



security testing **in continuous integration**

Agenda

whoami?

why?

what?

how?

whoami?



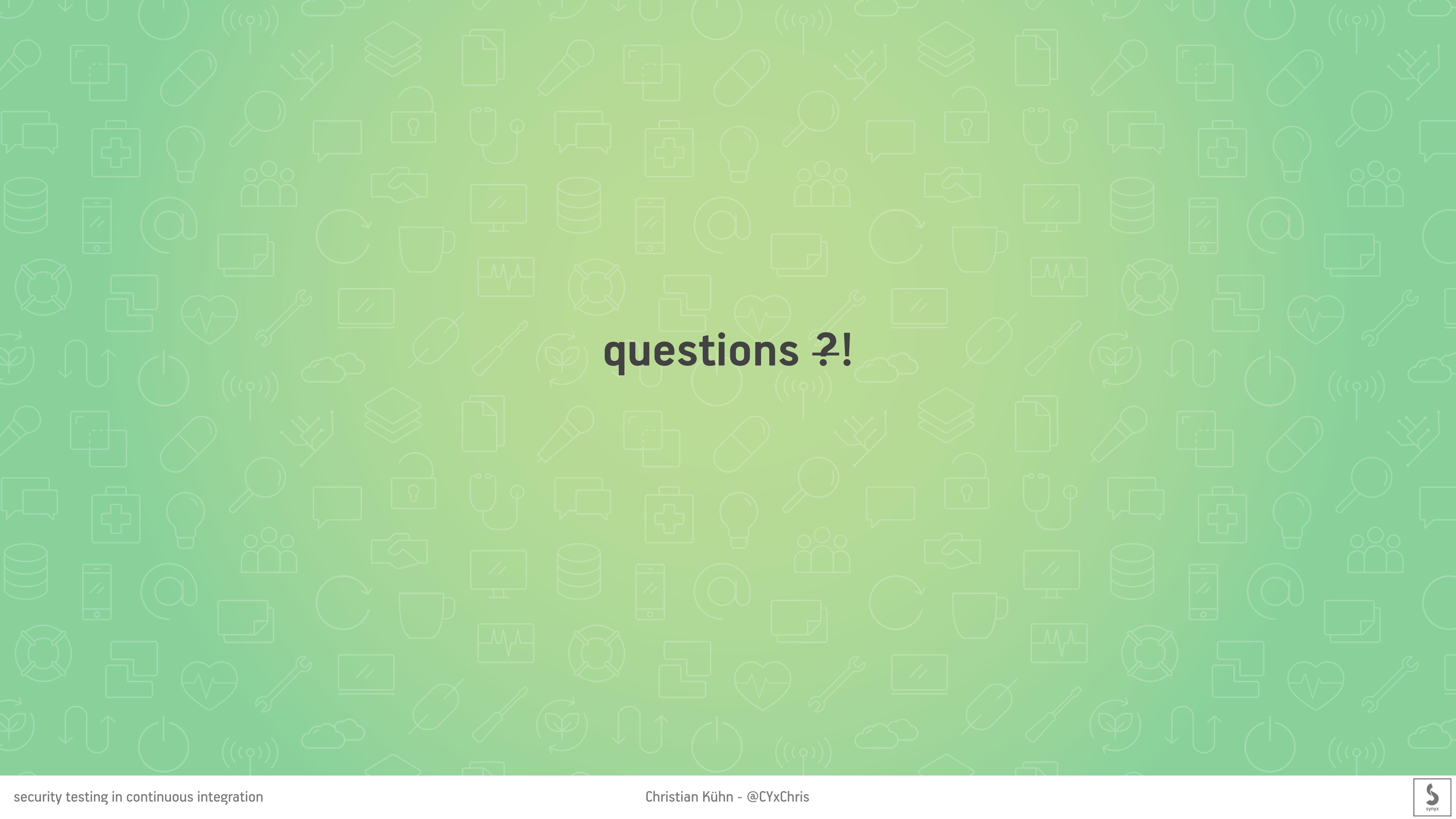
Christian Kühn
Senior System Developer

#java #kubernetes #devops

synyx GmbH Karlsruhe

Software nach Maß
Code Clinic
Open Source





questions ?!

software security issues: what could possibly go wrong?

- leakage of business data
- leakage of user/customer data
- service interruption
- industry malfunction
- death (😱)

examples:

equifax - “Credit Monitoring”
hacked 2017
vulnerability in Apache Struts dependency

143,000,000 SSN

209,000 credit card numbers

182,000 “consumers” with PII

<https://krebsonsecurity.com/2017/09/the-equifax-breach-what-you-should-know/>

examples:

Mossack Fonseca - "Law Firm and corporate service provider"

**hacked 2015
vulnerability in Drupal**

**11.5 million leaked documents about
money laundering
tax avoidance
corruption**

https://en.wikipedia.org/wiki/Panama_Papers



what stops developers from patching?

negligence

priorities / lack of time

skills / training

insight

“security - not my department” (or is it?)



security issues = technical debt

Solution?

