Influence of HTTP Header Entries on the Forensic Analysis of Web Browser Artifacts

Tobias Scheible
Faculty of Computer Science
Albstadt-Sigmaringen University
Albstadt, Germany
scheible@hs-albsig.de
https://orcid.org/0000-0002-8602-6919

***Abstract*— The traces (digital evidence) created by users can be influenced by the parameters transmitted by the web server. HTTP header entries instruct the web browser or web server to perform certain functions. These include, for example, the criteria according to which content is to be cached on the client. If these effects are unknown to the investigators, this can be erroneously classified as manipulation. In addition, it is now being considered that new traces can be created by header entries that are not yet in the focus of the common investigation procedures. In this work, a demo application was used to investigate which parameters have an influence on the forensic investigation of traces of Mozilla Firefox.**

Keywords—webbrowser, webserver, http, web forensics, browser history analysis

I. Introduction

More and more frequently, web applications are being used, replacing classic desktop applications in some areas. This can be seen in the example of mail clients, where a webmail service is now positioned above desktop software in the email client market share ranking [1]. Due to the wide range of functionalities and the intensive use of web applications, forensic analysis of the traces generated by a web browser is therefore of great interest. For this reason, there are many instructions and tools in this area to analyze the chronicle.

However, the rapid development of web browsers ensures that new functions are constantly being introduced, which cannot then be processed by the forensic tools. And at the same time the generated traces can be influenced by setting header entries by the web server. HTTP headers let the client and the server pass additional information with an HTTP request or response. These entries can, for example, cause no cached contents to be saved. During a forensic investigation, this could lead investigators to classify the missing data as manipulation. This would result in an incorrect investigation report. Header entries can also create new traces, which can be used by investigators to prove that a specific website has been visited even though the user has deleted the history data. Consequently, this led us to the following research questions:

R1 Which header entry has influence on the storage of the data on the client?

R2 Which header entry generates additional information outside the chronicle?

To answer this question a demo application was developed, which contains the corresponding entries. These were called up and then the resulting traces were analyzed.

A. Limitations

The focus of this paper is on the investigation of possible web browser artifacts through HTTP header entries. The web browser Mozilla Firefox was selected for the investigation. Only the changes on the client side are examined and the effects on the server side are therefore not considered any further. At the same time, the most important options were selected that have an influence on the cache or result in an entry.

B. Related work

In the area of Web Browser Forensics there are several research projects. They often focus on privacy modes [2][3] and the Tor Browser [4][5]. In addition, there are other papers that have dealt with the general analysis of artifacts [6] or have studied the Firefox web browser in particular [7]. This work differs from the other works in that it focuses on the connection between the header entries of the web server and the behavior of the web browser.

II. Methodology

In order to systematically investigate this area, the relevant header entries were first identified, then a demo application was developed and finally the investigation was executed and the results were evaluated.

A. HTTP Header

An overview of the available HTTP headers is described in RFC4229[8]. However, since not all header fields are supported by Mozilla Firefox, the overview of the Mozilla Developer Network[9] was also used.

The next step was to select the headers according to the following criteria: They change the behavior of the web browser as content is cached or generate additional entries that are not linked to the history. Only response headers are considered. Therefore, the following header entries are examined:

- Cache-Control
- Clear-Site-Data
- Strict-Transport-Security

B. Investigation Environment

A virtual system was used for the analyses. As a host system a computer running Windows 10 (Version 1903 - Build 18362.535) was used. VMware® Workstation 15 Player (Version 15.5.1 build-15018445) was utilized as virtualization software. Ubuntu 19.10 was chosen as the system for the investigation. There the Apache web server in version 2.4.41

with PHP (libapache2-mod-php) in version 7.3.11 was installed.

On the host system the web browser Mozilla Firefox (version 71.0(64-bit) as a portable version were used for the investigation. In Firefox, due to the portable configuration, the browser.cache.disk.enable option was reset to the default value and the start page was configured to an empty page.

The demo application was built in PHP. A virtual separate subpage was created for each header entry and for each parameter. Each page was assigned a unique ID, which was then searched for in the analysis. Additionally, an individual CSS and graphic file was integrated. To have basic data for the output, the first page has no header entries. For the investigation, first the start page was called up, then the subpage with the header entry being investigated and finally the web browser was closed. Then the traces were examined. After each cycle, a cleansed copy of the web browser was used. For the test with the HTTPS connection the domain https://scheible.it has been accessed.

III. Results Overview

The following table 1 gives an overview of the resources that are temporarily stored with each header entry.

