International Journal of Cyber-Security and Digital Forensics (IJCSDF) 7(2): 175-182
The Society of Digital Information and Wireless Communications (SDIWC), 2018 ISSN: 2305-001

Acquisition of Browser Artifacts from Android Devices

Emrah Sariboz and Cihan Varol
Department of Computer Science
Sam Houston State University
1803 Ave I, AB1 214, Huntsville, Texas, 77341, USA
exs060@shsu.edu, *cvx007@shsu.edu

ABSTRACT

Providing quality of information in a quick manner is getting more doable each day with the increase of modern technologies such as HTML5. HTML5 came out with a recommended data storage feature called client-side data storage, i.e. web storage. Web storage, aims to store crucial and meaningful client-side data in a persistent and secure manner. Before the web storage feature, cookies were the solution to store a modest amount of data specific to a client and a website. The main limitation of the cookies was their size, which was 4 Kilobytes at maximum. Therefore, due to lack of capability of cookies, the usage of web storage feature has rapidly increased and become the main storage of the five major web browsers, Google Chrome, Internet Explorer, Mozilla Firefox, Opera, and Apple's Safari. Due to increased usage of the web storage, it has become a research topic of interest to forensic investigators.

In 2015, Mendoza et al worked on the desktop implementation of the web storage feature and investigated the extraction and mining of web browser artifacts for forensic purpose. This study will go beyond Mendoza's work and reveal the usage of web storage feature in mobile devices, specifically on Android platform. The five major mobile browsers (Google Chrome, Samsung, Firefox, Opera, and Web Explorer) are investigated in a forensic manner for web storage. Specifically, mostly visited 15 websites are investigated for local storage implementation on Android platform. Obtained results show that there are extensive similarities on the implementation of web storage on desktop platforms and mobile devices, such as stored information differences among the visited website and used web browser. Overall, this research will help an investigator to acquire web storage data that can potentially hold forensically important evidence for a crime in which the mobile device can be an instrument to commit a crime or even when a computer and its data is the target of a crime.

KEYWORDS

Android Forensics, Browser Artifacts, Local Storage, Web Storage.

1 INTRODUCTION

Within the last decade, mobile phones have become an integral part of humans. Formerly, mobile phones that are used only for calling and messaging now have access to all kinds of information from the internet. The fact that mobile phones are being used so much created another venue to be investigated for digital forensics investigators. It can possibly provide forensically important evidence for a criminal who has used web browser as an instrument to commit a crime or even when a computer or its data is the target of the crime [1]. This is also supported by Damshenas et al by pointing out that smartphone forensics is one of the leading trends in digital investigation and is only expected to grow in future [2].

Variety of digital forensic research work were conducted on mobile devices, such as exploring mobile Firefox operating system from forensics point of view [3]. Although this paper focuses on web storage feature on android smartphones, the closest work to ours were conducted on a desktop platform by Mendoza et al [1]. Specifically, authors investigated the implementation of local and session storage on most widely used web browsers and created a software product to collect and share the data from desktop platform with an investigator [1].

However, according to marketingland.com, mobile platform usage now represents 65 percent of all digital time spent by a user [4]. On the

other hand, desktop platforms have lost 12 percentage points since 2013 and has retreated to 35 percent of digital time spent [4]. Therefore, it is vital to not only investigate the web storage on the desktop implementation, but also on the mobile devices as well. Therefore, this work goes beyond what Mendoza et al. provided and it will offer findings on local storage data on Android platform.

HTML5 Web Storage consists of two browser-based Application Programming Interface (APIs) that allow for persistent client-side storage for web applications. These native browser APIs, known as sessionStorage and localStorage, allow websites to store domain-specific key/value pairs of data. These browsers-based APIs allow data to be stored in the client-side. Previous solutions to store data such as cookies were not enough to store the great amount of data. Now, developers also realized the importance of the web storage because of the ability of storing large amount of data. In addition, unlike cookies, web storage is not transmitted over each request to and from a web server, which thereby reduces the bandwidth overhead.

In this work, investigation started with surfing different web sites from variety of browsers (Google Chrome, Samsung, Firefox, Opera, and Web Explorer) to collect data. Then, local storage contents are explored to reveal the left artifacts of web activity. Overall, this research compared the details and nuances of web storage implementation in Android platform among the five major mobile web browsers: Google Chrome, Samsung, Firefox, Opera, and Web Explorer.

The remainder of the paper is organized as follows. In Section 2 related work about web storage is discussed. In Sections 3 and 4, the details of persistent storage and HTML5 client-side storage in different browsers are summarized. The findings on web storage implementation and stored data are discussed in Section 5. Finally, the paper is concluded with summary and brief discussion on future research direction in Section 6.

2 PRIOR WORK

To the best of our knowledge, there is no proven web storage investigation conducted on the Android platform indicating the location of the files and contents of it. According to developer.mozilla.org, the web storage feature is supported in mobile web browsers [5]. Another guideline of the web storage feature could be W3C (World Wide Web Consortium) which is also concentrated on the desktop implementation [6]. Research on the web storage feature has often been limited to where the information is stored and the privacy concerns on the web storage, and even these is limited with the desktop implementation.

Daniel and Daryl have explained the web storage features and vulnerabilities of web storage [7]. Authors indicated that the web storage is a crucial feature that can serve as evidence of a crime when web browser activities are valuable for the case. One of the raised concerns on the paper was that the criminals try to avoid leaving any footprints on their computers; instead, they look for a solution to store the file, so the found files won't be associated with them on a possible investigation. Again, they claimed that this information could be stored on various client systems if the malicious user has access to a domain. The study illustrates personal data as a potential security infraction and a hint in application of digital forensics principals.

Undoubtedly, one of the greatest achievement of the web storage was privacy. West and Monisha [8] demonstrated that web storage feature is more secured than the cookies. When a website wants to save the users' preferences for next visit, the website will store the key/value pairs in local storage while cookies create a unique ID; thus, server would have access to user's information constantly. When it comes to security, and when key/value pairs need to be changed, JavaScript can do this without having a server connection. But the cookies needed to be connected to the server. As a result of this introduced local storage feature, the packet sniffing attack was reduced to the smallest.

Janik and Kiebzakc designed a framework that integrates Java Server Pages technologies with the web storage feature. The main purpose of the study was to use the power of client-side storage and integrate it with the Java to create powerful client web application without the utilization of the JavaScript API's [9]. In addition, they concentrated on the feature of storing data on the client-side, which reduces the queue of data on the server-side. Their claim was

the data on the victim's local storage can be stolen by the malicious script. They claimed that if the data, which are stored on the client-side, is sanitized good enough, still client will be vulnerable to the attackers.

Mendoza et al. has created new perspective on the web forensic analysis with the study of web storage implementation on desktop computers [1]. They have proved that the extraction of the information from web storage artifacts is possible even they may not be associated with other browser artifacts such as cookies. They have studied the implementation of the web storage feature in five major browsers. To extract browser artifact, they created a tool, BrowStEx, to collect and parse Xml and SQLite files from web browsers and present it to the web investigator for further investigations.

Literature study has shown the importance of web storage feature and how it is also applicable to the one on the Android platform. Therefore, there is a dire need to investigate web storage feature on Android platform.

Table 1. Web storage implementation of 15 websites on mobile devices

Rank	Websites	Web Storage Feature
1	Facebook	Yes
2	Google	Yes
3	YouTube	Yes
4	Yahoo!	Yes
5	Amazon	Yes
6	Wikipedia	Yes
7	Twitter	Yes
8	Bing	Yes
9	eBay	Yes
10	Live.com	Yes
11	Microsoft	Yes
12	LinkedIn	Yes
13	Pinterest	Yes
14	Ask	No
15	WordPress	No

3 HTML WEB STORAGE CONCEPTS AND USAGE

Web storage consists of two basic APIs known as localStorage and sessionStorage [1]. These two APIs generate the key and value pairs from websites that one surfed via mobile or desktop devices. Today, the five most popular web browsers (Chrome, Mozilla Firefox, Samsung, Web Explorer and Opera) according to on mobile platform, store all key and value pairs in local and session storage [10]. The main difference between localStorage and sessionStorage is that localStorage stores data with no expiration date, but sessionStorage is a per-origin-per-window, in other words, the life of session is limited with the life of the browser window. This study particularly concentrated on localStorage, which contains information for all visited webpages. Another reasoning for investigation local storage is that forensic investigations are usually conducted when the phone is in rest mode, so browser windows is likely closed, and an investigator may be forced to get web data from previously visited websites via localStorage.

