DFRWS

Reconstructing streamed video content: A case study on YouTube and Facebook Live stream content in the Chrome web browser cache

Graeme Horsman

School of Science, Engineering and Design; University of Teesside, Campus Heart, Southfield Rd, Middlesbrough TS1 3BX

Email: g.horsman@tees.ac.uk

Abstract

With the increased popularity of online video streaming comes the risk of this technology's subsequent abuse. With a number of cases noted in 2017 where individuals have engaged with illegal or policy breaching video content, digital forensics practitioners are often tasked with investigating the subsequent 'fingerprint' of such acts. This is often to determine both the content of a stream in question, and, how it has been interacted with, typically from an analysis of data residing on a suspect's local device. This article provides an examination of the forensic procedures required to identify and reconstruct cached video stream data using both YouTube and Facebook Live as example case studies. Stream reconstruction methodologies are offered where results show that where a YouTube and Facebook Live video have been played, buffered video stream data can be reassembled to produce a viewable video clip of content.

Keywords: Digital Forensics; Video Stream; Video Forensics; YouTube; Facebook Live.

1 Introduction

To highlight the issues surrounding on-line video streaming, initial reference is drawn to the following comments made by the National Crime Agency in December 2017.

> "*The use of live streaming platforms by online sex offenders is increasing…During a recent week of intensification to tackle child sexual exploitation and abuse, police and NCA operations across the UK safeguarded 245 children and arrested 192 people, 18 of whom were in a position of trust. 30% of those cases involved some of the highest harm offences including live streaming, blackmail and grooming…Intelligence from the NCA and police forces shows that that dangerous offenders are capitalising on the immediacy of contact that live streaming offers" (National Crime Agency, 2017a).*

Online video streaming platforms now provide users with an opportunity to share content and to observe (via streaming) video material posted by others, without exhibiting ownership of it in terms of intentionally downloading and storing video content. A significant proportion of Internet users now watch video content online (Statista, 2018b) where 'as of 2017, 81.2% of online users in the U.S. alone (over 200 million) accessed digital video content' (Statista, 2018c; 2018d), a figure which is predicted to rise. With such volumes of traffic come regulatory problems linked to both the uploading and distribution of video content in breach of law and platform policies, and, the subsequent viewing and engagement with such material. Whilst mainstream vendors may have the resources to tackle such issues, smaller services may not, creating a challenge for law enforcement when attempting to effectively respond to an incident of this type. Whilst the discovery of an illegal/policy breaching video online may lead to consequences for the video 'owner' or a hosting/streaming service provider, identifying who

has viewed and interacted with the video may lead to further liability for such individuals. This is apparent in cases of streamed indecent content where the National Crime Agency (2017b; 2017c; 2017d) in 2017 have noted numerous instances of users prosecuted for indecent imagery offences under English law after interacting with online indecent video material. Extremist video content has also attracted regulatory interest and response, with the United Kingdom Home Secretary Amber Rudd seeking to impose stronger penalties on those who repeatedly view terrorist material online in an attempt to strengthen existing regulation under areas such as section 58 of the Terrorism Act 2000 (Travis, 2017).

Acts of video streaming (whether live or the replay of pre-recorded hosted content) can be associated with a number of potential offences and where a suspect's device has been seized, forensic analysis may be required to identify any potential streamed content. Whilst Internet history records may in some instances provide a pointer to a hosted video that has been accessed, this may not always be an effective at identifying any streamed content. Where a video has since been removed by a provider (no longer accessible online by a practitioner for verification of content), locally cached stream data (providing it can be interpreted) may be the only source of information remaining to identify a streams content and context. Further in offences involving indecent imagery, the identification and recovery of imagery left behind by a stream on a local device may facilitate a charge of possession or making indecent imagery under English law (see Protection of Children Act 1978 and Criminal Justice Act 1988).

With regards a forensic examination of the impact and recovery of streamed video on a local device, limited information exists. This article provides one of the first commentaries in this area, and aims to support those carrying out investigations of this type to ensure effective evidence recovery and interpretation. In doing so, this work addresses the following questions.

1. Is streamed video content stored on a local device when viewed? And if so;

    a) Can streamed video content be recovered and viewed?

    b) Is it possible to determine how much of a video has been viewed?

Within the confines of this article two case studies are presented, an examination of YouTube and Facebook Live video streams. Due to limitations with article size, only the Chrome Internet browser has been examined as a platform for accessing and streaming video content. Both testing methodologies and results are offered.

2 YouTube

YouTube (`www.youtube.com`) is a video sharing and streaming service owned by Google and maintains significant popularity with a reported estimate of 184 million users in the U.S. alone (Statista, 2018), with a reported 400 hours of video uploaded every minute (Schindler, 2017). Whilst the platform offers a popular source of material across a number of topic areas, it has also attracted criticism, particularly focused at its regulation of resident content. Mechanisms for child protection and their apparent failures have been highlighted (BBC News, 2017b) with reports of up to 100,000 predatory accounts leaving indecent comments on video material (BBC News, 2017c). Further, reports of indecent content and videos depicting child characters in inappropriate situations (designed to trick child viewers into watching) have been noted (BBC News, 2017d; 2018b). In November 2017, YouTube were reported to have removed almost 50,000 videos documenting extremist content, however, were criticised for an apparent slowness to act (BBC News, 2017a). In addition, concerns have also been raised due to the hosting of videos depicting anti-Semitic and gang culture (BBC News, 2017f; 2018a).

