Automated Identification and Reconstruction of YouTube Video Access

Jonathan Patterson[1], Christopher Hargreaves[2]

Centre for Forensic Computing, Cranfield University,
Shrivenham SN6 8LA

[1]j.patterson@cranfield.ac.uk, [2]c.j.hargreaves@cranfield.ac.uk

Abstract

YouTube is one of the most popular video-sharing websites on the Internet, allowing users to upload, view and share videos with other users all over the world. YouTube contains many different types of videos, from homemade sketches to instructional and educational tutorials, and therefore attracts a wide variety of users with different interests. The majority of YouTube visits are perfectly innocent, but there may be circumstances where YouTube video access is related to a digital investigation, e.g. viewing instructional videos on how to perform potentially unlawful actions or how to make unlawful articles.

When a user accesses a YouTube video through their browser, certain digital artefacts relating to that video access may be left on their system in a number of different locations. However, there has been very little research published in the area of YouTube video artefacts.

The paper discusses the identification of some of the artefacts that are left by the Internet Explorer web browser on a Windows system after accessing a YouTube video. The information that can be recovered from these artefacts can include the video ID, the video name and possibly a cached copy of the video itself. In addition to identifying the artefacts that are left, the paper also investigates how these artefacts can be brought together and analysed to infer specifics about the user's interaction with the YouTube website, for example whether the video was searched for or visited as a result of a suggestion after viewing a previous video.

The result of this research is a Python based prototype that will analyse a mounted disk image, automatically extract the artefacts related to YouTube visits and produce a report summarising the YouTube video accesses on a system.

1. Introduction

YouTube is a popular site for sharing videos which contains a wide range of content. While the majority of YouTube visits are perfectly innocent, there may be circumstances where YouTube video access is related to a digital investigation, e.g. viewing instructional videos on how to perform potentially unlawful actions or how to make unlawful articles. This paper investigates the extent to which these video accesses can be identified. This paper makes a number of contributions. Firstly it documents some of the artefacts that can be left as a result of a YouTube visit, particularly that the title of the video can be recovered. It also demonstrates that it is possible to infer about how a user arrived at a particular page, either from information extracted from the URL or by creating a timeline of the artefacts and examining other events that occurred immediately before or after. The paper also describes a prototype tool that will identify the patterns of artefacts and report a summary of the YouTube activity.

2. Background

2.1 Overview

This section provides a brief discussion of YouTube, followed by an explanation of the need for automation in digital forensics, highlighting the increase in the volume of data that analysts are faced with, but also the need to maintain an audit trail from raw data to interpreted data and any inferences drawn. The section also includes a discussion of work related to YouTube digital forensics.

2.2 YouTube

YouTube was founded in 2005 as a video sharing website where users can find, watch and share videos with others. YouTube has currently over 100 million users across the world with approximately 2 billion views per day. It also has over 24 hours of video uploaded every minute [1] and as a result of this, YouTube videos inevitably cover a large range of video topics including political, instructional and educational videos.

YouTube has strict policies relating to what can and cannot be uploaded to their website. Videos containing extremism, animal abuse, bomb making or sexual activities are examples of video content that violates these policies and uploading videos with content along those lines is prohibited. Although YouTube have these policies in place, YouTube [2] relies on users to detect inappropriate content, stating that "when a video gets flagged as inappropriate, we review the video to determine whether it violates our Terms of Use—flagged videos are not automatically taken down by the system". A consequence of this review process is that it is possible that videos containing illegal or inappropriate content may exist on YouTube for anyone to view for a period of time. Furthermore, it is important to remember that digital investigations do not necessarily have to be part of a law-enforcement investigation and may be an investigation into a violation of acceptable use policies in a corporate environment [3], or even an investigation in a education environment into 'cyber-bullying'[4].

2.3 Automation in Digital Forensics

Event reconstruction is an integral part of any digital investigation; as described in Casey [5]; it is the process of piecing together evidence during the initial stages of an investigation to help develop a better understanding of what events actually took place.

As part of a digital investigation it is necessary to manually examine and extract artefacts from a suspect system and carry out event reconstruction. This process can be difficult and time consuming due to both the complexity of today's systems and the volume of data they can hold. Over the past few years the amount of data on digital devices has increased dramatically and as a result, the workload for investigators has also increased. By introducing a system that automatically identifies artefacts and reconstructs them into events, the time spent by investigators on the higher level analysis phase of an investigation can be increased. In addition, given the increase in number of digital devices that can be involved in an investigation, it can also be necessary to prioritise the examination of one system over another, either due to resource limitations or the time sensitive nature of an investigation [6]. Automation could help in this prioritisation.

However, there is a difficulty with automating part of a digital forensics investigation, in that conclusions drawn may ultimately need to be presented in court. Principle 3 of the ACPO Guidelines on Computer-Based Electronic Evidence is particularly relevant, which states that "an audit trail or other record of all processes applied to computer-based electronic evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result" [7].This is also discussed in Carrier [8] which describes that error can be potentially introduced at each layer of abstraction when using forensic analysis tools, and that this error rate should be captured and taken into account when considering the results. Automated tools should therefore be able to 'explain' any results produced or conclusions that are drawn, preferably making it straight forward to manually verify the results and highlighting any potential error.

2.4 Related Work

There has been a small amount of published research on the forensic analysis of YouTube use. Sureka et al [9] describes a semi-automated system that mines YouTube with the purpose of discovering both extremist videos and hidden communities. Their work has been developed to aid law enforcement in dealing with cyber-crime in the area of radicalisation and has demonstrated that the automation of even part of a manual process can be beneficial to investigators in terms of time and overall success rate. However, while the paper is related to YouTube, is focused on detection of online data and does not assist in the examination of hard disks that are typically the source of evidence in digital forensic analyses.

There has also been some research carried out in the area of automated tools for answering 'higher-level' questions in digital investigations. While not related to YouTube, Adelstein and Joyce [10] describes an automated extraction tool capable of determining the presence of peer-to-peer software installed on a system and extracting evidence based on those results in a forensically sound manner. The developed tool automates what is described as a "manual and labor-intensive process" and demonstrates that this sort of automated analysis is feasible, and suggests that in some cases it is desirable.

2.5 Summary

This section has shown that evidence of YouTube activity may be of interest in the course of some digital investigations and that there is little published work on the topic. It has also shown that due to the volume and complexity of data encountered in the course of investigations some level of automation is advantageous. However, it has also discussed that while automation is useful to address these problems, since the output of an analysis may eventually need to be used in court, an automated tool should provide a full audit trail of how results were produced.

3.0 Methodology

3.1 Overview

As discussed in the introduction, this research investigates the possibility of producing an automated summary of a user's activity on YouTube for the purpose of highlighting to an investigator that there may be related evidence of interest. The research described in this paper is split into several stages. First, the artefacts that are left on a computer after a user has visited YouTube are identified. Secondly, experiments are conducted to determine what specific user behaviour can be inferred from the presence of the artefacts. Finally, in order to automate the process, a tool is developed that can extract the artefacts and supply an appropriate summary of user activity based on what has been extracted. The remainder of this section discuses these steps in detail.

3.2 Scope

As discussed earlier, YouTube videos are usually accessed through a web browser, which in turn runs on an operating system. There are a several browsers (e.g. Internet Explorer, Firefox, Chrome, Safari, Opera) and several operating systems (Windows XP, Vista and 7, Mac OS X, multiple distributions of Linux), not to mention dedicated YouTube apps as found on iOS. This paper particularly examines the artefacts left when Internet Explorer 8 is used for this viewing on Windows 7. Also, the paper considers viewing YouTube videos directly on the YouTube website rather than embedded videos in other sites.

3.3 Artefact Identification

The first step in the reconstruction of a user's visit to YouTube is to identify the artefacts left behind. There are a number of different methods available that can be used to help identify changes to a system. These are discussed in Hargreaves & Chivers [11] the first of which is live logging, i.e. the process of recording changes

to a live system with the help of tools such as Procmon[1]. The second method is using snapshots, which involves taking a copy of a test system before and after an action is carried out and identifying the differences between the two states. The last method involves creating a list of all the files on the system after the interaction has taken place and sorting them by date and time to identify any changed files.

For this research the last method is considered most appropriate. Since the aim of the research is to automate some of the event reconstruction process, meta-data associated with files e.g. dates and times, are particularly important. As a result, in terms of file system artefacts, only files with records that remain in the Master File Table (MFT) are considered, rather than supplementing them with those recoverable using file carving techniques. However, the time-based listing of files is extended, and in addition to the Modified, Accessed, Created and Entry Modified (MACE) times recovered from the file system, some compound files (e.g. the Registry hives[12] and index.dats [13] used by Internet Explorer) are also processed and integrated with the entries from the file system.

Test data is generated through the use of experiments with virtual machines. In this case VMware Workstation[2] is used to virtualise a Windows 7 environment and various YouTube sites are visited. At all times the actions taken when using the test system are recorded. After visiting the websites, duplicates are made of the virtual machine's hard disk (the .vmdk file), and in addition to the generation of the sorted list of file system, Registry and index.dat times, where further details need to be extracted, tools such as X-Ways Forensics[3], NetAnalysis[4], and Nirsoft's suite of History, Cache and Cookie Viewers[5] are used.

3.4 User Behaviour

Once the artefacts that are produced during a YouTube visit are identified, the next step is to compare each set of artefacts with the user's behaviour. As discussed in the previous section, the actual user behaviour in the test environment is documented, which allows artefacts and user actions to be compared. There are several distinct 'behaviours' that are examined. These are:

- The viewing of a video,
- The viewing of a video as a result of a YouTube suggested video,
- A YouTube search,
- The viewing of a video as a result of a YouTube search.

