Chapter 4

RECOVERY OF FORENSIC ARTIFACTS FROM DELETED JUMP LISTS

Bhupendra Singh, Upasna Singh, Pankaj Sharma and Rajender Nath

Abstract Jump lists, which were introduced in the Windows 7 desktop operating system, have attracted the interest of researchers and practitioners in the digital forensics community. The structure and forensic implications of jump lists have been explored widely. However, little attention has focused on anti-forensic activities such as jump list evidence modification and deletion. This chapter proposes a new methodology for identifying deleted entries in the Windows 10 AutoDest type of jump list files and recovering the deleted entries. The proposed methodology is best suited to scenarios where users intentionally delete jump list entries to hide evidence related to their activities. The chapter also examines how jump lists are impacted when software applications are installed and when the associated files are accessed by external storage devices. In particular, artifacts related to file access, such as the lists of most recently used and most frequently used files, file modification, access and creation timestamps, names of applications used to access files, file paths, volume names and serial numbers from where the files were accessed, can be recovered even after entries are removed from the jump lists and the software applications are uninstalled. The results demonstrate that the analysis of jump lists is immensely helpful in constructing the timelines of user activities on Windows 10 systems.

Keywords: Windows forensics, Windows 10, deleted jump lists, recovery

1. Introduction

Microsoft launched the Windows 10 operating system on July 29, 2015. As of November 2017, Windows 10 was the second most popular desktop operating system with a market share 31.85%, after Windows 7 with a market share 43.12% [8]. Forensic investigators are encountering large numbers of Windows 10 workstations for evidence recovery and analysis. The initial version of Windows 10 (v1511) was shipped with many new features, including Cortana, Edge Browser, Action (or Notification) Center, Universal App Platform, OneDrive, Continuum, Windows Hello and Quick Access. These features have direct implications on digital forensic investigations [11]. Windows 10 File Explorer opens up the Quick Access view by default to ease access to frequently used folders and recent files. These files and folders are stored in the *C:\Users\UserName\AppData\Roaming\Microsoft\Windows\Recent* directory in the form of Windows shortcut (LNK) files. The LNK file format has not changed in the Windows 10 operating system, but Microsoft has modified the structure of Windows 10 jump lists, especially the DestList stream [10]. Also, the number of items to be displayed in a list is now hard coded.

Microsoft introduced the jump list feature in the Windows 7 desktop operating system to improve user experience by providing the lists of recently opened files and directories. Before the introduction of jump lists, forensic investigators were dependent on the Windows registry to identify the most recently used (MRU) and most frequently used (MFU) items. Compared with the Windows registry, jump list data files provide more valuable artifacts related to user activity history. For instance, it is possible to extract useful information about file accesses, including MRU and MFU lists for users and applications, file names, full file paths, modified, accessed and created (MAC) timestamp values, volume names and volume serial numbers from where files were accessed, unique file volumes and object IDs. These artifacts appear to persist even after the files and the software applications that accessed them have been removed. Jump list information is maintained on a per application basis. However, not all applications create jump lists; these include host-based applications such as Regedit, Command Prompt and Run [12].

This chapter presents a methodology for recovering deleted jump list entries in Windows 10 systems. The open-source JumpListExt tool was used to parse and view information in jump list data files. Several experiments were conducted to detect and recover deleted jump lists and carve their artifacts, such as the applications used to access files, MRU and MFU lists, volume names and serial numbers used for access and files accessed during specific boot sessions. These artifacts appear to persist even after files have been deleted and their target software applications have been uninstalled.

2. Jump Lists in Digital Forensics

The structure and applications of jump list files have been widely discussed in the forensics community since the release of Windows 7.

Barnett [1] has described how jump lists function and has investigated the behavior of the jump list of a browser when files are uploaded and downloaded using the browser. Lyness [5] has explored jump lists further and has identified the structure and types of information recorded in a DestList stream contained in an AutoDest data file in a Windows 7 desktop operating system. Lyness executed anti-forensic actions on data files such as removing entries from a jump list and discovered that the attempts can be detected in the DestList stream data. Lallie and Bains [3] have presented an overview of the AutoDest data file structure and have documented numerous artifacts related to file accesses and program execution; they suggest that research should focus on timeline development based on the information extracted from jump list files.

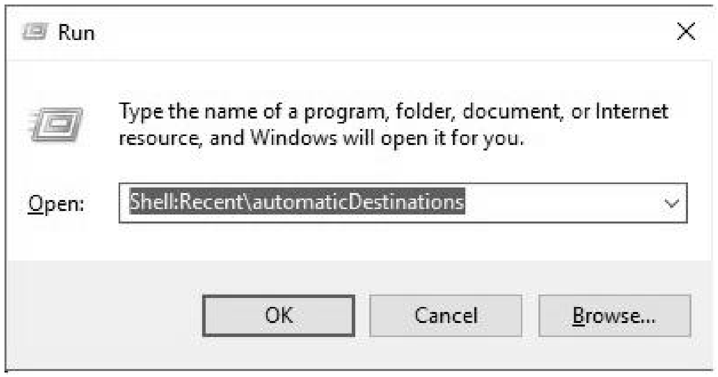
Smith [13] has used jump lists to detect fraudulent documents created on a Windows system. In particular, Smith showed that the information maintained in jump list files is useful forensic evidence in financial fraud cases because the files record all file creation and opening activities. More recently, Singh and Singh [10] have conducted the first investigation of jump lists in the Windows 10 operating system and have discovered that modifications of and/or additions to certain portions of jump list data files prevent existing forensic tools from working properly. Singh and Singh also examined the new DestList structure and compared it against the DestList structures in older Windows versions. They developed the JumpListExt tool for extracting information stored in jump list data files, individually as well as collectively. This information is very useful for constructing user activity timelines.

