

7 security tools and libraries every developer should know about



JavaOne 2014

Dominik Schadow | [bridgingIT](#)

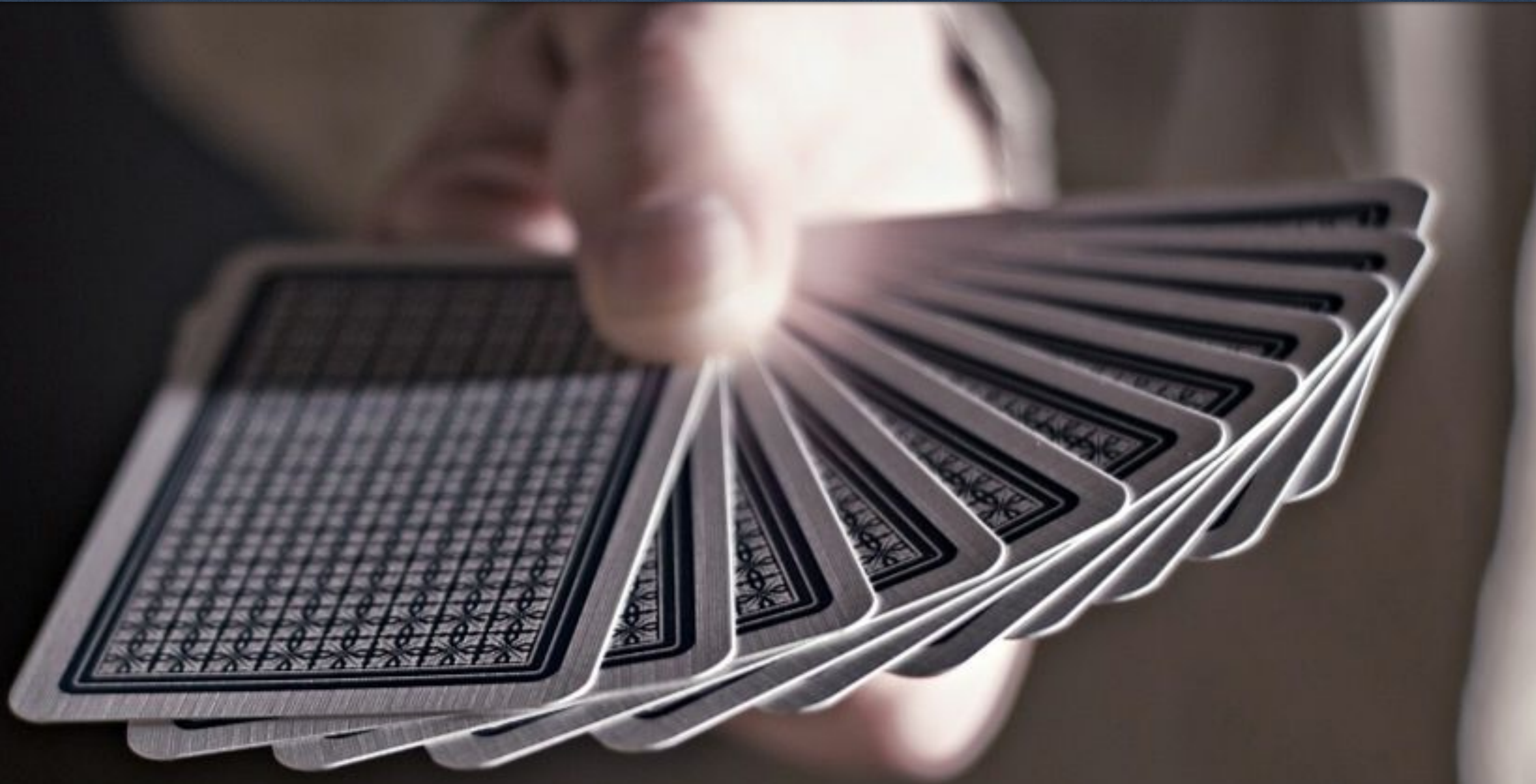
Security is a complex topic

**But required in any
(web) application**

**Complexity makes it
error prone**

**Reduce complexity
with libs and tools**

Spoilt for choice: frameworks and libraries



Which is the best one to solve my problem?