---

[1] Process Monitor - http://technet.microsoft.com/en-us/sysinternals/bb896645

[2] VMware Workstation - http://www.vmware.com/products/workstation/

[3] X-Ways - http://www.x-ways.net/forensics/

[4] NetAnalysis - http://www.digital-detective.co.uk/netanalysis.asp

[5] Nirsoft tools - http://www.nirsoft.net/

The purpose of testing these different behaviours is to examine the variation in artefacts left by a user visiting a YouTube site in different ways. Hargreaves [14] discusses the importance of demonstrating that a user had intent to visit a particular website. In this case there may be a question over whether the user viewed a page containing a video as a result of following a link from another page, a link in an instant message or email, from a suggested video, or as a result of using a particular search term.

3.5 Automation

The final stage of this research is the development of a tool that automatically examines artefacts left (determined from the earlier experiments) and produces a summary describing the interaction of a user with YouTube, including any specific behaviour that it is possible to infer. The tool also must provide full explanation of how any inferences are made. The tool is tested against other virtual machine disk images where known actions were performed, and the results compared to determine if the inferred behaviour mirrors reality.

4.0 Results

4.1 Overview

This section describes the results that were obtained using the methodology described in the previous sections, i.e. the identification of artefacts, inference of behaviour and automation of event reconstruction. This section is divided up to correspond with these stages in the methodology.

4.2 Artefact Identification

As described in Section 3.3, videos were viewed on YouTube using a Virtual Machine (VM) of Windows 7. As a result of the examination of the VM disk images, a number of key artefacts were identified that relate to a user visit to YouTube. These can include a URL that incorporates the video ID, a cached video file, the video name and references to Google ads. While all these artefacts can be found, it should be noted that not all of them are always present. The following sub-sections describe the artefacts in more detail.

4.1.1 YouTube 'watch' URL

The first of these artefacts is the YouTube URL. The URL primarily identifies the address of the video that was accessed by the user and can be divided into two sections. The URL identified in Figure 1 is described in detail below.

Figure 1: The 'basic' YouTube URL

The first section of the URL identifies the domain name for the YouTube website i.e. '*http://www.youtube.com/'*. The second section identifies the unique video ID

for the video that was accessed, e.g. '*watch?v=m8S718JvwX8'*. In this example the unique video ID is '*m8S718JvwX8'*. This URL can be identified and extracted from the index.dat files that make up the history of Internet Explorer (shown in Figure 2) and can be found in the master, daily or weekly index.dat files[6].

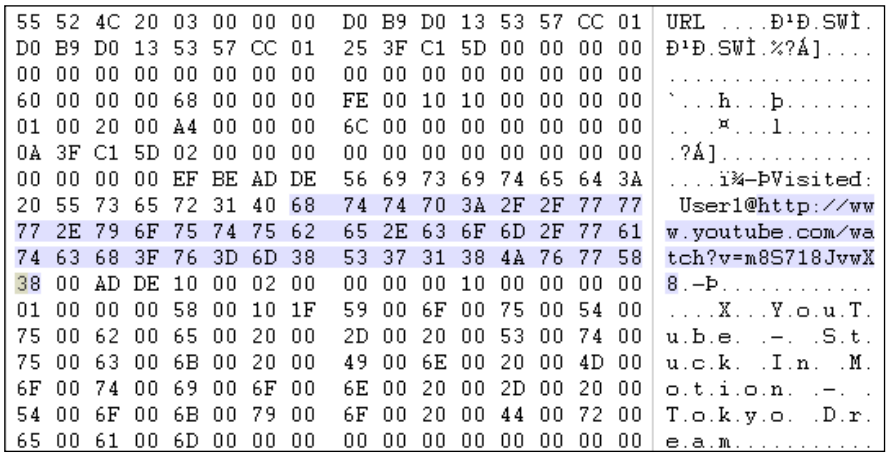

Figure 2: Example of a YouTube video URL in the master index.dat

4.1.2 Video file

When a YouTube video is accessed, it is possible that a copy of the video is stored within one of the sub-folders in the Temporary Internet Files cache. It is given the name *videoplayback* and is usually assigned a version number, e.g. *videoplayback[1]*. It is not clear at this time what determines if a video is definitely stored, how long these videos remain in the cache and under what circumstances they are deleted.

4.1.3 Video name

The title assigned to a YouTube page that contains a video can also be extracted from within the index.dat in the History folder. The title is stored within the same record as the page URL. For example in Figure 3 the title for the page is '*YouTube - Stuck In Motion - Tokyo Dream'* and the title of the video can be obtained by removing the initial 'YouTube' string.

[6] However, care must be taken when extracting dates and times from these files as they are not consistent across different types of index.dat.

