498.3
Quick Win Forensics

FOR498.3 | Battlefield Forensics & Data Acquisition

Quick Win Forensics

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

https://twitter.com/ericrz
https://twitter.com/kevinripa

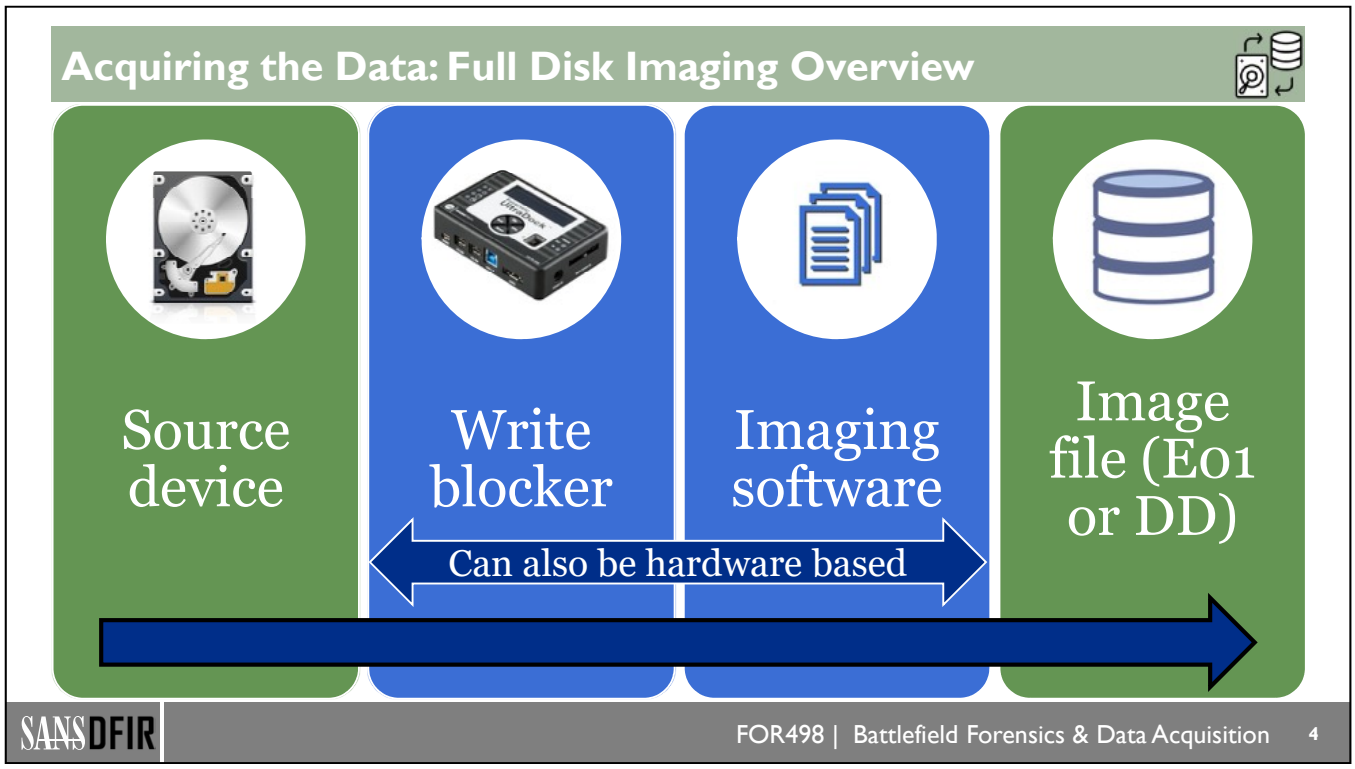
FOR498.3: Quick Win Forensics Agenda

- **3.1 Memory Acquisition & Encryption Checking**
- **3.2 Mounting Evidence**
- **3.3 Triage Acquisition**
- **3.4 Host Based Live Acquisition**
- **3.5 Dead Box Acquisition**

This page intentionally left blank.

Memory Acquisition & Encryption Checking

This page intentionally left blank.



A full disk image is ideally an exact bit for bit copy of a source device stored as a file (rather than copying all the bytes to another hard drive). The term "ideally" is used there because while that is the goal, there are some circumstances (such as when imaging SSDs) where it may not be possible to generate an exact copy of the data. Other possible issues that may prevent this are bad sectors on the source device which cannot be read. These will often appear as empty spaces in the corresponding image.

We have seen the first two pieces in this process already: removing the source from a device and connecting it to a write blocker. In this situation, we need to introduce a third item to the equation; imaging software. The job of imaging software is to read the source device and make a copy of the data into an image file, which will typically be in Expert Witness (E01) format, or in raw (DD) format. As we saw earlier, the primary difference between E01 and DD is compression. For this reason, E01 is the most common type of image format you will create and encounter. There are many software imagers out there, including FTK Imager, X-Ways Imager, Guymager, and trusty old DD. Each has their pros and cons, but all of them are software based, which means a host computer must be present. Once the write blocker is connected to the host computer, the imaging software is started, and the source device is selected. From here, a destination file is selected, and the imaging process initiated.

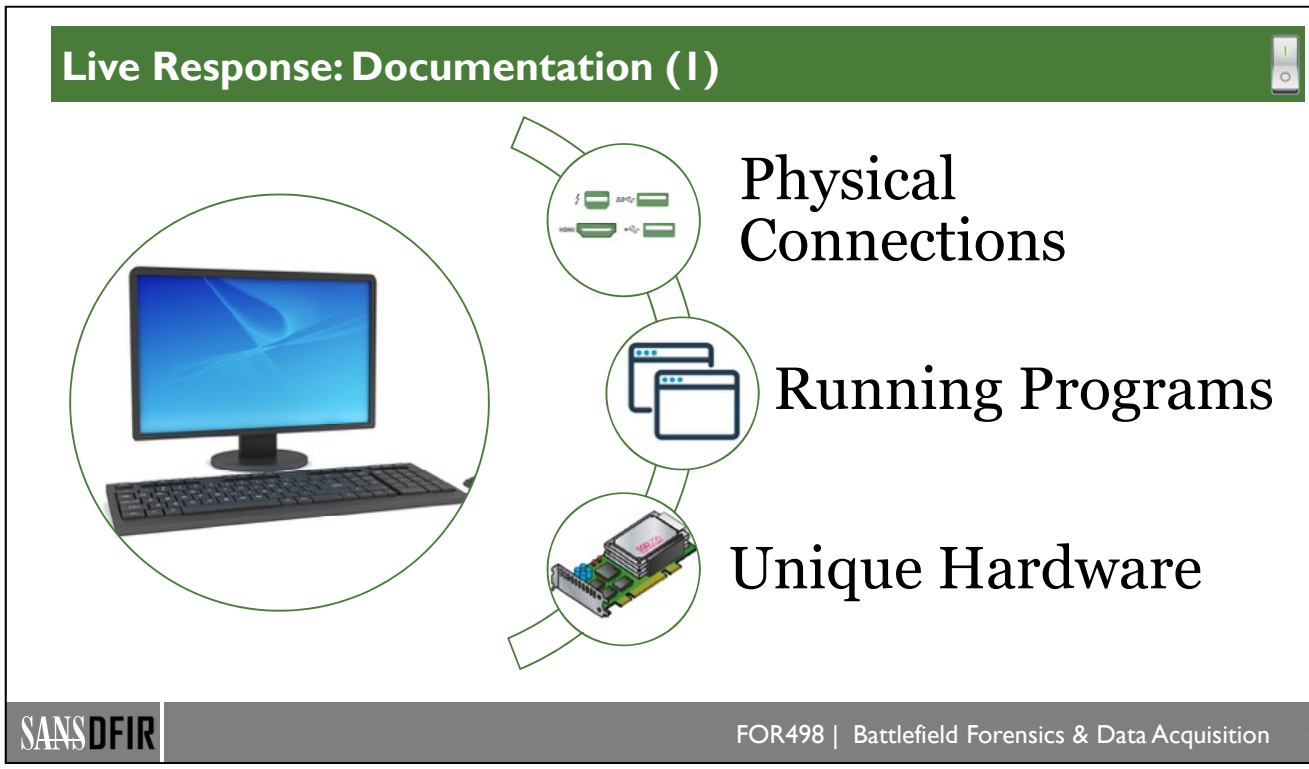
At this point the process is automatic. At the end of the process, the imaging software generates a log of the imaging process including information about the source drive (capacity, number of sectors, etc.), the hash of the source, and whether the destination image verified successfully. In the case where there were unread sectors due to damage, this is also generally included in the report. This report should be kept along with the image file.

Another option instead of a host computer, write blocker, and imaging software is using a dedicated piece of hardware for image creation. In this scenario, a single piece of hardware acts as the host, write blocker, and provider of the imaging software. The source drive is connected to a read only port on the hardware imager and a destination is selected. Some hardware imagers only allow for connecting another hard drive as the destination. In this case, the image file will be created on the destination drive. Other, more advanced hardware imagers can write the image file to a network share.

When using hardware imagers, it is important to not swap the source and destination devices, especially if you need to format the destination device prior to acquisition!

In either scenario, once the imaging process is started, it will take a long time to complete (usually a few hours, but it depends on several factors such as the size of the source drive, how full it is, and so on). So what can you do to pass the time? Review the battlefield forensics data of course!

Whether done on a live system or against a source device connected to a write blocker, performing battlefield forensics on a device and then initiating full disk imaging gives you plenty of time to review key data to start answering questions and generating leads.

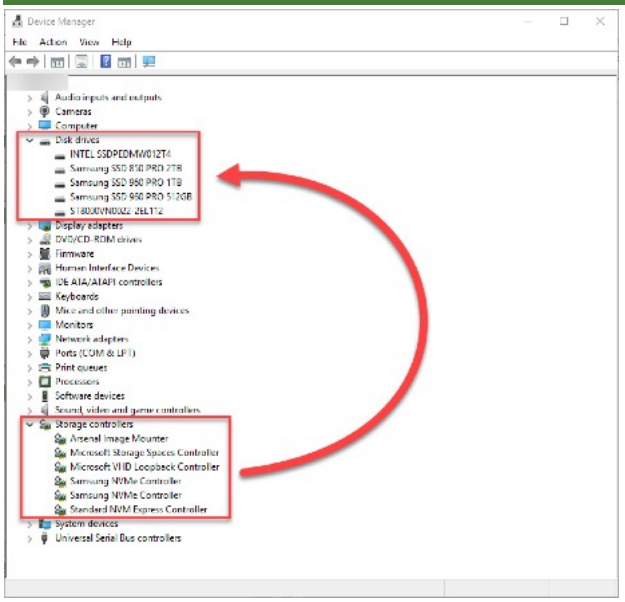


A running system is a blessing, as it gives us several investigative avenues that would otherwise not exist if the system were powered off. These avenues contain rich sources of information that are pertinent to just about every investigation, so we want to take advantage of the situation to maximize the amount of intelligence extracted from the running system. The process of interacting with a running system is typically called live response or triage, but triage can also be done against a system that is turned off, with some work (but not to the same degree as a running machine).

When a live system is encountered, the first thing to do is document the state of the device, including what is displayed on the screen, running applications, etc. When we earlier talked about scene management, we discussed several techniques that can be used to accomplish this. In addition to what is displayed on the monitor, a survey of devices connected to the running system should also be completed. The primary focus for external devices is locating those devices that can store data, such as an external hard drive, thumb drive, etc.

Another thing to consider when documenting a running system is what kind of hardware is present. More specifically, we are not talking about what kind of CPU or how much memory is in a device, but rather, are there any addon cards that provide RAID capabilities? In this situation, how can you determine what devices are present in the system? A visual inspection of the inside of the case may yield clues, but it is often easier, especially if the device is running, to look for third party software on the system that allows for management of the specific hardware.

Live Response: Documentation (2)



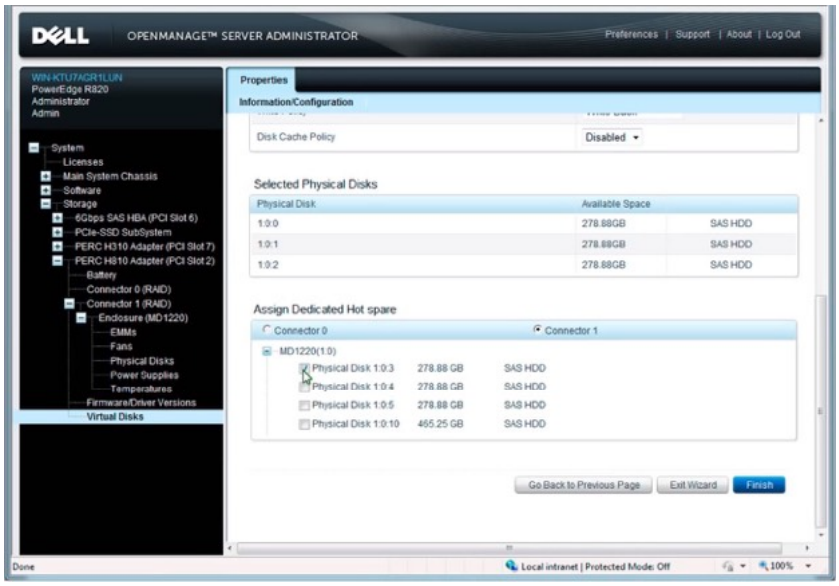
Device Manager provides insight into both controllers and storage

As we saw on the previous slide, we want to ensure that we collect the most volatile data first. Once this is done (by dumping memory), additional documentation can be done by looking at desktop shortcuts, checking the Start menu for vendor specific folders, or checking under the Control Panel's Add/Remove applet to see what software is installed. Another possibility is looking at the Device Manager to see what hardware is installed. While in the Device Manager, inspecting the "Storage controllers" and "Disk drives" section shows any storage volumes present that are being managed by a RAID card, for example.

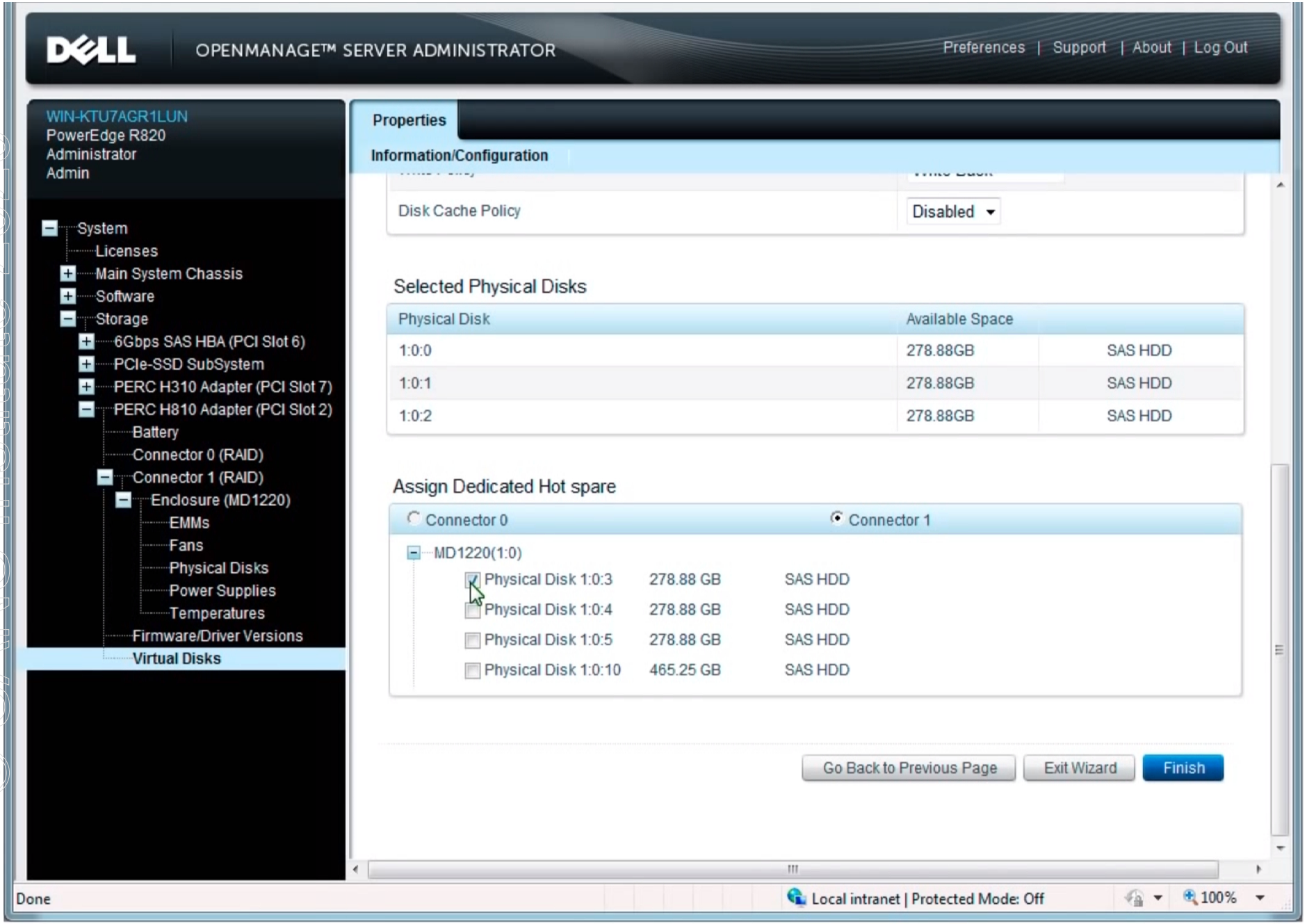
By being diligent and thorough during the documentation phase, future tasks will be easier and less error prone, especially when it comes time to fully acquiring the data from any devices that are present.

Live Response: Documentation (3)

Look for vendor specific software as well



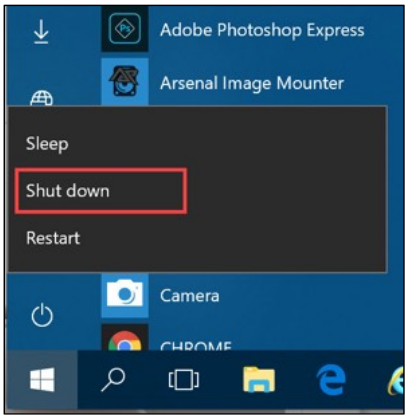
This page intentionally left blank.



The Old Way

Non-server

Server



In the early days of digital forensic acquisition, it was a commonly accepted practice to perform all acquisitions using a "dead box" approach. Those were the days when computers were relatively easy to take apart, at least to the point of accessing the hard drive.

Imaging RAM from a running machine was not being considered, and there were no tools to parse it anyway. Add to that the fact that RAM was relatively small in its size, and it was easy to see why it was not collected.

It was certainly a rare machine that had any kind of encryption running on it, and even then, it was almost certainly not at a physical disk level. With the advent of TPM (Trusted Platform Module), the user could leverage the BIOS and TPM to restrict the boot process by asking for a boot up password at the moment of turning on the computer. Even then, any data recovery company had the ability to remove such passwords, as they were merely written to negative cylinders (or firmware area) of the hard drive, and completely accessible with the right (albeit expensive) tools.

The acquisition process was quite standard, and the methods that were taught to all forensicators if they came across a running machine were thus:

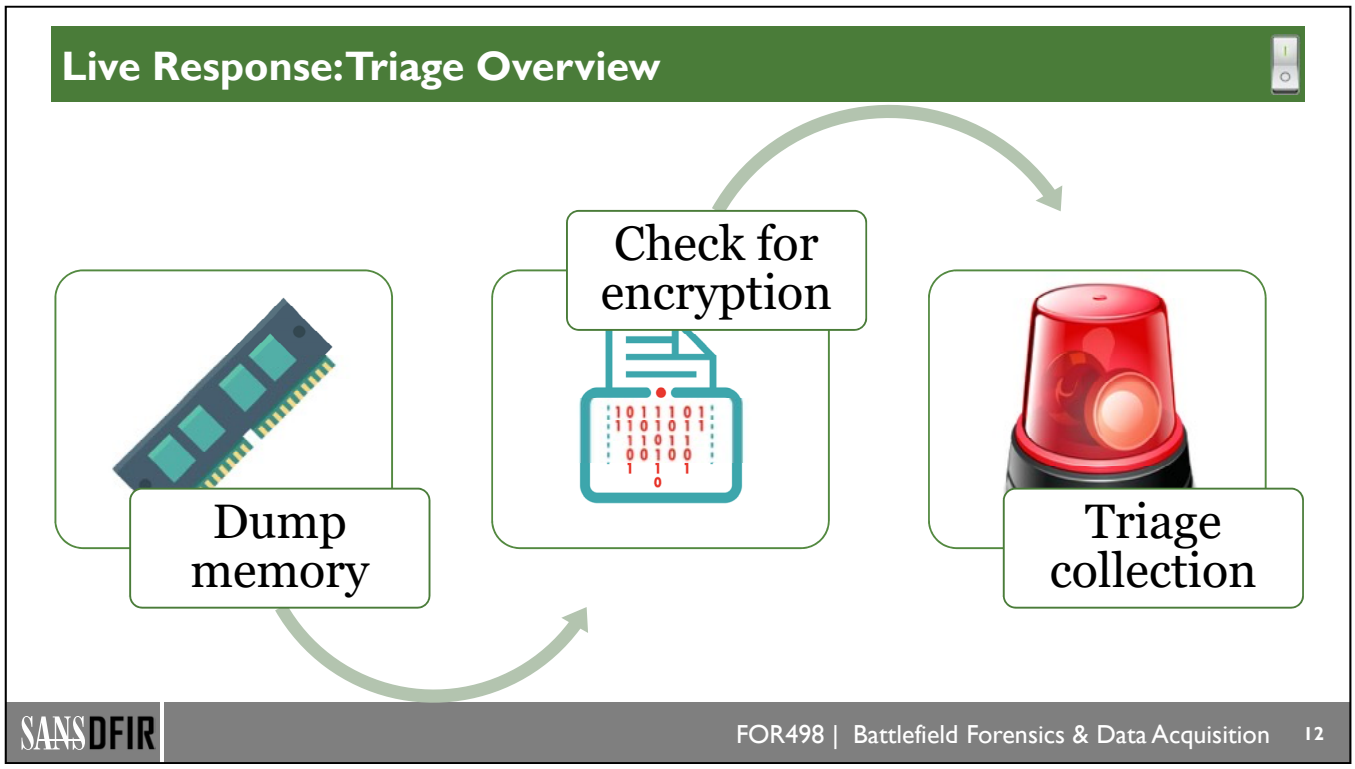
If the machine was a non-server system, pull the plug from the back of the machine.

If the machine was a server, power down in the normal fashion.

The thinking was that if it was a non-server system, it could survive being unplugged with very little to no risk of damage to the data on the hard drive, even though it was not shut down properly.

In the case of servers, they usually needed to be put back into production once the acquisition process was completed. Because of the various configurations of servers (usually multi-disc array), they were set up to run continuously from the first day they were turned on. These machines could run for years and never be turned off. With this type of “burn in”, it meant that the drives in the server got used to operating within a certain set of parameters, such as heat most importantly. This could mean that once the computer had power removed, and the drives had an opportunity to cool, they would be outside of their working parameter and may not come back online properly. This is known as temperature cycling.[1] Disrupting the power to a server could also cause issues with things like RAID tables. Without a properly configured RAID table, the hard drives could not interact with each other appropriately. When balancing the risk of losing data in the enterprise against the potential changes to the data on the discs, it was determined that turning the server off properly presented much less of a risk overall than simply pulling the power.

[1] Temperature cycling | `http://for498.com/zp9ni`



Once the status of the device is documented, live response can be performed on the running system. While we will cover the mechanics of exactly what introducing tools looks like in a future section, for now we want to understand the steps that are typically taken during live response.

ORDER OF VOLATILITY

Dump memory
Since memory is among the most volatile data on a computer, it should be collected first. Dumping memory allows for recovery of running programs, network connections, Registry hives, and a wide range of other useful artifacts.

Check for encryption
While not volatile like memory, running into encryption on a storage device can cause problems if not handled properly. In most cases, when encryption is found on a running machine, it can be bypassed while it remains running. We leverage this by creating a full disk image (usually at the logical volume level) before shutting the computer down. In this way, the unencrypted data is preserved in the forensic image, which can later be more fully analyzed.

If no encryption is found, the machine can be powered down and collected. The process for accessing the data in this scenario would be the same as the dead box scenario, discussed next.

Triage collection
Depending on the case, it may make sense to perform triage collection on the live device using a program like KAPE or similar, that can extract key artifacts from the running system for later analysis. In many cases this is a best practice because it allows for an examiner to look at a key set of artifacts to generate leads, while the time-consuming process of full disk imaging is happening. Triage collection will be covered in more detail in the dead box section.

RAM: To Collect or Not?

Yes! (Whenever possible)

Amount of data lost by not imaging is far greater than amount of data potentially changed when you do

Large amounts of data that exist nowhere else

There is a major tenet to digital investigations that suggests that an examiner must forensically acquire data in such a way as to verifiably ensure that no data has been altered or affected by the acquisition. In fact, many examiners will avow that if the data has been changed in any way, it is no longer evidence. However, this is an erroneous assertion and is simply untrue. Furthermore, it leads to incomplete data collection, and improper examinations and conclusions.

When an examiner or other first responder arrives at a computer to forensically acquire the data on it, notwithstanding proper evidentiary intake, the first determination must be whether the computer is still on. Even if the screen is black and everything appears off, the computer may just be in Sleep or Hibernation mode. Simply jiggling the mouse or tapping the space bar will determine whether this is the case. There should also be activity lights visible on the computer box itself. If the computer is in one of these suspended states, all data is still resident in RAM, and can be collected.

The collection of RAM data undeniably causes changes to the state of the computer data, and certainly leaves a visible footprint. The examiner must weigh the potential damage to data with the potential goldmine of information that will be extracted. In virtually all cases, it is a very necessary evil.

There are several different ways to properly collect RAM data, however the focus is primarily on Windows computers. Collecting RAM on Macs is a bit more problematic because the options for analysis are much more limited.

Especially with Apple laptops, they rarely ever get turned off. Most importantly for laptops, ensure that they are plugged into the power supply before doing anything. Any good lab will have an assortment of power cables in case a device didn't arrive with one. Press the spacebar and/or the mouse a couple of times, and if the computer was simply asleep, it will come on.

The first thing a first responder (or examiner) should be doing upon determining that a computer is in a running state is to isolate it from any networks. The simplest way to achieve this is to disconnect the Ethernet (network) cable from the computer. In the case of wireless connectivity on laptops, this can be achieved by disabling Wi-Fi connectivity. The computer is now prepared to have its RAM imaged.

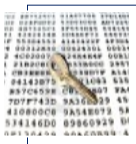
RAM Collection

Processes

Network connections

Open files, Registry keys, and devices

Configuration parameters



Encryption keys and passwords

Memory-only exploits/rootkit technology

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 14

Examiners have been trained to shut the computer off immediately upon reaching it, if it is not already off. Sadly, this is still happening in some cases today!

Although the premise is correct, in today's computing world, if this function is performed, the examiner will have irretrievably lost potentially the most important evidence in a case; that being all data residing within the RAM of the computer.

During the functioning of a computer, the transfer of data to and from the hard drive is the single largest bottleneck in the speed of the user experience. To overcome this, a computer has another area that it stores information, called the RAM. The RAM typically consists of one or more Printed Circuit Boards (PCB) containing solid-state memory chips. The reading and writing of data from these chips is exponentially faster than from the hard drive itself.

The hard drive is the main storage component on a computer and holds all the information that has been saved, including programs and the operating system. When a user needs to use a program, all the program components necessary are brought into the RAM for use. Besides the program components, there is a great deal of other activity going on in the RAM. Computer system processes such as network connections, user activity, passwords, program activity, chat activity, and many other functions and data live in the RAM. In fact, there is malicious software in use now that only lives in RAM. It is clear to see then, that a great deal of data resides in the RAM that does not exist anywhere else on the machine. Once the machine is turned off, this data is irretrievably cleared.

In today's computers, it is quite normal to have 8 GB of RAM, with many having 16 GB or more. Compare that to a mere 15 years ago when entire hard drives were that size, and it quickly becomes apparent that if an examiner is not collecting all the data in the RAM, they are missing a vast amount of data. As a result, the approach to seizing and examining computers has changed.

Entire investigative outcomes can hinge on information that exists solely in RAM, and examiners ignoring this, and/or not collecting it are doing a disservice to themselves, their clients, and the industry. As the saying goes, you cannot “unbake” the cake. During a forensic examination of the data, it is possible to absolutely determine whether the computer was booted and running at the time an examiner took it into custody. The first question to anyone providing you with a forensic image should be whether the RAM contents were collected, and if not, why not. One good reason for “why not” is if the computer was found in the “off” state. Another good reason would be if there was life hazard, and you had to grab the computer and run. Another very relevant reason would be something like the case of a computer in a workplace where you are trying to extract evidence of something that happened months ago. A new, unrelated user has been using the computer, and has re-cycled it numerous times since then. There would be no reason to image the RAM, because it couldn’t possibly retain anything of relevance.

What is sitting in memory? You have all the processes, files, directories, and any other information that could be sitting in the address space of memory. You can use this information to piece together old history and commands that a previous individual may have typed on the system. You might discover old emails or websites that the user surfed to. You might find residue from exited processes. And probably most importantly, you will likely have passwords for both encryption and other programs in clear text still sitting in memory.

With the increased use of encryption, particularly whole disk encryption utilities like Windows BitLocker, PGP, and VeraCrypt, it is more important now than ever before for incident responders to image RAM and collect volatile data on any powered-on system they respond to. While it is the most volatile piece of evidence, it is also one of the most valuable.

In most cases, programmers will not obfuscate or encrypt these sensitive areas in memory. It will merely be sitting there in plain text. However, there won't be ASCII art surrounding it stating that “THIS IS THE PASSWORD”. The string would exist though.

If you are interested in diving deeper into the kinds of artifacts that can be recovered from memory, FOR508[1] or FOR526[2] should be the next course you take. These courses contain advanced forensic techniques including analyzing RAM to find running and exited processes, extracting files from memory, and other ways to examine volatile data. [3]

Up until recently, memory analysis was essentially limited to performing string searches and byte searches through what was seemingly random data. The memory image file format has been recently reverse engineered and new tools exist that will allow for a more granular approach to examining the contents of memory.

[1] FOR508 | `https://for498.com/dlr6x`
[2] FOR526 | `https://for498.com/ld2bt`
[3] Encryption Keys -> BitLocker | `https://for500.com/u8m3q`

RAM Acquisition Considerations

- Will the benefit outweigh the harm?
- Can you explain and support your decision?
- More often today, you must support decision NOT to collect RAM
  - Can be because data destruction is visibly happening, so machine must be shut down immediately
  - Scene safety

Initially you will need a removable media such as a USB thumb drive, or external hard drive. This media is required both as a place to run the acquisition software from, and a location to store the extracted data. It would be more efficient to run the software from one location and deposit the extraction to another, however you are now introducing two new pieces of hardware to the computer instead of one. The goal is to make as few changes as possible. As a result, it is recommended that the media used should be a solid state external hard drive. The loss in efficiency from reading and writing to the same device is more than gained by the exponential efficiency of data handling on solid state media versus traditional rotating media.

When choosing storage media, an examiner can certainly use most any USB drive, however you get what you pay for in terms of acquisition and write speeds. A “run of the mill” USB drive of 64 GB that can be purchased quite cheaply will potentially take 20-30 minutes or more to write out an acquisition of 8-16 GB of RAM. An external solid-state hard drive will cost more but will perform the task in a matter of a few minutes. The difference cannot be exaggerated.

