

## Live-Streamed Video Reconstruction for Web Browser Forensics

Mahmoud El-Tayeb<sup>1\*</sup>, Ahmed Taha<sup>1</sup>, Zaki T. Fayed<sup>2</sup>

<sup>1</sup> Faculty of Computers & Artificial Intelligence, Benha University, Benha 13518, Egypt

<sup>2</sup> Faculty of Computer & Information Sciences, Ain Shams University, Cairo 11566, Egypt

Corresponding Author Email: [mahmoud.eltayeb18@fci.bu.edu.eg](mailto:mahmoud.eltayeb18@fci.bu.edu.eg)



<https://doi.org/10.18280/isi.270107>

### ABSTRACT

**Received:** 24 December 2021

**Accepted:** 23 February 2022

#### Keywords:

*digital forensics, browser cache, live-streams, Nimo TV, YouTube live, google chrome, Firefox, chromium*

The way we use video streaming is evolving. Users used to broadcast their videos on social media platforms. These platforms enable them to interact from anywhere they want. Recently, there has been a wide range of people who use live video streaming platforms regularly. Thanks to high-speed Internet connections, live video streaming is now easier than ever. Many of these platforms broadcast live video feeds of electronic games, so young streamers use them to make money. Live streaming refers to media that is simultaneously broadcasted and recorded online in real-time. Despite the growing popularity of these platforms, there is a risk that this technology will be abused. Several other recorded cases of abuse have resulted in the emerging popularity of live streaming platforms. Many criminal and public proceedings may rely on information linked to a normal Web user's Online activity. Examining the web browser's history or cache may reveal helpful information about the suspect's activities. The evidence can reveal keys that might lead to this individual being convicted or clear. This work continues what was previously done to reconstruct cached video streams from YouTube and Twitter on Firefox. Our main aim in this paper is to examine data from a cached live stream using YouTube Gaming/Live and Nimo TV on Firefox and Chromium browsers.

## 1. INTRODUCTION

At the dawn of the Internet, if a web admin wished to add videos to his website, he was forced to post them as a link. Web-users then had to download the file completely before playing it back. Now, video has been served so that files can be played almost immediately after the file starts to be downloaded. Today's technology allows even the normal user to record a video and post it on his social media page. Due to the evolution of this technology, the user can now perform a live broadcast at any time from any place. Live streaming simply allows him to "go live" with some clicks on a large platform. Platforms such as YouTube Live and Facebook Live provide free access to any user alongside specialized gaming sites such as Twitch and Nimo. Besides, Smartphones achieve an excellent live streaming experience instead of using costly equipment. It has become more popular amongst the public due to its availability and its infinite options. It also eliminated boundaries and allowed people to use live streaming fully. In addition, there was an increase in demand for it among teenagers and children during the Corona epidemic. However, these technologies in their infancy break barriers with their range of applications. Its usage has increased significantly since 2015 [1]. In 2018, the Ofcom organization revealed that people 16-34 years old most likely to switch to streaming video platforms than traditional TV [2]. Viewers spend at least eight times more time watching live videos on traditional TV [2]. While online streaming platforms allow many to interact in benign activities, a minority of people continue to abuse these platforms. In some cases, it can be seen on the surface the abuses of this technology. Child abuse videos were

