Ingénierie des Systèmes d'Information
Vol. 26, No. 4, August, 2021, pp. 337-344
Journal homepage: http://iieta.org/journals/isi

Streamed Video Reconstruction for Firefox Browser Forensics

Mahmoud El-Tayeb[1*], Ahmed Taha[1], Zaki Taha[2]

[1] Faculty of Computers & Artificial Intelligence, Benha University, Benha 13518, Egypt
[2] Faculty of Computer & Information Sciences, Ain Shams University, Cairo 11566, Egypt

Corresponding Author Email: mahmoudtayeb79@gmail.com

https://doi.org/10.18280/isi.260401

Received: 12 June 2021
Accepted: 9 August 2021

Keywords:
digital forensics, browser cache, social media (SM), video stream, data fragments, YouTube, twitter, Firefox

ABSTRACT

In criminal investigations, the digital evidence extracted from social media may provide exceptional support. Reviewing the history or cache of the web browser may provide a valuable insight into the activity of the suspect. The growing popularity of Internet video streaming creates a risk of this technology misuse. There are a few published research on video reconstruction forensics on the Chrome browser. There is a difference in the methods applied to reconstruct cached video on Chrome from the methods applied to Firefox or any browser. Our primary focus in this research is to examine the forensic procedures required to reconstruct cached video stream data using Twitter and YouTube on the Firefox browser. Some work has been done to reconstruct a cached video on the Chrome browser, but we need more work on the rest of the browsers, most notably the Firefox browser used in this research. Both examination strategies and contemplations displayed are approved and suitable for the forensic study of various streaming platforms as well as the web browser caches.

1. INTRODUCTION

Social media content can give extraordinary support to investigators in the criminal investigation process. It is an infinite source of information about possible suspects, victims, and witnesses. It offers a dynamic and new subdivision of data sources created by individuals. This includes friend lists, text posts, images, videos, geolocation data, demographic information, and so forth. Online Social Network (OSN) is a social structure that contains websites such as Facebook, Instagram, YouTube, or Twitter [1]. In 2018, about 3.196 billion users actively shared their everyday activities on social media sites [2]. Video streaming websites currently allow users to share information and identify (by streaming) video content provided by others without revealing ownership in terms of intentionally downloading and saving video content. Forensic analysis may be necessary to detect any potentially streaming content. Trials involving social media evidence are continually growing. 689 cases with social media evidence were published in 2012 [1]. The information posted on social media websites about a person, activities, and actions is sometimes used as a potential tool by investigators to backtrack a crime. In 2018 [2], the Internet Watch Foundation highlighted the role of streams depicting child abuse not only as a primary source of abusive material but also as a secondary means for imagery to be harvested and subsequently redistributed. The use of social media evidence is increasing significantly since 2015 [2]. Fourteen thousand decisions were discovered in 2016 in one year, only in the US. Nine thousand five hundred were mainly dependent on social media evidence among those verdicts [2]. It is needed to manually examine and extract artifacts from a suspect system and carry out event reconstruction as part of a digital investigation. The objective of this research is to analyze and reassemble the forensic artefacts of cached video streams from the installed Mozilla Firefox web browser. The study attempts to answer the following questions: When viewed, is streamed video content kept on the device? And, if so, can online content that has been streamed be recovered and displayed? Is it possible to find out how much of a video has been watched? This research employed an effective technique to forensically analyze YouTube/Twitter video streams utilizing Mozilla Firefox browser as a streaming video content platform. The processes for testing as well as the outcomes are presented. This research paper is organized as follows: Section 2 describes previous and related work in web browser cache reconstruction. Section 3 illustrates the characteristics of the Firefox browser and its cache structure contents; it also gives an overview of the concept of video reconstruction. Section 4 discusses the proposed cached video reconstruction technique using YouTube/Twitter as a streaming video platform. Section 5 presents the implementation details and experiments we made on both YouTube & Twitter websites. Section 6 concludes the paper with some open questions and future work.

2. RELATED WORK

Although there are many types of research in digital forensics, there has been a small number of published research on the forensic analysis of video reconstruction. Graeme Horsman [3] has provided a base for local analysis of video streams. He highlights investigatory approaches to discover both extremist videos and hidden communities. He presented two case studies, one on YouTube and the other on Facebook Live, both of which rely on single file viewing' as a method of identifying and authenticating video material. He utilized Google Chrome browser to view and stream video material.

The authors [4] investigated the possibility of reconstructing a web page from browser cache using a post-processing approach without distorting the evidence. It is also checked to see whether enough information is gathered to construct a web page. Their study aims to provide a better knowledge of online page reconstruction based on browser cache. Various browsers store cache data in different ways. Although browsers differ in the amount of data types saved, normalized data contains the same cache data fragments. They also exhibited two methods of rebuilding websites: pre- and post-processing, however they were unable to rebuild cache-case video stream material. Marrington et al. [5] made an experimental methodology to forensically examine and investigate the forensic remnants of both installed and portable web browsers. The experiment tested the privacy of Google Chrome Portable through forensic analysis of the forensic artifacts left by the portable web browser on the local hard disk, compared to the artifacts left by a normal, installed version of Google Chrome. Their experiment did not show how to reconstruct video stream content on both installed and portable web browsers. Besides, a methodology is offered for the analysis of private and portable artifacts [6]. They showed that further data could be reconstructed on host computers without the external storage device being present. The majority of reconstructed artifacts were discovered in slack/free space, FTK software directories, and RAM. Their method could not reconstruct video stream contents on browsers' cache.

