Modern Governance

Security Differently

John Willis
Global Transformation Office
@botchagalupe
“What would DevSecOps be like if DevOps never existed?”
Blog

What is DevSecOps?

June 01, 2015 / Shannon Lietz

The purpose and intent of DevSecOps is to build on the mindset that "everyone is responsible for security" with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.

Today, these same factors have led traditional security leadership to argue hard for a seat at the Executive table. And while having a seat at the table has increased the effectiveness of security decisions, it's since caused friction and a significant slow down in business outcomes because of a scarce supply of security skill sets to embed in the value creation process. Without enough people, the desired speed by business operators cannot be achieved and a change in how security value is contributed becomes necessary or risks to increase.
“Everyone is Responsible for Security”
Creating Better Pipelines

So how do we design, measure, and improve our pipelines to avoid the above?

Pipeline Design

At Capital One, we design pipelines using the concept of the “16 Gates”. These are our guiding design principles and they are:

- Source code version control
- Optimum branching strategy
- Static analysis
- >80% code coverage
- Vulnerability scan
- Open source scan
- Artifact version control
- Auto provisioning
- Immutable servers
- Integration testing
- Performance testing
- Build deploy testing automated for every commit
- Automated rollback
- Automated change order
- Zero downtime release
- Feature toggle

These gates are used to understand each and every product's progress through the DevOps process.
Industry Working Groups

- Automated Governance (Risk)
- Automated Cloud Governance (Defense)
“Traditional Security Models are Anti-Patterns”
The Three Lines of Defense Model

Governing Body / Board / Audit Committee

Senior Management

1st Line of Defense
- Management Controls
- Internal Control Measures

2nd Line of Defense
- Financial Control
- Security
- Risk Management
- Quality
- Inspection
- Compliance

3rd Line of Defense
- Internal Audit

Source: The Institute of Internal Auditors
“Any organization that designs a system will produce a design whose structure is a copy of the organization's communication structure”
Modern Governance
Modern Governance

● Modern Risk
  ○ Toil and Efficacy related to Risk
  ○ Governance and Compliance
  ○ Attestation and Enforcement

● Modern Defense
  ○ Toil and Efficacy related to Cyber Defense
  ○ MIRTE/NIST/FedRAMP
  ○ Intelligence/Cyber Data Lake

● Modern Trust
  ○ Toil and Efficacy related to Identity
  ○ Platform-Agnostic Authentication, Cryptographic Identities
  ○ Zero Trust Models/Secure Production Identity Frameworks (SPIFFE)
DevOps Automated Governance (Risk)
Verifiable Data Audit

- How do I prove that I’m safe?
- How do I demonstrate that I’m secure?
- How do I know that I can make those statements in a way that’s trustful, that I can actually have evidence that stands behind it?
Move from an implicit trust based model for controls to an explicit, proof based model.

Automated Governance (What)
Automated Governance
(How)

Changing **Subjective**
attestation into **Objective**
attestation
Objective Evidence and Closed Feedback Loops

- Reduce Audit Time
- Increase Audit Efficacy
- Shorten Feedback Loops
- Local Authority
- Minimize Handoffs
- 90% of Controls are Manual
- Enable Trust
Devops automated Governance Reference Architecture

Common Control
1. Access Control
2. Audit Train/log
3. Everything source control
4. Usage policies

Common Actors
1. Auditor, Risk/Compliance Office
2. (system)
3. Tools Admin
<table>
<thead>
<tr>
<th>Attestation Name</th>
<th>Stage</th>
<th>Attestation</th>
<th>Source</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Quality</td>
<td>Development</td>
<td>Numeric</td>
<td>SonarQube</td>
<td>&lt;4</td>
</tr>
<tr>
<td>Information Leakage</td>
<td>Development</td>
<td>Pass/Fail</td>
<td>Custom</td>
<td>pass</td>
</tr>
<tr>
<td>Unit Test Coverage</td>
<td>Development</td>
<td>Percentage Code Coverage</td>
<td>SonarQube</td>
<td>80%</td>
</tr>
<tr>
<td>Unit Test Execution</td>
<td>Development</td>
<td>Pass/Fail</td>
<td>SonarQube</td>
<td>Pass</td>
</tr>
<tr>
<td>Change Size</td>
<td>Development</td>
<td>Pass/Fail</td>
<td>Jenkins</td>
<td>Pass</td>
</tr>
<tr>
<td>Cyclomatic Complexity</td>
<td>Development</td>
<td>Pass/Fail</td>
<td>SonarQube</td>
<td>Pass</td>
</tr>
<tr>
<td>Pull Request</td>
<td>Development</td>
<td>Number of Approvers</td>
<td>Source Control</td>
<td></td>
</tr>
<tr>
<td>Branching Strategy</td>
<td>Development</td>
<td>Pass/Fail</td>
<td>SonarQube</td>
<td>Pass</td>
</tr>
<tr>
<td>Clean Dependencies</td>
<td>Development</td>
<td>Validation</td>
<td>Nexus</td>
<td>validated</td>
</tr>
</tbody>
</table>
## Build Stage

