CHAMPLAIN COLLEGE

LCDI

The Senator Patrick Leahy
Center for Digital Investigation

Forensic Acquisition of Websites (FAW) Tool Review

Written by
Nicholas Aspinwall
Researched by
Nicholas Aspinwall

175 Lakeside Ave, Room 300A
Phone: 802/865-5744
Fax: 802/865-6446
http://www.lcdi.champlin.edu

March 27, 2014

Disclaimer:

This document contains information based on research that has been gathered by employee(s) of The Senator Patrick Leahy Center for Digital Investigation (LCDI). The data contained in this project is submitted voluntarily and is unaudited. Every effort has been made by LCDI to assure the accuracy and reliability of the data contained in this report. However, LCDI nor any of our employees make no representation, warranty or guarantee in connection with this report and hereby expressly disclaims any liability or responsibility for loss or damage resulting from use of this data. Information in this report can be downloaded and redistributed by any person or persons. Any redistribution must maintain the LCDI logo and any references from this report must be properly annotated.

Contents

Introduction

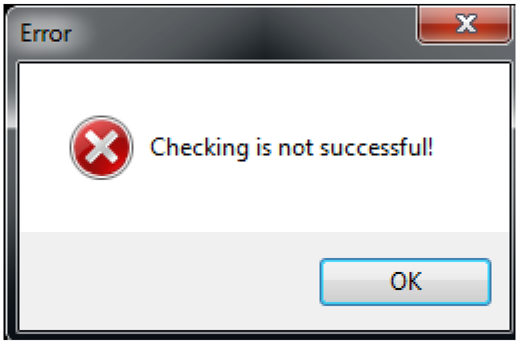
Websites and webpages contain information that can be incredibly valuable to an investigator. Forensic Acquisition of Websites (FAW) is a way to forensically acquire a website or webpage as it is viewed by the user. FAW preserves what is publicly available at the time. It is a helpful tool in non-solicitation cases (where an employee has violated the non-solicitation clause by posting about a project on Facebook) as it will preserve the evidence before it can be taken down by the user. .

There are a number of forensic tools that are able to retrieve Internet artifacts on a system or a system image. Internet Evidence Finder by Magnet Forensics[1] allows investigators to view Internet artifacts, such as social networking sites, webpages, and chat logs that could be stored on a system. FAW in unique in that, unlike IEF, this is a tool that can be used to capture posts and evidence in a live environment. FAW keeps a log of all the actions that take place during the acquisition in real time.

In order to capture a webpage with FAW, the investigator must go to the webpage in a live environment. This tool would be used in a scenario where an investigator may want to take a screenshot of a webpage as evidence of a crime such as cyber terrorism, narcotics sales, cyberbullying, etc. For example, an active investigation of a narcotics dealer may be ongoing, where his or her social media presence is being monitored. If the suspect tweets about the latest substance he has for sale, then that tweet and the entire page it is viewed on may be captured with FAW.

FAW allows the investigator to take a screenshot of the tweet, at the same time also recording many other useful artifacts, such as iFrames, advertisments, links, and streaming data. FAW records and logs every action within the tool for forensic documentation purposes. FAW also hashes the objects that are acquired. Each hash is stored in a text document, and the document is then hashed and stored in a checking file to determine if the hash list file has been altered. This ensures that files cannot be changed or added to the case without the hash list file being changed to accommodate the alteration. If the hash list is altered, this can then be checked using FAW's checking function. **Figure 1** shows the command box within the checking function after either the hash list file or the checking file was altered. **Figure 2** shows the command box if neither is altered. It is still important that the investigator takes care that no files are added to the directory as the checking function only checks the two aforementioned files (hash list and the checking file).

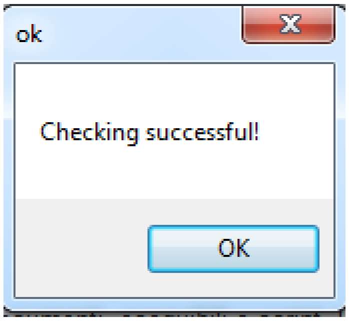
Figure 1



Checking is not successful!

[1] http://www.magnetforensics.com/software/internet-evidence-finder/

Figure 2


ok

Checking successful!

FAW is also helpful when analyzing webpages infected with malicious code. A common way to introduce malware into a system is through the use of JavaScript files, or frames that have no visible size. FAW is capable of capturing both frames and JavaScript files. Even if the frame or file is not viewable in the screenshot, they can still be analyzed.

