

# Continuous Security in Modernen Webanwendungen

**Sicherheit eingebaut**

Martin Reinhardt (Holisticon AG)

 @mreinhardt



# Agenda

- Continuous Delivery
- IT-Sicherheit
  - Methodik
  - Agil & Security
- Continuous Security
  - Automatisierung
  - Tools
- Ausblick
- Links

# About me

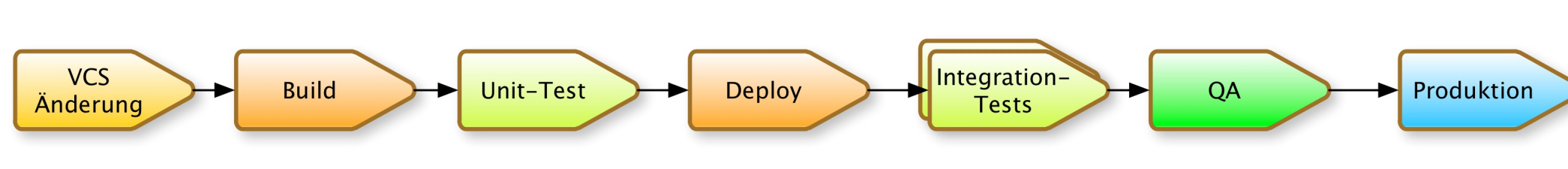
- Martin Reinhardt (Holisticon AG)



- [github.com/hyper2k](https://github.com/hyper2k)
- [twitter.com/mreinhardt](https://twitter.com/mreinhardt)

# Continuous Delivery

- Agile Softwareentwicklung arbeitet kleinteilig
  - Software oft und zuverlässig in Produktion
  - Software mit agilen Methoden kann nicht komplett (manuell) getestet werde
  - Alle 2 Wochen gesamten Funktionsumfang abtesten ist utopisch
  - Wesentlich ist dabei die Build Pipeline
- Wie?
  - Geschwindigkeit
  - Automatisierung



# Build Pipeline

- wesentliches Ziel ist schnelles Feedback, also Geschwindigkeit
  - Einzelne Schritt schnell abarbeiten (5 - 10 Minuten)
  - Möglichst früh Fehler finden (**Unit-Tests**)
- Quellcode muss Continuous Delivery gerecht werden
- Warum nicht auch für Security?
- Continuous Delivery auch eine Chance für Sicherheitsaspekte