TABLE I. Overview of the Results

HTTP header	HTML	CSS	Image
Clear-Site-Data: "*"	not saved	saved	saved
Cache-Control: no-store	saved	saved	saved
Cache-Control: no-cache	not saved	not saved	not saved

A. Clear-Site-Data

For the experiment the header *Clear-Site-Data: "*"* was set. The result of the investigation showed that the HTML files are not stored in the cache with this header entry. This applies to both the start page and the actual test page. Although the entry was only set for the test page. Other resources such as CSS files or images were not affected. This header entry ensures that no version of the page is stored in the cache and therefore only the entry in the chronicle is available during a forensic analysis.

B. HTTP header Cache-Control

Next, the header entry Cache-Control with the options no-store and no-cache was tested. Here, the no-cache option had no effect on storage, since the Web browser is instructed not to use the page from the cache without prior validation. Compared to A. Clear-Site-Data, the no-store option caches the home page, but does not save the actual test page. With this header entry in conjunction with the no-store option, specific individual pages cannot be saved and thus the examination is also disturbed.

C. HTTP Strict Transport Security

If a website has set an HSTS header entry, Firefox checks whether an entry already exists. If this is the case, it will be updated and if not, a new entry will be created. This attempt has shown that a new entry is created. Firefox manages these entries in a file named SiteSecurityServiceState.txt. Each entry for a domain is written in a single line. Besides the time information there is a counter for the number of visits. This is independent of the entry in the chronicle. An example for this file can be seen in the following figure 1.

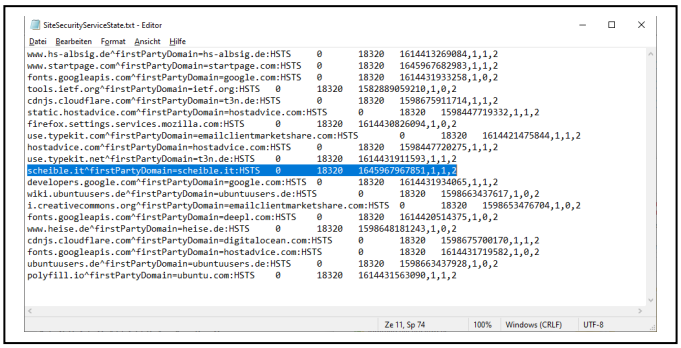

Figure 1: Display of the file with the HSTS entries

If, for example, the option is set that the complete chronicle is to be deleted when closing Firefox, the entry remains present. This entry allows to see how often the page has been visited and when the last visit took place.

IV. Conclusion and Future Work

The results of this work have shown that the transmitted header entries of the web server must be included in the analysis of the client in order to be able to perform a correct analysis of the traces. This allows additional insights to be gained during the forensic investigation. Since the found artifacts are not automatically removed by the option that the chronicle should be deleted automatically when closing, is particularly interesting for forensic analysis. This enables the proof of the access to a website, even if the history data has been purposefully deleted by a user.

In order to systematically investigate this area, a more comprehensive demo application should be created, covering all relevant parameters. In addition, the investigation should be extended to web browsers from other vendors to get a more comprehensive result. These additional entries could also be used to uncover inconsistency in history, to uncover manipulations performed by users. In addition, further HTML functions and, where appropriate, anti-forensic methods could be integrated to create a test environment for analyzing the coverage of forensic software.

References

[1] Litmus Email Analytics, Email Client Market Share. Accessed on: Feb. 28, 2020. [Online]. Available: https://emailclientmarketshare.com

[2] Gabet, R.M.: A Comparative Forensic Analysis of Privacy Enhanced Web Browsers (Doctoral dissertation, Purdue University) (2016)

[3] Tsalis, N., Mylonas, A., Nisioti, A., Gritzalis, D., Katos, V.: Exploring the protection of private browsing in desktop browsers. Comput. Secur. 67, 181–197 (2017)

[4] Keller, K.: The Tor browser: A forensic investigation study (Doctoral dissertation, Utica College) (2016, December)

[5] Boggs, R.J., Fenger, T., Sammons, J., Winkler, D.: Online anonymity: forensic analysis of the tor browser bundle (2017)

[6] Oh, J., Lee, S., Lee, S.: Advanced evidence collection and analysis of web browser activity. Digital Invest. 8(S62), S70 (2011)

[7] Mahaju, S., Atkison, T.: Evaluation of Firefox Browser Forensics Tools. In Proceedings of the SouthEast Conference, pp. 5–12. ACM (2017)

[8] Network Working Group, Request for Comments: 4229, HTTP Header Field Registrations. Accessed on: Feb. 28, 2020. [Online]. Available: https://tools.ietf.org/html/rfc4229

[9] Mozilla Developer Network – web docs, HTTP headers, Nov. 11, 2019. Accessed on: Feb. 28, 2020. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers