Shadow Volume Trash:

$Recycle.Bin Forensics for Windows 7 and Windows Vista Shadow Volumes

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

According to Microsoft, over one-third of all data loss is the result of accidental file deletion or modification (Microsoft, 2003). The Volume Shadow Copy Service is a Windows operating system service that archives key data and system settings. This allows Windows 7 and Windows Vista to recover from accidental data deletion and from destabilizing events, such as a virus attack or the incorrect installation of a software or hardware device. This archiving service also makes it possible for a user to view "previous versions" of documents. Because of the amount of data that this service archives, it has been referred to as a gold mine of forensic evidence.

One of the key sets of data that gets copied by the Volume Shadow Copy Service is the user's Recycle Bin data. Recycle Bin data includes records of the most recently discarded ("deleted") files of the user. The process of archiving Recycle Bin data by the Volume Shadow Copy Service is achieved by taking a "volume snapshot." "Volume snapshot" data is stored in what is known as a "shadow volume." Because a "shadow volume" is not located within the traditional file-tree structure of the operating system, the usual methods employed by forensic computer examiners to analyze this data cannot be used. A new approach for examining this data is required.

In the following pages, the author explains how the Volume Shadow Copy Service archives Recycle Bin data into shadow volumes. The use of the "vssadmin list shadows" command is introduced as a way to identify the shadow volumes that exist within an operating system. The author further explains how to create "symbolic links" to access individual shadow volumes. The challenges that a forensic computer examiner faces when attempting a manual examination of a shadow volume are also explained. The author concludes his exposition by suggesting the forensic computer examiner should use the software tool Shadow Miner, a tool that automates the forensic examination of Recycle Bin data that has been archived into a shadow volume.

1. INTRODUCTION

In the previous paper (Leschke, 2010), the implementations of the Recycle Bin as found in the Windows XP, Windows Vista, and Windows 7 operating systems were explored. Drastic changes to the Recycle Bin have occurred as it evolved from Windows XP to Windows Vista, but minimal changes to the Recycle Bin were noticed as it evolved from Windows Vista to Windows 7. Users learned that the implementation of the Recycle Bin in Windows Vista and Windows 7 are almost identical to each other. As the Recycle Bin implementations from these three operating systems were explored, those details that were thought to be of greatest interest to the forensic computer examiner were highlighted.

In the previous paper, an overview of the Volume Shadow Copy Service was given. This Windows service plays a very significant role in both Windows Vista and Windows 7. One of the most interesting aspects of the Volume Shadow Copy Service is its ability to create what is known as a "shadow volume". A shadow volume is a copy of key data that is used by the system to recover from an unstable state, such as after a virus attack or the incorrect installation of a software application or hardware device. Shadow volume data is also used to allow a user to view "previous versions" of a document, and to recover from the accidental modification or loss of data.

In the following pages, the reader will learn, (1) an explanation of the Volume Shadow Copy Service, (2) how to identify the shadow volumes that are present within a Windows Vista or Windows 7 system, (3) how to create symbolic links to each shadow volume, and (4) how to analyze the Recycle Bin data that are found in a shadow volume. The discussion will conclude by bringing to light some of the challenges and impossibilities that a forensic examiner faces when trying to conduct a manual, command-line, examination of shadow volume data. The discussion ends with the reader being encouraged to use Shadow Miner, a software tool that facilitates the examination of shadow volume data

2. VOLUME SHADOW COPY SERVICE

The Volume Shadow Copy Service was introduced in Windows Server 2003 and was known as "Shadow Copies for Shared Folders" (Microsoft, 2003). This technology was not known as the "Volume Shadow Copy Service" until it was included in the Windows Vista operating system, released in 2007. By the time this technology was included in Windows Vista, the Volume Shadow Copy Service evolved into a much more robust application with a much more prominent role.

The Volume Shadow Copy Service allows for the creation of a backup copy of key data. The process of creating the backup copy is known as taking a "volume snapshot" and the actual backup copy of data is known as a "shadow volume". Having a backup copy of key data, i.e. system settings, is useful when trying to recover from a virus attack or the incorrect installation of software that has put the computer into an unstable state. In addition to helping the operating system recover from an unstable state, the Volume Shadow Copy Service also makes it possible to work with a *previous version* of a file and to recover from the accidental modification or deletion of data.

Previously, it was thought that 15% of a volume was set aside by default for storing shadow volumes. However,

experimentation with both Windows 7 and Windows Vista systems has resulted in shadow volumes that are only 2-4% of the volume's size. An explanation for this inconsistency cannot be provided. Regardless of what the default size of a shadow volume is, the amount of space allocated for the storing of shadow volumes can be changed by using the "Vssadmin Resize ShadowStorage" command. (See http://technet.microsoft.com for further details.)

The Volume Shadow Copy Service makes copies of key files and directories at various times; including just prior to installing new software, as well as when a restore-point is established. One of the key directories that gets copied as part of the Volume Shadow Copy Service is the $Recycle.Bin.

When the $Recycle.Bin directory is copied by the Volume Shadow Copy Service, the entire contents of this directory appear to be copied. The data that is copied is stored in a "shadow volume". This "shadow volume" is located on the logical drive volume, in a location that is allocated by the operating system for the storing of backup copies of files.

Even if a Recycle Bin is "emptied" and all of its contents are "permanently deleted", if a shadow copy was made of the Recycle Bin data prior to the Recycle Bin being emptied, then it is still possible to recover Recycle Bin data from the shadow volume. Thus, the forensic examination of Recycle Bin data does not end until the examiner has also examined the backup copy of the data that has been archived into a shadow volume.

As an example of how forensic evidence might be archived by the Volume Shadow Copy Service, consider the following scenario. Suppose there is a malicious user with either a contraband file (child pornography) or a software tool that has been "deleted" (sent to the Recycle Bin) - perhaps as an attempt to hide a crime. Now suppose that the user attaches a new USB thumb drive to the computer. Since this is a new USB thumb drive, the operating system will need to install new drivers to support this USB device. When the new drivers get installed, the Volume Shadow Copy Service will be triggered to create a new shadow volume. When this occurs, the contraband image or software tool that is in the Recycle Bin will be copied to a shadow volume. At this point, if the user were to empty his Recycle Bin with the intention of permanently deleting the discarded contraband image or tool, the user may incorrectly conclude that he has removed the only copy of the file that there is. The user would not be aware of the copy of the file that was archived into the shadow volume. Since the only remaining copy of the evidence has been archived into a shadow volume, being able to access the shadow volume is now a critical step in the successful examination of this evidence.

3. VSSADMIN LIST SHADOWS

The successful examination of a shadow volume begins with being able to access the shadow volume. However, before one can access a shadow volume, one must first find out which shadow volumes are actually present within a system. This is easily accomplished by using the command "vssadmin list shadows". When one types this command at a command-line prompt with Administrator privileges, he or she will see a report that displays the information about the shadow volumes that are found within the system. Figure 1 below is an example of a "vssadmin list shadows"

report that was generated on a Vista machine. Highlighted in red is the name of the one "shadow copy volume" that exists on the target machine. The name "\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1" will be used to identify the shadow volume when creating a symbolic link to this data blob.

A source that explains each line of the "vssadmin list shadows" report could not be found, however the material provided by Microsoft (http://technet.microsoft.com) and some independent experimentation has provided some insight. What follows is conjecture to make sense of this report.

The first line of the "vssadmin list shadows" report is most likely the name of the "service" that provides the shadow volume. A service is simply an application that is capable of generating a shadow volume. A service can be written by Microsoft, an independent software vendor, or an individual user. In this case, the line states "vssadmin 1.1 – Volume Shadow Copy Service administrative command-line tool". This line is immediately followed by the Microsoft copyright marking that designates this as a Microsoft application.

The next line of text states "Contents of shadow copy set ID: {6bb7123a-61c7-4890-984e-a617f9599a37}". This is understood to be the identifier of the shadow volume "copy set". When shadow volumes are created, they are created in a "set". A set can consist of one or more shadow volumes. For example, if a computer has two volumes (i.e. two hard drives), a volume snapshot can be taken of both shadow volumes at the same time. The volume snapshots of each of the two volumes are combined in one set which has its own unique identifier. This identifier is the "shadow copy set ID".

The next line of this report states "Contained 1 shadow copies at creation time: 11/5/2009 10:54:16 AM". This entry in the report is a time-stamp for when the shadow volume was created. It also specifies that there is only one shadow copy in the set. Since there is only one shadow copy in the set, there is also only one shadow copy listed below this line. The listing of the shadow

```
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2005 Microsoft Corp.

Contents of shadow copy set ID: {6bb7123a-61c7-4890-984e-a617f9599a37}
   Contained 1 shadow copies at creation time: 11/5/2009 10:54:16 AM
      Shadow Copy ID: {3b4d5d3d-394a-4fa9-af55-5c8249a9affb}
         Original Volume: (C:)\\?\Volume{1d6e1a07-bf34-11de-b87b-806e6f6e6963}\
         Shadow Copy Volume: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
         Originating Machine: ForensicExam-PC
         Service Machine: ForensicExam-PC
         Provider: 'Microsoft Software Shadow Copy provider 1.0'
         Type: ClientAccessibleWriters
         Attributes: Persistent, Client-accessible, No auto release, Differential, Auto recovered
```

Figure 1

volume begins with "Shadow Copy ID", which is followed by a 32 alpha-numeric unique identifier. Had there been two shadow copies in the set, the report would have stated "Contained 2 shadow copies at creation time…". Below this line there would have been two entries, one for each shadow volume. Experimentation has shown that if a restore point is created within a Vista machine that has two volumes that are included in the restore point, then the "vssadmin list shadows" report will show two shadow volumes, one for each volume. Furthermore, if after the two shadow volumes are created, one of the volumes is removed, the "vssadmin list shadows" report will still state "Contained 2 shadow copies at creation time…". However, since only one shadow volume is currently present, there will only be one entry that states "Shadow Copy ID…". This illustrates how being able to read the "vssadmin list shadows" report might allow the forensic examiner to recognize that one of the original volumes from a Vista system is missing.

In Figure 1, the Original Volume entry is "(C:)\\?\Volume{1d6e1a07-bf34-11de-b87b-806e6f6e6963}\". The "C:" is understood to be the letter assigned by the operating system to the particular volume that is the target of the volume snapshot. The remainder of the entry is most likely the unique identifier for that particular volume. This conclusion is arrived at because the string appears to be a set of hexadecimal values that follow the format of a Microsoft Globally Unique Identifier (GUID).

The next entry in the report is the most import for this investigation. It is the "Shadow Copy Volume" entry and it is the identifier of the shadow volume. This name is important because it is used in conjunction with the "mklink" command to create a symbolic link to the shadow volume. An example of the command is provided in Figure 2 below.

The next two entries in the "vssadmin list shadows" report are the "Originating Machine" and the "Service Machine". A review of the available literature did not reveal what these entries refer to. However, a logical conclusion is that they refer to the "requestor" and the "writer". A requestor is "An application that requests that a volume shadow copy be taken. A backup application is an example" (Microsoft, 2009). The same source states a "writer" is "A component of an application that stores persistent information on one or more volumes that participate in shadow copy synchronization. Typically, this is a database application like SQL Server, or Exchange Server, or a system service like Active Directory." If this is the case, then our example "vssadmin list shadows" report (Figure 1) states the "requestor" and the "writer" are the same, the "Forensic Exam-PC" machine.

Another inference as to what the "Originating Machine" and "Service Machine" refer too, is that they refer to the "source volume" and the "storage volume". According to Microsoft TecNet, the *source volume* is "The volume that contains the data to be shadow copied" (Microsoft, 2009). On the other hand, the *storage volume* is "The volume that holds the shadow copy storage files for the system copy-on-write software provider."

```
mklink /d C:\myShadowVolume1 \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\
```

Figure 2

The next entry in the report is the "provider". The provider is the service that creates the shadow copies. In Figure 1, the provider is listed as 'Microsoft Software Shadow Copy provider 1.0'.

The final two entries in the "vssadmin list shadows" report are the "type" and the "attributes". A review of the available literature did not reveal what these terms refer to. However, based on naming convention and how the terms are used, it seems that "type" refers to the type of "writer" that was used to create the shadow volume. The "attributes", on the other hand, seems to be the attributes for the shadow volume itself. No further information could be found about either of these terms.

4. ACCESSING SHADOW VOLUMES

Now that a way of identifying which shadow volumes are present on a machine has been established, a procedure for accessing these shadow volumes must be explained. A shadow volume cannot be accessed through traditional Windows tools such as Windows Explorer or the command-line prompt. This might be because the shadow volume is not part of the Windows file-tree structure, or at least, it is not part of the file-tree structure that can be accessed directly by the user. If a user wants to gain access to a shadow volume, the user must be able to modify the file-tree structure. This is done by creating a "symbolic link".

Although symbolic links were first seen in the UNIX operating system, they were not made available to the Windows user until the release of Windows Vista in 2007. In order for Vista to support symbolic links, the New Technology File System (NTFS) is required. A symbolic link is a file-system object within the NTFS that points to another file system object. This file-system object appears to the user as a normal file or directory. A symbolic link is essentially a virtual name for a directory path (Arehart, 2009). Symbolic links were designed to provide application compatibility with POSIX operating systems. A user with Administrator privileges can create a symbolic link by using the "mklink" command as shown in Figure 2.

4.1 THE "MKLINK" COMMAND

The mklink command has the following syntax:

mklink [switch] [link] [target]

There are three possible switches that can be used with the mklink command. These switches are "/d", "/h", and "/j". The "/d" switch is the default switch and it is used to create a "directory" symbolic link. A directory symbolic link can link objects from different file systems. This linking method is the preferred method for linking to shadow volumes. The "/h" switch creates what is known as a "hard link", which is a link that can only link objects on similar file systems. The "/j" switch is used for creating a directory "junction." A directory junction can only be used to link to a directory on a local volume.

The "[link]" of the mklink command specifies the name of the symbolic link that is being created. In the example shown in Figure 2, the symbolic link is called "shadowCopy1". This is a name that is chosen by the user.

The "[target]" specifies the relative or absolute path to what the user wants the link to refer to. In this example, the new link to refer to a

shadow volume. If Figure 1 is used as an example of a shadow volume report, the name of the shadow volume that the user wants to link to is found after the string that states "Shadow Copy Volume" entry. In this case, the entry is "\\?GLOBALROOT\Device\HarddiskVolumeShadowCopy1".

If all of these parts of the mklink command are put together, one will get the command that is found in Figure 2. Notice that a back-slash ("\") was added to the end of the command. This additional character is needed for the command to execute properly.

If the command found in Figure 2 is executed properly, (1) from a command-line prompt that has Administrator privileges, (2) from a Windows Vista or Windows 7 operating system that supports the Volume Shadow Copy Service, and (3) on an NTFS file system that supports symbolic links - one should be able to navigate to the C:\ directory and find a new directory named "shadowCopy1". This directory's icon is distinguishable from the other directory icons because it has a small arrow superimposed over the traditional folder icon. This arrow suggests to the user that the associated directory is a "pointer" (a symbolic link) to a different location within that computer system.

5. $RECYCLE BIN ANALYSIS

After the command in Figure 2 is executed, and a symbolic link is created to allow access to the shadow volume, one can access the shadow volume by executing the command "cd C:\shadowCopy1" at a command-line prompt. Once the user is in this directory, the command "dir /a" can be used to display a listing of the contents of that directory. One of the directories that will be displayed is the $Recycle.Bin directory. For a thorough analysis of how to conduct the forensic examination of a Recycle Bin outside of a shadow volume, see Leschke, 2010. However, this current paper will explain how to examine a $Recycle.Bin directory in the context of it being one of perhaps many $Recycle.Bin directories that have been archived into a shadow volume.

5.1 THE QUANTITY OF DUPLICATE DATA

One of the first problems that a forensic examiner faces when he begins the examination of shadow volume data is how to manage the large amount of data that is available. This problem seems to exist because the archiving of shadow volume data appears to multiply the amount of forensic data that is available for investigation. For example, suppose a user has deleted a single file and this file has been sent to the $Recycle.Bin directory. As explained in the previous paper, when this file gets sent to the $Recycle.Bin, it is separated into two files (Leschke, 2010). The $R-file is a copy of the original file's contents and the $I-file is a file that holds the original file's metadata. Thus, the deletion of one file results in there being two files in the $Recycle Bin.

The number of $Recycle Bin files that need to be reviewed by the forensic examiner becomes even greater because the taking of a volume snapshot appears to create duplicate copies of the $Recycle Bin data. For example, suppose a Windows Vista or Windows 7 is configured to create a volume snapshot every evening. For every file that is in the $Recycle Bin, a copy of that file *appears* to be made with every volume snapshot. Therefore, if a file is sent to the $Recycle Bin through deletion, that one

file becomes two files (an $R-file and an $I-file). Furthermore, if the operating system is configured to take a volume snapshot every evening, the $R-file and $I-file of a deleted file *appears* to be copied every night. Thus, after 30 nights of making copies by way of the volume snapshot process, the original deleted file can quickly grow into what appears to be 60 files (30 copies of the $R-file and 30 copies of the $I-file.)

As a real word example; the forensic examination of a Vista machine was conducted in which the user deleted about 100 different files. These 100 files quickly grew to 200 files as the original files were separated into $R-files and $I-files. Furthermore, these 200 files were archived in a volume snapshot every evening for a period of about 30 days. Thus, what was once a manageable number of files (100), quickly grew into a mess of about 6,000 $R-files and $I-files. Being able to manage the great quantity of duplicate data is a concern for the forensic examiner.

5.2 RECOGNIZING DUPLICATE DATA

Another challenge with analyzing Recycle Bin data within a shadow volume is the inability to quickly recognize duplicate data. Because the naming conventions used by Windows Vista and Windows 7 for deleted file names is unknown, it is not known for certain if one can assume that two files in the Recycle Bins of different shadow volumes are identical simply because their names are identical.

A version of this same problem occurs when files with the same content have different names. To illustrate this issue better, a simple text document named "test.txt" with the content "Hello" was created. When this file was deleted and sent to the Recycle Bin, this file was separated into two files, namely $IY57PWC.txt and $RY57PWC.txt. The original file was restored and then sent to the Recycle Bin again by deleting it a second time. This time, the file was separated into the files named $IC1IRLV.txt and $RC1IRLV.txt. Notice that the name of the files changed even though the content of the file did not change. This suggests the naming conventions used for deleted files are not based solely on content. Perhaps the deleted file's name is also based on the file's metadata (such as its deletion time). Or, perhaps the file name is purely random. Whatever the case, it seems clear that one cannot determine a deleted file's content based on the name assigned to it by the operating system.

If we assume the name of the deleted file (ignoring the "$I" or "$R" characters that are common to all deleted file names) is always 6 characters long, and each character is either a letter of the alphabet or a number between 0 and 9, then each character can be one of 36 possibilities. This means that there are only 36^6 or 2,176,782,336 possible names for the deleted files. Thus, if the deleted file's name is generated randomly, there is a 1 in about 2 billion chance that any two deleted files will have exactly the same deleted file name. Furthermore, the accidental duplication of randomized file names can be avoided by keeping track of which file names are currently in use, and then re-randomizing a file name if a file name is ever duplicated. The probability of having to re-randomize a file name is very low and therefore quite acceptable. If this scenario is accepted, then one can conclude that the names of the records of the deleted files (the $R-files and the $I-files) are also unique. Thus, duplicate records can be eliminated based on this uniqueness. In other words, if the user

ever encounters what appears to be multiple copies of the $R-file or $I-file (based on the names of the files), then the user can eliminate the extra copies of the files because these are assumed to be duplicates of the same file.

Unfortunately, this approach is still just a theory and perhaps should not be embraced too strongly. It is a theory that offers great confidence, despite the fact that it has not yet been proven correct.

5.3 THE ILLUSION OF "DUPLICATE" DATA

Experimentation has shown that despite what appears to be duplicate records of deleted files being maintained in different shadow volumes, this data really is not duplicated at all. It is an illusion. It is a false presentation that seems to be made by the operating system. This conclusion is arrived at after having conducted the following experimentation.

A file was deleted, which created an $R-file and $I-file in the $Recycle.Bin within the user profile that is associated with deleting the file. Three restore-points were created, which was theorized should result in a backup copy of the $R-file and $I-file being archived into each of the three shadow volumes. Thus, it was expected that there would be three actual copies of the $R-file and three actual copies of the $I-file, one of each in each of the three shadow volumes. The shadow volumes were accessed by way of symbolic links that were created by using the "mklink" command as described in section 4.1. Upon inspection of each of the three most recently created shadow volumes, it appeared as if each of these shadow volume had its own copy of the $R-file and $I-file. This was exactly as expected.

However, the theory fell apart when the contents of the Master File Table of the computer was viewed with EnCase 6.14.3. This tool revealed that only one copy (not three copies) of the $R-file and $I-file existed. Thus, the duplication of the $R-file and the $I-file is an illusion. This finding suggests that there is only one copy of the $R-file and the $I-file, and each shadow volume might have its own "pointer" to that one copy. This pointer can make the $R-file and $I-file appear to be in each shadow volume when in fact there is just one copy of these files. Thus, the duplication of data is really just an illusion that is created by the operating system and perhaps contributed to by how symbolic links are used for inspecting each shadow volume.

5.4 THE NEED FOR A NEW SOFTWARE TOOL

As was just explained, there are several issues with conducting a manual examination of shadow volume data. For one, the great volume of data makes a manual examination too time-consuming to be practical. Second, the amount of duplicated data merely adds to the frustration of conducting a manual examination of the data. These issues are best handled with a software tool that can quickly "hide" duplicate data so that each file only has to be inspected once by the forensic examiner.

In addition to the issues that were previously mentioned, there is also the issue of identifying identical files. For example, if the subject of an investigation made several copies of an illicit picture, and then "deleted" each copy of this picture, sending each of the deleted copies to the Recycle Bin, there would be duplicate copies of the same file in the Recycle Bin; but each deleted file would

have different $R-file and $I-file names. The duplication of these files might best be discovered by matching the file's MD5 values. Although it is possible to manually run an MD5 program against every file in a Recycle Bin and then compare the MD5 values manually, this is still a task that is best automated with a software tool.

In conclusion, because of many of the issues that face a forensic examiner, whom is conducting a manual examination of the Recycle Bin data that is found within the Windows 7 and Windows Vista shadow volumes, the reader must agree that a tool to assist in this type of examination would be greatly appreciated. This need for an automated tool was the motivation behind the development of Shadow Miner.

6. SHADOW MINER

In response to the need for an automated tool for doing examinations of shadow volumes, the Defense Cyber Crime Institute has begun development of a software tool known as Shadow Miner[1]. This tool automates the process of generating the "vssadmin list shadows" report, and then uses the results of this report to create symbolic links to the shadow volumes that are present within the target environment.

In order to run the current version of Shadow Miner (v. 1.0), the software tool needs to be executed from within a virtual copy of the evidence that is being examined. The current version of Shadow Miner has been found to work when an image copy of the evidence is booted with LiveView and VMware while Shadow Miner is executed from a CD that is in this virtual machine's CD drive. Future versions of Shadow Miner are not expected to require the evidence to be bootable. Efforts are being made to allow Shadow Miner to be run from a Vista examination station while the evidence being examined is attached to that station by way of FireWire and a write-blocking device.

One of the restrictions for using Shadow Miner is that it can only be used with operating systems that support persistent shadow volumes and the "vssadmin list shadows" command. The only operating systems that meet this requirement are Windows Vista and Windows 7.[2]

When Shadow Miner runs, it executes the command "vssadmin list shadows" via a batch file, and the report that is generated is redirected to a text file which is stored at location that is determined by the Shadow Miner settings. The selected location for this text file is usually a shared directory on the examination station. Once the text file is generated, Shadow Miner parses the data in this file to get the names of each shadow volume. Another batch command is executed that creates the symbolic links to these shadow volumes. Once these symbolic links are created, accessing the data within these shadow volume is as easy as traversing a typical Windows file structure.

