498.6

Beyond the Forensic Tools: The Deeper Dive

FOR498.6 | Battlefield Forensics & Data Acquisition

Beyond the Forensic Tools: The Deeper Dive

Authors:
Eric Zimmerman – saericzimmerman@gmail.com
Kevin Ripa – kevin.ripa@gmail.com

https://twitter.com/ericrzimmerman
https://twitter.com/kevinripa

FOR498.6: Beyond the Forensic Tools: The Deeper Dive

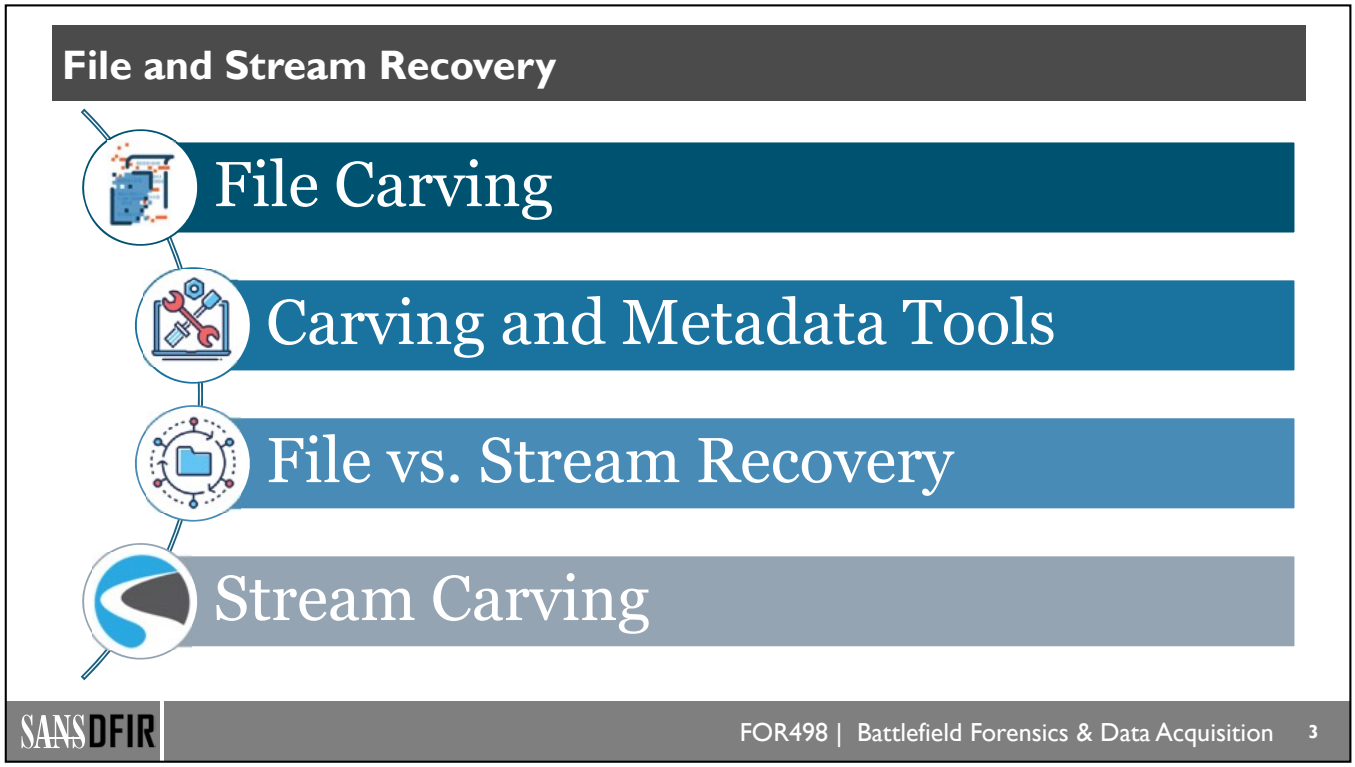
6.1 File & Stream Recovery

6.2 Data Recovery

6.3 Data Carving & Rebuilding

6.4 Where Do We Go from Here?

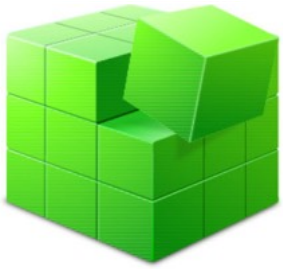
This page intentionally left blank.



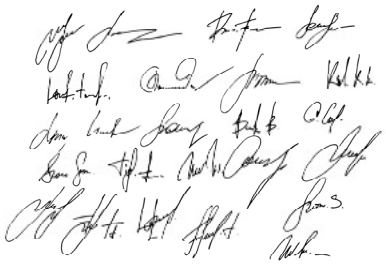
This page intentionally left blank.

File Signatures

File extensions identify a file type



Windows Registry keeps track of file extensions, and the programs that are meant to open them



File signatures allow the program to handle the file

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 4

Files by themselves on a computer would be largely useless without a program to open them. But how does the computer know what to do with the file that it sees? There are a few factors at play here, but the important ones are the file extension of the file, as well as the file signature of the file. The Windows registry keeps track of file associations. In other words based on the file extension of a particular file, the registry will have a record of what program the computer should use to open a file of that type. For example, a .DOCX file will be opened by Microsoft Word, and the reason this automatically happens is because the registry has that file association in its records. If the user tries to open a file for which there is no program associated with the extension, Windows will present the user with a list of choices, and then the user can pick an appropriate program.

Although the file extension looks in the registry for the associated program in order to open it, as the program starts to open the file, it is relying on a format that it understands. Back to our example of the word document, Microsoft Word will be looking at the first few bytes of the file to determine its signature so that the program can know if it can deal with such a file.

Any given file type will have a file signature that is common to every file of that type. Every .PDF file will have the same file signature; every .DOCX will have the same file signature; every .JPG file will have the same file signature; etc.

Unless you have opened a file using a hex editing program, you will be unfamiliar with the concept of file signatures. A file at its most basic is nothing more than hexadecimal code. While we certainly don't expect an examiner to be able to decode hexadecimal by sight, any good examiner should be able to recognize hexadecimal code for what it is. Experience will allow the examiner to look at the beginning of various common files and be able to recognize the type of file simply by the first few hexadecimal bytes of the file.

It is worth noting that .TXT and some email files have no file signature. The only way that Notepad knows to open such a file, is based on the file extension alone.

Sample File Signatures (1)

- **Office Documents – Pre 2007 version**

ASCII	HEX
ÐÏ·à¡±·á	**D0 CF 11 E**0 A1 B1 1A E1

- **Office Documents – 2007 & later version**

ASCII	HEX
PK··	50 4B 03 04

- **PDF Documents**

ASCII	HEX
%PDF	25 50 44 46

- **JPG Images**

ASCII	HEX
ÿØÿ	FF D8 FF

File Signature Listings | `https://for498.com/4x1jc`

Sample File Signatures (2)

- **WebPages**

ASCII	HEX
<HTM	3C 48 54 4D

- **MP3**

ASCII	HEX
ID3	49 44 33

- **ASF**

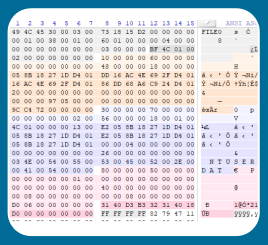
ASCII	HEX
0&²u	30 26 B2 75

- **AVI**

ASCII	HEX
RIFF	52 49 46 46

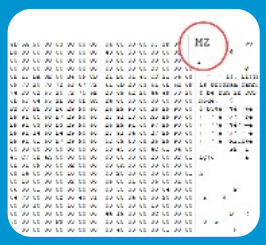
This page intentionally left blank.

File Carving Techniques



Metadata method

- File system metadata tracks
  - Data runs to clusters
  - File names, timestamps, file size, etc.
- Uses
  - Find starting cluster and handle fragmentation
- Example: Deleted metadata points to a file that is 12812 bytes in size and starting at cluster 782



Data layer method

- File signatures (a.k.a. file headers)
  - Example: (.exe = “MZ” header, or “4d 5a 90 00” in hexadecimal
- Scans free clusters
  - At start of each free cluster, look for signature bytes
- Caveats
  - File types may or may not have a footer signature
  - Can’t carve forever, so a maximum threshold must be set

Digital forensics is probably best known to the average individual for the ability of investigators to recover “lost” or deleted files from a file system. Most of the time, the recovery of files is limited to metadata layer recovery. What this means is that a file is recovered by examining the file properties such as the starting cluster, the file size, filename, and parent directory. On a typical system, recovering deleted data is easiest using this methodology. The problem with this technique is that many modern operating systems recycle deleted metadata locations quickly and as a result, overwrite the data that is stored there.

However, even if the metadata of a file is overwritten, in many cases it is still possible to recover a file from the data layer and clusters of a file system’s volume. Tools that focus on unallocated space extraction can scan the beginning of every cluster looking for file headers that match known file types.

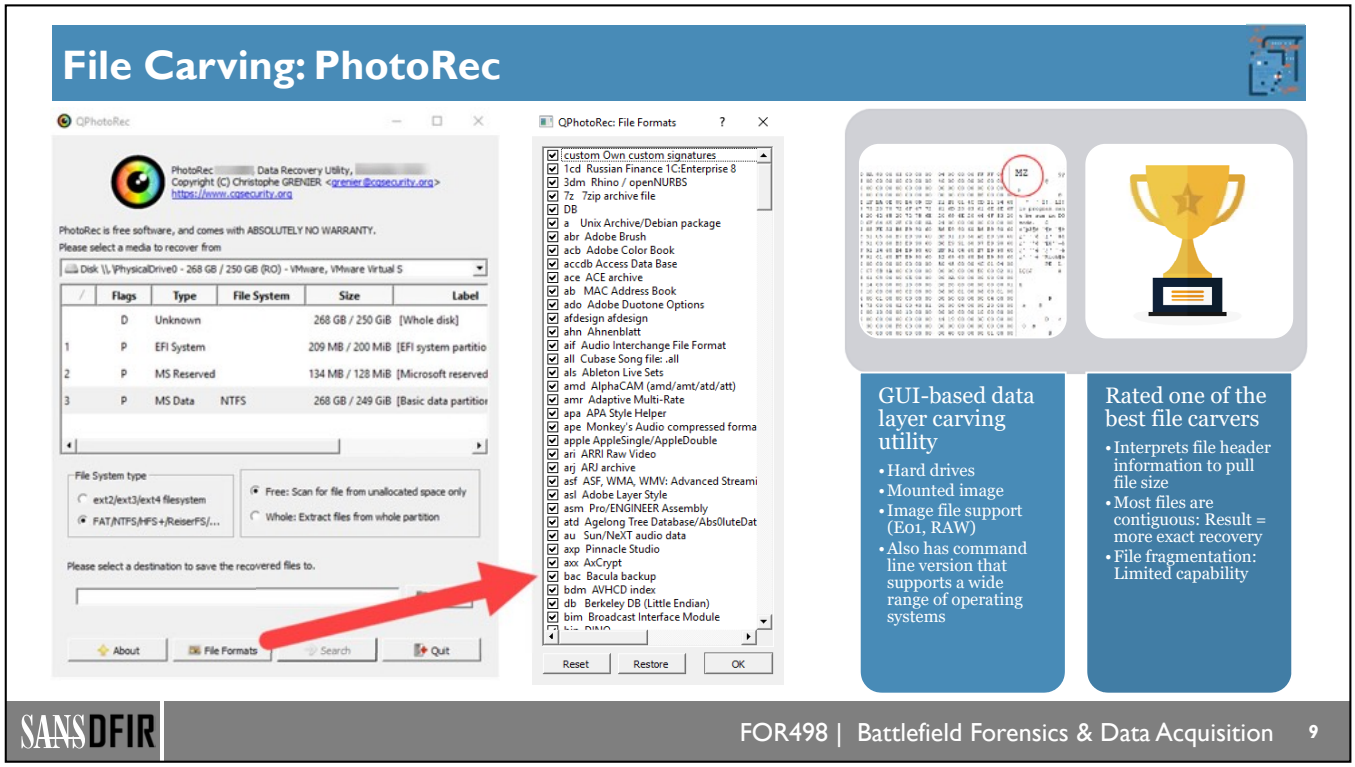
Data layer file recovery, typically also referred to as “file carving”, is the notion of looking in the free, or unallocated, clusters on a file system for known signatures. A signature is essentially a unique characteristic for a file that is found at a (generally) consistent location within the data. For example, if the first four bytes of a file are “0x72 0x65 0x67 0x66”, it is probably a Windows Registry file. If you convert these bytes to ASCII, you end up with the string “regf”, which is the signature for Registry hives. There are hundreds and hundreds of file signatures that have been documented, which means these signatures can be searched for in free space. For another example, a Windows Executable has a file header commonly called the "MZ" header. In hexadecimal, the header is 0x4D, 0x5A, 0x90, 0x00, where 0x4D corresponds to an ASCII "M“ and 0x5A corresponds to an ASCII "Z“. If a carving utility finds the exact header above, the data in unallocated space that it has found is likely to be a Windows executable (.exe) or a dynamic link library (.dll). The file signature can be as small as two bytes and as large as 64 bytes in some instances. So with the concept of signatures in mind, how are they used to recover files?

When file recovery software starts looking at unallocated clusters, it starts looking at the beginning of each cluster for each of the signatures it is trying to locate. It is essentially matching the signatures against the data it finds in the cluster. If it finds a match, it can then start recovering the data and saving it to a file with an extension that matches the signature that was found. In our Registry example, if a cluster was found with "regf" in it, the carving tool might save out the data to a file that ends in '.hbin' for example. If the signature for a jpg was found, the file extension would be '.jpg'. In both these examples, we only mentioned the file extension, but not the file name. So, what file names are used when file recovery software finds signatures? Since the data was found in free space, there is no reference back to any kind of file name that the data used to be associated with, and because of this, a random file name, such as 'image0001.jpg' might be generated. For every other jpg found, the counter would be incremented by one, and so on.

Some software, such as X-Ways, can do what it refers to as "intelligent naming", and based on other information in the data it carves, renames files based on this other data. If you have heard the term "metadata" before, this is what it is using. We will talk about metadata when we discuss tools, but for now, think of metadata as "data about data". As an example, for a jpg image the metadata may contain things like GPS coordinates, the make and model of the camera or device that took the photo, the time it was taken, and so on. With this concept in mind, it becomes clearer how X-Ways can use the metadata to "intelligently" name files it recovers. Going back to jpg files again, it uses the make and model of the camera to name the recovered files, so you might end up with something like "Canon PowerShot S910.jpg" vs "image0001.jpg" which is obviously more useful, especially if you know you are interested in Canon cameras.

Another question related to file carving is how does software know when to stop carving? In many cases, carving software will use a combination of techniques, such as carving until an allocated cluster is found, carving until a certain maximum length is found, or when a file's footer is found. A footer is basically a signature that is found at the *end* of a file, vs the beginning. In these situations you can think of the signature and footer being the bookends where everything in between is the data from the file. It is important to note that not all file types have file footers, or accurate file footers.

It should also be noted that file carving is not flawless and depending on how the data is arranged on a storage device, you may end up with a lot of false positives, or junk files, that cannot be viewed. Consider the case where a fragmented file gets deleted. Because the data is not contiguous, a file carver has little chance to find all the fragments without additional help from file system metadata (like where all the other clusters are for a file). When such metadata IS available, more accurate recovery is possible. When it is not however, as would be the case if someone quick formatted a volume, file carving may be all that is possible. With this in mind, we recommend you limit your file carving to signatures that are relevant to a particular investigation, so you reduce the amount of noise in your case. In other words, just because you can carve for 900 different file types, should you? With that, let's take a closer look at file signatures.



PhotoRec [1] is a file-carving program that comes in both command line and GUI flavors. The GUI version, shown above, has drop down menus that allow for selecting a disk in the machine that PhotoRec is running on, as well as providing the option to select an image file, such as a RAW or E01 file. You can also mount an image file using Arsenal Image Mounter and then scan the newly mounted device as well, should you have a need to mount the image for other purposes. The GUI also exposes options for file systems and the thoroughness of the search (either unallocated space only or everywhere). The last bit of configuration includes where to save the recovered files and, optionally, a list of file types for PhotoRec to look for. In the File Formats dialog, the Reset button unchecks everything, while the Restore button checks everything.

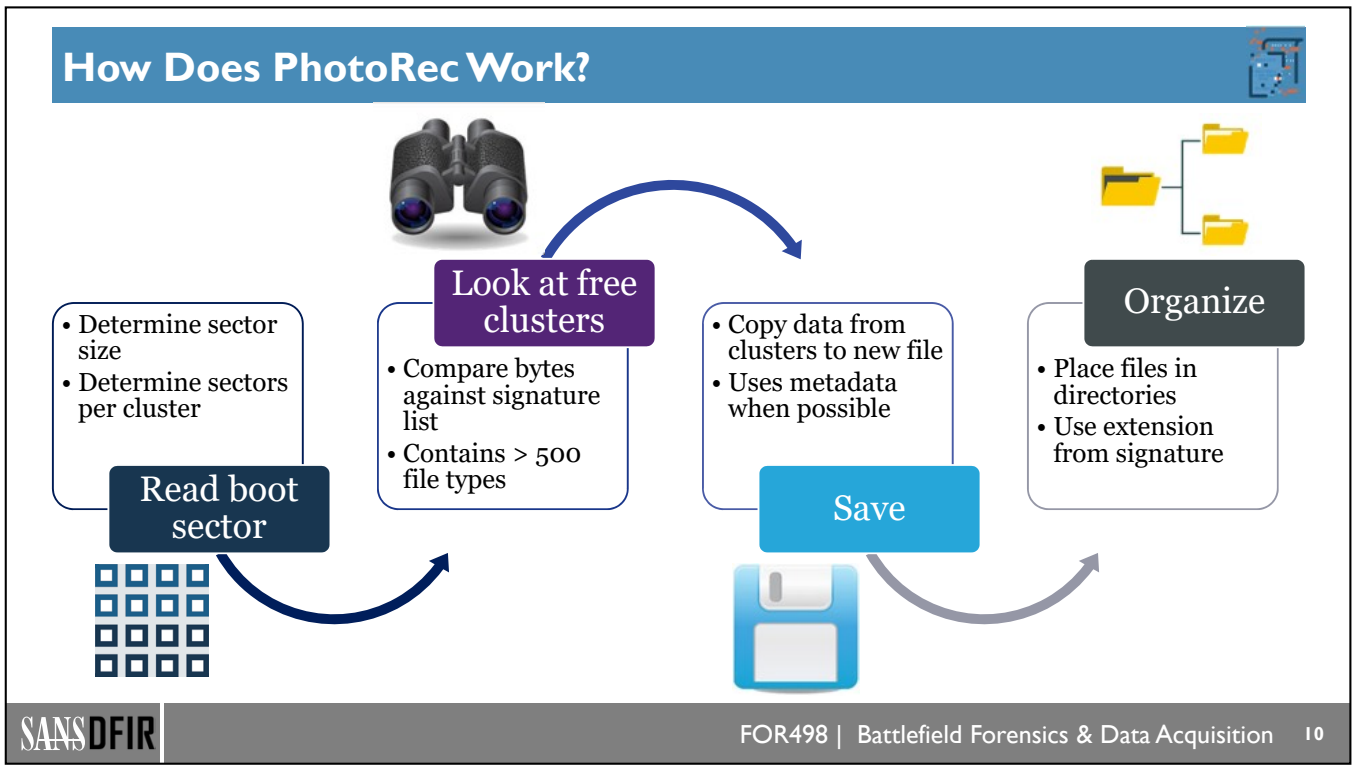
PhotoRec has been rated one of the best file carvers by the National Institute of Standards and Technology (NIST) [2]. According to Wikipedia, "The National Institute of Standards and Technology (NIST) is a physical sciences laboratory, and a non-regulatory agency of the United States Department of Commerce. Its mission is to promote innovation and industrial competitiveness. NIST's activities are organized into laboratory programs that include nanoscale science and technology, engineering, information technology, neutron research, material measurement, and physical measurement."

The only other comparable file carver that is equal or better is X-Ways Forensics.

There is a lot going on under the covers when using this program to recover data, and it is important to understand how PhotoRec or any other tool you are using goes about accomplishing its task. You should always strive to understand as much about HOW a program works as you do about how to use it. For this reason, let's take a deeper look at what is going on when PhotoRec is looking for data to recover.

[1] PhotoRec - Digital Picture and File Recovery | `https://for498.com/o2q6l`

[2] National Institute of Standards and Technology (NIST) | `https://www.nist.gov`



How does PhotoRec do what it does so well? There are a few pieces to this puzzle, including determining the cluster size in use for a given file system, and then scanning clusters for signatures.

The first thing PhotoRec does is read the Boot Sector of the NTFS or FAT partition since this is where the cluster size is recorded (well, its sector size and the number of sectors in a cluster, to be exact). Once the cluster size has been determined, it reads the target file system volume by cluster and examines the initial header of each cluster.

Once a cluster is read, PhotoRec uses a list of 500+ different file signatures [1] and compares the bytes found in the cluster to each signature. If a match is found, the data from that cluster on is saved to a new file, and the file is given a proper extension depending on the signature that was found. As PhotoRec finds files, it saves them to a base directory and child directories named **recup_dir.<number>**, where number is an incrementing value starting at 1. A new directory is created for every 500 files recovered. This potentially leads to a lot of different file types scattered across a lot of different directories, but we will see a way to address this issue soon, so it is not much of a concern.

One of the better features of PhotoRec is the capability of the tool to read the header of the files being recovered. This allows it to do a much better job of accomplishing file carving by being able to interpret file metadata that is stored in the MFT. When this is possible, PhotoRec truncates the size of the file to match this exact size. On the other hand, if a carved file ends up being smaller than what the file size shows in the MFT, then the recovered file would be discarded since it is not complete. This is not always optimal! It could discard a partial file that may have still held enough data to be interesting to an investigation.

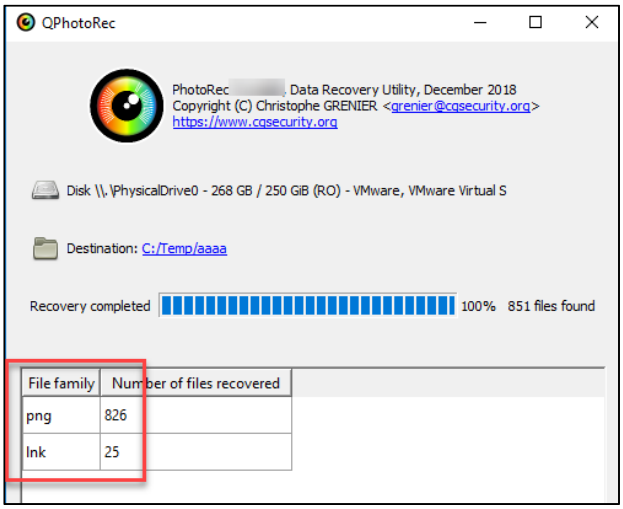
PhotoRec also has some limited capability to handle fragmentation. It can accomplish this by looking at the previous clusters to see if a file signature was found and the previous file was not carved for some reason. It will then attempt to try and carve the file again with the additional data. Even with this additional capability

to look back and attempt to join the data together, if a file is severely fragmented in more than two places, we have found that the recovery of the file is probably going to fail. However, it is one of the few tools that even attempts to accomplish this and is unique in that perspective.

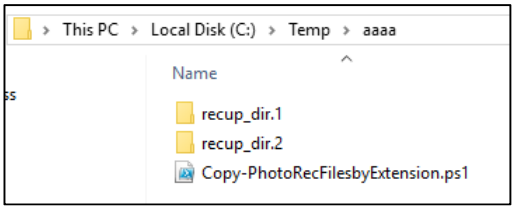
PhotoRec includes support for many Windows artifacts including executables, shortcut (.lnk) files, Event logs (.evt), and Internet Explorer history (index.dat). Besides that, it does a wonderful job at carving out archives and multimedia file types. While the interface is not the most intuitive and visually appealing, PhotoRec is a powerful tool indeed.

[1] File Formats Recovered By PhotoRec | `https://for498.com/abcvn`

PhotoRec Pro Tip: Sorting by File Type (1)



Multiple directories full of recovered files



SANS DFIR FOR498 | Battlefield Forensics & Data Acquisition 12

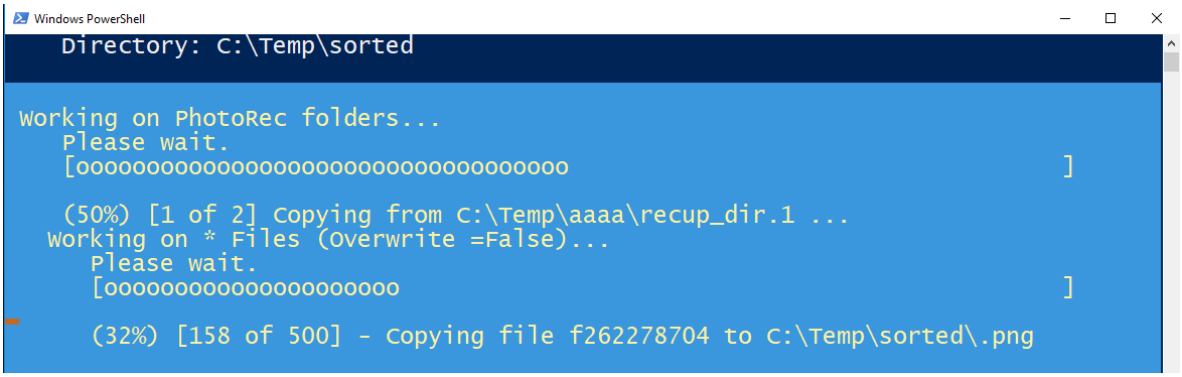
Copy-PhotoRecFilesbyExtension [1] is a PowerShell script written by Luca Conte that copies all files from the PhotoRec folders to new folders named by file extension. The script can be placed anywhere and does not have to be in the same directory that PhotoRec dumped its output to. Once PhotoRec is finished, open a PowerShell window and execute the script. The required parameters are **`-RootPhotoRec`** which is where PhotoRec dumped its output to, and **`-RootDestFolder`** which is where you want the sorted results to end up. Here is an example:

```
Copy-PhotoRecFilesbyExtension.ps1 -RootPhotoRec C:\Temp\aaaa\
-RootDestFolder C:\Temp\sorted
```

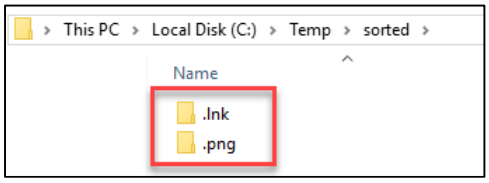
Once the required parameters are supplied, the script will begin.

[1] Copy-PhotoRecFilesbyExtension.ps1 | `https://for498.com/38qxz`

PhotoRec Pro Tip: Sorting by File Type (2)



Files copied to new folders, based on extension!



The script provides overall progress along with how many directories were found. When the script is finished, the destination folder will contain new directories, one for each unique file extension that was found.

Other optional parameters include:

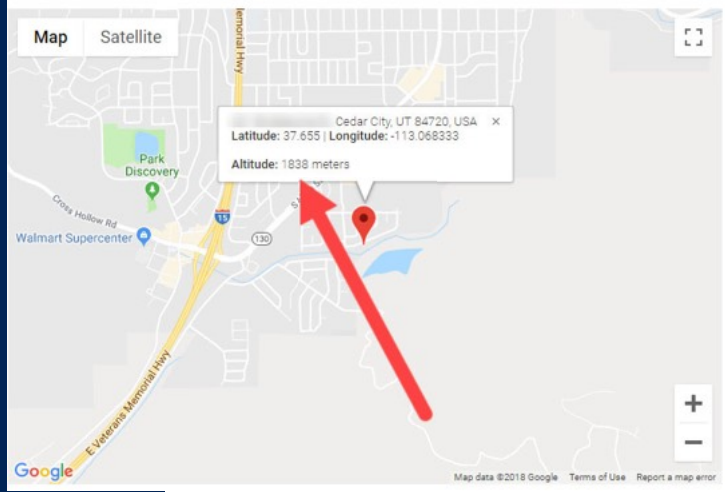
- **`OverWriteDuplicatedTo`**: When true, overwrite any duplicate files **with the same name** in the same destination folder. Only the last file copied is kept. False by default.
- **`CustomFileFilter`**: When set, filter out files that do not match the supplied pattern. The filter must start with a *, such as *jpg for example.

The nice thing about sorting the output from PhotoRec is that you can then leverage dedicated parsers against the newly gathered files. For example LECmd, a command line tool to parse .lnk files, could be pointed at the .lnk directory to process everything at once.

While sorting is by no means required, it is often very helpful to combine related files together for a variety of reasons.

Metadata Tools: ExifTool

```
PS C:\Tools>    '.\exiftool(-k).exe' C:\Temp\photo.JPG
ExifTool Version Number         : [redacted]
Make                            : Apple
Camera Model Name               : iPhone 4S
Orientation                     : Horizontal (normal)
X Resolution                    : 72
Y Resolution                    : 72
Resolution Unit                 : inches
Software                        : 6.1.3
Modify Date                     : 2013:03:26 11:59:49
Y Cb Cr Positioning             : Centered
Flash                           : Off, Did not fire
GPS Latitude Ref                : North
GPS Longitude Ref               : West
GPS Altitude Ref                : Above Sea Level
GPS Time Stamp                  : 17:59:49
GPS Img Direction Ref           : True North
GPS Img Direction               : 264.1028037
Compression                     : JPEG (old-style)
Thumbnail Offset                : 902
Thumbnail Length                : 10019
Image Width                     : 3264
Image Height                    : 2448
GPS Altitude                    : 1822 m Above Sea Level
GPS Latitude                    : 37 deg 39' 18.00" N
GPS Longitude                   : 113 deg 4' 6.00" W
GPS Position                    : 37 deg 39' 18.00" N, 113 deg 4' 6.00" W
```



ExifTool [1] is a powerful command line utility that finds and extracts metadata from a wide range of files. For our purposes, we will look at its ability to extract and display Exchangeable Image File Format (EXIF)[2] data from pictures. In the example above, ExifTool was used on a jpg file which resulted in dozens of EXIF tags [3] being displayed (the image above has the less important ones removed).

Of particular interest are the tags related to the device that took the photo, when the photo was taken, and, in this example, detailed GPS coordinates (including the altitude!) for the exact location the picture was taken.

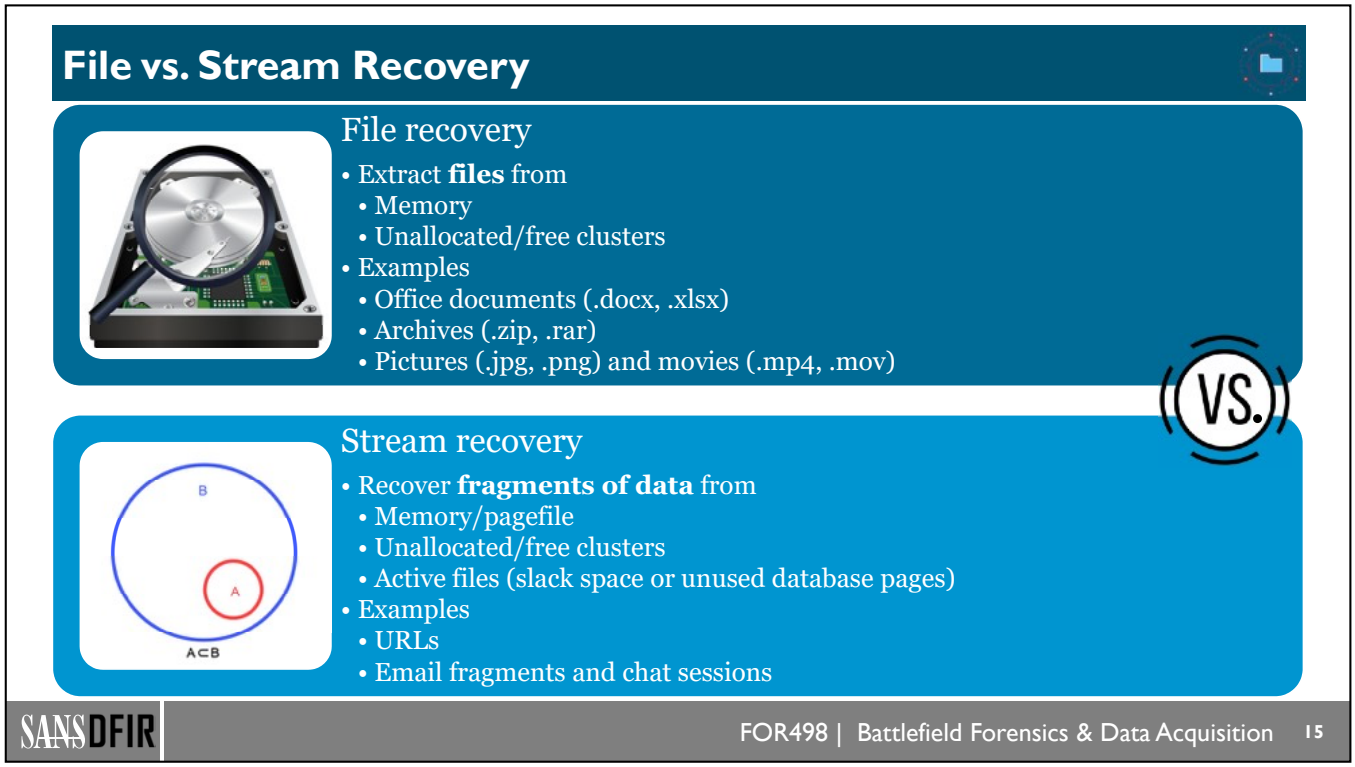
Using ExifTool is very easy:

```
exiftool.exe <filename.ext>
```

ExifTool will then look at the file and extract out any details it knows how to extract. This can be redirected to another file as well vs. only looking at it in the shell. ExifTool also knows how to handle data that gets piped into it, so it is useful in batch processing scenarios where you have a lot of pictures to process.