Where the investigation of a suspect leads to the analysis of their YouTube viewing habits, resident Internet history may provide some support. A standard YouTube URL is structured as follows: `https://www.youtube.com/watch?v=`**`mXFjwihUO00`** where there URL itself is prefixed with a unique identifier (bolded above) for the YouTube video itself. In some cases, a practitioner can search for the video using this identifier and verify its content. However, this process alone may not address the following two points of concern.

1. Video removal: A user may view a video that has since been removed before a practitioner inspection can take place. In this case, a practitioner may identify a suspected URL, but be unable to locate the video on the YouTube site. Whilst it may be possible to request an account disclosure from YouTube, a record of such information may no longer exist, or limited organizational resources may deem disclosure routes impractical.
2. Behavior: Where a video is of large length, determining how much of a video a user has watched and what particular content may be of evidential value and could provide.

In the cases noted above, resident cached video data may provide the only source of determining the context of a streamed video. As a result, the remainder of Section 2 offers an examination of the impact of YouTube streams in the Chrome web browser cache.

2.1 Preliminary Approach

To provide an initial insight into the challenges of investigating stream caching, an initial test designed to explore the use of file identification, parsing and recovery processes to examine the browser cache following the viewing of a test stream was ran. This was intended to simulate traditional analysis approaches, which involve large-scale file recovery and viewing processes typically undertaken through the running of automated procedural scripts. The following methodology has been implemented.

Preparation: To start, a standard clean install of the Windows 10 operating system was implemented and the Chrome (version 63.0.3239.132 (latest at time of testing)) browser was installed (and unused).

Test data: A uniquely identifiable YouTube video was chosen as suitable test data and its content recorded. This would allow for a visual identification and verification of any subsequently recovered streamed content (following the analysis stage) on the local machine resulting from the test stream. The chrome cache folders (`C:\Users\Staff\AppData\Local\Google\Chrome\User Data\Default\Cache`) were verified as empty to prevent contamination by any existing data.

Viewing the stream: The test YouTube video's URL was entered into the Chrome browser window and the video was played in full. The browser was then closed and the machine was shut down and imaged.

Analysis: X-Ways forensics version 19.3's comprehensive search options were utilized to recover (identify or carve, and reconstruct) all potential image, video and internet related data. Reliance was placed on automated media gathering processes to simulate traditional case procedures that are often used in forensic investigations to pre-process any existing media files *en-masse* for later review. On completion, four still thumbnail-sized cached images (`.jpg`) denoting content (video frames) contained within the stream were recovered by both tools (located at `C:\Users\Staff\AppData\Local\Google\Chrome\User Data\Default\Cache`). 41 `.webm` (a compressed video stream format (FileInfo, n.d.)) files were also located following the parsing of the Chrome cache metadata and cache data files. All `.webm` were exported given they are reported to be video stream files and opened using

VLC media player version 2.2.6 where only one file was playable, containing content from the first three seconds of the test video stream. All other `.webm` files returned errors upon attempting to play.

2.2 Does this mean the video content is not there?

To provide an initial indicator of the presence of content being cached locally, when a YouTube video is loaded, buffering takes place (the download and storage of a portion of video data, ready for playing), indicated by the grey video bar (see Figure 1). To test for the presence of local data, once a portion of the stream has been buffered, the removal of an Internet connection allows some of the buffered portion of the stream to be replayed. Without the ability to access data on the YouTube server, it would appear that this information is being replayed from locally resident content.

Figure 1: An example of a buffered YouTube stream.

Google Chrome's developer mode (accessed by `Ctrl+Shift+i`) allows users to monitor network activity generated by a web page within a browser window. Figure 2 provides an example of the network activity generated live during an active YouTube stream. Of notable interest are the `videoplayback?lmt=` entries which coincide with the addition of a new buffered partition of the stream. For example, every time that the YouTube stream video bar adds an additional buffered portion of the stream, directly preceding this event is a `videoplayback?lmt=` request entry. Each entry of this type maintains a `MIME` type of `video/webm`. Further, each request results in the downloading and local storage of chunks of data, in some cases being almost 2MB in size.

