Forensic Analysis of Spoliation Cases

Part 2 of a 2 Part Series: Windows Examinations

by Steve Bunting

As a quick refresher, in part one of this two-part series, we defined spoliation as the intentional, reckless, or negligent withholding, hiding, altering, fabricating, or destroying of evidence relevant to a legal proceeding. Thus, in simple terms, withholding, deleting, or hiding evidence are forms of spoliation. To add legal specificity to this definition, we quoted an Arkansas court ruling, referencing Black's Law Dictionary, in which they defined spoliation as "the intentional destruction of evidence and when established, [the] fact finder may draw [an] inference that [the] evidence destroyed was unfavorable to [the] party responsible for its spoliation." Thus, spoliation carries with it a very specific penalty in that the aggrieved party may legally infer the destroyed evidence was unfavorable, which often has a devastating impact on the party who destroyed the evidence.

Often, penalties far exceed the unfavorable inference. Sometimes the court simply rules in favor of the aggrieved party. Courts also levy heavy fines and costs when spoliation is established. Finally, as false statements under oath often accompany spoliation, there is often the possibility of criminal proceedings stemming from the act.

Yet despite these penalties, many times parties to litigation tempt fate by deleting, hiding, or withholding digital evidence. It seems so easy to do and they don't foresee getting caught. Fortunately for the aggrieved

party, however, the digital landscape is laden with evidence of spoliation when it occurs. It's difficult to use files and programs and subsequently delete all traces of that activity. Thus, a competent forensic examiner who is actively searching for this evidence will most often discover this vital evidence and establish to the court that spoliation did occur. It is to this purpose, assisting the digital forensic examiner with finding evidence of spoliation, that this article was written. In part one, we focused on the Macintosh operating system, while in part two, this article, we will focus on the Windows operating system, specifically Windows 10.

We often find that there are degrees of spoliation that correspond with the technical skills that the offending party possesses. Those with simple skills delete files and emails and often accompany that activity with a commercial evidence or artifact cleaner or evidence removing tool. The more sophisticated delete files and emails as well and, like their less sophisticated counterpart, they use commercial tools to eliminate artifacts. However, they often take added steps to remove artifacts not often removed by tools and they also make attempts to cover their use of tools. You will often find they will use command line tools or other advanced deletion techniques. Perhaps they will even attempt to delete the system restore points. Regardless of their sophistication, they will leave behind evidence and also traces of their activities.

In the previous article, I cited a physical world analogy for the spoliation process, which involved using a branch to cover one's tracks in the snow. While one covers their tracks, the branch itself leaves behind a pattern in the snow. Often the branch does not cover up the entire track. Perhaps faint imprints remain behind. And finally, when one is done using the branch to erase the footprints, the offender is left holding the branch, or tool. What does one do with it? You have to discard it and any traces of its use, leaving behind more tracks in the snow. It is quite difficult to hide all traces or tracks, especially from a skilled forensic examiner.

In our examples in the article, we are using snippets or situations from actual cases on which we have done spoliation consultations. The situations exemplified herein have involved persons who have downloaded copyright protected movies using various torrent applications. The parties owning the digital rights to these movies contracted with a monitoring service. The defendants were subsequently discovered and identified as having downloaded and shared copyrighted movies. Thus, the discovery process will seek to obtain their computers and related equipment for forensic examination.

Let's begin our focus with the litigant who has average skills or abilities. This person will typically go on a file deletion spree to remove files and programs that are incriminating. This will occur shortly after the O.S. moment that we previously discussed. This is the moment when the party realizes that litigation is imminent and is faced

with the duty to preserve evidence for discovery, in which they'll be required to turn over their data to the opposing party.

Of course after deleting data, they start worrying about getting caught and feel the need to cover their tracks even more, using one of the more popular Windows cleanup tools to remove evidence from their computers. They use their browser to search for and download the tool. They typically use the first one that they encounter in the search results and best of all it is free. They use it and then they uninstall it. In their mind, they've cleaned up their trail. They provide an image of their hard drive for discovery, signing a sworn document acknowledging they've fully complied with discovery, provided all relevant data and haven't deleted or destroyed data. The words will vary, but the essence remains constant.

This party does not understand at all how data is stored on modern operating systems and does not understand at all the complexities of the modern digital landscape. Unfortunately for them, they will soon find out. Whether they deleted through the Recycle Bin and emptying it or whether they deleted directly using the Shift – Delete key combination, the cleaning tools will do a good job of clearing out the recycle bin. The most common cleaning tool encountered, currently, is CCleaner from Piriform.com. It is listed first among the best five in online reviews and appears first during most searches for such tools. When it is downloaded from the internet, it will reside in the downloads folder as ccsetup###.exe, where ### will represent the version number. This is important as often when uninstalling the tool after its use, the downloaded setup file is often overlooked, as seen in EnCase 8 in Figure 1 below.

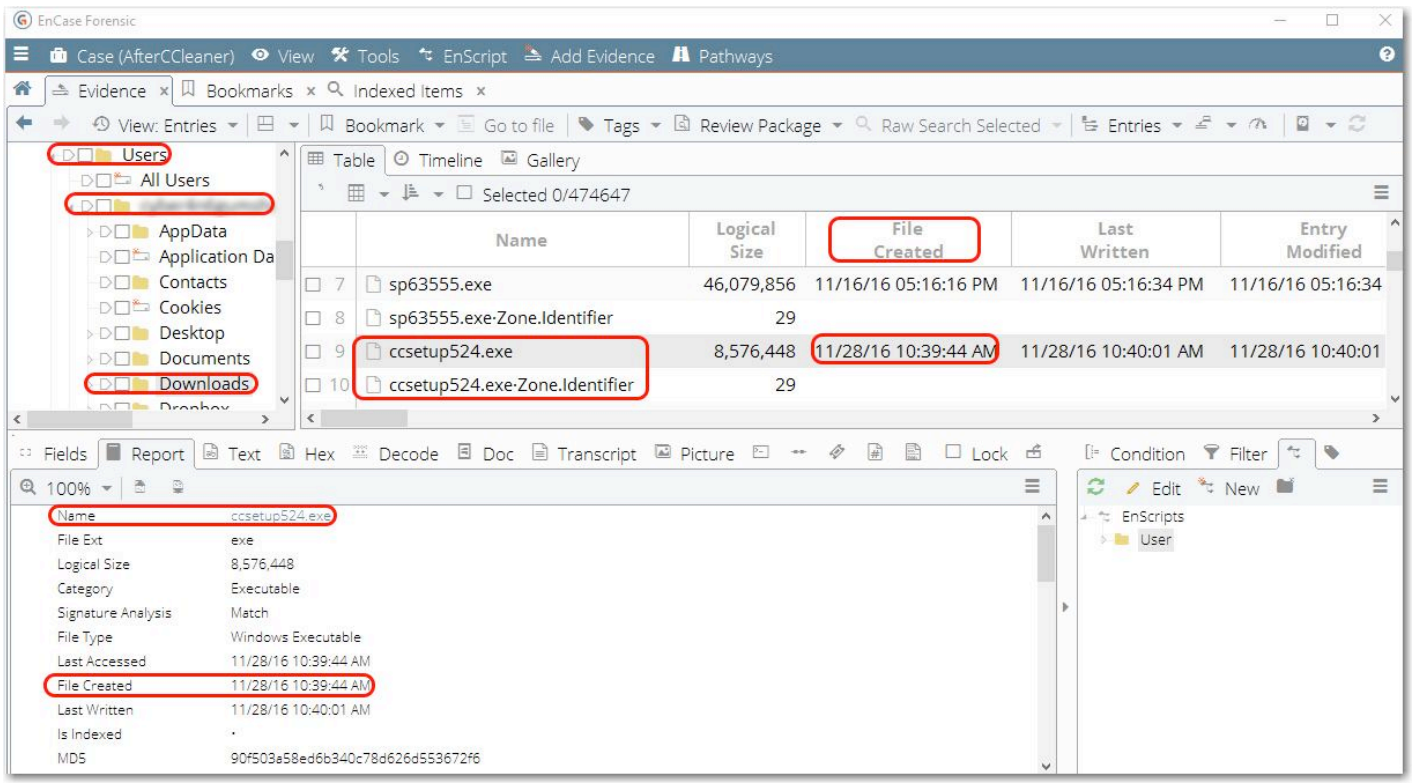


Figure 1 - Remnants of CCLEAN setup file in Download folder

If you find the remnants of the setup download file, its creation date will be quite important. If it was downloaded, i.e. created, after the duty to preserve attached, that certainly stands as evidence of intent to destroy or cover up data. This tool, CClean, does a fine job of removing many evidentiary artifacts to be certain, especially internet browsing artifacts but no tool is perfect and leaves its own distinctive set of tracks as well. This tool, when used to wipe the free space of your disk, creates a unique set of artifacts that I call the 'zzzz' folders and files. These are temporary files and folders created as it overwrites data. I'm not certain why they do so in this manner, but they do and have done so consistently since at least 2007. I like that kind of consistency.

After the wiping is completed, CClean deletes these files and folders, but that is akin to being left holding that branch after wiping footprints in the snow. Where does the branch go? You can hide the branch, but it is still there somewhere. The same thing goes for the 'zzzz' folders and files. You can delete them, but not wipe them again with the same tool, as yet another set of 'zzzz' folder and files will be created and deleted. As these files and folders are deleted, they are unseen by the offending party, but quite visible to the forensic examiner in the root of the volume wiped, as seen below in Figure 2.

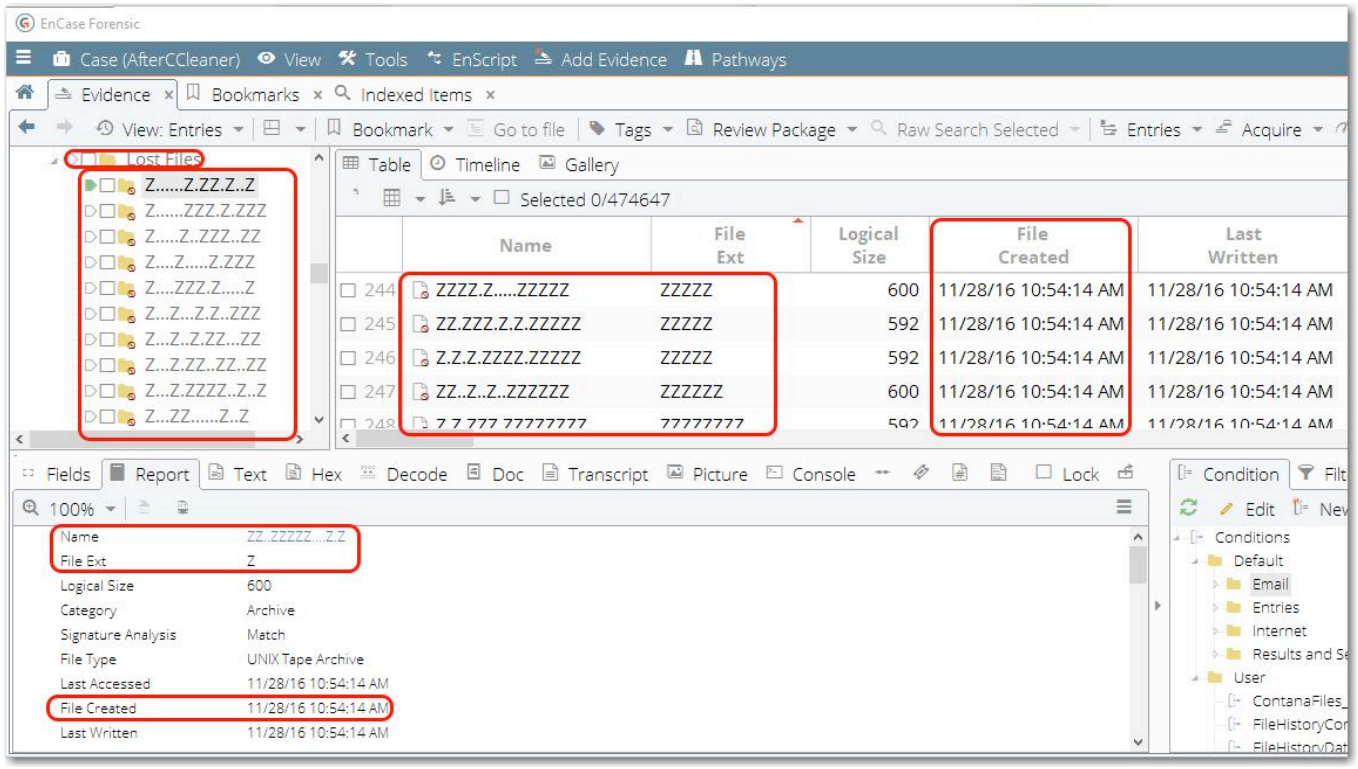


Figure 2 - CClean leaves 'zzzz' files and folders as artifacts

The icon on these files and folders shows they are deleted. Their presence in the "Lost Files" category indicates they are orphaned files with no parent folder. There are approximately 10,000 files and folders created with various iterations of "Z" as both a file name and file extensions, interspersed with dots. They were created within a period of less than two minutes during the wiping process and the creation timestamp clearly indicates

the time at which the wiping occurred. This timestamp also becomes important evidence, especially if it was after the duty to preserve data attached.