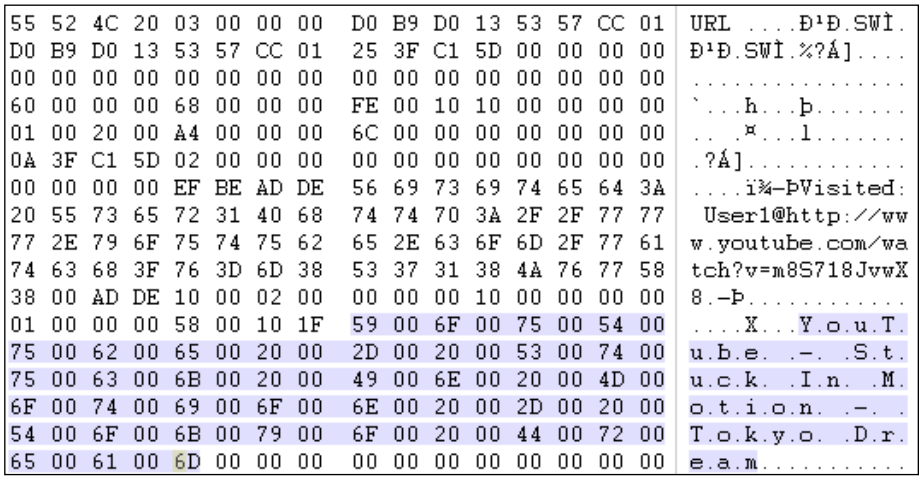

Figure 3: Example of a YouTube video title in index.dat

4.1.4 Google Ad

A Google Ad artefact may also be present relating to a specific YouTube video access. The artefact is located within the index.dat file for the Temporary Internet Files and the URL contains one or more references to the video ID of the video, which is shown in Figure 4.

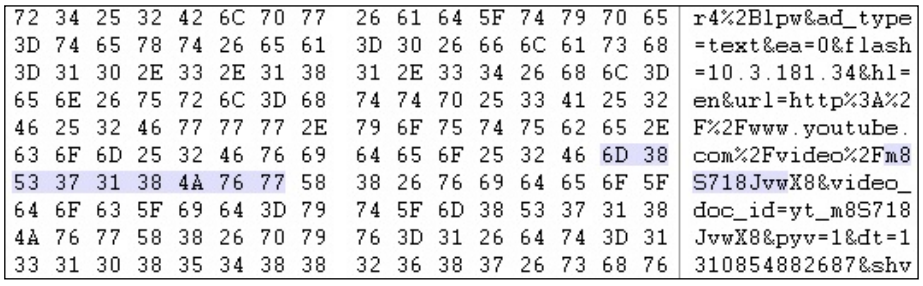

Figure 4: The Cache index.dat showing the same video ID

4.2 User Behaviour

As described in Section 3.4 there are various user behaviours that can be inferred from the artefacts identified as a result of a YouTube visit. Behaviours including searches, video accesses and watching related videos are discussed in this section. Figure 5 shows the output of the custom timelining tool showing a set of file and index.dat changes[7].

[7] Some of the changes are omitted and paths shortened to maintain the clarity of the figure.