3. Locations and Structures of Jump Lists

The data files created by the jump list feature are hidden by default, but users can view them by browsing the complete path. Two types of data files are associated with the feature: (i) AutoDest (automatic destinations); and (ii) CustDest (custom destinations). Users may also locate the data files by entering *Shell:Recent\automaticDestinations* in the Run command as shown in Figure 1. The files are located at the following paths:

- **AutoDest:** *C:\Users\UserName\AppData\Roaming\Microsoft \Windows\Recent\AutomaticDestinations*
- **CustDest:** *C:\Users\UserName\appData\Roaming\Microsoft \Windows\Recent\CustomDestinations*

Larson [4] was the first to study jump lists as a resource for user activity analysis and presented the anatomy of jump lists in Windows 7


Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:
OK
Cancel
Browse...

```
Shell:Recent\automaticDestinations
```

Figure 1. Locating jump list data files in Windows.

platforms. Each jump list file has a 16-digit hexadecimal number called the AppID (application identifier) followed by an extension. The AppID is computed by the Windows operating system using a CRC-64 checksum based on the file path of the application. An application that is executed from two different locations has two different AppIDs associated with the application. A script is available for computing AppIDs based on the file paths [2]. A list of known AppIDs is available at [6, 9].

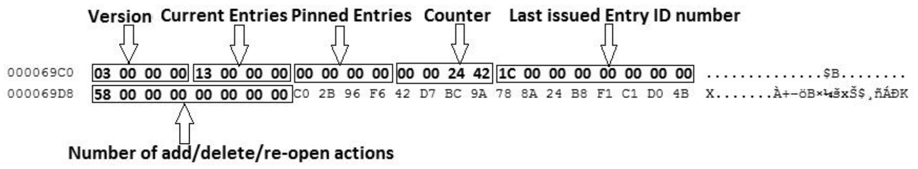
The structure of an AutoDest data file conforms to the Microsoft OLE (object linking and embedding) file format [7]. The AutoDest file has two types of streams: (i) SHLLINK; and (ii) DestList. A data file may have multiple SHLLINK streams, all of which have a structure similar to a shortcut (LNK) file. On the other hand, the DestList stream, which has a special structure, records the order of file accesses and access count of each file, which could serve as the MRU and MFU lists, respectively. The header length of 32 bytes in a DestList stream is fixed in all Windows distributions. However, the semantics of the fields in the DestList stream header differ in Windows distributions.

Table 1 presents the DestList stream header fields and their semantics in four Windows operating system distributions. The DestList stream header records useful information such as the version number, total current entries, last issue entry ID number, pinned entries and the numbers of added/deleted entries in the jump list.

Figure 2 shows the binary data in a DestList stream header in Windows 10 Pro v1511. Comparison of the DestList header in Windows 10 against the header in Windows 7/8 reveals that most of the fields are same, but the semantics of two fields are different. For example, the value of the first four bytes in Windows 10 is 3 or 4 (version number) whereas it is 1 (first issued entry ID number) in Windows 7/8. Also,

Table 1. DestList headers and entry lengths in four Windows distributions.

Operating System	**Header Length (Bytes)**	**Header Version (Offset 0)**	**Entry Length (Bytes)**
Windows 7 Profesional	32	`0x00000001` (1)	114
Windows 8/8.1 Pro x64	32	`0x00000001` (1)	114
Windows 10 Pro v1511	32	`0x00000003` (3)	130
Windows 10 Pro v1607	32	`0x00000004` (4)	130


Version
Current Entries
Pinned Entries
Counter
Last issued Entry ID number
Number of add/delete/re-open actions

```
000069C0  03 00 00 00 13 00 00 00 00 00 00 00 00 00 24 42 1C 00 00 00 00 00 00 00  ..............$B........
000069D8  58 00 00 00 00 00 00 00 C0 2B 96 F6 42 D7 BC 9A 78 8A 24 B8 F1 C1 D0 4B  X.......À+–öB×¼šxŠ$¸ñÁÐK
```

Figure 2. DestList stream header structure in Windows 10 Pro v1511.

the value of the last eight bytes of the DestList header appears to be double that of the total current entries in the list. When an entry is added, deleted or pinned, or an existing entry is re-opened, the value is incremented by one. Singh and Singh [10] have described the process for computing the access counts of removed entries. Table 1 presents the DestList headers for various Windows distributions.

CustDest files are created by applications with their AppIDs followed by the extension `customDestinations-ms`. These jump lists are specified by the applications using the ICustomDestinationList API [4]. Compared with AutoDest files, CustDest files have a different structure of sequential MS-SHLLINK binary format segments. CustDest files record the artifacts related to user web history on the system. For example, the file `969252ce11249fdd.customDestinations-ms`, which is created by Mozilla Firefox, records the web history and timestamps. Windows Media Player creates the file `74d7f43c1561fc1e.customDestinations-ms`, which records the file paths of music files that have been played.