Thus far we've seen that CClean leaves a very unique trail in its wake, one that clearly links to CClean. Along with it, we have a timestamp left by those unique 'zzzz" files and folders, both of which are significant pieces of evidence. I mentioned early that CClean is good, but not perfect, meaning it doesn't remove everything of import. Specifically, it doesn't touch the data contained in the Restore Points, meaning an entire set of duplicate data often remains behind, hidden in the Volume Shadow Copies in a location that even the administrator can't normally access.

Initially, (Windows XP and earlier) Restore Points did not capture user data, but rather just system settings needed to recover the operating system. Starting with Windows Vista, the process evolved into a Volume Shadow Copy Service or VSS that operates at the block level and not the file level. The volume shadow copies are made before Windows updates, before unsigned drivers are installed, periodically, and can even be user created. When the Restore Points evolved into the VSS, they captured not only system data, but user data. Starting, therefore, with Vista an option within Windows Explorer was to "Restore previous versions" of a given file or folder, as seen below in Figure 3.

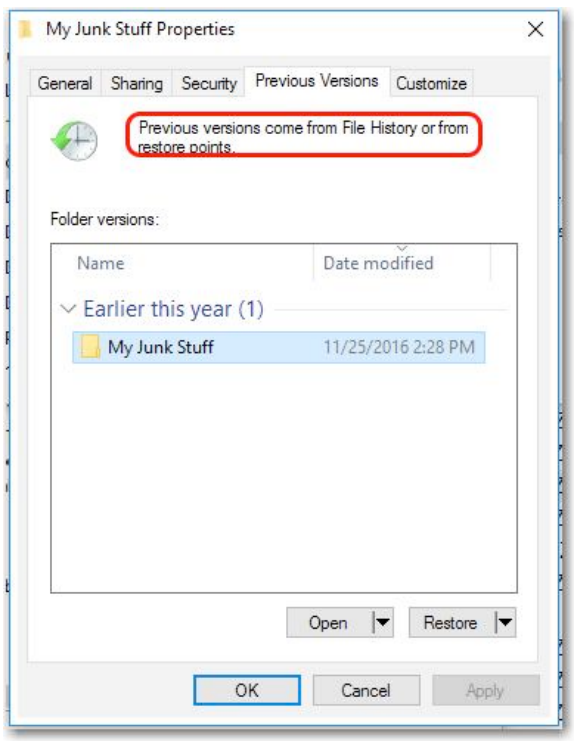


Figure 3 - Restore Previous Versions option in Windows Explorer

In the Previous Versions Window, you can see that I have data available in a folder dating back almost a month from when this was captured, meaning it could easily contain data that has since been deleted, which is a forensic goldmine. As Windows has evolved, the Volume Shadow Copy Service has done likewise. Starting with Windows 10, an interesting note appears in the Previous Versions tab, which reads "Previous versions come from the File History or from restore points." Currently, with Windows 10, I'm not seeing any previous versions coming from Restore Points, only from File History, which is running on this particular system. However, there is yet another version of "My Junk Stuff" in the Restore Points that can be seen below in Figure 4. In this case, I'm using ShadowExplorer to browse the Restore Points on this machine.

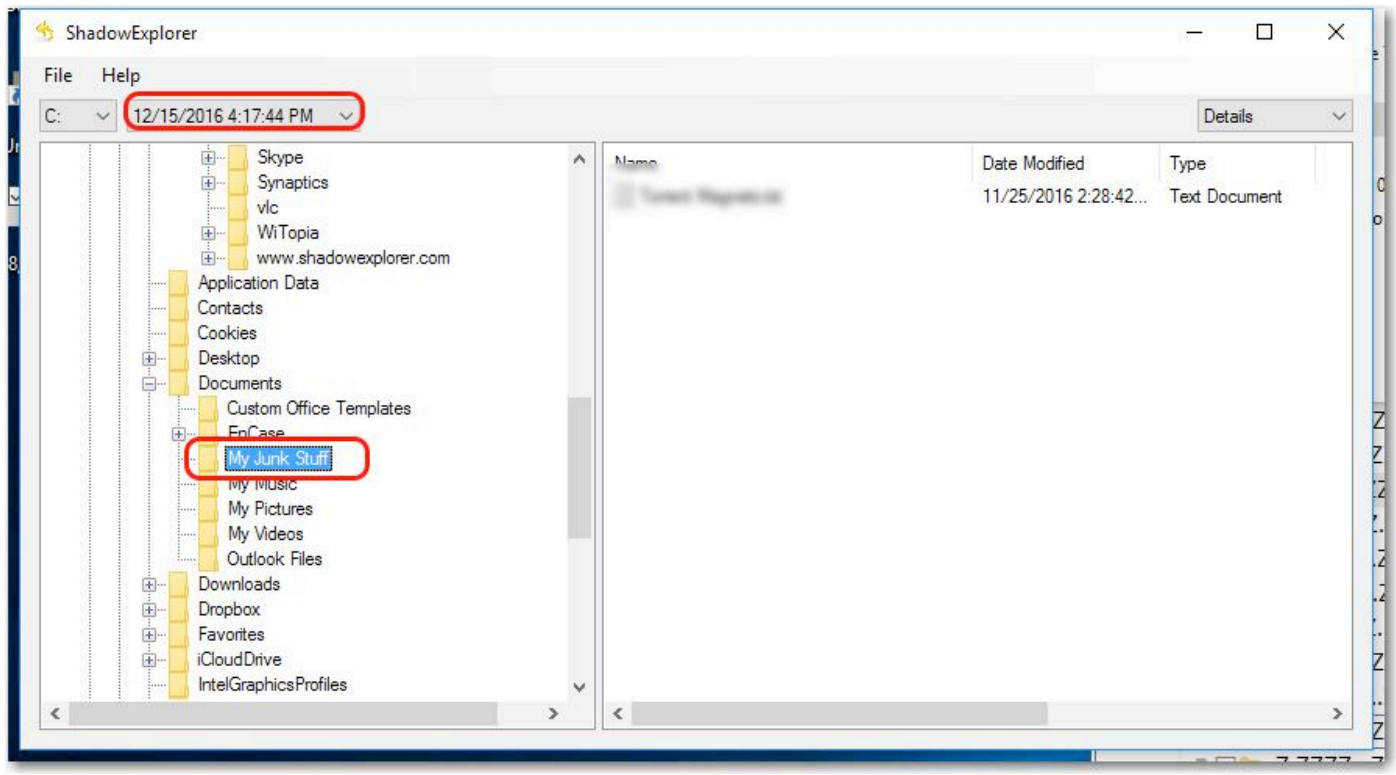


Figure 4 - Restore Point has version not showing in Previous Versions on Windows 10

If you have File History enabled (similar to Time Machine on OS X), you can have a version of deleted files in both File History and in Restore Points. We discussed Restore Points somewhat, let's talk about File History. Windows 8 introduced File History as a feature. The user simply opens up Settings and goes to Update & Security > Backup. At that point, you configure a drive that you either leave attached or periodically attached as a backup for your user files. If you detach the File History drive, very much like OSX, Windows File History still maintains a hidden cache backup of your files as they are changed, modified, etc. The user is unaware that this hidden backup is occurring and CClean does not touch it. It is located at Users/UserName/AppData/Local/Microsoft/Windows/FileHistory.

The FileHistory folder contains a Configuration and a Data folder. The Data folder contains all the backup data in numbered folders that are easily sequenced and identified by their timestamps as to when they were created. Figure 5, below, shows the path and contents. The AppData folder is normally hidden and the average user will never see or know about this hidden backup.

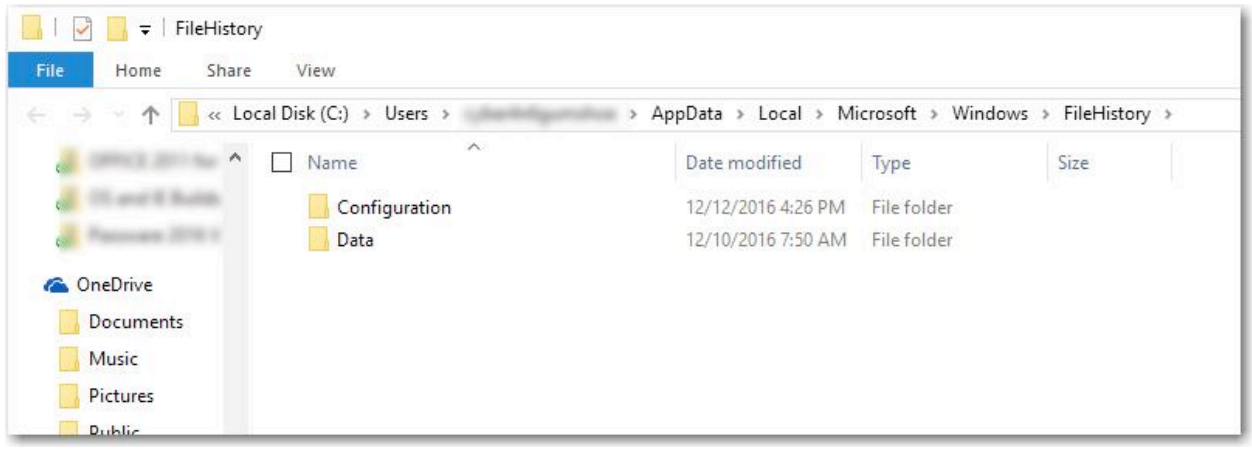


Figure 5 - Location of hidden FileHistory folder

To view or parse the FileHistory, you need only any file browsing utility. There is no encryption or special scheme. The file and folder names are stored in their normal hierarchical format. To view Volume Shadow Copies, most mainstream forensic tools process them automatically or with separate scripts. ShadowExplorer can allow you to view them on a live system or you can mount a write protected physical image and use ShadowExplorer in that fashion. The important takeaway here is that most cleaner tools will not alter FileHistory backups nor will they alter Volume Shadow Copies (Restore Points). If an offending party has deleted files, often those files can be found in these locations. When found in these locations, you usually have clear evidence of spoliation.

As previously mentioned, the user must enable File History, and often does, but the user is normally unaware that File History is active, caching in a hidden directory, even though the File History drive is not mounted. The really good news is that Volume Shadow Copy service is enabled by default, which results in a forensic goldmine of information that is normally going to be present unless the user is knowledgeable enough to know of its existence and disable it long before they engage in nefarious activities.

When our case involves a certain file or file type, in this case videos, we often look to the default location for storage of those files, which in this case would be the user's "Videos" folder. Because the offending party didn't wish to show us his unlawfully downloaded video files, this folder was found to be empty, as shown in Figure 6 below.