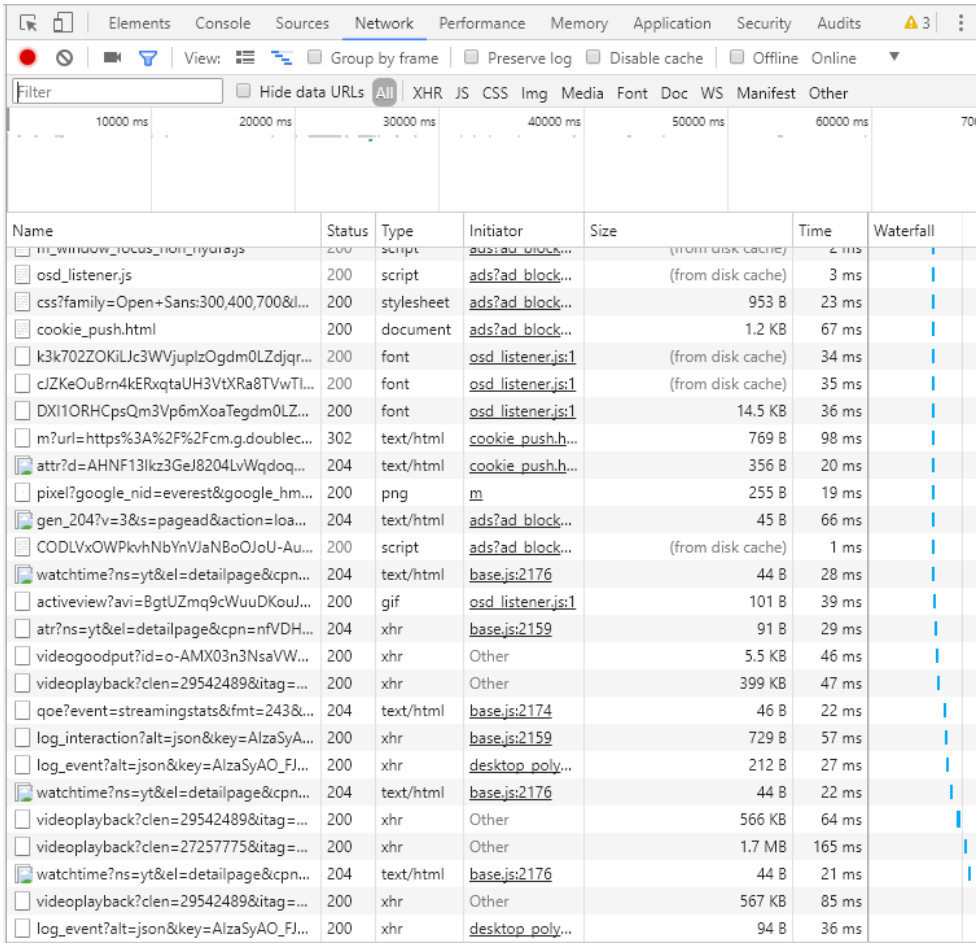


Figure 2: Google's developer mode during a YouTube video stream.

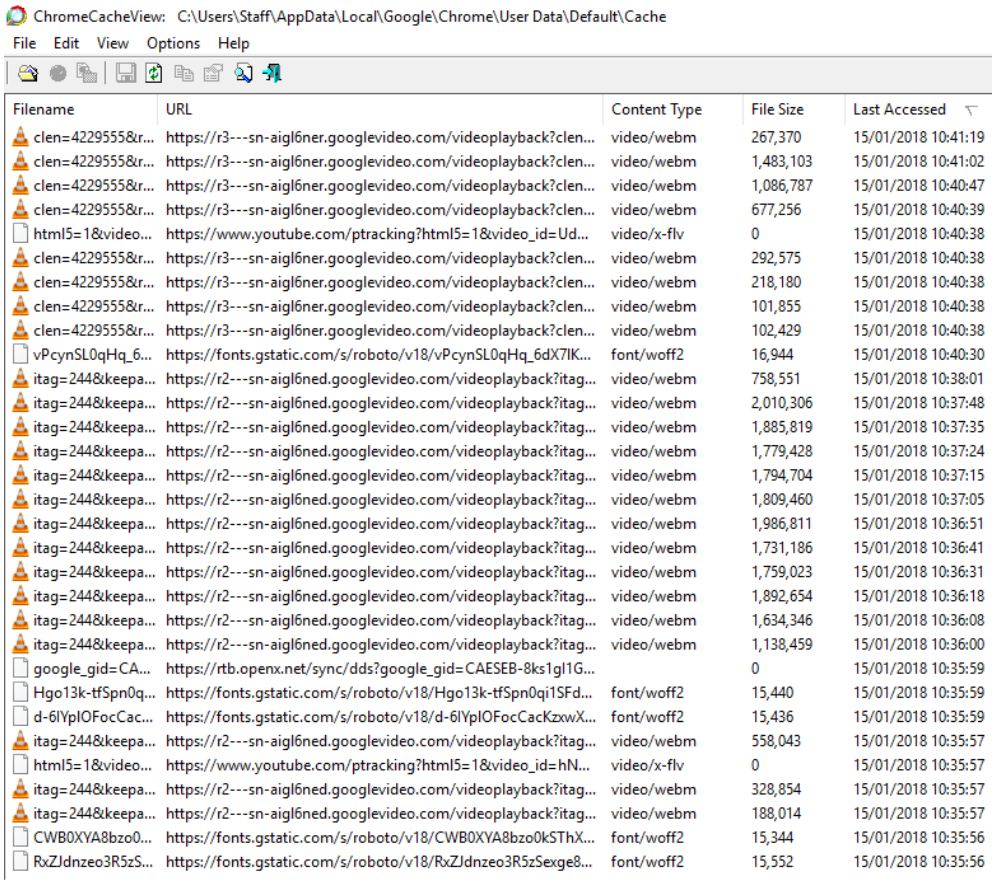
Google's developer mode suggests that caching is actually taking place on the local disk in relation to stream content. Using ChromeCacheView v1.76 (available `http://www.nirsoft.net/utils/chrome_cache_view.htm`) Chrome's cache folders can be parsed and monitored in real time during test conditions to assess the incremental impact of web browsing actions on locally stored content. Figure 3 provides an example of the cached video files (video file filter applied) following a test stream view. Test results indicate that when a YouTube stream is accessed, the process of buffering does result in data being incrementally stored on the local device.

At this point it is also necessary to draw reference back to the preliminary testing carried out in section 2.1. Such work was designed to resemble typical '*en-masse*' automated media recovery processes followed by a single file review (placed in an appropriate media player). The problem with such processes in relation to analyzing cached streams lies with media files being reviewed as single entities (complete videos in their own right). This consensus sits in conflict with the process of streaming, where a video is broken down and transferred via smaller data packages. Whilst when examined as single files, only the start of a stream can be reviewed, the remaining stream content can be viewed, but only following an effective reassembly of the buffered stream fragments (see Section 2.3).

2.3 Video reconstruction

To reconstruct the YouTube stream, all `.webm` entries must first be collected. Whereas preliminary testing indicated that 1 of the 41 `.webm` files is playable, all files collectively form

1 complete stream, but to view this content they must be processed correctly in the following way.