4. Experiments and Results

This section discusses the experimental objectives, methodology and results related to the recovery of forensic artifacts from AutoDest jump lists in a Windows 10 Pro v1511 system.

4.1 Experimental Objectives

The principal objectives of the experiments were to: (i) retrieve deleted entries from jump list data files; (ii) identify the names of applications based on their AppIDs; (iii) determine the connected removable media properties using jump list information; (iv) list the most recently used and most frequently used files on a per application basis; and (v) identify the files accessed during a particular boot session.

4.2 Experimental Methodology

Ever since the introduction of jump lists in Windows 7, the recovery of forensic artifacts from deleted jump lists has always been a challenge. This section describes the methodology for identifying deleted entries in AutoDest jump list files and recovering the deleted entries. The methodology is best suited to scenarios where users have deleted entries from jump lists.

The first step in the methodology is to locate the AutoDest data files by entering *Shell:Recent\automaticDestinations* in the Run command. If the data files are available at the specified location, then it is possible to obtain the AppIDs and, thus, the names of the applications. The data files are then parsed individually or as a whole using the JumpListExt tool. The procedure outlined in Figure 3 is used to detect entries removed from an AutoDest data file. The jump list file must be parsed manually in order to carve the deleted entries. Manual analysis of the DestList header provides information about the number of deleted entries and their access counts. In the experiments, FTK Imager 3.4.2.6 was used to acquire the jump list files.

4.3 Experimental Set-Up

Two Windows systems were set-up for the experiments, one as the suspect system and the other as the forensic server. Several applications were installed on the suspect system, including host-based user applications (default Windows installation), user applications, portable applications (without installation and writing any configuration settings to the disk) and Windows Store apps, were used to create the jump lists. Various hypothetical activity scenarios were simulated by opening test

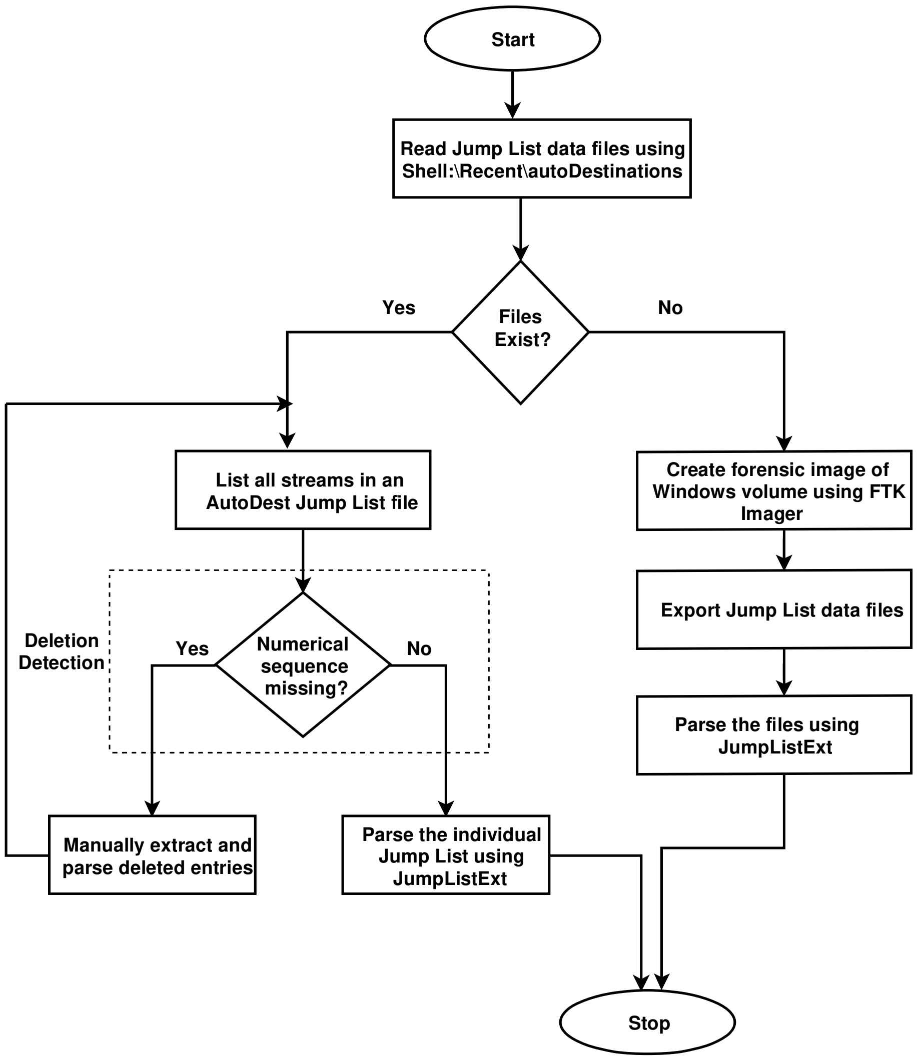

Start
Read Jump List data files using Shell:\Recent\autoDestinations
Files Exist?
Yes
No
List all streams in an AutoDest Jump List file
Create forensic image of Windows volume using FTK Imager
Export Jump List data files
Deletion Detection
Yes
Numerical sequence missing?
No
Parse the files using JumpListExt
Manually extract and parse deleted entries
Parse the individual Jump List using JumpListExt
Stop

Figure 3. Identifying and recovering deleted entries in AutoDest jump lists.

files with the applications; the test files were opened from the internal hard disk. Also, in some cases, the applications and files were installed and opened from removable storage devices. The CCleaner application was used to delete jump lists and recent items. The forensic server was loaded with FTK Imager, which was used to recover and examine the deleted jump lists. The JumpListExt tool was used to parse and view

Table 2. Systems and tools used in the experiments.

