CHAPTER

Volume Shadow Copies 3

CHAPTER OUTLINE

INFORMATION IN THIS CHAPTER

- What are "Volume Shadow Copies"?
- Live Systems
- Acquired Images
- Windows 8

INTRODUCTION

Every time a new version of the Windows operating system is announced or made public, a collective shudder ripples throughout the forensics community. What new features are going to be available in the next operating system version? What's going to remain the same? What new challenges will we face? Some changes are minor; for example, the binary structure of the Windows Registry hasn't changed between versions, from Windows 2000 all the way through to Windows 7, although how the Registry is used (where keys are located, what keys and values are created and modified, etc.) by the operating system and applications has changed in

many cases. Other changes can be quite significant, such as those that change the very core of how Windows operates. In this chapter, we'll address one of those changes, specifically the introduction of Volume Shadow Copies (VSCs). However, we will discuss this topic not from the perspective of a developer or programmer, but instead from the perspective of an analyst, and how this technology might be utilized to further an investigation.

What are "volume shadow copies"?

VSCs are one of the new, ominous sounding aspects of the Windows operating systems (specifically, Windows XP, in a limited manner, and more so with Vista and Windows 7) that can significantly impact an analyst's examination. VSCs are significant and interesting as a source of artifacts, enough to require their own chapter.

With the release of Windows XP, Microsoft introduced the Volume Shadow Copy Service to provide functionality for backing up critical system files in order to assist with system recovery. With Windows XP, users and administrators saw this functionality as System Restore Points which were created automatically under various conditions (every 24 hours, when a driver was installed, etc.) and could also be created manually, as illustrated in Figure 3.1.

As illustrated in Figure 3.1, users can not only create Restore Points, but they can also restore the computer to an earlier time. This proved to be useful functionality, particularly when a user installed something (application, driver, etc.) that failed to work properly, or the system became infected with malware of some kind. Users could revert the core functionality of their systems to a previous state through the System Restore functionality, effectively recovering it to a previous state. However, System Restore Points do not back up everything on a system; for example, user data files are not backed up (and are therefore not restored, either), and all of the data in the SAM hive of the Registry is not backed up, as you wouldn't want users to restore their system to a previous point in time and have them not be able to access it, as a previous password had been restored. So, while System Restore Points did prove useful when a user needed to recover their system to a previous state, they did little to back up user data and provide access to previous copies of other files. From a forensic analysis, a great deal of historical data could be retrieved from System Restore Points, including backed up system files and Registry hives. Analysts still need to understand how backed up files could be "mapped" to their original file names but the fact that the files are backed up is valuable in itself.

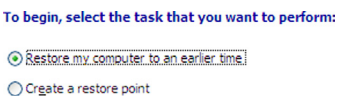

FIGURE 3.1

Windows XP System Restore Point functionality.

TIP

System Files in Restore Points

One use of system files being backed up to Windows XP System Restore Points is that when malware is installed as device driver (executable file with a ".sys" extension), it would be backed up to a Restore Point. If the installation process had included modifying the file time stamps so that the file appeared to have been created on the system during the original installation process, the true creation date could be verified via the master file table (MFT; see Chapter 4). Further, if there were six Restore Points, and the system file was not backed up in the older five Restore Points, and was only available in the most recent Restore Point, this would also provide an indication that the observed creation date for the file was not correct.

With the release of Vista, the functionality provided by the Volume Shadow Copy Service to support services such as Windows Backup and System Restore was expanded. In particular, the amount and type of data captured by System Restore was expanded to include block-level, incremental "snapshots" of a system (only the modified information was recorded) at a given point in time. These "snapshots"—known as VSCs—appeared in a different manner to the user. VSCs operate at the block level within the file system, backing up and providing access to previous versions of system and user data files within a particular volume. As with System Restore Points, the actual backups are transparent to the user, but with VSCs, the user can restore previous versions of files through the Previous Versions shell extension, as illustrated in Figure 3.2 (from a Windows 7 system).

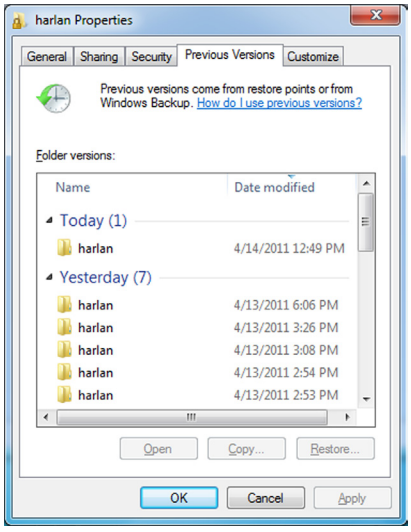

FIGURE 3.2

Windows 7 "Previous Versions" shell extension.

Okay, so what does this mean to the forensic analyst? From an analyst's perspective, there is a great deal of historical information within backed up files. Accessing these files can provide not just historical data (previous contents, etc.) but additional analysis can be conducted by comparing the available versions over time.

Registry keys

As you'd expect, there are several Registry keys that have a direct impact on the performance of the Volume Shadow Copy Service (VSS; the service which supports the various functions that lead to VSCs). As this is a Windows service, the primary key of interest is:

```
HKLM\System\CurrentControlSet\Services\VSS
```

However, it is important to understand that disabling the VSC Service may affect other applications aside from just disabling VSCs, such as Windows Backup. As such, care should be taken in disabling this service on production systems. Also, forensic analysts examining Vista and Windows 7 systems that do not appear to have any VSCs available should check this key to see if the service had been disabled prior to the system being acquired.

There's another key within the System hive that affects VSC behavior, and that is:

```
HKLM\System\CurrentControlSet\Control\BackupRestore
```

Beneath this key are three subkeys: FilesNotToBackup, FilesNotToSnapshot, and KeysNotToRestore. The names should be pretty self-explanatory, but just in case, the FilesNotToBackup key contains a list of files and directories that (according to Microsoft; additional information is available online at http://msdn.microsoft.com/en-us/library/bb891959(v=vs.85).aspx) backup applications should not backup and restore. On a default Windows 7 installation, this list includes temporary files (as in those in the %TEMP% directory), the pagefile, hibernation file (if one exists), the Offline Files Cache, Internet Explorer index.dat files, as well as number of log file directories. The FilesNotToSnapshot key contains a list of files that should be deleted from newly created shadow copies. Finally, the KeysNotToRestore key contains lists of subkeys and values that should not be restored. It should be noted that within this key, values that end in "\" indicate that subkeys and values for the listed key will not be restored, while values that end in "*" indicate that subkeys and values for the listed key will not be restored from backup, but new values will be included from the backup.