FAW is a tool designed primarily to provide screenshots with additional supporting data. It can capture iFrames, advertisments, links, and streaming data, among other artifacts. FAW also has built-in features that make it very useful in recreating events. Before a user deletes a post on a social networking site, FAW provides a way to capture what was posted. Not only can it grab social networking posts, but it is also capable of capturing streaming video and the frames on the web page[2]. Throughout our work, we will be looking at how effectively FAW captures this data, and how useful it can be in an investigation.

Background:

Currently there is not a large amount of research pertaining specifically to FAW. There has been some discussion of the tool on Forensic Focus; however, there is no documentation found other than the FAW website, fawproject.com.

Purpose and Scope:

The intent of our research is to comprehensively evaluate the tool. As there is little research on this specific tool, our research will provide a source of reference for the tool's abilities and claims. We will be discussing the experience of using the application and how useful the data captured could be in an investigation. We will then compare FAW to other internet artifact tools.

Research Questions:

1. Does FAW forensically acquire websites?
2. What data is captured and how can this be helpful in an investigation?
3. Does FAW provide accurate and repeatable artifacts?
4. How does FAW differ/compare to other web artifact forensic tools?

[2] www.fawproject.com/ en/characteristics.aspx

Terminology:

FAW – Forensic Acquisition of Webpages – tool used to capture live webpages.
Frame – A frame is used to divide webpages into different sections, mainly for design purposes.
Hash – An algorithm used to generate a unique string based on the data of the file.
RSS – Rich Site Summary - Web Feed format – A format to present frequently updated data to the user.
User-agent – the string used to identify the browser being used.
XML – Extensible Markup Language (XML) is the human and machine readable document.

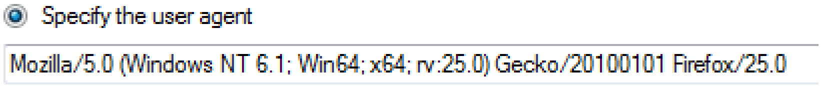
Methodology and Methods

To conduct research, a virtual machine running Windows 7 Pro 64bit and the current version of Internet Explorer was set up (Table 1: Virtual Machine Settings).   FAW version 2.1.0.0 was downloaded and installed on the machine. The default configuration of FAW uses the user-agent (a string sent to the webserver) found with the installed version of Internet Explorer. Internet Explorer version 11 was installed on the VM. In order to replicate a different browser, the user-agent must be specified. With FAW, it is possible to directly type the user-agent in, rather than selecting from a provided list. We used the user-agent for Firefox 25.0 (**Figure 3**) and Chrome 32.0.1667.0 (**Figure 4**). **Figure 5** Shows the default user-agent settings, which uses the version of Internet Explorer installed on the host OS. FAW can also narrow down the objects that the investigator wants collected. For this project, we had FAW capture as much data as possible.

Table 1: Virtual Machine Settings

VM Machine	
RAM	4GB
Processors	4
OS	Windows 7 Pro 64 Bit Service Pack 1
VM Ware	VMware Workstation 10.0.0
LCDI System	Research 11

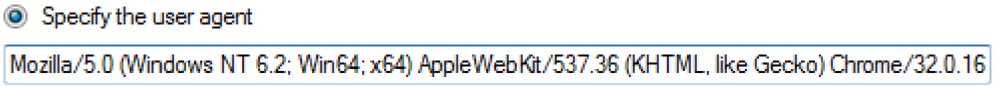
Figure 3



● Specify the user agent

Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:25.0) Gecko/20100101 Firefox/25.0

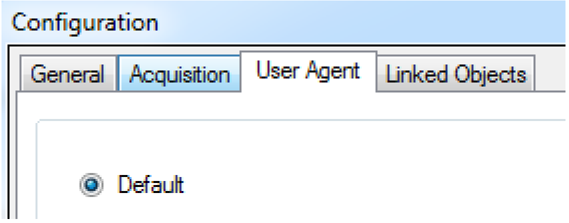
Figure 4



● Specify the user agent

Mozilla/5.0 (Windows NT 6.2; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.16

Figure 5



● Default

Equipment Used

Our host machine was a Windows 7 Pro 64 bit machine with Service Pack 1. This hosted the Virtual Machine that had FAW installed (2.1.0.0). The virtual machine was also Windows 7 Pro 64bit with Internet Explorer version 11. This is the user-agent that was used by FAW for the default settings.