So how does this tie into data recovery? Recall in many cases, a file name will not be available unless the file carving used the metadata method to find the data to recover. By using ExifTool on recovered files, additional details can be gleaned (sometimes including a file name!) that would otherwise not be obvious to discover. Recall from earlier we mentioned that X-Ways names recovered files using “intelligent naming”, which it derives from information it finds in EXIF tags. Now, should you not have access to X-Ways, you have a way to achieve something similar, in that you can look at the EXIF tags for a given file and then give the recovered file a more meaningful name.

[1] ExifTool by Phil Harvey (Good Canadian boy!) | `https://for498.com/ie7mb`
[2] Exchangeable Image File Format (EXIF) | `https://for498.com/tfukq`
[3] EXIF Tags | `https://for498.com/i38jv`



Data recovery typically falls within two categories: file recovery and stream recovery. With an understanding of file carving behind us, let's look at the other possibility: stream carving. Stream carving is different from file carving in that we are looking for a subset of data within a file vs. an entire file itself. For example, consider index.dat, which is used by older versions of Internet Explorer to store things like browser history in the form of URLs. index.dat may contain thousands of URLs inside of it and these are what stream carving would recover. In this situation, index.dat can be thought of as a container holding a bunch of other data (URLs in our example) that can be recovered.

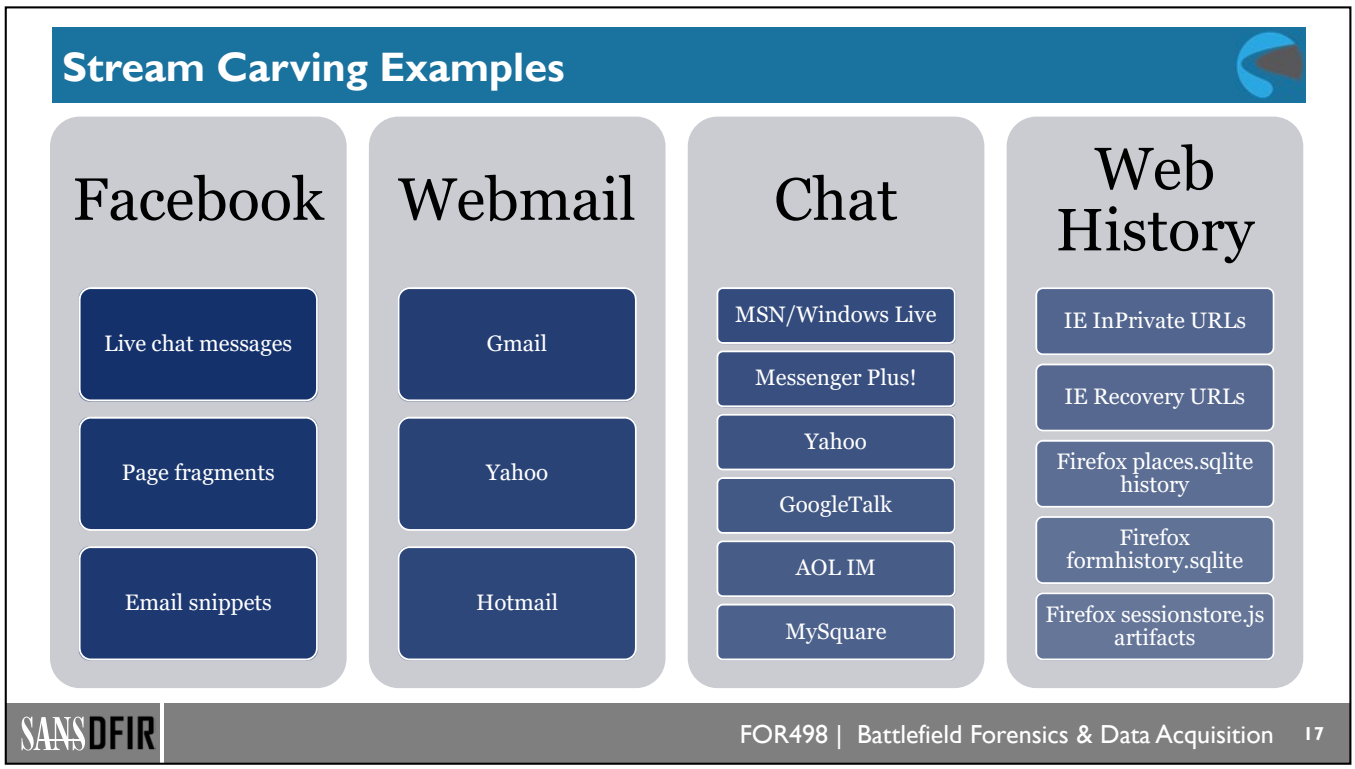
Where this becomes even more useful is when we can only partially recover data from a file. Let's stick with our index.dat example again. Suppose only the first 70% of an index.dat file could be recovered, which isn't enough for index.dat parsers to fully dump the contents of the file. Enter stream carving! A program that knows how to look for things inside of the partial index.dat file can look for signatures inside the index.dat file's data and extract them out as is. This powerful technique allows investigators to more fully exploit the data on a machine, even when traditional file carving fails.

To clarify, data carving works on all manner of files such that it can be used on allocated files (tracked by a file system), files carved fully from free space, and fragments of files. Tools such as Magnet Forensics Axiom do a thorough job extracting out such things as chat history, email fragments, browsing history, and so on from things like memory, free space on a hard drive, and so on.

Because of the nature of what is happening with all this data, and the large amount of signatures that must be looked for, stream carving can take quite a long time to finish. Recent forensic research has allowed for deleted data in SQLite databases, such as those used in most mobile applications, as well as programs like Firefox, Chrome, and Skype[1]. This is a rich source of detail that can be of significant help in investigations.

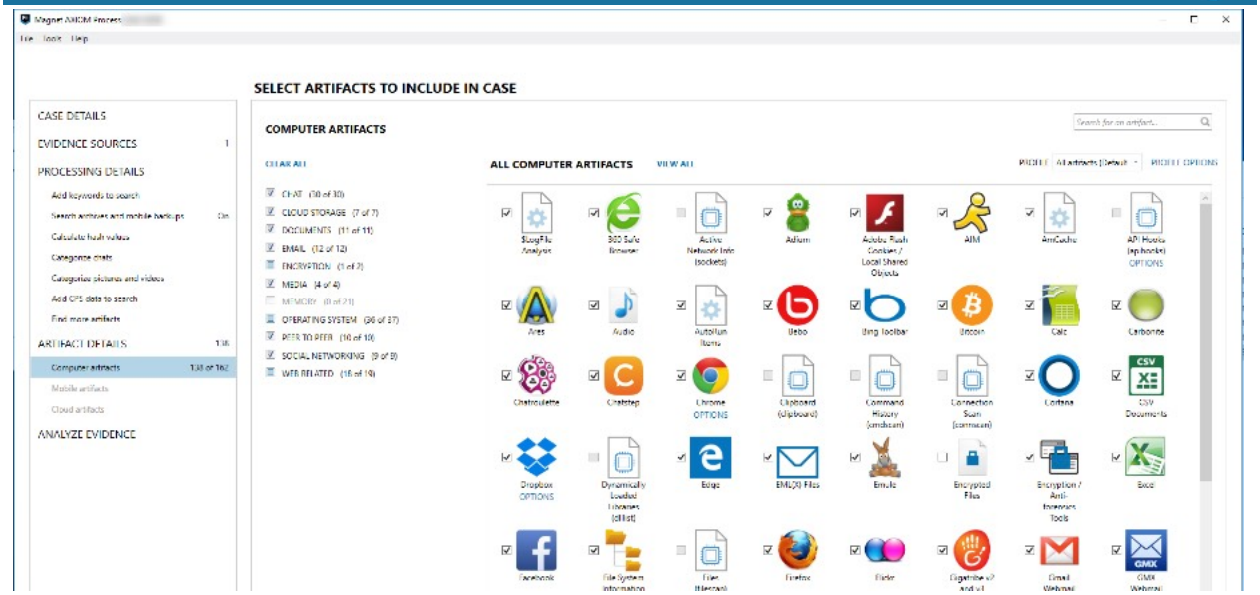
As a final example of stream recovery, consider another file type; a PST email archive generated from Outlook. The PST file contains emails that Outlook has archived, but what happens when a user deletes an email message from a PST file? The pointer to the email message is simply removed and the PST no longer maintains a reference to the email message. The data, however, is still there until something else overwrites it, free space is consolidated, and so on. Stream carving can look in the free space of the PST though to find and make available again the contents of the previously deleted email message.

[1] Recovering Data from Deleted SQLite Records | `https://for498.com/s024-`



This chart represents some of the types of streams that can be recovered. This is by no means an exhaustive list, and programs that know how to do stream carving typically allow for selecting all available options or targeting just a few types of artifacts for recovery. There are many tools that can perform stream carving such as X-Ways and Axiom, and we will spend a bit of time looking at Axiom and what it can do for us from a stream carving perspective. Axiom searches images, drives, folders, or individual files (memory dump, pagefile.sys, etc.) for a wide range of artifacts, including those listed above, along with many more.

Stream Carving Tools: Axiom (1)



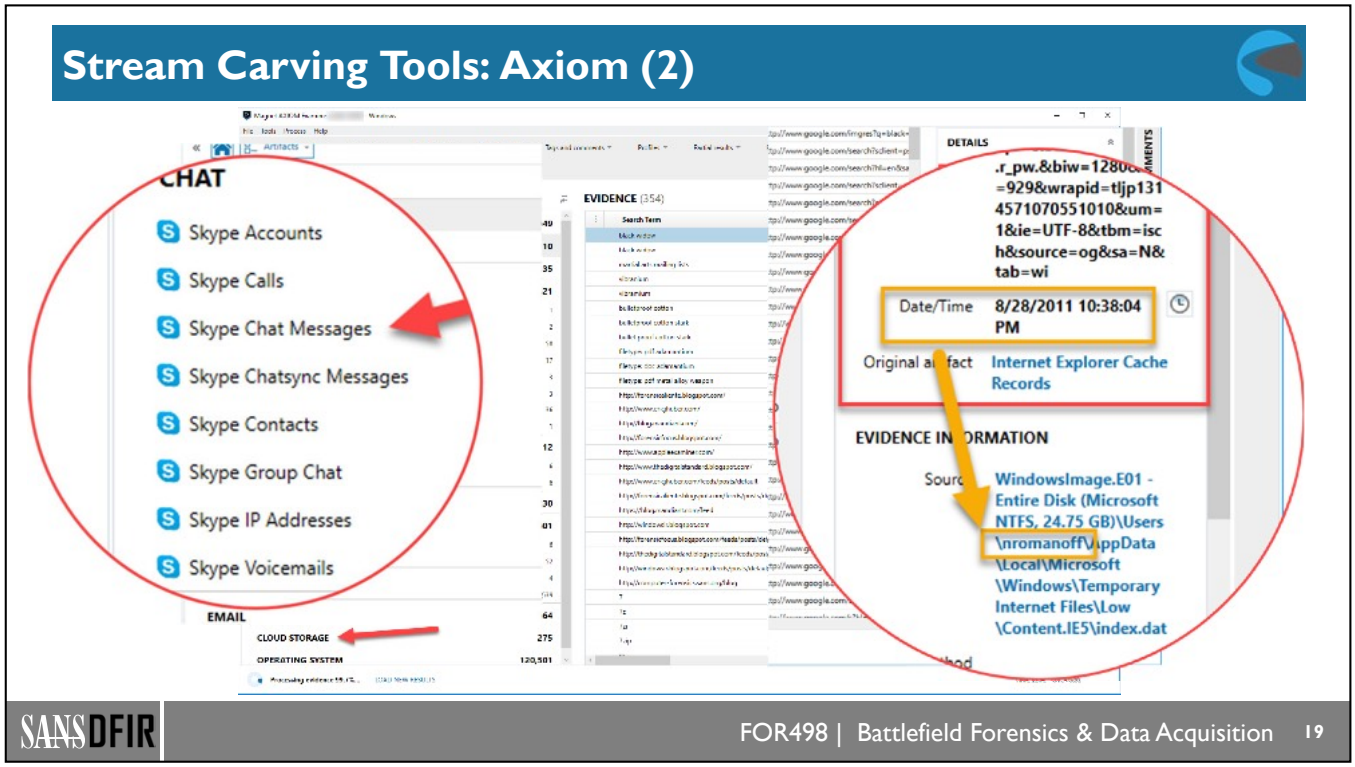
Axiom is a program that allows you create a case, add evidence, and then select different artifacts to look for. Once it is configured, the software then begins processing all the evidence according to the settings you chose.

When creating a new case, you need to supply details such as the name of the case and where to save the case data Axiom will generate.

Evidence sources include mobile devices, cloud providers, or the local computer. When selecting the local computer, you can select multiple forensic images (including volume shadow copies), memory captures, and other forms of evidence to be added to a case and processed as a unit. Of course, the more data added to a case, the longer it will take to process.

Next, additional processing options are configured, such as hash calculation, matching against hash sets, and so on. The last step in configuration is selecting which artifacts to look for and process. This is what is shown in the screen shot above. Notice how there are categories of artifacts available related to things like chatting, cloud storage, email, pictures and videos, and so on. This is where you will want to tune the selected options (either as an entire category, or individual entries) depending on your case, as it just takes additional time to process artifacts you may not ever use.

After everything is configured, Axiom starts processing evidence. As the case is processed, everything that has been found so far can be reviewed. This lets you start to review data is it is available vs. having to wait for an entire case to finish processing first.



This is a screenshot from Axiom after it has processed a case. The tool has scoured a forensic image in order to extract out key fragments that might be found useful. Although extremely useful in carving out interesting data, the flaw with Axiom and similar tools is that, in most cases, the tool lacks context behind the "who" (the user that did it) and/or the "when" (the time something happened) associated with the artifact recovered. Without the associated data that would tie the artifact to a time and a user, this information is useful only to perform additional searches that will help identify additional fragments.

With that said, not every fragment recovered by Axiom lacks a username or timestamp. Chat messages regularly include a date and time associated with a particular user's conversation. In addition, if Axiom is asked to extract data from a hard drive, it does parse regular files and you can source the timestamp information from the file metadata directly and the user associated with it by examining the file's path, especially if it is found in the user's home directory.

In the example above, notice on the left how we have artifacts grouped categorically. Each category can be expanded and collapsed, allowing you to drill into different areas as you look for things. Selecting a specific artifact loads the results in the grid on the right. Selecting an entry in the grid populates the details pane on the far right. Notice how you can see the search term "black widow" separately from the URL where it was found along with the fact this URL came from "Internet Explorer Cache Records". This is a perfect example of stream carving search terms out of a broader file, index.dat. The full path to the file is shown below the details box. Most artifacts can be tied back to a user, and in this case, you can see the index.dat file was found under the "nromanoff" profile. Finally, notice that we not only get the search term, but when the search happened. This is recorded in the "Date/Time" field in the details view.

Summary

- File signatures are how the OS identifies the type, starting point, and sometimes ending point of files
- An examiner should be familiar on sight with a few of the most common file types
- File carving is how we recover files that are no longer addressed by the OS
- Stream carving allows us to recover data that may not reside in a normal file structure

This page intentionally left blank.

Exercise 6.1

Data & Stream Carving

Synopsis: In this exercise, you will use PhotoRec to carve deleted files from a mounted drive. You will then filter them through ExifTool. After this, you will use a number of EZ Tools to parse through the recovered data. Finally, you will use Axiom to perform stream carving.

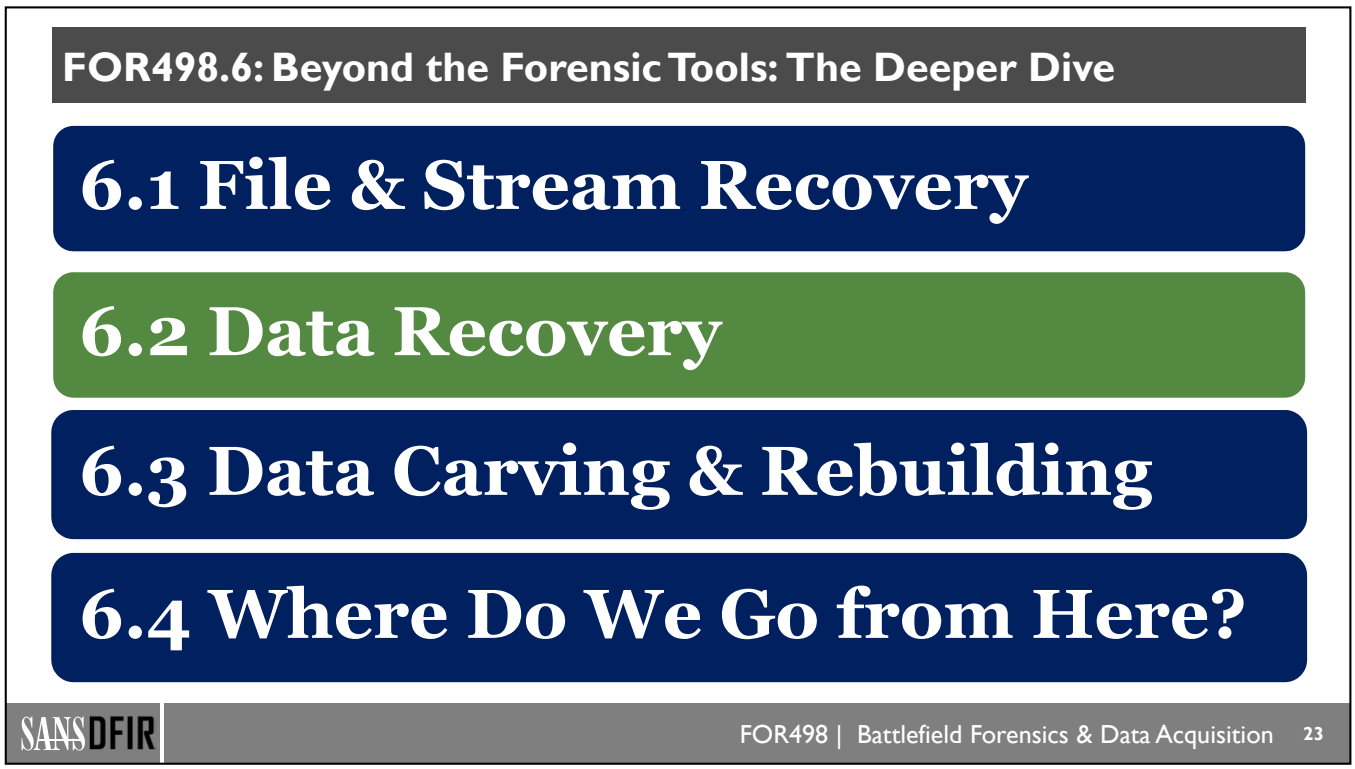
Average Time: 45 Minutes

This page intentionally left blank.

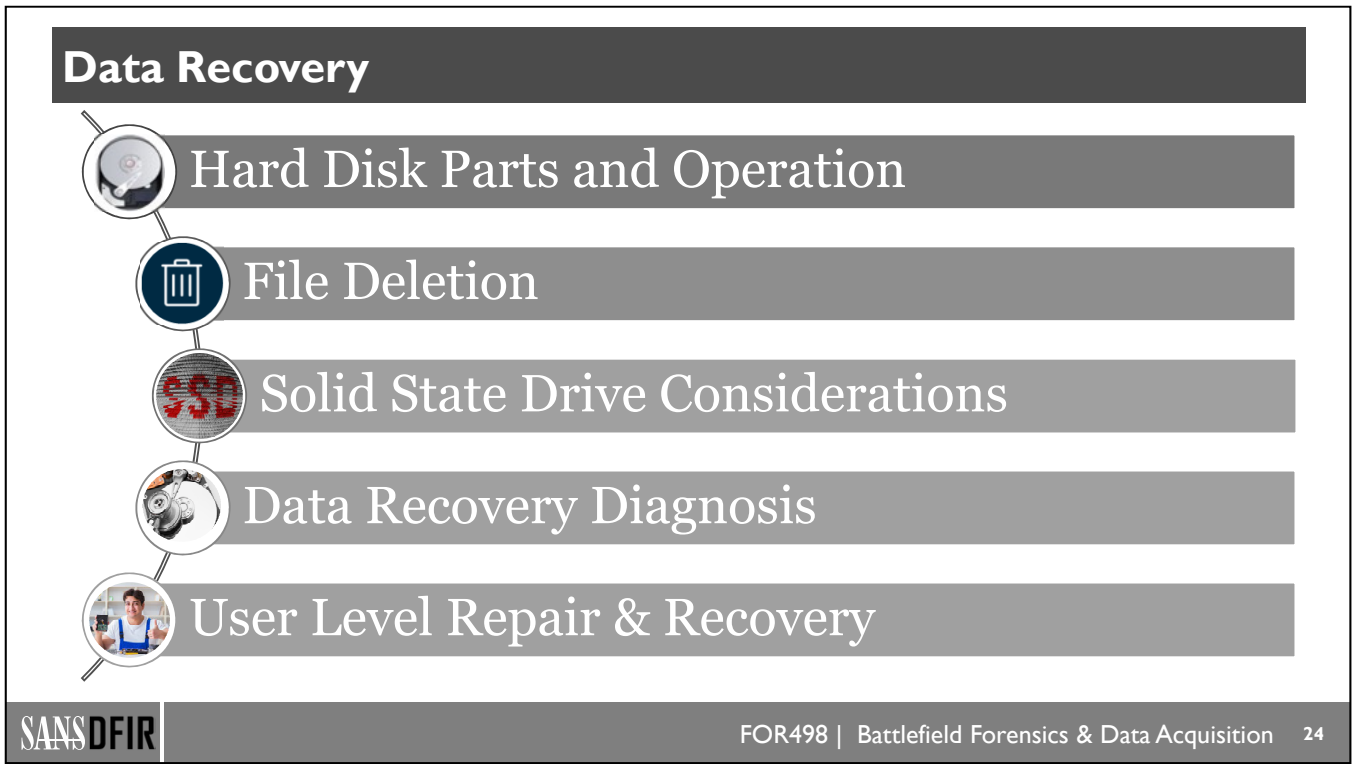
Exercise 6.1 Takeaway

- Data and stream carving can bring new leads and insight to your cases.
- There is no guarantee that files recovered via carving will be useful.
- Different tools can produce different results. Test your tools!
- Tools like Axiom can find data embedded inside files, such as URLs, email addresses, and more.

This page intentionally left blank.



This page intentionally left blank.



This page intentionally left blank.

How Does Data Live on a Drive

Email

Spreadsheet

Database

Photos

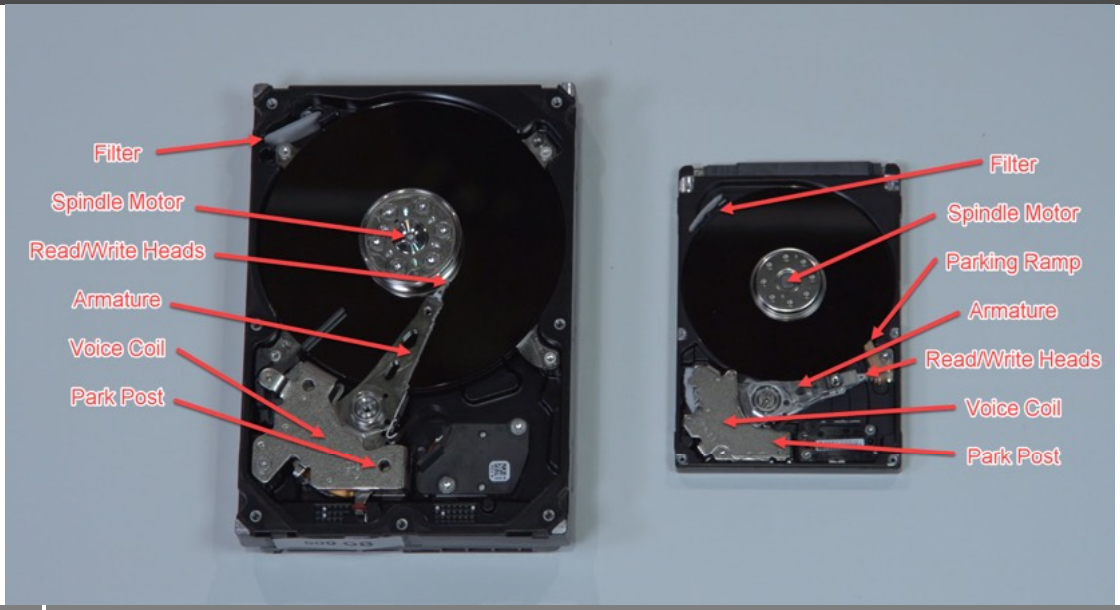
This module is not intended to make you a data recovery specialist, and as such, we will not be going terribly deep into the intricacies of data recovery. Two weeks of training would merely make you dangerous. The idea behind this module is to educate the examiner on how data lives physically on spinning and solid-state drives in a general sense, as well as give some guidance on what an examiner can and cannot do when faced with a non-functioning hard drive.

There are many different areas of data recovery, all with their own disciplines. Rare is the person that can function in all arenas equally. In each arena, there are two subsets. These being rotational (or magnetic) media, and solid-state media. These again are their own disciplines.

A great many organizations including law enforcement have what they consider data recovery capabilities. A very few of these are dedicated, qualified practitioners that are set up properly. It is often seen that attempts are made to recover data based on YouTube videos and Google research. The problem with this is that often, the attempt will destroy the media to a point where no one can ever get the data back.

We will start by exploring the physical components of a hard drive, and then move into how the data resides on it. After this, we will look at various damage and failure points of storage media and finally, what we can or can't do about it.

Hard Disk Drive Interior



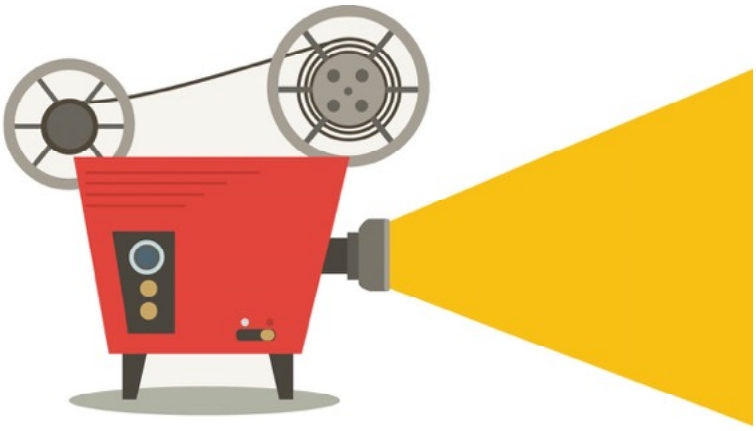
A spinning hard drive is made up of 1 or more platters. These platters are made of various materials laid down in multiple layers on a metal, ceramic, or glass platter surface. They will usually (but not always) have a head on either side of each platter to read and write data to the platter surface. Each of these heads is connected to an armature, which is in turn, connected to each other in a head stack assembly. In this way, all heads move back and forth across all platters at the same time. A portion of the assembly at the far end from the heads themselves rides between two magnets. This is called the voice coil, or actuator. The voice coil works on the same principal as a speaker, in that the assembly will move one way or the other based on the direction of the current on the coil. The amount of movement is dictated by the amount or intensity of the current.

The head stack assembly communicates with the printed circuit board (PCB) on the outside of the drive, via a sealed series of pins and a ribbon cable. The drive contains a motor that the platters are placed on, and this spindle motor is what spins the platters.

The read/write heads themselves park in one of two places on a hard drive when the drive is not powered on. They may park on the platters themselves, up against the spindle motor on something called a parking area, or they may rest off of the platters, on something called a parking ramp. In the left example on the slide, the head is on a parking area. In the right example in the slide, they are on a parking ramp.

Two other important pieces inside a hard drive are a dust filter and park posts. The park post limits how far a head stack assembly can travel, to protect it from hitting the spindle, or flying off the edge of the platter.

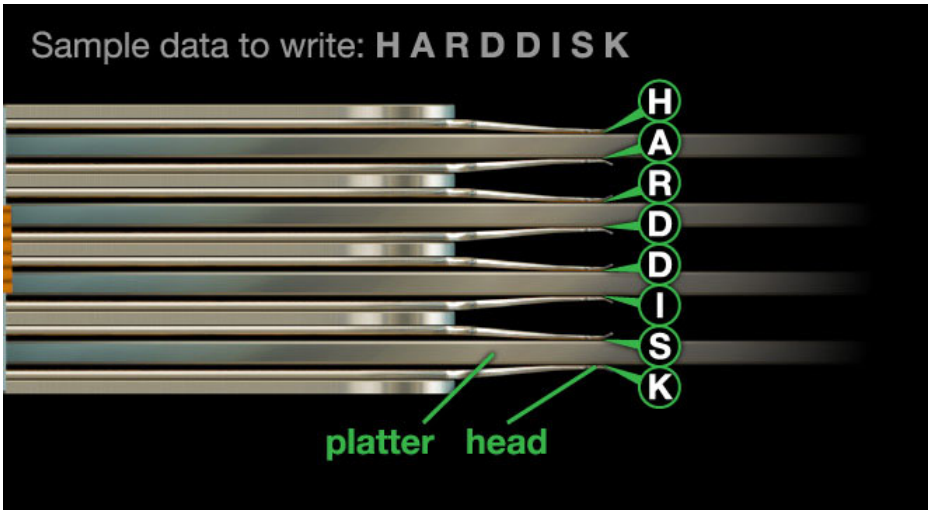
Movie Time!



- 6_1: HDD Parts ID

This page intentionally left blank.

Hard Disk Drive (HDD) Operation



Graphic courtesy of Animagraffs

Imagine a Boeing 747 flying over the earth at Mach 800, ½ an inch (1 cm) off the ground, counting each blade of grass on the way by, with less than 10 errors for every few hundred miles. [1] This is the approximate comparison of how a hard drive functions. A 3 TB HDD today has a read/write head stack that flies about 40 angstroms above the platter surface. How big is 40 angstroms? That is 4 nanometers [2]. Or 8 atoms (specifically silicon atoms). In a more relevant comparison, a human hair is about 75 000 nanometers in size.

As described in the previous slide, the read write heads do not sit directly on the platter. When the drive is not operating, the heads will sit either on the parking area of the platter close to the spindle, or they will sit on a parking ramp off of the platter surface. So then how do the heads get over the data? When power is introduced to the drive by pressing the power button on the computer or plugging the external drive into a computer, the drive platters start to spin. Hard drives are designed to spin at a predetermined RPM. Whatever that speed might be, the heads do not move until the platters are spinning at that predetermined speed. The inside of the drive is designed so that the wind created by the spinning platters creates something called an air bearing. This air bearing causes the read/write heads to float just above the platter surface. If this air bearing is interrupted, the heads may potentially contact the platters causing irreparable damage. As a point of reference, the turbulence inside a hard drive that has a rotation speed of 7200 RPM is approximately equivalent to an 80 MPH (120 KPH) wind.

It is a common misconception that hard drives are hermetically sealed units. In fact there is a breather hole on the top, bottom, or side of every hard drive from 8 TB and smaller. There is new technology now where the hard drive is helium-filled, and in these cases the drive is a sealed unit. The reason manufacturers are moving towards helium-filled drives is because it allows the drive to run faster but cooler because it does not have the friction found within normal air. When speaking of normal, the breather hole is there to allow air to pass in and out of the drive, in measured quantities, to allow the necessary turbulence required for the read/write heads to ride on the air bearing.

Once the drive is spinning at the appropriate speed and the read/write heads are riding on the air bearing, the space between the heads and the platter is called the fly height or flying height. This fly height has changed over time based on storage space, and hard drives getting larger and larger in capacity. When hard drives were 10 GB in size spread out across two platters, the fly height was considerably more than the fly height of today where we have 4 TB on two platters. Hard drives of today are able to pack 1.1 TB of data per platter side. The largest production drive (rotating) on the market today is 20 TB packed onto 9 platters, all in the ubiquitous 3.5" form factor. This is pushing the limits of data storage.

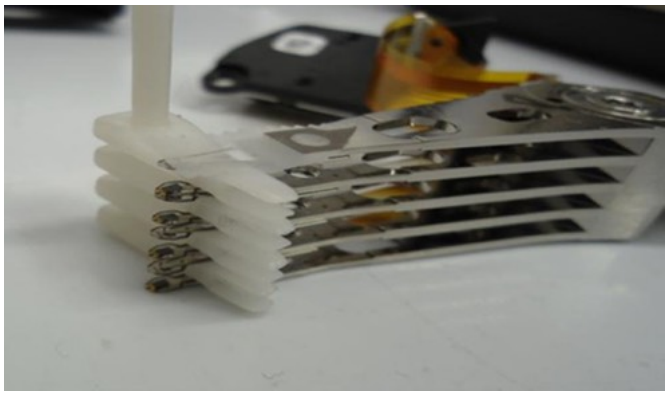
You may have wondered how hard drives have gotten larger and larger in storage size but have never gotten larger physically. One of the way manufacturers have been able to accomplish this is by changing something called the areal density [3] on the drive. As the hard drive is broken down into tracks and sectors at the factory, the manufacturer is able to put more and more of them closer and closer together, thereby allowing more of them to exist on a platter. When this happens, something must occur with the fly height and the read/write heads in order to accommodate the data being packed more closely together. Among other things, two very important things need to change. The first is the amount of voltage being applied to the read/write heads. This must be reduced to create a tighter focus. The second thing that must change is the fly height. The read/write heads must fly much closer to the platter because of the lower voltage. On hard drives of today, a fingerprint on the platter surface is quite a few times thicker than the actual fly height and will destroy the heads as the platter spins and pulls the fingerprint under the head. Such are the tolerances that hard drives must operate with today.

