KickaaS Security with DevOps and Cloud

Rich Mogull
Analyst and CEO, Securosis LLC
@rmogull
Stick with me… I have code and live demos…
The Problem

New Tools and Applications

Threats

Technical Debt
The Problem

New Tools and Applications

Cloud + DevOps
Cloud and DevOps

- Cloud is a new operational model.
- It requires a re-thinking of fundamental architectures.
- DevOps is a new operational framework, highly attuned to cloud.
- Both shatter existing security approaches.
Cloud tourists deploy their existing operational models and frameworks onto a cloud service, losing most of the benefits of cloud.

Typically due to lack of knowledge, institutional momentum, and arbitrary economic models.

Natives vs. Tourists
Why DevOps?
Application Deployment

- Develop
- Commit
- Test
- Deploy
Complexity Breeds Error
Environments, Requirements, and Configurations Drift

- Dev
- Prod

Time

Configuration Delta
Cycle Times Matter

Pre-Alpha  Alpha  Beta  Gold

Agile? Waterfall?  All the same…
The Dev and Ops Problem

- Configurations drift over time.
- The less automated, the more the drift.
- Manual intervention creates error, reduces standardization.
- Environments become de-synced.
- Different teams work in different environments.
- The longer the update cycle, the greater the chances for error.

No standards = More complexity
Enter DevOps

• DevOps is an *operational framework* that increases standardization, agility, and reliability.
• It relies heavily on virtualization, cloud, and automation.
• The principles come direct from Denning and Lean Manufacturing.
Development Pipelines and Continuous Deployment

- Source Code
- Cloud Formation Templates
- Chef Recipes
- Git
- Functional Tests
- Non-Functional Tests
- Security Tests
- Jenkins
- Chef Server
- Test
- Prod
- Chef Server
- Chef Server
- Chef Server

Securosis
- No fixed servers.
- No fixed connections.
- Workloads scattered across IaaS and PaaS.
- All components elastic.
- Host security breaks.
- Firewalls break.
- DAM/WAF break.
- IDS/IPS breaks.
- Assessment and monitoring break.

If you try to apply the traditional security operational framework to the new operational model!
10 Deploys a Day?

Commit

Etsy

facebook

NETFLIX

Adobe
Why DevOps Works

• All environments, including supporting third-party applications, are consistent.

• The deployment pipeline is automated, for code, configurations, and toolsets.

• There is no drift. There are no human-induced errors.

• Faster deployment cycles reduce error and improve business agility.

• Version control and consistency support instant rollback.
The Security Profession Problem

- The discipline that is most resistant to change and least likely to adapt is “Security”
- This resistance is usually excused due to a lack of trust and a reliance on people because we don’t trust security automation.
- “Security” continues to rely on a manual supply chain operated by the “Meat Cloud”
- Trustable automation and an operational model to support it is needed
Our security operational model (tools) and framework (process) are decoupling from the business and rest of IT.
[Sec]DevOps
DevOps is a Security Dream!
Trustable Security Automation!
DevOps + Security = ❤️

- DevOps provides a **consistency** and **control** impossible with manual application deployments.
- Security can easily **embed** and **automate**.
- Security can steal DevOps techniques to apply to diverse workloads and infrastructure requirements.
Automating Security Policies in Chef (or whatever)

1. Launch Instance
2. Inject startup script
3. Pull secure credentials
4. Register with config mgmt server
5. Pull configuration
How do you do this without automation?
Software Defined Security in Action

• Meet SecuritySquirrel, the first warrior in the Rodent Army (apologies to Netflix).
• The following tools are written by an analyst with a Ruby-for-Dummies book.
• Automated security workflows spanning products and services.
Problem: Identify Unmanaged Servers for Compliance

1. Scan the network
2. Scan again and again for all the parts you missed
3. Identify all the servers as best you can
4. Pull a config mgmt report
5. Manually compare results
1. Get list of all servers from cloud controller (can filter on tags/OS/etc).
   • Single API call

2. Get list of all servers from Chef
   • Single API call

3. Compare in code
Problem: Incident Response

1. Detect Compromise
2. Pull server information (If you have it)
3. Quarantine
4. Image
5. Analyze
6. Recover

= Hours!

Each step is manual, and uses a different set of disconnected tools.
1. Pull metadata
2. Quarantine
3. Swap control to security team
4. Identify and image all storage
5. Launch and configure analysis server
6. Can re-launch clean server instantly
A Software Defined Security/SecDevOps Unicorn

- Automating a secure vulnerability assessment involving a cloud service and two commercial security products.
- Open firewall, open host firewall, trigger scan, close firewalls.
Integrating Security into DevOps and Cloud

- DevOps increases **standardization and resiliency**. It is our ally, not our enemy.
- Don’t be distracted by “10 deploys a day”
- Integrate **security testing and configuration** into the DevOps deployment pipeline.
- Directly establish **automated security policy enforcement** in cloud computing environments.
- Use **Software Defined Security** to enhance and automate security operations and workflows.
Presents

KickaaS Security with DevOps

Rich Mogull
rmogull@securosis.com
@rmogull
http://securosis.com
http://devops.com