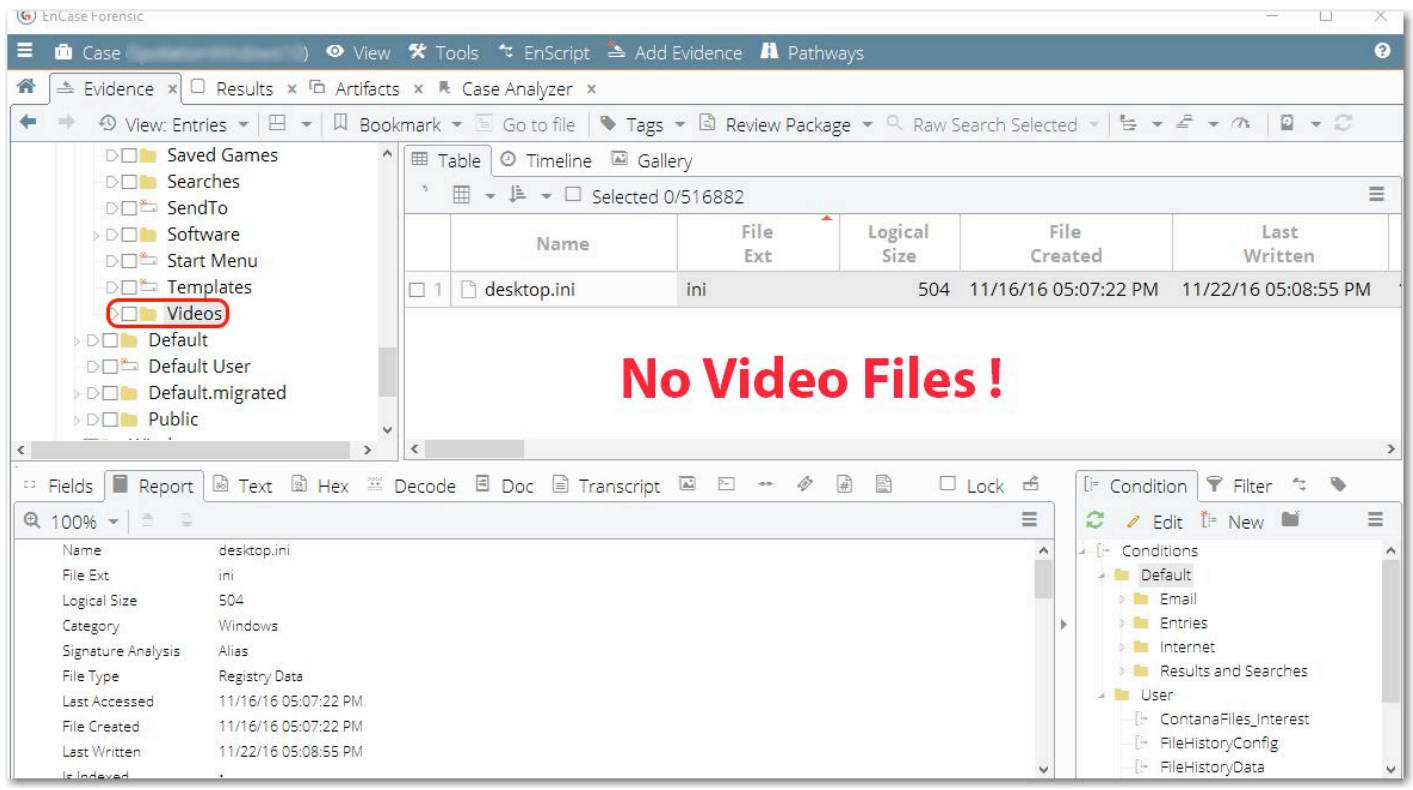


Figure 6 - Video folder is empty, aside from the desktop.ini file

When we see this folder empty when the case is all about videos, we become, quite naturally, suspicious. Windows is a prolific producer and user of link files. When a user clicks on files to open them, a link file is typically created in the user's 'recent' folder. The link file contains metadata about the target file, such as its MAC times, path where it was located, etc. If we search the link files, wherever they now exist (recent folder, in the restore points, unallocated clusters, etc) for certain key strings, such as the path to the user's "videos" folder and the file extensions of various popular video formats (.wmv, .avi, .mkv, .m4v, .mp4, etc.), we are often rewarded with some significant findings, as shown below in Figure 7.

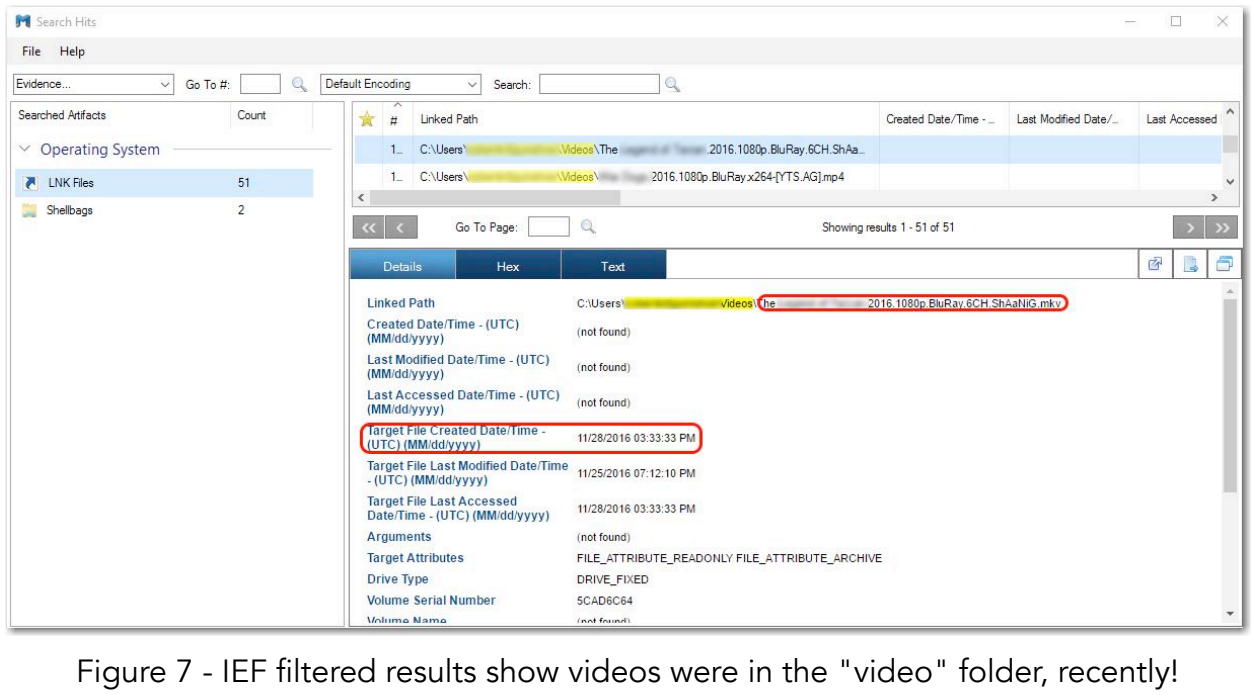


Figure 7 - IEF filtered results show videos were in the "video" folder, recently!

In Figure 7, we see the results of a search for the strings we described in the previous paragraph (path to user's video folder and common video file extensions). The tool used for this search was Magnet Forensics Internet Evidence Finder (IEF). We are showing two video files that were recently in the "videos" folder and there were many. This image was made on November 30, 2016 and these videos were present in this folder on November 28, 2016, just two days prior. Clearly we have evidence of spoliation, as those video files were deleted within the past two days. This is quite common to find, which is a mad dash to delete files just before turning over their data!

Since we know there were video files in the "videos" folder on November 28th, it stands to reason that any Restore Point made shortly after the files were present and before their deletion just may have copies of those deleted videos. Let's take a closer look at all the details shown in the lower right portion of Figure 7, shown below. The details tab contains some incredibly important information. This the detail information for a link file.

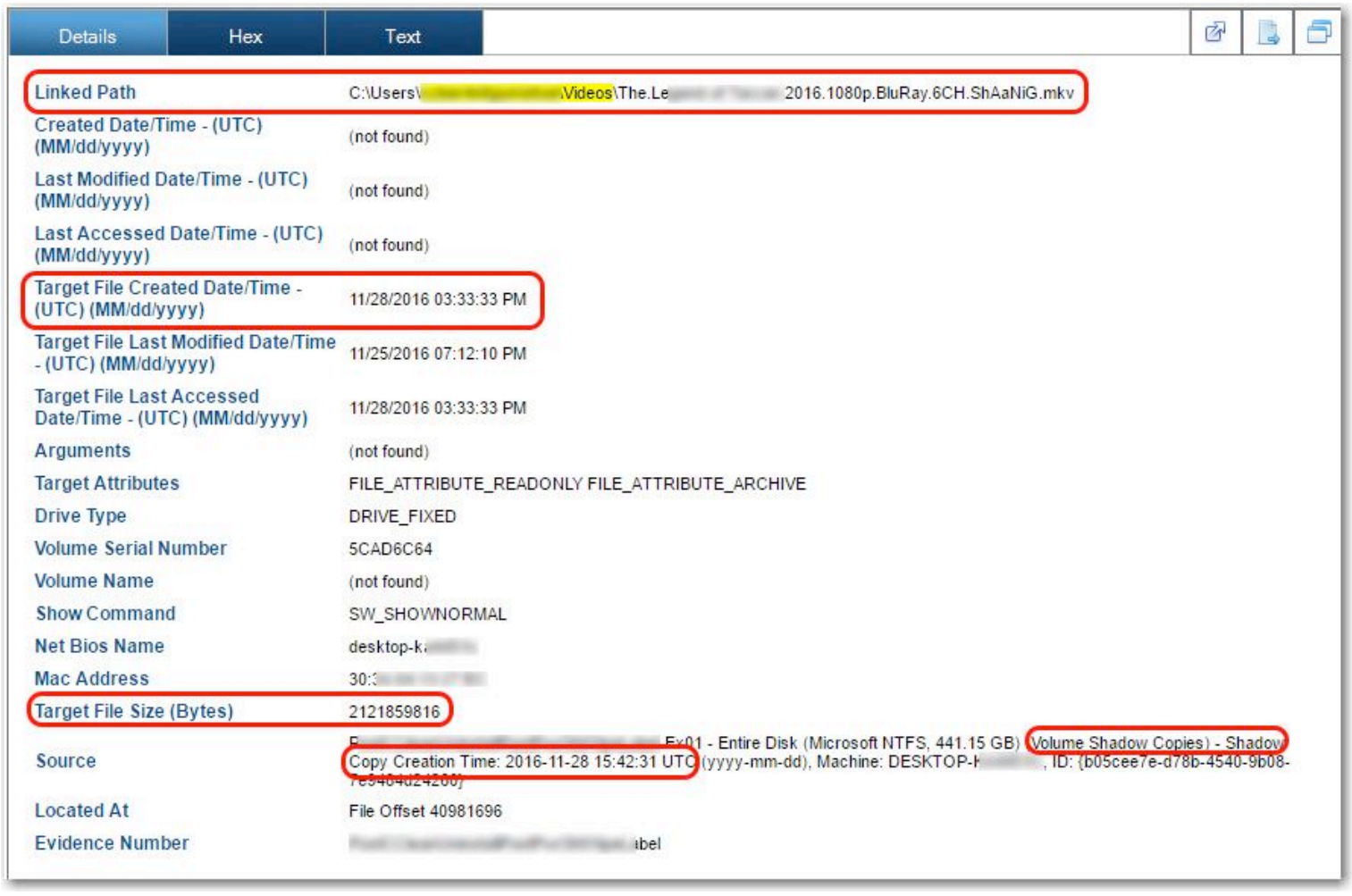


Figure 8 - IEF Detail pane contains important metadata concerning the target file

First, consider the source, which is nearer the bottom and circled in red. This link file was found inside a Volume Shadow Copy and was untouched by CClean! Note that the timestamp is provided for the Shadow Copy, which is November 28, 2016 at app. 3:42 pm UTC. Since this Shadow Copy was made after the creation time of the target file in question, there's a good chance that this Shadow Copy contains the video file that was subsequently deleted. Often the easiest way to view a Volume Shadow Copy is to attach the suspect drive, better yet a clone, via a write-blocker to a Windows machine running ShadowExplorer, which we will do for our purposes here. You could also use any one of the mainstream forensic tools or you could use Reconnoitre from Sanderson Forensics, which was designed to forensically examine Volume Shadow Copies.

Using ShadowExplorer, in Figure 9 below, we see that there are several videos in the "videos" folder on November 28, 2016 when this Volume Shadow Copy was made, but none in this same folder on November 30, 2016 when we made our image. Further, we see the target file name from Figure 8, above, appearing in the Shadow Copy. Further, if you look at the file size in Figure 8, above and circled in red (2,121,859,816 bytes), you will find it matches the file size in the Shadow Copy, shown below in Figure 9, which is 2,072,128.72656 KB and rounds up to 2,072,129 KB (as shown below). The MAC times also match.