3. METHODOLOGY

This section explores the concept of Web caching in general and Firefox in particular. It discusses the video reconstruction term; also, it reveals the most significant challenges facing the reconstruction from Firefox.

3.1 Browser cache

The web browser is a program that allows users to access web applications and web pages on the Internet. Web browser usage grows as more online applications are migrated to the Web in web applications. Browser caching is a mechanism for temporarily storing files obtained from the visited websites on a local disk. When the web page is visited again at a later time, it will load considerably quicker. The data from the online web page is compared to the data stored in the cache folder by the web browser. If this web page hasn't changed, the cache or parts will be used, and the page will be downloaded, displayed, and most probably cached again. When the browser is closed, the web cache is saved in a particular location on the hard disk. Different options like the amount of cache that can be saved and the cache deletion are available in utmost browsers. Online and offline caching are the two types of caches known. Offline caching differs in that the web page developer specifies which portions of the visited web page are cached. These elements are defined by the web developer in a manifest [6], which is a predefined file. When using online caching, the browser decides what should be cached and what should be left out.

3.2 Mozilla Firefox

Mozilla Firefox is a free and open-source browser for Mac OS X, Linux, and Windows [7]. The changes we make in Firefox, as well as our bookmarks and passwords, are all saved in the profile folder. A brief look into Firefox's caching folder reveals three types of files that reassemble the cache data. There are three cache block files, as well as separate cache data files and a cache map file. In order to reconstruct web pages from Firefox Cache data, the cache map file will be the primary file (see Figure 1). The structure of this map file includes a file header, followed by allocated space, known known as “buckets”, which contains information about the mapping to the cached data. The CACHE MAP file is divided into 32 buckets and within each bucket there is room for 256 records (total of 8192 records) [4]. Each record represents a single cached instance of data. A Hash Number, an Eviction Rank, the Data Location, and the Metadata Location are all 32-bit integers in a single record.

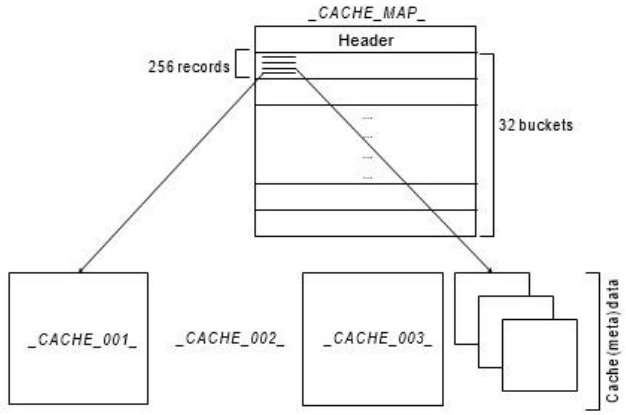

Figure 1. Mozilla Firefox file cache structure [6]

3.3 Video reconstruction

Video reconstruction is an essential part of any digital investigation process. It is the process of putting pieces of evidence together during the initial phases of an investigation to improve the understanding of what events occurred. This paper discusses if it is possible to recover the streamed video content and determines how much of a video has been viewed. When dealing with the Firefox browser, there are some challenges. One of them is that no reconstruction experiments on Twitter and YouTube have been made before on Firefox. Experiments were only done on Chrome browser [3]. Also, Chrome's cache file structure differs from Firefox's structure. Besides, the forensic tools used in recovering Chrome video are different from the tools in Firefox. The next section discusses the technique of the proposed cached video reconstruction.

4. THE PROPOSED TECHNIQUE

This section presents the proposed cached video reconstruction technique. Figure 2 shows the general stages of reconstructing cashed video files from the Firefox browser. The proposed technique consists of three major phases: Collecting, Analysis, and Recovery phase. Each phase is illustrated in the following subsections.

4.1 Watching session

A short watching session was performed on the Firefox browser on PC. After that, the browser was closed, and the device was turned off and imaged. The grey video bar displays

the buffering process when a YouTube / Twitter stream is playing, according to the initial test. After disconnecting the Internet connection, some of the buffered portions of the stream can be replayed. It means that this information is being replayed from locally stored local content rather than data on the YouTube / Twitter server (see Figure 3).

