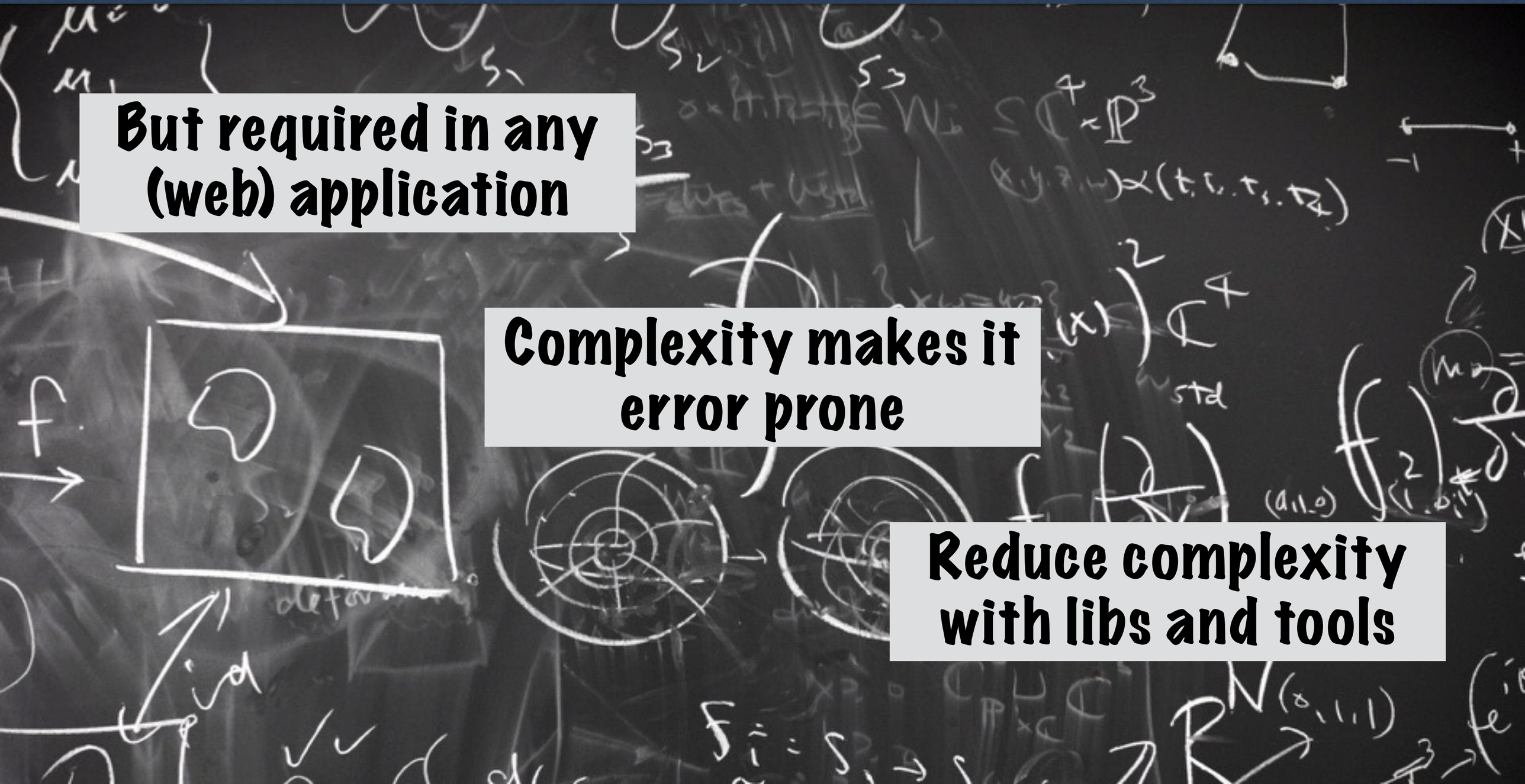
# 7 security tools and libraries every developer should know about

