# Real World Java Web Security

## Java User Group Karlsruhe

Dominik Schadow | bridgingIT

# Who thinks about ...

... architecture while coding?

... architecture before coding?

# Who thinks about ...

... security while coding?

... security before coding?

# OWASP TOP 10 2013

(1) Injection
(2) Broken Authentication and Session Management
(3) Cross-Site Scripting (XSS)
(4) Insecure Direct Object References
(5) Security Misconfiguration
(6) Sensitive Data Exposure
(7) Missing Function Level Access Control
(8) Cross-Site Request Forgery (CSRF)
(9) Using Components with Known Vulnerabilities
(10) Unvalidated Redirects and Forwards

# Software that is secure by design

Know the web application

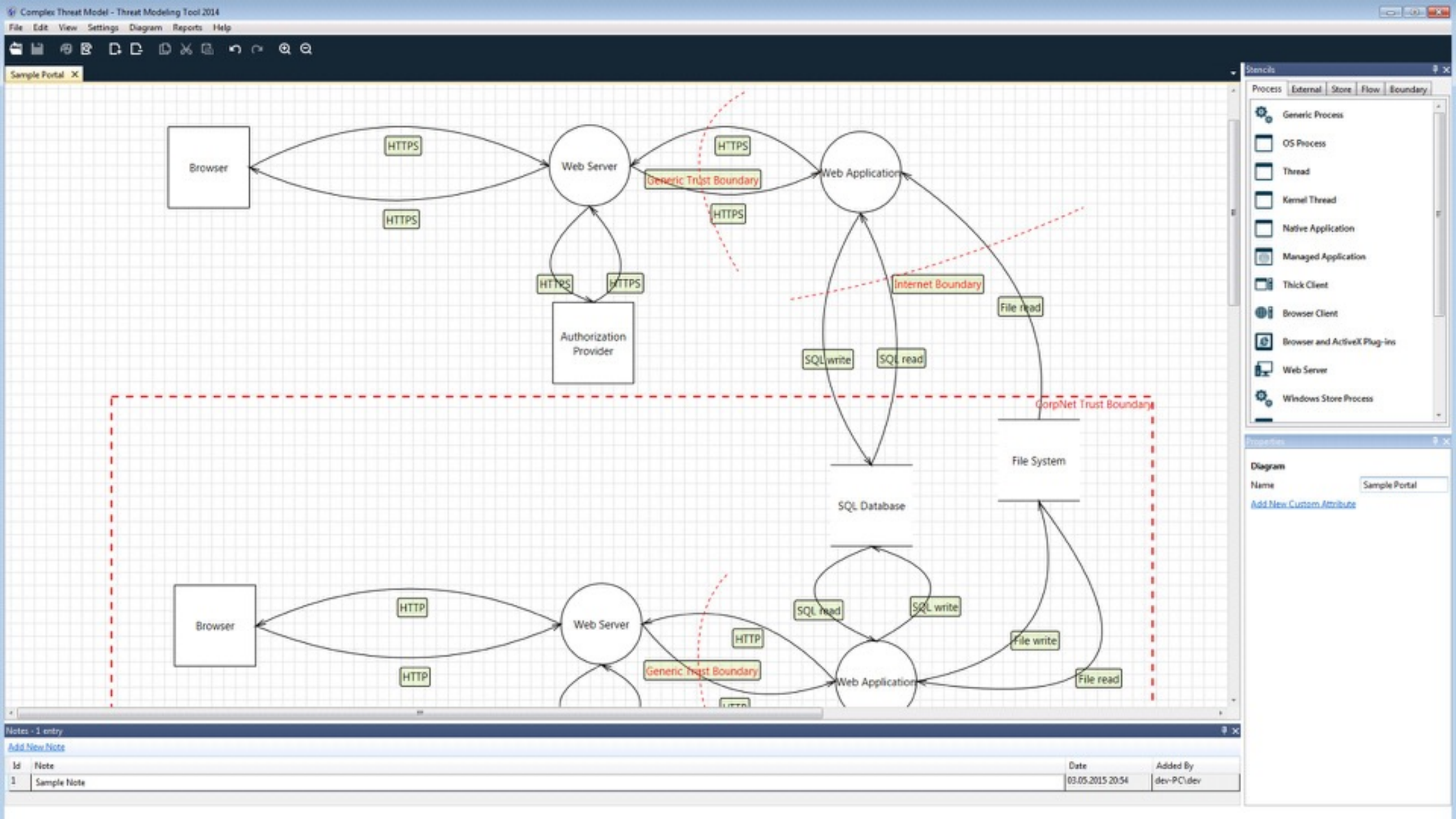
Know all external entities

Identify all data flows

**Identify all risks**

# Threat model

# Avoid design flaws


Complex Threat Model - Threat Modeling Tool 2014
File Edit View Settings Diagram Reports Help
Sample Portal
Browser
HTTPS
HTTPS
Web Server
HTTPS
Generic Trust Boundary
HTTPS
Web Application
HTTPS
HTTPS
Authorization Provider
Internet Boundary
File read
SQL write
SQL read
CorpNet Trust Boundary
File System
SQL Database
Browser
HTTP
HTTP
Web Server
HTTP
Generic Trust Boundary
SQL read
SQL write
File write
File read
Web Application
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
Properties
Diagram
Name
Sample Portal
Add New Custom Attribute
Notes - 1 entry
Add New Note
Id
Note
Date
Added By
1
Sample Note
03.05.2015 20:54
dev-PC\dev


Fight the
identified
threats


Maintain all threat models

# Instrument the Browser

# Defense in Depth

# Force HTTPS

```
@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);
  }
  // …
}
```

```
@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);
  }
  // …
}
```

```
@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);
  }
  // …
}
```

```
@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);
  }
  // …
}
```

```
@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);
  }
  // …
}
```

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

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

# Prevent framing

```
response.addHeader(
   "X-Frame-Options",
   "DENY"
);
```

```
response.addHeader(
    "X-Frame-Options",
    "DENY"
);
```

```
response.addHeader(
   "X-Frame-Options",
   "DENY"
);
```

```
response.addHeader(
   "X-Frame-Options",
