Cyber Dumpster-Diving:

$Recycle.Bin Forensics for Windows 7 and Windows Vista

Timothy R. Leschke
Forensic Computer Engineer
U.S. Department of Defense Cyber Crime Institute
911 Elkridge Landing Road, Suite 450
Linthicum, MD 21090
Timothy.Leschke.ctr@dc3.mil

ABSTRACT

Analysis of deleted files often provides useful information for the forensic computer examiner. Knowing where to find the deleted files, and how to interpret the metadata associated with the file's deletion, make up the cornerstone of a successful forensic computer examination. Much like an office trash-can, the Microsoft Windows Recycle Bin is a temporary holding container for files that have been recently discarded (deleted) by the user. Microsoft first introduced the Recycle Bin with its Windows 95 Operating System (released in 1995). This original Recycle Bin implementation was modified for the implementations of the Windows XP Operating System (released in 2001), the Windows Vista Operating System (released in 2007) and the Windows 7 Operating System (released in 2009). Although the Windows XP Recycle Bin is well understood by the forensic examiner community, the Recycle Bins found in Windows Vista and Windows 7 are generally, significantly less understood.

In this paper, The author compares and contrasts the similarities and differences of the Recycle Bin of the Windows Vista and Windows 7 Operating Systems, and the Recycle Bin of the Windows XP Operating System. In this investigation, the author points-out the details of each implementation that are of interest for the forensic computer examiner.

1. INTRODUCTION

"Dumpster diving" - the act of rummaging through a person's garbage while looking for evidence of a crime - is a technique utilized by many successful law enforcement officers. The forensic examination of a Microsoft Windows Recycle Bin is the cyber equivalent of this activity. The Recycle Bin remains a good source of evidence because it often contains the most recently discarded files, and these files remain resident until either the user "empties" the Recycle Bin, or the files

are automatically deleted by the Operating System as it makes room for new trash.

Microsoft first introduced the Recycle Bin in 1995 with the Windows 95 Operating System. The release of the Windows 7 Operating System in 2009 and the release of the Windows Vista Operating System in 2007 reflect some significant changes in the implementation of the original Recycle Bin technology. Understanding the new implementation of this technology is essential for the modern forensic computer examiner.

The release of every new version of the Windows Operating System causes many forensic examiners to quickly investigate how the new version is different from its predecessor. The release of Windows 7 is no exception to this rule. For the forensic examiner that is concerned with examining Recycle Bin data, he or she will be pleased to learn that the Windows 7 implementation of the Recycle Bin is almost identical to the implementation found in Windows Vista.

In the following pages, the reader will learn how the Windows 7 and Windows Vista Operating Systems[1] are essentially the same in their implementation of the Recycle Bin. In contrast, the reader will also learn how these two implementations are different that the Recycle Bin that is found in the Windows XP Operating System.

In investigating these changes, cosmetic changes will be highlighted, as well as some changes to the configuration options. Also, changes to the name of the directory in which the deleted files are maintained, and changes to where the deleted file's metadata will be discussed. Next, the difference in the naming conventions of the deleted files will be explored followed by an explanation of how to use WinHex for a bit-level examination of the deleted file's metadata in order to reveal (1) the encoding of the file's size at the time of deletion, (2) the time and date of deletion, and (3) the original file path. There is finally further discussion of the Recycle Bin settings and what happens when a file gets restored.

The goal of this paper is to leave the reader with a better understanding of the Windows 7 and Windows Vista Recycle Bins. This understanding will allow the reader to be more successful with their next examination of a Recycle Bin on one of these Operating Systems. This understanding will also allow the reader to be better prepared for their next experience as a "cyber dumpster-diver."

2. USER INTERFACE

2.1 RECYCLE BIN ICON

With every new release of a Windows Operating System, Microsoft is known for making cosmetic changes to the software's graphical user interface. The implementation of Windows 7 is no exception. However, despite the overall change in appearance of Windows 7 from Windows Vista, the icon used for both of these Operating System's Recycle Bins appears to be unchanged. This is a relief for the forensic examiner that has grown weary of icons that evolve with each new version of the Windows Operating System.

The Recycle Bin icon that is used by both Windows 7 and Windows Vista

[1] Windows 7 "Ultimate" and Windows Vista "Ultimate" are the two versions that are compared.

looks like a very modern, full-sized, cylindrical trash-can. This icon has two versions. The see-through version is used to represent the Recycle Bin when it is completely empty. The other version appears to be the same cylindrical trash-can, but is full of paper, and is used to represent the Recycle Bin when it contains at least one discarded file.

If one compares the Recycle Bin icon that is shared by Windows 7 and Windows Vista, with that used by Windows XP, one notices a significant difference. The Windows XP recycle Bin icon might appear to the reader to look like a much smaller office trash-can, or perhaps an antique bin once used for fireplace kindling. Its shape is more elliptical than cylindrical. This icon is not see-through and only comes in one version. It appears as a solid white container displaying what can be described as a green "recycle logo" (two arrows that curve in a clock-wise circle) on its side.

2.2 RECYCLE BIN MENU

If one right-clicks the Windows 7 Recycle Bin icon, one is presented with the menu choices of "Open", "Empty Recycle Bin", "Create Shortcut", "Rename", and "Properties". This set of menu choices is an improvement from the Vista Recycle Bin menu which includes all of these choices as well as the choices of "Explore" and "Delete". The "Explore" option that is found as part of the Vista Recycle Bin appears to have the same functionality as the "Open" option that is present in both Windows Vista and Windows 7. Eliminating this duplicated function in the implementation of Windows 7 is a welcome improvement. Likewise, not including the "Delete" option in Windows 7 is also a welcome improvement. Anyone that tested the "Delete" option of the Vista Recycle Bin discovered that it is easy to delete the Recycle Bin icon, but somewhat difficult to restore it.[2]

Although most of the Recycle Bin menu options are essentially the same for Windows XP, Windows Vista, and Windows 7, the one option that should be mentioned is the "Properties" option. Within Windows 7 and Windows Vista, the "Properties" option appears to be exactly the same. Selecting this menu option presents the user with the cosmetically same interface. This interface provides information about the "Recycle Bin Location" and the "Space Available" for that Recycle Bin.

The Recycle Bin "Properties" interface that both Windows 7 and Windows Vista have in common allows the user to toggle between the two settings. The first setting is one that allows the user to set the "Custom size: Maximum size (MB)". This is the setting that allows the user to set the size of the Recycle Bin in megabytes (MB). Windows XP also allows the user to configure this size, but it does so as a percent of each drive's size, rather than a specific size defined in megabytes.

[2] To restore a deleted Recycle Bin icon in Windows Vista, go to Start→Control Panel → Appearance and Personalization → Personalization and then click "change desktop icons" on the left-hand column. From there just check Recycle Bin and click OK. If you still don't see the recycle bin, right click anywhere in an unused space on your desktop (minimize all windows first) & choose 'Refresh' from the menu that pops up.

Windows 7, Windows Vista, and Windows XP each allow the user to configure the Recycle Bin to 'Do not move files to the Recycle Bin. Remove files immediately when deleted.' Furthermore, each of these three Operating Systems allow the user to configure each Recycle Bin independently. Windows XP is unique, however, in that it allows the user to also "use one setting for all drives." Further discussion of configuring the Recycle Bin settings will occur later in section 5.1, "Setting the Recycle Bin Size."

3. FILE LEVEL ANALYSIS

3.1 RECYCLER vs. $Recycle.Bin

For the forensic examiner, in order to truly understand the differences and similarities between these three Operating Systems, one needs to move past the Recycle Bin icons and the different cosmetic appearances of each interface and investigate the individual file-tree structures which are the foundation of each Recycle Bin. By investigating each Recycle Bin through a command-line interface, the more forensically significant details become apparent. The findings are summarized in Figure 1 and Figure 2 on the next page.

Figure 1 provides a simplified file-level view of the Windows XP Recycle Bin whereas Figure 2 provides a similar view for the Windows 7 and Windows Vista Recycle Bins. Because Windows 7 and Windows Vista appear to share the exact same Recycle Bin file-structure, Figure 2 represents them with the same illustration.

In our example, although the root of each file-tree begins at C:\, we notice the very first folders of each are already different. With the Windows XP implementation, the first folder is named "RECYCLER". With Windows 7 and Windows Vista, the first folder is named "$Recycle.Bin". Each of these folders is normally hidden from the user by the Operating System, so the forensic examiner must remember to unhide these folders before they can be viewed.[3]

Experimentation has shown that if either Operating System is installed on a system with several "drives," a RECYCLER folder for Windows XP, or a $Recycle.Bin folder for Windows 7 or Windows Vista, will be placed on each "drive". A "drive", in this case, refers to a logical partition that is assigned a drive letter by either of these Windows platforms. A logical partition may consist of a single physical partition on a single hard-drive, or a partition that spans several hard-drives.

Furthermore, when installing Windows 7 or Windows Vista with several partitions, it was observed that the name of the $Recycle.Bin folder on the first partition (C:\) was displayed by command-line tools as "$Recycle.Bin" (with upper and lower-case characters). However, the same folder name as found on the other partitions was displayed as "$RECYCLE.BIN" (in all capital letters). An explanation for this anomaly cannot be provided.

[3] If one is using command-line tools for displaying the folders, the command "dir /a" will display all files and folders, including those that are hidden. If one is using Windows Explorer in Classic View, open the Control Panel, double-click on the Folder Options icon, select the "View" tab, and under the folder that states "Hidden Files and Folders", select the option to "show hidden files and folders".

Windows XP Recycle Bin File Structure

```
C:\RECYCLER\
        S-1-5-21-51003140-4199384537-3980697693-500
        S-1-5-21-3345512350-4226073239-312180513-1000
                DC1.txt
                INFO2
        S-1-5-21-3345512350-4226073239-312180513-1001
                DC2.pf
                DC3.pf
                INFO2
```

Figure 1

Windows 7 and Windows Vista Recycle Bin File Structure

```
C:\$Recycle.Bin\
        S-1-5-21-51003140-4199384537-3980697693-500
        S-1-5-21-3345512350-4226073239-312180513-1000
                $IPTEYOA.txt
                $RPTEYOA.txt
        S-1-5-21-3345512350-4226073239-312180513-1001
                $IGDRVPB.pf
                $IW1EQ3V.pf
                $RGDRVPB.pf
                $RW1EQ3V.pf
```

Figure 2

3.2 THE SID FOLDERS

Figures 1 and 2 show that if one keeps traversing the file-tree, inside either the "RECYCLER" folder in Windows XP or the "$Recycle.Bin" folder in either Windows 7 or Windows Vista, there are several sub-folders that contain the actual discarded files of the users, along with the associated meta-data. Each of these sub-folders will be referring to as a Recycle Bin, and they will be distinguished from each other based on the user they belong to. Before looking at the contents of these Recycle Bin folders, the naming conventions of these folders will be explored.

In Figures 1 and 2, the first of these Recycle Bin folders is named S-1-5-21-51003140-4199384537-3980697693-500. This folder's naming convention is the same for Windows 7, Windows Vista, and Windows XP. This naming convention is based on the Security Identifier (SID) of the user.

According to the Microsoft Developer Network (2009), the SID is an alpha-numeric string that is used by Windows to uniquely identify an object - like a *user* or a *group*. Experimentation has revealed that, although a user profile may have been created, the SID does not exist until the user logs-on for the first time.

In this example SID, "S" means the string is a Security Identifier. "1" refers to the Revision Level. (This value has always been 1). "5" is the identifier for the Authority Level or "Identifier-Authority" as Microsoft calls it. In our case, "5" refers to NT authority. The remaining part of the string, from "21-51003140-…" up-to, but not including, the "500" at the end of the string, is the Domain or Local Computer Identifier (or one or more sub-authority values). This is a variable number that identifies which computer (or network) created the number. The "500" at the end is known as the Relative ID, and in this case, "500" means the user is a system administrator. By default, only one user account is allowed to be the "system administrator" and is given full control over the computer. A user or group that was not created by default will have a Relative ID of 1000 or greater. We see in Figures 1 and 2 that there are two folders that were not created by default (they end in "1000" and "1001").

As a forensic examiner, it is often important to be able to match a SID with a specific user. Although there are many ways to find a user's SID, one way to do so on a Windows 7 or Windows Vista platform is to left-click on the "perl", which is found on the desktop's bottom tool-bar, and type "RegEdit" in the "Start Search" window. This will open the Registry Editor interface. Then, navigate to HKEY_LOCAL_MACHINE \SOFTWARE\Microsoft\Windows NT\ CurrentVersion\ProfileList. While in the ProfileList folder, click-on the folder with the name that corresponds to the SID of the user that one is trying to look-up. In the window on the right-side of the Registry Editor interface, there will be a list of documents, one of which will be named "ProfileImagePath". The corresponding information under the "Data" column heading will be the file path for that user's profile. The name of the last folder in that path will be the user name associated with the SID.

3.3 EACH USER HAS A PRIVATE BIN

Again, looking at Figure 1 for Windows XP, and Figure 2 for Windows 7 and Windows Vista, one sees that each user (or group) has an individual folder that is associated with their SID. In this example, these are the folders that have names that end in '500", "1000" and "1001". For the purpose of the experiment, each of these folders has been called a Recycle Bin. Conceptually, this is like each user has a private Recycle Bin. This is because when a user deletes a file, the file appears in one's own SID folder. Files that are deleted by different users are not placed in the same SID folder. Thus, there is no co-mingling of deleted files or folders. Furthermore, a user (with the exception of the administrator) cannot even "see" (i.e. read) the deleted items that are in another person's SID folder. If a user without administrator rights attempts to navigate to another person's confidential trash folder, they will receive an "access denied" message. However, a user with full administrator

rights can freely access another user's Recycle Bin, read their files, create new files, and delete existing files.

Although not reflected in Figures 1 and 2, a user will have a private Recycle Bin on each "drive" (a logical partition assigned a drive letter by the Operating System). This means, for example, that if there are three users and four drives, there will be four folders named $Recycle.Bin (one on each drive) and within each of these $Recycle.Bin folders will be three sub-folders with names that correspond to the SID of each of the three users. So, in a sense, there will be twelve Recycle Bins (four bins for each of the three users).

4. METADATA ANALYSIS

4.1 "INFO2" IS NOT USED

Figure 1 shows that for the Windows XP Recycle Bin, in the folder that ends with "1000" there are two files; DC1.txt and INFO2. The examiner that is familiar with Windows XP forensics will recall that these two files contain the information about a deleted file. The file named DC1.txt contains the actual content (text, in this case) of the deleted file, whereas the INFO2 file contains the metadata (file size, time-of-deletion, etc.).

The actual name of the DC1.txt file is significant and it follows the following rules: According to Microsoft Help and Support (2009), the first letter "D" stands for "drive". "C" refers to the identifier of the drive (in Figure 1, we are dealing with "Drive C:\"). "1" is the unique identifier of the file. In this example, this file was the first to be sent to the recycle bin. Subsequent deleted files on the C:\ drive would be named DC1, DC2, DC3, etc. The extension "txt" simply refers to the file extension that was on the file that was deleted.

In addition to storing the actual content of the deleted file, Windows XP also stores the metadata associated with the file's deletion – such as the file-name, file-path, file-size, and time-of-deletion. All of this metadata is stored in the file named INFO2. (According to Kleiman (2007) and others, the name INFO was used for the metadata file that was found in a Windows operating systems prior to Windows 95 and for those operating systems not installed on an NTFS file system.) There is only one INFO2 file for each user's Recycle Bin, where all of the metadata for all of the files/folders that are found in that Recycle Bin is stored. This is a different implementation than that of Windows 7 and Windows Vista.

When looking at Figure 2, for the Windows 7 and Windows Vista file-tree, within the folder that ends with "1000", there are two files; $IPTEYOA.txt and $RPTEYOA.txt. Like the DC1.txt file of our Windows XP example, the $RPETEYOA file contains the content of the file that was deleted by the user that has the SID that also ends in "1000". The name of the file appears to follow the naming convention $R<FileID>.<ext> (this file will be referred to as a "$R-file"). A search of the available literature did not reveal why this naming convention is used, however logical guesses may suggest "R" stands for "Recycle" or "Restore".

The <FileID> portion of the file name is a six-character alpha-numeric string that is most likely intended to uniquely identify this file. Again, a search of the available literature did not reveal more about this six character string. Unlike the Windows XP

implementation that uses an incremental naming convention (DC1, DC2, DC3, etc.), the naming convention for this Windows 7 and Windows Vista file does not seem to follow a noticeable pattern, except to say that it matches the convention used by the $I<fileID>.txt (or "$I-file") found right above it.

Right above the file named $RPTEYOA.txt in Figure 2 is a file named $IPTEYOA.txt. With the exception of the second character ("I"), this file's name is identical to the file below it that starts with "$R". The file $IPTEYOA.txt contains the metadata about the deleted file, such as the file-path, file size, and time the file was deleted. This file extension (.txt) is the same file extension of the file that was deleted. (If a directory/folder is deleted, no file extension is present). Again, the naming convention of this file is not known for certain, but perhaps the leading $I stands for "Information" – as in 'information about the deleted file'.

One way to view the metadata that is contained in the $I<fileID>.<ext> file is with a software tool called WinHex[4]. When using WinHex (v. 10.54) to open the file $IPTEYOA.txt, one will notice several sets of "hexadecimal" values. According to Mitchell Machor (2009), at offset 0x8 is a 64 bit integer that represents the file size. At offset 0x10 is a 128 bit integer that represents the time the file was deleted. At offset 0x18 is the original file-path which includes the file-name.

4.2 INTERPRETING METADATA

Suppose, for example, that one used WinHex to open the file $IPTEYOA. At offset 0x8, one will see the 64 bit hexadecimal value "6A F7 0F 00", which represents the file size. Because we are working with a PC that uses Intel Hardware, this number is in Little Endian format. (Motorola/Macintosh hardware uses the Big Endian format.) "Little Endian" format simply means the lower-ordered byte of the number is stored in the memory at the lowest address. Or, in other words, the little end is read first. However, when one reads English, one reads the larger order first (or, what is known as Big Endian). For example, one reads the number "2008" as two thousands and eight ones, with "thousands" being the larger order and "ones" being the lower order. Therefore, to convert "6A F7 0F 00" from Little Endian to Big Endian format, one needs to read the 2-digit integers from right to left. In doing so, 6A F7 0F 00 is read as

00 0F F7 6A

Converting this hexadecimal value to a decimal (using a conversion calculator)[5] one gets 1,046,378 bytes (or about 1 MB), which represents the actual size of the file that was deleted.

At offset 0x10 is the hexadecimal value

D0 DD 76 3C 2B A9 C8 01

Machor explains that this is the time at which the file was deleted (represented as an offset from January 1,

[4] WinHex is a tool provided by X-Ways Software Technology AG, Carl-Diem-Str. 32, 32257 Bünde (Germany), phone +49 221-420 486 5, http://www.x-ways.net

[5] See http://www.tonymarston.net/php-mysql/converter.php

1601, and expressed in 100 nanoseconds).[6] Since this is a number in Little Endian format, one converts it to Big Endian and gets this result

01 C8 A9 2B 3C 76 DD D0

Converting this hexadecimal value to a decimal value (again, by using a conversion calculator) we get the number 128,538,592,543,170,000. Machor states that this number is the time the file was deleted, but expressed as the number of 100 nano-seconds from January 1, 1601. To convert this number to a more usable size, one multiplies the number by 100 (to convert it from 100 nano-seconds to nano-seconds) and then divides it by 1,000,000,000 to convert it from nano-seconds to seconds.

To calculate the exact time and date at which the file was deleted, one just needs to add the result of the calculation, 12,853,859,254.3170 seconds, to January 1, 1601. Fortunately, the WinHex tool does this calculation. The resulting date and time is 4/28/2008, 12:27:34 (in 24 hour notation), expressed in Universal Coordinated Time (UTC). To convert this time to the time-zone on the East Coast of the United States (during Daylight Savings Time), one subtracts 5 hours for the time-zone difference and then adds 1 hour for the Daylight Savings Time conversion. The final result is 4/28/2008 8:27:34, which is precisely when the file was deleted.[7]

The experienced forensic examiner knows to view every time-stamp critically, including this one. Experimentation has shown that the recorded time of the file's deletion can be manipulated by simply changing the operating system's clock (the displayed time) prior to the file being deleted. The deleted file gets it's time-stamp from the displayed clock, and if this clock time is incorrect, so too will be the recorded deletion time of the file. Furthermore, anyone that is skilled with a bit manipulation tool (like WinHex) can directly manipulate the deletion time-stamp of perhaps any file or folder. Therefore, although the Windows 7 and Windows Vista Recycle Bins record the time and date at which a file was deleted, the forensic examiner must view this time-stamp critically, keeping in mind that each time-stamp is subject to manipulation.

Mitchell Machor (2009) states that at offset 0x18 is the start of the file-path. In this case, we have what appears to be the following "hexadecimal" values;

00 43 00 3A 00 5C 00 55 00 73 00 65 00 72 00 73 00 5C 00 55 00 73 00 65 00 72 00 20 00 31 00 5C 00 44 00 65 00 73 00 6B 00 74 00 6F 00 70 00 5C 00 31 00 20 00 4D 00 42 00 2E 00 74 00 78 00 74

If these values are treated as "hexadecimal" values, they can be translated into the following characters, with a space between each one of them;

C : \ U s e r s \ U s e r 1 \ De s k t o p \ 1 M B . t x t

If one ignores the spaces between the characters, one then has the file-path

[6] Mitchell Machor states it is an offset from January 1, 1601. It is through inspection that we see that the time is expressed in nano-seconds.

[7] The example used is from a Windows Vista Operating System, but experimentation has shown that Windows 7 will yield the same result.

of "C:\Users\User1\Desktop" as well as the file name of "1MB.txt".

A further study of these values however, reveals they are not actually hexadecimal. They are Unicode. Unicode is a character encoding system that allows for many more character possibilities than that of hexadecimal. When trying to encode the English Language character set – Basic Latin – the number of possible characters is handled just fine by a set of pairs of alpha-numeric characters that make up a hexadecimal value. However, if one wants to be able to encode many more of the world's known characters - like those of the Armenian, Coptic, Cyrillic, Greek, and Hebrew alphabets to name a few - then one needs a much larger encoding set, and each character has to be encoded with four alpha-numeric characters rather than just the two found in the hexadecimal encoding.

Fortunately, the encoding of Basic Latin characters in hexadecimal and Unicode are very similar. For example, the character "A" is encoded as "43" in hexadecimal and as "0043" in Unicode, "B" is encoded as "44" in hexadecimal and as "0044" in Unicode, etc. Therefore, it is easy to see how one could easily misread Unicode as a set of hexadecimal values with spaces ("00") between them.

Returning to the alpha-numeric values found at offset 0x18, one can now interpret the codes as 0043, 003A, 005C, 0055, etc. These codes, when translated as Unicode, are the encoding for "C:\Users\User1\Desktop\1MB.txt", which, as we stated earlier, is the original path and file name of the deleted file.

5. CONFIGURATION ISSUES

5.1 SETTING THE RECYCLE BIN SIZE

Windows 7, Windows Vista, and Windows XP each allow the user to set the *maximum* size of the Recycle Bin, but each has a unique way of accomplishing this. Windows XP only allows the user to set the maximum size of the Recycle Bin in terms of a percent of the size of the volume upon which the Recycle Bin resides. Windows 7 and Windows Vista both allow this same feature, but they also allow the user to set the maximum size of the Recycle bin in terms of a specific megabyte (MB) value. The advantage that Windows 7 and Windows Vista have with this added feature, is that this setting is now more precise.

For each of the three Operating Systems, to set the *maximum* size of the Recycle Bin is to merely set the upper-limit of how big the Recycle Bin can be. This is not a setting for the *actual size* of the Recycle Bin as one may assume. For example, experimentation was done with a Windows 7 volume that is 15.9 GB in size.[8] The "Recycle Bin Properties" interface allows the user to set the maximum Recycle Bin size to 16,382 MB, which is equal to the volume's size of about 15.9 GB. This setting is a theoretical maximum, not an actual maximum. Because this volume also contains Operating System files and other user data, about 6.9 GB of the 15.9 GB volume is already occupied. Therefore, the volume only has about 9 GB of space for the Recycle Bin to

[8] Similar experimentation was done with a Windows Vista volume. Windows XP does not allow for similar experimentation.

occupy. So in this case, despite the fact that the user is able to set the maximum size of the Recycle Bin to be 15.9 GB, the actual room available for the Recycle Bin is only 9 GB. Thus, the forensic examiner needs to keep in mind that the setting associated with the maximum size of the Recycle Bin is a theoretical upper limit of its size. The actual maximum size of the Recycle Bin is further limited by the amount of actual space that is available on the volume on which the Recycle Bin is attempting to reside.[9]

Further experiments with Windows 7 and Windows Vista show that if the user attempts to set the maximum Recycle Bin size to a value that is greater than the actual size of the volume, the properties interface will reject this setting and default to the maximum size allowable (a size equal to the size of the volume).

Experimentation was also done with setting the maximum size of the Recycle Bin to zero. For a Windows XP Recycle Bin, if one sets the parameter so that the maximum size is zero percent of the drive, no files are allowed into the Recycle Bin. Experiments with deleting files of 1 byte showed that even files this small were not allowed into the Windows XP Recycle Bin that was configured with a zero percent maximum size.

On the other hand, when experiments were done with Windows 7 and Windows Vista Recycle Bins, a different result was achieved. When the maximum Recycle Bin size to 0 MB was set, the settings interface rejected this setting and replaces it with a 1 MB setting. This setting was tested by sending a 1 byte file to the Recycle Bin and it was found that the Recycle Bin accepted it. Further experimentation found that files that were less than 1 MB were accepted by the Recycle Bin whereas files that were greater than 1 MB were rejected by this Recycle Bin. Experimentation also showed that a file that is exactly 1 MB (1,048,576 bytes) was accepted, whereas a file that was just one byte larger was rejected by the Recycle Bin.[10]

Upon installation of all three Operating Systems, it was noticed that each sets the maximum Recycle Bin size to be about 10% of the size of the volume by default.

Regardless of how the size is set, the rules that govern what size files will be accepted by the Recycle Bin are very strict. A file that is as little as 1 bit larger than the Recycle Bin maximum size parameter will not be accepted. If a user deletes a file that is too big for the Recycle Bin, the user will be prompted to confirm that the file or folder will be deleted "permanently" and not placed in the Recycle Bin. Furthermore, the size that is examined is the size of the file content (the $R-file), not the "size on disk" – which can be a little larger than the size of the content.

Although the Recycle Bin has been described as being an individual folder (named after the SID of the user),

[9] Experimentation with a Windows 7 Recycle Bin showed that if the space occupied by user files within a particular volume grew enough as to force the Recycle Bin to shrink in size below the size of its current contents, some files that are in the Recycle Bin will get permanently removed – a process that follows the usual first-in-first-out algorithm.

[10] The size used was the actual size, not the "size on disk", which can be slightly larger because the Operating System appears to allocate space in 4,096 byte increments.

setting the size of the Recycle Bin does not set the size of this folder as one might think. The folder that contains the deleted file holds two types of files – the $R-file, which is a copy of the deleted file's contents, and the $I-file, which is the metadata about the deleted file. When a Recycle Bin size is configured, the value that is set is the maximum combined size of all of the $R-files (not including the $I-files). The combination of the sizes of the $R-files and $I-files may be larger than the configured size of the Recycle Bin, but the combined size of the $R-files is limited by the configured Recycle Bin size.

Lastly, with Windows 7 and Windows Vista, each user is allowed to have their own Recycle Bin that they can configure independently of the other users of the computer. This is different than Windows XP, which allows the Recycle Bin to be configured only by the Administrator (or a user with administrator privileges). With Windows 7 and Windows Vista, the user is allowed more control over one's individual computing environment.

5.2 TURNING OFF RECYCLING

One of the features that Windows XP implemented that Windows 7 and Windows Vista retained, is the ability to configure the Recycle Bin to 'Do not move files to the recycle bin. Remove files immediately when deleted'.[11] When a Recycle Bin is configured this way and a file is deleted, the corresponding $R-file and $I-file are not created and placed in the Recycle Bin folder of the Windows 7 or Windows Vista Operating System. Instead, the Operating System handles the file in what is assumed to be the traditional way, presumably by changing the file allocation table so as to no longer protect the memory location that holds the newly deleted file. This memory location will then be free to be re-used and its contents will be overwritten as soon as the Operating System needs to use that space.