Java User Group Hamburg
Dominik Schadow | bridgingIT

# Security is a complex topic



But required in any (web) application

Complexity makes it error prone

Reduce complexity with libs and tools

# Spoilt for choice: frameworks and libraries

# Security has to be tested like any other functionality

# **My** seven tools and libs

Microsoft Threat Modeling Tool

OWASP Java Encoder

OWASP ZAP

Recx Security Analyser

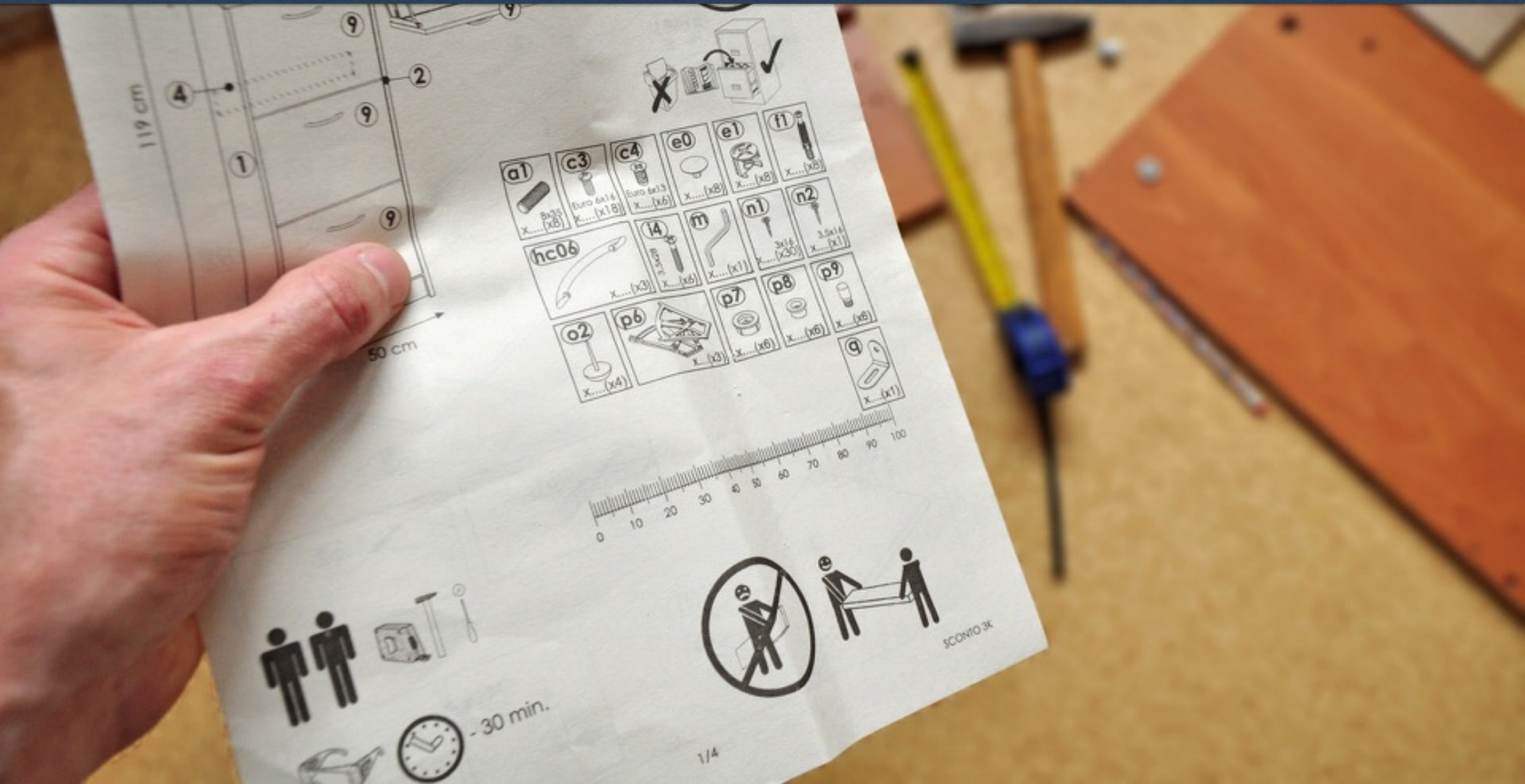
Jasypt

Keyczar

OWASP Dependency Check

# What about Spring Security?

# So you are developing a new web application?!



# Who thinks about ...

... architecture while coding?

... architecture before coding?

# Who thinks about ...

... security while coding?

... security before coding?

