IE9 Self-XSS Blackbox Protection bypass

IE9 Self-XSS Blackbox Protection bypass - Soroush Dalili, soroush.me.

- Published: date not stated
- Original: <https://soroush.me/blog/ie9-self-xss-blackbox-protection-bypass>
- Preserved from: https://soroush.me/blog/ie9-self-xss-blackbox-protection-bypass (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.

IE9 Self-XSS Blackbox Protection bypass

Introduction:

There is a defense-in-depth technique in IE9 that protects users against self XSS attacks which are growing very fast among social networking users (http://nakedsecurity.sophos.com/2010/02/02/anatomy-free-starbucks-gift-card-scam/ & https://www.facebook.com/video/video.php?v=956977232793).

IE9 protects users against copying and pasting a javascript or vbscript in URLs simply by detecting and removing the script protocols. For example, if you try to copy and paste “javascript:alert(1)” in the address bar, it will be converted to “alert(1)”. In the latest versions, it can also detect the script protocol if it starts with special characters such as Space Character (0x20), Control Characters (0x00-0x1F – not 0x00 and 0x7F), and Colon (0x3A) (Google chrome is currently vulnerable to this http://code.google.com/p/chromium/issues/detail?id=123213). As a result, even if you copy and paste the decoded equivalence of the following string, IE9 will remove the “javascript:” protocol:

However, IE9 still allows any other URL to be copied into the address bar.

Description:

I accidentally realised that there is a strange behaviour in IE9 and “file” protocol that can lead to execution of a Javascript/VBScript in URL (or browsing the file system). In order to replicate the issue, follow these steps:

1- Add a letter before file protocol (e.g. “Xfile:”), or maximum three letters after the “file” protocol (e.g. “fileXXX:”), or add one letter before and after the file protocol (e.g. “XfileX:”)

2- Now, add one or more space characters (or any other control characters) after the colon character (you can use URL-encoded values) (e.g. “XfileX:%20%0A%1F”)

3- Add the result to “javascript:Your Code Here” (e.g. “XfileX:%20%0A%1F javascript:Your Code Here”.

4- Open IE9, and go to facebook.com

5- Try to copy and paste the final string into the address bar and press enter. (e.g. “XfileX:%20%0A%1Fjavascript:alert(document.cookie)”)

6- You should be able to see your cookies.

Finally, two simple examples are:

I have also noticed that the file system can be browsed by the following vector (in different versions of IE):

It is almost the same as using “file:c:/” which is not a security issue on its own. However, this new vector can lead to file system access in kiosk devices that use IE and have blacklist filter on the address bar.

Ctrl+Shif+L (Go to copied address) in IE9 – Can be used in Self-XSS:

There is an interesting feature in IE9 that can be used to make the exploitation of this issue even easier by using social engineering techniques. An attacker needs to deceive the user to copy something into his/her clipboard and then encourage him to press “Ctrl+Shift+L”! This attack is feasible when you are able to control an IFrame inside the target website such as Facebook.

Note 1: This issue has already been reported to MS as a low issue (**msrc #12866**).

Note 2: This issue is **not** detectable by Shazzer.

This entry was posted in Security Posts

Creation date: August 14, 2012

Previous Don’t trust a string based on TryParse or IsNumeric result! (.Net/VBScript)[

Next

Microsoft IIS tilde character “~” Vulnerability/Feature – Short File/Folder Name Disclosure

---