498.2
Portable Devices and Evidence Acquisition

FOR498.2 | Battlefield Forensics & Data Acquisition

Portable Devices and Evidence Acquisition

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

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

FOR498.2: Portable Devices & Evidence Acquisition Agenda

2.1 Portable Device Acquisition

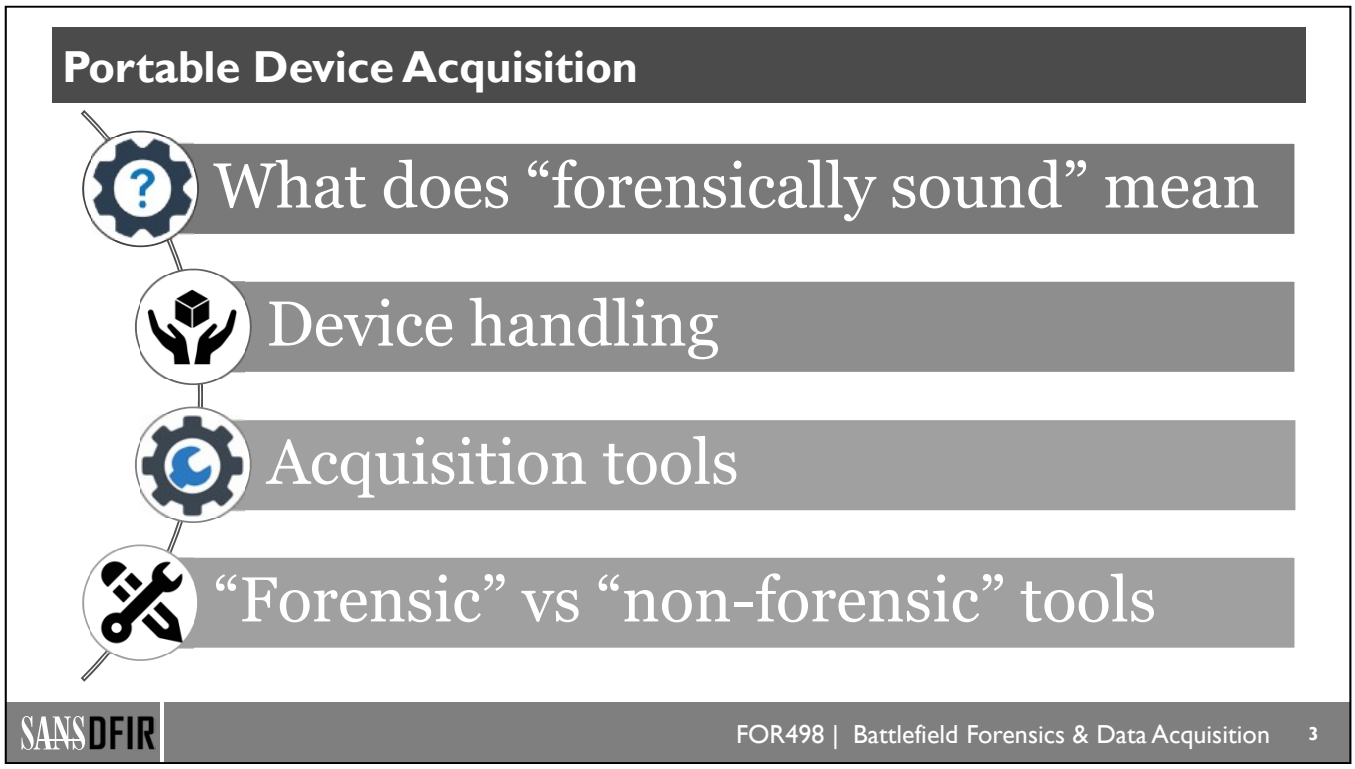
2.2 Portable Device Analysis

2.3 Acquisition Hardware & Software

2.4 Acquisition Methodology

2.5 Discovering & Interacting with Data

This page intentionally left blank.



This page intentionally left blank.

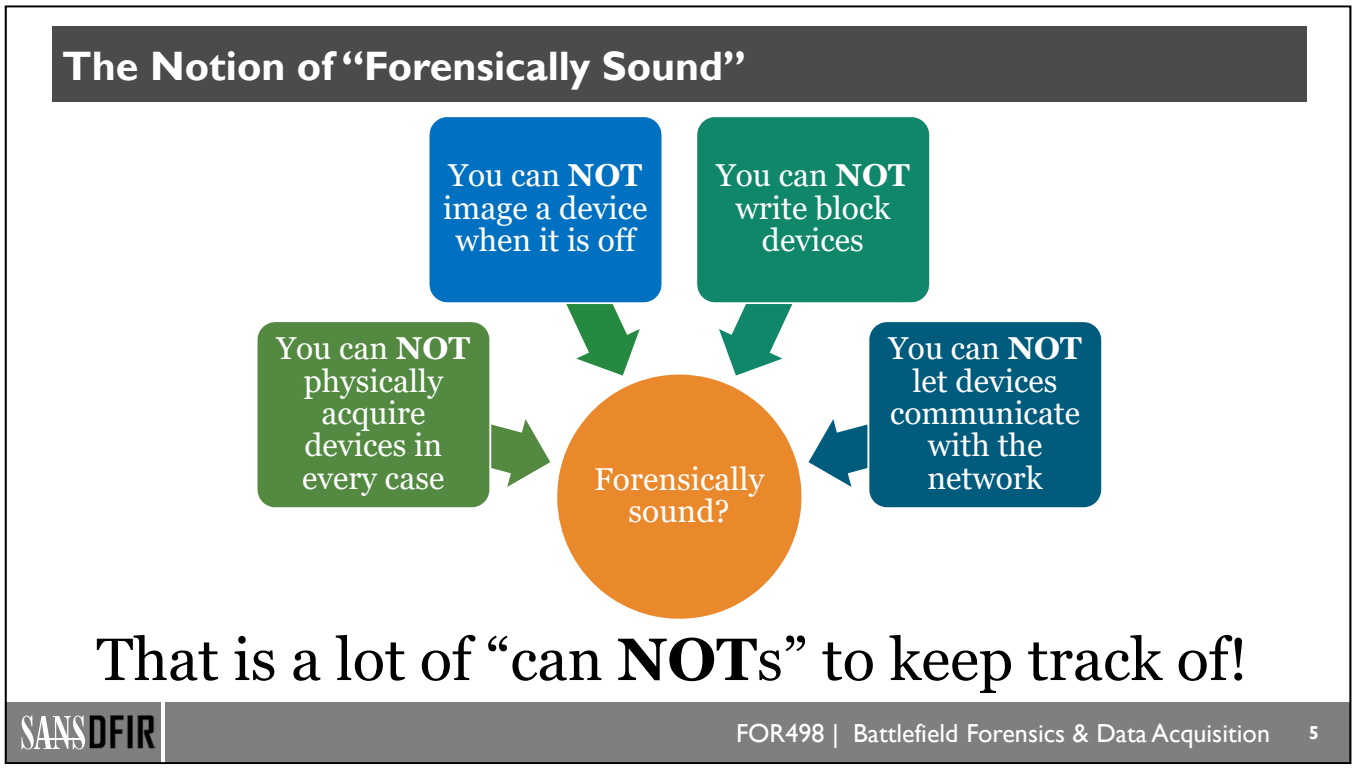
Portable Devices

You want me to image what?!

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 4

With the ubiquity of portable devices, it will be extremely common to have them in an investigation, even when traditional computers are non-existent. Certainly in many cases, your first contact in an investigation may be a cellular device, and proper triage will assist in leading you to other devices that you may not otherwise have known about. To be clear, when we say, “portable devices”, we are referring to cellular phones, smart phones, and devices such as iPads and Samsung Galaxy devices, to mention just two.

Over the years, portable devices have increased in performance and storage capacity, and adopted new connection types as well as different operating and file systems. Because there is such a wide range of portable devices out there, this is one of the more challenging types of devices to deal with.

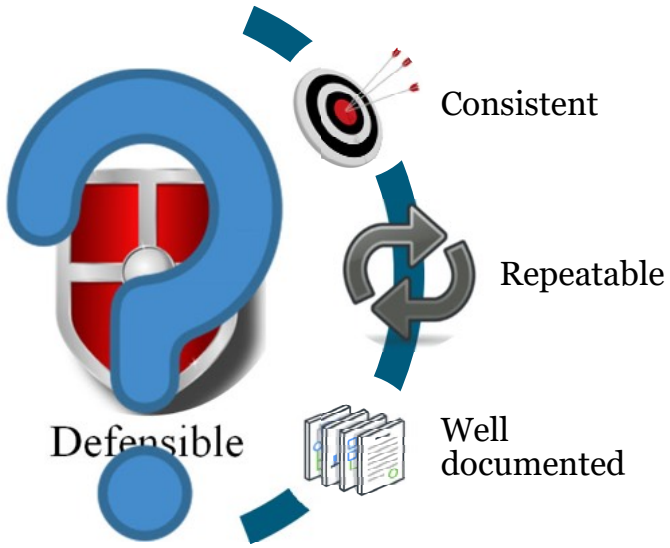


Best practices in portable device acquisition have certainly created a number of arguments in the forensic community. The thought is that we must always handle the evidence in a "forensically sound" manner. But what does that mean?

When anyone says they created a forensic image of a smartphone, how did they go about it? Two near universal truths are the fact that you cannot image a portable device when it is turned off, and you cannot write block it because the acquisition devices and software often times must communicate bi-directionally with it to perform the imaging process.

"Forensically sound" collection generally means that the data was somehow protected from change or alteration DURING the acquisition phase. In order for this to happen, the subject media needs to be write-blocked during acquisition.

What Is "Forensically Sound"?



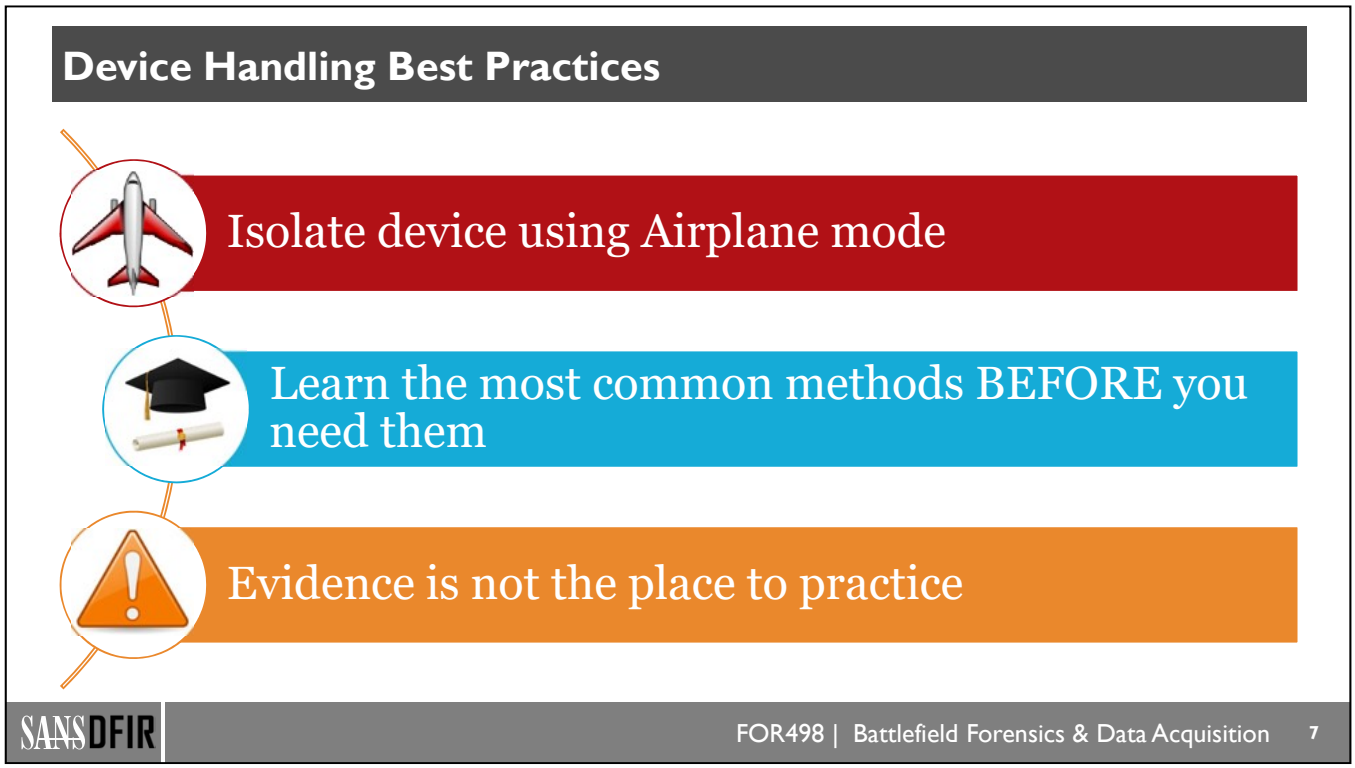
In most cases, do these concepts apply to portable device acquisition?