DevSecOps

CAUTION
ENTERING
BUZZWORD
ZONE

DevOps

DevOps is a set of practices that automates the processes between software development and IT teams, in order that they can build, test, and release software faster and more reliably.

CAUTION
ENTERING
BUZZWORD
ZONE

continuous delivery

✓ testtage 1 >

Pipeline Änderungen Tests Artefakte ⌂ ⌒ ⌓ ⌔ Ausloggen X

Branch: master ↗ 1m 41s Keine Änderungen
Commit: 35bd92b ⌒ 7 minutes ago Branch indexing

Show runtime version mvn cleanup Unit Tests package artifact upload docker build docker push End

The screenshot shows a pipeline interface with a green header bar. The header includes a checkmark icon, the text "testtage 1 >", and several navigation links: Pipeline, Änderungen, Tests, Artefakte, and icons for refresh, edit, settings, and logout. Below the header, it displays "Branch: master" and "Commit: 35bd92b" along with their timestamps. A progress bar at the bottom shows the pipeline stages: "Show runtime version", "mvn cleanup", "Unit Tests", "package", "artifact upload", "docker build", and "docker push". The first six stages have green circular markers with checkmarks, while "docker push" has a blue circular marker with a checkmark, indicating it is the current step or the most recent one.

continuous delivery extended

✓ testtage < 3

Pipeline Änderungen Tests Artefakte ⌂ ⚙️ ↗ Ausloggen X

Branch: master ⓘ 22s Änderungen von chris
Commit: f096488 ⓘ a few seconds ago Replayed #2

```
graph LR; Start((Start)) --> ShowRuntime[Show runtime version]; ShowRuntime --> MvnCleanup[mvn cleanup]; MvnCleanup --> UnitTests[Unit Tests]; UnitTests --> DependencyCheck[dependency check]; DependencyCheck --> Package[package]; Package --> ArtifactUpload[artifact upload]; ArtifactUpload --> DockerBuild[docker build]; DockerBuild --> DockerPush[docker push]; DockerPush --> ContainerScan[container security scan]; ContainerScan --> ApiScan[api security scan]; ApiScan --> DockerCleanup[docker cleanup]; DockerCleanup --> End((End))
```