# Software that is secure by design

Know the web application

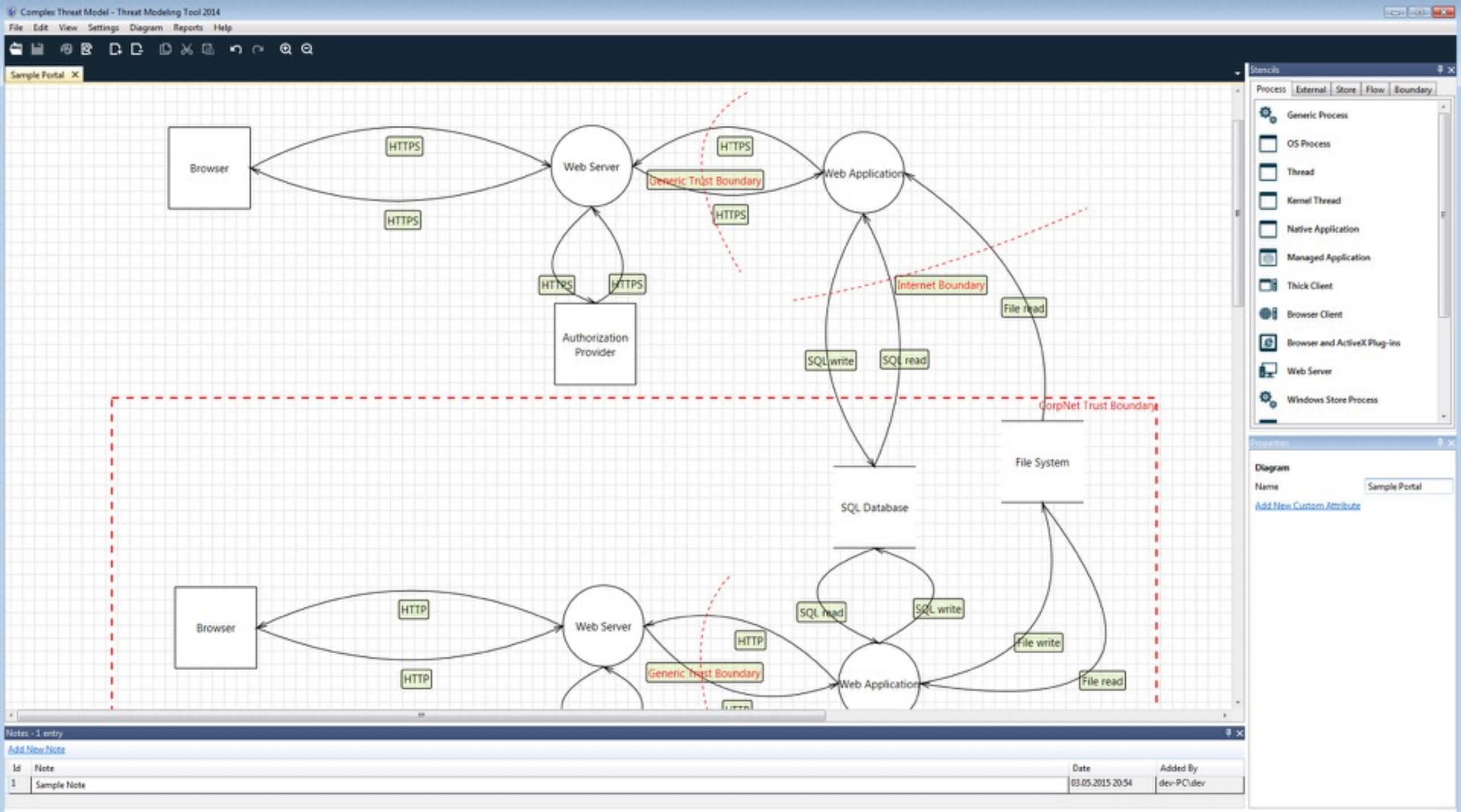
Know all external entities

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
Generic Trust Boundary
HTTPS
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
Generic Trust Boundary
HTTP
SQL read
SQL write
Web Application
File write
File read
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

SeaSponge

mozilla.github.io/seasponge/#/draw

SeaSponge | File | Zoom-In | Zoom-Out | Web Application Sample | Refresh Sharable Link | Share link | Help

