Reducing the friction of vulnerability scanning in continuous integration

Allan Cascante
This presentation is for informational purposes only. INTEL MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. No computer system can be absolutely secure.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

* Other names and brands may be claimed as the property of others.

Copyright © 2018, Intel Corporation. All rights reserved.
About Me

@allancascante

http://linkedin.com/in/allancascante
Some Key Terms

- **SAST** – Static Application Security Testing
- **DAST** – Dynamic Application Security Testing
- **Security Testing** – Validating software for vulnerabilities
- **DevOps** – Cultural change to bring development and operations together
- **DevSecOps** – DevOps + Security
- **CI** - Continuous Integration
- **CD** - Continuous Delivery
- **Delivery Pipeline** – Automated Process to Deliver Software.
When developing software

HOW TO GO FAST AND SECURELY?
Continuous Delivery (Pipeline)

*Continuous Delivery*. Reliable Software Releases through Build, Test, and Deployment Automation. by Jez Humble and David Farley.
Lack of alignment

- Different Direction and Goals
- Lack of Alignment
- Cancel each other out
- Feeling of constant work with no real progress
Walls of confusion

Business

Development

Operations

InfoSec
Why DevOps?
DevOps Process

Continuous (days, hours, minutes)

Our Problem

- SAST and DAST process where slow and time consuming
- Deployments were gated due to having to complete Static and Dynamic analysis
- We were asked to go faster but still be complaint with (our) InfoSec requirements
- Save time by automating scan manual process
- DAST & SAST duration was non-deterministic
In the DevOps flow

HOW CAN WE INTEGRATE SECURITY GATES?
Continuous Delivery (Pipeline)

*Continuous Delivery. Reliable Software Releases through Build, Test, and Deployment Automation. by Jez Humble and David Farley.*
Static Application Security Testing

- Find security bugs
- ‘Faster’ inside out
- Reads your code
- Works at rest
Commit Stage

Commit Stage

Commit • Compile • Tests • Assemble

Code Analysis
- SAST
- Code Quality
Integrated SAST Process
Tools to Integrate your Own

- Git (Git Hub*)
- Jenkins*
- SonarQube*
- Any OWASP SonarQube Project Plugin

*Names and brands are the property of their respective owners
Open Source Alternative
Dynamic Application Security Testing

- Find ‘other’ security bugs
- ‘Slower’ outside in
- Plays with your application
- Works at play
Acceptance Stage

- Configure Environment
- Deploy Binaries
- Smoke Tests
- Acceptance Test
- DAST
Integrated DAST Process
Integrating more security validations into our delivery pipeline

GOING FURTHER, SECURITY TESTING
Why?

• Enhanced assurance
• Faster feedback
• Innovation
• DAST has some ‘deficiencies’
ZAP Integration into our pipeline
Advantages in the new approach

• Acceptance test allow a ‘knowledgeable’ scan with ZAP
• Reporting from ZAP integrated into builds give traceability
• Easy integration, just needed to change proxy settings into the testing boxes
Some Highlights

• While DAST and SAST showed no issues, ZAP reported vulnerabilities.
• ZAP approach turned to be faster than DAST or SAST scans.
• ZAP scan duration is deterministic (same as acceptance tests).
• According to State of DevOps high performer teams spend 50% less time remediating security issues.