

CSS visited pages disclosure

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2002-02/0271.html>

From: Andrew Clover (and@doxdesk.com)

Date: 02/20/02

Date: Wed, 20 Feb 2002 10:06:45 +0000
From: Andrew Clover <and@doxdesk.com>
To: bugtraq@securityfocus.com

Affected: web browsers with CSS support

Vendor: various

Risk: low

Background

=====

In <http://www.cs.princeton.edu/sip/pub/webtiming.pdf>, Felten and Schneider outline a method for pages on an attacking server to detect whether pages on another server have been visited, by trying to fetch a URL from the target server and using the time taken to fetch it to guess whether the URL was in the browser's local cache.

A method is also suggested to use the browser cache, read this way, as a store for persistent user data ("cache cookies").

CSS has a feature that can be abused to exactly the same ends. It is simpler, more accurate, and more easily abused than the timing attacks described in the above paper.

Issue

=====

The CSS `:visited` pseudo-class can be used to apply different on-screen styling to links leading to pages the user has already visited. However the styling can have side effects which can be detected by the attacking server. For example, the page at <http://www.smith-widgets.foo/> could include the following markup:

```
<a id="jones" href="http://www.jones-widgets.foo"></a>
```

with the style:

```
#jones:visited { background: url(/visited.cgi?site=jones); }
```

SecurityFocus Bugtraq: CSS visited pages disclosure

In this case the side-effect of the style will be a call to the CGI at smith-widgets if the user has visited jones-widgets. The script there could log this information, associate it with any cookies passed, then return a transparent background image set to expire soon.

Any property that can be given with a <uri> parameter could be abused this way. CSS2 defines background-image, list-style-image (trickier to use without fiddling with display properties or using CSS3 selectors, as a list cannot normally go inside a link), content and cursor (trickier to use due to poor browser support), and various Aural CSS properties (again, terrible browser support).

The simple answer to this problem would be to have all URIs associated with :visited conditions be fetched regardless of whether the link has been visited or not. However, apart from the performance penalty this would incur, it does not solve the problem for browsers with the capability to read calculated styles. JavaScript can then be used to detect other side-effects, if it is enabled.

IE gives each document element a 'currentStyle' object which can be queried to read which the calculated styles applied to that element, which can be used to determine whether a :visited rules was applied:

```
a { color: blue; }
a:visited { color: red; }
```

```
if (document.getElementById('jones').currentStyle.color=='red')
    document.writeln('<p>Hello! I see you've been to Jones.');
    document.writeln('Don\'t buy from Jones - their widgets');
    document.writeln('are made from recycled babies.</p>');
```

Mozilla's support of DOM Views should be able to do the same sort of thing. Even without direct access to calculated style objects, there are ways to imply which rules have been used, for example using the on-screen positions of elements:

```
#jones { position: absolute; top: 0; }
#jones:visited { top: 100px; }
```

```
if (document.getElementById('jones').offsetTop>50)
    ...
```

IE's offsetFoo properties are also supported by Mozilla, and, I believe, Konqueror.

The primitive one-bit-cache-storage "cache cookies" idea can also be used with one-bit-history-list-storage to get "CSS

cookies". To write to the history list would require an actual visit to the page, not just an attempt to load it; this could be achieved using an invisible frame. Mozilla also counts an <iframe> as being a visit, but IE does not. using :visited rules then gives non-destructive read capability to the history list. Many single bits (documents) would have to be used to store any practical amount of user data, which is presumably why 'cache cookies' have not been exploited so far (as far as I know).

Possible solutions to the problem would be:

- (a) as well as fetching all URIs independent of :visited conditions, removing all access to calculated styles and other run-time properties such as positioning. Unfortunately these features can be very useful to web authors! There is no practical way to limit access to elements unaffected by :visited styles.
- (b) as well as fetching :visited URIs, advising users to turn scripting off in non-trusted sites. This is probably a good idea in any case, but users never do it.
- (c) something similar to Felten and Schneider's proposed 'domain tagging' – make 'visited' links only look 'visited' when they point to documents in the same domain as the current page. This would be a severe blow to the functionality and usability expected of visited links.

Can anyone think of a better approach?

Vendor response

=====

This is a general problem with implementing CSS, not a browser bug, and not one with a simple fix. For this reason I have not contacted vendors.

IE and Mozilla are known to be vulnerable; Netscape 4 and Opera are probably not, as their layout algorithms seem to be incapable of applying properties with side-effects to pseudo-classes, and their object models do not allow access to calculated styles. However the next major version of Opera will probably be affected. It is expected some other browsers I have not tested (Konq?) will also be vulnerable.

--

Andrew Clover
mailto:and@doxdesk.com
<http://and.doxdesk.com/>

- *Previous message:* [Jonathan G. Lampe: "Whose X do I need to X to get on CERT?"](#)
- *Messages sorted by:* [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)