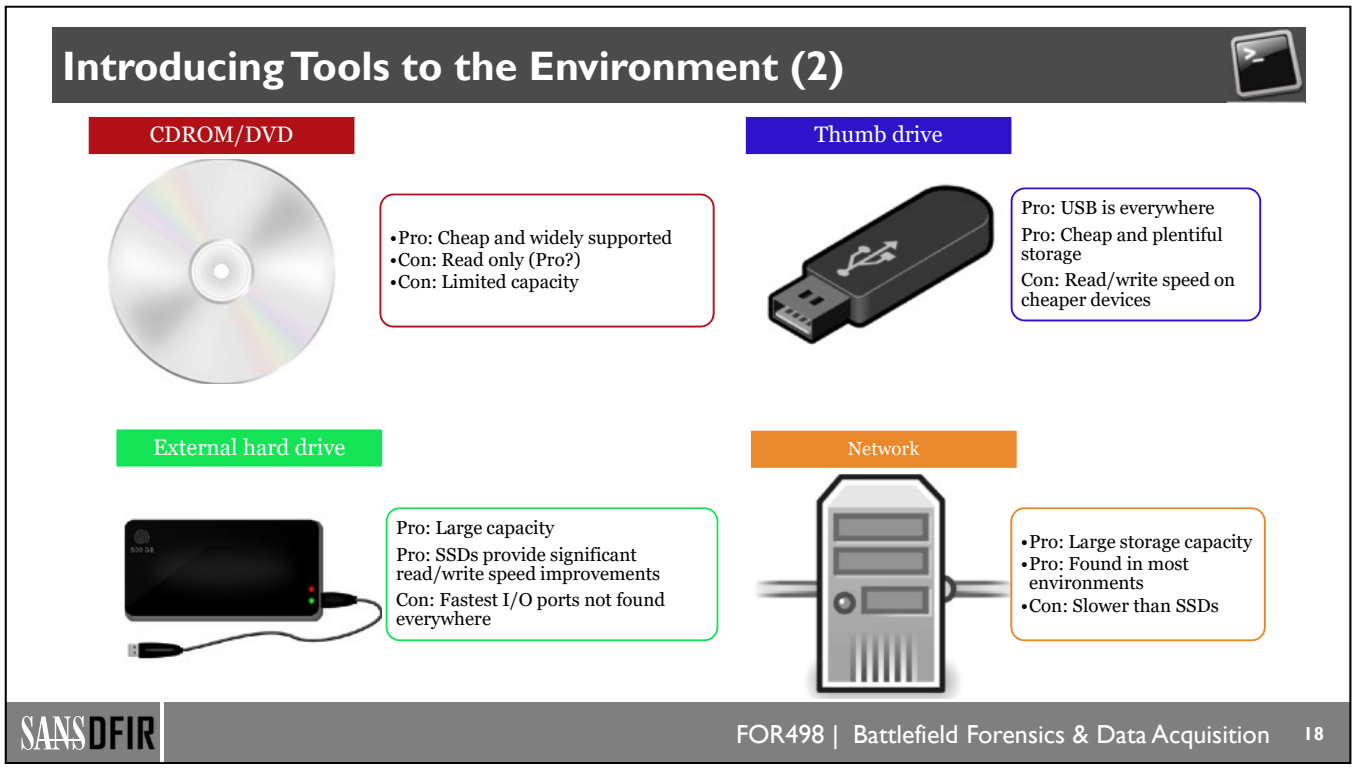
It goes without saying that if you are doing a RAM dump, the computer is ON and LIVE. As such, you need to be extra careful about your processes and steps. The number one rule to live by is DOCUMENT YOUR ACTIONS. Everything you do on the live machine needs to be recorded either via video/pictures, or in writing. Better yet, both. You WILL be changing system settings, and you WILL be overwriting data. This is quite alright, if you have a GOOD reason, and can justify your actions.

Introducing Tools to the Environment (1)

Ah, the classic dilemma as to the best way to introduce a set of live response tools onto a running computer! If only if it were that simple, but alas, there are many more questions to consider:

- How much space will you need?
- How should the drive be formatted?
- What other resources are available?
- Can I trust the software on a computer, or do I bring my own?
- What kind of restrictions are in place in the environment I have to operate in?
- If my primary choice fails or is not available, do I have a backup?

Most computers these days will have several things in common, such as the availability of a network, several USB ports, optical drives, and so on. But is this universally true? In the cases where you know what to expect when you walk in the door, it becomes easier to ensure you have the right equipment from the start, but for those people who operate in more uncertain environments, such as law enforcement or the military, it pays to have several backup plans.



As you can see above, there are many different ways to go about getting your toolset onto a target system. In every case, there will be pros and cons, but as a general best practice, we recommend the following:

- Large capacity, quality made external storage from a reputable vendor
- Ideally, use an SSD for maximum read/write speeds
- Format drive as NTFS for Windows analysis, exfat for everything else
- Always introduce your own trusted binaries to a computer

Once you have decided on an external device, it is often a good idea to document the details about the device in order for future forensic examination to account for your connecting of your external device to a target machine. For USB devices, a tool like Nirsoft USBDeview[1] provides details such as make, model, and serial number of the USB device. To make effective use of this technique, connect your USB device to your own machine and run USBDeview, then record the details for your particular device. This information can then be included in any subsequent documentation or report generated which should then be passed on to forensic examiners conducting a more extensive forensic review of the computer.

Another consideration to keep in mind is where to save forensic images and/or memory dumps. This depends on several things, such as the size of the hard drives being imaged, the amount of memory on a computer, and how many computers you have to potentially interact with. Getting a good answer to this question is made much easier if you know what to expect before you get on scene, but even in those cases, if you think you need one terabyte, bring four, because you will inevitably run into a situation where there will be more computers than were initially known about.

With that in mind, possible solutions include:

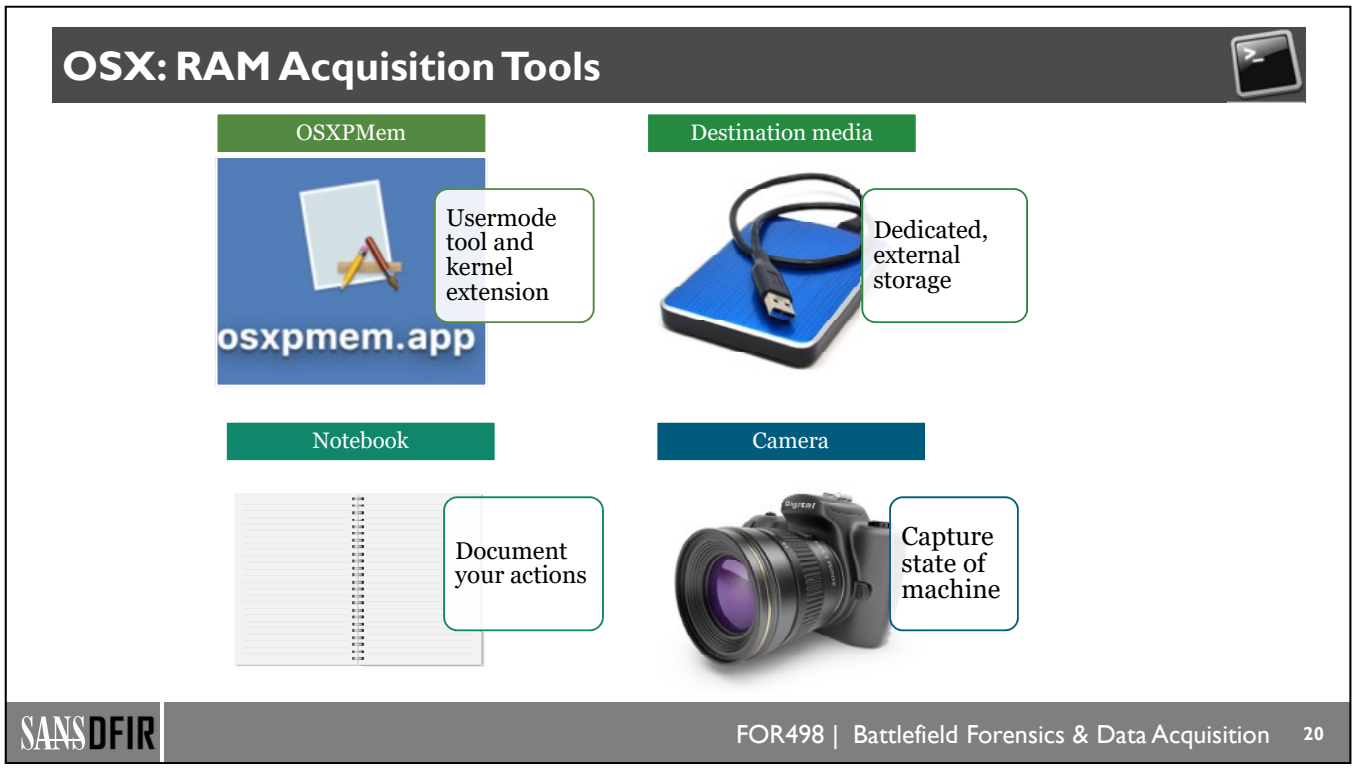
- If your external drive is big enough (it should be. Remember the rule of x4!), write to the drive. Keep in mind that low quality drives combined with large RAM capacities can result in RAM smear[2];
- Is a network target available? This is a possible (though much less optimal) solution.

Finally, always be sure to take the security of the environment into consideration. Who else has access to where the image/memory is being captured to? This is especially concerning when using a network share to save files to. In the case of an external drive, have you maintained a proper chain of custody to ensure no one had the ability to tamper with a device and what is stored on it?

Once you have your device in place, it's time to start deciding what tools to copy to the device for use in an environment. This will often be a mix of various tools, both command line and GUI based, that can be leveraged on a system to perform some level of live response, memory acquisition, or even full disk imaging.

[1] Nirsoft USBDeview | `http://for498.com/qhvdt`

[2] RAM smear (Page 3) | `http://for498.com/2idbv`



Imaging RAM on a Mac requires mostly the same process as with Windows. You need good quality DESTINATION media that already contains the software you will use for your acquisition. You will need something to visually document your work as it happens. The camera from a cellular telephone will serve the purpose. You should have a notebook to take notes of what is happening, what you have done, and when it occurred. Keeping track of this time is not as simple as the time you started and the time you ended. You should be noting the time of each step in the process as well. Nobody was ever accused of documenting something too much.

In the case of Mac RAM acquisition, a capable free tool is Rekall's OSXPMem[1][2]. It is a tool that is run from the command line. It consists of two components; the usermode acquisition tool 'osxpmem', which parses the accessible sections of physical memory and writes them to disk in a specific format, and a generic kernel extension 'pmem.kext', that provides read only access to physical memory. After loading it into the kernel it provides a device file ('/dev/pmem/'), from which physical memory can be read. If you are not conversant with command line, this process may seem confusing, however it is quite simple in its use. It does require constant testing though, because its ability changes potentially every time that Apple changes how they do things (which is often).

Two other tools (not free) are MacQuisition by Blackbag Technologies, and Recon by Sumuri.

[1] OSXPMem Description | `https://for498.com/l40ew`

[2] Download OSXPMem v3.2 | `https://for498.com/tj73o`

Windows CLI: Comae Dumpit (1)

```
Usage: DumpIt [Options] /OUTPUT <FILENAME>

Description:
  Enables users to create a snapshot of the physical memory as a local file.

Options:
  /TYPE, /T             Select type of memory dump (e.g. RAW or DMP) [default: DMP]
  /OUTPUT, /O           Output file to be created. (optional)
  /QUIET, /Q            Do not ask any questions. Proceed directly.
  /NOLYTICS, /N         Do not send any usage analytics information to Comae Technologies. This is used to improve our services.
  /NOJSON, /J           Do not save a .json file containing metadata. Metadata are the basic information you will need for the analysis.
  /LIVEKD, /L           Enables live kernel debugging session.
  /COMPRESS, /R         Compresses memory dump file.
  /APP, /A              Specifies filename or complete path of debugger image to execute.
  /CMDLINE, /C          Specifies debugger command-line options.
  /DRIVERNAME, /D       Specifies the name of the installed device driver image.
```

DumpIt has many optional command line switches, but when executed without any options, DumpIt will use recommended defaults and dump memory to the same directory it was executed from.

When collecting memory, remember that we want to take every step we can to minimize changing anything on a running computer. While “zero changes” is impossible, our choice of programs to execute on a host will directly impact how much change is happening as a result of us performing memory collection (or any type of triage for that matter). For this reason, in almost every case, it makes sense to use a command line to as opposed to a graphical user interface (GUI) application to capture memory. By minimizing how much memory a program takes to execute, we ensure we do as much as possible to minimize the changes on the running system.

The first tool we will look at is DumpIt, from Comae[1]. DumpIt is a command line tool that requires administrator privileges to run. The options for DumpIt are shown below that can be used to tailor your collection, avoid DumpIt asking for confirmation, providing a filename to capture memory to, etc. In its simplest usage, running DumpIt without any arguments will also capture memory. We will look at an example of this next.

Usage: DumpIt [Options] /OUTPUT <FILENAME>

Description:

Enables users to create a snapshot of the physical memory as a local file.

Options:

/TYPE, /T	Select type of memory dump (e.g. RAW or DMP) [default: DMP]
/OUTPUT, /O	Output file to be created. (optional)
/QUIET, /Q	Do not ask any questions. Proceed directly.
/NOLYTICS, /N	Do not send any usage analytics information to Comae Technologies. This is used to improve our services.
/NOJSON, /J	Do not save a .json file containing metadata. Metadata are the basic information you will need for the analysis.
/LIVEKD, /L	Enables live kernel debugging session.
/COMPRESS, /R	Compresses memory dump file.
/APP, /A	Specifies filename or complete path of debugger image to execute.
/CMDLINE, /C	Specifies debugger command-line options.
/DRIVERNAME, /D	Specifies the name of the installed device driver image.

Examples:

Create a local memory snapshot:

```
DumpIt.exe /OUTPUT snapshot.bin
```

Enable live kernel debugging session:

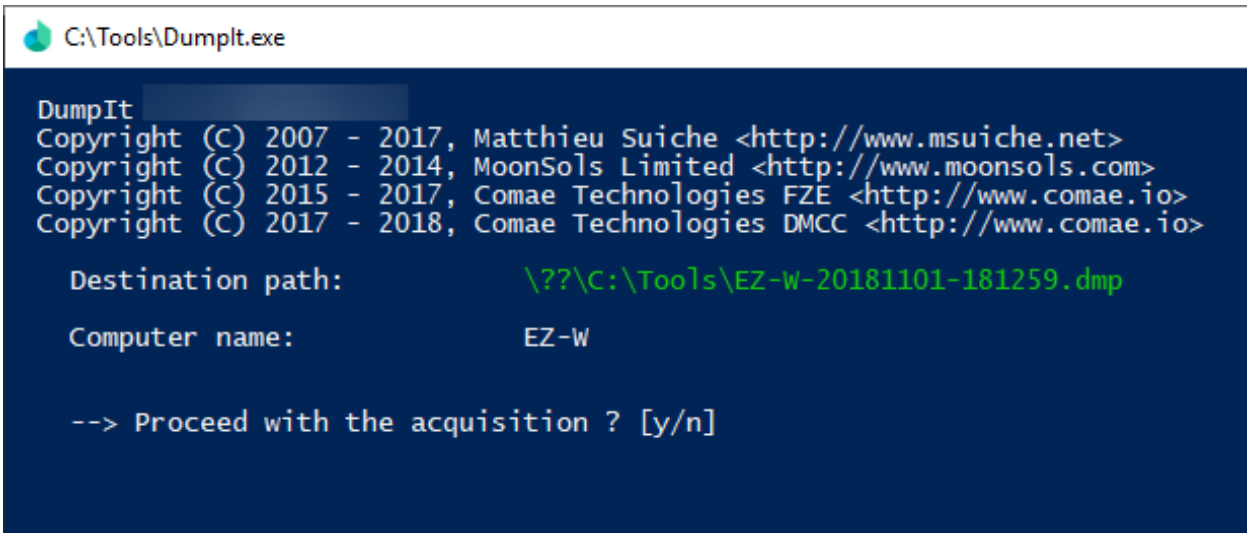
```
DumpIt.exe /L /A <debugger image path>
```

Extract metadata from machine in live kernel debugging session:

```
DumpIt.exe /L /A Dmp2Json.exe /C "/Y srv*C:\Symbols*http://msdl.microsoft.com/download/symbols /C \"/live /all /datetime /archive /snapshot C:\Snapshots\Snapshot\""
```

[1] Comae Technologies | `http://for498.com/xawuj`

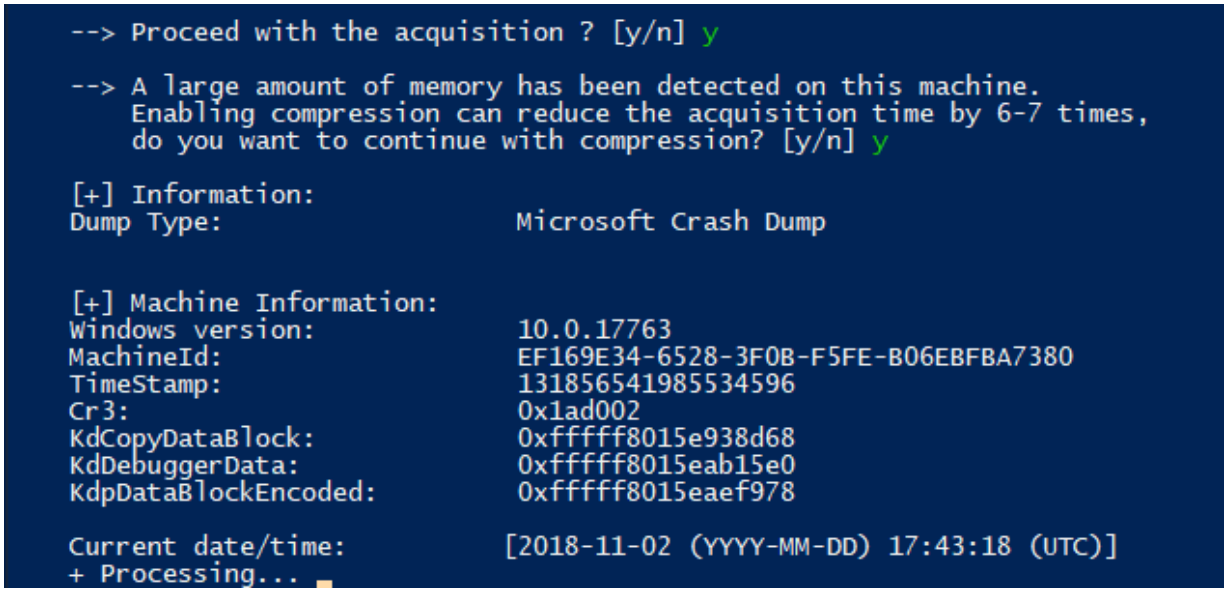
Windows CLI: Comae Dumpit (2)



DumpIt.exe lives in C:\Tools directory and is run without options

The easiest way to run DumpIt is to simply execute it without any arguments. Notice that when this is done however, DumpIt asks you to confirm each step in the process along the way. Once the acquisition is started, there is little else to do. In some cases, when DumpIt sees a large amount of memory on a host computer, it will prompt to enable compression of the capture which can save significant space. Keep in mind however, that some memory analysis tools may not know what to do with a compressed file, and therefore it will be necessary to decompress the memory capture back to a RAW form.

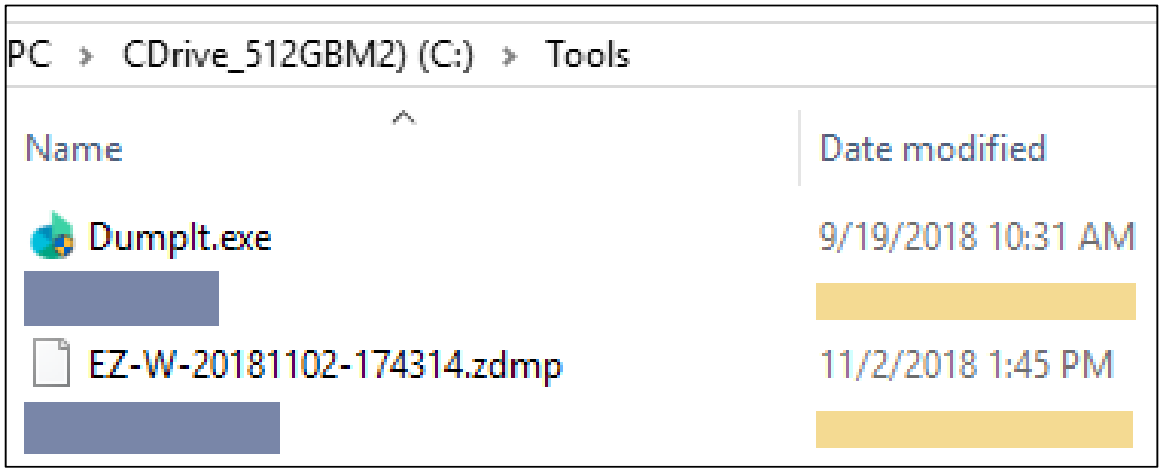
Windows CLI: Comae Dumpit (3)



Once all questions have been answered, DumpIt displays details about the computer it is running on and begins dumping memory to the same directory where DumpIt.exe was executed from.

Notice that by default, the name of the memory capture includes the machine name and a timestamp in the filename. This makes it easy to keep track of where memory captures originated from, as well as when they were collected. The size of the file generated by DumpIt will of course be a factor of both the amount of memory in the computer, as well as how much of that memory contains compressible data (as well as to what degree that data is compressible).

Windows CLI: Comae Dumpit (4)

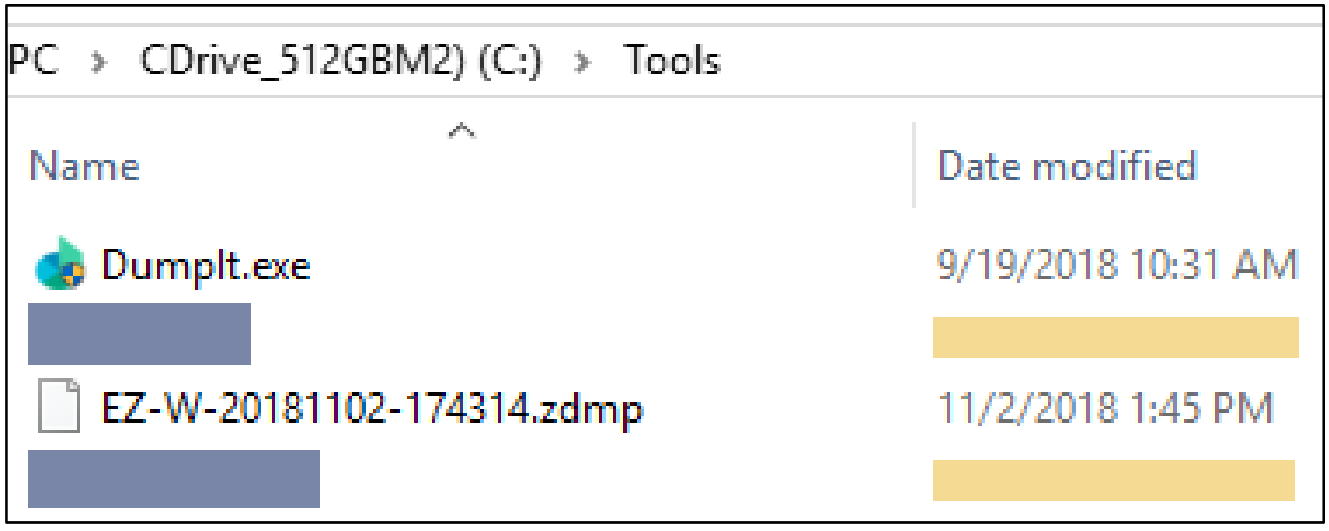


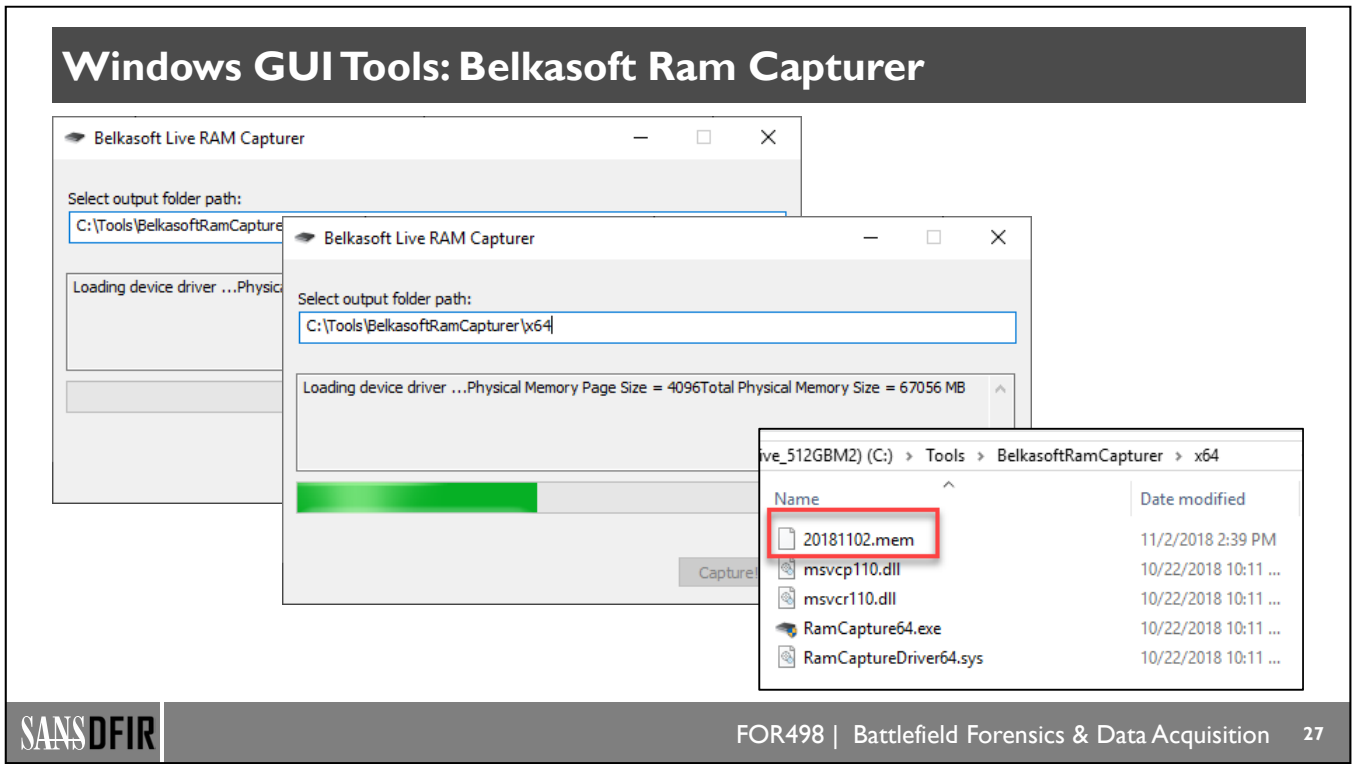
Here we see an example of the file that DumpIt generated.

C:\Tools\DumpIt.exe

```
DumpIt
Copyright (C) 2007 - 2017, Matthieu Suiche <http://www.msuiche.net>
Copyright (C) 2012 - 2014, MoonSols Limited <http://www.moonsols.com>
Copyright (C) 2015 - 2017, Comae Technologies FZE <http://www.comae.io>
Copyright (C) 2017 - 2018, Comae Technologies DMCC <http://www.comae.io>

  Destination path:           \??\C:\Tools\EZ-W-20181101-181259.dmp

  Computer name:              EZ-W


  --> Proceed with the acquisition ? [y/n]
```

```
  --> Proceed with the acquisition ? [y/n] y

  --> A large amount of memory has been detected on this machine.
      Enabling compression can reduce the acquisition time by 6-7 times,
      do you want to continue with compression? [y/n] y

  [+] Information:
  Dump Type:                  Microsoft Crash Dump


  [+] Machine Information:
  Windows version:            10.0.17763
  MachineId:                  EF169E34-6528-3F0B-F5FE-B06EBFBA7380
  TimeStamp:                  131856541985534596
  Cr3:                        0x1ad002
  KdCopyDataBlock:            0xfffff8015e938d68
  KdDebuggerData:             0xfffff8015eab15e0
  KdpDataBlockEncoded:        0xfffff8015eaef978

  Current date/time:          [2018-11-02 (YYYY-MM-DD) 17:43:18 (UTC)]
  + Processing...
```





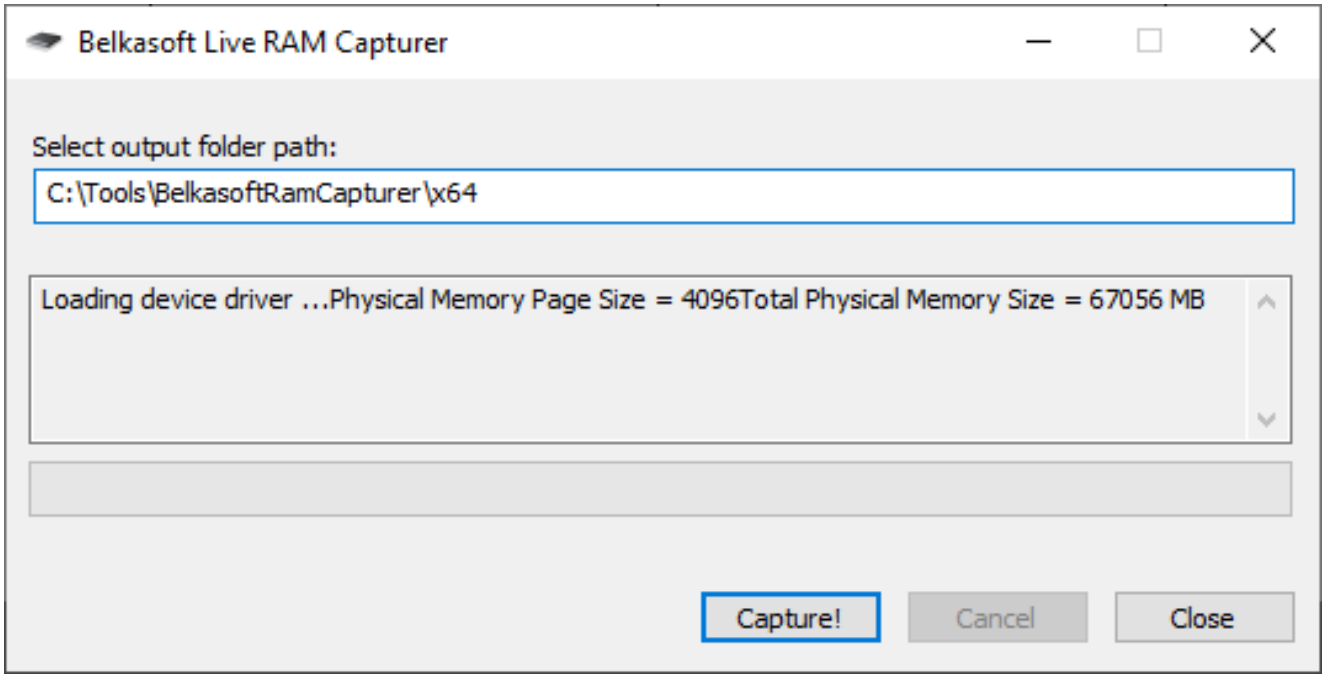
The next tool, bridges the gap between a pure CLI tool and a GUI tool.

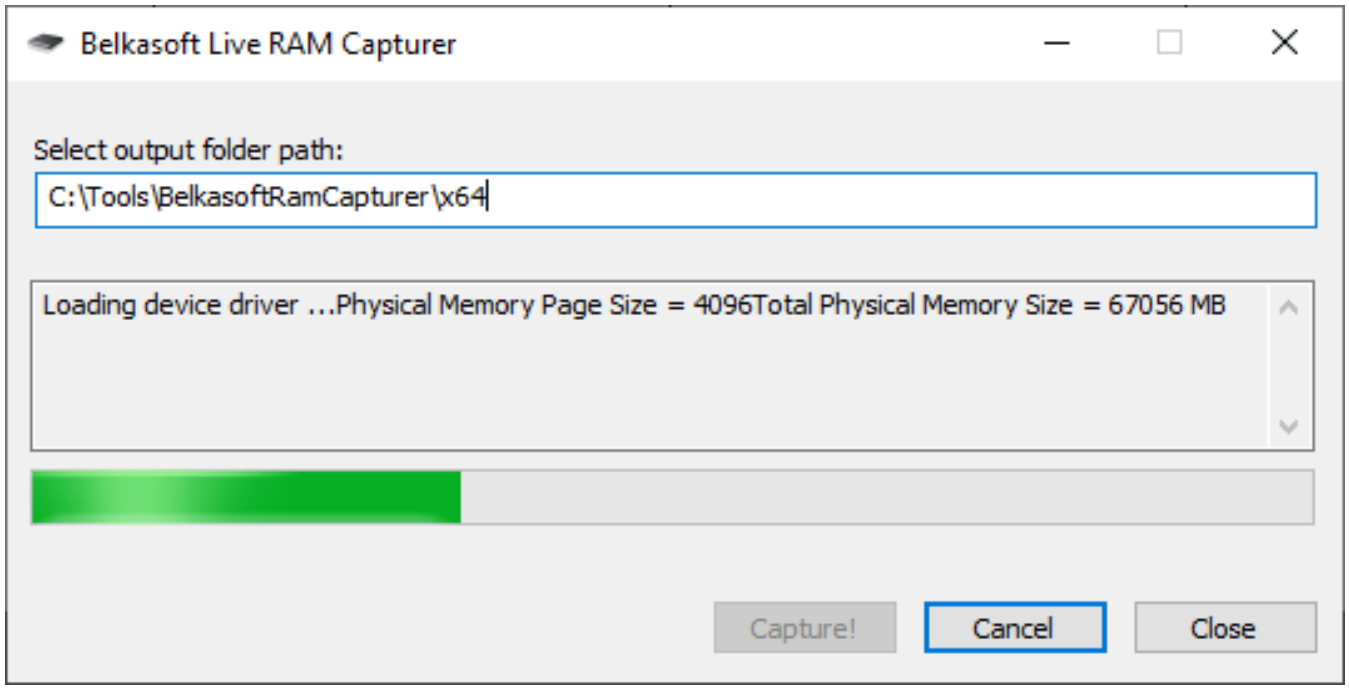
Belkasoft Ram Capturer[1] is a minimal GUI with only a few options, such as the output folder, but contains advanced features designed to bypass active anti-forensics techniques. This software, unlike FTK Imager, which we will take a look at next, uses a kernel mode driver to ensure accurate memory captures occur (other tools run in user mode).