spotlighted by the Internet Watch Foundation (IWF) in 2018 [3] as a main source of abuse and as a means of capturing and sharing images. Sexual offenders may use trickery, sexualized games, or fake clips to manipulate children and young people into undressing or creating self-generated indecent images while live streaming [3]. On January 15, 2014, 29 people were arrested for an international case of live webcam child abuse [3]. In exchange for payment, the criminal organization arranged for children to be sexually abused live on a webcam in the Philippines. The use of webcams to stream live abuse, particularly from the developing world, is a significant and emerging threat, according to the NCA's CEOP command [3]. Terrorist attacks and killings cases have increased on live streaming platforms recently. On March 15, 2019, a terrorist shot dead 51 Muslims in two Christchurch mosques in New Zealand [4]. A total of 51 innocent prayers was killed, and another 40 were injured [4]. Before his attack, he published the first shooting on Facebook Live and Twitch. As a result, the video spread quickly across the Internet. The video and its illegal and hateful content were streamed on Twitch as part of the 'Artifact' digital card game. Spammers targeted the section after the game was named the site's lowest rated. Including the video, there was hate speech directed at Muslims in the comments section. Another case In Saxony-Anhalt, Germany, two people were killed and two others injured in a fire shooting on October 9, 2019 [4]. The shooting was live-streamed on Twitch. On March 23, 2021, in a mass shooting in USA, a man shot and killed ten people. A lived shooting clip at a supermarket was streamed on YouTube Live by a spectator [4]. From these incidents, we need to develop a technique to analyze and reconstruct live streams. To detect any potentially

streaming content, forensic examination may be required. Investigators can exploit information about a person, their activities, and their actions on social media sites as a potential tool to trace down a crime. Since 2015, the use of social media information has been increasing considerably [3]. It is important to manually examine and retrieve data from a suspected machine as part of a digital investigation, as well as perform event reconstruction. The proposed technique includes reconstructing live video fragments stored in the cache watched live on Nimo TV and YouTube Live/Gaming. This model also effectively retrieves the stored video fragments from YouTube Live/gaming and the playback videos on Nimo TV. Unfortunately, the cached video files could not be retrieved during the live broadcast of Nimo TV due to its limitation of Caching during the live broadcast of the video being watched. Many experiments were made to demonstrate the feasibility of the proposed techniques. The impact of live video streaming on the browser cache streams of installed Google Chrome, Mozilla Firefox, and Opera is examined in this work. This study uses an experimental methodology to reconstruct YouTube Live and Nimo stream platforms forensically. Stream reconstruction methodologies are provided where results indicate that where live-stream video has been played, it is possible to reassemble buffered video stream data to build a viewable video clip. Both testing procedures and results are provided. The following is the structure of this research paper: The previous and related web browser cache and video reconstruction work is described in Section 2. Section 3 depicts the features of Chromium (v92.0.4515.159), Mozilla Firefox (v91.0.2), and the contents of their cache structures. The proposed cached live video reconstruction technique is discussed in Section 4 using YouTube Gaming/Live and Nimo as a streaming video platform. The implementation details and trials we conducted on these streaming platforms are presented in Section 5. Section 6 concludes the paper with several unanswered questions, future work, and discussion.

## 2. RELATED WORK

Although there is a large number of studies in digital forensics, there are just a few studies on video reconstruction analysis that have been published. Previously, we offered a foundation for recovering local video streams using the Firefox browser. A method for reconstructing cached videos has been presented. We developed a technique of extracting fragments while maintaining the video's efficiency and accuracy. Graeme Horsman [5] has provided a framework to evaluate video streams locally. He highlights analysis techniques for identifying both violent and underground communities' videos. He offered two case studies, one on YouTube and the other on Facebook Live, as a method of identifying and validating video content based on "single file viewing." He used the Chrome browser as a platform for accessing and streaming video content. Horsman [6] also provided some insight into six additional platforms: Twitch, Ustream.tv, Mixer, Smashcast.tv, Facebook Live, and Younow. He provides the analysis of localized stream caching with methodologies. Horsman published a detailed digital forensic study of live broadcasting Periscope in 2018 [7]. It provided a process for Periscope forensic experts to employ while looking into incidents of abuse through the application. The authors [8] tested whether a post-processing approach

might be used to recreate a web page from browser cache without altering the evidence. Their research aims to improve understanding of online page reconstruction using browser cache. They also demonstrated pre- and post-processing approaches for rebuilding websites; however, they were unable to reconstruct cache-case video stream information. Marrington et al. [9] developed an experimental methodology for forensically analyzing and examining both installed and portable web browser artifacts. On both installed and portable web browsers, their experiment did not show how to rebuild video stream content. Their study was unable to recover the contents of video streams stored in browser cache.

