

# [NT] Internet Explorer Method Cache Location Variant Trust Leads to Script Execution

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-07/0096.html>

---

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

**Date:** 07/27/04

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

Date: 27 Jul 2004 20:12:29 +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>

-----

Internet Explorer Method Cache Location Variant Trust Leads to Script Execution

---

## SUMMARY

Internet Explorer is "Microsoft's core browser that is a part of any Windows operating system and is the dominant browser currently in the world".

A condition exists in which Internet Explorer can be fooled to execute JavaScript code in a popped-up window. The vulnerability is based on method cache unwarranted trust when redirecting execution to a script function with the same name.

## DETAILS

Vulnerable Systems:

\* Internet Explorer version 6.0.2800.1106 (even with Windows XP SP1)

Vulnerable Components

\* MSHTML.DLL version 6.00.2800.1400

In an HTML page containing script that redirects a function to another

## Securiteam: [NT] Internet Explorer Method Cache Location Variant Trust Leads to Script Execution

function with the same name, the later can be called without security restrictions. An example of the issue can be easily demonstrated using the following script code:

```
<script>
var var1=location.assign;
alert("Assign function of the current window:\n"+var1);
var w=window.open("about:blank","_blank");
var var2=w.location.assign;
var w=alert("Assign function of the new window:\n"+var2);
w.close();
</script>
```

You should get two alerts describing the assign() function as being:

```
function assign(){
[Native code]
}
```

Notice how both functions appear to be same. Although it might seem that since their names are identical, IE marks the function assign() as safe. However, that is not enough as each assign() function has a unique ID that distinguishes between them, instead of just using their toString() methods. Instead, IE tries to determine whether the function call is safe based on the level of trust it has to the object that the method resides on.

It appears then that IE trusts objects in such a manner that would enable a complete range of method caching vulnerabilities by circumventing the object security check. This is possible because IE does not take into account cross-window function calls, which is the vessel used to circumvent the trust checks performed by IE. This can be demonstrated by creating a cached reference to the location.assign method from the first window on the second windows location object, not just on the location.assign method but also on the location.replace method and the non-existent location.whatever property. Thor Larholm's demo of this can be found at

<http://www.pivx.com/research/2004/7/PaulsimilarMethodNameRedirection/test2.html>  
<http://www.pivx.com/research/2004/7/PaulsimilarMethodNameRedirection/test2.html>.

As Thor Larholm pointed out, it isn't a problem of similar method name redirection, but a problem with the location variant itself. A new proof of concept HTML page was constructed and can be found at

<http://freehost07.websamba.com/greyhats/evilchild.htm>  
<http://freehost07.websamba.com/greyhats/evilchild.htm>.

EvilChild creates a child pop-up in a new window. It then redirects the page. As the page is loading, the pop-up is shown and saves the reference of parent.window.open to location.cache. As soon as the evil child pop-up cannot access the parent.document, an error handler is fired calling parent.window.open to load JavaScript into the main window.

## Securiteam: [NT] Internet Explorer Method Cache Location Variant Trust Leads to Script Execution

When JavaScript code can be executed by a malicious HTML page on the client side, the variety of attack possibilities is staggering.

### Workaround

Since the vulnerability can be used via any script function call, the only viable option for now is to completely disable scripting in Internet Explorer, or switch to another browser.

### ADDITIONAL INFORMATION

The information has been provided by <mailto:paul@greyhats.cjb.net> Paul.

=====

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

In order to subscribe to the mailing list, simply forward this email to: 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.