Model Information

Diagram Information

New Threat

Threat Information

High SQL Injection

Stencils

Stencil Properties - Generic Trust Boundary

Boundary

Delete Element

Title

Database

Category

Boundary

Tags

Comma separated tags

Icon

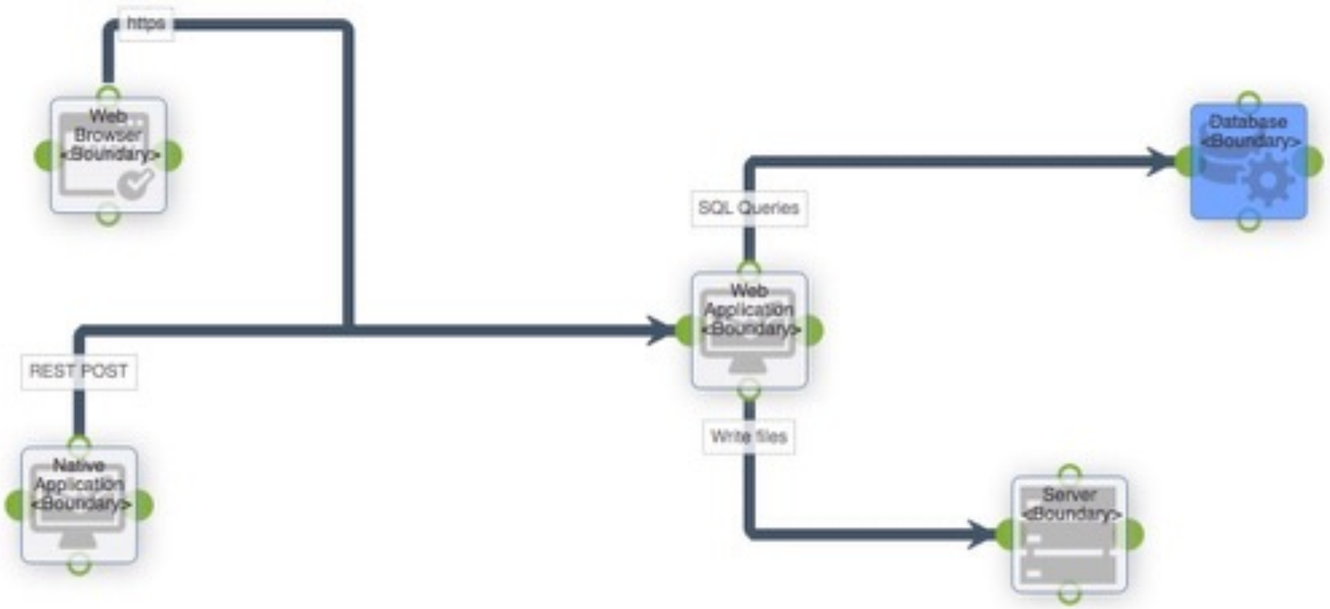
images/icons/database11.svg

Code Type

Managed

Running As

Web Application Sample | + Diagram | Reports



# Threat modeling key questions*

1. What are you building?
2. What can go wrong?
3. What should you do about those things that can go wrong?
4. Did you do a decent job of analysis in steps 1 to 3?

* by Adam Shostack - Threat Modeling

Fight the
identified
threats

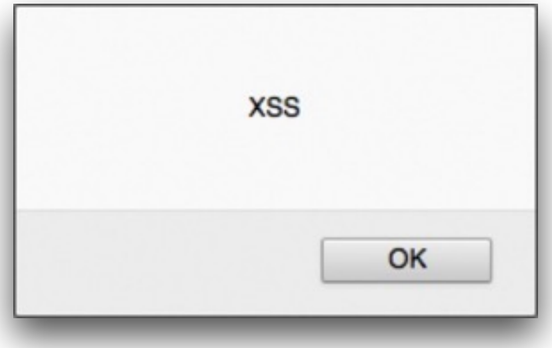
Maintain all threat models

# Demo

# Know all data entering and leaving your application

# Escape all untrusted data for the proper context

```
<script>alert("XSS")</script>
```




## HTML

```
&lt;script&gt;alert(&#34;XSS&#34;)&lt;/script&gt;
```

## CSS

```
\3cscript\3e alert(\22XSS\22)\3c\2fscript\3e
```

## JavaScript

```
<script>alert(\x22XSS\x22)<\/script>
```

# OWASP Java Encoder

```
String encHTML = Encode.forHtml(name);
String encCSS  = Encode.forCssString(name);
String encJS   = Encode.forJavaScript(name);
```

```
forUri(String input)                                  String
forJavaScript(String input)                           String
forJava(String input)                                 String
forJava(Writer out, String input)                     void
forCssString(String input)                            String
forHtmlAttribute(String input)                        String
forHtml(String input)                                 String
forHtml(Writer out, String input)                     void
forHtmlContent(String input)                          String
forCDATA(String input)                                String
forCssUrl(String input)                               String
forHtmlUnquotedAttribute(String input)                String
forJavaScriptAttribute(String input)                  String
forJavaScript(Writer out, String input)               void
forJavaScriptBlock(String input)                      String
forJavaScriptSource(String input)                     String
forUriComponent(String input)                         String
forUri(Writer out, String input)                      void
```

```
forXml(String input)                                  String
forXmlAttribute(String input)                         String
forXml(Writer out, String input)                      void
forXmlComment(String input)                           String
forXmlContent(String input)                           String
forCDATA(Writer out, String input)                    void
forCssString(Writer out, String input)                void
forCssUrl(Writer out, String input)                   void
forHtmlAttribute(Writer out, String input)            void
forHtmlContent(Writer out, String input)              void
forHtmlUnquotedAttribute(Writer out, String input)    void
forJavaScriptAttribute(Writer out, String input)      void
forJavaScriptBlock(Writer out, String input)          void
forJavaScriptSource(Writer out, String input)         void
forUriComponent(Writer out, String input)             void
forXmlAttribute(Writer out, String input)             void
forXmlComment(Writer out, String input)               void
forXmlContent(Writer out, String input)               void
```

```
<%@ page language="java" contentType="text/html;
    charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="org.owasp.encoder.Encode" %>
<%@ taglib prefix="e" uri="https://www.owasp.org/
    index.php/OWASP_Java_Encoder_Project" %>

<html>
<body>
 <%=Encode.forHtml(request.getParameter("name"))%>

 <e:forHtml value="${param.name}"/>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html;
    charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="org.owasp.encoder.Encode" %>
<%@ taglib prefix="e" uri="https://www.owasp.org/
    index.php/OWASP_Java_Encoder_Project" %>

<html>
<body>
 <%=Encode.forHtml(request.getParameter("name"))%>

 <e:forHtml value="${param.name}"/>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html;
    charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="org.owasp.encoder.Encode" %>
<%@ taglib prefix="e" uri="https://www.owasp.org/
    index.php/OWASP_Java_Encoder_Project" %>

<html>
<body>
 <%=Encode.forHtml(request.getParameter("name"))%>

 <e:forHtml value="${param.name}"/>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html;
    charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="org.owasp.encoder.Encode" %>
<%@ taglib prefix="e" uri="https://www.owasp.org/
    index.php/OWASP_Java_Encoder_Project" %>

<html>
<body>
 <%=Encode.forHtml(request.getParameter("name"))%>

 <e:forHtml value="${param.name}"/>
</body>
</html>
```