[1] Fly height comparison | `https://for498.com/yo013`

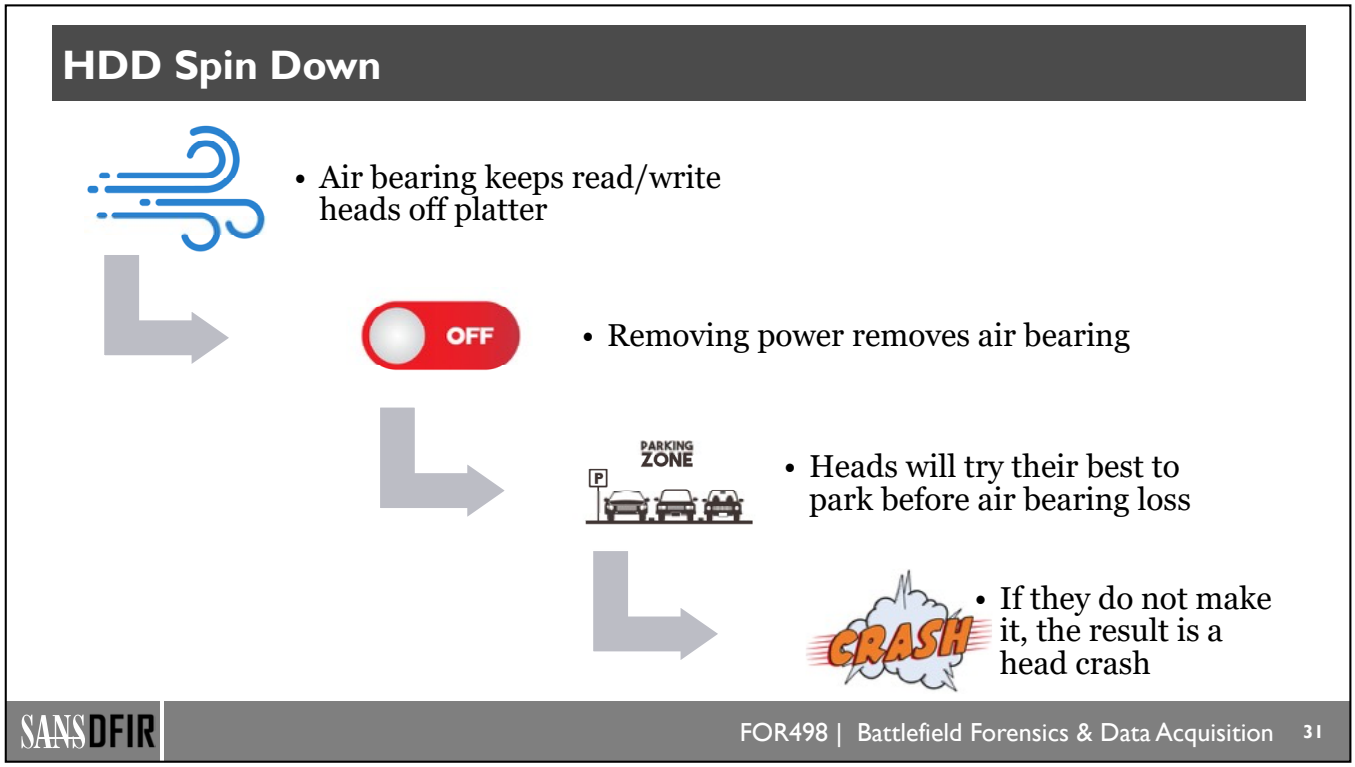
[2] How big is a Nanometer? | `https://for498.com/tieym`

[3] Areal Density | `https://for498.com/pqmrs`

Head Stack Assembly



Head stack assembly both on the parking ramp, and out of the hard drive, being separated by a head comb.



One of the most critical times in the life of a hard drive is at spin down. This is the time for which power has been detected to be removed from the hard drive. As previously mentioned, the read/write heads ride on something called an air bearing. This is the only thing holding the read/write heads off of the surface of the platters. On a normal internal hard drive, spin down is not usually an issue, because the operating system and the computer are handling this behind the scenes. Things change considerably with external hard drives.

It is unfortunately very common today for users to simply unplug the thumb drives or external hard drives when they're done using them. This is probably the number one failure point of external hard drives. In an orderly process of spin down for a hard drive, an instruction is issued that read/write activity to the hard drive must end because power is about to be removed. At this point the hard drive finishes its read/write operations and moves the heads to whatever parking area is relevant to that drive. Only at that time is it okay to disconnect the external drive. In instances where someone merely pulls the plug on the external hard drive because they are done with it, there may be times where the heads are still performing functions. If it is a case where heads are meant to sit on a parking ramp off the edge of the platters, but yet they are reading and writing data on the platter surface close to the spindle motor, when power is interrupted and the air bearing is reduced and disappears, heads will not have enough time to get back to the parking ramp before the air bearing disappears. This will cause the heads to contact the platter surface and potentially cause a sticking issue which we will discuss later in this module.

This is not a problem unique to spinning hard drives. This can also cause significant issue on solid-state drives as well. In the case of solid-state drives, blocks of data are constantly in motion. We will describe this in more detail later in this module, but suffice to say for now if blocks of data are being moved around inside the solid-state drive at the point that power is removed, that block of data may not get its address marker written to the proper database, thereby causing geometry issues that may cause the drive to no longer function.

Hard Drive Geometry

- Binary means two
- Binary Digit = bit
- A bit is 0 or 1
- 8 bits = 1 byte
- 1 byte is a single letter
- 512 bytes in a sector

0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	
2	x2	x2	x2	x2	x2	x2	x2	=256
2x1 =2	x2 =4	x2 =8	x2 =16	x2 =32	x2 =64	x2 =128	x2	= 256

How much data does your 1 TB hard drive hold? Surprise! Not 1 TB. But this depends on whose math you use. There is real math, and then there is marketing math. Let's have a look at each. But to understand it, we need to understand bits and bytes.

The smallest piece of digital information is a binary digit, or bit. [1] To be clear, these bits are not actual data, so much as an electrical state. A bit is either set or not set. A bit is either on or off. In a spinning hard drive, where magnetism is used to set a bit or not, a bit is then a magnetic polarization. The important takeaway is that there are only two possible values that can be obtained from a bit. Either a zero or a one. This does not give us very many options! That is why we string eight of them together. If we understand that 8 zeros and 8 ones equals 16 different possible states, we start to see why hexadecimal (base 16) is used in computing. 16 possible bit positions allows for 256 possible combinations of these bits. Another way to look at it is this:

1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0

2 X 2 X 2 X 2 X 2 X 2 X 2 X 2 = **256**

2 X 2 = 4 X 2 = 8 X 2 = 16 X 2 = 32 X 2 = 64 X 2 = 128 X 2 = **256**

So why 256? Why not have 9 bits in a byte? Or 10? Efficiency mostly, but that goes beyond the scope of this module. Being that the bit is the basis of all, let us now look at how that affects drive sizes. If there were 10 bits in a byte, then the power of ten would mean that 1000 bytes equals 1 KB, and 1000 KB equals 1 MB, and 1000 MB equals 1 GB, and 1000 GB equals 1 TB. But that's not how it works. With 8 bits in a byte, then 1000 bytes is actually 8000 bits, and not 10 000 bits. But 10 000 bits is actually 1024 bytes. 1024 bytes is what we

call a KB, even though it is technically a little bit (no pun intended) more. This spread between actual 1000 and represented 1000 gets larger and larger as data sizes go up. By using this actual math, 1 TB would actually be about 25 GB more than a TB, and manufactures are not going to give you 25 GB of data for free!

In the real world, it takes 8 bits (or any combination of 8 zeros and ones) to make a byte. For many languages including English, 1 byte equals one character (referred to as single byte languages)[2]. The letter 'A', the number 1, a space, or a period all utilize 1 byte of space on a hard drive, in order to be represented. Because 1 byte is 8 bits, then 2 bytes is 16 bits. 200 bytes is 1600 bits. It starts to become clearer when we look at it the other way. 1000 bits equals 125 bytes. In other words, not an even number. Because of the binary nature of data, if a company said their hard drive was 1 TB in size, it would actually be more than that. This is not good for business. If you can convince people that 931 GB is 1 TB, then you can tell people they are getting 1 TB, when in reality you are only giving them 931 GB, and you are coming out ahead.

Tera represents trillion. So 1 TB can also mean 1 trillion bytes. If you do the math that way, that is to say backwards, 1 trillion bytes converts to just slightly over 931 real GB. That is almost 70 real GB you are losing to the manufacturer. That is some weird math.

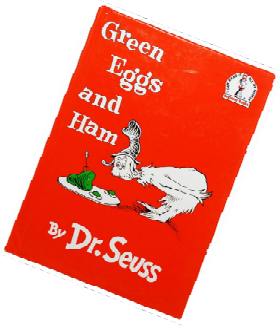
[1] Binary Digit | `https://for498.com/6-psz`

[2] Single/Multi byte languages | `https://for498.com/tkc3q`

How Much Data Is That? (1)

KiloByte (KB) – 1,024 bytes

- 2 KB: Typewritten page
- 3.4 KB: Green Eggs & Ham
- 100 KB: Photograph, low-resolution



KiloByte (KB) - 1,024 bytes

2 KB: Typewritten page

3.4 KB: Green Eggs & Ham

100 KB: Photograph, low-resolution

How Much Data Is That? (2)

MegaByte (MB) - 1,048,576 bytes

- 1 MB: Small novel
- 2.1 MB: War and Peace
- 5 MB: Complete works of Shakespeare
- 100 MB: 1 meter of books on a shelf

MegaByte (MB) - 1,048,576 bytes
1 MB: Small novel
2.1 MB: War and Peace
5 MB: Complete works of Shakespeare
100 MB: 1 meter of books on a shelf

How Much Data Is That? (3)

GigaByte (GB) - 1,073,741,824 bytes

- 1 GB: Paper in the bed of a pickup
- 2 GB: 20 meters of books on a shelf
- 11 GB: Stack of typed paper as high as CN Tower
- 20 GB: Audio collection of the works of Beethoven
- 30 GB: Read one novel per day, every day, for 80 years
- 50 GB: Library floor of books on shelves
- 160 GB: 25,000 feet (7.6 km) of printed data

Gigabyte (GB) - 1,073,741,824 bytes
1 GB: Paper in the bed of a pickup
2 GB: 20 meters of books on a shelf
11 GB: Stack of typed paper as high as CN Tower
20 GB: Audio collection of the works of Beethoven
30 GB: Read one novel per day, every day, for 80 years
50 GB: Library floor of books on shelves
160 GB: 25,000 feet (7.6 km) of printed data

How Much Data Is That? (4)

TeraByte (TB) - 1,099,511,627,776 bytes

- 1 TB: 50,000 trees made into paper (26 trees to build 200 m^2 house)
- 10 Terabytes: Printed collection of U.S. Library of Congress
- 15 TB: 100 photos per day, every day, for 80 years
- 42 TB: Music 24/7/365 for 80 years
- 120 TB: 24/7/365 of 1080p Video for 5 years

TeraByte (TB) - 1,099,511,627,776 bytes

1 TB: 50,000 trees made into paper and printed (26ish trees to build 200m2 or 2000ft2 house)

10 Terabytes: Printed collection of U.S. Library of Congress

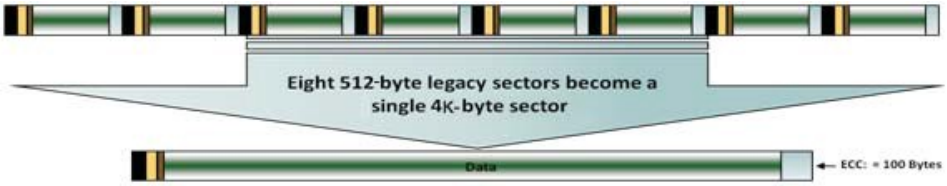
15 TB: 100 photos per day, every day, for 80 years

42 TB: Music 24/7/365 for 80 years

120 TB: 24/7/365 of 1080p Video for 5 years

Sector Layout

512 byte



4096 byte

Picture courtesy
https://for498.com/bsm4k

The smallest user addressable area on a hard drive is called a sector. In most cases up until about 2011, a sector, by default, was 512 bytes in size. You now know that this means, for example, 512 number 6s typed in a row with no spaces. Most drives today use something called Advanced Format, and these sector sizes are 4096 bytes. These are certainly not the only sizes for sectors. Many specialty drives are 520 bytes in size (usually only seen in big storage), and there are certainly a number of others.

Sector Layout Description [1]:

Gap section: The gap that separates sectors.

Sync section: The sync mark indicates the beginning of the sector and provides timing alignment.

Address Mark section: The address mark contains data to identify the sector's number and location. It also provides status about the sector itself.

Data section: The data section contains all of the user's data.

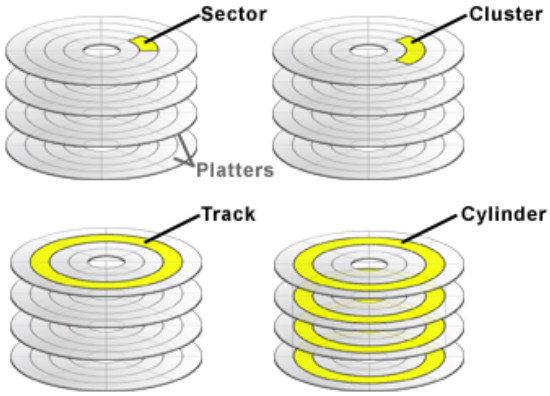
ECC section: The ECC section contains error correction codes that are used to repair and recover data that might be damaged during the reading or writing process. Consider this a checksum of the data section.

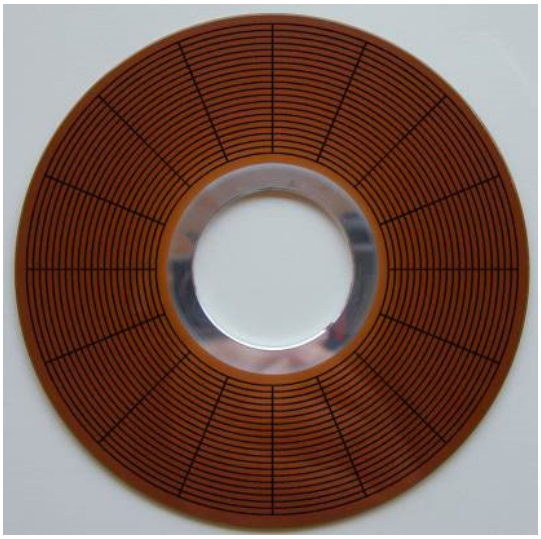
The Gap, Sync, Address Marker, and ECC are not part of the data portion of the sector. In other words, where a 512-byte sector will hold 512 bytes, these extra parts are in addition, and also take up space on a drive, although they are not addressable by the drive as we see it. This accounts for a significant amount of physical data space that could be better used to store actual user data. This was why the 4096-byte sector model was created. Instead of having 8 sectors of 512 bytes with their commensurate overhead per sector X 8, we now have 4096 bytes with only one set of overhead space. This has allowed hard drive manufacturers to increase drive size by 18-20% without having to change any of the physical parameters of the hard drive. Having said that, operating systems expect to see a 512-byte sector, so sector emulation occurs in order for the new drive to function properly.

As a point of reference, a 1 TB hard drive has 1 953 525 168 sectors. Using 512-byte sectors, this means that almost 98 GB of space is being given over to non-addressable platter surface. Moving to 4096-byte sectors means that less than 24 GB is being used for non-addressable platter surface. On top of all of this, there is the issue of recoverability of a sector based on platter surface damage, given the very tight areal densities in use today.

[1] Sector Layout Explanation | `https://for498.com/bsm4k`

Platter Layout





When a hard drive is created, there are certain standard layouts that are used. A platter has concentric rings across its entire surface on both sides. These rings are called tracks, and there can be as many as 300,000 tracks per inch of disk width when measured from center to the outer edge of the platter. This is part of what makes up the previously described areal density.

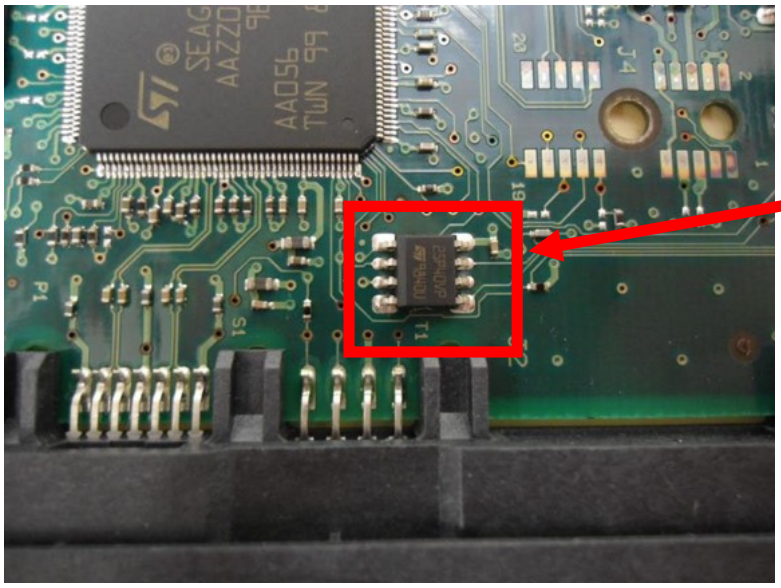
Within a given track, there are further divisions. These divisions are the previously discussed sectors. Given the shape of a platter, it stands to reason that 512 bytes in a sector closer to the center of the platter would have to be closer to each other than 512 bytes on the outer edge of the platter. In the early days of hard drives, this was the case, and as a result, sectors on the outer edges of a platter were read and written much more quickly than sectors near the spindle. As well, read/write errors were higher near the spindle. Platters of today employ a geometry called Zone Bit Recording [1] where sector density is the same throughout the platter. In other words, there are more sectors per track on the outer tracks than on the inner tracks. This again allows for creating higher data density without changing physical parameters of the disk.

The next layout parameter is the cylinder. Given a platter has tracks on both surfaces, it is easy to understand that if you were to draw an imaginary line through the platter, you would be on the same numbered track on the bottom side as you are on the top side. In the case of multiple platters, this means that these tracks line up through all of them. As a result, if you have a hard drive with 4 platters and you draw an imaginary line straight down through them, your line goes through 8 sectors. With 4096 byte sectors of today, this is a total of 32 KB of data that can be written or read without the read/write head stack having to move. Having said that, this Cylinder–Head-Sector (CHS) method of geometry has largely been replaced by Logical Block Addressing (LBA), as drives have become much more efficient, and thus need not harness the CHS method.

Although the diagram shows a value called a cluster, this is an abstraction that is utilized by the formatting utility, and is thus not discussed here.

[1] Zone Bit Recording | `https://for498.com/ry4xp`

P-List & G-List



When a hard drive is manufactured, an operating system is placed on it. This is not the Windows operating system or any other operating system that the average user is familiar with. In fact this operating system is not even visible to the average user and the average user has no way of seeing it, or indeed interacting with it.

Even many advanced users believe that zero sector is the first sector on a hard drive, but that is not true. It is the first positive sector on a hard drive. Hard drives have negative sectors and negative cylinders, and these are sometimes referred to as the firmware, service area, or servo tracks of the hard drive. This firmware is the operating system of the hard drive and tells the hard drive how to be a hard drive. It tells a hard drive it's make and model as well as its addressing scheme, and many other important parameters. It keeps track of many of the potential error conditions on a hard drive. It is so important that there are two copies in different places on the platter, and they are both under a different read/write head. On some hard drives, the second copy is actually on a chip on the PCB.

This firmware area can be gigabytes in size and is made up of many different files (called modules) that allow the hard drive to function. Two critical files within this firmware area are the P-list file and the G-list file.

The P list file or Permanent list file is created at the factory. No platter surface is absolutely perfect. There are going to be bad sectors on the platter when the hard drive is manufactured. As a manufacturing process lays the tracks in the areas on the hard drive, it reads and addresses them from the beginning. For example it will read zero sector, and if zero sector returns as being in good condition it will be marked as LBA zero. Sector one will now be read, and if it is in proper working order it will be labeled as LBA one. If the hard drive gets to sector two and notes that sector two is not in optimal condition, it will write that location to the P list and it will remove it from the addressable space. It will then go to the next properly functioning sector and label that sector as LBA two and so on. It is not uncommon through the manufacturing process for the P list to have thousands and thousands of bad sectors that are not addressable by the hard drive. In other words the hard drive does not know they exist and they do not interrupt the numbering scheme of the logical block addressing.

The second critical file we mentioned is the G list or Growing list. Once the hard drive is delivered to the customer and it is put into use, the firmware is functionally capable of keeping track of the sectors on the drive and watching them for performance issues. If it detects a sector that is going bad, for example sector 500, it will extract the data from that sector that is now going bad, refer that sector to the G list and take it out of service, and then label another available sector as sector 500 and place the copied data there. In this way the drive can continue to function even in the face of failing platter surface areas. A relatively common problem in hard drives is when sectors go bad so quickly that the firmware area does not have a chance to reallocate the bad sector. This will be discussed in further detail later in the module.

The information in the G-List and P-List (among other things) are kept track of in something called the translator. This is another module in the firmware area of the drive. If this translator becomes corrupted, your drive will not read properly, because there is no longer a translator keeping track of what sectors are bad. This translator can also be on the ROM chip, described below.

Something very significant to the forensics world is something called data hiding. With the right hardware, someone can create a significant amount of data on a hard drive, and then commit the sectors for that data to the G-List. Now no amount of forensics is going to find it. If the drive is wiped, it will show as having all zeros on it, but the P-List and G-List will remain untouched, with whatever data they had in them.

An important chip on the PCB, also known as the hard drive's motherboard, is the ROM (Read Only Memory) chip. This is typically an 8 pin chip (but can be 40+ pin and tied in with motor controller chip), and will usually contain data singularly unique to the hard drive it is on. This started around the 750 GB drive size territory. For many technical reasons beyond the scope of this class, drive manufactures had to address the vagaries of platter construction on a PER PLATTER basis. As a result, when the drive is built, one of the last functions involves laying the tracks and other geometry onto the platters. This geometry layout, as well as the translators it may create, is unique to that one hard drive in the whole world. This data is now written to the ROM chip. Damage or destroy that ROM chip, and NOBODY will get your data back. In most data recovery cases nowadays where the PCB is the problem, you must transplant the ROM chip, or where possible, write the data in the ROM chip from the patient drive to the donor drive, in order for data recovery to be possible.

Deleting Data

- Library of Congress
  - Dewey Decimal Card System
    - Helps us find where the book is
    - Very difficult to find the book without it

If card is destroyed, does book still exist?

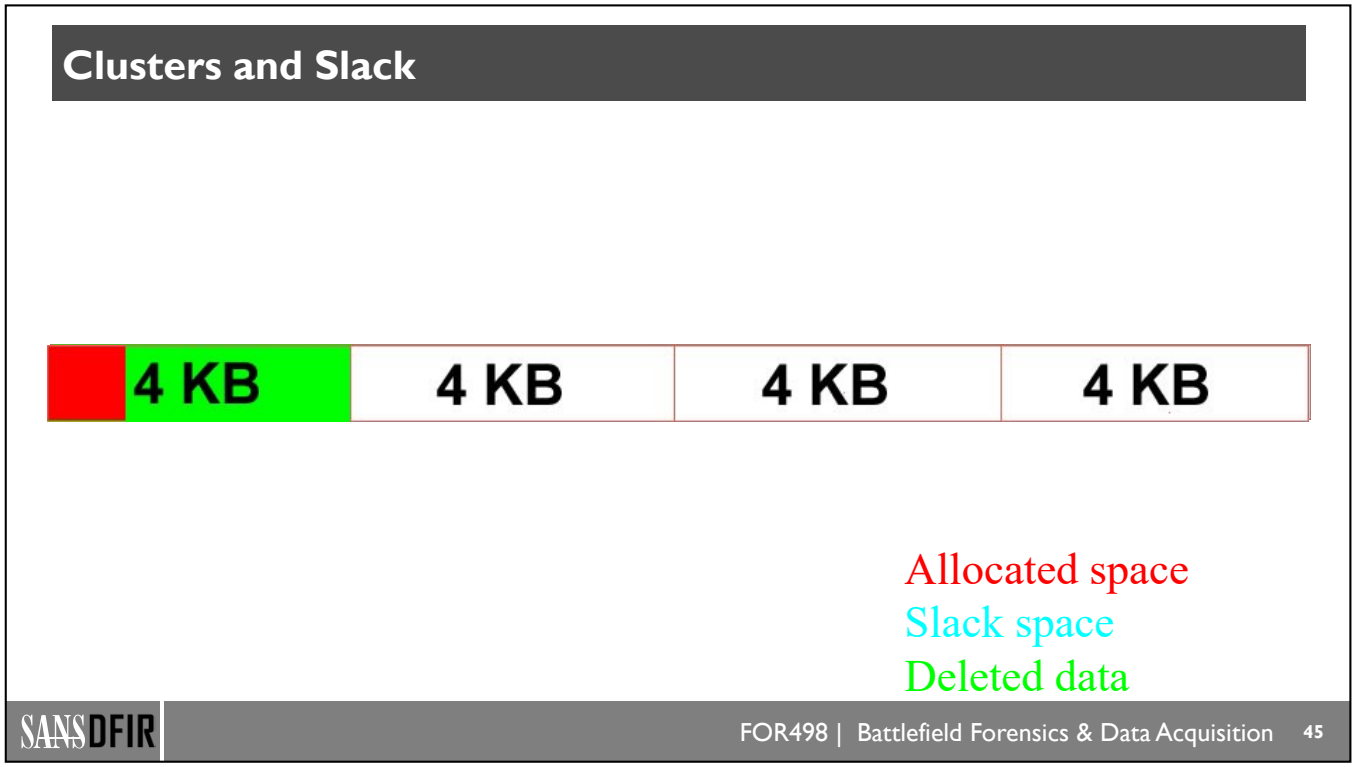
When a file gets deleted (and emptied from the Recycle Bin), it is not gone. Many of you have heard this, but how does it actually work? The Master File Table (MFT) on your computer is the table of contents for all resident files and folders. Its job is to keep track of a great deal of information about these files, but most importantly is the physical location of the file on the actual platters of the hard drive. It represents the file to the user in the form of an icon styled after the application it represents. For example, a Word document will have a Microsoft Word icon. This icon may be on your desktop, and you double click on it to open the file. However this is not your file. It is merely a pointer to the file. Although I should not use the term 'merely'. Let's look at what happens when a user clicks on an icon, and then we will come back to the deletion issue.

When you double click on the file, the computer first looks at the file extension of the file. In this case it is .docx. It then consults the Registry to find out what program it should use to open such a file. In this case, it would be told to use Microsoft Word. Microsoft Word will now consult the MFT to find out where the file actually lives on the hard drive. Once the MFT provides the information, Microsoft Word will reach out to the actual sector, and attempt to read the file. If the data at the sector is in a format that Microsoft Word understands, then the file opens for the user to interact with. For the sake of example, we will say the MFT had told Microsoft Word that the file lived at sector number 3 000 000.

When the file gets deleted and the Recycle Bin gets emptied, the file itself does not change. The icon for the file is no longer visible, and the MFT entry for that file gets altered to a state where it is no longer recognized by the live system. But the data is still there starting at sector 3 000 000. Since sector 3 000 000 is not referenced any longer though, new data can be written to that sector, effectively destroying the old data beyond recovery using today's capabilities. Unless and until new data writes to sector 3 000 000 though, the data that resides there can be recovered.

A great analogy is that of a library. For those of us old enough to remember, when you wanted to find a book in a library, you used something called the Dewey Decimal System. This was a bank of small drawers along a

wall that had filing cards in them. These cards had information (most importantly the location) about where a particular book could be found in the library. The Dewey Decimal System is analogous to the MFT. Deleting a file really only changes the MFT so that it doesn't visibly point to the file anymore. If I walked up to the Dewey Decimal System, opened a drawer, pulled out a card, and burned it, does the book still exist? Yes it does. But we can no longer find it using normal means.



We have discussed both 512-byte sectors, as well as 4096-byte sectors, and although the standard today is 4096 bytes per sector, because an emulator is telling the operating system that a sector is 512 bytes, that is what we will use in our discussion. A sector is the smallest addressable space for a hard drive. The next size up is a cluster, and clusters are made up of sectors. A cluster can be one sector in size, or it can be multiple sectors in size. The user has the option to set the size of the clusters on a given operating system. For the NTFS file system, although the default size of a cluster, also known as a block, is 4096 bytes, or 4 KB, the user can choose block sizes from 512 bytes through to 2 MB. For the ExFAT file system, the user can choose any block size from 512 bytes through to 32 MB!

Given the 512-byte sector size and the NTFS default of 4096-byte cluster size, it then takes 8 sectors to make a cluster. When this is used by the formatting function, the cluster becomes the smallest user addressable space on a hard drive. This means that if a 1 KB file was created, it would still occupy 4 KB of space on the hard drive. The cluster would now have 3 KB of unused space within the cluster. No other file can use this space. If a new file is to be created, it must start at the beginning of the next cluster boundary. As a result, your 1 KB file is using 4 KB of disk space. This is not an efficient use of space.

Slack space is that space from the end of the logical file to the end the cluster. There are two types of slack space. RAM slack, and file slack. RAM slack is the space from the end of the logical file to the end of that sector, and file slack is the space from the end of the logical file to the end of the cluster. In some instances, RAM slack and file slack can refer to the same space. Today we do not concern ourselves with RAM slack, as the file system writes zeros to this area.

How do we decide what cluster size to pick? Or is 4 KB always good enough? Why is there even a choice? It all comes down to efficiency. Let's use an example of a 1 TB hard drive. If you have 4 KB clusters, but all the files on your hard drive are 1 KB files, then even though you only have 250 GB of logical data, your 1 TB hard drive is now full. In a situation like this, it would be better to format your hard drive using a smaller cluster size. So why don't we just format a hard drive where each cluster is only 1 sector in size? Again, efficiency. Unless you have a specific purpose, requiring the file system to address so many clusters may be inefficient. If

most of your files are more than 1 KB in size, then a 512-byte cluster would cause your system to run very slowly. The sweet spot then, has been determined as 4 KB. Certainly if you have an external hard drive that contains only movies, you could format it with a large cluster size, allowing it to read/write more quickly.

Using the default 4 KB cluster size, if we were to write a 4 KB file to the volume, it would fill a complete cluster. Once we delete it, we know that the data still resides in that cluster, and will continue to do so unless and until new data is written to the cluster, thus overwriting the old data; and only enough of the old data that it needs to make space for the new data. This means that if the new file is 1 KB in size, it will cover the first 1 KB of the old file that still exists in that space. But the last 3 KB of the old file will still remain in the slack space. More importantly, nothing will ever happen to it unless the new 1 KB file is made larger, or is deleted, thus freeing up the cluster again. The remaining 3 KB will be there, available for recovery.

Now if this 3 KB of remaining data happens to be the last 3 KB of a photo, or some type of non-ASCII data, then it will certainly be unusable. But if it happens to be email data, or internet history, or some other type of typically small, plain text data, it is recoverable. As an example, 3 KB is 3 average emails without attachments! Cases have been won and lost based on data found in slack space. Never underestimate its importance.

Typical Photo File

So far, we have looked at how data exists inside one cluster of data. In the case of files that are larger than one cluster, the file at the initial time of writing will span as many contiguous clusters as it needs. In the slide, the photo is 25 KB in size. It completely covers 6 clusters, and 1 KB of a 7th. In this case, the only slack space that will exist is the last 3 KB of the 7th cluster.

Deletion (1)

Here lived once upon a time a wicked prince whose heart and mind were set upon conquering all the countries of the world, and on frightening the people; he devastated their countries with fire and sword, and his soldiers trod down the crops in the fields and destroyed the peasants′ huts by fire, so that the flames licked the green leaves off the branches, and the fruit hung dried up on the singed black trees. Many a poor mother fled, her naked baby in her arms, behind the still smoking walls of her cottage.

We will have one last look at what deletion looks like on spinning media. In this example, there is exactly 512 bytes of data. You can count every letter, space, and punctuation and fact check it. This amount of data is exactly 1 sector, and is resident data. It is intended to be viewed as though in a hex editing program at the disk level. As a forensic comparison, Guidance Software's EnCase would show this as above, as a sector of data.

Deletion (2)

Here lived once upon a time a wicked prince whose heart and mind were set upon conquering all the countries of the world, and on frightening the people; he devastated their countries with fire and sword, and his soldiers trod down the crops in the fields and destroyed the peasants′ huts by fire, so that the flames licked the green leaves off the branches, and the fruit hung dried up on the singed black trees. Many a poor mother fled, her naked baby in her arms, behind the still smoking walls of her cottage.