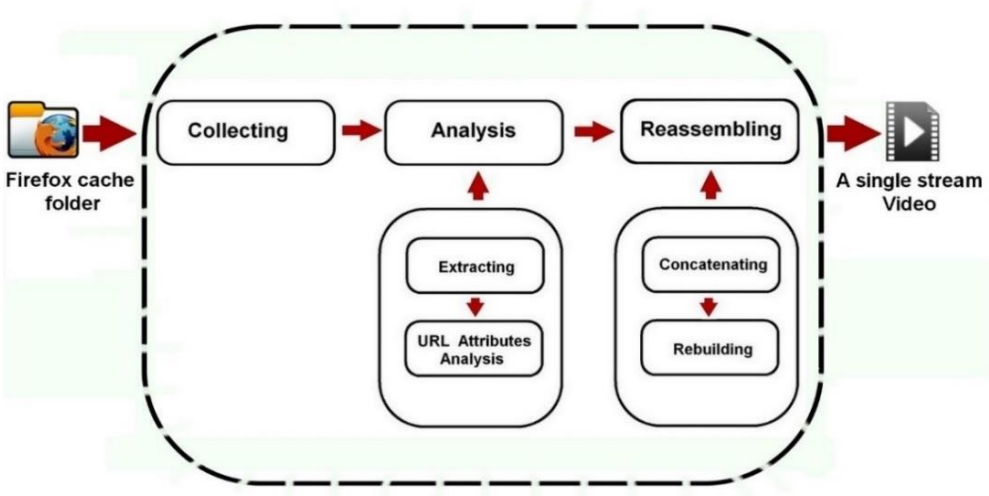

Figure 2. The proposed technique of cached video reconstruction

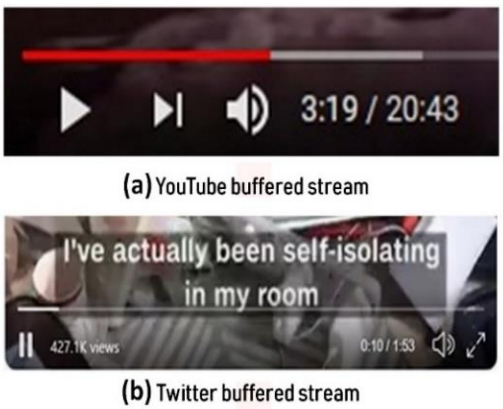

Figure 3. A buffered stream examples

4.2 Collecting phase

This process aims to determine the location of the Firefox folder cache on the local disk. Cache folder contains various file types, including audio, video, text, etc. The cached video is temporarily stored as fragments in the default location of Firefox on the local disk "**C:\Users\<user>\AppData\Local\Mozilla\Firefox\Profiles\2mto9q8n.default-release\cache2\entries**". Each cached fragment contains a small part from the cached video stream. After locating the cache folder, the analysis of folder contents begins.

4.3 Analysis

The main objective is to analyze the attributes of each extracted cached Uniform Resource Locator (URL) fragment before the reassembling phase. This process consists of two phases: extracting and URL attributes analysis.

4.3.1 Extracting

In this phase, the analyst uses a suitable tool (MZCacheView version 1.90 [8]) to explore the browser cache folder's contents depending on its structure. The extracted fragments are prepared in a separate folder for the next phase.

4.3.2 URL attributes analysis

There are two types of URLs to deal with, the standard and the cached URL. The standard URL of YouTube/Twitter are post-fixed with a unique identifier (see Figure 4). The analyst can use this identifier to search for the video and validate its content. The cached URL consists of the main attributes to reconstruct the cached video (see Figure 5). Each cache entry has its associated cache URL and must be examined to identify its "fragment order". During the buffering process, data is stored on the local disk while a YouTube and Twitter stream is accessed. There are clear differences between the cached URL of YouTube and Twitter. Unlike Twitter YouTube cache URL contains a number of attributes. In YouTube, the range is one of the main attributes to reassemble video fragments. It determines the frame order in the cached video stream. In addition, the "dur" attribute refers to the whole length of the video, not the amount of cached video. In Twitter cached URL, the value of the cached video clip is always start with zero. The cached video in the example has a resolution of 720*720 pixels Figure 7. Each cache entry maintains a Multipurpose Internet Mail Extensions (MIME) type. All MIME types are managed by the Internet Assigned Numbers Authority (IANA) [9]. It is developed to support more formats in the form of image, audio, video, or executable files. Browsers use MIME to decide how to process a URL rather than file extension. For example, the MIME Type of a Video Transport Stream File (TS) format video is "video/MP2T" with "video" being the type and "MP2T" being the subtype. A slash (/) is used to separate type from subtype. The (TS) extension is found next to the video filename.

https://www.youtube.com/watch?v=**AMCssEhaQN8**

https://twitter.com/i/status/**1248223164364193795**

Figure 4. The standard URL of YouTube & Twitter

Figure 5. The cached URL of YouTube and Twitter

4.4 Reassembling

A full cache analysis is required for this process. The primary focus is to reassemble YouTube/Twitter cached fragments to build a single concatenated video file. This process contains two phases: concatenating and rebuilding.