# Never ever do the escaping yourself! Never!

You don’t know if a web application is secure if you don’t test security

# Test your validation and escaping boundaries

~~validated~~ input -> backend
~~escaped~~ output -> frontend

Hack yourself first

# Intercept and manipulate HTTP requests and responses

Untitled Session - OWASP ZAP 2.4.2

Standard mode

Sites

- Contexts
  - Default Context
- Sites
  - http://localhost:8080
    - GET:xss
    - xss
      - resources

Quick Start | Request | Response | Break

Method | Header: Text | Body: Text

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: de,en-US;q=0.7,en;q=0.3
DNT: 1
Referer: http://localhost:8080/xss/
Cookie: JSESSIONID=B5A47E79DAB125DBA641664FCF576000
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 23
```

```
inputValidatedName=<script>alert(document.cookie)</script>
```

History | Search | Alerts | Output | Active Scan | Break Points | Forced Browse | Fuzzer | Http Sessions | Params | Spider

- Alerts (4)
  - X-Frame-Options Header Not Set (2)
  - Cookie set without HttpOnly flag
  - Web Browser XSS Protection Not Enabled (2)
    - GET: http://localhost:8080/xss/
    - GET: http://localhost:8080/xss/resources/css/styles.c
  - X-Content-Type-Options Header Missing (2)

Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

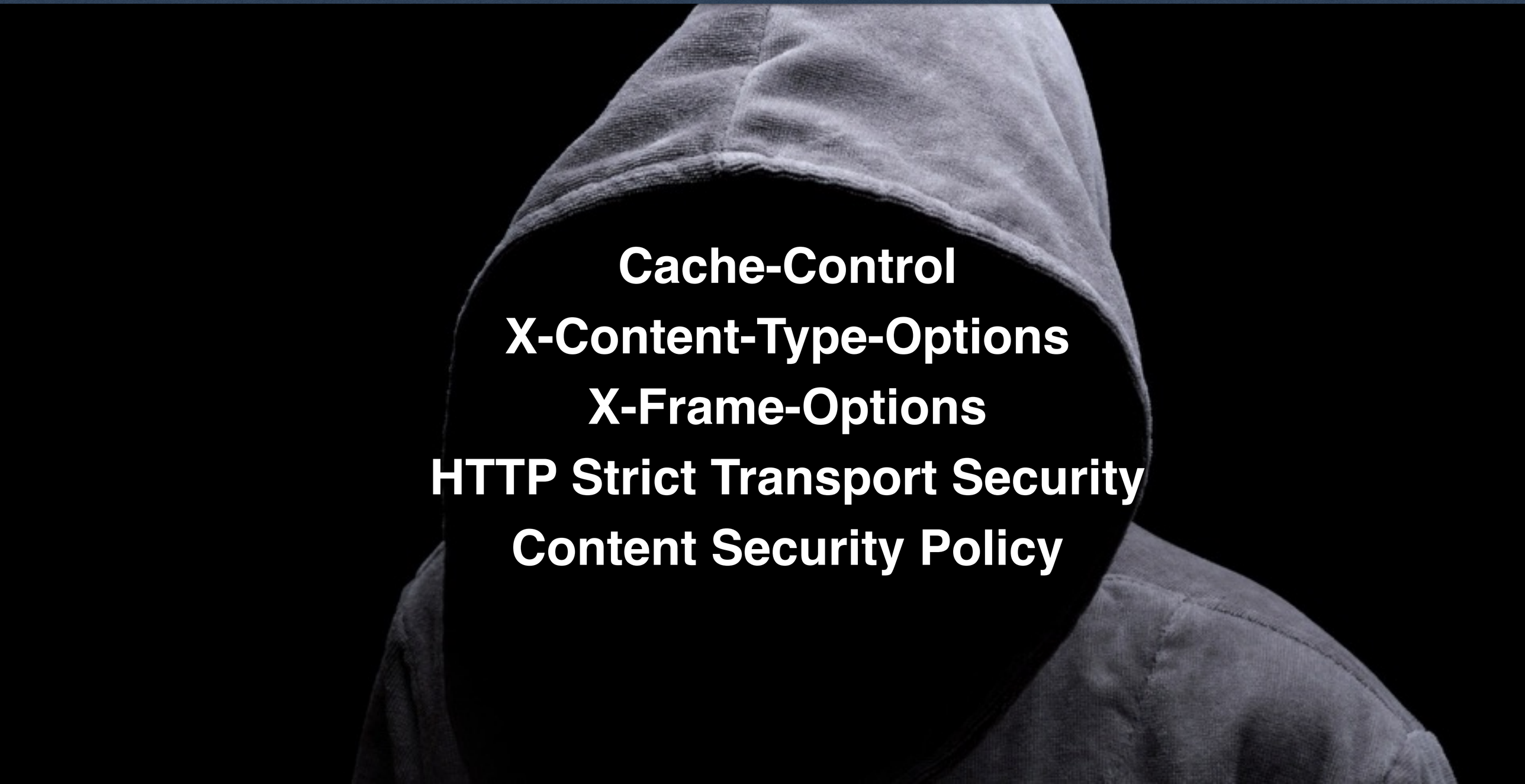
You can also edit existing alerts by double clicking on them.

Alerts 0 1 3 0

Current Scans 0 0 0 0 0 0 0

# Demo

# Security response header instruct the browser



Cache-Control

X-Content-Type-Options

X-Frame-Options

HTTP Strict Transport Security

Content Security Policy

# Use browser as part of your defense-in-depth strategy