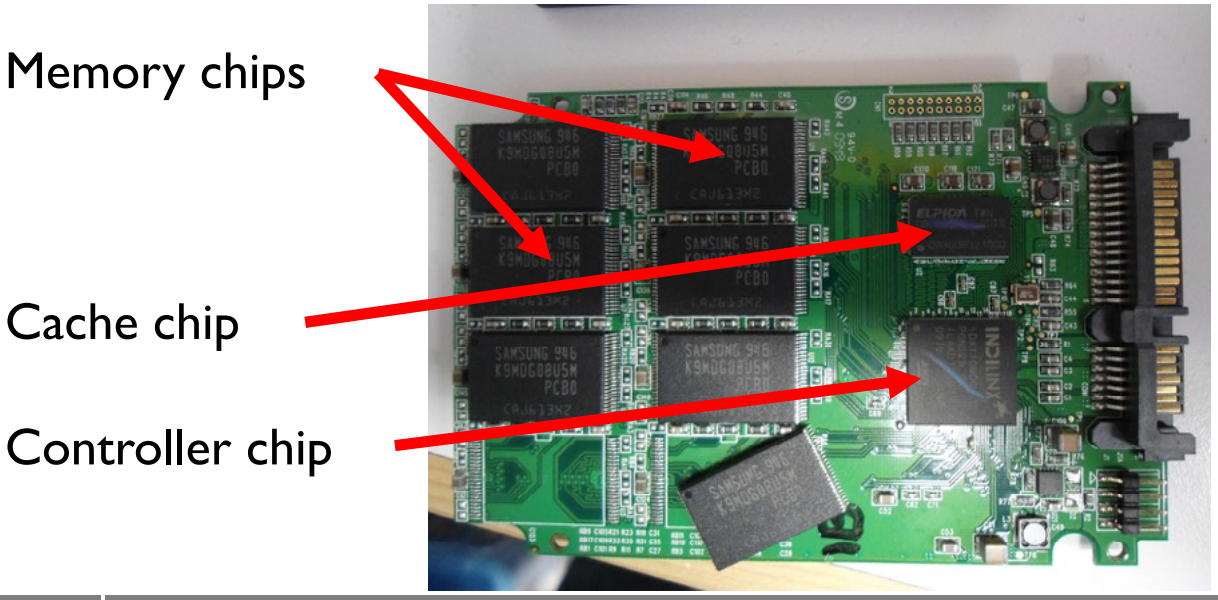
We now delete the data, and see that nothing has been changed. As viewed in EnCase, the only difference to this sector of data is that is now presented in red color, which is EnCase's default color to show deleted data. If our goal was to recover this data, you can see that it is entirely recoverable.

Overwriting

Mary had a little lamb, its fleece was white as snow. Everywhere that Mary went, the lamb was sure to go.ies of the world, and on frightening the people; he devastated their countries with fire and sword, and his soldiers trod down the crops in the fields and destroyed the peasants' huts by fire, so that the flames licked the green leaves off the branches, and the fruit hung dried up on the singed black trees. Many a poor mother fled, her naked baby in her arms, behind the still smoking walls of her cottage.

A user has now created a new file, and through quite a random process, Windows chose to save it to the above sector, simply because it was marked as available to receive data. The new data was written over top of the old data, but only for as much space as was needed for the new data. When viewed in a hex editor the remaining old data is still visible, or in EnCase you can see that resident data being black and deleted data being red, the actual representation is very clear. The remaining data from the old file is still visible, and although not complete, is certainly recoverable.

Solid State Drive Interior



Although some solid-state drives (SSD) appear to have the same connections on the end as a SATA hard disk drive (HDD), this is where the similarity ends. The obvious differences are the lack of moving parts, and the multitude of chips on the circuit board, not to mention the exponentially faster read/write speeds. Not as obvious are the significant power savings for not having to spin a motor continually, and the weight. Lifting an SSD that has the same form factor (shape and size) as a regular SATA HDD would have you believe that you are holding an empty case.

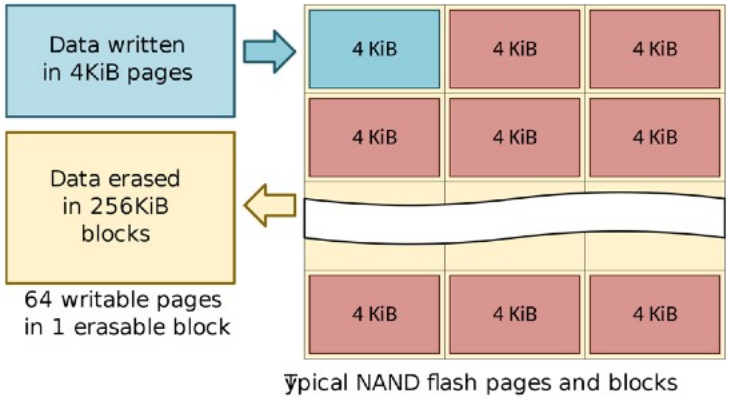
The chips on the circuit board that are used for storing data are known as NAND (NOT AND) chips. [1] The data on these chips is not written sequentially across one chip and then to the next as chips fill with data. The data is spanned across multiple chips based on parameters laid out on the controller chip. This process is called interleaving[2] and is done to allow for higher speed performance when reading and writing data. Hard drive manufacturers use proprietary algorithms to create this interleaving, so there is no given standard across various makes and models. Each drive can be entirely unique from the last one, and these differences can extend to hard drives for whom the labeling on the outside is identical.

There are three main problems that can occur on solid-state hard drives. The first is failure of the controller. In this case the controller no longer knows how to rebuild the data on the chips. A data recovery lab can potentially rebuild this interleaving manually, or access databases of algorithms that may be used to successfully recover the data. In some cases, each memory chip must be removed from the circuit board, placed in a reader, dump the data, and then pull all the data dumps together and try to rebuild their interleaving manually. The second is something called bit rot. This term comes from the spinning hard drive world and is somewhat of a misnomer. What happens in a case like this is that each cell holding a bit of data floats back to neutral, thereby losing the data position it was holding. This can be caused by long periods of time where the hard drive has not had power introduced to it. A third very common problem with solid-state hard drives is attributable to the previously mentioned unplugging the drive without properly ejecting it. The drive may be performing trim, garbage collection, or wear leveling functions when the power is removed, causing the data that happened to be in flux at that point in time to not be currently addressed. More on trim, garbage collection, and wear leveling later in this module.

[1] NAND memory | `https://for498.com/hteov`
[2] Interleaving | `https://for498.com/cz5in`

Cell/Page/Block/Layer

- Cell – 1 or more bits
- Page – 2 to 8 KB
- Block – normally 256 KB
- Layers – stacks of cells



In a solid-state hard drive at the chip level there is no magnetization as on the spinning hard drive. On a solid-state hard drive each bit is represented in something called a cell, and the cell is either charged or not charged. There are millions of these cells in a solid-state hard drive across each of the chips in the hard drive. One of the major differences between a solid-state hard drive and a spinning hard drive is that on a solid-state hard drive the cell needs to be returned to the neutral state before it can be used for future data. As solid-state hard drives get larger and larger, the methods that are used to make them larger without increasing form factor is through the use of layering of these cells. Technology has allowed for the stacking of the cells, and even the placing of more than one bit in a cell, however there is a performance degradation with having more than one bit in a cell. Having said this, a solid-state hard drive with stacked cells is still exponentially faster than a spinning hard drive. As previously described, one bit is represented in one cell. A row of cells is called a page (usually 2-8 KB) and a group of pages is called a block (usually about 256 KB).

Garbage Collection

Unlike spinning hard drives, a solid-state hard drive cannot hold deleted data unless and until new data needs the space. Each cell of deleted data must be returned to neutral before that cell can be utilized to hold new data. Even in the case of creating a file and then editing the file, the addition to the pre-existing file must now be written to entirely new sections of the hard drive, because one cannot simply change the cells where the existing file sits. In background operations of a solid-state hard drive, deleted data is kept track of for a function called garbage collection[1]. This garbage collection works from the idea that deleted data will be marked to be returned to neutral during certain idle times within the hard drive.

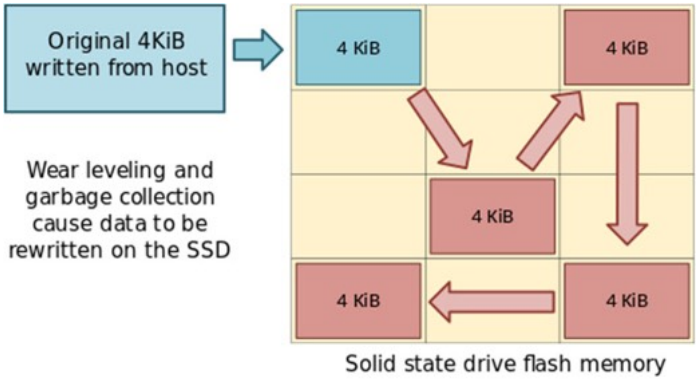
Garbage collection cannot be performed on anything smaller than a block of data, which is usually 256 KB of data. It is very possible that a resident file will exist within the block along with deleted data. Since garbage collection cannot zero anything smaller than a block it must first write out the resident data to a new block so that it can then send the pre-existing block with deleted data off to be reset.

Garbage collection is something that is performed at the drive level, and the operating system has no control over this. There is another function that is commonly mistaken for garbage collection and this function is called the trim[2] function. The trim function is a function that is controlled by the operating system. The trim function is a more efficient manner of garbage collection and will actually allow the drive to have a longer storage life than a drive that is doing garbage collection alone.

[1] Garbage Collection | `https://for498.com/73j6d`

[2] Trim Function | `https://for498.com/-t2s5`

Wear Leveling



Solid-state hard drives have a finite life. Each cell can only be used a certain amount of times before it wears out and becomes unusable. Because of this, if we had a 500 GB hard drive, but only used the first 10 GB, the usage and re-usage of the first 10 GB of the hard drive would cause the hard drive to die prematurely while 490 GB of the hard drive will have never been used. Because of this, hard drives utilize algorithms to perform something called wear leveling. This wear leveling moves data around all of the different cells of the hard drive so as to evenly wear out all of the cells on the drive. This contributes to a much longer life for the hard drive.

This creates a scenario not seen on spinning hard drives. Imagine you have data written from sector 0 to sector 128 of the drive. Because of wear leveling, these 128 sectors will be moved to a new 128 sector space somewhere else on the hard drive. However the operating system must still be able to access these 128 sectors and believe that they are in the same place they've always been. As you can see, this causes a high degree of intricacy in the solid-state hard drive, having to keep track of the movement of all data while still representing the location of this data as a static address for the operating system.

This causes a new layer of complexity in the forensics world as well as the data recovery world. It causes deleted data to be overwritten and/or destroyed far more quickly, and it also means that when orphaned data is found in unallocated space on the hard drive, attribution and context become much more difficult to determine.

It stands to reason that there must be free space on the solid-state hard drive to allow this movement of data. The solid-state drive has a certain percentage of drive size in addition to the reported drive size. This additional drive space is not provisioned by the hard drive and does not carry addressable space unless or until the hard drive wants to use it for the process of wear leveling. If a hard drive gets too full, it becomes harder and harder for the hard drive to perform wear leveling functions because there is not enough available space to perform this efficiently. As a result, if the drive gets beyond 70 to 80% capacity, speeds degrade significantly. The user has the ability to create more space for wear leveling as well as garbage collection and trimming if they so choose. It is as simple as formatting your hard drive and creating the partition to be smaller than the actual drive size reported. For example if you have a 500 GB drive and you create your volume at 400 GB, the solid-state drive will use the remaining 100 GB as additional space to perform trim and wear leveling functionality, thereby allowing the drive to remain exponentially quicker then it's spinning relative.

Data Recovery Introduction

- Recovering data that has been lost for any reason:
  - Accidental deletion
  - Accidental formatting
  - Intentional destruction
  - Storage media failure of electronic components, failed firmware code, platter surface damage, or read/write head stack failure

The data recovery practice is incredibly deep, and we can't get into every facet of data recovery, but the goal is to hit the main points, explain some of the things that can go wrong, and dispel some of the rumors that are found on the Internet.

Data loss covers a range of possibilities. A user can experience data loss from something as simple as accidentally deleting a file, to accidentally formatting a hard drive, and even something as extreme as reinstalling the operating system on top of their data.

Data loss can also be as a result of malfunctioning of the storage medium itself. This can extend to things such as platter surface degradation, failure of the read/write head assembly, failure of the microcode in the service area, and any number of electronic failures present on the PCB. Half of the battle in terms of successful recovery comes down to a proper diagnosis. There is no one size fits all. You must know what you're up against before you start throwing ill-mannered attempts at the recovery.

In the case of mechanical failure, a situation where the user can no longer access the data, a data recovery lab may be contacted. This is usually the first indication that anyone has of how expensive such a venture can be. Proper professional data recovery labs commonly have recovery prices starting at US$1000 and easily climbing to $4000 and more. People are quite shocked at this, complaining that they could buy a whole new computer for less money than that. While this is true, no amount of new computers will have the customer's data on them.

There are very few actual professional data recovery labs with the ability to perform all manner of repairs and recovery. As well there are very limited training opportunities. There are no schools that offer data recovery as a curriculum. Data recovery engineers have typically learned on the job, and the difficulty among data recovery labs is retaining these engineers after they have been trained. Another consideration in the pricing of data recovery labs is the cost of tooling. It can cost well past $100,000 to set up a lab environment. Couple that

with the fact that any reputable data recovery lab will not charge any money if they are unsuccessful at recovering data and it becomes easier to see why pricing is as it is.

Data recovery, quite frankly, is an industry that should not exist. If users maintain proper backups of their data, then hard drive failure should be nothing more than a minor inconvenience.

Diagnosis

Diagnosis of Problems

Smell

Look

Listen

When it comes to diagnosing potential issues with a hard drive, you have a starting point. For example, you will know if you accidentally deleted a file or accidentally reinstall the operating system on top of your files. In these cases, software recovery solutions will suffice, but which one, and how should it be performed?

Suggesting the above is not the case, you may see a situation where you plug in an external hard drive and it fails to be recognized. Or you hear noise. Or worse, you don't. This crosses over into the realm of potentially mechanical failure. In these cases, advanced data recovery technology and expertise may be necessary. Having said that, there are certain steps to take before any attempts are made.

In the case of a drive that you have come across that is not working, or more importantly a drive that someone has given you to look at, there are three very important steps you must perform first before any data recovery attempts are made. The first is to smell the drive. Yes that sounds odd, however there are certain issues on a hard drive that can cause electronics to fail. These things may include things like substandard power supplies. When electronics fail on the circuit board the smell is very obvious. If you smell a hard drive and smell burnt electronics, do not plug the hard drive into a computer. This will be discussed further in the module.

Suggesting the device does not smell, the next issue is to look at it. Are there any visible dents or scrapes on the chassis of the hard drive? Visually inspect the PCB for cracked or blown chips. If all of this seems in order, the next step would be to introduce power to the device. If it is a bare drive, just plug in the power cable and not the data cable. At this point listen very carefully to the drive as it spins up. In cases like this a stethoscope can be very helpful in listening to the drive. You may not know exactly what you should be listening for but some of the more obvious problems will be easy to determine. If you are unsure of whether or not the drive sounds like it is operating properly, it is a good idea to run the same test on a known good hard drive and then compare the two.

Data Recovery Software

- Free tools, in general, are not that good.
- Good tools give you a free preview to see if they will work ON YOUR PARTICULAR PROBLEM

R-Studio	GetDataBack	RAID Reconstructor	ReclaiMe	PhotoRec
• r-studio.com	• runtime.org	• runtime.org	• reclaime.com	• cgsecurity.org

There is no shortage of free data recovery software on the Internet, purporting to help anyone with any data loss problem they may be having. As with anything that is free, you're getting what you pay for. Understand that many of these free tools have been placed online by people who have written them for a particular problem. In other words someone with the knowledge to write a program had a data loss situation, was able to identify why the situation was caused (or got lucky with a guess), and wrote a piece of software to address that particular problem. They then decided that it would be a gracious thing to place it on the Internet for other people potentially having a problem. The issue with this is that unless you're having exactly that same problem, the software will not help you, and in many cases may cause bigger problems. Unfortunately, the average user will have no way of knowing or understanding this.

There are a number of paid programs on the Internet that are quite cost-effective. Most any of this software retails for less than $100. Any reputable tools that are designed to handle software-based data loss will usually offer a trial version of the program to run your recovery and see what can be recovered and how. This is a 'try before you buy' method. If the software shows that it can recover your data, you then have a feature available to purchase the software and actually perform the recovery.

The best program generally for a software-based data loss situation is a program called R-Studio. R-Studio will rebuild a drive in a number of different ways and for a number of different types of filesystems. R-Studio is also a very capable tool in assisting in the rebuilding of RAID arrays. Be careful because there are copycat companies that market their products under the name rstudio. This is not the same product!

Probably the best program on the market today to address the situation where an operating system has been re-installed and you need to recover the data that existed prior to this reinstallation, is a program called GetDataBack.

If you have a RAID array that has failed and you have no visibility into, or knowledge of the setup of the particular array, a program called RAID Reconstructor is a fantastic tool for doing this.

If the pre-existing volume or partition information is so damaged that no software can see it, a great tool of last resort is PhotoRec. This tool is especially useful if you are in a data loss situation with a hard drive that contains thousands of photos. There is probably no more capable tool than PhotoRec for recovering lost photos.

The key is in understanding that no single tool is the be all and end all of software data recovery solutions. The best tool for the job will always depend on what your particular data loss situation is. Given that these tools all have trial versions, it makes sense to use more than one tool in an attempt to recover your data. Different tools will present your data in different ways and then you can select the best tool for your situation. It is worth noting that, for example in an operating system reinstallation scenario, GetDataBack can conceivably take days to scan a hard drive prior to rebuilding.

It is extremely important that you do not place the data recovery software on the media that contains the data you are trying to recover. It is also a very good idea to write block the media you are trying to recover so that tools do not inadvertently overwrite the very data you are trying to recover.

R-Studio – r-studio.com

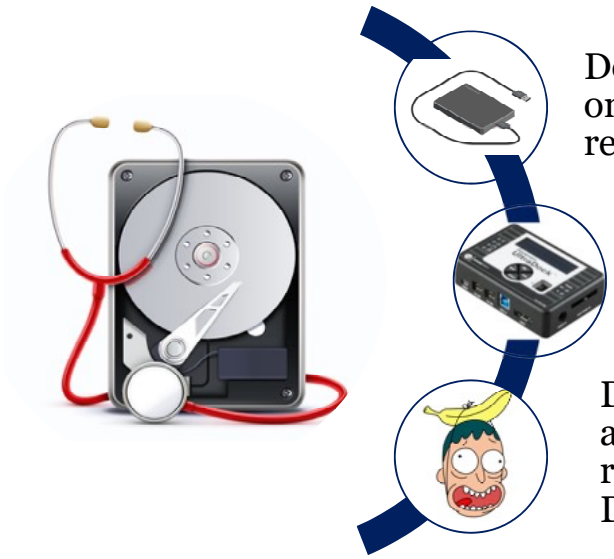
Get Data Back – runtime.org

RAID Reconstructor – runtime.org

ReclaiMe - reclaime.com

PhotoRec - cgsecurity.org

Local Diagnosis and Solutions



Do NOT run data recovery tools on the drive you are trying to recover data from!

Plug drive in through write blocker, or you risk overwriting the very data you are trying to recover.

Do not throw feeble, unknown attempts at a hard drive, if data recovery is mission critical. (i.e. Do not do anything crazy!)

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 60

Here are some guidelines to consider when trying to decide what might be wrong with a hard drive, and what to do about it.

If the drive is making a knocking or clicking noise, what kind of knocking noise is it? More on this clicking in the next slide.

Don't leave a bad hard drive running simply because you think you may never get it going again. If you have failing heads or platter degradation, continuing to run it and trying to throw (insert free downloaded tool here) at it will not work, and will just hasten the demise of the heads, and or destroy the data to the point of non-recovery. If it is clicking, shut it off and get help.

It is true that nobody wants to pay for data recovery, and many think it is a rip off. The other side of the thought process is this. It is shocking that a user will take their most valued things (baby pictures, corporate data, etc.) and not keep it safe (backups).

This is also not a time to be looking for budget solutions. Nothing will render your data unrecoverable more quickly than a cheap lab's attempt at recovery. Every reputable lab has a host of stories they have seen and heard about someone who has taken their drive to a cheap lab first. Online searches will show these labs in almost every location. In many cases these are nothing but mail drops and not actual physical locations. These places will advertise data recovery for a mere few hundred dollars. You must understand how these labs operate. In an industry whose tenet is 'no data no pay', lesser labs must turn more hard drives in order to make money. If your drive is sent to them with a head stack failure, very often the drive will be called unrecoverable and returned, without any repair attempt being made because head stack changes are hard, time consuming, and never guaranteed. The customer will then be left with the idea that their data can never be recovered and walk away from it. As it turns out, this is not the case.

Simply put, data recovery cannot be done for a few hundred dollars. In the case of a head stack failure for example, there is no hard drive parts store for a data recovery lab to go to. Data recovery labs must source the hard drive from companies who stockpile hard drives. When they contact these companies, these companies know exactly who they are because they're not just asking for a simple hard drive. They are asking for a certain hard drive of a certain family of a certain make and model with a certain revision code. There is only one industry that asks for that level of detail. The wholesalers know exactly what data recovery companies charge and so if a data recovery company needs to buy a hard drive of a very specific make model and revision number, that hard drive, although it may only be a 500 GB hard drive, is going to cost them sometimes $300 or $400 to purchase. If the data recovery is subsequently unsuccessful the data recovery lab eats that cost. As you can see, in the case of mechanical failure no lab can do a successful data recovery job for just a few hundred dollars.

At the end of the day, the point here is to know what you are doing and why you are doing it, because otherwise you can destroy your drive irretrievably, and never know it. If you must have the data, don't play around with it. The most well-meaning efforts can cause the data to never again be retrievable.

If the data is expendable, and/or you don't want to spend the money, then you can experiment, but since your experiments may often destroy things along the way, have a reasoned plan and expectations, and apply sound methods that have been thought out. Cold plates get used frequently. Freezers do not. Nor do we pour liquid into our drives.

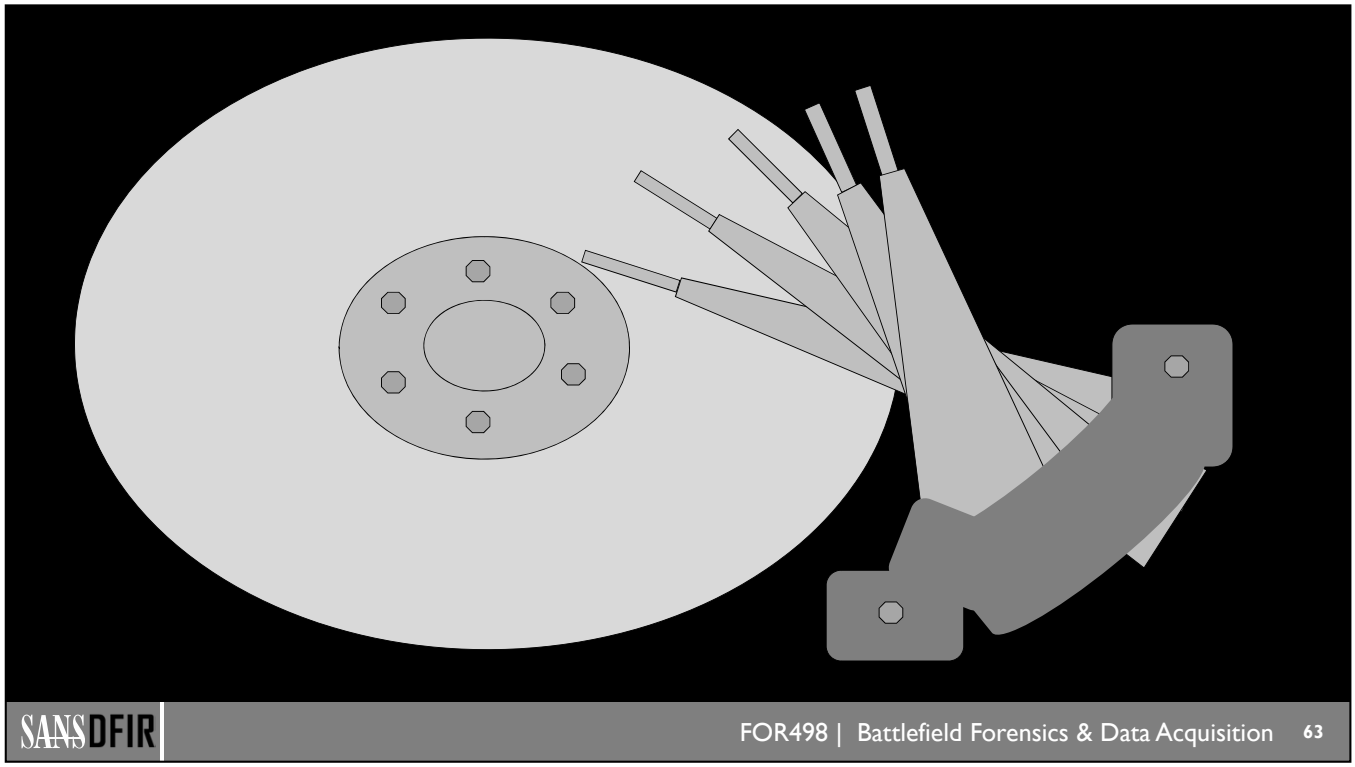
Some great, extra-curricular reading can be had here:

My Hard Drive Died – Scott Moulton | `https://for498.com/9-hxj`
HDD Guru | `https://for498.com/-uxs9`

Click of Death

The aptly named 'Click of Death' is widely believed to be failed heads and is further widely believed to be the sound of read/write heads banging on the platter surface. This is not correct. The dreaded sound is actually the head stack voice coil hitting a park post that limits movement of the head stack, and not the heads actually hitting anything. When a hard drive first spins up it will attempt to find zero sector and start reading from there, simply because that is the sector where the information resides that tells the hard drive where the data is. If the heads cannot find zero sector, for example in the case of platter surface degradation or head failure, the head stack will reset itself back to the parking ramp location, or the outer edge of the platter, and restart its attempt to find zero sector. Each time the head stack re-parks itself, it will have reached the outer limit of its range of motion. Inside the hard drive housing exists a metal or rubber post that will stop the head stack from traveling any further than it is supposed to go.

Is it a cyclic, nonstop knocking that starts immediately when the drive spins up and never changes in tempo? This is usually indicative of head stack failure. The only thing that will get your data back is a head stack transplant. If it clicks intermittently, that may be a sign of failing heads, degrading platter surface, corrupted service area module, corrupted G-List, etc. The point is, there is nothing an end user is going to do about this.



This page intentionally left blank.

Professional Tools

- In some situations, software may not be enough to recover data
- There are many advanced tools that can be leveraged by a laboratory using specialized equipment
- The tools below are some examples of popular tools used by many labs

PC-3000	PC-3000 Flash	Data Extractor	Deepspar Disk Imager
• acelaboratory.com	• acelaboratory.com	• acelaboratory.com	• deepspar.com

Among many tools like rework stations, heat guns, soldering irons, oscilloscopes, microscopes, clean stations, finger cots, toolsets, and an unending supply of patience, a data recovery lab will use extremely specialized tools like head combs, platter removal tools, spindle motor tools, laser measuring tools, etc. Many of these tools are available from a company called HDDSurgery [1].

Any professional data recovery lab will also use a tool called PC-3000, PC-3000 Flash, and Data Extractor. These are all manufactured by Ace Laboratories [2]. Finally, for the data recovery portion itself, a tool called Deepspar Disk Imager, from Deepspar Data Recovery Systems [3].

On top of tooling, there is the expense of a hard drive parts inventory. The author's lab has over 3000 hard drives in inventory, and yet probably 30-40% of the time, we will not have the right one for a recovery and have to order one specifically for a job.

Most any professional data recovery company will provide FREE estimates (except on RUSH jobs), and will have a No Data, No Pay policy for the vast majority of jobs (usually only excluding accidental deletion recovery, or attempts at self-repair).

[1] HddSurgery | `https://for498.com/jw2-h`

[2] Ace Laboratories | `https://for498.com/3n5ti`

[3] Deepspar | `https://for498.com/5imp3`

A professional data recovery lab will contain many tools that will not be found in an average person's garage. Let's take a look at just a few of these tools.

A class 100 clean station is a must have in a lab where hard drives will be opened up. A hard drive cannot be opened up in normal air. The smallest piece of dust in the air landing on a platter surface will destroy a head stack once the drive is spun back up. Hard drives are opened up in the clean station so this does not happen. The station is merely a large box with an open front, and a large fan on the top. Sitting on top of the fan will be a HEPA filter that blocks particulate to a certain size. This station is a positive pressure environment where the fan sucks air through the HEPA filter and blows it down into the station and out the front.

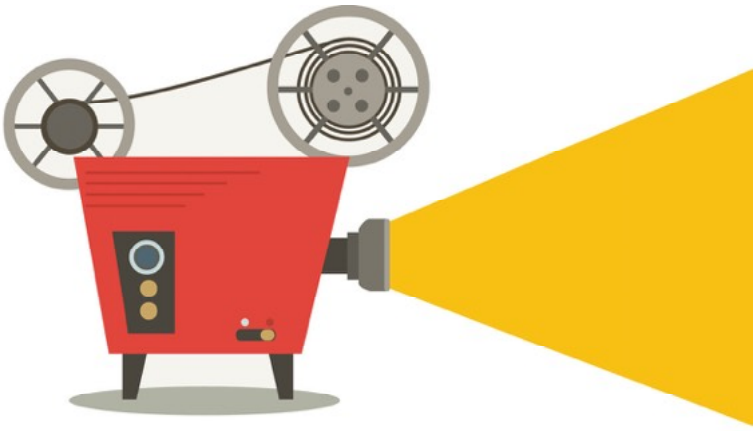
A heat rework station is used to desolder chips that have multiple pins. For example, if a ROM chip needs to be transplanted, it has eight pins that all need to have solder melted at the same time. More complex situations with certain chips called ball grid array chips have dozens of pins or connection points underneath them connecting them to a circuit board. A soldering gun with multiple tips is necessary because in some circumstances you may have to improvise with certain wires coming off of certain pads of the circuit board to perform various functions.

A microscope is used to assist in performing microscopic work, as well as in assisting with diagnosis of cracks, and reading the very tiny writing on many chips.

A laser plane is used to scan the surface of a platter to check for any warpage.

There are many more specialty tools used in professional data recovery labs, such as tools that will help unstick seized motors, and devices that fit around a stack of platters to prevent rotation of the platters during a transplant.

Movie Time!



- 6_2: Data Recovery Tools

This page intentionally left blank.

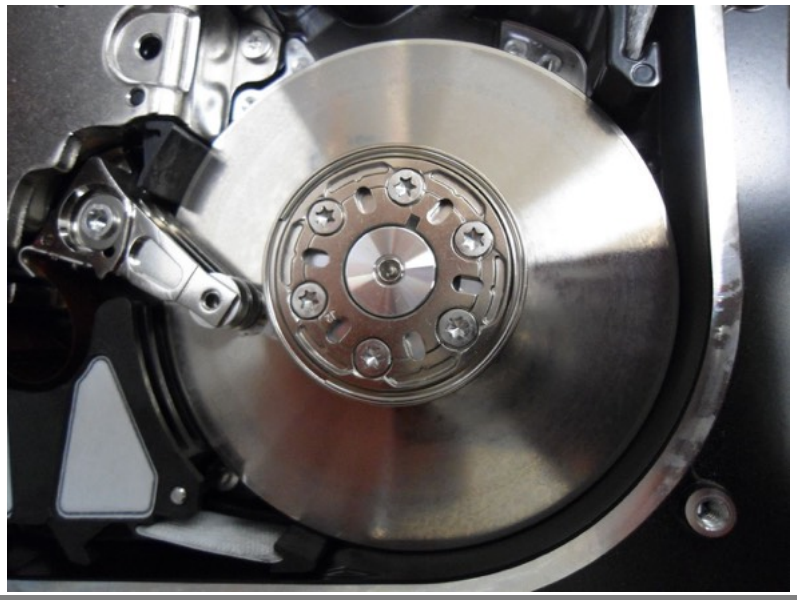
Hand Tool Set

This page intentionally left blank.

Hard Drive Platter (1)

This is a very clear representation of how perfectly polished the surface of a hard drive platter is.

Hard Drive Platter (2)



This is very clearly a destroyed hard drive platter…

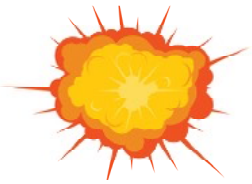
Hard Drive Platter (3)

…as is this one. All of the dark, dusty looking matter throughout the inside of the housing used to be data on the platter surface.