4.4.1 Concatenating

Generally, Reassembling is based on concatenating all the fragments in sequential/chronological order. YouTube typical streams have a header frame that indicates the beginning of the video for a duration of five seconds. Starting with the header, data fragments must be concatenated in ascending order to build a single file (see Figure 6). The range variable and its associated metadata must be reassembled to determine the order of all fragments and identify their MIME types and related URLs that contain the range attribute. Reassembling without the range value is based on guessing the file order. Attempts with incomplete range or with the wrong order of stream result in a nonviewable video.

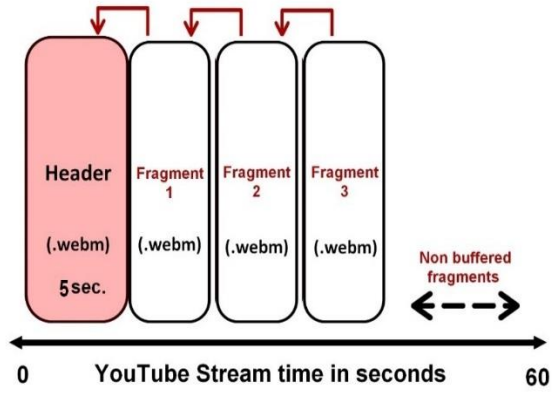

Figure 6. A reconstructed YouTube stream file structure

Twitter stream contains a frame header identifying the video beginning. Identified with 0-<number> range value via ".MP2T" MIME signature. Each stream portion is in Video Transport Stream File (TS), a video media storage format [10]. Testing indicates that the header and all fragments have a length of about three seconds. Reassembling must start in chronological order from the header file to the second, third fragment, etc., to build a single file. (See Figure 7).

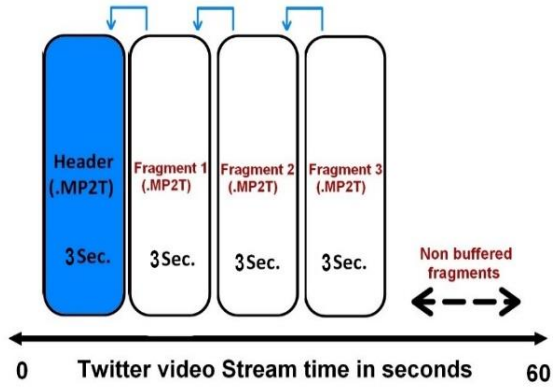

Figure 7. A structure of the reassembled Twitter stream file

4.4.2 Rebuilding

After concatenating, the fragments must be combined to a single concatenated file. This can be done using the MoviePy module. It is a Python module for joining and editing video files. After the files are merged, a single video file is built and played using MPC media player software. Reassembling fragments in the wrong order results in an unplayable stream.

5. EXPERIMENTAL RESULTS

This section evaluates the performance of the proposed technique experimentally. Both YouTube/Twitter have been tested to reconstruct videos from the Firefox cache folder. As we aim for experiments to be easily replicated by digital forensic analysts, we decided to use programs that are used in many digital forensic laboratories.

5.1 Reconstructing cached video stream on YouTube

The test video stream contained a header frame that defines the video's start. It can be detected by a range value of 0-<147011> with WebM signature as shown in (see Table 1). After reassembling all fragments using both range and ordering variables, a single built file with a new cached URL is created. This experiment offers a review of YouTube streams' impact in the Firefox cache folder. It aims to recover the video stream on YouTube using the proposed technique.

Table 1. The theoretical analysis of YouTube stream reconstruction

File Order	Range (Test video values)	File Signature
Header	0-147011	0x1A 0xF4 0xDF 0x23 09F 0x62 0x82 0x41 0x01 0x12 0xF7 0x81 0x08 0x42 0x F7 0x41 0x04 0x02 0xF4 0x81 0x08 0x42 0x82 0x84 0x77 0x65 0x62 0x6D 0x42
Fragment 1	147012-212547	-
Fragment 2	212548-354838	-
Fragment 3	354839-404630	-

- **Instruments**

The experiment was done on several machines. Each one was freshly imaged with an institutional standard operating environment Windows 10 Pro 64-bit 10.0 Build 17134. The watching session was carried out on Firefox version 74.0.1 64-bit. The forensic acquisition was carried out with MZCacheView.

- **Experiment**
  - **Setup & preparation**

The device's hard disk was forensically wiped by overwriting all sectors several times and re-imaged. This ensures that no artifacts from the previous web browser session remained. After booting, Firefox browser version 74.0.1 64-bit was downloaded using Internet Explorer browser and installed.

  - **Acquisition**

A uniquely identifiable YouTube video URL https://www.youtube.com/watch?v=DnWs_AFUrNk played on Firefox as a suitable test video. The browser was then closed, and the device was shut down, and an image was created.