# IT-Sicherheit



Pete Cheslock

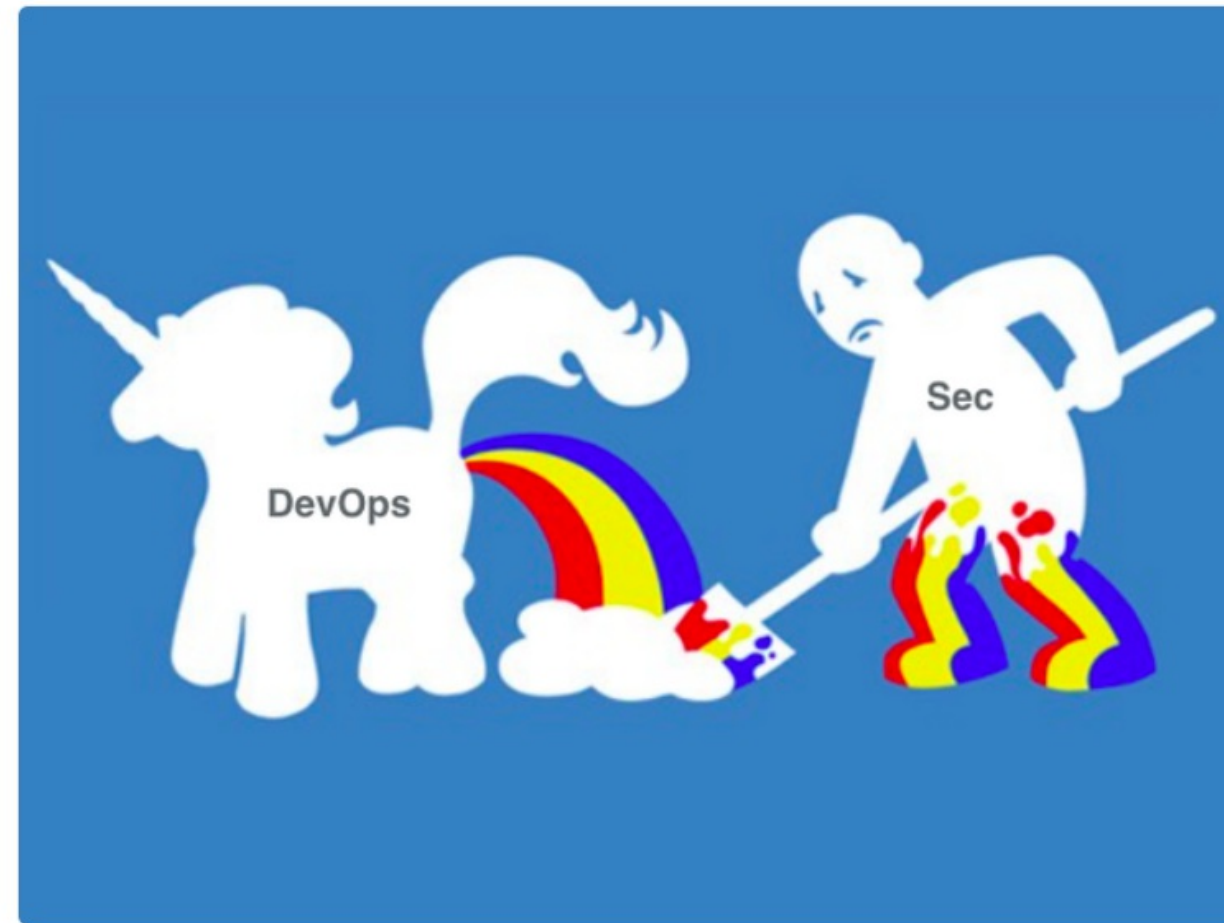
@petecheslock



Suivre

Everyone seemed to like this representation of DevOps and Security from my talk at [#devopsdays Austin](#)

[Voir la traduction](#)



RETWEETS  
2 378

J'AIME  
1 948



08:53 - 5 mai 2015



Retweets 2,4 k

Likes 1,9 k



# Warum das Ganze?

- NSA
- BDSG
- DSGVO
- Kosten
- Exploits
  - CVE-2016-5000 - Apache POI Information Disclosure via External Entity Expansion (XXE)
  - CVE-2016-4216 - Adobe XMP Toolkit for Java Information Disclosure via External Entity Expansion (XXE)
  - CVE-2016-3081 - Remote code execution vulnerability in Apache Struts when dynamic method invocation is enabled
  - CVE-2015-8103 - Remote code execution vulnerability in Jenkins remoting; related to the Apache commons-collections

# Black Duck - Open Source Security Analysis

- Stand von Open Source Security in kommerziellen Anwendungen [bit.ly/2yfsD2x](https://bit.ly/2yfsD2x)
  - 95% der Anwendungen enthalten OSS
  - 67% der Anwendungen enthalten OSS Schwachstellen
  - Durchschnittsalter von bekannten Schwachstellen in OSS: 1894 Tage





# OWASP Top 10

- Kritischsten Risiken in Webanwendungen
- A9 - Nutzung von Komponenten mit bekannten Schwachstellen
- Schwer zu erkennen
- Bewusstsein auf Entwicklungsseite
- Sichtbarkeit
- Toolunterstützung nicht vorhanden
- Patching erfordert erneut Codeänderungen

# Arten von Tests

- Funktionale Sicherheitstests
- Schwachstellen-Scanning / Fuzzing
- Penetrationstests

# Continuous Security

- Testen ist nicht alles, bei Entwicklung auch nicht
- Warum also nicht auch bei Security?
- logischer Schritt für Automatisierung
- Security muss auf verschiedenen Ebenen betrachtet werden
  - Code & Architektur (Sonar)
  - Integrations-Tests (bdd-security, zap, owasp)

# Security ganzheitlich

## ■ Thema einarbeiten

- Erstellung von Security Guidelines
- Berücksichtigung von Sicherheit im Rahmen der Erstellung von User Stories
- Codescans durchführen
- Peer-Reviews durch einen Security Champion durchführen
- Penetrationstests einplanen