Date	Time	Type	Details
2011-07-15	09:40:58	MFT - File created	/Program Files/Google/Google Toolbar/Component/GoogleToolbar_32_79A4E6A8AACC0F12.dll
2011-07-15	09:40:58	MFT - File accessed	/Program Files/Google/Google Toolbar/Component/GoogleToolbar_32_79A4E6A8AACC0F12.dll
2011-07-15	09:40:58	MFT- File modified	/Program Files/Google/Google Toolbar/Component/GoogleToolbar_32_79A4E6A8AACC0F12.dll
2011-07-15	09:40:58	Index.dat – Last visited	http://www.youtube.com/watch?v=RCwCw6XbsgY
2011-07-15	09:40:58	Index.dat – Last visited	/Users/User1/AppData/LocalLow/Microsoft/Internet Explorer/DOMStore/index.dat
2011-07-15	09:41:01	MFT - File created	/Low/Content.IE5/LQE9P3FQ/videoplayback[1].mp4
2011-07-15	09:41:01	MFT - File accessed	/Low/Content.IE5/LQE9P3FQ/videoplayback[1].mp4
2011-07-15	09:43:13	MFT- File modified	/Low/Content.IE5/LQE9P3FQ/videoplayback[1].mp4
2011-07-15	09:44:34	Index.dat – Last visited	http://www.youtube.com/results?search_query=wireless+hacking&aq=f
2011-07-15	09:44:34	Index.dat – Last visited	http://i2.ytimg.com/vi/uDhPeee7TM0/default.jpg
2011-07-15	09:44:34	Index.dat – Last visited	http://i1.ytimg.com/vi/DwqnGm4S5oo/default.jpg
2011-07-15	09:44:39	Index.dat – Last visited	http://www.youtube.com/watch?v=jETwvEDaJeQ
2011-07-15	09:44:39	Index.dat – Last visited	http://googleads.g.doubleclick.net/pagead/ads?client=ca-pub-8174875793926223&output=js ……3DjETwvEDaJeQ&fu=4&ifi=1&dtd=172
2011-07-15	09:44:39	Index.dat – Last visited	http://i1.ytimg.com/vi/40S8G2ZuRpE/default.jpg
2011-07-15	09:44:40	MFT - File created	/Low/Content.IE5/NQUV5M1Y/videoplayback[1]
2011-07-15	09:44:40	MFT - File accessed	/Low/Content.IE5/NQUV5M1Y/videoplayback[1]
2011-07-15	09:44:40	MFT - File modified	/Low/Content.IE5/NQUV5M1Y/videoplayback[1]
2011-07-15	09:44:56	Last updated	/CMI-CreateHive{3D971F19-49AB-4000-8D39-A6D9C673D809}/Microsoft/Reliability Analysis/RAC
2011-07-15	09:44:56	MFT - File created	/Low/Content.IE5/LQE9P3FQ/videoplayback[1]
2011-07-15	09:44:56	MFT - File accessed	/Low/Content.IE5/LQE9P3FQ/videoplayback[1]
2011-07-15	09:47:46	MFT- File modified	/Low/Content.IE5/LQE9P3FQ/videoplayback[1]
2011-07-15	09:49:10	Index.dat – Last visited	http://i1.ytimg.com/vi/40S8G2ZuRpE/hqdefault.jpg
2011-07-15	09:49:10	Index.dat – Last visited	http://i3.ytimg.com/vi/FBpfrg3z0TI/hqdefault.jpg
2011-07-15	09:49:18	Index.dat – Last visited	http://www.youtube.com/watch?v=L4iJ5ElIDwc&feature=related
2011-07-15	09:49:18	MFT - File created	/Low/Content.IE5/1N77V0FL/videoplayback[1]
2011-07-15	09:49:18	MFT - File accessed	/Low/Content.IE5/1N77V0FL/videoplayback[1]
2011-07-15	09:49:18	Index.dat – Last visited	http://googleads.g.doubleclick.net/pagead/ads?ca_h=250&url=http%3A%2F%2Fwww.youtube.com%2Fvideo%2FL4iJ5ElIDwc……%3DjETwvEDaJeQ
2011-07-15	09:50:25	Index.dat – Last visited	PrivacIE:ytimg.com/vi/nYPxZt2naK0/*/default.jpg
2011-07-15	09:50:25	Index.dat – Last visited	PrivacIE:ytimg.com/vi/pye_hIg1ymM/*/default.jpg
2011-07-15	09:52:30	MFT – File modified	/Low/Content.IE5/1N77V0FL/videoplayback[1]

Figure 5: Timeline of file and index.dat changes

The following sub-sections explain how certain user behaviour can be inferred from these results.

4.2.1 YouTube video access

As described in the previous sections, when a user visits a YouTube video page there are a number of artefacts that are created on their system. Figure 6 shows an example of the artefacts that may be present after such a visit.

2011-07-15	09:40:58	Index.dat – Last visited	http://www.youtube.com/watch?v=RCwCw6XbsgY	← Video access URL
2011-07-15	09:40:58	Index.dat – Last visited	/Users/User1/AppData/LocalLow/Microsoft/Internet Explorer/DOMStore/index.dat	
2011-07-15	09:41:01	MFT - File created	/Low/Content.IE5/LQE9P3FQ/videoplayback[1].mp4	← Creation of videoplayback
2011-07-15	09:41:01	MFT - File accessed	/Low/Content.IE5/LQE9P3FQ/videoplayback[1].mp4	
2011-07-15	09:43:13	MFT- File modified	/Low/Content.IE5/LQE9P3FQ/videoplayback[1].mp4	

Figure 6: First video access

The first and most common artefact for a 'basic' video access is the video page URL. This is stored within an index.dat in Internet Explorer's History folder. Within the same record is the video ID, title and the time and date that video was last visited. Having both the URL and the video title provides more detail that can be used to determine the nature of the video accessed by the user.

A videoplayback file may also be present within the cache of the system that relates to the video access. If present, it is a copy of the YouTube video accessed by the user[8]. The videoplayback metadata does not contain any information relating to the associated video ID, title or URL, so currently, the only method of confirming that the video file is related is by comparing the contents of the video. As a result, if the file has been deleted (but the MFT entry is still recoverable) it is not currently possible to conclusively link the videoplayback file to the other artefacts.

Note that in this case the Google Ad artefact mentioned in Section 4.1.4 is not present.

4.2.2 YouTube video access as a result of a suggested video

As discussed in Section 4.1.1 when a user watches a video, a copy of the URL for that video is stored in the index.dat. While the example in the previous section contains no indication of how the user navigated to the page, in some cases the structure of the URLs can be used to infer specific user behaviour. Figure 7 shows a slightly more complex example of a URL containing additional information.