Common Problems

Stuck heads

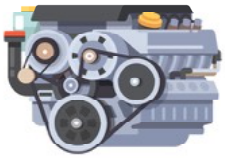
SMART exceeded



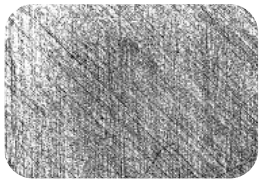
Blown TVS

Corrupt ROM

Dead heads

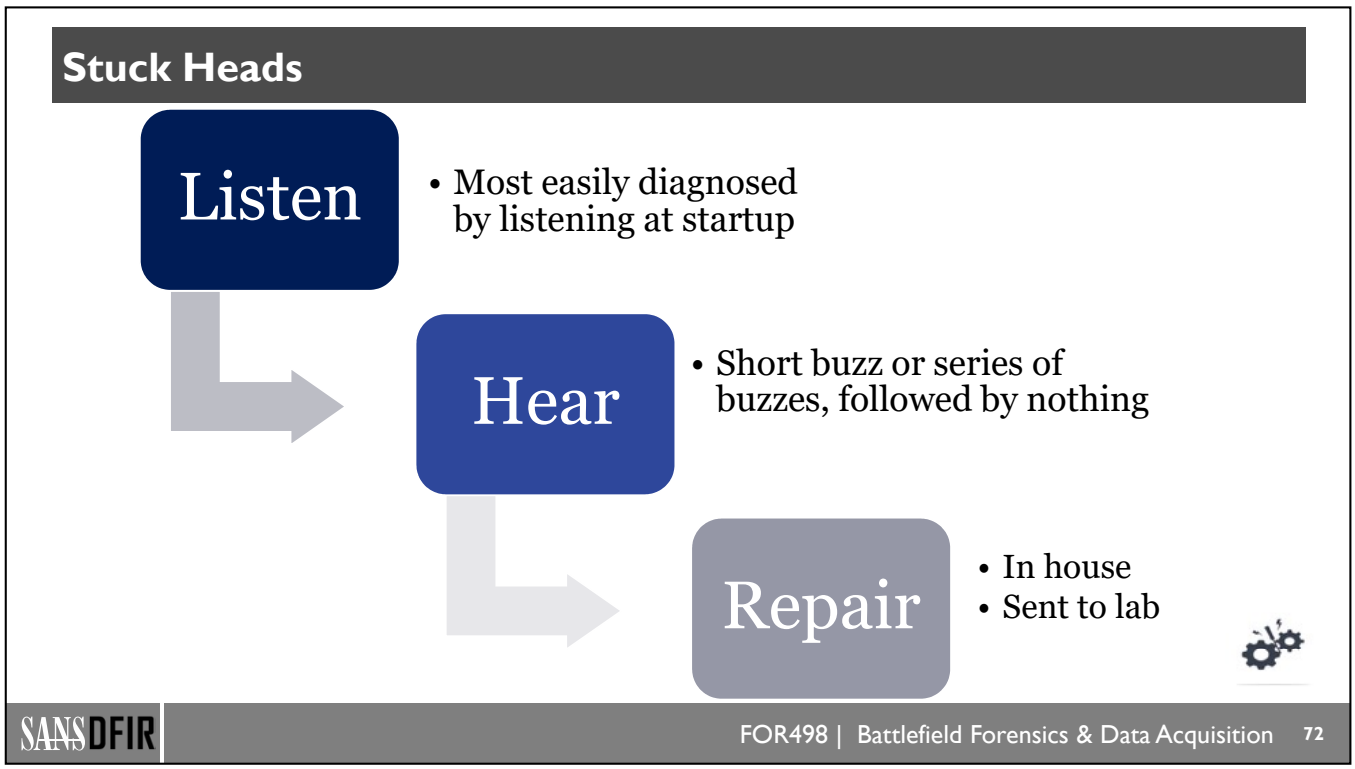


Seized motor



Degraded platter surface

There are many complicated issues that can befall a hard drive and cause a data loss situation. We will go through a short list of the more common things that can happen to a drive in the following slides. Although we will talk about these problems in general, the correction or repairing of each given situation is different virtually for every make and model of hard drive you might encounter. This is not a case of where ‘a hard drive is a hard drive is a hard drive’. As well, with any of these situations, no tool on the Internet, nor the oft touted “freezer trick” is going to fix any of them.



This is most commonly seen on external drives, where you unplug the USB without actually properly ejecting it first. On most hard drives today, the heads move off the platters to a parking ramp before the drive powers down. If you unplug the drive without properly ejecting it, sometimes the heads won't get over to the parking area before the platters spin down and the air bearing is lost. If this happens, the heads fall directly on the platter and stick there. The reason for this sticking is due to the surface tension. The platter surface is polished to such a fine finish as seen in a previous slide that when the head makes contact with it, it sticks to the surface so tightly that the motor is not strong enough to rotate off of the heads. If you have ever tried to separate two pieces of glass that were laying on top of each other, you will know that you had to slide them apart in order to separate them, as opposed to pulling them apart. The reason you couldn't pull them apart is again because of surface tension.

In a data recovery lab the technician will use a specialized technique in the form of special tools as well as an ultrasonic cleaning bay to separate the head stack from the platters with minimal damage. There are a number of YouTube videos that actually show people opening hard drives in normal air and simply using their fingers to move the head stack back over to the parking ramp. If you attempt this and are successful, you are truly lucky indeed. The surface tension is so strong that in many cases your actions may tear the head off of the armature and leave it stuck to the platters. If this occurs on the underside of a second or third platter, you may not see this until the armature that no longer has a head scrapes along the entire platter surface on its way to the parking ramp.

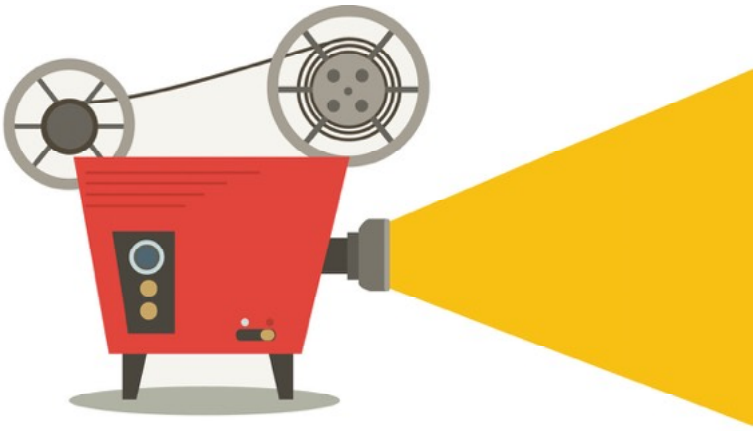
If you have a situation like this, it is obviously best to refer it to a data recovery lab, however there may be reasons why you will not consider this. One of the reasons could be the cost. Another reason could be the material that's on the hard drive. If you have thought of all of the possibilities and have decided that going to a data recovery lab is simply not an option, there are ways to manually unstick the heads in such a manner as to minimize the possibility of damage. This technique is NOT recommended, however if you are going to try to unstick the heads yourself, you might as well be armed with the best possible information. We have created a video of the procedure to help maximize the possibility of success. It must be understood, and cannot be overstated, that this is not recommended.

If you are going to attempt this procedure, you must ensure you are in a clean area that has been wiped down. You must also be wearing gloves or finger cots. Once you have completed the task, ensure there is no dust particulate on the platter surface. If there is, do NOT blow it off. Moisture from your blowing could eject onto the platter surface, creating other problems. Do not use compressed air either! Use a bulb syringe [1] only. These can be found online, or at most any pharmacy.

This situation is commonly misconstrued as a dead motor, because the drive is not spinning. In many cases though, at the moment of the introduction of power, there may be a faint beeping of the drive, and/or possibly a very quick and quiet click.

[1] Bulb Syringe | `https://for498.com/w-eru`

Movie Time!

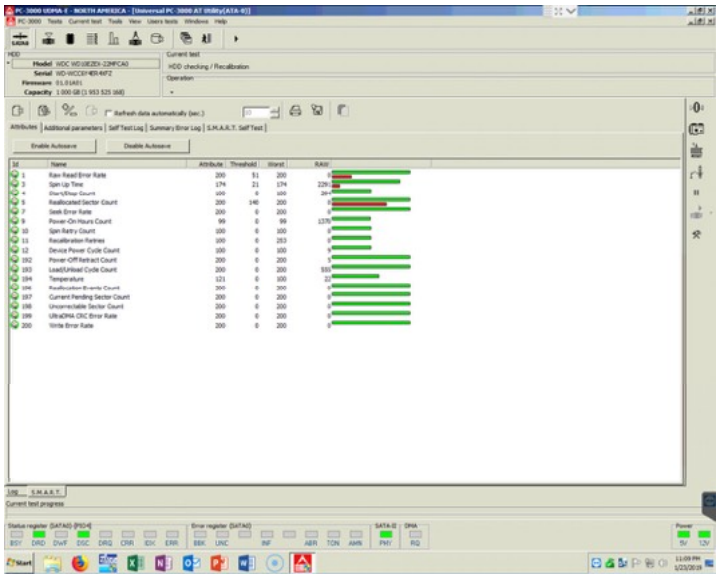


- 6_3: HDD With Stuck Heads

This page intentionally left blank.

SMART

- **S**elf
- **M**onitoring
- **A**nalysis
- **R**eporting
- **T**echnology



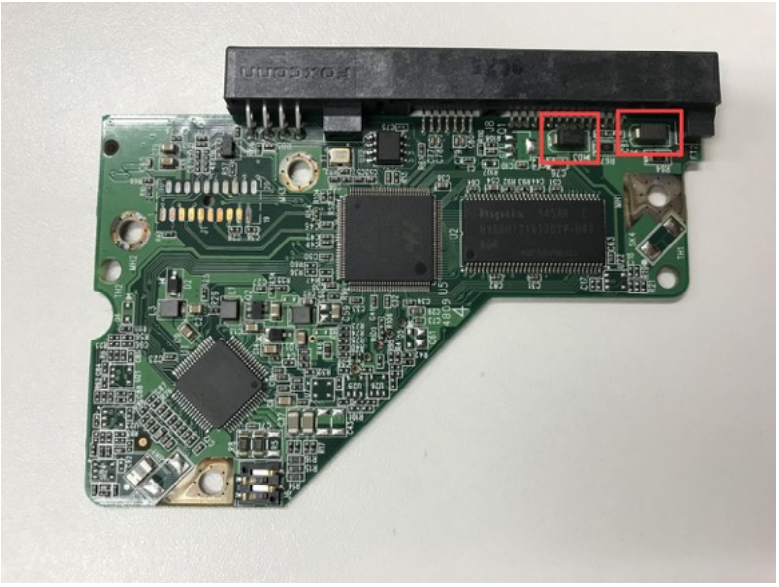
The SMART [1] on a hard drive is a component of the system area of the hard drive. It monitors a number of different parameters of the hard drive, including things like how long it takes to read a sector of data or write a sector of data. It can also monitor drive temperatures and other variables. For each of these variables, the smart defines a maximum threshold that it will allow before it will push the drive into a fail state. Once the SMART exceeds these thresholds the drive can become unstable or unusable. Although this will present itself as a full data loss situation, it can be as simple as either resetting the SMART, or telling the drive to turn SMART off. This sounds simpler than it is, because you normally can't do it just from the computer itself. GSmartControl on your VM will perform this function, but the drive must be readable by the OS first, in order for GsmartControl to address it. Lab level data recovery software can manipulate SMART in ways that will allow a technician to ignore it or disable it. There are tools online that will also provide the ability for this, however we make no recommendations.

[1] SMART | `https://for498.com/j5u4k`

Blown TVS

- Transient
- Voltage
- Suppression

Simply remove TVS



A hard drive contains at least one and sometimes more than one fuse. Fuse is an incorrect term though, in that the actual component does not work like a normal fuse. When a normal fuse blows, what has actually happened is a piece of wire inside the fuse has burned through, thereby breaking the flow of current. Rather than a fuse, a hard drive has a component called the TVS or Transient Voltage Suppressor. This device works opposite of a fuse. During normal operation there is no connection between two metal contacts inside the TVS. In the case of an overvoltage, the suppressor will fail and cause the metal contacts to touch each other, thereby completing the circuit, allowing power to pass through, and creating a failed state for the hard drive.

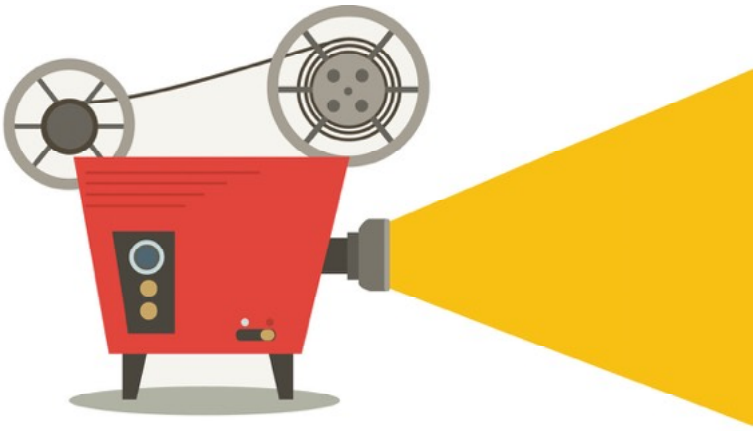
In a situation like this, the drive will present as being completely dead. When you plug it in there will be no sound or activity on the drive whatsoever.

This is caused almost exclusively by a cheap power supply in a computer. One of the jobs of a power supply on a computer is to regulate the current that flows to the motherboard and all of the computer components. There are capacitors on the board that will store electricity so as to provide a constant supply in the case of brown outs or reduced current flow. A capacitor can also absorb a certain amount of power. A good quality power supply will also have other components within it to deal with over-voltages. If the current travels through the power supply and the power supply does not stop the overcurrent, the last line of defense is the TVS.

This is a very easy situation to diagnose because you can smell the burned component. If you unscrew the PCB off the hard drive and look at the chips, you will see in the vast majority of cases, the chip that is damaged. It will be bubbled, and/or cracked, and it will smell very strongly like burnt electronics.

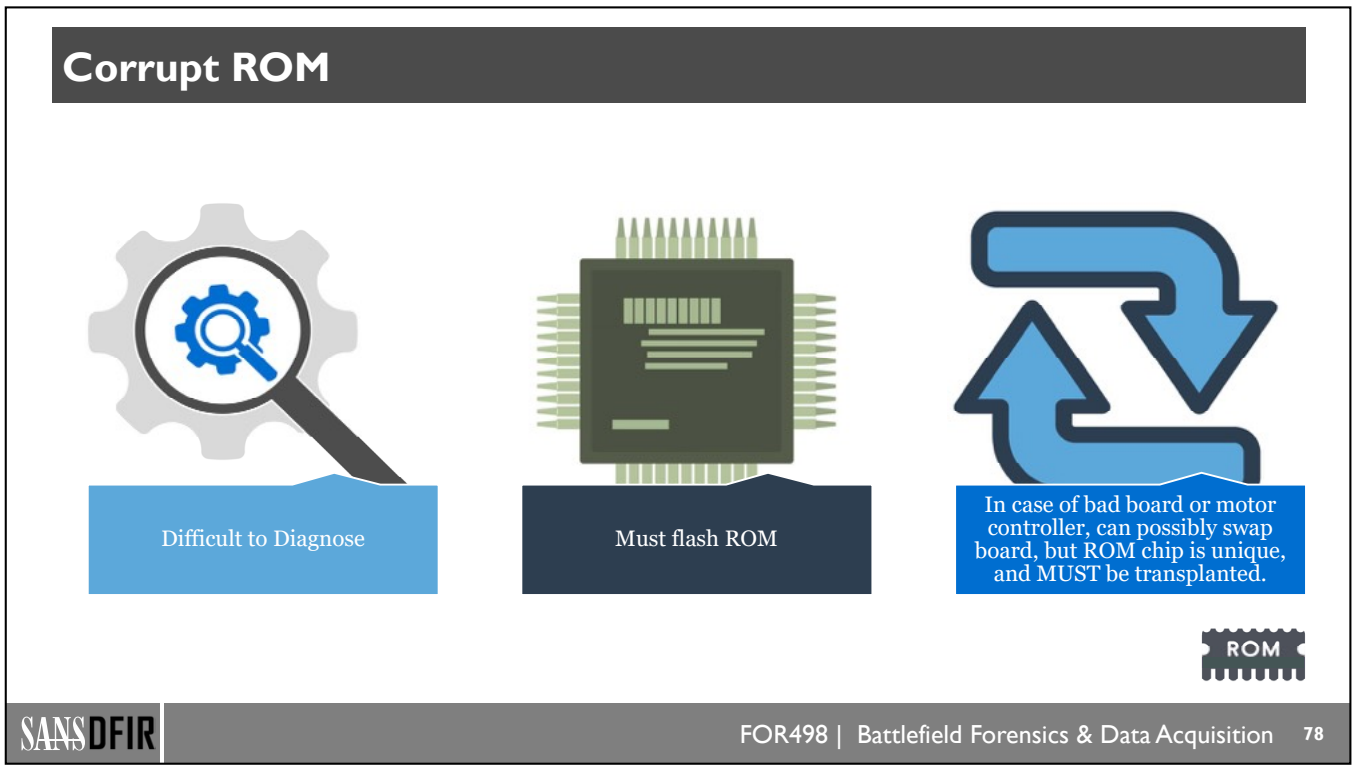
Once you have identified the burnt TVS, simply remove it using a soldering iron, and the drive will work again. Do NOT plug it back into the computer that caused it to fail in the first place! Go to a computer shop and buy a high-quality power supply. Remember that the hard drive now has no over voltage protection, and if there is another power surge, it will almost certainly blow the head stack.

Movie Time!



- 6_4: HDD Blown TVS

This page intentionally left blank.

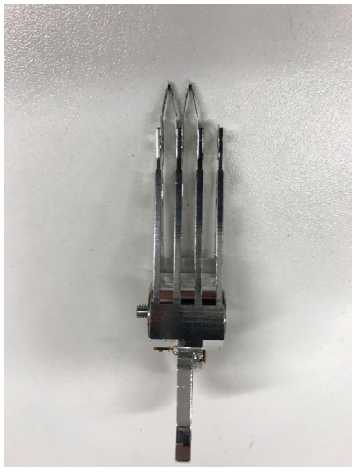


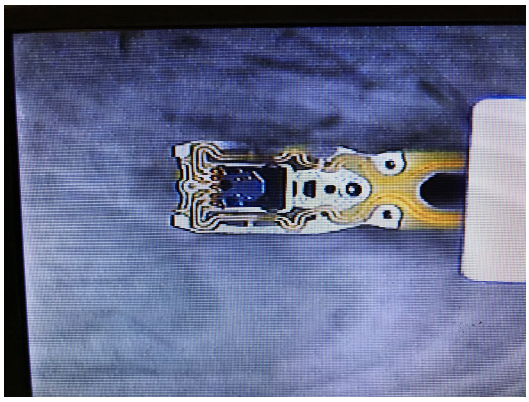
Just like Windows, the drive OS (modules) can get damaged or corrupted. Because they control the drive, it would be catastrophic to lose them, so most drives even have 2 separate copies written under 2 different heads, in case one gets corrupted. A copy may also live in the ROM, depending on the make and model of the hard drive. If you don't repair the modules causing the problem, you can't get at your data.

ROM data is another data set that can become corrupted. Depending on age, make, and, model of drive, if this goes bad, you may be completely out of luck. In newer hard drives, there are file sets called adaptives written to the ROM based on information on the platters at the time of manufacture. If these are lost or corrupted, there is no other set of that data in the world. It is unique to the hard drive that it was built with. In other less frequent cases, you can rewrite the ROM data from a donor drive.

A third area that can become corrupted is the motor controller. If it goes bad, it will need to be replaced. These controller chips have typically a 40-pin connection to the PCB, meaning that a soldering iron will be useless. Sometimes you can "hot swap" a PCB (well, not you, but a data recovery lab), or transplant a ROM chip to a donor board. The key is that this is very brand and version specific! You can't just swap a PCB, as is widely believed. Western Digital hard drives tend to almost always require ROM chip transplanting due to how they program their ROMs.

Damaged Head Stack





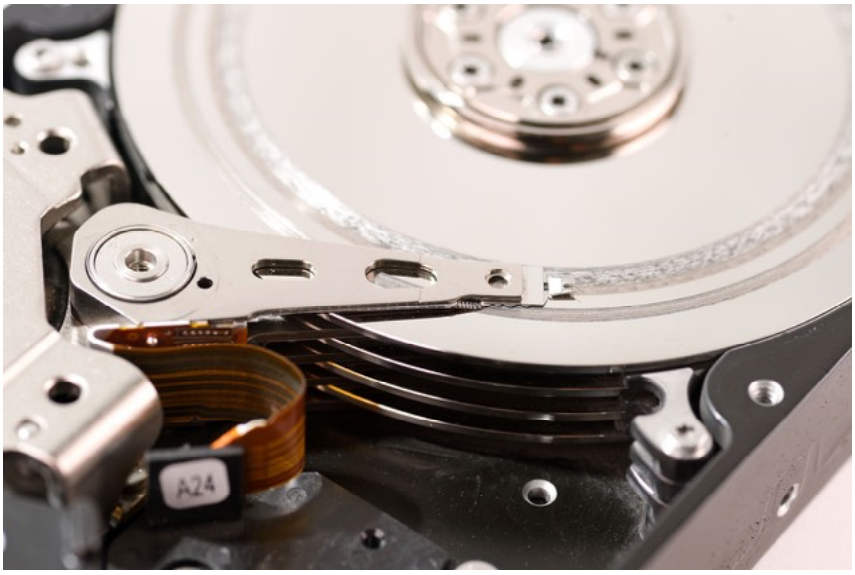
If a head stack fails, it will no longer read or write data. In this case, the head stack will have to be changed before the data can be recovered.

In limited circumstances, it is possible that one head on a head stack fails but other heads on the stack will still operate. In a professional data recovery lab, the hardware and software that is utilized has the ability to isolate heads. In other words a technician can determine which specific head has failed and can turn that head off. This will allow the drive to communicate with the rest of the heads and recover the data that can be read by the rest of the heads. Obviously this means that the data that is under the failed head will not be recovered. It is possible that the data that would normally be read by the failed head is not data that you would need and maybe the data recovered by the rest of the heads is sufficient for the purpose. In any event a technician will always use any working heads that are available to recover whatever data is available under those heads, before attempting a head stack swap.

Seized Motor

With hard drive motors, bearings will sometimes seize up, stopping the motor from turning. Fortunately this is a very uncommon issue and can usually be fixed using specialized tools and HEAT, not cold. Statements on the Internet say that if you drip liquid into the screw hole at the top of a drive, it will get it running again. In actual fact, all that will happen is that liquid will get splattered all over the platters when it spins up. Considering the distance between today's heads and the platter surface area is only marginally thicker than a strand of DNA, this will not work. Don't squirt anything into the drive. There is nothing to be gained and everything to be lost.

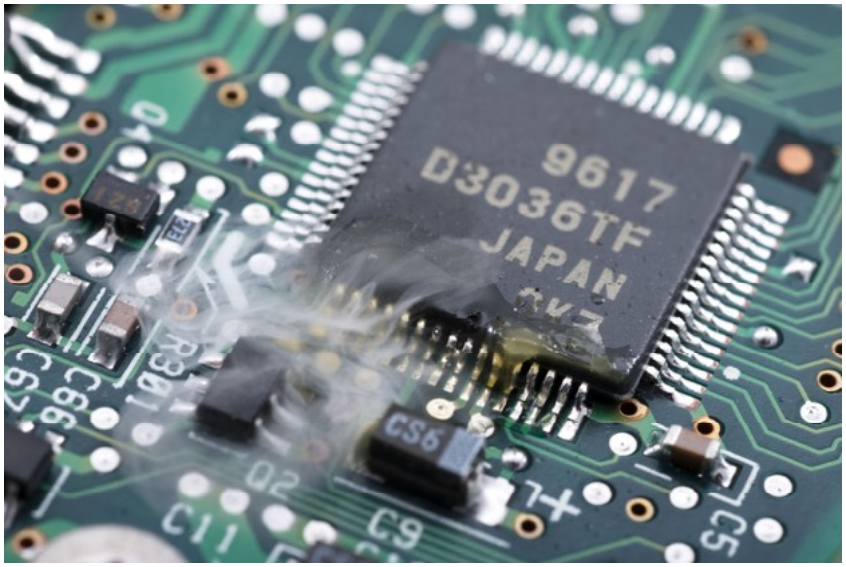
Degraded Platter Surface



If the platter surface degrades or gets dirty, this is the worst of all situations, because this is where the data lives, and so nothing can be replaced to fix it. If the sector can no longer be read, that is the end of the exercise. Having said that, a data recovery lab reads the platter surface much differently than a Windows operating system. An operating system needs the checksum from the ECC sectors to match before it will return data. A data recovery lab can read the sector without first seeing the ECC sectors. There is more detail coming up in this module.

The first symptom of platter surface degradation will be when you navigate to a folder and try to open a file, or copy files and/or folders to new media. The copying or opening will fail after a long time out, with no indication as to why this has happened. Many people get confused with this because they say that they can see their data, but they just can't move it. We know this is not true though because we have learned that the MFT is actually what you're seeing when you see the icon for a file, but the data itself actually lives elsewhere on the hard drive. In this situation, the part of the hard drive that holds the MFT is perfectly fine but the part of the hard drive where the data lives is not fine.

SSD Issues

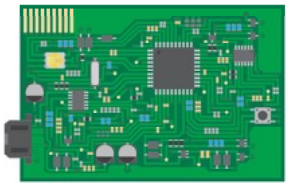


With SSDs, there are no moving parts, but the controllers can still get corrupted. When this happens, there is basically nothing the end user can do about it. In the early days of solid-state storage, the technician would remove all memory chips, dump the data from them with a special reader, then use algorithms to rebuild the interleaving among the data on the different chips. Today, data recovery equipment allows the technician to emulate controllers on the drive without removing the chips (but not in all cases). This would be considered an 'easy' SSD problem.

The worst problem is data leakage. Some labs call it data evaporation, and there are other names, like bit rot. The premise is that if the drive is not powered for long periods of time, the electrons in the cells (the zeros and ones) will reset themselves, (or drift back to neutral). Now the data is gone completely, and there is nothing to recover. It is very much like shaking an etch-a-sketch. In some cases, a data recovery lab can recover partial data.

In both cases, spinning or SSD, a lab can tell you (in the case of partial recoveries), what files they can recover and which ones they can't, before you have to commit to the recovery. It would be a bad situation indeed, if you paid $3000.00 for a partial recovery, and the only part you received was the Windows folder!

Things to NOT Do



Swap the PCB

Put fluid on the spindle

Remove the platters

Put the drive in a freezer

To reiterate, because it is that important, we will briefly review what not to do.

Swapping a PCB - It is often heard that people talk about just swapping out the PCB card with one from another drive of the same make and model. This used to be possible 15 or more years ago. With the drives of today, this simply will not work in 99% of the cases. If you do not first transplant the ROM chip or flash over ROM data for example, you can swap every card on eBay with ZERO success. Another issue is power. You could take a PCB from a seemingly identical donor drive and try to swap it onto a bad drive. Because of a revision in the code of the PCB, the voltage to a head, or some other component has been increased by just a small amount. When the card is swapped, the new donor card may destroy the head stack when powered up. In this instance, there is no warning, and you may never know the damage you did.

Fluid on spindle – There is simply no basis in truth to this myth. The fluid and bearing area of the motor are sealed at the factory. Without a drill or Dremel tool, you cannot access the area of the motor. Removing the screw from the top of the drive and presenting oil into it will usually just spew oil all over the platter surfaces and destroy them at the time of spin up. In any event, very few hard drives today have screws or holes into the spindle from the top.

Remove Platters – An often stated, but not tenable, idea. There are very few, and very specific incidents where you would remove platters. In fact, in the data recovery world, this is the last of all possible resorts, for many reasons. The biggest reason is that in some cases, the data is written to the platter surfaces in a cylinder, and not linearly, as many would think. If you loosen the platters and rotate one against another even a hair's width, your data is lost forever. The proper approach is to change everything around the platters. The only need to remove platters is to change a motor or because the housing is damaged, and then there are very specialized tools that get used for the process.

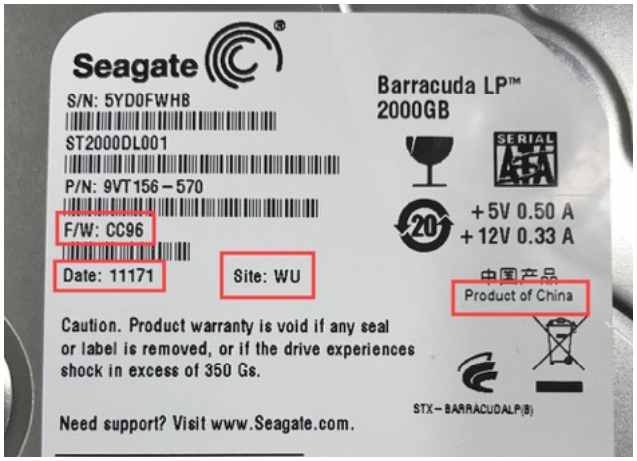
Freezer Trick – This idea is not without merit, but you need to understand where it came from, why, and how it worked. Back many years ago, (think back to very small drives of 10 GB and under), there was a common problem with a certain brand of drives, whereby when they heated up from use, the PCB would expand, and paradoxically STOP making contact on a certain bus (the little wire paths on a circuit board). Freezing the drive kept it cold enough for long enough to pull data before the components heated up again and expanded.

Another situation had to do with flawed material used in the head stack armature that holds the read/write heads. The flaw caused the heads to ride too close to the platters when the assembly got hot and freezing caused the arm to flex just enough to create the necessary space, or fly height, again to read data. These were the only two documented issues solvable by freezing a drive. Also, the conditions of the freezing had to be properly performed. The drive was placed in a Ziploc bag with as little air as possible, then put into another Ziploc bag and sealed. The drive was frozen for about 4 hours. Once the drive was removed, you would push the drive connections onto the drive without opening the bags. This stopped the condensation inside the bag for as long as possible. Once plugged in, you would keep everything wrapped with a freezer pack.

Comments online suggest the bag is unnecessary because the freezer is very dry air. If you believe this, you have clearly never been in a Canadian winter with glasses on! Outside is so dry we have to humidify our houses. Drier than the inside of a freezer. Go outside and shovel your driveway at -20 degrees for 15 minutes, and then walk into the house. Your glasses fog up instantly. Why? Condensation. Put a hard drive in the freezer (one you don't want), for even a couple of hours. Take it out and put it on your counter, and watch it fog over like a mirror after a hot shower. To think otherwise is to not understand the phase state of matter. Condensation inside the drive will destroy it. When the condensation dries, it can leave a mineral deposit. That deposit will destroy the heads and platter surface.

Placing a hard drive into a freezer as a method of data recovery will not fix any of the more common reasons for drive failure. How does freezing the drive fix corrupted sectors? Or bad platter surface? Or blown heads? Or corrupted translator? Or seized motor? Or voltage fluctuation? It won't.

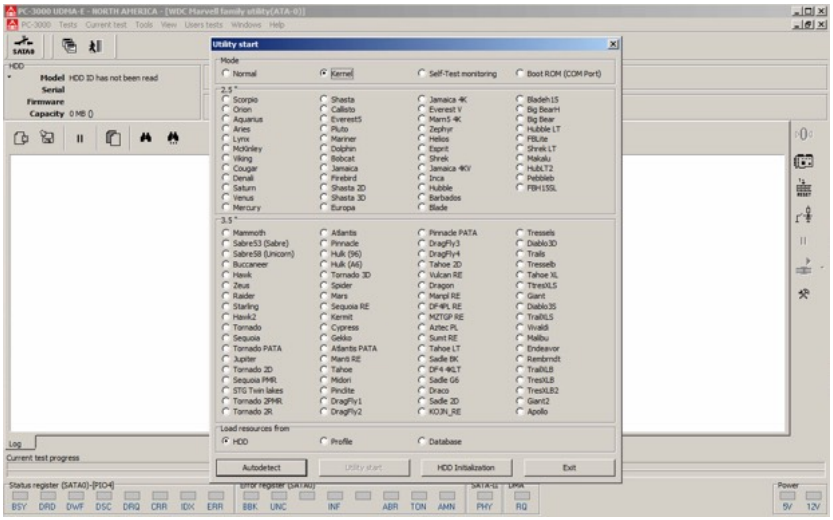
Sourcing Hard Drives



Sourcing hard drives is no small task. Most users believe that a hard drive is a hard drive. They believe that a Seagate 2 TB hard drive is like any other Seagate 2 TB hard drive that appears to have a matching label. Although this is technically correct in terms of the size of data that the hard drive will hold, this has no bearing on the compatibility of parts from one drive to another. Two seemingly identical hard drives with the same name and size will not have compatible parts inside for a reason as simple as where the hard drive was manufactured.

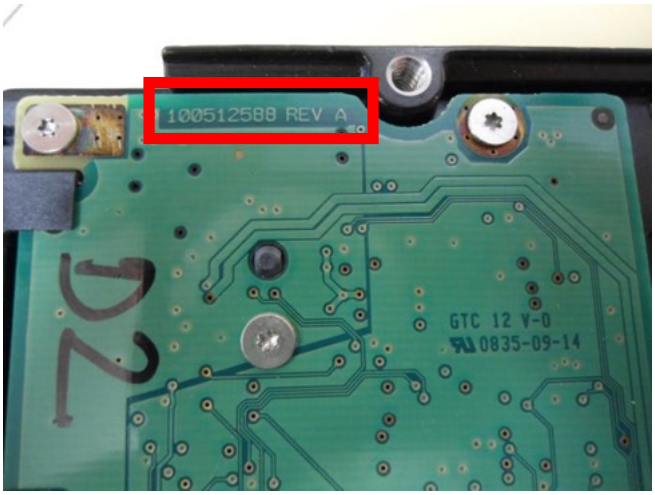
For example, a hard drive manufactured in Thailand with the same specs as a hard drive manufactured in Taiwan will not have compatible parts even though for all intents and purposes everything should be identical. As well, especially with Western Digital hard drives, there are dozens of families of hard drives, and compatibility of parts very much comes down to what the family of the drive is. In the examples in the slide, we have a Seagate hard drive and a Western Digital hard drive. The Seagate hard drive shows a firmware code highlighted as well as a date code and a site code and a product code. In order for things like a head stack to be compatible, all of this information must match, not to mention the model number. When looking at the Western Digital hard drive in the example, for the second part of the model number, it states 00MMMB0. The MM in the middle is the family name and there are reference charts online as well as in the PC-3000 software to make these compatibility comparisons. For Western Digital hard drives the date and DCM code are quite important as well in terms of compatibility, as is the previously mentioned product code.

