Java-Web-Security Anti-Patterns

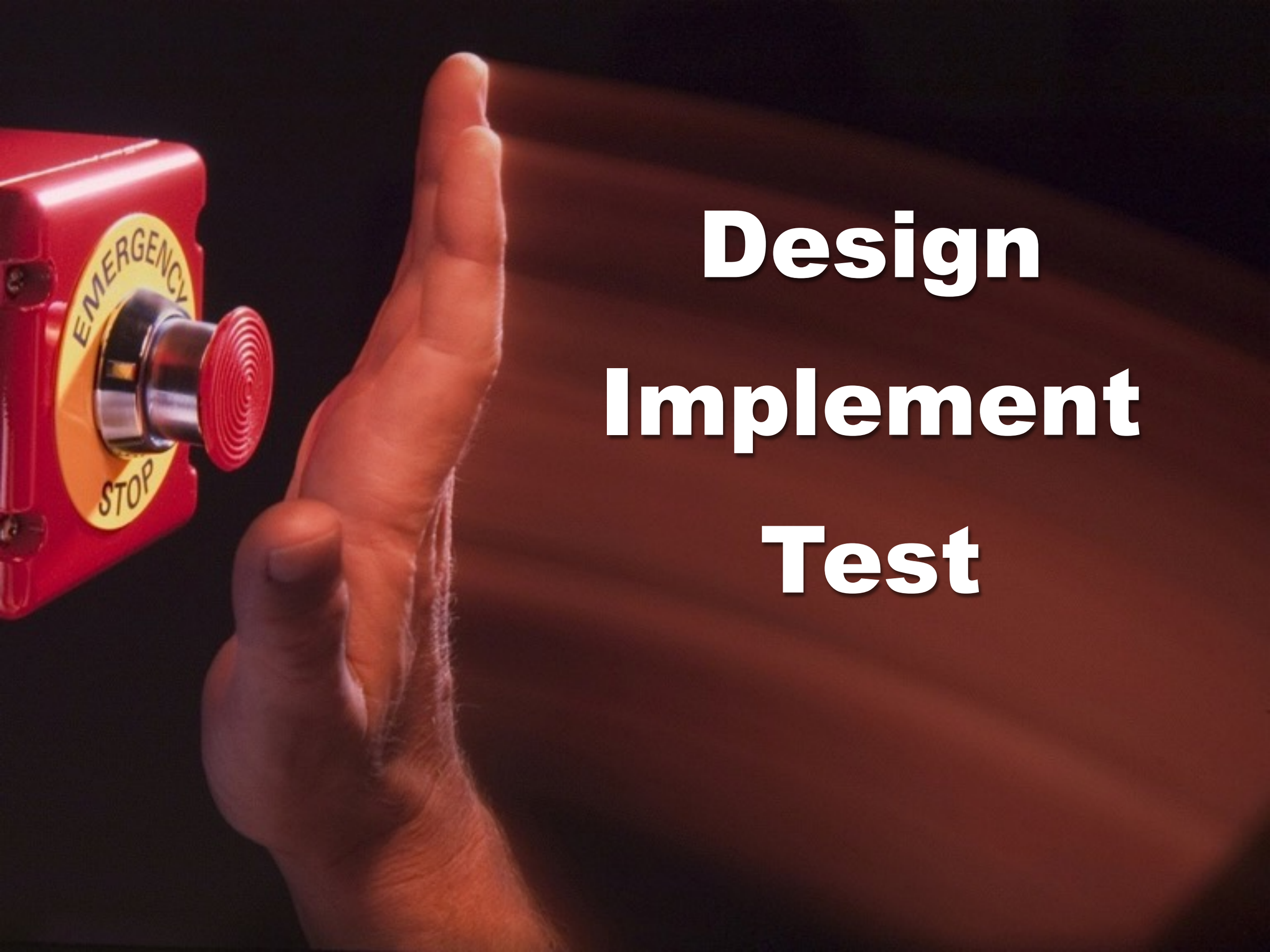
JAX | 23.04.2015

Dominik Schadow | bridgingIT

Failed with nothing but the best intentions

```
PreparedStatement pstmt =
   con.prepareStatement(
      "SELECT * FROM customer " +
      "WHERE name = '" + name + "' " +
      "ORDER BY CUST_ID");
ResultSet rs = pstmt.executeQuery();
```

```
PreparedStatement pstmt =
  con.prepareStatement(
    "SELECT * FROM customer " +
    "WHERE name = ? " +
      "ORDER BY CUST_ID");
pstmt.setString(1, name);
ResultSet rs = pstmt.executeQuery();
```