continuous delivery extended

testage < 2

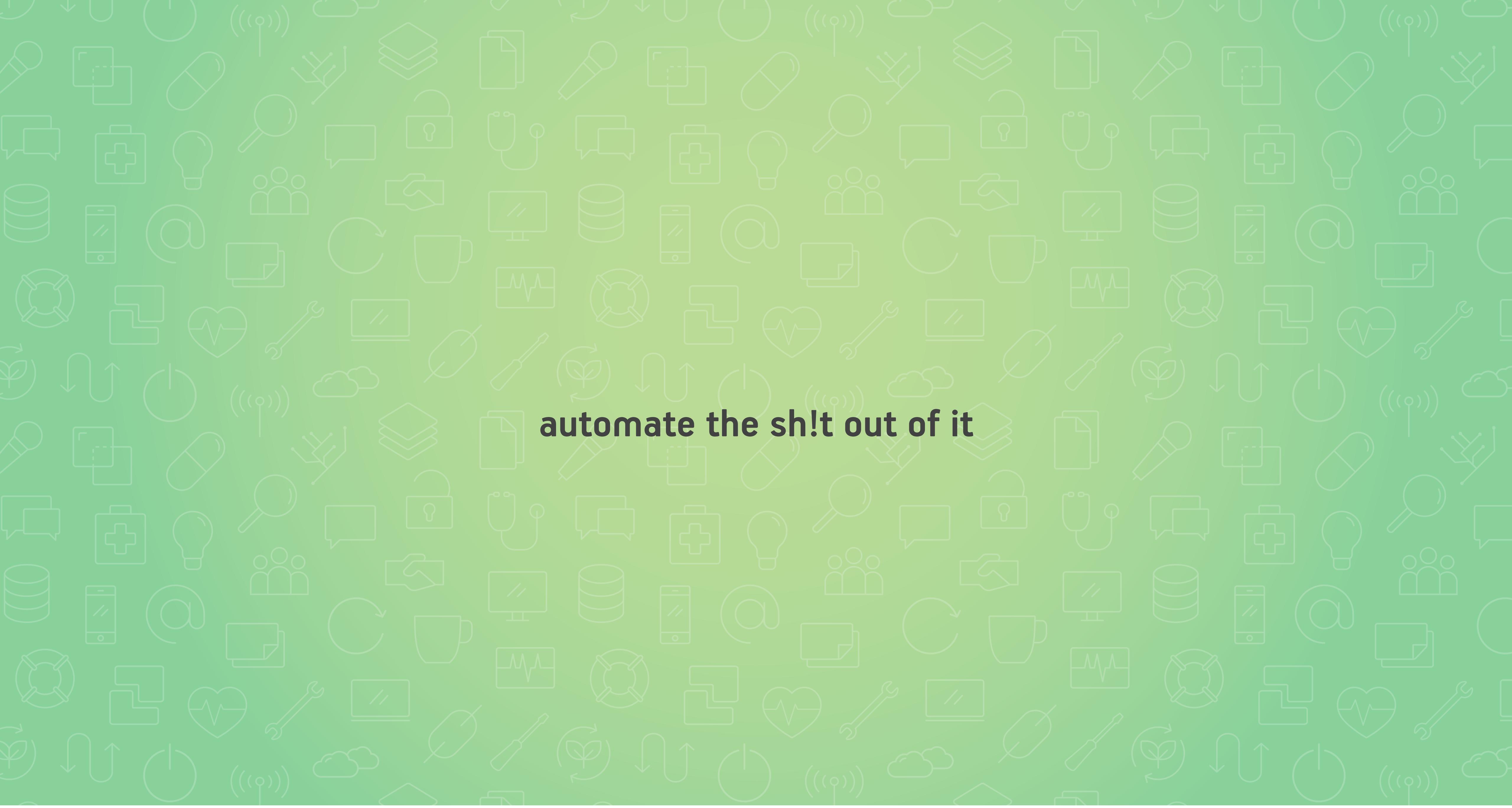
Branch: master ↗
Commit: f096488

⌚ 2m 40s
⌚ an hour ago

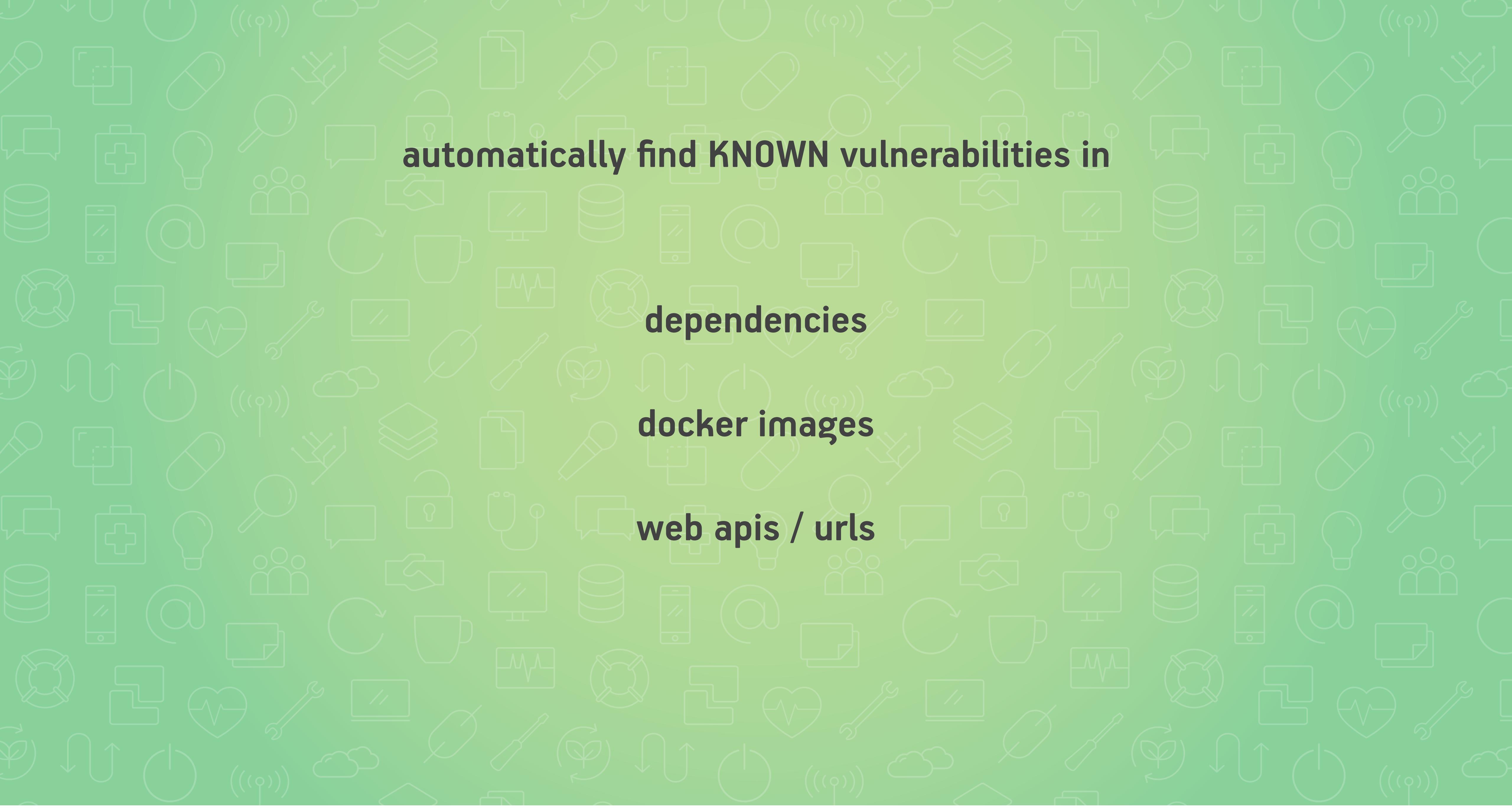
Änderungen von chris
Branch indexing

Pipeline Änderungen Tests Artefakte ⚡ 🖊️⚙️🔗 Ausloggen

The screenshot shows a continuous delivery pipeline interface. At the top, there's a header with the text "testage < 2", a branch dropdown set to "master", a commit hash "f096488", and timestamps "⌚ 2m 40s" and "⌚ an hour ago". Below the header, it says "Änderungen von chris" and "Branch indexing". The main area features a horizontal timeline with circular nodes representing different steps: "Start", "Show runtime version", "mvn cleanup", "Unit Tests", "dependency check", "package", "artifact upload", "docker build", "docker push", "container security scan", "api security scan", "docker cleanup", and "End". The "dependency check" node is highlighted with a blue border and exclamation mark icon, indicating a failure or warning in that step.



automate the sh!t out of it



automatically find KNOWN vulnerabilities in

dependencies

docker images

web apis / urls

CVE - Common Vulnerabilities and Exposures

vulnerability

/vʌln(ə)rə'biliti/
noun

1. the quality or state of being exposed to the possibility of being attacked or harmed, either physically or emotionally.

CVE - reference for publicly known information-security vulnerabilities and exposures