System	OS	Version	Additional Info	RAM
Host Machine	Windows 7	Pro 64 bit	Service Pack 1	16gb
Virtual Machine	Windows 7	Pro 64 bit	Service Pack 1	4gb
Vm Ware	VmWare Workstations	10.0.0		
FAW		2.1.0.0		

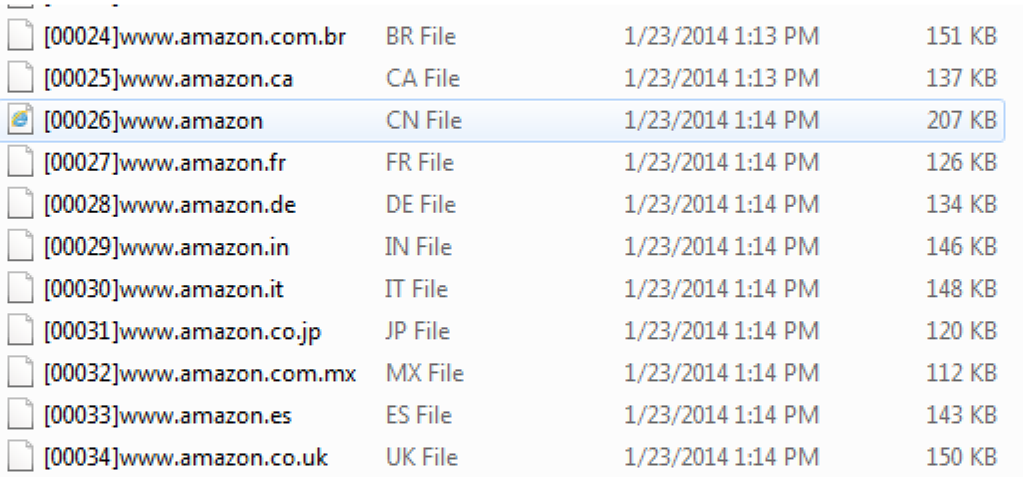
Data Collection:

For each browser user-agent string the same process was followed. Each webpage visited was designated as a new "case" to ensure that all data was separated. A single VM was used, so each time the user-agent string was simply changed within the tool.

FAW's website reports that the tool is capable of collecting a number of data sources from a single webpage. The table below (Table 2: Number of Files from Objects Folder) displays the number of files collected from each website. FAW is able to specify a User-Agent in order to mimic different browsers. Table 3: User-agent Strings shows the user-agent used for each browser. FAW was able to extract images in the form of JPEG, GIF, PNG, and icon files. The tool was also able to retrieve java-script files and MS-DOS Application files, along with AU formatted audio files and RSS files.

Table 2 contains a category labeled "other files." This category encompasses a range of file types, such as Cascading style sheets, HTML, and XML files. Several of these files are html files that Windows doesn't recognize because they are missing file extensions. Many of the HTML files are versions of the webpage in a different language or from a different region. **Figure 6** shows examples from the Amazon results. There is one file with a ".jp" extension which Windows incorrectly labels as a JP file. The file is actually the html file of Amazon in Japanese. There are several additional files with unknown file types.

Figure 6



[00024]www.amazon.com.br	BR File	1/23/2014 1:13 PM	151 KB
[00025]www.amazon.ca	CA File	1/23/2014 1:13 PM	137 KB
[00026]www.amazon	CN File	1/23/2014 1:14 PM	207 KB
[00027]www.amazon.fr	FR File	1/23/2014 1:14 PM	126 KB
[00028]www.amazon.de	DE File	1/23/2014 1:14 PM	134 KB
[00029]www.amazon.in	IN File	1/23/2014 1:14 PM	146 KB
[00030]www.amazon.it	IT File	1/23/2014 1:14 PM	148 KB
[00031]www.amazon.co.jp	JP File	1/23/2014 1:14 PM	120 KB
[00032]www.amazon.com.mx	MX File	1/23/2014 1:14 PM	112 KB
[00033]www.amazon.es	ES File	1/23/2014 1:14 PM	143 KB
[00034]www.amazon.co.uk	UK File	1/23/2014 1:14 PM	150 KB

Table 2: Number of Files from Objects Folder

Browser	Images (JPEG, GIF, etc.)	Java-Script Files	Application Files	Sound/Video Files	RSS	Other Files*	Total
Internet Explorer							
Amazon	16	1	34	1	0	20	72
Woot	42	3	2	0	2	50	99
LCDI	4	2	0	0	0	12	18
Chrome							
Amazon	16	1	34	1	0	28	80
Woot	42	3	2	0	2	50	99
LCDI	4	2	0	0	0	12	18
FireFox							
Amazon	16	1	34	1	0	28	80
Woot	42	3	2	0	2	50	99
LCDI	4	2	0	0	0	12	18

*Other Files contains files with missing file extensions.

Table 3: User-agent Strings

Browser	User-Agent String

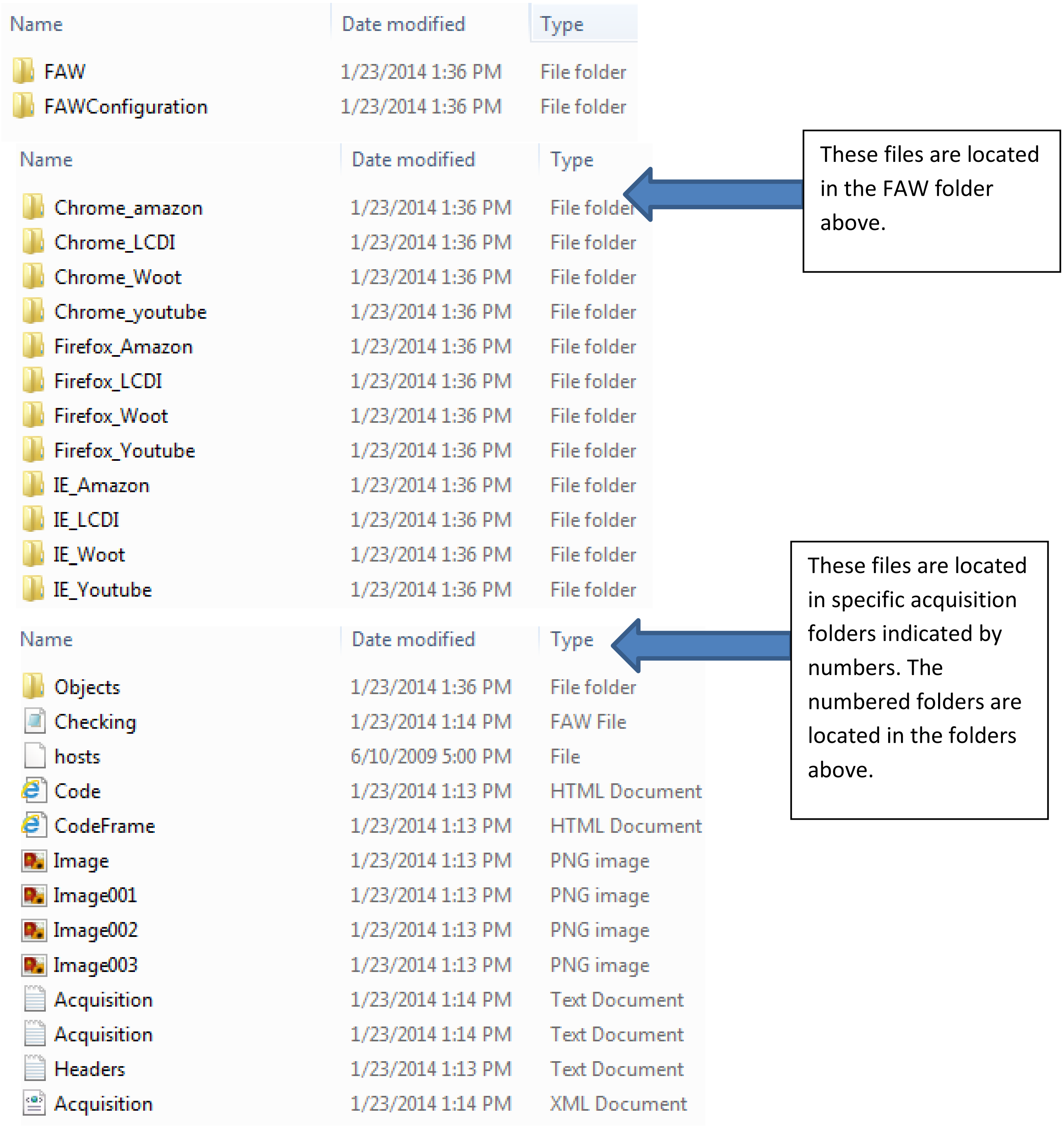
Internet Explorer	Default. Uses the user-agent of Internet Explorer that is currently installed on Host
Google Chrome	Mozilla/5.0 (Windows NT 6.2; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1667.0 Safari/537.36
Mozilla Firefox	Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:25.0) Gecko/20100101 Firefox/25.0

Analysis

File Hierarchy and Data

Figure 7 shows the files and folder that FAW creates during a case. For each browser, the file structure was identical and the majority of data was as well. **Figure 7** is a series of screenshots of the file structure that FAW creates.