To be forensically sound, a data collection process must be defensible, meaning that it is consistent, repeatable, and well documented. A forensically sound data collection process should be accompanied by an audit trail describing every step that was taken in collecting electronically stored information (ESI). The process should be subject to authentication, or proof that the collected data is the same data that a litigant used, unchanged from its original state. In short, the entire data collection process should be correct and explainable so that it can withstand scrutiny in a court of law [1].

Because this is not possible with a portable device, it cannot be said that portable device acquisitions are forensically sound [2].

[1] Forensically sound data collection | `https://for498.com/pe1jn`

[2] Acquisition comparison methods | `https://for498.com/gd9v6`



What we CAN control is how the device is handled from the moment of seizure, through the acquisition process. Proper intake and device handling are key in being able to maintain best practices and minimize the amount of change to the subject device.

The first and most important task that needs to be performed is to isolate the device from the cellular, wireless, and Bluetooth networks it may have a connection with. This is typically (but not always completely) achieved by placing the device into airplane mode. In the vast majority of cases, devices can be placed into this mode even if the device is locked with a passcode.

If the device is not isolated at the time of seizure, there are any number of things that can happen. New data can be introduced, as well as data being removed. Add to this the ability of someone to send a command to the device to start a wiping process, and it is easy to see why this function simply cannot be ignored.

WARNING

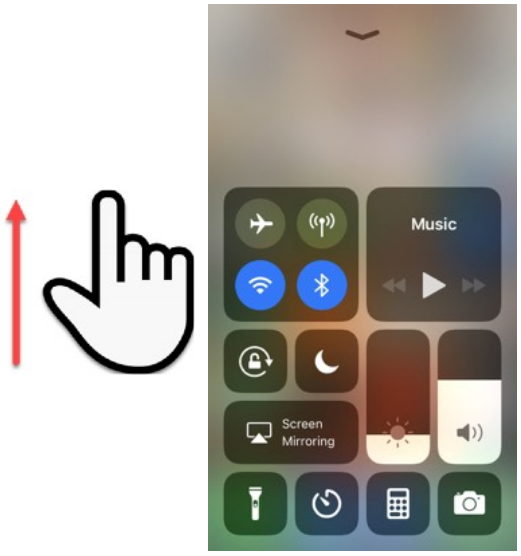
We cannot possibly account for a methodology that works for all devices, nor could we even address all the major ones here.

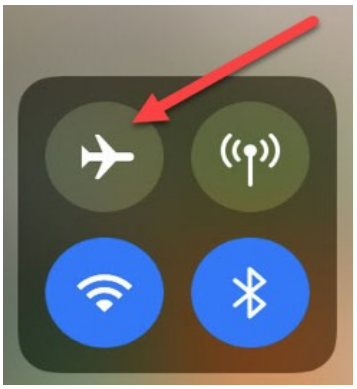
If the methodology provided does not work for the device you have seized, or you have a brand other than Apple, Android, or BlackBerry, you MUST find a method for that device as soon as possible.

While portable device acquisition has come a long way, it is not a 100% solved problem. Legacy devices, cloned devices, and a myriad of other factors introduce many variables that make this as much art as it is science.

When the tools and techniques you have fail due to one or more of these variables, it is necessary to forge a new technique or fall back to a more "brute force" like method. For example, if acquisition fails for a given device and no solution is in sight, your only alternative may be to interact with the device and take photographs of each screen that needs to be documented. While tedious, it is sure to work for just about any device that you can gain access to.

Apple Isolation



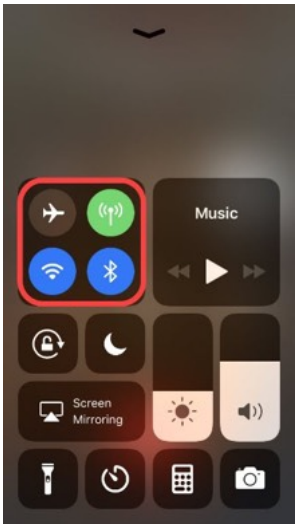


Clicking on the Airplane Mode icon places the device into airplane mode

Verify for your iOS and device, as this can differ

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 9

For Apple devices, the most efficient way to place the device into airplane mode is by swiping up from the bottom of the screen[1].

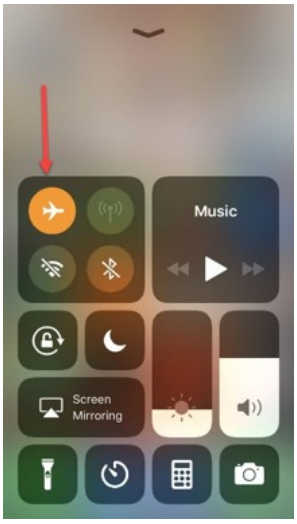


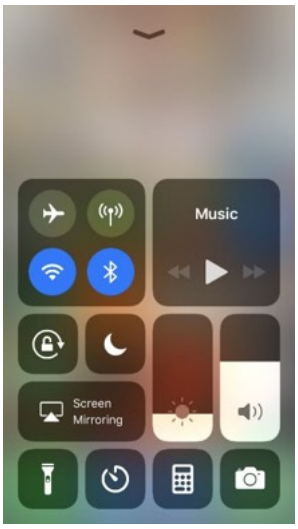
You will be presented with a screen called the Control Screen. Note the "Connections" box on the screen. You will see four icons.

Top left is the Airplane Mode button
Top right is the cellular connection button
Bottom left is the Wi-Fi connection
Bottom right is the Bluetooth connection

Clicking on the Airplane Mode icon will place the device into, you guessed it, airplane mode.

Verify that the other three icons in the Connections box are no longer colored. Further, the Bluetooth and Wi-Fi icons have strikes through them.





The examiner must be aware that even if Wi-Fi and Bluetooth were disabled at the time of placing the device into airplane mode, they will automatically become enabled when turning off airplane mode. Testing has also shown various combinations of this.

If you are unable to obtain the Control Screen by swiping up, it could be that the feature has been disabled in the settings of the device and cannot be performed without first bypassing the access code. In fact, this is a common issue with iPads.

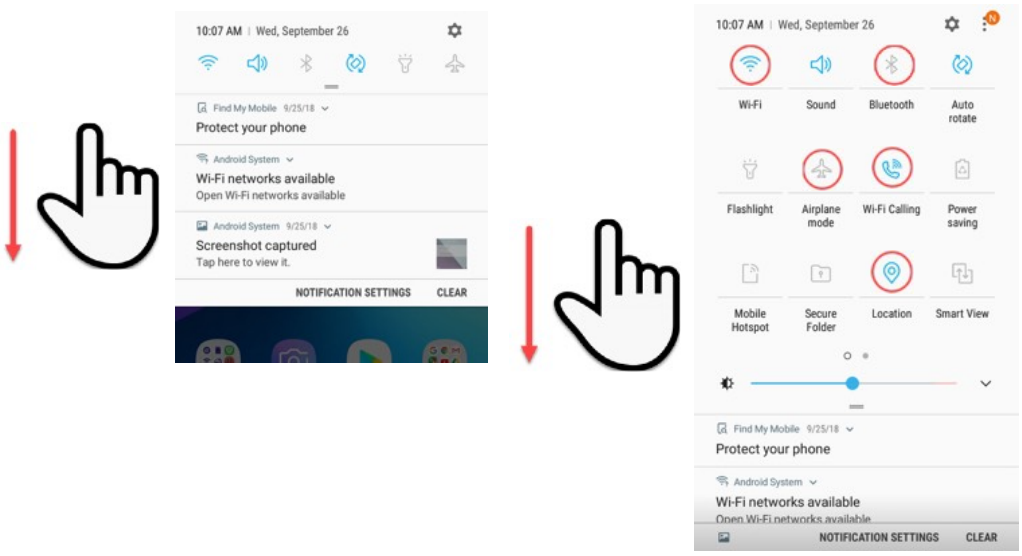
It is imperative that you verify the method for a particular device, as it can change from device to device, even if the iOS is the same. For example, on an iPad Air 2 running iOS 12 and higher, the Control Center is accessed by swiping from the top down in the upper right of the screen. Alternatively, on an iPhone 7 Plus running the same iOS 12 version, you access the Control Center as described in the slide.

In more recent version of the iOS, it is important to note that Airplane Mode and Flight Mode (observed on newer Android devices) are synonymous. Flight Mode is often being seen on newer Android devices, so the examiner must be aware of the terminology used between different eco-systems, even if they mean the same thing.

It is also very significant that turning Airplane Mode/Flight Mode (AM/FM) off may not necessarily turn off Wi-Fi or BT connections. These may have to be done separately from placing in AM. Double check that these other connections are also turned off once AM/FM is invoked.

[1] Airplane mode | `https://for498.com/y-umx`

Android Isolation (1)

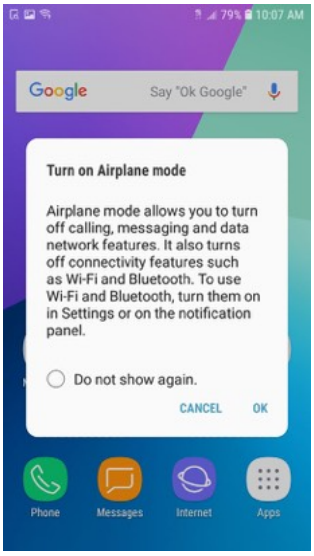


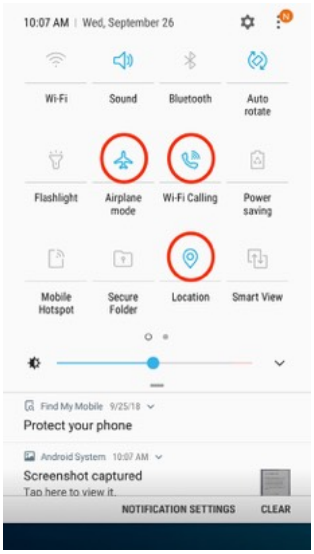
For Android devices, the instructions follow. When talking about swiping down, we mean putting your finger at the top edge of the screen and swiping your finger downwards until a menu is pulled down. In some cases, the down swipe will cause the full menu to show, and in the example above, the down swipe will have to be done twice.

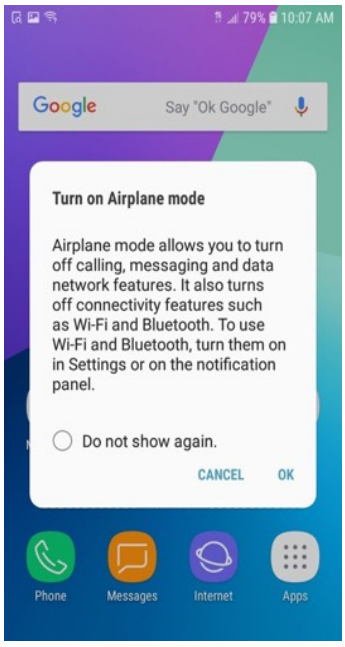
Swipe down on the Android screen as shown. A menu screen will appear, covering half the screen of the device. Swipe down a second time.

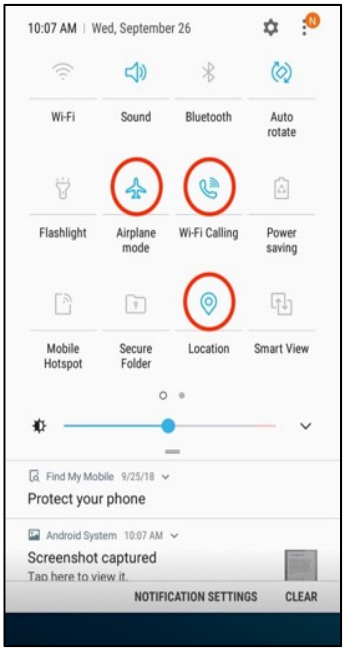
A menu appears covering the full screen. There are six items you potentially have to be aware of. Five are shown in the image above, and the sixth will be shown next. The five noted in the photo are Wi-Fi, Bluetooth, Airplane mode, Wi-Fi Calling, and Location. The first thing to do is tap the Airplane mode icon to place the device in airplane mode.