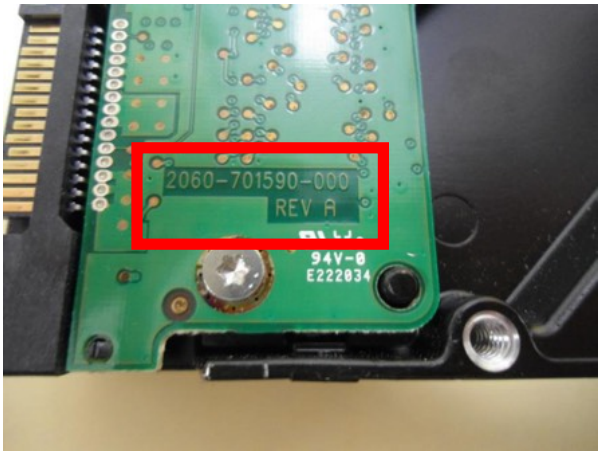
Western Digital Example



This is a sampling of the number of families of Western Digital hard drives. A hard drive may look the same on the outside, but that is where the similarity ends.

PCB Compatibility



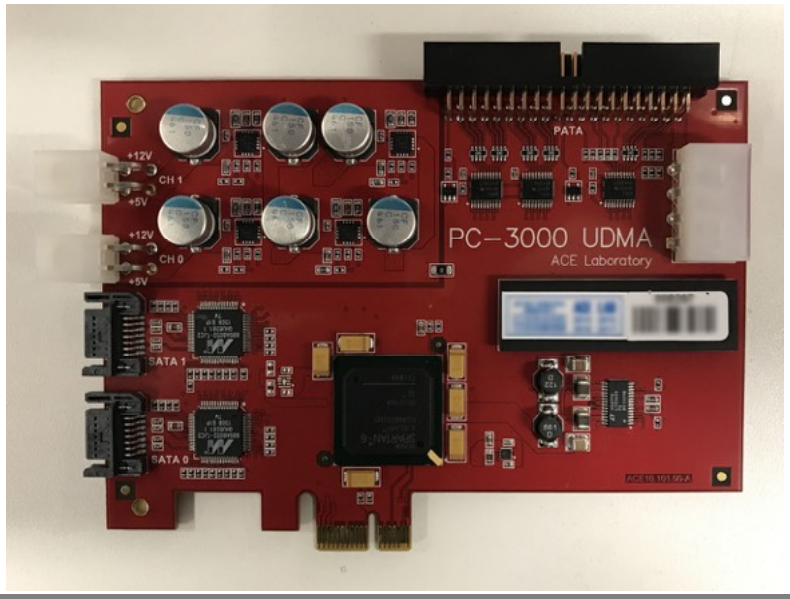


When looking for a donor PCB, the revision code is what needs to match. Even then, as stated before, you will need to transplant the ROM chip, or somehow read the data off of the ROM to flash it to the donor ROM.

Hard Drive Inventory

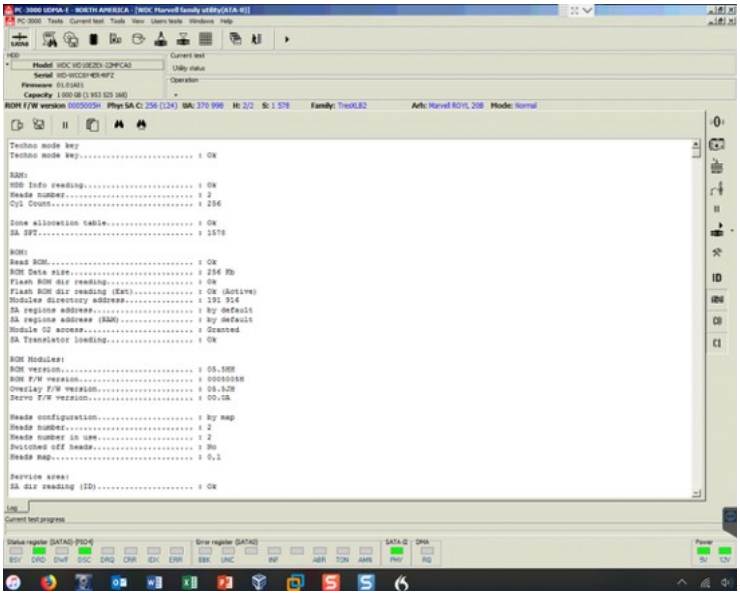
A small sample of the author's hard drive parts inventory.

PC-3000



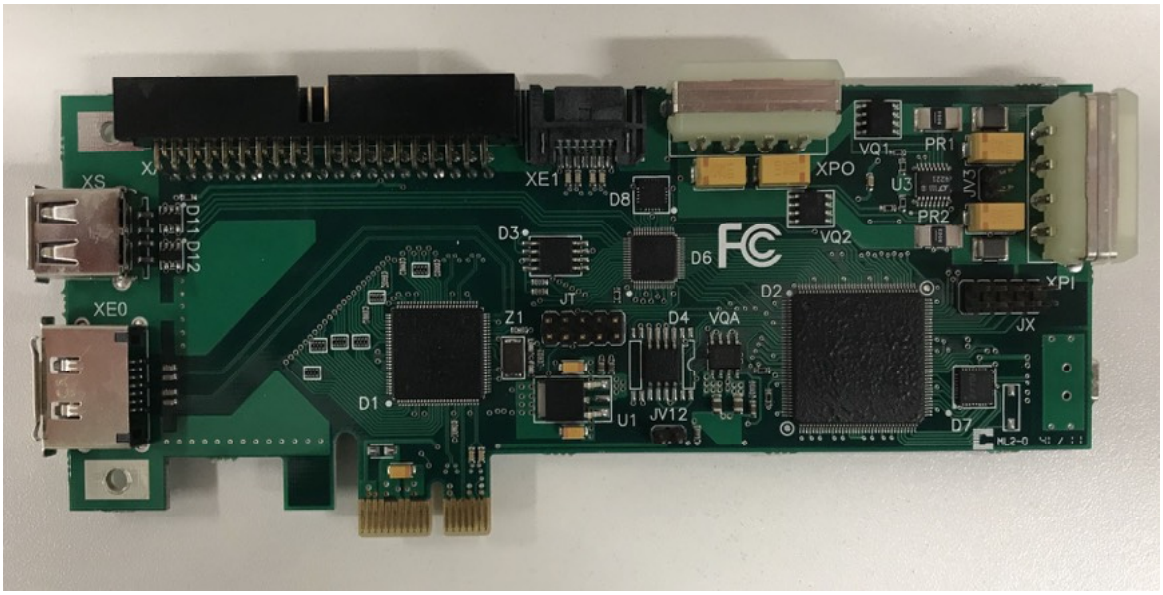
PC-3000 is both a piece of hardware and a piece of software. The hardware portion pictured in the slide is a PCIe card. It is supplied with power from a Molex connector and it has separate outputs to SATA drives as well as IDE drives. As can be seen in the upper left corner there are also two power ports. This allows the technician to plug in a hard drive to PC-3000 for diagnosis and have the ability to control power to the hard drive. This allows the technician to start and stop the hard drive at will and as necessary, as well as in ways that the computer would not allow. PC-3000 is primarily a diagnosis tool. Beyond that it also has the capability of manipulating modules for the hard drive operating system, as well as turning heads on and off for testing capability. This tool will also allow for the extraction of ROM data as well as all modules on the drive. In cases where some modules may be corrupted, PC-3000 can take an aggregate extraction of both copies of the modules and create one good copy for rewriting back to the hard drive. Volumes have been written about PC-3000 and its capabilities, and we can only scratch the surface here. Suffice to say this is the last word in professional data recovery and it has a price to match.

PC-3000 Output



This is an example of one of the simpler screens within PC-3000.

Deepspar Disk Imager



Deepspar Disk Imager (DDI) [1] is a tool manufactured by Deepspar Data Recovery Systems. Its function is purely and simply data recovery. It does not fix hardware or any software functionality. Its singular purpose is to attempt to access the sectors of the drive in various different ways, so as to extract data.

All of the parts of a hard drive have work to do. Platter surfaces are expected to magnetize and demagnetize on each particular bit. Heads need to receive the correct voltages in order to read and write these bits. The data bits that are before and after the actual addressable sector data such as gap, sync, address marker and error correction code (ECC), all have to work properly to even present the correct data. The G-list and P-list need to be read correctly in order to map the drive properly. But what if these things don't happen as they should? If there is any actual re-coding that needs to be done, that is the job of PC-3000. But once it is as operational as PC-3000 can make it, the rest is up to DDI, and the specialist's ability to craft the correct parameters to ensure the highest degree of success.

When Windows, OS X, Linux, or any given OS is reading data, it does so with predetermined instruction based both on internal code, as well as interfacing that is done at the ATA level of the drive. This will usually mean that, for example, if there is a degraded area of a platter that the OS can't read, it will try over and over again to read the area. It will also potentially increase the read time of the heads. While the OS is trying to read this area over and over, nothing else is allowed to happen, and a user sees their computer freeze up. Potentially, the OS will continue to do this to the point of failure.

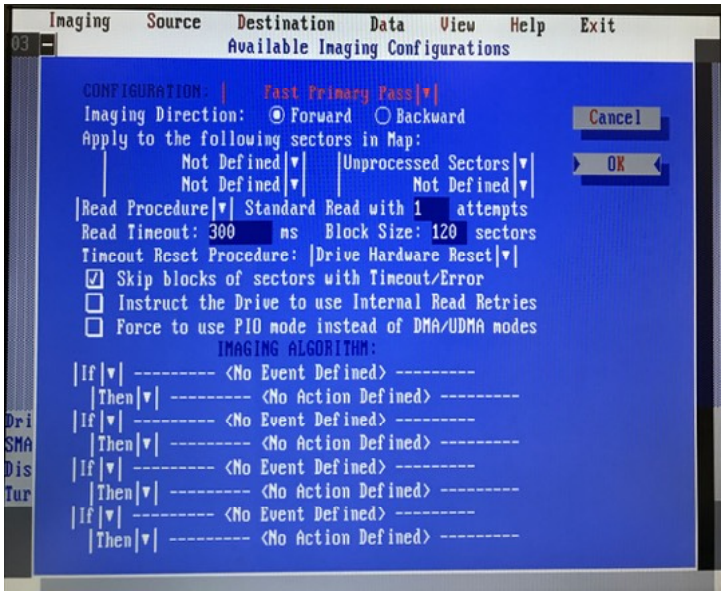
DDI can detect such bad surface areas and treat them differently. For example, the analyst can tell DDI how long to try to read a sector, and if it is taking longer than a certain period of time, skip the sector and go to the next one. Once all the easily read sectors have been read off to new media, we can go back and apply more time on the sector. Or in a case where the reason for the poor read is because ECC is not correlating properly, the analyst can tell DDI to stop looking at ECC and just pull raw data from the sector.

A very common problem with hard drives is the failure of the caching. Caching on a hard drive has to do with something called “Read Ahead” [2]. As the head is flying over sectors, it will grab any data around where it is working and push it into cache. This allows the hard drive to respond much faster, rather than waiting for the disk to rotate again and place the data under the head for reading. When this caching fails, the drive will no longer function. DDI has the capability of reading a hard drive backwards, which allows for reading the drive without having caching enabled. The speed difference is extremely significant. It is not uncommon for a drive to read ahead at rates of 80 000 KB/s. Reading backwards without caching is done at approximately 2000 KB/s. At this speed, it takes 11.5 DAYS to recover data from an otherwise healthy 2 TB hard drive.

[1] DDI | `https://for498.com/h1zdf`

[2] Read Ahead Caching | `https://for498.com/h1zdf`

Fast Primary Pass

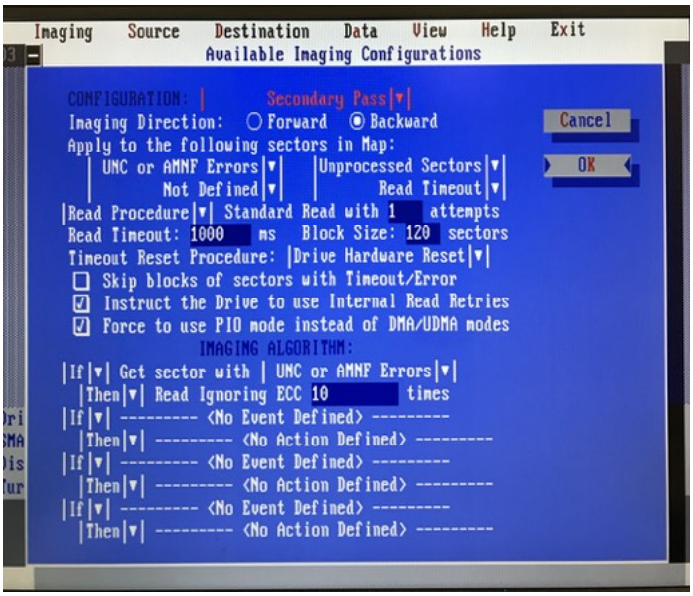


Once the drive has been repaired either mechanically or using PC-3000, it is ready for data extraction. Understand that when a professional data recovery lab refers to data extraction, it does not mean that the drive has been fully repaired and is ready for normal operation. If a hard drive has been submitted to a data recovery lab, it is highly recommended that this drive never be put into production again. In many cases the drive won't operate in a normal computer, however with hardware such as DDI, the hard drive can be accessed in various different ways to extract the data even if the drive is not normally operational.

When the drive is connected to DDI one of the first operations that needs to be performed is determining what kind of passes the technician is going to use to recover the data. There are a number of methods at the technician's disposal, however three are commonly used. The first is a fast primary pass, the next is a secondary past, and finally the data retrieval pass. We will look at all three briefly.

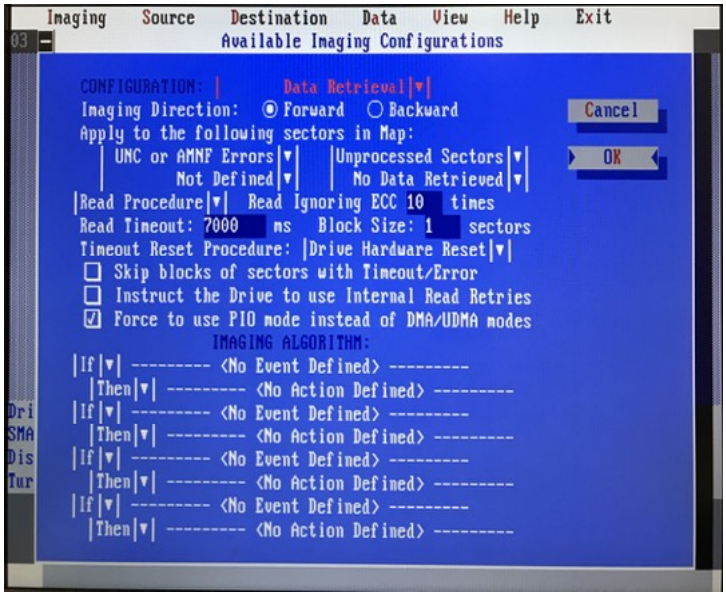
With the fast primary pass mode, this reads the data in a forward direction with a relatively low read timeout, meaning it does not spend more than 3/10 of a second trying to read a sector before it moves on. It will also read the sectors in blocks, in this case,120 sectors per block. The technician can change both of these parameters, and the higher the block size the faster the read operation will occur. In the case of a read timeout or a sector that cannot be read within 3/10 of a second the technician can define what will happen to the sector. In the fast primary past, it just merely marks that entire block of sectors as read timeout and it continues on. This pass is designed for a very rapid recovery of any data on the hard drive that doesn't need extraordinary measures to recover. As the drive recovers data from good sectors it creates a map. Using this map the technician can go back to work on sectors that hadn't been recovered yet without having to worry about the data that has already been recovered.

Secondary Pass



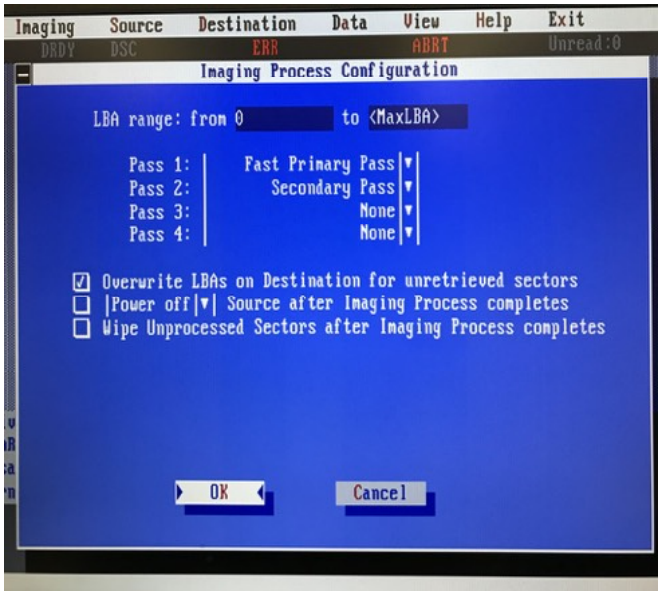
The second pass is called a secondary pass. This secondary pass is designed to read the hard drive backwards, although it does not need to. Again there are a great deal of options the technician can select, however the secondary pass's claim to fame is to be able to read a hard drive backwards and recover data in situations where things like the buffer or the read ahead will not allow the drive to operate properly when it's being read in a forward state. The read timeout for secondary pass will usually be higher than a primary pass, and in the case of the slide, is set to read a sector for up to one second before skipping and moving to the next sector. The secondary pass is also set up so that if there is an error reading a sector and the error is of a certain type, such as an address marker not found, it will read the data in the sector ignoring the ECC 10 times. Yes, it reads the sector 10 different times. Because ECC is not being used, it has no way of verifying whether the data is correct or not, so if it reads the data 10 times, it takes an average of the 10 times and wherever the same data showed up the most frequently is what it will use to be true.

Data Retrieval Pass



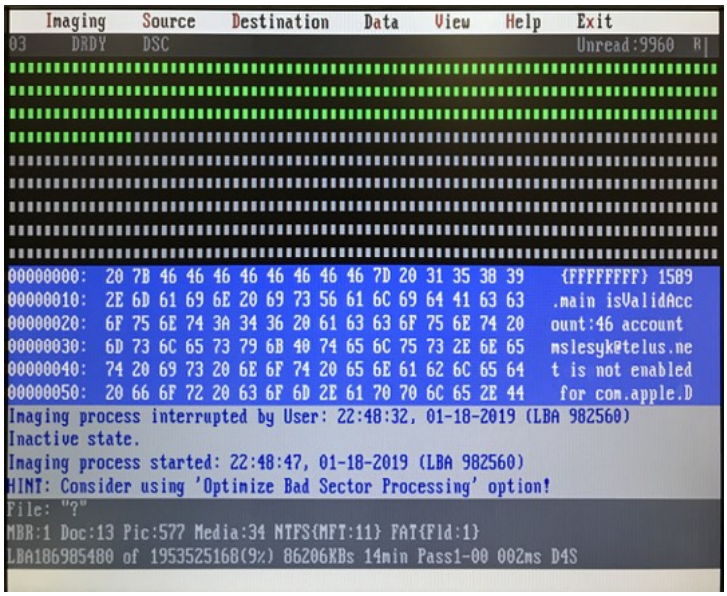
Once the secondary pass is complete, the drive will start on the data retrieval pass. This data retrieval pass is a forward pass and moves quite a bit slower than any of the other passes. As noted in the slide, the read timeout is actually seven seconds per sector. In each of the three passes, the work that the hard drive has to do becomes more and more taxing on the resources. In the first pass, the technician grabbed what we would consider the low hanging fruit, or the 'easy to get' data. Then each pass worked successively harder and harder to retrieve data from unrecovered sectors. This methodology will allow for maximum amount of data recovery before a drive potentially stops working for good.

Image Process



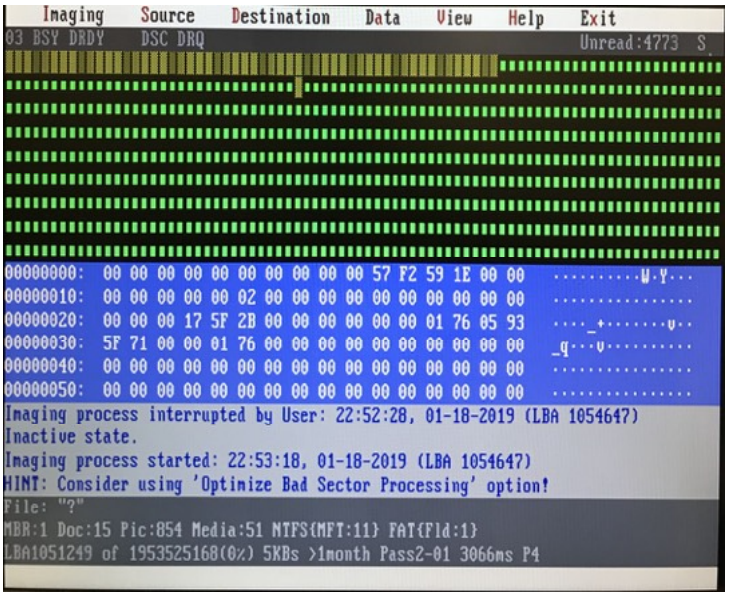
DDI also allows us to extract data just from certain ranges on the hard drive. For example we can tell DDI what sector range to recover if we know exactly where our data is. The technician can decide which passes to run. In more advanced screens within DDI, the technician can even scan the drive and write a database for what data lives under which heads, and extract data per head. The technician can also outline the data on the drive based on MFT, and target just resident data, or even more specifically, a particular type of file.

Reading Data (1)



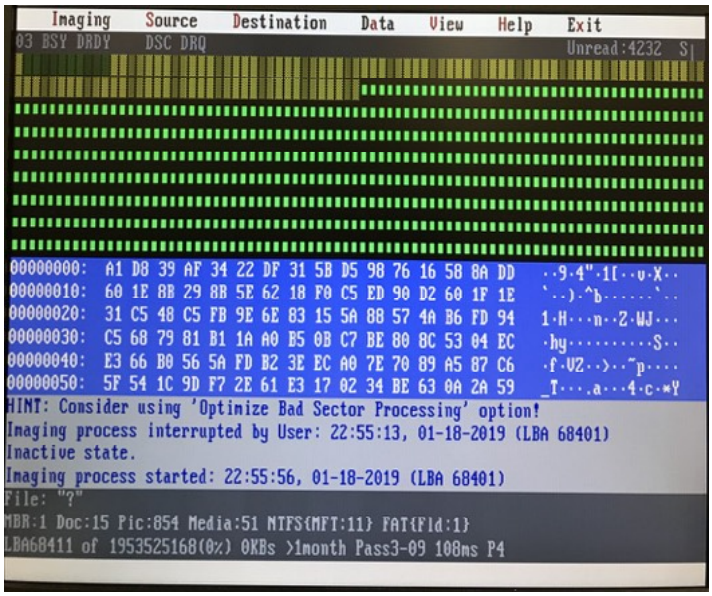
In the screen above we can see grey blocks and green blocks. Anything marked as a green block is data that has been read and committed to a destination hard drive. This data does not need to be revisited for any reason. In the middle of the screen colored in blue is the hex representation of the data in the particular sector currently being read. The information towards the bottom of the screen shows date and time as well as a guideline for what has been recovered to that point. It will also show what sector is being read out of how many sectors total on the drive, and the read speed.

Reading Data (2)



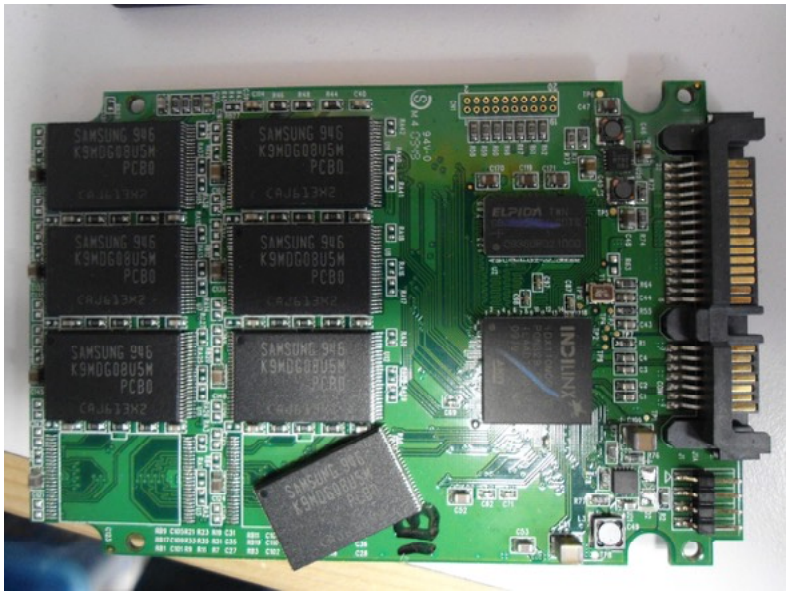
When DDI is reading a drive and it gets to a read timeout, it will mark those sectors in yellow boxes. This means that they have had a read timeout and have not been read yet. On the secondary pass, which is reading backwards, DDI will try to read those sectors using more advanced parameters. In the slide above we can see that some of the yellow sectors have been read up to this point, however there was one sector in the second row that could not be read even in the secondary pass. The idea would be that in the data retrieval pass, this sector would finally be read and committed to the destination hard drive.

Reading Data (3)



This slide shows an example of the data retrieval mode. This is the third pass, and most taxing pass on the components of the hard drive. The dark green boxes indicate data that has been recovered in a data retrieval mode, and if it is green it has been successfully recovered.

Tooling Examples



As indicated earlier in the module, data recovery on certain solid-state drives might involve the removal of the memory chips. In this slide we see one chip having been removed and a second ship desoldered but still sitting on the board

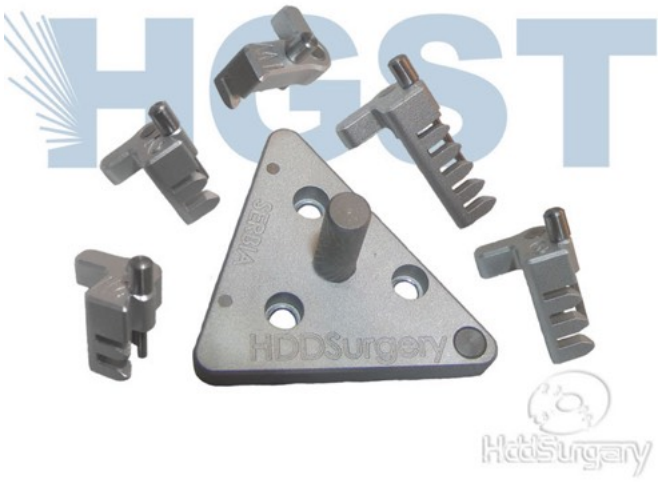
PC-3000 Flash

This device is one of a number of what is called TSOP (Thin Small Outline Package) readers, and the memory chip that was removed in the last slide would now be placed inside this reader and PC-3000 flash software would read all of the data off of this chip into a single raw file. As each memory chip is planted in this device and another raw image created, the technician will finally be left with the same number of files as there are memory chips on the hard drive. Other software is then run to try to rebuild the interleaving and extract the data from the chip.

Head Comb (1)

This is an example of a head comb, or head removal tool. This particular model is from HDDSurgery, and can retail for approximately 400 to 600 Euros. This price is for one comb. In order to do a head swap you will need 2 of them.

Head Comb (2)



This slide shows an assortment of head combs specifically for Hitachi hard drives.

Spindle Motor Tool

This tool is a Seagate spindle motor replacement tool, and retails for approximately 2500 Euros.

Summary

- Hard drives are far more unique than people believe
- Data recovery is expensive and complicated
- No matter how simple a problem seems, it usually is not
- There are tools for dealing with issues; for a price
- Many times you only have one chance at recovery

This page intentionally left blank.

Exercise 6.2

Data Recovery

Synopsis: In this exercise, you will use R-Studio data recovery software to recover files from a damaged volume and overwritten MFT.

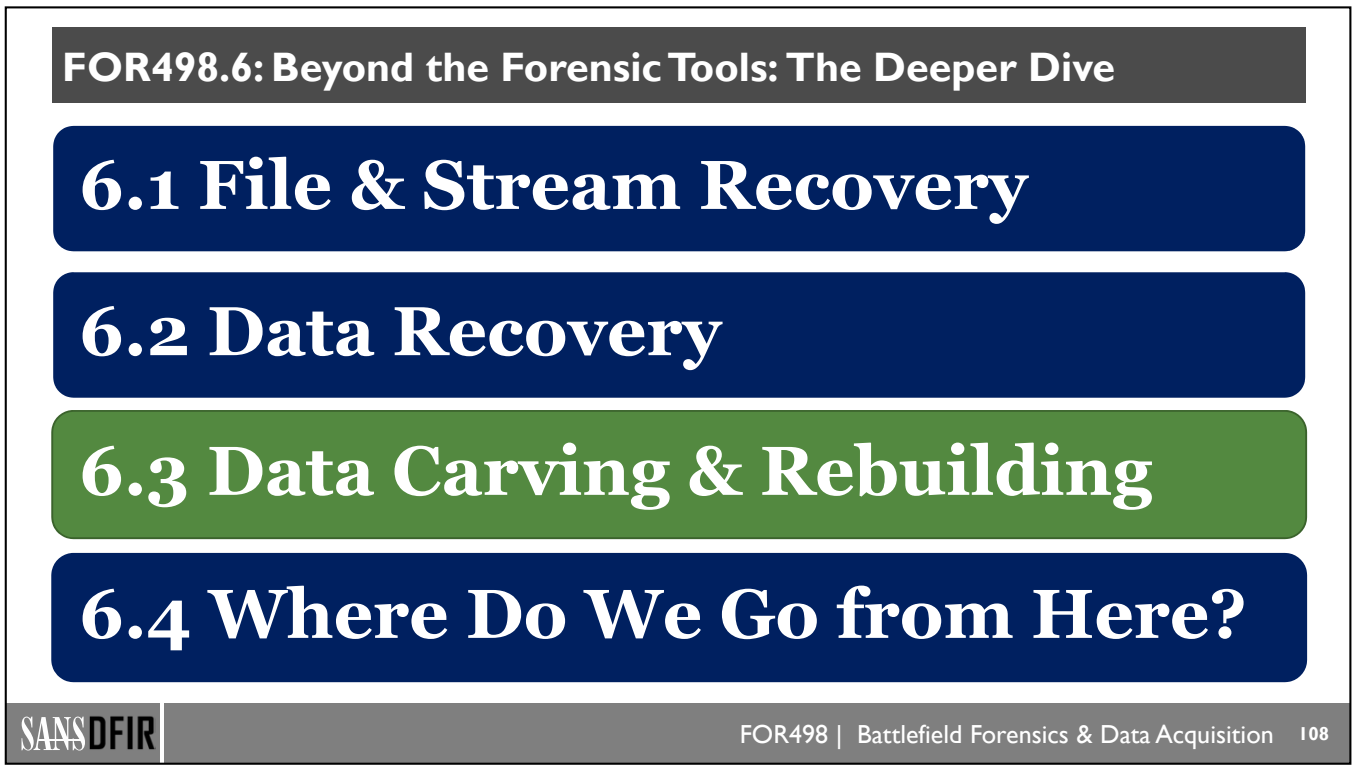
Average Time: 20 Minutes

This page intentionally left blank.

Exercise 6.2 Takeaway

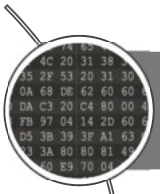
- Data recovery can find data that was otherwise thought to be lost forever.
- Not all data recovery programs are equal in their recovery capabilities.

This page intentionally left blank.



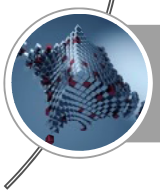
This page intentionally left blank.