2011-07-15	09:49:18	Index.dat – Last visited	http://www.youtube.com/watch?v=L4iJ5ElIDwc&feature=related	← Suggested video URL
2011-07-15	09:49:18	MFT - File created	/Low/Content.IE5/1N77V0FL/videoplayback[1]	← Creation of videoplayback
2011-07-15	09:49:18	MFT - File accessed	/Low/Content.IE5/1N77V0FL/videoplayback[1]	
2011-07-15	09:49:18	Index.dat – Last visited	http://googleads.g.doubleclick.net/pagead/ads?ca_h=250&url=http%3A%2F%2Fwww.youtube.com%2Fvideo%2FL4iJ5ElIDwc……%3DjETwvEDaJeQ	← Google Ad URL
2011-07-15	09:50:25	Index.dat – Last visited	PrivacIE:ytimg.com/vi/nYPxZt2naK0/*/default.jpg	
2011-07-15	09:50:25	Index.dat – Last visited	PrivacIE:ytimg.com/vi/pye_hlg1ymM/*/default.jpg	
2011-07-15	09:52:30	MFT – File modified	/Low/Content.IE5/1N77V0FL/videoplayback[1]	

Figure 7: Example URL accessed as a result of a suggested video

[8] There may also be additional video content, e.g. advertisements before the actual video played.

This URL holds information relating to how the URL was navigated to. The additional string that reads *'&feature=related'* and is the result of the user clicking on a suggested video link as seen in Figure 8. There are a number of different 'feature' topics that can be appended to a URL, e.g. 'top music' appears as 'feature=topvideos_music'.

Also in this example, both the videoplayback file and the associated Google Ad cache artefact are present.

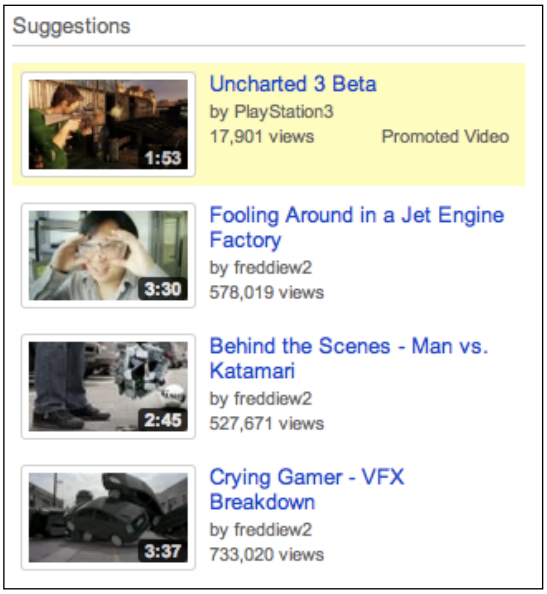

Figure 8: YouTube suggestions column

4.2.3 YouTube search

The previous section shows one way in which a YouTube video could be reached (clicking a link for a suggested video). Another route is as a result of a YouTube search. When a user searches for a video using YouTube they are directed to a page of video results related to their search term and the URL for this page is added to an index.dat file within the History folder. The following URL in Figure 9 is an example result of a user's search for 'wireless hacking tools'.

```
http://www.youtube.com/results?search_query=wireless+hacking+tools&aq=f
```

Figure 9: Example of a manually created YouTube search URL

The first part of this URL is similar to the video URL discussed in Section 4.1.1, in that it begins with YouTube's domain name. The second part of the URL is specific to the search carried out by the user, containing the actual search term i.e. *search_query=wireless+hacking+tools.* The final part of the URL is related to

YouTube's auto-complete query option and provides further insight into the user's actions including whether they typed the complete search term or typed part and selected the full term from YouTube's drop down search menu. In the previous example in Figure 9, the complete search term was typed by the user. The URL in Figure 10 is an example of a search that was partially typed by the user and then automatically completed by clicking on one of the suggested search terms.

http://www.youtube.com/results?search_query=wireless+hacking+tools&aq=6&oq=wireless+hack

Figure 10: Example of a partially automated YouTube search URL

In this example the search term executed was 'wireless hacking tools'. The following part of the URL, '*&aq=6',* indicates which search term was selected from the drop down menu. In this case it was the seventh suggestion as numbering starts from zero. This can be seen in Figure 11. The final part of this URL, '*&oq=wireless+hack',* indicates how much of the search term was physically typed in by the user before selecting the suggested search term. In this case 'wireless hack' was typed, also shown in Figure 11.

Figure 11: Choosing a suggested search term in YouTube

4.2.4 YouTube video access as a result of a search

Unlike the example provided of viewing a page as a result of a suggested video, accessing a video as a result of a search does not contain any information in the URL that can easily demonstrate this. To identify this behaviour, displaying the file and index.dat changes in a timeline is essential since a video viewed immediately after the result of a search can be said to be likely to be viewed as a result of that search, particularly if the video title is examined and shown to be related. This is shown in Figure 12.