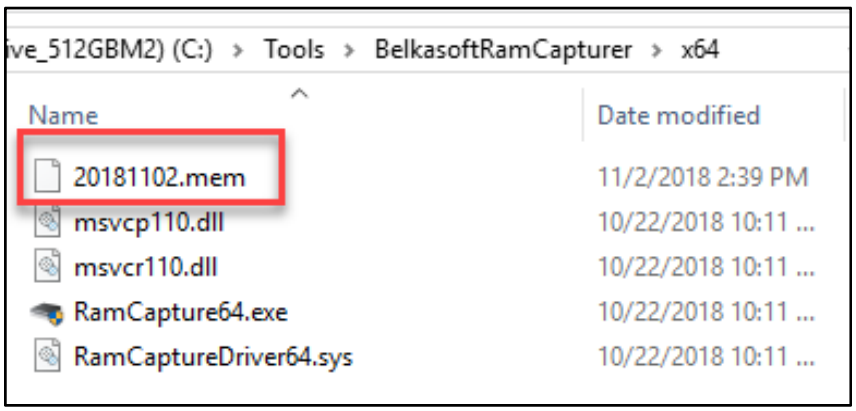
The tool comes in separate 32-bit and 64-bit versions to minimize the amount of code that is loaded prior to capturing memory.

Using the tool is very simple. Once the software is started, adjust the output folder path as necessary. By default, the software will dump memory to the same directory the executable was launched from. Once the **Capture!** button is clicked, memory will be dumped to a file named after the date the collection started.

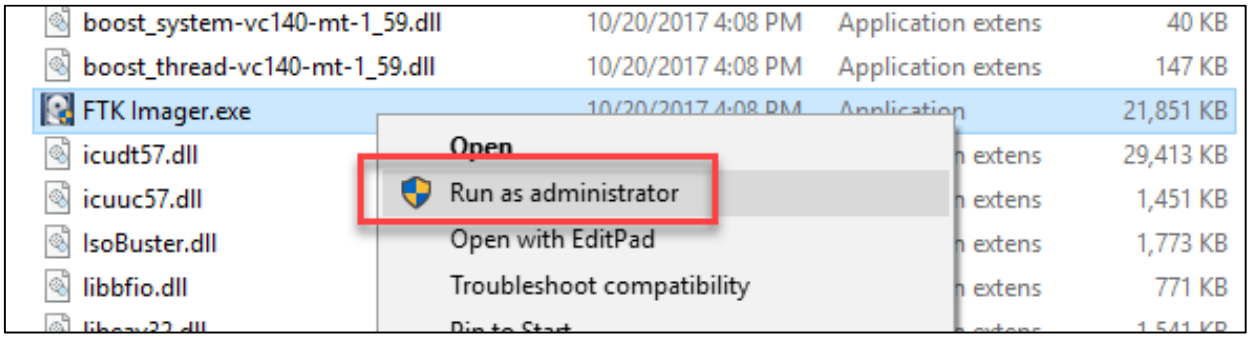
[1] Belkasoft RAM Capturer: Volatile Memory Acquisition Tool | `http://for498.com/oua4n`







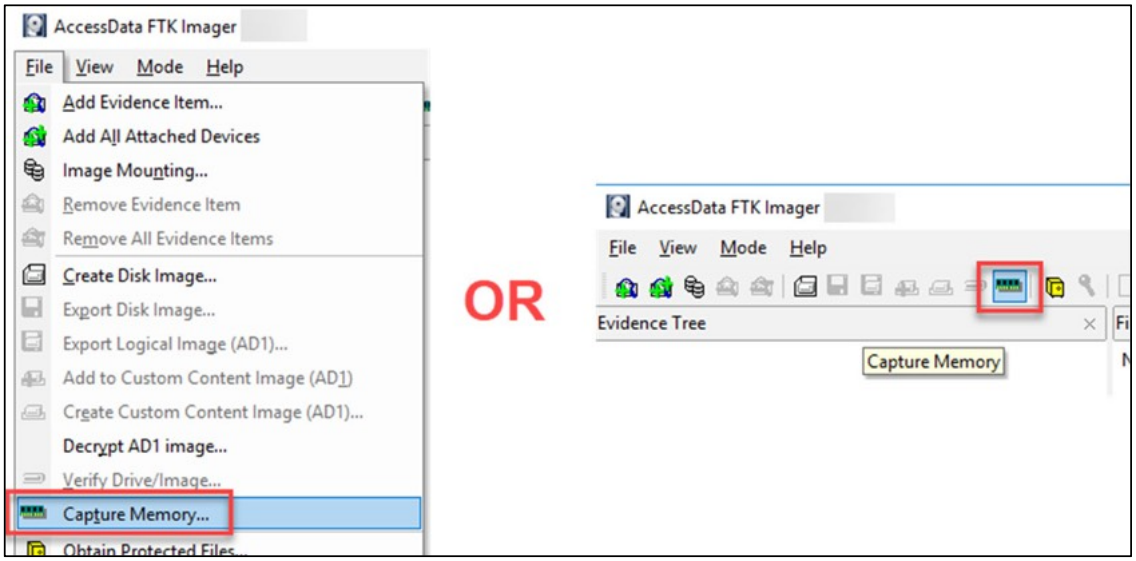
Windows GUI Tools: FTK Imager (1)



FTK Imager is another tool that can be used to collect memory, with some limitations over what we saw with the command line tools. The biggest difference between FTK Imager and other tools is one you cannot really see at all: the means by which memory is collected. As we saw previously, FTK Imager uses a usermode level driver to access RAM, whereas other tools use a kernel mode driver. While this isn't necessarily a deal breaker, you need to be aware of this distinction. If you have reason to believe that malware or other software on a computer may be interfering with memory collection, the best choice would be to use something other than FTK Imager to ensure all available memory is captured.

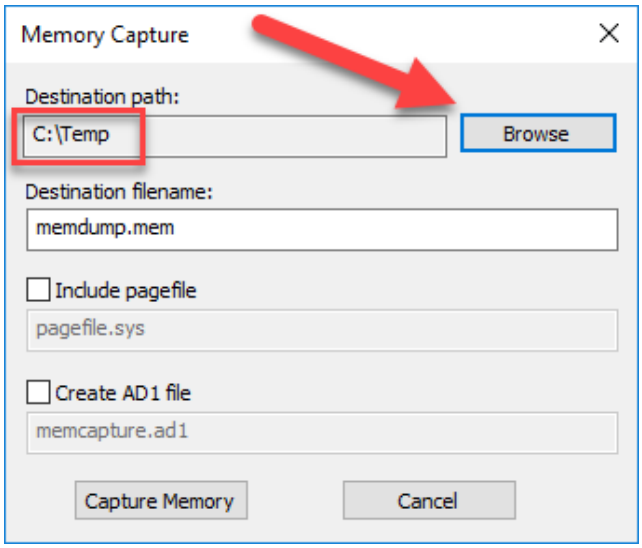
Using FTK Imager is easy. First, start FTK Imager as an administrator via right clicking on the main executable and selecting the appropriate option.

Windows GUI Tools: FTK Imager (2)



Once FTK Imager is started, select the Capture Memory option from the File menu, or use the button on the toolbar.

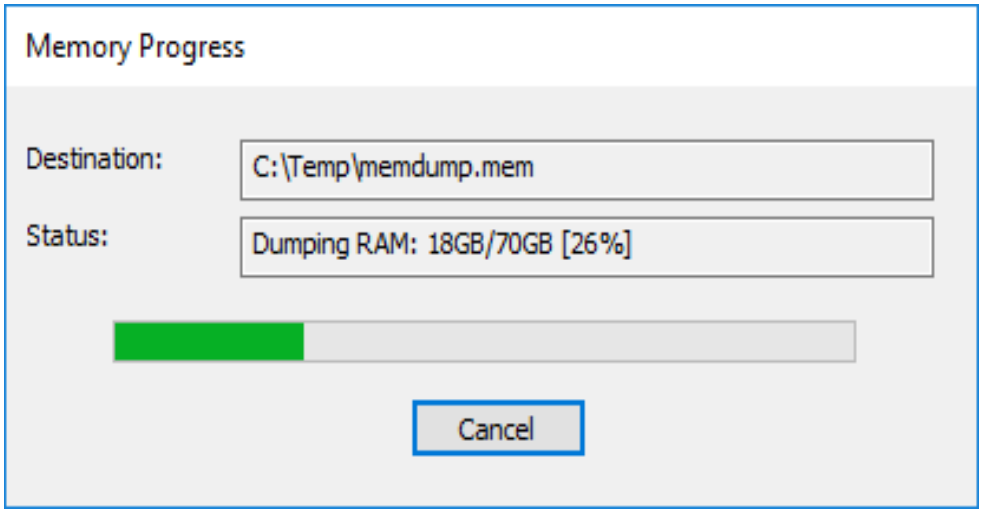
Windows GUI Tools: FTK Imager (3)



Do not check the pagefile or AD1 file boxes

Finally, use the Browse button to select the directory where you want to save your memory capture to.

Windows GUI Tools: FTK Imager (4)



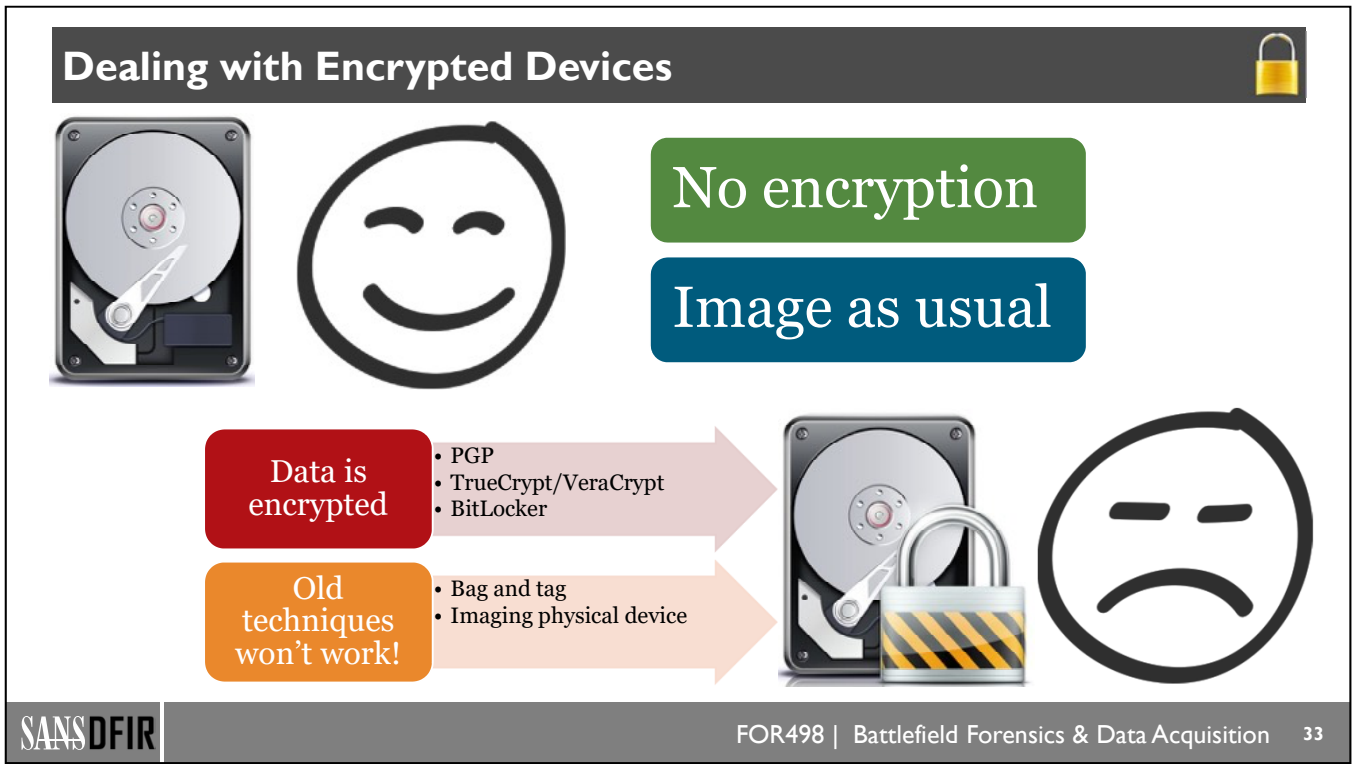
Once the memory collection has started, simply wait for it to finish. How long the collection will take depends on several factors, including the total amount of memory being collected as well as the speed of the device that the memory file is being written to.

Ideally, FTK Imager (and pretty much all other live response tools) would be run from writable external media, such as a hard drive or thumb drive. This external device also serves as a good place to save the memory image to. Whenever possible, avoid saving images of any kind to the host computer's file systems, as this overwrites large portions of storage that may contain recoverable data.

Once the memory collection is finished, the resulting file can be analyzed using a wide range of tools such as Volatility[1], and Rekall[2].

[1] Volatility | `https://for498.com/n-lib`

[2] Rekall | `https://for498.com/8zwb-`



Usually it is possible to access the data on a storage device regardless of the program that is used to image the drive. In most cases, forensic images will be in the E01 format, but you may also see images that are in raw format.

But what happens when someone introduces encryption to their system, such as PGP or VeraCrypt? These programs effectively make the data inaccessible unless the proper key is provided. When such solutions are in place, traditional techniques start to fail.

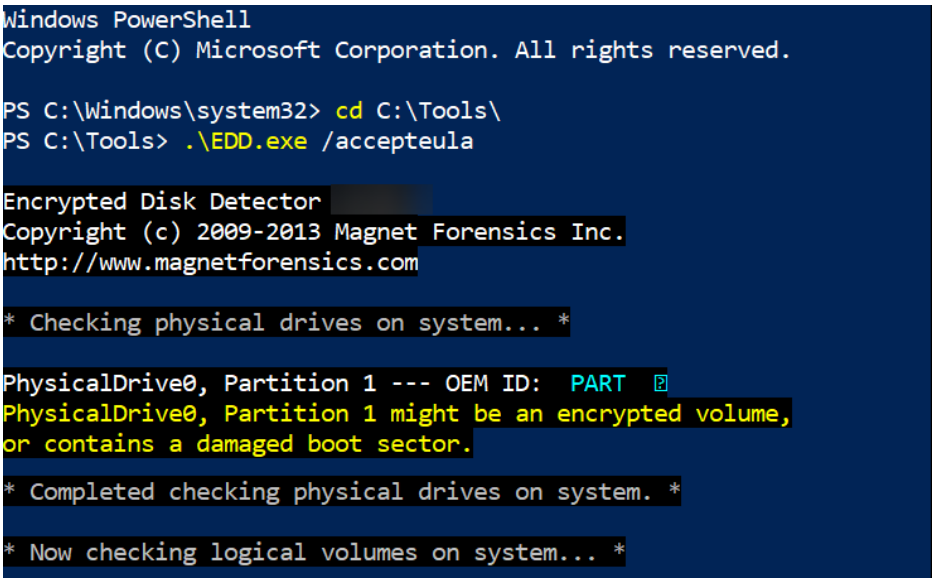
Since the data on the hard drive is encrypted, using traditional techniques such as imaging the physical volume connected to a write blocker would result in a forensic copy of the encrypted data being made which, of course, is not what we want. Traditionally, imaging takes place after a device is seized which generally implies the device has been powered off. But what happens to the data on a running computer when encryption software is in use? When power is lost, we would lose access to the data we previously had access to while the computer was running. Because of this, it is extremely important to be sure that no encryption software is in use before shutting a computer down (for whatever reason).

When encryption is detected, pivoting to alternate techniques becomes necessary, such as imaging logical devices while a computer is running, or performing triage data collection to ensure at least the most critical evidence is preserved from the computer.

To avoid falling into this trap, we want a way to know when this scenario would be necessary. Different triage tools can detect encryption and provide guidance on how to deal with it to ensure a proper collection happens.

One example of such a program is Magnet Forensics EDD tool, which we will look at next.

Locating Encryption with EDD (1)



Encrypted Disk Detector (EDD)[1] is a command-line tool that checks the local physical drives on a system for TrueCrypt, PGP®, BitLocker®, Safeboot, BestCrypt, Checkpoint, Sophos, or Symantec encrypted volumes. If no disk encryption signatures are found in the MBR, EDD displays the OEM ID and, where applicable, the Volume Label for partitions on that drive when checking for Bitlocker® volumes.

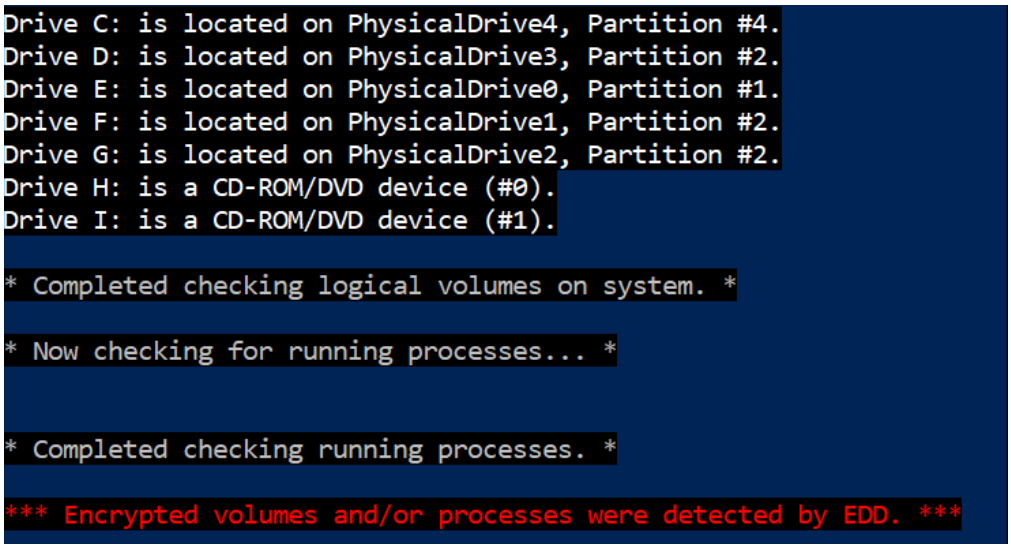
EDD does not attempt to locate encrypted volumes that are not mounted; its purpose is to alert the user of ***currently accessible*** drives/volumes that may be encrypted and therefore may be inaccessible if the system was shut down. Put in other words, EDD does not scan drives for files that might be encrypted containers. If this is what you're looking for, there are other software packages available elsewhere that attempt to do this.

EDD is useful during incident response to quickly and non-intrusively check for encrypted volumes on a computer system. The decision can then be made to investigate further and determine whether a live acquisition needs to be made in order to secure and preserve the evidence that would otherwise be lost if the plug was pulled.

The newest version of EDD has added a splash screen asking for you to accept the End User License Agreement (EULA). You can create a shortcut to the EDD.exe with the /accepteula switch to bypass this EULA, for example: ***edd.exe /accepteula***

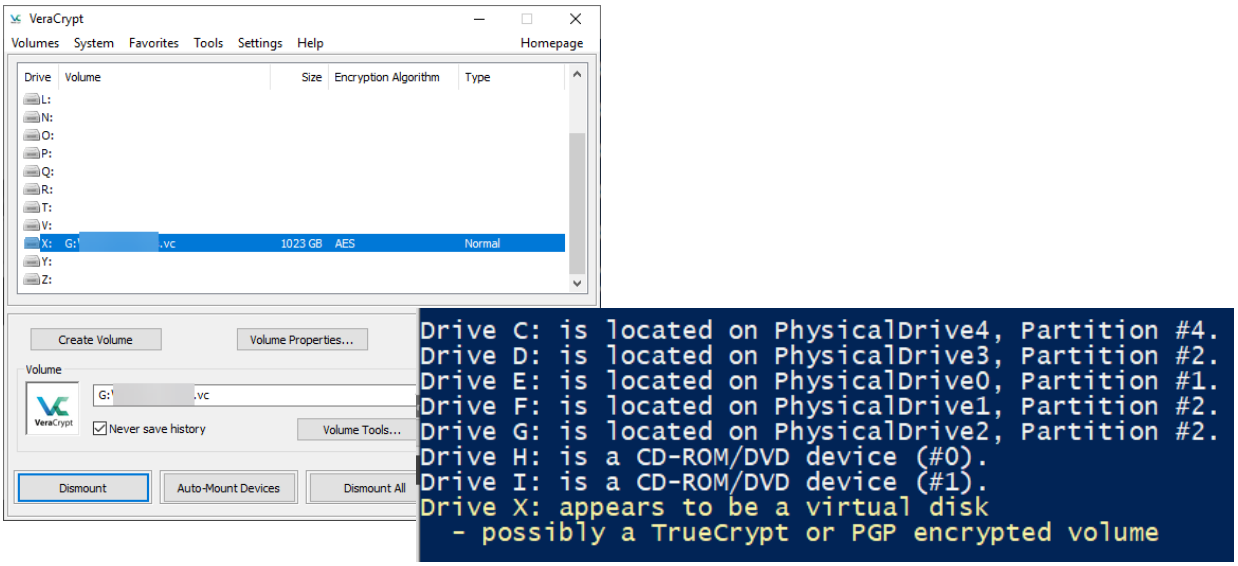
[1] Encrypted Disk Detector | `http://for498.com/vaxow`

Locating Encryption with EDD (2)



This page intentionally left blank.

VeraCrypt Detection



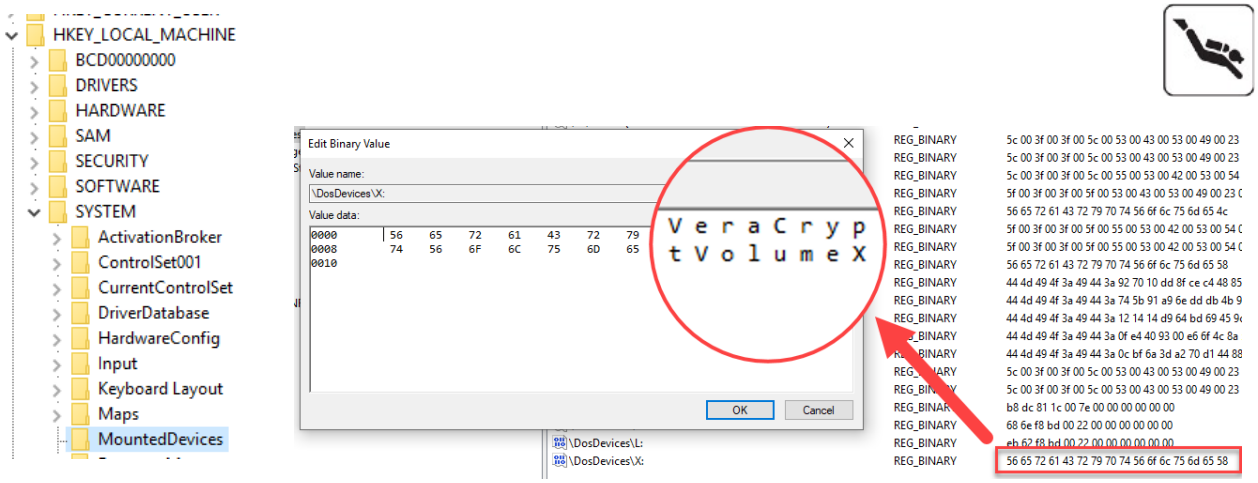
One of the types of encryption you are likely to run into is VeraCrypt[1], which is essentially what became of TrueCrypt after development on TrueCrypt was stopped. In many cases, the two tools behave the same and, for the most part, leave the same forensic artifacts on a computer. One main difference of course is the name itself, and as such, you will have to look for both TrueCrypt and VeraCrypt when looking for these products.

Both function the same in that they allow an end user to make an encrypted container that can be mounted as a drive letter. The tools offer many configuration options such as file system to use, capacity, the ability to have a hidden partition inside the primary container, and so on.

When either is used and a container is currently mounted, EDD locates and displays a message indicating this (It appears the description needs updating as it currently only says it could be a TrueCrypt volume).

[1] VeraCrypt - Free Open source disk encryption | `http://for498.com/4d07m`

VeraCrypt Registry Artifacts



Note: This Registry value is ONLY present when the container is mounted. The value is removed when the container is unmounted.

DEEP DIVE: Encryption detection via the Registry

VeraCrypt and TrueCrypt usage can also be seen in the Registry, specifically, in the SYSTEM hive's MountedDevices key.

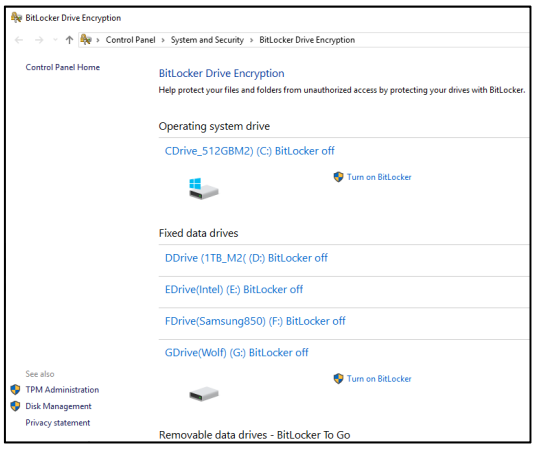
TrueCrypt and older versions of VeraCrypt artifacts could be seen in MountedDevices values, even after the container was unmounted. However, as of the latest version of VeraCrypt, when the container is unmounted in VeraCrypt, the value for the drive letter is DELETED. It is still possible of course to find remnants of this activity in the Registry by using tools such as Registry Explorer.

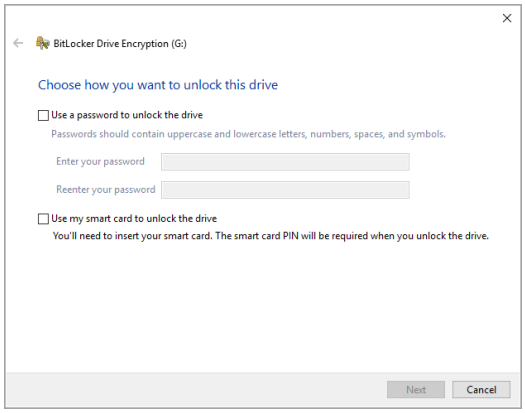
To be sure of what artifacts are in place on a given computer, determine the exact version in use on a machine, and then conduct your own tests using the same version of the software on a control machine.

PRO TIP: When looking for encrypted containers, it is often very easy to locate them by looking at a file system and locating the largest files on any given volume, removing such things as pagefile.sys, or hiberfil.sys. Since most users will want to store a not insignificant amount of data in an encrypted container, the container files tend to be among the largest files on the system.

Note that this may change at any time, should the VeraCrypt developers decide to clean things up a bit better. Then, depending on the version of VeraCrypt in use, this artifact may or may not be there.

BitLocker Introduction (I)





SANS DFIR FOR498 | Battlefield Forensics & Data Acquisition 38

BitLocker[1] is a data protection feature developed by Microsoft and is included in certain editions of Windows including Ultimate, Enterprise, or Pro [2]. It is most secure when the computer has a Trusted Platform Module (TPM), but it can work without a TPM being present. BitLocker uses the Advanced Encryption Standard (AES) with key lengths of either 128 or 256 bits.

Like other encryption programs, its primary job is to prevent data from being accessed by someone without the proper key, be it a passphrase, pin, or external USB key.

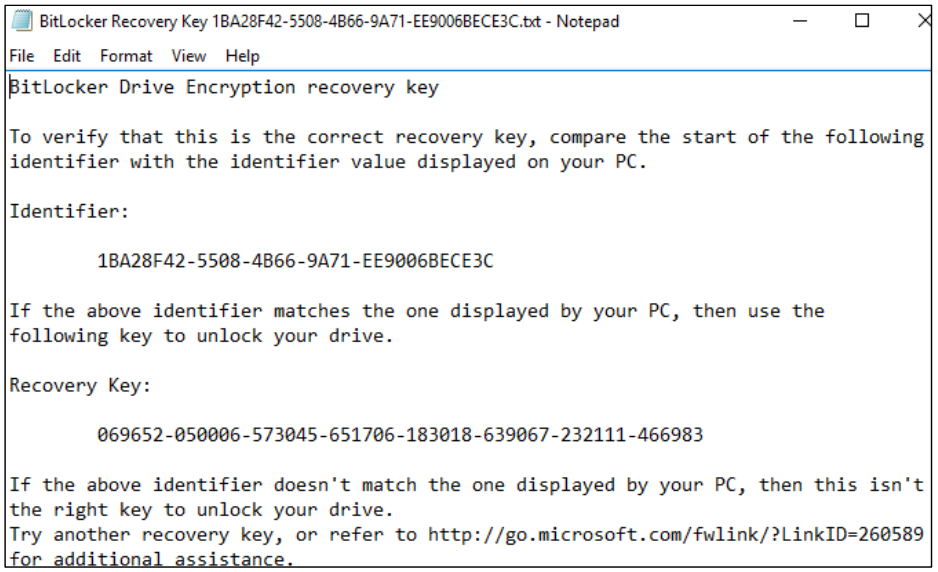
BitLocker is very easy to setup and use. Once in place, it essentially operates in a transparent fashion. A wizard walks the end user through the process. During this process, Windows will generate and save a BitLocker recovery key that has to be stored on a separate drive, or printed. This key can be used to gain access to the data should the primary method be unavailable. Once the wizard is finished, encryption begins in the background. Depending on the size of the drive being encrypted, this could take hours to complete.

However, if BitLocker is in use and the physical drive is removed from a computer and imaged, none of the data will be accessible without a recovery key. Another option that can be used with BitLocker (or any other encryption program) is to image the logical drive while the computer is running. This essentially bypasses the encryption since the encryption is unlocked while the computer is in use.

Because BitLocker is included with Windows, it can be deployed via group policy as well. In these cases network administrators can facilitate access to encrypted drives by providing the necessary recovery keys. Various policies exist to allow network administrators control over BitLocker properties[3].

[1] BitLocker (Windows 10) | `http://for498.com/7iasz`
[2] A beginner's guide to BitLocker, Windows' built-in encryption tool | `http://for498.com/vqy74`
[3] Overview of BitLocker Device Encryption in Windows 10 | `http://for498.com/udtcy`

BitLocker Introduction (2)



This page intentionally left blank.

Summary

- Discovery and documentation are critical
- We cannot keep responding in the ways that we used to
- Not collecting RAM when it is available is unacceptable, and considered to be destruction of evidence
- Introduce tools in a way that will make the least amount of changes
- Always consider level of volatility
- Shutting down a computer with whole disk encryption may render the storage completely inaccessible

This page intentionally left blank.

Exercise 3.1

RAM Acquisition & Encrypted Media

Synopsis: In this exercise, you will perform RAM acquisition, as well as check for any disk encryption that might be in use.

Average Time: 30 Minutes

This page intentionally left blank.