Item	Description
Suspect System	Windows 10 Pro v1709 i3-2328M CPU @ 2.2 GHZ, 4 GB RAM
Forensic Server	Windows 7 Professional i7-3770S CPU @ 3.1 GHZ, 4 GB RAM
FTK Imager v3.4.2.6	For acquiring the jump list files
CCleaner v5.05	For deleting shortcut (LNK) and jump list files
JumpListExt	For viewing and parsing jump list files

the jump list files. Table 2 shows the details of the systems and tools used in the experiments.

4.4 Artifacts in AutoDest Jump Lists

This section discusses the experiments and their results. Note that the applications and file types considered in the experiments are mere examples; numerous other applications and file types could be considered without affecting the experimental results.

Detecting and Recovering Deleted Jump Lists. Users can delete jump list files from hidden locations to destroy evidence. The evidence may be removed in two ways:

- Removing individual entries from a jump list file using "Remove from this list" on the Taskbar or Startbar.
- Deleting all the jump list files using SHIFT+DELETE or by executing a privacy cleaner tool.

It is difficult for a normal user to detect the removal of individual entries from a jump list file. However, a forensic analyst can detect the removal by parsing the header of the DestList stream of the AutoDest jump list file. If a few entries have been removed from the list, then the values of the total number of current entries and the last issued entry ID number would be different. The difference gives the number of entries removed from the list. Note that (partial) data belonging to a removed entry may still reside in the corresponding jump list file. An automated tool is not available for carving the individual entries that have been removed. However, a forensic analyst can carve the entries via manual analysis of binary data in the jump list file.

Table 3. Effective size of an AutoDest file before and after entry removal.

Before Entry Removal			**After Entry Removal**		
	Entry	**Size**		**Entry**	**Size**
SHLLINK Streams	1	995	**SHLLINK Streams**	1	995
	2	941		2	941
	3	952		3	952
	4	997			
DestList Stream		902	**DestList Stream**		668
Effective Size		4,787	**Effective Size**		3,576

The Adobe Reader application was used to open four PDF files, which created the file `de48a32edcbe79e4.automaticDestinations-ms`. The individual SHLLINK streams and the DestList stream were extracted and their sizes (in bytes) were recorded (Table 3). The sizes of the two types of streams were determined using the following Python script:

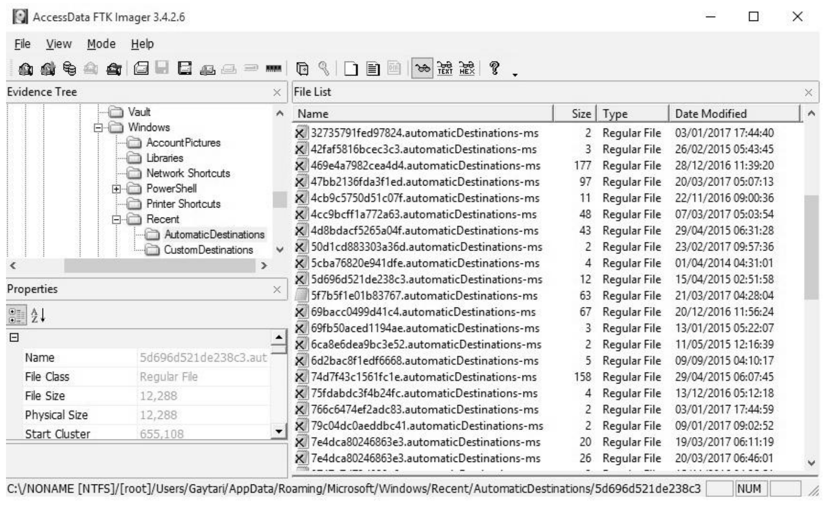
```
import olefile as J
ole = J.OleFileIO("AppID.automaticDestinations-ms", "rb")

# lists all streams in the AutoDest file
print(ole.listdir(streams = True, storages = False))

# lists size of all streams in the AutoDest file
for item in ole.listdir():
       print(ole.get_size(item))
```

The total size of the two types of streams, SHLLINK and DestList, was (995 + 941 + 952 + 997) + 902 = 4,787 bytes. Next, entry number 4 was removed from the jump list associated with Adobe Reader by selecting the option "Remove from this list" on the Startbar. The two types of streams were extracted once again and their total size was computed to be 3,576 bytes. If all the data related to entry number 4 (997 bytes) had been removed, then the number of remaining data bytes can be computed as follows:

$$Remaining\ Bytes = E_B - (E_A + Entry\ Size) \qquad (1)$$

where E_B and E_A are the effective sizes before and after removal, respectively.

