Java ist doch schon sicher?!

Entwicklertag - 21.05.2014

Dominik Schadow - bridgingIT

java security

Ungefähr 56.800.000 Ergebnisse (0,31 Sekunden)

Java | heise **Security** - Heise Online
www.heise.de › Security › Erste Hilfe › Browsercheck
Wegen einer aktuellen Sicherheitslücke sollten Sie **Java** derzeit nicht einsetzen. Falls **Java** oben als "aktiv" angezeigt wird, deaktivieren Sie die Erweiterung im ...

Java security - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Java_security** Diese Seite übersetzen
The **Java** platform provides a number of features designed to improve the **security** of **Java** applications. This includes enforcing runtime constraints through the ...

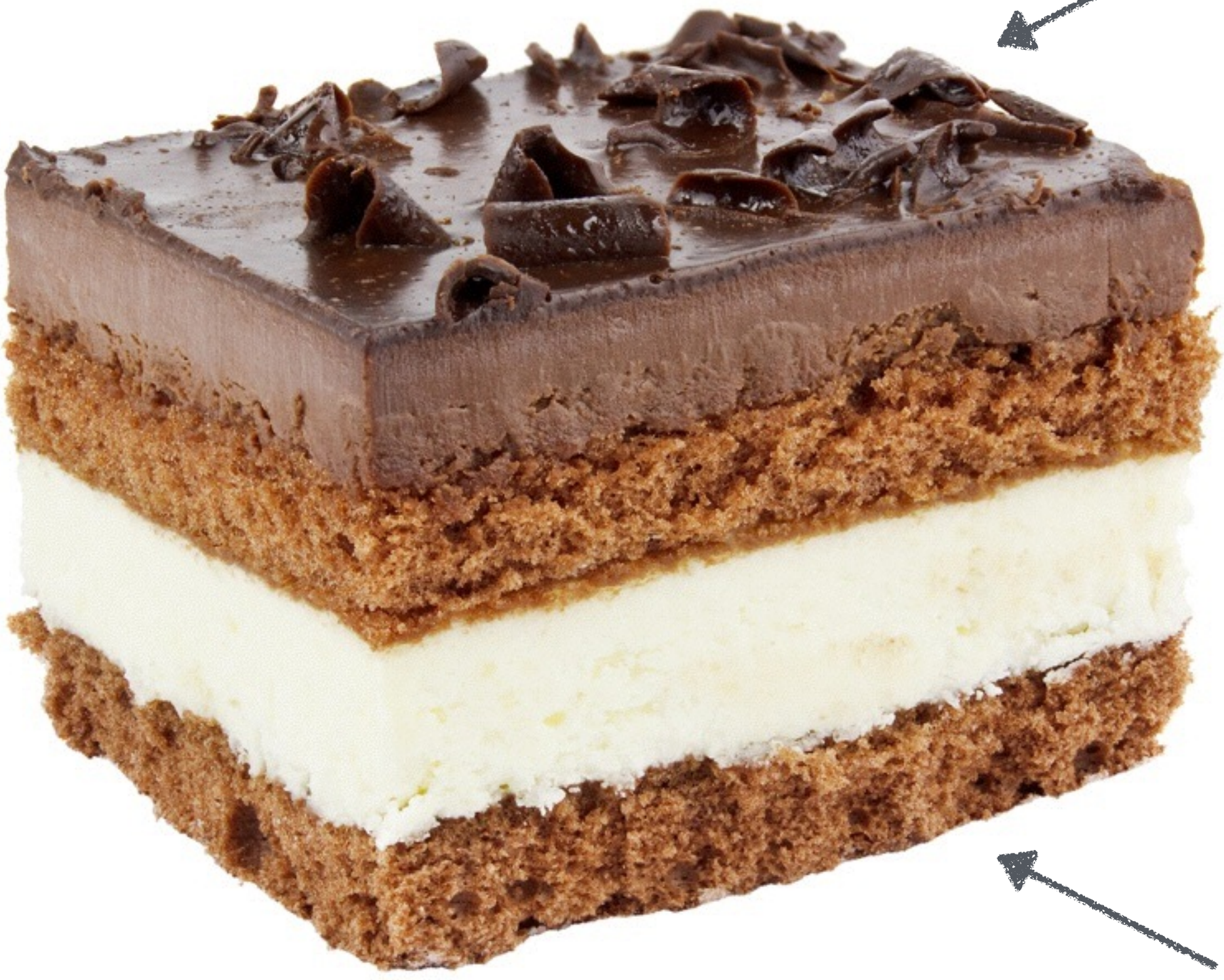
Was muss ich tun, wenn ich einen Sicherheitshinweis ... - **Java**
https://www.**java**.com/de/download/help/app**security**dialogs.xml
Das Update 21 für **Java** 7 hat Veränderungen im Verhalten des **Java**-Browser-Plug-ins eingeführt, durch die Sie besser informierte Entscheidungen treffen ...

java.com **Java Security** Resources
www.**java**.com/en/**security**/ Diese Seite übersetzen
This page is a reference on **security** for **Java**. It also provides additional resources to related sites.