Web storage implementation and the idea of usage of it significantly differs from cookies. As introduced earlier, cookies were the solution to store data before web storage feature, but cookies were not sized enough to store necessary data for a webpage visit. On the other hand, web storage can store data up to 10 megabytes(MB) for each domain. Another difference between cookies and web storage is the use of the network. Every HTTP request includes cookies turns out more data transmission over the network. In web storage, data is not included with every request [11].

Mendoza et al, showed the web storage adoption by top 15 US websites, which has been ranked by Alexa. In this research, the same websites are investigated for usage of the web storage feature on mobile browsers. Web storage feature implementation on the most visited websites on mobile devices is shown in Table 1. Particularly this table shows the 15 most popular websites in the US, sorted and ranked by Alexa in April 2017 [12].

4 COLLECTION OF DATA

According to netmarketshare.com, Android is the most popular operating system in mobile devices [13]. There are a couple of reasons why Android is the major browser in mobile world. One, and the most effective reason is customization of the phone. Rooting is one of the well-known option to gain complete access of the mobile devices, which provides access to hidden features and yield customization of the phone. There are multiple ways to root mobile devices, such as Kingo root, su binary, or Android Package Kit (APK), that can be installed to mobile devices without computer access [14].

Arguably, the best way to achieve root access to devices is installing su binary because of the easy utilization. After rooting process, the phone is no longer limited by the default Android system. Manipulation on the security, using not authorized apps (application), and most importantly using root enabled-apps like Titanium back-up are become achievable in the mobile device. In this project, the phone needed to be rooted to reach out hidden features and back-up those to a computer for further analysis. Titanium back-up was used to back-up the whole phone content to an external storage [15]. The process of obtaining web-forensics data is shown in Figure 1.