Figure 7



Name	Date modified	Type
FAW	1/23/2014 1:36 PM	File folder
FAWConfiguration	1/23/2014 1:36 PM	File folder

Name	Date modified	Type
Chrome_amazon	1/23/2014 1:36 PM	File folder
Chrome_LCDI	1/23/2014 1:36 PM	File folder
Chrome_Woot	1/23/2014 1:36 PM	File folder
Chrome_youtube	1/23/2014 1:36 PM	File folder
Firefox_Amazon	1/23/2014 1:36 PM	File folder
Firefox_LCDI	1/23/2014 1:36 PM	File folder
Firefox_Woot	1/23/2014 1:36 PM	File folder
Firefox_Youtube	1/23/2014 1:36 PM	File folder
IE_Amazon	1/23/2014 1:36 PM	File folder
IE_LCDI	1/23/2014 1:36 PM	File folder
IE_Woot	1/23/2014 1:36 PM	File folder
IE_Youtube	1/23/2014 1:36 PM	File folder

These files are located in the FAW folder above.

Name	Date modified	Type
Objects	1/23/2014 1:36 PM	File folder
Checking	1/23/2014 1:14 PM	FAW File
hosts	6/10/2009 5:00 PM	File
Code	1/23/2014 1:13 PM	HTML Document
CodeFrame	1/23/2014 1:13 PM	HTML Document
Image	1/23/2014 1:13 PM	PNG image
Image001	1/23/2014 1:13 PM	PNG image
Image002	1/23/2014 1:13 PM	PNG image
Image003	1/23/2014 1:13 PM	PNG image
Acquisition	1/23/2014 1:14 PM	Text Document
Acquisition	1/23/2014 1:14 PM	Text Document
Headers	1/23/2014 1:13 PM	Text Document
Acquisition	1/23/2014 1:14 PM	XML Document

These files are located in specific acquisition folders indicated by numbers. The numbered folders are located in the folders above.

FAW configuration folder

Configuration file

The configuration file contains the information for a number of settings. The configuration file lists the NTP sever that it connects to, which in our case wastime.windows.com. This file is also where the start page of the browser is kept. By default the start page is "www.fawproject.com/itsdone.aspx." Additionally, this file contains the default directory where acquisitions are saved. By default, this location is in the user documents; however, FAW has an option that allows all acquisitions to be sent by email directly after the capture is completed. This file also contains email addresses with passwords, the SMTP ports, and the TO and FROM fields. This includes a field indicating if SSL is enabled or not.

Application

The application file contains a time stamp and a message of Send" Email failed" The address cannot be empty. If the investigator has asked the tool to send data to a specific email, this file will log the email information. This could prove to be somewhat problematic if some files captured exceed content limits within the email. Additionally, this file holds timestamps and logs for the tool and its loading process.

FAW Folder

Case Directories

FAW creates a directory that it uses to store all of the information that it gathers. Once the investigator names a case, a folder with the case name is created inside the FAW directory. For each "Acquisition" or capture, a new folder is created within that case folder. These folders are named using a numbering system. For our research, we used the yellow box to include the entire web page so there was one acquisition per webpage, per browser.

Numbered Folders

Objects Folder

The objects folder was released with the 2.1.0.0 version of FAW. This folder contains all the objects that FAW is able to capture from each website. This includes java-script files, images, audio, and html. Depending on the scope of the investigation, the file types collected can be narrowed down. For this project, we had FAW capture everything that it could.

Because the file type is determined by the file extension, some files assume the incorrect type due to the naming convection of some files. As an example, certain files have a dot within the name so the OS incorrectly labels that as the file type. The file extension should not prevent FAW from obtaining these files.

FAW is capable of capturing most files that are of interest. It was able to capture other html files, JavaScript files, MS-DOS Application files, Sound and Video Files, and RSS Feed Files among others.

Two Acquisition text files

The first acquisition text file we recovered is the log file of all the actions that took place during the investigation, along with the UTC time stamps for each. The first action that is logged is the creation of the case followed by the start page loading up and the time the load completed in the next entry. From this point on, FAW records all the URLs that are being loaded, including any other plugin that is being loaded from the webpage. Additionally, the start of acquisition mode is recorded along with the time that the acquisition is completed.

The second acquisition file contains the hash value of every file created by FAW during the selected acquisition. It records the URL, User-Agent, Capture Time start, Capture Time end, and the NTP server. If a proxy is in use, the proxy address, the DNS address of the local machine, the Case name or

ID, the Acquisition ID, and the Detective name are also recorded. The Acquisition ID is the Case ID combined with the numbered folder in which the acquisition data is stored. The first hash value is of the Image.png file. Although other image files are created the only one that is hashed is the original image that was screen captured within the yellow box.