CVE-2017-5638 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands via a crafted Content-Type, Content-Disposition, or Content-Length HTTP header, as exploited in the wild in March 2017 with a Content-Type header containing a #cmd= string.

Source: MITRE

Description Last Modified: 09/22/2017

[+View Analysis Description](#)

Impact

CVSS v3.0 Severity and Metrics:

Base Score: 10.0 CRITICAL

Vector: AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H (V3 legend)

Impact Score: 6.0

Exploitability Score: 3.9

Attack Vector (AV): Network

Attack Complexity (AC): Low

Privileges Required (PR): None

User Interaction (UI): None

Scope (S): Changed

CVSS v2.0 Severity and Metrics:

Base Score: 10.0 HIGH

Vector: (AV:N/AC:L/Au:N/C:C/I:C/A:C) (V2 legend)

Impact Subscore: 10.0

Exploitability Subscore: 10.0

Access Vector (AV): Network

Access Complexity (AC): Low

Authentication (AU): None

Confidentiality (C): Complete

Integrity (I): Complete

Availability (A): Complete

QUICK INFO

CVE Dictionary Entry:

[CVE-2017-5638](#)

NVD Published Date:

03/10/2017

NVD Last Modified:

03/03/2018

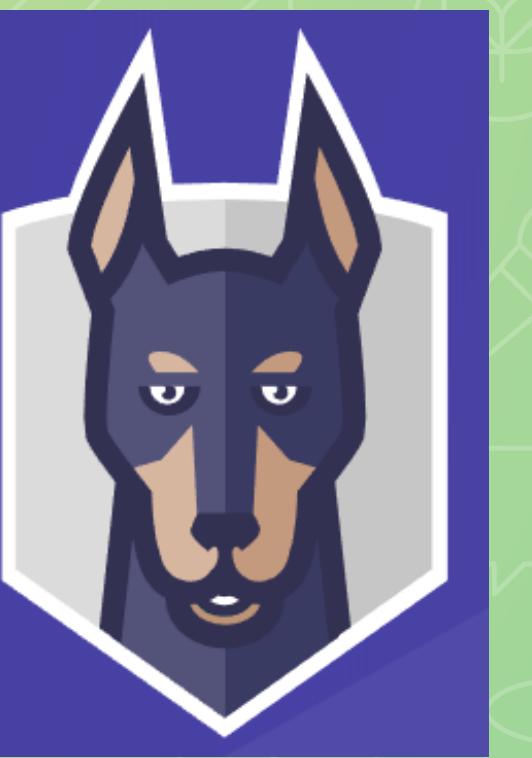
dependencies

**example: little maven/springboot demo-project:
github.com/cy4n/broken**

6 maven dependencies

71 transitive dependencies

find vulnerable dependencies



OWASP dependency-check



pulls CVE data from NIST database

check local project and write report

supports multiple Languages

can be run as docker-container, maven/gradle package...