Although there are several analytical tools that are built into the current version of Shadow Miner, the one that is the most relevant for this discussion is the Recycle Bin analysis tool. This tool displays the contents of the Recycle.Bin directories for all of the user profiles that have been archived into

[1] Shadow Miner is currently under development and is not available for general use. When Shadow Miner is fully developed, it is expected to be accessible through the National Repository for Digital Forensic Intelligence (https://www.nrdfi.net).

[2] Only the *Ultimate* version of each operating system has been tested.

shadow volumes by the Volume Shadow Copy Service. The user has the option of viewing the Recycle Bin data associated with just one user, or viewing the Recycle Bin data from several users at the same time. Furthermore, the user can view all of the shadow volumes at once or select a sub-set of shadow volumes to review. The design of the tool makes it easy for the forensic examiner to perform meaningful analysis on the Recycle Bin data.

Shadow Miner displays Recycle Bin data in columns, and this data can be further sorted to support analysis. One of the most interesting columns, is the column that holds the original name of the file that was deleted. This data is obtained from the $I-file that is associated with the original file's deletion. Since file names are usually very descriptive and they often reflect the contents of the file, being able to do a quick review of the original file names is often very useful for the forensic examiner. Thus, the column that holds these original file names is expected to be of great interest for the forensic examiner.

When the forensic examiner finds a Recycle Bin file that needs to be inspected further, he or she can use a special Shadow Miner tool that allows the user to copy the selected file to the examination station for analysis. Because each "deleted" file is separated into a $R-file and a $I-file when it gets sent to the Recycle Bin, these files need to be examined together. Therefore, these two files are paired-up by Shadow Miner before they are copied to the examination station for review. When these two files are copied out, Shadow Miner includes a small text file that provides useful information about the original deleted file, including where the file was found within the evidence computer, when the file was created, when the file was deleted, and which user profile is associated with that file. This information is included by Shadow Miner because this information is often needed by the prosecution when trying to obtain a successful conviction. Shadow Miner also updates the user interface to show which files have been copied to the examination station for review. This helps the forensic examiner keep track of which files still need to be reviewed.

Shadow Miner further supports the analysis of Recycle Bin data by providing tools that allow the forensic examiner to mark which files are "relevant" and "not relevant". "Relevant" file names are displayed in green and "not relevant" file names are displayed in red. A separate column entry also allows the examiner to sort files based on relevancy. Additional analysis tools allow the examiner to unmark previously marked files, as well as hide and unhide file names for easier analysis.

7. CONCLUSION

As this discourse comes to a close, the reader will recall that it began with an investigation into the Volume Shadow Copy Service. This Windows operating system service is responsible for taking "volume snapshots" of key data and storing this data in what is commonly known as a "shadow volume". These shadow volumes are found by using the "vssadmin list shadows" command. Once the shadow volumes are located, the "mklink" command is used to create symbolic links that can be followed to access each shadow volume. From this point, accessing the contents of each shadow volume is as simple as traversing a traditional Windows file system.

Once a forensic examiner is able to access a shadow volume, he or she must then address the issues that arise

from doing a shadow volume forensic examination. As was previously stated, he or she must deal with (1) the great volume of data, (2) the great amount of duplicated data, and (3) the need to identify files that have a common MD5 hash value. It was determined that these issues are best handled by an automated tool that is designed specifically for this task. The tool that was introduced to handle these issues is Shadow Miner – a tool developed at the Defense Cyber Crime Institute.

REFERENCES:

Arehart (2009). "Symbolic links on Windows: why and how" by Charlie Arehart. http://bluedragon.blog city.com/symbolic_links_on_windows_why_and_how.htm (accessed December 14, 2009)

Leschke (2010) "Cyber Dumpster Diving: $Recycle Bin Forensics for Windows 7 and Windows Vista" presented at the Department of Defense Cyber Crime Conference 2010.

Microsoft (2003) "Introduction to Shadow Copies of Shared Folders", http://www.microsoft.com/windowsserver2003/techinfo/overview/scr.mspx (accessed December 23, 2009)

Microsoft (2009). "How Volume Shadow Copy Service Works." http://technet.microsoft.com/en-us/library/cc785914%28WS.10%29.aspx (accessed December 14, 2009)