  - **Steps & Analysis**

All WebM records must be exported using MZCacheView in order to reconstruct the YouTube video stream. The video was watched for five minutes. Testing indicates that there were 24 chunks with a typical naming convention for the cache file. There was only one chunk file playable from a total of 28 WebM files (See Figure 8). All the other 27 WebM chunks returned errors upon playing. To view the content of the video, all WebM entries must be concatenated in chronological order. Each WebM fragment has its URL, examined in order, starting with the header to identify the fragment order. A header frame identifies the start of the video in typical YouTube stream with WebM signature. (See Figure 9).

Filename	Content Type	URL	File Size	Fetch Count	Last Modified	Last Fetched	Expiration Time
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	537,024	1626086410	7/12/2021 12:40:10...	7/12/2021 12:40:10...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	152,666	1626086199	7/12/2021 12:36:41...	7/12/2021 12:36:41...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	542,325	1626086262	7/12/2021 12:37:42...	7/12/2021 12:37:42...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	538,320	1626086480	7/12/2021 12:41:20...	7/12/2021 12:41:20...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	418,102	1626086520	7/12/2021 12:42:01...	7/12/2021 12:42:01...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	65,536	1626086199	7/12/2021 12:36:41...	7/12/2021 12:36:41...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	554,782	1626086550	7/12/2021 12:42:31...	7/12/2021 12:42:31...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	454,213	1626086450	7/12/2021 12:40:51...	7/12/2021 12:40:51...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	534,637	1626086232	7/12/2021 12:37:12...	7/12/2021 12:37:12...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	546,614	1626086370	7/12/2021 12:39:30...	7/12/2021 12:39:30...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	68,074	1626086198	7/12/2021 12:36:41...	7/12/2021 12:36:41...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	556,981	1626086590	7/12/2021 12:43:11...	7/12/2021 12:43:11...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	529,409	1626086302	7/12/2021 12:38:22...	7/12/2021 12:38:22...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	260,970	1626086201	7/12/2021 12:36:41...	7/12/2021 12:36:41...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	2,524	1626086198	7/12/2021 12:36:38...	7/12/2021 12:36:38...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o	audio/webm	https%2Cyoutube.com%29;https://r4---sn...	550,527	1626086211	7/12/2021 12:36:52...	7/12/2021 12:36:52...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	1,425,978	1626086563	7/12/2021 12:42:44...	7/12/2021 12:42:44...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	420,371	1626086199	7/12/2021 12:36:41...	7/12/2021 12:36:41...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	3,780	1626086198	7/12/2021 12:36:41...	7/12/2021 12:36:41...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	1,466,855	1626086488	7/12/2021 12:41:29...	7/12/2021 12:41:29...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	4,096	1626086198	7/12/2021 12:36:38...	7/12/2021 12:36:38...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	65,536	1626086199	7/12/2021 12:36:41...	7/12/2021 12:36:41...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	1,420,831	1626086228	7/12/2021 12:37:09...	7/12/2021 12:37:09...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	1,140,117	1626086207	7/12/2021 12:36:47...	7/12/2021 12:36:47...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	1,405,506	1626086292	7/12/2021 12:38:12...	7/12/2021 12:38:12...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	1,362,613	1626086414	7/12/2021 12:40:15...	7/12/2021 12:40:15...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	3,780	1626086198	7/12/2021 12:36:38...	7/12/2021 12:36:38...	N/A
expire=1626107785&ei=KRvsYNbOGrPCmwfc2YRw&ip=196.219.94.96&id=o.mp4	video/mp4	https%2Cyoutube.com%29;https://r4---sn...	59,258	1626086199	7/12/2021 12:36:41...	7/12/2021 12:36:41...	N/A

Figure 8. All 24 Webm cached fragments on MZCacheView

https://r4---sn-hgn7rne7.googlevideo.com/videoplayback?expire=1610633867&ei=K_7_X9uvHpTmxwLIha6YBw&ip=41.237.155.183&id=o-ALqI67wtO9JNjVVUIgm2Qo_WejApBpH3bC2AP-FjuHs3&itag=244&aitags=133%2C134%2C135%2C160%2C242%2C243%2C244%2C278%2C394%2C395%2C396%2C397&source=youtube&requiressl=yes&mh=2g&mm=31%2C29&mn=sn-uxaxjvhxbt2u-5atr%2Csn-hgn7rne7&ms=au%2Crdu&mv=m&mvi=4&pl=19&gcr=eg&initcwndbps=411250&vprv=1&**mime=video%2Fwebm&ns=7V6pTMMnfqQleRhFLCrc5AUF&gir=yes&clen=47454923&dur=1327.526**&lmt=1540907862961675&mt=1610611968&fvip=4&keepalive=yes&c=WEB&txp=5432432&n=D32W9xpghVuuMw&sparams=expire%2Cei%2Cip%2Cid%2Caitags%2Csource%2Crequiressl%2Cgcr%2Cvprv%2Cmime%2Cns%2Cgir%2Cclen%2Cdur%2Clmt&sig=AOq0QJ8wRAIgJ_XbaE_M4fhGuy8aSW_5j_ePI2xQWKVt5apsejdQCnQCIDLzAP39Co6ZKqE3nfqrULuVje0ZCkA36npy_H53aPI&lsparams=mh%2Cmm%2Cmn%2Cms%2Cmv%2Cmvi%2Cpl%2Cinitcwndbps&lsig=AG3C_xAwRAIgXfTtkhvWvHQInwhtU4M-hxrSdL9jNZdjGCRUmDYqRvMCIFdUBmijPeCu8Dj2k6QI-i4jgarko4Rc4V_Af838Vf9V&alr=yes&cpn=Q3FvKN1wRJ6n2AoF&cver=2.20210112.08.00&fallback_count=1&**range=0-147011**&rn=8&rbuf=0