What Developers Need to Know About **Java Security**
www.**java**.com/en/**security**/developer-info.jsp Diese Seite übersetzen
What Developers Need to Know About **Java Security**. Duke and security shield. Java is the most powerful tool you can use as a developer to deliver full featured ...

How to Control Concurrent Active User session in **Java** Web ...

https://plus.google.com/.../dkn7Nw9r1YN Diese Seite übersetzen
Javarevisited
04.04.2014 - How to Control Concurrent Active User session in **Java** Web Application using Spring **Security** Spring **security** offer many Out Of Box feature required in a Secure ...



Frontend

Services

Backend

Java Runtime

Java Runtime takes care of the security baseline

Cross-Site Request Forgery
Authorization
SQL Injection
3rd Party Libraries
Web Application Firewall
Authentication
Access Control
Redirects
Cross-Site Scripting
Transport Layer Security
Security Misconfiguration
Clickjacking
Presentation Layer Access Control
Session Fixation
Direct Object References
Cookie Theft
Cryptography
Privacy
XPath Injection
Session Management
Output Escaping
Input Validation
Sensitive Data Exposure
Vulnerabilities
Man in the Middle Attacks

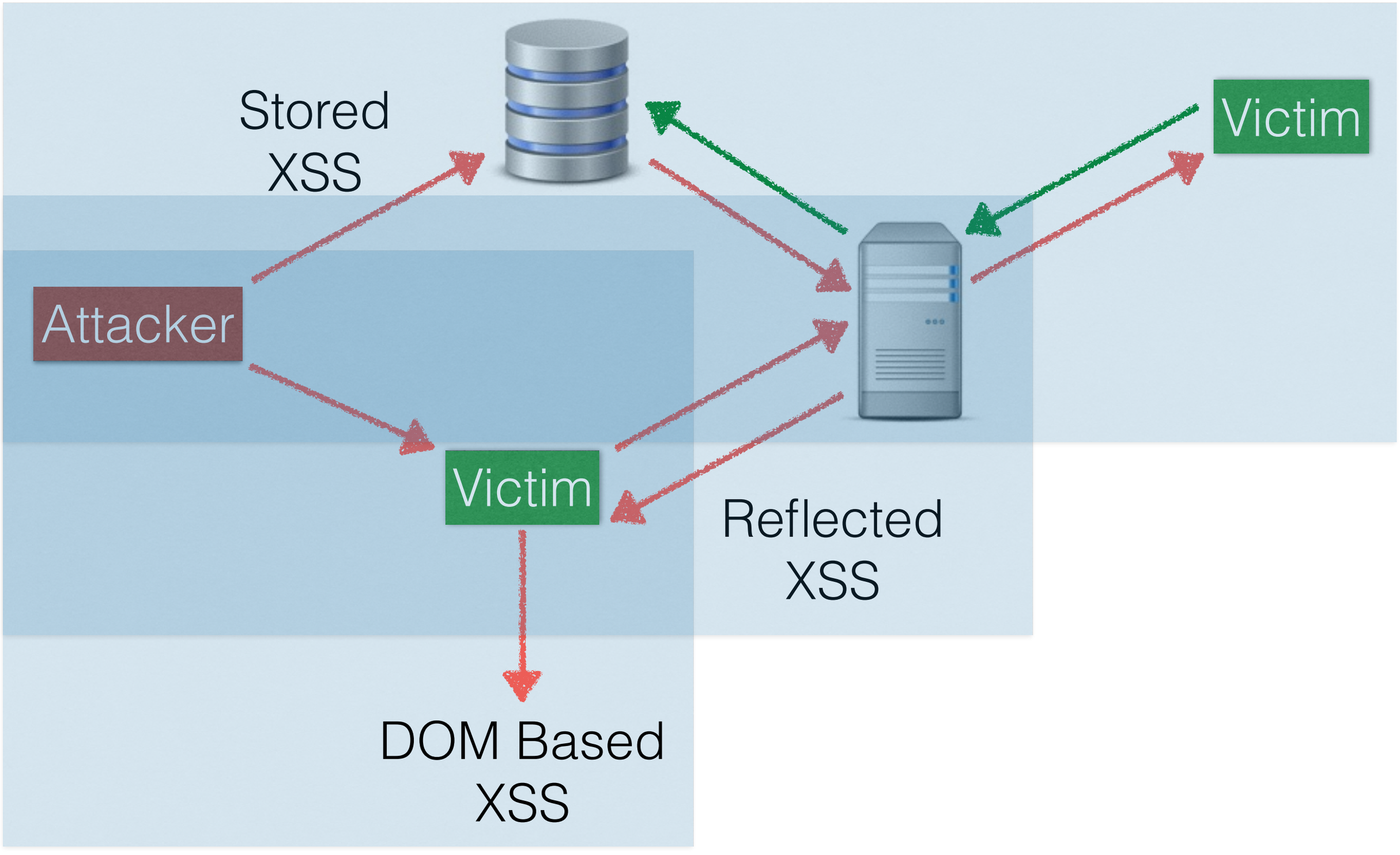

3rd party
library usage
Cross-Site
Request Forgery
Cross-Site
Scripting

Cross-Site Scripting (XSS)