Another Registry key to be very aware of is the following:

```
HKLM\Software\Microsoft\Windows NT\CurrentVersion\SPP\Clients
```

This key contains a value named "{09F7EDC5-294E-4180-AF6A-FB0E6A0E9513}," and the data within that value will tell you which volumes are being monitored by the Volume Shadow Service. The data for this value can contain

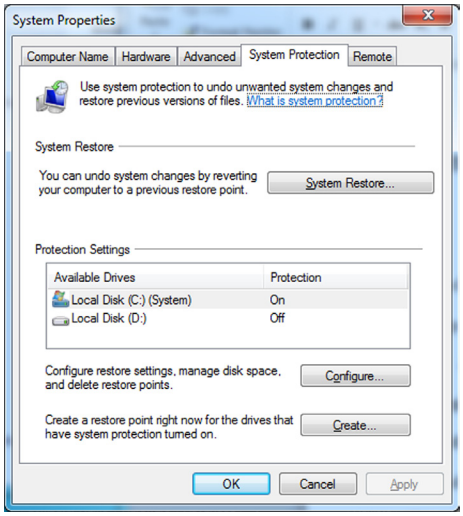

FIGURE 3.3

System Properties dialog.

multiple strings, each of which references a volume GUID and the drive letter for the volume, separated by a colon. This value will mirror what is listed in the Protection Settings section of the System Properties dialog, as illustrated in Figure 3.3.

> **TIP**
>
> **Finding VSCs**
>
> I've run into and used the SPP\Clients key quite a bit during examinations. One of the steps I include in order to orient myself to an image prior to an examination, I will check (via FTK Imager or ProDiscover, usually) to see if there are any difference files available within the System Volume Information folder. In a number of cases, I've found none, and upon further examination, found that the VSS service was set to run automatically upon system boot. During examinations in which historical information would be very valuable, I will then verify the LastWrite time on the SPP\Clients key, and check the data of the "{09F7EDC5-294E-4180-AF6A-FB0E6A0E9513}" value. Using this information, I can then state my findings based on those values in my report; many times, I find from the client that deleting or clearing the value is actually part of the standard system configurations for the enterprise.

Live systems

Accessing VSCs on live Vista, Windows 2008, and Windows 7 systems is a relatively simple task, as Windows systems ship with the necessary native system tools to access VSCs. In order to see the available VSCs for the C:\ drive of the Vista

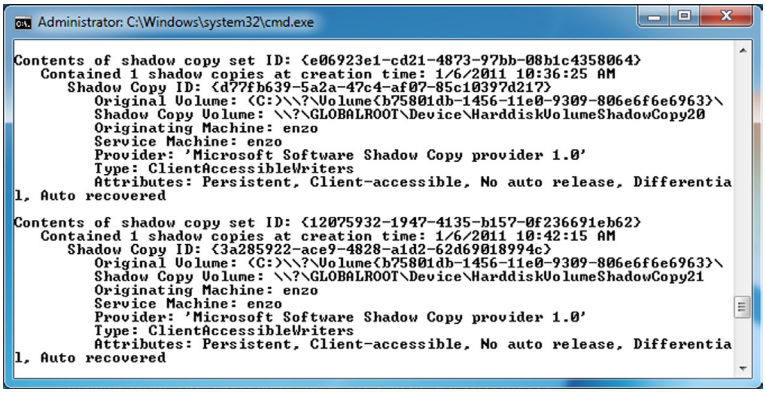

FIGURE 3.4

Sample output of the *vssadmin* command.

or Windows 7 system that you're logged into, type the following command into a command prompt using elevated privileges (you may need to right-click the command prompt window and choose "Run as Administrator"):

```
C:\>vssadmin list shadows /for=c:
```

Example results of this command are illustrated in Figure 3.4.

As you can see illustrated in Figure 3.3, we can use the *vssadmin* command to gather considerable information about available VSCs on the system.

WARNING

WMI

The Windows Management Instrumentation (WMI) class Win32_ShadowCopy (documentation found online at http://msdn.microsoft.com/en-us/library/aa394428(v=VS.85).aspx) provides an interface for programmatically extracting much of the same information from Windows systems made available by the *vssadmin* command. However, according to information available at the Microsoft web site (see the "Community Content" section of the previously linked page) at the time of this writing, this class is not supported on the 64-bit version of Windows 2008. Testing using a Perl script indicates that this is also true for Windows 7; the script didn't work at all on 64-bit Windows 7, but ran very well on the 32-bit edition. A sample of what is available via Perl (or other methods for accessing WMI classes) appears as follows:

```
Computer: WIN-882TM1JM2N2
DeviceObject: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
InstallDate: 20110421125931.789499-240
<snip>
VolumeName: \\?\Volume{d876c67b-1139-11df-8b47-806e6f6e6963}\
```

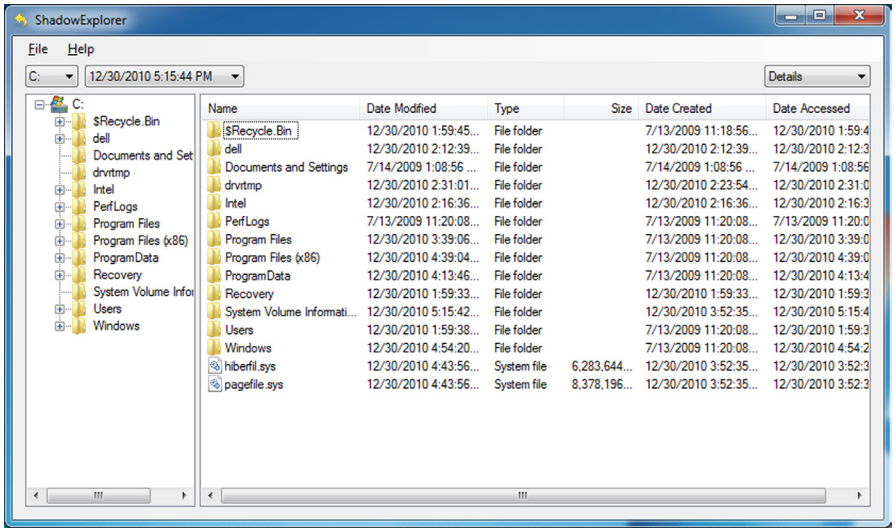

FIGURE 3.5

ShadowExplorer v0.8 interface.

Don't like the command line approach? Hey, that's okay—it's not for everyone. Head on over to ShadowExplorer.com and get a copy of ShadowExplorer (at the time of this writing, version 0.8 is available). Download and run the setup file on your system in order to install ShadowExplorer on the system in question. The web site describes ShadowExplorer as being useful to all users, but especially so to users with Windows 7 Home Edition, who don't have access to VSCs by default. Once you install and launch ShadowExplorer, you will see the interface as illustrated in Figure 3.5.

As illustrated in Figure 3.5, you can use the drop-down selector beneath menu bar to select the date of the VSC you would like access to; unfortunately, ShadowExplorer will only show you the VSCs available within the volume or drive (i.e., C:\, D:\) on which it is installed. Therefore, if your system has a D:\ drive, you'll need to rerun the installation program and install it on that drive, as well, in order to view the VSCs on that drive. Navigating through the tree view in the left-hand pane, locate the file for which you'd like to see a previous version, right-click the file and choose "Export" to copy that file to another location.

Going back to the command prompt, in order to access the VSCs on your live system and have access to the previous versions of files within those VSCs, you'll need to make a symbolic link to a VSC. To do that, go to the listing for a VSC, as illustrated in Figure 3.3, and select (you'll need to have Quick Edit mode enabled in your command prompt) the VSC identifier, which appears after "Shadow Copy Volume:." Then go back to the prompt and type the following command:

```
C:\>mklink /d C:\vsc
```

Do not hit the Enter key at this point. Once you get the far with command, right-click to paste the selected VSC identifier into the prompt and then add a trailing slash ("\"), so that the command looks like the following:

```
C:\>mklink /d C:\vsc \\?\GLOBALROOT\Device\
HarddiskVolumeShadowCopy20\
```

Remember to add the trailing slash to the command—this is very important! This is not something that is clearly documented at the Microsoft site, but has been found to be the case by a number of forensic analysts, to include Rob Lee, of SANS fame, and Jimmy Weg, a law enforcement officer from Montana. Now, go ahead and hit the Enter key, and you should see that the symbolic link was successfully created. Now you can navigate to the C:\vsc directory, and browse and access the files via the command prompt or Windows Explorer. Once you're done doing whatever you're going to do with these files (review, copy, etc.), type the following command to remove the symbolic directory link:

```
C:\>rmdir C:\vsc
```

This series of commands is going to be very important throughout the rest of this chapter, so it's important that we understand some of the key points. First, use the *vssadmin* command to get the list of VSCs for a particular volume; note that when you run the command from the command prompt, you do not have to be *in* that volume. For example, if you want to list the VSCs for the D:\ volume, you can do so using the following command, run from the C:\ volume.

```
C:\>vssadmin list shadows /for=d:
```

Once you know which VSC you'd like to access, you can use the *mklink* command to create a symbolic link to that VSC. Remember, you must be sure that the VSC identifier (i.e., \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy20\) ends with a trailing slash. Finally, once you've completed working in that VSC, you remove the symbolic link with the *rmdir* command.

ProDiscover

A number of commercial forensic analysis applications provide access to VSCs within acquired images, and ProDiscover is just one of those applications. However, ProDiscover is also the only commercial forensic analysis application to which I have access. As such, I briefly mention its ability to access VSCs on live systems here. For those who want more detailed information on how to use ProDiscover for this purpose, Christopher Brown posted a five-page PDF format paper at the Technology Pathways, LLC, web site that describes how to use ProDiscover IR (the Incident Response Edition) to access and acquire VSCs on remote live systems. This can be very valuable to an investigator who needs to quickly access these resources in another location, or to do so surreptitiously. The paper can be found on the web at http://toorcon.techpathways.com/uploads/LiveVolumeShadowCopyWithProDiscoverIR.pdf.

F-Response

If you're a user of the fantastic F-Response tool from Matt Shannon, particularly the Enterprise Edition (EE), you'll be very happy to know that you can use this product to access VSCs on remote systems. This may be important for a variety of reasons; a user within your enterprise environment may have "lost" an important file that they were working on, you may need to access an employee's system surreptitiously, or you may need to quickly acquire data from a system located in another building in another area of the city. While I generally don't recommend acquiring full system images over the network, even over a VPN, you can use tools like F-Response EE, which provides read-only access to the remote system drive, in order to collect specific information and selected files from remote systems very quickly. This will allow you to perform a quick triage of systems, and potentially perform a good deal of data reduction and reduce the impact of your response activities on your organization by identifying the specific systems that need to be acquired.

That being said, perhaps the best way to discuss F-Response EE's ability to provide access to VSCs is through a demonstration. Before describing the setup I used and walking through this demonstration, I need to make it clear that I used F-Response EE because Matt was gracious enough to provide me with a copy to work with; this process that I'm going to walk through can be used with all versions of F-Response, including the Consultant and Field Kit editions.

TIP

F-Response VSC Demo Setup

For my demonstration, I don't have a full network to "play with," so I opted to use the tools that I do have available. I booted my 64-bit Windows 7 Professional analysis system, and then started up a 32-bit Windows 7 Ultimate virtual machine (VM) in VMPlayer. I had set the Network Adapter in the settings for the VM to "bridged," so that the VM appeared as a system on the network. For the demonstration, the IP address of the running VM was 192.168.1.8, and the IP address of the host was 192.168.1.5. On both systems, the Windows firewalls were disabled (just for the demonstration, I assure you!) in order to simulate a corporate environment. Also, it is important to note that Windows 7 ships with the iSCSI initiator already installed, so I didn't need to go out and install it separately.

Again, this demonstration makes use of F-Response EE (thanks to Matt Shannon for allowing me the honor to work with this wonderful tool!). Once I logged into my analysis system, I plugged in my F-Response EE dongle and launched the F-Response License Manager Monitor to install and start the License Manager service. I then launched the F-Response Enterprise Management Console (FEMC) and started by configuring the credentials that I would be using to access the remote system. I clicked File→Configure Credentials… from the menu bar, and entered the appropriate username/password information to access the remote system (if you're in an Active Directory domain, check the "Use Current User

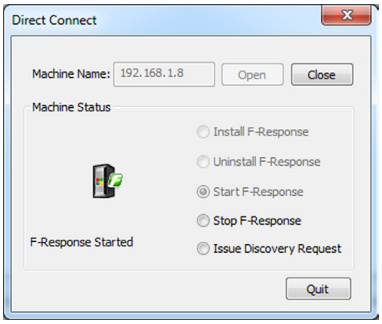

FIGURE 3.6

FEMC Direct Connect UI.

Credentials" option). Next, I clicked File→Configure Options… and configured my deployment options appropriately (for this demo, I didn't select the "Physical Memory" option in the Host Configuration section).

As I was going to connect to a specific system, I selected Scan→Direct Scan from the menu bar, and entered the IP address of the target system (i.e., 192.168.1.8), and clicked the Open button. Once the connection was made, F-Response was installed and started on the target system, as illustrated in Figure 3.6.

From there, I logged into the C:\ volume on the target host, and that host's C:\ drive appeared on my analysis system as the F:\ volume. I then ran the following command on my analysis system:

```
C:\>vssadmin list shadows /for=f:
```

In order to access the oldest VSC listed (HarddiskVolumeShadowCopy17, created on January 4, 2011), I entered the following command in a command prompt on my analysis system:

```
C:\>mklink /d d:\test \\?\GLOBALROOT\Device\
HarddiskVolumeShadowCopy17\
```

This command created a symbolic link on my analysis system called "d:\test" that contained the contents of a VSC created on the target system on January 4, 2011, and allowed me to access all of the files with that directory, albeit via the read-only access provided by F-Response EE.

WARNING

Accessing VSCs on Live Systems

It is very important to remember that when you're accessing VSCs on live systems, that system, whether accessed remotely or locally, is still subject to operating normally. What this means is that if you're accessing the oldest VSC that you found, the system itself is still going about its normal operations, and that VSC could be overwritten to make

room for another VSC, as under normal conditions, the VSCs are subject to the "first-in-first-out" (FIFO) process. This actually happened to me while I was working on some of the demonstrations listed in this chapter. The remote live system continued to operate normally, and the VSC I was accessing was removed simply because I had taken too long to complete the testing (I was just browsing through some of the files). I had to back out of my demonstration and restart it. When I did, I found that the output of the *vssadmin* command was quite a bit different, particularly with respect to the dates on which the available shadow copies had been created.