Security has to be tested like any other functionality

**How can I test the
security of my (web)
application?**





My seven tools and libs

Microsoft Threat Modeling Tool

OWASP Java Encoder

OWASP ZAP

Recx Security Analyser

Jasypt


Keyczar



OWASP Dependency Check

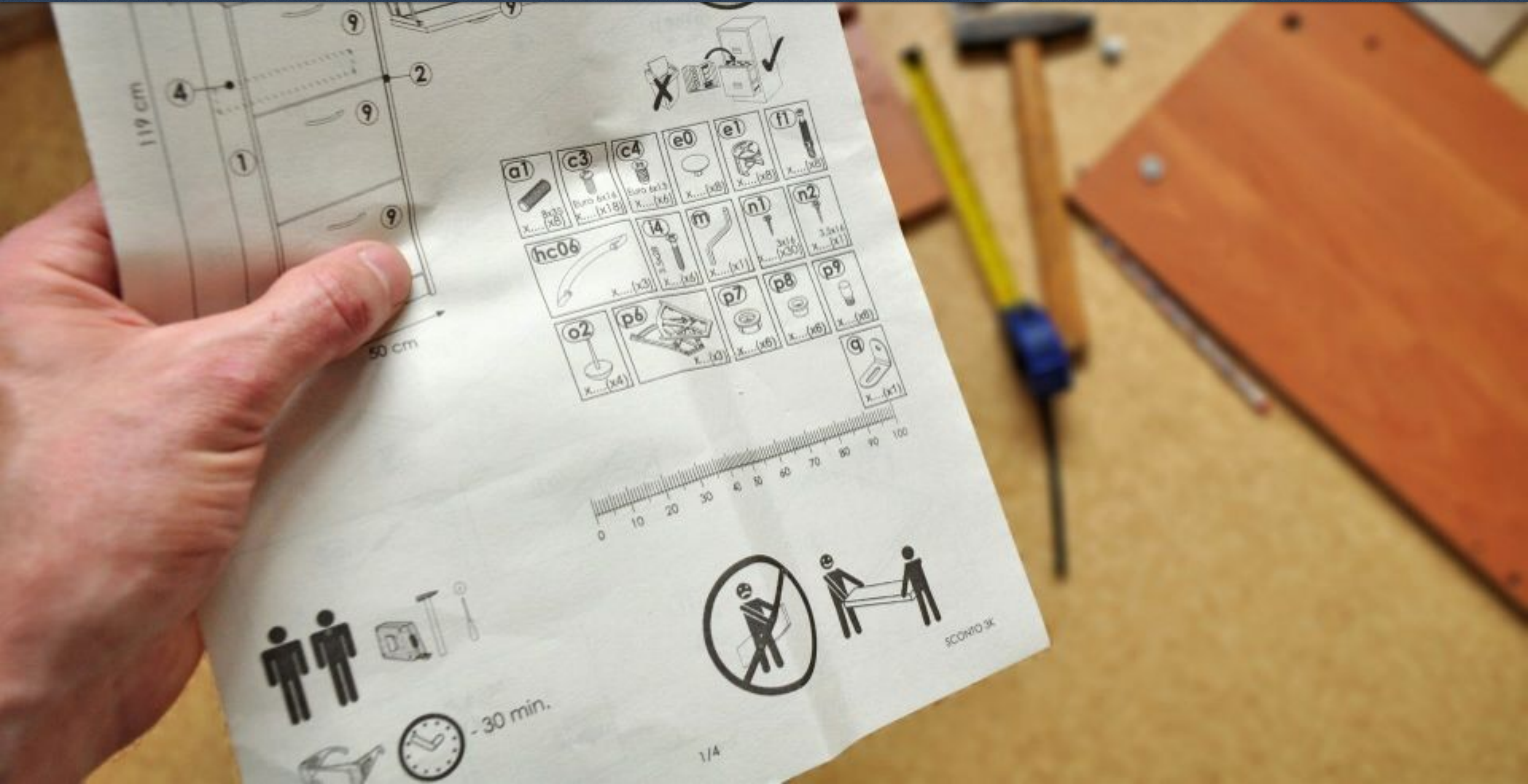
What about the well-known all-rounder frameworks?

Spring Security
Apache Shiro

A close-up photograph of a person's open palm, facing upwards. The skin is light-toned. On the palm, the text "PIN:" is written in dark ink above the number "1234". The background is a blurred blue and white pattern, possibly a wall or a screen. A black watch strap is visible at the bottom left corner of the frame.

PIN:
1234

So you are developing a new web application?!



