6

Windows Artifact Analysis

The world runs on the Microsoft Windows operating system, with Microsoft accounting for nearly 90 percent of the operating system market share (https://netmarketshare.com/). In my personal experience, I have examined far more Windows operating systems than any other operating system; macOS would be the next most common operating system, with Linux running a distant third. While you have to be prepared to analyze all operating systems, whichever is the most common within the realm you are working in is where you should focus your attention.

This chapter will provide you with an understanding of the Windows operating system and the artifacts you may find. There are entire books written about the Windows operating system; this chapter's goal is to provide you with an understanding of the more common operating system artifacts you may encounter during your investigation. You will start by going through user profiles where an examiner can find most user data. Then, we will look at the Windows Registry to identify the Windows settings. You will also look at artifacts to determine the user's activities and learn how to identify which USB devices were used on the system. Finally, we will cover all of this in the following topics:

- Understanding user profiles
- Understanding Windows Registry

- Determining account usage
- Determining file knowledge
- Identifying physical locations
- Exploring program execution
- Understanding USB/attached devices

An operating system manages the hardware resources and allows the user to run other applications that are essentially programs within the operating system environment. It can be a treasure trove of artifacts to recreate user or system activity at any given moment in time. When we discuss the Windows operating system, this topic could cover multiple versions. At the time of writing, the current version of the Windows operating system is Windows 11, but Windows 10 is still on the majority of systems. That does not mean every system you examine will have Windows 10 installed on it. It is possible that even in a corporate environment, you could still examine a Windows XP client, although Microsoft released it in 2001 and no longer supports it.

I will focus on Windows 7, 8, and 10 for the rest of this chapter. First, however, there may be references to Windows XP because of Microsoft's legacy support for the operating system.

The first item I want to discuss is the different types of user profiles and where the operating system will store the user's data.

Understanding user profiles

When the Windows operating system is installed, it creates a default folder structure to store user and application data. Sometimes, just looking at the folder structure can tell you which version is or isn't installed.

When you are looking for user account profiles, the location can vary depending on the version of the operating system:

- For Windows XP, WinNT, and Win2000
  - `C:\Documents and Settings\%UserName%`
- For Windows Vista, 7, 8, and 10
  - `C:\Users\%UserName%`

When the user first logs on to the system, it will create a user profile. That profile will then be used for any subsequent logins and is now the user's environment for their activity on the system. Microsoft defines the different types of user profiles:

- **Local user profile**: This profile is created when the user logs on to a computer for the first time. You will find the profile stored on the hard disk. When changes are made to the profile, the changes will be specific to the user and stored on the local computer.
- **Roaming user profile**: This profile is an administrator-created, network-based profile. The profile will be downloaded to the localhost when the user logs in to the system. When any changes are made to the profile on the localhost, changes will also be made to the server copy when the user logs off from the localhost. This profile type removes the requirement on the part of the user to create a profile when they log on to different hosts on the network. (You will only find this type of profile in Enterprise environments.)
- **Mandatory user profile**: This profile is a profile created by the network administrators to lock users down to a specific set of settings when they use a host on the network. The user will not be allowed to change the profile without the administrator's approval. Any changes

made by the user to the localhost environment will be lost when the user logs off from the localhost.

- **Temporary user profile**: This profile is created when an error occurs when the system loads the user's profile. When the user logs off, the profile is deleted. You will find the use of temporary profiles on computers running Windows 2000 and later.

Each user profile will have a registry hive – `NTUSER.DAT` – and is mapped to the system registry key of **HKEY Current User** when the user logs in. This registry hive contains the user's preferences and configuration settings.

Each user profile contains the following folders:

- `\Users\$USER$\Documents`
- `\Users\$USER$\Music`
- `\Users\$USER$\Pictures`
- `\Users\$USER$\Videos`

The `AppData` folder is a hidden folder that contains user-specific preferences and profile configurations and is further divided into three subfolders:

- `\Users\$USER$\AppData`
- `\Users\$USER$\AppData\Local`
- `\Users\$USER$\AppData\LocalLow`
- `\Users\$USER$\AppData\Roaming`

The `Roaming` folder contains data that can be synced within the server environment. Data such as web browser favorites or bookmarks will travel with the user as they log on to different workstations:

- `\Users\$USER$\AppData\Roaming\Microsoft\Windows\Cookies`

- `\Users\$USER$\AppData\Roaming\Microsoft\Windows\Network Shortcuts`
- `\Users\$USER$\AppData\Roaming\Microsoft\Windows\Printer Shortcuts`
- `\Users\$USER$\AppData\Roaming\Microsoft\Windows\Recent`
- `\Users\$USER$\AppData\Roaming\Microsoft\Windows\SendTo`
- `\Users\$USER$\AppData\Roaming\Microsoft\Windows\Start Menu`
- `\Users\$USER$\AppData\Roaming\Microsoft\Windows\Templates`

The `Local` folder contains data related to the installation of programs. It is workstation specific and will not sync with the server (in a server environment). Temporary files are also stored here:

- `\Users\$USER$\AppData\Local`
- `\Users\$USER$\AppData\Local\Microsoft\Windows\History`
- `\Users\$USER$\AppData\Local\Microsoft\Windows\Temporary Internet Files`

The `LocalLow` folder includes low-level access data, such as the temporary files of your browser when running in protected mode.

That completes our discussion on user accounts, so let’s move on to the registry, which is the heart and soul of the Windows operating system.

Understanding Windows Registry

The Windows Registry is the very heart of the Windows operating system and will be the source of many artifacts we will discuss later in the chapter. First, I will provide a high-level view of the registry. Then, suppose you want to dig deeper into the nuts and bolts of the registry. In that case, I highly recommend Harlan Carvey's book *Windows Registry Forensics – Advanced Digital Forensic Analysis of the Windows Registry*. Harlan Carvey is also the developer of the tool RegRipper, which is a tool we will use in this chapter.

What is the registry? Microsoft defines the registry as a central hierarchical database. This database is used to store configuration information about users, hardware devices, and applications.

But what does that mean for the forensic investigator? Windows continually references the information in the registry during operations. Information in the registry will contain profiles for each user, installed applications, different document types, and property settings for folders and application icons. The registry will also contain information about the hardware on the system, including networking information such as the ports used.

Wow. That was a mouthful, but in simple terms, the registry contains information about… almost everything on the computer system.

The components of the registry are found in the `%SystemRoot%\System32\Config` folder and are called hive files. You will find the `SAM`, `SECURITY`, `SOFTWARE`, and `SYSTEM` hives. Below is a brief description of the hives:

- The `SAM` hive is the Security Accounts Manager and contains login information about the users.
- The `SECURITY` hive contains security information and, potentially, password information.
- The `SOFTWARE` hive contains information about application information and the default Windows settings.

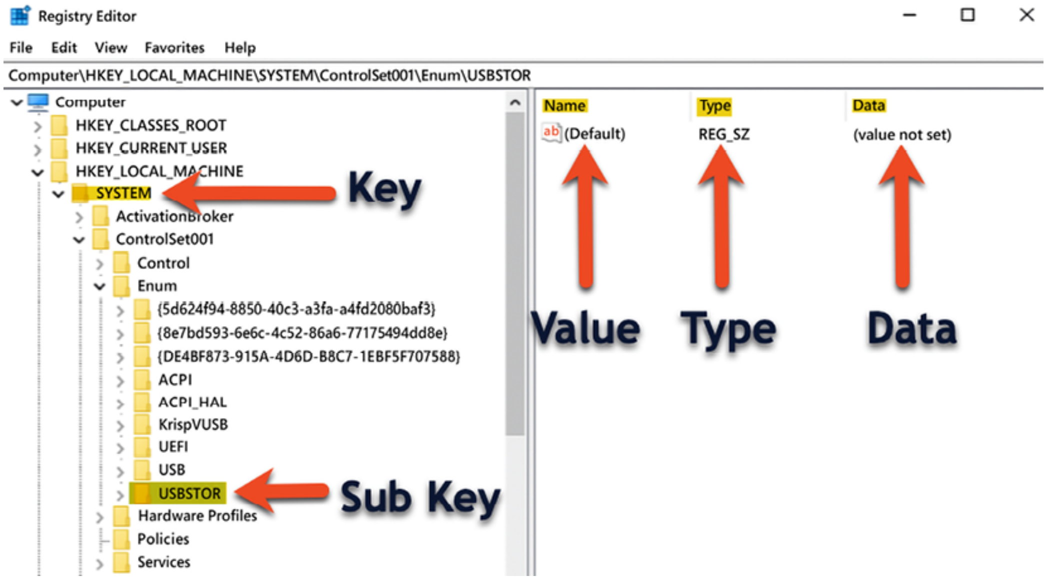
- The `SYSTEM` hive includes information on the hardware and system configuration.

There is an additional hive, `NTUser.dat`, which is stored in the root of the user profile. This hive contains information about user behavior and their settings.

Another file in the hive format is the `UsrClass.dat` file, which is found in the `\AppData\Local\Microsoft\Windows` folder of the user account. You will find information concerning **user access control** (**UAC**) configuration and information about the **graphical user interface** (**GUI**) display for the user experience.

The hive comprises subkeys that contain the **Value**, **Type**, and specific **Data** or settings being saved. This will give us a frame of reference as we explore the artifacts contained within the registry.

As you can see in the following screenshot, it is difficult to decipher the meanings of the subkeys and values and what they represent:



Computer\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\USBSTOR

Name	Type	Data
(Default)	REG_SZ	(value not set)

Figure 6.1: Registry Editor showing the USBSTOR registry key

As we go through the artifacts, I will show you the view you will see with the Registry Viewer and the easier-to-read parsed version created by the forensic tools.

We will use some open-source tools during this chapter:

- RegRipper (available for download from https://github.com/keydet89/RegRipper3.0), created by Harlan Carvey.
- Eric Zimmerman (whose work is available for download from https://ericzimmerman.github.io/#!index.md) has created several open-source utilities to parse Windows artifacts.

There are several categories in which we look for artifacts. I like to use the SANS catalog descriptions of the artifacts, which can be found at https://digital-forensics.sans.org/community/posters and are listed as follows:

- Account usage
- File knowledge
- Physical location
- Program execution
- USB/drive usage
- Browser usage (which we will discuss in *Chapter 9*, *Internet Artifacts*)

With this understanding of the user profile, we will now discuss the artifacts that determine what actions are associated with the user accounts.

Determining account usage

Identifying the user behind the keyboard is one of the hardest things you must do when conducting a digital forensic examination. You will have to parse through many artifacts to help make that determination. First, you will want to gather as much information about the user account in question and see whether you can relate it to the physical person. You will want to gain as much information about that user account and its activity related to the matter you are investigating. We will now go over some artifacts from a Windows-based operating system that will help you determine and identify that account activity starting with the user's last login or password change.

