Reconstructing web pages from browser cache

Iwan Hoogendoorn
&
Edwin Schaap

University of Amsterdam

July 4, 2013
Demo I

- Open Safari
- Clear Safari’s cache
- Visit www.tweakers.net
Criminal research

- planning a crime
- committing the perfect crime
- Internet used as a resource
Evidence by a witness

- looking at content that is against the law
- content is removed by a suspect in a later stage
- Internet used as a resource
Forensic crime investigation

- computer forensics
- browser forensics
- web cache data forensics
Research question

In what ways can one visually reconstruct websites with information retrieved from normalized browser caches that can be use for computer forensic examiners to build a case?

- Raw caching data
- Reconstruction methods
- Reliability after reconstruction
Current forensic web cache tools

- Nirsoft
- Web Cache View
- Digital Detective
- Siquest
- Foxten Software
Netherlands Forensic Institute Tools

- XIRAF
- HANSKEN
- Traces
Figure 1: Browser popularity - Worldwide
Figure 2: Chrome web cache structure
Web cache data structure - Mozilla Firefox

Figure 3: Firefox web cache structure
Figure 4: Safari web cache structure
<table>
<thead>
<tr>
<th></th>
<th>Chrome</th>
<th>Firefox</th>
<th>Safari</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique identification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Eviction</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>URL request string</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Time/Date (first request)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Time/Date (last request)</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Time/Date (expire)</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Fetch count</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Client request headers</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Server response header</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Server response body</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

*Table 1: Firefox, Chrome and Safari web cache comparison table*
Traces - normalised cache data

- Unique identification
- URL request string
- Time/Date (first request)
- Server response body
Web page reconstruction methods - I

- pre-processing
- post-processing
Pre-processing

Advantages:
1. Requires no configuration of the rendering browser.
2. Can even run in the browser of the user enabling interaction.

Disadvantages:
1. Tampering the evidence.
2. Hard to parse all resource identifiers, especially if JavaScript is used.
3. Non-parsed resource identifiers are circumventing the application.
Post-processing

- **Advantages:**
  1. All resource identifiers are captured by the proxy.

- **Disadvantages:**
  1. Requires proxy configuration of rendering browser.
  2. SSL traffic is hard to deal with.
Proof of Concept
Figure 5: Web page reconstruction application
Demo II

- Reconstruct web page visited at the beginning of this presentation
- Compare before and after

`tweakers.net`
Result - Simple websites I

Original

Reconstructed
Result - Simple websites II

Original

NetAnalysis
Result - complex websites I

Original

Reconstructed
Result - complex websites II

Original

NetAnalysis
Analysis - Dynamic resources

1. Browser S displays website W1 on time A.
2. Website W1 contains resource R.
3. Browser S displays website W2 on time B.
Analysis - Runtime dependencies

1. Browser S visits website W.
2. Website W contains a dynamic time T.
3. Time T is taken from the local system time.
Conclusion

- Prefer post-processing
- Normalized data is sufficient
- Reliability depends on cache data