Upon applying Equation (1), a total of 4,787 – (3,576 + 997) = 214 bytes of the removed entry persisted in the AutoDest jump list file of



Name	5d696d521de238c3.aut
File Class	Regular File
File Size	12,288
Physical Size	12,288
Start Cluster	655,108

Name	Size	Type	Date Modified
32735791fed97824.automaticDestinations-ms	2	Regular File	03/01/2017 17:44:40
42faf5816bcec3c3.automaticDestinations-ms	3	Regular File	26/02/2015 05:43:45
469e4a7982cea4d4.automaticDestinations-ms	177	Regular File	28/12/2016 11:39:20
47bb2136fda3f1ed.automaticDestinations-ms	97	Regular File	20/03/2017 05:07:13
4cb9c5750d51c07f.automaticDestinations-ms	11	Regular File	22/11/2016 09:00:36
4cc9bcff1a772a63.automaticDestinations-ms	48	Regular File	07/03/2017 05:03:54
4d8bdacf5265a04f.automaticDestinations-ms	43	Regular File	29/04/2015 06:31:28
50d1cd883303a36d.automaticDestinations-ms	2	Regular File	23/02/2017 09:57:36
5cba76820e941dfe.automaticDestinations-ms	4	Regular File	01/04/2014 04:31:01
5d696d521de238c3.automaticDestinations-ms	12	Regular File	15/04/2015 02:51:58
5f7b5f1e01b83767.automaticDestinations-ms	63	Regular File	21/03/2017 04:28:04
69bacc0499d41c4.automaticDestinations-ms	67	Regular File	20/12/2016 11:56:24
69fb50aced1194ae.automaticDestinations-ms	3	Regular File	13/01/2015 05:22:07
6ca8e6dea9bc3e52.automaticDestinations-ms	2	Regular File	11/05/2015 12:16:39
6d2bac8f1edf6668.automaticDestinations-ms	5	Regular File	09/09/2015 04:10:17
74d7f43c1561fc1e.automaticDestinations-ms	158	Regular File	29/04/2015 06:07:45
75fdabdc3f4b24fc.automaticDestinations-ms	4	Regular File	13/12/2016 05:12:18
766c6474ef2adc83.automaticDestinations-ms	2	Regular File	03/01/2017 17:44:59
79c04dc0aeddbc41.automaticDestinations-ms	2	Regular File	09/01/2017 09:02:52
7e4dca80246863e3.automaticDestinations-ms	20	Regular File	19/03/2017 06:11:19
7e4dca80246863e3.automaticDestinations-ms	26	Regular File	20/03/2017 06:46:01

C:\/NONAME [NTFS]/[root]/Users/Gaytari/AppData/Roaming/Microsoft/Windows/Recent/AutomaticDestinations/5d696d521de238c3

Figure 4. Deleted jump lists shown in FTK Imager.

the application. However, these bytes could only be carved by analyzing the binary data of the jump list file.

The following observations were made when conducting the experiments:

- Removing individual entries reduces the DestList stream size by 130 bytes plus the size (in bytes) of the file name in Unicode.
- Partial data pertaining to the deleted entries may reside in the SHLLINK streams.
- The overall size of the jump list may or may not be reduced.
- The size of the residual data may be computed using Equation (1).

Another set of experiments was performed to reproduce a situation where a user deliberately deletes a few or all the jump list files to hide activities performed on fixed or removable media. A user may also destroy evidence in the files by browsing the locations and manually deleting them using SHIFT+DELETE. In the experiments, jump lists and recent documents were deleted by running a privacy protection tool (CCleaner).

FTK Imager was used to create an image of the Windows volume (comprising the Windows 10 operating system) to recover the deleted jump list files (Figure 4). The deleted jump list data files were then exported to a different volume. Following this, the exported files were parsed and analyzed using the JumpListExt tool. Useful information

Table 4. AppIDs and application names.

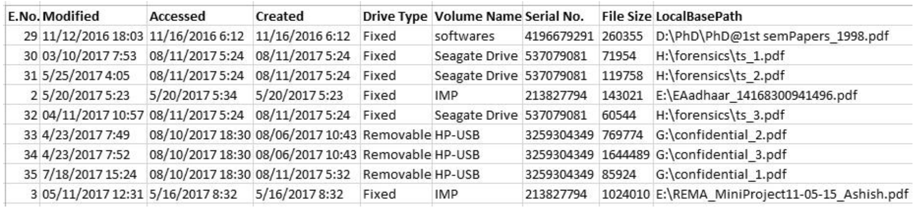
AppID	Application Name
1bc392b8e104a00e	Remote Desktop
*5f7b5f1e01b83767	Quick Access
*4cb9c5750d51c07f	Movies and TV (Windows Store App)
4cc9bcff1a772a63	Microsoft Office PowerPoint 2013 x64
9b9cdc69c1c24e2b	Notepad (64-bit)
9ce6555426f54b46	HxD Hex Editor
12dc1ea8e34b5a6	Microsoft Paint
47bb2136fda3f1ed	Microsoft Office Word 2013 x64
69bacc0499d41c4	Microsoft Office Excel 2013 x64
*a52b0784bd667468	Photos (Windows Store App)
*ae6df75df512bd06	Groove Music (Windows Store App)
f01b4d95cf55d32a	Windows Explorer Windows 8.1/10
faef7def55a1d4b	VLC Media Player 2.1.5 x64
ff103e2cc310d0d	Adobe Reader 11.0.0 x64

related to file accesses was obtained. This included the most MRU and MFU lists corresponding to each user and application, file names, full file paths, file MAC timestamps, volume names and serial numbers from where the files were accessed, unique file volumes and object IDs.