Android Isolation (2)



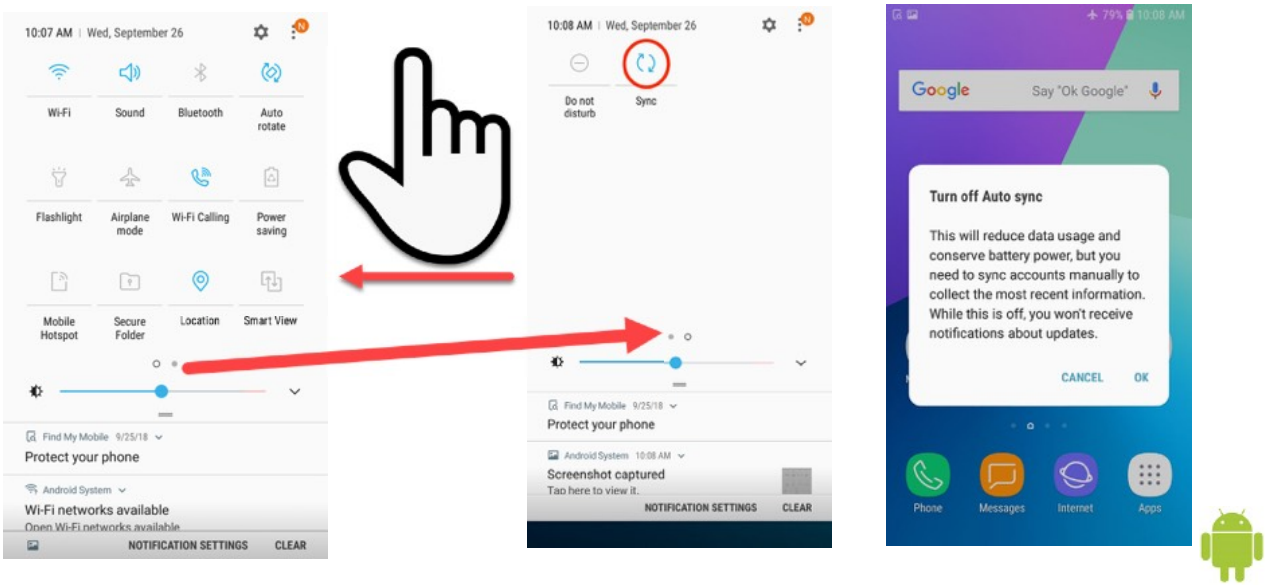






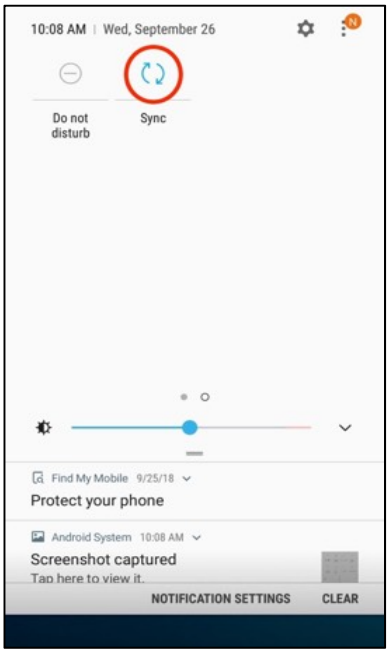
You may receive a warning message. If you do, click OK, and you will be returned to the screen on the right. If you are not given the menu screen, simply swipe downwards twice again. Observe the changes that have been made. As indicated in the screenshot on the right, the Airplane mode is now enabled. Check the icons previously indicated to see if activating Airplane mode turned them all off. Chances are that it did not. Manually disable any of the other listed icons, if they are not already. In the example on the right, we see that Wi-Fi Calling and Location were not disabled by activating Airplane mode. Tap on them to disable them.

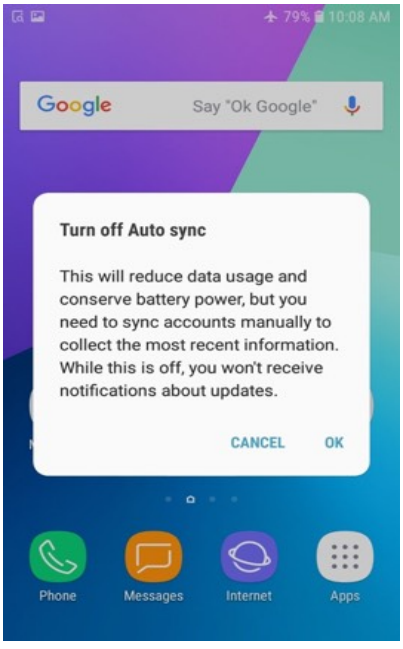
Android Isolation (3)



SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 13

Once the previous step is complete, the only network related icon that should be highlighted is Airplane mode. Now on to the sixth icon we mentioned earlier. Swipe your screen from the right edge of the device to left. This takes you to the second page of options (if available). Notice how the small dot at the bottom of the icons changes as you swipe left. You should see the screen as indicated in the following screenshot. Because Android can be customized by the end user, you may see these icons on different locations than shown above. When this happens, just search for the icons and/or their descriptions to ensure you interact with the settings as outlined.





Once done, you should see an icon labeled Sync. Tap this to disable it. You may receive a confirmation message as shown in the screenshot on the right. Click OK.

Faraday Isolation

Sometimes, Airplane mode is not an option

Faraday Bag

Faraday Tent

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 14

If you are unable to perform network isolation through the use of the device's airplane mode, other measures will need to be employed. The most commonly used method of isolation is to place the device in a "Faraday" bag. A Faraday bag is designed with Radio Frequency (RF) shield lining built in. Although this isolates the device during the time it is properly sealed in the bag, the device may still be exposed to signals while an examiner opens it to attach acquisition cables. Some environments will utilize a Faraday cage to ensure complete isolation during any work on a device that cannot be isolated [1].

After either using Airplane mode, or employing a Faraday device, you now have to decide the next step. This may involve imaging immediately, or shutting it down for transport. Consider that if the device has a passcode in place, you need to disable the passcode immediately, if possible. Typically, the imaging process will cause the device to require confirmation of the passcode, and this would end the good luck in finding an unlocked phone!

[1] Examples of Faraday devices | `https://for498.com/l6nz1`

Portable Device "Forensic" Tools

Once the device is properly seized and isolated, the next step is acquisition. There are a number of different tools for the task, with varying features, and certainly varying prices!

Cellebrite [1] is a widely used tool within the law enforcement and investigative community. It comes in a standalone hardware device, or a software package that requires a host computer or laptop. Both configurations allow for extraction of resident data, as well as an "add-on" package that will allow the examiner to acquire deleted data from certain devices. It will acquire and analyze data, as well as create reports, and allow for the forensic handling and verification of a dataset, once it has been collected.

The Cellebrite kit will typically come with a myriad of cables to address many different types of devices. It also contains adapters and devices to read SIM (Subscriber Identification Module) cards.

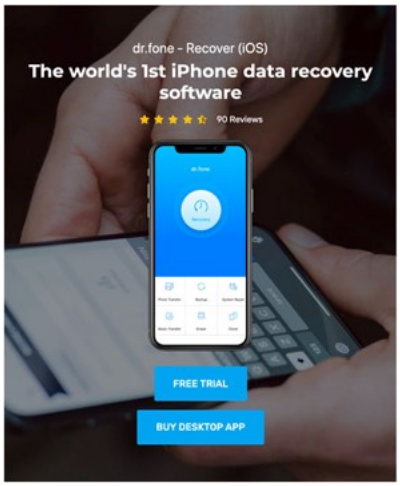
Magnet's IEF/Axiom [2] is another very widely used tool and is equally as effective with portable devices as it is with traditional computers. Its power lies in its parsing abilities. IEF/Axiom will ingest most any data set and look through its structure for anything recognizable. Its ability to parse complex data structures gives it abilities that few other tools can match. Magnet Acquire is the free acquisition tool that will acquire data from portable devices, as well as computers.

We will be using both tools later in the course.

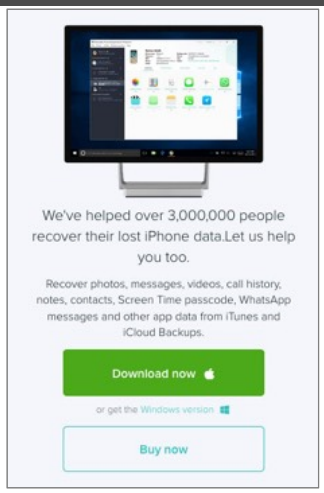
[1] Cellebrite | `https://for498.com/2msbl`

[2] Magnet | `https://for498.com/7z8l6`

Non-Forensic Tools (iDevice)



dr. fone
Recover



iPhone Backup
Extractor

Although the previously mentioned tools are quite effective in their abilities, they also come with a price that puts them out of the reach of many. There are also tools available that allow for gaining information from devices, however they do not have the end to end continuity previously mentioned. They were designed to provide the ability to move data from a portable device to a computer. In some cases, these applications can even recover deleted data. This last point is typically limited to text and iMessages.

Their price point is significantly lower, but so is their ability to interact with data in the way that the more expensive tools do. Having said that, it is very interesting to note that for many times when a new OS was released, it can take major tools sometimes weeks to be able to properly support them, and yet the budget tools have support the next day!

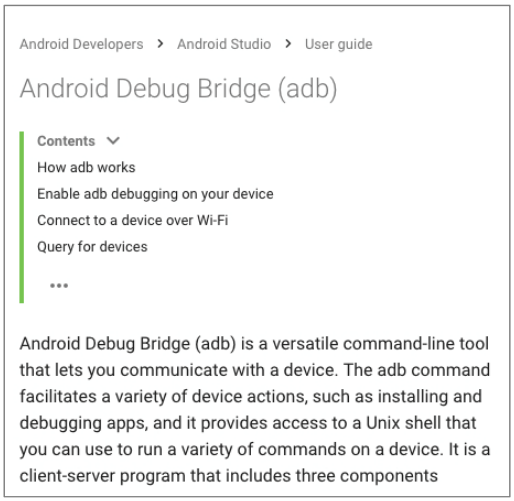
It goes without saying though, that it may be more difficult to use the budget tool findings in a court proceeding. In any case, regardless of the tool used, writing full, detailed, and extensive documentation of the process you followed will never hurt you.

Some very effective tools for iOS device examination are listed here and come in Windows or Mac versions.

- dr.fone [1]
- iPhone Backup Extractor from Reincubate [2]

[1] dr.fone | `https://for498.com/5y2i4`
[2] iPhone Backup Extractor | `https://for498.com/tp8rf`

Non-Forensic Tools (Android)



In the same vein, Android also has its tools, and they also come in Mac and Windows versions.

- MobiKin Doctor for Android [1]
- Android Debug Bridge [2]

[1] MobiKin Doctor | `https://for498.com/ps5uz`
[2] Android Debug Bridge | `https://for498.com/9ow7q`

Surprising Findings (1)

iPhone Backup Extractor
1874 messages

Another interesting thing to note is the difference in the amount of data that is sometimes recovered. In many cases, the budget tools have recovered more deleted messages than their more expensive counterparts. In any event, it is never a bad idea to verify your findings, as those findings may be surprising.

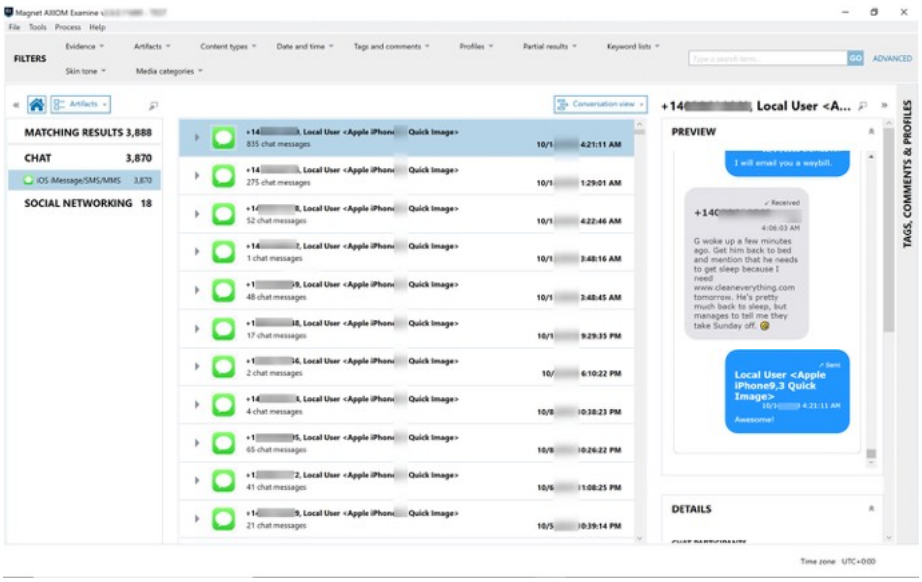
We used an iPhone as a test device, and used Cellebrite, Axiom, dr.fone, and iPhone Backup Extractor to parse its data, in an attempt to extract resident and deleted data from SMS (Short Message Service), MMS (Multimedia Message Service), and iMessages. The findings were as follows:

iPhone Backup Extractor	1874 messages
Axiom	3870 messages
Cellebrite	621 messages
dr.fone	1924 messages

It is important to note that these numbers are based on how a particular tool interprets data and counts a message. Having said that, there were certainly instances where a tool had recovered a message that another tool had not. The above statistics are not to be interpreted as showing one tool's superiority over another. They are simply to show that different tools will provide different results for various different reasons.

A proficient examiner will be aware of this, and not rely on any one tool, because, at the end of the day, the best tool for the job is the one that finds the evidence you need to make your case.