Exercise 3.1 Takeaway

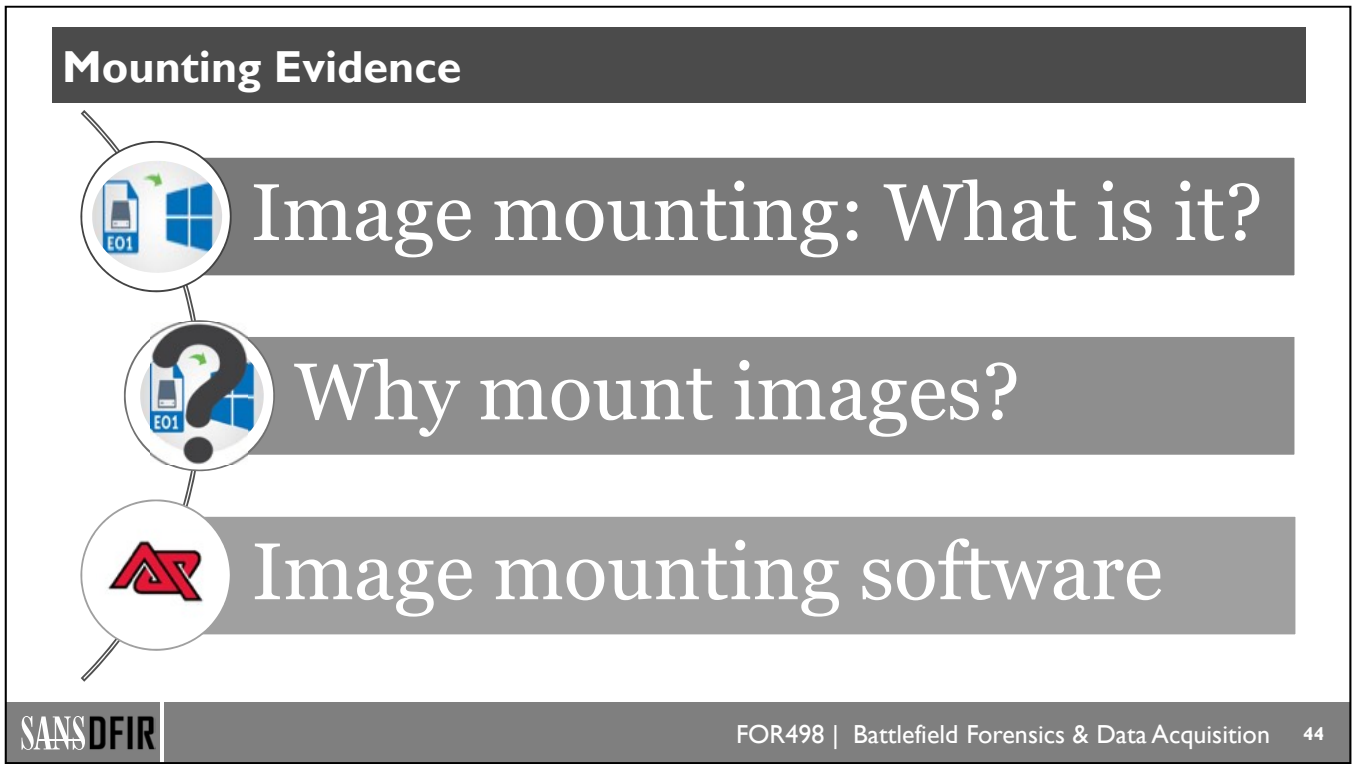
- Memory capture is an event that an incident responder will typically only get one opportunity to perform. Having multiple tools like FTK Imager and Dumpit provides several avenues for memory collection.
- Once memory is collected, due diligence should be done to check for disk encryption.
- When encryption is located, appropriate steps need to be taken to preserve the data in its current, unencrypted state before shutting down the computer.

This page intentionally left blank.

FOR498.3: Quick Win Forensics Agenda

- 3.1 Memory Acquisition & Encryption Checking
- 3.2 Mounting Evidence
- 3.3 Triage Acquisition
- 3.4 Host Based Live Acquisition
- 3.5 Dead Box Acquisition

This page intentionally left blank.



This page intentionally left blank.

Image Mounting: What Is It?

Access file systems in forensic images

- Raw/DD
- E01

Ensures read-only access

- Prevents changes to evidence
- Ensures the data you start with is what you end with

Access all parts of a forensic image

- Example: Volume Shadow Copies
- Requires physical disk emulation

Before we talk about why we want to mount a forensic image and the tools we will use to do so, it is helpful to understand what is going on when we mount a forensic image. When mounting an image you will typically be dealing with two different kinds of images: a full forensic image or a container image. We will talk more about these concepts on the next slide.

The concept of mounting an image is making what is inside the image file available to other tools. What this boils down to is that we want access to the file systems that are contained within the image files. In some cases, there are no non-proprietary file systems present, so special software must be used to mount these kinds of images.

Like everything we do in digital forensics, we always want to be as minimally intrusive as possible. In the case of mounting images, this means we want to mount an image in read-only mode to prevent the data contained in the image from being altered. Working against a read-only copy of the data minimizes risk associated with being accused of creating, modifying, or deleting data.

The last very important concept is that we want to be able to access all available parts of an image. Depending on how a tool is written, it may not present an image to the host computer as a physical disk. This limits the interaction that is possible because certain features (like volume shadow copies) will only be made available by Windows when it thinks it is looking at a physical disk. For this reason, when working with any version of Windows after Windows XP, it is critical that the software you use to mount images is capable of emulating a physical disk.

Image Mounting Characteristics

Forensic image

- Represents all data collected against a physical device or logical partition
- Examples: E01, DD/Raw

Container

- A subset of data that is copied into a proprietary format
- Examples: AD1, L01, Ctr, vhdx

We previously saw mention of two types of image files we may encounter: a full forensic image or a container image. But what is the difference between the two?

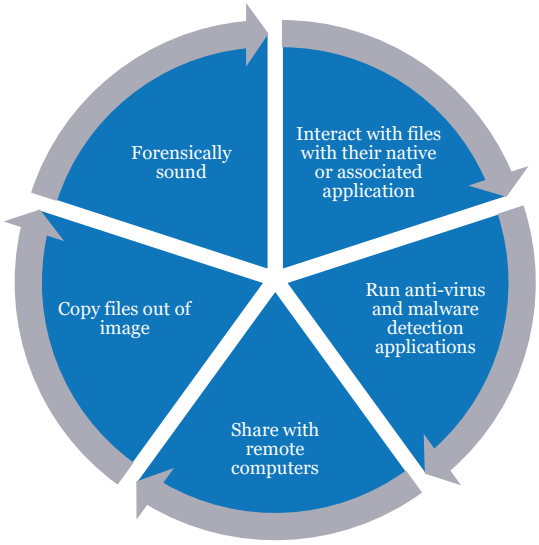
In the case of container files, most are proprietary in nature. This means you may need specific software to access the container once created. If this is the case, why use them in the first place? The answer is convenience, primarily. Using a container allows for copying only specific files into the image, which means less time taken to create the image and an image that is much smaller in size than a full forensic image.

In the case of AccessData's AD1 container or Encase's L01 container, they contain full file structures, including deleted files in some situations. This prevents them from being mounted physically. In addition, when you mount them logically, the drive or partition size will not be displayed correctly (because it does not have that information). When you create an E01, S01, or RAW/dd image of a properly working drive, the images contain all the appropriate drive data, disk, partition, and full file structure, which overcomes the issues we just mentioned about container files.

For some container image formats, third-party tools exist to access the data contained within them. For example, Mount Image Pro[1] can handle .E01, EX01, .L01, .LX01, .AD1, DD/RAW, .AFF, .MFS01, ProDiscover images, Safeback v2, .S01, and XWays .CTR formats!

[1] Mount Image Pro | `http://for498.com/d6u3r`

Why Mount Images?

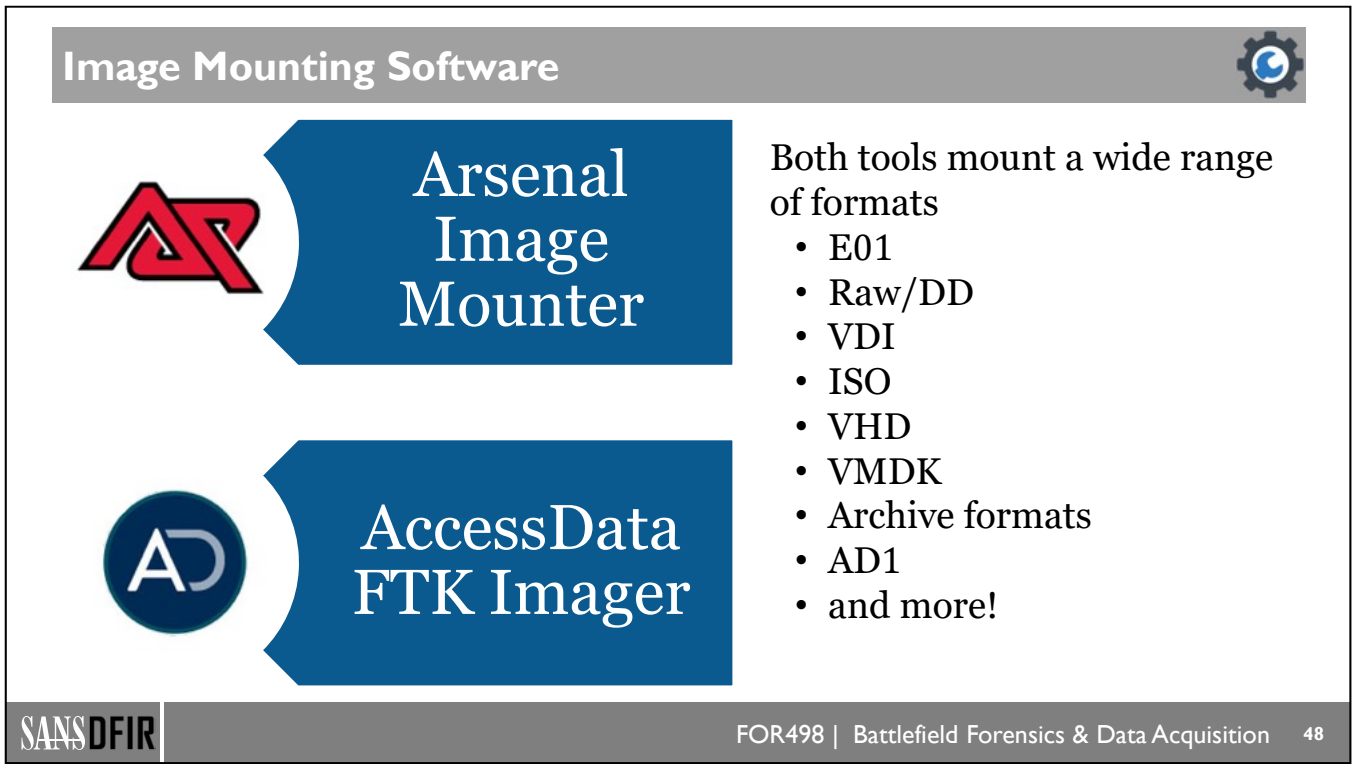


The primary benefit is being able to directly access the data in an image without being tied to any specific forensic tool.

Some of the many benefits of mounting forensic images are that examiners, or even investigators with no forensic training, can view and interact with the mounted files in their native or associated application installed locally on the review machine. This allows the reviewer to copy files out of the mounted filesystem. Because the image is mounted read-only, there are no worries that files can be copied into the mounted image or that the mounted image will be changed in any way.

A forensic image that is mounted is seen as another drive attached to the host system, which means antivirus and malware detection applications can run against the mounted filesystem. This could be a great first step to determining whether a virus or malware has infected the system.

In short, we like to mount forensic images because it allows us to access the data in any way we need to, and not be tied to any specific tool (short of the one we used to mount the image in the first place, of course!)

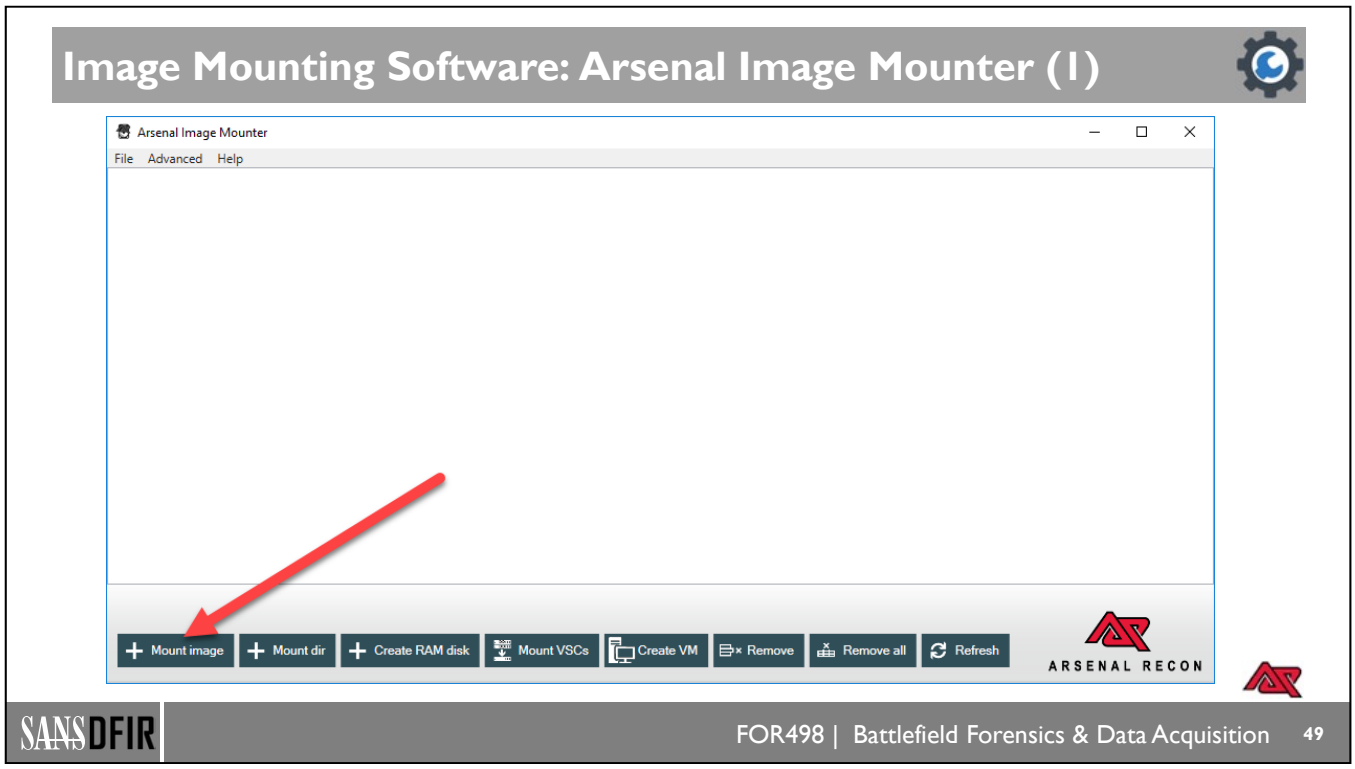


While there are other tools that can mount various image files, such as Mount Image Pro, we will be focusing on two tools. Both can mount a wide range of image and archive formats.

We recommend using Arsenal Image Mounter (AIM) as your primary tool due to its ability to emulate a physical disk (which means we get access to volume shadow copies!). Because it can mount a wide range of image formats, it fits into a standard process nicely.

As previously mentioned, if you run across a proprietary image format, you may have to use a different program to access the data contained within the image. An example of this would be an AD1 file. In these cases, using software from the company that created the AD1 specification is a good choice as it should provide the best compatibility with the image, especially if the image file was also created with software by the same company.

We will first take a look at how to use AIM to mount images and then follow up with how FTK Imager is used to mount an image.



From the Arsenal website: "Arsenal Image Mounter mounts the contents of disk images as complete disks in Microsoft Windows. Arsenal Image Mounter includes a virtual SCSI adapter (via a unique Storport miniport driver), which allows users to benefit from disk-specific features in Windows, like integration with Disk Manager, access to Volume Shadow Copies, Bitlocker, Bitlocker To Go, and more. As far as Windows is concerned, the contents of disk images mounted by Arsenal Image Mounter are 'real' SCSI disks. Arsenal Image Mounter is the first and only open source solution for mounting the contents of disk images as complete disks in Windows."

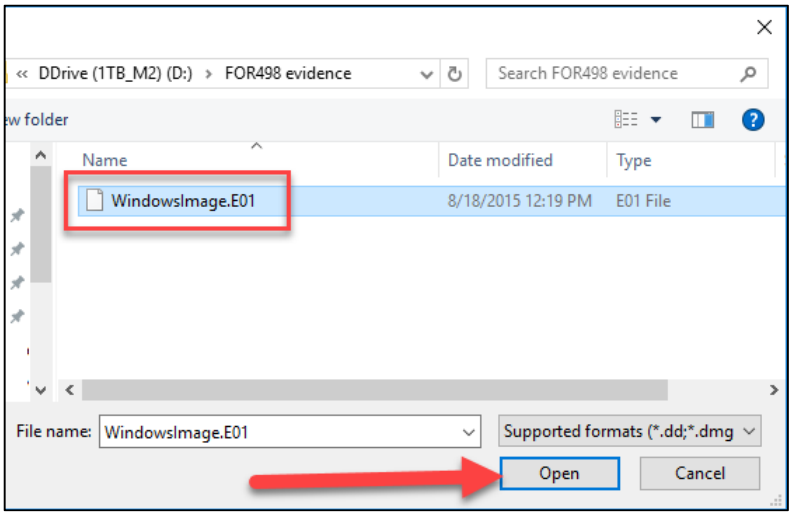
"Many image-mounting solutions mount the contents of disk images in Windows as shares or partitions (rather than complete disks), which limit their usefulness. While developing the digital forensics tool Registry Recon, Arsenal found existing image-mounting solutions lacking and built Arsenal Image Mounter. Ongoing development of Arsenal Image Mounter has broader goals that benefit the entire information technology community in addition to digital forensics and incident response practitioners." [1]

Using Arsenal Image Mounter (AIM) is easy. After starting the program, click the Mount image button and navigate to the directory where your forensic image is located. Selecting an image and pressing Open takes you to the Mount options dialog. You will almost always choose the Write temporary option. Finally, click OK to complete the process.

Reference

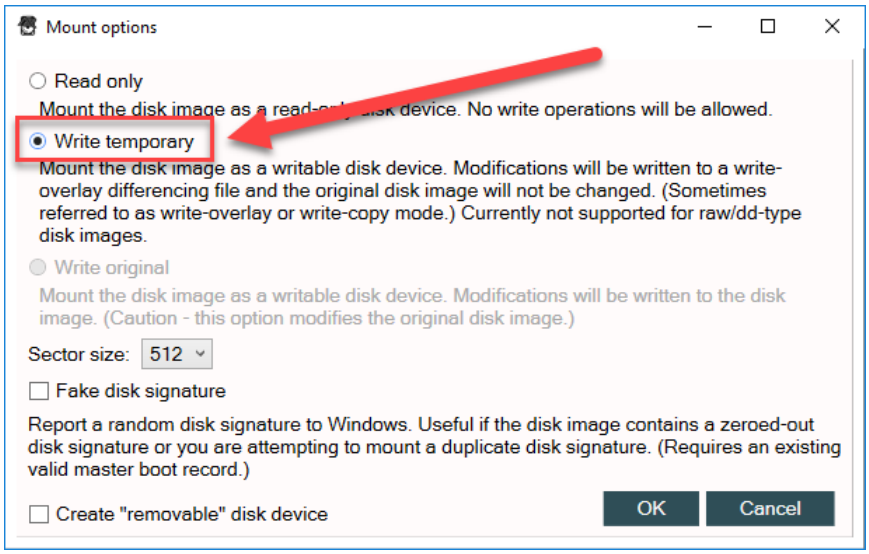
[1] Arsenal Recon products | `http://for498.com/08fty`

Image Mounting Software: Arsenal Image Mounter (2)



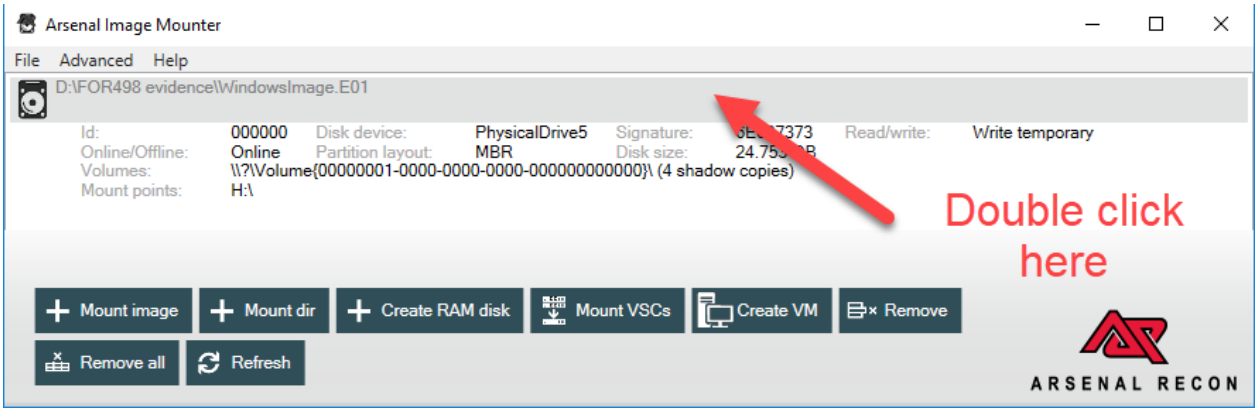
This page intentionally left blank.

Image Mounting Software: Arsenal Image Mounter (3)



This page intentionally left blank.

Image Mounting Software: Arsenal Image Mounter (4)



Detailed information, including mount points and volume shadow copy information, is available for each mounted image.

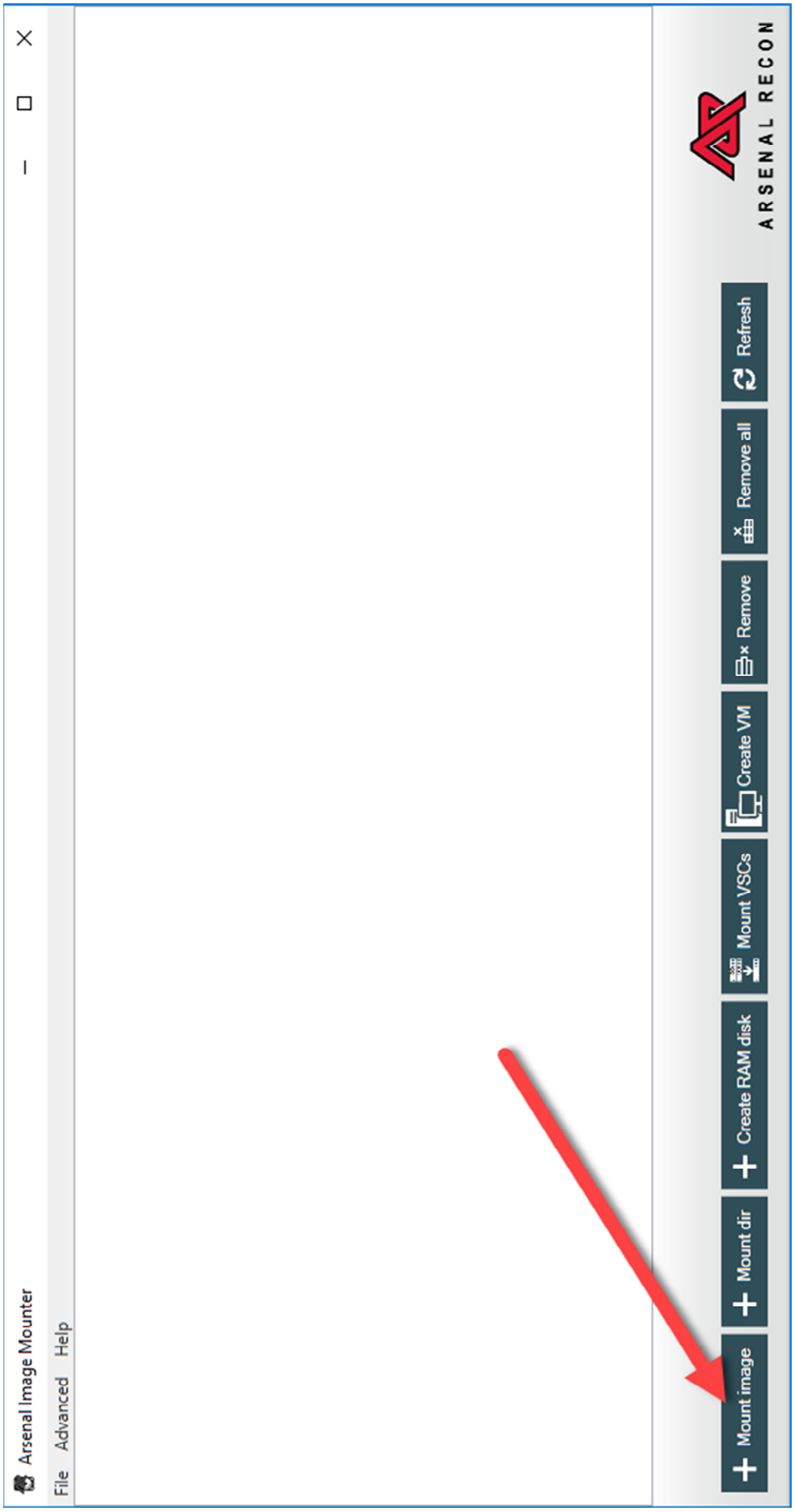
After an image is mounted, it is displayed in the interface. AIM keeps track of a lot of different information about each mounted image, including the device name, size, the number of volume shadow copies (VSC) available, and drive letter where the image is mounted. To see this information, double click anywhere on the grey bar where the full path to the image is displayed. This will expand out the details for that image.

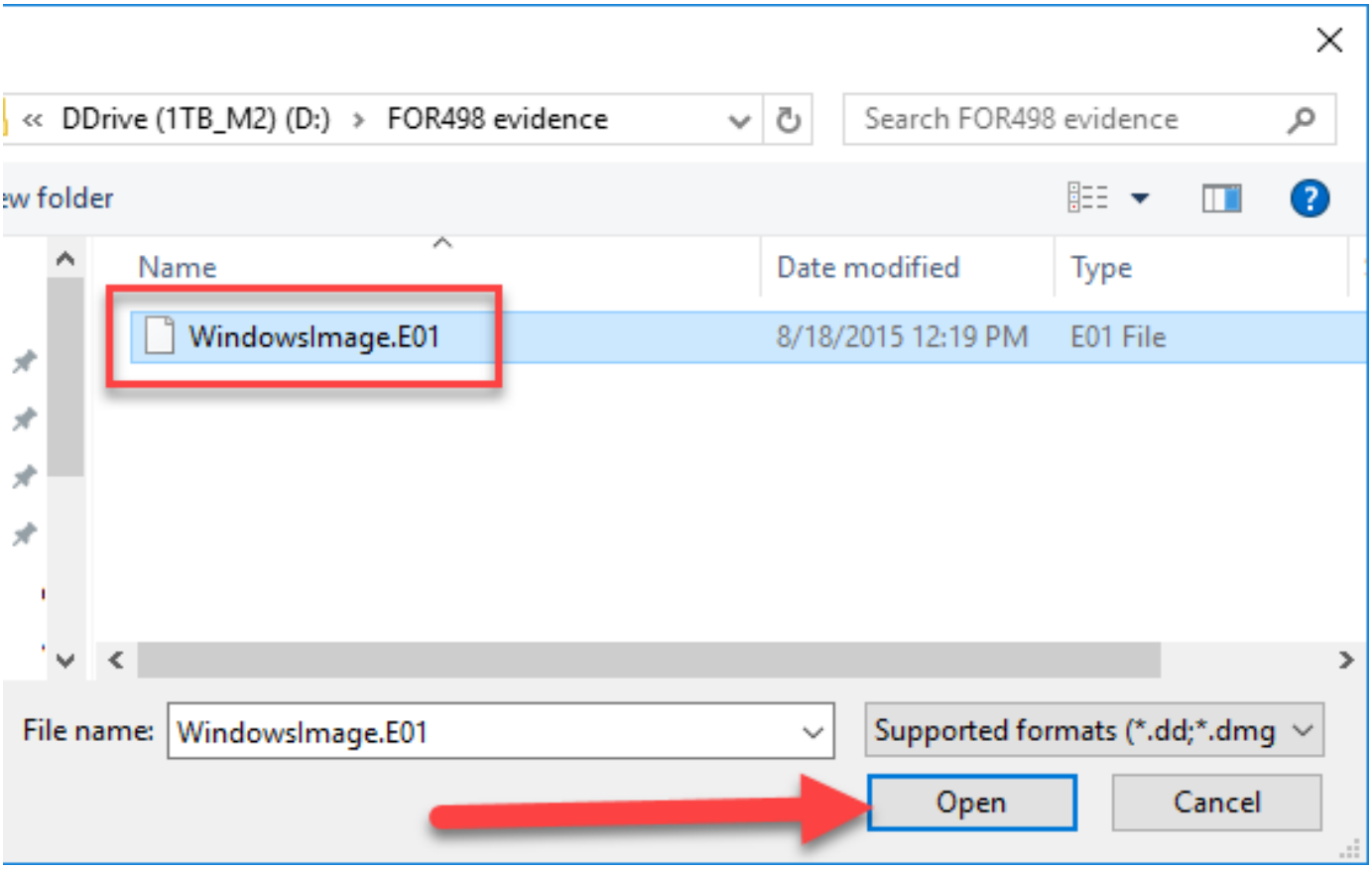
To unmount an image, select it from the list and click Remove. To remove all images, use the Remove All button. AIM also has additional features such as mounting VSCs and creating a virtual machine from the image, but these features are only available in the paid version and require a specific host machine configuration.

At this point you are free to interact with the data in any way you choose, from browsing with File Explorer, to pointing forensic tools at the mounted drive letter (H:\ in the example above).

AIM makes it easy to go from a forensic image to having access to the file system's data, and because it emulates a physical disk, this includes any volume shadow copies that may be on the drive.

It should be noted that you will only see a drive letter for an image if Windows understands the file system the image contains. In other words, if you mount a forensic image of an Ext4 file system, Windows will see the fact that there is a new physical disk present, but it will not mount the file system to a drive letter (because it has no idea, without a third party driver, what to do with the Ext4 partition or partitions in the image.) With that said, other forensic tools that do understand Ext4 would be able to access the physical disk via the device name/number (PhysicalDrive5 in the image above).