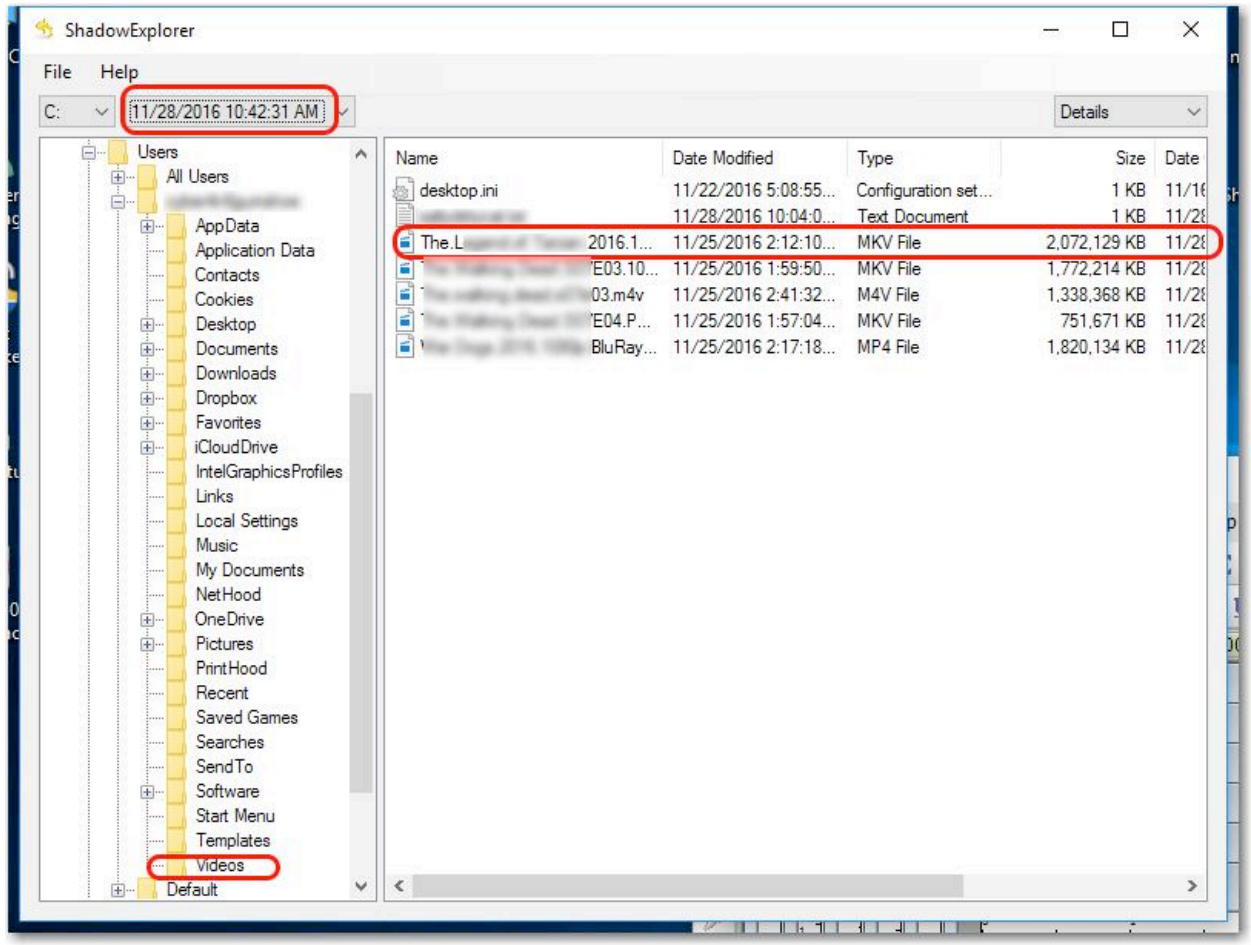


Figure 9 - ShadowExplorer view of a Shadow Copy containing video files that were deleted!

We have solid evidence establishing that video files were present in the "videos" folder two days prior to imaging, but removed before imaging. Spoliation has occurred and we have proof using link files and Volume Shadow Copies. We've also established attempts, thus far, to use CClean to remove evidence We found the remnants of its installer in the "Downloads" folder and clear tracks left behind in the form of 'zzzz' files and folders that result from wiping free space.

Another location to find solid evidence that certain programs have been used in the past is the Windows Prefetch folder. When a program is launched, Windows tracks metadata about that program to speed up its launch in the future. Among other data, it tracks the number of times a program was launched, from which path, and a timestamp indicating its last run time.

In our case, even though CCleaner has been uninstalled, there is still a prefetch file for it. Even if CClean removed prefetch entries for uninstalled programs, it could not do so for itself after it was uninstalled. This situation, again, it like sitting at the end of the trail holding the branch after you've wiped your trail clean. What do you do with the branch? In this case, you'd have to know it existed and manually remove it, however that would do nothing to the prefetch files in the Volume Shadow Copies, about which the average user knows not,

nor can they be easily accessed. You can readily see how difficult it becomes to hide or remove all tracks in the snow!

Figure 10, below, shows the parsed prefetch data for the program 'ccleaner64.exe', which is the full executable for the CClean program. It has been run eighttimes during the 28th and 29th of November, 2016, the day just before it was turned over for imaging. It's not too hard to arrive at the conclusion that the offending party intentionally destroyed data.

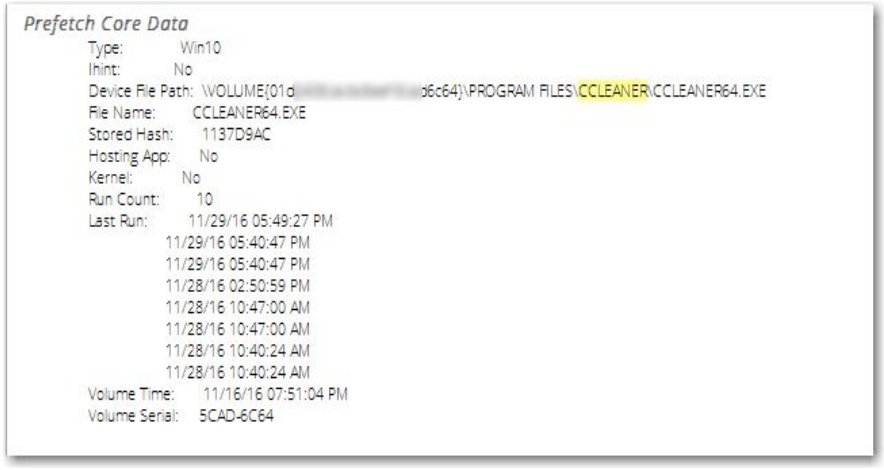


Figure 10 - Prefetch data parsed in EnCase 8 showing CCleaner64.exe was run 8 times and when

While we are on the topic of prefetch, the target of this litigation is alleged to have used torrent services to download copyright protected movies, but there are no torrent programs currently installed. Now that we know that prefetch files are excellent artifacts for determining if programs have been run in the past, regardless if they have since been uninstalled, let's look at the prefetch folder for torrent programs. Figure 11, below, shows an EnCase 8 view of the Prefetch folder. We sorted by filename and have located four prefetch files associated with the program 'utorrent.exe', a very popular torrent application. By this alone, we know that this program has been installed and used in the past, despite the fact that it is no longer present.

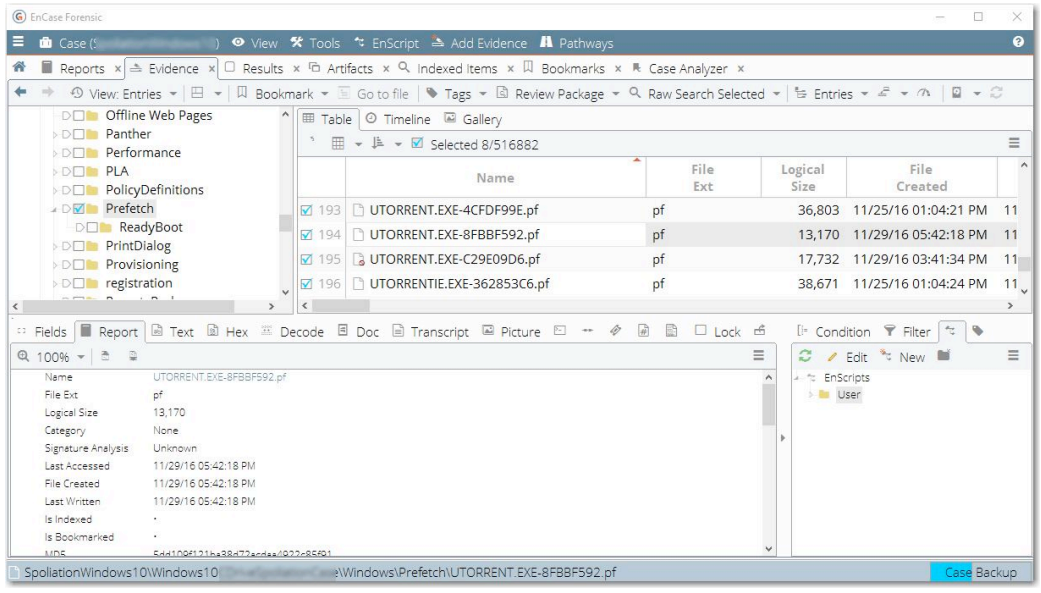


Figure 11 - EnCase 8 view of the Windows Prefetch folder showing utorrent.exe prefetch files

EnCase uses EnScripts to further process and parse data. Simon Key has written one, which is free, that does an excellent job of parsing prefetch data. By selecting only these four prefetch files, we obtain a near instantaneous result, which is shown in our Bookmarks view. The parsed data is shown below in Figure 12. We can easily see that the 'utorrent' program was run eight times and up until the day before the system was imaged. Since it is no longer present on the day of imaging and was running the day prior, we have additional evidence of spoliation, as the offending party was under a clear duty to preserve data on the day prior to imaging and certainly well before.

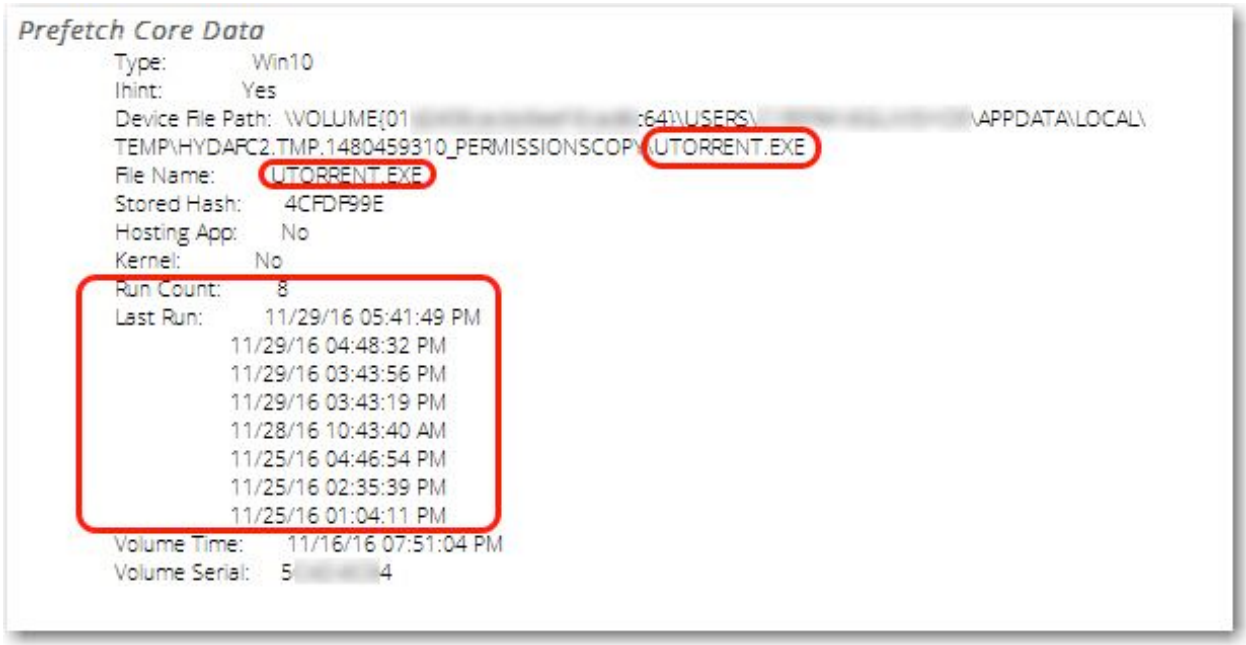


Figure 12 - EnCase 8 EnScript report shows prefetch data for utorrent program

There is nothing wrong with having and using 'utorrent' or any other torrent application. They are excellent programs and have many legitimate uses. Many large files are distributed by torrents legitimately every hour of every day. The problem comes into play when they are used to download copyrighted materials without the owner's permission or images depicting the sexual exploitation of children. In our case, the program was used to download copyright protected movies and the offending party chose to destroy the program used to do so.

Before we move on to the more advanced user, there is another aspect of spoliation that occurs when the accused party does not turn over all of the media and has also failed to advise the other party of the existence of these devices. Link files often reveal these devices to the careful examiner. We previously discussed that Windows creates a link file to a target when the user clicks on it and opens it. This link file contains metadata concerning the target file, which, among other metadata, tracks MAC times, location / path, MAC address of the volume, and file size. It is the location that becomes important when looking for other involved devices, especially those that have not been disclosed and made subject to discovery. Figure 13, below, shows a movie file that is located on a Removable Device that was assigned to drive letter "I" at that time. When we make a

discovery such as this, our employing counsel will file a motion seeking additional discovery of this device, among other motions. Since the parsed information includes the Volume Name and Serial Number, this motion can be quite specific.

