Implementing SDL at Scale

Secure Development Lifecycle - DevSecOps

Sam Sehgal
Program Leader, SDL automation and DevSecOps
Sep 19, 2022
Dell Context

Scale and complexity

By product heterogeneity

By maturity journeys

By numbers

By tech stack & tools

Pain points

Manual, slow security feedback

Security findings not actionable in CI/CD

Time consuming security assessments

Security control bypassed?
7-point implementation strategy for SDL at scale

1. Solid SDL* foundation
2. Multiple consumption options
3. Customer agnostic architecture
4. Act one team everyday
5. Integrate at LCD**
6. Optimize for DevOps experience
7. Instrument for measurement

*Secure Development Lifecycle
**Lowest common denominator
### SDL Overview

**Foundations**

- **62+ Controls**
  - Compliant
  - Cutting Edge
  - Standard
  - Leading

- **One Standard**
  - Four Maturity Practice Levels

**Objectives**

- SDL for Scale
  - Embrace & Adopt.
- Measure and progress Apps. against four Maturity Practice Levels
- Effectively reduce risk prior to release (GA)
- Invest and train security champions

**DevOps & SRE**

- **Fostering Continuous & Frictionless Security**
  - Threat Modeling
  - Static Code Analysis
  - Open Source Component Mgmt.
  - Container Scanning
  - Web Security Testing
  - Network Vulnerability Scanning
  - Independent Security Testing
  - SDL Security Assessment

**Services**

- Design
- Development
- Testing
- Release

- **Champion**
- **Security Engineer**

**Services**

- DevOps & SRE
  - Fostering Continuous & Frictionless Security

**Five Levels of Security Training**

- L1: Awareness
- L2: Knowledge
- L3: Knowledge/Skill change
- L4: Demonstrate Skill
- L5: Demonstrate Skill
Offer multiple consumption options

1. SDL Engineer Led
   - Led by Security (SDL) Engineer
   - Manual
   - For high value applications
   - Most comprehensive

2. SDL as Self Service
   - Led by Security Champion
   - Manual
   - For all applications
   - Comprehensive

3. SDL as API
   - Performed pragmatically via Git workflows and CI/CD events
   - Automated
   - For apps with frequent & automated deployments
   - Balanced
Customer agnostic automation architecture

DevOps and SRE teams’ development and deployment environment

Security of DevOps processes

Multiple Integration Options for DevOps teams

SEAL – SDL Enforcement and Automation Library

Security of SRE provisioned infra

SDL Foundation
Customer agnostic automation architecture

Integration Options for Dell Products and Dell Digital

On-premise, Cloud deployment env.

Apps, VMs, Containers, Storage, Networking

Security Design and Architecture

Continuous Build and Integration

Continuous Test

Continuous Delivery and Deployment

Continuous verification, monitoring & redeployment

Maintained by Dell DevOps & SRE teams

SDL controls as code (OSCAL compliant YAML)

System (infrastructure) security policy as code (OSCAL compliant YAML)

Provided by Product and Application Security

Security of DevOps processes

Security of SRE provisioned infra

Seal – SDL Enforcement and Automation Library

SDL Verification Tools, Dell Systems & Infrastructure

Security Policy as Code Implementation

Push data to Metrics

Validation and enforcement of golden images

Configuration drift mgmt

API / CLI

CI/CD Plugins

Container Image

API

Docker

Unified SDL Scanning and Controls Verification Interface

API based SDL enrollment

Scan state management

Security Policy as Code Implementation

Generate dependencies, SBOM

Invoke Scans

Taxonomy alignment

SDK Policy validation (pre-deploy)

Infrastructure vuln. scan (on demand)

Continuous infra. vuln. scan

Runtime infra. cfg drift monitoring

Runtime container vuln. defense

IaC policy validation

Continuous Infra. vuln. scan

Runtime app vuln. defense

Fuzz testing

SBOM generation (in progress)

Dynamic app scan

Container image Scan

Open Source / 3rd party software composition Scan

Static application code scan (IDE/Git PR triggered)

Static infrastructure code scan

Static application code scan

Static infrastructure code scan

CI/CD

Triggers

Stages

A

B

B1

B2

On roadmap

On roadmap
Act as one team

• Do not “confront” – Build partnerships instead
• Establish joint scrum teams
• Get into a common backlog
• Resource challenged?
  – Bring security champions to the challenge
  – Reinforce through security awareness and training
Integrate at the *Lowest Common Denominator*

Identify opportunities for platform integration...

...without losing sight of the “downstream” DevOps teams
DevSecOps strategy to scale Dell wide

A win-win-win scenario

Feedback Channels:
- Verbal and subjective
- Programmatic (metrics)

Delivers
- Git + CI/CD platform
- Standardized blueprint for commonly used patterns
- Automatically invoked security scans
- Apply security policy programmatically

Accomplishes
- Platform with security scan built-in and run automatically
- Mitigates security findings in code, open-source components, container images & applications

Product & Application Security

Delivers
- Security scanning services
- Reusable SDL automation artifacts
- Security policy as code
- Threat modeling service
- Secure architecture guidelines

Telecom (TSB)

ISG Products, APEX

Client Solutions Group

Dell.com, Internal apps

Dell Technology Services
## N factor model for optimization

<table>
<thead>
<tr>
<th>Scope</th>
<th>Be precise about “what” needs to security verified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch</td>
<td>Not all branches need same treatment for security verification</td>
</tr>
<tr>
<td>Stage</td>
<td>Allocate security activities appropriately among CI/CD/CD</td>
</tr>
<tr>
<td>Trigger</td>
<td>Security activity triggered with: Time / Git event / Pipeline event</td>
</tr>
<tr>
<td>Frequency</td>
<td>The sweet spot: how often to perform a security activity</td>
</tr>
<tr>
<td>Enforcement</td>
<td>Audit mode vs. Strict mode (block the merge/build/pipeline)</td>
</tr>
<tr>
<td>Packaging</td>
<td>As package / As container image / As code</td>
</tr>
</tbody>
</table>

---
Thank you