OWASP Indonesia Meetup

Strengthen and Scale security using DevSecOps

www.teachera.io  @secfigo  secfigo@gmail.com
Mohammed A. Imran
Senior Security Engineer

# whoami

Author, Speaker and Community Leader.

Speaker/Trainer at Blackhat, AppSec EU, Pycon, All Day DevOps, DevSecCon London, DevSecCon Singapore, Nullcon etc.,

Organizer of DevSecOps Track in OSS 2018.

Project Leader for OWASP DevSecOps Studio, DevSlop, Integra and Awesome-Fuzzing projects.

Organised around 100 monthly security meetings and about 50 workshops.

SCJP, OSCP, OSCE. AWS-CP, AWS-CSA, AWS-SS
Agile and DevOps
Long Long time ago

Trivia: how is this related to Singapore?
Traditional SDLC

- **Requirements**: Gather Requirements from the client/customer
- **Design**: Design the software according to the requirements
- **Implementation**: Implement the design agreed upon
- **Deploy**: Deploy the software to the production
- **Maintain**: Maintain of the software in production
Then Agile Happened

Everything changed after agile, much shorter development cycles and faster deploys to production.

Speed with which changes are being made is beyond security's operations reach.
Wall of confusion
DevOps

**DevOps** is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality. - Bass, Weber, and Zhu
Plan & Create
Plan and implement the code using source code management (SCM)

Verify
Test and verify the code does what business wants.

Package
Package the code in a deployable artifact & test it in staging environment

Monitor
Monitor the application for its performance, security and compliance

Configure
Configure the application/stack using configuration management

Release
Release the artefact as production ready after change/release approvals

DevOps Cycle
Do you speak Chinese?
DevOps

Wall of compliance

Security
Tradition Secure SDLC

<table>
<thead>
<tr>
<th>1. TRAINING</th>
<th>2. REQUIREMENTS</th>
<th>3. DESIGN</th>
<th>4. IMPLEMENTATION</th>
<th>5. VERIFICATION</th>
<th>6. RELEASE</th>
<th>7. RESPONSE</th>
</tr>
</thead>
</table>
Security is Outnumbered!

Dev / Ops / Security

100 / 10 / 1
DevSecOps

DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality - Bass, Weber, and Zhu

By definition, security is part of DevOps.
Flexibility

With ever changing technology, businesses have to be flexible and fast to deliver value to their customers otherwise they risk losing the business.

Resilience

DevOps helps organisations in designing and implementing resilient systems.

Speed

Speed is competitive advantage and DevOps helps to go to market faster.

Automation

Automation helps to reduce complexity of modern systems and can scale as per needs

Reliability

Customers need more reliable & available systems. DevOps reduces failure rates and provides faster feedback
How to DevSecOps?

**Culture**
DevOps is about breaking down barriers between teams; without culture other practices fail.

**Measurement**
Measuring activities in CI/CD helps in informed decision making among teams.

**Automation**
Often mistaken as DevOps itself but a very important aspect of the initiative.

**Sharing**
Sharing tools, best practices etc., among the teams/organization improves confidence for collaboration.

Core Values of DevOps
Build bridges, not walls!
Build guard rails, not gates!

Embed security early and often
Conway’s Law

Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.
Continuous Integration/Deployment
CI/CD

**Plan**
- Requirements
  - Functional req.
  - Non Functional req.
  - Design
- Code Repository
- Code Server
  - Compile
  - Basic tests
- Branching
- Third party components
- Hooks

**Code**
- Code Repository
- CI Server
  - Compile
  - Basic tests
  - Lint(analyze)
  - Package
  - Security

**Build**
- Integration Testing
  - Integration
  - Performance
  - Security
- Test on staging
- Release
- Schedule

**Test**
- Artefact Repository
- Test on staging
- Release
- Schedule

**Release**
- CD Orchestration
  - Configuration
  - Inventory
  - Infrastructure

**Deploy**
- Monitor
  - Metrics
  - Monitoring
  - Alerting

**Operate**
Scale security with DevOps
DevSecOps Implementation

So far we have looked at Principles and Ideas behind DevSecOps but how do we start implementing DevSecOps?

We can use the techniques (see towards your right hand side) discussed in this course to implement a full blown security pipeline.

- **Shift Security Left**
  Use CI/CD pipeline to embed security

- **Self Service**
  Gives developers and operations visibility into security activities

- **Security Champions**
  Encourage security champions to pick security tasks.