Identifying the Names of Installed Applications. AppIDs of individual applications are computed by the Windows operating system based on the application file paths. AppIDs can be used to name the individual applications. Thus, if an AppID is known, it is possible to identify the name of the associated application. Table 4 lists the AppIDs of common applications. The AppIDs marked with asterisks correspond to default applications introduced in Windows 10.

During the experiments, it was observed that the AppID of an application gives the correct name only when the application is installed at its default location. Different AppIDs are produced when an application is installed at its default location and subsequently installed at another location. Also, the data files associated with applications remain on the hard disk even after the applications have been uninstalled.

Determining Connected Removable Media Properties. When applications are installed and files are opened from a removable media drive, drive properties such as drive type, removable media label, drive serial number and full paths to the accessed files can be determined from



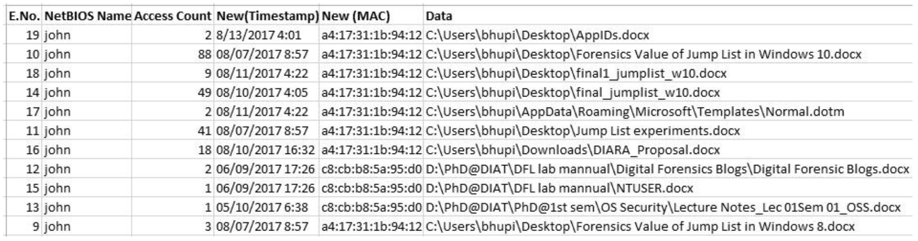
E.No.	Modified	Accessed	Created	Drive Type	Volume Name	Serial No.	File Size	LocalBasePath
29	11/12/2016 18:03	11/16/2016 6:12	11/16/2016 6:12	Fixed	softwares	4196679291	260355	D:\PhD\PhD@1st semPapers_1998.pdf
30	03/10/2017 7:53	08/11/2017 5:24	08/11/2017 5:24	Fixed	Seagate Drive	537079081	71954	H:\forensics\ts_1.pdf
31	5/25/2017 4:05	08/11/2017 5:24	08/11/2017 5:24	Fixed	Seagate Drive	537079081	119758	H:\forensics\ts_2.pdf
2	5/20/2017 5:23	5/20/2017 5:34	5/20/2017 5:23	Fixed	IMP	213827794	143021	E:\EAadhaar_14168300941496.pdf
32	04/11/2017 10:57	08/11/2017 5:24	08/11/2017 5:24	Fixed	Seagate Drive	537079081	60544	H:\forensics\ts_3.pdf
33	4/23/2017 7:49	08/10/2017 18:30	08/06/2017 10:43	Removable	HP-USB	3259304349	769774	G:\confidential_2.pdf
34	4/23/2017 7:52	08/10/2017 18:30	08/06/2017 10:43	Removable	HP-USB	3259304349	1644489	G:\confidential_3.pdf
35	7/18/2017 15:24	08/10/2017 18:30	08/11/2017 5:32	Removable	HP-USB	3259304349	85924	G:\confidential_1.pdf
3	05/11/2017 12:31	5/16/2017 8:32	5/16/2017 8:32	Fixed	IMP	213827794	1024010	E:\REMA_MiniProject11-05-15_Ashish.pdf

Figure 5. Determining connected removable media properties.

the jump lists. These properties can be extracted from the individual SHLLINK streams in the AutoDest files.

To validate these hypotheses, experiments were conducted with two removable drives, an external hard drive labeled Seagate and a removable USB drive labeled HP-USB. Both the drives were seeded with applications and files. Adobe Reader XI application was installed from the HP-USB drive. Six test PDF files were opened with the application. Three files (`confidential_1.pdf`, `confidential_2.pdf` and `confidential_3.pdf`) were from the HP-USB drive and the other three files (`ts_1.pdf`, `ts_2.pdf` and `ts_3.pdf`) were from the Seagate drive. Adobe Reader XI was then uninstalled, all the test files were deleted and both the drives were removed safely.

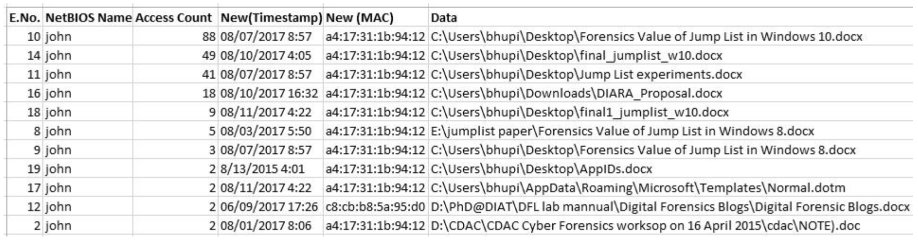
In order to determine the drive properties, the jump list for the AppID `ff103e2cc310d0d` corresponding to Adobe Reader XI was parsed and exported to a CSV file using the JumpListExt tool. Figure 5 presents the identified drive properties: (i) drive type; (ii) volume name/drive label; (iii) drive serial number; and (iv) complete paths to the test files.