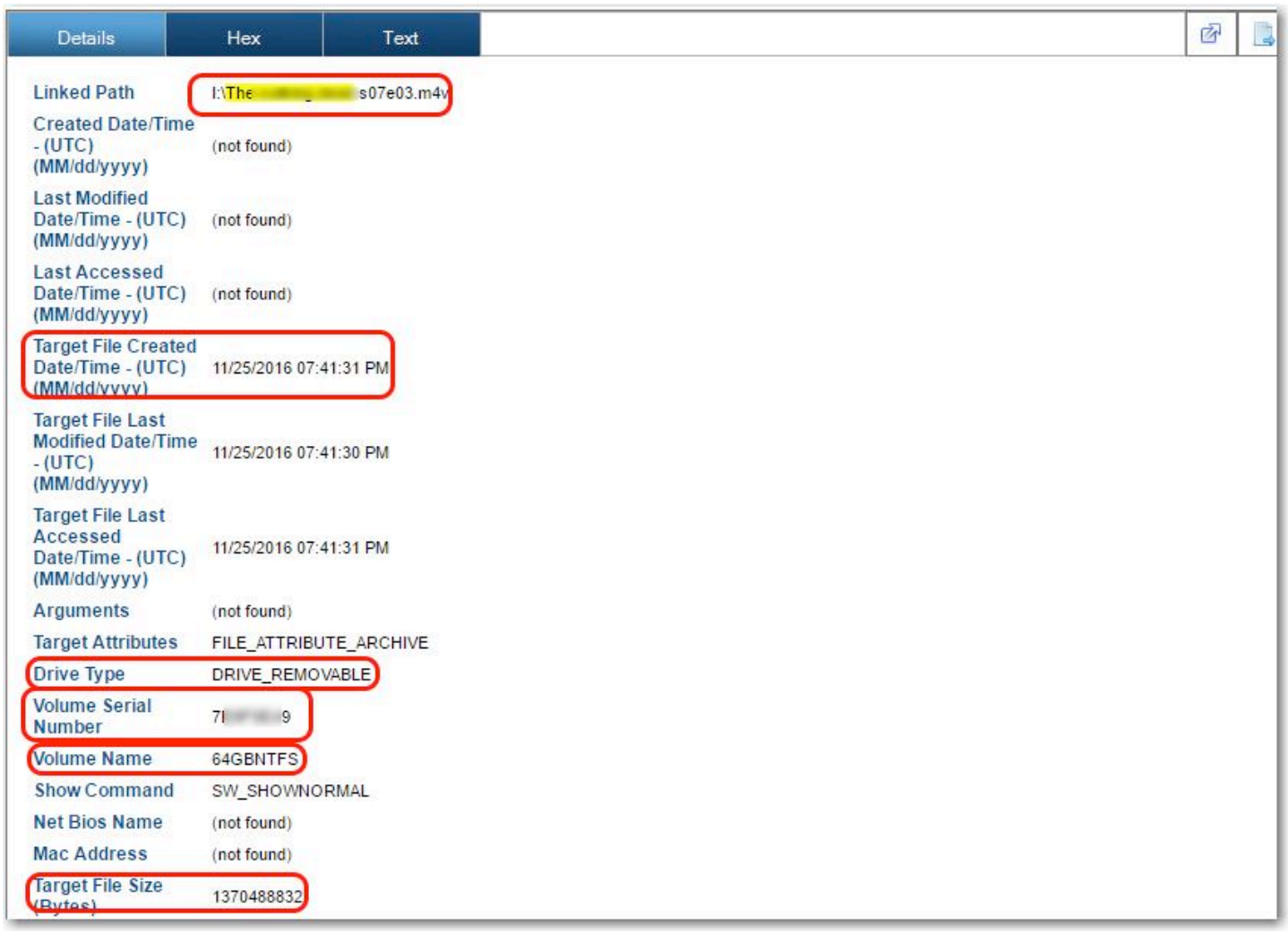


Figure 13 - Parsed link file revealing a relevant file on Removable Media, along with specifics (Vol Name & Serial #)

Figure 14, below, depicts yet another parsed link file in which the location is a network resource. By the non-public IP address (10.0.1.13), we know it is an internal network device and it also contains a shared folder named "movies", which is quite relevant to the case at hand. Furthermore, the target file on that shared resource was a movie that was involved in the litigation, making it quite relevant, but nevertheless not disclosed during discovery. Depending on other factors and the judge, these non-disclosures may or may not constitute spoliation alone. The offending party could argue that it was an unintentional oversight and may prevail in that argument. However, in the case we are seeing here, where there are blatant and intentional acts of spoliation, this would likely be treated as additional evidence to add to the accumulating list of evidence.

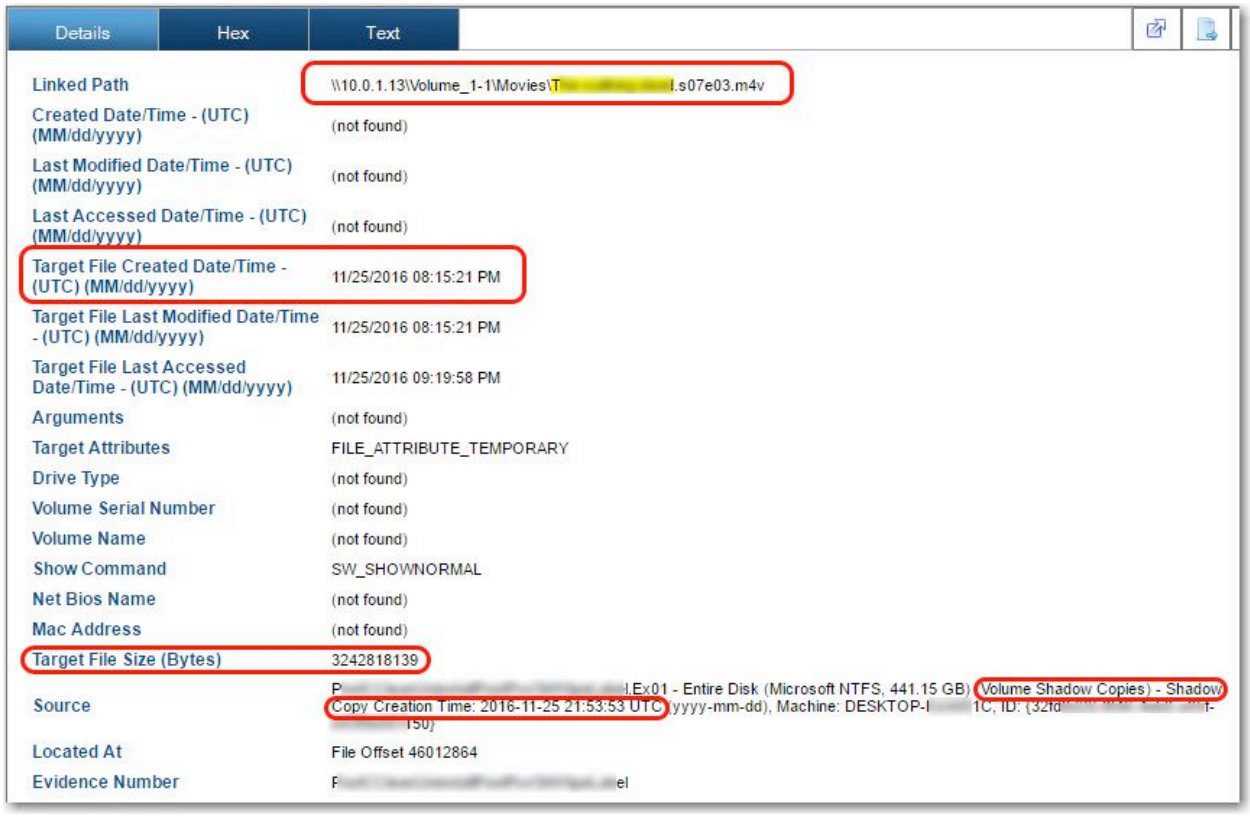


Figure 14 - IEF parsed link file metadata showing a relevant target file on a network resource

Before we leave the topic of other devices, let's consider what Shellbags might reveal about other networked resources. Shellbags have been around since Windows XP, but were an underutilized artifact when they first appeared on the Windows operating system. Among other things, Shellbags track views, sizes, and positions of a folder window when viewed through Windows Explorer. In as much as these views include My Network Places and Removable Drives, this should pique your interest. Since they may contain references to resources no longer available or not in hand, in the case of discovery, their importance becomes obvious. If their importance is not obvious by verbal description, then one look at the data should remove all doubt. Figure 15, below, shows Shellbags parsed by Magnet Forensic's IEF. The day before ("Last Explored Date") this machine was imaged, the user was connected to several network resources on the same MAC-PRO computer. The paths are clearly delineated along with a timestamp indicating when those folders were last explored or browsed using Windows Explorer. Since that machine was not disclosed during discovery, we have another discovery violation that builds the ever-growing list in our spoliation claim.

#	Path	Last Explored Date/...	Mode
1	My Network Places:	11/29/2016 10:31:30 ...	
2	My Network Places:	11/29/2016 01:28:00 ...	
3	My Network Places:\MAC-PRO [redacted]\Desktop\		Details
4	My Network Places:\MAC-PRO [redacted]\		Details
5	My Network Places:\MAC-PRO [redacted]\6TBStripedRAID\	11/29/2016 09:43:07 ...	Details
6	My Network Places:\MAC-PRO [redacted]\Desktop\Binary Computations\		Details
7	My Network Places:\MAC-PRO [redacted]\Desktop\VMSharedFolder\	11/29/2016 06:53:43 ...	Details
8	My Network Places:\MAC-PRO [redacted]\Pictures\	11/29/2016 01:29:41 ...	Icons
9	My Network Places:\MAC-PRO [redacted]\6TBStripedRAID\Input\	11/29/2016 10:31:30 ...	Details
10	Control Panel:		Tiles

Figure 15 - IEF Shellbag results showing connections to a networked resource with timestamps

For the careful observer, there is a folder in Figure 15 that is described as "VMSharedFolder". While one can't be certain, it suggests to me that "VM" might refer to a virtual machine, so when this machine is eventually included in discovery, I'd certainly be looking for virtual machines on that device. I also see a 6TB Striped Raid folder in Figure 15. 6TBs is a lot of storage and I would strongly expect this drive to contain a large cache of videos.

So far, we covered the offending party that possesses minimal or average technical skills. What might a more technically savvy user do that we thus far have not seen? Typically, that user will use command line tools to dig deeper and delete data. Also, that user is typically aware of Restore Points and will try to delete them. Finally, that user is likely to back up data as they realize the import of such practices. Let's look first at the command line activity.

On Windows, one may simply use the command prompt (cmd.exe) or they may prefer the Windows PowerShell. PowerShell is a command line shell with an associated scripting language built on the .NET Framework. As the name would imply, PowerShell is for power users, the IT crowd. For those who float in between Windows and Unix environments, it is particularly nice as it recognizes many Unix shell commands.

For purposes of deleting files and folders and wiping, using built-in commands (cipher), the two are no different in effect. If one were using the simple command line, the keystrokes you enter would be remembered only for the duration of the shell, as they are retained only in RAM. If you are lucky, perhaps the RAM was written to the hiberfil.sys when the machine went to sleep. Otherwise, you'll likely not find any trace of commands entered into this command shell.

If PowerShell were used, the keystrokes you entered would be stored a little differently. If you type in the command "history" and press enter, you will see the history of commands for this session of the shell. If you close the PowerShell, reopen the PowerShell, and type history again, you will not see the history from the previous session.

You might surmise from this that PowerShell, like its lesser cousin 'cmd.exe', stores its history in RAM also. However, if you use the up and down cursor to step through your history, you'll find that PowerShell retains the history for this purpose for multiple sessions, which suggests that the history is actually retained in a file. In Figure 16, below, you see that the PowerShell console history is in fact retained in a file. Interestingly enough, the command "clear-history" does clear the history for the current session, but still does not delete any of the content of this file, making it one of those forensic nuggets when investigating the activity of a 'power user'.