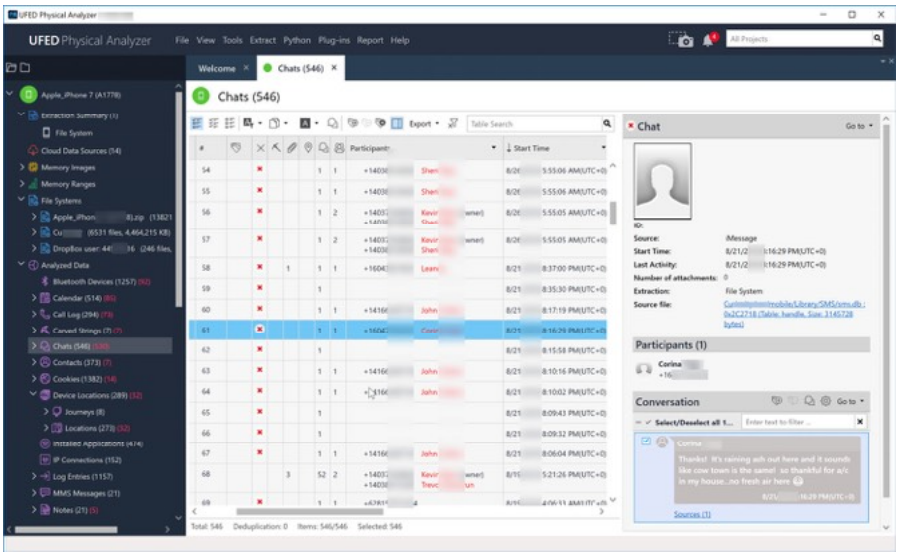
Surprising Findings (2)



Magnet Axiom
3870 messages

Magnet Axiom findings.

Surprising Findings (3)



Cellebrite UFED 4PC
621 messages

Cellebrite UFED 4PC/Physical Analyzer findings.

Surprising Findings (4)

1785		unknown158	7/28/	Received	Yes! Where are you going to be exactly and when?
1786		unknown153	7/28/	Send	Hey brother. Do we have a game plan for tomorrow? I know logistically it may not be possible_
1787		unknown162	7/27/2	Received	We're practicing.
1788		unknown157	7/27/2	Received	Perfect brunch starts at 11 but we can start at any time although there will be several of us I an
1789		unknown149	7/27/2	Received	I need to look after myself for a change as well. And my wife. And maybe a youngster.
1790		unknown143	7/27/2	Received	Thanks brother. You know I've been working for them for over 15 years. I just made the plans fo
1791		unknown138	7/27/2	Received	I hope you can come to Del Rey yacht club.
1792		unknown156	7/27/2	Received	My niece Mallory is in town Martin is in town and the family is in town it is Olivia's birthday pa
1793		unknown148	7/27/2	Received	And I can stay near Neil.
1794		unknown137	7/27/2	Received	I have 150 restaurants hotels and casinos. And 15K employees under me.
1795		unknown160	7/27/2	Received	Real job. Real office real benefits etc.
1796		unknown155	7/27/2	Received	Officially two months ago.
1797		unknown127	7/27/2	Received	Check your email.
1798	Lov	125	7/27/2	Received	Hey there_ are you guys home? I'd told Sheri I would try to FT tonight. I was at work early this r
1799	Lov	125	7/27/2	Send	They are at home. I am still in DC.
1800	Lov	125	7/27/2	Received	Ok. Are you ok?
1801	Lov	125	7/27/2	Send	Yeah fine. Just tired and need a break.
1802	Lov	125	7/27/2	Received	That makes 2 of us!! I miss you very much
1803	Lov	125	7/27/2	Send	Miss you too....gotta get out there for a weekend with G!
1804	Lov	125	7/27/2	Received	Just say when! 😀😘
1805	Lov	125	7/27/2	Received	Cool message! I take whatever I can get whenever I can get it!
1806	Lov	125	7/28/	Send	You around for Thanksgiving weekend?

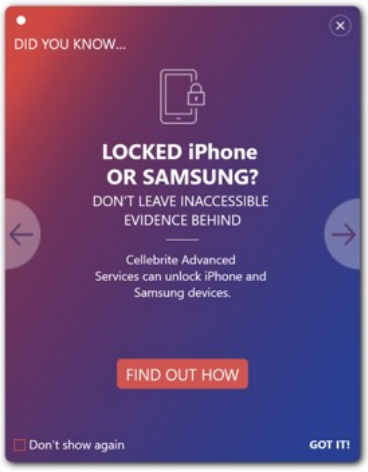
Messages

dr.fone
1924 messages

dr.fone findings.

Locked Devices

GrayKey device from Grayshift



Cellebrite Advanced Services

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 22

There are a great deal of arguments surrounding portable device encryption, and the ability to access these devices for the purposes of investigation. Although the San Bernardino shooting in 2015 was one of the most publicized cases of conflict between law enforcement and Apple, it certainly was not the only case. The argument surrounded law enforcement's inability to access an iPhone seized from the suspects, and Apple's position to not unlock the device for the FBI.

While the debate between the need for timely access to potential evidence vs. the demand for privacy rages on, various companies work to provide for the needs of law enforcement while the portable device manufacturers attempt to create ever more secure devices.

Two of the most popular options on the market are GrayKey from a company called Grayshift, and Cellebrite Advanced Services. Both are positioned to provide in-house services for law enforcement and government agencies, while the Cellebrite product will offer services to private enterprise in certain circumstances.

The data recovery company DriveSavers advertises that they will unlock any device of any version, with a "no unlock, no pay" policy, however it is quite expensive! Information would suggest that they are also not doing the work in-house. It would seem they send the device to an independent third party for the actually unlocking and data dump.

IMEI: International Mobile Equipment Identity

- Unique identifier for the device
- Device can be disabled by provider using this number in case of theft
- IMEI ties device to provider, and therefore, subscriber

The International Mobile Equipment Identity (IMEI) number[1] is a number much like a serial number. It is assigned to a device, and then tacitly assigned to the user. In the case of device theft, a user can call the provider, and the provider can disable the device permanently so that it cannot be used by anyone.

The IMEI on some devices can be found on a label under the battery, on the back of the device, or possibly printed on the SIM tray. If there is an IMEI value on the SIM tray and the device is accessible, check device settings to make sure the IMEI values match. It is disturbingly common, specifically in certain types of iPhones, that the IMEI value on the tray does not match the IMEI from the device. On devices without removable batteries, such as Apple devices, the IMEI can be found in the "Settings -> General ->About" option.

Another method of gathering the IMEI if the phone is on and accessible, is to proceed to the telephone number entry screen and enter *#06# (star-pound-zero-six-pound) into the keypad.[2] This will display the IMEI value of the device. Understand that commonly, this technique will not work on Apple devices that have been placed in Airplane Mode.

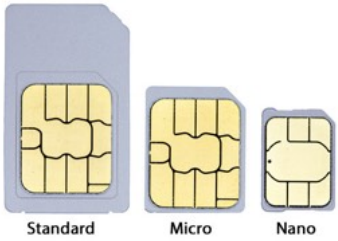
[1] IMEI description | `https://for498.com/1cvhk`

[2] Showing IMEI on phone screen | `https://for498.com/hk10x`

SIM Cards & Reader

SIM card reader from Cellebrite

3 most commonly used SIM cards.



SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 24

SIM (Subscriber Identification Module) [1] cards are the key to being able to use cellular technology today. The SIM card is essentially what causes the cellular device to connect to a cellular network. Not every type of device requires one, but the overwhelming majority today do.

The criticality of the SIM card data is in the information that it contains. A SIM card contains unique information about the mobile network the device is connecting to. This unique information is tied to the user, is issued by the provider, and identifies a device on the network. In other words, it authenticates and identifies a subscriber on the network [2].

The SIM card contains the following critical pieces of information [3]:

Integrated Circuit Card Identification (ICCID) – The SIM card's unique identification number
International Mobile Subscriber Identity (IMSI) – The telephone number assigned to the SIM by the network

The SIM card "may" include security information in the form of a Personal Identification Number (PIN) and a Personal Unblocking Code (PUC) which is only usable to unlock certain data areas of the SIM card that are inaccessible during a SIM card extraction. This PIN is typically set by the device user. The PUC can only be provided by the mobile network service provider.

In the case of older devices and many Android devices, the SIM card may also contain data such as contact lists, call logs, and text messages.

It is not hard to see why the information on a SIM card is important to an investigation, although on many smart phones today, the information on a SIM card can also be derived from the device itself.

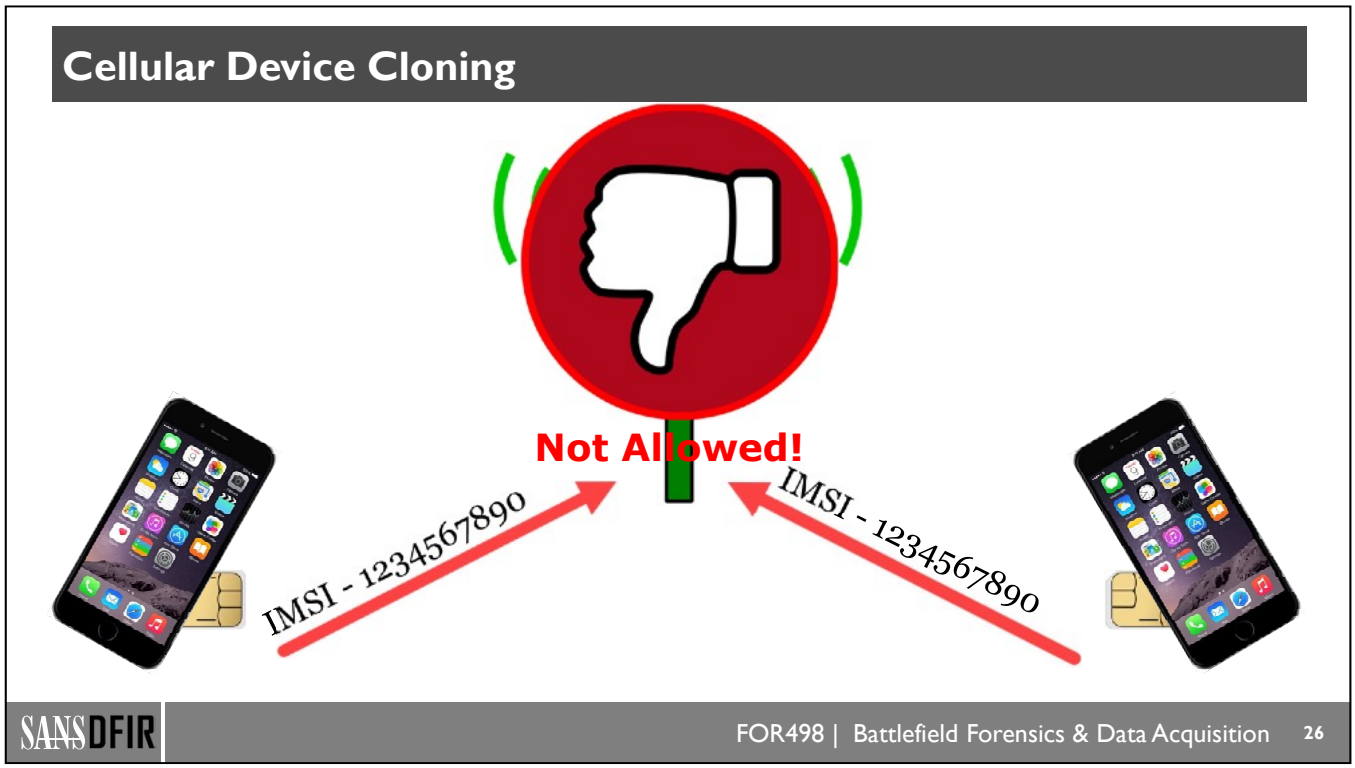
From SIM to eSIM

Newer smartphone devices from 2018 onward are migrating to, or adding eSIM (embedded SIM card) capability. [4]

If you come across a device that you can easily identify as being from 2018 and newer and it does not appear to have a SIM, keep this in mind.

eSIM will give customers the ability to more easily change providers, as well as pave the way for dual SIM capability. This means you could have more than one phone number on the device at the same time [5]!

[1] What are SIM cards | `https://for498.com/t405k`
[2] SIM details | `https://for498.com/8of9g`
[3] SIM card contents | `https://for498.com/d56g4`
[4] eSIM details | `https://for498.com/5iteb`
[5] eSIM capabilities | `https://for498.com/snpur`



It is fairly common to hear about the notion of device cloning [1]. As in many things, often the story does not survive scrutiny.

Is cloning possible? Yes, but only in very specific circumstances. With that said, it is not something that can be performed by some cheap tool from the Internet.

The cloning of a device is creating a situation where two devices identify on the network using the same identifiers. Should this happen, the customer gets the bill for all activity from both devices, and they may not realize this is happening until they receive the bill. Historically, this has typically been done by criminals to evade detection or steal service. Kevin Mitnick committed crimes through the use of cellular cloning, back when it was usefully possible. He has written books and become a very successful speaker regarding his escapades, showing that sometimes crime DOES pay!

In general terms, device cloning was prevalent back in the days of analog signal networks (1G) [2] and early Code Division Multiple Access (CDMA) [3]. Since the transfer to digital signal technology, the ability to clone devices has essentially been eliminated.

In order for a device to be cloned, the second device would have to have the same SIM card data. This will cause, among other things, two different devices to be on the same network at the same time with the same IMSI. The network will almost instantaneously detect this anomaly through comparison of this data (SIM card data) against each device's radio fingerprint [4], and kick both devices off the network.

[1] Device cloning | `https://for498.com/0irq4`
[2] Analog signals | `https://for498.com/3dkil`
[3] CDMA | `https://for498.com/e7h3u`
[4] Radio fingerprinting | `https://for498.com/q5wev`

Summary

- There are an almost endless number of device types
- “forensically sound” acquisition is not possible
- Isolating device from communication is priority number 1
- Tools are not built equally
- Sometimes the cheap\free tool is the best\only solution

This page intentionally left blank.

Exercise 2.1A-B

Portable Device Acquisition

OPTIONAL, OUT OF CLASS EXERCISE

Synopsis: In this exercise, you will use a program called Physical Analyzer to perform the acquisition of an Apple device or you will use a program called UFED4PC to perform an acquisition of an Android device.

This page intentionally left blank.

FOR498.2: Portable Devices & Evidence Acquisition Agenda

2.1 Portable Device Acquisition

2.2 Portable Device Analysis

2.3 Acquisition Hardware & Software

2.4 Acquisition Methodology

2.5 Discovering & Interacting with Data

This page intentionally left blank.

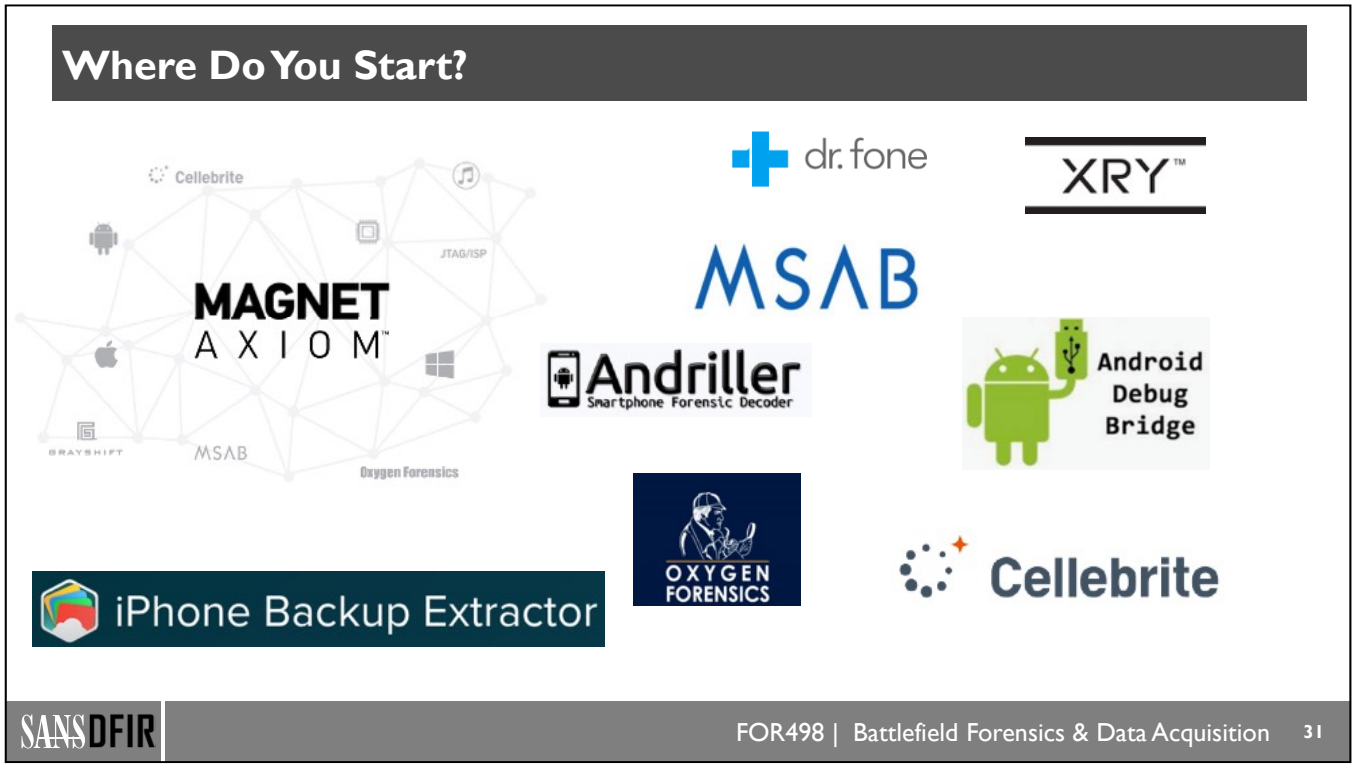
Portable Device Analysis

Analysis Tools

Quick Win Data

Apple vs Android Analysis

This page intentionally left blank.



You have seized the phone and in the best of all scenarios, you have been able to create a working image of the phone. Given that smart phones contain not only the fastest, but also the most personal and immediate of all user data, it is not hard to see the importance of being able to gain access to this information early in an investigation.

Although every effort should be made at the moment of seizure to create and preserve an image of the device for evidentiary purpose, this may not always be the best approach. Where life risk might be a factor, imaging a device is a luxury the examiner may not have.

Even once an image of the device has been created, this still may not be the best, or most rapid method of analysis. It may take quite an amount of time to extract a database from a smart phone image and parse it into usable information. Alternatively, looking at the device itself and clicking on an icon to review data may take seconds. As with the handling of all devices, the situation will dictate the approach, and the person holding the device will have to defend their actions.

No matter what approach is taken, the first and most important action by the first responder must be to isolate the device from all types of networking connections. Keep in mind that airplane mode is not enough. There are also Bluetooth and wi-fi to consider.

The next issue would be determining how best to access the data. There are so called forensic tools that are the preferred method, however due to things like cost and availability, these may not be an available choice. There are a number of low-cost tools that have the ability to perform rapid analysis of devices, however they would not be considered forensically sound (whatever that means to portable devices).

DANGER!

Anything you do (including nothing) may have an adverse effect on the device you are analyzing.

The mere act of tapping the screen on a phone in preparation to unlocking it is already creating changes within the data on the device. Anything you do with the device and any type of interaction you have will change data. This also includes doing nothing. As long as the device is connected to cellular, Wi-Fi, or Bluetooth, changes can be made to the device remotely.

Simply picking up the phone and tapping on icons to look at the data within may render evidence inadmissible, if it doesn't accidentally destroy it first. This is not to suggest that a first responder shouldn't do this, but the first responder needs to understand that they will have to answer for their actions at some point. Where life risk is at stake, the actions may be perfectly acceptable.

There is no standard operating procedure when dealing with cellular devices other than one that might be created within a certain department. Even then, not every eventuality can be covered by such a document.

Analysis with Cellebrite

Cellebrite is one of the leading tools in the smart phone forensics field. It has the ability to allow for a quick look at a device, a deep dive into the device, or even the ability to remove passcodes from devices where it is believed to not be possible.

Whether it be UFED4PC, Physical Analyzer, or Cellebrite's hardware device, the first responder has the ability to access data in minutes depending on the collection method.

Cellebrite can analyze and extract data from many of the most common smart phone apps, however its effectiveness is dependent on the make and model of the device in question.

Analysis with AXIOM

Axiom is another leader in the data analysis workspace. Its ability to address and parse data from an incredibly wide array of devices makes it a versatile tool in any triage kit.

There is no single tool that does it all. In any given circumstance the preferred tool may not be the tool that the examiner has. Axiom gives the analyst the ability to examine smart phone data sets from an extremely wide variety of devices, and extract data, both deleted and resident, that many other solutions simply do not have the ability to do.

Axiom is relatively intuitive to use, and its acquisition interface gives the examiner the ability to target certain types of data in order to rapidly zero in on whatever data is immediately necessary. One of the really great features of AXIOM is its ability to ingest datasets created by a large number of competing software solutions.

Device/User Information

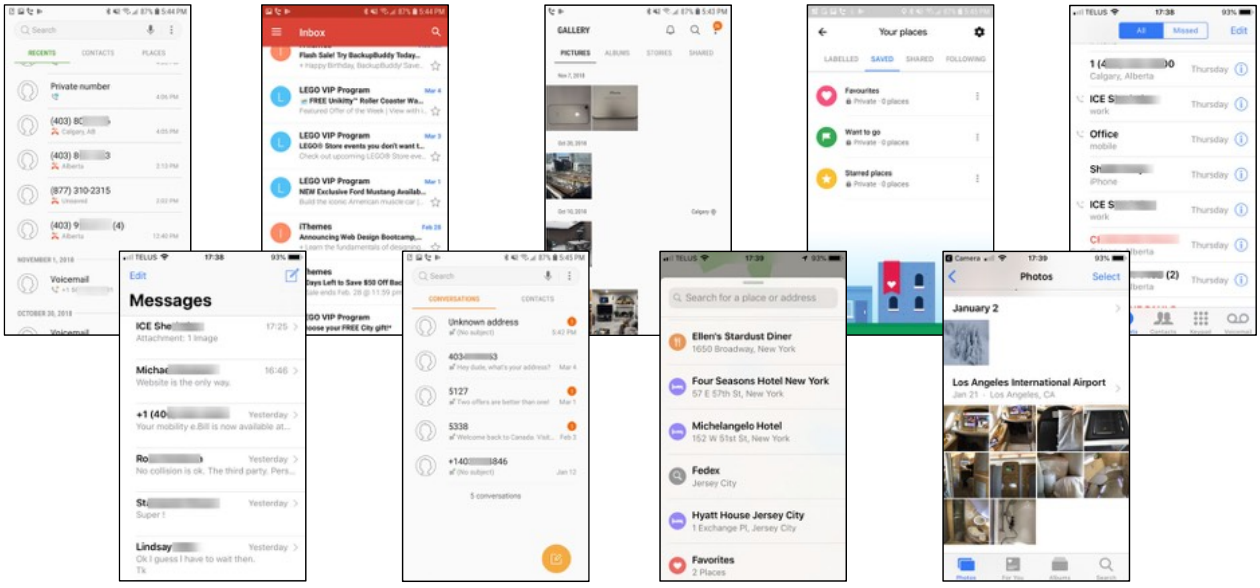
- Account ID (Gmail address or Apple ID)
- Username
- Backup locations
- iCloud
- Information sharing

A great deal of information can be extracted from a device that may be useful to an investigation but does not reside within the applications on the device. While not being able to consider every eventuality, there may be times when it is not immediately obvious who the owner of the device is, either because the subject is not forthcoming or quite frankly, the subject cannot be forthcoming. As a result it is necessary to examine the device for things like account IDs. Depending on the device this may come in the form of a Gmail email address or an Apple ID. Various usernames, backup locations, and other information within the device settings can assist in rapidly moving the ball forward in an investigation.

The settings of the device could also be used to outline backup locations for various data on the device giving the examiner potentially another avenue of investigation. Certainly in the case of iPhones it can quickly be determined whether or not data is being uploaded to the iCloud. Settings will also show any privacy or location details that apply on a "per app" basis on the device. Where cellular data usage is concerned, the examiner can also see very quickly what apps are in use and how much data they are using, or have recently used.

By this point it should go without saying that most if not all of this is impossible without the appropriate credentials to access a device or its backup.

Quick Win Data



This module is all about "quick win" data when it comes to portable devices. The purpose of this course is to allow for rapid and surgical extraction of data to further an investigation without having to wait hours, days, or even weeks before proper analysis can be done.

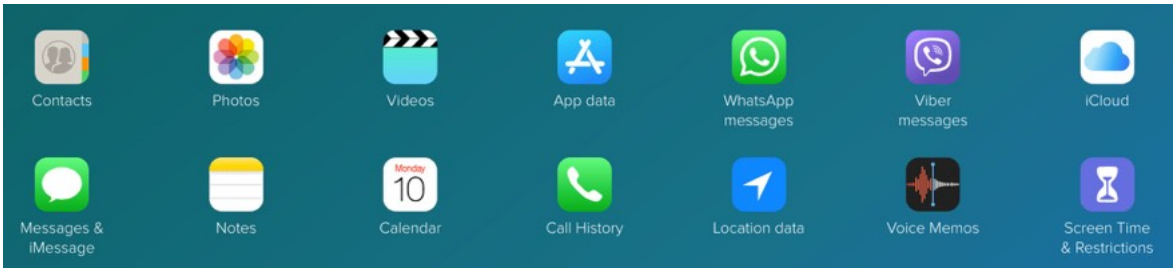
Although the "what if" game could be played forever, there are a certain few pieces of data within the smart phone that most everyone will agree can be considered immediately useful in the vast majority of investigations.