EMERGENCY
STOP
Design
Implement
Test

Design

Ignore

… threat modeling

… defense in depth

Threat model prior to implementation

Develop software that is secure by design

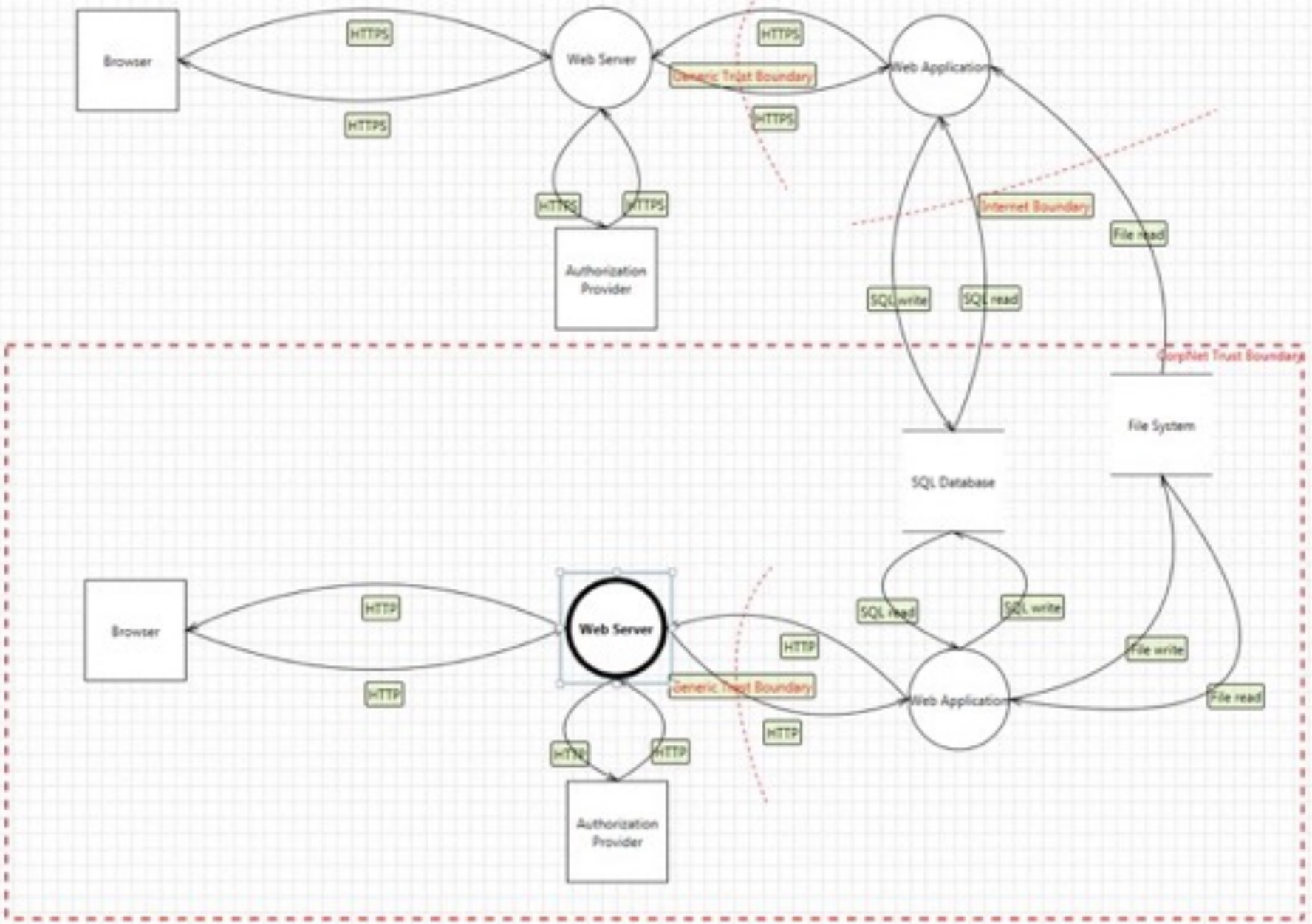
Know the web application

- Environment
- All external entities

Complex Threat Model - Threat Modeling Tool 2024

File Edit View Settings Diagram Reports Help

Portal



Stencils

Process | External | Store | Flow | Boundary

- Generic Process
- OS Process
- Thread
- Kernel Thread
- Native Application
- Managed Application
- Thick Client
- Browser Client
- Browser and ActiveX Plug-ins
- Web Server
- Windows Store Process
- Win32 Service
- Web Application