Identify all incoming and outgoing data



Microsoft Threat Modeling Tool 2014

The screenshot displays the Microsoft Threat Modeling Tool 2014 interface. The main window shows a threat model diagram for a "Web Application Sample". The diagram includes a "Browser Client" and a "Web Server" connected by "HTTPS Request" and "HTTPS Response" flows. The "Web Server" is connected to an "SQL Database" via "SQL Commands" and "Resultset" flows, and to a "File write" process. Boundaries are marked with dashed red lines: "Internet Boundary" around the Browser Client and Web Server, and "Database Boundary" around the SQL Database. The "Web Server" is highlighted with a thick black border.

On the right side, the "Threat List Filter" pane shows the following threat states:

- Threat States (71)
 - Not Started (67)
 - Mitigated (2)
 - Not Applicable (1)
 - Needs Investigation (1)

The "Properties" pane for the selected "Web Server" shows the following attributes:

- 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:

The "Threat Information" pane at the bottom displays a list of 71 threats. The first four are visible:

Threat	Category	Status	Severity
Potential Data Repudiation by Web Serv	Repudiation	Mitigated	High
Cross Site Scripting	Tampering	Mitigated	High
Spoofing the Browser Client Process	Spoofing	Needs Investigation	High
Elevation by Changing the Execution Flo	Elevation Of Privilege	N/A Not Applicable	High