<table>
<thead>
<tr>
<th>Attestation Name</th>
<th>Stage</th>
<th>Attestation</th>
<th>Source</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>Build</td>
<td>ID</td>
<td>Source Control</td>
<td>2.0.3-16-98092ba</td>
</tr>
<tr>
<td>Build Performance</td>
<td>Build</td>
<td>Verification</td>
<td></td>
<td>verified</td>
</tr>
<tr>
<td>Build Version</td>
<td>Build</td>
<td>Version</td>
<td>Source Control</td>
<td>98092ba</td>
</tr>
<tr>
<td>Build Conconfiguration</td>
<td>Build</td>
<td>Config Identification</td>
<td>Source Control</td>
<td></td>
</tr>
<tr>
<td>Linting</td>
<td>Build</td>
<td>Pass/Fail</td>
<td>SonarQube</td>
<td>Pass</td>
</tr>
<tr>
<td>SAST Scan</td>
<td>Build</td>
<td>Validation</td>
<td>SonarQube</td>
<td>Invalid</td>
</tr>
<tr>
<td>Attestation Name</td>
<td>Stage</td>
<td>Attestation</td>
<td>Source</td>
<td>Example</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Artifact Versioning</td>
<td>Package</td>
<td>Pass/Fail</td>
<td>Nexus</td>
<td>Pass</td>
</tr>
<tr>
<td>Package Metadata</td>
<td>Package</td>
<td>Pass/Fail</td>
<td>Nexus</td>
<td>Pass</td>
</tr>
<tr>
<td>Code Signing</td>
<td>Package</td>
<td>Validation</td>
<td>Cryptographic Hash</td>
<td>validated</td>
</tr>
<tr>
<td>Container Scan</td>
<td>Package</td>
<td>Validation</td>
<td>Openscap</td>
<td>validated</td>
</tr>
<tr>
<td>Attestation Name</td>
<td>Stage</td>
<td>Attestation</td>
<td>Source</td>
<td>Example</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>Trusted Packages</td>
<td>Pre-Prod</td>
<td>Validation</td>
<td>Openscap</td>
<td>validated</td>
</tr>
<tr>
<td>Approved Configuration</td>
<td>Pre-Prod</td>
<td>Validation</td>
<td>Ansible</td>
<td>validated</td>
</tr>
<tr>
<td>Threat Monitoring</td>
<td>Pre-Prod</td>
<td>Validation</td>
<td>Tanium</td>
<td>validated</td>
</tr>
<tr>
<td>Autom Alert Tooling</td>
<td>Pre-Prod</td>
<td>Validation</td>
<td>Dynatrace</td>
<td>validated</td>
</tr>
<tr>
<td>Deployment Strategy</td>
<td>Pre-Prod</td>
<td>Validation</td>
<td>Ansible</td>
<td>validated</td>
</tr>
</tbody>
</table>
Risk as Code

- Human Readable (YAML)
- Machine Interpreted
- Version Controlled
- Models Attestations and Enforcement
rode
Services and tools for enabling Enterprise Automated Governance, Policy as Code
🔗 https://liatrio.com  📧 rode@liatrio.com

sigstore
A non-profit, public good software signing & transparency service

Trillian: General Transparency
SUNSPOT: An Implant in the Build Process

In December 2020, the industry was rocked by the disclosure of a complex supply chain attack against SolarWinds, Inc., a leading provider of network performance monitoring tools used by organizations of all sizes across the globe. CrowdStrike and another firm have been supporting SolarWinds in its
<table>
<thead>
<tr>
<th>Tactic</th>
<th>ID</th>
<th>Technique</th>
<th>Attestation</th>
<th>Attestation Source</th>
<th>Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reconnaissance</td>
<td>T1592.002</td>
<td>Gather Victim Host Information</td>
<td>Custom</td>
<td>TSSC/PyPi/Pipeline as Code</td>
<td>StellarParticle had an understanding of the Orion build chain before SUNSPOT was developed to tamper with it.</td>
</tr>
<tr>
<td>Resource Development</td>
<td>T1587.001</td>
<td>Develop Capabilities – Malware</td>
<td>Custom</td>
<td>TSSC/PyPi/Pipeline as Code</td>
<td>SUNSPOT was weaponized to specifically target the Orion build to replace one source code file and include the SUNBURST backdoor.</td>
</tr>
<tr>
<td>Defense Evasion</td>
<td>T1140</td>
<td>Deobfuscate/Decode Information</td>
<td>Configuration</td>
<td>Inspect</td>
<td>The configuration in SUNSPOT is encrypted using AES128-CBC. It contains the replacement source code, the targeted Visual Studio solution file name, and targeted source code file paths relative to the solution directory.</td>
</tr>
<tr>
<td>Defense Evasion</td>
<td>T1027</td>
<td>Obfuscated Files or Information</td>
<td>Configuration</td>
<td>OSCAP, Rekor, Trillion</td>
<td>The log file SUNSPOT writes is encrypted using RC4.</td>
</tr>
<tr>
<td>Defense Evasion</td>
<td>T1480</td>
<td>Execution Guardrails</td>
<td>Code Signing</td>
<td>Cryptographic Hash</td>
<td>The replacement of source code is done only if the MD5 checksums of both the original source code file and backdoored replacement source code match hardcoded values.</td>
</tr>
<tr>
<td>Defense Evasion</td>
<td>T1480</td>
<td>Execution Guardrails</td>
<td>Image Scanning</td>
<td>OSCAP</td>
<td>The replacement of source code is done only if the MD5 checksums of both the original source code file and backdoored replacement source code match hardcoded values.</td>
</tr>
<tr>
<td>Defense Evasion</td>
<td>T1036</td>
<td>Masquerading</td>
<td>Custom</td>
<td>Rekor, Trillion</td>
<td>SUNSPOT masquerades as a legitimate Windows Binary, and writes its logs in a fake VMWare log file.</td>
</tr>
</tbody>
</table>

MITRE ATT&CK Framework Analysis from “Crowdstrike Blog - SUNSPOT: An Implant in the Build Process
Thank you

jwillis@redhat.com
@botchagalupe

linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHat