May 19th, 2:30 PM

Investigating Forensics Values of Windows Jump Lists Data

Ahmad Ghafarian
University of North Georgia, Department of Computer Science and Information Systems,
ahmad.ghafarian@ung.edu

EMBRY-RIDDLE
Aeronautical University.
SCHOLARLY COMMONS

(c)ADFSL

INVESTIGATING FORENSICS VALUES OF WINDOWS JUMP LISTS DATA

Ahmad Ghafarian
University of North Georgia
Department of Computer Science and Information Systems
Dahlonega, Ga 30597
Ahmad.ghafarian@UNG.edu

ABSTRACT

Starting with Windows 7, Microsoft introduced a new feature to the Windows Operating Systems called *Jump Lists*. *Jump Lists* stores information about user activities on the host machine. These activities may include links to the recently visited web pages, applications executed, or files processed. Computer forensics investigators may find traces of misuse in *Jump Lists* auto saved files. In this research, we investigate the forensics values of *Jump List*s data. Specifically, we use several tools to view *Jump Lists* data on a virtual machine. We show that each tool reveal certain types of information about user's activity on the host machine. This paper also presents a comparative analysis of the tools' performances. In addition, we suggest different method of viewing contents of hidden folders, present another approach for deleting files from hidden folders, and propose an innovative way of gaining access to application identification numbers (AppIDs.)

Keywords: Windows 7, *Jump Lists*, operating systems, computer forensics tools, virtual machine, VM

1. INTRODUCTION

Jump Lists is a new feature of Windows 7 Operating Systems that shows the files and tasks that most recently or most frequently used by a user. They are similar to *shortcuts* in that they take user directly to the files or directories that are regularly used. They are different than the normal *shortcut* in that they are more extensible in what information they display. For example, Internet Explorer will use *Jump Lists* to display websites frequently visited; Microsoft Office products like Excel, PowerPoint and Word, on the other hand, will show most recently opened documents. From a user's standpoint, *Jump Lists* increase one's productivity by providing quick access to the files and tasks associated with the applications. From a forensics investigator's standpoint, *Jump Lists* is a good indicator of which files were recently opened or which websites were visited frequently. Limited research results have been reported in the area of forensic value of *Jump Lists* data. Barnett (2011) has reported on the forensic value of Windows *Jump Lists* data. However, in his experiment he

did not use any computer forensic tool. The author used a PC running Windows 7 with various web browsers to download pictures from a website. Then the amount and type of information that was stored by *Jump Lists* were compared manually for different web browsers. Roblyness (2012) has evaluated the data being stored by *Jump Lists* for different applications such as Notepad, MS Word, etc. He concluded that the programs that use default applications to open a related file, store less information than when the application is chosen by a user to open the same file. This researcher also did not use any tool and all the information was retrieved manually. In Windows 7, details of accessed files, such as opening a file by right-clicking the application taskbar square, are held within structured storage files which themselves are stored within the user's profile. The files are named with 16 hexadecimal digits, known as the AppID, followed by two hidden file extensions called *automaticDestinations* and *customDestinations*. The first set store information about data file usage. Items are sorted either by Most Recently Used (MRU) or by Most

Frequently Used (MFU), depending on the application. The latter set is the type file. The content contained within, and the tasks specified by this category of file, are maintained by the specific application responsible for that specific *Destination* file. These two sets of files can be parsed to obtain forensics data. Cowen (2011) has tested several applications on Windows 7 Professional SP1 and noted that the application identification numbers (AppID) of those applications are different for different versions of the same applications.

The purpose of this research is to further investigate various aspects of *Jump Lists* auto saved data. To do this, we needed to decide on the applications we intend to use. There are many applications that a user or suspect can execute on a machine. *Jump Lists* keep different type of information for each type of application and for each action (e.g. open, update, delete, etc.) on the file. For example, the type and amount of the *Jump Lists* hidden files for a Microsoft Word file would be different than the same data for graphic file. In this work, we will limit our experiment to Microsoft Office 2010, standard web browsers, and portable web browsers. In contrast to most of the previous work, we perform our experiment on a *virtual machine (*VM*)*, i.e. vmWare. This is because we wanted to make sure that the applications we use in this experiment are the only ones that are installed on the VM. The impact of this restriction is that we will only have limited AppIDs to evaluate.