Things like call logs that will show calls that were received and made may quickly identify other people of interest to an investigation. Many people use various types of different chat applications thinking that their messages are either anonymous, encrypted, or both. Having said that, it is quite common to find all types of chat activity residing within the default applications on the device. Thinking outside the box slightly, if we want to know the movements of the subject, we can use mapping data, GPS data, location data, and geolocation data in photographs on the device. These would definitely be considered critical data for the rapid progression of an investigation.

Once again we cannot possibly imagine all of the scenarios that might play out, and any given investigation may be aided by information gathered from other areas on the device. It is the first responder's job to understand these possibilities, and have a mechanism in place to rapidly target locations on the device. As previously stated, this may simply be interacting directly with the device.

Non-Forensic Solutions

- Will create very quick backup of device to dump data
- Often can repair corrupted backups
- Accesses newest iOS much quicker than expensive tools



iPhone Backup Extractor

dr.fone

There are a number of tools available to the examiner for little or no cost. Especially with the iPhone there are a few options that can be used besides expensive tools. First of all, in order to do most any form of iPhone analysis, whether it be from the device itself, or from a backup of the device, you must have iTunes installed on your machine. To add a level of difficulty, Windows 10 introduced some new features that have caused the downloading of iTunes to happen differently than it did in the past. By default, Windows 10 will offer you the iTunes program from the Windows store. The most significant difference between iTunes from the Windows store and iTunes from the iTunes.com website seems to be that the iTunes download from the Windows store will never prompt the user to update the program. But there are clearly other differences. In testing with a number of iPhone analysis tools, none would work with the iTunes version from the Windows store. Some were very clear about this issue, and others just gave very cryptic error messages. For some reason, Windows does NOT want you downloading iTunes from anywhere other than the Windows store. The full version (the one you want) can be found here. [1]

Many of the low-cost tools that are available will have modules for both the iPhone and Android devices. Some of them will also recover deleted data. In the majority of cases, this software was never designed for forensic analysis. It was designed to assist the user in transferring data from an old device to a new device, allow for the transfer of data across different platforms, or allow the user to recover accidentally deleted information.

Most of these tools are quite economic in cost and the vast majority have a “try before you buy” option. This option allows for full functionality of the software but potentially limits the amount of data that can be extracted, or the amount of time that you can use the tool for free. It gives you enough time to see if the software will work for your situation.

Viewing the recovered data can be a challenge all its own. When analyzing an iPhone at the file system level, you will encounter an overwhelming amount of plist files. A great tool for mounting and viewing the data in these files is called plist Editor Pro. [2]

Although you may have difficulty introducing evidence in court based solely on the findings of such tools, there are other reasons why you may want to use them. As has been stated a number of times to this point, validation and correlation are cornerstones of any examination.

Maybe your forensic tool is not giving you the information you expect, or any information at all. It may be a case where the tool cannot handle such a new operating system, or it may be that it is not presenting you the data you seek as intuitively as possible. It can be a great idea to use a low-cost tool to possibly find data you're looking for and then correlate it with a forensic tool for submission as evidence. In a case where the low-cost tools will allow you access to an operating system when the forensic tools will not, these low-cost tools can help further your investigation much more rapidly. Lastly, these tools can help you verify your findings from a different tool.

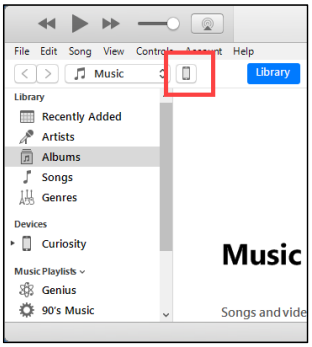
No tool does it all. In fact some just simply give BAD data. This applies both for expensive forensic tools and low priced or free tools. No matter what tool you are using, one of the key factors in becoming an effective examiner is in understanding these tools. Whether it be an expensive forensic tool or a free or inexpensive tool used for other purposes, the only consistency is their inconsistency. You can use the same data set in five different programs and get five different results. This is why the examiner should absolutely know their tool and be proficient with it. Simply downloading a tool and planning to use it is a recipe for failure.

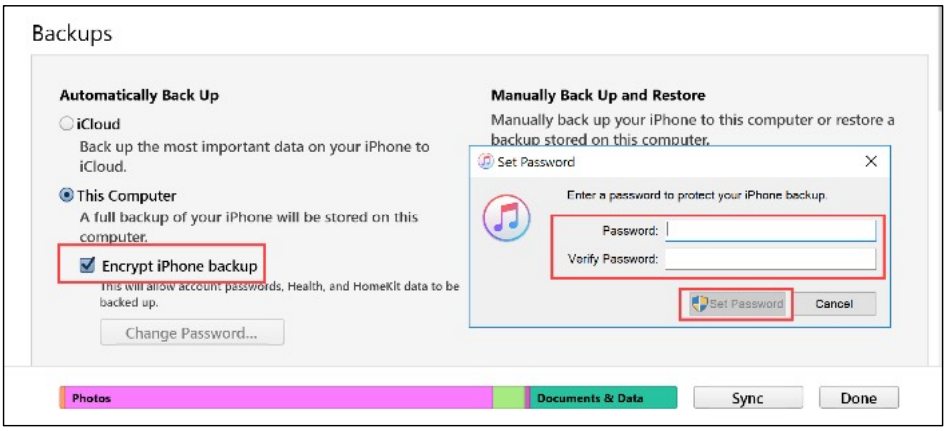
To be effective you must know what a given tool looks like under controlled circumstances. This means using test devices with known information that has been created on the device, then looking at this device with more than one tool to see how the data is presented. In our previous module we noted differences in the data even between two expensive forensic tools.

So why should we use free or inexpensive tools for smart phone examination in an investigation? As regards forensic tools, the standard is incredibly high in terms of releasing an update to the program. Because of this, a great deal of testing must occur before any new releases can happen. Low cost utilities that were never intended for forensic use do not have to meet the same standard, and as a result they will tend to be released much more quickly when a new smart phone operating system has been released. It is quite common upon the release of a new version of iOS, for example, for low-cost tools to be able to access and parse the data within days. It can often take forensic tools weeks to be able to access the same data set, let alone parse it. Data availability per tool changes almost daily. A simple update on a device can cause any given tool to stop working properly. This is another reason to have more than one tool in your toolbox.

[1]”Full” Apple iTunes Download | `https://for498.com/itunes`
[2] plist Editor Pro | `https://for498.com/8w3po`

Encrypted Backup

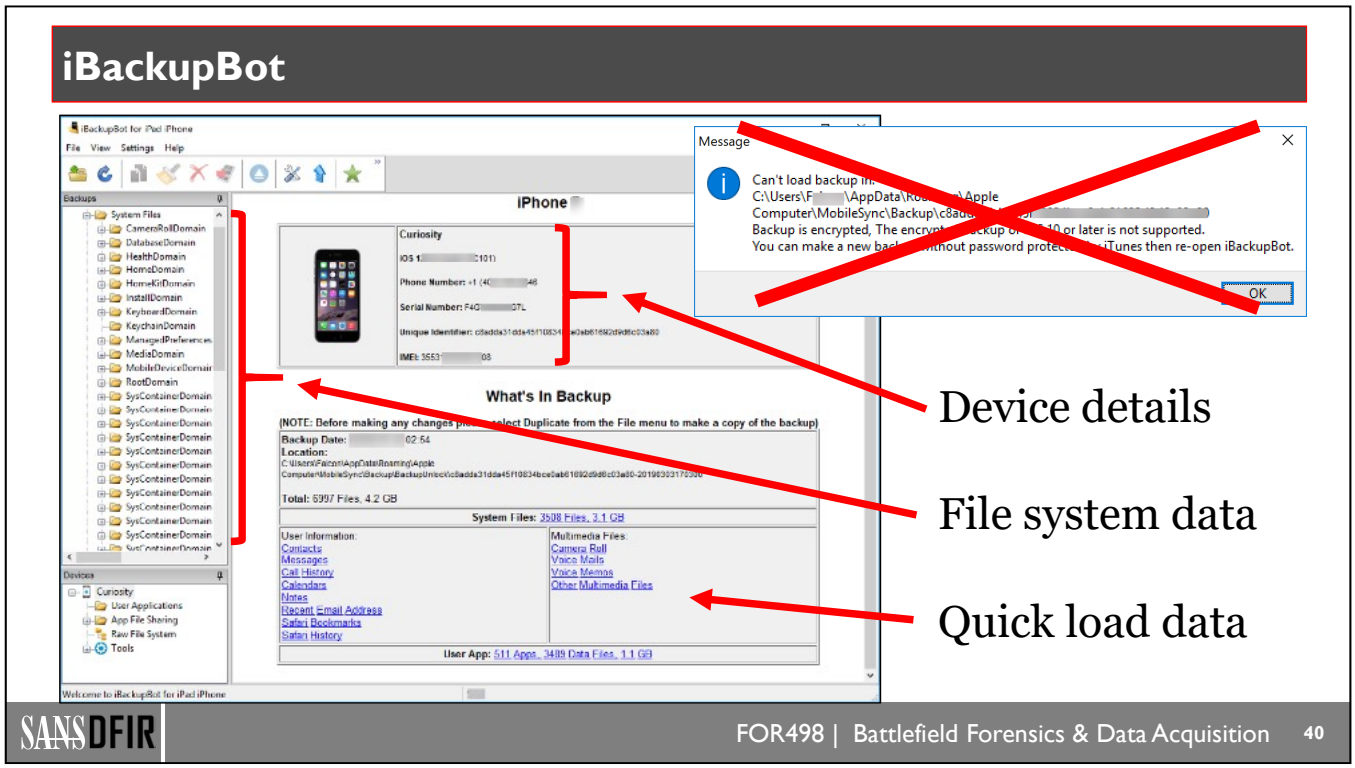




In a perfect world we would always have the ability to capture a full physical dump of all data on a device. Sadly this is often not possible. The next best option is a file system dump. Many tools that create this dump cannot access and extract a good portion of data unless the backup that has been created is an encrypted backup. As mentioned earlier, at least with iOS devices, a backup of the device is created, and the program works from this backup. Most any tool used for iOS analysis and/or recovery is using iTunes to create this backup.

An encrypted backup will allow the examiner access to many secure areas of the device that contain potentially sensitive information, such as passwords, chats, and any other areas that Apple sees fit at any given time.

You can let the software create the backup for you or you can use iTunes to create the backup as well. If using iTunes to create the backup, you will see the option to encrypt the iPhone backup and you will be prompted for a password. You can create any password you want however ensure that you remember the password because you will need it perform the analysis.



iBackupBot [1] is a "try before you buy" program that will parse the backup of an iPhone. Note that it will not work from the device itself. Use iTunes to create a backup, or iBackupBot can create a backup as well.

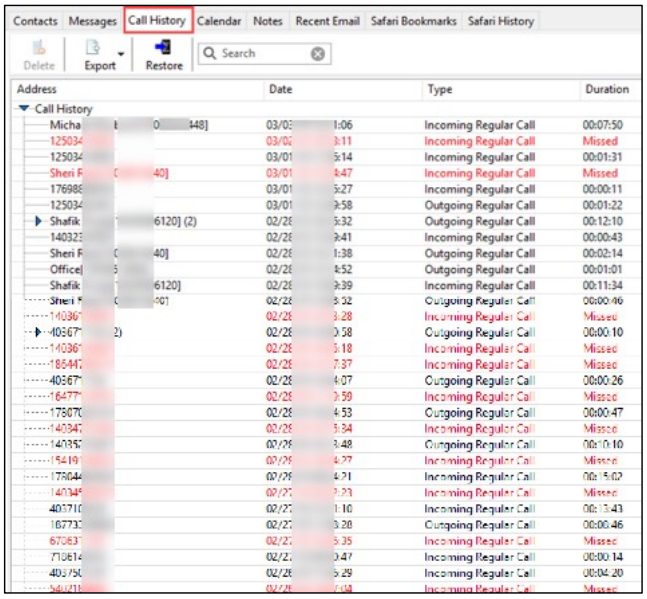
Since approximately iOS version 10, iBackupBot does not play well with encrypted backups, and has no mechanism to decrypt encrypted backups. When trying to load an encrypted backup, the program will present errors and tell you to redo the backup without encryption. DON'T! In other words do not use iBackupBot to create your backup. It would be recommended to use iTunes so that you can create an encrypted backup.

iBackupBot is a great tool when it comes to analysis though, which is why we continue to use it. But if it creates an error message when trying to read an encrypted backup, and it does not provide you with a mechanism in which to decrypt the encrypted backup, how can it be useful? The forensicator can use a tool called AnyTrans by iMobie. AnyTrans is a great tool by itself for analysis and or deleted recovery of data on iPhones and Android devices. Before using it on an Android, understand that it is seriously invasive to the device, and is not recommended in a forensic analysis capacity. However, it will also take an encrypted backup and decrypt it. We can then use this decrypted version of the backup to analyze within iBackupBot.