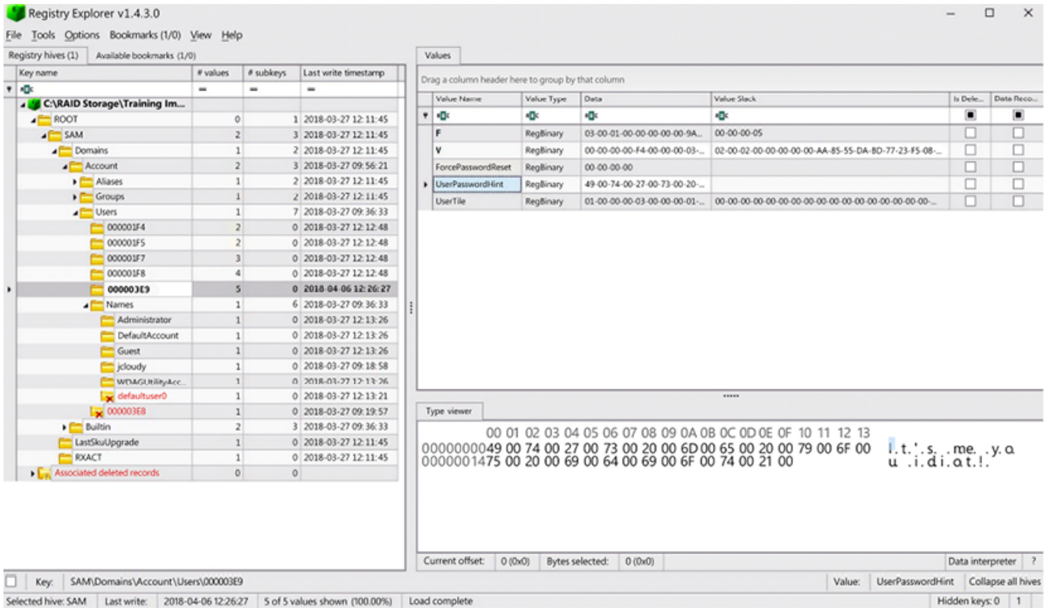
Last login/last password change

The following path will contain information about the user accounts on the system:

```
C:\windows\system32\config\SAM\Domains\Account\Users
```

To navigate to the location that contains the user account information, I will use Eric Zimmerman's Registry Explorer. I have exported the registry hive files from the forensic image to run Registry Explorer and RegRipper.

In the following screenshot, we can see that I have already opened the folder path and the subkeys, and within the **Users** subkey, there are folders with hexadecimal names and a folder entitled `Names`. Within the `Names` subkey, you see a listing of the accounts on the machine:



Key name	# values	# subkeys	Last write timestamp
C:\RAID Storage\Training Im...			
ROOT	0	1	2018-03-27 12:11:45
SAM	2	3	2018-03-27 12:11:45
Domains	1	2	2018-03-27 12:11:45
Account	2	3	2018-03-27 09:56:21
Aliases	1	2	2018-03-27 12:11:45
Groups	1	2	2018-03-27 12:11:45
Users	1	7	2018-03-27 09:36:33
000001F4	2	0	2018-03-27 12:12:48
000001F5	2	0	2018-03-27 12:12:48
000001F7	3	0	2018-03-27 12:12:48
000001F8	4	0	2018-03-27 12:12:48
000003E9	5	0	2018-04-06 12:26:27
Names	1	6	2018-03-27 09:36:33
Administrator	1	0	2018-03-27 12:13:26
DefaultAccount	1	0	2018-03-27 12:13:26
Guest	1	0	2018-03-27 12:13:26
jcloudy	1	0	2018-03-27 09:18:58
WDAGUtilityAcc...	1	0	2018-03-27 12:13:26
defaultuser0	1	0	2018-03-27 12:13:21
000003E8	1	0	2018-03-27 09:19:57
Builtin	2	3	2018-03-27 09:36:33
LastSkuUpgrade	1	0	2018-03-27 12:11:45
RXACT	1	0	2018-03-27 12:11:45
Associated deleted records	0	0	

Drag a column header here to group by that column

Value Name	Value Type	Data	Value Slack	Is Dele...	Data Reco...
F	RegBinary	03-00-01-00-00-00-00-00-9A...	00-00-00-05	□	□
V	RegBinary	00-00-00-00-F4-00-00-00-03-...	02-00-02-00-00-00-00-00-AA-85-55-DA-BD-77-23-F5-08-...	□	□
ForcePasswordReset	RegBinary	00-00-00-00		□	□
UserPasswordHint	RegBinary	49-00-74-00-27-00-73-00-20-...		□	□
UserTile	RegBinary	01-00-00-00-03-00-00-00-01-...	00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-...	□	□

```
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13
0000000049 00 74 00 27 00 73 00 20 00 6D 00 65 00 20 00 79 00 6F 00   I.t.'.s. .me. .y.o
0000001475 00 20 00 69 00 64 00 69 00 6F 00 74 00 21 00               u. .i.d.i.o.t.!.
```

Current offset: 0 (0x0) | Bytes selected: 0 (0x0) | Data interpreter | ?

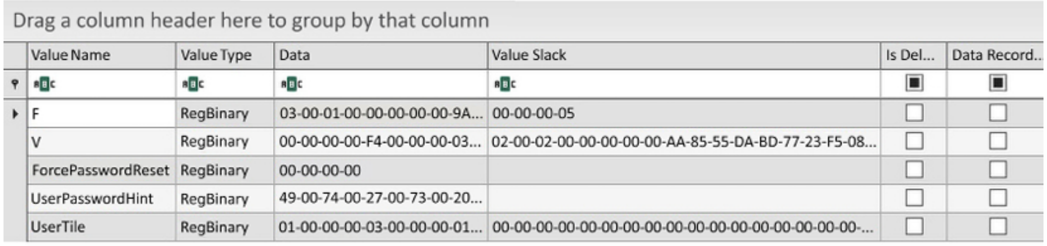
□ Key: SAM\Domains\Account\Users\000003E9 | Value: UserPasswordHint | Collapse all hives

Selected hive: SAM | Last write: 2018-04-06 12:26:27 | 5 of 5 values shown (100.00%) | Load complete | Hidden keys: 0 | 1

Figure 6.2: Registry Explorer showing the USERS key and subkeys

It lists the names in English, so they are easily readable. Out of the six accounts being shown, one has been deleted (`defaultuser0`), and one has the username of `jcloudy`. The value for the `jcloudy` subkey will point to the subkeys with the hexadecimal values. Here, `jcloudy` points to `x3E9`.

In subkey `x3E9`, as shown in the following screenshot, I see that I have an **F** and a **V** value and below that, I can see information pertaining to the user's passwords:

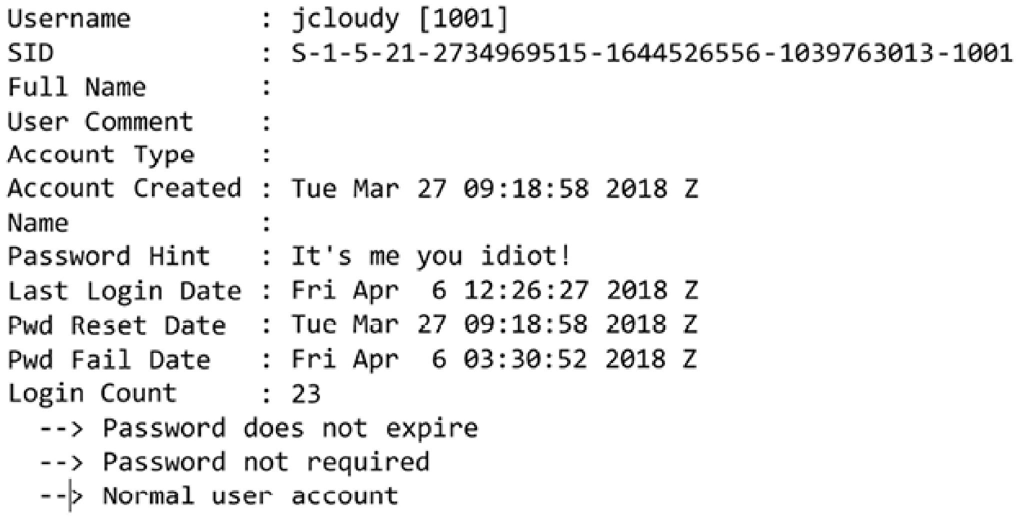


Drag a column header here to group by that column

Value Name	Value Type	Data	Value Slack	Is Del...	Data Record...
F	RegBinary	03-00-01-00-00-00-00-00-9A...	00-00-00-05	□	□
V	RegBinary	00-00-00-00-F4-00-00-00-03-...	02-00-02-00-00-00-00-00-AA-85-55-DA-BD-77-23-F5-08...	□	□
ForcePasswordReset	RegBinary	00-00-00-00		□	□
UserPasswordHint	RegBinary	49-00-74-00-27-00-73-00-20...		□	□
UserTile	RegBinary	01-00-00-00-03-00-00-00-01...	00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-...	□	□

Figure 6.3: Registry subkey X3E9

To make it easier, we can run RegRipper and see whether we can get an easier-to-read output. An example of the output for the `jcloudy` account is as follows:



```
Username        : jcloudy [1001]
SID             : S-1-5-21-2734969515-1644526556-1039763013-1001
Full Name       :
User Comment    :
Account Type    :
Account Created : Tue Mar 27 09:18:58 2018 Z
Name            :
Password Hint   : It's me you idiot!
Last Login Date : Fri Apr  6 12:26:27 2018 Z
Pwd Reset Date  : Tue Mar 27 09:18:58 2018 Z
Pwd Fail Date   : Fri Apr  6 03:30:52 2018 Z
Login Count     : 23
  --> Password does not expire
  --> Password not required
  --> Normal user account
```

Figure 6.4: RegRipper output for the jcloudy account

RegRipper parses the data and presents it in an easy-to-read format. And we can see when the account was created, the password hint, the last time the user logged in, and the number of times the user has logged in to the system.

As you look at the username `jcloudy`, you can see the numerals `1001`, and below that, an entry marked `SID`.

SID is the **security identifier** used by the Windows operating system to identify objects within. This is how Windows addresses components internally. At the end of the SID is the **relative identifier** (**RID**), which is the last digits after the SID. For example, if you see `500` as the RID, that will identify the administrator account for that system. The guest account would have an RID of `501`. In this case, as shown in the following diagram, we see the RID of `1001`. This informs me that the `jcloudy` account is user-created, and is not an account created by the system through an automated process:

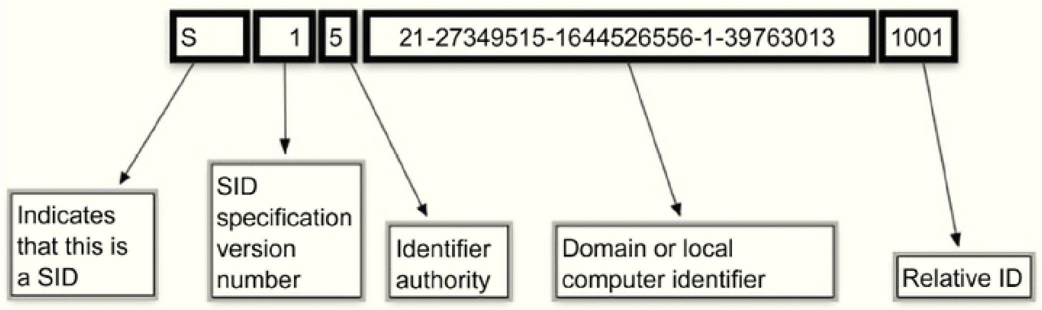

S
1
5
21-27349515-1644526556-1-39763013
1001
Indicates that this is a SID
SID specification version number
Identifier authority
Domain or local computer identifier
Relative ID

Figure 6.5: Breakdown of the SID

When doing your exam, the most commonly looked at portion of the SID is the RID. We can associate the RID with a specific user account. As the user creates accounts on the system, the RID will increase by one digit. For example, we could have a user, `user X`, with an RID of `1005`, and if I cannot find accounts `1001` through `1004`, it is possible that someone/something deleted those user accounts.