Access victims' session credentials

- Site defacement
- Undermine CSRF defense
- Redirects (phishing)
- Load scripts
- Data theft

Attacker injected code executed in web application



Always validate all input and escape all output

Libraries for Cross-Site Scripting countermeasures

Output escaping

Coverity Security Library

github.com/coverity/coverity-security-library

OWASP Java Encoder

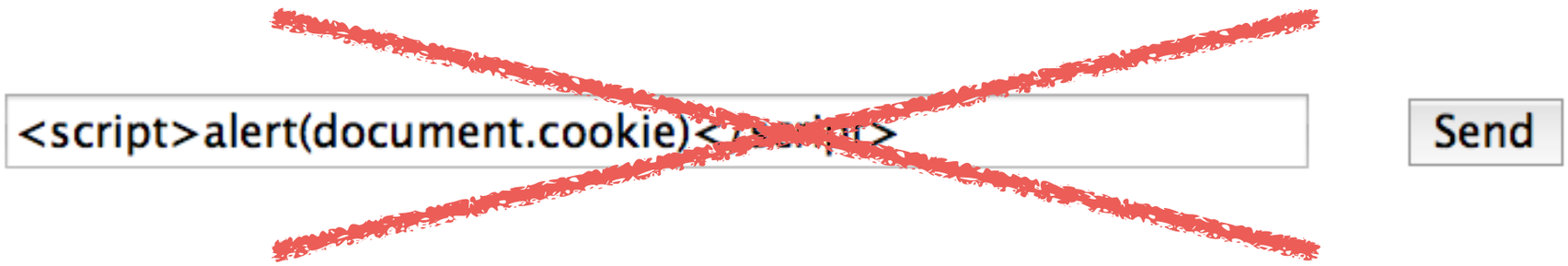
www.owasp.org/index.php/OWASP_Java_Encoder_Project

Allow selected HTML tags/ attributes

OWASP HTML Sanitizer

www.owasp.org/index.php/OWASP_Java_HTML_Sanitizer_Project

Content Security Policy is framework independent



Blocks ALL inline scripts

Whitelist valid resource URLs

```
response.setHeader("Content-Security-Policy",
"default-src 'self'; img-src *; object-src
applets.sample.de; script-src
scripts.sample.com; style-src *.sample.com");
```

Report only as test mode

```
response.setHeader("Content-Security-Policy-
Report-Only", "default-src 'self'; report-uri
CSPReporting");
```

