Java Security Myths

DOAG, 21.11.2013

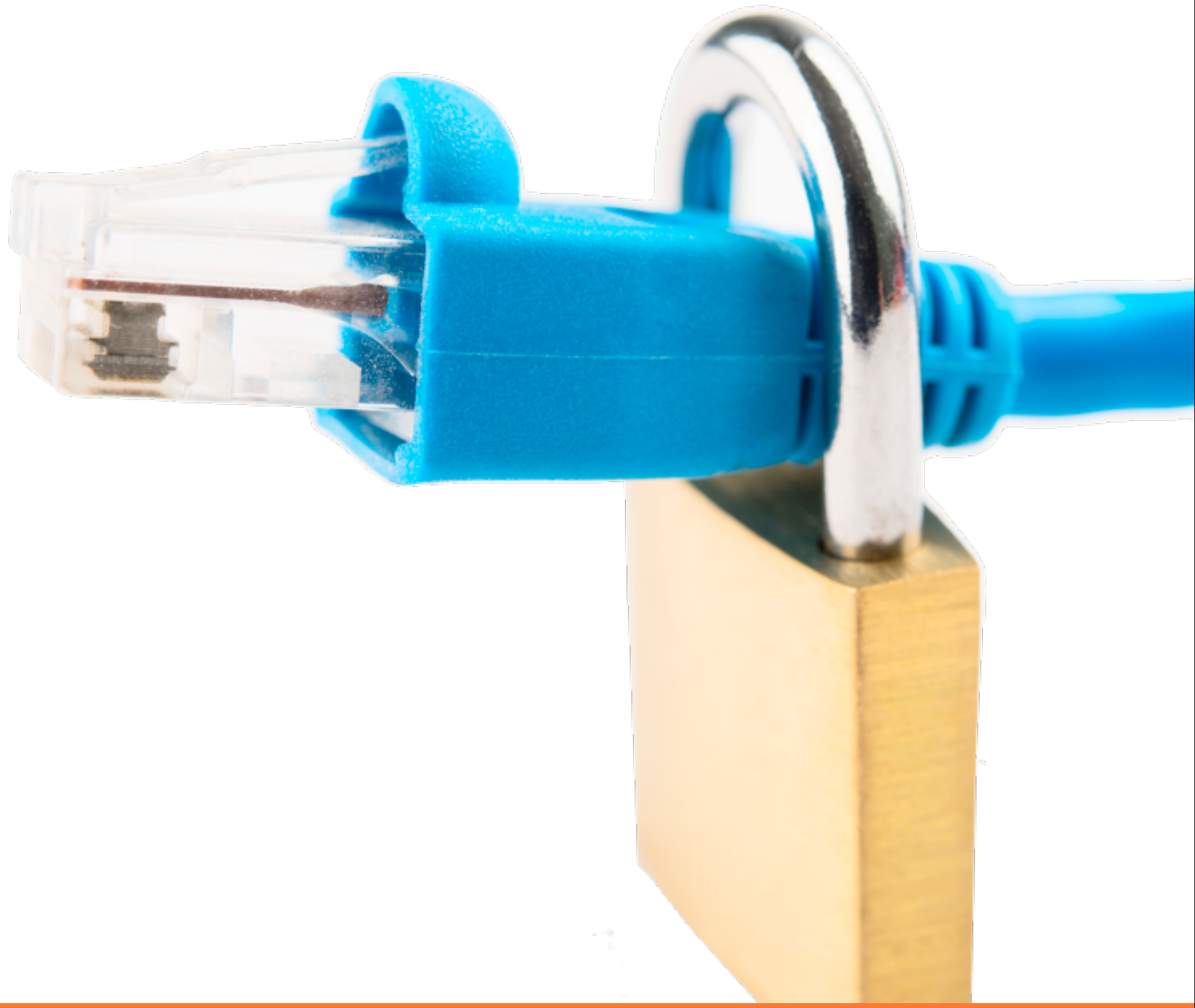
Dominik Schadow
BridgingIT GmbH

Secure Communication

Cross-Site Scripting

Cross-Site Request Forgery

Secure Communication



HTTPS after log-in form is enough

Login form manipulation and interception

HTTP Strict Transport Security Header (HSTS)

```
protected void doPost(HttpServletRequest request,
  HttpServletResponse response) {
  response.setHeader("Strict-Transport-Security",
    "max-age=2592000; includeSubDomains");
```

