Cloud Security at Scale via DevSecOps

Integrating Security with DevOps

Camil Samaha, AWS Solutions Architecture
What Is (True) Cloud Computing?

The on-demand delivery of IT resources over public or private networks with zero up-front costs, no long-term contracts, and pay-as-you-go pricing.
AWS Global Infrastructure
New Business Model

Focus on differentiating your company

Innovate at start-up like speed

Reduce risk
Move Fast

OR

Stay Secure
Move Fast AND Stay Secure
Infrastructure Evolution

Then
• Big Perimeter
• End-to-End Ownership
• Build it all yourself
• Server-centric approach
• Self-managed Services
• Static Architecture
• De-centralized Administration

Now
• Micro-Perimeters
• Own just enough
• Focus on your core value
• Service-Centric
• Platform Services
• Continuously Evolving
• **Control Plane API**
Security Evolution

Security as code
Security as Code

1. Use the cloud to protect the cloud

2. Security infrastructure should be cloud aware

3. Expose security features as services via API

4. Automate everything so everything scales
What is DevOps?

Cultural Philosophy  Practices  Tools
What is DevOps?

**Culture**
- Breakdown the barriers
- Work as one team end to end
- Support business and IT agility
- Collaboration & Communication

**Technology**
- Treat Infrastructure as code
- Automate everything
- Test, measure & monitor everything
Security as Code: Innovation, Stability & Security

- Business
  - Build it faster
  - Protect it
  - Keep it stable

- Development
- Security
- Operations

© 2016 Amazon Web Services, Inc. and its affiliates. All rights reserved.
Security as Code: Deploying More Frequently Lowers Risk

Rare release events: “Waterfall methodology”

Frequent release events “Agile methodology”

“Increased risk”

“Minimized risk”
What is DevSecOps?

DevOps = Efficiencies that speed up this lifecycle
DevSecOps = Validate building blocks without slowing lifecycle

Software development lifecycle

developers

delivery pipeline

plan

feedback loop

monitor

test

build

release

Security

customers
Who is DevSecOps?

DevSecOps is

- Team/Community, not a person
- Automated and autonomous security
- **Security at scale**

DevSecOps role

- **Not** there to audit code
- Implement the control segments to validate and audit code and artifacts as part of the CI/CD process
CI/CD for DevOps

Version Control

Package Builder

Deploy Server

AMIs

AWS CloudFormation templates for Environment

Repo

Generate

Install Create

Push

Distributed Builds
Run Tests in parallel

Get / Pull Code

Staging Env

Test Env

Prod Env

Commit to Git/master

Code Config Tests

Send Build Report to Dev
Stop everything if build failed

© 2016 Amazon Web Services, Inc. and its affiliates. All rights reserved.
CI/CD for DevSecOps

Version Control

Code Config Tests

Continuous Scan

Checksum AMIs

AWS CloudFormation templates for Environment

Audit/Validate

Verify

Checksum

Get / Pull Code

Log for audit

Send Build Report to Security
Stop everything if audit/validation failed
Promotion Process in Continuous Deployment

CONTINUOUS DELIVERY
1. CODE DONE
2. UNIT TESTS
3. INTEGRATE
4. ACCEPTANCE TEST
5. DEPLOY TO PRODUCTION

CONTINUOUS DEPLOYMENT
1. CODE DONE
2. UNIT TESTS
3. INTEGRATE
4. ACCEPTANCE TEST
5. DEPLOY TO PRODUCTION
What Does DevSecOps CI/CD Give Us?

• Confidence that our code is validated against corporate security policies.
• Avoid infrastructure/application failure in a later deployment due to different security configuration
• Match DevOps pace of innovation
• Audit and alert
• Security at scale!
AWS CloudFormation primer

Infrastructure is code
AWS CloudFormation Primer

Allows you to define a “template”
- Composed of different “resources”
- Provision that template into repeatable, live, “stacks”.

CloudFormation (CFn) provides a single service interface
- Let CFn perform state changes and govern who calls CFn

Treat as Code
- Check in your templates

CFn templates can hook into external configuration management frameworks
- Jenkins/Chef/Puppet/etc.
Split Ownership Configurations

Who knows your solution best?

- Dev, Infra, Sec…?
- **Delegate ownership**
  - Infra – VPC design, IGW Deployment, Subnets, etc.
  - DevOps – EC2, Elastic BeanStalk, RDS, DynamoDB, etc.
  - OS Patching, Security Agent Deployments, IAM Roles, etc.

Use Yaml and split file into chunks or functions

- Separate file sources with access control – Use IAM/VPC-E/etc.
- Push files -> **Validate** -> Merge files -> **Validate** -> Deploy -> **Validate**

Jenkins for deployment

- **Promotion flows**
  - Move from manual to Automation based on validation quality
- Excellent for merging jobs of split configurations

© 2016 Amazon Web Services, Inc. and its affiliates. All rights reserved.
Merging

From single file or multiple files
- Maintain access control using policies
- Use different source stores if needed

Based on function/state

Reusable patterns

Maintain order, especially of validation
- Security validation last to execute
- Security should always win
Validation

Keep track of what section you are validating
  • Stage vs Prod
  • Merged vs separated

Validate often and log/alert
  • Validate part and end result
  • Run-time validation

Use external agents/services
  • Amazon Simple WorkFlow
  • AWS Lambda
  • AWS Config
  • Amazon CloudWatch Logs
  • Etc.
Where else can this be applied?

AWS CloudFormation

Amazon EC2 Container Service

AWS CodeDeploy

CFn Template

Task Definition

AppSpec File

…and more.
Other tools to keep close

Tracking
- AWS Config Rules *(Preview)*
- Amazon CloudWatch Events
- AWS CloudTrail
- AWS Inspector

Execution
- AWS Lambda

Track/Log
- Amazon CloudWatch Logs
- Amazon DynamoDB

Alert
- Amazon SNS
Why should I do this?

Implement “Compliance Status” for easy overview
• Use predefined checks
• Create extended custom checks
• Fix the issue while checking

Evaluate/remediate changes/events in your account
• Doesn’t replace log analysis (Machine Learning FTW)
• Protect against changes made by (un)authorized accounts
• Automatic remediation for critical events
• Do forensics on the fly

Always Log and Alert!