We are going through the registry to find artifacts that support (or do not support) our hypothesis about what occurred. Another source of information to help determine what happened on the system are the event logs.

Windows categorizes events into three different classes:

- **System**: Information generated by the Windows operating system
- **Application**: Information generated by applications on the local machine
- **Security**: Information related to login attempts

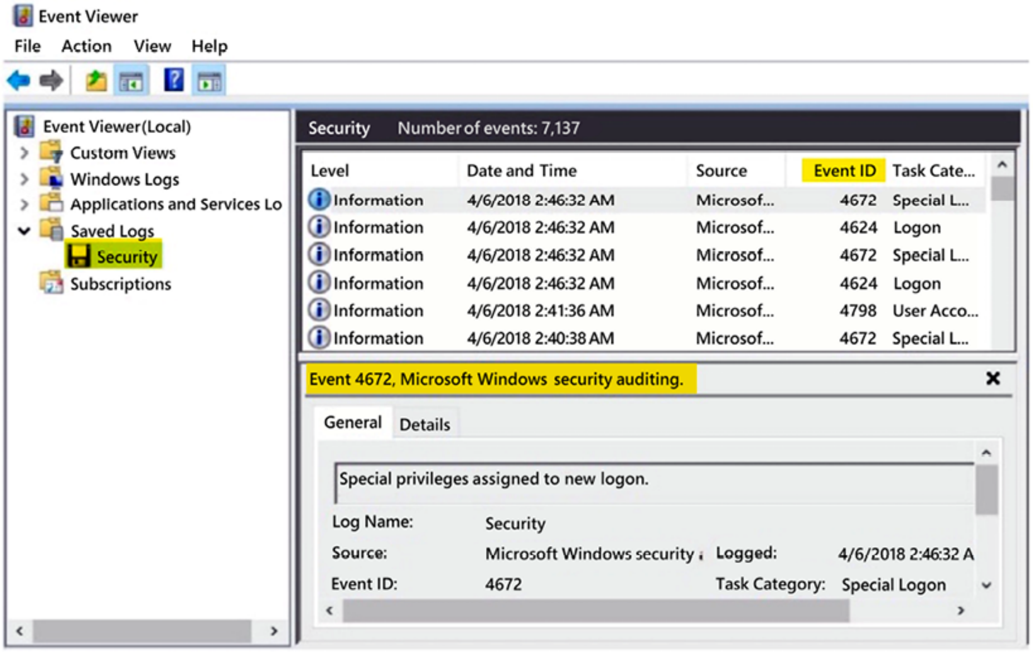
In Windows Vista through Windows 10, we can find the event logs at the following path:

```
C:\Windows\System32\winevt\logs
```

A common excuse that users give when they are accused of using the system for criminal or inappropriate reasons is that someone else had access to their system. **Remote Desktop Protocol** (**RDP**) is a way to access a host from another location. The security log will record any access using the RDP protocol. You will want to look for event ID numbers **4778** and **4779**, which would show you when the service connected/reconnected and when it disconnected.

You can also search for the type of logon into the system. For example, when we examine the security log for event ID **4624**, this will tell us the day, time, username, and the means with which the login was successful. As you can see in the following screenshot of **Event Viewer**, you can use this application to review the exported log files.

Once you have loaded the selected log file you want to examine, you can filter the results only to show the events that are relevant to your investigation:



Security Number of events: 7,137

Level	Date and Time	Source	Event ID	Task Cate...
Information	4/6/2018 2:46:32 AM	Microsof...	4672	Special L...
Information	4/6/2018 2:46:32 AM	Microsof...	4624	Logon
Information	4/6/2018 2:46:32 AM	Microsof...	4672	Special L...
Information	4/6/2018 2:46:32 AM	Microsof...	4624	Logon
Information	4/6/2018 2:41:36 AM	Microsof...	4798	User Acco...
Information	4/6/2018 2:40:38 AM	Microsof...	4672	Special L...

Event 4672, Microsoft Windows security auditing.

Special privileges assigned to new logon.

Log Name: Security

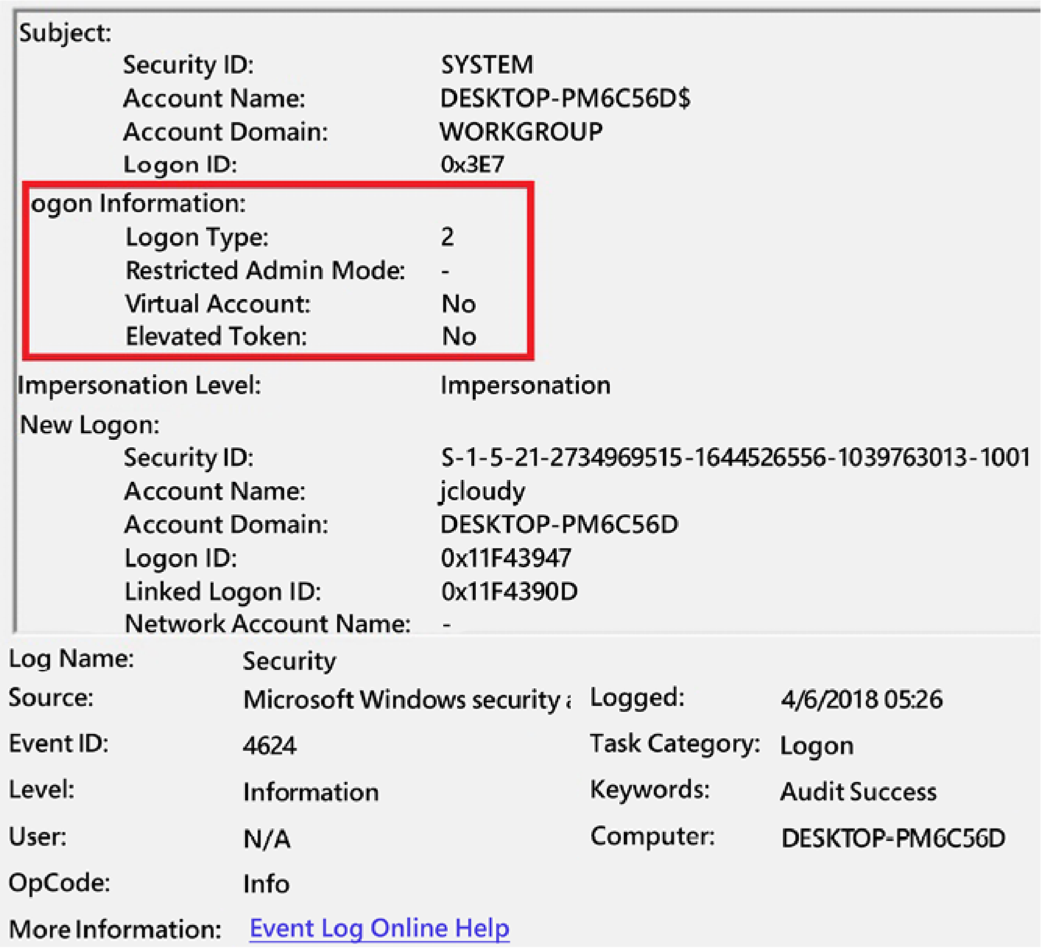
Source: Microsoft Windows security a Logged: 4/6/2018 2:46:32 A

Event ID: 4672 Task Category: Special Logon

Figure 6.6: Event Viewer displaying event information

The type of logon is also significant. Was the user sitting at the keyboard, or did the user log in from a remote site? Event ID `4624` will identify the login type used by the user. In the following screenshot, you can see the output of Event Viewer showing when the user logged in and the login type.

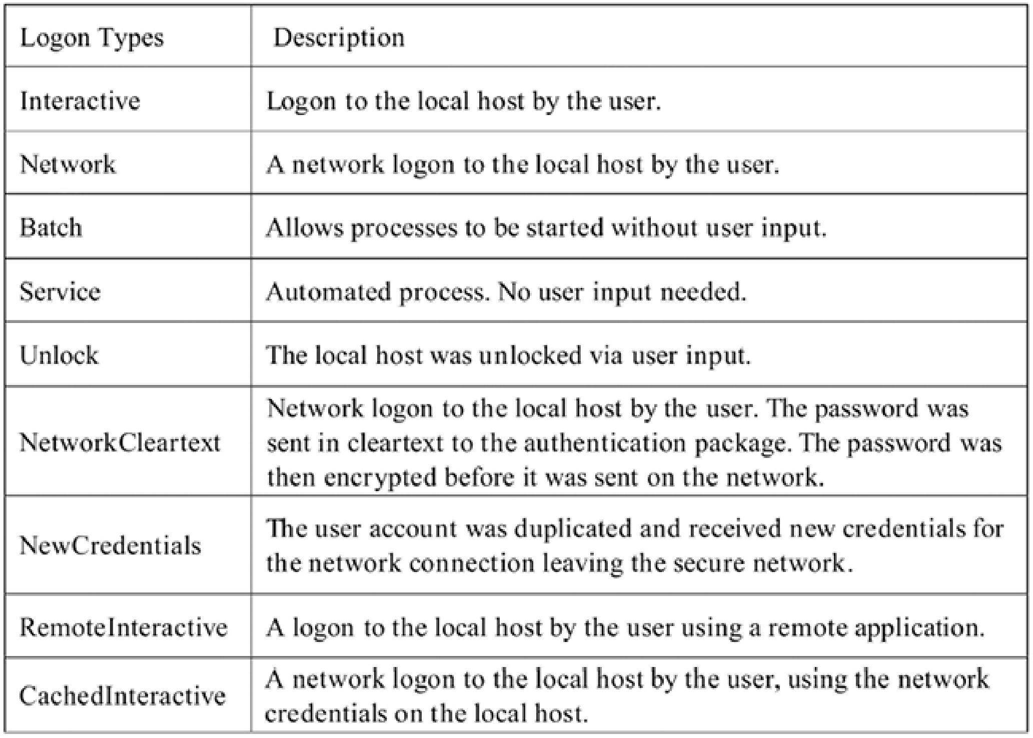
Here, it shows the user's login was type 2, which is "interactive":



```
Subject:
        Security ID:                SYSTEM
        Account Name:               DESKTOP-PM6C56D$
        Account Domain:             WORKGROUP
        Logon ID:                   0x3E7
ogon Information:
        Logon Type:                 2
        Restricted Admin Mode:      -
        Virtual Account:            No
        Elevated Token:             No
Impersonation Level:                Impersonation
New Logon:
        Security ID:                S-1-5-21-2734969515-1644526556-1039763013-1001
        Account Name:               jcloudy
        Account Domain:             DESKTOP-PM6C56D
        Logon ID:                   0x11F43947
        Linked Logon ID:            0x11F4390D
        Network Account Name:       -
Log Name:          Security
Source:            Microsoft Windows security a  Logged:          4/6/2018 05:26
Event ID:          4624                          Task Category:   Logon
Level:             Information                   Keywords:        Audit Success
User:              N/A                           Computer:        DESKTOP-PM6C56D
OpCode:            Info
More Information:  Event Log Online Help
```

Figure 6.7: Event Viewer showing the logon type

The following is a list from Microsoft of the other logon types you may encounter, together with their descriptions:



Logon Types	Description
Interactive	Logon to the local host by the user.
Network	A network logon to the local host by the user.
Batch	Allows processes to be started without user input.
Service	Automated process. No user input needed.
Unlock	The local host was unlocked via user input.
NetworkCleartext	Network logon to the local host by the user. The password was sent in cleartext to the authentication package. The password was then encrypted before it was sent on the network.
NewCredentials	The user account was duplicated and received new credentials for the network connection leaving the secure network.
RemoteInteractive	A logon to the local host by the user using a remote application.
CachedInteractive	A network logon to the local host by the user, using the network credentials on the local host.