```
@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);
  }
  // …
}
```

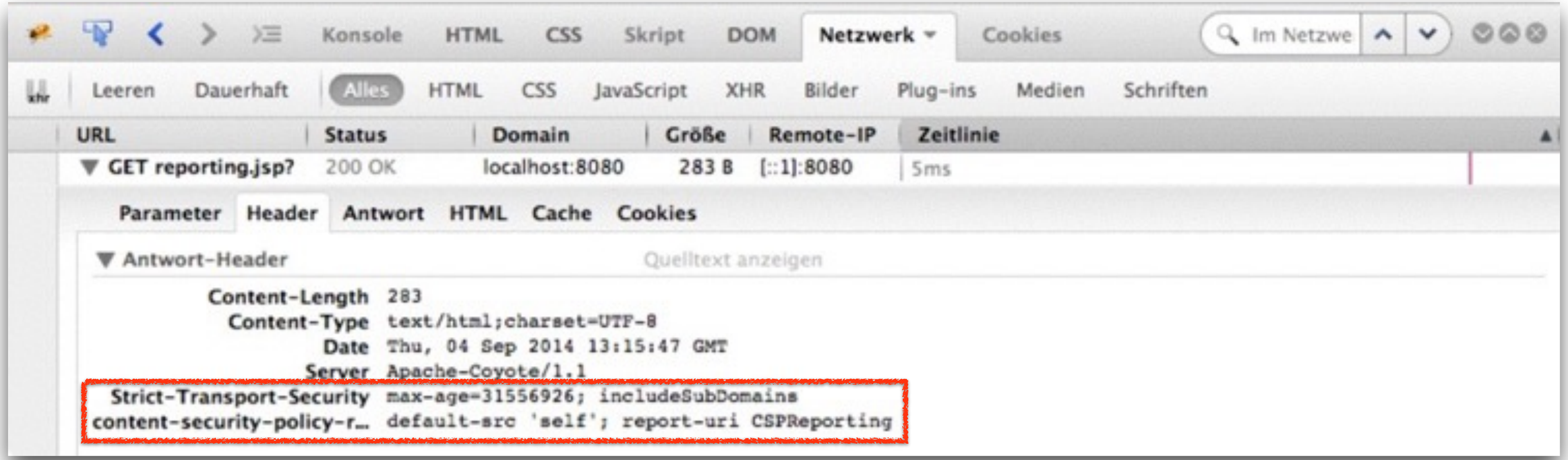
```
@WebFilter(urlPatterns = {"/*"})
public class HSTS implements Filter {
  public void doFilter(…) {
    HttpServletResponse response =
     (HttpServletResponse) res;
    response.addHeader(
     "Strict-Transport-Security",
     "max-age=31556926");

    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);
  }
  // …
}
```

# Verify headers manually with Firebug



*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

# Keep your configuration passwords confidential

Spring, Hibernate, ...

```
<context:property-placeholder
  location="classpath:spring.properties" />

<bean id="dataSource"
class="org...BasicDataSource">
  <!-- … -->
  <property name="url" value="${jdbc.url}"/>
  <property name="username" value="${jdbc.un}"/>
  <property name="password" value="${jdbc.pw}"/>
</bean>
```

```
<context:property-placeholder
  location="classpath:spring.properties" />

<bean id="dataSource"
class="org...BasicDataSource">
  <!-- … -->
  <property name="url" value="${jdbc.url}"/>
  <property name="username" value="${jdbc.un}"/>
  <property name="password" value="${jdbc.pw}"/>
</bean>
```

```
<context:property-placeholder
  location="classpath:spring.properties" />

<bean id="dataSource"
class="org...BasicDataSource">
  <!-- … -->
  <property name="url" value="${jdbc.url}"/>
  <property name="username" value="${jdbc.un}"/>
  <property name="password" value="${jdbc.pw}"/>
</bean>
```

```
jdbc.driver=org.hsqldb.jdbcDriver
jdbc.url=jdbc:hsqldb:mem:sampleDB
jdbc.un=sampleUser
jdbc.pw=samplePassword
```

```
jdbc.driver=org.hsqldb.jdbcDriver
jdbc.url=jdbc:hsqldb:mem:sampleDB
jdbc.un=sampleUser
jdbc.pw=samplePassword
```

dos — bash — 125×29

```
Marvin:~ dos$ listAlgorithms.sh

DIGEST ALGORITHMS:   [MD2, MD5, SHA, SHA-224, SHA-256, SHA-384, SHA-512]