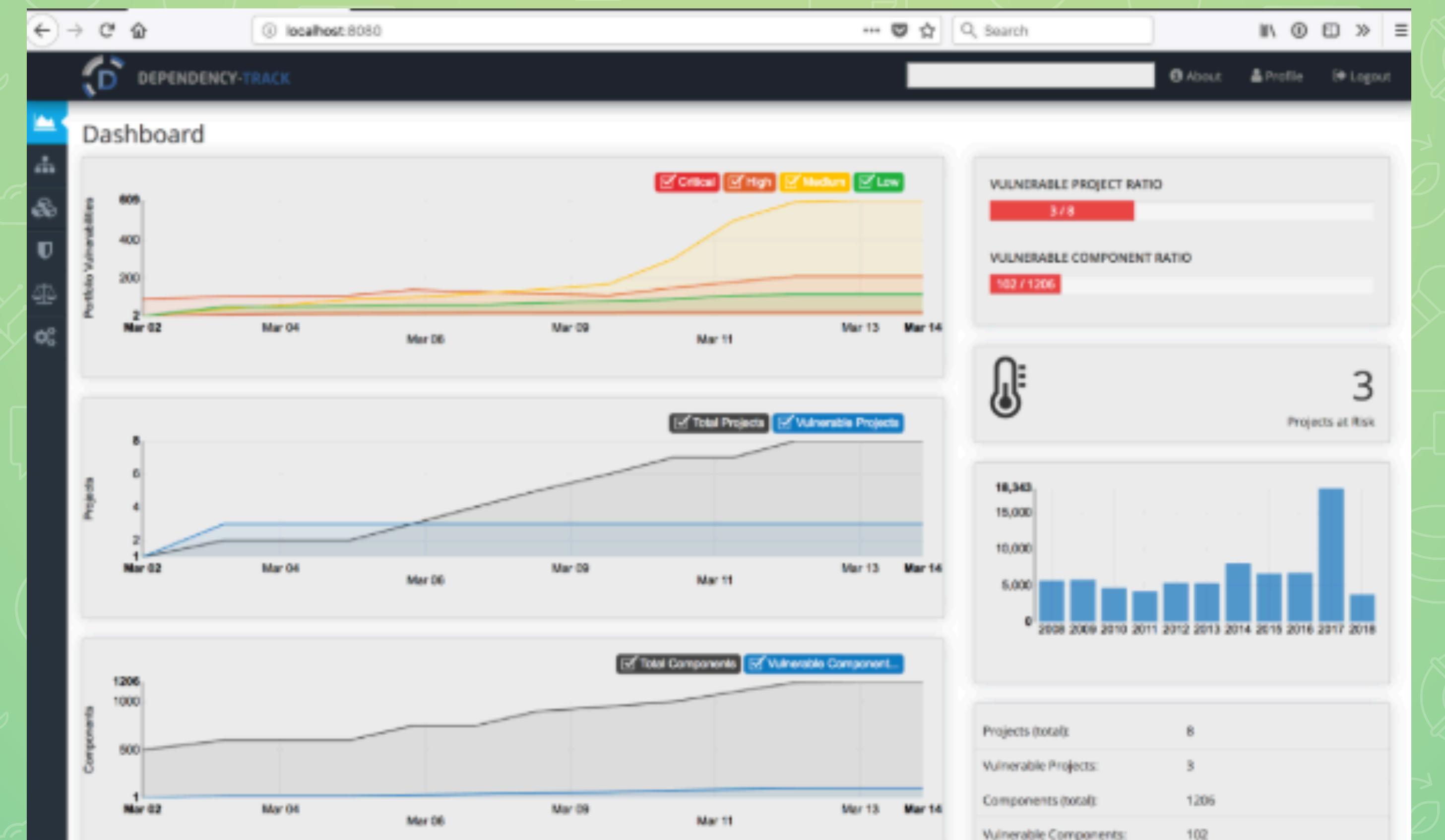
OWASP dependency-check



> Reliability ?	
▼ Security ?	
Overview	
On new code	
Vulnerabilities	0
Rating	A
Remediation Effort	0
Overall	
Vulnerabilities	14
Rating	D
Remediation Effort	3h 40min
> Maintainability ?	
> Coverage	
> Duplications	
> Size	
> Complexity ?	
> Issues	
▼ OWASP-Dependency-Check	
Critical Severity Vulnerabilities	0
High Severity Vulnerabilities	1
Inherited Risk Score	9
Low Severity Vulnerabilities	1
Medium Severity Vulnerabilities	1
Total Dependencies	178
Total Vulnerabilities	5
Vulnerable Component Ratio	1.7%



OWASP dependency-check



container

docker pull cy4n/broken

FROM cy4n/broken:latest

find vulnerable packages in container images

CoreOS Clair

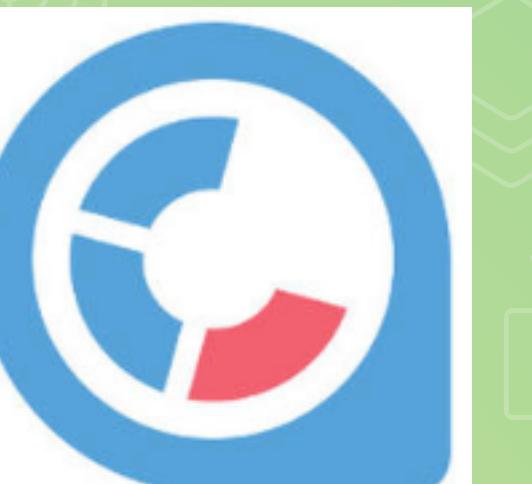


database

clair-server

clair-scanner (client)