## 3. METHODOLOGY

In this section, we'll look at how Web caching works in the Firefox and Chromium browsers. It outlines the most critical obstacles that live reconstruction from various Web browsers faces.

**Browser Cache.** The web browser is a tool that allows Internet users to view web applications and web sites. Browser caching is a method that saves the retrieved files from visited websites to a specific location on a local device for later use. The visited web pages can be loaded much quicker when this page is visited once more at a later time. The web browser compares the online web page's data with the one held in the cache folder. If this web page has not changed, its cache or parts will be used, and the page will be downloaded, displayed, and most likely cached once more. The web cache is saved on that specific location even after the browser is closed. Although Google Chrome and other Chromium-based browsers Edge, Opera, etc., use the same core engine, they differ in many ways. Each browser has significant differences and unique options that make this choice more than just about branding.

### 3.1 Google Chrome

Chrome is a Google-developed freeware web browser that uses the blink layout engine [8]. Chrome manages web caches with at least five files. One is the index file, and there must be at least four data files entitled data n, where n is the file number that begins with zero (see Figure 1).

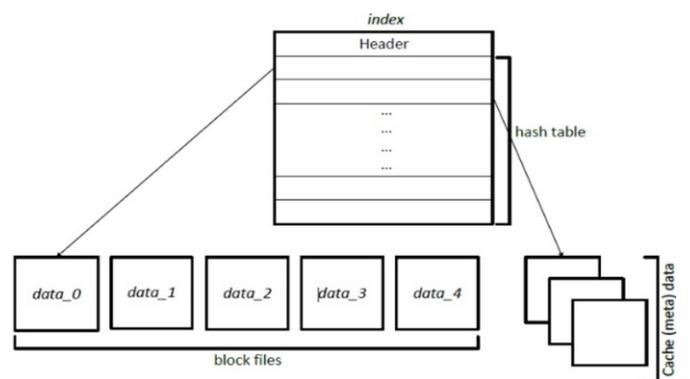


Figure 1. Chrome file cache structure [8]

If any of these files are missing or corrupted for whatever reason, it will be rebuilt. The index file contains a hash table that is used to locate cached file entries. The data files contain information about HTTP headers as well as data about a

specific request. These files are also known as block files since the file format is designed to hold data in fixed-size blocks. A block file can store up to 256 bytes of data in blocks, with the data stored across one to four of these blocks. When the cached data exceeds 16 KB in size, the web cache data will no longer be placed within a conventional regular block file, but rather in a separate file with no special headers and is just the raw data we are saving/caching. Typically, the name is f\_xx, where xx is a hexadecimal number that identifies the file.

### 3.2 Opera and Microsoft Edge

Since 2013, Chrome and Opera have shared the same engine, so they both load web pages in the same way. Opera announced that its desktop, mobile, and embedded web browsers switched from its closed source Presto engine to Blink. Blink is the same rendering engine used by Chrome and Safari. Besides, Microsoft Edge uses Chrome's engine as its base. It's been two years since Edge Chromium browser was released, and the company has been steadily adding new features to the cross-platform browser.

### 3.3 Mozilla Firefox

Mozilla Firefox is a free and open-source web browser. Firefox uses a Gecko layout engine [10]. All changes we make in Firefox, including passwords and bookmarks, are saved in a profile folder. The cache folder is made up of three main types of files that reconstruct the cached data. There are three cache block files, a cache map file, and separate cache data files. The cache map file will be the primary file used to rebuild web pages using Firefox Cache data (see Figure 2). Table 1 shows a comparison between different actively developed browsers engines. We discover that Safari browser works on Apple IOS, which runs on Webkit engine and supports most operating systems, whereas the Google Chrome,

Firefox, and Flow browsers all work on the same operating systems, but they use different engines such as Blink, Gecko, and Flow in that order. we also note that the Pale Moon browser, which uses Goanna engine, however does not support macOS or Android.