Spoofing

Pretending to be somebody else

Tampering

Modifying data that should not be modifiable

Repudiation

Claiming someone didn't do something

Information Disclosure

Exposing information

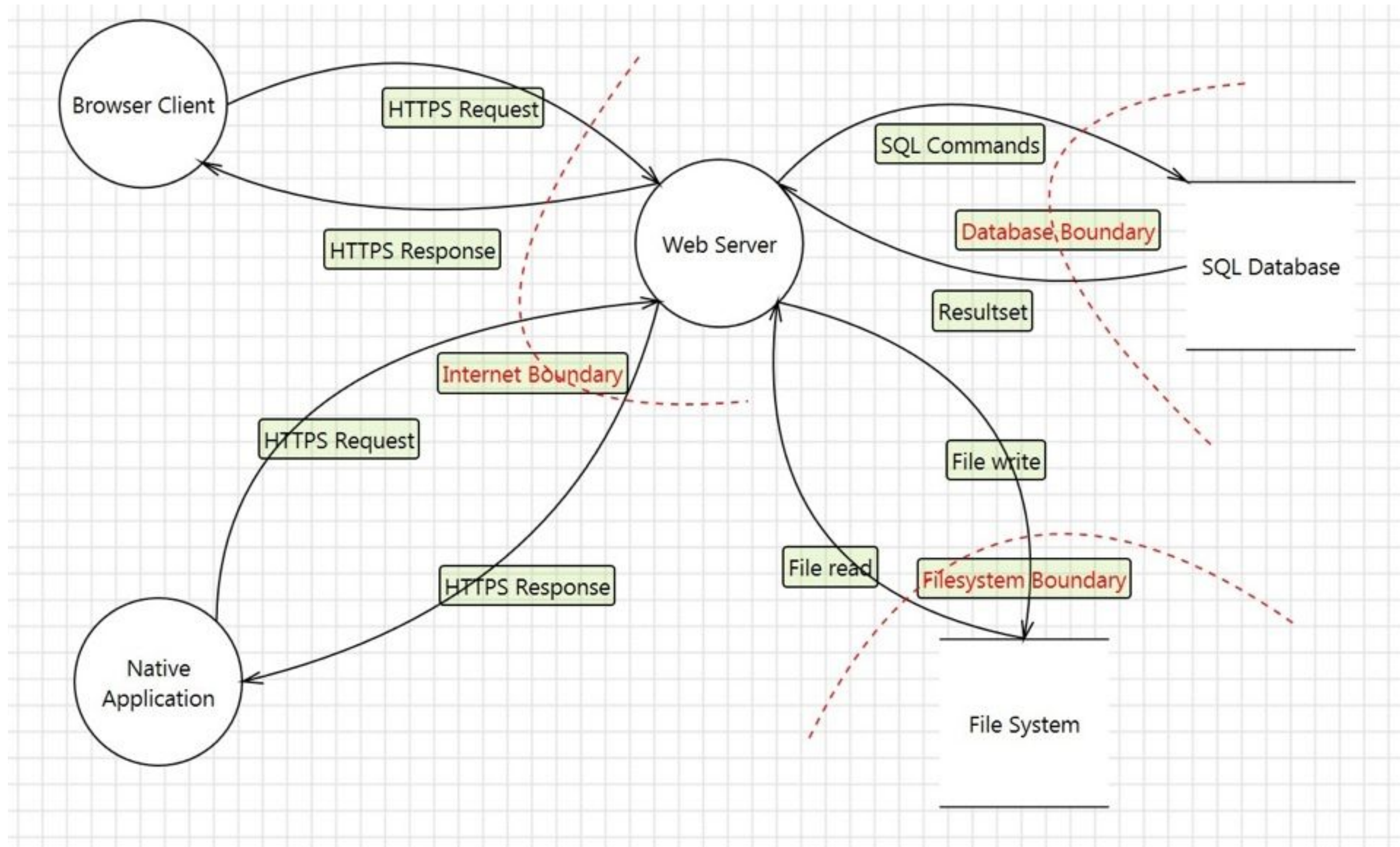
Denial of Service

Preventing a system from providing service

Elevation of Privilege

Doing things that one isn't supposed to do

Model your system and address the essential threats



Participating systems and their boundaries

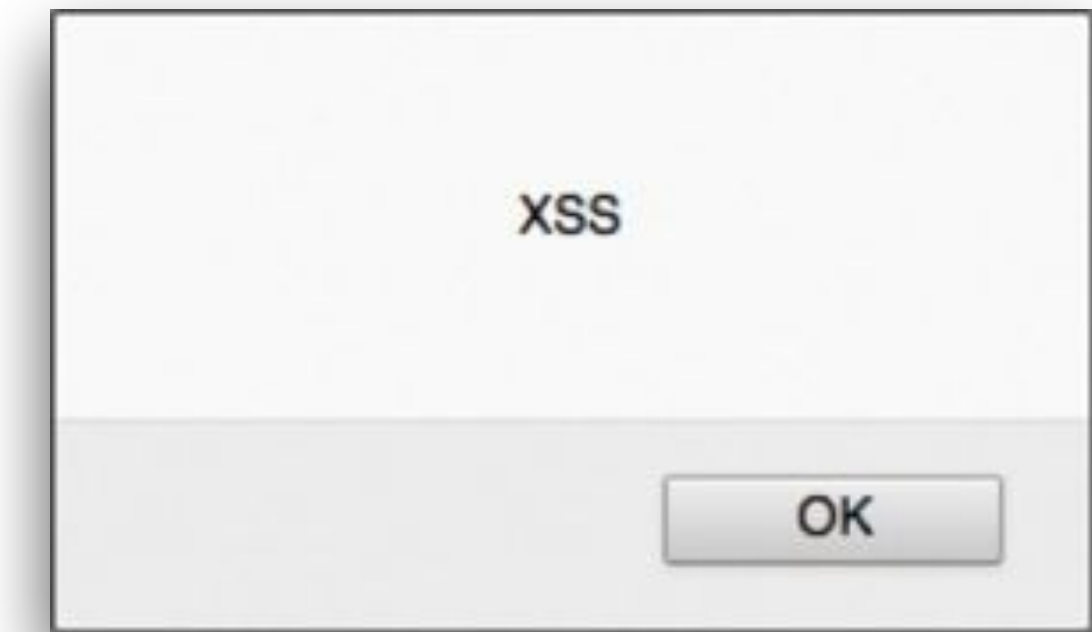
- ▣ Design and analysis view
- ▣ Security issue identification
- ▣ Issue tracker integration
- ▣ Report generation

Now you know all data entering and leaving your system



Escape all untrusted data for the proper context

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



For HTML

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

For CSS

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

For XML

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

OWASP Java Encoder

```
String encHTML = Encode.forHtml(name);
```

```
String encCSS = Encode.forCssString(name);
```

```
String encXML = Encode.forXml(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
```