CoreOS Clair



Quay Repositories Tutorial Docs Blog jaketd -

← example/repository 56c0fa71e47b

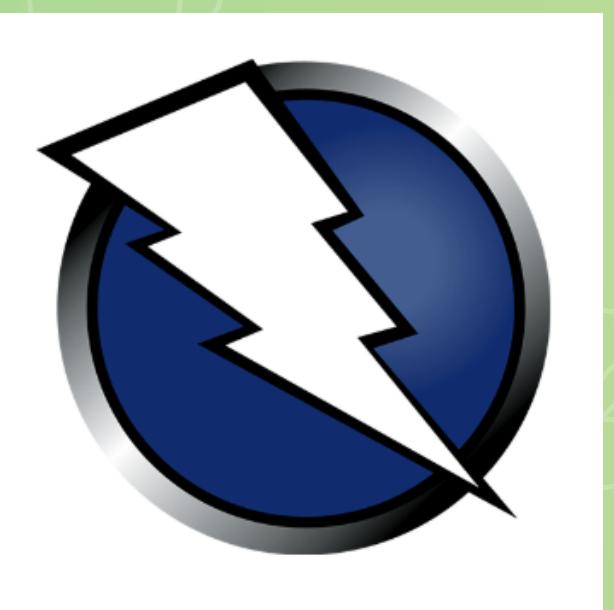
Quay Security Scanner has detected **13** vulnerabilities.
Patches are available for **4** vulnerabilities.

⚠ 1 High-level vulnerabilities.
⚠ 1 Medium-level vulnerabilities.
⚠ 2 Low-level vulnerabilities.
⚠ 5 Negligible-level vulnerabilities.
⚠ 4 Unknown-level vulnerabilities.

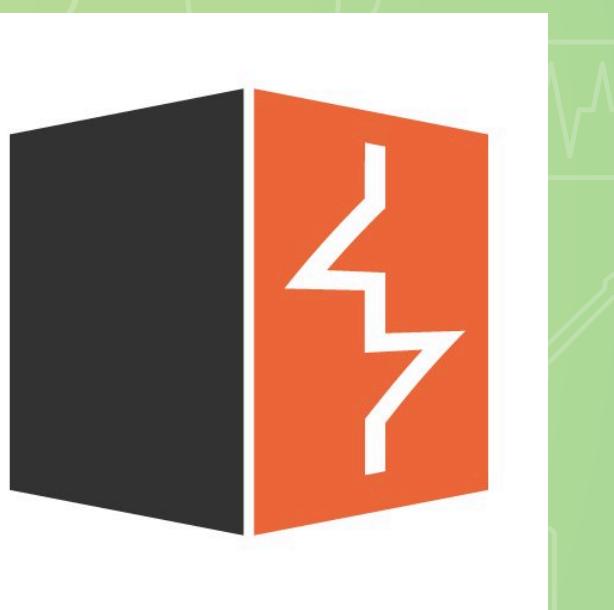
Image Vulnerabilities Filter Vulnerabilities... Only show fixable

CVE	Severity	Package	Current Version	Fixed In Version	Introduced In Image
CVE-2013-7445	⚠ High	linux	3.16.7-ckt20-1+deb8u3	(None)	<button>RUN</button> <code>apt-get up.</code>
CVE-2015-5276	⚠ Medium	gcc-4.9	4.9.2-10	(None)	<button>ADD</button> <code>file:b5391.</code>
CVE-2016-2856	⚠ Low	glibc	2.19-18+deb8u3	(None)	<button>ADD</button> <code>file:b5391.</code>
CVE-2016-0823	⚠ Low	linux	3.16.7-ckt20-1+deb8u3	(None)	<button>RUN</button> <code>apt-get up.</code>
CVE-2005-3660	⚠ Negligible *	linux	3.16.7-ckt20-1+deb8u3	(None)	<button>RUN</button> <code>apt-get up.</code>
CVE-2015-4003	⚠ Negligible *	linux	3.16.7-ckt20-1+deb8u3	(None)	<button>RUN</button> <code>apt-get up.</code>
CVE-2008-4108	⚠ Negligible *	python-defaults	2.7.9-1	(None)	<button>RUN</button> <code>apt-get up.</code>
CVE-2015-8830	⚠ Negligible	linux	3.16.7-ckt20-1+deb8u3	3.16.7-ckt20-1+deb8u4	<button>RUN</button> <code>apt-get up.</code>
CVE-2013-4392	⚠ Negligible *	systemd	215-17+deb8u3	(None)	<button>ADD</button> <code>file:b5391.</code>
CVE-2015-7515	⚠ Unknown	linux	3.16.7-ckt20-1+deb8u3	(None)	<button>RUN</button> <code>apt-get up.</code>
CVE-2015-8816	⚠ Unknown	linux	3.16.7-ckt20-1+deb8u3	3.16.7-ckt20-1+deb8u4	<button>RUN</button> <code>apt-get up.</code>
CVE-2016-2547	⚠ Unknown	linux	3.16.7-ckt20-1+deb8u3	3.16.7-ckt20-1+deb8u4	<button>RUN</button> <code>apt-get up.</code>
CVE-2016-2545	⚠ Unknown	linux	3.16.7-ckt20-1+deb8u3	3.16.7-ckt20-1+deb8u4	<button>RUN</button> <code>apt-get up.</code>