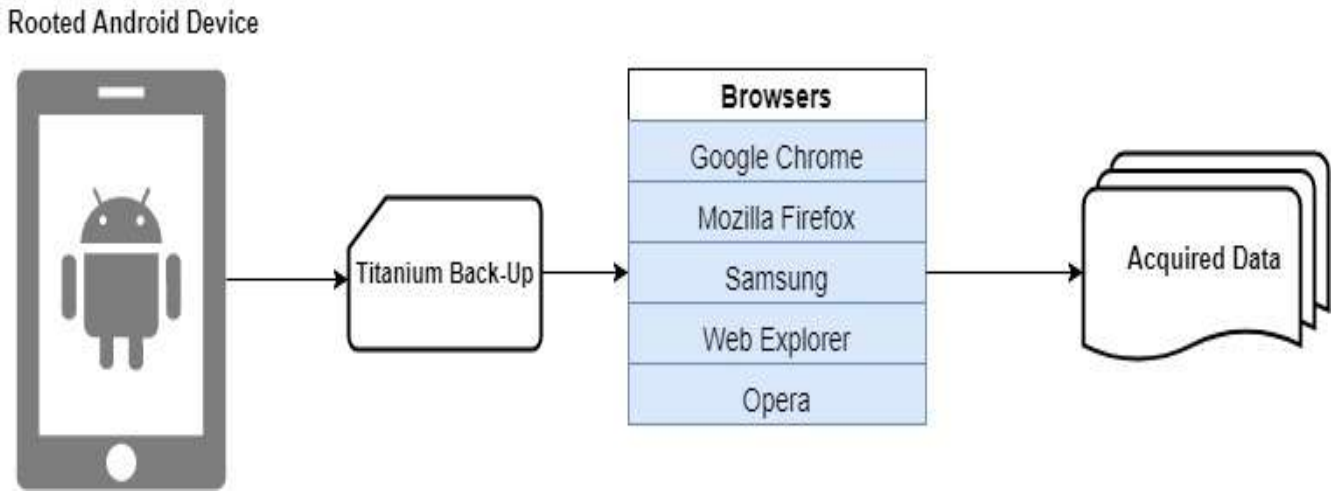

Figure 2. Collection of local storage

5. RESULTS

This work is conducted research on the five major web browsers according to androidauthority.com, to evaluate usage of web storage adaption, these are Chrome, Mozilla Firefox, Samsung, Opera, and Web Explorer [10]. Five of the web browsers that are used in this research proved the adoption of the web storage feature. Details of the web storage implementation on Android platform is shown in Figure 2. Table 2 provides the file types of the local storage data, along with extensions and number of files available for each website. Table 3 provides the storage areas of the local storage files that have found in the Android devices. As it is clear from the table, the storage location varies among the web browser.

For each web browser, top 15 websites are visited to collect web storage data. Some of the actions that are practiced on the websites were: liking a news/page, searching, tweeting, and commenting. The local storage details of each browser are shared below.

5. 1 Chrome

The Chrome version that is used in this research was 57.0.2987.132. Chrome stores the local storage as a [protocol] [domain]. localstorage which is the same as the desktop implementation of the web storage feature. [Protocol] stands for HTTP \ https and domain are the address of the website. For example, the local storage file for data collected from YouTube is https_m.youtube.com_0. localstorage. SQLite database used to access the contents of files with extension localStorage. After the examination of the localStorage file, key-value pairs were located within the SQLite database.

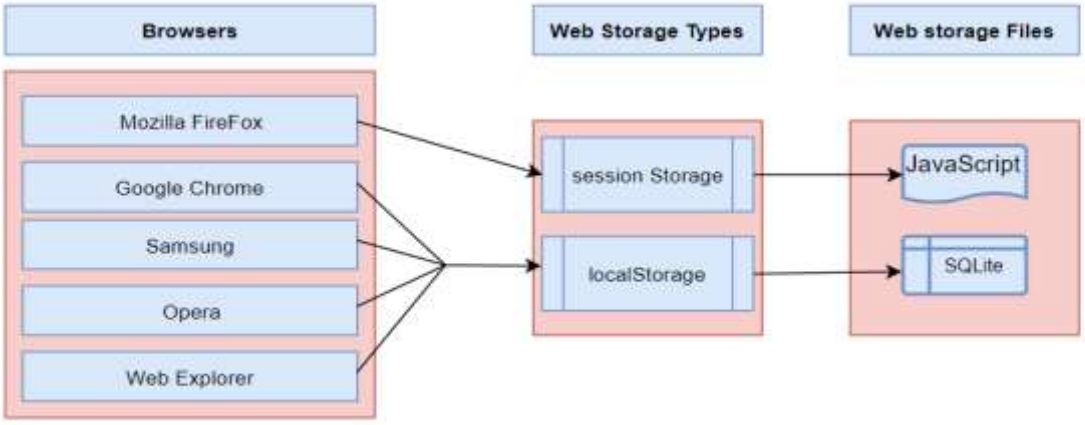

Figure 2. Web Storage Implementation on Android Platform

Table 2. Web storage file location on Windows Platform

	File Extension	File Type	Number of Files
Chrome	.localStorage .localStorage-journal	SQLite	# of origins * 2
FireFox	.sessionStorage	JavaScript	# of origins * 2
Samsung	.localStorage.localStorage-journal	SQLite	# of origins * 2
Opera	.localStorage .localStorage-journal	SQLite	# of origins * 2
Web Explorer	.localStorage .localStorage-journal	SQLite	# of origins * 2

Table 3. Web storage file location on Windows Platform

Browser	Directory
Chrome	"\data\data\com.android.chrome\app_chrome\Default\Local Storage"
FireFox	"\data\data\org.mozilla.firefox\files\mozilla\9qb6g46a.default\Session Storage"
Samsung	"\data\data\com.sec.android.app.sbrowser\app_sbrowser\Default\Local Storage"
Opera	"\data\data\com.opera.browser\app_opera\Local Storage"
Web Explorer	"\data\data\webexplorer.amazing.speed\app_webview\Local Storage"