Properties

Web Server

Name	Web Server
Out Of Scope	
Reason For Out Of Scope	

Configurable Attributes

Code Type	Managed
Sanitizes Input	Not Selected
Sanitizes Output	Not Selected

As Generic Process

Running As	Not Selected
Isolation Level	Not Selected
Accepts Input From	Not Selected
Implements or Uses an Authentication Mechanism	No
Implements or Uses an Authorization Mechanism	No
Implements or Uses a Communication Protocol	No

Add New Custom Attribute

Messages - 4 issues found

Description	Severity	Diagram	Ignore
External interactor should communicate over trust boundary.	Warning	Portal	
External interactor should communicate over trust boundary.	Warning	Portal	
External interactor should communicate over trust boundary.	Warning	Portal	
External interactor should communicate over trust boundary.	Warning	Portal	

Messages - 4 issues found | Notes - no entries

Defense in depth

Utilize every software involved

Implement

Passwords stored

… as plaintext (+/- db encryption)

… encrypted

… as a simple hash (+/- salt)

.com

Password Retriever

Forgotten your password? No problem! Just enter your email address and postcode
with your password will be also sent to you, please make sure the email address er

Email *

Display password on screen directly

Retrieve

Important: for your security, please change your password after you get the old one back. To change your password, please Go Here

Storing passwords

… as plaintext

… encrypted

… simply hashed

Password hash algorithms

1. **PBKDF2 (multiple rounds)**
   - Supported on many platforms
   - Iteration count against brute force attacks
2. **bcrypt (multiple rounds)**
   - Password length up to 56 bytes
   - Iteration count against brute force attacks
3. **scrypt (resource (RAM) intensive)**
   - Relatively new
   - Hardware resources against brute force attacks

Demo

Plan for the future

 # iterations must change with new hardware

- Make it configurable
- Define period of time to update all passwords

Update hashed user passwords during log-in

- Change the salt
- Calculate new hash with new # iterations

Deactivate not updated user accounts

- Set password to null
- Setup password reset process to change

Passwords

Change salt during password change

- Never reuse same salt

Add length limit to password fields

- 1024 or 2048 characters are reasonable
- Hashing always reduces it to the same size

Ask for old password during password change

- Form prone to Cross-Site Request Forgery?
- Attacker gained access to session id?

Do you ask for the password when changing the email address?

Login form

… prevents pasting passwords

… delivered via HTTP

Say yes to pasting into password fields

- Does not stop any attack
- Does not provide any more security
- But does frustrate users

.com

Email address

Password

Forgot password?

Remember me on this computer

Sign in

"Remember me" uses a cookie. View our Cookie Policy.

Sign in with Twitter

HTTP login page puts security in jeopardy

Attacker might change the log-in form action

Load the log-in form over HTTPS

Link to a dedicated HTTPS log-in page

HSTS to force HTTPS for whole page

```
@WebFilter(urlPatterns = {"/*"})
public class HSTS implements Filter {
  public void doFilter(…) {
    HttpServletResponse response =
     (HttpServletResponse) res;
    response.addHeader(
      "Strict-Transport-Security",
      "max-age=31556926");

    chain.doFilter(req, response);
  }
  // …
}
```

Session

… without configuration

… without changing session id

```
<plugin>
 <groupId>
  org.apache.maven.plugins
 </groupId>
 <artifactId>
  maven-war-plugin
 </artifactId>
 <version>2.6</version>
 <configuration>
  <failOnMissingWebXml>
   false
  </failOnMissingWebXml>
 </configuration>
</plugin>
```

web.xml

session-timeout: 30 (or 60 or 90)