Escape directly inside JavaServer Pages

```
<%@ 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!



Test your validation and escaping boundaries

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

Hack yourself first

Intercept and manipulate HTTP requests and responses



OWASP Zed Attack Proxy (ZAP) and FoxyProxy

The screenshot displays the OWASP ZAP interface. The top toolbar includes buttons for 'Standard mode', 'Schnellstart', 'Request', 'Response', 'Break', and 'Skripting-Konsole'. The left sidebar shows a tree view of sites, with 'http://localhost:8080' expanded to show 'XSS' and 'GET:styles.css'. The main pane shows a POST request to 'http://localhost:8080/XSS/javaEncoder.jsp HTTP/1.1'. The request headers are: Host: localhost:8080, User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:31.0) Gecko/20100101 Firefox/31.0, 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, Referer: http://localhost:8080/XSS/, Cookie: JSESSIONID=E571C442465BD0F22CED2F65668BD315, Connection: keep-alive, Content-Type: application/x-www-form-urlencoded, Content-Length: 10. Below the headers is a table of parameters:

Type	Parameter Name	Value	Functions
form	name1	<script>alert("XSS")</script>	Addins
form			Addins

The bottom toolbar includes 'History', 'Search', 'Break Points', 'Alerts', 'Active Scan', 'Spider', 'Forced Browse', 'Fuzzer', 'Params', and 'Http Sessions'. The 'History' pane shows a single entry: Id 1, Req. Timestamp 30/08/14 13:16:14, Method GET, URL http://localhost:8080/XSS/styles.css, Code 200 OK, Reason, RTT 43 ms, Size Resp. Body 111 bytes, Highest Alert Low. The bottom status bar shows 'Alerts 0 0 2 1' and 'Current Scans 0 0 0 0 0 0'.

Intercepting proxy

- ▣ Active and passive scanners
- ▣ Forced Browsing
- ▣ Fuzzer
- ▣ Spider

- ▣ HTTP Session Management

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



Tell the browser how to secure your web application

Cache-Control

X-Content-Type-Options

X-Frame-Options

HTTP Strict Transport Security

Content Security Policy

Verify headers manually with Firebug

The screenshot shows the Firebug Network tab with the following details:

URL	Status	Domain	Größe	Remote-IP	Zeitlinie
GET reporting.jsp?	200 OK	localhost:8080	283 B	:::1):8080	Sms

Below the table, the 'Header' tab is selected, showing the response headers:

```
Antwort-Header Quelltext anzeigen
Content-Length 283
Content-Type text/html;charset=UTF-8
Date Thu, 04 Sep 2014 13:15:47 GMT
Server Apache-Coyote/1.1
Strict-Transport-Security max-age=31556926; includeSubDomains
content-security-policy-r... default-src 'self'; report-uri CSPReporting
```

The last two lines of the header list are highlighted with a red dashed box.

Recx Security Analyser



https://localhost:8443/security-header/



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

HTTP security headers

Name	Value	Setting secure
content-security-policy	default-src 'self'	✓
x-content-type-options	nosniff	✓
x-frame-options	deny	✓
strict-transport-security	max-age=31556926; includesubdomains	✓
cache-control	no-store, no-cache, max-age=0, must-revalidate	✓
x-xss-protection	1; mode=block	✓
access-control-allow-origin	Header not returned	✓
		+

DEMO

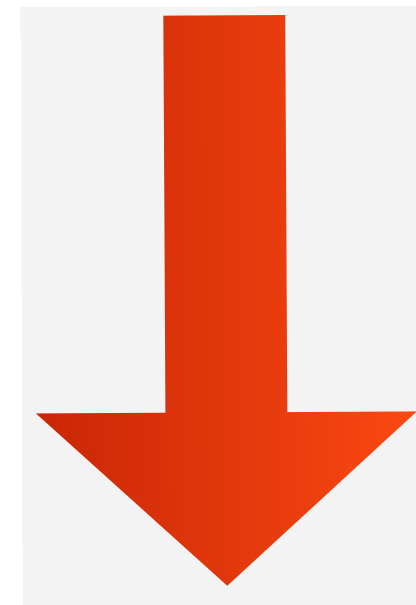
Keep your configuration passwords confidential

Spring, Hibernate, ...



Typical DataSource configuration in a Spring XML file