Figure 9. The cached URL of the header chunk

5.2 Reconstructing cached video stream on Twitter

The second experiment aims to analyze Twitter streams' impact in the Firefox cache and build a single viewable video.

- **Instruments**

The experiment was done on several machines, following the same steps as the previous study.

- **Experiment**
  - **Setup & preparation**

The same measures as in the first experiment have been taken.

  - **Acquisition**

A standard Twitter video URL https://twitter.com/i/status/1248223164364193795 played on Firefox. After that, the browser was closed, and the device was turned off and imaged.

  - **Steps & Analysis**

As a precaution, to avoid contamination by existing data, the cache folder was verified as empty. Using MZCacheView, thirty-nine fragments with ".MP2T" MIME type have been collected and exported (see Figure 10). The initial test indicates that all files are running, playing about three seconds from the video stream.

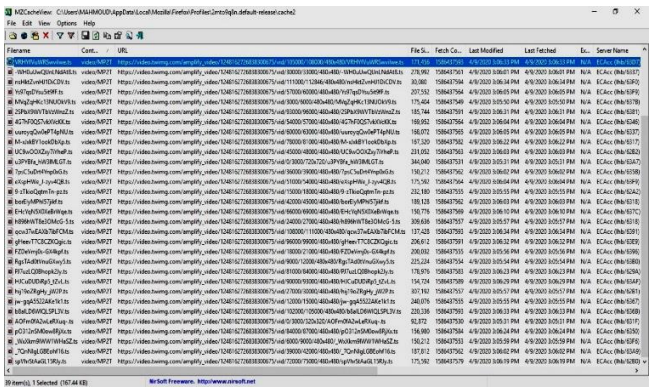

Figure 10. All 39. MP2T cached video chunks on MZCacheView

As previously, all fragments must be concatenated correctly to reconstruct the video stream. The associated URL of each ".MP2T" cache entry is required. The header frame has a starting value of 0-<3000>. Both the range order and the creation date & time attributes are used to determine frames' order (see Figure 11). All data fragments must be concatenated in sequential order, starting with the header. Reassembling without the range value is unsuccessful and is likely based on guessing the fragment order. After concatenating, A single video file is built using Shotcut software and viewed using MPC-HC media player [11] (see Figure 12).

```
Filename          : AOFm0fA2wLeRXuq-.ts
Content Type      : video/MP2T
URL               : https://video.twimg.com/amplify_video/1248162726838300675/vid/0/3000/320x320/AOFm0fA2wLeRXuq-.ts
File Size         : 92,872
Fetch Count       : 15864437530
Last Modified     : 4/9/2020 3:05:31 PM
Last Fetched      : 4/9/2020 3:05:31 PM
Expiration Time   : N/A
Server Name       : ECAcc (lhb/631F)
Server Response   : HTTP/2 200 OK
Server Time       : 4/9/2020 3:06:40 PM
Server Last Modified: 4/9/2020 10:13:58 AM
Content Encoding  :
Cache Name        : F9E88BB5C2DFFAF81F5248912ADFC697C82AFE2A
Subfolder         :
Missing File      : No
Cache Control     : max-age=604800, must-revalidate
ETag              :
URL Length        : 96
```

Figure 11. The characteristics of Twitter header fragment

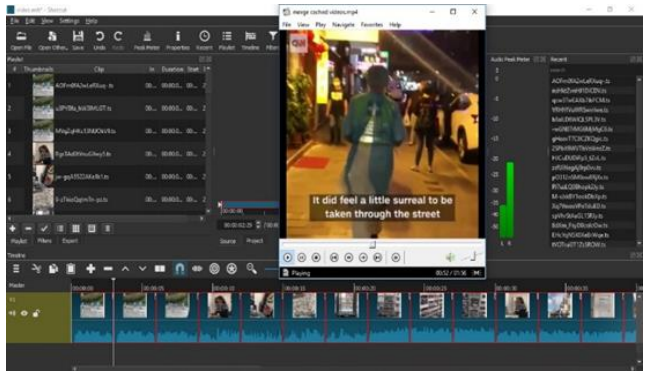

Figure 12. Playing a single built video file by the MPC-HC media player

Two case studies are presented within this paper's scope, an examination of YouTube/Twitter video streams. The experimental results show that the cached video from the installed Firefox can be reconstructed. A table with a summary of the main experimental results is shown in Table 2. The results obtained can be used to examine other streaming services and web browser cache characteristics. To test the scalability, we conducted some experiments on a larger scale with different machines' scenarios. The goal is to evaluate the efficiency of the proposed technique and examine its shortcomings. Over 100 experiments were performed using the proposed technique on YouTube and Twitter. Five key scenarios have been applied for multiple videos of varying lengths and duration. The process of reconstructing cached videos has begun at different times. (See Table 3).