Another very important aspect of accessing VSCs (and this applies to accessing VSCs within images, as well) is that you need to be very careful about the files you click or double-click on. Remember, if you double-click a file that is in a VSC on a remote system, your analysis system is going to apply its own rules to accessing and opening that file. This means that if you see a PDF file that you'd like to click on, you should be very sure that it wasn't what led to the remote system being infected in the first place. If it is a malicious PDF, and your system isn't protected (updated antivirus (AV) and PDF viewer, etc.), then your system may become infected, as well.

As I mentioned, there are a number of commercial forensic analysis applications and tools that provide analysts and responders with the ability to access VSCs on remote systems, and what we've discussed here are only a few of your (and my) available options. The application and methodology you choose to use depends largely on your needs, abilities, and preferences (and, of course, which tool or set of tools you can afford).

Acquired images

Since discussion of VSCs first started, one of the biggest and most often asked questions within the forensic analysis community has been, "how do we access VSCs within acquired images?" First of all, accessing VSCs within images is not the same thing as accessing those on live systems. Figure 3.7 illustrates what the VSCs "look like" within an acquired image.

As illustrated in Figure 3.7, the VSC difference files within the System Volume Information directory are binary files, and we need some means for translating

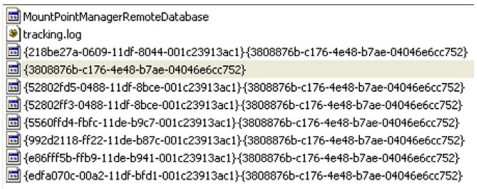

FIGURE 3.7

Acquired image of Vista system opened in FTK Imager v3.0.

this binary data into accessible information. On live systems, this is usually done through the use of the available API; therefore, one means of accessing the same data on an acquired image would be to boot the image through the use of LiveView and VMWare.

TIP

LiveView

LiveView, freely available online from http://liveview.sourceforge.net/, is a Java-based graphical tool developed by a student at Carnegie Mellon University. LiveView creates VMWare configuration files for acquired raw/dd-format images or physical disks and supports Windows versions from Windows 98 through Windows 2008 (Windows 7 is not listed among the supported operating systems).

However, even with the ability to "zero out" (not crack, but reset to a new value, possibly using a tool such as ntpwedit, found online at the time of this writing at http://cdslow.webhost.ru/en/ntpwedit/) the Administrator password so that you can log into the now-running system, this may still not be a viable option. So, the question becomes, with nothing more than an acquired image of a system that may contain VSCs, what are some options for gaining access to the data within those VSCs?

I asked myself this question seriously during the break between Christmas 2010 and the New Year, and I began researching it in order to find a solution. After all, I'd encountered several systems that contained VSCs, including Windows 7 and even a Vista system. In my case, neither instance required access to the VSCs in order to complete my analysis, but it was still clear to me that like other analysts, I could fully expect to see more of these systems. Subsequently, I was going to have to come up with a way to access the VSCs.

I began my search by going to Google… of course. I found a number of references to accessing VSCs within acquired images, but in each case, the materials included mounting the acquired image using EnCase (from Guidance Software) and the Physical Disk Emulator (PDE) module, as part of the process. Well, I don't have access to EnCase, nor to the PDE module, and I thought that there just *had* to be some way to access data within the VSCs of an acquired image, without using either one.

For my testing, I had an image acquired from a personal system that was running a 32-bit version of Windows Vista. This was an image of the physical hard drive, and as the system was a Dell laptop, the image contained several partitions including the Dell maintenance partition. As such, I used FTK Imager version 3.0 to extract the active operating system partition from the image, as I wanted to isolate the partition that contained the VSCs. The disk image was called "disk0.001," and the image of the active partition was called "system.001." My analysis workstation was a Dell Latitude E6510 laptop, running a 64-bit version of Windows 7 Professional. On that laptop, I had a copy of FTK Imager version 3.0.0.1443, as well as ImDisk 1.3.1.

VHD method

A "VHD" file is a virtual hard disk file used by virtualization software such as Microsoft's Virtual PC or Virtual Server (but can also be used by Oracle's VirtualBox application, as well). The VHD file represents a physical hard disk and can be used by a VM as if it were a physical hard disk. Additional information regarding VHD files can be found online at the Microsoft web site at http://technet.microsoft.com/en-us/library/cc708315%28WS.10%29.aspx.

As part of my research for this little project, I found vhdtool.exe at the Microsoft site (found on the web at http://code.msdn.microsoft.com/vhdtool). I also found that Microsoft's Virtual Server application includes a tool named "vhdmount" (information about this tool can be found online at http://technet.microsoft.com/en-us/library/cc708295%28WS.10%29.aspx) for mounting VHD files. In reading about vhdtool.exe, it has an option ("/convert") for converting a raw/dd-format image file into a fixed-format VHD file. I ran the tool against a copy of the system.001 file (the active OS partition image described above, on an external USB wallet drive) and although the file name was not changed to ".vhd," the tool reported that it had successfully modified the file (apparently by adding a footer). From there, the next step was to mount the new VHD file; I did this by opening the Computer Management console, selecting Disk Management and clicking Action, then Attach VHD from the menu bar. The system.001 file was recognized as a valid VHD file, resulting "Attach Virtual Hard Disk" dialog is illustrated in Figure 3.8.

Notice in Figure 3.8 that I had selected the option to mount the VHD file read-only. Even though I was using a working copy of the image file, and it had already been modified (via the use of vhdtool.exe, which I documented), I wanted to be sure to follow best practices in my procedures.

As a result of attaching the VHD file, the Disk Management console showed a 136.46 gigabyte (GB) partition mounted as Disk 2, and listed as the G:\ drive/volume, as illustrated in Figure 3.9.

Opening Windows Explorer, I could clearly see the files within in the G:\ volume; I confirmed this using the *dir* command to generate a file listing from the command prompt. The next step was to determine which VSCs were available, if any. To do this, I ran the following command from the command prompt:

```
vssadmin list shadows /for=g:
```

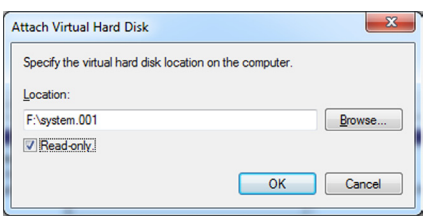

FIGURE 3.8

Windows 7 Disk Manager "Attach Virtual Hard Disk" dialog.