Data Carving & Rebuilding



Data Recognition & Carving

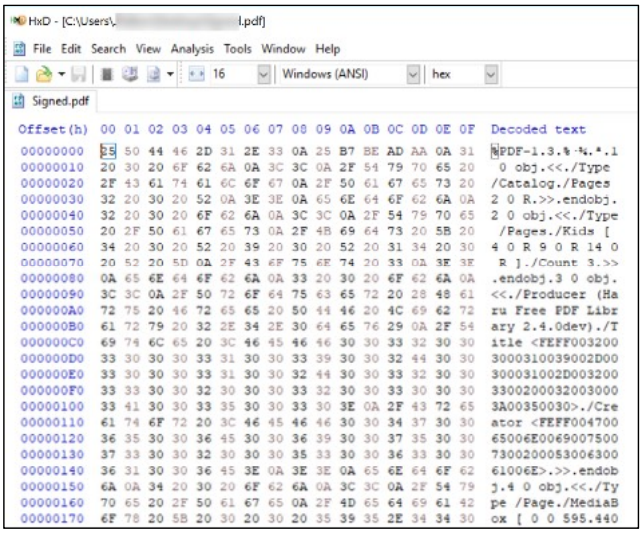
Data Structures & Rebuilding

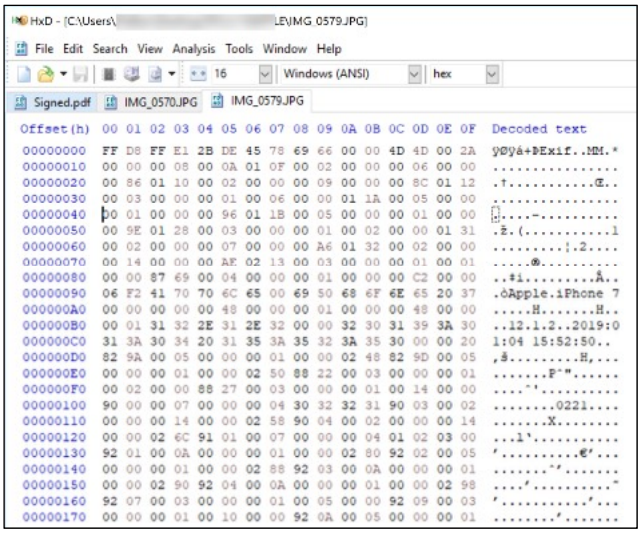


Repairing Various Data Types

This page intentionally left blank.

Hex Editor





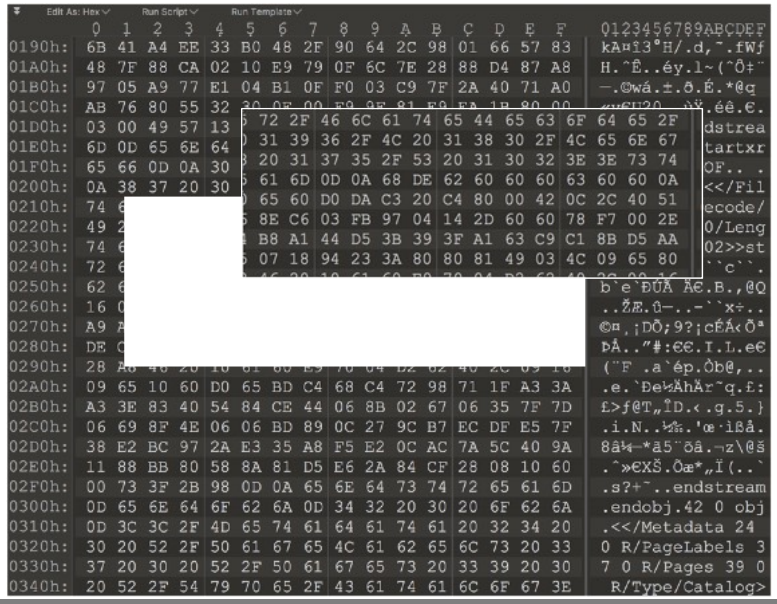
In order to perform manual data extraction, a good, solid hex editor is a must. There are free hex editors, and there are paid hex editors. Paying for a hex editor does not necessarily mean that you are going to get a better product. In this course, we use a product called HxD as our hex editor of choice, and it is free.

There are two significant factors to consider when reviewing a hex editor for use. The first thing you must determine is whether or not it allows access to a physical drive. In the case of many paid products that also have free versions, this is one restriction they will have. They will not allow you to access a physical drive unless you get the paid version. You are restricted to file level access only. As well, many free hex editors do not allow access to the physical drive. The second factor is whether or not the hex editor will allow you to actually edit the contents of the file. Again many paid versions will not allow for editing in their free versions, and many of the free hex editors will also not allow editing at all. It may be best said then, that many hex editors are not editors at all, but simply viewers.

When viewing data in a hex editor, although there may be different features and options that can be used within the hex editor, the layout is generally the same. There are usually three columns to the viewing pane of a hex editor. The first column will be the offset or address column. The second column will be the hexadecimal column and will be 16 bytes wide. The third column will be the decoded text. This does not mean that the data in this column will necessarily be readable. If the file in its native state is not plaintext then it won't be readable, and in many cases such as video or photos, will merely be the closest approximation of an ASCII (American Standard Code for Information Interchange) [1] character that the hex editor can create.

[1] ASCII conversion table | `https://for498.com/kazup`

Why Data Carve?



SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 111

The debate continually rages over whether or not we should be data carving. Although we cannot answer this question for you, simply because it depends on the situation you are faced with, it can be categorically stated that this is an absolute 'must have' skill. If you are not data carving you are missing significant amounts of data. In the case of deleted material, data carving is often the only option. Performing a 'recover folders' operation in EnCase is not the answer.

Essentially, data carving is necessary if data in unallocated file space is no longer referenced by the MFT.

In many cases examiners are using the techniques they believe are available within their forensic suite of choice. The forensic suites themselves, although having data carving capabilities, do not present these capabilities in necessarily a user-friendly manner. It is usually only the advanced examiners that are even aware of the features and how to use them. Data carving is an advanced skill in an industry where more advanced skills are desperately needed.

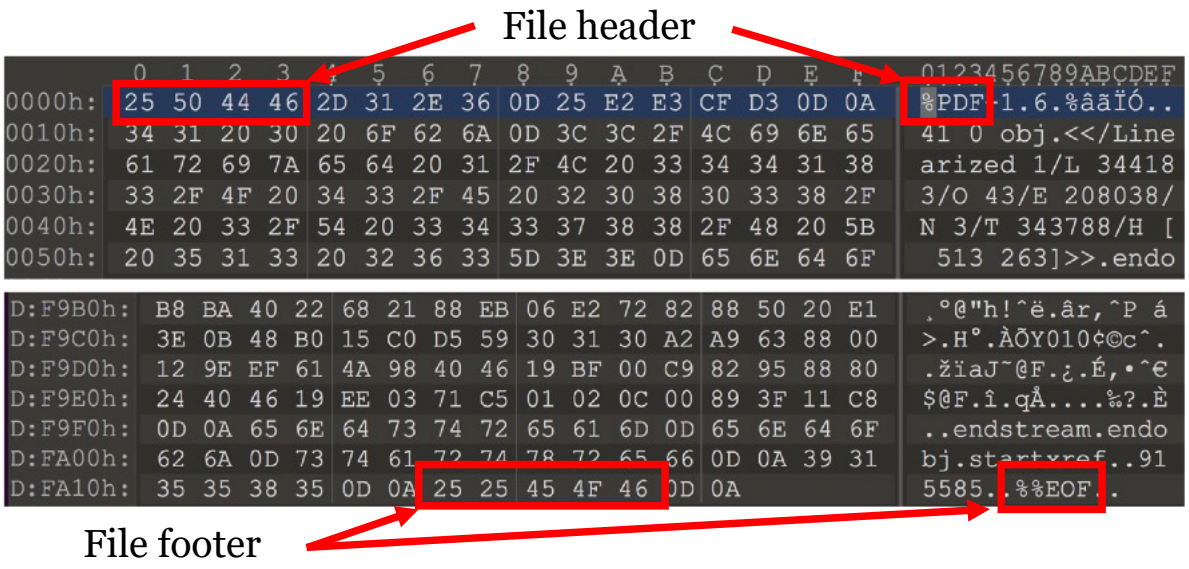
Much of the difficulty in this area comes from the fact that sadly in many investigative areas, if there is not a tool with a 'get me the evidence button', the data is believed to not be available. Especially when it comes to manual data carving, there simply are no tools. It comes down to expertise, patience, tedium, and understanding data at the hexadecimal level.

Even for those that are doing data carving operations, in the majority of cases, programs are doing the carving for you. For efficiency's sake this is not necessarily a bad thing but if you're going to use a program to do your carving for you it is important to understand the limitations of that program. If you do not, you end up with exactly no efficiencies. A worse outcome is in not understanding the data you have just carved. In this module, you will learn about data that has been carved, seems to be corrupt and goes ignored, but actually contains important data.

It is often said that data carving is useless simply because we cannot provide attribution for data that is recovered. While this is sometimes true, and especially applicable to child exploitation cases for law enforcement, nothing could be further from the truth when it comes to many other kinds of cases. For example, if your investigation involves theft of intellectual property or data destruction, being able to recover deleted files becomes incredibly important. These techniques have been used in cases where documents have been altered and the originals have been deleted.

Cases have been won and lost based on information extracted from unallocated file space. It is data carving that does this extraction. The only question that needs to be asked is, “How are you doing the data carving?” Data carving essentially falls into two categories. The first category, and the one that is in use most is a traditional carving methodology. The second broad category is manual carving. We will explore both the next few slides.

Traditional Carving



Traditional data carving is the type of data carving that most examiners have been doing, if they been doing carving at all. This involves the use of some kind of a tool such as EnCase, FTK, or XWays. In this and other forensics classes, we also introduce you to a tool called PhotoRec. PhotoRec is probably one of the most capable data carving tools on the market. A commonly used carving tool in the command line world is a tool called scalpel. This is another very capable tool.

As you can see, there are many tools at your disposal to assist in the data carving scenario. But for many examiners, if these tools cannot carve the data they are looking for, it must not be available. This is an inaccurate and dangerous assumption. If you are an examiner that may be asked to support your findings in court, you had better be absolutely sure of your statements. Nothing will destroy a reputation (at least for a non-law enforcement examiner) quicker than making a categorical statement on a dataset, only to have an examiner on the other side of the case bring up data that you said didn't exist.

To understand what you may be missing with traditional carving methods, you must first understand how traditional carving operations are performed. We discussed previously the idea of file signatures. A file signature includes a file header, or the first few bytes at the beginning of a file that make it unique to that file type, and it also includes a file footer. This file footer is the demarcation of the end of the file. Not every file has a file footer, nor needs a file footer. In certain cases such as .JPG files, HTML files, and .ASF files, even if the end of the file is damaged and the footer no longer exists, the file will work correctly up until the damage point.

Traditional file carving tools use these file signatures to determine what they should be carving. In the example of a Word document, the file carving tool will start looking at every single sector of data on the computer. The examiner can limit this carving to just unallocated file space if they choose. As a file carving tool examines each sector of data, it is looking for the file signature or the first few unique bites of the Word document. When it identifies this file header, it will start collecting, or carving, the data that follows. It will continue carving this data out until it finds the file footer. Once it sees this, it will take this chunk of data that has just been carved and move it to a folder in another location on an examiner hard drive for later analysis. Once done it continues through the data set and does this for every single Word document that it can find.

As should be very apparent by now, the only way that the file carving utility can operate is through the identification of the file header. But what happens in the case where the first sector of data has been damaged, or even just the first few bytes? In a case like this, the Word document may only be missing the first 10 bytes of the file, but because the file signature exists in that first 10 bytes, the carving tool will not identify it and will keep moving on and miss this entire file. Even though only the first 10 bytes are missing, the rest of the file is completely intact, and a skilled examiner can recover that file and absolutely rebuild it in such a way as to extract potentially critical data. That falls under the purvey of manual data carving.

A very significant issue that an examiner is faced with when discussing automated data carving is the issue of the substantial amount of unnecessary, unusable, corrupt, and duplicate files. Anyone who has done this traditional style of carving even once has suffered through the pain of going to the folder that holds the carved files and potentially seeing thousands and thousands of files that were carved. In the case of a recovery of Word documents, what advantage have you gained now that you have a folder with 20 or 30,000 carved Word documents? As is commonly done, the examiner's next task is to start opening these files one by one to find out which ones work, and what their contents are. And what happens in most cases? You double-click to open the file and you get an error message saying that the file is corrupt and unreadable. What is your next step? In most cases, the next step is to close the corrupt file and move on to the next file.

In our previous example of partial overwriting, maybe the Word document that you just saw that was corrupted actually only has the last five bytes that have been overwritten. This is enough for the file to present as corrupted, and yet the file contents in their entirety are contained within the file and absolutely recoverable by the examiner who knows how.

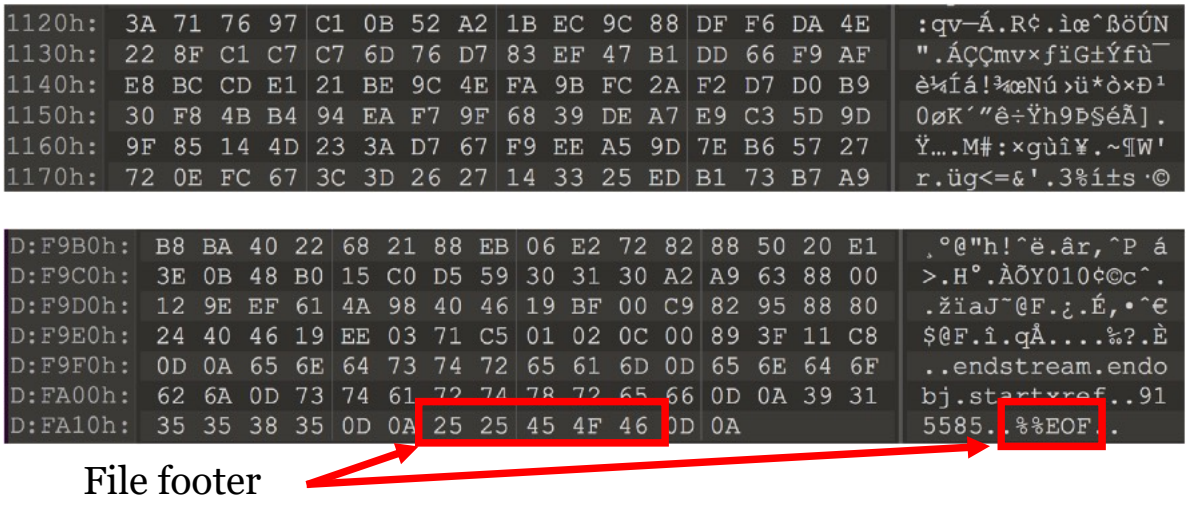
Another great example is that of an .EPP file. This file is of a type particular to professional cameras. Its file signature is proprietary, and so not easily carved. If your tool has not carved the .EPP file, you will miss it entirely. If your tool has actually carved the .EPP file, you will go to the file on your examiner's machine and double-click, and it will not open. It will present is unreadable. And you'll probably close it and move to the next picture. Such are the shortcomings of automated file carving. The reason it did not open is because very few computers have a viewer for viewing .EPP files. In that same case on the manual carving side of things, we would go into that file manually with a hex editing program and extract the actual photograph. We will perform this in an exercise at the end of this module.

It is worth noting that traditional carving, or automated carving absolutely has its place. If you were involved in an investigation with a file that has a proprietary format, you can actually write a carving signature for that file. Then you can analyze the entire hard drive in an automated fashion pulling out every instance of that file. This may be efficient in cases where there are not going to be many of these file types on the computer in the first place. Understand that you are still faced with the same limitations as previously discussed.

We do not want to leave you with the impression that traditional carving methods are useless or unnecessary. We want you to understand how they work, and the possible limitations that might exist. Sadly though, if you are only doing this type of carving, you will never know what you missed, or left behind. It could make the difference between someone getting fired or not. Someone getting charged or not. Someone going to jail or not. The stakes can be incredibly high. No pressure though.

Manual Carving

File header - NONE



In the mid-1990s, an entire forensic investigation was handled through the process of manual carving. There were no "forensic suites" to automate the task for examiners. You opened the data set in a hex editor, and you started examining from there. As data sets got larger and larger, this became untenable. Tools were developed to help look at data sets in a more manageable way, as well as extract specific data based on a given investigation. As time went on, manual data carving fell by the wayside and became almost a lost art. Make no mistake, manual data carving is an art as much as a science.

The most important tool besides a capable hex editing program is your brain. Manual data carving is not something that can be approached in a haphazard manner. It is also not something that can be learned by a '9 to 5' examiner. Manual data carving takes a great deal of practice, not to mention a great deal of research into understanding data structures. You must have the ability to understand what you're looking at when you see it. This is not a skill that you want to learn in the middle of your investigation.

It is expected that an examiner who plans to perform manual data carving can take a great deal of commonly seen files and recognize them for their file format based on their file signature. The only way this is possible is through practice, practice, and more practice. It is not expected that an examiner will be able to recognize every file format by rote, as there are thousands. However when it comes to forensics, the overwhelming majority of cases will involve the recovery of maybe 15 to 20 very common file types. For example Microsoft office files, executable files, PDF files, picture files, graphics files, zip files, Internet pages, and email files. The majority of these files should be identifiable by sight. If you cannot currently perform this task, there is no time like the present to learn.

In fact, manual carving is so rare that one of the largest forensics conferences in the world awarded their "Investigator of the Year" Award to a police officer who solved a very high-profile case by, quite frankly, doing nothing more than manually carving data from a floppy disk. A skill that any examiner should possess.

Once you have your image file mounted in your favorite viewer, you should be able to save out the unallocated file space to a separate file. This is not necessary, but this would certainly cut down on the amount of files that you might find, by excluding everything that is resident.

If your search is specific enough, this should not be an issue in any event. You would now open your dataset in a hex editor and run a search. If your search is for a specific term that you expect to be found in plain text, then this is the simplest of all scenarios. If you find your search term in the middle of a bunch of data, you can now use the hex editor to start viewing the data before your search term hit. Ideally you are looking for a file header, and if you are lucky enough in your scan to find it, you can start collecting data from that point down past your search term hit, and all the way to (hopefully) the file footer. Once you have highlighted all of this in the hex editor you should be able to right-click and export this data out to a new location as any type of file you would like. Depending on the hex editor, your best option may be to copy this data, open a new file in the hex editor, paste the data and then save it out from there.

If you are not lucky enough to find an intact file, then at least export the data around your search term that you believe to be part of that file. It is not always easy to identify the pieces of that partial file in a case where the beginning or the end may have been overwritten by new data. Experience will allow you to tell though, a separation between the data. For example if you are trying to recover a webpage and you are scanning for the file header and you see a bunch of non-ASCII data, you know that whatever this non-ASCII data is has overwritten your webpage file. This is where becoming familiar with what a file looks like in a hex editor becomes very advantageous.

In the case of a compound file such as a Word document, your task becomes much more difficult. But not impossible. It is more a question of what is at stake and what kind of time you are willing to invest. Although, as we know, we cannot run a word search against Office files of 2007 and later, we do know that we can still find files of that type. Once we understand the structure of an Office file we can now search for the names of the components of that file. Knowing that the text portion of a Word document, for example, is named document.xml, we can use this as our search term. Our search term can include something such as word/document.xml, as this is the path within the compound file. This appears exactly 2 times in a Word document. The first time is the beginning of the actual data. The second time is merely metadata. When we see it in both these scenarios, it is very obvious which is which. If we do a search across our unallocated file space for word/document.xml we can then begin to locate these documents. Once you find a particular search hit based on this parameter, you can start looking at the data in the hex editor prior to your search hit and get a sense of how much of the document still exists.

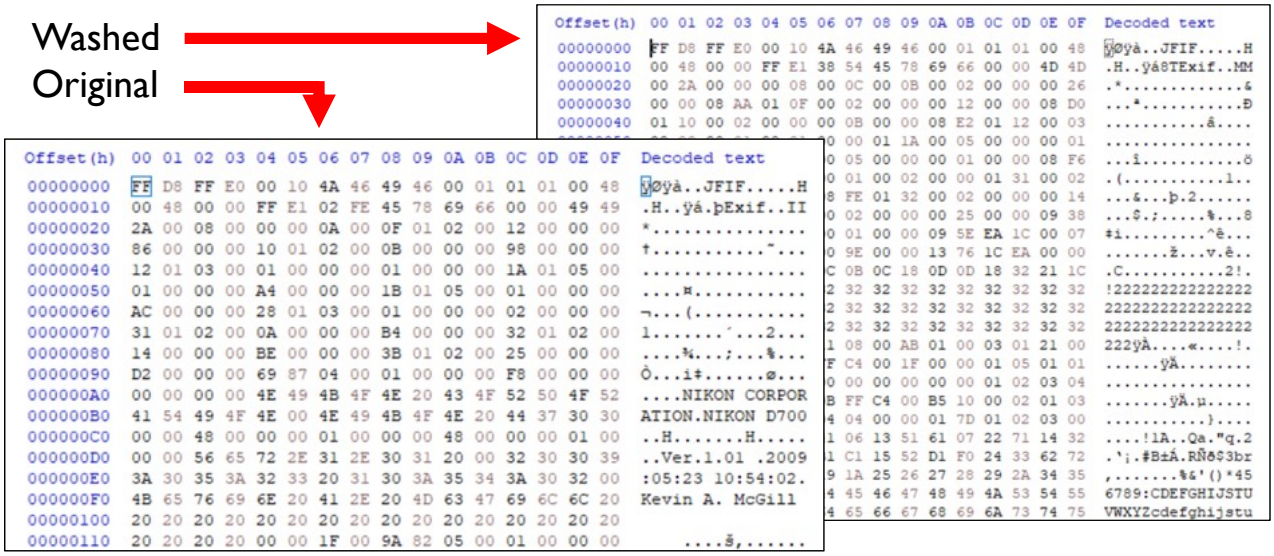
Even if all you are able to find is the document.xml title, you can start carving from that point downwards to whatever is the end of the file. Once you save this out, if converting it to a zip file still does not get the information you are seeking, there are various tools that will help you repair the zip file and potentially recover the data.

This type of carving is not for the faint of heart. If the search term hit happens to be in something like a file path in a PDF document, and so you start carving the data looking for the file header, you could be engaged for a very long time. In the case of something as small as even 4 or 5 MB in a PDF file, you could be performing a page up command in the hex editor for literally hours.

Data Structure

Washed

Original

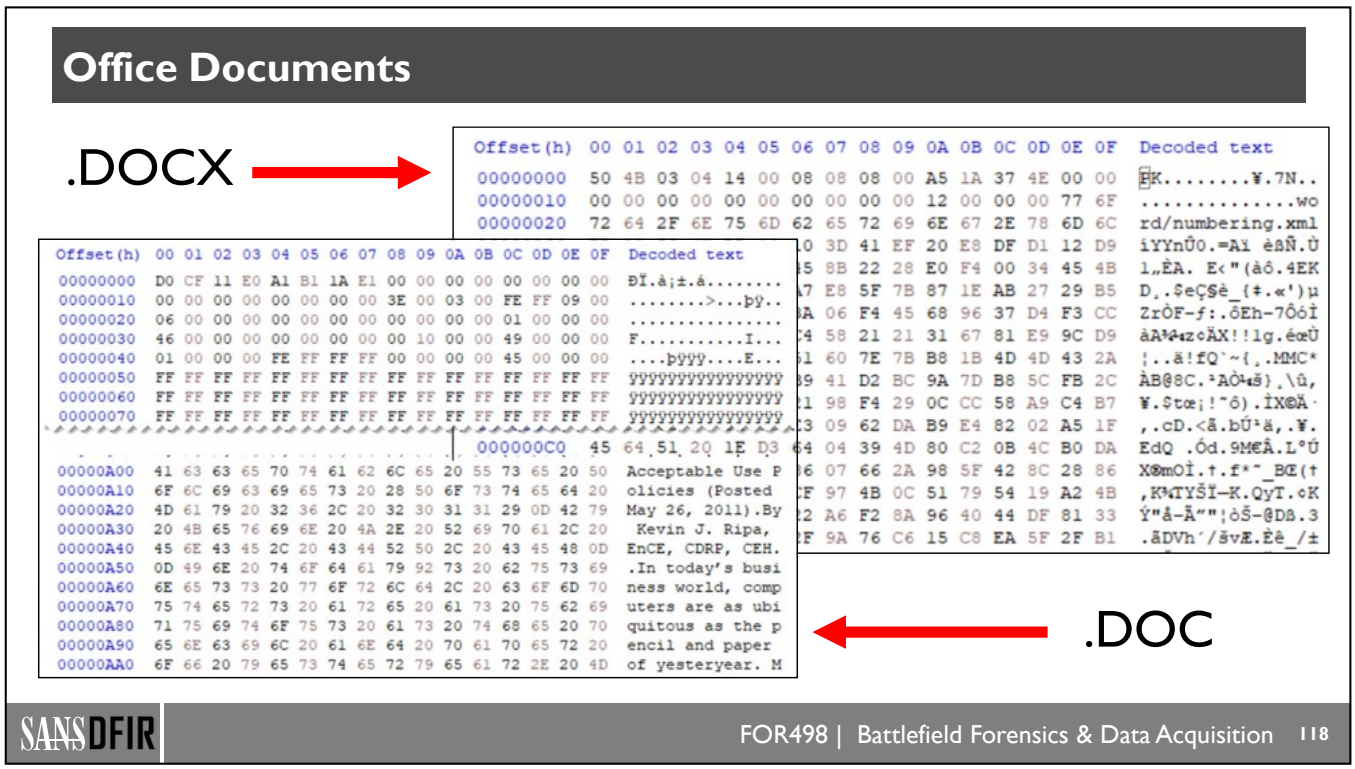


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	48	ÿØÿà..JFIF.....H
00000010	00	48	00	00	FF	E1	02	FE	45	78	69	66	00	00	49	49	.H..ÿá.þExif..II
00000020	2A	00	08	00	00	00	0A	00	0F	01	02	00	12	00	00	00	*..............
00000030	86	00	00	00	10	01	02	00	0B	00	00	00	98	00	00	00	†...........˜...
00000040	12	01	03	00	01	00	00	00	01	00	00	00	1A	01	05	00	....
00000050	01	00	00	00	A4	00	00	00	1B	01	05	00	01	00	00	00	....¤...........
00000060	AC	00	00	00	28	01	03	00	01	00	00	00	02	00	00	00	¬...(...........
00000070	31	01	02	00	0A	00	00	00	B4	00	00	00	32	01	02	00	1.......´...2...
00000080	14	00	00	00	BE	00	00	00	3B	01	02	00	25	00	00	00	....¾...;...%...
00000090	D2	00	00	00	69	87	04	00	01	00	00	00	F8	00	00	00	Ò...i‡......ø...
000000A0	00	00	00	00	4E	49	4B	4F	4E	20	43	4F	52	50	4F	52	....NIKON CORPOR
000000B0	41	54	49	4F	4E	00	4E	49	4B	4F	4E	20	44	37	30	30	ATION.NIKON D700
000000C0	00	00	48	00	00	00	01	00	00	00	48	00	00	00	01	00	..H.......H.....
000000D0	00	00	56	65	72	2E	31	2E	30	31	20	00	32	30	30	39	..Ver.1.01 .2009
000000E0	3A	30	35	3A	32	33	20	31	30	3A	35	34	3A	30	32	00	:05:23 10:54:02.
000000F0	4B	65	76	69	6E	20	41	2E	20	4D	63	47	69	6C	6C	20	Kevin A. McGill
00000100	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000110	20	20	20	20	00	00	1F	00	9A	82	05	00	01	00	00	00	....š,......

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 117

When it comes to familiarizing yourself with various file types, it need not be a protracted process. Using a hex editor, simply open known file types. If you open a.jpg file in a hex editor and look at the first few bytes, and then continue this through a number of.jpg files, it will not take long for you to become familiar with what the first few bytes of every single .JPG file looks like.

With .JPG files though, outside of the file signature, the hex data can differ widely. A full and intact .JPG taken with a digital camera or smartphone will typically have that device's identifiers at the beginning of the file. In the case of photos that have been resized or 'washed', there may be no identifiable metadata in the hex output. But the one thing that does not change is the first 3 bytes of FF D8 FF.



It is also very important to understand what compound files look like, as well as their structure. Microsoft Office documents are a great example of this. Since Microsoft Office 2007, the structure of office files has changed. [1] Prior to this, office documents had their data within the structure in plain text. In the case of a Word document prior to office 2007, opening the document in a hex editor would allow you to see all of the contents in plaintext. As of 2007, Microsoft went to an OpenXML [2] format for their files. As well, they created a proprietary compression methodology in order to keep all components in one place. To look at an office document in a hex editor is to potentially say or misinterpret the file as being a zip file. Both types of files start with the ASCII characters PK. This is because, like a zip file, office documents are also compressed.

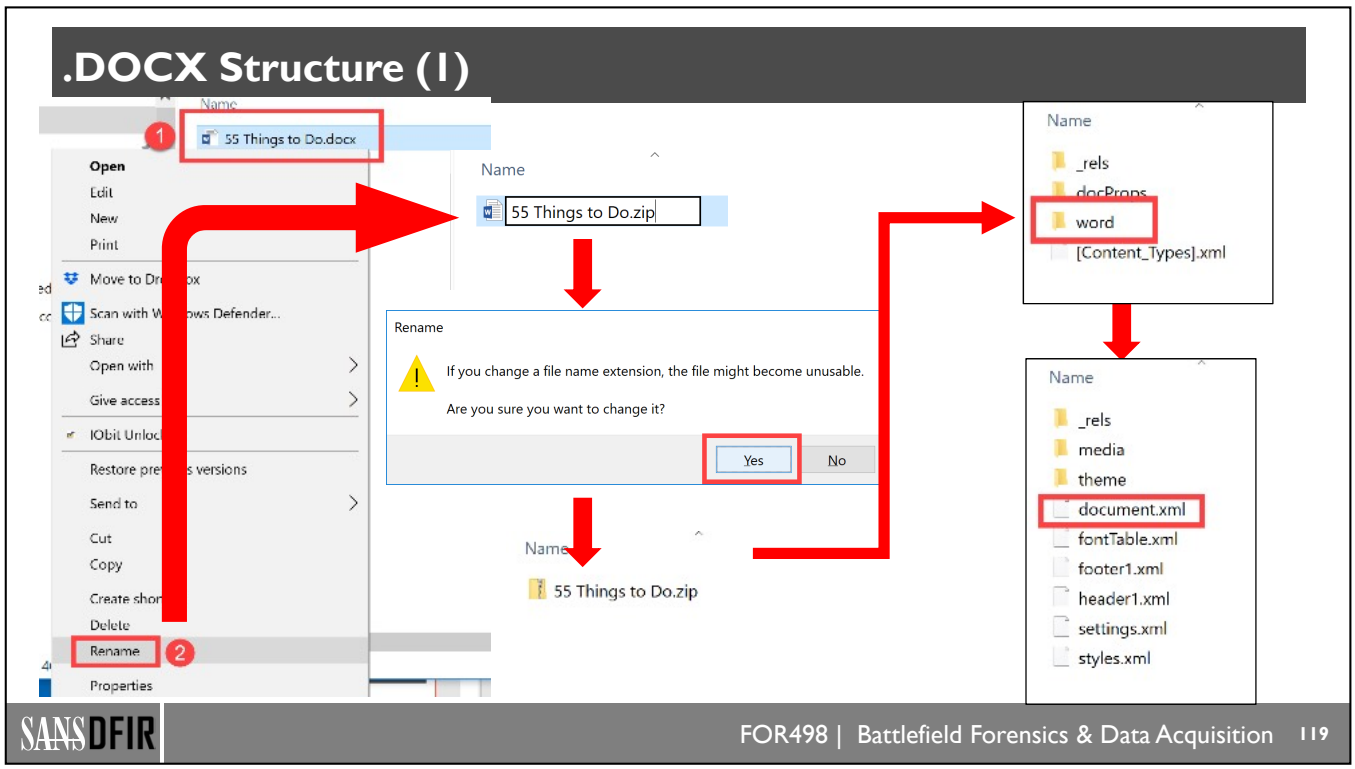
Critically, when a word search is done against a data set that is in a compound file format, it will not hit on the search term even if the term exists. As seen in the slide, the DOCX is not in a human readable format. The file needs to be mounted, or unpacked, before it can be searched. A great deal of potential evidence has been missed in cases where examiners using expensive forensic suites failed to first mount compound files before running search terms. If a compound file exists in unallocated file space though, how can you mount it? The answer is that you can’t. You need to carve it first. This is the paradox. You cannot find the data by search term, because the file is compound, and you can’t use automatic carving, because you may not have an intact file header to trigger the carve.

Understanding the various data structures within an office document are imperative if you wish to do partial recovery of damaged office documents. [3]

[1] .DOC vs .DOCX | `https://for498.com/7vxps`

[2] OpenXML | `https://for498.com/oldas`

[3] Forensic implications of OpenXML | `https://for498.com/3db8q`

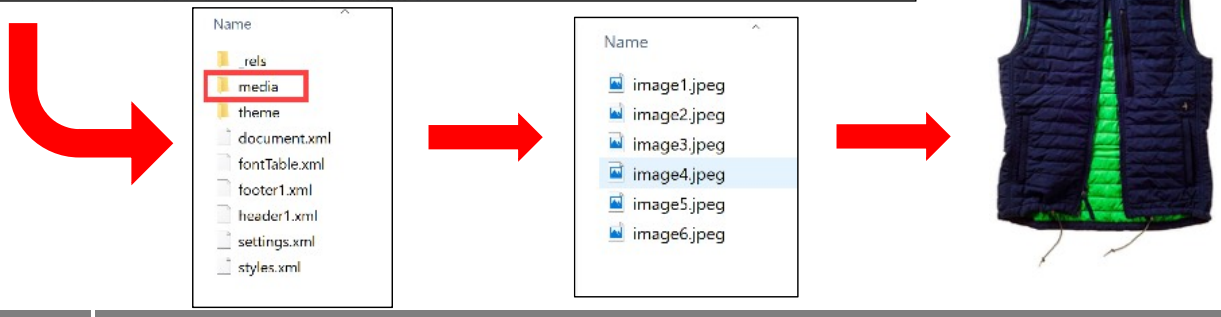


Being that office documents, whether they be PowerPoint, Word, Excel, etc. are compound files, it is interesting to see how they're constructed using OpenXML. The easiest way to see the construct of an office file is to simply rename it. Find any given office file and right-click on it and change the file extension to .zip. Confirm the change and you will see the icon changed to that of a zip file. Now double-click on the zip file and when it opens, you will see a number of folders and files inside.