Figure 6.8: Microsoft logon types

You may also want to establish the attempted login events to determine whether an attacker compromised the account. The following event IDs will help you make that determination:

- `4624` - Successful logon
- `4625` - Unsuccessful logon
- `4634` - Logon session terminated
- `4647` - Logon session terminated by the user
- `4648` - User logon was attempted by a user using different credentials
- `4672` - User logon with Admin rights
- `4720` - User account created

A full list of Microsoft Windows Event IDs can be found at:

https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/

Suppose you see many failed logins, or a user was granted administrator rights when they usually do not possess superuser rights. In that case, these event ID clues provide you with additional investigative avenues to determine what occurred.

Now that we've examined the user's account activity, next we will discuss the artifacts associated with user account file access.

Determining file knowledge

Some incidents you investigate may deal with contraband images, stolen data, or unlawful access to data. You will have to determine whether the user had knowledge of the file(s) in question, or whether the file(s) existed on the user's system.

We will now talk about some artifacts you can find in the Windows operating system that will help you make that determination.

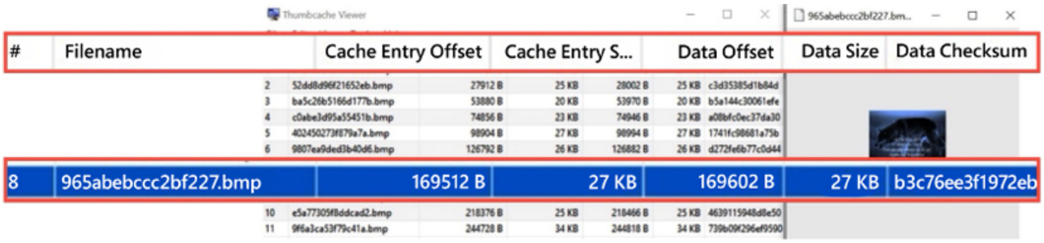
Exploring the thumbcache

A thumbcache is a database of thumbnail images created when using Windows Explorer in a thumbnail view. Depending on the size of the thumbnail, you may have multiple databases with the same image but with different sizes. It depends on the view the user selected while in Windows Explorer. The existence of an image found in the database is not substantial proof that the user knew the image was on the system. The system can add a thumbnail to the cache without the user’s knowledge. The thumbcache can be found in the user’s profile at the following path:

```
AppData\Local\Microsoft\Windows\Explorer
```

Your commercial forensic tools will process the thumbcache with no issues. If you want to use an open-source utility, you can use Thumbcache Viewer (which can be downloaded at https://thumbcacheviewer.github.io/).

The following is an example of the output of Thumbcache Viewer:



#	Filename	Cache Entry Offset	Cache Entry S...	Data Offset	Data Size	Data Checksum
2	52dd8d96f21652eb.bmp	27912 B	25 KB	28002 B	25 KB	c3d35385d1b84d
3	ba5c26b5166d177b.bmp	53880 B	20 KB	53970 B	20 KB	b5a144c30061efe
4	c0abe3d95a55451b.bmp	74856 B	23 KB	74946 B	23 KB	a08bfc0ec37da30
5	402450273f879a7a.bmp	98904 B	27 KB	98994 B	27 KB	1741fc98681a75b
6	9807ea9ded3b40d6.bmp	126792 B	26 KB	126882 B	26 KB	d272fe6b77c0d44
8	965abebccc2bf227.bmp	169512 B	27 KB	169602 B	27 KB	b3c76ee3f1972eb
10	e5a77305f8ddcad2.bmp	218376 B	25 KB	218466 B	25 KB	4639115948d8e50
11	9f6a3ca53f79c41a.bmp	244728 B	34 KB	244818 B	34 KB	739b09f296ef9590

Figure 6.9: Thumbcache Viewer output

As you can see, the thumbnail does not have the same filename as the source image. To identify the original file that was used to create the thumbnail, we need to look in the Windows Search Indexing database, `Windows.edb`, which can be found at the following path:

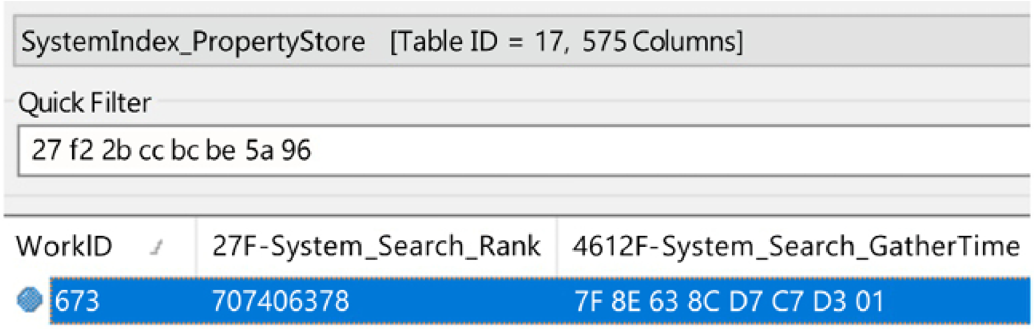
```
C:\ProgramData\Microsoft\Search\Data\Applications\Windows\Window
```

You will need an additional tool to find the information about the image used to create the thumbnail. You can use `ESEDatabaseView` (located at https://www.nirsoft.net/utils/ese_database_view.html).

The thumbnail name is `96 5a be bc cc 2b f2 27`, which is made up of hexadecimal characters. We need to reverse the values to search the database, so we will want to search for `27 f2 2b cc bc be 5a 96`. The information we are looking for is in different locations depending on the operating system.

- On a Windows 7 system, you want to examine the contents of the table `SystemIndex_0A`.
- On a Windows 8/10 computer, you want to examine the contents of the table `SystemIndex_PropertyStore`.

Once we input the hexadecimal values into the filter, it reduces the data to a single row:



SystemIndex_PropertyStore [Table ID = 17, 575 Columns]

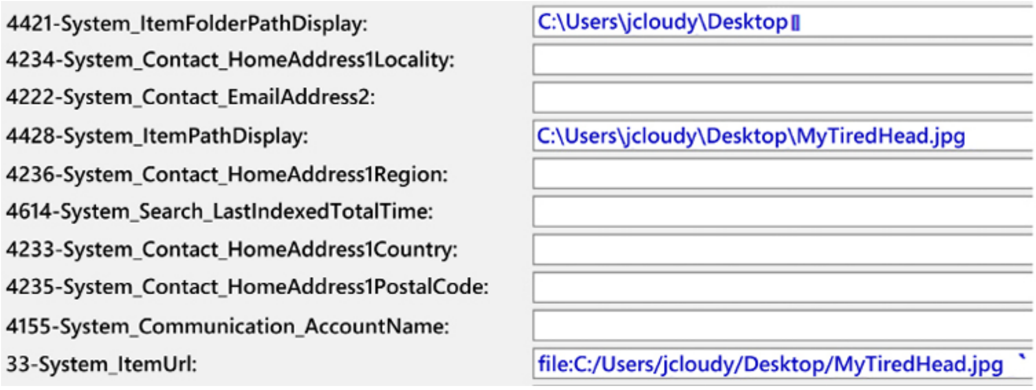
Quick Filter

27 f2 2b cc bc be 5a 96

WorkID	27F-System_Search_Rank	4612F-System_Search_GatherTime
673	707406378	7F 8E 63 8C D7 C7 D3 01

Figure 6.10: Filtered database results

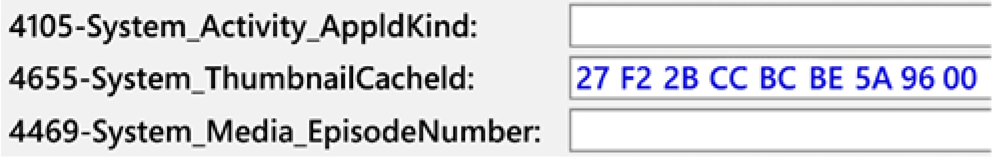
In the following screenshot, we can see that the file came from the desktop of the user `jcloudy`. The name of the image is `MyTiredHead.jpg`:



4421-System_ItemFolderPathDisplay:	C:\Users\jcloudy\Desktop
4234-System_Contact_HomeAddress1Locality:	
4222-System_Contact_EmailAddress2:	
4428-System_ItemPathDisplay:	C:\Users\jcloudy\Desktop\MyTiredHead.jpg
4236-System_Contact_HomeAddress1Region:	
4614-System_Search_LastIndexedTotalTime:	
4233-System_Contact_HomeAddress1Country:	
4235-System_Contact_HomeAddress1PostalCode:	
4155-System_Communication_AccountName:	
33-System_ItemUrl:	file:C:/Users/jcloudy/Desktop/MyTiredHead.jpg

Figure 6.11: Filename display in the database

In the following screenshot, we can verify that this is the correct file when we look in the `System_ThumbnailCacheID` field:



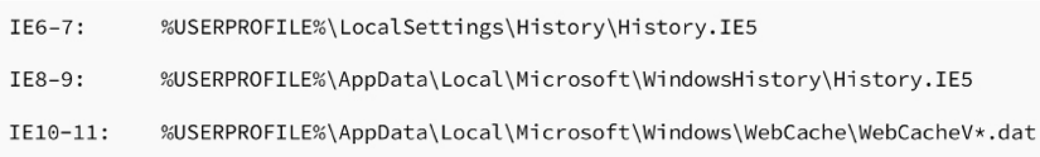
4105-System_Activity_AppIdKind:	
4655-System_ThumbnailCacheId:	27 F2 2B CC BC BE 5A 96 00
4469-System_Media_EpisodeNumber:	

Figure 6.12: Thumbnail name in the database

That completes the discussion on the thumbcache. We will now explore the artifacts created by the Edge/Internet Explorer/File Explorer browsers.