Of course, when a file is "deleted", the file might still be recoverable by a forensic examiner as long as the memory location that held the file has not been re-used. This is accomplished through the use of special data recovery or data-carving tools.

6. FURTHER CONSIDERATIONS

6.1 EXTRA "TRASH" IN THE BIN

If one plays with a Windows 7 or a Windows Vista Recycle Bin long enough, they will notice "extra trash" in the bin. This extra trash is the "." (dot-directory), the ".." (dot-dot-directory), and the "desktop.ini" file.

Within each Windows 7 or Windows Vista user's Recycle Bin, if one has displayed all of the "hidden" directories, one will likely find a directory named "." (dot). Stepping-into this directory one finds that it leads nowhere. It is simply a pointer back to the directory where they were before stepping-into the dot-directory. Stephen Chen (2009) states this "." (dot) is a relative path name that refers to the current directory.[12]

[11] Whereas the Windows XP Operating System uses the wording "Do not", Windows 7 and Windows Vista use "Don't".

[12] The use of the dot and dot-dot symbols for the current and parent directory might be a technique that was inherited from the Disk Operating System (DOS).

Stepping into the ".." (dot-dot) directory, one will find they have moved to the parent directory of the directory they were in previously. This dot-dot is also a relative path name and it refers to the parent directory of the current directory. Windows 7 and Windows Vista appear to add the dot-directory and dot-dot-directory to every user's Recycle Bin by default.

Another surprise that the forensic examiner may find in a Recycle Bin is a file named "desktop.ini." Ieinfosite (2009) states the desktop.ini file is a hidden Windows system file that provides information to Windows Explorer about how to display the contents of a folder. For example, if a particular thumbnail image needs to be displayed when a file icon is displayed inside a folder, this information is provided by the desktop.ini file.

Experimenting with the desktop.ini file is difficult, primarily because it is difficult to observe consistent behavior with this file's creation. It is not clear if the desktop.ini file is created when the Operating System is first installed, or if the file is created on-the-fly when a user first opens the Recycle Bin by double-clicking on it. Or, perhaps there is another event that triggers its creation. For our discussion, the forensic examiner need only be aware that the desktop.ini file is a hidden system file that the Operating System uses to help display the contents of a folder.

6.2 RESTORING A FILE

When a file is restored from a Windows 7 or Windows Vista Recycle Bin, the $R-file and $I-file are used to recreate a new copy of the original file and place it where it was prior to deletion. If a file that is deleted was originally in a folder that no longer exists, Windows 7 and Windows Vista will each re-create the folder (or folders) that would be necessary to hold the file. If a folder has to be re-created, an entirely new folder is created and the original deleted folder will not be removed from the Recycle Bin.

When a file is sent to a Recycle Bin, it has a "deleted" time-stamp and a "created" time-stamp value, which are recorded in the $I-file. When this file gets restored, it loses the "deleted" time-stamp, but gains a "modified" and "accessed" time-stamp. Each of the "created", "modified", and "accessed" time-stamps of the newly restored file are set to the "created" time-stamp that was associated with the file before it entered the Recycle Bin.

When a folder is sent to the Recycle Bin, it too has the "deleted" and "created" time-stamp, but when it is restored, it only retains the "created" time-stamp, and never gains the "modified" or "accessed" time-stamp, unlike what happens with a file.

When a file is restored, the corresponding $I-file and $R-file are erased from the Recycle Bin folder. If the restored file is deleted again, a new $I-file and $R-file are generated, and these files have names that are different than the previous $I-file and $R-file names.

When a folder that contains a file is sent to the Recycle Bin, each item - the folder and the file - is treated as a separate deletion. There will be an $I-file and $R-file for the folder and there will also be a $I-file and $R-file for each file that was in the deleted folder. When the deleted folder is restored, so too is the folder's contents. Furthermore, all of

the records of these deletions are removed from the Recycle Bin.

6.3 RECYCLE BINS AS QUEUES

Experimentation has shown that the Windows 7 and Windows Vista Recycle Bins each behave like a first-in-first-out queue. That is to say, the first item into the Recycle Bin is the first item to be taken out and permanently deleted. For example, if a Recycle Bin is full and a new file is added to the Recycle Bin (by having been deleted), the Recycle Bin will remove (delete) one or more of the first files that were sent to the Recycle Bin in order to make room for the new incoming file. Furthermore, it will permanently delete the file that was the first to be added to the Recycle Bin, much like how items are added and removed from a queue according to a first-in-first-out algorithm.