Filename	URL	Content Type	File Size	Last Accessed
clen=4229555&r...	https://r3---sn-aigl6ner.googlevideo.com/videoplayback?clen...	video/webm	267,370	15/01/2018 10:41:19
clen=4229555&r...	https://r3---sn-aigl6ner.googlevideo.com/videoplayback?clen...	video/webm	1,483,103	15/01/2018 10:41:02
clen=4229555&r...	https://r3---sn-aigl6ner.googlevideo.com/videoplayback?clen...	video/webm	1,086,787	15/01/2018 10:40:47
clen=4229555&r...	https://r3---sn-aigl6ner.googlevideo.com/videoplayback?clen...	video/webm	677,256	15/01/2018 10:40:39
html5=1&video...	https://www.youtube.com/ptracking?html5=1&video_id=Ud...	video/x-flv	0	15/01/2018 10:40:38
clen=4229555&r...	https://r3---sn-aigl6ner.googlevideo.com/videoplayback?clen...	video/webm	292,575	15/01/2018 10:40:38
clen=4229555&r...	https://r3---sn-aigl6ner.googlevideo.com/videoplayback?clen...	video/webm	218,180	15/01/2018 10:40:38
clen=4229555&r...	https://r3---sn-aigl6ner.googlevideo.com/videoplayback?clen...	video/webm	101,855	15/01/2018 10:40:38
clen=4229555&r...	https://r3---sn-aigl6ner.googlevideo.com/videoplayback?clen...	video/webm	102,429	15/01/2018 10:40:38
vPcynSL0qHq_6...	https://fonts.gstatic.com/s/roboto/v18/vPcynSL0qHq_6dX7lK...	font/woff2	16,944	15/01/2018 10:40:30
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	758,551	15/01/2018 10:38:01
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	2,010,306	15/01/2018 10:37:48
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,885,819	15/01/2018 10:37:35
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,779,428	15/01/2018 10:37:24
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,794,704	15/01/2018 10:37:15
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,809,460	15/01/2018 10:37:05
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,986,811	15/01/2018 10:36:51
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,731,186	15/01/2018 10:36:41
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,759,023	15/01/2018 10:36:31
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,892,654	15/01/2018 10:36:18
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,634,346	15/01/2018 10:36:08
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	1,138,459	15/01/2018 10:36:00
google_gid=CA...	https://rtb.openx.net/sync/dds?google_gid=CAESEB-8ks1gl1G...		0	15/01/2018 10:35:59
Hgo13k-tfSpn0q...	https://fonts.gstatic.com/s/roboto/v18/Hgo13k-tfSpn0qi1SFd...	font/woff2	15,440	15/01/2018 10:35:59
d-6lYplOFocCac...	https://fonts.gstatic.com/s/roboto/v18/d-6lYplOFocCacKzxwX...	font/woff2	15,436	15/01/2018 10:35:59
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	558,043	15/01/2018 10:35:57
html5=1&video...	https://www.youtube.com/ptracking?html5=1&video_id=hN...	video/x-flv	0	15/01/2018 10:35:57
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	328,854	15/01/2018 10:35:57
itag=244&keepa...	https://r2---sn-aigl6ned.googlevideo.com/videoplayback?itag...	video/webm	188,014	15/01/2018 10:35:57
CWB0XYA8bzo0...	https://fonts.gstatic.com/s/roboto/v18/CWB0XYA8bzo0kSThX...	font/woff2	15,344	15/01/2018 10:35:56
RxZJdnzeo3R5zS...	https://fonts.gstatic.com/s/roboto/v18/RxZJdnzeo3R5zSexge8...	font/woff2	15,552	15/01/2018 10:35:56

Figure 3: ChromeCacheView displaying Chrome's cache containing video content.

Each `.webm` entry maintains a portion of a stream and reassembly must take place in order to create a viewable video. Where a YouTube video has been cached, using `ChromeCacheView` to order cache entries by their last accessed date and time provides the order in which artefacts are cached in Chrome on the local disk (as shown in Figure 3). Each `.webm` cache entry must have its associated URL (see Magnet Forensics (2017) for an overview of the Chrome cache functionality) examined in order to identify its 'fragment order' (an attribute coined in this article). A typical `.webm` cached artefact URL is structured as follows:

```
https://r2---sn-
aigl6ned.googlevideo.com/videoplayback?itag=244&keepalive=yes&lmt=15
15578817467917&key=yt6&signature=76C58D7F78D783433894A5035F5782BC42B
24479.1267C0A3034DC4EBDA3C7968798118B3810E0BE3&ms=au&mv=m&mt=1516012
566&requiressl=yes&ip=152.105.118.127&ipbits=0&gcr=gb&pl=16&id=o-
AE1mirNM9fvhqmgotXSh29VDXx1bmxZr2dzVu_HMWonX&mime=video%2Fwebm&mn=sn
-aigl6ned&mm=31&expire=1516034260&ei=dIRcWoSZI4LgV-
T7ucAH&initcwndbps=1595000&gir=yes&dur=272.440&source=youtube&clen=2
1255658&sparams=aitags%2Cclen%2Cdur%2Cei%2Cgcr%2Cgir%2Cid%2Cinitcwnd
bps%2Cip%2Cipbits%2Citag%2Ckeepalive%2Clmt%2Cmime%2Cmm%2Cmn%2Cms%2Cm
v%2Cpl%2Crequiressl%2Csource%2Cexpire&aitags=133%2C134%2C135%2C136%2
C137%2C160%2C242%2C243%2C244%2C247%2C248%2C278&ratebypass=yes&alr=ye
s&cpn=QhnO2WvdKbz3nFlQ&c=WEB&cver=2.20180111&range=0-
188013&rn=0&rbuf=0
```

Of particular interest is the `range=` value (noted in bold), which can be used to determine the order of frames within the cached video stream. Typical YouTube streams which are `.webm` maintain a header frame which identifies the start of the video. This is identifiable via its `.webm` signature (shown in Table 1) and will have a `range=` value of `0-<number>`. During testing, this was found to be the only `.webm` file which was playable when accessed individually. The `dur=` attribute notes the entire length of the video, not the amount of video which has been cached to the local disk.