Since the tools behave differently for different applications, we present a comparative analysis of the performances of the tools we used to view *Jump Lists* data. Additional contribution of this research include proposing different methods of viewing contents of hidden folders, presenting another approach for deleting files from hidden folders and suggesting an innovative way of gaining access to AppIDs.

2. VIRTUAL MACHINE AND TOOLS

In order to make our experiment consistent, we use virtual machine to examine forensics values of Windows *Jump Lists*. This is because the experiment is done at different date and time during the course of this research. At any given time, a physical machine may have different status as far as the application running on the machine and resource usages of the system. However, with the virtual machine, we use a bare machine with only the activities that are related to this research.

Throughout this research, we use some tools to retrieve information from Windows *Jump Lists*. In general, there are two types of tools, namely tools like Jumplist-Launcher, which allow user to create customized jump lists, and tools like JumpLister, which parses the jump lists and deliver details about the activities of the user on the Windows machine.

2.1 Virtual Machine (VM)

A virtual machine is a software implementation of a computing environment. The virtual machine typically emulates a physical computer, but requests for physical resources are managed by a hypervisor which translates these requests to the underlying physical hardware (vmWare, 2014).

2.2 Jumplist-Launcher

Jumplist-Launcher is a free portable tool for Windows 7 and 8 that allows computer forensics investigators to add their favorite programs in a single *Jump Lists* for easy accessibility. We can add up to 60 jump list-items and they can be categorized into self-defined groups for easy accessibility (Madalina, 2014).

2.3 JumpListsView

JumpListsView is an open source tool that is used to display the information stored by *Jump Lists*. For every record found in the *Jump Lists*, JumpListsView displays the following information: The filename that the user accessed, the date/time of the file opening event, the ID of the application that was used to open the file, the size/time/attributes of the file on the time the file was opened (NirSoft, 2013.)

2.4 JumpLister

JumpLister is designed to open one or more *Jump Lists* files, parse the compound file structure, and then parse the link file streams that are contained within. It uses the LNK parser (Woanware, 2012).The latest version also parses out the

Destination Lists (DestList) and performs a lookup on the AppIDs (Cowen, 2011.) For example, when a user opens a file and saves it as a new name, in JumpLister the Count will increase by one in Root and the DestList will be updated. Besides, the path of file, type of file, and name of file will be shown to the examiner.

2.5 Jump Lister Parser (JMP)

JMP is a command line version of a Windows parser that parses *Jump Lists*. This tool is geared for outputting data in a parseable comma delimited (CSV) format. For example, the statement, Jmp <Destinations filename> > results.txt,
parses an individual destination file and saves the results on results.txt.

2.6 Jump List File Extract

Jump Lists File Extract is a program that extracts file information from *Jump List*s data. This information contains link to the files accessed by *Jump Lists* that are called destination files and are introduced in section 1 of this paper.

3. OUR EXPERIMENT

In this section we describe the environment and the setup in which we performed our experiment. We installed vmWare 8.0 on a Windows 7 machine. We then set the logical environment for JumpLister, JumpListsView and Jmp on VM. In order to view *Jump Lists* data we need to run an application on our Windows machine. To make data more meaningful, we limited ourselves to three applications namely, Microsoft Office 2010, Mozilla Firefox, and Google Chrome Portable. We installed all these three applications on vmWware (VM).

3.1 Results of Actions on Various Application Files

First we worked with MS Word. We created a sample MS Word document on VM. We then performed some actions such as open, rename, delete on the file. After each action, we used several tools to open *Jump Lists* auto saved data. Three of the most significant *Jump Lists* data that we monitored their changes include AppID, Count, and DestList. The results of this experiment are shown in Table 1 below. AppID

was briefly described in section 1 above. Count and DestList are briefly described below.

Count indicates the number of times a file has been referred to and DestList represents the action on the file. The DestList stream acts as a most recently/frequently used list. This stream consists of a 32-byte header, followed by the various structures that correspond to each of the individual numbered streams. Each of these structures is 114 bytes in size, followed by a variable length Unicode string (NirSoft, 2013.)

For installed web browser experiment, after installing Firefox web browser and connecting to the Internet for the first time, we noticed that the AppIDs was not created even after viewing the Welcome Firefox HTML. Further examination showed that the AppID was created when Count increased for the first time. Each increase of the Count indicates some actions such as visiting a web site, downloading a picture or a video clip. Table 1 shows the changes on Count, AppID and DestList for opening a web page.