[1] iBackupbot | `https://for498.com/vsinc`

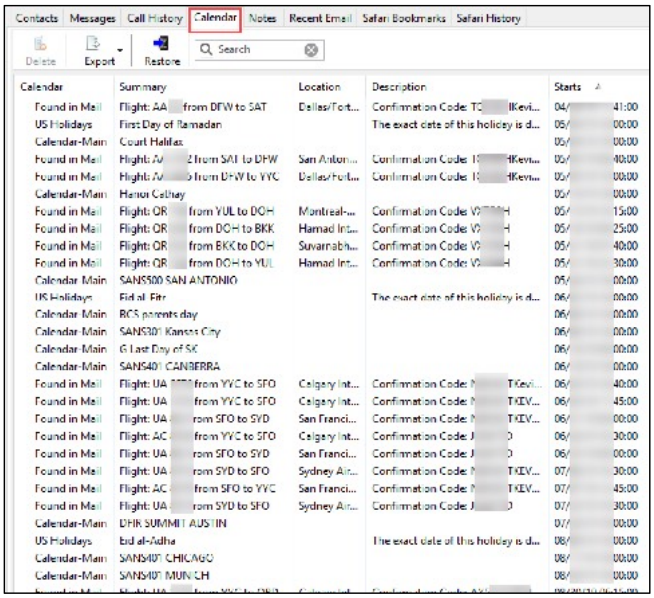
[2] AnyTrans | `https://for498.com/9wq45`

Call History



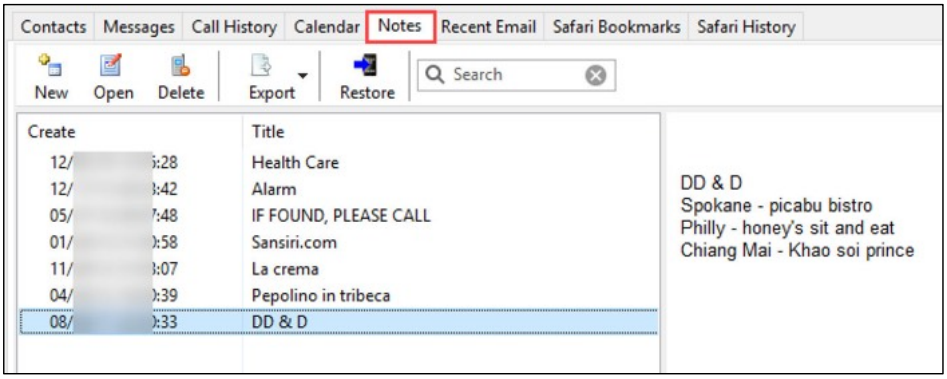
iBackupBot allows you to see some "quick win" data in terms of clickable links for some of the most commonly used artifacts such as call logs, chats, etc. In the example above, call history is being displayed, however you can see from the tabs along the top of the box, the types of data you can access. Not all of them will contain data. For example, on an iOS device you will not get recent emails from newer filesystems.

Calendar



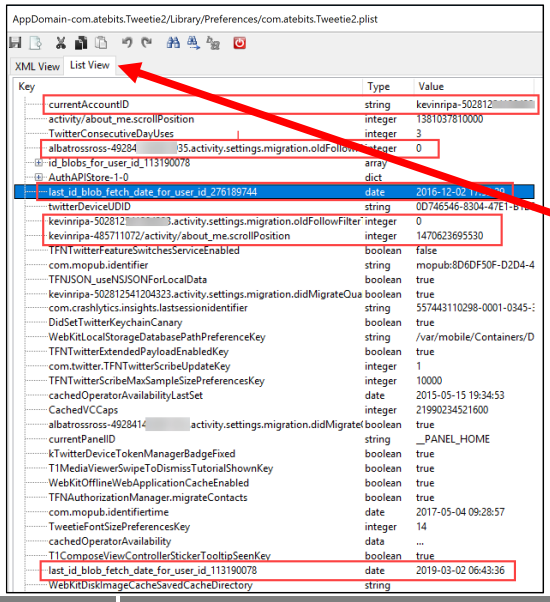
Details from the Calendar tab are displayed with iBackupBot.

Notes



Details from the Notes tab are displayed with iBackupBot.

Twitter (1)



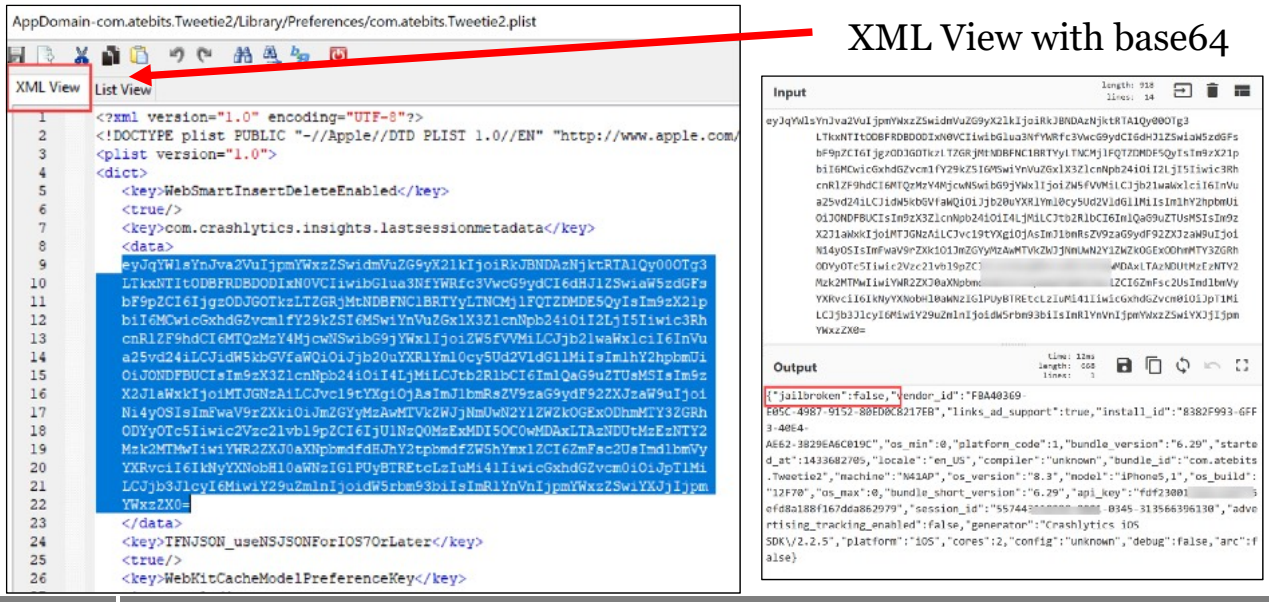
AppDomain/com.atebits.Tweetie2/Library/Preferences/com.atebits.Tweetie2.plist

Note the two views. List view is cleaner, but may hold less data

Twitter account information being shown with iBackupBot. Note that the List View tab is selected.

`AppDomain/com.atebits.Tweetie2/Library/Preferences/com.atebits.Tweetie2.plist`

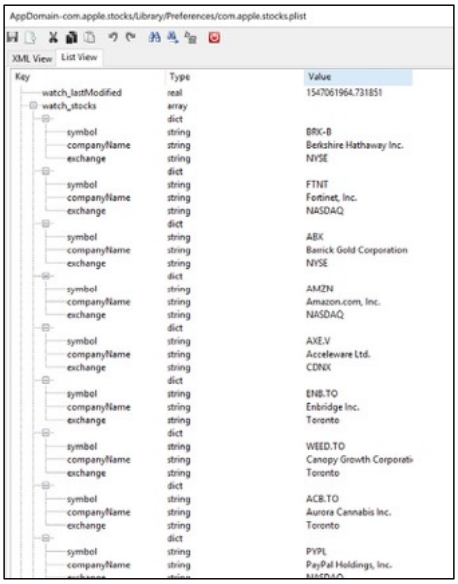
Twitter (2)



In the XML View tab, there may be important data that is not seen in List View. Essentially, XML view shows all data, and List View shows clearer, more usable data. In the example above, the base64 data holds information including whether or not the device is jailbroken. Extract this base64 data and translate it using a tool like CyberChef. [1]

[1] CyberChef – The Cyber Swiss Army Knife | `https://for498.com/hd2un`

Stock Listings



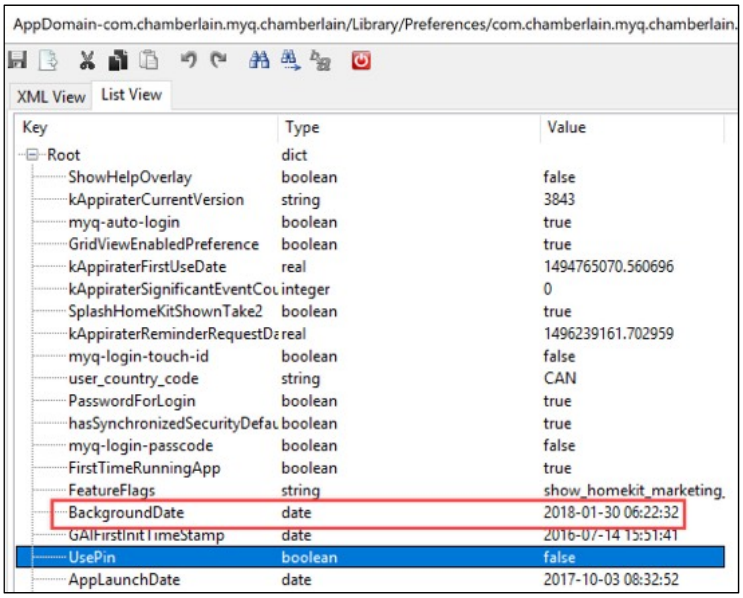
AppDomain/com.apple.stocks/Library/
Preferences/com.apple.stocks.plist

Note: Seeing entries here does not mean these stocks are held

Although many artifacts may hold data, the examiner must be very careful about how she interprets this data. In the above case, the stocks have merely been added to the app. It does not make any determination as to whether the user owns these stocks.

`AppDomain/com.apple.stocks/Library/Preferences/com.apple.stocks.plist`

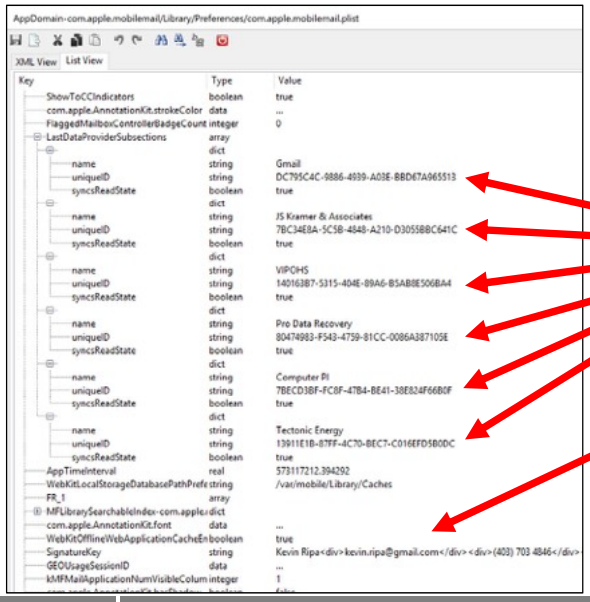
Home Automation



This data is from the Chamberlain garage door opener app. Obviously this will not be on all devices; however this is a clear example of not knowing what information might be available until you start looking. The highlighted data shows the last time the app was used to open a garage door.

Do not jump to conclusions! This does not mean that the device was at the door when the door was opened. The app can open the door from anywhere in the world.

Mail Account Names



AppDomain/com.apple.mobilemail/Library/Preferences/com.apple.mobilemail.plist

Listing of email accounts by name

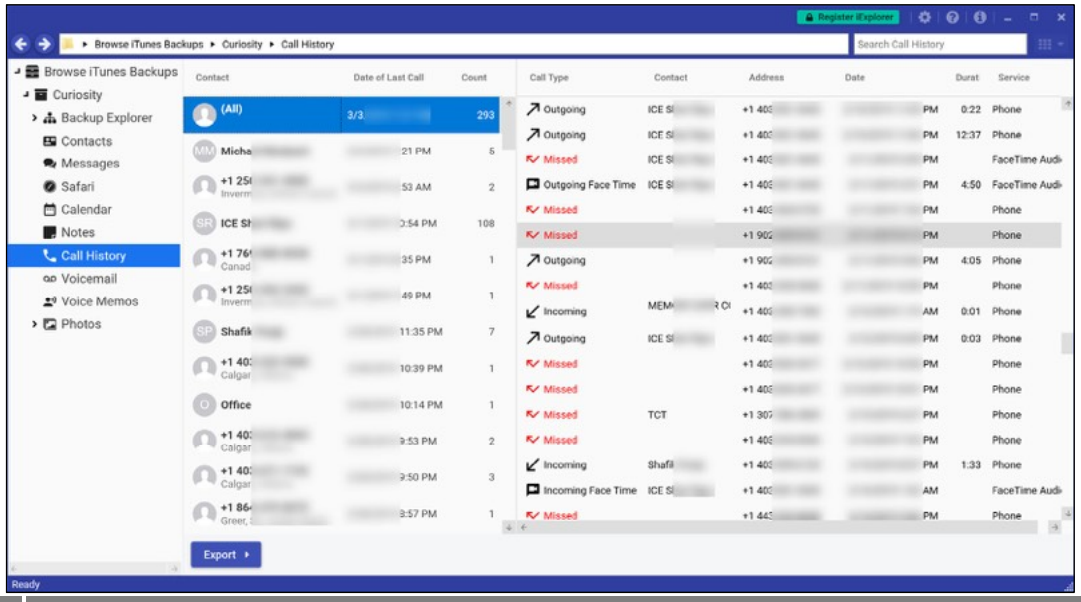
Mail signature block

Mail account names are available from a plist file on an iPhone. This does not list the email addresses themselves, although they are available in a different plist on the device. This listing merely shows the name a user has given to a particular email account. Various testing has shown that in the case of multiple email accounts, not all email addresses may be available in a single plist. This mobilemail.plist shows all of the account names though, so that the examiner will at least know how many accounts are available on the device as compared to how many email addresses they are actually seeing.