   "SAME-ORIGIN"
);
```

```
response.addHeader(
   "X-Frame-Options",
   "ALLOW-FROM http://www.safe.de"
);
```

# Prevent Cross-Site Scripting

```
response.addHeader(
   "Content-Security-Policy",
   "default-src 'self'"
);
```

```
response.addHeader(
    "Content-Security-Policy",
    "default-src 'self'"
);
```

```
response.addHeader(
   "Content-Security-Policy",
   "default-src 'self'"
);
```

# Content Security Policy Directives

| | |
|---|---|
| **default-src** | default if specific directive is not set |
| **object-src** | Sources in object, embed or applet tags |
| **script-src** | Script sources (includes XSLT) |
| connect-src | XMLHttpRequest, WebSocket, … |
| font-src | Font sources |
| frame-src | Sources embeddable as frames |
| img-src | Image sources |
| media-src | Video and audio sources |
| style-src | CSS sources (does not include XSLT) |

www.w3.org/TR/CSP

```
response.addHeader(
   "Content-Security-Policy",
   "default-src 'none';
    script-src 'self';
    style-src 'self';
    img-src 'self';
    report-uri CSPReporting"
);
```

```
response.addHeader(
   "Content-Security-Policy",
   "default-src 'none';
    script-src 'self';
    style-src 'self';
    img-src 'self';
    report-uri CSPReporting"
);
```

# Violation Report


```
{
 "document-uri":"http://.../reporting.jsp?
   name=%3Cscript%3Ealert(%27XSS%27)%3C/script%3E",
 „referrer“:"http://www.sample.com/security-header/
   index.jsp",
 "blocked-uri":"self",
 "violated-directive":"default-src http://www.sample.com",
 "source-file":"http://.../reporting.jsp?
   name=%3Cscript%3Ealert(%27XSS%27)%3C/script%3E",
 "script-sample":"alert('XSS')",
 "line-number":10
}
```

# Content Security Policy Level 2

| | |
|---|---|
| **frame-ancestors** | Allow resource frame embedding<br>Obsoletes X-Frame-Options header |
| **reflected-xss** | (De-)activate user agent XSS heuristics<br>Obsoletes X-XSS-Protection header |
| child-src | Replaces frame-src |
| form-action | Form targets to send data to |
| plugin-types | Allowed plug-ins (their MIME type) |
| referrer | Referrer URL exposed to others |
| sandbox | Load resource in restricted sandbox |

www.w3.org/TR/CSP2

```
response.addHeader(
   "Content-Security-Policy",
   "default-src 'self';
    frame-ancestors 'none'"
);
```

```
response.addHeader(
   "Content-Security-Policy",
   "default-src 'self';
    frame-ancestors 'none'"
);
```