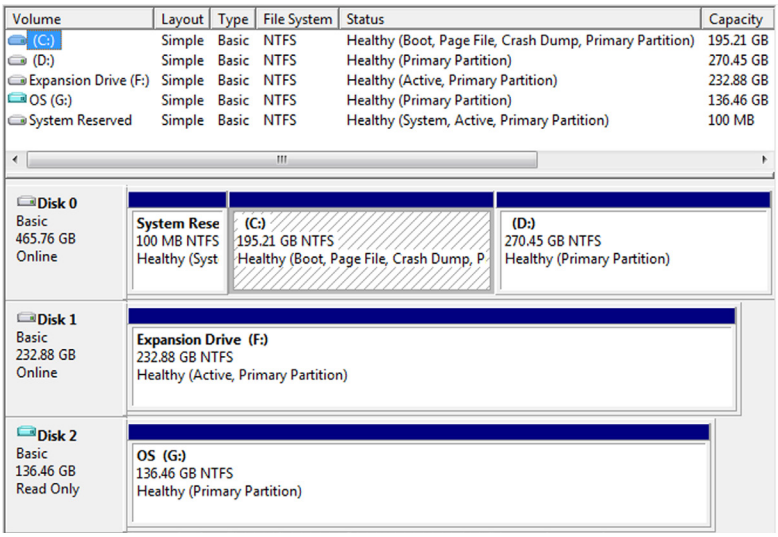

FIGURE 3.9

Disk Management console showing G:\ volume.

The output of this command indicated that there were a total of seven VSCs available in the image, with creation dates ranging from January 10, 2010 to January 20, 2010. I opted to mount the oldest VSC; to do so, I selected \\?*GLOBALROOT\Device\HarddiskVolumeShadowCopy23*, which appeared after "Shadow Copy Volume:" in the output of the above *vssadmin* command, and right-clicked to copy this string to the clipboard. I then returned to the command prompt and typed in the following command:

```
D:\>mklink /d d:\vsc23
```

After typing the above command, I right-clicked to paste the \\?\GLOBALROOT… string that I'd copied to the clipboard at the end of the command, and then I made sure to add a closing "\" to the end of the command, and hit the Enter key. The result was that the symbolic link from the VSC to D:\vsc23 was successfully created.

TIP

Final Backslash

The final backslash at the end of the *mklink* command is critically important! Without it, you won't be able to access the mounted VSC properly.

At this point, I had the image file mounted as a VHD file, and the oldest VSC within the image also mounted and accessible from my analysis system (confirmed via the *dir* command). Using robocopy.exe (which is native to Windows 7) to preserve file metadata (time stamps), I copied the contents of a user's profile directory (albeit not the subdirectories) from both the mounted VHD file (the imaged Vista operating system partition) and the mounted VSC within the VHD file in order to run a quick comparison against the NTUSER.DAT files, and in particular the contents of the UserAssist key. I could have run RegRipper (specifically rip.pl or the "compiled" executable version of the tool, rip.exe) from the analysis system against the mounted VHD and VSC to obtain the information I was looking for, but copying the files gave me an excuse to run the *robocopy* command (until then, I hadn't ever used the command). To get information from the UserAssist keys from the two copied NTUSER.DAT hive files, I ran the following command:

```
C:\tools>rip.pl -r <path>\ntuser.dat -p userassist2>output.txt
```

Running the command against each hive file, redirecting the output to the appropriate text file, allowed me to then open the output files in an editor and compare them. From the NTUSER.DAT hive file from the oldest VSC within the image, I found the following entries:

```
Sat Jan 9 11:40:31 2010 Z
 UEME_RUNPATH:C:\Program Files\iTunes\iTunes.exe (293)
Fri Jan 8 04:13:40 2010 Z
UEME_RUNPATH:Skype.lnk (5)
 UEME_RUNPATH:C:\Program Files\Skype\Phone\Skype.exe (8)
```

Then, from the NTUSER.DAT hive file from the VHD image file itself, I found the following entries:

```
Thu Jan 21 03:10:26 2010 Z
 UEME_RUNPATH:C:\Program Files\Skype\Phone\Skype.exe (14)
 UEME_RUNPIDL:C:\Users\Public\Desktop\Skype.lnk (1)
Tue Jan 19 00:37:46 2010 Z
 UEME_RUNPATH:C:\Program Files\iTunes\iTunes.exe (296)
```

What this clearly demonstrates is the changes that occur between various VSCs and the actual running system, as well as the forensic value of VSCs. As you can see from the above examples, in the space of 12 days, the user had run the Skype application six times, and in about 10 days, had run the iTunes application three times. As the UserAssist key records the date and time that the application was most recently run, all we would normally be able to determine from the image of the Vista was that as of January 21, 2010, the Skype application had been run a total of 14 times by the user. However, by accessing the VSCs, we're able to obtain historical information regarding previous times that the user had run the Skype application. This same concept applies to other Registry keys, as well, particularly

those that maintain lists of subkeys and values. Specific keys that may be of interest during an examination may include "most recently used" or "MRU" lists; these keys usually contain a number of values, and the LastWrite time of the key corresponds to the date when the last file was accessed. However, we may be able to use data from hive files within VSCs to determine the dates and times when other files within the MRU list were accessed, as well. Being able to access this type of temporal information allows an analyst to infer certain things about a user's behavior on the system, particularly if (per this example) the fact that the user launched Skype six times in the space of approximately 12 days is pertinent to the goals of the examination (additional information regarding the user's activity could then be obtained from the application's log files). It should be clear from this that there is significantly more value to VSCs than simply previous versions of graphic image files.

TIP

Registry Analysis

A more detailed discussion of analysis of the Windows Registry hive files can be found in Chapter 5 of this book, as well as within *Windows Registry Forensics* [1].

Once I had completed all that I wanted to do (mostly just browsing), I removed the symbolic link that I'd created to the VSC using the following command:

```
D:\>rmdir d:\vsc23
```

As the symbolic link was created to a directory (i.e., *mklink /d*), I needed to treat the symbolic link as a directory in order to remove it (i.e., *rmdir* or *rd*). I then returned to the Disk Management console (see Figure 3.8), right-clicked on the "Disk 2" box to the left of the G:\ volume (displayed in the lower pane) and chose "Detach VHD" from the context menu.

TIP

Diskpart

The *diskpart* command (a reference for the command, albeit specifically for Windows XP, can be found online at http://support.microsoft.com/kb/300415) can be used to attach and detach VHD files from the command line. First, you need to simply type "diskpart" at the command prompt in order to being working in the diskpart shell. In order to attach a VHD file, use the following commands:

```
selectvdisk file=<path to VHD file>
attachvdisk
```

Using these commands, the VHD file is automatically mounted using the next available drive letter. In order to detach the VHD file, use the following command:

```
detachvdisk
```

In summary, the process you would follow to access VSCs using this method would be to:

- Convert a working copy of your image file to a VHD file using vhdtool.exe.
- Attach/mount the newly created VHD file to your Windows 7 analysis workstation, using either the Disk Management console, or diskpart.exe. Be sure to check the "Read-Only" box (see Figure 3.7) when mounting the VHD file.
- Determine how many VSCs you have available within the image, and for which dates, using vssadmin.exe (i.e., *vssadmin list shadows /for=n:*).
- Create a symbolic directory link to the VSC (or VSCs) of interest using mklink.exe (i.e., *mklink /d C:\mountpoint\\.?\GLOBALROOT\Device\HarddiskVolumeShadowCopyn*). Note: The trailing backslash in the *mklink* command is critically important!
- Perform whatever work is part of your analysis plan (i.e., copy files via robocopy, scan the mounted VSC with AV scanners)
- Remove the symbolic link with the *rmdir* command. When you've completed working with the VHD file itself, detach it via the Disk Management console or diskpart.exe.