## ■ Planing

## ■ Secure Scrum

## ■ Thread Model

## ■ Analog zur restlichen Softwareentwicklung

# Continuous Security Testing

- Tools mittlerweile verfügbar
- meist setzen diese auf OWASP auf
- Integration nicht schwieriger als bei DevOps

# Node Security Project (NSP)

- Prüfung der Abhängigkeiten auf bekannte Schwachstellen
- Schlägt korrigierte Version vor

Insecure Defaults Allow MITM Over TLS	
Name	engine.io-client
Installed	1.5.4
Vulnerable	<= 1.6.8
Patched	>= 1.6.9
More Info	<a href="https://nodesecurity.io/advisories/99">https://nodesecurity.io/advisories/99</a>

# OWASP dependency-check

- Seit 2012 verfügbar (basiert auf [A9 - Komponenten mit bekannten Schwachstellen](#))
- Verfügbar in verschiedenen Varianten: Ant, Maven, Gradle, SBT, Jenkins
- Analyse der Abhängigkeiten zu bekannten Schwachstellen für Java & .NET
- Experimentielle Unterstützung
  - CocoaPods
  - Swift Package Manager
  - Python
  - PHP (composer)
  - Node.js
  - Ruby

- Prinzipiell kann jede gefundene Schwachstelle zu Buildfehler führen

```
[ERROR] Failed to execute goal org.owasp:dependency-check-maven:1.4.0:check (default) ...  
[ERROR]  
[ERROR] Dependency-Check Failure:  
[ERROR] One or more dependencies were identified with vulnerabilities  
         that have a CVSS score greater than '5.0':  
[ERROR] commons-httpclient-3.1.jar: CVE-2014-3577  
[ERROR] mysql-connector-java-5.1.37.jar: CVE-2014-0001, CVE-2013-2378, ....  
[ERROR] tomcat-embed-core-8.0.33.jar: CVE-2016-3092, CVE-2013-2185, CVE-2002-0493
```

- Nicht praktikabel, deswegen sind Ausnahmen nötig
  - Kann reduzierter Common Vulnerability Scoring System-Score (CVSS) gewählt werden
  - Ausnahmen festlegen



# HTML - Report



Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

## Project: angular-spring-boot-webapp

Scan Information ([show all](#)):

- *dependency-check version*: 1.4.0
- *Report Generated On*: Oct 1, 2016 at 08:24:04 CEST
- *Dependencies Scanned*: 128
- *Vulnerable Dependencies*: 3
- *Vulnerabilities Found*: 106
- *Vulnerabilities Suppressed*: 0
- ...

Display: [Showing Vulnerable Dependencies \(click to show all\)](#)

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
<a href="#">commons-httpclient-3.1.jar</a>	cpe:/a:apache:commons-httpclient:3.1 cpe:/a:apache:httpclient:3.1	<a href="#">commons-httpclient:commons-httpclient:3.1</a>	Medium	3	LOW	17
<a href="#">mysql-connector-java-5.1.37.jar</a>	<a href="#">cpe:/a:mysql:mysql:5.1.37</a>	<a href="#">mysql:mysql-connector-java:5.1.37</a>	High	98	HIGHEST	22
<a href="#">tomcat-embed-core-8.0.33.jar</a>	<a href="#">cpe:/a:apache:tomcat:8.0.33</a>	<a href="#">org.apache.tomcat.embed:tomcat-embed-core:8.0.33</a>	High	5	HIGHEST	16

# ■ Mit festgelegten Ausnahmen



Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

## Project: angular-spring-boot-webapp

Scan Information ([show all](#)):

- *dependency-check version*: 1.4.0
- *Report Generated On*: Oct 1, 2016 at 08:47:00 CEST
- *Dependencies Scanned*: 102
- *Vulnerable Dependencies*: 0
- *Vulnerabilities Found*: 0
- *Vulnerabilities Suppressed*: 0
- ...

Display: [Showing Vulnerable Dependencies \(click to show all\)](#)

Dependency CPE GAV Highest Severity CVE Count CPE Confidence Evidence Count

## Dependencies

This report contains data retrieved from the [National Vulnerability Database](#).