Using the header file as a starting position, additional `.webm` files must be concatenated (a binary file concatenation, joining for example the header fragment to a second fragment in sequence order to create a separate combined file) to it in order to recreate the video (see Figure 4). This must be done in frame order using the values stored in the `range=` attribute. Whilst the header file maintains an identifiable `.webm` signature, testing indicates that the following stream chunks do not maintain a consistent header structures. Therefore, to identify the order of all stream fragments, this must be done using the `range=` ordering variable and via the parsing of Chrome cache artefacts and their associated metadata to identify their `MIME` types and associated URL containing the `range=` attribute (see Table 1).

Table 1: A breakdown of a hypothetical reconstruction of a YouTube stream

File Order	File Order	Range (example values)	File Signature
Header	1	0-188013	0x1A 0x45 0xDF 0xA3 0x9F 0x42 0x86 0x81 0x01 0x42 0xF7 0x81 0x01 0x42 0xF2 0x81 0x04 0x42 0xF3 0x81 0x08 0x42 0x82 0x84 0x77 0x65 0x62 0x6D 0x42
Data Fragment	2	188014-35644	N/A
Data Fragment	3	35645-611485	N/A
Data Fragment	4	611486-983432	N/A

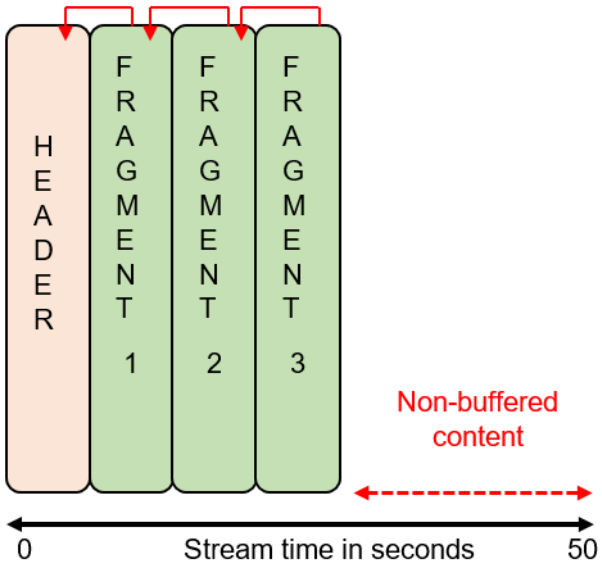


Figure 4. A hypothetical structure of the reassembled (concatenated) stream file.

2.4 Points to note

First, only the buffered part of a stream can be reconstructed and it was not possible to identify which sub-portion of the buffered content a user had viewed on screen. Therefore, where a user buffers 40 seconds of a 50 second video, the final 10 seconds cannot be reconstructed as the buffered content is not local (see Figure 4). Reconstruction is not effected by a user watching the video, therefore where a user loads a video but immediately pauses it, buffered but 'non-watched' content is still stored and can be reconstructed. However, buffered content is not evidence of 'viewed' content despite YouTube's buffering being dynamic where unless the user plays the stream, a complete buffering of the video does not occur. Typically, testing indicated that when a YouTube video is loaded but immediately paused, around 30 seconds of stream content is buffered locally and can be reconstructed.

Second, stream reconstruction requires a full cache investigation in order to parse cached content and the associated metadata belonging to any cached artefacts. Without the `range=` value, reassembly is unlikely to be successful and likely be based on guessing the relevant order of files. This issue also means that there is a potentially low success rate of recovering and rebuilding fragmented streams where content is no longer in the cache (for example an unallocated cluster recovery) as relevant stream metadata needed for rebuilding may be missing.

Third, attempts to rebuild streams with an incomplete set of stream fragments or in the wrong order typically results in a non-viewable rebuilt stream. This is even the case where one fragment appears out of order.

Fourth, during testing a small number of `.mp4` formatted YouTube streams were also encountered. Their behavior in the cache is comparable to `.webm` streams, where a rebuild can be obtained via ordering of the `range=` URL attribute (see also discussions in section 3 for signature information for `.mp4` formatted streams).

3 Facebook Live

Facebook Live is an additional feature of the Facebook platform giving users the ability to live stream video content. Streamed content becomes available as part of the Facebook profile where existing privacy and permission settings regarding the availability of the video apply. Public broadcasts can be viewed by those who passively access the account, whereas private broadcasts can be limited to those who are 'friends' of the account holder. Once a live broadcast has finished, the video will remain available (subsequent to the author deleting or adjusting viewing settings) and can be viewed later (taking the form of a recorded stream). As with many video platforms, large amounts of traffic is harmless, yet instances of the Facebook Live service abuse have been noted. These include reports of live broadcasts depicting sexual assaults (BBC News, 2017g), threatening behaviors (BBC News, 2017h), potential copyright infringement (BBC News, 2017i) and broadcasted murder (BBC News, 2017j).