I should note that mounting a working copy of your acquired image as a VHD file can be used for much more than accessing VSCs. For example, all of those tasks we mentioned performing against a mounted/linked VSC (scanning with AV, performing other malware detection steps, etc.) can be performed on just the mounted VHD file.

VMWare method

After I figured out how to access the VSCs within an acquired image via the VHD method, I began discussing this with others, and found out that folks like Rob Lee (of SANS and Mandiant fame) and Jimmy Weg (a law enforcement officer from Montana) have been using VMWare in a very similar manner to access VSCs. Discussing the VMWare method with both of them, I got an idea of the process that they used, and decided to try it on my own to see if I could get it to work. In order to work through this process you'll need the following:

- The ability to run a VMWare virtual machine, such as VMPlayer (freely available, and found online at http://www.vmware.com/products/player/) or VMWare Workstation (a 30-day evaluation version is available online from http://www.vmware.com). Using VMWare Workstation, you can create your own VMs.
- A Windows 7 VM (I used a 32-bit Windows 7 Ultimate VM for this demonstration).
- A copy of LiveView or ProDiscover Basic Edition (BE).

The first thing I did was downloadVMPlayer from the VMWare web site and get a copy of LiveView. Having only an image in raw/dd format, I needed a way to get

the data within the image recognized as a disk or partition by the VMWare tools. LiveView provides that capability by generating a VMWare virtual machine disk format (.vmdk) file that points to the image; however, for this demonstration, I just wanted the vmdk file, and I didn't necessarily want to boot the VM.

TIP

ProDiscover

The ProDiscover forensic analysis application, from Technology Pathways, LLC, includes functionality for creating VMWare VMDK files (similar to LiveView). This functionality is included in the BE, a freely available version of the application. After you've installed ProDiscover BE, open the application, and under the Tools menu option, choose Image Conversion Tools and then "VMWare Support for "DD" Images…," as illustrated in Figure 3.10.

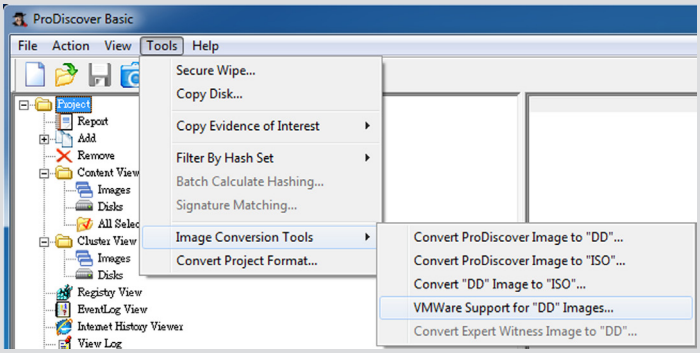

FIGURE 3.10

Selecting VMWare Support for "DD" Images in ProDiscover BE.

When the resulting dialog opens, browse for and select the raw/dd-format image file you're interested in (remember, we're going to use the one named "system.001") as illustrated in Figure 3.11 and click OK.

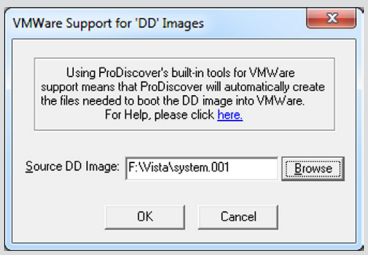

FIGURE 3.11

ProDiscover BE VMWare Support for "DD" Images dialog.

You won't see any progress bar or notification, but a .vmdk file pointing to the raw/dd-image file will be created. You can then add the .vmdk file to an existing VM as a hard disk.

Next, launch VMPlayer and select your VM, but do not start it; instead, edit the VM settings to add the newly created .vmdk file to the VM as an additional disk.

WARNING

Nonpersistent Disk

In the following section, we'll be adding an independent, nonpersistent disk to an existing VM via VMWare Workstation. The option to add a new hard disk that is nonpersistent is *not* available in VMPlayer, at least not at the time of this writing. As such, if you choose to use this method to access VSCs, you need to be sure to use a working copy of your image, or use other mechanisms to ensure that the image itself isn't modified.

When the Add Hardware Wizard opens and allows you to select a disk, choose "Use an existing virtual disk" and click Next, as illustrated in Figure 3.12.

In the "Select an Existing Disk" dialog, browse to the newly created .vmdk file (in our example, system.vmdk) and click Finish. At this point, if you get a message from VMPlayer (or Workstation) about converting the virtual disk format to a newer format, simply choose to keep the existing format. After you've added the new hard disk to the VM, boot it, log in, and open Windows Explorer to see the file system for the added disk. From here, you can view and access the VSCs using the same process we discussed previously in the chapter.

If you're using VMWare Workstation, when you get to the "Select an Existing Disk" dialog, you will be presented with some additional options, as illustrated in Figure 3.13.

When adding the new .vmdk file to as a hard disk to your VM, go to the Mode section of the dialog and select "Independent," and then "Nonpersistent." This will help ensure that any changes made to image file as a result of your analysis (or by

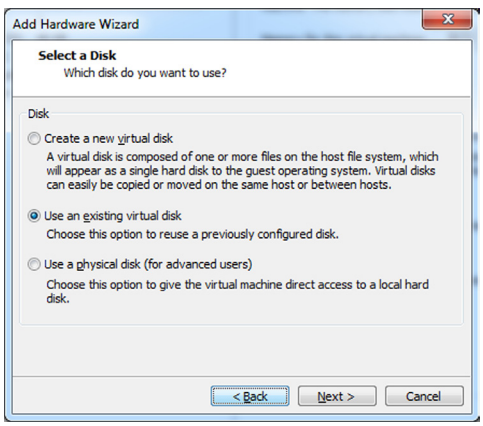

FIGURE 3.12

VMWare Workstation "Select a Disk" dialog.

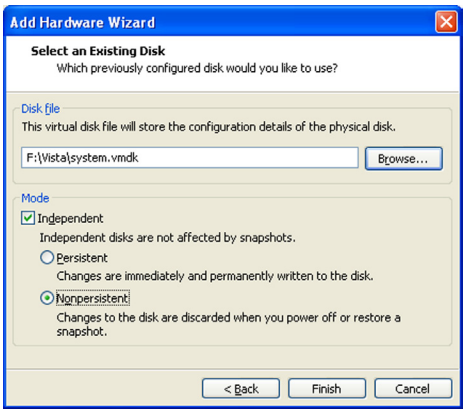

FIGURE 3.13

Adding an independent, nonpersistent disk.

the operating system) are not written to the image. This is simply an additional step you should take as part of sound analysis practices; you should already be working with a copy of your image, not the original image.