For portable web browser, we used Google Chrome portable web browser. Generally, we cannot pin a portable app to the taskbar since the AppID of the launcher is different from the actual app executable. Therefore, the windows taskbar cannot group them into one place. However, we followed the solution that is offered by (Roblyness, 2012) and were able to create shortcuts and pin it on the Taskbar. After we opened a web page, we checked *Jump Lists* data and noticed that the AppIDs was not created. The reason is because there were both installed and portable web browsers and the operating systems probably did not know which one to use. After we uninstalled Firefox browser, tried to open a page with the portable browser, the related web page was opened. We tried this action several times to make sure that this observation is accurate. In Windows XP, we can set a portable web browser as a default browser. However, in Windows 7 and 8 this cannot be done easily. See Table 1 for the results.

Table 1-Results of Actions on MS Word, Installed Browser, and Portable Browsers

Row	Action	Result	AppID	Count	DestList
1	Open fixed disk Word file	After opening the file	Visible after Count changed	Changed	Updated
2	Open Word file from Removable media	After opening the file	Not visible	Not changed	Not updated
3	Right mouse click & delete Word file	After deleting the file	Visible after Count changed	Changed	Updated
4	Rename, Word file	After action finished	Visible after Count changed	Changed	Updated
5	Regular browser,	After opening a page	Visible after Count changed	Changed	Updated.
6	Portable browser	After opening a page	Not visible	Not changed	Not updated

Comparison of Table 1 entries with the results reported in (Larson, 2011) shows that; *Jump Lists* data revealed on our VM and on a Physical machine for the most parts are the same. The exception is when we use removable media and portable browser. In case of removable media such as flash drive, Count and DestList were not changed. In the case of portable web browser, we should not have an installed version of any browser together with the portable web browser on the same machine. Otherwise, portable web browser would not work. In addition, with portable web browser, when the saved web page was opened and changed followed by saving these changes, Date/Time was not updated in *Jump Lists*. Also, when we used portable web browsers to open a page, the AppID was not visible, Count was not changed, and DestList was not updated either. For regular web browsers, after opening a web page and saving it as a new web page, the Count and DestList were updated. Overall, examining traces of using removable media and web browsing activity using portable web browser is a challenging task for computer forensics investigators. However, for non-removable media, the details of a user activity can be viewed and possible misuse can be identified.

3.2 Comparisons of the Tools

In section2, we introduced several Windows *Jump Lists* tools. Table 1 shows comparative performances of some of those tools on Windows *Jump Lists*. However, we plan to report more results in future paper covering all the tools.

Table 2- Comparison of the Tools

Tool Name	User friendliness	Displays information	Search Option	Recognizes DestList	Recognizes AppIDs
Jmp	CMD	Yes	Has	Does	Doesn't have
JumpLister	GUI	Yes	Doesn't have	Doesn't have	To certain extent
JumpListView	GUI & CMD	Yes	Has	Doesn't have	Has

With JMP we can extract more data than the other two. JMP and JumpListView have search option but Jumplister does not have. Overall, there is no one tool that does everything. Rather each tool has unique feature. We recommend that one should use a combination of the tools.

4. ADDITIONAL RESEARCH

During the course of this research, we propose different ways of handling issues such as deleting files, accessing applications, etc. In this section, we present the details of our approaches to handling those issues.

4.1 Detecting Files from Hidden Folders

As we discussed earlier, *Jump Lists* creates hidden files and folders on the host machine. There are specific methodologies and tools to detect these files. Two methods of detecting files of hidden folders have been discussed by Madalina (2014). In this work, we propose a third method of copying hidden files to a new destination. We can do that by typing the following command in MS-DOS, the hidden files will be copied to a new media called d.

C:\copy   c:\ %appdata%\Microsoft\Windows\Recent\Automati cDestinations*.* d:\new folder

4.2 Deleting Jump List Data

A suspect may decide to delete file entries from the *Jump List*s so that a trace of it cannot be found. Various methods of deleting the entries from a *Jump Lists* have been tested by Harvey (2011). In here we propose another way of deleting file entries. In this approach, we suggest to use Track Eraser Pro software for free (AceSoft, 2014) to delete the *AutomaticDestanation* folder and its content. Therefore, an investigator should consider that a suspect may have used this utility for erasing his/her foot print.