```
<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
```

Jasypt

```
bin — bash — 120x32

marvin:bin dos$ ./listAlgorithms.sh

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

PBE ALGORITHMS:    [PBEWITHHMACSHA1ANDAES_128, PBEWITHHMACSHA1ANDAES_256, PBEWITHHMACSHA224ANDAES_128, PBEWITHHMACSHA2
24ANDAES_256, PBEWITHHMACSHA256ANDAES_128, PBEWITHHMACSHA256ANDAES_256, PBEWITHHMACSHA384ANDAES_128, PBEWITHHMACSHA384AN
DAES_256, PBEWITHHMACSHA512ANDAES_128, PBEWITHHMACSHA512ANDAES_256, PBEWITHMD5ANDDES, PBEWITHMD5ANDTRIPLEDES, PBEWITHSHA
1ANDDESEDE, PBEWITHSHA1ANDRC2_128, PBEWITHSHA1ANDRC2_40, PBEWITHSHA1ANDRC4_128, PBEWITHSHA1ANDRC4_40]

marvin:bin dos$ ./encrypt.sh input="samplePassword" password="spring-jasypt" algorithm="PBEWITHSHA1ANDDESEDE"

----ENVIRONMENT-----

Runtime: Oracle Corporation Java HotSpot(TM) 64-Bit Server VM 25.5-b02

----ARGUMENTS-----

algorithm: PBEWITHSHA1ANDDESEDE
input: samplePassword
password: spring-jasypt

----OUTPUT-----

s6eA5w90tgVM0vZ1hjKH/w1ndwJiGMr0

marvin:bin dos$
```

Replace plaintext password with encrypted password

`jdbc.pw`
=

`ENC (s6eA5w90tgVM0vZ1hjKH/w1ndwJiGMr0)`

Provide the encryption password at server start



**Environment variable
Servlet launched after startup**

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



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



KeyczarTool for command line key management

```
symmetric — bash — 120x36
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.
```

Keyczar uses JSON for key sets and keys

**No direct support
in Java KeyStore**



DEMO

Vulnerabilities in widespread libs affect many applications



Always keep your dependencies up-to-date



OWASP Dependency Check