# Here is your Policy!

**Content-Security-Policy: default-src 'none' ; script-src 'self' ; style-src 'self' ; img-src 'self' ; font-src 'self' ; frame-ancestors 'none' ; form-action 'self' ; upgrade-insecure-requests; block-all-mixed-content; reflected-xss block;**

## Import a policy

Enter your domain

Content-Security-Policy

Import

## Build your CSP

1) Default Source
2) Script Source
3) Style Source
4) Image Source
5) Font Source
6) Connect Source
7) Media Source
8) Object Source
9) Child Source

Default Source View Info

- [x] None
- [ ] All
- [ ] Self
- [ ] Data
- [ ] Unsafe Inline
- [ ] Unsafe Eval

Space separated list of hosts.

*Page, Header & Cookie Security Analyser*

## Analysis results for:

https://blog.dominikschadow.de/

Click the icons in the tables below for a more detailed explanation.

## HTTP security headers

| Name | Value | Setting secure |
|---|---|---|
| x-content-type-options | nosniff | ✓ |
| x-frame-options | deny | ✓ |
| cache-control | no-cache, must-revalidate, max-age=0, no-store, no-cache, must-revalidate | ✓ |
| content-security-policy | default-src 'self'; img-src *; font-src *; style-src 'self' https://fonts.googleapis.com 'unsafe-inline'; frame-ancestors 'none' | ✓ |
| strict-transport-security | max-age=31556926 | ✓ |
| x-xss-protection | 1; mode=block | ✓ |
| access-control-allow-origin | Header not returned | ✓ |

# Demo

# And now?

# OWASP TOP 10 Proactive Controls

(1) Parameterize Queries
(2) Encode Data
(3) Validate All Inputs
(4) Implement Appropriate Access Controls
(5) Establish Identity and Authentication Controls
(6) Protect Data and Privacy
(7) Implement Logging, Error Handling and Intrusion Detection
(8) Leverage Security Features of Frameworks and Security Libraries
(9) Include Security-Specific Requirements
(10) Design and Architect Security in **Threat Modeling**

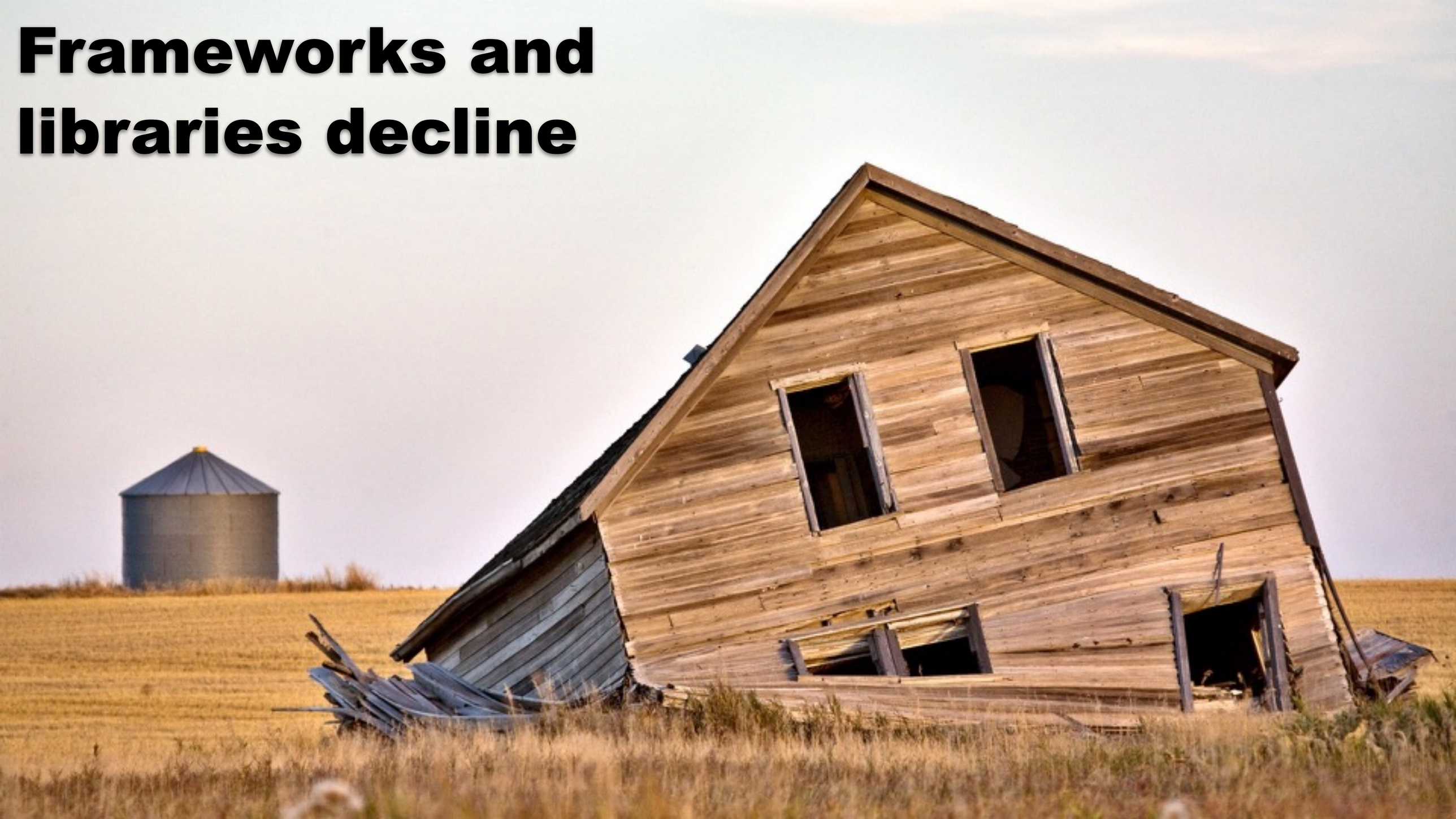
# Leverage Security Features of Frameworks and Security Libraries