Exploring Microsoft browsers

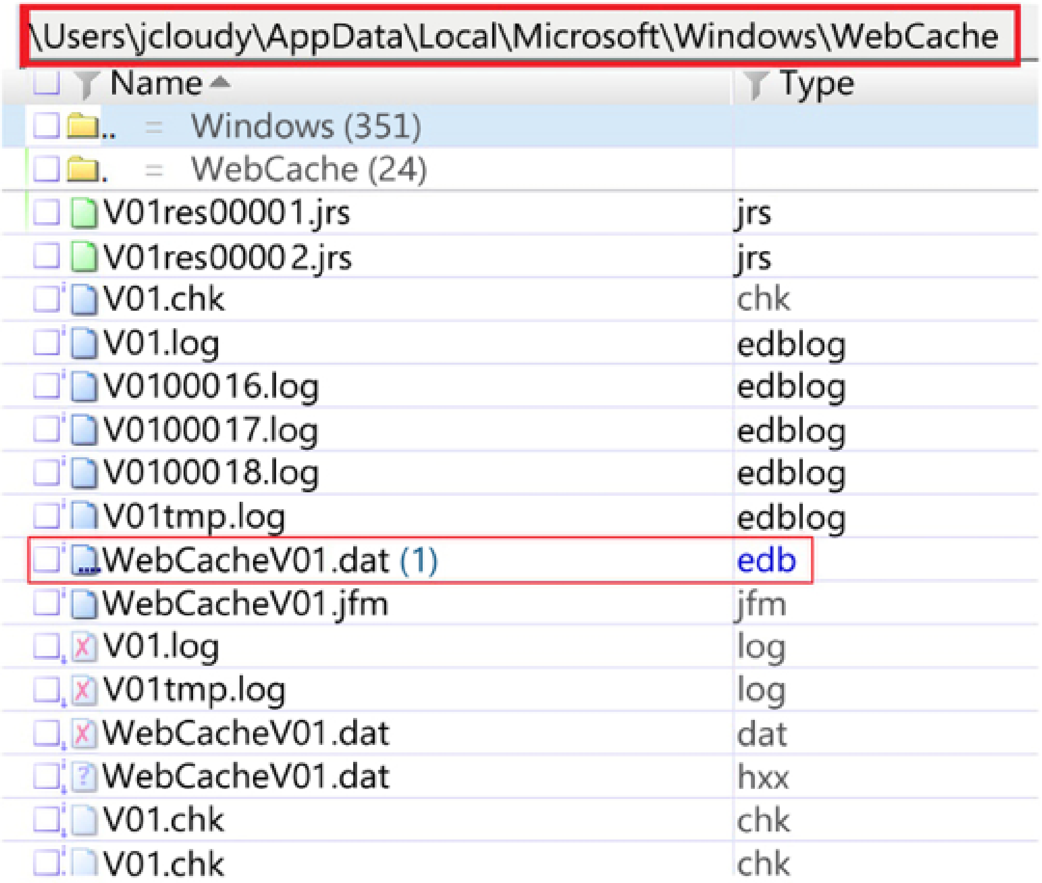
Microsoft uses the same method to record a user's file activity and internet history with the Internet Explorer/File Explorer/Edge browsers. In addition, it records local and remote file access. Most commercial forensics tools parse these files easily. Depending on the version, the history file will be in the following areas:



```
IE6-7:      %USERPROFILE%\LocalSettings\History\History.IE5
IE8-9:      %USERPROFILE%\AppData\Local\Microsoft\WindowsHistory\History.IE5
IE10-11:    %USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat
```

Figure 6.13: IE Locations

In the following screenshot, you can see that the user is using version 10/11 because of the existence of the `WebCacheV01.dat` file:



\Users\jcloudy\AppData\Local\Microsoft\Windows\WebCache

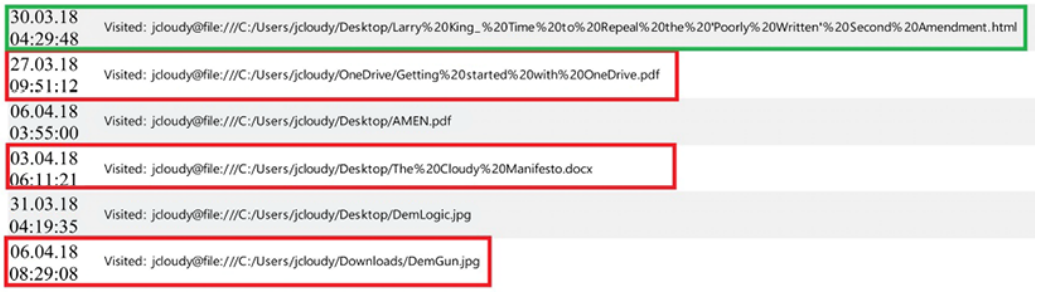
Name	Type
.. = Windows (351)	
. = WebCache (24)	
V01res00001.jrs	jrs
V01res00002.jrs	jrs
V01.chk	chk
V01.log	edblog
V0100016.log	edblog
V0100017.log	edblog
V0100018.log	edblog
V01tmp.log	edblog
WebCacheV01.dat (1)	edb
WebCacheV01.jfm	jfm
V01.log	log
V01tmp.log	log
WebCacheV01.dat	dat
WebCacheV01.dat	hxx
V01.chk	chk
V01.chk	chk

Figure 6.14: File Explorer showing the WebCacheV01.dat file

The `.dat` file is an ESE database. If you want to use a single-use forensic tool, you can export the `.dat` file out of the forensic image and view it with an open-source forensic tool such as `ESEDatabaseView`.

(Located at https://www.nirsoft.net/utils/ese_database_view.html)

You will want to navigate to the `Containers` table. The following screenshot is the output from X-Ways Forensics:



30.03.18 04:29:48	Visited: jcloudy@file:///C:/Users/jcloudy/Desktop/Larry%20King_%20Time%20to%20Repeal%20the%20'Poorly%20Written'%20Second%20Amendment.html
27.03.18 09:51:12	Visited: jcloudy@file:///C:/Users/jcloudy/OneDrive/Getting%20started%20with%20OneDrive.pdf
06.04.18 03:55:00	Visited: jcloudy@file:///C:/Users/jcloudy/Desktop/AMEN.pdf
03.04.18 06:11:21	Visited: jcloudy@file:///C:/Users/jcloudy/Desktop/The%20Cloudy%20Manifesto.docx
31.03.18 04:19:35	Visited: jcloudy@file:///C:/Users/jcloudy/Desktop/DemLogic.jpg
06.04.18 08:29:08	Visited: jcloudy@file:///C:/Users/jcloudy/Downloads/DemGun.jpg

Figure 6.15: X-Ways display of the contents of the WebCache

As you can see, we have a date and timestamp and the file path of the file that was viewed. We have one offline HTML file (the first line), which was located on the user's desktop. We see the user opened two PDF files, two JPEG files, one HTML file, and one DOCX file.

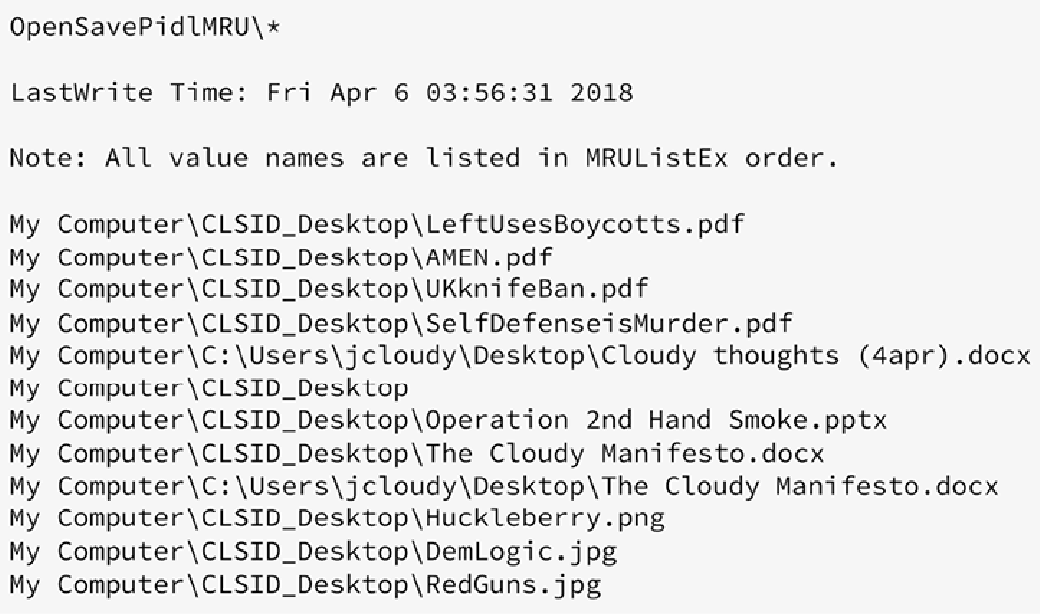
There are additional artifacts that show that a user account accessed a file, which we will discuss next.

Determining most recently used/recently used

An **MRU** (**Most Recently Used**) is a list of recently used files stored in the user's **NTUSER.DAT** hive. When you open an application and see the history list of prior files that the application has used, you are looking at an

MRU. There are a lot of MRU lists stored within the registry file. We will go over some more common locations.

OpenSavePidlMRU from the user’s **NTUSER.DAT** file tracks the last 20 files opened/saved via the Windows Common Dialogue (these are the commonly encountered **Open/Save As** dialog boxes). In the following example, we can see the last 20 files used by the user:



```
OpenSavePidlMRU\*

LastWrite Time: Fri Apr 6 03:56:31 2018

Note: All value names are listed in MRUListEx order.

My Computer\CLSID_Desktop\LeftUsesBoycotts.pdf
My Computer\CLSID_Desktop\AMEN.pdf
My Computer\CLSID_Desktop\UKknifeBan.pdf
My Computer\CLSID_Desktop\SelfDefenseisMurder.pdf
My Computer\C:\Users\jcloudy\Desktop\Cloudy thoughts (4apr).docx
My Computer\CLSID_Desktop
My Computer\CLSID_Desktop\Operation 2nd Hand Smoke.pptx
My Computer\CLSID_Desktop\The Cloudy Manifesto.docx
My Computer\C:\Users\jcloudy\Desktop\The Cloudy Manifesto.docx
My Computer\CLSID_Desktop\Huckleberry.png
My Computer\CLSID_Desktop\DemLogic.jpg
My Computer\CLSID_Desktop\RedGuns.jpg
```

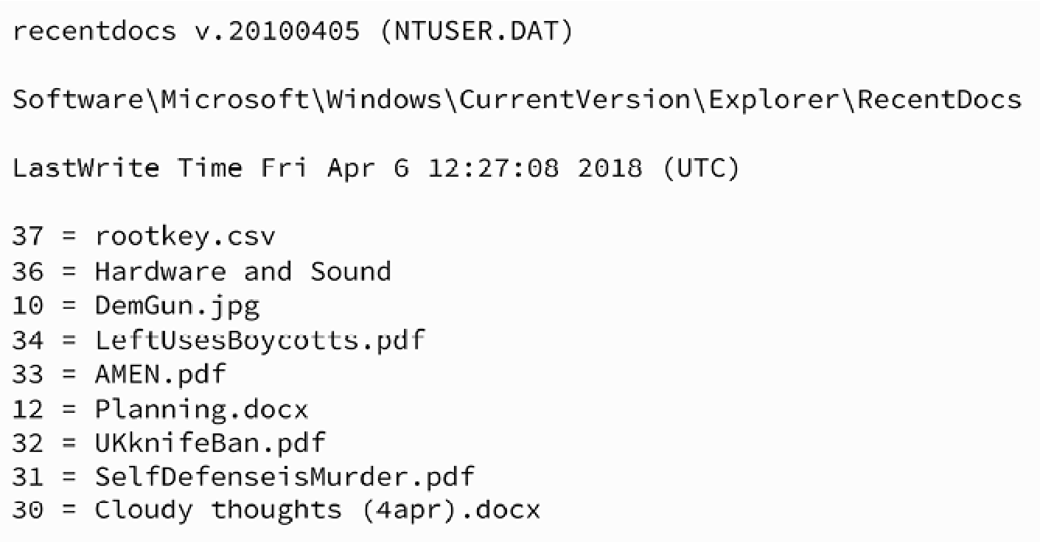
Figure 6.16: Content of NTUSER.DAT key - OpenSavePidlMRU

Another key to look at is:

```
NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\Re
```

This key contains a list of files executed/opened by the user through the Windows Explorer application. You will also have subkeys, based on file extensions, listing those files that were executed/opened. The system will store the entries in chronological order of when the files were executed/opened by the user.