> **TIP**
>
> **VMDKs and SIFT**
>
> I mentioned in Chapter 1 that I had used the SANS SIFT v2.0 Workstation VM that Rob Lee put together. I was working out the kinks in some ideas that I had and was going to try to access a raw/dd-format image of a Windows XP system, but this specific experiment required that I access the image as a .vmdk file. In short, I found that LiveView did a much better job of creating the necessary VMWare files for use with the SIFT Workstation than did ProDiscover BE. When I added the .vmdk file created via ProDiscover BE as an additional hard drive to the SIFT VM and ran the *fdisk* command, I got some very odd output. However, when I did the same thing, using the same image file, but using the VMWare files created through LiveView, everything worked just fine.

Automating VSC access

During conferences or training courses, I'm often asked by other analysts, "How can I fully exploit VSCs?" To that, my response is, "what do you want to do?", and that's where the conversation usually comes to a screeching halt. What does "fully exploit" really mean? As we've discussed so far in this chapter, once you've attached an image to your analysis system using either the VHD or VMWare method, you'll be able to access to the available VSCs, and from that point, you can use almost all of the techniques and processes that you already use to "fully exploit" the data available in the VSCs.

One way to collect information from available VSCs is to image the entire VSC. So, you have an image attached to your analysis workstation, and you can image an available VSC from the attached volume, using George M. Garner, Jr's Forensic Acquisition Utilities (found online at http://gmgsystemsinc.com/fau/). Download the archive and be sure to the use the appropriate version (32- or 64-bit) for your platform. You can then use the appropriate version of dd.exe to create a logical image of a VSC using the following command (substituting for the appropriate VSC number, of course):

```
C:\tools>dd.exe if=\\.\HarddiskVolumeShadowCopy20 of=D:\vsc20.img
-localwrt
```

One thing to consider about this method is that you will likely need a considerable amount of storage space. For a 70-GB volume, if there are nine VSCs, you will need a total of 700 GB of space; 70 GB for the original volume, and another 70 GB for each of the VSCs. This method for acquiring data from VSCs is resource-intensive, but there may be times when it is absolutely necessary. However, keep in mind that VSCs are not complete backups of the system at a point in time, and as such, acquiring an image of the entire VSC may be more effort than is required.

When it comes to accessing and collecting information from the VSCs, you can also use Windows native batch file functionality to automate a great deal of your data collection. Automation in this manner not only increases efficiency and reduces the chance of errors (i.e., typing the wrong command, or commands in the wrong sequence), but it's self-documenting, as well; simply keep a copy of the batch file (and any output) as your documentation. While we're discussing accessing VSCs within acquired images, you will see you can also use these same automation techniques to access VSCs on live remote systems, as discussed earlier in this chapter. Doing so will help you mitigate issues with the oldest VSCs being deleted through the normal function of the system while you're accessing it, as a batch file will run much quicker than typing all of the commands manually.

As we've discussed, once you've run the *vssadmin* command, you should see a list of the available VSCs in the output. You will see the list in the command prompt, or you can redirect the output of the command to a file and view the list that way. So let's say that you have four VSCs, listed as HarddiskVolumeShadowCopy20 through 23, and you'd like to run the same series of commands on each of these VSCs, in succession. You can do this using batch files, which is a capability native to Windows systems. For example, we can use the following command in a batch file (call it "vsc_sweep.bat," or something that you'd find meaningful) as the initial command that handles creating a symbolic link to each VSC:

```
for /l %i in (20,1,23) do mklink /d C:\vsc\vsc%i \\?\GLOBALROOT\
Device\HarddiskVolumeShadowCopy%i\
```

Once this command has completed, you should have four symbolic links created, C:\vsc\vsc20 through vsc23. At this point you can run through the directories,

running whichever commands you choose. On April 13, 2010, a post to the "Forensics from the sausage factory" blog (http://forensicsfromthesausagefactory.blogspot.com) illustrated a command for using robocopy.exe to retrieve copies of specific files from the VSCs. That command, modified to work along with the previous command, looks as follows:

```
for %i in (20,1,23) do robocopyC:\vsc\vsc%i\Users C:\vsc_output\
vsc%i *.jpg *.txt /S /COPY:DAT /XJ /w:0 /r:0 /LOG: C:\vsc_output\
Robocopy_log_SC%i.txt
```

This command copies (via robocopy.exe) all of the files that end with .jpg and .txt extensions from the user profiles within the VSCs to a specific directory on the analysis computer, and logs the activity. As such, a copy of robocopy.exe must be located in the same directory as the batch file, and you should make sure that the C:\vsc_output directory exists before running the commands.

After you're done accessing the VSCs, you can remove the symbolic links using the following command:

```
for /l %i in (20,1,23) do rmdir C:\vsc\vsc%i
```

In April 2011, Corey Harrell (author of the "Journey into IR" blog at http://journeyintoir.blogspot.com) contacted me with the interesting idea of running RegRipper (more specifically, rip.exe) against successive VSCs in order to collect specific information. Using "&&" to append commands together in a single line in a batch file, Corey's idea was to collect information (Corey's original submission made use of recentdocs.pl RegRipper plugin) from a specific user's NTUSER.DAT hive file. The specific command (modified for use in this example) that Corey had put together was as follows:

```
for /l %i in (20,1,23) do (echo ----------------------
---------------- » output-file.txt && echo Processing
HarddiskVolumeShadowCopy%i» output-file.txt && C:\tools\
rip.exe -r c:\vsc\vsc%i\Users\user-profile\NTUSER.dat -p
userassist2»userassist.txt)
```

The batch file and various commands that we've discussed here are just a few simple examples of what you can do using batch file functionality that is native to Windows systems.

TIP

Batch Files

There are a number of very useful resources available online that provide references for batch file commands, such as http://www.computerhope.com/batch.htm and http://ss64.com/nt/. You can also find tutorials, such as http://commandwindows.com/batch.htm, that will assist you in writing batch files.

Corey also created a more comprehensive and functional batch file, which he graciously consented to allowing me to include in the additional materials associated with this book. The batch file is named "rip-vsc.txt" and can be found in the associated materials associated with this book (which can be found online at http://windowsir.blogspot.com/p/books.html). Corey spent some time in documenting and explaining the use of the batch file, by adding comments (lines that begin with "REM") to the file.

Internet Evidence Finder version 4 (available online at http://www.jadsoftware.com/go/?page_id=141) is a software application that can search files or hard drives for indications of a wide range of Internet-related artifacts, including Facebook, MySpace, mIRC, and Google chat, web-based emails, etc. The web page for IEF4 states that the application can also be used to search mounted VSCs.

ProDiscover

On March 3, 2011, Christopher Brown released version 6.9.0.0 of ProDiscover (all versions, including the BE). I've had a license for ProDiscover IR Edition since version 3, for which I'm very grateful to Chris. Over the years, I've had the privilege of watching the evolution of this product and used it to analyze a number of images. The latest update (as of this writing) provides access to VSCs, which (as we've discussed) can be extremely valuable to the examiner. I should note that in September 2011, Christopher released version 7.0.0.3 of ProDiscover.