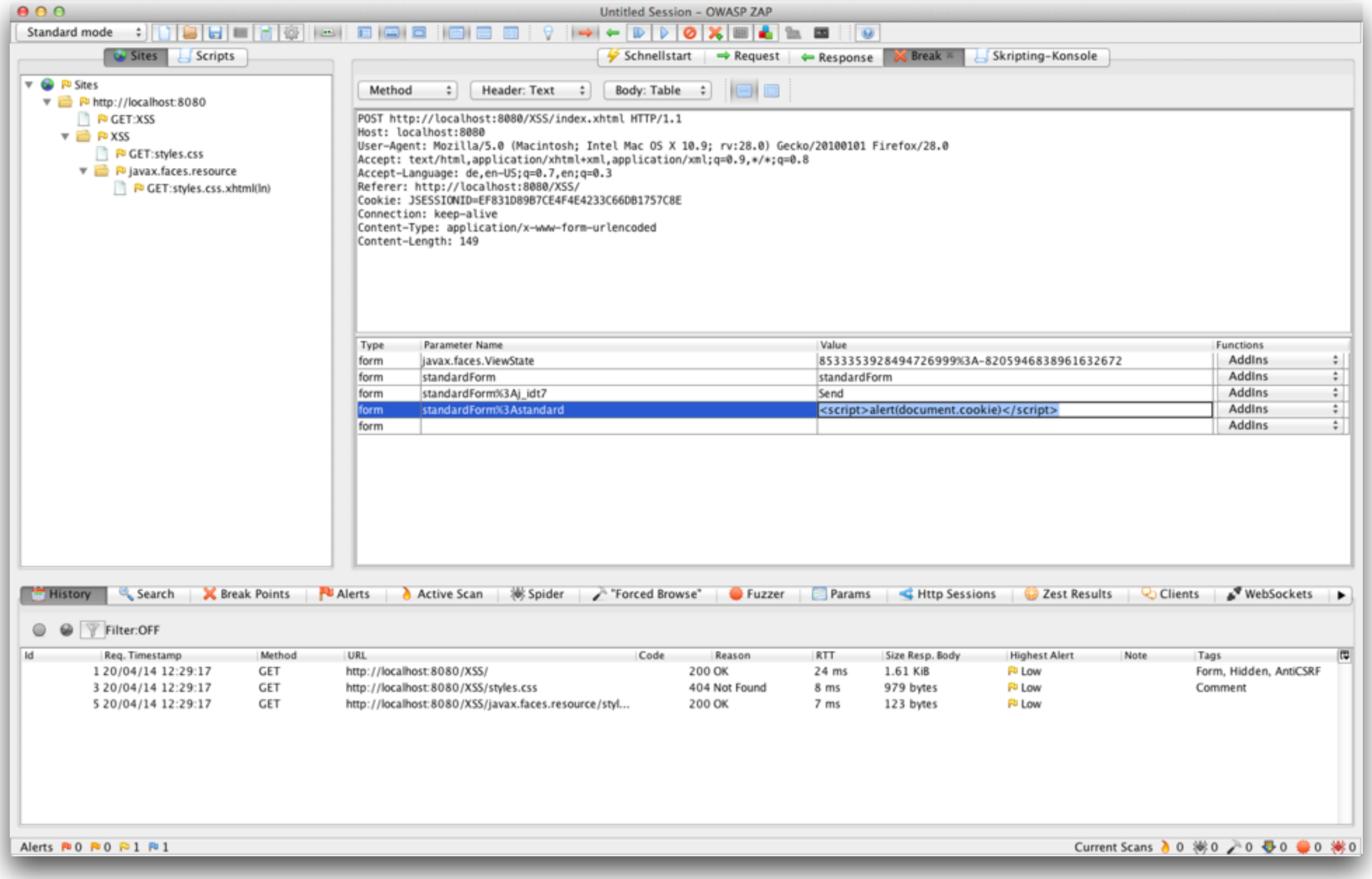
www.w3.org/TR/CSP

Session-Cookie protection via web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app ... version="3.1">
    <!-- ... -->
    <session-config>
        <session-timeout>30</session-timeout>
        <cookie-config>
            <http-only>true</http-only>
        </cookie-config>
        <tracking-mode>COOKIE</tracking-mode>
    </session-config>
</web-app>
```

Tomcat 7

Intercept requests/ responses with OWASP ZAP



