

[NT] Script Injection to Custom HTTP Errors in Local Zone

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-06/0066.html>

From: SecuriTeam (*support_at_securiteam.com*)

Date: 06/19/03

To: list@securiteam.com

Date: 19 Jun 2003 19:28:11 +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

Beyond Security in Canada

Toronto-based Sunrays Technologies is now Beyond Security's representative in Canada.

We welcome ISPs, system integrators and IT systems resellers

to promote the most advanced vulnerability assessment solutions today.

Contact us at 416-482-0038 or at canadasales@beyondsecurity.com

Script Injection to Custom HTTP Errors in Local Zone

SUMMARY

Internet Explorer ships with various internal HTML resource files. The majority of these files are meant to handle custom HTTP errors in web sites (also called "Friendly HTTP error messages"). They all use the same basic pieces of code, with minor changes to the actual content of each resource.

One of the main functions included in the resources is a method to extract the real URL from the resource URL hash. For example, if "site.com" generated a 404 HTTP error, the following URL will be internally requested by IE: `res://shdoclc.dll/404_HTTP.htm#http://site.com/file.html`.

The function takes the part after the # sign and attempts to extract the domain of the site, in order to embed it in the content of the custom message.

DETAILS

Securiteam: [NT] Script Injection to Custom HTTP Errors in Local Zone

Vulnerable systems:

* Microsoft Internet Explorer 5.01, 5.5 and 6.0.

Note that any other application that uses Internet Explorer's engine (WebBrowser control) is affected as well (AOL Browser, MSN Explorer, etc.).

GreyMagic found that the above-mentioned parsing procedure has a flaw in it that may cause arbitrary script commands to be executed in the Local Zone. Leading to potential arbitrary commands execution, local file reading and other severe consequences.

However, exploiting this procedure requires user-interaction. The user must click the URL presented to it by the resource for the malicious code to execute.

Here is the vulnerable function, precisely as it appears in the resources:

```
function Homepage(){
// in real bits, urls get returned to our script like this:
// res://shdocvw.dll/http_404.htm#http://www.DocURL.com/bar.htm

//For testing use DocURL =
"res://shdocvw.dll/http_404.htm#https://www.microsoft.com/bar.htm"
DocURL = document.location.href;

//this is where the http or https will be, as found by searching for
:// but skipping the res://
protocolIndex=DocURL.indexOf("://",4);

//this finds the ending slash for the domain server
serverIndex=DocURL.indexOf("/",protocolIndex + 3);

//for the href, we need a valid URL to the domain. We search for the #
symbol to find the begining
//of the true URL, and add 1 to skip it – this is the BeginURL value.
We use serverIndex as the end marker.
//urlresult=DocURL.substring(protocolIndex – 4,serverIndex);
BeginURL=DocURL.indexOf("#",1) + 1;
if (protocolIndex – BeginURL > 7)
urlresult=""

urlresult=DocURL.substring(BeginURL,serverIndex);

//for display, we need to skip after http://, and go to the next slash
displayresult=DocURL.substring(protocolIndex + 3 ,serverIndex);

// Security precaution: must filter out "urlResult" and
"displayresult"
forbiddenChars = new RegExp("[<>\\\"]", "g"); // Global search/replace
urlresult = urlresult.replace(forbiddenChars, "");
```

Securiteam: [NT] Script Injection to Custom HTTP Errors in Local Zone

```
displayresult = displayresult.replace(forbiddenChars, "");

document.write('<A target=_top HREF="' + urlresult + "'>' +
displayresult + "</a>");

}
```

The comments in this function teach us that Microsoft had indeed attempted to protect this resource from being exploited in this way, but unfortunately failed to do so. A specially crafted value appended after the # sign can fool this function to write a "javascript:" URL in the displayed link.

Exploit and Demonstation:

This URL will cause the resource to output a "javascript:" link to the document, which will execute when the user clicks on it:

```
res://shdoclc.dll/HTTP_501.htm#javascript:%2f*:*%2falert(location.href)/
```

Copy and paste the above URL in your browser, then click the red link in order to test it.

Solution:

Microsoft was notified on 20-Feb-2003. They were able to reproduce this on IE6 Gold and all versions below it. We managed to reproduce it on all versions, including IE6 SP1, with no exceptions.

They plan to fix this flaw in a future service pack.

ADDITIONAL INFORMATION

The original advisory can be downloaded from:
<<http://security.greymagic.com/adv/gm014-ie/>>
<http://security.greymagic.com/adv/gm014-ie/>

The information has been provided by <<mailto:security@greymagic.com>>
GreyMagic Software.

=====

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,

Securiteam: [NT] Script Injection to Custom HTTP Errors in Local Zone

loss of business profits or special damages.