PBE ALGORITHMS:      [PBEWITHHMACSHA1ANDAES_128, PBEWITHHMACSHA1ANDAES_256, PBEWITHHMACSHA224ANDAES_128, PBEWITHHMACSHA224AND
AES_256, PBEWITHHMACSHA256ANDAES_128, PBEWITHHMACSHA256ANDAES_256, PBEWITHHMACSHA384ANDAES_128, PBEWITHHMACSHA384ANDAES_256,
PBEWITHHMACSHA512ANDAES_128, PBEWITHHMACSHA512ANDAES_256, PBEWITHMD5ANDDES, PBEWITHMD5ANDTRIPLEDES, PBEWITHSHA1ANDDESEDE, PBE
WITHSHA1ANDRC2_128, PBEWITHSHA1ANDRC2_40, PBEWITHSHA1ANDRC4_128, PBEWITHSHA1ANDRC4_40]

Marvin:~ dos$ encrypt.sh input="sampleDbPassword" password="samplePasswordEncPassword" algorithm="PBEWITHHMACSHA1ANDAES_128"

----ENVIRONMENT----------------

Runtime: Oracle Corporation Java HotSpot(TM) 64-Bit Server VM 25.25-b02



----ARGUMENTS------------------

algorithm: PBEWITHHMACSHA1ANDAES_128
input: sampleDbPassword
password: samplePasswordEncPassword



----OUTPUT---------------------

dBhdXawEhtCITbdZW+tDPPTtKlbS6m4ZJdLe3RMiUabL7RZW17nK/zcwXBw9u5W8
```

jdbc.pw

=

samplePassword

jdbc.pw

=

ENC(s6eA5w9OtgVM0vZ1hjKH/w1ndwJiGMr0)

# Provide the encryption password at server start

# Encrypt all sensitive data with safe algorithms

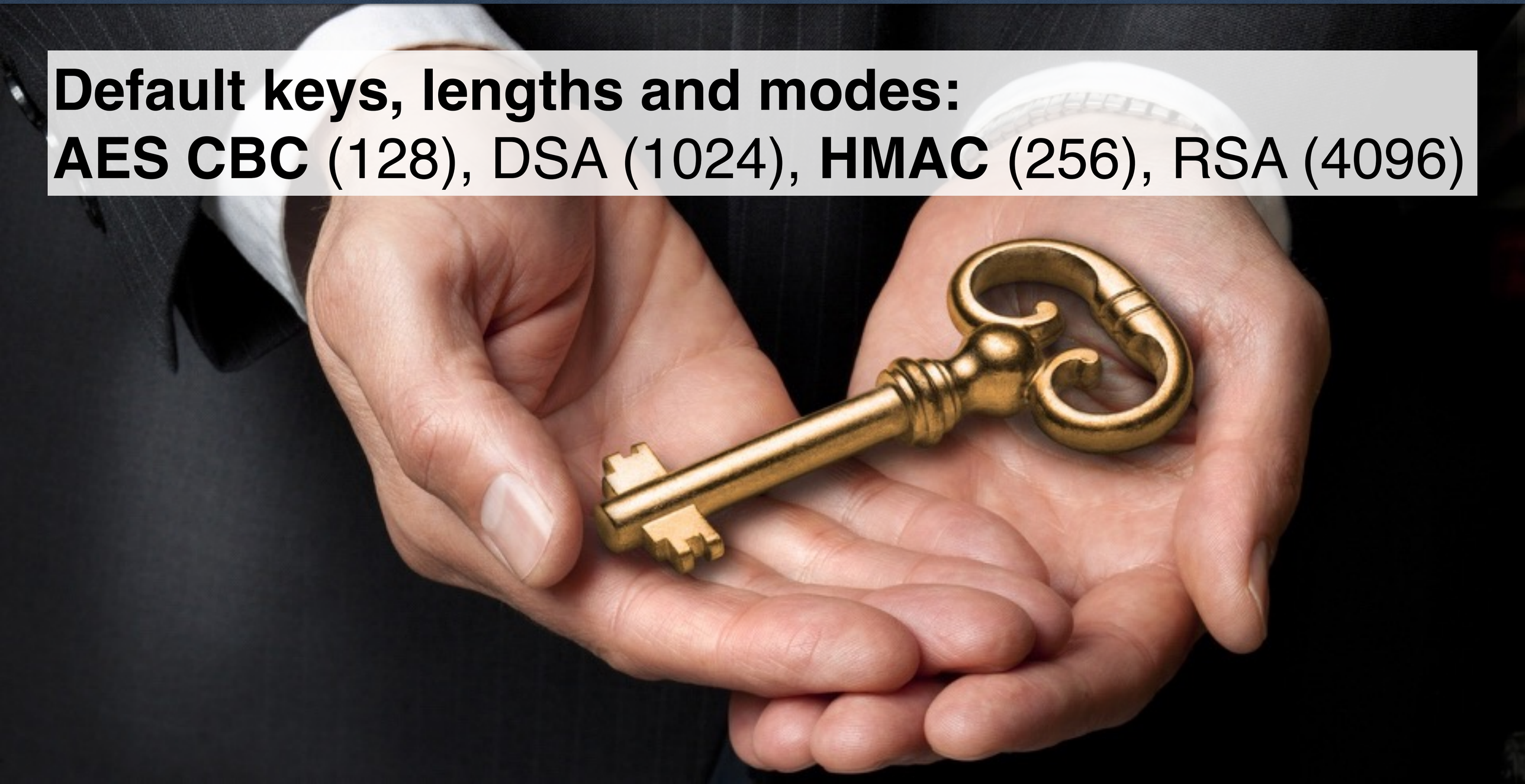
**AES Modes**
ECB, CBC, CFB,
OFB, CTR, ...

Which algorithm, mode, padding, ...?

# Swap all keys regularly like any other password

# Keyczar

**Default keys, lengths and modes:**
**AES CBC** (128), DSA (1024), **HMAC** (256), RSA (4096)



# One key per purpose: encrypt/decrypt & sign/verify

dos — bash — 180×58

