

# [REVS] DOM Based Cross Site Scripting

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-08/0013.html>

---

**From:** SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

**Date:** 08/02/05

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 2 Aug 2005 18:16:11 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

-----

DOM Based Cross Site Scripting

---

## SUMMARY

We all know what Cross Site Scripting (XSS) is, right? It's that vulnerability wherein one sends malicious data (typically HTML stuff with Javascript code in it) that is echoed back later by the application in an HTML context of some sort, and the Javascript code gets executed.

Well, wrong. There is a kind of XSS that does not match this description, at least not in some of its fundamental properties. The XSS attacks described above are either non-persistent / reflected (i.e. the malicious data is embedded in the page that is returned to the browser immediately following the request) or persistent / stored (in which case the malicious data is returned at some later time).

But there s also a third kind of XSS attacks – the ones that do not rely on sending the malicious data to the server in the first place! While this seems almost contradictory to the definition or to common sense, there are, in fact, two well described examples for such attacks.

This technical note discusses the third kind of XSS, dubbed "DOM Based XSS". No claim is made to novelty in the attacks themselves, of course, but rather, the innovation in this write-up is about noticing that these belong to a different flavor, and that flavor is interesting and

important.

DETAILS

Application developers and owners need to understand DOM Based XSS, as it represents a threat to the web application, which has different preconditions than standard XSS. As such, there are many web applications on the Internet that are vulnerable to DOM Based XSS, yet when tested for (standard) XSS, are demonstrated to be "not vulnerable". Developers and site maintainers (and auditors) need to familiarize themselves with techniques to detect DOM Based XSS vulnerabilities, as well as with techniques to defend against them, both there which are different than the ones applicable for standard XSS.

The reader is assumed to possess basic knowledge of XSS ([1], [2], [3], [4], [8]). XSS is typically categorized into "non-persistent" and "persistent" ([3], "reflected" and "stored" accordingly, as defined in [4]). "Non-persistent" means that the malicious (Javascript) payload is echoed by the server in an immediate response to an HTTP request from the victim.

"Persistent" means that the payload is stored by the system, and may later be embedded by the vulnerable system in an HTML page provided to a victim. As mentioned in the summary, this categorization assumes that a fundamental property of XSS is having the malicious payload move from the browser to the server and back to the same (in non-persistent XSS) or any (in persistent XSS) browser. This paper points out that this is a misconception. While there are not many counterexamples in the wild, the mere existence of XSS attacks which do not rely on the payload embedded by the server in some response page, is of importance as it has a significant impact on detection and protection methods. This is discussed in the document.

To read more about the subject please visit:  
<<http://www.webappsec.org/projects/articles/071105.shtml>>  
<http://www.webappsec.org/projects/articles/071105.shtml>

ADDITIONAL INFORMATION

The information has been provided by <<mailto:contact@webappsec.org>>  
Robert Auger.

The article has been written by Amit Klein  
The original article can be found at:  
<<http://www.webappsec.org/projects/articles/071105.shtml>>  
<http://www.webappsec.org/projects/articles/071105.shtml>

=====

This bulletin is sent to members of the SecuriTeam mailing list.  
To unsubscribe from the list, send mail with an empty subject line and body to:  
[list-unsubscribe@securiteam.com](mailto:list-unsubscribe@securiteam.com)

Securiteam: [REVS] DOM Based Cross Site Scripting

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto:list-subscribe@securiteam.com)

=====  
=====

**DISCLAIMER:**

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.