6.4 SHADOW COPIES

A thorough investigation of Recycle Bin data cannot be considered complete until one has also investigated the Recycle Bin data that can be archived by the Volume Shadow Copy Service. The Volume Shadow Copy Service is a Windows function that makes copies of key data so that the operating system can easily recover from an event that has placed the system into an unstable state. This Windows function is also what makes it possible for a user to view a "previous version" of a document.

One of the key data sets that can be archived by the Volume Shadow Copy Service is the user's Recycle Bin data. The process of copying the data is known as taking a "volume snapshot". The data that is copied is stored in what is called a "shadow volume". Thus, any forensic examiner that is conducting a thorough examination of Recycle Bin data must also examine the "shadow volume".

A continuation of this investigation into the forensic examination of Recycle Bin data that is found within a "shadow volume" is provided in the paper titled "Shadow Volume Trash: $Recycle.Bin Forensics for Windows 7 and Windows Vista Shadow Volumes".[13]

7. CONCLUSION

In conclusion, the implementation of the Recycle Bin in Windows 7 and Windows Vista is slightly different than that of Windows XP. The introduction of the $Recycle.Bin folder, $R-file, and $I-file in both Windows 7 and Windows Vista have replaced the RECYCLER folder, DC<#> file, and INFO2 file of Windows XP.

Although Windows 7 and Windows Vista use the SID as part of their Recycle Bin folder naming convention just like Windows XP does, Windows 7 and Windows Vista are different because each creates a private Recycle Bin, on each partition (one for each user). Furthermore, this private Recycle Bin can be configured to be a specific size, rather than only a %-size of the drive-size, which is all that Windows XP allows.

The WinHex tool was also examined as it relates to viewing the metadata and file content that is found in the $I-file and $R-file for each deletion. This provides the forensic examiner with a way to determine the time the file (or folder) was deleted, the size of the file, and the file's original name & path.

[13] Presented at the U.S. Department of Defense Cyber Crime Conference 2010.

The setting of the Recycle Bin size was also explored, as well as how turning off the recycling option affects its behavior. The process of restoring files and folders was explained. It was also explained that the Recycle Bin behaves much like a queue – where the first file in is also the first one to be moved out and permanently deleted.

A closer look was also given to hidden files that are found in the Recycle Bin folder and it was noticed that some extra "trash" is already present by default, namely the dot-folder, dot-dot-folder, and the desktop.ini file.

How a file or folder within a Windows 7 or Windows Vista Recycle Bin gets restored was explained. The reader saw how the $R-file contains the original file content, whereas the $I-file provides the metadata that is needed to be able to place the restored file back in its original location. In instances in which a file is being restored to a folder that no longer exists, it was explained that the required folder is also recreated as the file gets restored.

Lastly, it was explained how the Volume Shadow Copy Service found in Windows 7 and Windows Vista may make backup copies of Recycle Bin data. This means that although a computer user may believe they have permanently deleted certain files, a skilled forensic examiner may be able to recover these files from the data that has been archived by the Volume Shadow Copy Service. shadow volume. This is a topic that will be explained in our next paper.[14]

In conclusion, it is vital to remember the importance of examining the Recycle Bin as part of a thorough forensic computer examination. The evidence that the Recycle Bin can provide may be the key piece of evidence that law enforcement needs for a successful prosecution. Being able to properly interpret the evidence found in the Recycle Bin is sometimes essential for a forensic computer examiner to be able to conduct a successful exam. After reviewing the guidance provided in this paper, the forensic computer examiner should now be better prepared to complete a successful forensic examination of a Windows 7 or a Windows Vista Recycle Bin. Furthermore, this guidance should help the forensic examiner hone his skills as a "cyber dumpster-diver."

[14] "Shadow Volume Trash: $Recycle.Bin Forensics for Windows 7 and Windows Vista Shadow Volumes."[14] Presented at the Department of Defense Cyber Crime Conference 2010.

References:

Chen, Stephen Y. (2009). "Introduction to file systems".
https://www.atkinson.yotku.ca/~sychen/ITEC1620/UNIX/unixhelp.html
(accessed September 2, 2009).

Ieinfosite (2009). "What does desktop.ini do".
http://www.ieinfosite.co.uk/tip_view.asp?id=29
(accessed September 2, 2009).

Kleiman (2007). "The Official CHFI Study Guide (Exam 312-49)". Published by Amorette Pederson. Copyright by Elsevier, Inc.

Machor, Mitchell (2008). "The Forensic Analysis of the Microsoft Windows Vista Recycle Bin." http://www.forensicfocus.com/downloads/forensic-analysis-vista-recycle-bin.pdf
(accessed September 2, 2009).

Microsoft Developer Network (2009). "SID Components."
http://msdn.microsoft.com/en-us/library/aa379597(US.85).aspx
(accessed September 1, 2009).

Microsoft Help and Support (2009). "How the Recycle Bin Stores Files".
http://support.microsoft.com/kb/136517
(accessed September 1, 2009).