Table 4 shows the experimental results that have been made on YouTube/Twitter. The table's main metrics are video duration, the reconstructed duration, rebuilding process starts, and usage scenario. Every experiment has its playing scenario.

Adblock Plus [12] add-on has been installed and enabled to block ads and maintain the experiment's efficiency temporarily.

Table 2. A comparison between YouTube/Twitter cache characteristics

Comparison	YouTube	Twitter
The possibility to recover and view streamed video content	Yes	Yes
The capability to figure out how much of a video has been played	Yes	Yes
MIME Type	WebM	MP2T
The duration of the cached fragments	Only the header file plays five seconds of the streaming video. The rest returns an error upon playing.	Each chunk plays for about three seconds of the streaming video.
The main steps of the reassembling process	• MIME attribute. • The range ordering variable • The last accessed date and time.	

Table 3. A statistical list of cached video reconstruction experiments on YouTube/Twitter

Comparison		Total No. of Videos	
		YouTube	Twitter
		52	48
Video Duration	Video ≤ 30 Minutes	10	25
	31 Minutes ≤ Video ≤ 119 Minutes	30	16
	Video ≥ 120 Minutes	19	12
Video Playing Scenario	Normal watching	15	14
	Skipping	8	10
	Pausing and resuming	9	13
	Commercial Ads enabled	13	8
	Closed Captioning (CC) Enabled	7	8

Table 4. Test analysis for YouTube/Twitter stream reconstruction with the usage scenario

#	URL	Video Duration	Watching Duration	Buffered Bar Time	The length of the recovered	MIME Type	No. of Fragments	Rebuilding process start	Video Playing Scenario
1	**https://www.youtube.com/watch?v=ISQVx3tntqM**	**21:41**	**21:41**	**21:41**	**21:41**	**WebM**	**70**	**After 5 minutes from watching session**	**Normal watching**
2	**https://www.youtube.com/watch?v=mVcZsJpuxec**	**21:44**	**18:22**	**19:12**	**19:31**	**WebM**	**41**	**After 12 hours from watching session**	**Skip from 2:15 to 18:00**
3	**https://www.youtube.com/watch?v=OUYqQcHKtZ8**	**52:58**	**42:12**	**44:13**	**44:19**	**WebM**	**255**	**After 3 days from watching session**	**Pause and resume**
4	**https://www.youtube.com/watch?v=yE9TZxevU34**	**52:29**	**52:29**	**-52:29**	**52:29**	**WebM**	**367**	**After 7 days from watching session**	**Commercial Ads enabled**
5	**https://www.youtube.com/watch?v=TLpbfOJ4bJU**	**42:25**	**06:36**	**07:47**	**42:25**	**WebM**	**29**	**After 5 hours from watching session**	**Closed Captioning (CC) Enabled**
6	**https://twitter.com/engineers_feed/status/1342801199444193280**	**00:07**	**00:07**	**00:07**	**00:07**	**MP2T**	**3**	**After 5 minutes from watching session**	**Normal watching**
7	**https://twitter.com/i/status/1251243294157348864**	**02:17**	**02:07**	**02:17**	**02:17**	**MP2T**	**79**	**After 3 days from watching session**	**Skip from 0:15 to 01:30**
8	**https://twitter.com/Reuters/status/1343174792699064320**	**01:21**	**00:40**	**00:48**	**00:50**	**MP2T**	**18**	**After 12 hours from watching session**	**Pause and resume**
9	**https://twitter.com/CNN/status/1342917583859806208**	**03:58**	**01:10**	**01:21**	**01:46**	**MP2T**	**28**	**After 7 days from watching session**	**Commercial Ads enabled**
10	**https://twitter.com/i/status/1319292428575232000**	**02:57**	**02:57**	**02:57**	**02:57**	**MP2T**	**60**	**After 24 hours from watching session**	**Closed Captioning (CC) Enabled**