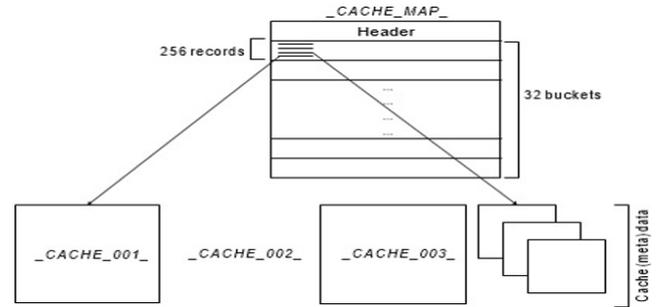


Figure 2. Mozilla Firefox file cache structure [8]

**Live video reconstruction.** Video reconstruction is an essential phase in digital analysis process. It is the process of putting together live pieces of evidence during the early stages of an investigation to improve understanding of what happened. With the proliferation of live streaming websites, it is becoming increasingly important to develop live video reconstruction techniques. This paper examines whether it is possible to recover live-streamed video content. There are some difficulties when working with the Chromium and Firefox browsers. One of them is that no previous reconstruction experiments on YouTube Gaming / Live and Nimo TV on any browser have been conducted. Aside from the forensic tools used in recovery, Chromium's cache file structure differs from Firefox's structure. The tools used in Chromium are not the same as in Firefox. The technique of the proposed cached video reconstruction is discussed in the following section.

Table 1. Comparison between actively developed browsers engines

#	Engine	Host	Embedded in	Supported Operating Systems
1	WebKit	Apple	Safari and iOS browsers	Windows, macOS, iOS, Android, Linux, and BSD
2	Blink	Google	Google Chrome, Opera and Microsoft Edge, and all other Chromium-based browsers	Windows, macOS, Android, Linux, and BSD
3	Gecko	Mozilla	Firefox browser and Thunderbird email client	BSD
5	Flow	Ekioh	Flow browser	
4	Goanna	M.C.Straver	Pale Moon and Basilisk browsers, K-Meleon browser beginning with version 76.2G	Windows, Linux, and BSD

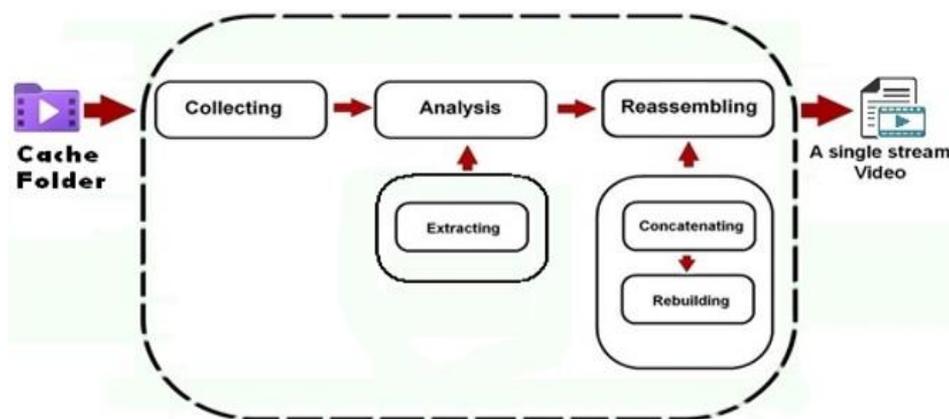


Figure 3. The proposed cached video reconstruction technique

#### 4. THE PROPOSED TECHNIQUE

The general stages of reconstructing live cached video files from Chromium/Firefox browsers are represented in Figure 3. The proposed technique is divided into three major phases: collecting, analyzing, and reassembling. Each phase is depicted in the following subsections.

**Watching session.** On a PC, Using Firefox / Chromium browsers, a brief watching session was conducted. According to the initial test, there is no buffering process during live streaming (see Figure 4). After disconnecting the Internet connection, most of the buffered portions of the stream can be replayed. (see Figure 5).