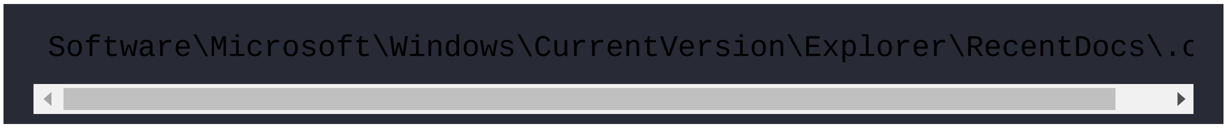
When looking at the last entry/modified time of the key, it will correspond to the last entry in the list. This key will keep track of the previous 150 files that were opened/executed. The following is the output of the key (I am only showing the top-level entries for brevity's sake):



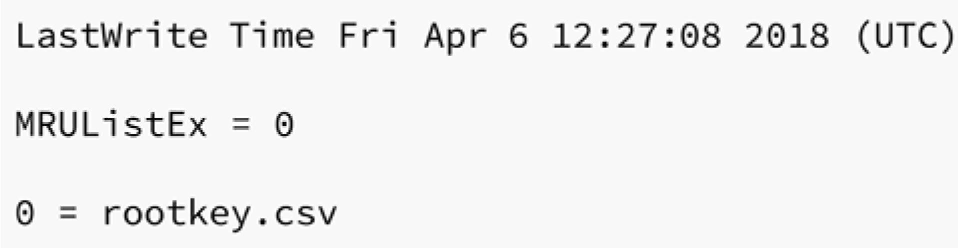
```
recentdocs v.20100405 (NTUSER.DAT)

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

LastWrite Time Fri Apr 6 12:27:08 2018 (UTC)

37 = rootkey.csv
36 = Hardware and Sound
10 = DemGun.jpg
34 = LeftUsesBoycotts.pdf
33 = AMEN.pdf
12 = Planning.docx
32 = UKknifeBan.pdf
31 = SelfDefenseisMurder.pdf
30 = Cloudy thoughts (4apr).docx
```

Figure 6.17: Recent Docs entries

This is an example of the file extension subkeys I described earlier, and it shows the recently used CSV files:



```
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.c
```

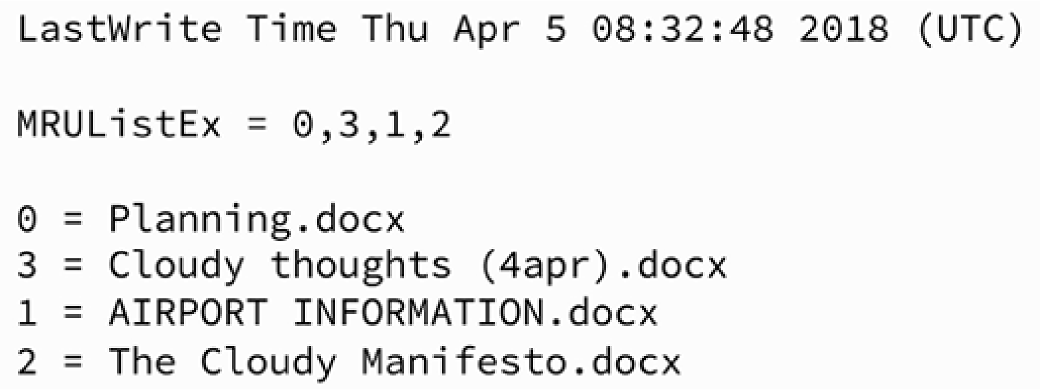


```
LastWrite Time Fri Apr 6 12:27:08 2018 (UTC)

MRUListEx = 0

0 = rootkey.csv
```

Figure 6.18: Content of NTUSER.DAT key - Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs for CSV files

This is an example of the file extension subkeys I described earlier, and it shows the recently used DOCX files:

```
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.
```

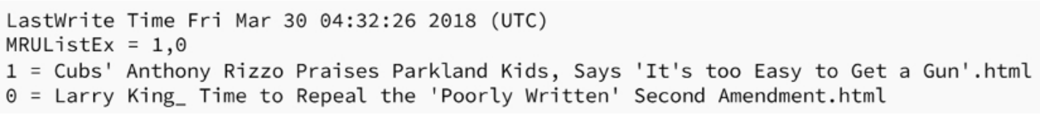


```
LastWrite Time Thu Apr 5 08:32:48 2018 (UTC)

MRUListEx = 0,3,1,2

0 = Planning.docx
3 = Cloudy thoughts (4apr).docx
1 = AIRPORT INFORMATION.docx
2 = The Cloudy Manifesto.docx
```

Figure 6.19: Content of NTUSER.DAT key - Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.docx

This is an example of the file extension subkeys I described earlier, and it shows the recently used HTML files:

```
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.
```



```
LastWrite Time Fri Mar 30 04:32:26 2018 (UTC)
MRUListEx = 1,0
1 = Cubs' Anthony Rizzo Praises Parkland Kids, Says 'It's too Easy to Get a Gun'.html
0 = Larry King_ Time to Repeal the 'Poorly Written' Second Amendment.html
```

Figure 6.20: Content of NTUSER.DAT key - Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.html

There is also an additional subkey, `\Folder`, that lists when the user opened folders on the system, which is shown as follows:

```
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\F
```

Graphical user interface, text, application Description automatically generated

Figure 6.21: Content of NTUSER.DAT key - Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\Folders

Entries of potential interest include OneDrive and CloudLog. If I am looking for evidence of specific files, the subject may store the data in cloud storage. When I see artifacts showing the use of cloud storage, it provides additional locations that I will have to locate and acquire the digital evidence to continue my digital forensic investigation.

As you can see, these are great artifacts to see what files were accessed by the user, but what happens when the user deletes a file? That leads us on to our next topic, the Recycle Bin.

Looking into the Recycle Bin

The **Recycle Bin** is Microsoft’s effort to protect users from their actions. It provides an intermediary step for when a user deletes a file. Windows will move the file into a holding area known as the **Recycle Bin**.

The **Recycle Bin** is a hidden folder stored in the root directory of every fixed disk on the system. The folder name is `$Recycle.Bin`. On an NTFS formatted disk, there will be sub-folders named with the user’s SID. These sub-folders are created whenever a user logs on to the system for the first time:

- `$Recycle.Bin`
    - `S-1-5-21-2734969515-1644526556-1039763013-1001`

When a user deletes a file, the original file gets renamed and becomes part of a set of `Recycle.Bin` files. The system will rename the original file with `$R` and then six random alphanumeric characters for the filename. The file extension will remain the same. The system will create a second file, which will start with `$I` and then have the same six alphanumeric characters that the `$R` file has. The `$I` file will also have the same file extension as the `$R` file.

The `$I` file will track the time of deletion and the path to the original file location:

- **Size**: 4.9 MB
- **Moved to recycle bin**: `04/05/2018 02:20:17 +0` `C:\Users\jcloudy\Desktop\Larry King_ Time to Repeal the 'Poorly Written' Second Amendment_files`

As you can see, we have the size of the original file, when the user deleted it, and the original path that includes the filename.

If a user deletes a directory, you will still have the `$R` and `$I` files for the directory. The `$R` file will contain all the subdirectories and all the files with the original names, as shown in the following screenshot:

Figure 6.22: Deleted directory

The user can empty the **Recycle Bin**. When that occurs, the filesystem updates that the clusters are now available for use. Until the system overwrites the data, you may recover data from the unallocated clusters. Just be aware that the `$I` (on an NTFS volume) will be resident data in the

MFT. NTFS is very efficient in reusing the file entries in the MFT, so it’s challenging to recover the information in the `$I` file.

If the **Recycle Bin** is emptied, other artifacts may be referencing the file(s). That brings us to our next topic, link (LNK) files.

Understanding shortcut (LNK) files

A `.lnk` file is used by the Windows operating system as a shortcut or link to files, applications, or resources. It is a simple, easy-to-use method for users to gain access to frequently used documents or applications. The link file will contain useful information for the digital forensic investigator, including the following:

- File MAC times
- File size
- File path
- Volume details

This information will remain even if the destination file has been deleted. The system will create a link file every time the file is double-clicked or when using the **File Open** dialog box. These link files will be stored in the `Recent` folder located at the following path:

```
%Username%\Appdata\Roaming\Microsoft\Windows\
```

Most commercial forensic tools can analyze link files. An open-source option is Eric Zimmerman’s LECmd tool (which can be found at https://ericzimmerman.github.io/).

When we analyze the contents of the link file, we can see a large amount of information that could be helpful to the digital forensic investigator:

Graphical user interface, text, application Description automatically generated

Figure 6.23: Link File contents

We can see that the destination file is a Microsoft Word document stored on the user’s desktop. When we look at the field ID list, we can also see the file’s internal metadata (MAC values). This data can be fundamental when trying to tie knowledge of the file to a specific user. We can also see the date/time when the system created the link file. Additional information is the volume type/serial number and hostname, which allow us to tie this link file to the specific location of the destination file. Be aware that this is an option that the user or systems administrator can turn off. Another artifact similar to LNK files is the JumpList.

Deciphering JumpLists

JumpLists were introduced with Windows 7 and are very similar to the `Recent` folder (which we discussed with LNK files). They allow the user to access frequently used/recently used files from the Windows taskbar. Even if the user clears out the `Recent` folder, it will not clear out the information stored in the JumpLists.

JumpLists can be found at the following paths:

- `%UserProfile%\AppData\Roaming\Microsoft\Windows\Recent\Automaticdesti nations`

- `%UserProfile%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations`

There are two types of JumpLists:

- Automatic – system-created. Records information about file usage.
- Custom – application-created. Records task-specific information about the application.

In the following screenshot, you can see the `AutomaticDestinations` folder, and inside the folder will be files containing the JumpLists:

Figure 6.24: JumpList display

The system names the JumpLists based on their JumpLists IDs. For example, in the preceding screenshot, we see `5d696d521de238c3.automaticDestinations-ms`. A search of the JumpLists ID list (which can be found at https://community.malforensics.com/t/list-of-jump-list-ids/158) shows that this is the JumpLists ID for the Google Chrome browser.