6. RESULT AND DISCUSSION

The experiments' results have provided us with valuable insights into improving the proposed technique. It is noted that the rebuilding process takes time depending on the cached video duration and the analyst performance. The following significant observations were recorded after performing the experiments: the entire first and fifth videos have been cached and reconstructed after five minutes of viewing with no errors. In the 1st experiment, 70 fragments, 49 audio fragments, and 21 video fragments with WebM type were found. In the 6th experiment, the test video has been recovered after reassembling three fragments with TS extension. When there are multiple watched videos cached on the disk, it would be identified by date and time, and range attributes. In the 2nd and 7th experiments, a skipping video playback scenario has been implemented at various times. The entire videos have not been cached, but only what has been watched before and after skipping was reconstructed. The skipped portions during online watching have not been cached. In 3 and 8, the test videos have been paused while playing multiple times. The aim is to consider the effect of this scenario on the order of the fragments being cached. It has been examined that there is no negative impact on the cached stream. All the cached video fragments have been rebuilt and played without disruption. Before beginning the 4th and 9th experiments, Adblock Plus add-on has been disabled. Afterward, the online test videos have started with ads to figure out how the cached ad video could be recovered from the cache folder. The ads have been verified to be located in the same cache folder with a different URL. It was easy retrieval using the proposed technique. Closed Captioning (CC) [13] has been enabled in 5 and 10 during watching to see whether or not the subtitle displays after the reconstruction. It has been found that the cached videos have been rebuilt with no subtitles. It has also been detected that the sound plays well after reconstructing, even if the online video is mute.

7. CONCLUSION AND FUTURE WORK

This paper is considered one of the preliminary contributions in the video reconstruction field. As online video streaming becomes more popular, there's a risk of abuse of this technology. This work provides a basis for local video stream recovery using the Firefox browser. A proposed technique is presented for reconstructing cached videos. It implements a unique way of extracting fragments without compromising on accuracy and efficiency. Two experiments were made to demonstrate the feasibility of the proposed technique. The procedures for testing as well as the results are offered. The proposed technique applies to a forensic analysis of various streaming platforms. There is still a scope of improvement in this technique. This work aims to enable forensic analysts to ensure effective video reconstruction. It would also be a compelling resource for law enforcement, digital forensic experts, and the academic community of digital forensics. After testing, it is possible to reconstruct the unwatched content when the user loads a video and pauses it. Future work involves expanding the framework in two possible directions. First, extending the analysis into mobile browsers and other applications such as Twitter, YouTube, etc. Second, testing various streaming services require further research, such as Nimo, TikTok, Dailymotion, etc.

REFERENCES

[1] Zhou, C.F. (2017). Handbook of Research on Creative Problem-Solving Skill Development in Higher Education: Advances in Higher Education and Professional Development. Denmark: IGI Global. https://doi.org/10.4018/978-1-5225-0643-0

[2] IWF. (2018). Trends in Online Child Sexual Exploitation: Examining the Distribution of Captures of Live-streamed. Annual Report, Cambridge: The Internet WatchFoundation.https://www.basw.co.uk/system/files/resources/Distribution%20of%20Captures%20of%20Live-streamed%20Child%20Sexual%20Abuse%20FINAL.pdf.

[3] Horsman, G. (2018). Reconstructing streamed video content: A case study on YouTube and Facebook Live stream content in the Chrome web browser cache. Digital Investigation, 26: S30-S37. https://doi.org/10.1016/j.diin.2018.04.017

[4] Schaap, E., Hoogendoorn, I. (2013). Reconstructing Web Pages from Browser Cache. University of Amsterdam, Amsterdam: Neverlands Forensics Institute.

[5] Marrington, A., Baggili, I., Al Ismail, T., Al Kaf, A. (2013). Portable web browser forensics: A forensic examination of the privacy benefits of portable web browsers. In 2012 International Conference on Computer Systems and Industrial Informatics (ICCSII 2012). Piscataway, pp. 1-6. https://doi.org/10.1109/ICCSII.2012.6454516

[6] Shashidharm, N., Ohana, D.J. (2013). Do private and portable web browsers leave incriminating evidence? A forensic analysis of residual artifacts from private and portable web browsing sessions. Proceedings of the 2013 IEEE Security and Privacy Workshops. 1730 Massachusetts Ave., NW Washington, DCUnited States: IEEE Computer Society, pp. 135-142. https://doi.org/10.1109/SPW.2013.18

[7] Statcounter. (2021). Browser Market Share Worldwide. May 11. https://gs.statcounter.com/browser-market-share#monthly-202006-202105-bar, accessed on June 19, 2021.

[8] NirSoft. (2021). MZCacheView v2.02 - View the cache files of Firefox Web browsers. June 11. https://www.nirsoft.net/utils/mozilla_cache_viewer.html, accessed on June 19, 2021.

[9] MIME Media Types. June 11. https://www.iana.org/assignments/media-types/media-types.xhtml, accessed on June 19, 2021.

[10] FileInfo. (2021). TS File Extension. June 11. https://fileinfo.com/extension/ts, accessed on June 19, 2021.

[11] Codec, MPC-HC. (2021). MPC-HC media player program version 1.7.13.112. June 11. https://mpc-hc.org/, accessed on June 19, 2021.

[12] Adblock Plus. https://addons.mozilla.org/en-US/firefox/addon/adblock-plus, accessed on June 16, 2021.

[13] Closed Captioning. https://en.wikipedia.org/wiki/Subtitle, accessed on June 16, 2021.