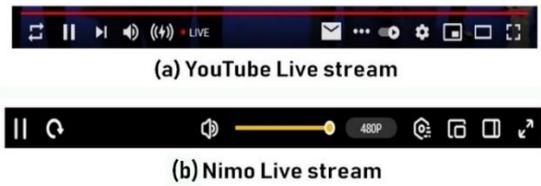


Figure 4. An example of a live stream



Figure 5. An example of buffered stream

#### 4.1 Collecting phase

Generally, Web browsers store user information in four different sections: cache, cookies registry, history records, and downloaded files [8]. The goal of this phase is to find the location of the browser's cache folder on the local disc. After locating the cache folder, the analysis of its contents begins. The cached video is temporarily stored as fragments on the local hard disk in the default location of any browser. Figure 6 shows the locations on multiple operating systems where web browsers save data.

Web Browser	Cache folder path	Operating System
Google Chrome	C:\Users\%username%\AppData\Local\Google\Chrome\User Data\Default\Preferences	Windows 7 and latest versions
Microsoft Edge	C:\Users\%username%\AppData\Local\Microsoft\Windows\Temporary Internet Files\	
	C:\Users\%username%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\	
Firefox	C:\Users\%USERNAME%\AppData\Roaming\Mozilla\Firefox\Profiles\%PROFILE%\default\places.sqlite	
Safari	C:\Users\%username%\AppData\Roaming\Apple Computer\Safari\	
Opera	C:\Users\%username%\AppData\Roaming\Opera\Opera\	

Figure 6. Locations on OSs were web browsers store data

#### 4.2 Analysis

The primary goal is to examine the properties of each extracted live cached fragment URL. This process fully depends on the extracting phase.

#### 4.2.1 Extracting

During this stage, the analyst employs a suitable tool to investigate the contents of the browser cache folder based on its structure. The extracted fragments are saved in a separate folder for the next step.

#### 4.3 Reassembling

The main goal is to reassemble the cached live streaming pieces into a single rebuilt video file. This procedure is divided into two stages: concatenating and rebuilding.

#### 4.3.1 Concatenating

Concatenating all fragments in sequential order is the basis for reassembling. Beginning with the header, fragments must be concatenated in an ascending order to a second, third, etc. to create a single concatenated file. To specify the order of all fragments, reassembling must be done using the range variable and its associated metadata. Reassembly is based on estimating the ordering of files. Attempts with an incomplete range or the incorrect stream order result in an unplayable video. While replayed YouTube stream restoration has previously been addressed, live broadcasts are now properly considered. A header frame denotes the beginning of the video in the YouTube Live / Gaming stream. It is identifiable via .MP4 MIME signature with a range value of 0-<number>. Testing indicates that the header and all fragments have a length of about five seconds. Reassembling must start in sequential order from the header file to the second, third fragment, ...etc. to build a single file (See Figure 7). Testing indicates the performance is the same for Firefox and all Chromium browsers.