```
XSS — bash — 120x30
Last login: Tue Sep 23 20:37:45 on ttys000
marvin:XSS dos$ dependency-check.sh -a XSS -s target/dependency/
Sep 23, 2014 8:39:05 PM org.owasp.dependencycheck.data.update.task.DownloadTask call
INFORMATION: Download Started for NVD CVE - Modified
Sep 23, 2014 8:39:16 PM org.owasp.dependencycheck.data.update.task.DownloadTask call
INFORMATION: Download Complete for NVD CVE - Modified
Sep 23, 2014 8:39:16 PM org.owasp.dependencycheck.data.update.task.ProcessTask processFiles
INFORMATION: Processing Started for NVD CVE - Modified
Sep 23, 2014 8:39:18 PM org.owasp.dependencycheck.data.update.task.ProcessTask processFiles
INFORMATION: Processing Complete for NVD CVE - Modified
Sep 23, 2014 8:39:18 PM org.owasp.dependencycheck.data.update.StandardUpdate update
INFORMATION: Begin database maintenance.
Sep 23, 2014 8:39:25 PM org.owasp.dependencycheck.data.update.StandardUpdate update
INFORMATION: End database maintenance.
Sep 23, 2014 8:39:29 PM org.owasp.dependencycheck.Engine analyzeDependencies
INFORMATION: Analysis Starting
Sep 23, 2014 8:39:38 PM org.owasp.dependencycheck.Engine analyzeDependencies
INFORMATION: Analysis Complete
marvin:XSS dos$
```

dependency-check.sh -a [NAME] -s [JAR-DIR]

Dependency-Check Report

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: CSRF-spring-security

Scan Information ([show all](#)):

- *dependency-check version: 1.2.4*
- *Report Generated On: Sep 1, 2014 at 14:25:17 MESZ*
- *Dependencies Scanned: 14*
- *Vulnerable Dependencies: 1*
- *Vulnerabilities Found: 3*
- *Vulnerabilities Suppressed: 0*
- ...

Display: [Showing Vulnerable Dependencies](#)

Dependency	# Related	CPE	GAV	CVE Impact	CVE Count	CPE Confidence	Evidence Count
spring-security-core-3.2.4.RELEASE.jar	1	cpe:/a:springsource:spring_framework:3.2.4 cpe:/a:vmware:springsource_spring_framework:3.2.4 cpe:/a:vmware:springsource_spring_security:3.2.4		Medium	3	HIGHEST	13

Dependencies

spring-security-core-3.2.4.RELEASE.jar

File Path: target/dependency/spring-security-core-3.2.4.RELEASE.jar

Use Dependency Check as Jenkins plug-in

Post Steps

Run only if build succeeds Run only if build succeeds or is unstable Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Invoke OWASP Dependency-Check analysis

Path to scan	ear/target/ear-1.0.0/lib, engine/target/engine-1.0.0/WEB-INF/lib, web/target/web-1.0.0/WEB-INF/lib
Output directory	
Data directory	/jenkins_workspace/dependency-check-data
Suppression file	
ZIP extensions	
Disable CPE auto-update	<input type="checkbox"/>