3.1 Initial Testing

It is first key to note what is and what is not cached when interacting with Facebook Live. When a user 'live broadcasts' and a suspect account watches the broadcast live, testing indicated that no caching occurs in the suspect's Chrome browser cache. To test this, following the same procedural steps to create a clean test environment as noted in Section 2.1, a separate lab machine was used to initialize a test Facebook Live broadcast. On the test machine, the URL of the live broadcast was entered into the Chrome browser in order to take

the user directly to this live broad cast. For the duration of the 1-minute-long broadcast, the suspect Chrome cache was reviewed live using ChromeCacheView (refreshing the application every 2 seconds). On completion of the broadcast, the suspect's browser was closed and the cache was finally examined with no video caching activity apparent (in comparison to the impact of a replayed stream discussed in Section 3.2). Therefore, testing indicates that those who only view live broadcasts do not have stream content cached in their Chrome browser cache.

3.2. Stream Replays

In contrast to watching live broadcasts, when a user replays a hosted Facebook Live broadcast (i.e. a suspect watches a video which a user has left hosted after a live broadcast – essentially replaying the content), browser caching does take place. Following the replay of a Facebook Live hosted video, Figure 5 demonstrates the typical impact of this process on the Chrome browser cache. Stream fragments are noted to be in `.mp4` format, yet none are playable as individual files (tested using VLC media player version 2.2.6).

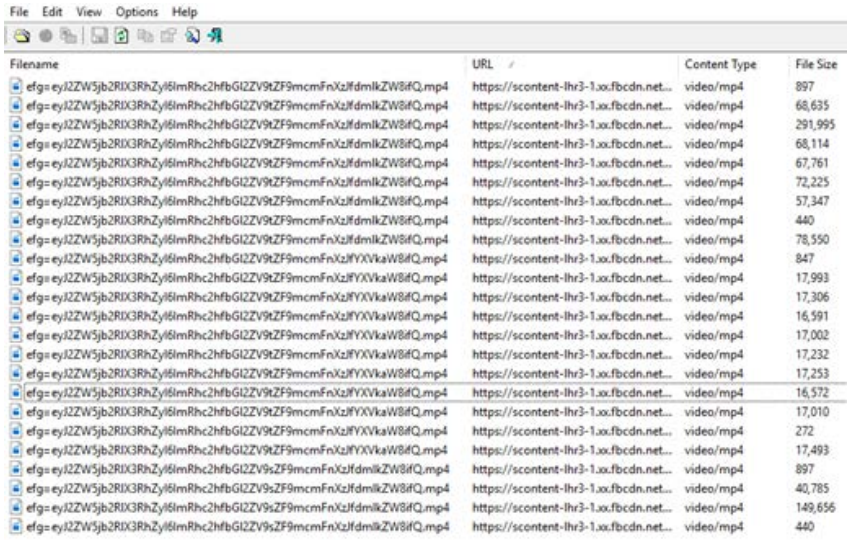


Figure 5: Replay of a Facebook Live video displayed in ChromeCacheView.

As with YouTube streams, these fragments can be reassembled (binary concatenated as with YouTube stream fragments) to reconstruct stream content, but only through an analysis of the URL of the cached artefact. A typical Facebook Live video cached artefact URL is structured as follows when analyzed using ChromeCacheView.

`https://scontent-lhr3-1.xx.fbcdn.net/v/t42.1790-29/26947798_1548879368500753_3538282435986849792_n.mp4?efg=eyJ2ZW5jb2RlX3RhZyI6ImRhc2hfbGl2ZV9tZF9mcmFnXzJfYXVkaW8ifQ%3D%3D&oh=a5c44f0172736933195c1eaf2e35bb9d&oe=`**`5A5E4AFF`**`&`**`bytestart=`**`52910&`**`byteend=`**`69481`

To rebuild the stream, the `oe=`, `bytestart=` and `byteend=` attributes are important. Testing indicates that the `oe=` attribute acts as a stream identifier. Figure 6 provides an example where despite only one stream being viewed, cached stream fragments are sorted by their `oe=` attribute, where only matching `oe=` values form part of the same stream rebuild. The `bytestart=` and `byteend=` attributes denote the order of concatenation.

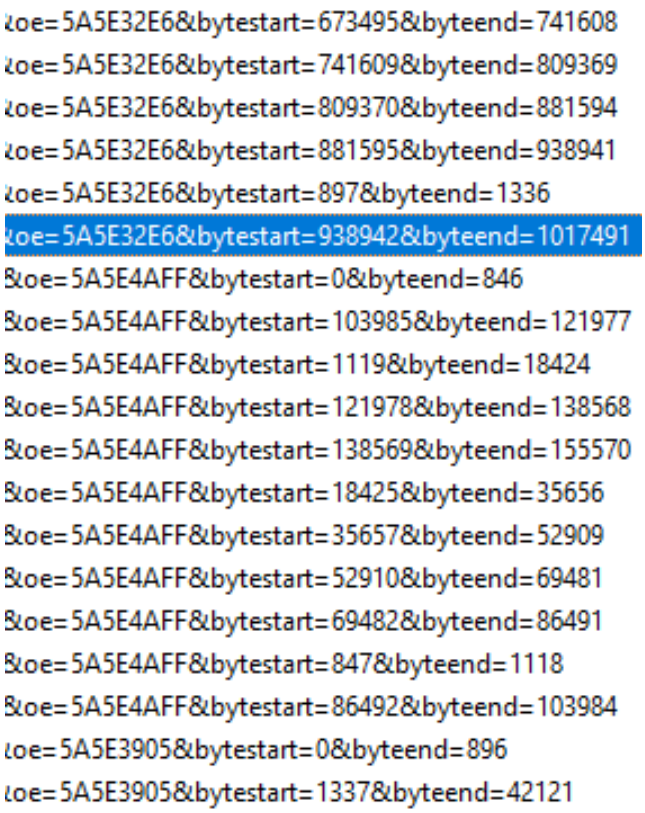


Figure 6: An example of `oe=` attribute values in the cache.