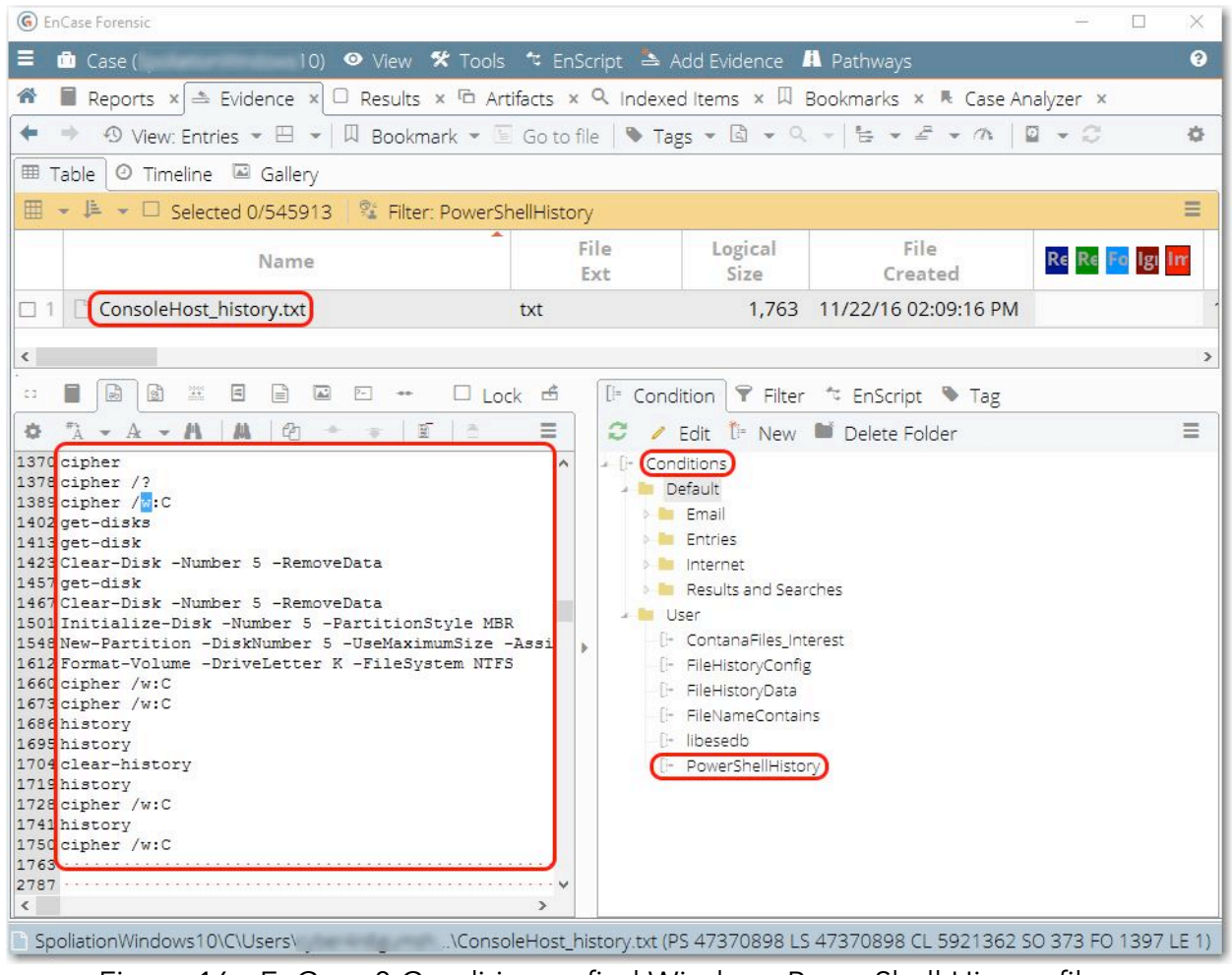


Figure 16 - EnCase 8 Condition to find Windows PowerShell History file

In the lower left you will see an extensive history of commands entered in the Windows PowerShell. It would seem the only way to remove this history would be to manually edit this file. Of course, you would have to know it existed and what it does, which isn't very likely. In the above EnCase 8 view, I have created a User condition

to immediately locate any PowerShell History by simply setting a condition to filter for Name matches "ConsoleHost_history.txt". It will find it for any user, as it is based on file name. The path to this file is "Users \UserName\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt" You will note that among these commands, you will see the 'cipher' command, specifically 'cipher /w:C'. What this command does is to wipe the free space of this C drive. There's nothing intuitive or obvious about this feature by its name. You'd simply have to know what it does to realize the import of this finding. Figure 17, below, shows this command running in PowerShell.

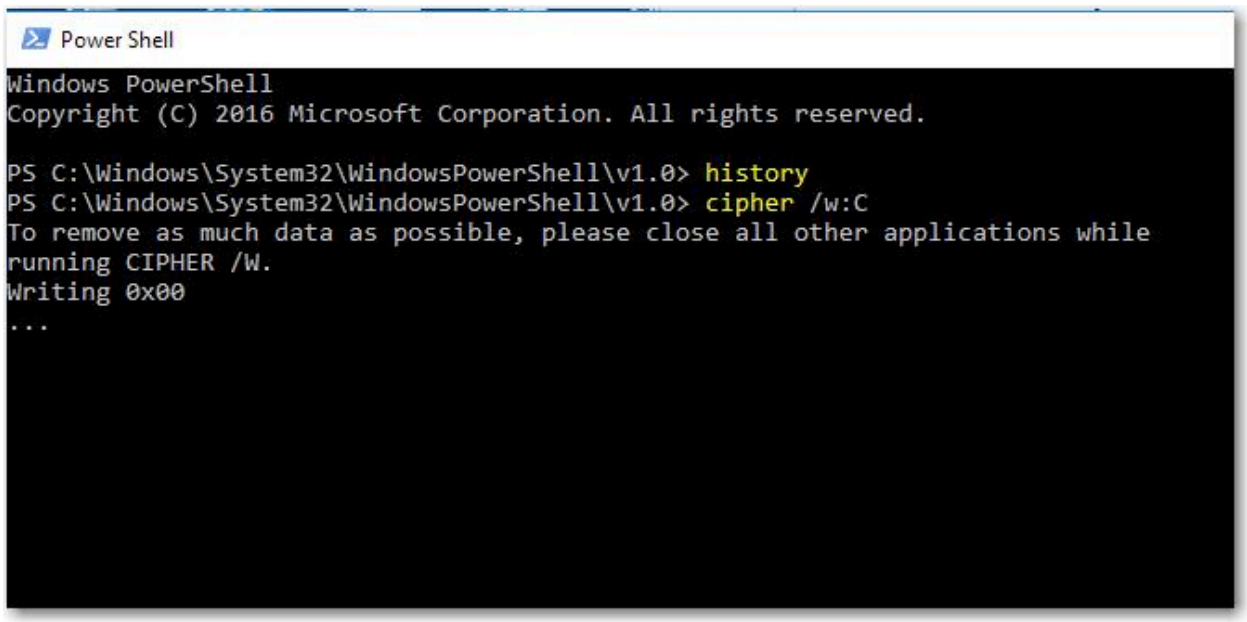


Figure 17 - Cipher /w:C running in PowerShell - This command wipes free space

It is important to understand what the various commands do that you find in this file. If you don't know, it is easy enough to research them. You will note that they do not have timestamps on the commands. Opening and closing the PowerShell will not change the timestamp on this file. Entering a command, however, does change the last modified timestamp. The best you can do, therefore, is to have a timestamp for the very last command in the list.

Beyond that, you may have to extrapolate or approximate based on commands in the history that can be associated with objects that have timestamps related to that activity. For example, a format command will have no timestamp in the history file, but if you can locate the volume that was formatted, you could determine from that volume when it was created or formatted. That could then establish time ranges for other commands between the format command and the last command in the list. Sometimes time ranges are more than sufficient for the purpose.

An advanced user with some technical background will likely be aware of the Volume Shadow Copy Service and Restore Points. Normally, Restore Points are good to have and can rescue your system from Windows issues, hence a technical person would likely want to keep them enabled. I certainly do! However, when that technical person finds themselves in a legal mess and is faced with turning over an image of their computer, the last thing they want to have on their system are copies of files they have intentionally deleted. Hence, they will disable the system protection feature in the Restore Systems configuration menu (a tab on the System Properties menu). As this feature is toggled on and off by a registration setting, the key for this feature is at HKLM \SOFTWARE\Microsoft\WindowsNT\CurrentVersion\SystemRestore\RESessionInterval. If it is enabled, it is set to 1. If it is disabled, it is set to 2. Figure 18, below, shows the feature enabled and the corresponding registry setting. Figure 19, also below, shows the feature disabled and the corresponding registry setting. Keep in mind, by default, this feature is normally enabled when Windows is installed, which is good for digital forensics examiners.

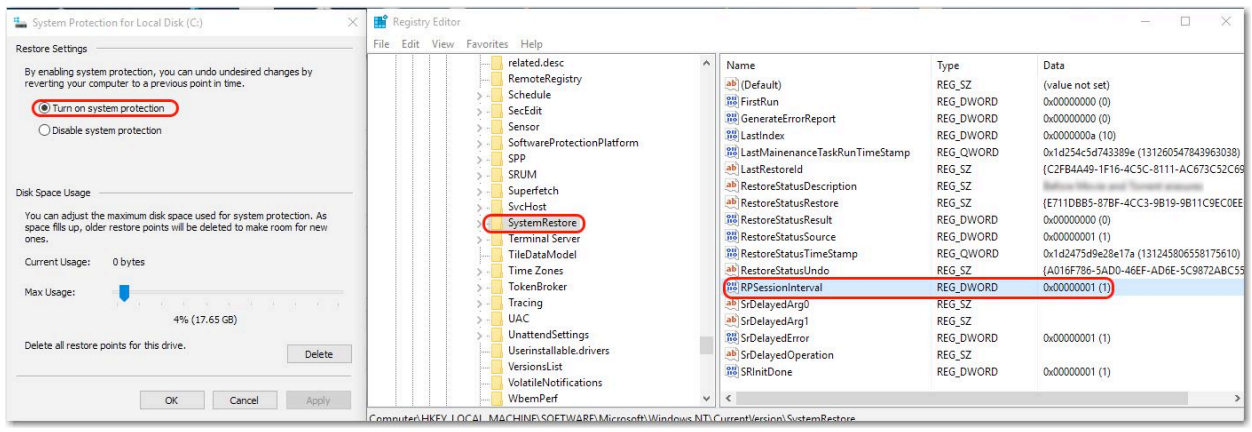


Figure 18 - System protection is enabled with corresponding registry settings

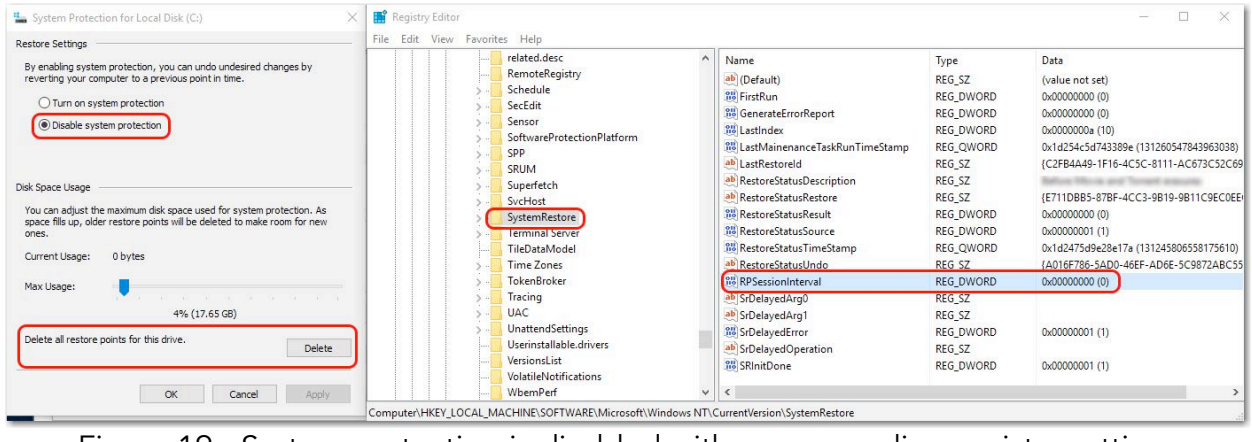


Figure 19 - System protection is disabled with corresponding registry settings

In Figure 19, above, you can see that the system protection has been disabled. Circled in red in the lower left is the option to "Delete all restore points for this drive", which is what will happen when the advanced user is attempting to cover his or her tracks. While they will succeed in deleting the Shadow Copy data (Restore

Points), especially if they wipe the drive after doing so, they will leave very significant tracks that will show that they deleted the restore points. First, let's look at some of the features of the restore point data structures.

Figure 20, below, shows a snippet from ShadowExplorer on the right. It is showing a list of the restore points that are available for the C: drive. The times for when the restore points were created are displayed in local time. The left side of the image shows a list of what are called "OnDiskSnapshotProp" files , located in "System Volume Information\SPP\OnlineMetadataCache" These are prefixed with a GUID, but it is important to note the timestamps correspond with those shown on the right in the ShadowExplorer, except they are displayed in UTC. The take-away here is that there is one of these property files for each restore point. The second take-away is that when the restore points are deleted, these property files will NOT be deleted. Specifically, they will remain behind and intact.