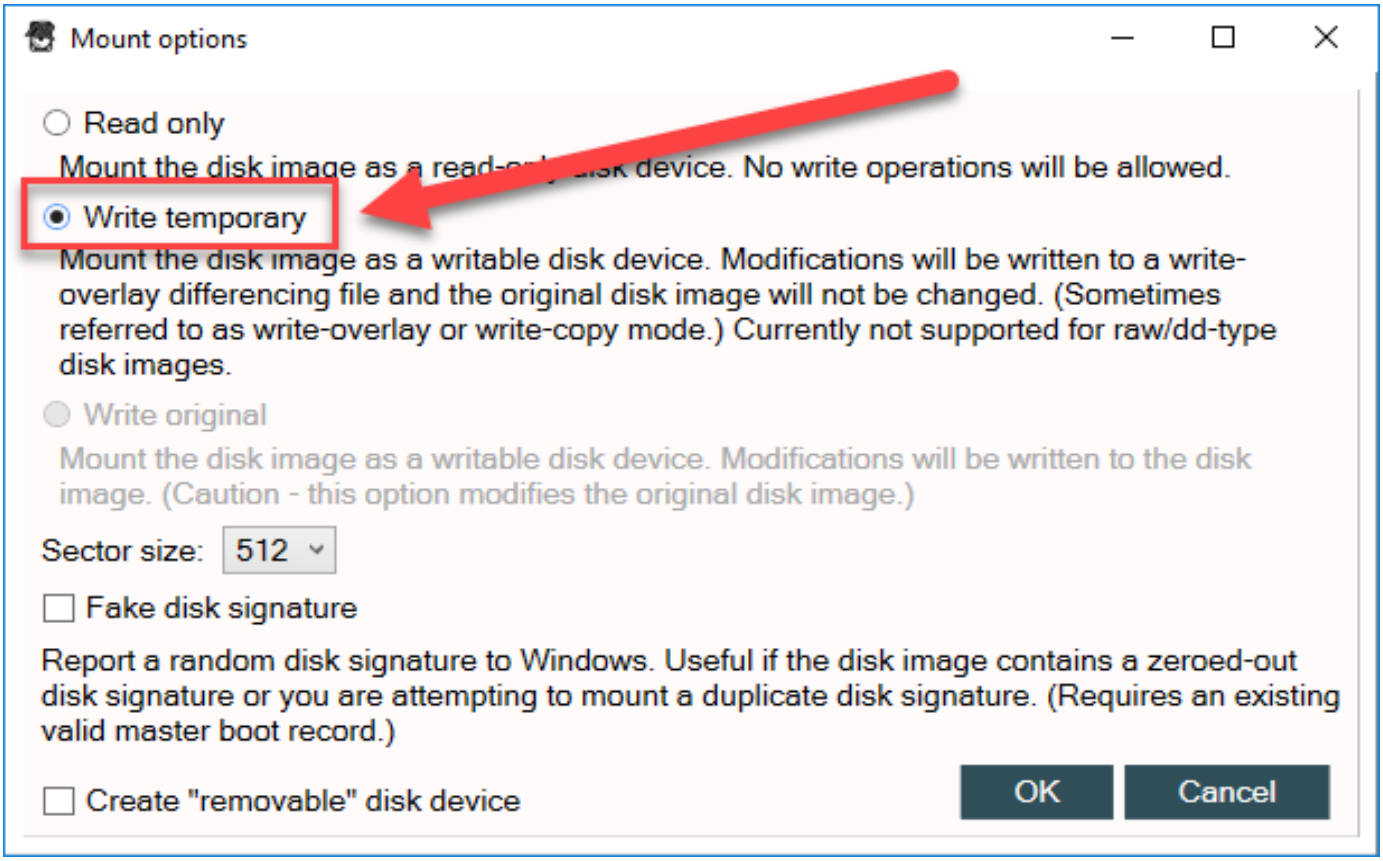
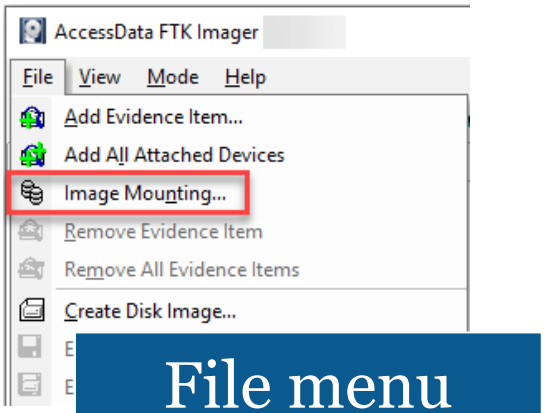
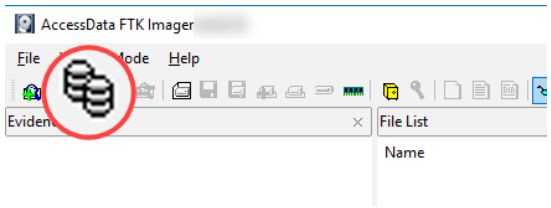


Image Mounting Software: FTK Imager (1)

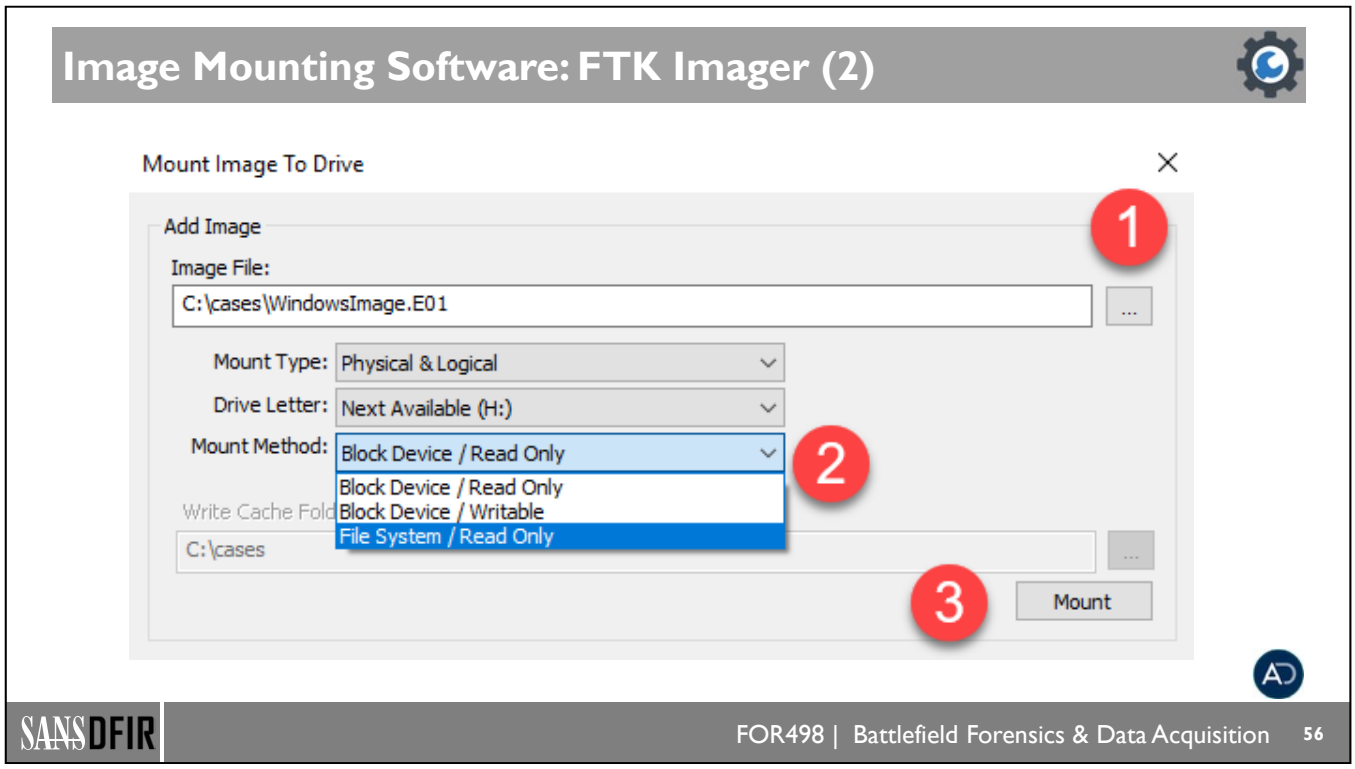


File menu



Icon on toolbar

With FTK Imager open, there are two ways to mount an image: the File menu, or the icon on the toolbar. Clicking File -> Image Mounting or the icon on the toolbar begins the process of mounting an image.



In the Mount Image To Drive dialog, we first have to select the image we want to mount. To the far right of the Image File text box is a button with three dots on it. Clicking this brings up a file selection dialog. Navigate to the forensic image to mount, then click Open.

With the image selected, we can look at the three options in the middle of the dialog. Mount Type contains three different map types: Physical & Logical, Physical Only, and Logical Only. Physical & Logical is generally the option you want (and the default). The Drive Letter automatically defaults to the next available drive letter, but you can optionally select the drive letter to assign the mount point manually. Finally, depending on the kind of image you have (we are working with an E01 above), select the appropriate Mount Method (File System /Read Only in our case).

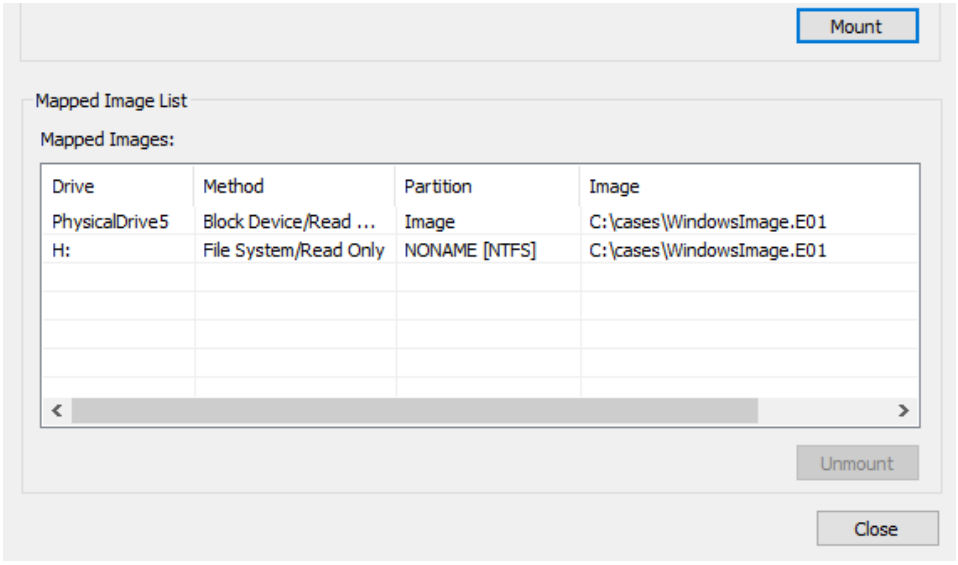
The options (from the FTK Imager User Guide) for Mount Method work as follows:

Block Device / Read Only: Treats the mounted image as a block device (disk). Image will be subject to NTFS permissions and Windows file/folder protections.
Block Device / Writable: Mounts image as a writable device, saving any changes in a cache file (no changes made to original image).
Filesystem / Read Only: Creates a virtualized folder structure, circumventing Windows file/folder protections. Shows deleted files. Filesystem starts in [root] folder.

Notice that the middle option mentions "writable" but note the description here. While it will seem like changes are being made to the underlying image, these changes just LOOK like they took place. If you were to unmount and then remount the same forensic image after "altering" it using this option, you would see that nothing in fact has changed.

With these properties set, click Mount and the chosen image file will be mounted. You can see details of the mounted images in the Mounted Image List at the bottom half of the "Mount Image to Drive" area.

Image Mounting Software: FTK Imager (3)

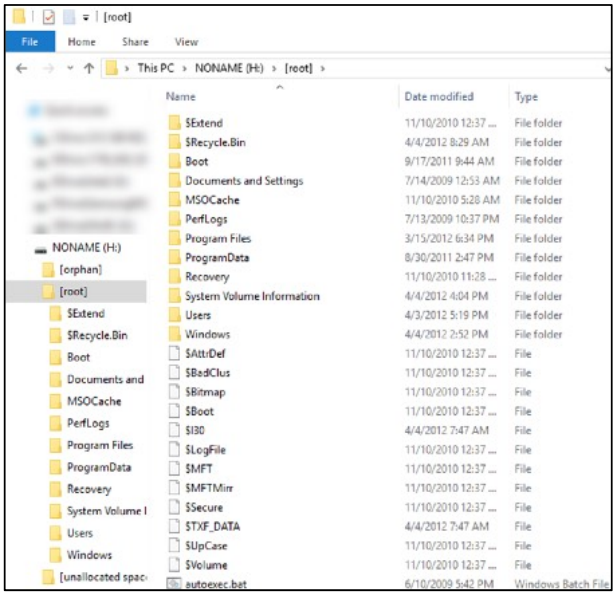


After clicking the Mount button, the Mapped Image List will be populated with the details of the newly mounted image. In the example above, notice that we have both the physical disk (as PhysicalDrive5) available, as well as the H: drive, which contains an NTFS partition.

To unmount images, select an entry in the Mapped Images list under the Mapped Image List section, then click Unmount.

WARNING: Mapped images will be automatically unmounted if FTK Imager is closed!

Exploring Mounted Images: FTK (1)



Here we see what a mounted image looks like using FTK Imager. FTK Imager assigned drive letter H:\ to this image.

Right away we can see some differences with how the data is presented. There are several folders at the root of the disk that do not really exist such as **[orphan]** and **[root]**. The **[root]** folder contains the usual directory structures we are used to seeing with a Windows image.

In the case of FTK Imager, notice also that we can see special files, like $MFT, and special directories like $Extend.

This can be useful if you want to copy out $MFT using File Explorer.

Exploring Mounted Images: FTK(2)

```
PS C:\Windows\system32> vssadmin list shadows /for=h:
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

No items found that satisfy the query.
```

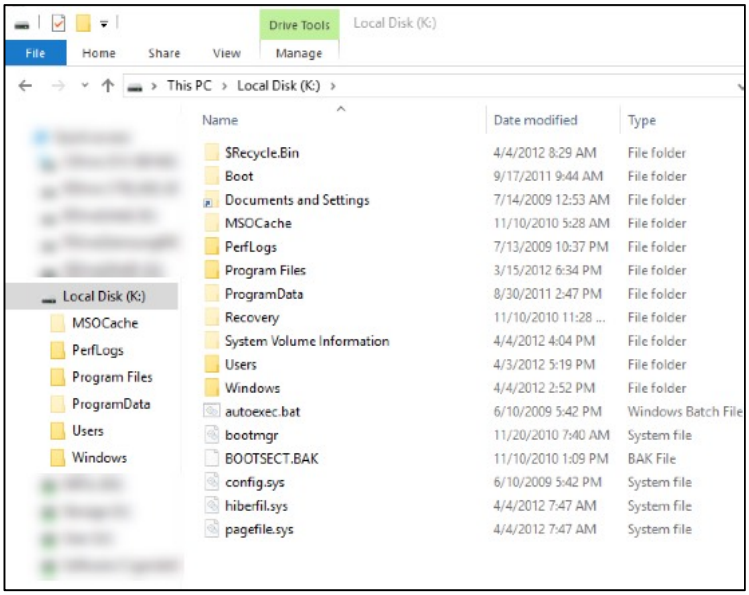

But what about Volume Shadow Copies? The command…

`vssadmin list shadows /for=h:`

…confirms the fact that we do not have access to VSCs when using FTK Imager to mount images.

We will get into the particulars of vssadmin and other Volume Shadow Copy information in a future section, but for how, understand that vssadmin is the command included in Windows to interact with VSCs.

Exploring Mounted Images: Arsenal (1)

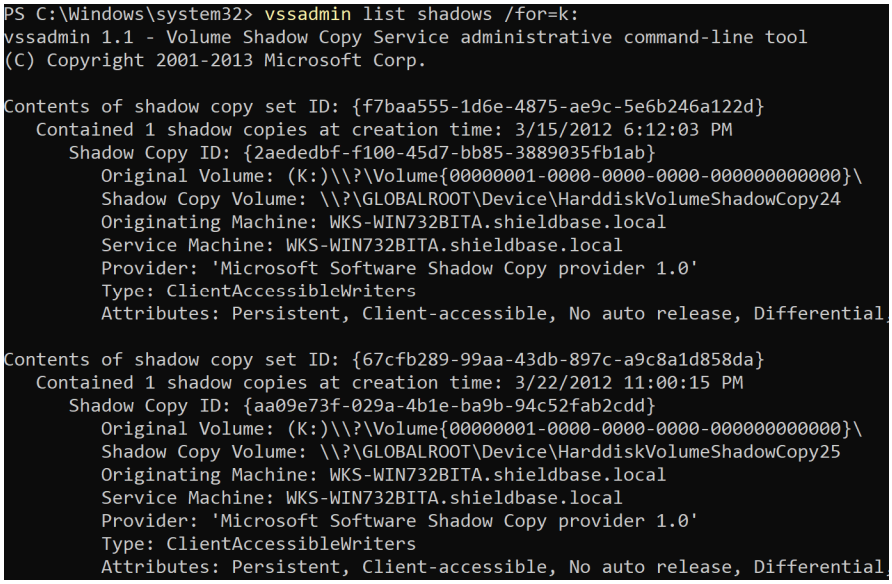


Here we see the same forensic image mounted in Arsenal Image Mounter (AIM) that was assigned drive letter K:\.

Notice that it looks like any other drive you may have looked at, in that you see folders such as Windows, Program Files, etc.

What you do not see when using AIM are the special files, such as $MFT or special directories, such as $Extend. It is not that these files are not there and available via forensic tools, but since AIM emulates a physical disk, Windows is hiding the special files.

Exploring Mounted Images: Arsenal (2)



```
PS C:\Windows\system32> vssadmin list shadows /for=k:
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

Contents of shadow copy set ID: {f7baa555-1d6e-4875-ae9c-5e6b246a122d}
   Contained 1 shadow copies at creation time: 3/15/2012 6:12:03 PM
      Shadow Copy ID: {2aededbf-f100-45d7-bb85-3889035fb1ab}
         Original Volume: (K:)\\?\Volume{00000001-0000-0000-0000-000000000000}\
         Shadow Copy Volume: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy24
         Originating Machine: WKS-WIN732BITA.shieldbase.local
         Service Machine: WKS-WIN732BITA.shieldbase.local
         Provider: 'Microsoft Software Shadow Copy provider 1.0'
         Type: ClientAccessibleWriters
         Attributes: Persistent, Client-accessible, No auto release, Differential

Contents of shadow copy set ID: {67cfb289-99aa-43db-897c-a9c8a1d858da}
   Contained 1 shadow copies at creation time: 3/22/2012 11:00:15 PM
      Shadow Copy ID: {aa09e73f-029a-4b1e-ba9b-94c52fab2cdd}
         Original Volume: (K:)\\?\Volume{00000001-0000-0000-0000-000000000000}\
         Shadow Copy Volume: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy25
         Originating Machine: WKS-WIN732BITA.shieldbase.local
         Service Machine: WKS-WIN732BITA.shieldbase.local
         Provider: 'Microsoft Software Shadow Copy provider 1.0'
         Type: ClientAccessibleWriters
         Attributes: Persistent, Client-accessible, No auto release, Differential
```

What you gain by using AIM, however, is access to any Volume Shadow Copies (VSC) that exist. The command…

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

…shows us all available shadow copies for the K:\ drive, which is where AIM mounted the image file.

Summary

- Mounting a forensic image allows for a great deal of investigative functionality
- There are a number of tools that can be used for mounting; however the examiner must understand the advantages and limitations of each
- FTK Imager does not allow for the accessing of Volume Shadow Copies

This page intentionally left blank.

Exercise 3.2

Mounting Evidence

Synopsis: In this exercise, you will mount an evidence file and extract quick win data from it. You will then process prefetch data, and analyze it using Timeline Explorer.

Average Time: 35 Minutes

This page intentionally left blank.

Exercise 3.2 Takeaway

- Mounting evidence files in a forensically sound manner allows for copying out forensic artifacts for analysis.
- Manual collection of key forensic artifacts can be time consuming and, for more than a handful of files, can also be error prone.
- For specific artifacts, manually locating and copying files out of an image might be the fastest way to get to the data you need.
- Timeline Explorer can help make sense of a wide range of forensic data.

This page intentionally left blank.

FOR498.3: Quick Win Forensics Agenda

- 3.1 Memory Acquisition & Encryption Checking
- 3.2 Mounting Evidence
- 3.3 Triage Acquisition
- 3.4 Host Based Live Acquisition
- 3.5 Dead Box Acquisition

This page intentionally left blank.

Triage Acquisition

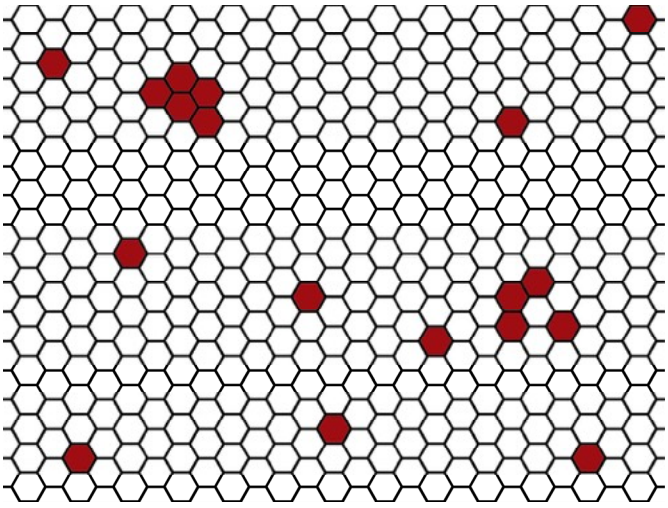
Triage Introduction

Triage Acquisition Techniques

Triaging Original Media vs. Image File

This page intentionally left blank.

Triage Introduction



Find the signal among the noise

Given that 99% of the necessary evidence typically will exist in 1% of the data acquired when creating a full disk image, it is easy to see how a great deal of time is wasted following the normal procedures in today's digital forensics world. Instead, let's focus on this 1%. By doing so, we can do a very rapid triage collection that we can use to start our investigation sooner!

Furthermore, as hard drive size keeps expanding, it becomes more and more impractical to take a full disk image of every device encountered, which means having to wait for hours before it is even possible to start looking for answers. By triaging devices before full disk images, a significant amount of time can be saved.

Before we look at some triage techniques, let's take a moment to discuss what exactly we will be collecting during the triage process. Because the whole point is to not collect everything, we need to understand what it is that we want to collect (the 1% amidst the rest of the data), as well as understand the value of the data being collected.

What Content to Collect When Triaging?

Artifact	
Registry Hives and Backups	• FOR500
LNK Files	• FOR500
Jump Lists	• FOR500
Prefetch	• FOR500
Event Logs and PnP Logs	• FOR500
Browser Data (IE, Firefox, Chrome)	• FOR500
Recycle Bin	• FOR500
Master File Table	• FOR500/508
NTFS Log Files and Journal Log	• FOR508
Pagefile and Hibernation Files	• FOR508

Given the hundreds of thousands of files and directories on a typical hard drive, how do you know what to grab?

This is obviously going to be driven by the facts of your investigation but for nearly any triage collection, we would want to collect the following listed files/artifacts:

- **$MFT:** The master file table, which contains information about every file and folder on the system.
- **$Logfile and $USN $J (journal):** These files record $MFT and file change activity (file open, close, creation, deletion).
- **All registry hives and perhaps backup registry hives:**
  - SAM
  - SYSTEM
  - SOFTWARE
  - DEFAULT
  - NTUSER.DAT
  - USRCLASS.DAT
  - AMCACHE.HVE
- ***.evtx:** These are the event logs, located in the %WinDir%\System32\winevt\Logs folder.
- **Other log files:** These include setupapi.dev.log (plug and play logfile), firewall logs, IIS logs, and so on.
- ***.lnk files:** These are shortcut files which tell us about files and folders that were opened.
- ***.pf:** These are prefetch files, which is an evidence of execution artifact.
- **The RECENT Folder and subfolders:** These include jump lists (Win Vista/7/8/10).
- **The User's Home Folder of "APPDATA:"** This should be extracted as it contains the cache, history, cookies files, and more.

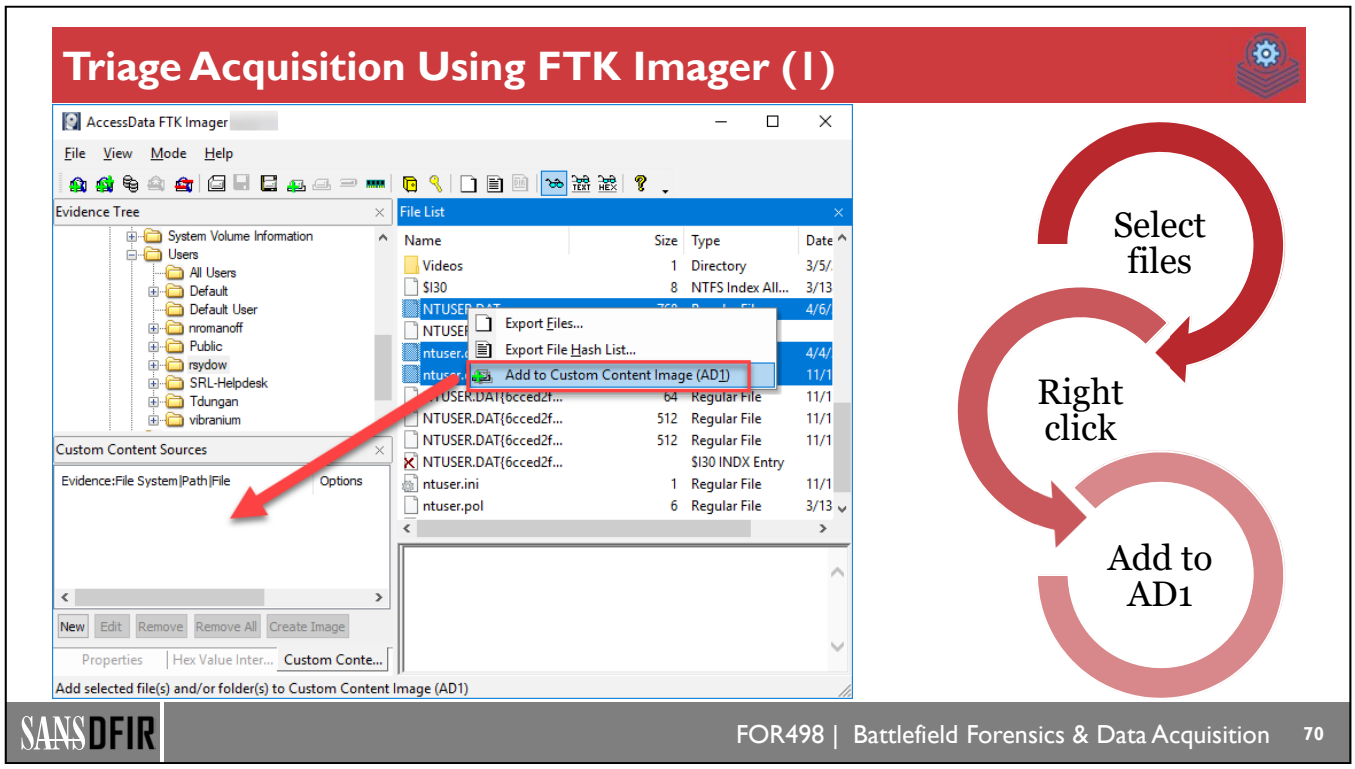
Optional, but useful artifacts:

- **Pagefile.sys:** The Windows pagefile (an extension of RAM).
- **Hiberfil.sys:** The Windows hibernation file is a compressed image of RAM the last time the system was placed into hibernation.

Tools used for triage collection will usually allow for the use of wildcards, which makes looking for files by extension much easier. For example, using FTK Imager's Custom Content Image feature (we will show you how to do this next) allows for finding files by extension, regardless of where they are on the drive.

The data above is where you will typically spend approximately 80% or more of your time doing analysis, so the faster you can get these kinds of files, the faster you will find the answers you are looking for.

Between FOR500 and FOR508, we have a lot of artifacts to cover. We list which course will cover, in depth, the examination of that specific artifact for processing. This way, it makes sense why we are obtaining it.



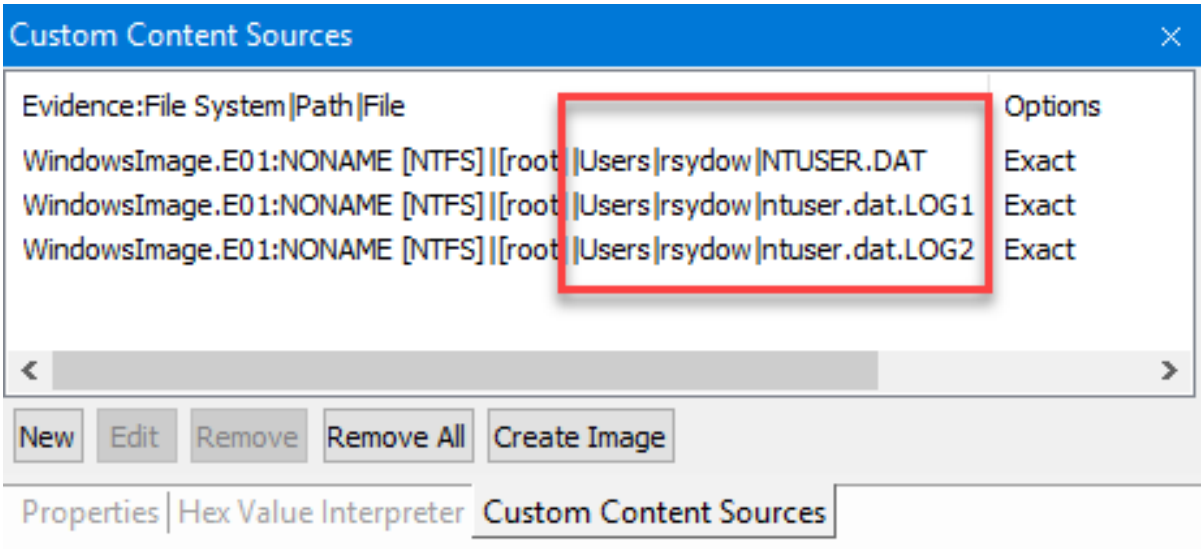
Custom Content Image is a feature of FTK Imager that allows a user to selectively choose a subset of data available in a forensic image or on a running computer, and place the data into a "container" that can be accessed later using different tools. To get started making a custom content image, evidence must be loaded into FTK imager so the files you want to add to a container are visible.

In the lower left corner of FTK Imager is a multi-tabbed window. One of the tabs is labeled **Custom Content Sources**. You can click on the **Custom Content Sources** tab, or it may be automatically selected when you add your first item for inclusion in the custom content image.

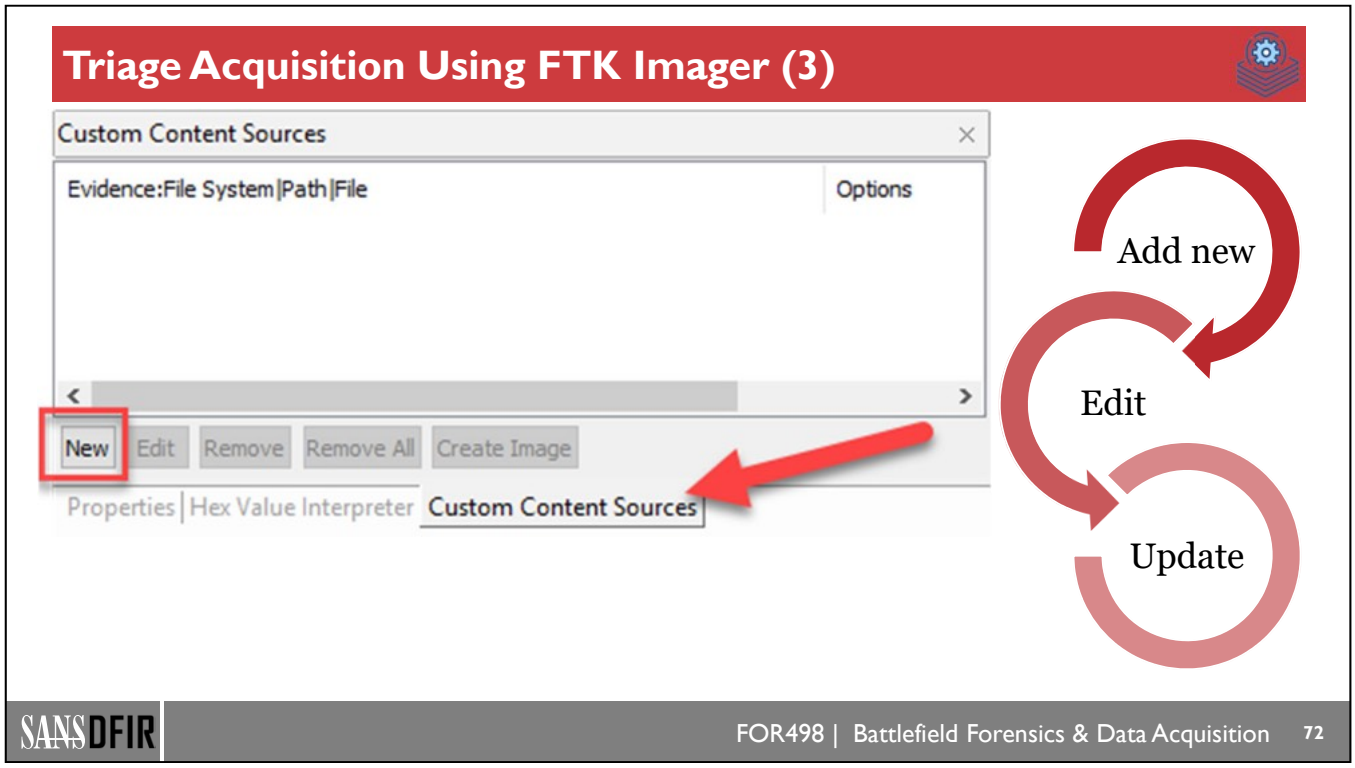
There are several ways to include files in a custom content image. The most flexible method is using the **Custom Content Sources** feature to add files using wild cards, by path, and so on. We will see this in detail in the next few slides.

The other option is to simply right-click on files or directories as you navigate the evidence. A context menu will be shown where one of the options is **Add to Custom Content Image (AD1).** Selecting this option will add the selected files and directories to the custom image. Note that when adding a directory to a custom image, all files and folders under that folder will also be added.

Triage Acquisition Using FTK Imager (2)

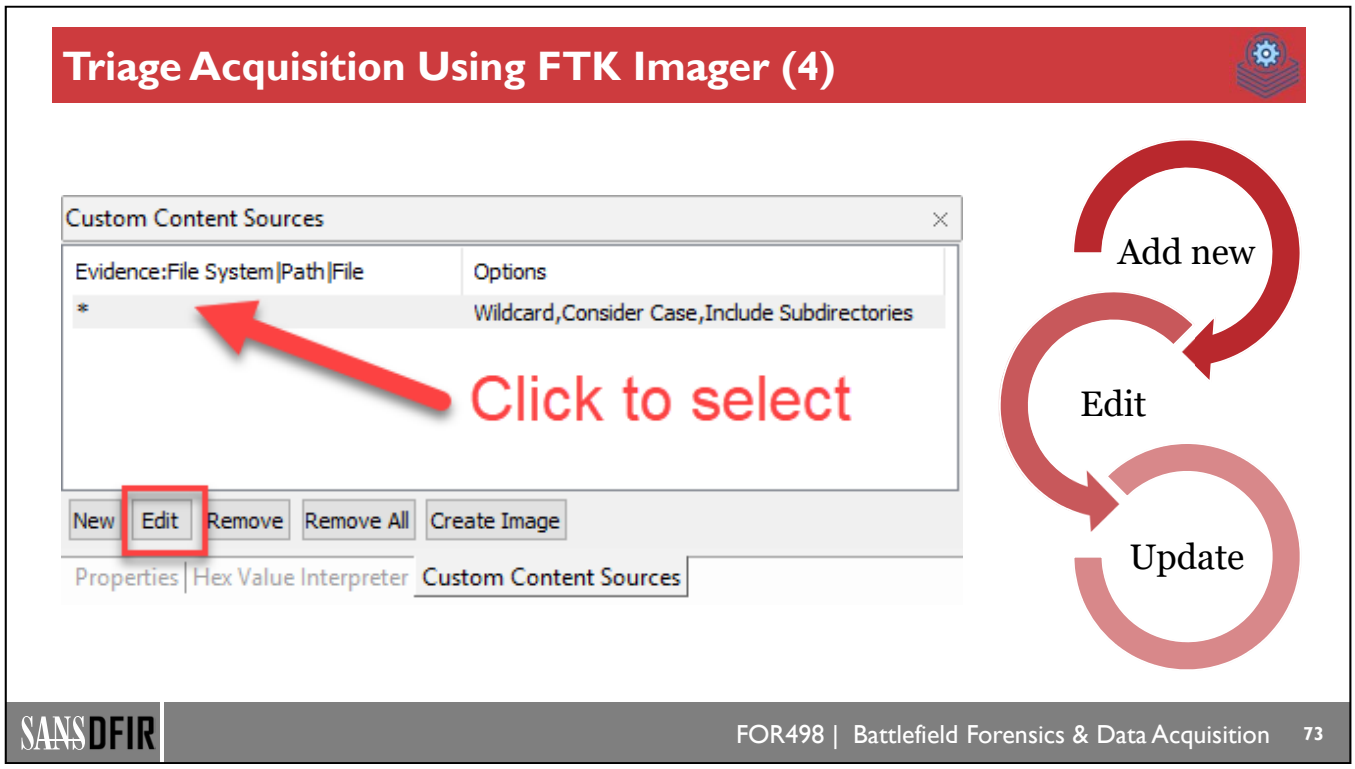


Here is an example of a Registry hive and its associated LOG files that have been added as Custom Content Sources. Note there is a button to add a **New** entry to the list, as well as the ability to edit a selected entry, remove an entry, and so on. Let's take a closer look at how we can achieve this manually.