www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

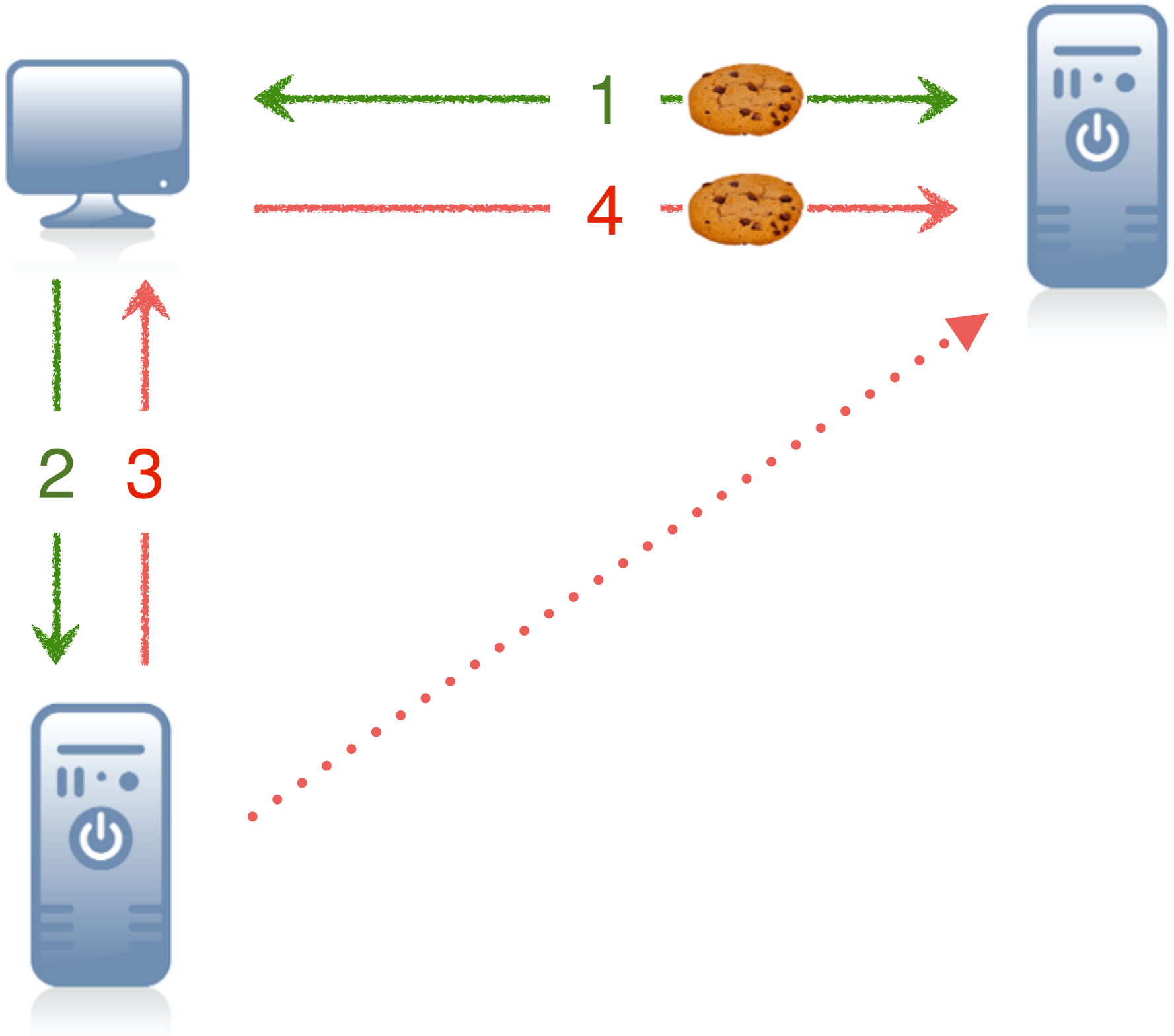
Demo

Content Security Policy as second layer of defense

Cross-Site Request Forgery (CSRF)

Using victims' credentials to gain access

CSRF utilizes fire and forget requests



Stop fake requests with random anti CSRF tokens

CSRF protection in libraries and frameworks

Built-in protection

JavaServer Faces (improved in 2.2)

jcp.org/en/jsr/detail?id=344