- **Everything as Code (EAC)**
  Compliance as Code and hardening via configuration management systems

- **Secure by Default**
  Use secure by default frameworks and services

- **Use maturity models**
  Use DevSecOps Maturity Models to improve further
1. Shift Security left

Use CI/CD pipeline to embed security early on
DevOps: Typical Activities

Requirements
- Functional req.
- Non Functional req.
- Design

Code Repository
- Code
- Branching
- Third party components
- Hooks

CI Server
- Compile
- Basic tests
- Lint(Analyze)
- Package
- Security

Integration Testing
- Integration
- Performance
- Security

Artefact Repository
- Test on staging
- Release
- Schedule

CD Orchestration
- Configuration
- Inventory
- Infrastructure

Monitor
- Metrics
- Monitoring
- Alerting
DevOps: Typical Security Activities

Requirements
- Threat Modelling
- ASVS
- Git secrets
- Dependency Scanning
- Dependency Scanning
- Code Analysis(SAST)
- Security Unit Tests
- Docker security Testing
- Git secrets scanning
- Component scanning

Code Repository
- ZAP testing - baseline
- Container Scanning
- Modsecurity CRS

CI Server
- Docker/Third Party
- SSL scanning
- Nikto/dirbuster
- WPScan/JoomScan
- ZAP + selenium + python
- Component scanning

Integration Testing
- Component scanning

Artefact Repository
- Docker Benchmark
- System Hardening
- Application Hardening

CD Orchestration
- Compliance as code
- SOC with ELK
- Verify Controls

Monitoring

PLAN
- Code
- Build
- Test
- Release
- Deploy
- Operate
2. Self Service
Gives developers and operations visibility into security activities
3. Security as Code (EaC)

Compliance as Code and hardening via configuration management systems
4. Secure by default

Use secure by default frameworks and services
DevSecOps Maturity Model
DevSecOps Maturity Model (DSOMM)

Source: https://www.slideshare.net/cschneider4711/hackpra-2015-security-devops-free-pentesters-time-to-focus-on-highhanging-fruits
DevSecOps Maturity Model (DSOMM)

Static Depth: How deep is static code analysis?

Dynamic Depth: How deep are dynamic scans executed?

Intensity: How intense are the majority of the executed attacks?

Consolidation: How complete is the process of handling findings?

Source: https://www.slideshare.net/cschneider4711/hackpra-2015-security-devops-free-pentesters-time-to-focus-on-highhanging-fruits
Security Tools in CI/CD

1. Anything which takes more than 10 minutes (me being optimistic), isn't fit for CI/CD
2. SAST/DAST without creating custom rules/tweaks is of not huge benefit down the line.
3. Create separate jobs for easy debugging later.
4. Roll out tools in phases.
5. Fail builds when critical/high severity issues are found (after you have given devs/ops enough time to learn and get used to the security tools)
6. Link wiki in the scan outputs if someone needs some answers.
7. Tools which provide APIs are huge wins but make sure you at least have a CLI
8. See if your tools does incremental/baseline scans.
9. Some Ability to control the scope and false positives locally is nice (see brakeman/zap/dependency checker).
10. When in doubt ask Developers/QA for the help.
11. Everything as Code (EaC). Auditable, measurable and secure
Let’s see **DevSecOps pipeline** in Action
OWASP

DevSecOps Studio

DevSecOps Studio is a virtual environment to learn and teach DevSecOps concepts. It’s easy to get started and is mostly automatic.

It takes lots of efforts to setup a DevSecOps environment for training/demos and more often, its error prone when done manually.

https://github.com/teacheraio/DevSecOps-Studio/
DevSecOps Studio Benefits

A Easy to setup
Takes only few mins to setup and start using with just one command

B Reproducible
The aim of this project is to setup reproducible DevSecOps Lab environment for learning and testing different tools.

C Free & Open Source Software
This project is a free and open software to help more people learn about DevSecOps
Our Setup for On-Premise

- Push Code to git repo
- Triggers Build
- Run tests
- Deploys to Production
Our Setup for On-Premise

- Push Code to git repo
- Triggers Build
- Run tests
- Deploys to Production
## Python security tools

<table>
<thead>
<tr>
<th>Security Test</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAST</td>
<td>Bandit</td>
</tr>
<tr>
<td>DAST</td>
<td>ZAP Baseline</td>
</tr>
<tr>
<td>Hardening</td>
<td>Ansible</td>
</tr>
<tr>
<td>Compliance</td>
<td>Inspec</td>
</tr>
<tr>
<td>Git Secrets</td>
<td>Trufflehog</td>
</tr>
</tbody>
</table>
Conclusion

In conclusion, we don't need large sums of money to implement DevSecOps. We can use free and open source tools to showcase the benefits and value DevSecOps provides to the organization(s).

Go on, embed security as part of CI/CD

- **Shift Security Left**
  Use CI/CD pipeline to embed security early on

- **Self Service**
  Give developers and operations visibility into security activities/tools

- **Security Champions**
  Encourage security champions to pick security tasks.

- **Everything as Code (EAC)**
  Use Configuration management (IaC) to implement Security as Code

- **Secure by Default**
  Use secure by default frameworks and services

- **Use maturity models**
  Use DevSecOps Maturity Models to improve further
Thank you!
You folks are awesome.

www.teachera.io  @secfigo  secfigo@gmail.com