5. 2 Firefox

The version of Firefox used in this research was 52.0.2. In the top five mobile web browsers, Firefox was the only one, which does not store the localStorage file (it does store sessionStorage data) in the data/data subdirectory. Security and privacy could be the reasons that are not storing

them in the files. Again, unlike the others, only backup files were found in sessionStorage. The extension of this file was named sessionStorage is JavaScript. The fact that all data is kept in a single file gives the researcher a great advantage over time because the researcher will only deal with one file.

5. 3 Samsung

The version of the Samsung browser in this research was 3.3.8. The way the Samsung web browser stores the data is same as Chrome (see Table 3). Samsung stores the local storage as a [protocol] [domain]. localstorage. The SQLite database was used to access the contents of localStorage.

5. 4 Opera

The version of the Opera in this research was 47.7.2246. Opera stores the data from web storage feature in a single file called localStorage and SQLite is used to reach out data for files. It is worthy to note that Opera Mini web browser was also investigated and concluded that it is not supporting the web storage feature.

5. 5 Web Explorer

The version of the Web Explorer in this research is 11.0. Explorer stores the data of the web storage feature in a single file called localStorage. Again, SQLite is used to reach out data for files. Chrome, Opera, Samsung, and Web Explorer were storing the data of web storage file within a single file with the extension of localStorage. SQLite was the main application used to find out saved data for all web browsers.

Previous research shows that the internal data of all applications present on the device (either system or user-installed applications) is automatically saved in the data/data subdirectory, named after the package name (DOM STORAGE) [16]. In this research, the tradition of data storage and web storage has not changed, and top five web browsers which is discussed above, have stored data in the data/data subdirectory. However, the use of the web storage feature has shown significant differences between browsers.

The first difference that was observed was that Firefox recorded data as just one file and in a different extension. This diversity is also seen in the desktop version of the web storage feature analyzed by Mendoza et al. Google Chrome, Web Explorer, Samsung, and Opera's localStorage file extension type was SQLite, but Firefox stored the saved data in a file with the sessionStorage extension, and the type of this file was JavaScript. Despite all the differences, localStorage and sessionStorage record the data in the form of key-value pairs. Depending on the amount of data collected and the number of operations, the stored key and value pairs are incremented. Another difference that was observed is on the stored data. The stored data varies among the websites and browsers. Specifically, each browser store different amount of information for a given website. Moreover, as shown below, CNN.com site stores information such as where the logged-in user is logged in, browser type, login date and time, while yahoo.com stores the news headline that the user has read, etc. as shown in Table 4.

Compared to Mendoza et al. work [1], the data collected from the browser artifacts were different in this study. For example, collected Cnn.com data from the Android device stored the type of search engine used in browsing activity. However, this information was not available on the desktop implementation of web storage. The IP address of the user was stored on the desktop implementation of the web storage, but same data was not available on the mobile implementation. As can be seen from these results, the stored data may show differences on both the desktop and mobile implementation of web storage. Another difference between two research works is the created web storage file for Mozilla Firefox. Despite the creation of the SQLite file for Mozilla Firefox in the desktop application, a JavaScript file has been created in the mobile application.