Acquisition xml file

The majority of the data from the other files is stored in this XML file, which record the hash of each file created by FAW. Located in this file is also the username of the user on the host system. It records the machine name of the host system, the working directory, the user-agent, the detective's name, the OS system name, OS version, the system architecture of the host, if a proxy is in use, and the DNS of the host machine. This is known as the "Run Environment." This file also contains the build_enviroment section, which records the compilation date in UTC and the compiler used. This file contains the Program Name (FAW) and the current version of the program that is running.

The Source section contains the URL of the page alongside the width and height of the page in pixels, and the acquisition start and end times.

The Configuration section contains the NTP server, the homepage of the program, and the capture folder (where data is saved to).

The results section of this file contains the same results as the second acquisition text file; however, this section also includes the MAC times of each file and the MD5 and SHA1 value for each file.

Checking.faw

This is a file type created specifically for FAW that is used to ensure that the acquisition was completed successful. Once the acquisition is complete, a check acquisition button may be clicked which will check to ensure that all the hashes of the files are verified and that data has not been altered.

Code html page

This is a copy of the code from the page that the capture was taken from. This can used to reload the page exactly as it was seen when the acquisition took place.

Code Frame html page

This is the code from the Frames on the page that was captured. It may sometimes contain code from scripts and advertisement systems.

Headers text file

This is a simple text file that contains the header information of the page.

Hosts File

This is where the host file appears if it is able to be captured. We were able to capture the sample host file every time.

Image PNG file

This file is the image or screenshot of the entire selected area (using the yellow box). The yellow box can be used to stretch past what is visible on the screen, making it possible to capture an entire webpage.

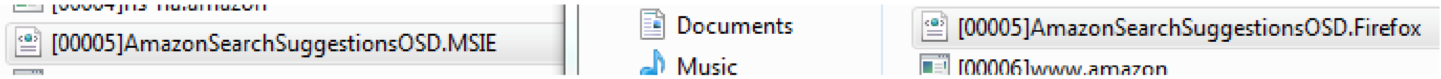
Image00X PNG File

Image00X PNG files are screen captures of each section on the monitor as seen by the user. They are created in such a way that it is as if the user   had scrolled down to a section of the screen, taken a screenshot and then scrolled down to a different section of the screen and taken another screenshot, continuing until the entire page was collected. During an acquisition, the screen will visibly move and take screen shots.

Results

When going through our results, we immediately noticed the difference in the number of objects collected in the “Objects” folder (Table 1: Number of Files from Objects Folder). The Objects folder is where the “objects” that were captured off the webpage are stored. FAW can capture client side effects as well as a number of other file types. Client side effects are classified as data streaming or java script-like applications. It should be noted that when FAW was acting as Internet Explorer and preforming an acquisition on Amazon.com, it captured 8 less files than it did when acting as Google Chrome or Mozilla Firefox. Additionally, among similar files there was a variation in the results. FAW acting as Internet Explorer and Firefox was able to capture different XML documents for both.

Figure 8



[00005]AmazonSearchSuggestionsOSD.MSIE

Documents
Music

[00005]AmazonSearchSuggestionsOSD.Firefox
[00006]www.amazon

Figure 8 shows two files that were acquired. The file ending in .MSIE was captured when FAW was using the Internet Explorer user-agent. The file ending in .Firefox was captured when FAW was using the Firefox User-agent. This file was not captured at all while FAW was using the Google Chrome user-agent.

Potentially, this could be the effect of different style sheet files that were used for each browser. It is also possible that the webpage had changed in-between captures. Additionally, it is possible that a larger advertisement was loading on the page, or other advertisements were loaded to the page. Object 1 is a screenshot of the images captured from Amazon.com from all three browsers. In the screenshot, you can see that each time we visited the website different items were suggested, and therefore different files were captured. Clicking on the object will open the image.

Live Changes.PNG

Object 1

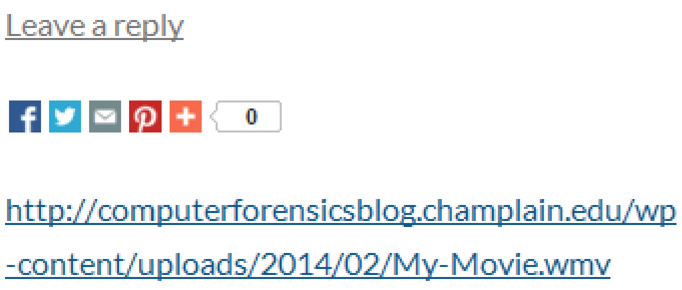
The files captured from a webpage provide a lot of information about that page. First, the entire html code of the webpage is captured, allowing for the webpage to be recreated. Many of the files also allow the investigator to see what pages or files that specific webpage has a connection to. An example of this would be when we acquired Amazon.com; we also captured the other language version of the html file. Some of these files could be used for any number of purposes. Frames and JavaScript files can be captured and analyzed for malware. If a JavaScript file contains malicious code and is downloaded to the host machine, it is possible to capture that file using FAW in a Virtual Machine and then examine it.