# Use it!

# Spring Security (Java config) adds headers automatically

```
X-Content-Type-Options
Cache-Control
X-Frame-Options
HTTP Strict Transport Security
X-XSS-Protection
```

# Frameworks and libraries decline



```
Marvin:xss dos$ dependency-check.sh --project XSS --scan target/dependency/
[INFO] Checking for updates
[INFO] NVD CVE requires several updates; this could take a couple of minutes.
[INFO] Download Started for NVD CVE - 2007
[INFO] Download Started for NVD CVE - 2006
[INFO] Download Started for NVD CVE - 2008
[INFO] Download Complete for NVD CVE - 2007  (20151 ms)
[INFO] Processing Started for NVD CVE - 2007
[INFO] Download Started for NVD CVE - 2009
[INFO] Processing Complete for NVD CVE - 2007  (7298 ms)
[INFO] Download Complete for NVD CVE - 2009  (19850 ms)
[INFO] Download Started for NVD CVE - 2010
[INFO] Processing Started for NVD CVE - 2009
[INFO] Processing Complete for NVD CVE - 2009  (6511 ms)
[INFO] Download Complete for NVD CVE - 2006  (60332 ms)
[INFO] Processing Started for NVD CVE - 2006
[INFO] Download Started for NVD CVE - 2011
[INFO] Download Complete for NVD CVE - 2008  (60883 ms)
[INFO] Download Started for NVD CVE - 2012
[INFO] Processing Complete for NVD CVE - 2006  (3920 ms)
[INFO] Processing Started for NVD CVE - 2008
[INFO] Download Complete for NVD CVE - 2010  (24857 ms)
[INFO] Download Started for NVD CVE - 2013
[INFO] Processing Complete for NVD CVE - 2008  (4563 ms)
[INFO] Processing Started for NVD CVE - 2010
[INFO] Processing Complete for NVD CVE - 2010  (10100 ms)
[INFO] Download Complete for NVD CVE - 2012  (34614 ms)
[INFO] Processing Started for NVD CVE - 2012
[INFO] Download Started for NVD CVE - 2014
[INFO] Processing Complete for NVD CVE - 2012  (11777 ms)
[INFO] Download Complete for NVD CVE - 2014  (25895 ms)
[INFO] Processing Started for NVD CVE - 2014
[INFO] Download Started for NVD CVE - 2015
[INFO] Download Complete for NVD CVE - 2013  (59494 ms)
[INFO] Download Started for NVD CVE - Modified
```

```
<reporting>
 <plugins><plugin>
  <groupId>org.owasp</groupId>
  <artifactId>dependency-check-maven</artifactId>
  <version>1.3.1</version>
  <reportSets>
   <reportSet>
    <reports>
     <report>aggregate</report>
    </reports>
    </reportSet>
   </reportSets>
  </plugin></plugins>
</reporting>
```

## Post-build Actions

### Publish OWASP Dependency-Check analysis results

Dependency-Check results

Fileset includes setting that specifies the generated raw Dependency-Check XML report files, such as **/dependency-check-report.xml. Basedir of the fileset is the workspace root. If no value is set, then the default **/dependency-check-report.xml is used. Be sure not to include any non-report files into this pattern.