E.No.	NetBIOS Name	Access Count	New(Timestamp)	New (MAC)	Data
19	john	2	8/13/2017 4:01	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\AppIDs.docx
10	john	88	08/07/2017 8:57	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\Forensics Value of Jump List in Windows 10.docx
18	john	9	08/11/2017 4:22	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\final1_jumplist_w10.docx
14	john	49	08/10/2017 4:05	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\final_jumplist_w10.docx
17	john	2	08/11/2017 4:22	a4:17:31:1b:94:12	C:\Users\bhupi\AppData\Roaming\Microsoft\Templates\Normal.dotm
11	john	41	08/07/2017 8:57	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\Jump List experiments.docx
16	john	18	08/10/2017 16:32	a4:17:31:1b:94:12	C:\Users\bhupi\Downloads\DIARA_Proposal.docx
12	john	2	06/09/2017 17:26	c8:cb:b8:5a:95:d0	D:\PhD@DIAT\DFL lab mannual\Digital Forensics Blogs\Digital Forensic Blogs.docx
15	john	1	06/09/2017 17:26	c8:cb:b8:5a:95:d0	D:\PhD@DIAT\DFL lab mannual\NTUSER.docx
13	john	1	05/10/2017 6:38	c8:cb:b8:5a:95:d0	D:\PhD@DIAT\PhD@1st sem\OS Security\Lecture Notes_Lec 01Sem 01_OSS.docx
9	john	3	08/07/2017 8:57	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\Forensics Value of Jump List in Windows 8.docx

Figure 6. MRU list for Microsoft Word 2013.

Determining MRU and MFU Lists. The DestList stream in an AutoDest file contains the entry ID numbers of the file entries. This information can be used to verify the order of the entries added to the list and, thus, the order of file accesses. Indeed, the DestList stream serves as the MRU list for the files accessed by an application. Figure 6

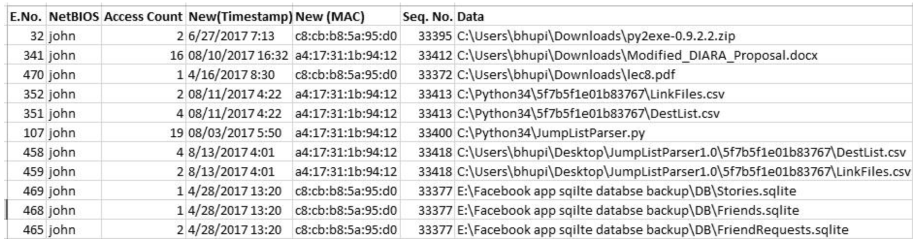


E.No.	NetBIOS Name	Access Count	New(Timestamp)	New (MAC)	Data
10	john	88	08/07/2017 8:57	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\Forensics Value of Jump List in Windows 10.docx
14	john	49	08/10/2017 4:05	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\final_jumplist_w10.docx
11	john	41	08/07/2017 8:57	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\Jump List experiments.docx
16	john	18	08/10/2017 16:32	a4:17:31:1b:94:12	C:\Users\bhupi\Downloads\DIARA_Proposal.docx
18	john	9	08/11/2017 4:22	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\final1_jumplist_w10.docx
8	john	5	08/03/2017 5:50	a4:17:31:1b:94:12	E:\jumplist paper\Forensics Value of Jump List in Windows 8.docx
9	john	3	08/07/2017 8:57	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\Forensics Value of Jump List in Windows 8.docx
19	john	2	8/13/2015 4:01	a4:17:31:1b:94:12	C:\Users\bhupi\Desktop\AppIDs.docx
17	john	2	08/11/2017 4:22	a4:17:31:1b:94:12	C:\Users\bhupi\AppData\Roaming\Microsoft\Templates\Normal.dotm
12	john	2	06/09/2017 17:26	c8:cb:b8:5a:95:d0	D:\PhD@DIAT\DFL lab mannual\Digital Forensics Blogs\Digital Forensic Blogs.docx
2	john	2	08/01/2017 8:06	a4:17:31:1b:94:12	D:\CDAC\CDAC Cyber Forensics worksop on 16 April 2015\cdac\NOTE).doc

Figure 7. MFU list for Microsoft Word 2013.

shows the MRU list for the files accessed by Microsoft Office Word 2013. The MRU list enables a forensic analyst to identify recently used files on a per application basis.

The newly-added four-byte field in the DestList entry structure from offset `116` to `119` is a counter that consistently increases as files are accessed. The field records the access counts of the individual entries. Sorting the entries based on decreasing order of access counts gives the list of MFU entries. Figure 7 shows the MFU list for the files accessed by Microsoft Office Word 2013. The MFU list enables a forensic analyst to identify the frequently used files on a per application basis.



