Feasibility and Real-world Implications of Web Browser History Detection

Artur Janc, Łukasz Olejnik
What the Internet Knows About You

W2SP 2010
Outline

Attacks on privacy using CSS :visited to inspect users’ Web browsing histories

1. Basics (quick) and history
2. Analysis
   - What can be detected, performance
   - Building a history detection system
3. Results
4. Current work / Countermeasures
How it Works

• CSS :visited, :link styling

• Browsers apply additional styles to links which the user had visited (requirement)

Attack:

• Insert a link with a URL to check for

• Check if visited style was applied (JS) or if a visited “marker” resource was downloaded
Examples

CSS

```html
<style>
#foo:visited {background: url(/?yes-foo);} 
#bar:link {background: url(/?no-bar);} 
</style>
<a id="foo" href="http://foo.org">foo</a>
<a id="bar" href="http://bar.biz">bar</a>
```

JavaScript

```javascript
<script>
var rl = 'a {color:green;}';
var r2 = 'a:visited {color:red;}';

document.styleSheets[0].insertRule(rl, 0);
document.styleSheets[0].insertRule(r2, 1);

var a_el = document.createElement('a');
a_el.href = "http://foo.org";

var a_style = document.defaultView.
    getComputedStyle(a_el, "");

if (a_style.getPropertyValue("color")
    == 'red') { // link was visited }
</script>
```

A known Mozilla “bug” since at least 2000
History (of) Detection

- Issue described by:
  - (Felten & Schneider), Ruderman, Jakobsson & Stamm., Jackson et al., others
- Several analyses of Web security issues (including Google’s BSH)
- Rediscovered on multiple occasions (PoCs)
- Life always goes on
What Changed Since Then

• Browsers still support :visited selectors
• The Web has changed
  • More apps are Web-based
  • More personal interactions with the Web (social networks/news, forums)
• Browsers are much faster
What Can Be Detected?

- Protocols
- Framed content
- HTTP status codes

- Usually: if in address bar \(\Leftrightarrow\) detectable
- Can detect parameters from forms submitted with HTTP GET (not POST)
- Affected by history expiration policies
How Long Does it Take?

- Modern browsers are fast
- Can do a few smart things to improve performance & avoid resource limits
- Can optimize JS detection code for each browser (can be significantly faster)
- Fallback CSS-only technique still good
How Long Does it Take?

- **JavaScript:** ~ 20,000 links/second
How Long Does it Take?

- CSS: up to 25,000 links/sec (small sets)
Detection System

- Demonstrate browser history detection
- Thousands of websites, categorized
- Detect secondary resources (subpages) and other information (usernames, etc)
- Educate users, describe issue
- Gather real world data (analyze impact)
This page checks your browser history and determines which of the 5000 most popular Internet websites you've recently visited.

**Popular websites you've recently visited (8)**

- **Google** (4 visited pages detected)
- **MSN** (5 visited pages detected)
- **ieee.org** (1 visited page detected)
- **cuteoverload.com** (1 visited page detected)
- **xkcd** (3 visited pages detected)
- **Reddit** (1 visited page detected)
- **Bing** (1 visited page detected)
- **Yahoo!** (1 visited page detected)

Quantcast top #1 site
Quantcast top #3 site
Quantcast top #2425 site
Quantcast top #4789 site
Popular webcomic
Reddit social news
Search engine
Tech site in the category: Web
<table>
<thead>
<tr>
<th>Website</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>Quantcast top #1 site</td>
</tr>
<tr>
<td><a href="http://www.google.com/services/">http://www.google.com/services/</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.google.com/">http://www.google.com/</a></td>
<td></td>
</tr>
<tr>
<td>MSN</td>
<td>Quantcast top #3 site</td>
</tr>
<tr>
<td><a href="http://moneycentral.msn.com/home.asp">http://moneycentral.msn.com/home.asp</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://entertainment.msn.com/">http://entertainment.msn.com/</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://g.msn.com/UTO_repus">http://g.msn.com/UTO_repus</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://msn.com/">http://msn.com/</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://lifestyle.msn.com/">http://lifestyle.msn.com/</a></td>
<td></td>
</tr>
<tr>
<td>.ieee.org</td>
<td>Quantcast top #2425 site</td>
</tr>
<tr>
<td><a href="http://ieee.org">http://ieee.org</a></td>
<td></td>
</tr>
<tr>
<td>cuteoverload.com</td>
<td>Quantcast top #4789 site</td>
</tr>
<tr>
<td><a href="http://cuteoverload.com/">http://cuteoverload.com/</a></td>
<td></td>
</tr>
<tr>
<td>xkcd</td>
<td>Popular webcomic</td>
</tr>
<tr>
<td><a href="http://xkcd.com/archive/">http://xkcd.com/archive/</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://blaz.xkcd.com/">http://blaz.xkcd.com/</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://xkcd.com/">http://xkcd.com/</a></td>
<td></td>
</tr>
<tr>
<td>Reddit</td>
<td>Reddit social news</td>
</tr>
<tr>
<td><a href="http://www.reddit.com/">http://www.reddit.com/</a></td>
<td></td>
</tr>
<tr>
<td>Bing</td>
<td>Search engine</td>
</tr>
<tr>
<td><a href="http://www.bing.com/">http://www.bing.com/</a></td>
<td></td>
</tr>
<tr>
<td>Yahoo!</td>
<td>Tech site in the category: Web</td>
</tr>
<tr>
<td><a href="http://www.yahoo.com/">http://www.yahoo.com/</a></td>
<td></td>
</tr>
</tbody>
</table>
How it Works