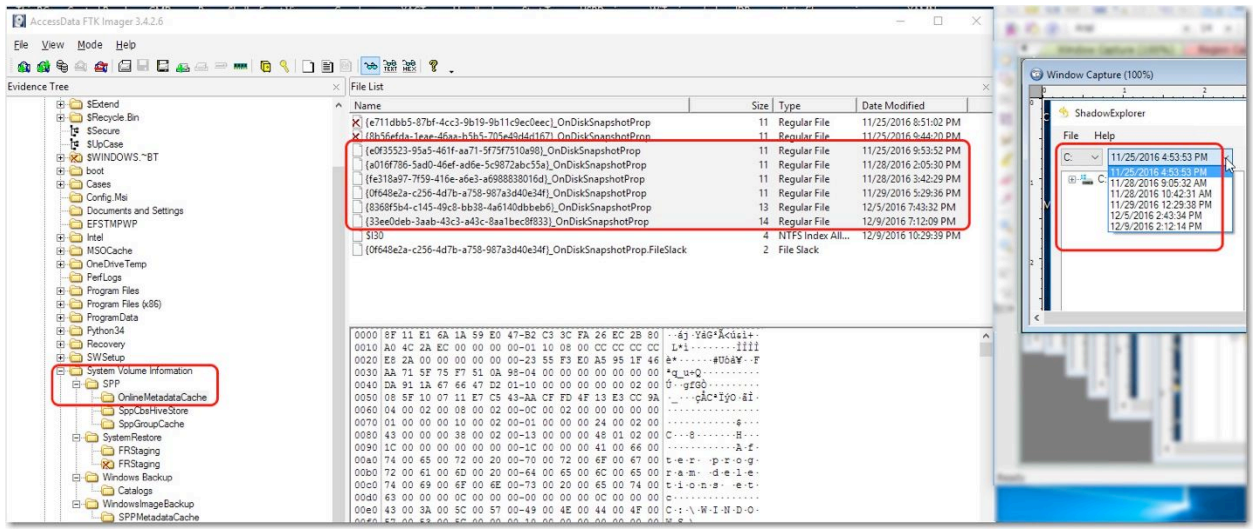


Figure 20 - An OnDiskSnapShotProp exists for each Restore Point and will not be deleted if Restore Points deleted

Figure 21, below, shows restore points that have been deleted. For clarity, they are circled in red. You'll see that the timestamps of these deleted restore points correspond with the "OnDiskSnapshotProp" files shown in Figure 20, above. If the Restore Points have been deleted and there hasn't been much activity on the system since the deletion, you may still see the MFT entries for those deleted files as seen in Figure 21, below. If there is significant activity or there are cleanup activities undertaken, you may not see the deleted restore point entries, however, you will see the "OnDiskSnapshotProp" files shown in Figure 20, above. As there should be corresponding restore points to match these property files, the absence of those restore point files is evidence that they were deleted. If there were restored points in existence during the period when the duty to preserve attached and are since gone, you have evidence of spoliation.

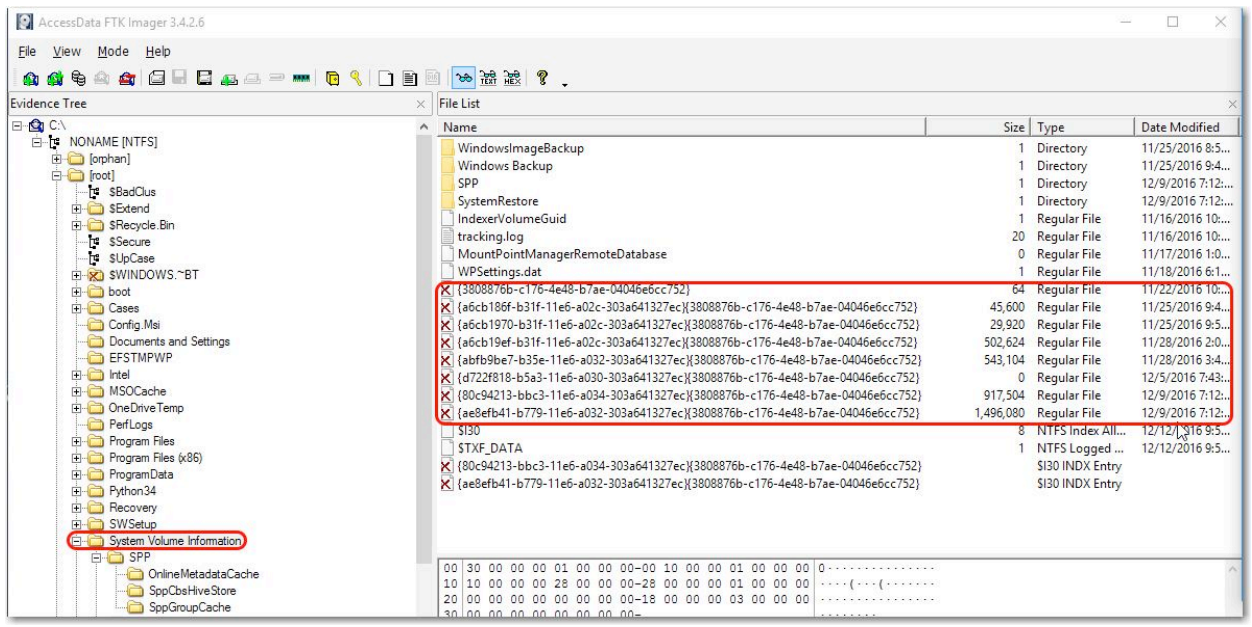


Figure 21 - FTK Imager view showing deleted Restore Points

This may be hard to understand, so it is better to reinforce this conclusion with other evidence that supports the previous existence of these restore points. To do so, we turn to our Windows Event Logs. If you filter the System logs for Event ID 98, you will get a list of the system checks on the volumes on the system, which includes the Volume Shadow Copies, aka Restore Points. Figure 21, below, shows Event ID 98 for a specific time, reporting the Shadow Copy #3 was healthy. This establishes that there were in fact Volume Shadow Copies present. There is no abstract thinking involved. It is clearly stated. You can find all of these up to a point where they are no longer being reported, because they were deleted. This will dovetail with and support your findings regarding the presence of "OnDiskSnapshotProp" files with no corresponding restore point files.

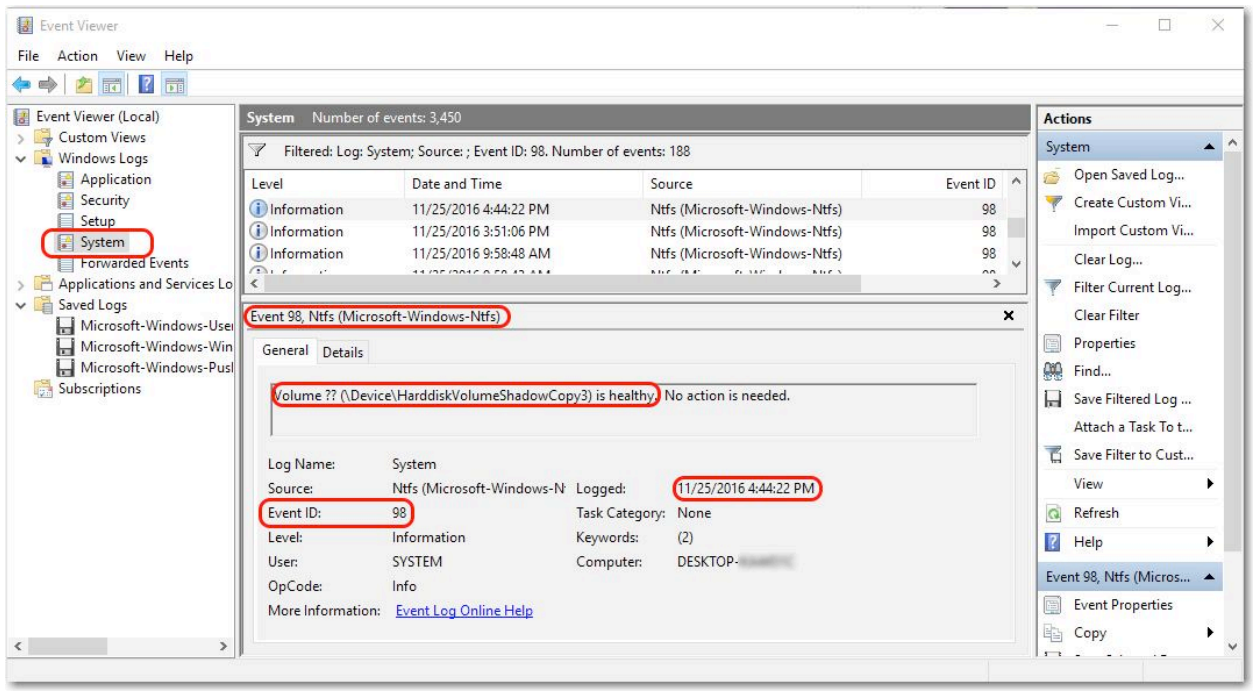


Figure 22 -System Event Logs filtered for Event ID 98

Thus far, we have seen how an advanced user may use command line tools and how they may be tracked with history files. We have also seen how deleting restore points leaves considerable evidence. Let's circle back to the command line for a moment. If the traditional command line was used (cmd.exe), we won't expect a history file, but don't forgot to look for things like prefetch files for cmd.exe. They won't tell you what commands were run within the shell, but at least you'll know cmd.exe was in use, when and how often. Figure 23, below, shows that cmd.exe was run significantly in the two days preceding the imaging of this machine.

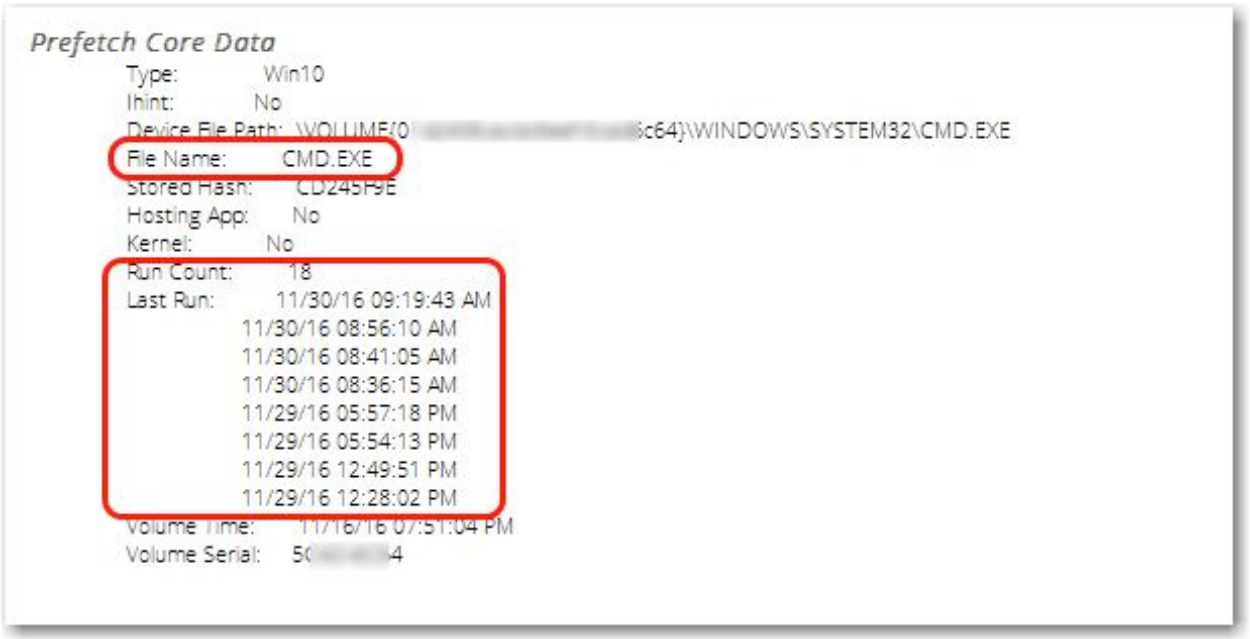


Figure 23 - EnCase 8 EnScript report for cmd.exe prefetch metadata

It also pays to know some of the cleanup tools for the command line. One of the most popular and free ones is "sdelete.exe" or secure delete. This does not ship with Windows, but is available as a free download from Microsoft Technet. Thus, if it is present, it was intentionally downloaded and placed on the machine. If your evidence is indexed, a quick search will find it. Figure 24, below, shows the results for a search for 'sdelete'. In a short amount of space, you can see where it was downloaded as a zip file and the zip was clicked to open it, thus creating the link file. Next it was deployed in different locations. Finally, you see when the prefetch file was created at the time it was first launched, which happened to be the evening before the day of imaging. Last minute attempts to delete data are actually quite common to find.