Rebuilding the stream is a similar process to that of YouTube where a binary concatenation of files will potentially create a viewable stream.

Typically, stream rebuild fragments will appear as noted in Figure 7, with a typical `.mp4` structured header (`ftyoiso` identifier), followed by a `sidx` identifier fragment and finally a series of `moof` identifier fragments. Only buffered content of a Facebook Live replayed video can be recovered.

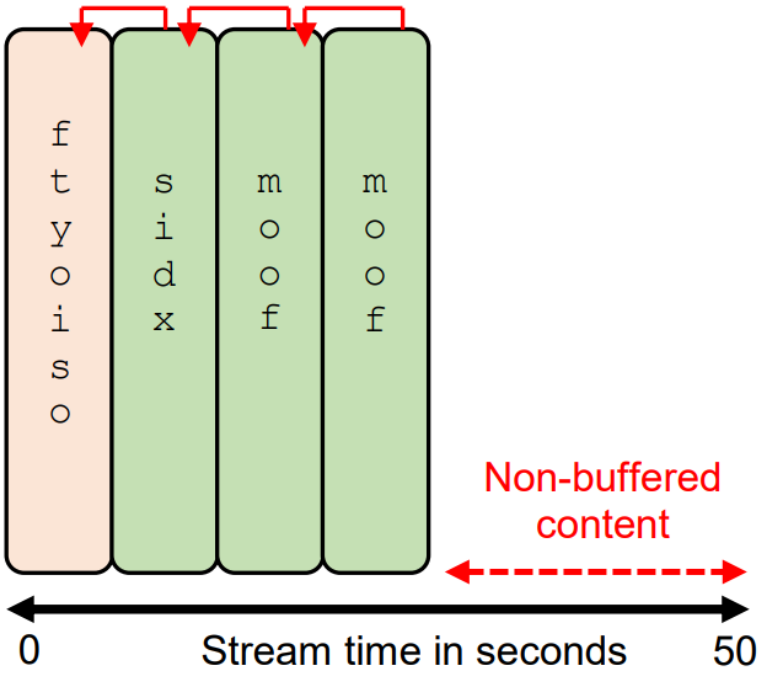


Figure 7: A hypothetical structure of the Facebook Live stream rebuild.

***Points to note*: Whilst the example in Figure 6 shows three potential `oe=` attribute streams, only one contains the actual video stream itself when rebuilt in the correct order. Testing was unable to determine which `oe=` attribute contains the stream before rebuilding; therefore all must be built in order to create a viewable stream. In addition, whilst the `bytestart=` and `byteend=` attributes must be used in incremental order to determine the order of concatenation, they are not always perfectly numerically aligned (for example, not always 1, 2, 3, 4 – sometimes 1, 3, 4, 6). Providing they were in incremental numerical value order, testing indicated that a stream rebuild could still be achieved.

4 Concluding Points

Streaming platforms are likely to continue to pose regulatory issues with future incidents of abuse almost certain to be reported. In response to such incidents, digital forensics practitioners will likely be tasked with effectively reconstructing streamed data to establish the presence of policy/law breaching material. This article has offered an introductory case study on the forensic processing of cached video stream data in the Chrome web browser to support forensic practitioners. The rebuilding of video stream fragments has been demonstrated in order to produce a viewable video clip of locally buffered data.

In both cases, traditional '*single file*' media analysis strategies for identifying and examining media content as single entities are ineffective. Stream fragments must be identified from within the cache where an analysis of both the cached artefact and their associated metadata contained within the cache files is required. The ChromeCacheView application facilitates a parsing of the Chrome cache folders and this process is needed in order to carry out an effective stream rebuild, where a suspect's cache folders can be exported from a case and examined separately using this tool. Cached metadata surrounding each artefact is needed to allow stream fragments to be correctly ordered during a stream rebuild. An absence of this data would result in a practitioner having to guess the order of the fragments, which is arguably

not feasible, particularly where a stream is of large length and a number of fragments have been cached.

Whilst only two streaming services have been analyzed, it is hoped that the examination methodologies and considerations presented are applicable to a forensic analysis of other streaming services and web browser caches, which requires future analysis to determine. However, this work provides an indication of the need to consider the possibly of analyzing video media files as a collective rather than relying on '*single file viewing*' as a means of identify and validating video content.

4.1 Future work

This work has offered a starting point for local video stream analysis whilst highlighting investigatory approaches. Future work involves the expansion of analysis in three possible directions. First, Chrome as a platform to access the stream has been utilized and work must expand analysis into both additional browser types and caching via mobile applications (mobile application browsers and direct applications such as the YouTube app). Second, there are multiple streaming platforms, which are in need of further testing and analysis with examples including 'Twitch'. Finally, characteristics of cached streams should be further examined. This includes an analysis of the persistence of cached stream data in the browser cache and the potential for recoverability following a 'cache clear' should be tested. In addition, the identification and recoverability of stream content from caches that have be subject to heavy use requires further investigation.