• For each test send primary links to user

• For each found link check ~100 popular secondary links (subpages & resources)
  • Crawling, search engine API, manual

• For certain sites, enumerate resources
  • Usernames, search terms, zipcodes
Test Categories

- Popular websites (Alexa, Quantcast, ...)
- Categorized sites
  - Online stores, .gov / .mil sites, banks, dating sites, universities, adult
- Social news sites: Slashdot, Digg, Reddit
- Sensitive sites (also zipcodes, search terms)
- 21 tests, 72k primary URLs, 8.6M secondary
**General Results**

- Gathered between 09/2009 and 02/2010
- 271,576 users, 703,895 tests executed

<table>
<thead>
<tr>
<th></th>
<th>JS</th>
<th>CSS</th>
<th>JS</th>
<th>CSS</th>
<th>JS</th>
<th>CSS</th>
<th>JS</th>
<th>CSS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Users</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>top5k</strong></td>
<td>206,437</td>
<td>8,165</td>
<td>76.1%</td>
<td>76.9%</td>
<td>12.7 (8)</td>
<td>9.8 (5)</td>
<td>49.9 (17)</td>
<td>34.6 (9)</td>
</tr>
<tr>
<td><strong>top20k</strong></td>
<td>31,151</td>
<td>1,263</td>
<td>75.4%</td>
<td>87.3%</td>
<td>13.6 (7)</td>
<td>15.1 (8)</td>
<td>48.1 (15)</td>
<td>51.0 (13)</td>
</tr>
<tr>
<td><strong>all</strong></td>
<td>32,158</td>
<td>1,325</td>
<td>69.7%</td>
<td>80.6%</td>
<td>15.3 (7)</td>
<td>20.0</td>
<td>49.1 (14)</td>
<td>61.2</td>
</tr>
</tbody>
</table>
90th percentile: ~30 primary, ~120 secondary
# Browser Differences

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>Firefox</th>
<th>Safari</th>
<th>Chrome</th>
<th>Opera</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JS</td>
<td>CSS</td>
<td>JS</td>
<td>CSS</td>
<td>JS</td>
</tr>
<tr>
<td>top5k</td>
<td>73</td>
<td>92</td>
<td>75</td>
<td>77</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>93</td>
<td>100</td>
<td>70</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>top20k</td>
<td>81</td>
<td>95</td>
<td>69</td>
<td>86</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>100</td>
<td>88</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>all</td>
<td>78</td>
<td>97</td>
<td>62</td>
<td>79</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>87</td>
<td>98</td>
<td>85</td>
<td>83</td>
<td></td>
</tr>
</tbody>
</table>
Social News

• Links from RSS feeds of popular social news sites and 32 regular news services

<table>
<thead>
<tr>
<th></th>
<th>Median secondary</th>
<th>Average secondary</th>
</tr>
</thead>
<tbody>
<tr>
<td>All news</td>
<td>7</td>
<td>45.0</td>
</tr>
<tr>
<td>Slashdot</td>
<td>3</td>
<td>15.2</td>
</tr>
<tr>
<td>Digg</td>
<td>7</td>
<td>51.8</td>
</tr>
<tr>
<td>Reddit</td>
<td>26</td>
<td>163.3</td>
</tr>
</tbody>
</table>

• Monitored for visited profile pages to detect usernames (Reddit: 2.4%)
Some Random Results

Percentage of visitors with adult sites in their browsing history

- Found some zipcodes (9.8%) and search engine queries (~0.2%)
- Can identify Wikileaks power users
Fixing It

- All browsers susceptible
- A server-side fix won’t help (impractical)
- Hard to get adoption for a plug-in (has been tried with SafeHistory)
- Hard to change browser behavior to close the hole (standards; developers get angry)
- But...
Coming Soon

• David Baron’s/Mozilla Corp.’s proposal
• Apply only *-color rules to visited styles
• Make JS functions lie about actual style
• Should be in Firefox 4.0 (~November)
• Similar changes rumored for WebKit
• Not ideal, but a big step forward; now we **must** get other browsers to do the same
Thank you