool

Automated testing is key to driving the DevOps pipeline

As noted - Security tools must be tightly integrated throughout the DevOps pipeline

Testing using tools should be metric driven a few key metrics include:

- Availability: Amount of uptime/downtime in a given time period, in accordance with the SLA.
- Change Failure: Percentage of production deployments that failed.
- Change Lead Time: Time between a code commit and production deployment of that code.
- Mean Time to Failure (MTTF): Time that a system is online between outages or failures.
- Mean Time to Recovery (MTTR): Time between a failed production deployment to full restoration of production operations.
- Number of False Positives: The number of mistakenly flagged vulnerabilities for an application.
- ISC2 list in appendix.
DevSecOps Tools Drive the DevOps Pipeline Via Logging

Logging pipeline

- Analyze usage
- Analyze security incidents

DevOps teams may not know how to identify security breaches, hacking attempts

Log management tool
Reading & parsing logs
Distinguishing unauthorized activity
The Case for DevSecOps

2014 — 16% DevOps teams

2017 — 27% DevOps teams

2018 — 29% DevOps teams

High performers spent 50% less time remediating security issues

This drives the need to:

Involve security and quality teams in the development process early and often.
A Security Strategy for Implementing DevSecOps

**Keys to Successful Implementation**

- **Culture** of Collaboration and Contribution
  - Everyone has something to offer
  - Everyone is responsible for security
  - Goal = safely distributing security decisions

- **Process** – signification changes to existing processes
  - Need mechanisms for communications, measurement, reporting
  - Need to establish a group including Security, Development and Operations
    - This group is responsible for end-to-end security:
      - App development
      - Implementing changes
      - A continuous loop – CI/CD

- **Tools** – required to automate processes for:
  - Managing code repositories
  - Testing – attacking surface analysis, threat modeling, penn & fuzz testing, etc.
Thank You

Eddie McAndrew
COO
AIS Network
(804) 779-7754
Email:
eddie.mcandrew@aisn.net

Barry Davis
CISSO
Virginia Dept. of Social Services
(804) 726-7153
Email:
barry.davis@dss.virginia.gov
Appendix 1 – ISC2 DevSecOps KPIs

Key Performance Indicators

TERMS: DEFINITION

Availability: Amount of uptime/downtime in a given time period, in accordance with the SLA.
Change Failure: Percentage of production deployments that failed.
Change Lead Time: Time between a code commit and production deployment of that code.
Change Volume: Number of user stories deployed in a given time frame.
Customer Issue Resolution Time: Mean time to resolve a customer-reported issue.
Customer Issue Volume: Number of issues reported by customers in a given time period.
Defect Burn Rate: Amount of time to fix vulnerabilities in an application.
Defect Density: The number of bugs identified divided by the codebase of an application.
Deployment Frequency: Number of deployments to production in a given time frame.
Logging Availability: Amount of uptime/downtime of the logging pipeline in a given time period.
Mean Time Between Failures (MTBF): The amount of time that elapses between one failure and the next. Mathematically, this is the sum of MTTF and MTTR, the total time required for a device to fail and that failure to be repaired.
Mean Time to Failure (MTTF): Time that a system is online between outages or failures.
Mean Time to Recovery (MTTR): Time between a failed production deployment to full restoration of production operations.
Number of False Positives: The number of mistakenly flagged vulnerabilities for an application.
Number of Functional/Acceptance Tests: Number of automated functional or acceptance tests for an application.
Number of Passed/Failed Security Tests: Number of automated security tests for an application.
# Security Professional’s role

Today, modern application security programs feature centralized governance by security, but testing and fixing are owned by development in an automated fashion throughout the build process. In this approach, security owns setting policies, tracking KPIs, and providing security coaching to developers.

In addition, security is responsible for providing developers with support in integrating scalable tools into their SDLs. Developers own testing applications in their development environment, fixing flaws to pass policy and continuing to build code.

In this process, security-related defects are just another bug during the build process, and developers have the tools and guidance needed to fix them. At the same time, security can govern the program to make sure KPIs and policies are met.

In this vein, security professionals will have new responsibilities and new skill requirements.

## NEW SKILL REQUIREMENTS

<table>
<thead>
<tr>
<th>Skill Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable developers to find and fix security-related code defects</td>
<td>Ability to provide remediation coaching and guidance on security-related code defects</td>
</tr>
<tr>
<td>Govern the use of open source components</td>
<td>Basic understanding of application development and why and how third-party components are used</td>
</tr>
<tr>
<td>Implement developer training on secure coding</td>
<td>Understanding of the basics of software development</td>
</tr>
<tr>
<td>Manage and report on application security policy, KPIs and metrics</td>
<td>The ability to measure meaningful metrics at each point in the SDLC process</td>
</tr>
<tr>
<td>Understand the requirements for security testing solutions in a DevSecOps environment</td>
<td>Including the need for immediacy and accuracy of results to avoid impacting the delivery cycle — and enable dev to use these solutions</td>
</tr>
<tr>
<td>Create developer security champions</td>
<td>Be empathetic and consultative</td>
</tr>
</tbody>
</table>
DevSecOps Tooling