To demonstrate accessing VSCs via ProDiscover IR, I have the application installed on a Windows XP SP3 system, and I have an image of a hard drive from a Dell laptop running Vista. First, I opened ProDiscover and created a new project, and then added the image of the hard drive to the project. Once the image was added (and I saved the project file), I clicked on the image file listing in the Content View to see the context menu illustrated in Figure 3.14.

Then, I clicked on "Mount Shadow Volume…" in the context menu and saw the "Mount Shadow Volume" dialog box illustrated in Figure 3.15.

As you can see in Figure 3.15, the mounted image has four partitions available, which, for those familiar with systems from Dell, is fairly common for default installations (when I purchase Dell systems for myself, the first thing

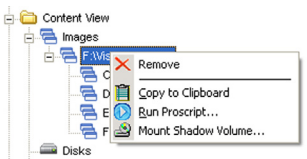

FIGURE 3.14

ProDiscover Mount Shadow Volume… functionality.

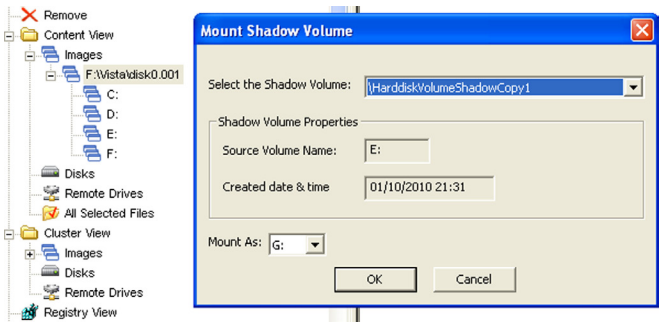

FIGURE 3.15

ProDiscover Mount Shadow Volume dialog.

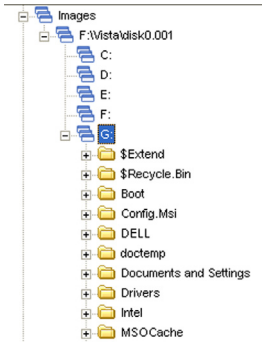

FIGURE 3.16

VSC mounted in ProDiscover.

I do is completely reinstall the operating system) as they include, at the minimum, a Dell maintenance partition. Once the "Mount Shadow Volume…" functionality was selected, ProDiscover located the volume where the VSCs reside (in this case, E:\) and populated a drop-down list with the available VSCs. There are a total of seven VSCs available, and as we progress through the VSCs in the drop-down list, the "Created date & time" will change to reflect the correct date and time for the selected VSC. Finally, whichever VSC was selected will be added to the "Content View" display as the G:\ volume (note that C:\ through F:\ are already populated).

When I clicked "Okay," the selected VSC was mounted as the G:\ volume within the ProDiscover Content View interface. I then clicked on the volume letter, and the files were populated within the volume, as illustrated in Figure 3.16.

At this point, there's a great deal I can do with the available data in the VSCs. For example, I can navigate to the Users folder and select files to be copied out of

the project for deeper examination, run ProScripts, etc. It all happened within the blink of an eye, right there while I was sitting in front of my analysis system. Along with the other functionality inherent to ProDiscover (parsing Vista and Windows 7 Recycle Bin files, locating and parsing email archives, etc.), being able to mount and access the VSCs puts a whole new level of capabilities in the hands of the analyst.

TIP

Other Image File Formats

Throughout this chapter so far we've discussed accessing VSCs with raw/dd-format image files. As such, I'm sure that at some point someone's going to ask, "…but I have an EnCase.EOx format image file, and it's compressed … what do I do?" or "I have a snapshot of a VMWare virtual machine/vmdk file, how can I use the VHD method?" Those questions are easy to answer. For the expert witness format images (such as acquired via EnCase), you can open the image in FTK Imager and reacquire it to raw/dd format, making yourself a working copy of the image file. You can do the same thing with the vmdk file and then use vhdtool.exe to prepare the image for mounting, or search for tools to convert the vmdk file to vhd file format.

Windows 8

Throughout this chapter, we discussed VSCs from the perspective of Windows Vista and Windows 7, and what we've discussed also applies to Windows 2008 R2. But, as I've been asked numerous times, what about Windows 8?

Windows 8 appears to take a bit of a different approach to VSCs than Vista and Windows 7. I say this because while I've been working on writing and updating this edition, I have a Windows 8 system on which I can try the various techniques and tools discussed in this chapter and see how they work. I started by logging into my Windows 8 system, launching a command prompt with administrator privileges, and running the "vssadmin" command to list the available VSCs, and found that there were several listed (numbered 3 through 6) in the output of the command. I then launched FTK Imager on the system, added the C:\ volume as an evidence item, and found four difference files in the "System Volume Information" folder.

My next step was to use FTK Imager to acquire an image of the C:\ volume, and then use vhdtool.exe to convert the image file to VHD format (as described early in this chapter, this process simply adds a footer to the image file, without changing the extension). When I attempted to attach the newly created VHD to my Windows 7 analysis system, it was accessible as the G:\ volume, but I received a message stating that the volume would need to be formatted before it could be used. When I attempted to attach the VHD on the Windows 8 system, I received an error message stating "A virtual disk support provider for the specified file was not found." However, changing the file extension from ".001" to ".vhd" allowed the image file to be mounted as a read-only volume, in accordance with the VHD method procedure described earlier in this chapter.

Once the VHD was mounted as the H:\ volume, I was able to run the "vssadmin list shadows /for=h:" command to obtain a list of available VSCs within the image. I then used the *mlink* command to create a symbolic link from the oldest available VSC (do not forget the trailing slash!!) to a folder on the D:\ volume for my system. From that point, I was able to access the logical files within the VSC, and having demonstrated that, I used the *rmdir* command to remove the symbolic link.

TIP

VHD File Extension

When mounting an acquired image as a VHD on a Windows 8 system, be sure to change the file extension to ".vhd" after using vhdtool.exe to convert the raw/dd-format image file to a VHD format.

For the sake of simply being complete, I ran additional tests and attempted to access VSCs within a Windows 7 image from the Windows 8 analysis system, via the VHD method discussed in this chapter. Again, as long as I changed the file extension to ".vhd" after I ran the vhdtool.exe command to convert the image file, everything worked just fine, and I was able to list and access the VSCs available within the Windows 7 image.

SUMMARY

While VSCs may initially be somewhat mysterious to many analysts, they do provide a very valuable resource with respect to historical data. VSCs can be accessed via a number of methods, depending upon how you're accessing them (i.e., on a live system or within an acquired image).

Keep in mind, however, that accessing VSCs on live systems can be a bit tricky, in that you have to move quickly and decisively, as VSCs are subject to the FIFO cycle, and you may be attempting gather information from a VSC that gets deleted during that process.

Finally, remember to be extremely careful with respect to how you access files within VSCs, both on live systems and within acquired images, as double-clicking the wrong file can lead to your analysis system being infected or compromised.

Reference

[1] Carvey H. Windows registry forensics. Burlington, MA: Syngress Publishing, Inc.; 2011.