Martin Reinhardt

Continuous Security at the next level

Security

code.talks

Agenda

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

About me

- Martin Reinhardt (Holisticon AG)

- github.com/hypery2k
- twitter.com/mreinhardt

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

Continuous Delivery, J. Humble, D. Farley

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
  - Tests liefern schnelles Feedback über Seiteneffekte und Regressionen
- Wie?
  - Geschwindigkeit
  - Automatisierung

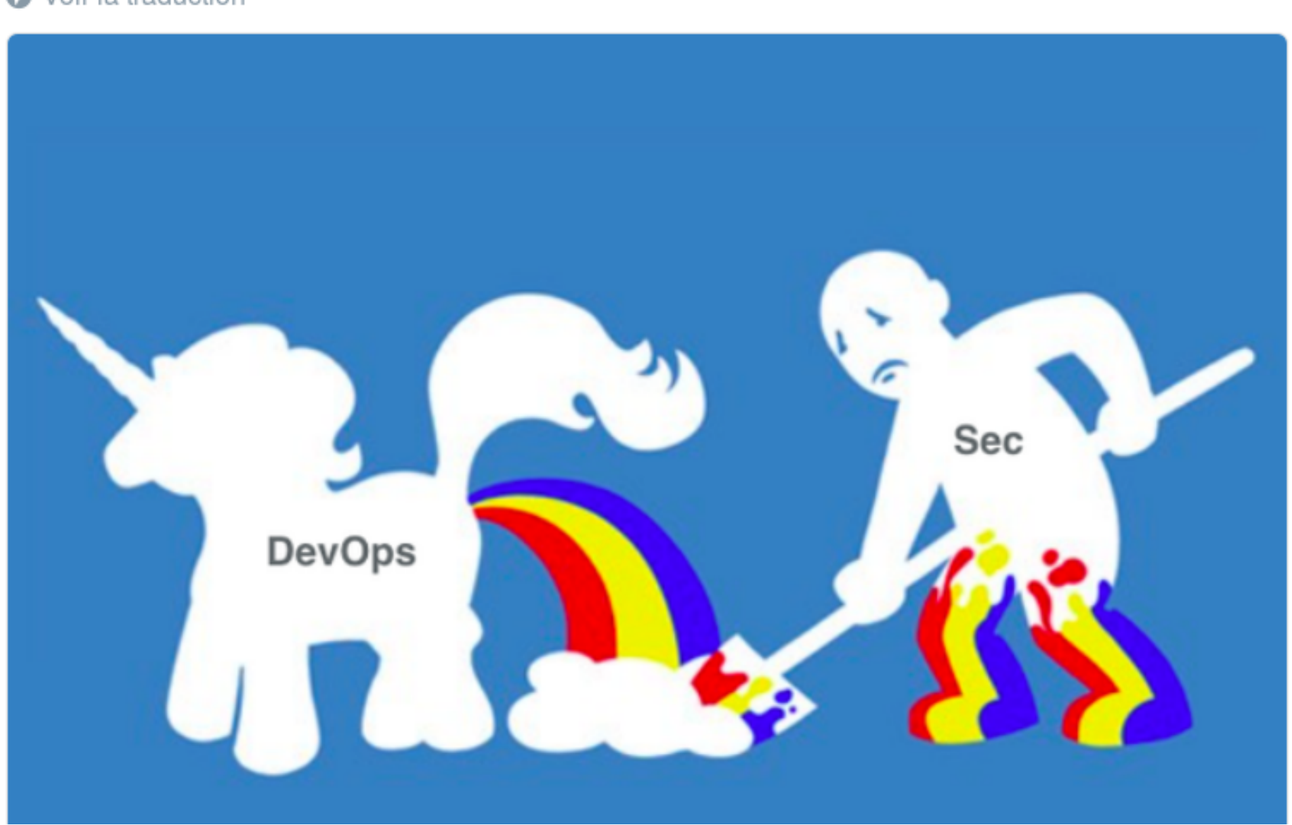
IT-Sicherheit

Pete Cheslock
@petecheslock

Suivre

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

Voir la traduction



Warum das Ganze?

- NSA, BND
- BDSG, DSGVO/GDPR
- 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
  - 95% der Anwendungen enthalten OSS
  - 67% der Anwendungen enhalten 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
- Patching erfordert erneut Codeänderungen

Arten von Tests

- Funktionale Sicherheitstests
- Schwachstellen-Scanning / Fuzzing
- Penetrationstests

Continuous Security

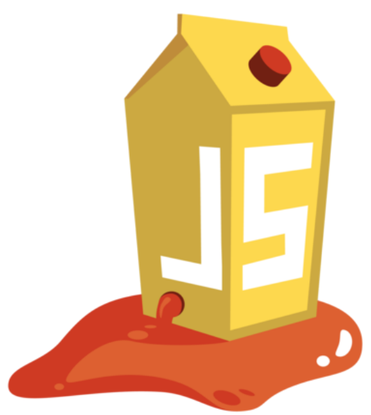
- Testen ist nicht alles, bei Entwicklung auch nicht
- Warum also nicht auch bei Security?
- logischer Schritt für Automatisierung
- Security muss auf verschieden Ebenen betrachtet werden
  - Code & Architektur (Sonar)
  - Integrations-Tests (bdd-security, zap, owasp)
  - Monitoring, Auditing & Logging