4.3 Finding AppIDs

Jump Lists file names are created using hash-like values that in turn are based on AppID. A forensics investigator may be interested in determining AppIDs which in turn identifies associated applications that have been used by a suspect. Two methods of finding AppIDs are listed in (Forensics Focus, 2012). We propose the third way of finding AppIDs. In this approach, we delete *AutomaticDestination* files (for example with Track Eraser). Recall that when we delete *AutomaticDestination,* the hidden files will still be there. We then use specific tools to retrieve AppIDs of the deleted files. From AppIDs we can determine the applications that have been used by a suspect. The AppIDs contain 16 characters. Table-3 shows AppIDs of several applications (List of Jump Lists IDs).

Table 3 – Selected AppIDs and Corresponding Applications

AppID	Application Description
271e609288e1210a	Microsoft Office Access 2010 x86
6e855c85de07bc6a	Microsoft Office Excel 2010 x86
3094cdb43bf5e9c2	Microsoft Office OneNote 2010 x86
9c7cc110ff56d1bd	Microsoft Office PowerPoint 2010 x86
a7bd71699cd38d1c	Microsoft Office Word 2010 x86
3094cdb43bf5e9c2	Microsoft Office One note 2010 x86
28c8b86deab549a1	Internet Explorer 8 / 9
6824f4a902c78fbd	Mozilla Firefox 29

5. CONCLUSSIONS

Our results show that *Jump Lists* data in both cases of physical and virtual machine are the same in most cases. However, when we use removable drive, traces of *Jump Lists* data are inconsistent. Similarly, when we have a standard web browser installed on the VM, we could not launch the portable web browser on the VM. Over all we conclude that forensics analysis of *Jump Lists* data for removable media and portable web browsers is more challenging for computer forensics investigators. Comparisons of the performances of the tools show that each tool has its own unique feature. We found that the type and the amount of data varied based on the tool we use. This is because the tools are designed with different features. Our suggestion is for analysis of *Jump Lists* data; a combination of the tools will yield better results. Finally, we made recommendations on how to detect *Jump Lists* hidden files, how to find AppIDs and how to delete *Jump Lists* data.

6. FUTURE WORK

We plan to experiment with more tools on a physical machine for parsing Windows Jump Lists data to extract forensically valuable data. This may include the type and the amount of data they can retrieve from *Jump Lists* information. We also plan on evaluating *Jump Lists* data on different applications such PDF, images, and multimedia files. Writing new parsing tools with different features will also be a good line of future research. Additional work can be done on the verification of the consistency of tools. This can be done by performing the same action more than once on the same application file to see if the same results can yield.

REFRENCES

Acesoft (2014). Track Eraser software. Retrieved from http://www.acesoft.net/download.htm

Barnett, A. (2011). The Forensics Value of the Windows 7 Jumplist, Purdue University. Retrieved from

http://www.alexbarnett.com/jumplistforensics.pdf

Cowen, D. (2011), Jump Lists Forensics: AppIDs Part 1 & 2, retrieved from http://www.4n6k.com/2011/09/jump-list-forensics-appids-part-1.html

Forensics Focus (2011). Windows Jumplist parser (Jmp). Retrieved from http://www.forensicfocus.com/Forums/viewtopic/t=9316/

FtpArmy (2011). Jumplist File Extract. Retrieved from http://ftparmy.com/143017-jumplist-file-extract.html

Harvey, H. (2011), Windows Incident Response, retrieved from http://windowsir.blogspot.com/2011/08/jump-list-analysis.html

Larson, T. (2011). Forensics Analysis of Windows 7 Jump Lists, retrieved from http://www.slideshare.net/ctin/windows-7-forensics-jump-listsrv3public#

Madalina, M. (2014). "How to create your own Windows 7 and 8.1 Jump Lists using Jumplists Launcher. Rretrieved 4/from http://en.www.ali.dj/jumplist-launcher/

NirSoft (2013). JumpListsView. Retrieved from http://www.nirsoft.net/utils/jump_lists_view.html

Roblyness, T. (2012), Forensics Analysis of Windows 7 Jump Lists. retrieved from http://articles.forensicfocus.com/2012/10/30/forensic-analysis-of-windows-7-jump-lists/

vmWare Virtualization for desktop. Retrieved from http://www.vmware.com/

Wiki. List of Jumplist IDs, retrieved from
      http://www.forensicswiki.org/wiki/
      List_of_Jump_List_IDs

WoanWare (2012). Jumplister info, retrieved from
      http://www.woanware.co.uk/forensics
      /jumplister.htm l