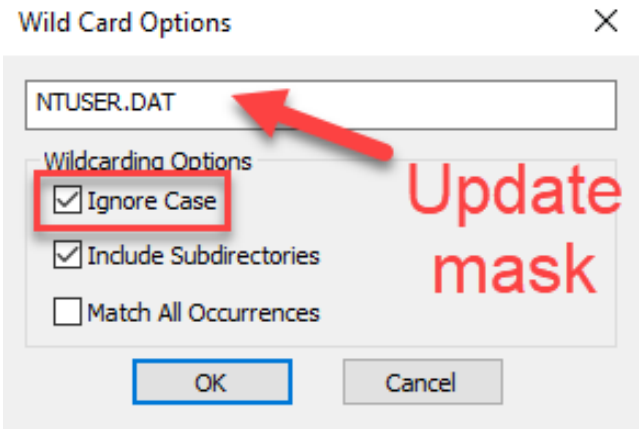
To create a new wildcard entry, select the **New** button in the lower left corner of the **Custom Content Sources** window. You can also press the **Alt+N** keyboard shortcut.

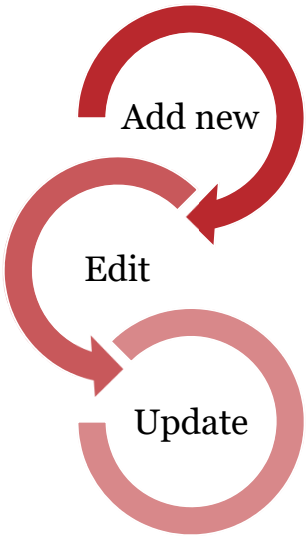
Clicking the **New** button adds an asterisk (*) in the **Custom Content Sources** window. Select the asterisk, and then click the **Edit** button (Shortcut: **Alt+E**).



After adding a new entry, select it, which will enable the **Edit** button so the details can be updated.

Triage Acquisition Using FTK Imager (5)



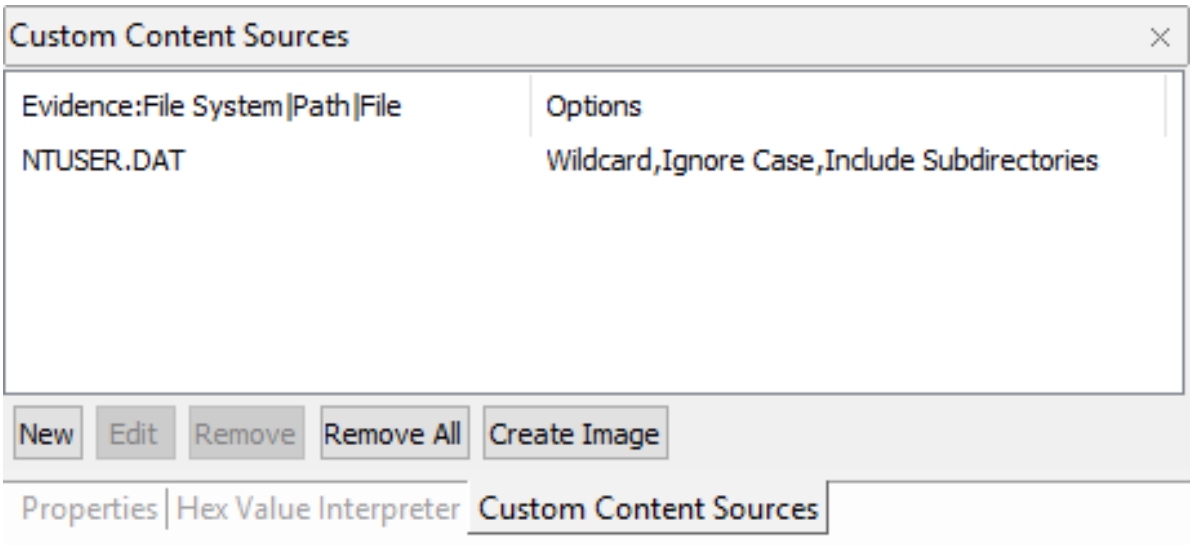


Once **Edit** is clicked, the **Wild Card Options** dialog box is shown. Here, enter the exact name of the file. For example, **`NTUSER.DAT`** would add every instance of "NTUSER.DAT" to the custom content image. You can also search for any variation of a filename with a wildcard. For example, **`*.evtx`** would include any file that ends with that file extension. You also have the option to ignore case and match all occurrences.

When finished editing the entry, click the **OK** button to save the changes.

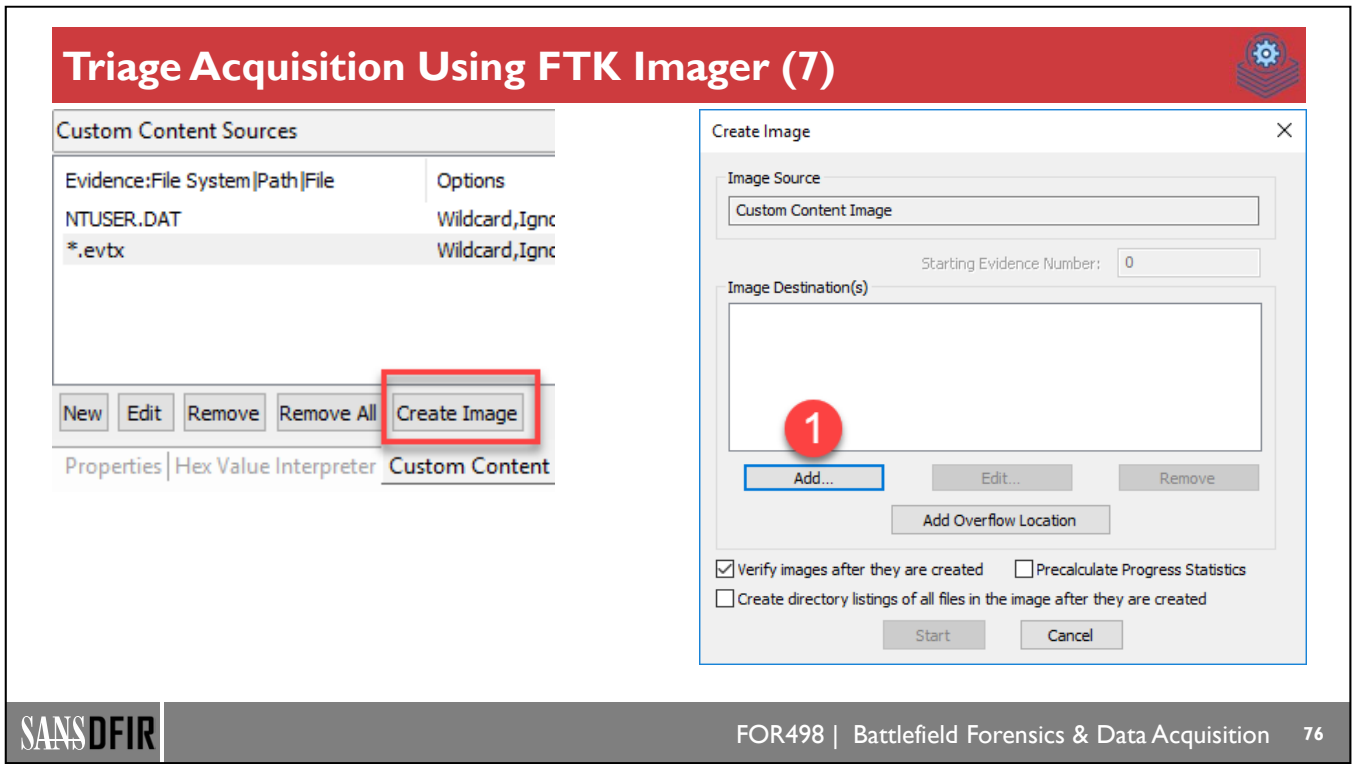
Repeat this process for as many different file masks as necessary.

Triage Acquisition Using FTK Imager (6)



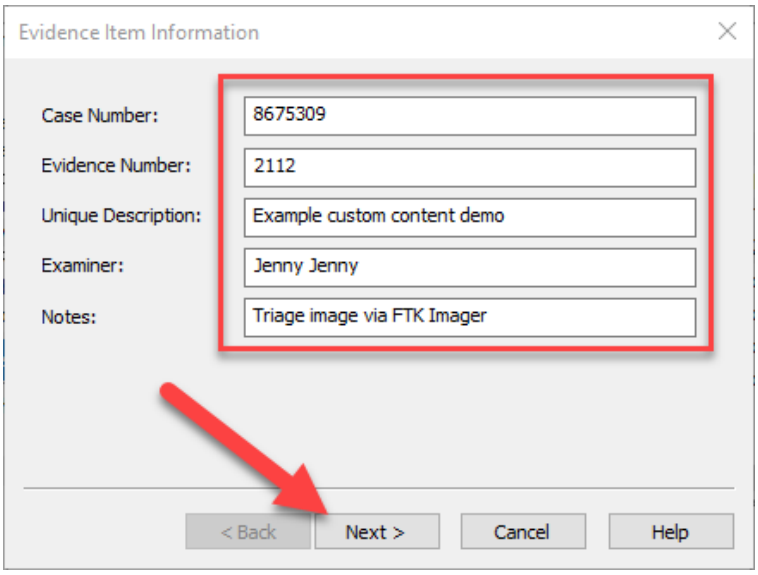
With the details updated and saved, the new information is reflected in the dialog.

Repeat this process for as many different file masks as necessary.



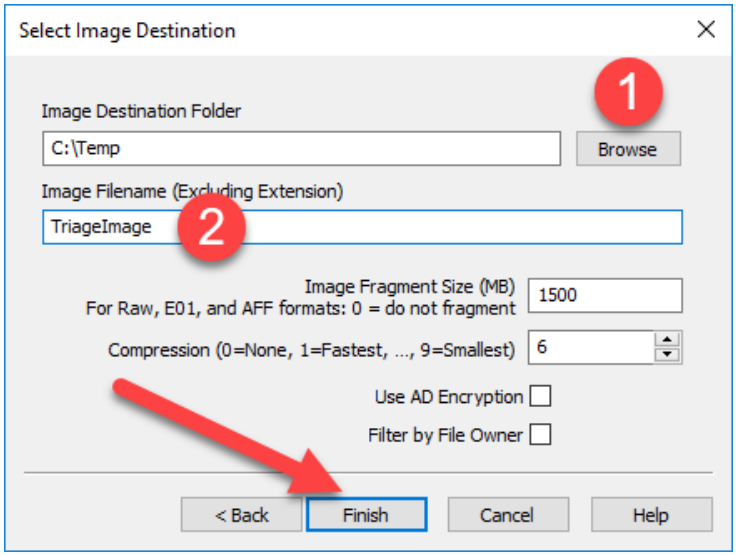
Once you have completed adding file masks and are ready to create the image, start the process by clicking the **Create Image** button. A wizard will then be displayed that walks you through the details related to case information, where to save the image, and so on.

Triage Acquisition Using FTK Imager (8)



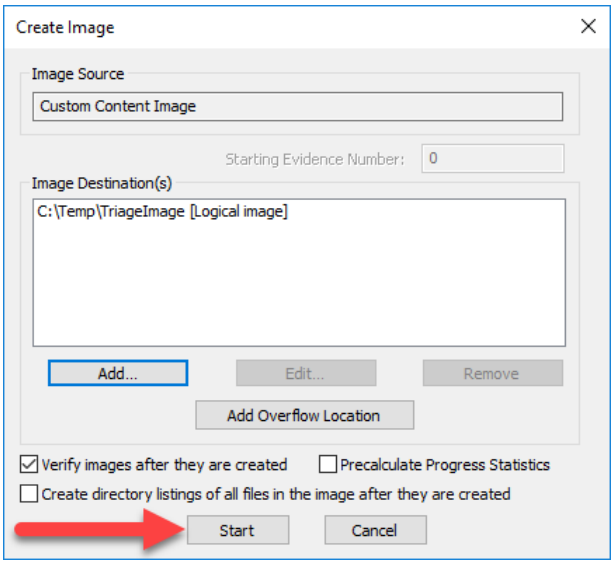
This page intentionally left blank.

Triage Acquisition Using FTK Imager (9)



This page intentionally left blank.

Triage Acquisition Using FTK Imager (10)

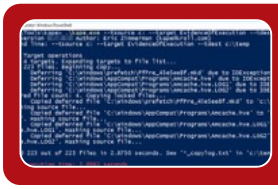


Once the wizard is done, click the **Start** button to create the custom content image.

The custom content image will be created in a proprietary format from AccessData, known as AD1. As such, only certain tools will be able to read the custom content image (FTK and Mount Image Pro for example).

The number of file masks added and the size of the source device determine how long the image creation process will take.

Triage Collection with KAPE (1)



Command-line collection and processing tool

- Targets to collect are defined via external text based files
- Works against live systems, mounted images, and via F-Response
- Can optionally run programs against collected data

Benefits

- Rapid file enumeration and locked file support including ADS
- Forensically sound extraction including timestamp preservation
- Automatic VSC processing and collection

Robust output options

- Maintains proper folder hierarchy
- Copy to directory or VHD(x) container
- Optional deduplication of data by SHA-1

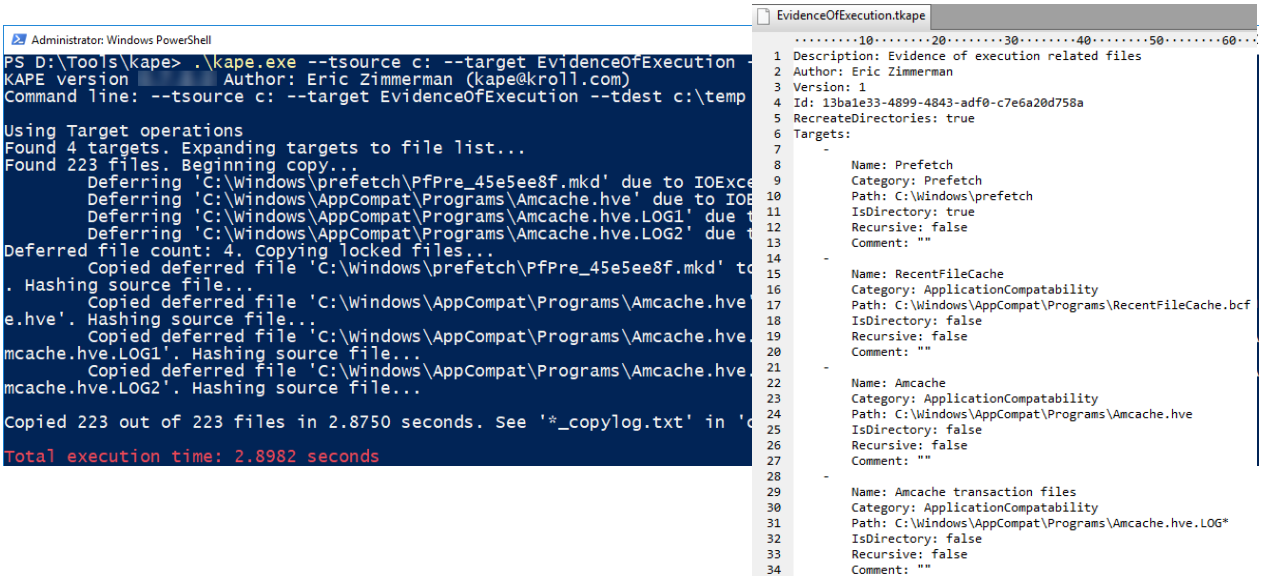
Kroll Artifact Parser & Extractor (KAPE) is written by Eric Zimmerman and was born out of prior work related to triaging computers both in his role with the FBI and in the larger forensics community. KAPE was designed to be as flexible and extensible as possible by end users, without the need for any coding or additional development. In and of itself, KAPE does not know how to do much. It requires configuration files to be used that define what files and directories to copy (in the case of targets) or which programs to run against what files (in the case of modules).

KAPE targets fully support wild cards, which makes it easy to create configurations that can locate files within a few seconds, ideally grouped by type (Registry hives or event logs, for example).

KAPE was designed to be forensically sound in that it preserves original timestamps for files and directories, logs everything it does, calculates the SHA-1 value of source files, deduplicates based on SHA-1 and much more. It also supports finding and searching available volume shadow copies (VSC). KAPE also handles in-use files when run on a live system, which makes it an equally effective tool against a running computer or against a mounted disk image.

We will be spending a significant amount of time with KAPE in a future section. For now it is enough to understand the basics of what it does. It quickly finds and extracts selected files and directories from a file system while maintaining the forensic integrity of the data being extracted.

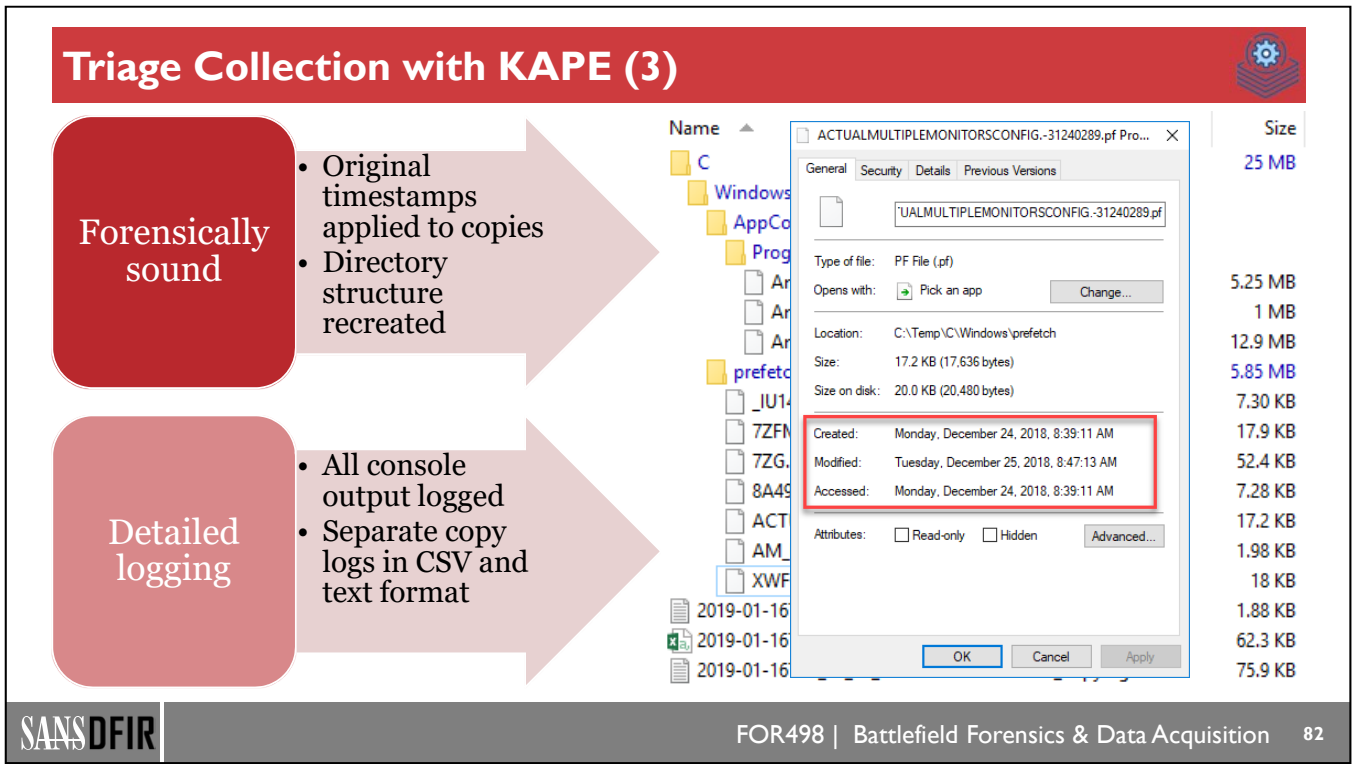
Triage Collection with KAPE (2)



Here we see an example of KAPE running against a live system. The EvidenceOfExecution target is specified, which means it will collect things such as prefetch files and amcache.hve. Notice that KAPE was able to handle both locked and unlocked files as necessary.

In this particular example, KAPE was told to dump collected files to c:\temp. If the --vhdx switch was used however, all of the collected files would have ended up in a VHDX container.

KAPE's configuration files use YAML (YAML Ain't Markup Language) to define the "rules" for collection and processing. The EvidenceOfExecution target shown above is an example of what the configuration syntax looks like.

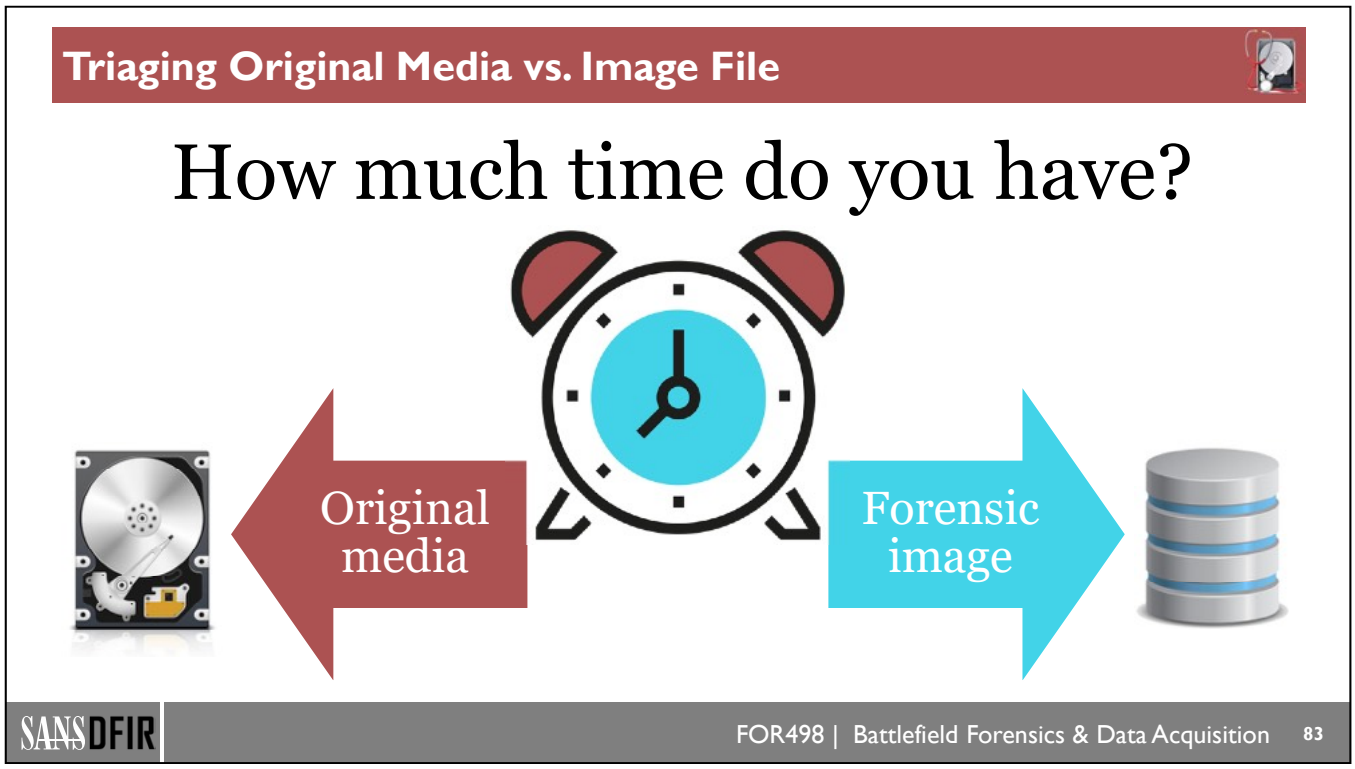


Here we see the partial output of the KAPE run we looked at previously. Some of the prefetch files were removed so as to include examples of everything else that is present in the directory.

Notice that the file system hierarchy was recreated, starting from the C:\ drive (this was the source that KAPE used to find files). Each file that is extracted is placed in the corresponding directory where it was found. All original timestamps are reapplied to each directory and all files that are copied as well.

One way to look at this example is that KAPE essentially removed all of the "noise" around evidence of execution artifacts as defined in the target configuration because it found and copied out only what it was told to. This lets you focus on the things you want to focus on much faster (the collection took ~2.8 seconds!).

The log files include a console log, which shows the output of everything that went across the console while KAPE was running (useful for long running operations or when using more robust logging options, like --debug and --trace). The "copylog" files contain full details of every file that was found and **actually copied**.



When dealing with evidence, it is rarely a good idea to work against original evidence. As such, in digital forensics, one of the first things we do when dealing with a device is to create a forensic image of that device. Once the image is created, we can work on the forensic image and therefore avoid working on the original.

But is this always practical? Is this always necessary? The answer, like in most things, is that it depends on the circumstances. Should you triage against original media? Of course, even if you do triage against original media, you always want to be as minimally intrusive as possible, and that means the use of a write blocker. This is no different than what you would do when imaging a device, so this is not out of the ordinary.

Now the question becomes this: Once you have a hard drive (for example) connected to a write blocker, how do you know whether you should triage against the original or image first, then triage against the image?

To answer this question, consider another question: How much time do you have? If time is not a concern, then the least intrusive thing to do is to make an image and triage from there, but in an emergency situation where there is a safety concern or other urgent circumstance, triaging directly against original media is the right move.

Each of these approaches has their pros and cons, but what if we can meet somewhere in the middle to find a solution that captures the benefits of both approaches? Can we find a win-win solution to this?

What if we, after connecting a device to a write blocker, and prior to collecting a full image, triaged the device using software such as KAPE to quickly locate and forensically extract key data? Once KAPE is finished and the full disk image process is started, analysis can be started using the data collected by KAPE. Recall that 99% of forensic analysis typically focuses on 1% of the data. Rather than wait hours for a full image to be created, why not let KAPE run for a few minutes to collect the 1% of data first? You then have the entire time the full disk image is being created to look at the data and generate leads.

This is our win-win situation!

Summary

- Triage acquisition allows an examiner to quickly extract relevant data
- Understanding how data is handled by a computer allows an examiner to know what to focus on for quick wins
- FTK Imager allows an examiner to create a Custom Content Image of any data the examiner wishes
- KAPE is a forensic magic button that allows for surgical extraction and analysis like no tool ever before
- Open source target and module files allow an examiner to customize their analysis in incredibly granular ways

This page intentionally left blank.

Exercise 3.3

Triage Acquisition

Synopsis: In this exercise, you will perform the creation of a Custom Content Image using FTK Imager. You will then analyze your findings in Timeline Explorer.

Average Time: 40 Minutes

This page intentionally left blank.

Exercise 3.3 Takeaway

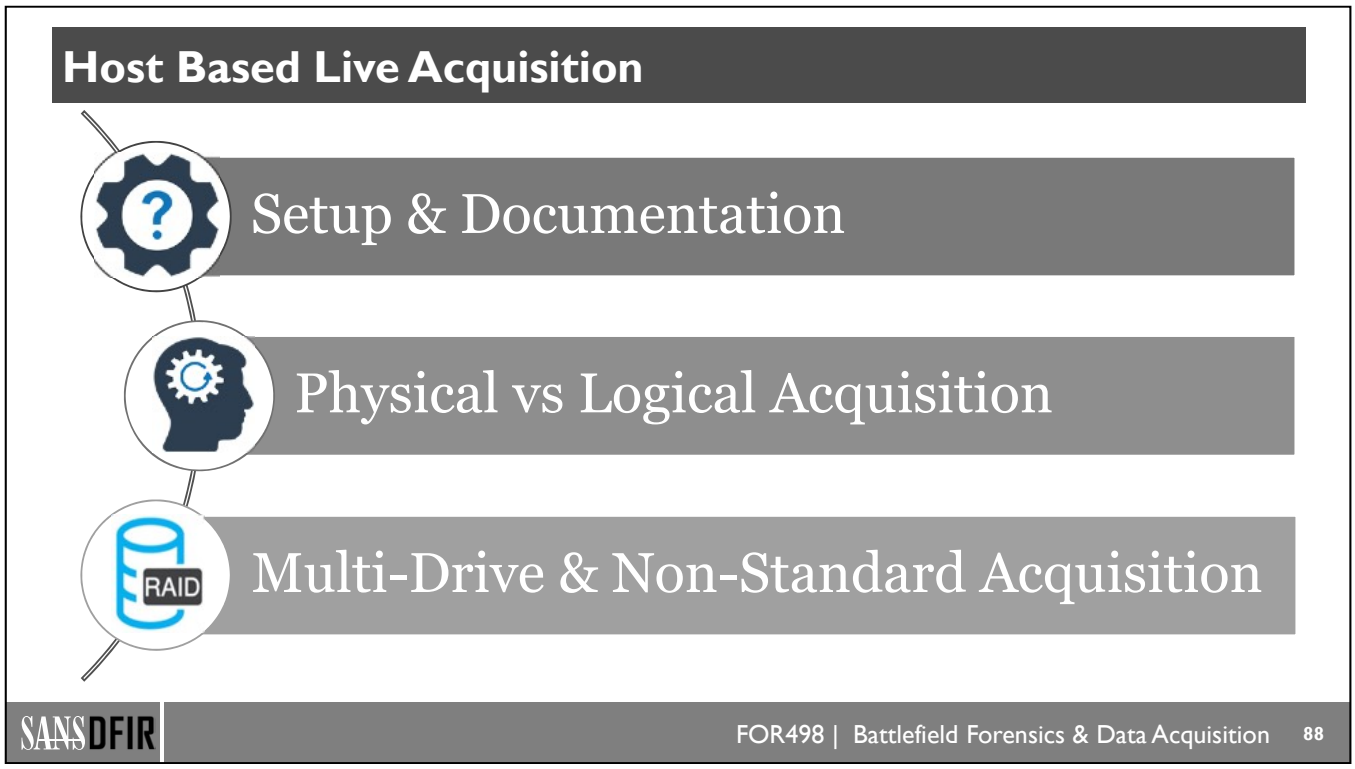
- Being able to manually locate and select files on a live system or from a mounted image can save considerable time, since you can quickly get to the data you need.
- FTK Imager allows for specifically adding files to a custom image as well as being able to specify things using wildcards. Using wildcards can save time because FTK Imager will find and add matching files automatically.
- Once data is collected into a Custom Content Image, the image file can be mounted using FTK Imager to allow access to analytical tools.

This page intentionally left blank.