Table 4. localStorage Data Sample

Websites	Key	Value
Yahoo.com	AF_5b02c0af-67cc-3990-b177-a313ec5c3637	**"title":"Tax story puts spotlight on MSNBC's Rachel Maddow"**,"link":"https://www.yahoo.com/tv/tax-story-puts-spotlight-msnbcs-rachel-maddow-032437070.html","
	AF_f7351af5-32a0-3a95-a741-333bb328d37e	**{"type":"story",**"is_eligible":"true","id":"b15dd0e1-5793-36dc-aad0-f6e8484698a7","licensed":"true**","title":"It's Still 'Friday' Six Years Later: What Rebecca Black Is Up**
Cnn.com	lastSEPromptDate	Fri Mar 17 2017 02:20:22 GMT+0000(GMT)
	Kxgeo	domain=suddenlink.net&**country=us**&longitude=-95.5724&latitude=30.6885&dma=618&zip=77340&**region=tx**
	optimizely_data$$oeu1491278426928r0.4959968577604741$$131788053 $$visitor_profile	"referrer":"**http://www.bing.com/search?setmkt**=en-GB&q=cnn&PC=SMSM&FORM=MBDPSB","source_type":"search","cookies"::"1","bounceClientVisit340v":":**"US","geoData":"Huntsville|TX|77340|US|NA"**

6 CONCLUSION AND FUTURE WORK

This research is an extension of web storage study conducted by Menzoda et al. The increase in daily use of the phone caused researchers to concentrate more on problems associated with the phones rather than computers. Thus, the main purpose of this article was to evaluate, extract, and present meaningful data stored by the local storage on Android platform via different web browsers. The information revealed by local storage artifacts, are beyond what is presented by other web stored data such as cookies and history. Therefore, it can possibly provide forensically important evidence, which was not possible to obtain with previous web technology. Currently, to the best of our knowledge, there is no other study in the literature that targets to obtain web storage artifacts from mobile devices. Therefore, this work was aimed to provide details of how web storage data is stored in Android devices and how one can extract the useful information from it.

Robust and correct extraction and efficient presentation of forensically important information from the web storage will help investigators to resolve a lawsuit in a correct and quicker manner. Thus, automatically retrieving web storage data and converting it to a readable format are other objectives of future work, so that the forensic investigators will save the significant amount of time during their investigation.

REFERENCES

1. Mendoza, A., Kumar, A., Midcap, D., Cho, H., Varol, C.: BrowStEx: A Tool to Aggregate Browser Storage Artifacts for Forensic Analysis. Digital Investigation Journal, 14. pp. 63-75 (2015).
2. Damshenas, M., Dehghantanha, A., Mahmoud, R.: A Survey on Digital Forensics Trends. International Journal of Cyber-Security and Digital Forensics (IJCSDF), 3. pp. 209-234. (2014).
3. Yusoff, M.N., Mahmod, R., Dehghantanha, A., Abdullah, M.T.: Advances of Mobile Forensic Procedures in Firefox OS. International Journal of Cyber-Security and Digital Forensics (IJCSDF), 3. pp. 183-189. (2014).
4. Digital Marketing & Martech News, Tactics & Strategies. Retrieved from: https://marketingland.com/
5. Web Storage API. Retrieved from: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API
6. HTML5 Web Storage. Retrieved from: https://www.w3schools.com/html/html5_webstorage.asp
7. Daniel, B., Daryl, J., Robert, P.: Browser Web Storage Vulnerability Investigation: HTML5 localStorage

Object. The 2012 International Conference on Security and Management (SAM 12). (2012).
8. William W., Pulimood S. M.: Analysis of Privacy and Security in HTML5 Web Storage. Journal of Computing Sciences in Colleges, 27. pp. 80-87. (2012).
9. Janik, J., Kiebzak, S.: Client-Side Storage: Offline Availability of Data. Journal of Automation, Mobile Robotics and Intelligent Systems, 8. pp. 3-10. (2014)
10. Android Authority. Retrieved from: https://www.androidauthority.com
11. Bajaj, P. Overview: Responsive Web Design. Retrieved from: https://www.slideshare.net/PankajBajaj7
12. Alexa Ranking. Retrieved from: https://www.alexa.com
13. Market Share Statistics for Internet Technologies. Retrieved from: https://www.netmarketshare.com
14. Sun, S.T., Cuadros, A., Beznosov, K.: Android Rooting: Methods, Detection and Evasion. Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM 15). pp. 3-14. https://doi.org/10.1145/2808117.2808126 (2015)
15. Titanium Back Up. Retrieved from: https://www.titaniumbackup.com
16. Resig, J., Ferguson, R., Paxton, J.: Pro JavaScript Techniques, Apress. pp 161-175. (2016)