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

Kenne deinen Feind

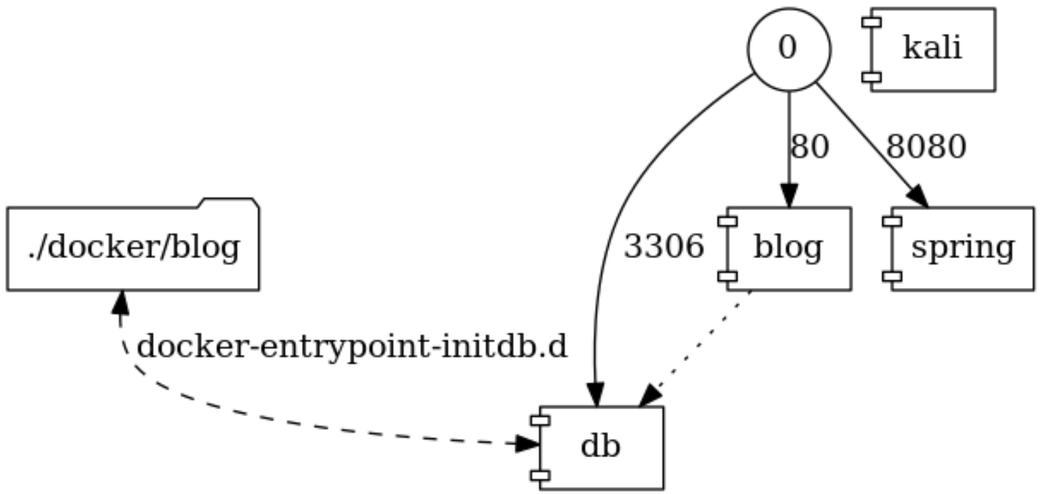
- Selber hacken
- Juice Shop der OWASP als Spielwiese
- Kombinierbar mit ctfd als Team-Challenge
- Per Heroku deployen und Teams gegeneinander spielen lassen
- Geringe Einstiegshürde



Hack yourself

- Welcome to the next level
- selber Ethical Hacker sein
- eigenen Stack per Docker attackieren
- verstehen wie Hacker vorgehen
- welche Tools genutzt werden
- Beispiel-Repo mit WordPress, SpringBoot und Kalilinux:
  github.com/holisticon/hack-yourself

Hack yourself



Continuous Security Testing

- Tools mittlerweile verfügbar
- meist setzen diese auf OWASP auf
- Integration nicht schwieriger als bei DevOps
- Mehr als Penetrationstests
- Neben BlackBox auch WhiteBox Testing nötig (Sonar, FindBugs)
- Keine Credentials im VCS ...
- Auf bekannte Schwachstellen prüfen

Node Security Project (NSP)

- Prüfung der Abhängigkeiten auf bekannte Schwachstellen
- Separates NPM-Modul, schlägt korrigierte Version vor

	Insecure Defaults Allow MITM Over TLS
Name	engine.io-client
CVSS	7.1 (High)
Installed	1.5.4
Vulnerable	<= 1.6.8
Patched	>= 1.6.9
path	devgui@0.1.4 > engine.io-client@1.6.9
More Info	https://nodesecurity.io/advisories/99

Time to say goodbye

- nsp gibt es nicht mehr
- `npm audit` to the rescue
- Setzt auf nsp auf, mit Integration in `npm install`
- Integriert in NPM 6+, schlägt korrigierte neben Version auch Kommandos vor

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 gefundenen 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 then '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 Ausnhamen nötig
  - Kann reduzierter Common Vulnerability Scoring System-Score (CVSS) gewählt werden
  - Ausnahmen festlegen