Most commercial forensic tools will parse out the JumpLists. An open-source option is Eric Zimmerman's JumpList Explorer.

(Located at https://ericzimmerman.github.io/)

The following is the information contained in the file. You can see that the user was using Chrome to view PDF files and offline HTML files. It also contains the date/time the user opened the files:

- `7 04/06/2018 03:56:32 +0 C:\Users\jcloudy\Desktop\LeftUsesBoycotts.pdf`
- `6 04/06/2018 03:55:00 +0 C:\Users\jcloudy\Desktop\AMEN.pdf`
- `5 04/05/2018 05:51:41 +0 C:\Users\jcloudy\Desktop\UKknifeBan.pdf`
- `4 04/05/2018 05:48:40 +0 C:\Users\jcloudy\Desktop\SelfDefenseisMurder.pdf`
- `3 03/30/2018 04:32:25 +0 C:\Users\jcloudy\Desktop\Cubs' Anthony Rizzo Praises Parkland Kids, Says 'It's too Easy to Get a Gun'.html`
- `2 03/30/2018 04:29:48 +0 C:\Users\jcloudy\Desktop\Larry King_ Time to Repeal the 'Poorly Written' Second Amendment.html`
- `1 03/27/2018 09:51:18 +0 C:\Users\jcloudy\OneDrive\Getting started with OneDrive.pdf desktop-pm6c56d`

JumpLists are artifacts for files; the next artifact will show which folders the user accessed.

Opening shellbags

Shellbags are a set of registry keys that remember the size and location of the folders and libraries that the user has accessed via the GUI. In addition, you may find artifacts showing user interaction with network devices, removable media, or encrypted containers.

You will find them in a registry hive called `USRCLASS.DAT`, located in the users, `AppData\Local\Microsoft\Windows` folder.

Most commercial forensic tools will parse out the shellbags from the `USRCLASS.DAT` file, but the presentation of the artifact will be different. I like to use Eric Zimmerman’s Shellbag Explorer as an open-source alternative.

In the following screenshot, you can see the graphical representation of the folders the user accessed via the Windows GUI. This screenshot is taken from Shellbag Explorer:

Figure 6.25: Shellbag Explorer: a graphical representation of shellbags

You cannot determine whether the user accessed any files from within the folder through this artifact. What this artifact shows is that the user accessed the folder. As I look at the display, I see that the user was using three cloud storage services. We have seen prior artifacts for Box Sync and Dropbox, but this is the first reference I have seen regarding Google Drive.

In the following output from RegRipper, we can see the access date and timestamps and the date/time of the first access:

Graphical user interface, text, application, email Description automatically generated

Figure 6.26: RegRipper output Google Drive

This artifact is important if the subject states that they did not know about a file/folder location. This artifact is created by the user’s actions. The next artifact can also be used to show user knowledge of a file.

Understanding prefetch

Prefetch is a feature Microsoft introduced to enhance the user experience with the Windows operating system. It allows faster response times by preloading data into the RAM in anticipation of its demand by the user or system. You will find the prefetch files at the following path:

```
%WINDOWS%\PREFETCH
```

The files will have a file extension of `.pf`. In addition, the prefetch file will contain information about the executable file associated with it, such as the list of files used by the executable, the number of times the user ran the executable, and the last date/time when the user ran the executable.

Most commercial forensic tools will parse out the prefetch files. For an open-source option, you can use NirSoft's **WinPrefetchViewtool**.

(Located at https://www.nirsoft.net/utils/win_prefetch_view.html)

In the following screenshot, we are looking at the output of `WinPrefetchView`. You can see the date and timestamp and the process path of the executable (be aware that due to the method with which the system monitors the prefetch files, you may have to subtract 10 seconds from the created/modified times to get an accurate time):

Figure 6.27: Prefetch files displayed by WinPrefetchView

By using these artifacts, you can determine which applications are being used by the user, which may lead to the discovery of hidden partitions, mobile devices, encrypted containers, or cloud storage.

As operating systems change or are updated, the artifacts may move or be removed. You will have to stay current as changes become known. We will now look at artifacts that help us determine the physical location of the system.

Identifying physical locations

Knowing the system's physical location may help you prove or disprove the allegations against the subject you are investigating. For example, there was an investigation into a compromise of the organization's network. A former employee was the suspect in the attack because of their threats during the termination process. When the suspect was interviewed, he denied being in the area and stated he was out of state. A judge authorized a search warrant for the suspect's mobile device and laptop computer. When conducting the forensic analysis of the laptop, the examiner found it to have been recently restored to a new version of the operating system. Artifacts in the unallocated space led us to believe the user had wiped the device. (The user overwrote all available sectors with hexadecimal 00 characters.) The suspect had not tampered with the mobile device, and we could analyze the device. We were able to map out the Wi-Fi hotspots the device had accessed in the immediate neighborhood when the suspect was allegedly out of state. When confronted with the digital evidence, the suspect confessed and admitted he forgot about his mobile device and that it was automatically connecting to Wi-Fi hotspots.

We will now talk about some artifacts you can look at in a system to help determine their physical location at the time of an incident.

Determining time zones

Time zone information on the system allows you to have a starting point with which to correlate activity that is recorded with the date/time that the incident occurred. All the internal dates and timestamps will be based on the time zone information recorded in the registry. We can find the time zone information within the system hive. We can find the key at the following path:

```
SYSTEM\CurrentControlSet\Control\TimeZoneInformation
```

This will give us the following output, courtesy of RegRipper:

Text Description automatically generated

Figure 6.28: RegRipper output - SYSTEM\CurrentControlSet\Control\TimeZoneInformation

`Tzres.dll` is the time zone resource `DLL`. You have the fields of `Bias` and `ActiveTimeBias`, which show the values of `300` and `240`, respectively, which is the number of minutes offset from GMT. And then you have the time zone common name, which in this case is `Eastern Standard Time`.

Time zones are not always accurate – the user can set the time zone to the zone of their choice. The next artifact we will examine may help in locating a physical location.

Exploring network history

Knowing which networks, be they wired or wireless, the suspect has connected to might give you location information about their whereabouts at the time in question. You will find the relevant information in the

`Software hive` or an XML document managed by the operating system. The Wi-Fi document will be found at the following path:

```
C:\ProgramData\Microsoft\Wlansvc\Profiles\Interfaces
```

This directory contains subfolders (using the GUID naming convention) for each interface. The XML document will contain the **SSID** (**Service Set Identifier**) of the networks the interface has connected to. The following output is consistent with the information you would find in the XML document:

Figure 6.29: XML Output of WLAN Profile

As you can see, the SSID of the network is `Net 2.4` and it is using `WPA2PSK` authentication.

If you go to the registry location, you will find sub hives that will contain networking information such as the `Profiles` subkey, which gives us additional information about the wireless network(s) the subject connected to:

```
SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList
```

The following is the RegRipper output of the `networklist` sub hive:

Figure 6.30: RegRipper output - SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList

The registry hive gives us a little bit more information, including the MAC address date and timestamp of when the last connection was made. There is also an additional log file we can examine: the WLAN event log.

Understanding the WLAN event log

Microsoft Windows also keeps an event log of wireless connections. The log can be found at the following path:

```
C:\windows\System32\winevt\Logs\Microsoft-Windows-WLAN-AutoConfi
```

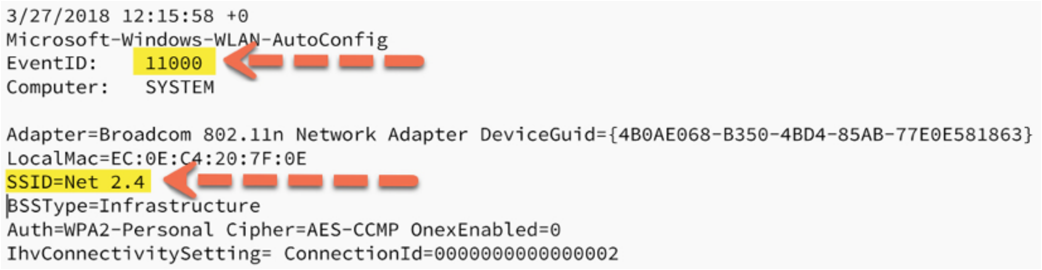
This log contains SSID information, MAC addresses, and the date and timestamps of the connection. The following event ID numbers may be pertinent to your investigation:

- `11000` - Wireless network association
- `8001` - Connected to a wireless network
- `8002` - Failed to connect to wireless network
- `8003` - Disconnected from a wireless network
- `6100` - Network diagnostics (System log)

Note

Everything you ever wanted to know about Microsoft Windows can be found at https://docs.microsoft.com/en-us/.

The following output is consistent with what you will see in the event log:



```
3/27/2018 12:15:58 +0
Microsoft-Windows-WLAN-AutoConfig
EventID:     11000
Computer:    SYSTEM

Adapter=Broadcom 802.11n Network Adapter DeviceGuid={4B0AE068-B350-4BD4-85AB-77E0E581863}
LocalMac=EC:0E:C4:20:7F:0E
SSID=Net 2.4
BSSType=Infrastructure
Auth=WPA2-Personal Cipher=AES-CCMP OnexEnabled=0
IhvConnectivitySetting= ConnectionId=0000000000000002
```

Figure 6.31: Event log for WIFI access

This is an `11000` event ID, which is the start of a wireless connection. So, based on this specific artifact, you can articulate that a connection was made to the wireless network `Net 2.4` on March 27, 2018, at 12:15:58 (GMT) by the computer `SYSTEM`.

If you know where the wireless network `Net 2.4` is located, you can associate this computer with that physical location.

Next, we will discuss the artifacts that allow us to determine whether the user executed a specific program.

Exploring program execution

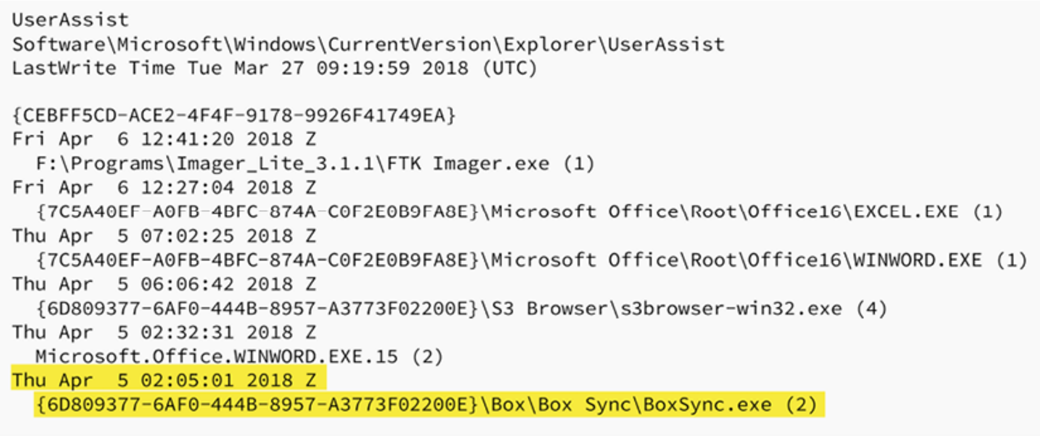
Program execution artifacts indicate programs or applications that were run on the system. The user could cause the execution, or an autostart/run event managed by the system. Some categories overlap with the file knowledge category we discussed earlier in the chapter. I am not going to re-examine those specific artifacts in this section. Just be aware that the artifacts from recent apps, JumpLists, an MRU, and prefetch files will also contain information about program/application activity.