Capturing streaming video from a website proved to be a challenge for our team. We wanted to be practical and test the function against the more commonly used streaming sites, so we used YouTube.com and Vimeo.com. We followed the same procedure as the other webpages, but were not able to capture the video files. We tried multiple approaches and were unable to discern what was blocking the capture. We attempted to capture the webpage with video playing, before it was playing, after it was fully loaded, and after it was completely viewed. All approaches were unsuccessful.

Most likely, YouTube and Vimeo have a setup where the player points to the video file so when the user is downloading and watching it, they are not directly interacting with the original video file. It is understandable that these streaming sites would not want users to be able to easily download the file from the website as this could lead to illegal downloads and file sharing, as well as a loss in revenue.

At this point, we chose to upload a video to the LCDI Blog website. We created a post that was a single video under 10mb. When we navigated to the page using a browser on the host machine, a video player similar to QuickTime Player was displayed. When we navigated to the webpage using FAW, we encountered difficulties viewing the player. Instead of a video player, it was just a link to the video file. Figure 9 shows a snippet of how the link was displayed inside FAW. Despite this, FAW was still able to extract the file from the webpage.

Figure 9



Leave a reply

http://computerforensicsblog.champlain.edu/wp-content/uploads/2014/02/My-Movie.wmv

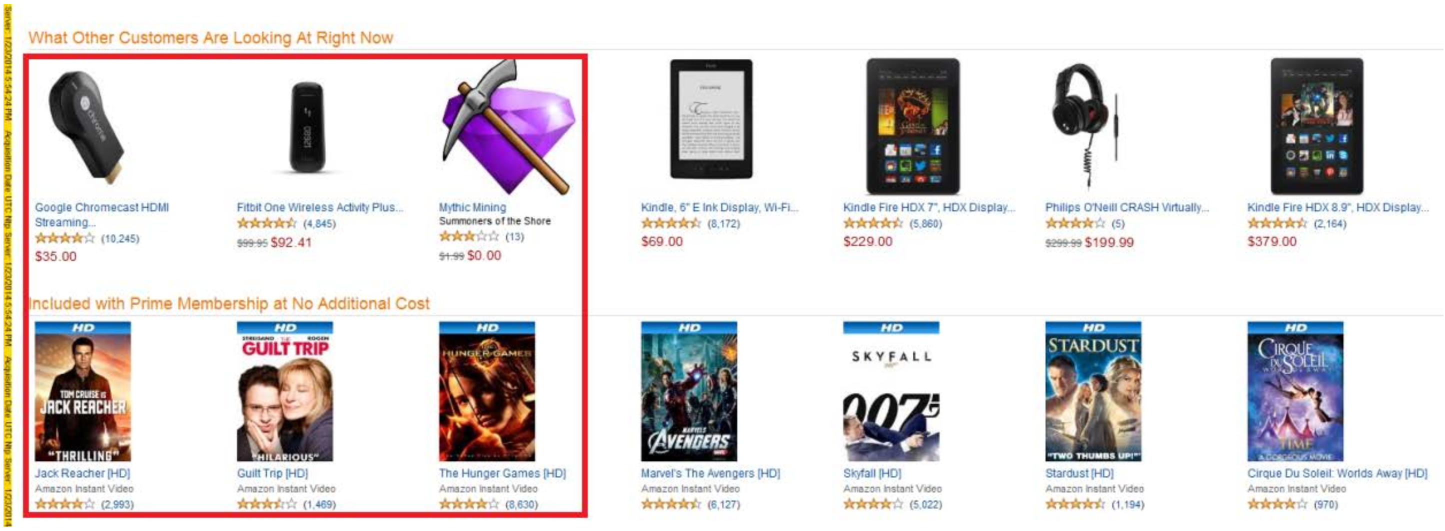
Instead of looking at history or cache files as IEF or Histex, FAW creates an image of a webpage with as many files as it can capture. This approach is similar to Web Page Saver or HTtrack. Programs like Internet History Evidence Finder works by looking for artifacts left on a system, but FAW serves a different purpose. While a post is active and up, FAW can be used to forensically acquire it exactly as it is seen by the user. If the poster takes the post down the next day, then IEF, Encase, or FTK would have to be used to determine if the post was created. FAW would have created a snapshot of the post with timestamps, the code, images, and nearly anything on the page.

