Drag and Drop XSS in Firefox by HTML5 (Cross Domain in frames)

Drag and Drop XSS in Firefox by HTML5 (Cross Domain in frames) - Soroush Dalili, soroush.me.

- Published: date not stated
- Original: <https://soroush.me/blog/drag-and-drop-xss-in-firefox-by-html5-cross-domain-in-frames>
- Preserved from: https://soroush.me/blog/drag-and-drop-xss-in-firefox-by-html5-cross-domain-in-frames (stored) on 2026-08-11
- 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.

Drag and Drop XSS in Firefox by HTML5 (Cross Domain in frames)

First of all, this vulnerability and the related techniques have already been reported to Mozilla on 21st Nov 2011, without having any specific result till the date of this report (issue ID 704354 – works on all the latest versions which support HTML5). I had raised this bug as a major issue, but it seems it was not important from Mozilla Firefox point of view and its risk is not high at all.

However, **NoScript can protect the users against it from version 2.2.3** [released about three weeks ago] (http://noscript.net/changelog) – thanks to Giorgio Maone for the fast response and quick fix.

As there is already a solution for this issue and its impact is not high, I am going to publish my research results as they belong to 2011!

As you may have noticed, most of the modern browsers are recently protecting their users from running unwanted JavaScript by copying and pasting it in the address bar or even by dragging and dropping it into a web page. In this research, I have found a technique to bypass Drag/Drop protection in Mozilla Firefox to run a JavaScript. As a final result, it is possible to drag and drop a hidden JavaScript into a predefined HTML5 box and run the Javascript code. Unfortunately, if you put this page in an IFrame, the Javascript code can be run on the context of the main site that includes the IFrame. For instance, When Facebook opens any URL in a frame, it is possible to run a JavaScript code on Facebook website by drag and drop jacking.

In order to understand the Mozilla Firefox protection against JavaScript Drag and Drop, follow these steps:

1- Go to Mozilla Firefox address bar and type “javascript:alert(1)” without pressing Enter.

2- Select all the string that you have just typed (“javascript:alert(1)” without quote signs).

3- Drag and drop it on a new tab or on the context of the same tab that you currently have. You will not receive any alert message.

Now, in previous steps, capitalize one or more letters in the “javascript:” string (for instance “jAvAscript:”) and drag/drop it into the page. You should be able to see an alert message as you have bypassed the Mozilla Firefox protection!

I have also found another interesting protocol in Mozilla Firefox that can lead to running a JavaScript. This protocol can be used as follows to bypass the Mozilla Firefox prevention method:

“feed:javascript:alert(1)”

“feed:feed:feed:javascript:alert(1)”

“feed:javascript:javascript:feed:alert(1)”

“feed:feed:javascript:javascript:feed:alert(1)”

” feed:feed:feed:javascript:alert(1)”

In this step, I had to find a way to use the issue and exploit the system to prove that it can be an important security risk; however, there are two facts that made it a bit difficult:

1- There is no point if we cannot run the JS code on the context of another site.

2- We need the user interaction to d/d a JS code. And it is not easy to deceive the users to d/d a JavaScript code when it is visible.

The first problem has been solved by using HTML5 D/D functionality that I have found from the following URL: “http://html5demos.com/drag“; I found out, if I drag and drop the “feed:javascript:alert(1)” to the drop location, the JavaScript will run due to the redirection. And interestingly, if this drop location is inside an IFrame, the main page will be redirected and therefore we can conduct an XSS attack on the context of the main website.

The second problem was also solved by using a hidden “textarea” tag that I found during my tests! In Mozilla Firefox, if you select a text with a hidden textarea, all the texts in that hidden textarea will be selected as well.

I have created a proof of concept which can be found in the following link:

PoC: http://soroush.secproject.com/downloadable/demo/FF_DragDrop_XSSHost_simp.html

In this research, I was able to bypass Mozilla Firefox – Javascript Drag and Drop by using capitalization and Feed protocol. Then I was able to exploit this issue to run a JavaScript code in the context of another website which can accept an external frame by using the HTML5 drag and drop functionality.

It is still possible to bypass Mozilla Firefox prevention method by finding another protocol or maybe by using the encoding techniques.

If someone drags and drops a JavaScript into a page with “chrome://” protocol, it can lead to a local code execution; however, this protocol is highly protected by Mozilla Firefox and I was not able to find a way to make it possible. As a PoC, drag and drop the following Javascript code into the “chrome://global/content/config.js” page to run the local Windows Calculator:

“feed:jAvAscript:file=Components.classes[‘@mozilla.org/file/local;1’].createInstance(Components.interfaces.nsILocalFile);file.initWithPath(‘c:\\windows\\system32\\calc.exe’);process=Components.classes[‘@mozilla.org/process/util;1’].createInstance(Components.interfaces.nsIProcess);process.init(file);process.run(true,[],0);void(0);”

This entry was posted in Security Posts

Creation date: December 31, 2011

Previous Sometimes no Ninja skill is required to receive money from security bug bounty programs![

Next

Flash ExternalInterface.call() JavaScript Injection – can make the websites vulnerable to XSS