Rugged Building Materials and Creating Agility with Security

David Etue (@djetue)
Click ‘engage’ to rate session.

Rate 12 sessions to get the supercool GOTO reward.
Rugged Building Materials

• SecDevOps, Rugged DevOps, DevSecOps, DevOps: Whatever you want to call it, we all need security (and compliance)

• Very little security can exist without asset, configuration and change management

• If we write good code, choose our components wisely, and manage it well, what else is left?
“Security” Holding Up DevOps Deployments

<table>
<thead>
<tr>
<th>Issue</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security or compliance concerns</td>
<td>28%</td>
</tr>
<tr>
<td>Difficult to justify from an ROI standpoint</td>
<td>27%</td>
</tr>
<tr>
<td>Organizational complexity</td>
<td>27%</td>
</tr>
<tr>
<td>Identifying the right DevOps consulting firm</td>
<td>26%</td>
</tr>
<tr>
<td>Roles &amp; responsibilities across dev and ops not aligned</td>
<td>25%</td>
</tr>
<tr>
<td>Lack of understanding of the phases of the dev lifecycle and who is responsible</td>
<td>19%</td>
</tr>
<tr>
<td>No support from leadership</td>
<td>18%</td>
</tr>
<tr>
<td>No budget/lack of clarity over whose budget is responsible for what</td>
<td>17%</td>
</tr>
<tr>
<td>Lack of the right skills within development and operations</td>
<td>16%</td>
</tr>
<tr>
<td>No time/resources available to develop a strategy or plan</td>
<td>8%</td>
</tr>
<tr>
<td>Finding collaborative tools to enable the teams to work together</td>
<td>5%</td>
</tr>
</tbody>
</table>

*Figure 6. What are the major obstacles to implementing a DevOps strategy in your organization? Total: 1,425*
Security Struggling With Cloud Too...

Cloud Computing Pain Points – Time Series of Top Categories

- Security: 31% (2H '13), 37% (1H '14)
- Pricing/Budget/Cost: 14% (2H '13), 17% (1H '14)
- Human Change Management*: 12% (2H '13), 12% (1H '14)
- Security of Data, Control of Data Locality, Sovereignty*: 11% (2H '13), 11% (1H '14)
- Compliance: 9% (2H '13), 11% (1H '14)
- Migration/Integration: 10% (2H '13), 12% (1H '14)
- Internal Resources/Expertise: 11% (2H '13), 9% (1H '14)
- Management: 7% (2H '13), 8% (1H '14)
- Lack of Internal Process: 7% (2H '13), 10% (1H '14)
- Vendor/Provider Issues: 3% (2H '13), 7% (1H '14)
- Organizational Challenges: 7% (2H '13), 13% (1H '14)
- Contractual/Legal Issues: 7% (2H '13), 7% (1H '14)


Q. What are your top cloud computing-related pain points? Select up to three. 2H '13, n=117; 1H '14, n=163. * New category in 1H '14.
Traditional Security Controls Don’t Map Well to Cloud and DevOps

Microservices, Agility and Portability Require Focus “Up The Stack”

IaaS and PaaS Redefining How Security Controls Are Evaluated and Deployed (and Who Owns Them)
Security Wants Automation Too…

...They just might not know it yet

- DevOps wants security to be:
  - Orchestrable
  - API-driven
  - Automatically assessed
  - Portable
  - Risk-based / appropriate

- Security wants:
  - Security closer to the data
  - Lower cost of Compliance
  - Analyst productivity
  - Better inventory / asset management
  - More uniformity
  - Faster updates (and patches)
  - Not to be “Dr No”

Big Gap Between Desired State and Security Solutions “As Code”
Core Security Building Blocks

- **Identity** to determine who (or what) did (or failed to do) something
- **Controls** on what privilege users and privilege infrastructure (code) can do
  - Separation of duties
  - Least privilege
- **Encryption** as a tool to separate data (and secrets) from inappropriate access
  - Privilege Users (internal)
  - Privilege Users (cloud / service provider)
  - Government Agencies
  - Adversaries
- **Logging and Auditing** to enable:
  - Granular what, where, when, and how (and sometimes why)
  - Demonstration of compliance
  - Incident response
Identity

- Lots of solutions for humans
  - IAM, PIM/PAM, Cloud IAM, etc.
  - APIs and Provisioning becoming a key platform feature
  - Key focus: He/She Who Can Deploy (or Un-deploy) is god…

- Less solutions for systems, services, processes and things, but evolving
  - UUIDs (or similar) matter
  - Automation means infrastructure and code becoming “privileged”

Credentials To The Production Stack Are Critical!
What Is A Secret

- **m-w.com**: kept hidden from others: known to only a few people
- Examples of Secrets
  - Password
  - Symmetric Encryption Key
  - Private Encryption Key
  - API Key
  - Token

Important Secret attributes:
- Where is it stored?
- Where is it used?
- Who / what is authorized to use it?
- What is it authorized to do?
How Not To Protect a Secret

• Embed it in source code
  – Bonus points for posting to Github once it's in there...
• Put it in a configuration file or script, next to what the secret opens
• Encrypt it with a key embedded in the code (or script)
Protecting A Secret

Attributes of Securing a Secret (from Conjur)

• Self-Auditing
• Fully programmable with fine granularity
• Highly available across any cloud, hybrid, and global architecture
• The secrets should be encrypted when "at rest" in the secrets server
• Each secret should be encrypted with a unique key, which is itself encrypted by a master key (or set of master keys)
• Cryptography should be professionally audited, and ideally open-sourced.
• Secrets should be encrypted in transit, using e.g. TLS
• SSL verification must be ON!

My Addition: Secrets to secrets is a recursive problem…“Distributed” or “derived” secrets should be granular and less trusted.
Secret (and Crypto) Management Systems

• DIY (Do It Yourself)
• Traditional Crypto Key Managers
  – Definitely for “Keys”
  – But also for other objects (e.g., KMIP Blobs)
• Cloud Solutions
  – AWS CloudHSM
  – AWS KMS
  – AWS S3 (+KMS +IAM)
  – Azure Key Vaults…

• Conjur Secrets Management
• Vault from Hashicorp
• KeyWhiz (open source from Square)
• Barbican (OpenStack)
• Chef-Vault?
• And More…

Know Your Capabilities and Security Needs

Join the conversation #gotokdn
Crypto

• Powerful tool, but crypto fail hurts
  – Accidentally destroy a key = destroy data/value
  – Poor implementations easily breakable

Crypto Allows Your To Put Data In Hostile Environment With Near Mathematical Reliability...
If Implemented Properly
Key Hierarchies and Roots of Trust

Trust Anchor (e.g., HSM, TPM, etc.)

Key Management

Applications and Workloads

Most Flexible

Highest Assurance

Key Management and Assurance Levels Matter...
Logging and Auditing

• Can be boring, but is essential
• Great starting point to automate security and compliance testing
• DevOps teams better prepared than anyone—if you can do a rollback…
• Capture and maintain key attributes (6 “W”s)
• Secure / tamper evident
• Work with compliance team to automate reports
Takeaways

• Find common ground with security on security and compliance automation
• Focus on privilege users and infrastructure
• If you have a secret, make it secret
  – Don’t take crypto lightly…
• Make security portable
Please
Remember to rate this session

Thank you!
Thanks!