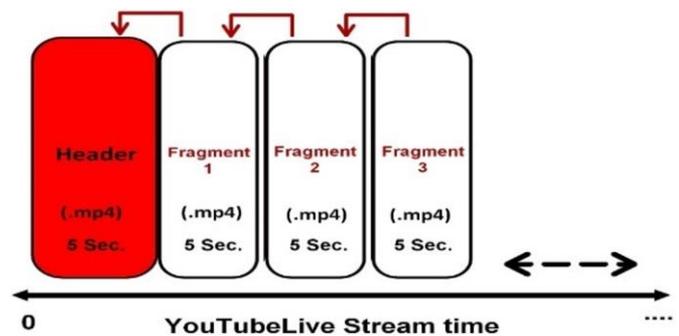


Figure 7. A structure of the reassembled YouTube Live stream

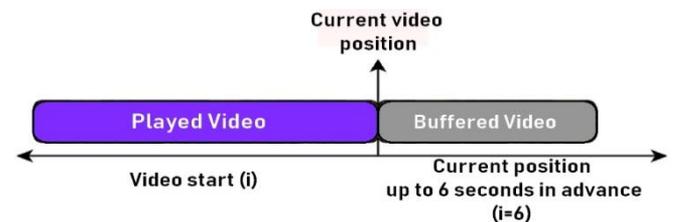


Figure 8. A breakdown of Nimo stream buffering

Testing indicates that Nimo live streams cannot be reconstructed. Users of this platform have no control over the live streams they are watching since no buffering occurs. Replayed Nimo stream has a header frame that identifies the beginning of the video. It is identifiable via .MP2T MIME signature with a range value of 0-<number>. Each stream fragment is preserved in the Video Transport Stream File format (TS). According to testing, the header and all parts have

a length of about six seconds (See Figure 8). To create a single-stream file, reassembling must start in chronological order, from the header fragment to the second, third, and so on. Testing indicates that Firefox and all Chromium browsers also have the same performance.

## 5. EXPERIMENTAL RESULTS

This section conducts an experimental evaluation of the proposed technique's performance. YouTube Live and Nimo have been tested for reassembling videos from the cache folder of Firefox/Chromium. Over 100 experiments were performed using the proposed technique on YouTube Live and Nimo. Many scenarios such as: normal watching, skipping videos, pausing & resuming, commercial ads. and closed captioning (cc) have been applied for multiple videos of varying lengths and duration. Reconstructing cached videos has begun at different times. Before starting the experiment, Adblock Plus [10] add-on was installed and enabled to block ads to maintain the experiment's efficiency.

### 5.1 YouTube gaming / live

YouTube Gaming competes with Twitch and Facebook Gaming by providing a platform for livestreaming. Testing indicates that YouTube Gaming has the same cache structure as YouTube Live. The tools available for YouTube Gaming are the same as those available for YouTube Live.

#### 5.1.1 Live broadcasts (Online stream)

YouTube stream content is cached in .mp4 format (video header: 0x00 00 00 1C 66 74 79 70 64 61 73 68) and .m4a (audio header: 0x00 00 00 18 66 74 79 70 64 61 73 68) formats (See Figure 9). The first test reveals that all fragments can be played separately for about five seconds after the video stream has ended. To reconstruct the video stream, all fragments must be correctly concatenated. Starting with the header, it must be concatenated in the right order. Without the range value, reassembling fails and is most likely based on guessing the fragment order. After concatenation, Shotcut software is used to create a single video file, which is then viewed with an MPC-HC media player. The performance of the Chromium and Firefox browsers are the same.



Figure 9. A snapshot of cached content during YouTube Live/Gaming stream

#### 5.1.2 Replayed video

After the live broadcast ends on YouTube Live, users can save their content on the platform. Cached video fragments are stored in the browser cache as videos are replayed. The order

of the streamed content must be evaluated by examining the cached fragments' last accessed time and ordering them in ascending order. Typical YouTube streams have a five-second header frame that indicates the start of the video (See Figure 10). To create a single file, data fragments must be assembled in ascending order, starting with the header.

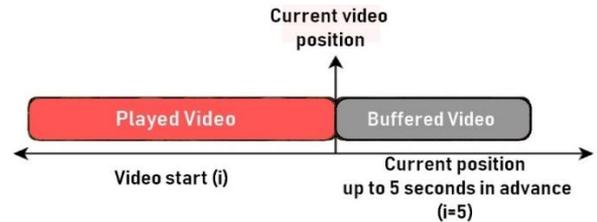


Figure 10. A breakdown of replayed YouTube stream buffering

### 5.2 Nimo TV

Nimo is one of the most extensive Chinese videos live streaming services. At the end of 2019, it had 150 million monthly active users. Nimo also has live streams for a variety of other genres, such as cooking, traditional sports, and reality shows.