The HSTS policy requires browser support

Cross-Site Scripting (XSS)

Stored, **Reflected** and DOM Based XSS

Input **validate**, output **escape**

Java Server Faces automatically **escape** all output

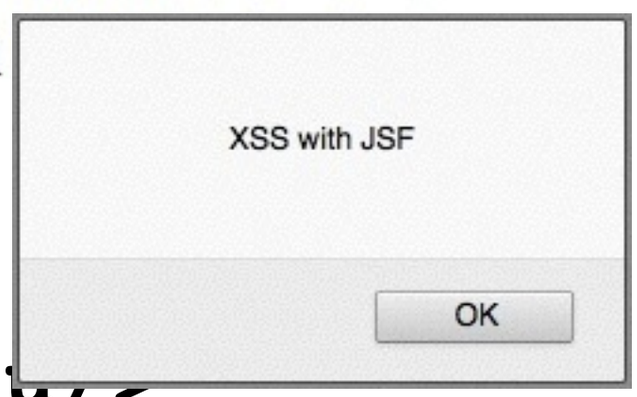
outputText to display user input is obvious, but…

<script>alert('XSS with JSF')</script>

Submit

<h:outputText/>

<script>alert('XSS with



<h:selectOneMenu/>

<script>alert('XSS with JSF')</script>

XSS in action

Don‘t take framework security for granted

Don‘t regard your code and libraries as separate units

Update third party libraries as soon as possible

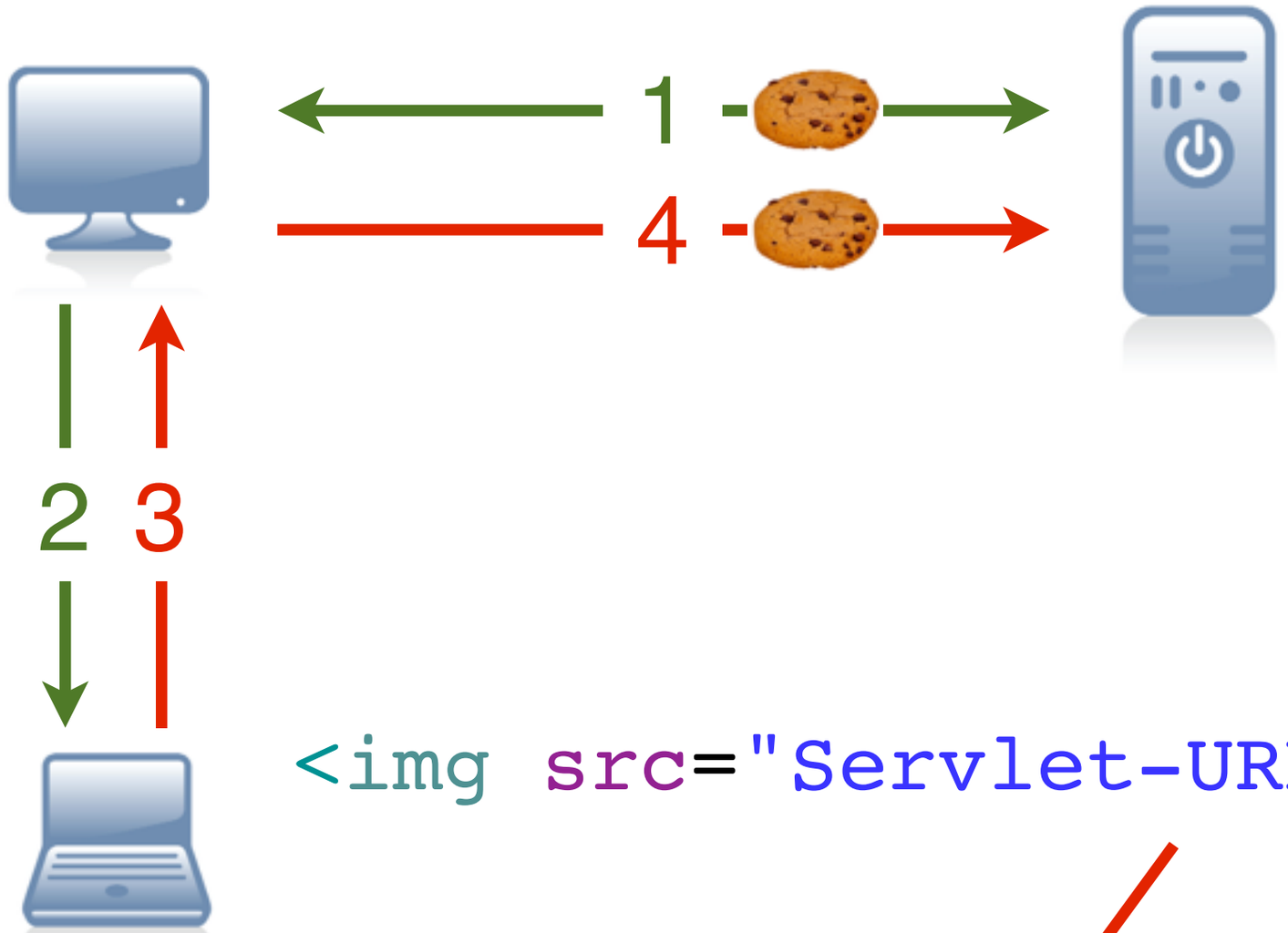
Use output escaping libraries with security focus