CSRF extension

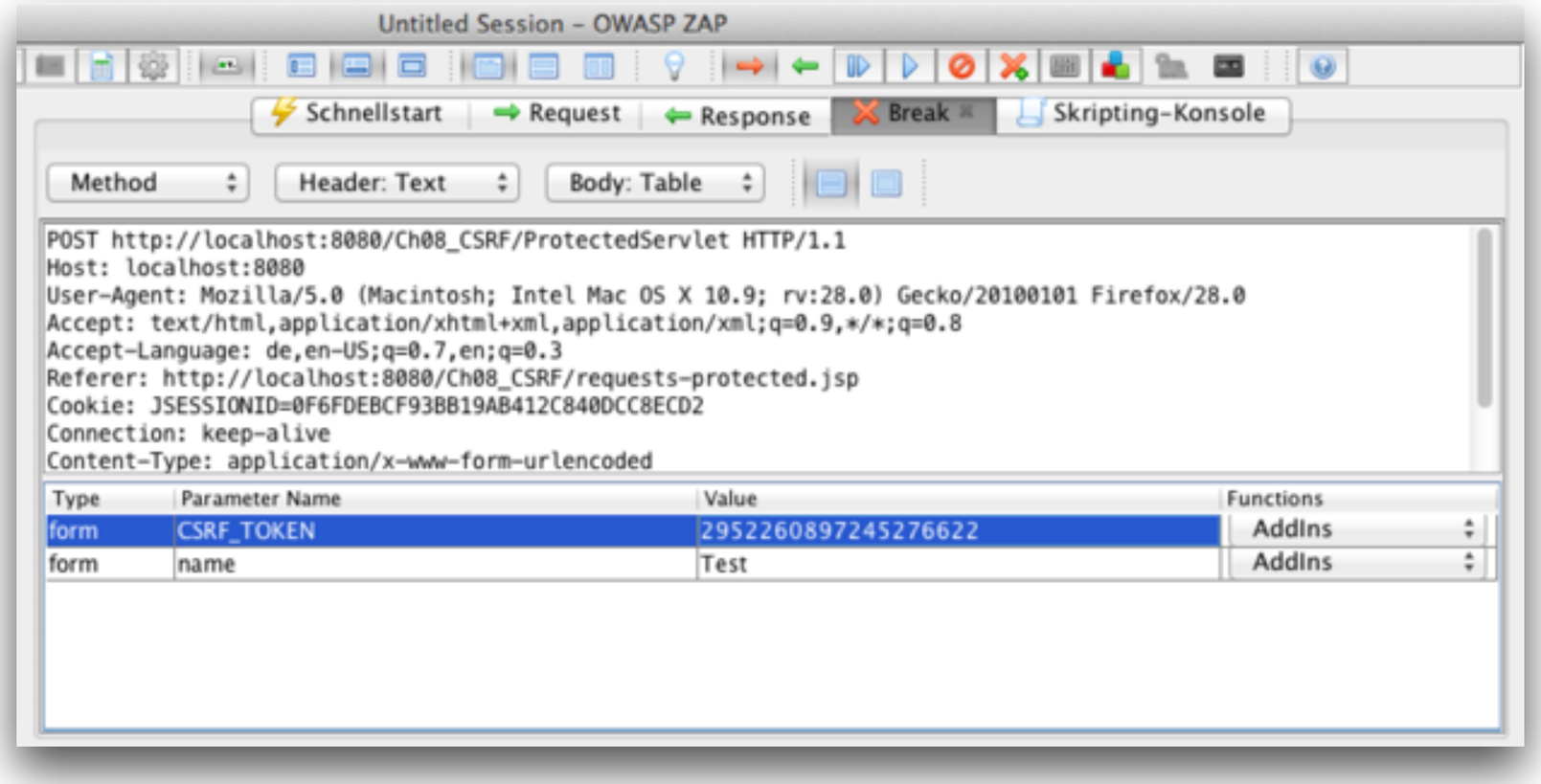
Spring Security 3.2

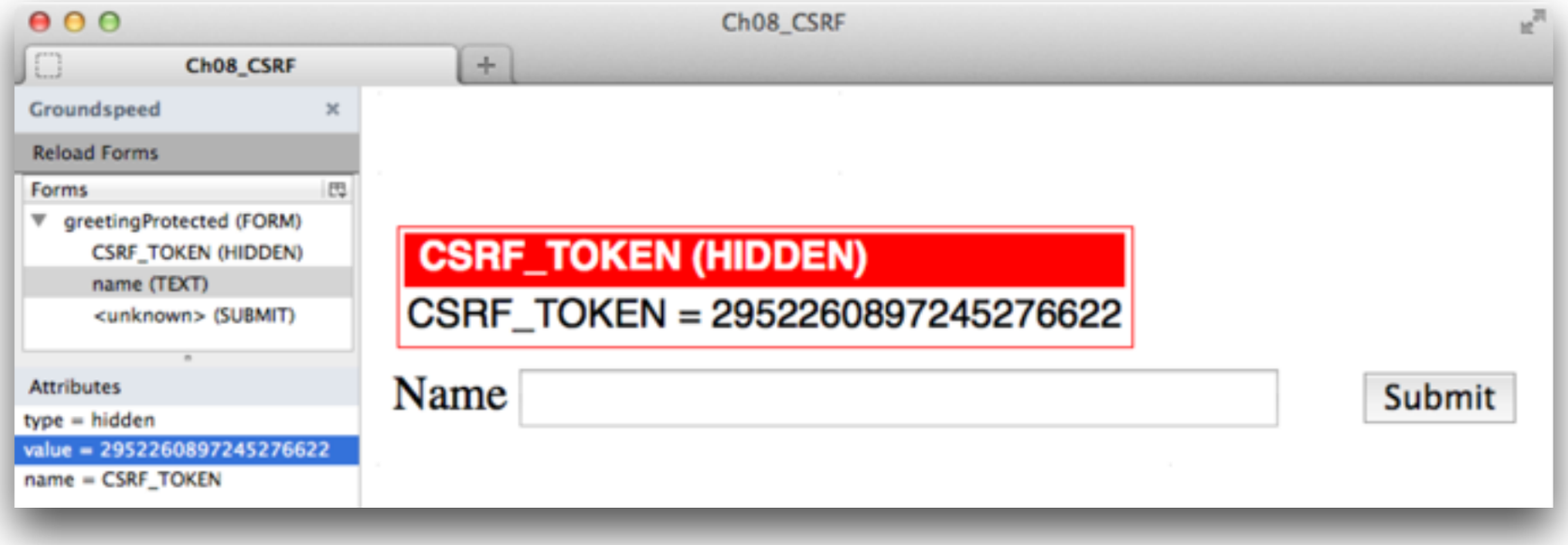
projects.spring.io/spring-security

Enterprise Security API

www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API

Test your CSRF protection with faked tokens



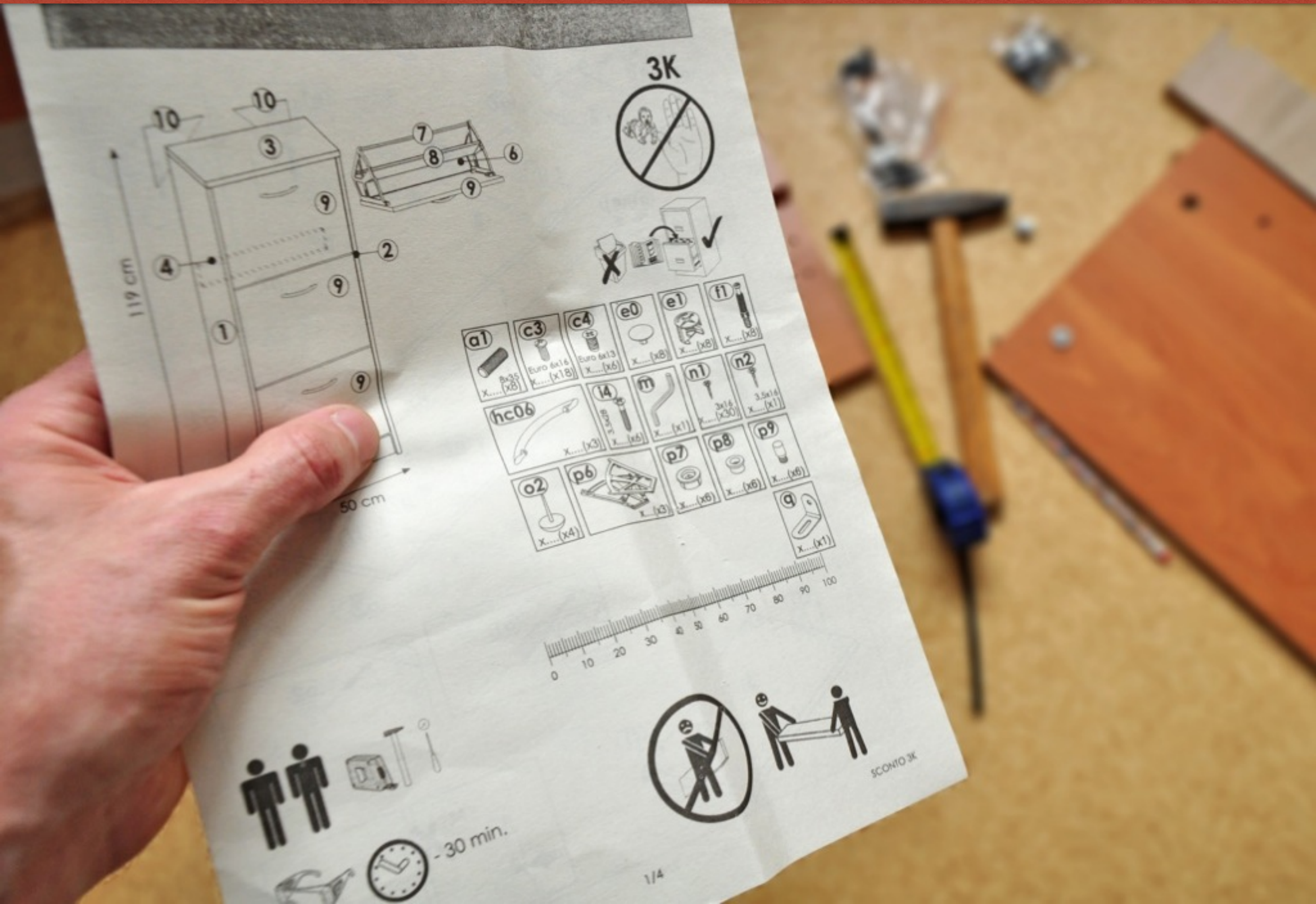


www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

addons.mozilla.org/en-US/firefox/addon/groundspeed

Demo