E.No.	NetBIOS	Access Count	New(Timestamp)	New (MAC)	Seq. No.	Data
32	john	2	6/27/2017 7:13	c8:cb:b8:5a:95:d0	33395	C:\Users\bhupi\Downloads\py2exe-0.9.2.2.zip
341	john	16	08/10/2017 16:32	a4:17:31:1b:94:12	33412	C:\Users\bhupi\Downloads\Modified_DIARA_Proposal.docx
470	john	1	4/16/2017 8:30	c8:cb:b8:5a:95:d0	33372	C:\Users\bhupi\Downloads\lec8.pdf
352	john	2	08/11/2017 4:22	a4:17:31:1b:94:12	33413	C:\Python34\5f7b5f1e01b83767\LinkFiles.csv
351	john	4	08/11/2017 4:22	a4:17:31:1b:94:12	33413	C:\Python34\5f7b5f1e01b83767\DestList.csv
107	john	19	08/03/2017 5:50	a4:17:31:1b:94:12	33400	C:\Python34\JumpListParser.py
458	john	4	8/13/2017 4:01	a4:17:31:1b:94:12	33418	C:\Users\bhupi\Desktop\JumpListParser1.0\5f7b5f1e01b83767\DestList.csv
459	john	2	8/13/2017 4:01	a4:17:31:1b:94:12	33418	C:\Users\bhupi\Desktop\JumpListParser1.0\5f7b5f1e01b83767\LinkFiles.csv
469	john	1	4/28/2017 13:20	c8:cb:b8:5a:95:d0	33377	E:\Facebook app sqilte databse backup\DB\Stories.sqlite
468	john	1	4/28/2017 13:20	c8:cb:b8:5a:95:d0	33377	E:\Facebook app sqilte databse backup\DB\Friends.sqlite
465	john	2	4/28/2017 13:20	c8:cb:b8:5a:95:d0	33377	E:\Facebook app sqilte databse backup\DB\FriendRequests.sqlite

Figure 8. Files accessed in a boot session.

Identifying Files Accessed in a Boot Session. An AutoDest jump list with AppID `5f7b5f1e01b83767` (Quick Access) keeps track of all the files opened with an application. The JumpListExt tool was used to parse and export the jump list data to a CSV file. Two fields were found to be important for identifying all the files accessed during a particular boot session: (i) Sequence Number; (ii) Birth Timestamp. Entries corresponding to the files accessed during the same boot session have the same Sequence Number; the Birth Timestamp value represents the session boot time of the system (Figure 8). This information enables a forensic analyst to construct a timeline of user activities on a system under investigation.

5. Conclusions

The Microsoft Windows 10 operating system introduced dozens of new features and modified the formats of older features such as jump lists. Jump lists are created by software applications to provide quick and easy access to recently-opened files associated with the applications. Jump lists are a rich source of evidence related to file accesses and applications that have been used. This chapter has proposed a new methodology for identifying and recovering deleted entries in the AutoDest type of jump list files. The methodology is best suited to scenarios where users have intentionally deleted entries from jump lists to hide evidence related to their activities.

The experimental research demonstrates that valuable information can be recovered about the files that were accessed and their timelines, applications used to access the files, MRU and MFU file lists, volume names and serial numbers of the devices from which the files were accessed, and the files accessed during a particular boot session. The experiments also empirically verify that user activity history can be retrieved from jump lists even after the associated software applications have been uninstalled and the files have been deleted. Additionally, jump list analysis reveal anti-forensic activities intended to thwart investigations. Indeed, jump list data files are a treasure trove of evidentiary artifacts that can be leveraged in forensic investigations.

Although major portions of DestList streams are well understood, the specific uses of certain fields are still unknown. Future research will investigate the unknown components of DestList streams. Also, research will focus on automating the carving of deleted entries from jump lists.

References

[1] A. Barnett, The forensic value of the Windows 7 jump list, in *Digital Forensics and Cyber Crime*, P. Gladyshev and M. Rogers (Eds.), Springer, Berlin-Heidelberg, Germany, pp. 197–210, 2011.

[2] Hexacorn, Jump list file names and AppID calculator, Hong Kong, China (`www.hexacorn.com/blog/2013/04/30/jumplists-file-names-and-appid-calculator`), 2013.

[3] H. Lallie and P. Bains, An overview of the jump list configuration file in Windows 7, *Journal of Digital Forensics, Security and Law*, vol. 7(1), pp. 15–28, 2012.

[4] T. Larson, Forensic Examination of Windows 7 Jump Lists, *LinkedIn SlideShare* (`www.slideshare.net/ctin/windows-7-forensics-jump-listsrv3public`), June 6, 2011.

[5] R. Lyness, Forensic Analysis of Windows 7 Jump Lists, *Forensic Focus* (articles.forensicfocus.com/2012/10/30/forensic-analysis-of-windows-7-jump-lists), October 30, 2012.

[6] M. McKinnon, List of Jump List IDs, *ForensicsWiki* (www.forensicswiki.org/wiki/List_of_Jump_List_IDs), December 19, 2017.

[7] Microsoft Developer Network, [MS-CFB]: Compound File Binary File Format, Microsoft, Redmond, Washington (msdn.microsoft.com/en-us/library/dd942138.aspx), 2018.

[8] NetMarketshare, Operating system market share (www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0), 2018.

[9] D. Pullega, Jump List Forensics: AppIDs, Part 1, *4n6k* (www.4n6k.com/2011/09/jump-list-forensics-appids-part-1.html), September 7, 2011.

[10] B. Singh and U. Singh, A forensic insight into Windows 10 jump lists, *Digital Investigation*, vol. 17, pp. 1–13, 2016.

[11] B. Singh and U. Singh, A forensic insight into Windows 10 Cortana search, *Computers and Security*, vol. 66, pp. 142–154, 2017.

[12] B. Singh and U. Singh, Program execution analysis in Windows: A study of data sources, their format and comparison of forensic capability, *Computers and Security*, vol. 74, pp. 94–114, 2018.

[13] G. Smith, Using jump lists to identify fraudulent documents, *Digital Investigation*, vol. 9(3-4), pp. 193–199, 2013.