Depending on the size and complexity of the document, there can be quite a few various directories, subdirectories, and files. In the case of a Word document, we are primarily concerned with a file named document.xml that can be found in the word folder at the root of the zip file.

.DOCX Structure (2)

```
        <w:rFonts w:hAnsi="Helvetica" w:ascii="Helvetica"/>
        <w:sz w:val="34"/>
        <w:szCs w:val="34"/>
        <w:shd w:color="auto" w:val="clear" w:fill="fefffe"/>
        <w:rtl w:val="0"/>
        <w:lang w:val="en-US"/>
    </w:rPr>
    <w:t>Driving around Flathead Lake in Montana, a friend and I stopped at what we thought was a small lakeside
        park. It was actually frontage on a massive piece of property. The house on the property was under
        construction, a big extravagant thing. They</w:t>
</w:r>
<w:r>
  - <w:rPr>
        <w:rFonts w:hAnsi="Helvetica" w:ascii="Helvetica" w:hint="default"/>
        <w:sz w:val="34"/>
        <w:szCs w:val="34"/>
```



Opening the document.xml is not pretty, but it will allow you to view the text contents of the document.

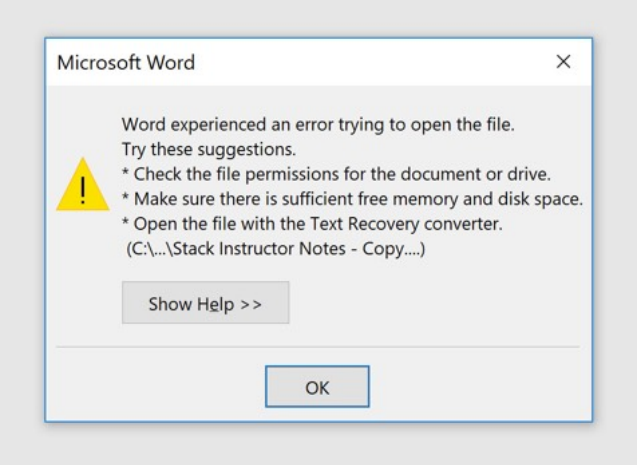
Inside this same word folder you may see another folder titled media. If the Word document has any images or videos or anything other than text embedded in it, each of these will be found as an individual file inside the media folder.

When it comes to recovering these files, it is very common for traditional carvers to recover corrupted versions of the documents. Then when you double-click on the document, open it, and it doesn't open, stating that it's a corrupt file, you don't realize that by simply changing the extension on the file you may very well be able to recover at least the text portion of the document.

Pretty much every component of a Word document is kept track of somewhere inside this XML structure. For example, it is possible to see when a document was altered, or edited, or maybe has had a paragraph added a year after the rest of the document. We have used these methods to prove the genesis and lineage of reused Word documents such as those found within law offices, or any company that uses the same document over and over again as a template for different files.

In the case of other types of Office documents, the same structure exists, with slightly different names. In the case of .PPTX, each slide has its own .XML file.

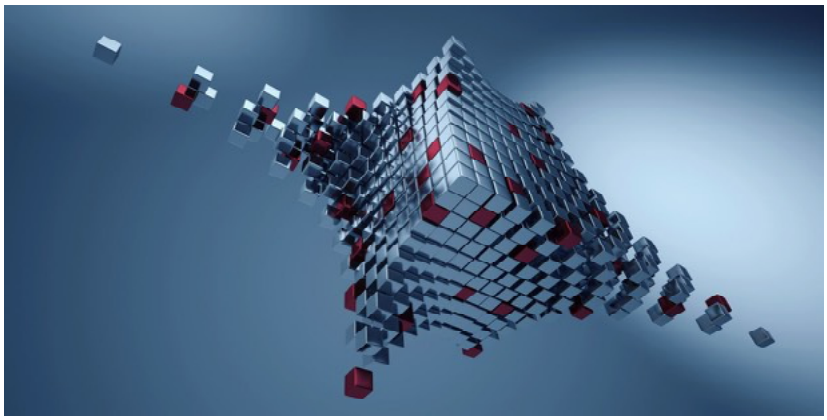
Misguided Message



No longer does this message have to lead to frustration. The only thing worse than getting this message is deleting the file thinking there is nothing in it, when in fact, it was recoverable.

Data Rebuilding

- Take known good header code & prepend to file
- Manipulate code so substantive content isn't changed, but it is now viewable



Data rebuilding is the task of taking partial data and manipulating it in such a way that the file will actually open with a viewer in order for a layperson such as your client or a jury to read or view the contents. Especially in the case of a PDF or a photograph, without the beginning of the file, even if the vast majority of the file still exists, the file will not open. This is where the tedium starts.

As we know by now, the beginning of any given file is unique to that file type. Opening any file of that type will have the same file signature, and in many cases the same data structure. Knowing this, the possibility exists that we can take a known good piece of data from the beginning of a known file and paste it on the beginning of a partial file. In other words, imagine we have been able to recover most of a PDF document, however the file header was overwritten, as well as part of the beginning of the file. Of course we don't know how much was overwritten, however we can take a known good PDF file, open it in a hex editor, and copy a certain amount of the beginning of the file. We can then paste this on the beginning of our partial PDF file that we recovered. With a mixture of patience, skill, and luck, we may very well be able to open the file. Although this may or may not work, it does work just enough of the time to make it a viable option.

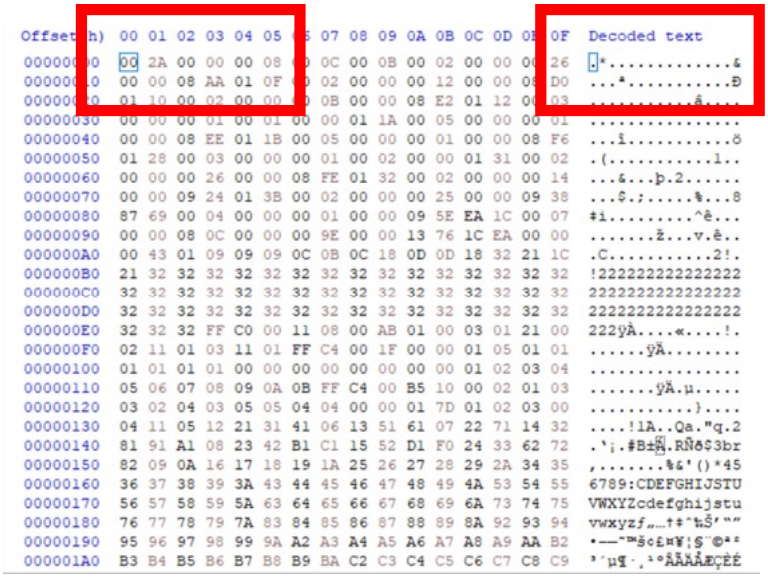
Be very cautious about using repair tools that you have found online. Many free repair tools are in fact malware in disguise. Having said that, some tools do exist that are very capable. Because of the diversity of these programs and the myriad host of problems you may be trying to solve, we cannot make recommendations across the board. When it comes to Office documents though, two very competent programs rise to the top.

Even after we have done our rebuilding of Office files, there may be times that we cannot open the file. We will first need to repair the .zip file [1] or try repairing the document itself. [2] DiskInternals also has a number of other corrupt file recovery utilities.

[1] DiskInternals Zip Repair | `https://for498.com/0r4is`
[2] Corrupt docx Salvager | `https://for498.com/gtvzw`

A Corrupt .JPG

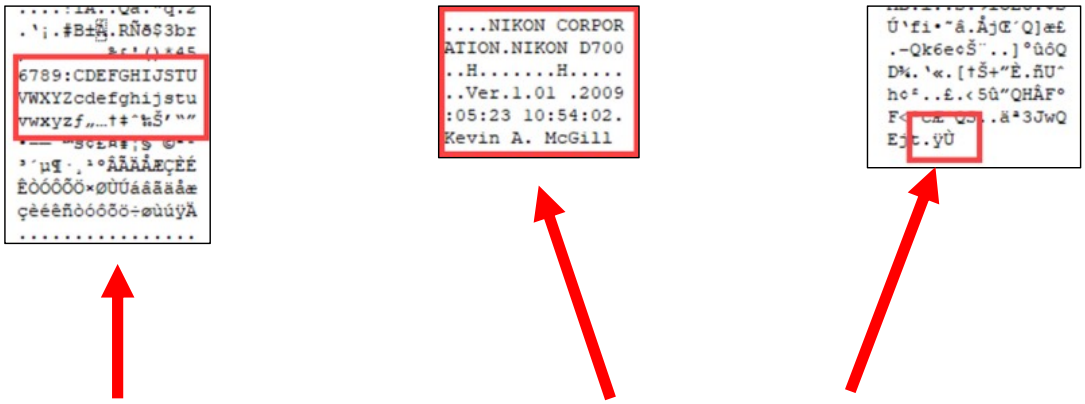
BustedPic.jpg
It appears that we don't support this file format.



Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	00	2A	00	00	00	08	00	0C	00	0B	00	02	00	00	00	26	.*.....&
00000010	00	00	00	08	AA	01	0F	00	02	00	00	00	12	00	00	00	...ª.....Ð
00000020	01	10	00	02	00	00	00	0B	00	00	08	E2	01	12	00	03	.....â....
00000030	00	00	00	01	00	01	00	00	01	1A	00	05	00	00	00	01	....
00000040	00	00	08	EE	01	1B	00	05	00	00	00	01	00	00	08	F6	...î...........ö
00000050	01	28	00	03	00	00	00	01	00	02	00	00	01	31	00	02	.(...........1..
00000060	00	00	00	26	00	00	08	FE	01	32	00	02	00	00	00	14	...&...þ.2......
00000070	00	00	09	24	01	3B	00	02	00	00	00	25	00	00	09	38	...$.;.....%...8
00000080	87	69	00	04	00	00	00	01	00	00	09	5E	EA	1C	00	07	‡i.........^ê...
00000090	00	00	08	0C	00	00	00	9E	00	00	13	76	1C	EA	00	00	.......ž...v.ê..
000000A0	00	43	01	09	09	09	0C	0B	0C	18	0D	0D	18	32	21	1C	.C...........2!.
000000B0	21	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	!2222222222222222
000000C0	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	2222222222222222
000000D0	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	2222222222222222
000000E0	32	32	32	FF	C0	00	11	08	00	AB	01	00	03	01	21	00	222ÿÀ....«....!.
000000F0	02	11	01	03	11	01	FF	C4	00	1F	00	00	01	05	01	01	......ÿÄ........
00000100	01	01	01	01	00	00	00	00	00	00	00	00	01	02	03	04	....
00000110	05	06	07	08	09	0A	0B	FF	C4	00	B5	10	00	02	01	03	.......ÿÄ.µ.....
00000120	03	02	04	03	05	05	04	04	00	00	01	7D	01	02	03	00	...........}....
00000130	04	11	05	12	21	31	41	06	13	51	61	07	22	71	14	32	....!1A..Qa."q.2
00000140	81	91	A1	08	23	42	B1	C1	15	52	D1	F0	24	33	62	72	.'¡.#B±Á.RÑð$3br
00000150	82	09	0A	16	17	18	19	1A	25	26	27	28	29	2A	34	35	,.......%&'()*45
00000160	36	37	38	39	3A	43	44	45	46	47	48	49	4A	53	54	55	6789:CDEFGHIJSTU
00000170	56	57	58	59	5A	63	64	65	66	67	68	69	6A	73	74	75	VWXYZcdefghijstu
00000180	76	77	78	79	7A	83	84	85	86	87	88	89	8A	92	93	94	vwxyzƒ„…†‡ˆ‰Š'""
00000190	95	96	97	98	99	9A	A2	A3	A4	A5	A6	A7	A8	A9	AA	B2	•—˜™š¢£¤¥¦§¨©ª²
000001A0	B3	B4	B5	B6	B7	B8	B9	BA	C2	C3	C4	C5	C6	C7	C8	C9	³´µ¶·¸¹ºÂÃÄÅÆÇÈÉ

When we try to open a corrupt .JPG file, we see a message as in the slide. When we open the file in a hex editor, we immediately see that we have no file header.

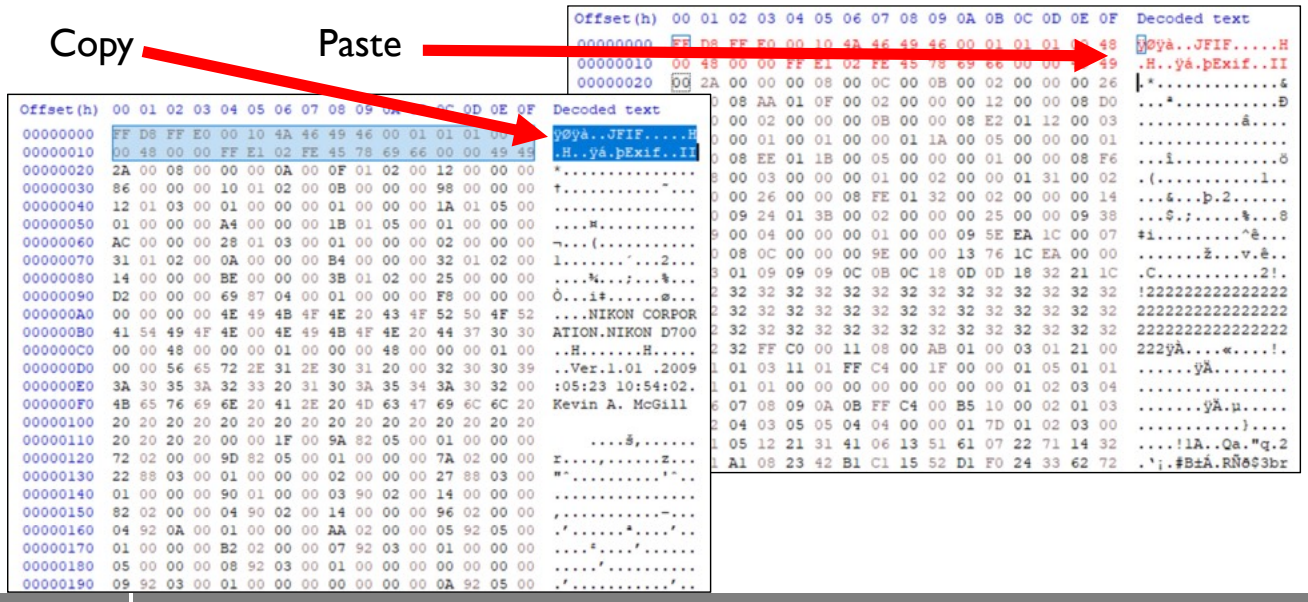
Repairing a .JPG (1)



Numbers and alphabet in sequence, Metadata, and the ÿÙ as the file footer are all indicative of a .JPG

In the case of non-ASCII types of files such as a .JPG, how do we know that it is a .JPG if the file header is no longer there? When we understand what a .JPG file looks like in a hex editor, we can get a good sense. First there is the file footer that looks somewhat like “yu”. As well, we can often tell by viewing metadata. The code shown in the slide, also shows a pattern that is commonly found at the beginning of photo files.

Repairing a .JPG (2)



Open a known good .JPG file in a hex editor and copy some of the first few bytes of the file, and paste them at the beginning of the broken file. How many? Trial, error, and experience will dictate. Start small, and if it doesn't work, come back and copy a bit more. If you have tried to copy and paste 600 bytes and it still hasn't worked, it probably won't work. It doesn't work every time, but when it does work, it will make you dance. Once the copy and paste is done, save the file, and try to open it.

Success

This page intentionally left blank.

Summary

- Automated carving only works on files with a header
- Just because a header no longer exists, does not mean the important data is not recoverable
- Understanding what signatures look like is a must
- Being able to carve with a hex editor is an important skill
- Patience is a virtue

This page intentionally left blank.

Exercise 6.3A-C

Data Carving & Rebuilding

Synopsis: In this exercise, you will use a hex editor to identify file types via signature, and manually carve for deleted .jpg, .doc, and image files. We will then carve, recover, and repair partially overwritten files that a carving tool would not have extracted.

Average Time: 35 Minutes

This page intentionally left blank.

Exercise 6.3A-C - Details

- Data is often available, even if it is not in a format that the examiner is familiar with.
- Data carving and rebuilding can be very time consuming and frustrating.
- If the data does not present itself in a manner that you are accustomed to, try something else.
- What is believed to be an entire file may not be.
- A forensicator could spend hours trying to repair a file and have no success.
- If you receive an error warning that a document cannot be opened, and you don't investigate further, this can have significant consequences.
- It takes practice to know what is and is not possible.

This page intentionally left blank.

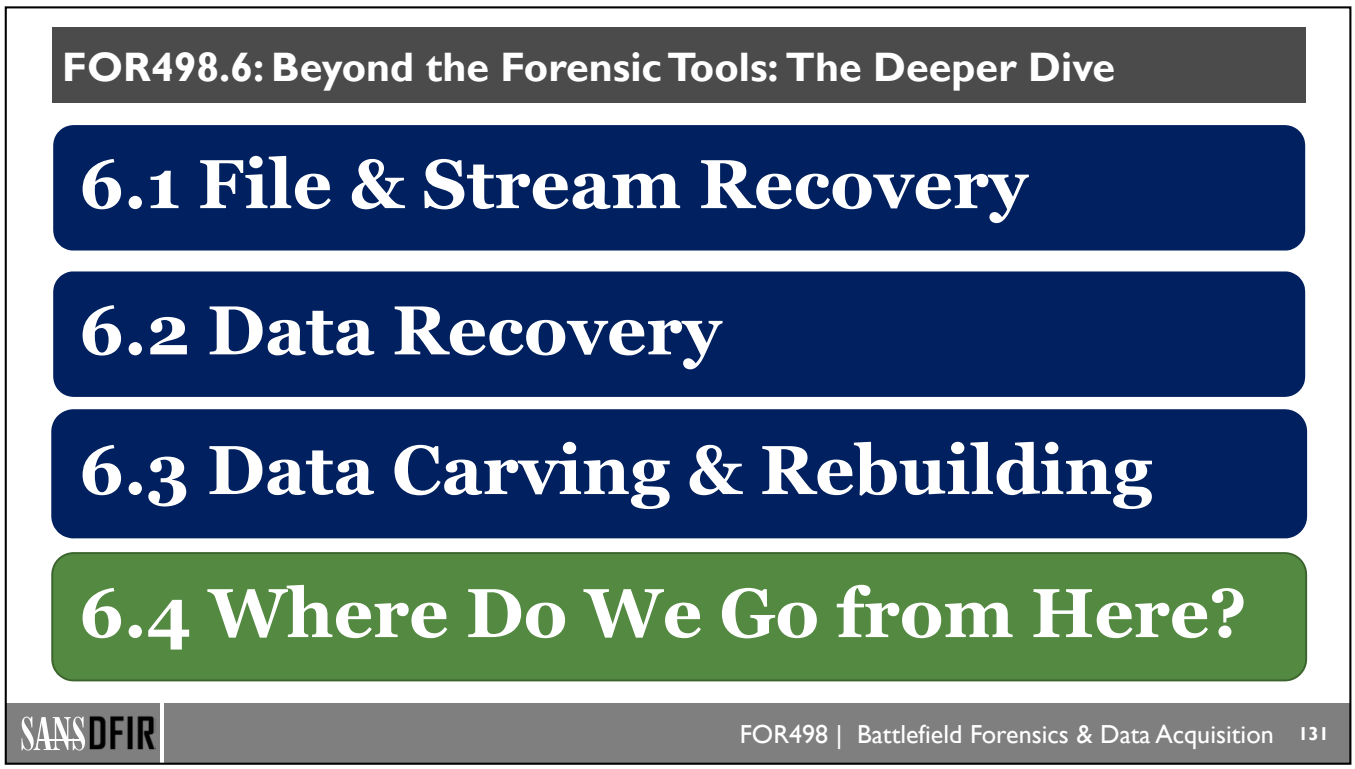
Exercise 6.3D-E

Data Carving & Rebuilding

OPTIONAL, OUT OF CLASS EXERCISE

Synopsis: In this exercise, you will carve for damaged files and determine/extract information from metadata that would not have been available or immediately visible. You will also carve for a broken .jpg file, and repair and open it.

This page intentionally left blank.



This page intentionally left blank.

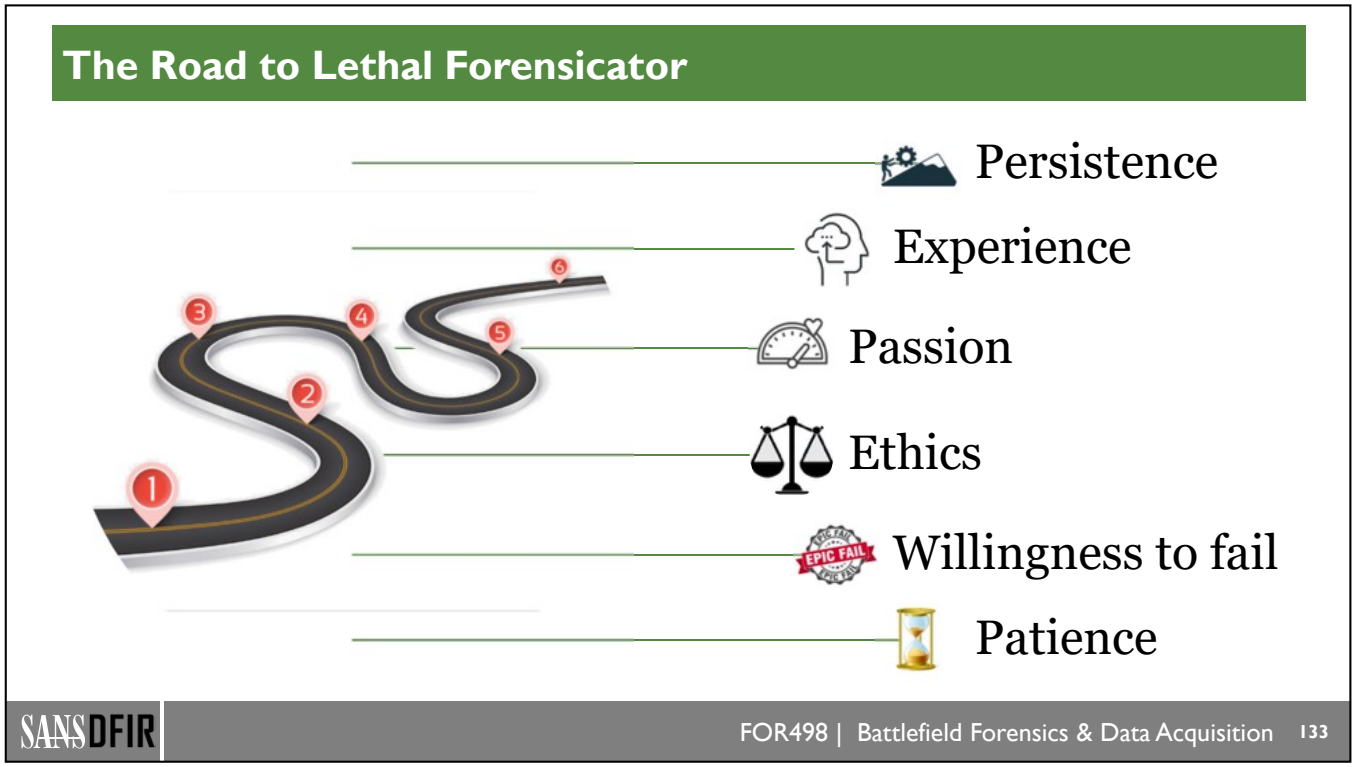
Where Do We Go from Here?

The Road to Lethal Forensicator

To Specialize, or Not?

Going Deeper

This page intentionally left blank.



They say it takes 10,000 hours of practice to become an expert. Now this does not mean that simply putting in the time makes that a guarantee, but rather, the takeaway from this notion is that it takes a long time to get good at anything, and DFIR is no exception. While it is essential to put the time in, the hours alone are not enough. So, with the fact that mastering aspects of DFIR will take time, let's look at some other qualities that will go into making you a lethal forensicator.

Persistence

DFIR is not an ideal field for those who expect immediate gratification. With many complex topics and an ever-expanding amount of research, it takes both time and dedication to stick to it and achieve some level of mastery of a given topic.

Experience

Perhaps most related to what was initially discussed, this is just simply time in the trenches, doing the job, analyzing the data, and writing the reports.

Passion

Having a love for the work, the data, the numbers, and the process goes a long way toward being successful. You can only stare at hexadecimal for so long if you do not love what you do (being a little crazy helps too, at times).

Ethics

"Doing the right thing, even when no one is watching" essentially sums this one up. There are times when someone will want you to do or say something contrary to what the data says. Resist the temptation, even though it may cost you personally in the short term. Doing otherwise will certainly catch up with you in the long run.

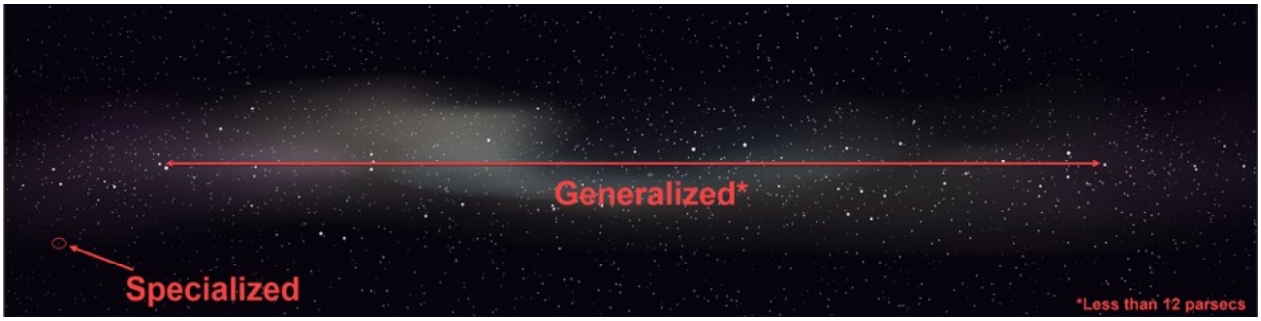
Willingness to fail

This can often be the hardest skill to learn, especially depending on your personality. When we say willingness to fail, we are not saying willingness to make careless mistakes, but rather, trying new things, moving beyond your comfort zone, and so on. If you are in the realm of programming or scripting forensic tools, you may go through many iterations of possible solutions before you have that "aha!" moment that solves the issue at hand. Many hours of work may need to be tossed aside when the wrong road is taken toward a solution. When this happens, move on, solve the problem, and count it toward experience.

Patience

Sometimes it takes time, a lot of time, to work a case, look for an answer, or understand the data. Sometimes it involves new research and producing a new analytical technique. When available tools fail or simply do not exist, will you have the patience to stay the course and see it through?

To Specialize, or Not?



Supplement a strong foundation with deep niche skill

The world of digital forensics and incident response (DFIR) is truly miles wide and miles deep. Because of the wide range of hardware, operating systems, networks, and various artifacts, it is impossible to know it all. At best, we as individuals can hope to master several aspects of DFIR, but even when this is possible, it is generally within a smaller area of expertise, such as network forensics or host-based forensics on Windows. There is just too much information already available, with more being discovered and published every day, to both ingest and retain this new research.

At some point in your career, you will most likely find an area of DFIR that greatly piques your interest, like reverse engineering, memory analysis, file systems, packet analysis, and so on. As we saw on the previous slide, a significant amount of effort is required to become a forensic expert. As you find your niche and invest the time and energy necessary, hopefully other people around you (peers, coworkers, etc.) will also be doing the same. This is ideal because it allows different people and groups to compliment the skills of others. By having other forensic experts around with deep knowledge in certain areas that may be very different than yours, you can get them involved for their insight into a problem you are working on. They, in turn, can do the same, when they need help with your area of expertise.

No one knows it all, and the longer you are in this field, the more you will realize just how little you know! Embrace this fact and strive to be the person who so understands the specialization you have chosen that people seek you out for the hard answers.

Going Deeper

- FOR572: Advanced Network Forensics
- FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting
- FOR500: Windows Forensic Analysis
- FOR518: Mac and iOS Forensic Analysis
- FOR526: Advanced Memory Forensics
- FOR585: Advanced Smartphone Forensics
- FOR578: Cyber Threat Intelligence
- FOR610: Reverse-Engineering Malware

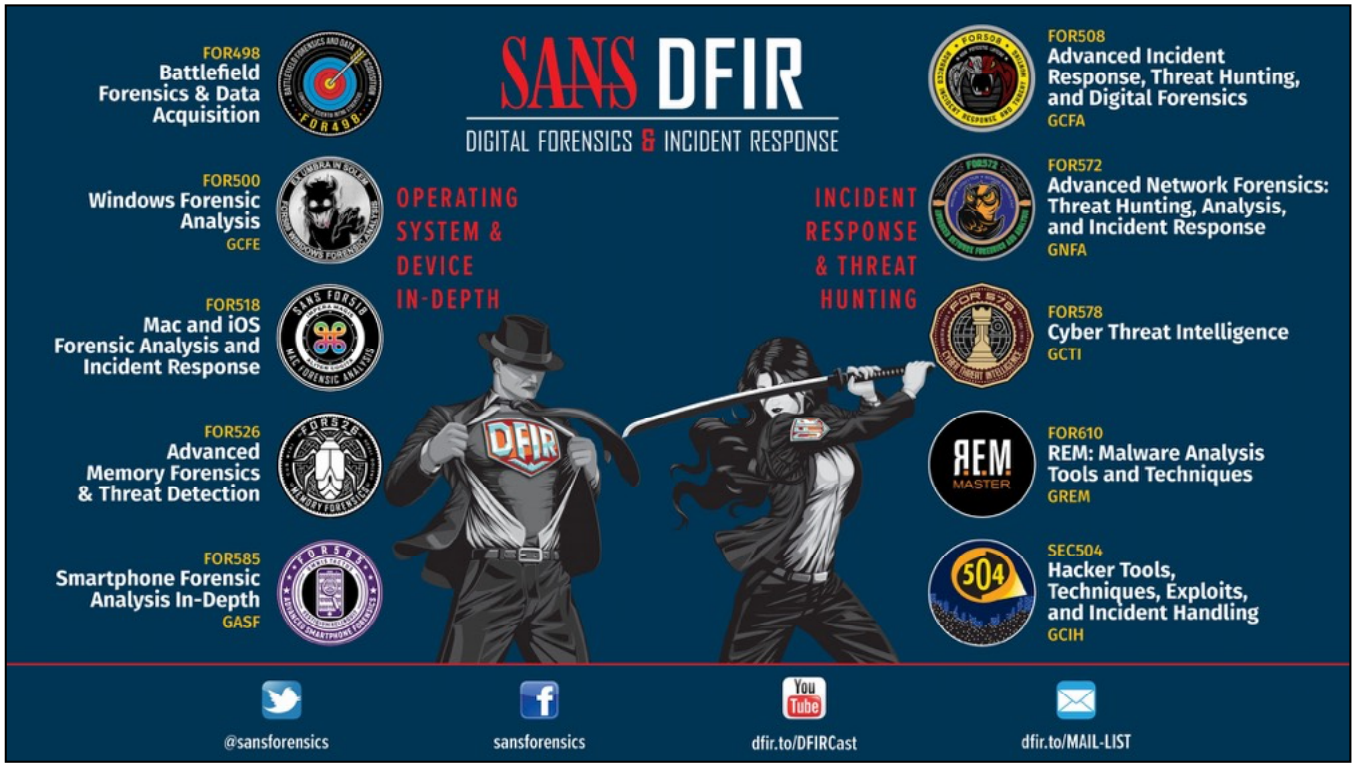
SANS DFIR FOR498 | Battlefield Forensics & Data Acquisition 136

We covered a lot of ground as it relates to acquisition and battlefield forensics, but where to go from here to continue your DFIR pursuit and pursue deeper knowledge in the areas we touched on?

Depending on what you found most interesting or based on the needs of your role with your employer, there are many additional courses to pursue. [1]

Courses ranging from Windows forensics and incident response to Apple and smartphone devices provide an immense amount of detail into key forensic artifacts that provide insights that are valuable in just about every investigation. For those of you who work on the wire, a network forensics course will help take your analytical skills to the next level when looking at network captures and log files alike. Other specialized courses include reverse engineering, memory forensics, and threat intelligence.

[1] SANS Digital Forensics and Incident Response | `https://for498.com/p3dus`



This page intentionally left blank.

COURSE RESOURCES AND CONTACT INFORMATION

Here is my lens. You know my methods. –Sherlock Holmes

INSTRUCTOR CONTACT
Eric Zimmerman
saericzimmerman@gmail.com
twitter @ericrzimmerman

Kevin Ripa
kevin.ripa@gmail.com
twitter @kevinripa

SANS INSTITUTE
11200 Rockville Pike
Suite 200
North Bethesda, MD 20852
301.654.SANS(7267)

DFIR RESOURCES
digital-forensics.sans.org
Twitter: @sansforensics

SANS EMAIL
GENERAL INQUIRIES: info@sans.org
REGISTRATION: registration@sans.org
TUITION: tuition@sans.org
PRESS/PR: press@sans.org

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 138

This page intentionally left blank.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z