#### 5.2.1 Live broadcasts (Online stream)

There is no video stream content stored in the browser cache when a user watches a Nimo live broadcast. Only images (jpeg/png/gif) and chat are cached on the web (See Figure 11). This is due to the lack of buffering, as the content is transmitted in real-time. Content is not buffered locally when a stream is paused, as it is when the stream is resumed. A user is transported to the current transmit position of a stream. The problem affects Firefox as well as all Chromium browsers and caches.

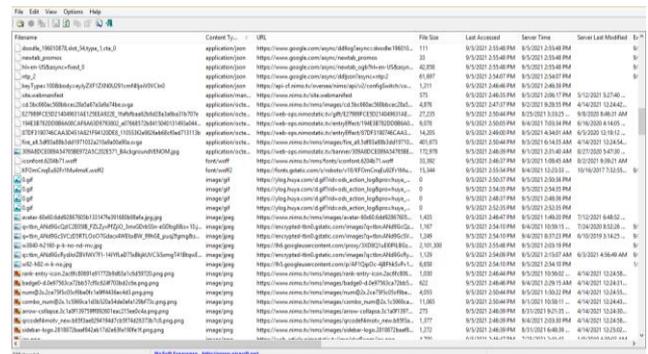


Figure 11. A snapshot of cached content during Nimo live video stream

#### 5.2.2 Replayed video

After the live broadcast has ended, Nimo users can watch their live broadcast again on the platform. The local browser cache stores video fragments that have been cached. To reconstruct the video stream, all fragments must be correctly concatenated. The order of fragments is determined by the creation date and time attributes (see Figure 12). Starting with the header, all data fragments must be concatenated in order. After concatenation, Shotcut software is used to create a single video file, which is then viewed with an MPC-HC media player [11].

Filename	Content Type	URL	File Size	Last Accessed	Service Time	Service Latency	File Size
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	181,18	9/5/2021 2:18 PM	9/5/2021 2:17 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,046,704	9/5/2021 2:18 PM	9/5/2021 2:17 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	181,190	9/5/2021 2:18 PM	9/5/2021 2:17 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	181,240	9/5/2021 2:18 PM	9/5/2021 2:17 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	181,290	9/5/2021 2:18 PM	9/5/2021 2:17 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,058,628	9/5/2021 2:18 PM	9/5/2021 2:18 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	181,779	9/5/2021 2:18 PM	9/5/2021 2:18 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,062,278	9/5/2021 2:18 PM	9/5/2021 2:18 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	953,520	9/5/2021 2:18 PM	9/5/2021 2:18 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	181,684	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,047,536	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,075,380	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,101,402	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	181,624	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,054,402	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,058,882	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	838,458	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	181,228	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,082,280	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,082,780	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,082,780	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,082,280	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,111,008	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,021,998	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	1,022,288	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	187,564	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	184,558	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/
resolutions-5A4805yaaad-26401-1723300yppomads-1581930300hy	video.MP4	https://vod.video.cdn.mozilla.net/166895/edge/CAC202004e1...	182,112	9/5/2021 2:40 PM	9/5/2021 2:40 PM	84/2021 9:55 PM	9/

Figure 12. A snapshot of cached content during Nimo replayed video stream

Comparison	YouTube Gaming / Live	NIMO TV
The possibility to recover and view live stream video content	Yes	No
The possibility to recover and view replayed video content	Yes	Yes
The possibility to determine how much of a live video has been viewed	Yes	No
The possibility to determine how much of a replayed video has been viewed	Yes	Yes
MIME Type	MP4	MP2T
The duration of the cached fragments	Each fragment plays for about five seconds of the streamed video.	Each fragment plays for about six seconds of the streamed video.
The main steps of the reassembling process	MIME attribute. fragments' name. The last accessed date and time.	