API / Webserver



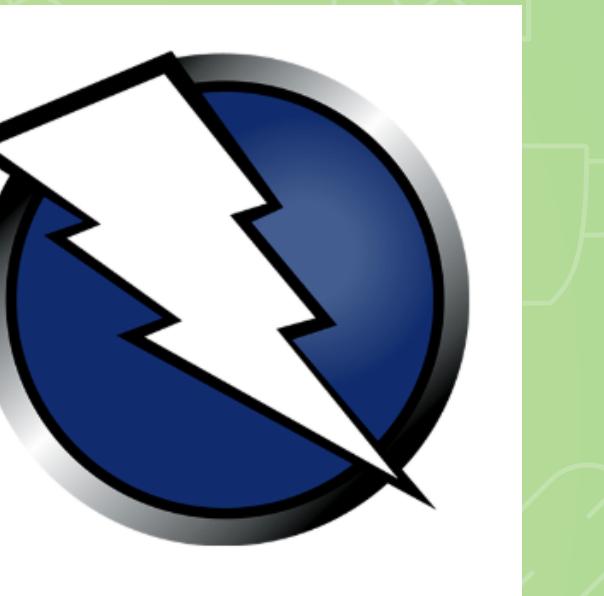
ZAPProxy



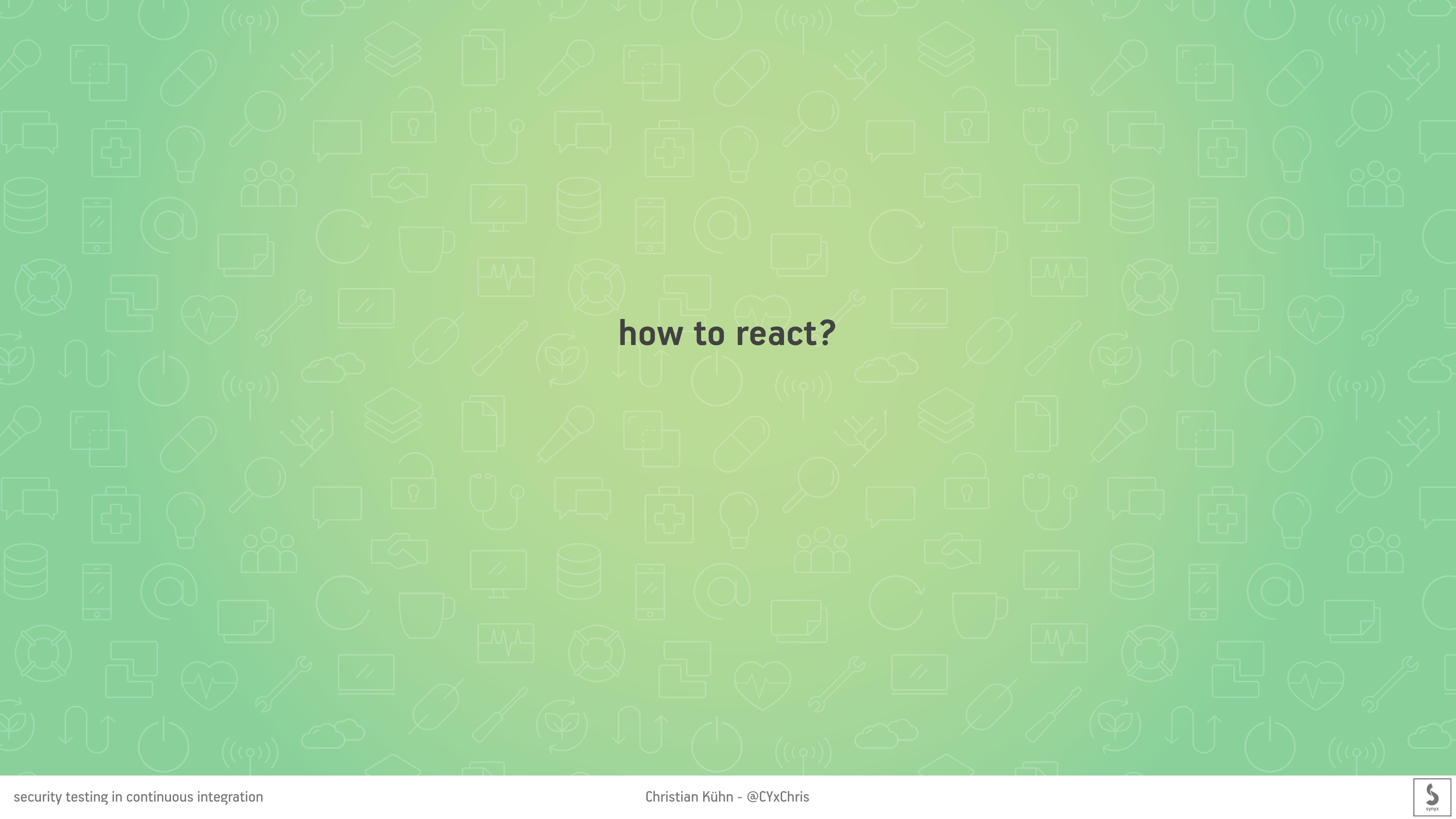
burp

API / Webserver

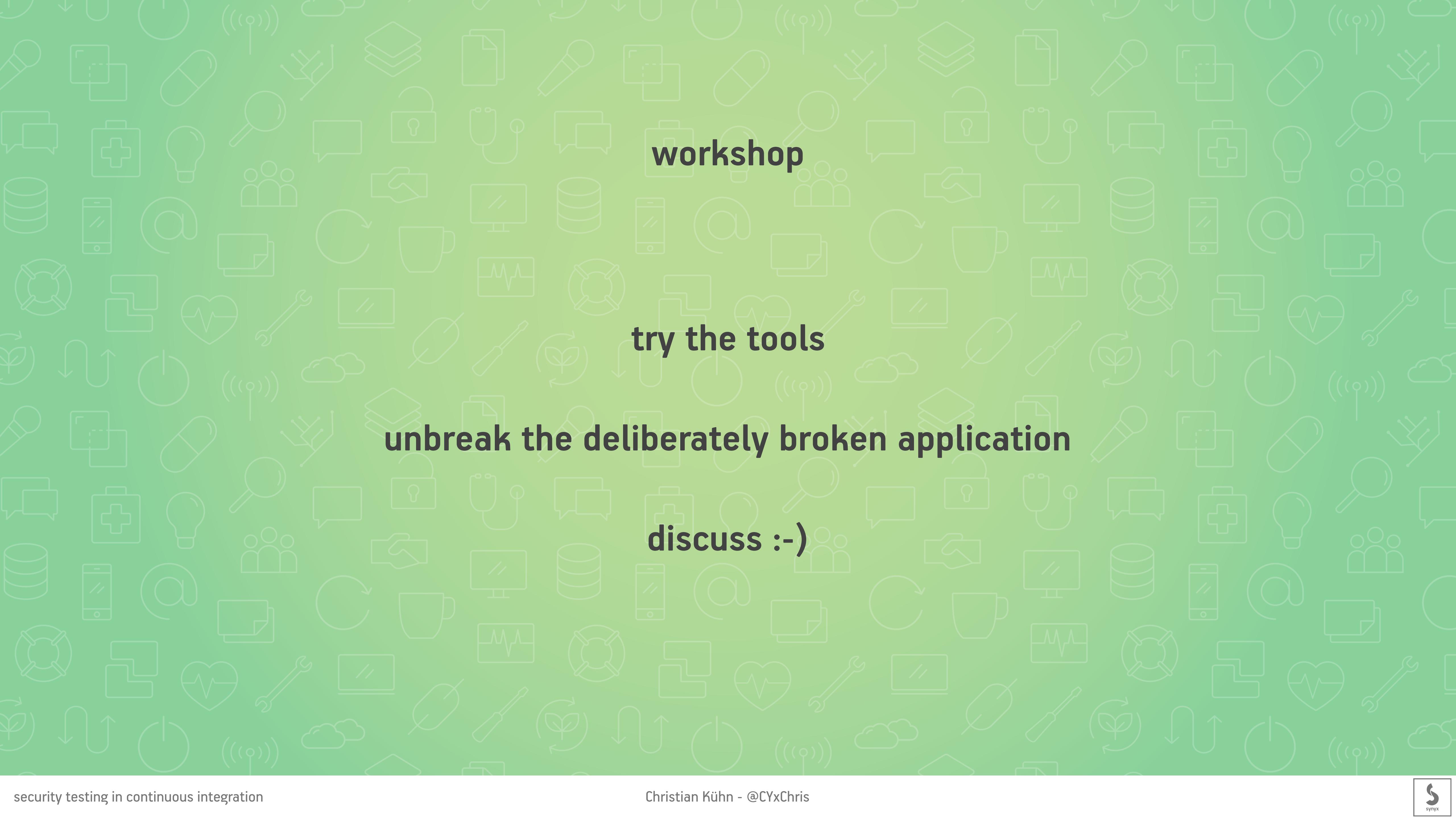
OWASP ZAPProxy



**url spider
passive (and active) modes
dynamic scanner
ajax supported**



how to react?



workshop

try the tools

unbreak the deliberately broken application

discuss :-)