FOR498.3: Quick Win Forensics Agenda

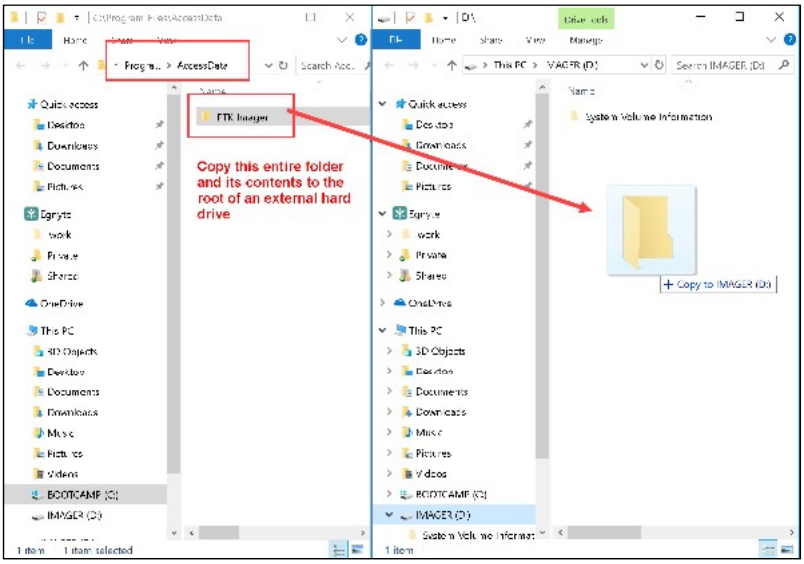
- 3.1 Memory Acquisition & Encryption Checking
- 3.2 Mounting Evidence
- 3.3 Triage Acquisition
- 3.4 Host Based Live Acquisition
- 3.5 Dead Box Acquisition

This page intentionally left blank.



This page intentionally left blank.

Create FTK Imager Lite



Conducting live acquisition of the hard drive consists of accurate preparation with the appropriate software. As has been stated previously, you obviously want to cause as little interaction with the subject machine as possible. Having said that, we know that to perform a live acquisition, we must at least insert a USB device to the system. We then have to run our acquisition program.

In setting up for this task, the acquisition software is placed on a USB device. Although FTK Imager is a great acquisition tool, it needs to be installed on a system before it can be run. What we need is something like FTK Imager that we can place on a USB drive, so that when we run it, it is not necessary to install it on the subject computer. Enter FTK Imager Lite.

The setup is quite simple. Pick an external hard drive that is large enough to accept a forensic image of the hard drive in the computer you are about to perform the imaging process on. Connect this external hard drive to a computer that already has FTK Imager installed on it. Navigate to C:\Program Files\AccessData. Inside the AccessData folder, you will find a folder called FTK Imager. Drag and drop, or copy and paste this entire folder to the root of your external hard drive. Once the folder is copied over to the external hard drive, the drive can be disconnected from the computer.

You now have FTK Imager Lite on the external hard drive, and it is ready to be used for acquisitions.

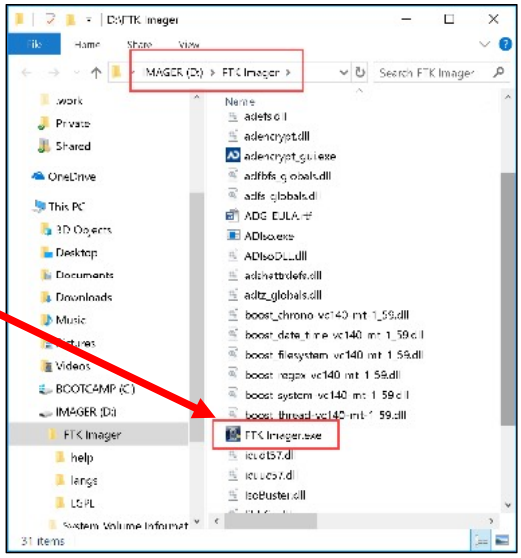
The Physical Setup

1

Insert external device with FTK Imager Lite into host computer

2

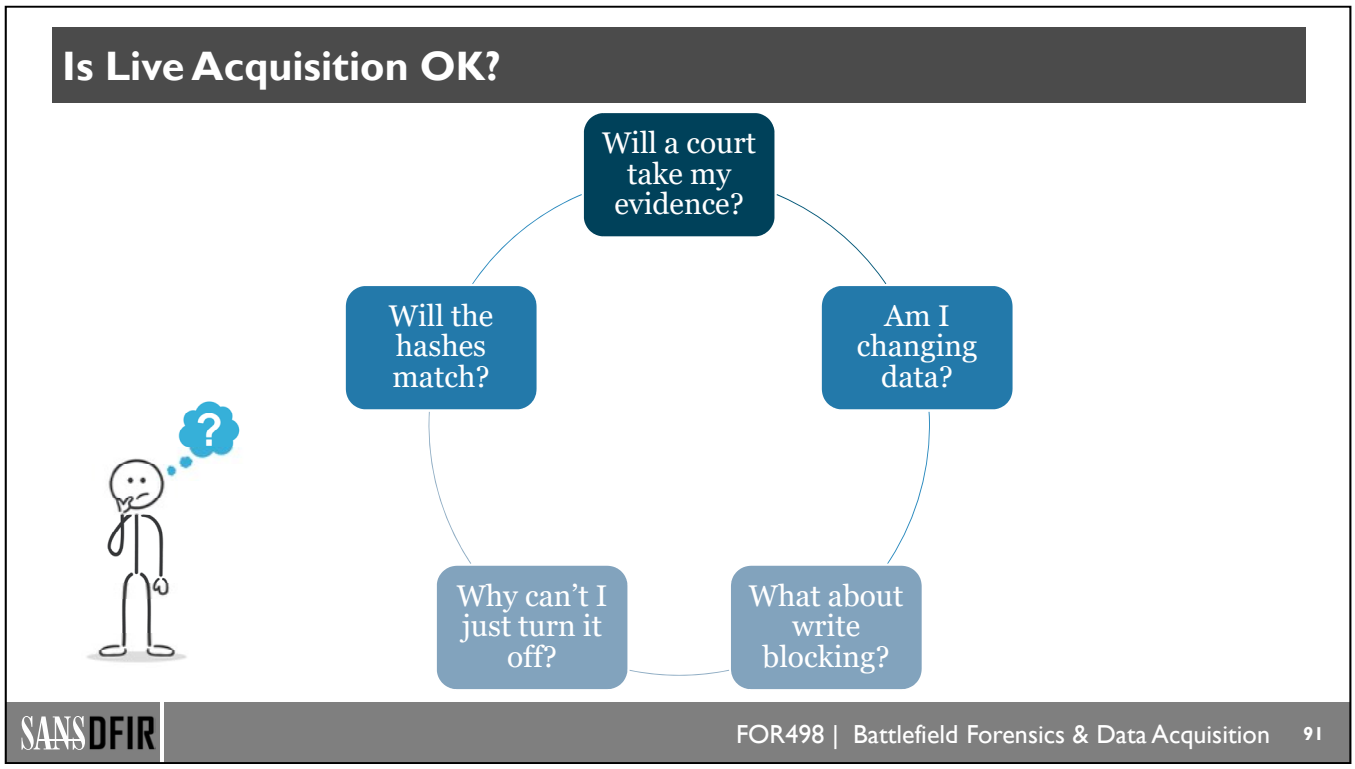
Using File Explorer on host, navigate to FTK Imager.exe



SANS DFIR FOR498 | Battlefield Forensics & Data Acquisition 90

Once you reach the point in your investigation where the acquisition process will take place, you will prepare the physical set up of the devices as necessary. This should already be in place if you have collected the memory image. In any event, if the external hard drive containing FTK Imager Lite is not already connected to the computer, connect it now and then access the drive through the host computer's File Explorer. Open the FTK Imager folder found at the root of your external drive and then locate the FTK Imager.exe program inside this folder. Once you double-click on the program executable, the program will start without the need for installation. At this time you would follow the steps outlined in a different module to perform the acquisition.

A couple of pitfalls to avoid include making sure the computer does not go to sleep, and also making sure that (with a laptop) you have a power cord AND it is plugged in AND it is getting power! Nothing is more frustrating than wasting time getting started, only to have it interrupted by the screensaver coming on, or worse, having the battery die 3 hours into a 5 hour acquisition.



Today the question often gets asked whether or not a live acquisition is okay. While there are a number of different variables involved in any acquisition process, the answer to this question would be, it depends. Realizing that this is a very common answer when discussing any types of digital forensic investigations, nonetheless it is very true. There are many different circumstances which would dictate whether the live acquisition is okay or not. As has been stated since the beginning of this class, the biggest consideration is to do as little damage, or make as few changes, to the data as possible. Having said that, once again we must weigh the potential of damage to data against the value of being able to extract data that will be lost if we turn the machine off.

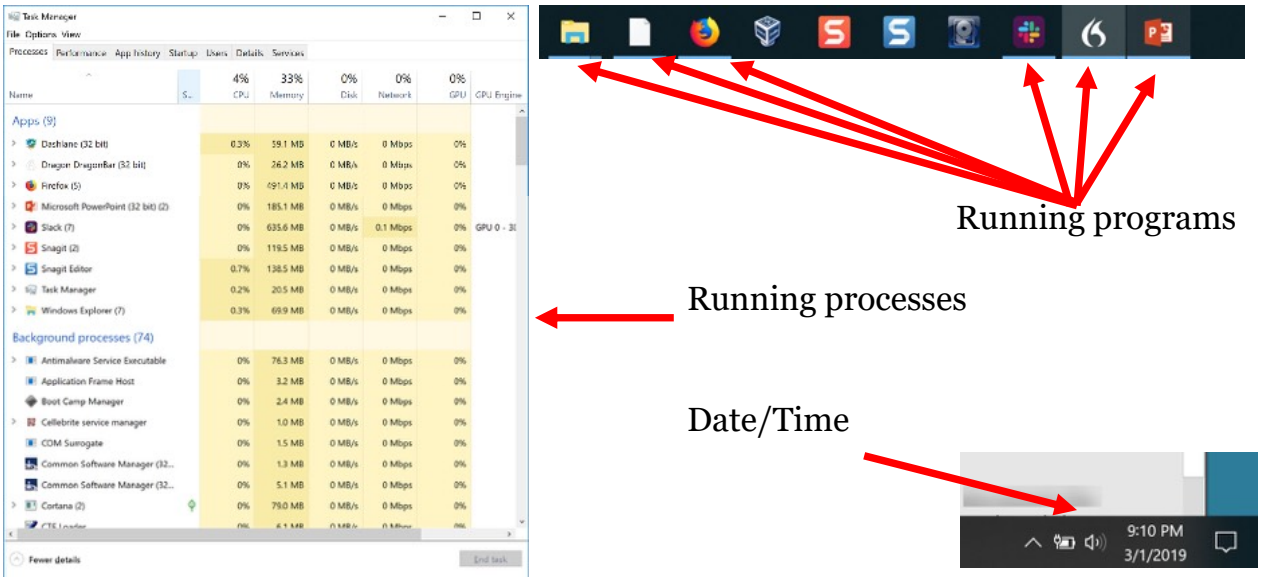
In the case of full disk encryption, unless it is a well-known process and the examiner actually has the unlock key it would be a fool's errand to turn off this type of machine.

There is also a fair amount of dialogue today regarding the imaging of solid-state hard drives. If the computer is on, it makes sense to image these types of drives live. When you turn the computer off, the reintroduction of power can cause various functions to occur on the drive below the reach of any write blocking hardware or utilities.

Making the decision to image a computer hard drive in a live state is a judgment call in any event. We cannot cover every possible eventuality in this text. Mentoring under someone, and or gaining your own experience, are the best teachers. Even then, careful consideration must be applied in every case and it is recommended that the thought process be directed to account for questions that may be asked after the fact.

Create a checklist, or even a small exercise for yourself, to direct your decision-making. The questions should be treated as though someone else was questioning you. The first question is, "Why did you choose to image the drive live, rather than shutting it down first?" If you do not have a clear answer to this question, and one that follows best practice and is defensible, then maybe the best choice would be to turn the computer off.

Documentation

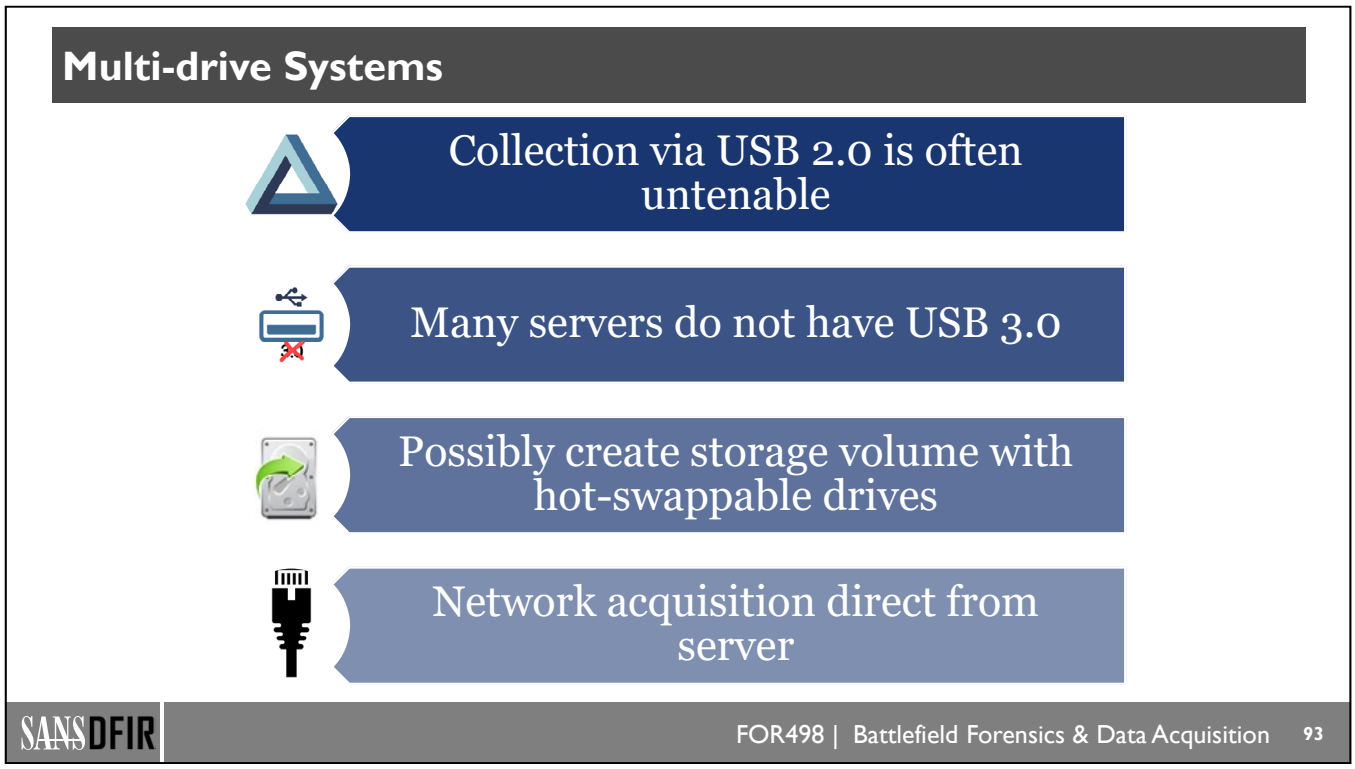


Although this has already been discussed in an earlier section of the class, it is important enough that it bears repeating. Any handling of the subject computer while it is on is causing changes and potential data destruction to the system. As a result you must have a very measured and logical approach to your interactions with the computer. You must also document meticulously so that you can explain and account for your activities much later. Don't ever fall victim to the notion that you will remember what you did, even days down the road. Especially in many civil cases, it is not uncommon for a forensicator to be called three or four or five years after the examination was done and be expected to answer questions. In any event, merely expecting someone to believe you at face value will almost never work. There is no replacement for detailed documentation.

Besides collecting the memory and determining whether or not any disk encryption exists, it is quite standard to photograph at least the desktop in the state in which you found it. This includes any screensavers, desktops, desktop notes, as well as any other open windows. Note any open programs by viewing the taskbar. Hovering your mouse over any icon will show whether the program is open, and indeed what file is being handled by that program. For example, if it is noted that Microsoft Word is open and a document is being worked on, you can click the program button in the taskbar to expand the file. You would then photograph this. Very importantly, also photograph and document the date and time being reported by the computer. In the case of Windows, this is usually found in the bottom right corner of the desktop, and on an Apple computer, this is usually found in the top right corner of the desktop.

Depending on the direction of the investigation, it may be prudent to access the Task Manager to photograph and document the running services on the system. You can also click on the start button and photograph and document any programs that are listed.

Above all, use common sense and need to dictate interacting with the computer. Just because it is on, does not mean the forensicator should click randomly throughout the computer to “see what is going on“.



Network addressable storage (NAS), RAID configurations, and other multi-disk storage solutions found usually in servers create their own unique acquisition challenges. Most servers will have USB ports on board; however it is not uncommon for these to be of the USB 2.0 variety. You can well imagine the challenge of performing a multi-terabyte data acquisition through a USB 2.0 port.

This course cannot possibly prepare you for every situation that you may encounter in the field. A great deal of ingenuity as well as thinking outside the box will be what will guide you towards a successful resolution.

As an example, let us suggest that you have arrived on scene and must image a multi-disk server. You note that the server only has USB 2.0 ports on board. You must first identify what it is that you are going to acquire. Does the acquisition involve imaging the entirety of the server? In many cases the answer will be no. It is quite often the case that a certain volume is all that is necessary to be imaged. In any event, let us suggest that you must acquire a 6 TB volume on the server. It would be unrealistic to acquire this through USB 2.0. Under the most ideal of conditions with USB 3.0, the theoretical transfer rate would be about 3.5 GB per minute [1], or over 28 hours just to acquire the data without verification. Realistically, transfer rates while hashing the acquisition would not be nearly that high. With USB 2.0, you could expect to see transfer rates closer to 1.5 GB per minute.

Enterprises today usually have robust bandwidth in the form of at least 1 Gbps, and many sport 10 Gbps. It may well be far more efficient to perform the acquisition using the server's network card as your connection interface. Using this interface would at least double the speed achieved using USB 2.0. Clearly the set up you have created with FTK Imager Lite on an external hard drive is not going to work in this scenario. Thinking outside the box, it may be better to build a network addressable storage device with the appropriate storage area and then connect it to the server. Depending on the environment, FTK Imager Lite or quite possibly a command line instruction, will be used to perform the acquisition process and push the image from the server to the NAS.

In cases where time is more important than cost, it might be acceptable to insert hot swappable drives into available bays and push the image to a purposely created volume. In this case, you could run FTK Imager Lite from a USB stick inserted into the USB 2.0 port.

[1] Realistic USB 2.0 transfer rates | `http://for498.com/r6gc2`

Determining What to Collect

Would you like the whole filing cabinet? (physical acquisition)

Or just one drawer? (volume acquisition)

Once all the drawers are closed, you can't access any of them. (whole disk encryption)

Especially with live acquisition, consideration has to be given to what specifically will be acquired. In the majority of cases, an examiner strives for a full physical image of the hard drive. This is the best of all worlds, but not always possible.

Depending on the direction of the investigation, it may be that all the examiner is collecting is material that can be gathered through triage imaging. It also may be that, in the case of whole disk encryption, you may not benefit from a physical image collection.

On a hard drive with whole disk encryption, once the decryption key is loaded into memory, it decrypts the data on the fly as a user requires it. In the case of a logical (volume) acquisition being performed on a live system, this would be exactly that type of activity. It is a common misconception that once the decryption key is entered, that it decrypts the drive in its entirety. Remembering that in whole disk encryption when the data is at rest, it is encrypted, then even if you are interacting with a live system but collect a physical image, it will be encrypted within the image upon completion. Without the key you will not be able to decrypt that physical image.

To review then, especially in the case of whole disk encryption, you will be collecting a volume image as opposed to a physical image of the entire drive. This can best be described as collecting a forensic image of the C: drive, or any other volume that may be present on the drive. In this manner you are still able to collect all resident data, as well as any slack space and unallocated file space.

Physical vs. Logical Imaging (1)

- The physical disk is outlined in green (Entire box Disk 2).
- The logical volume is outlined in red (Internal box Samsung_T5).
- In the case of whole disk encryption, acquiring physical disk means no access to data without the key.
- Acquiring logical volume means access to all data in volume.

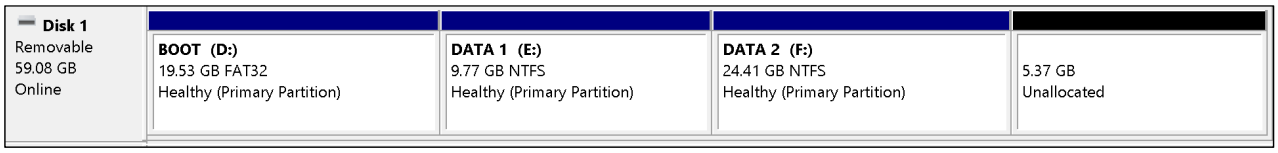
If you are in a whole disk encryption scenario and are just collecting a volume, be prepared to answer the questions that will inevitably come about what you may have missed.

If we think about what goes into creating a hard drive or preparing a hard drive for data storage, an empty drive first becomes formatted. Imagine this is the outside walls of a house. You will then divide this area into at least one big room, or storage area. In many cases this means that the walls you have created are going to contain one large room that will use all available space. This is often seen in the case of a hard drive that only has a C: drive and it uses all available space on the hard drive.

If we look back at our home building analogy, then collecting the volume as opposed to the physical would be more akin to gathering all of the contents of the room in our house but leaving the outer walls behind. We can gather information about those outer walls however, they're not going to contain data in so far as user activity is concerned.

Ultimately, in the case of a volume that occupies all available space on a physical disk, what are you conceivably missing between imaging physically, or just a volume? The answer in a nutshell, is Master Boot Record information. The lack of MBR information has no effect on resident and unallocated data within the volume in any way.

Physical vs. Logical Imaging (2)

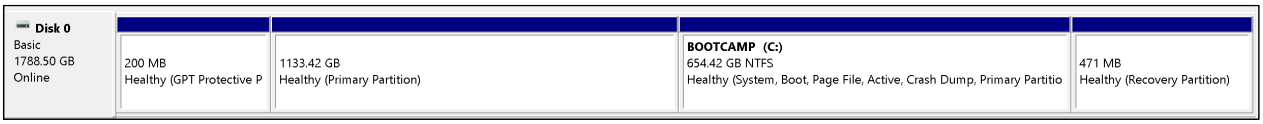


- In the above case, this will require 3 separate images.
- But what about the unused portion at the end, labelled Unallocated? (more on this later)

If there is more than one volume on the hard drive, in other words maybe a C: drive and a D: drive, then imaging both volumes separately may become necessary.

Where a drive area does not contain a drive letter, this will pose a problem for many forensicators. Unfortunately as you will soon see, some tools aren't exactly forthcoming about what is available.

Non-Standard Volumes

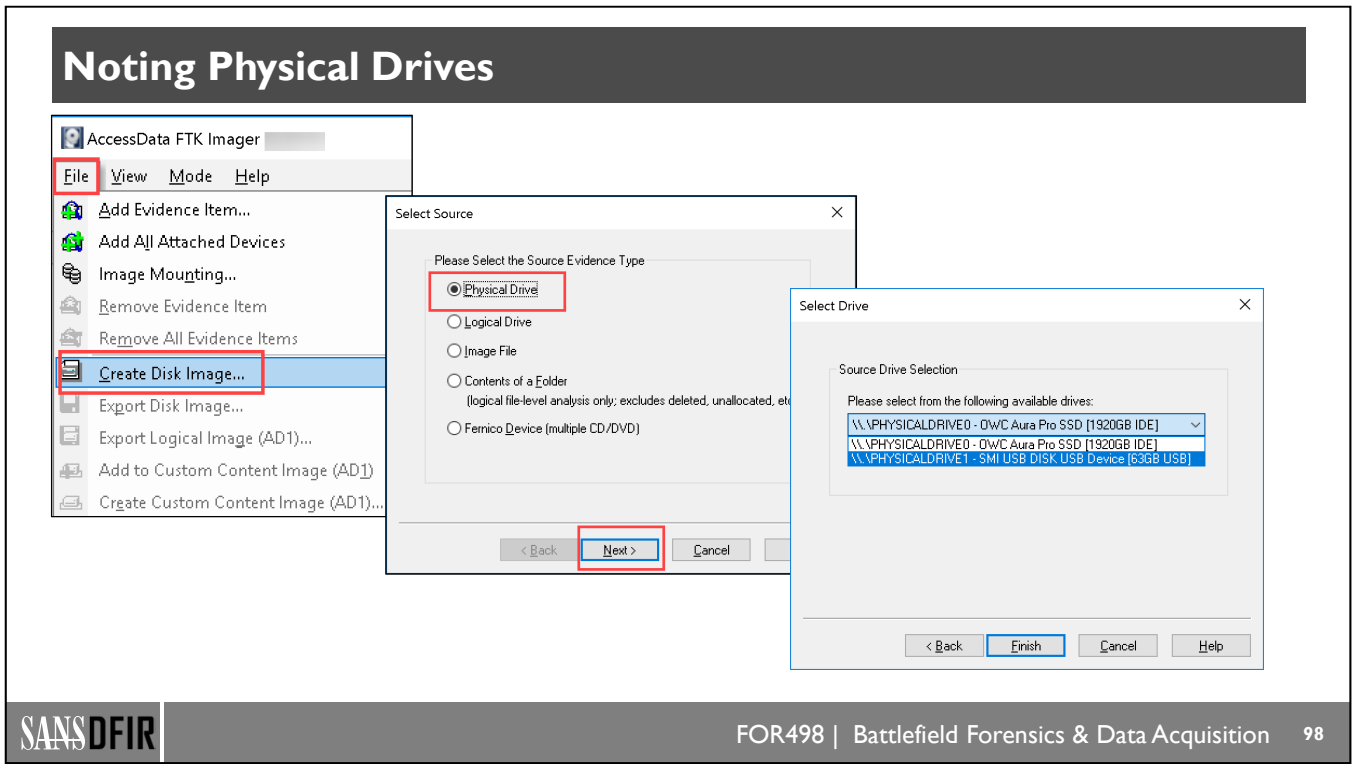


- No drive letter
- Many tools cannot see volume without drive letter
- Others are not very intuitive as to how to image

On hard drives in branded computers, there may be other volumes resident that are not normally visible to the user. Given that they do not have drive letters, it would take more than the tools and ability of the vast majority of users to access, and indeed attempt, to hide data. These areas fall under the category of Host Protected Area (HPA) and Device Configuration Overlay (DCO). This could also be things like Mac OS partitions, as well as Linux partitions. Something that might reside in the HPA for example, would be the recovery data needed to repair a computer in the case of failure, or simply to reinstall Windows back to factory settings on the computer. Again, although it is possible to use these areas for data hiding, the likelihood is extremely low. In any event though, when we attempt a live acquisition of the system, we should be presented with at least the recovery partitions. It would then be a judgment call on the part of the examiner as to whether they wish to collect this or not.

In the example in the slide, this is a "Bootcamp" partition[1] containing the C: volume, and would be found on an Apple computer. As you can see, this drive has 4 partitions, but only 2 would reasonably be expected to hold user data. Specifically in this case, the Apple OS partition (seen with a size of 1133.42 GB) would be encrypted at rest (today's Apple devices with the Apple File System encrypt by default), and so it may be a waste of time attempting to image that partition.

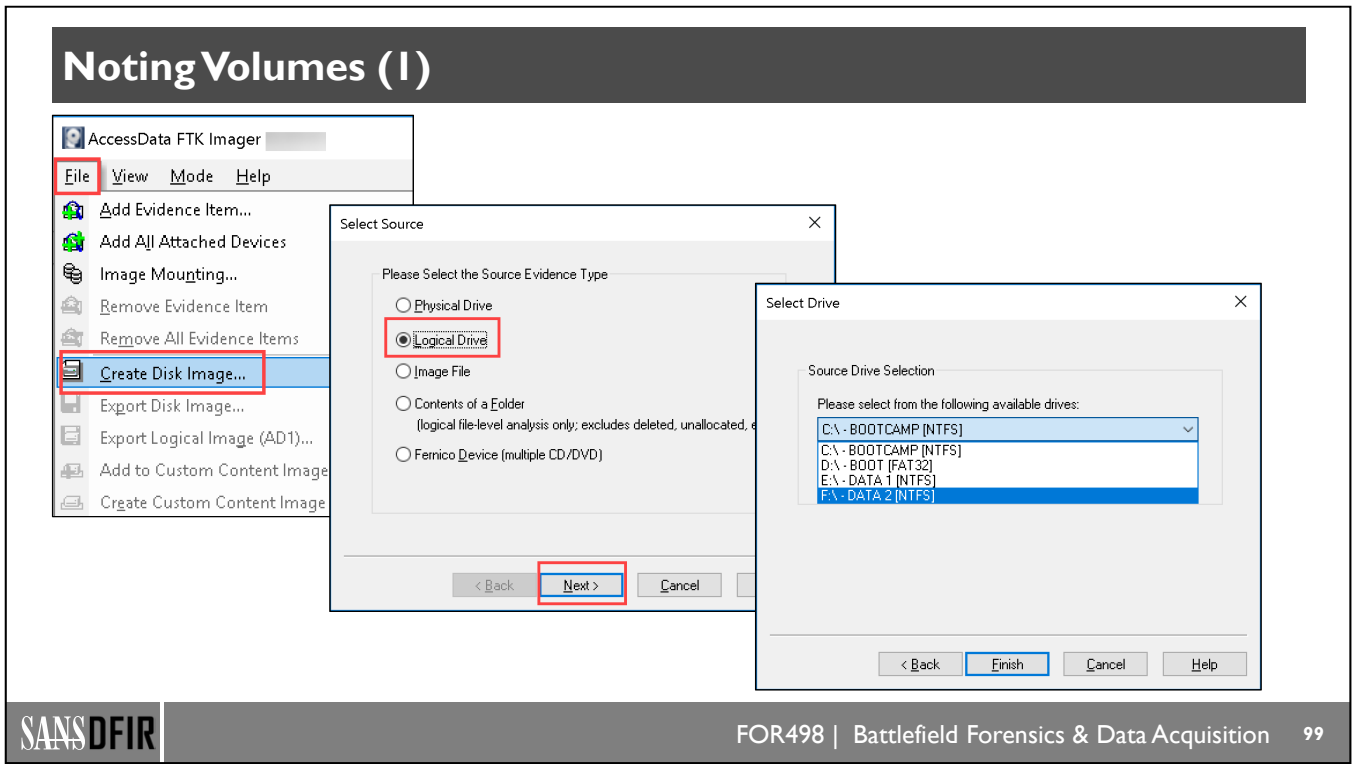
[1] Partition vs Volume | `http://for498.com/96spe`



In many cases, the same software that is used for dead box imaging can be used in host based live acquisition. Depending on the software, it may not be straightforward to identify all storage areas.

This slide shows FTK Imager Lite identifying any physical drives that it can see. In this case, there are two. Remember though, that if you are imaging this drive in a live state, there is usually a particular reason, and that reason is most commonly due to the detecting of whole disk encryption.

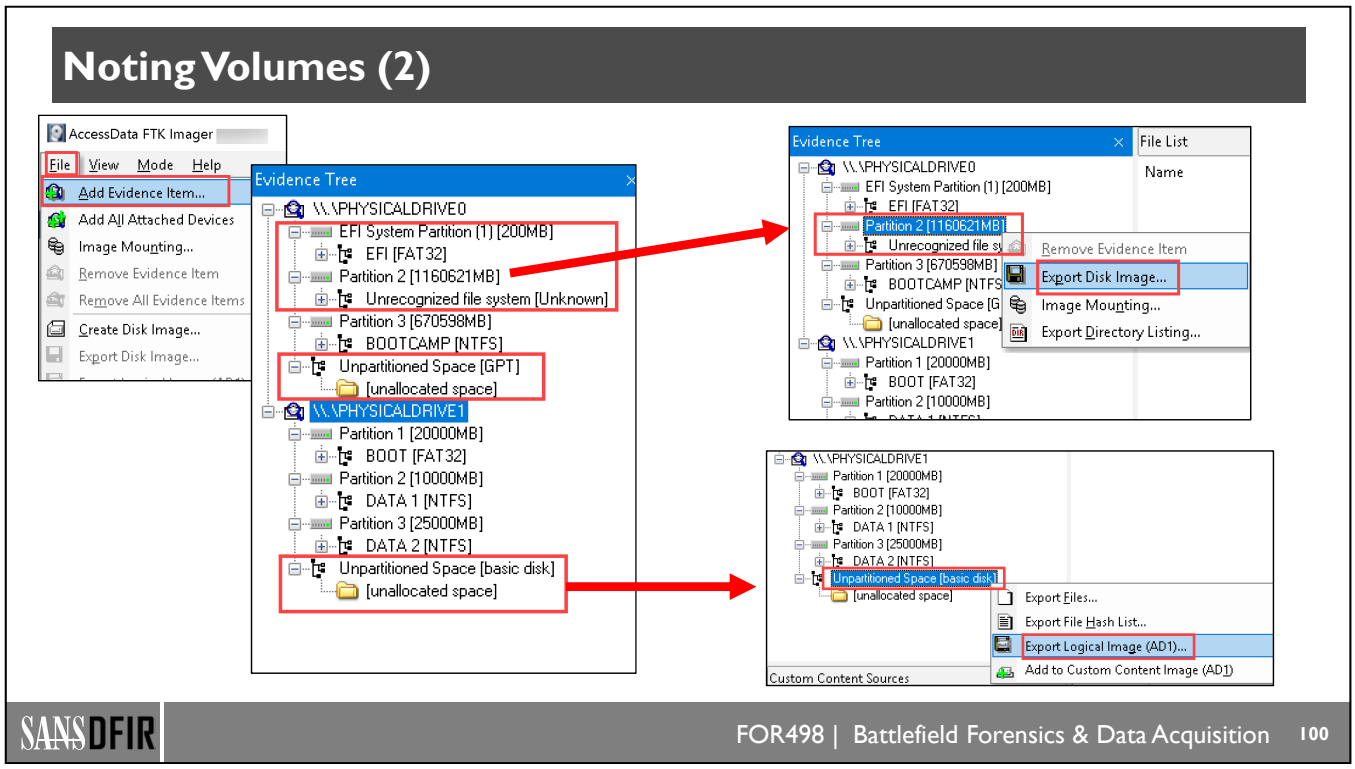
Two other reasons may be that this drive is a solid state drive, or it may be in an operational server that cannot be shut down.