2011-07-15	09:44:34	Index.dat – Last visited	http://www.youtube.com/results?search_query=wireless+hacking&aq=f	← Search URL
2011-07-15	09:44:34	Index.dat – Last visited	http://i2.ytimg.com/vi/uDhPeee7TM0/default.jpg	
2011-07-15	09:44:34	Index.dat – Last visited	http://i1.ytimg.com/vi/DwqnGm4S5oo/default.jpg	
2011-07-15	09:44:39	Index.dat – Last visited	http://www.youtube.com/watch?v=jETwvEDaJeQ	← Video access URL
2011-07-15	09:44:39	Index.dat – Last visited	http://googleads.g.doubleclick.net/pagead/ads?client=ca-pub-8174875793926223&output=js ……3DjETwvEDaJeQ&fu=4&ifi=1&dtd=172	← Google Ad URL
2011-07-15	09:44:39	Index.dat – Last visited	http://i1.ytimg.com/vi/40S8G2ZuRpE/default.jpg	
2011-07-15	09:44:40	MFT - File created	/Low/Content.IE5/NQUV5M1Y/videoplayback[1]	← Creation of videoplayback
2011-07-15	09:44:40	MFT - File accessed	/Low/Content.IE5/NQUV5M1Y/videoplayback[1]	
2011-07-15	09:44:40	MFT - File modified	/Low/Content.IE5/NQUV5M1Y/videoplayback[1]	

Figure 12: Video access as a result of a search

Furthermore, while cached copies of the search results page have not been found, if they were recovered, the source of the search results page could be examined to determine any hyperlinks to the videos that were subsequently viewed.

4.3 Automation

The previous sections described the artefacts left by particular behaviours related to YouTube video access. This section discusses the extent to which this reasoning can be automated.

Using a Python based prototype tool, the sequence of extracted artefacts' metadata was searched for the presence of the various sets of artefacts described in the previous section. For example in the case of the 'basic' YouTube Video access, a regular expression can be applied to each of the URLs recorded to detect the 'watch' URL:

```
re.search(r"http://[a-zA-Z0-9]*?\.youtube\.com/watch\?v=([^&]+)", url)
```

If found then it was also possible to test for the videoplayback file:

```
re.search(r".*/(videoplayback\[{1}[0-9]].*)$", path)
```

and to test for the Google Ad, within a similar time period:

```
re.match(r"(http://googleads.g.doubleclick.net)(.+)(video_doc_id=yt_)(.+?)&", url)
```

Similar searches were conducted for the YouTube searches and for videos that were watched as a result of a suggestion from another page. The output of this automated process is shown in Figure 13, where the results have been automatically compiled into an html document.

13:36:32	YouTube search	keylogger
13:37:14	YouTube video accessed	Title: * The best keylogger for free (Very easy to use !!!), - YouTube
13:38:05	YouTube video accessed (from Suggestions)	Title: * Facebook Keylogger, - YouTube
13:38:51	YouTube video accessed (from Suggestions)	Title: * How To Make KEYLOGGER, - YouTube
13:38:52	YouTube video accessed (from Suggestions)	Title: * How To Make KEYLOGGER, - YouTube
13:47:22	YouTube search	how to install a keylogger
13:47:43	YouTube video accessed	Title: No title found
13:47:45	YouTube video accessed	Title: * Install Undetectable & Untraceable Keylogger, - YouTube

Figure 13: The output from the automated process

A summary of the actual actions performed as logged during the experiment were:

13:35 Launched Internet Explorer
13:36 The address www.youtube.com typed into the address bar
13:36 YouTube search for 'keylogger'
13:37 "best keylogger ever" video partially watched
13:38 "Facebook keylogger" video partially watched from suggested
13:38 "How to make a keylogger" video watched in full from suggested
13:47 YouTube search for "install a keylogger" typed and "how to install a keylogger" selected from options presented.
13:47 "Install undetectable and untraceable keylogger" video watched in full
13:52 Closed Internet Explorer
13:53 Windows shutdown

As can be seen above, the events detected correlate with the actual events that occurred. In addition, as described in Section 2.3, it important to maintain an audit trail of how the results of an automated analysis tool are obtained. An extract from the log file is shown in Figure 14.

```
16:04:49 ------------------------------------
16:04:49 Matched 'watch' string with http://www.youtube.com/watch?v=AEjtXoxNnu8
16:04:50 Looking between 13:47:44 and 13:48:25 for additional artefacts...
16:04:50 Searching for videoplayback files...
16:04:50 Matched a videoplayback file in /Users/User1/AppData/Local/Microsoft/Windows/Tem
16:04:50 Searching for related Google Ad...
16:04:50 Found a google ad url in Users/User1/AppData/Local/Microsoft/Windows/Temporary I
16:04:50 Matched url was http://googleads.g.doubleclick.net/pagead/ads?client=ca-pub-8174
16:04:50 Checked ID and matched from watch(AEjtXoxNnu8) with Ad result (AEjtXoxNnu8)
16:04:50 Added YouTube event to list (YouTube video accessed)
16:04:50 ------------------------------------
```

Figure 14: Output from the log file created while searching for YouTube related events

The information recorded in the log file makes it relatively straightforward to manually examine the related artefacts if necessary.