- Idle time in minutes after session expires

http-only: true

- Prevent script access to session cookie

secure: true

- Transfer session cookie only via HTTPS

tracking-mode: cookie

- Prevent session rewriting by storing session id in cookie, not in URL

```
<web-app ... version="3.1">
 <!-- ... -->
 <session-config>
  <!-- idle timeout -->
  <session-timeout>30</session-timeout>
  <cookie-config>
   <!-- prevent script access -->
   <http-only>true</http-only>
   <!-- HTTPS only -->
   <secure>true</secure>
  </cookie-config>
  <!-- no session rewriting -->
  <tracking-mode>COOKIE</tracking-mode>
 </session-config>
</web-app>
```


4E01EF46D8446D1C1
0CB5C08EDA69DD1
User usually receives a session id when visiting web application

Attacker dictates user's session id

- Attacker uses physical access, URL manipulation or Cross-Site Scripting to create a session id
- Victim logs in, keeps using session id

Mitigate session fixation

- Limit session duration
- Always show logout link
- **Change session id after log-in**

Demo

Accessing data

… without verifying ownership

[…] **one end-point that did not properly validate that the user carrying out the action was a registered owner of the lock.** […] **an attacker could craft a request to add themselves as a Guest to any lock, provided they know the lock's UUID.**

blog.perfectlylogical.com/post/2015/03/29/Making-Smart-Locks-Smarter

Permissions

Don’t use any data from frontend unvalidated

- Form data, headers, cookie, …
- *Unchangeable* data can be manipulated too
  - Intercepted and manipulated during request

Don’t use only frontend values to load data

- User (id) is backend data only

Ensure user permissions for requested data

- Verify user actually has permission for the requested id

```
String details(@PathVariable int id,…){
 Contact c = template.queryForObject(
  "SELECT * FROM contacts WHERE id = ?
   AND username = ?",
  new Object[]{id, getUsername()},
    (rs, rowNum) -> {
     Contact contact = // …
 });

 // …
}
```

```
private String getUsername() {
    return ((User) SecurityContextHolder
        .getContext().getAuthentication()
        .getPrincipal()).getUsername();
}
```

Test

Deployment

… without tested security

… with outdated 3rd party libs

Test the functionality you rely on

Untitled Session - OWASP ZAP 2.4.0

Standard mode

Sites

Contexts
Default Context
Sites
http://localhost:8080
GET:xss
xss
POST:escaped(outputEscaped
POST:csp(cspName)
GET:index.jsp
POST:escaped.jsp(outputEsca
resources
css
GET:styles.css
GET:css
GET:resources
POST:validated(inputValidated
POST:unprotected(unprotectec

Quick Start | Request | Response

Header: Text | Body: Text

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 266
Date: Fri, 17 Apr 2015 10:57:51 GMT
```

```
<html><head>
<title>XSS - Input Validation</title>
<link rel="stylesheet" type="text/css" href="resources/css/styles.css" />
</head>
<body>
<h1>XSS - Input Validation</h1>
<p>[</p><script>alert(1);</script><p>]</p>
<p><a href="index.jsp">Home</a></p>
</body></html>
```

History | Search | Alerts | Output | Spider | Fuzzer | Active Scan

Alerts (5)
Cross Site Scripting (Reflected) (3)
POST: http://localhost:8080/xss/csp
POST: http://localhost:8080/xss/unprotected
POST: http://localhost:8080/xss/validated
X-Frame-Options Header Not Set (10)
Cookie set without HttpOnly flag (4)
Web Browser XSS Protection Not Enabled (10)
X-Content-Type-Options Header Missing (10)

Cross Site Scripting (Reflected)

URL:	http://localhost:8080/xss/validated
Risk:	High
Confidence:	Medium
Parameter:	inputValidatedName
Attack:	<script>alert(1);</script>
Evidence:	<script>alert(1);</script>
CWE Id:	79
WASC Id:	8

Description:

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also

Other Info:

Solution:

Alerts 1 1 3 0

Current Scans 0 0 0 0 0 0 0

Frameworks and libraries decline