- Ausnahmen werden in eigenem XML-Format festgelegt

```
<suppress>
  <notes>
    <![CDATA[This suppresses false positives identified on spring security. ]]>
  </notes>
  <gav regex="true">org\.springframework\.security:spring.*</gav>
  <cpe>cpe:/a:mod_security:mod_security</cpe>
  <cpe>cpe:/a:springsource:spring_framework</cpe>
  <cpe>cpe:/a:vmware:springsource_spring_framework</cpe>
</suppress>
```



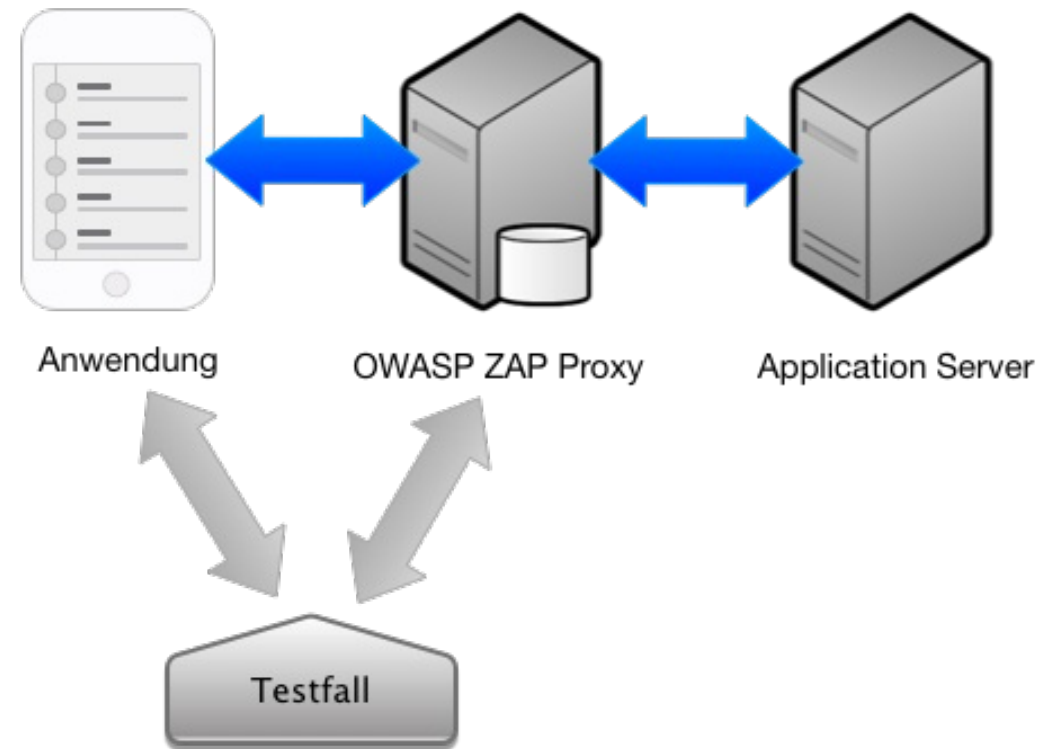
# OWASP Zed Attack Proxy (ZAP)

## Features

- Intercepting Proxy
- Automated Scanner
- Passive Scanner
- Brute Force Scanner
- Fuzzer
- Port Scanner
- Spider
- Web Sockets
- REST API Scanning (OpenAPI/Swagger)

# Funktionsweise

- Installation auf separaten Umgebung



- Scan der Anwendung
- Proxy während Testausführung



# Integration in Pipeline

- Maven Plugin: [github.com/hyperry2k/zap-maven-plugin](https://github.com/hyperry2k/zap-maven-plugin)

Jenkins > Public > ContinuousSecurity\_Demo > develop > #8

← Back to Project

**Status**

Changes

Console Output

View as plain text

View Build Information

Open Blue Ocean

Git Build Data

Test Result

Serenity Test Report

ZAP Report

Pipeline Steps





Embeddable Build Status

Previous Build

Next Build

## ! Build #8 (Sep 15, 2017 12:17:18 PM)

**Build Artifacts**

	<a href="#">ng-spring-boot.jar</a>	42,00 MB	 <a href="#">view</a>
	<a href="#">zapReport.xml</a>	4,55 KB	 <a href="#">view</a>
	<a href="#">zapSpiderResults.xml</a>	560 B	 <a href="#">view</a>

**Changes**

- chore(build): always clean up docker images ([detail](#) / [githubweb](#))

**Push event to branch develop at 12:17:12 on 15.09.2017**

**This run spent:**

- 3 ms waiting in the queue;
- 11 min building on an executor;
- 11 min total from scheduled to completion.