5.0 Evaluation

This section evaluates the research conducted. Each of the stages of the research is examined in turn.

5.1. Artefacts Recovered

This research has examined the artefacts left by visiting YouTube using Internet Explorer 8 on Windows 7 only. There are many other permutations of operating system and browser that are likely to produce artefacts in different locations and different formats. This could include other web browsers such as Firefox, Chrome, Safari and Opera, on different operating systems such as Windows XP, Vista and possibly Mac OS X and Linux. It has also not considered the artefacts left on dedicated apps such as those found on iOS. In addition, while the file system, Registry and index.dats are examined for artefacts, other locations that can contain relevant artefacts were not parsed e.g. Windows Search database.

5.2 User Behaviour

The behaviours examined were:

- The viewing of a video,
- The viewing of a video as a result of a YouTube suggested video,
- A YouTube search,
- The viewing of a video as a result of a YouTube search.

There are obviously other behaviours associated with YouTube video access that have not yet been explored, e.g. viewing a YouTube video embedded in another web site, or multiple visits to the same website. Nevertheless, the methodology used to determine the artefacts left can be easily applied to other behaviours.

5.3 Automation

The initial results of the automation are promising. Scanning for the identified artefacts has proven successful in detecting different types of visits and of YouTube searches. The tool also logs the tests performed and the reasoning for the output produced and makes it straightforward to check both the reasoning and for the presence of (or lack of) artefacts detected. However, it is important to remember that while it is relatively straightforward to perform an action and determine the artefacts left, care must be taken in stating that something definitely happened because certain artefacts are present. This is because there may be alternative explanations for the same set of artefacts. However, the artefacts found would be consistent with the inferred behaviour and this does not negate the validity and value of this approach. In fact the automation of the process combined with thorough logging means that larger numbers of artefacts that support a particular inference could be examined, which could reduce the chances of reaching incorrect conclusions.

6.0 Conclusions & Future Work

This research has identified several artefacts of interest related to YouTube activity and it has shown how they relate to user actions. In terms of future work, as discussed in the evaluation section, there is much work to be done on other operating systems and using different web-browsers. There is also much more research that can be performed to further understand the makeup of the YouTube URL strings and what else can be inferred from them. Also the artefacts deposited by uploading videos to YouTube may be of interest.

Finally, the research has shown that some automation in terms of summarising the activity on a computer system is possible, in this case the use of YouTube. However, detailed logging and transparent reasoning for conclusions being drawn is believed to be essential and requires further work. Nevertheless, given the increasing volumes of data, increasing numbers of digital devices and in most cases limited resources, having some indication of which specific systems need to be prioritised is desirable.

References

1. *Statistics*, www.youtube.com/t/press_statistics (visited July 2011)
2. *YouTube Community Guidelines*, http://www.youtube.com/t/community_guidelines (visited July 2011)
3. Nikkel, B. (2006), The Role of Digital Forensics within a Corporate Organization, IBSA Conference, Vienna.
4. Shariff, S. (2008), Cyber-bullying: Issues and Solutions for the School, the Classroom and the Home. Routledge.
5. Casey E, *Digital Evidence and Computer Crime*. Elsevier 2004, ISBN 0-12-163104-4
6. Rogers, M.K. and Goldman, J. and Mislan, R. and Wedge, T. and Debrota, S. (2006) Computer forensics field triage process model, Proceedings of the Conference on Digital Forensics Security and Law
7. ACPO (2007) Good Practice Guide for Computer-Based Electronic Evidence
8. Carrier (2002) Defining Digital Forensic Examination and Analysis Tools, Digital Forensics Research Workshop II
9. Sureka et al (2010).Mining YouTube to Discover Extremist Videos, Users and Hidden Communities, *Lecture Notes in Computer Science.* 6458, 13-24.
10. Adelstein F, Joyce R (2007) File Marshal: Automatic extraction of peer-to-peer data, *Digital Investigation.* 4, 43-48.
11. Hargreaves C, & Chivers H, (2010) A Virtualisation Based Computer Forensic Research Tool. Cybercrime Forensics Education and Training Conference Canterbury, UK.
12. Norris P. (2009) The Internal Structure of the Windows Registry, MSc Thesis, Cranfield University, http://amnesia.gtisc.gatech.edu/~moyix/suzibandit.ltd.uk/MSc/

13. Jones, K. (2003) Forensic Analysis of Internet Explorer Activity Files, http://www.mcafee.com/us/resources/white-papers/foundstone/wp-pasco.pdf
14. Hargreaves, (2010) Establishing Context When Investigating a Suspect's Internet Usage. Proceedings from 3rd Cybercrime Forensics Education & Training. Canterbury Christ Church University, Canterbury, UK.

Automated identification and reconstruction of YouTube video access

Patterson, J.

2011-09-01T00:00:00Z