Determining UserAssist

`UserAssist` is a registry key in the user's `NTUSER.DAT` file and can be found at the following path:

```
NTUSER.DAT\Software\Microsoft\Windows\Currentversion\Explorer\Us
```

The key tracks the GUI-based applications that were launched in the system. The system encodes the data in the key with ROT 13 encoding. RegRipper will decode the data automatically. The following represents the output you will see from RegRipper:



```
UserAssist
Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist
LastWrite Time Tue Mar 27 09:19:59 2018 (UTC)

{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}
Fri Apr  6 12:41:20 2018 Z
  F:\Programs\Imager_Lite_3.1.1\FTK Imager.exe (1)
Fri Apr  6 12:27:04 2018 Z
  {7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E}\Microsoft Office\Root\Office16\EXCEL.EXE (1)
Thu Apr  5 07:02:25 2018 Z
  {7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E}\Microsoft Office\Root\Office16\WINWORD.EXE (1)
Thu Apr  5 06:06:42 2018 Z
  {6D809377-6AF0-444B-8957-A3773F02200E}\S3 Browser\s3browser-win32.exe (4)
Thu Apr  5 02:32:31 2018 Z
  Microsoft.Office.WINWORD.EXE.15 (2)
Thu Apr  5 02:05:01 2018 Z
  {6D809377-6AF0-444B-8957-A3773F02200E}\Box\Box Sync\BoxSync.exe (2)
```

Figure 6.32: Contents of NTUSER.DAT\Software\Microsoft\Windows\Currentversion\Explorer\UserAssist

As shown in the preceding output, you can see the date and timestamp of the last execution and the path of the executable. The number in parentheses at the end indicates the number of times the user/system has activated the executable. Next, we will discuss the Shimcache, which also contains information about executed programs.

Exploring the Shimcache

This is the default location of the Shimcache:

```
SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache
```

The Shimcache is used to track compatibility issues with executed programs. Some information that is stored in this cache is as follows:

- File path
- `$Standard` Information Attribute Modify Time
- The update time of the Shimcache

The following represents the output you will see from RegRipper:

Text, letter Description automatically generated

Figure 6.33: Shimcache output

The artifacts found in the Shimcache can provide supporting evidence to the other artifacts found throughout the system, that is, the registry, event logs, filesystem, and so on.

Sometimes, the user will have programs or files contained within a portable device. The next set of artifacts will deal with the use of USB devices.

Understanding USB/attached devices

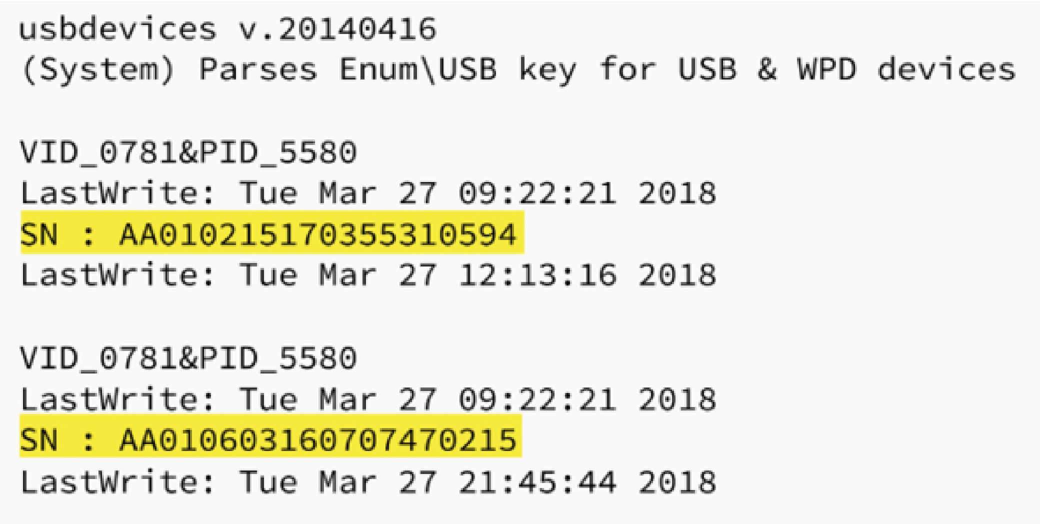
There are several security risks associated with a USB device. They are small, portable, high-capacity storage devices that can be used to exfiltrate data from an organization, or they can be used to deliver malware to an organization to compromise its security protocols.

As a digital forensic investigator, you will want to know whether there were any USB devices attached to the host you are examining. We will now talk about some Windows system artifacts that will allow you to identify USB device usage on the host.

We will now look at the results for two registry keys. The first key can be found at the following path:

```
SYSTEM\CurrentControlSet\Enum\USB
```

This registry key identifies the USB devices attached to the system, as shown in the following output:



```
usbdevices v.20140416
(System) Parses Enum\USB key for USB & WPD devices

VID_0781&PID_5580
LastWrite: Tue Mar 27 09:22:21 2018
SN : AA010215170355310594
LastWrite: Tue Mar 27 12:13:16 2018

VID_0781&PID_5580
LastWrite: Tue Mar 27 09:22:21 2018
SN : AA010603160707470215
LastWrite: Tue Mar 27 21:45:44 2018
```

Figure 6.34: Content of Registry key - SYSTEM\CurrentControlSet\Enum\USB

We can see there were two USB devices attached to the system at different times. We have different volume serial numbers and the last write times from when the system accessed the devices. The volume serial number found in the registry is not the physical device serial number.

> **Note**
>
> Devices that do not have a unique volume serial number will have an `&` as the second character of the volume serial number.

The next registry key you want to look at is the following:

```
SYSTEM\CurrentControlSet\Enum\USBSTOR
```

When we look at the values in `USBSTOR`, we get some additional information about the devices, including the commercial name of the device. We also confirm the serial numbers of the devices with these two entries in the `SYSTEM` hive:

Text Description automatically generated

Figure 6.35: Content of Registry key - SYSTEM\CurrentControlSet\Enum\USBSTOR

In the `MountedDevices` key in the `SYSTEM` hive, which can be found in `SYSTEM\MountedDevices`, we can map the USB device(s) via the serial number to a drive letter on the system:

Graphical user interface, text, application, email Description automatically generated

Figure 6.36: Content of Registry key - SYSTEM\MountedDevices

When we analyze the data, we can see that two USB devices (serial numbers `AA010215170355310594` and `AA010603160707470215`) were connected to the system. One was recognized as the `D:` drive and the second device was recognized as the `E:` drive.

Does the question remain as to which user account was responsible for the USB device usage? To determine the answer to that question, we would have to take the GUID from each of the USB devices and compare them to the user's `NTUSER.DAT` file. The GUIDs we are searching for are `3869c27a-31b8-11e8-9b12-ecf4bb487fed` and `5c3108bb-31c0-11e8-9b10-806e6f6e6963`.

RegRipper will also analyze the `NTUSER.DAT` file and give us the information about the devices that were used and associated with the user's account:

Graphical user interface, text, application Description automatically generated

Figure 6.37: Content of Registry key - Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2

As you can see, we find both GUIDs in the registry entry, which shows when the devices were last mounted. So we can now say that the user used a specific USB device on the system while the `jcloudy` account was logged in.

Summary

In this chapter, we have discussed how to locate artifacts on a Microsoft Windows-based operating system to determine the subject's culpability in the matter being investigated. You have learned about the different categories of artifacts and what actions of the user/system they represent. Using the knowledge you have gained from this chapter will allow you to quickly determine which accounts were active during the timeframe you are investigating and whether the incident involved a removable storage device. In addition, you have learned about the artifacts to analyze in determining whether a user had knowledge of a specific file or application. Finally, we have used several commercial and open-source forensic tools to access the artifacts. As a result, you now know how to find and analyze digital evidence found on a Microsoft Windows-based operating system.

The next chapter will deal with memory forensics.

Questions

1. Where would you find the registry files?

    a. `%SystemRoot%\System32\Config`
    b. `%SystemRoot%\System32`
    c. `%SystemRoot%\\Config\System32`
    d. `%SystemRoot%\System64\Config`

2. When examining log files, which event ID identifies a successful logon?

    a. 4624
    b. 4625
    c. 4672
    d. 4642

3. A thumbcache is a _______________.

    a. Database of toenail images

    b. Database of thumbnail images
    c. Database of deleted thumbnail images
    d. Database of deleted images

4. The user can use Internet Explorer/Edge to view files.
    a. True
    b. False
5. Which of the following will you find in a link (LNK) file?
    a. Volume serial number
    b. Router name
    c. Date of deletion
    d. Volume details
6. Which of the following Microsoft Windows operating systems uses JumpLists?
    a. Windows 98
    b. Windows ME
    c. Windows 7
    d. Windows 2000
7. In which registry hive would we find artifacts relating to USB devices?
    a. NT USER.DAT
    b. SYSTEM
    c. SOFTWARE
    d. SECURITY

The answers can be found at the rear of the book under *Assessments*.

Further reading

Refer to the following links for more information on topics covered in this chapter:

- Altheide, C., Carvey, H. A., and Davidson, R. (2011). *Digital Forensics with Open Source Tools*. Amsterdam: Elsevier/Syngress (available at `https://www.amazon.com/Digital-Forensics-Open-Source-Tools/dp/1597495867`)
- Carvey, H. A. (2005). *Windows forensics and incident recovery*. Boston: Addison-Wesley (available at `https://www.amazon.com/Windows-Forensics-Incident-Recovery-Harlan/dp/0321200985`)
- Bunting, S. (2012). *EnCase Computer Forensics: The Official EnCE: EnCase Certified Examiner; Study Guide*. Indianapolis, IN: Wiley (available at `https://www.amazon.com/EnCase-Computer-Forensics-Official-EnCE/dp/0470901063`)

Exercise

Data set

`Chapter 6 Owl Exercise.e01`

Software needed

Autopsy - `https://www.autopsy.com/`

Scenario

In a jurisdiction where owls are illegal to trade and buy, two users are discussing the illegal trade of owls. A computer is taken into evidence belonging to a user who is attempting to purchase owls illegally. It has been requested that you conduct an analysis of the digital evidence. A forensic image has been obtained and is ready for you. You may use Autopsy or any other tool.

Some artifacts you may want to look for include:

- Web searches
- Shopping searches
- Chat clients
- Email
- Documents
- Social networks
- OS artifacts
- LNK files
- Recycle Bin
- Shellbag

Potential keywords:

- Owl
- Owlet
- Feathers
- Eggs
- Crossbreeding
- Nocturnal

- Nest
- Hoot
- Conservation
- Wingspan

Join our community on Discord

Join our community's Discord space for discussions with the author and other readers:

https://packt.link/CyberSec