Enterprise Security API * Coverity Security Library

Cross-Site Request Forgery (CSRF)

Authorized user executes foisted request

POST forms protect from **CSRF**



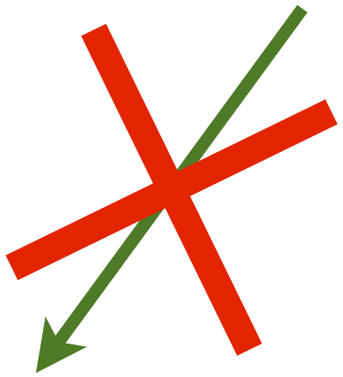

```
<img src="Servlet-URL?name=ATTACK">
```

```
protected void doGet(HttpServletRequest
request, HttpServletResponse response)
```

Replace GET with **POST requests**

```
<form method="post" action="MyServlet">
```

```
<img src="Servlet-URL?name=ATTACK">
```



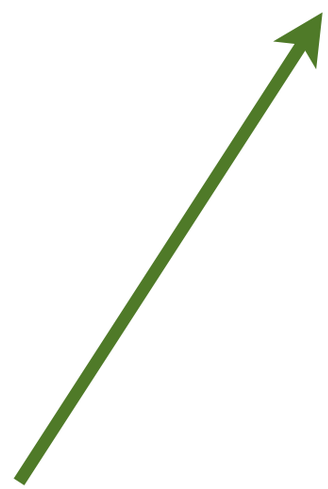
```
protected void doPost(HttpServletRequest
request, HttpServletResponse response)
```

Add more dynamic with **XMLHttpRequest**

JavaScript may submit POST forms

```
<form method="post" action="MyServlet">
```

```
protected void doPost(HttpServletRequest
request, HttpServletResponse response)
```