**Revision:** d63ac4c8d6b57b67321c7152d237e3d505a7a7d9

**git**

- develop

**Test Result** (no failures)

## ZAP Scanning Report

### Summary of Alerts

Risk Level	Number of Alerts
<a href="#">High</a>	0
<a href="#">Medium</a>	0
<a href="#">Low</a>	2
<a href="#">Informational</a>	0

### Alert Detail

Low (Medium)	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL	http://ngspring:41180/scripts/scripts.d13bbe30.js
Method	GET
Parameter	X-Content-Type-Options
URL	http://ngspring:41180/styles/vendor.0549f159.css
Method	GET
Parameter	X-Content-Type-Options
URL	http://ngspring:41180/styles/main.be748ac2.css



# Integration in Build

- Einfache Integration
- Erweiterung möglich (Selenium Tests)

```
<plugin>
  <groupId>de.martinreinhardt-online</groupId>
  <artifactId>zap-maven-plugin</artifactId>
  <configuration>
    <zapHost>localhost</zapHost>
    <zapPort>44444</zapPort>
    <failingRiskCodeThreshold>5</failingRiskCodeThreshold>
    <targetUrl>http://ngspring:41180/</targetUrl>
    <authenticationType>form</authenticationType>
    <username>user</username>
    <password>password</password>
    <shouldRunAjaxSpider>true</shouldRunAjaxSpider>
  </configuration>
  ...
</plugin>
```

# Sonar Integration

Total Alerts	High Risk Level	7
<u>236</u>	Medium Risk Level	<u>55</u>
	Low Risk Level	<u>174</u>
	Info Risk Level	<u>0</u>

ZAP Quality Gate Rename Copy Set as Default Delete

**CONDITIONS**

Only project measures are checked against thresholds. Sub-projects, directories and files are ignored. [More](#)

Add Condition:

ZAP Alerts	Value	is greater than	20	30	Update	Delete
ZAP High Alerts	Value	is greater than		0	Update	Delete
ZAP Medium Alerts	Value	is greater than	5	10	Update	Delete

[github.com/pdsoftplan/sonar-zap](https://github.com/pdsoftplan/sonar-zap)

# AWS absichern

- Security Monkey  
[github.com/Netflix/security\\_monkey](https://github.com/Netflix/security_monkey)
- Monitoring für Sicherheitsprobleme
- Für große verteilte AWS-Anwendungen

# Fazit & Ausblick

- Bibliotheken = Sicherheitsrisiko
  - gerade im modernen Umfeld
  - zeitnahe Aktualisierung nötig
  - automatisierbar
- Absicherung möglich
  - Penetrationstests durch DevOps einfach automatisierbar
  - viele Tools im Bereich Testing & Automatisierung
  - Spring Vault [projects.spring.io/spring-vault/](https://projects.spring.io/spring-vault/)
  - HashiCorp Sentinel [hashicorp.com/blog/sentinel-announcement-policy-as-code-framework/](https://hashicorp.com/blog/sentinel-announcement-policy-as-code-framework/)
- Feedback ist ein Muss
- unerlässlich ein Sicherheitsbewusstsein im Team aufzubauen

*"There is no one-size-fits-all solution to the complex problem of implementing a deployment pipeline."*

**Continuous Delivery, J. Humble, D. Farley**

*"There are only two types of companies: those that have been hacked, and those that will be"*

**Robert Miller, FBI Director, 2012**

# Links

- Beispiel Anwendung
- OWASP Top 10
- OWASP Cheat Sheet
- BSI Empfehlungen zu Webanwendungen
- DevOps - Testautomation I - Infrastructure as Code
- Rock CI mit Jenkins 2 und Docker
- Beispiel Serenity
- ZAP Blog
- Your code as a crime scene
- Secure Scrum
- Spring Vault
- HashiCorp Sentinel

# Präsentation



[holisticcon.github.io/presentations/2017\\_ISD\\_continuous-security/](https://holisticcon.github.io/presentations/2017_ISD_continuous-security/)