3rd party library usage

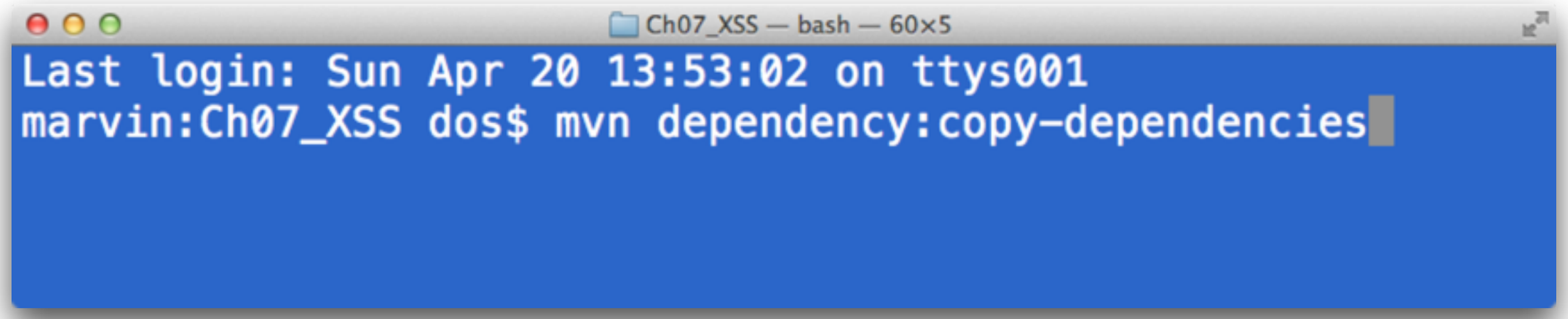


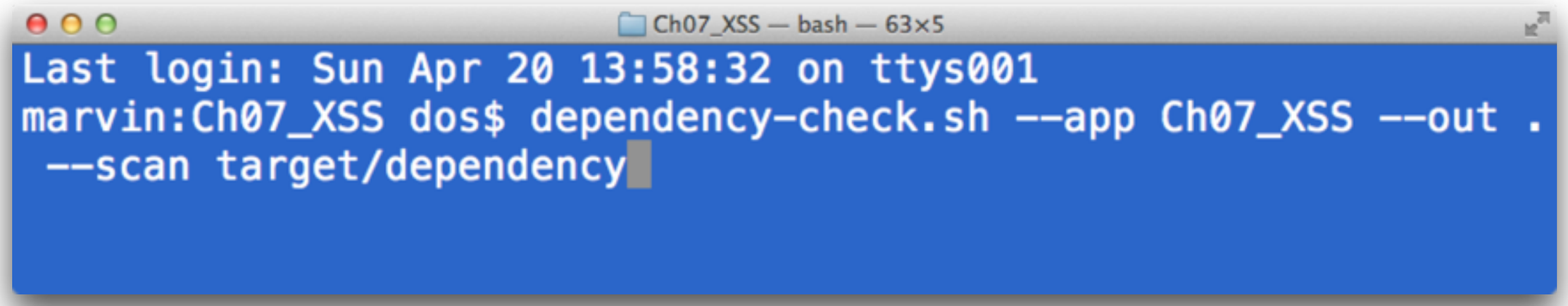
Code, libraries and configuration belong together

Test the functionality you rely on

Outdated libraries imply a false sense of security

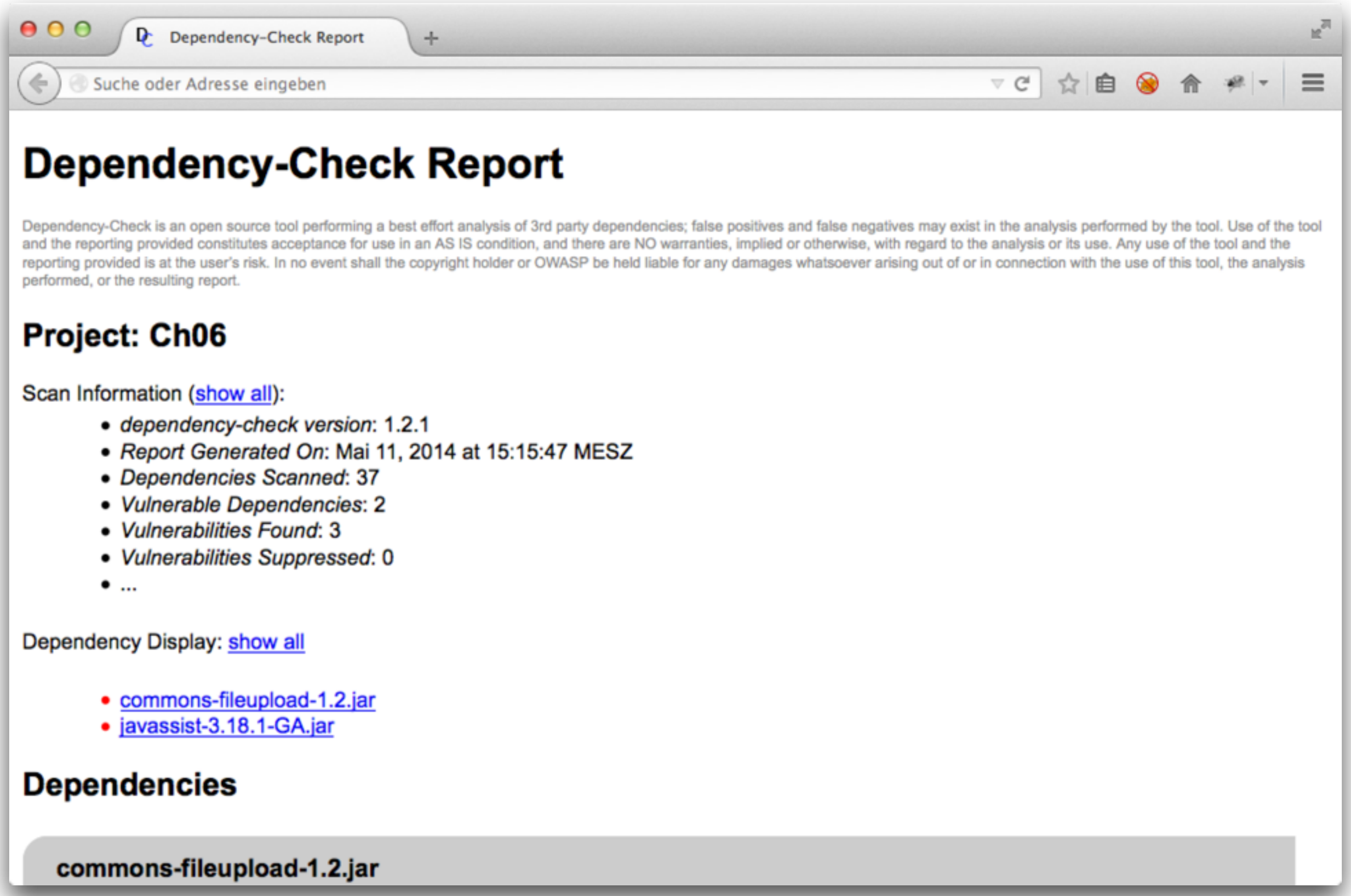
Find insecure libs with OWASP Dependency Check





www.owasp.org/index.php/OWASP_Dependency_Check

Not every vulnerability may affect your application



Widespread outdated libs impose a greater risk

- Java is as (in)secure as most other languages
- Java can't prevent every development bug
- (Web) application security is always the developers' job

Developers make the difference

Dominik Schadow

BridgingIT GmbH
Königstraße 42
70173 Stuttgart

dominik.schadow@bridging-it.de
www.bridging-it.de

Blog blog.dominikschadow.de
Twitter/ADN @dschadow

Demo Projects
github.com/dschadow/JavaSecurity

OWASP
www.owasp.org

Pictures
www.dreamstime.com