```
Usage: "KeyczarTool command flags"
Commands: create addkey pubkey promote demote revoke usekey importkey exportkey
Flags: location name size status purpose padding destination version asymmetric crypter pemfile passphrase
Command Usage:
create --location=/path/to/keys --purpose=(crypt|sign) [--name="A name"] [--asymmetric=(dsa|rsa|ec)]
        Creates a new, empty key set in the given location.
        This key set must have a purpose of either "crypt" or "sign"
        and may optionally be given a name. The optional version
        flag will generate a public key set of the given algorithm.
        The "dsa" and "ec" asymmetric values are valid only for sets
        with "sign" purpose.

addkey --location=/path/to/keys [--status=(active|primary)] [--size=size] [--crypter=crypterLocation] [--padding=(OAEP|PKCS)]
        Adds a new key to an existing key set. Optionally
        specify a status, which is active by default. Optionally
        specify a key size in bits. Also optionally specify the
        location of a set of crypting keys, which will be used to
        encrypt this key set.  The optional --padding flag is allowed
        only for key sets created with --version=rsa.  If omitted, it
        defaults to OAEP.
pubkey --location=/path/to/keys --destination=/destination
        Extracts public keys from a given key set and writes them
        to the destination. The "pubkey" command Only works for
        key sets that were created with the "--asymmetric" flag.
promote --location=/path/to/keys --version=versionNumber
        Promotes the status of the given key version in the given
        location. Active keys are promoted to primary (which demotes
        any existing primary key to active). Inactive keys are
        promoted to be active.
demote --location=/path/to/keys --version=versionNumber
        Demotes the status of the given key version in the given
        location. Primary keys are demoted to active. Active keys
        are made inactive.
revoke --location=/path/to/keys --version=versionNumber
        Revokes the key of the given version number.
        This key must have been demoted to inactive to be revoked.
        WARNING: The key will be destroyed.
usekey --location=/path/to/keys [--destination=/path/to/output] [--crypter=crypterLocation] [message]
        Encrypt or sign the message using the primary key of the key set. If the key set is
        encrypted, specify the decryption key with --crypter. If [message]
        is not specified, the message will be read from stdin.
        If --destination is specified, the output will be written to the file
        at that location; otherwise, it will be written to stdout.
importkey --location=/path/to/keys --pemfile=/path/to/pemfile [--padding=(OAEP|PKCS)] [--passphrase=passphrase]
        Imports a key into an existing keyset, reading it from an
        existing PEM or DER-formatted X.509 certificate file or PKCS#8 file.
        The optional --padding flag is only valid for certificates containing
```

# Keyczar uses JSON for key sets and keys

No direct support in Java KeyStore

Import PKCS and X.509 certificates

# Demo

# Vulns in widespread libs affect many applications

# Always keep your dependencies up-to-date

```
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
```

# 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: 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

```
<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

# Summary

Don’t reinvent the wheel, use existing libraries

Keep all libraries up-to-date

Hack your own applications

Enjoy secure programming

Königstraße 42
70173 Stuttgart

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

Blog blog.dominikschadow.de
Twitter @dschadow

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

**Demo Projects**
github.com/dschadow/JavaSecurity

**Microsoft Threat Modeling Tool**
www.microsoft.com/en-us/download/details.aspx?id=42518

**Recx Security Analyser**
chrome.google.com/webstore/detail/recx-security-analyser/ljafjhbjenhgcgnikniijchkngljgjda

**Firebug**
addons.mozilla.org/en-US/firefox/addon/firebug

**OWASP Zed Attack Proxy**
www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

**FoxyProxy**
addons.mozilla.org/de/firefox/addon/foxyproxy-standard

**OWASP Java Encoder**
www.owasp.org/index.php/OWASP_Java_Encoder_Project

**Jasypt**
www.jasypt.org

**Keyczar**
www.keyczar.org

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

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

**Pictures**
www.dreamstime.com