5 References

BBC News (2017a) 'YouTube removes dead extremist's videos' Available at: http://www.bbc.co.uk/news/technology-41969461(Accessed: 12 January 2017)

BBC News (2017b) 'YouTube child abuse reporting system 'flawed' Available at: http://www.bbc.co.uk/news/av/stories-42105526/youtube-child-abuse-reporting-system-flawed (Accessed: 12 January 2017)

BBC News (2017c) 'Glitch in YouTube's tool for tracking obscene comments' Available at: http://www.bbc.co.uk/news/blogs-trending-42060357 (Accessed: 12 January 2017)

BBC News (2017d) 'The disturbing YouTube videos that are tricking children' Available at: http://www.bbc.co.uk/news/blogs-trending-39381889 (Accessed: 12 January 2017)

BBC News (2017e) 'YouTube to restrict 'disturbing' children's videos, if flagged' Available at: http://www.bbc.co.uk/news/technology-41942306

(Accessed: 12 January 2017)

BBC News (2017f) 'Sadiq Khan urges YouTube to remove 'violent gang culture videos'' Available at: http://www.bbc.co.uk/news/uk-england-london-40849611(Accessed: 12 January 2017)

BBC News (2017g) 'Second teenager arrest for 'Facebook live assault'' Available at: http://www.bbc.co.uk/news/world-us-canada-39493838 (Accessed: 16 January 2017)

BBC News, (2017h) 'Facebook Live killer says he is looking for victims' Available at: http://www.bbc.co.uk/news/av/world-us-canada-39617888/facebook-live-killer-says-he-is-looking-for-victims (Accessed: 16 January 2017)

BBC News (2017i) ‘The bruising clash over a Facebook Live stream’ Available at: http://www.bbc.co.uk/news/world-australia-38876428 (Accessed: 16 January 2017)

BBC News (2017j) ‘Thai man kills baby on Facebook Live then takes own life’ Available at: http://www.bbc.co.uk/news/world-asia-39706205 (Accessed: 16 January 2017)

BBC News (2018a) ‘'Holocaust revisionist' on trial for anti-Semitic songs’ Available at: http://www.bbc.co.uk/news/uk-england-derbyshire-42637888 (Accessed: 12 January 2017)

BBC News (2018b) ‘Man pleads guilty over 'widely shared' abuse video’ Available at: http://www.bbc.co.uk/news/uk-england-42639072(Accessed: 12 January 2017)

FileInfo (n.d.) ‘.webm File Extension’ Available at: https://fileinfo.com/extension/webm (Accessed: 12 January 2017)

Magnet Forensics (2017) ‘Digital Forensics: Artifact Profile – Google Chrome’ Available at: https://www.magnetforensics.com/artifact-profiles/artifact-profile-google-chrome/ (Accessed: 17 January 2017)

National Crime Agency (2017a) ‘245 children safeguarded and 192 arrests for child sex abuse offences’ Available at: http://www.nationalcrimeagency.gov.uk/news/1252-245-children-safeguarded-and-192-arrests-for-child-sex-abuse-offences (Accessed: 12 January 2017)

National Crime Agency (2017b) ‘Man who entered child abuse forum 88 times in seven months had application form for paedophile ring’ Available at: http://www.nationalcrimeagency.gov.uk/news/1233-man-who-entered-child-abuse-forum-88-times-in-seven-months-had-application-form-for-paedophile-ring

National Crime Agency (2017c) ‘Three-year sentence for sex offender who watched live abuse’ Available at:

http://www.nationalcrimeagency.gov.uk/news/1216-three-year-sentence-for-sex-offender-who-watched-live-abuse

National Crime Agency (2017d) ‘UK man watched girl being sexually abused online’ Available at: http://www.nationalcrimeagency.gov.uk/news/1205-uk-man-watched-girl-being-sexually-abused-online

National Crime Agency (2017e) ‘Jail for chatroom sex offender’ Available at: http://www.nationalcrimeagency.gov.uk/news/1176-sex-offender-jailed-for-child-rape-messages

Schindler, Philipp (2017) ‘Expanded safeguards for advertisers’ Available at: https://www.blog.google/topics/ads/expanded-safeguards-for-advertisers/

Statista (2018) ‘Number of YouTube users in the United States from 2014 to 2019 (in millions)’ Available at: https://www.statista.com/statistics/469152/number-youtube-viewers-united-states/ (Accessed: 12 January 2017)

Statista (2018b) ‘Percentage of internet users who watch online video content on any device as of January 2017, by country’ Available at: https://www.statista.com/statistics/272835/share-of-internet-users-who-watch-online-videos/(Accessed: 12 January 2018)

Statista (2018c) ‘Digital video penetration in the United States from 2013 to 2021’ Available at: https://www.statista.com/statistics/271612/percentage-of-digital-video-viewers-in-the-united-states/ (Accessed: 12 January 2018)

Statista (2018d) ‘Number of digital video viewers in the United States from 2012 to 2021 (in millions)’ Available at: https://www.statista.com/statistics/271611/digital-video-viewers-in-the-united-states/ (Accessed: 12 January 2018)

Travis, Alan (2017) ‘Amber Rudd: viewers of online terrorist material face 15 years in jail’ Available at: https://www.theguardian.com/uk-news/2017/oct/03/amber-rudd-viewers-of-online-terrorist-material-face-15-years-in-jail (Accessed: 12 January 2018)

Biography

Graeme Horsman is a lecturer in Digital Forensics at Teesside University and has over 6 years experience in teaching in higher education. Graeme previously worked as a digital forensic analyst and was previously an EnCase Certified Examiner and Computer Certified Examiner (EnCE) and Certified Computer Examiner (CCE). He has a BSc (Hons) in Computer Forensics, a PhD, Graduate Diploma in Law, Masters of Jurisprudence and Post-Graduate Certificate in Higher Education Practice.

His research focuses on digital forensic examination techniques, methods for forensically investigating mobile devices, and knowledge-based systems for improving digital forensic examinations and evidence identification. In addition, Graeme is research active in the area of testing and validation in digital forensics and learning and teaching methods. Sub-research topics include the use of so-called anonymous communication services and the potential detection of users and legislation surrounding the possession, distribution and creation of illegal imagery.