Conclusion

FAW is a unique way to forensically acquire web pages. The tool provides a way to preserve the data as it is viewed by the user. This would be beneficial for a number of reasons and in a variety of settings. For example, if an employee signs a no solicitation clause with a company but posts updates to Facebook regarding other companies and employment, the employer, wanting to protect their agreement, can use FAW and navigate to the webpage. The employer can then take a forensically sound screenshot of the post as it is viewed publicly. Even if the next day the employee realizes that the post has been getting unwanted attention and takes it down, with FAW, the company has a record of the post.

Additionally, FAW easily extracts each image file on a webpage that is being viewed. The Woot, Amazon, and LCDI webpages all contained images, including logos, thumbnails, and full images. FAW was able to capture the majority of the images, but could not capture all of the thumbnails from Amazon. FAW was able to capture 6 of the 14 thumbnails for featured items with the Chrome user-agent. Using the Firefox user-agent, 6 of the 14 thumbnails was captured again. Using the default user-agent, acting as Internet Explorer, 6 of the 14 thumbnails were captured. It is unclear why FAW was not able to capture every image. Figure 10 shows the 14 thumbnails as seen in the browser. These 14 images were different for each capture that took place. Regardless of what the image was of, FAW captured the same 6 images from a specific part of the page (Figure 10).

Figure 10


Server: 1/22/2014 5:54:24 PM Acquisition Date UTC Ntp Server: 1/22/2014 5:54:24 PM Acquisition Date UTC Ntp Server: 1/22/2014
What Other Customers Are Looking At Right Now
Google Chromecast HDMI Streaming...
(10,245)
$35.00
Fitbit One Wireless Activity Plus...
(4,845)
$99.95 $92.41
Mythic Mining
Summoners of the Shore
(13)
$1.99 $0.00
Kindle, 6" E Ink Display, Wi-Fi...
(8,172)
$69.00
Kindle Fire HDX 7", HDX Display...
(5,860)
$229.00
Philips O'Neill CRASH Virtually...
(5)
$299.99 $199.99
Kindle Fire HDX 8.9", HDX Display...
(2,164)
$379.00
Included with Prime Membership at No Additional Cost
Jack Reacher [HD]
Amazon Instant Video
(2,993)
Guilt Trip [HD]
Amazon Instant Video
(1,469)
The Hunger Games [HD]
Amazon Instant Video
(8,630)
Marvel's The Avengers [HD]
Amazon Instant Video
(6,127)
Skyfall [HD]
Amazon Instant Video
(5,022)
Stardust [HD]
Amazon Instant Video
(1,194)
Cirque Du Soleil: Worlds Away [HD]
Amazon Instant Video
(970)

In many circumstances, FAW would be a useful tool to back up claims, and may be used to obtain further authorization for an investigation. FAW is capable of capturing a number of files. FAW was able to capture the JavaScript and css files on the site. These can be used to determine if a site contains malware. The tool can also capture Frames and coding, which would further an investigation of malware The tool can be quickly installed to a machine and can be run on varying hardware levels.

The data captured by FAW was shown to be reliable and repeatable. Each of the browsers obtained the majority of the same data with only minor differences. It is unlike other tools, such as Internet Evidence Finder or FTK. Internet Evidence Finder is used on entire systems to recover internet history. It uses previous cache and history files left on the system to see what a user has been doing. It can be used to rebuild webpages and can parse out social networking posts and chats, peer-to-peer data, instant messaging chat logs, images and videos. FAW seems equipped to preserve a live a webpage as it stands at that moment.

Further Work

When more updates are released for FAW, it will be interesting to see what functions will be added. The tool has additional applications beyond forensics, such as preserving public data for archiving or studying site malware behavior, and it would be beneficial to see what information it could extract from a malware infected website. Our team would also like the opportunity to test this tool on different machines, for different browsers.

References

FAW Project. (n.d.). *FAW Project*. Retrieved from http://fawproject.com/en/default.aspx

Authors of FAW:

Davide Bassani:

Davide Bassani works for ATHIX S.r.l. He has worked in the IT industry for over 15 years and specializes in computer security and computer forensics; he is a member of IISFA. He Has worked as a consultant for various companies and has been involved in all aspects on computing, network server, software engineering, Web applications, from IT security to computer forensics, field experience has played a key role in the formation of FAW.

Matteo Zavattari

Matteo Zavattari works for Zinformatica. He works as a software consultant in several Italian companies. Since 2007 he has been dedicated to computer security, with particular attention to the problems of code vulnerability, computer forensics expert solution and code scanning. Since 2011 he has worked as a CTP for cybercrimes mainly in the court of Rome, Milan and Varese.