Run always

By default, this plug-in runs only for stable or unstable builds, but not for failed builds. If this plug-in should run even for failed builds then activate this check box.

Detect modules

Determines if Ant or Maven modules should be detected for all files that contain warnings. Activating this option may increase your build time since the detector scans the whole workspace for 'build.xml' or 'pom.xml' files in order to assign the correct module names.

Health thresholds

100% 0%

Configure the thresholds for the build health. If left empty then no health report is created. If the actual number of warnings is between the provided thresholds then the build health is interpolated.

Health priorities

Only priority high Priorities high and normal All priorities

Determines which warning priorities should be considered when evaluating the build health.

Status thresholds (Totals)

| | All priorities | Priority high | Priority normal | Priority low |
|---|---|---|---|---|
| | 5 | 2 | 3 | 5 |
| | | | | |

If the number of total warnings is greater than one of these thresholds then a build is considered as unstable or failed, respectively. I.e., a value of 0 means that the build status is changed if there is at least one warning found. Leave this field empty if the state of the build should not depend on the number of warnings.

Compute new warnings (based on the last successful build unless another reference build is chosen below)

Default Encoding

# DEPENDENCY-CHECK



## Project: Java-Web-Security

Scan Information (show all):

- *dependency-check version*: 1.3.0
- *Report Generated On*: Aug 8, 2015 at 08:58:47 CEST
- *Dependencies Scanned*: 74
- *Vulnerable Dependencies*: 1
- *Vulnerabilities Found*: 2
- *Vulnerabilities Suppressed*: 0
- ...

Display: Showing Vulnerable Dependencies (click to show all)

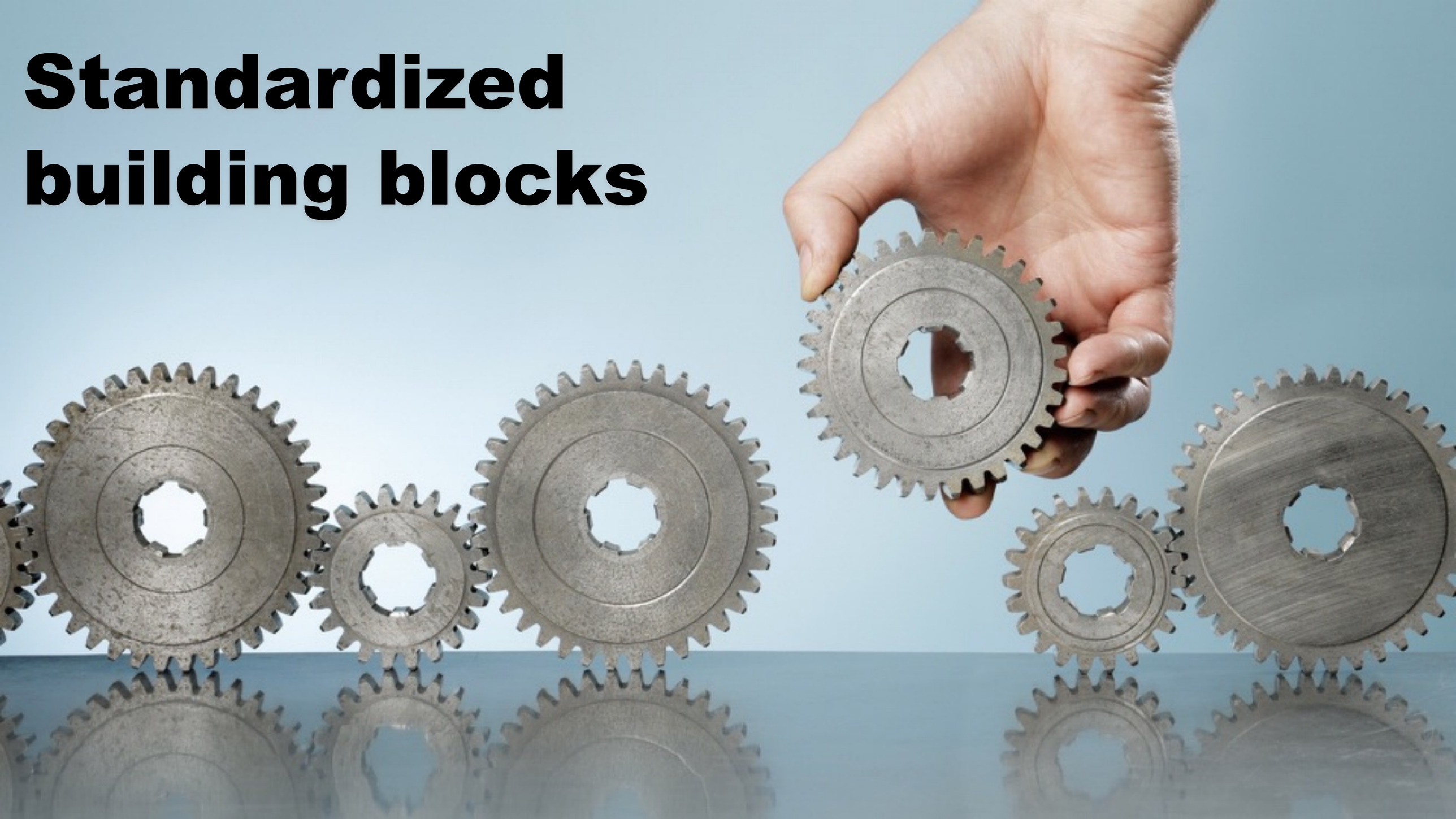
| Dependency | CPE | GAV | Highest Severity | CVE Count | CPE Confidence | Evidence Count |
|---|---|---|---|---|---|---|
| commons-fileupload-1.2.jar | cpe:/a:apache:commons_fileupload:1.2 | commons-fileupload:commons-fileupload:1.2 | Medium | 2 | HIGHEST | 23 |

