DEVSECOPS – USING CONTAINERS TO SPEED UP YOUR TESTING

Tim Chase
Director of Security – AppSec and Cloud
Nielsen
**DevOps** is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity – AWS Blog
What is DevOps

- Brings Dev, Testing, Ops together
- Culture Change
- Goal is to reduce cycle time
What is DevOps

“Until code is in production, no value is actually being generated, because it’s merely WIP stuck in the system.”– Gene Kim
Conventional AppSec Testing
Conventional AppSec testing is too slow
Traditional AppSec Testing Doesn’t Work

• Slow
• Too many people involved
• Manual
• Not flexible
DevOps forces security to change

• Faster
• More Accurate
• Flexible
• Shift Responsibility
DevSecOps enters the picture

- Incorporate security principles into DevOps
- Make developers responsible for security
- Move security team to auditors/SMEs
- Shift in culture
What’s the magic DevSecOps solution?

- No silver bullet
- It's about what works for you
- Keys
  - Increase Dev Responsibility
  - Automated
  - No PDFs!!
  - Flexible
One possible solution is containers
What's a container?

EXAMPLE OF A CONTAINER

Packages apps into a single container for deployment
Containers + Security

- Determine application set
- Group all of your applications together
- Install in a Docker container
- Configure in build automation tool
Security Container Example
SIMPLE EXAMPLE
Tools Needed

- Jenkins
- Docker
- OWASP Glue
- Jira
OWASP Glue

- Helpful for creating a pipeline of tools
- Docker container with security tools installed
- Easily extendable to new tools
OWASP Glue

- Examples of tools
  - Brakeman
  - FindSecBugs
  - NSP
  - Contrast (commercial)
  - Checkmarx (commercial) eslint

- Export
  - Jira
  - JSON
  - CSV/Text
Install Docker
Run Glue

- Decide what tools you want to use
- Decide how to export your data
- Build your command……

```bash
docker run -v "${PWD}":/src/ --rm owasp/glue -d -t OWASPDependencyCheck -d /src -f jira --jira-api-url https://adlm.company.com --jira-api-context "/jira" --jira-username "userID" --jira-password "Password123" --jira-project Project
```
Run Glue

docker run -v "$PWD":/src:--rm owasp/glue -d -t OWASPDependencyCheck -d /src -f jira --jira-api-url https://example.com/jira --jira-username <redacted> --jira-password <redacted> --jira-project BAPS --jira-component defect

Loading scanner...
Logfile nil?
calling scan
Running scanner
Mounting ... /src
Mounting target: /src
Checking about mounting /src with #<Glue::DockerMounter:0x000000022ef6d8>
In Docker mounter, target: /src became: ... wondering if it matched .docker
Checking about mounting /src with #<Glue::FilesystemMounter:0x000000022eef8>
Mounting /src with #<Glue::FilesystemMounter:0x000000022eef8>
Mounted /src with #<Glue::FilesystemMounter:0x000000022eef8>
Processing target.../src
Running tasks in stage: wait
Running tasks in stage: mount
Running tasks in stage: file
Running tasks in stage: code
code = OWASPDependencyCheck - #<Set:0x000000022b4948>
OWASP Dependency Check
Parsing report /src/dependency-check-report.xml
Fingerprint: CVE-2017-8886:postgresql-9.4.1208.jar
Push Glue to Jira
Connect Glue to Jenkins

- To make truly DevOps, connect to automation
- In Jenkins, add a Build Step
- In the build step, execute a shell command that runs your docker command
Connect Glue to Jenkins
Security Container Example

- DevOps Team
- Jenkins
  - git
  - Maven
  - Unit Tests
  - Find Security Bugs
- Security Container
  - OWASP ZAP
- JIRA
- DevOps Team
Why a Security Container?

- Easy change of tools
- Easy rollout (i.e. prereq handled)
- Easy update
- Flexible for different environments
- Automated and integrated
- Cost
Where to Start

Start Easy

• One App to test
• Determine Tools for container
• Integrate with Dev Team
Where to Start

- Open source container: OWASP Glue
- Support out of box for many free tools
  - PMD
  - FindSecBugs
  - Brakeman
  - RetireJS
  - Dependency Check
Work to mature solution

- Add more tools
- Add different types of tools
  - Dependency Checks
  - Code quality checks
  - Dynamic scans
  - Gauntlt
- Fail the build
Keys to success

- DevOps mature
- Security champion on teams
- Really smart people
- Really technical people
- Flexibility
- Be OK to fail
Apply What You Have Learned Today

- Next Week
  - Try the solution with one application

- In the next 3 months
  - Determine standard toolset for one group of apps
  - Create and rollout container for similar apps

- In the next 6 months
  - Review success of initial container rollout
  - Determine plan for remaining applications