Enable verbose logging	<input type="checkbox"/>
Generate optional HTML reports	<input type="checkbox"/>
Skip if triggered by SCM changes	<input type="checkbox"/>
Skip if triggered by upstream changes	<input type="checkbox"/>

Dependency-Check Warnings

Details

Details

richfaces-core-api-4.3.5.Final.jar , CVE-2014-0086 , Severity: Medium

The doFilter function in webapp/PushHandlerFilter.java in JBoss RichFaces 4.3.4, 4.3.5, and 5.x allows remote attackers to cause a denial of service (memory consumption and out-of-memory error) via a large number of malformed atmosphere push requests.

CVSSv2 Score: 4.3

NIST Reference: [CVE-2014-0086](#)

Category: CWE-20 Improper Input Validation

References:

Source: CONFIRM

Name: https://bugzilla.redhat.com/show_bug.cgi?id=1067268

URL: https://bugzilla.redhat.com/show_bug.cgi?id=1067268

Source: CONFIRM

Name: <https://github.com/pslegr/core-1/commit/8131f15003f5bec73d475d2b724472e4b87d0757>

URL: <https://github.com/pslegr/core-1/commit/8131f15003f5bec73d475d2b724472e4b87d0757>

Source: CONFIRM

Name: <https://issues.jboss.org/browse/RF-13250>

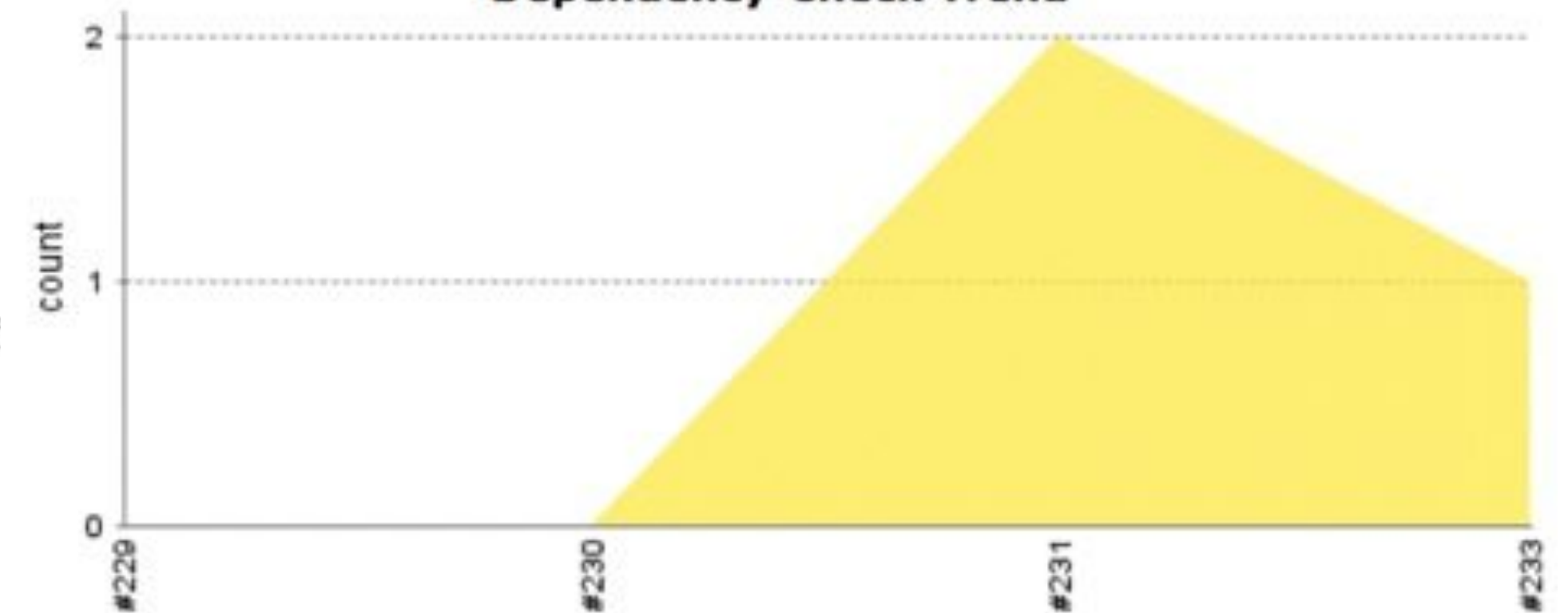
URL: <https://issues.jboss.org/browse/RF-13250>

Source: SECUNIA

Name: 57053

URL: <http://secunia.com/advisories/57053>

Dependency-Check Trend



Known vulnerabilities check (publicly disclosed)

- ▣ Automatic update of vulnerability information
- ▣ Execution via
 - ▣ Command line
 - ▣ Ant task
 - ▣ Maven plug-in
 - ▣ Jenkins plug-in

Restrict yourself on (security) library usage

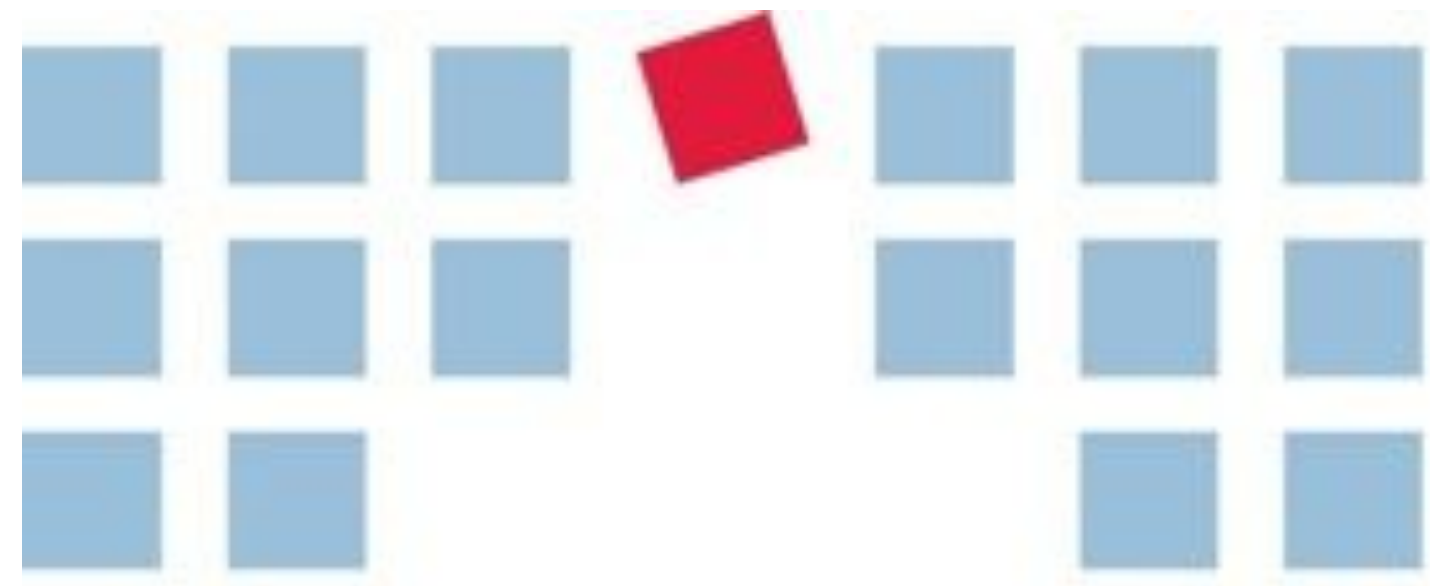
**Don't add overlapping
functionality**

A photograph of a long wooden pier extending from the bottom right towards the center of the frame. The pier is made of weathered wooden planks and leads into a vast, misty sea. The sky is filled with soft, grey clouds, and the overall atmosphere is serene and somewhat somber. The text is overlaid on the left side of the image.

- ❑ Don't reinvent the wheel, use existing libraries
- ❑ Keep libs up-to-date
- ❑ Test and hack your own applications

Developers make
the difference





bridging IT

BridgingIT GmbH

Koenigstr. 42

70173 Stuttgart/ Germany

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/download/details.aspx?id=42518

Recx Security Analyser

chrome.google.com/webstore/detail/recx-security-analyser/ljafjhbjenhgcniknijchknlgjda

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

Apache Shiro

shiro.apache.org

Pictures

www.dreamstime.com