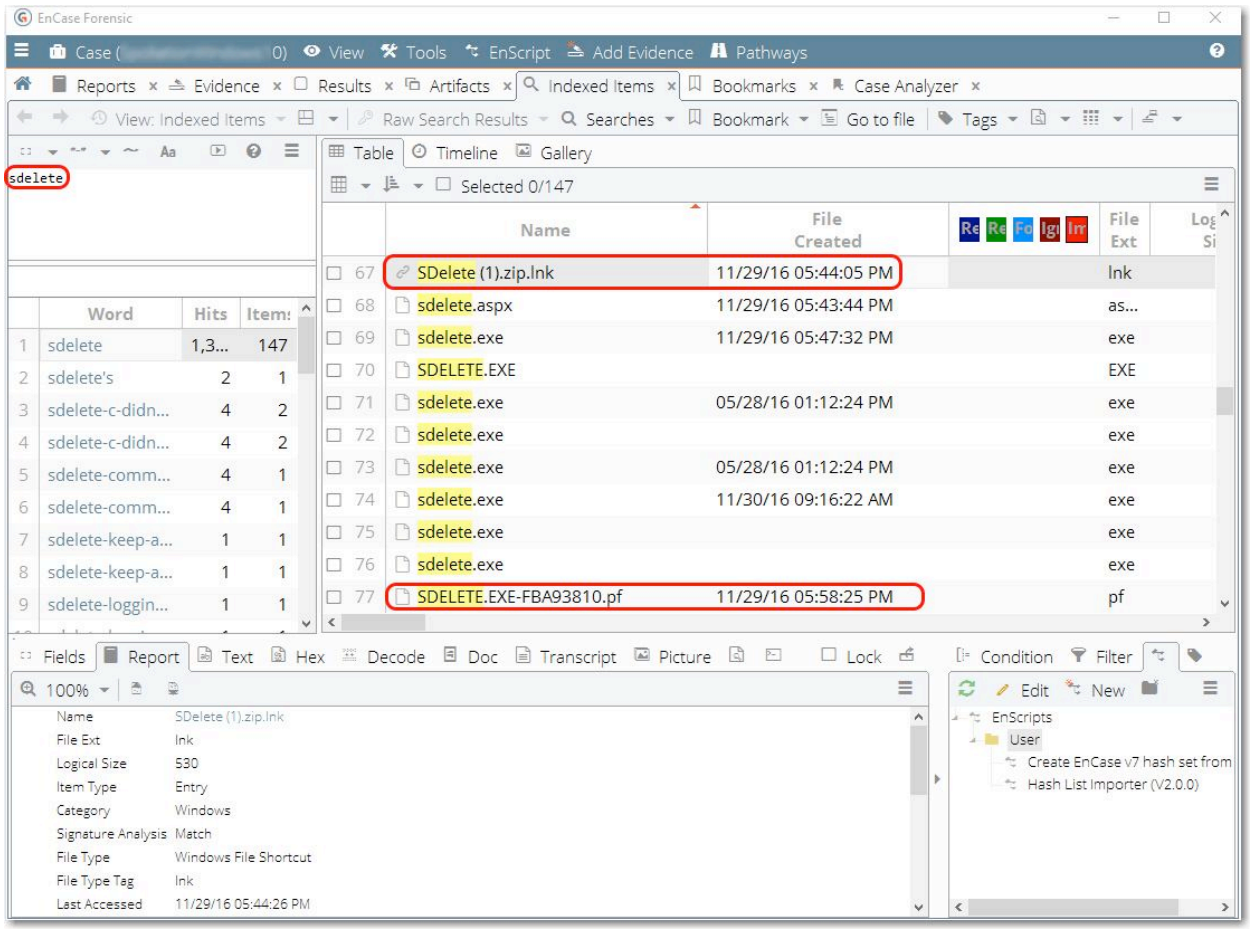


Figure 24 - "sdelete" is found on the system and used the day before the system was imaged.

Earlier we mentioned that there's one more thing that an advanced user will do. They are nearly all backup fanatics. They realize the importance of backups and they have them, only they won't produce them for discovery. I've actually had a case where they backed up their entire machine, cleaned it up for discovery, and turned it over for discovery. Once the backups were discovered it was clear they intended to restore their machine back to its pre-discovery state once the machine was returned to them.

We previously mentioned that Windows 10 (starting with Windows 8) offers File History as a feature. It's actually quite nice and is very much like Time Machine on Macintosh systems. If enabled, the path to the configuration file is: Users/UserName/AppData/Local/Microsoft/Windows/FileHistory/Configuration. Inside this folder are two XML files, Config1.xml and Config2.xml, which appear as a redundancy (matching hash values). Near the bottom of the XML file is key named "Target". The Target Name is the Volume Name and Drive Letter for the FileHistory backup drive. If it has not been provided as part of discovery, you can now ask for it by name.

Since the last time the FileHistory drive was plugged in (mounted), FileHistory is maintained in the folder: Users/UserName/AppData/Local/Microsoft/Windows/FileHistory/Data. You may get lucky and find copies of files that have been deleted in this location. If present, they will be easy to find as they exist in their original hierarchical format.

Macrium Reflect is probably one of the most popular and free backup utilities in use. It can integrate into the boot menu such that it is a startup option for backup or recovery. If you find this program installed, you had better start looking for the backup files. Very likely, the offending party will not have provided them during discovery. You may have to look at a list of attached USB devices, link files, jump lists, etc., to locate the backup drive and any timestamps that might indicate its last use. Clearly, you want to bring it to the discovery table. Figure 25, below, shows Macrium on the Start Menu. Figure 26, also below, shows a Macrium backup file, with the extension ".mrimg", which is clearly a string for which to search, along with Macrium, as it may land you on a link file pointing to a backup drive.

Figure 25 - Macrium installed and showing on the Start Menu

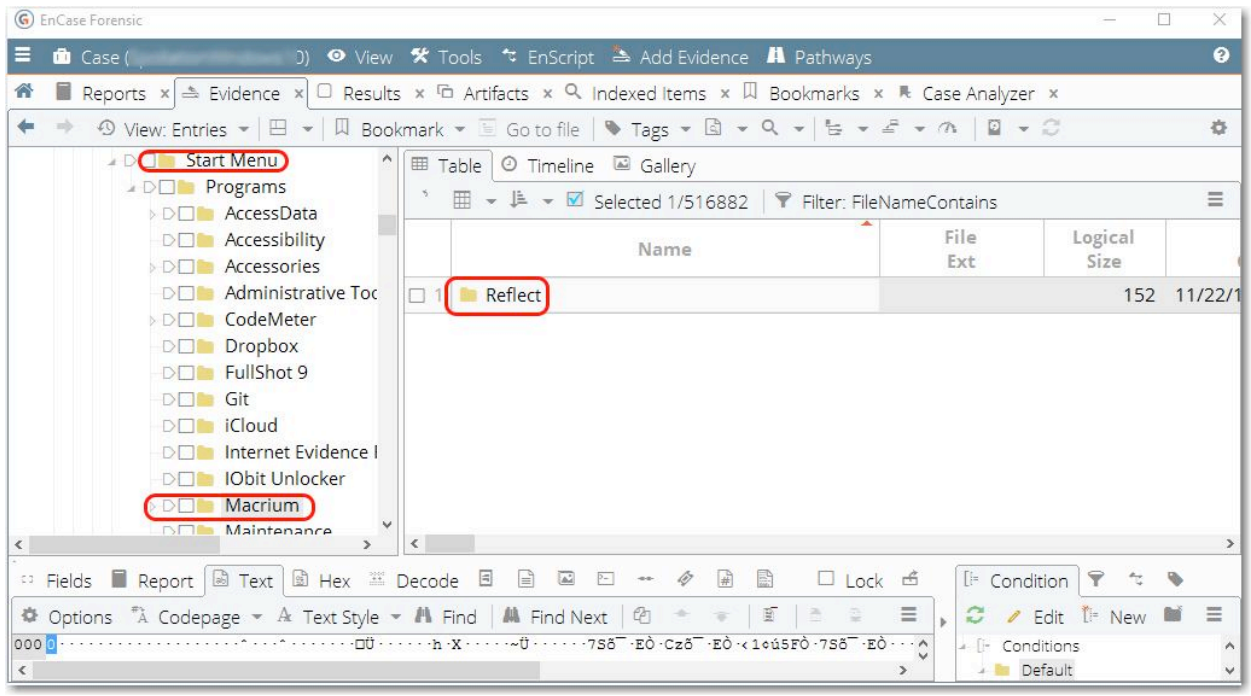
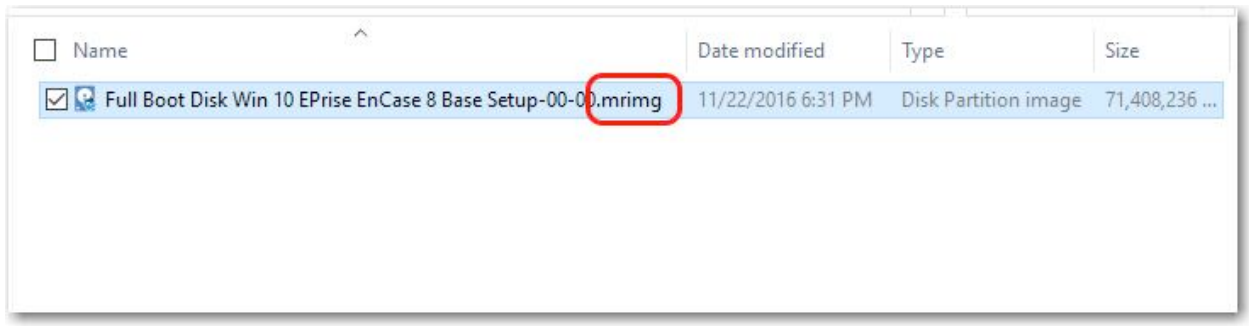


Figure 26 - Macrium file backup with an extension of ".mrimg"



Thus far, we've covered a lot of ground when it comes to finding evidence of spoliation and recovering copies of files that were intentionally deleted. Sometimes we may not actually recover the deleted files, but the evidence that they once existed and were deleted is often overpowering. It is truly hard to interact with a modern operating system without leaving some evidence behind. Any file created is tracked in so many ways and in so many places that it is hard to find them all if one decides to delete files. Every program installed or used likewise is tracked in so many ways. Systems are designed to have fail safe mechanisms, redundancies,

backups, and Shadow Copies to prevent data loss. Once again, removing data from these places is difficult as many are hidden or obscure.

Cortana is a personal assistant that has appeared in Windows 10. We haven't touched on the evidence of files and programs that are indexed in Cortana. Most certainly they are. While I'll likely cover this in depth in a future article, let me say for now, if you place a file on your system (movie, picture, audio file, document, etc), with almost certainty, Cortana will index it. If you install a program on your computer, Cortana will index it. When you delete these files, they persist in Cortana. For how long, I can't say, but they are there and they do persist beyond deletion. So Cortana is a treasure trove of forensic nuggets waiting to be understood and harvested.

Spoliation examinations are specialized examinations, which at this point should be very obvious. You are looking to uncover proof that data was deleted, to include when, how, by whom, etc. You are also trying to find traces of the original file or even another copy. Such examinations are invaluable to litigating parties, as there is a duty by all parties to a suit to preserve all relevant data and make it available for discovery. In theory, with all facts disclosed, a judge or jury can then render a fair judgement. When spoliation occurs, some of the most important relevant evidence is missing and the system can't properly evaluate the evidence. The spoliation examiner can right that wrong and bring out the fact that evidence was destroyed and in some cases recover that missing evidence. In this manner, the system can factor in the missing evidence by whatever remedy or directions the judge may decide.

These two articles, the previous one dealing with discovering evidence of spoliation on the Macintosh operating system, and this one, are guidelines or suggested workflows for discovering the most common types of spoliation. No roadmap is ever perfect or complete, but it can start you down a road to discovery when you don't know where to start or go next. Always remember to be alert, question the unusual, and be inquisitive. As always, test any questionable theory and attempt to replicate the behaviors you suspect in a controlled and like environment. Be in a position to testify firsthand about how something works or how any given artifact is created, always keeping in mind that the only thing certain is change. What behaved a certain way yesterday, may act differently tomorrow. That said, always test and validate your findings.

About the author: Steve Bunting

Steve Bunting, the author, is one of the pioneers in the field of digital forensics with over 17 years in the field. He spent ten of those years in digital forensics during his 35-year law enforcement career and seven of those years in support of the private sector.

He has been a presenter at several seminars and workshops, the author of numerous "white papers", including:

- the principal author of *EnCase Computer Forensics - The Official EnCE: EnCase Certified Examiner Study Guide, 3rd Edition*,
- the co-author of *Mastering Windows Network Forensics and Investigation* ,
- *the author of EnCase Computer Forensics—The Official EnCE: EnCase Certified Examiner Study Guide, 2nd Edition,*
- the co-author of *Mastering Windows Network Forensics and Investigation 2nd Edition,*
- *the author of EnCase Computer Forensics—The Official EnCE: EnCase Certified Examiner Study Guide, 3rd Edition* (all published by Wiley).

He is an instructor for the Anti-Terrorism Assistance Program, having trained law enforcement personnel in over 20 countries. He was an instructor for Guidance Software and is currently a contract instructor for Micro Systemation, AB, the Swedish company that makes the XRY Ecosystem of mobile device forensic software. He also maintains a digital forensic practice (Bunting Digital Forensics, LLC), which performs a variety of specialized services including spoliation examinations and consultations. He has recently joined forces with SUMURI, LLC, where he will manage their Services Division. Steve can be reached at stephenbunting@mac.com.