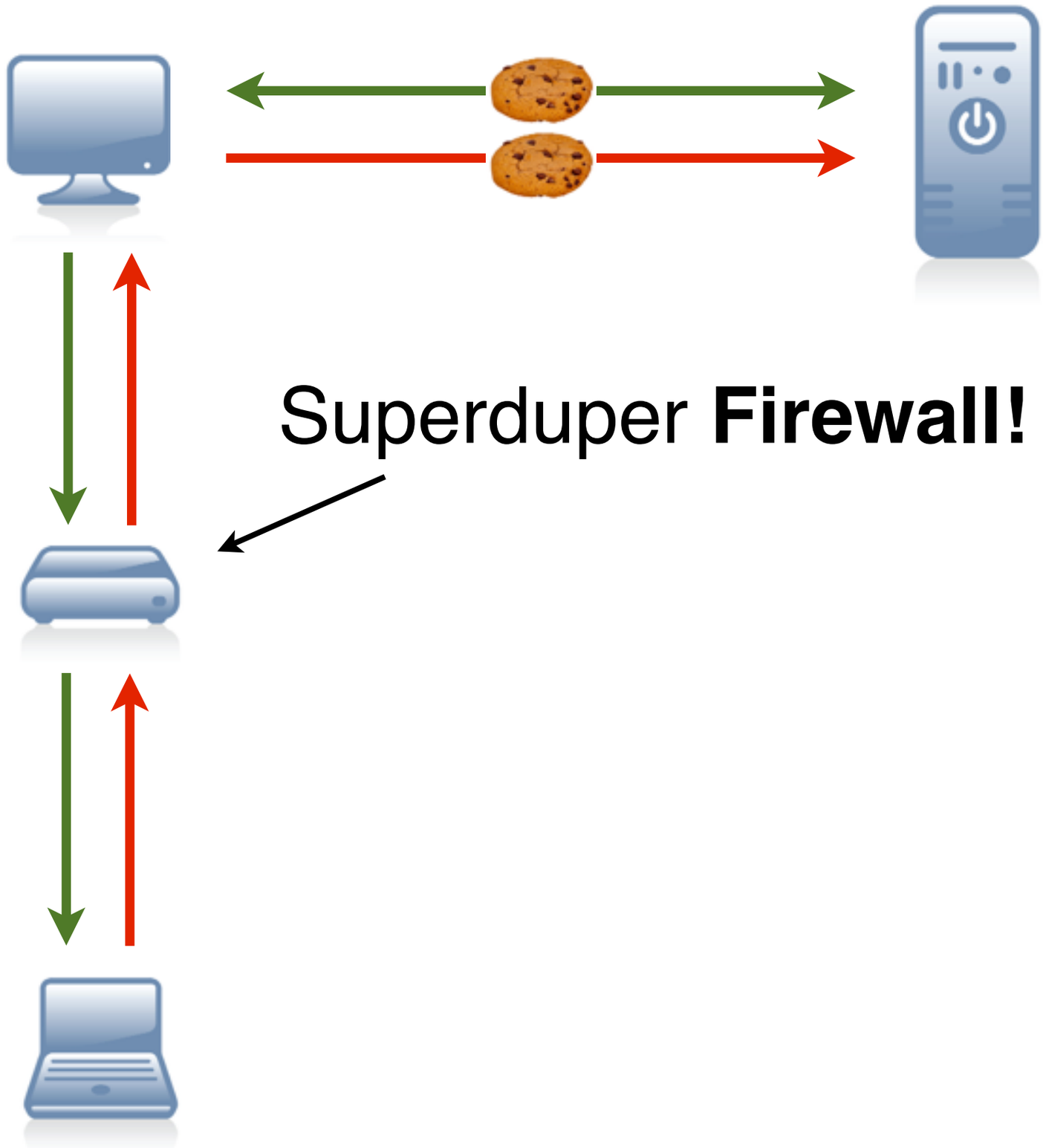
```
var request = new XMLHttpRequest();
request.open("POST", "Servlet-URL");
request.send("name=ATTACK");
```

Add hidden anti CSRF forgery token to each form

CSRF in action

Intranet applications are safe

CSRF attacks don‘t stop at firewalls



Basic security level is always required

Web application security is the developer's job

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
https://github.com/dschadow/JavaSecurityMyths

Enterprise Security API
https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API

Coverity Security Library
https://github.com/coverity/coverity-security-library

JSF Escaping Bug
https://java.net/jira/browse/JAVASERVERFACES-2747

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

Mozilla Firebug
https://addons.mozilla.org/en-US/firefox/addon/firebug