Figure 13. A comparison between YouTube Live / Nimo TV cache characteristics

Two case studies are presented within the scope of this paper, an examination of YouTube Live / Nimo TV video streams. The experimental results show that the cached video from the installed Firefox and all Chromium can be reconstructed on YouTube Live / Gaming. A table with a summary of the main experimental results is shown in Figure 13. The results obtained can be used to examine other streaming services and web browser cache characteristics. We conducted some experiments on a larger scale with different scenarios on multiple machines to test the scalability. The goal is to evaluate the efficiency of the proposed technique and examine its shortcomings.

## 6. CONCLUSION AND FUTURE WORK

Live video streaming is one of the most innovative technologies nowadays. This work provides a follow-up to El-Tayeb et al. [12-14] examining two live-streaming platforms accessible via desktop web browsers. It provides a framework for reconstructing live video streams in the Chromium/Firefox browser. A technique for rebuilding live cached videos is proposed. It utilises an innovative method of extracting fragments. The testing procedures as well as the results are available. The proposed technique can be used to perform forensic analysis on a variety of streaming services. This technique has a lot of improvement. This study aims to help forensic scientists in this field retrieve live video more effectively. It would also be a valuable resource for law enforcement, digital forensic experts, etc. The methodology will be expanded in two directions in the future. First,

expanding the analysis into mobile browsers and direct applications. Second, testing various streaming platforms requires further analysis, such as Dailymotion, TikTok, etc.

## REFERENCES

- [1] Zhou, C. (2016). Handbook of research on creative problem-solving skill development in higher education. IGI global.
- [2] TV streaming services overtake pay TV for first time. <https://www.ofcom.org.uk/about-ofcom/latest/media/media-releases/2018/streaming-overtakes-pay-tv>, access on September 15, 2021.
- [3] Children under 13 groomed on live streams. <https://www.bbc.com/news/uk-44233544>, access on September 15, 2021.
- [4] Distribution of Captures of Live-streamed Child Sexual Abuse FINAL. [https://www.iwf.org.uk/sites/default/files/inline\\_files/Distribution%20of%20Captures%20of%20Live-streamed%20Child%20Sexual%20Abuse%20FINAL.pdf](https://www.iwf.org.uk/sites/default/files/inline_files/Distribution%20of%20Captures%20of%20Live-streamed%20Child%20Sexual%20Abuse%20FINAL.pdf).
- [5] 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>
- [6] Horsman, G. (2019). Reconstructing cached video stream content -Part 2. Digital Investigation, 31: 200893. <https://doi.org/10.1016/j.fsidi.2019.200893>
- [7] Horsman, G. (2018). A forensic examination of the technical and legal challenges surrounding the investigation of child abuse on live streaming platforms: A case study on Periscope. Journal of Information Security and Applications, 42: 107-117. <https://doi.org/10.1016/j.jisa.2018.07.009>
- [8] Schaap, E., Hoogendoorn, I. (2013). Reconstructing web pages from browser cache. University of Amsterdam, Amsterdam: Neverlands Forensics Institute.
- [9] Marrington, A., Baggili, I., Al Ismail, T., Al Kaf, A. (2012). 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, Sharjah, United Arab Emirates, pp. 1-6. <https://doi.org/10.1109/ICCSII.2012.6454516>
- [10] Adblock Plus. <https://addons.mozilla.org/en-US/firefox/addon/adblock-plus/>, access on September 15, 2021.
- [11] Gecko Layout Engine. <https://developer.mozilla.org/en-US/docs/Mozilla/Gecko>, access on September 15, 2021.
- [12] MPC-HC media player program version 1.7.13.112. <https://mpc-hc.org/downloads/>, access on September 15, 2021.
- [13] El-Tayeb, M., Taha, A., Taha, Z. (2021). Streamed video reconstruction for Firefox browser forensics. Ingénierie des Systèmes d'Information, 26(4): 337-344. <https://doi.org/10.18280/isi.260401>
- [14] Reddy, S.T., Mothe, R., Sunil, G., Harshavardhan, A., Korra, S.N. (2019). Collecting the evidences and forensic analysis on social networks: Disputes and trends in research. Studia Rosenthaliana (Journal for the Study of Research), 11(12): 183-192.