## Dependencies

### commons-fileupload-1.2.jar

# Implement Appropriate Access Controls
# Establish Identity and Authentication Controls

# Standardized building blocks




# User usually receives a session id when visiting web application

# Demo

# Protect Data and Privacy

# Slow down brute force attacks

**PBKDF2**

Iterations against brute force attacks

Available in plain Java

# Demo

**bcrypt**

Iterations against brute force attacks

Integrated in Spring Security

```
@Configuration
@EnableWebMvcSecurity
public class WebSecurityConfig extends
 WebSecurityConfigurerAdapter {
 @Bean
 public PasswordEncoder passwordEncoder() {
  return new BCryptPasswordEncoder(10);
 }
}
```

```
@Configuration
@EnableWebMvcSecurity
public class WebSecurityConfig extends
 WebSecurityConfigurerAdapter {
 @Bean
 public PasswordEncoder passwordEncoder() {
  return new BCryptPasswordEncoder(10);
 }
}
```

```
@Configuration
@EnableWebMvcSecurity
public class WebSecurityConfig extends
 WebSecurityConfigurerAdapter {
  @Bean
  public PasswordEncoder passwordEncoder() {
   return new BCryptPasswordEncoder(10);
  }
}
```

```
@Configuration
@EnableWebMvcSecurity
public class WebSecurityConfig extends
 WebSecurityConfigurerAdapter {
 @Bean
 public PasswordEncoder passwordEncoder() {
  return new BCryptPasswordEncoder(10);
 }
}
```

**scrypt**

Memory against brute force attacks

Best protection against dictionary attacks

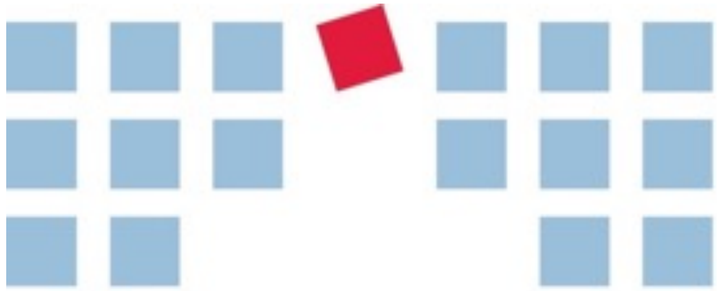
# Summary

# Plan security with threat modeling

# Think (like an attacker) during implementation

# Keep 3rd party libraries up-to-date

Enjoy secure programming



# bridging IT


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

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

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

**OWASP TOP 10**
www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

**OWASP TOP 10 Proactive Controls**
www.owasp.org/index.php/OWASP_Proactive_Controls

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

**Spring Security**
projects.spring.io/spring-security

**Pictures**
www.dreamstime.com

**Jobs@bridgingIT**
www.bridging-it.de/java