Where whole disk encryption is the reason that an examiner is imaging the device live, acquiring the volume or volumes is generally the only option available. As well, if this is a server with an extremely large storage area and multiple volumes, you may only be interested in certain ones.

If using FTK Imager Lite, as seen above, you can point it to any given volume that has a drive letter. In the above case, we see 4 volumes across 2 physical disks.

This is how FTK Imager and Imager Lite are most commonly utilized. For many first responders, and even seasoned examiners, the above is believed to be the whole story. Unfortunately, you are not seeing any volumes that don't specifically have a drive letter assigned. If you are unaware of, or not considering volumes without drive letters, you may have some uncomfortable questions to answer as the investigation unfolds.

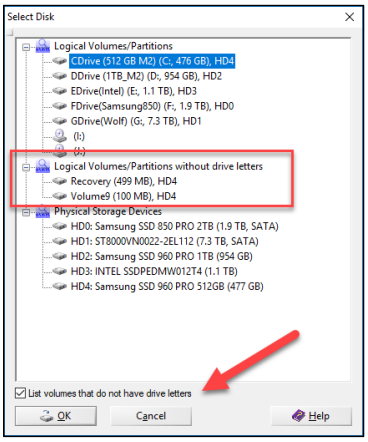


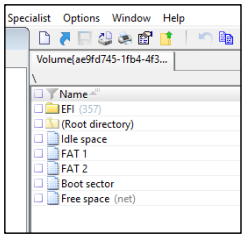
If a forensicator is using FTK Imager Lite, the measures to access non-standard volumes (recovery partitions or volumes without drive letters) are not quite intuitive, and not well understood.

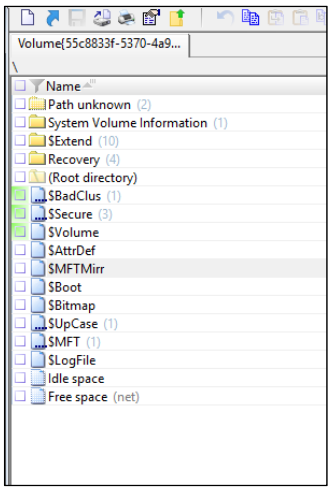
The Imager can still see these volumes, but not through the normally used acquisition steps. Extra measures must be taken. First, the physical drives must be mounted into FTK Imager Lite. Once mounted and expanded, you can see all partitions and volumes. Even then, the acquisition process is not standard. This can lead to confusion on the part of the examiner, and must be recognized for what it is. The difference, as FTK Imager Lite has determined, is that **Partitions** without drive letters are handled one way, while **Unpartitioned Space** gets handled another way. The pitfall here is that FTK Imager Lite has mis-identified a partition as **Unpartitioned Space**, as is the case in **PHYSICALDRIVE0**.

Comparing this process to the one seen in the previous slide, we see here that there are a total of 8 storage spaces, as compared to the 4 that were visible previously.

X-Ways as Alternative







Another approach when dealing with live acquisition might be the use of X-Ways Imager. As can clearly be seen above, X-Ways Imager does an outstanding job of clearly laying out all physical, and logical areas, whether carrying drive letters, or not.

X-Ways Imager, unlike FTK Imager, is not free though. The point of showing both products is to drive home the idea that different tools often display different results.

Summary

- Imaging a system while it is on is not only possible, but may be the best approach
- The level of documentation must be very high, and detailed
- You are making changes to a system with every interaction
- Make sure you can explain the choices you have made
- With larger data sets, especially in the enterprise, it may be better to think outside the box in terms of connection interface

This page intentionally left blank.

Exercise 3.4

Host-Based Live Acquisition

Synopsis: In this exercise, you will use FTK Imager to collect both a logical and a physical image of a live system.

Average Time: 15 Minutes

This page intentionally left blank.

Exercise 3.4 Takeaway

- Imaging a device is generally a straightforward operation but can be time consuming.
- If encryption is found, it may be necessary to image the logical partition in its unencrypted state while the machine is running.

This page intentionally left blank.

FOR498.3: Quick Win Forensics Agenda

- 3.1 Memory Acquisition & Encryption Checking
- 3.2 Mounting Evidence
- 3.3 Triage Acquisition
- 3.4 Host Based Live Acquisition
- 3.5 Dead Box Acquisition

This page intentionally left blank.

Dead Box Acquisition

Storage Device Removal

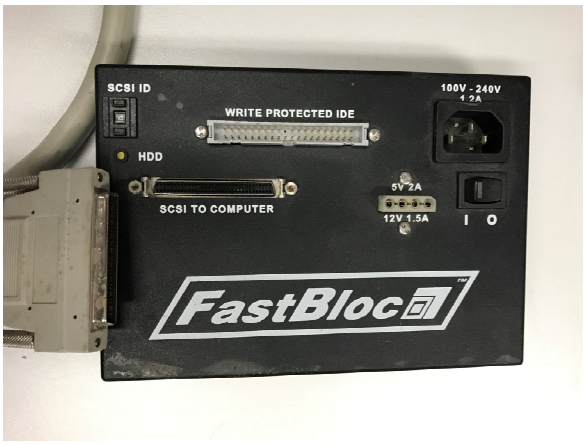
Hardware Acquisition Devices

Specialty Acquisition

This page intentionally left blank.

Traditional Acquisition

- Used to be the only approach
- Remove hard drive
- Connect to a write blocker
- Image with Linux, FTK, or EnCase



Ever since digital forensics became a thing, first responders and forensicators alike have been performing dead box acquisition. In the late nineties, when a drive was removed from the computer, it was connected in many cases, to the SCSI device that Guidance Software sold for the purpose. It was a large device that required conversion to SCSI via an adapter, and then it connected into the computer via a thick cable to a SCSI PCI card. Imaging speeds at the time were around 500 MB per minute, and we were happy to have it.

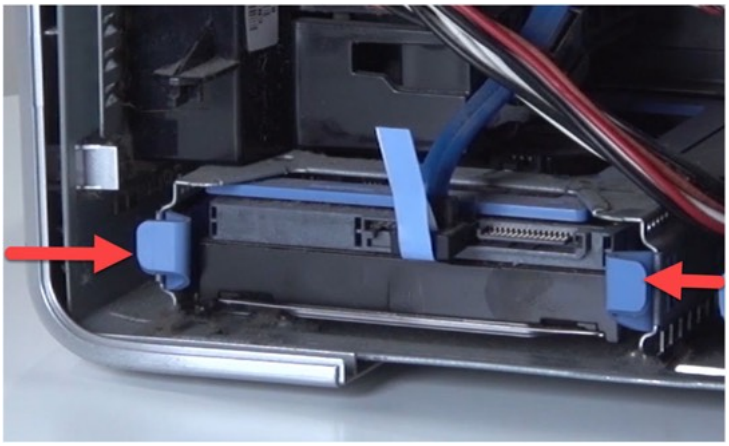
The biggest difficulty then was removing the hard drive. Not much was known about the construct of a computer, and figuring out how to dismantle it to remove the hard drive was a project in and of itself. The next challenge was figuring out how to set up the jumpers on the back of the hard drive. Besides the two rows of pins for the IDE connection, there are two rows of variously 3-5 pins each that were used along with jumpers to tell the computer how to treat the hard drive. Back then, two drives were allowed on a data cable, so the computer had to be told which hard drive was the Master, and which hard drive was the Slave. Even if there was only one drive on a cable, it still had to be jumpered in such a way as to tell the computer that it was the only hard drive.

Laptops were easier to figure out, because they had a slot on the edge whereby, after you removed one or two screws, the hard drive slide directly out. Hard drives were of the IDE variety, both in laptops and in desktops. It scaled very well because this was the interface of the day, and it was largely the only way that computers were built.

Dead box acquisition was the only approach. The notion of acquiring a computer while it was on hadn't really been considered in the mainstream, and the handling protocols even in the largest of agencies was to shut the machine off prior to seizure, if it was found to be powered on.

Today we still perform dead box acquisition, and it is indeed the most controlled way to acquire data, given that the computer is off at the moment of seizure.

Media Removal: Desktop



Today the removal of hard drives from desktop computers is both easier and harder, depending on the device in question. Gone are the days where a box was a box was a box. Today we have many different kinds of computer towers, from very small form factor gaming machines that are portable, to things like the Mac Mini and others very much like it. There is no one single way that can be shown to remove the hard drive from these. The important key is to be able to identify a hard drive by sight, and then be able to figure out how to remove it. Websites like ifixit.com can be very helpful, as can the plethora of YouTube videos available on the subject. The cheaper the computer, the harder it is to remove the hard drive. What is meant by this is that if a tower is built using inexpensive parts, we would expect to see hard screws everywhere, requiring screwdrivers to dismantle. As more money is spent on components, we see things like quick release latches and thumbscrews so the hard drives can be removed without needing tools.

When discussing branded computers, it is quite common for them now to have quick release methods to gain access to the hard drives. There is probably no better vendor than Dell in this regard. Dell computers today can be opened up using quick release latches and panels, and then the hard drive can be removed using quick release tabs. These devices are usually in the form of plastic enclosures with squeeze tabs, that when pinched, will facilitate the removal of the drive.

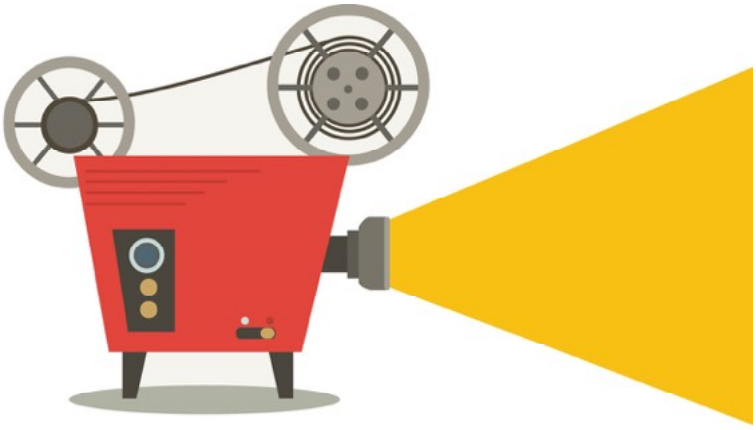
With computers built for gaming, or built for high quality video or audio purposes, the inside of the box can be quite cluttered with equipment attached to the motherboard. Very large video cards are commonly found in machines today, and multiples of these can be found in computers designed for the purpose of mining crypto currency. The first responder must be very careful when removing hard drives in machines such as this because it can become easy to bump a component or break something. In some cases it may become necessary to actually remove components from inside the computer prior to attempting to access the hard drive. In these cases it is important to photograph the inside of the box carefully and completely, in order to ensure that all components be replaced properly.

Consider that storage devices may not be in a common form factor that is easily recognizable. With M.2 solid-state hard drives that look very much like a stick of RAM, these can actually be screwed onto the motherboard. In yet another location, PCIe cards are being used as hosts for solid-state M.2 drives. If that isn't confusing enough the PCIe card itself could potentially be the hard drive.

Desktop computers commonly contain RAID configurations today. You cannot always determine whether multiple hard drives in a tower are a RAID configuration or not simply by looking at them. The mere existence of multiple drives is not automatically indicative of RAID. To determine if the computer has a RAID configuration, the first indication would be more than one hard drive. It is also possible that you will see a RAID card plugged into the motherboard via a PCI slot. Another indication can be labeling on the hard drives such as drive zero and drive one.

While the hard drives are removed for imaging and the examiner is checking or collecting information on the BIOS/UEFI, attention has to be paid to the boot process. If there is a card enabled to control a RAID configuration, this will almost certainly be noted in the boot up process. If it is an onboard software RAID controlled by the BIOS/UEFI, it will be noted within the BIOS/UEFI and also should be looked for. Depending on the circumstances and the goals, it may be necessary to stop the acquisition process, reinstall the drives, and acquire logical volumes. There is no single rule of thumb to dictate your response to a situation like this. Although it is normally recommended to acquire RAID logically, in the case of a standalone system with a two disc RAID, it is probably not going to hurt to acquire each drive separately. It is not an onerous task today to rebuild a RAID zero or a RAID one from the two physical disks using other software.

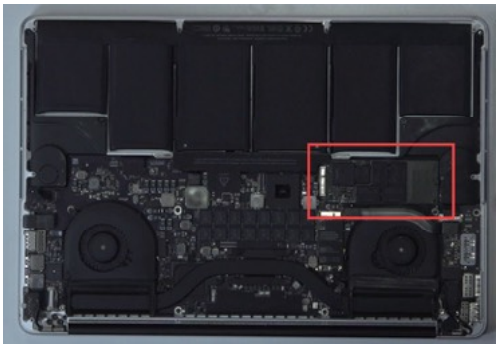
Movie Time!



- 3_1: Generic Desktop HDD Removal
- 3_2: Branded Desktop HDD Removal

This page intentionally left blank.

Media Removal: Laptop



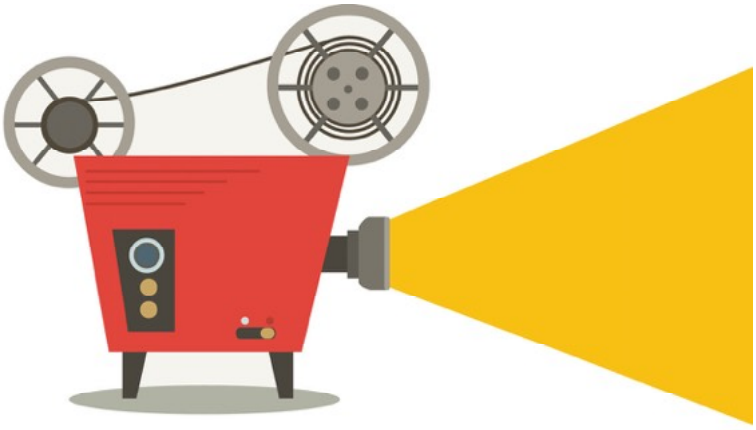
Laptops bring their own challenges to the table. Previously they were easily accessible either through a slot on the side of the computer or a small panel. With manufacturers trying to streamline the size of the computer as well as create new designs, hard drive locations have changed.

The old method still exists on certain larger laptops, whether they be a slot on the side of the computer or the aforementioned access panel on the bottom. In other cases you may have to unscrew the entire bottom of the laptop and then the hard drive will be visible underneath. With ultrathin laptops, you may be dealing again with an M.2 style of hard drive which will involve trying to locate it once the computer is dismantled. In the case of any Apple devices created since 2017, and mostly identified with the existence of the touch bar on the keyboard panel, these hard drives are components that are soldered directly onto the motherboard and cannot be removed.

Consideration must also be given for various different drive configurations such as RAID. Although not very common on laptops, they do exist and need to be accounted for. Never assume that a single hard drive is the only hard drive.

With the myriad different ways of accessing hard drives in today's laptops, there is no question that a first responder or examiner must be comfortable with what electronic components look like. This can only be gained through experience of opening many computers over a span of time and getting familiar with what is seen inside. Again websites like ifixit.com can be very helpful in assisting with the teardown of the computer to access the hard drive. Sometimes it can be difficult to find the correct instructions online. It is helpful to Google using the search term of the computer model followed by “hard drive replacement”.

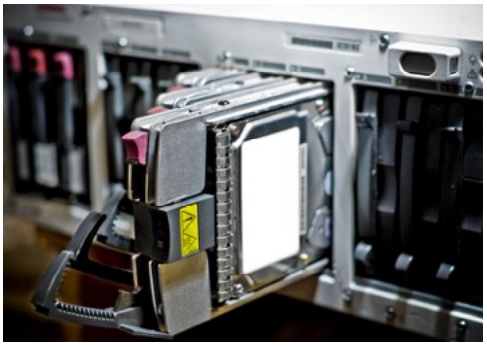
Movie Time!



- 3_3: Branded Laptop HDD Removal 1
- 3_4: Branded Laptop HDD Removal 2
- 3_5: MacBook Pro Laptop HDD Removal

This page intentionally left blank.

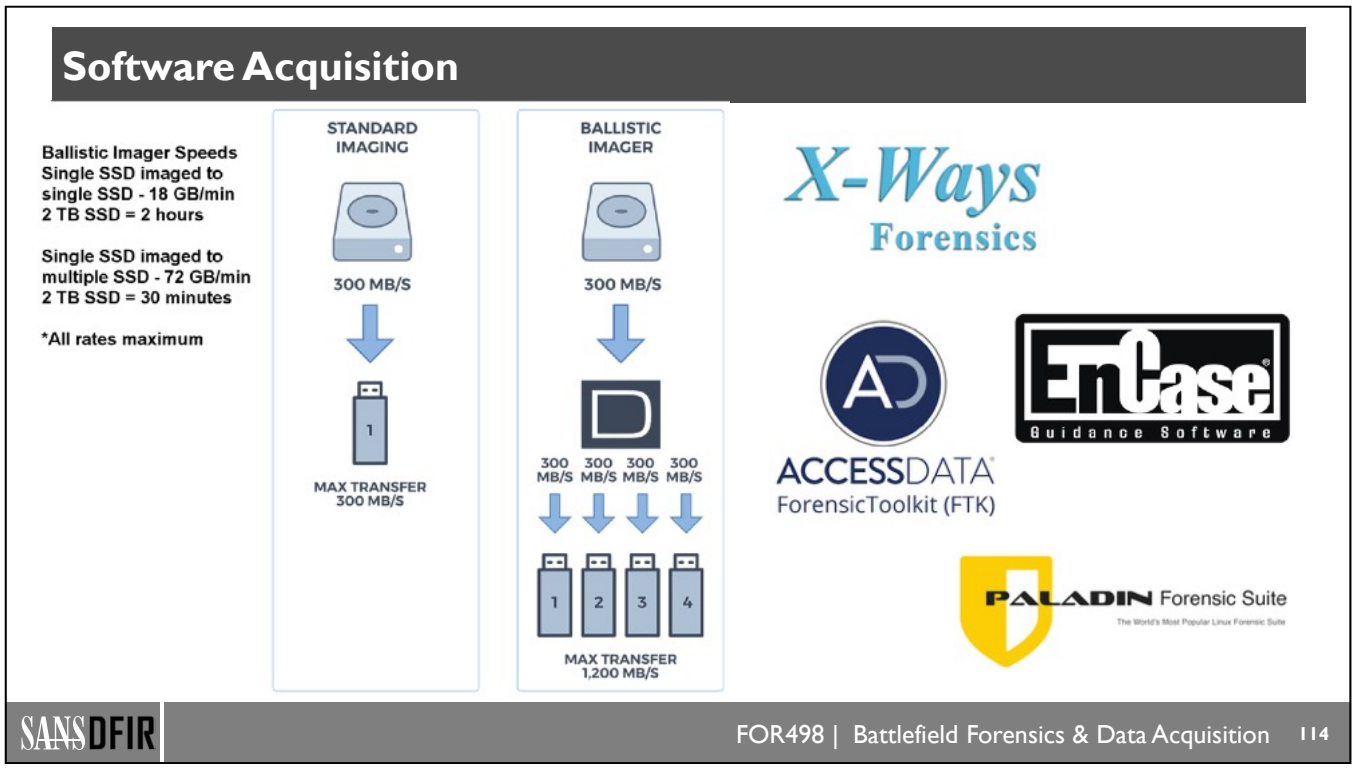
Specialty Media Removal



It is not a common occurrence today to remove hard drives from network addressable storage devices or large arrays for acquisition except in certain circumstances. However if it is necessary, typically the removal of drives from these systems is easier than drives from even desktops or laptops.

Hard drives such as these are typically housed on edge and slid into slots. Removing them is usually as simple as squeezing the plastic pinch bars on either edge of the hard drive and sliding the hard drive out of its slot. In the case of larger arrays and more expensive server equipment there will typically be a swingarm that will pull out from the front of the drive. As you continue to pull on this arm a cam will actually assist in the mechanical removal of the hard drive from its interface. At that point, it's just a matter of continuing to pull on the arm and the hard drive will slide out of its slot.

Often, hard drives of this type are going to be something other than a common interface. You will usually see SCSI, SAS, or fiber channel in arrays. When removing drives from equipment like this, is extremely important to number the drives properly, document extensively, take lots of pictures, and be careful in handling. These hard drives need to go back into their slots in exactly the same order, for the array to function properly again.



Software acquisition involves acquiring data from hard drives using a software solution. Although the hard drive will typically be connected through a write blocking device that may be considered a hardware device, the actual acquisition process is happening through a piece of software on the examiner or host computer.

Software acquisition reduces the speed by which an acquisition process can occur. 3 to 4 GB per minute would be considered a respectable acquisition speed using a software acquisition process. Speeds of 1 GB per minute are not uncommon, however you can see how that will certainly draw out the process in the case of drives in the terabyte sized territory.

A very innovative solution for rapid acquisition is the Ballistic Imaging System from a company called Detego. The Ballistic Imaging System works very loosely off the idea of plugging in a number of destination drives to the computer. Essentially inserting a destination drive into every available output from the host device. The Ballistic software controls the acquisition process and writes data out to all of the connected devices at once thereby allowing the acquisition process to occur at an incredibly high rate of speed. Once the acquisition process is complete, all the destination devices are taken away to a more stable environment and a full acquisition is built from each of these destination devices.

There are command line solutions that will allow for software acquisition. More commonly though, programs such as FTK Imager from Access Data, or the EnCase acquisition tool from Guidance Software can be used to perform the task.

The hard drive is plugged into a write blocking solution which is then plugged into the examiner's computer. The acquisition tool of choice would be started, and prior to the actual acquisition portion being started, the examiner would have to enter various pieces of information regarding the case. This would include the case number, evidence number, examiner name, and any notes that an examiner may want to enter. Depending on the software program that is being used, this data may become an integrated part of the evidence file itself in the case of an EnCase acquisition, or it may be a separate text file that accompanies the evidence file, as is the case with an FTK acquisition.

These tools should provide feedback as to the progress of the operation. Once the acquisition function is performed, a verification function should start immediately and automatically afterwards. The verification function can take almost as long as the acquisition function, but is a very necessary step. There may be certain circumstances where you will not perform the verification at the same time as the acquisition. You may be in a situation where time is of the essence and it is more important to complete the acquisition and clear the scene than it is to stay longer and verify the process. You would verify the process as soon as possible after the acquisition.

The next step after verification would be to create a second copy of the forensic image and re-verify that copy as well. The reason you need to re-verify the copy, is because you do not know the state of every sector on the destination hard drive. You also do not know the health of specifically every sector on the initial destination hard drive that is housing your collected image. If there is a problem with either of the two hard drives, they may function properly for all intents and purposes to the end-user, however certain sectors may not be copied over for various reasons. The only way you would notice what happened or did not happen was through the verification process of the second copy.

Once you have two verified copies of your forensic image, one gets marked as your gold standard and gets locked in whatever evidence storage facility you are using. It would never get accessed again unless or until there is a problem with your working copy.

Hardware Acquisition

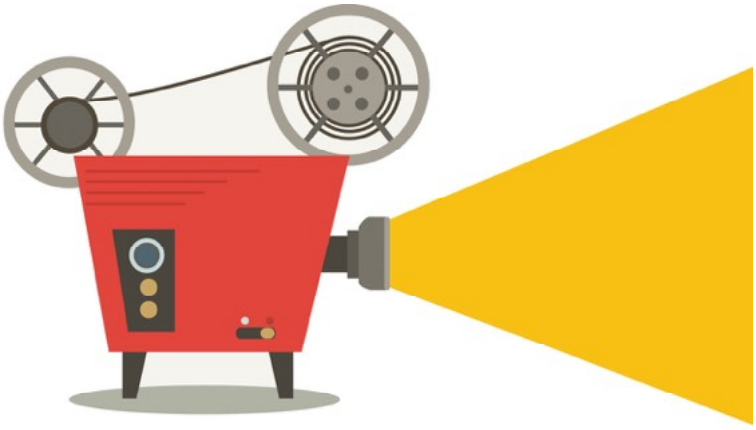
Where it is possible to remove the storage media from a device, hardware acquisition devices can be very helpful. Hardware acquisition devices such as the WiebeTech Ditto, Ditto DX, or devices such as Tableau, Logitech, etc. bring an entirely different dynamic to the acquisition process. It does not come without a price however, with some of these devices ranging into the thousands of dollars. This is money well spent in environments where acquisitions are a common part of daily activity.

Hardware acquisition devices can create two forensic images to two different destination drives simultaneously with no performance degradation during the acquisition process. Speed can also be a significant factor. Many hardware devices commonly acquire spinning media to destination spinning media at rates of 6 to 7 GB per minute. When acquiring solid-state devices these speeds can climb substantially.

Hardware acquisition devices can be controlled from the device itself or very commonly through a web interface. Every step of the process can be controlled including formatting the destination drives, outputting evidence files in predetermined block sizes, and collecting a number of data points regarding the subject hard drive. The Ditto and Ditto DX can also be shipped anywhere, and plugged into a network to be controlled remotely.

If there are any limitations to these devices, it is usually the limitation of the types of devices that can be acquired. Different acquisition devices may have different adapters in different interfaces for a variety of subject drives, and it is important when considering the purchase of one of these devices to explore what storage media can connect to them and what type of adapters are available. It is not as simple as using a normal adapter interface between the subject device and your hardware acquisition device. In some cases the hardware acquisition device will not be able to recognize the subject device on the other side of the adapter.

Movie Time!



- 3_6: Ditto Imaging Device
- 3_7: Acquisition With Ditto Imaging Device

This page intentionally left blank.

Specialty Acquisition

- Not all acquisition is the same!
  - Specialty devices such as NAS, RAID, JBOD not candidates for dead box acquisition
- In these cases, imaging these devices while they are running is often the best bet.
- Even post collection you may have to start such devices up and access them in order to properly collect the data. While hardware vendors can be of some use here, this can often be a slow process.

Besides desktops and laptops there are considerations for other storage devices as well. Larger RAID systems such as multidisc RAID 0, JBOD, RAID 5 and nested RAID bring their own challenges. In the case of RAID acquisition though, it is not common for a dead box acquisition to be performed. In fact, unless there is a mechanical issue with the controller device itself, it is probably best to acquire these devices live.

There are situations where you may need to acquire the physical drives. For example you may have a multi-drive system that has been turned off and disconnected, and you have no visibility into the prior circumstances. In this case, you don't know if all the drives are operational, if RAID tables are current, or if the system will even start up properly when you try. In cases like this if the stakes are high enough, at the risk of adding additional time and expense to the job it may be necessary and prudent to image each hard drive individually for preservation sake. Once they are all imaged, the device can be rebuilt and turned on and if everything mounts properly, then an acquisition can be performed on the volumes as per normal. However if the device is turned on and there are bad drives, the RAID table might start to rebuild the rest of the drives, destroying information. If a situation like this presents itself, you will be very happy that you took the time initially to image each drive individually.

Very commonly seen in many environments today are network addressable storage devices. QNap and Synology are just two common devices seen within a home network and they are typically multidrive devices. These devices have their own abstraction layer between how they communicate with the drive and how it is presented to the user. This abstraction layer usually exists on a chip on the motherboard within the NAS, and as a result, trying to image the drives individually and rebuild them outside of the system will almost always result in failure. For this reason, dead box acquisition is not recommended for these devices.

Microsoft Surface Pro

- Difficult to do dead box imaging
- Cannot remove hard drive
- Cannot rebuild RAID from boot disk – SAFE Block to Go is only viable solution

There are a great many different models and styles of ultralight and ultrathin laptops. In fact the line is blurred between what is considered an ultrathin laptop and a tablet, simply because in most of these cases the screen is removable and can be used as a touchscreen computer much like a tablet. Apple led the charge in terms of thinner and thinner devices, but Microsoft came out with their own, falling within the Surface family of devices.

As each new device comes out, it has different types and different numbers of potential connections that are available. One of the most drastic changes in recent time was the MacBook line of products, in that they converted all inputs/outputs to USB-C. Even so there is usually more than one of these on a given system. Probably one of the most difficult devices to address in terms of acquisition today is the Microsoft Surface Pro. This device has exactly one port that can handle input/output, and this is in the form of a USB 3 port.

When faced with the versions that have a single port, how can you begin to address them? Before we answer this, we need to understand a few of the idiosyncrasies of the Microsoft Surface Pro. First of all, the secure boot feature is enabled by default. We cannot access the hard drive until we turn that off. The next issue is that a Surface Pro is typically used as a touchscreen device, however when booting from an external drive to perform the acquisition, the touchscreen feature does not work. For those of you keeping score then, we need to somehow connect a boot device, a destination device, an input device in the form of a keyboard, and an input device in the form of a mouse. Somehow we have to plug four devices into a single port. This can be done, however it's not pretty, nor does it lend itself to a rapid acquisition process.

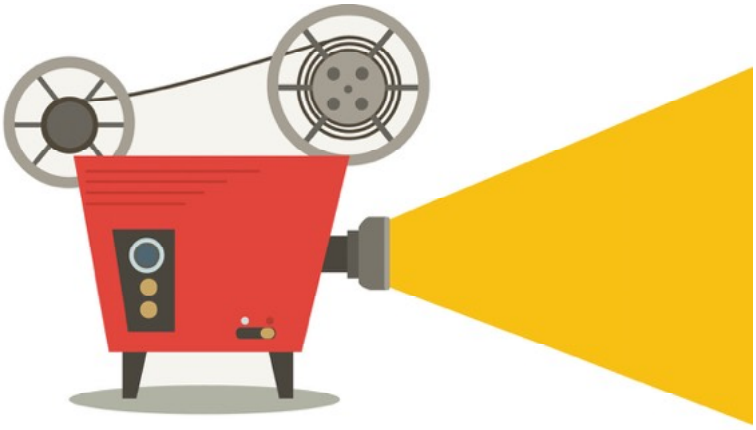
It starts with the insertion of a powered hub. This powered hub needs to be USB 3, and it needs to have at least four ports. Once we insert the hub into the Surface Pro, we then insert the device that we're going to use to boot the system. In our example we are using the Paladin boot disk. We must first boot the system into its UEFI. We do this by pressing and holding the power button and the volume up button simultaneously until we see the UEFI screen. Once inside the UEFI, we need to disable secure boot and we need to change our devices to ensure that the system will boot to our USB boot device. We recommend taking this one step further and actually disabling every other device that is seen by the Surface Pro.

The device is then shutdown and the mouse, keyboard, and destination drive are connected. The Surface Pro is started, and if all works properly, the system should boot into the Paladin operating system at which time the acquisition can be performed.

A relative newcomer to the space is a product called SAFE Block to Go. SAFE Block to Go is a device from ForensicSoft that is a special USB drive that has been created with the Windows to Go operating system as the boot platform, and SAFE Block is the write blocker. Using this configuration (although not free) will allow the examiner to perform the acquisition on a Surface Pro without having to disable secure boot. SAFE Block to Go also addresses the latest versions of Surface devices that have the 1 TB volume. Its 1 TB volume is being created by two 512 GB solid-state drives in a RAID configuration. Due to the security of the device, its drives are not accessible individually and cannot be rebuilt as a single volume using Paladin or any other kind of boot disk. Currently, SAFE Block to Go is the only product we are aware of that can perform an acquisition on this device.

SAFE Block to Go can also be used to boot servers, RAID servers, Apple devices and virtually any computer in a write blocked manner for acquisition.

Movie Time!



- 3_8: Image Surface Pro

This page intentionally left blank.

Summary

- Dead box acquisition can be done with drive removed, or in place
- Storage removal is not an option on some devices
- Storage recognition is very important
- A number of different form factors can be in play
- Many tools, both hardware and software based, are available to the examiner

This page intentionally left blank.

Exercise 3.5

Dead Box Acquisition

Synopsis: In this exercise, you will create a Paladin boot disk, and use it to perform a dead box forensic image collection.

Average Time: 15 Minutes

This page intentionally left blank.

Exercise 3.5 Takeaway

- Using properly configured USB boot disks, an examiner can create a forensically sound acquisition of a device without any other manner of write blocker
- A Paladin boot disk is a great option for the purpose

This page intentionally left blank.