HTML - Report

DEPENDENCY-CHECK

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
commons-httpclient-3.1.jar	cpe:/a:apache:commons-httpclient:3.1 cpe:/a:apache:httpclient:3.1	commons-httpclient:commons-httpclient:3.1	Medium	3	LOW	17
mysql-connector-java-5.1.37.jar	cpe:/a:mysql:mysql:5.1.37	mysql:mysql-connector-java:5.1.37	High	98	HIGHEST	22
tomcat-embed-core-8.0.33.jar	cpe:/a:apache:tomcat:8.0.33	org.apache.tomcat.embed:tomcat-embed-core:8.0.33	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>
```

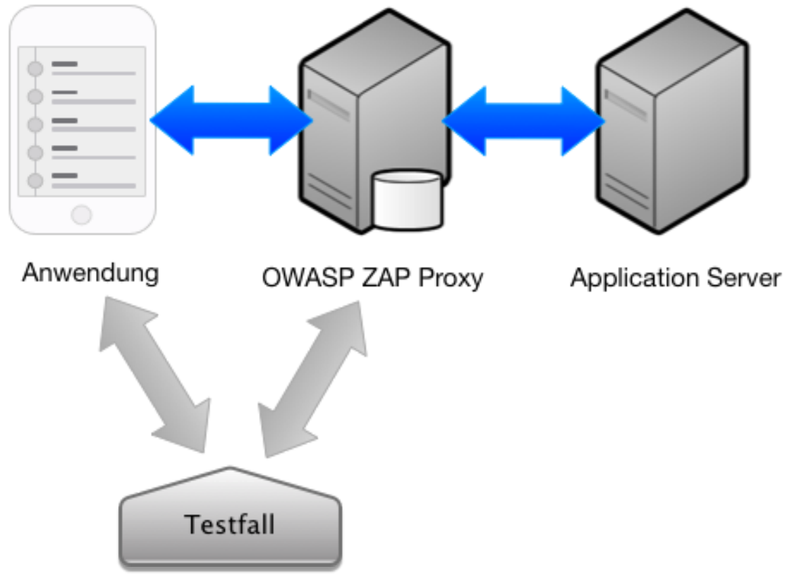
OWASP Zed Attack Proxy (ZAP)

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/ContinuousSecurityTooling/zap-maven-plugin

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	0
Low	2
Informational	0

Alert Detail

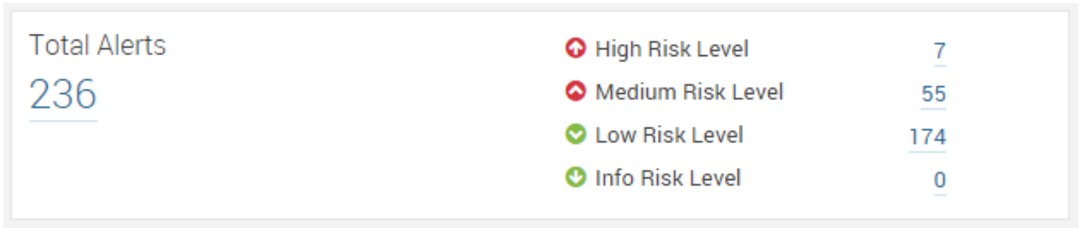
Low (Medium)	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted 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

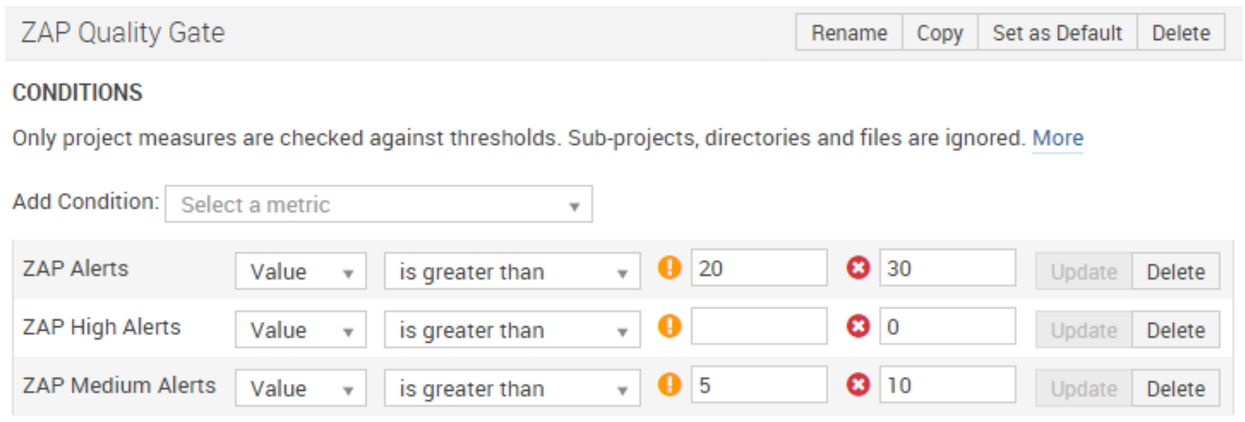
Integration in Build

- Einfache Integration
- Erweiterung möglich (Selenium Tests)

```
<plugin>
    <groupId>net.continuous-security-tools</groupId>
    <artifactId>zap-maven-plugin</artifactId>
    <version>0.2.0</version>
    <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





github.com/pdsoftplan/sonar-zap

holisticon

AWS absichern

- Security Monkey 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/
  - HashiCorp Sentinel hashicorp.com/blog/sentinel-announcement-policy-as-code-framework/
- Feedback ist ein Muss
- unerlässlich ein Sicherheitsbewusstsein im Team aufzubauen

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

Robert Miller, FBI Director, 2012

Links

- Hack Yourself
- Beispiel Anwendung
- OWASP Top 10
- OWASP Cheat Sheet
- BSI Empfehlungen zu Webanwendungen
- DevOps – Testautomation I – Infrastructure as Code
- Juice Shop
- ctfd als Docker Image
- Hack yourself
- ZAP Blog
- Your code as a crime scene
- Secure Scrum
- Spring Vault
- HashiCorp Sentinel

Präsentation

bit.ly/2ximOm3