```
Last login: Fri Apr 17 12:14:00 on ttys000
Marvin:sql-injection dos$ dependency-check.sh -a SQL-Injection -s target/dependency/
Apr 17, 2015 12:15:22 PM org.owasp.dependencycheck.Engine doUpdates
INFORMATION: Checking for updates
Apr 17, 2015 12:16:08 PM org.owasp.dependencycheck.data.update.task.DownloadTask call
INFORMATION: Download Started for NVD CVE - Modified
Apr 17, 2015 12:16:32 PM org.owasp.dependencycheck.data.update.task.DownloadTask call
INFORMATION: Download Complete for NVD CVE - Modified
Apr 17, 2015 12:16:32 PM org.owasp.dependencycheck.data.update.task.ProcessTask processFi
les
INFORMATION: Processing Started for NVD CVE - Modified
Apr 17, 2015 12:16:37 PM org.owasp.dependencycheck.data.update.task.ProcessTask processFi
les
INFORMATION: Processing Complete for NVD CVE - Modified
Apr 17, 2015 12:16:37 PM org.owasp.dependencycheck.data.update.StandardUpdate update
INFORMATION: Begin database maintenance.
Apr 17, 2015 12:16:44 PM org.owasp.dependencycheck.data.update.StandardUpdate update
INFORMATION: End database maintenance.
Apr 17, 2015 12:16:45 PM org.owasp.dependencycheck.Engine doUpdates
INFORMATION: Check for updates complete
Apr 17, 2015 12:16:45 PM org.owasp.dependencycheck.Engine analyzeDependencies
INFORMATION: Analysis Starting
Apr 17, 2015 12:17:13 PM org.owasp.dependencycheck.analyzer.CentralAnalyzer analyzeFileTy
pe
WARNUNG: Unable to download pom.xml for hibernate-jpa-2.1-api-1.0.0.Final.jar from Centra
l; this could result in undetected CPE/CVEs.
Apr 17, 2015 12:17:25 PM org.owasp.dependencycheck.Engine analyzeDependencies
INFORMATION: Analysis Complete
Marvin:sql-injection dos$
```

DEPENDENCY-CHECK

Project: SQL

Scan Information (show all):

- *dependency-check version*: 1.2.10
- *Report Generated On*: Apr 14, 2015 at 20:32:34 MESZ
- *Dependencies Scanned*: 44
- *Vulnerable Dependencies*: 3
- *Vulnerabilities Found*: 5
- *Vulnerabilities Suppressed*: 0
- ...

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
batik-css-1.7.jar	cpe:/a:apache:batik:1.7	org.apache.xmlgraphics:batik-css:1.7	Medium	1	HIGHEST	14
commons-fileupload-1.2.jar	cpe:/a:apache:commons_fileupload:1.2	commons-fileupload:commons-fileupload:1.2	Medium	2	HIGHEST	14
commons-httpclient-3.1.jar	cpe:/a:apache:commons-httpclient:3.1	commons-httpclient:commons-httpclient:3.1	Medium	2	LOW	11

Dependencies

batik-css-1.7.jar

License:

```
The Apache Software License, Version 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt
```

File Path: target/dependency/batik-css-1.7.jar
MD5: b0203e64b3c06729baa0ef84743ab119
SHA1: e6bb5c8575333153459 3f33fb9236acb41a0ab79

Evidence

Related Dependencies

Identifiers

- **cpe:** cpe:/a:apache:batik:1.7 *Confidence:HIGHEST* suppress
- **maven:** org.apache.xmlgraphics:batik-css:1.7 *Confidence:HIGHEST*

Published Vulnerabilities

CVE-2015-0250 suppress

Integrate ZAP and Dependency Check into your Jenkins toolchain

- Prepare implementation with threat modeling
- Think when implementing security functionality
- Test your application and 3rd party libraries

BridgingIT GmbH

Königstraße 42
70173 Stuttgart

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

Blog blog.dominikschadow.de
Twitter @dschadow

Demo Projects
github.com/dschadow/JavaSecurity

HTTP Strict Transport Security RFC
tools.ietf.org/html/rfc6797

Microsoft Threat Modeling Tool
www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx

Mozilla SeaSponge
air.mozilla.org/mozilla-winter-of-security-seasponge-a-tool-for-easy-threat-modeling

OWASP Dependency Check
www.owasp.org/index.php/OWASP_Dependency_Check

OWASP ZAP
www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

Recx Security Analyser
www.recx.co.uk/products/chromeplugin.php

Spring Security
projects.spring.io/spring-security

Pictures
www.dreamstime.com