The SignatureKey listing in this plist will show the information that is in a signature block that goes out with every email composed on the device. This information in and of itself can prove very useful for an examiner.

`AppDomain/com.apple.mobilemail/Library/Preferences/com.apple.mobilemail.plist`

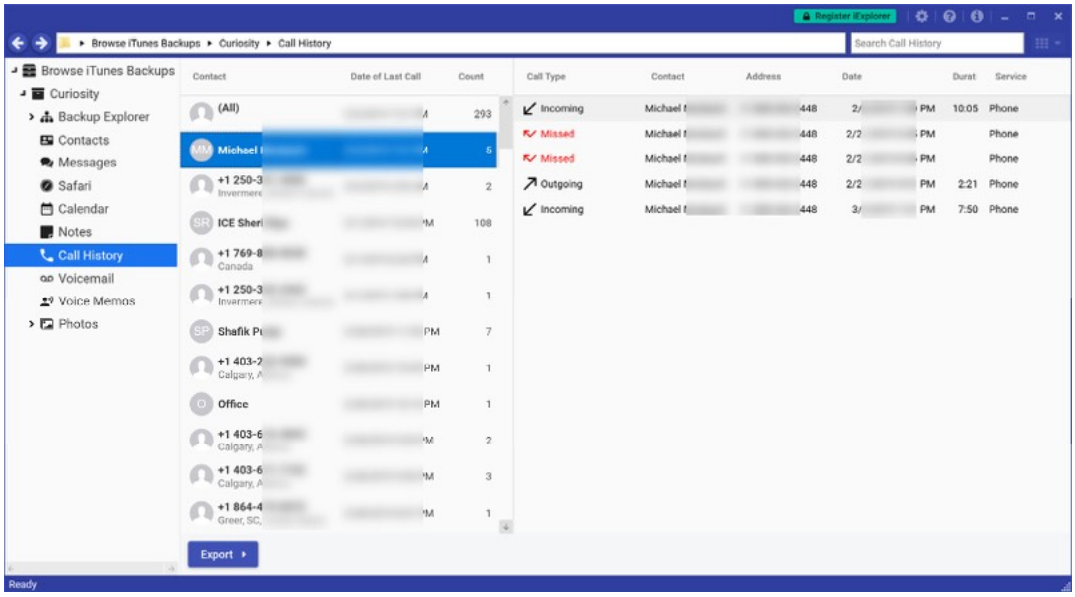
iExplorer Call History (1)



The program iExplorer [1] from a company called Macroplant is another tool that can be used to review an iPhone backup. As can be seen by the slide, the output is displayed differently than the previous tool. Depending on the information you are looking for, this can be helpful as it may be clearer than viewing in another tool. iExplorer can work with encrypted backups, and is a "try before you buy" tool.

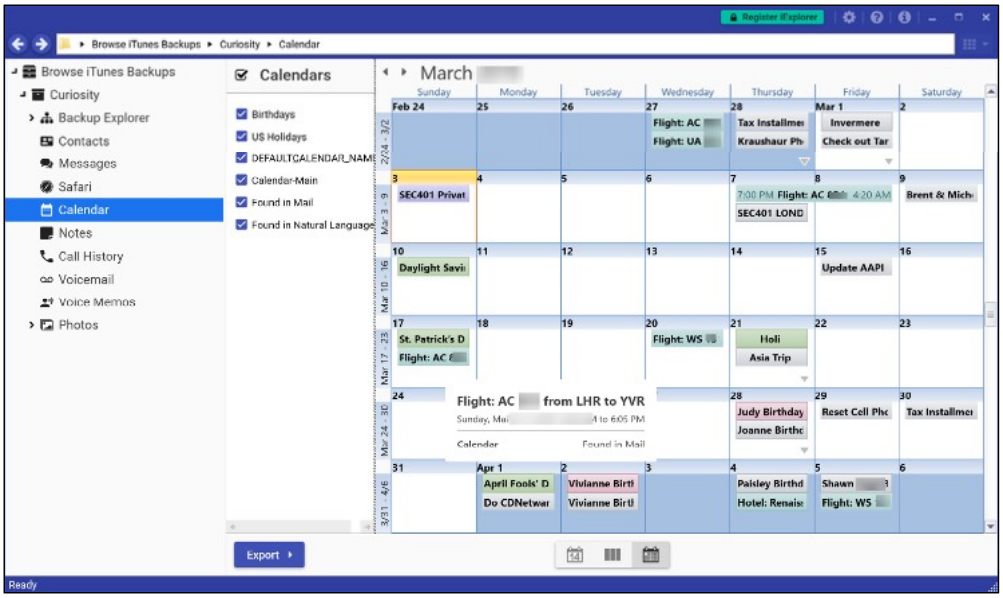
[1] iExplorer | `https://for498.com/jayh0`

iExplorer Call History (2)



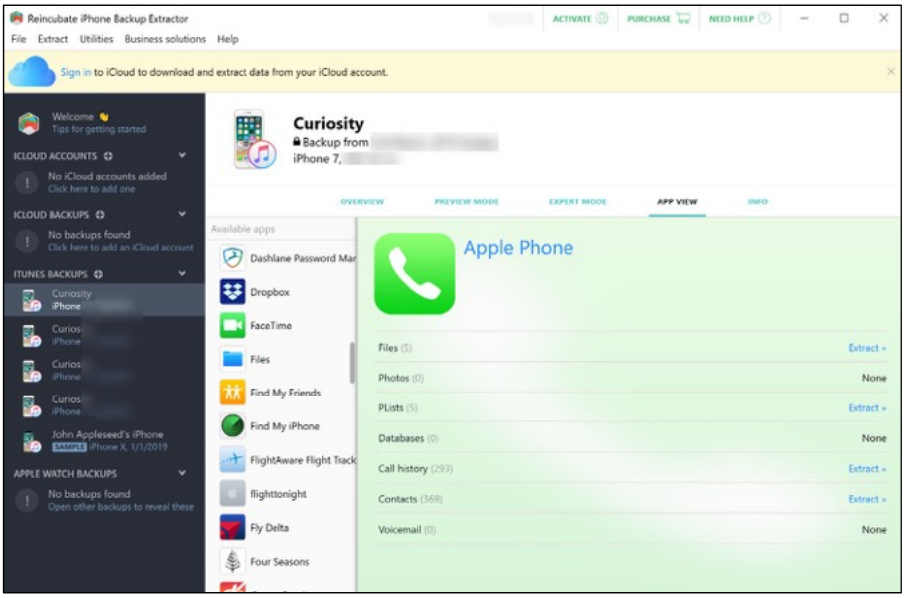
This page intentionally left blank.

iExplorer Calendar



This page intentionally left blank.

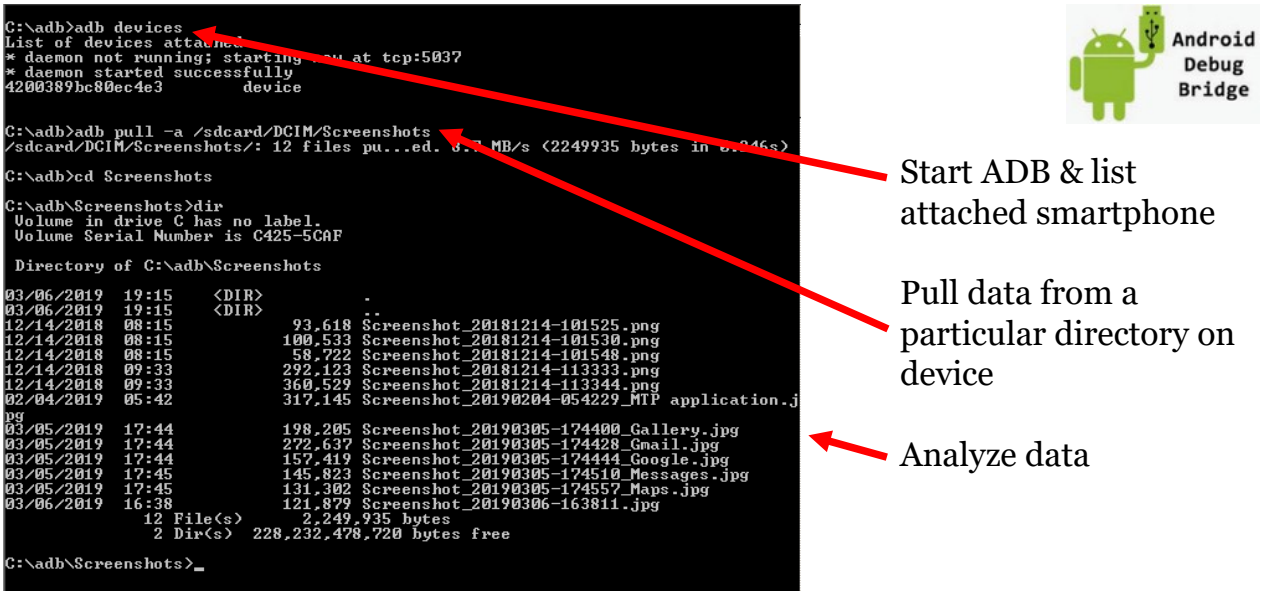
iPhone Backup Extractor



iPhone Backup Extractor [1] is yet another tool for reviewing iPhone backups on a “try before you buy” basis. It also recovers deleted data.

[1] iPhone Backup Extractor | `https://for498.com/gpyel`

Android Solutions



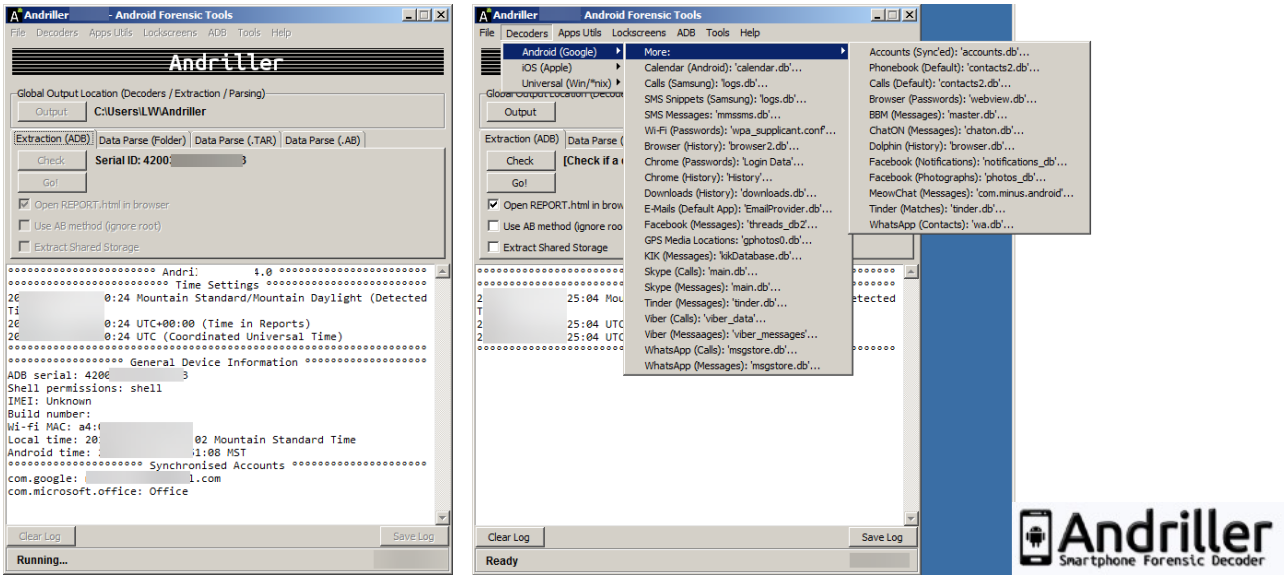
Many of the previous tools shown will work on Android as well as iPhone. For the best possible outcome in an Android device examination, the device needs to be rooted first. This involves interacting with the file system to place a file that will allow certain functions to occur on the device. With many forensic tools, this is also the process, although the forensicator doesn't typically see this activity happening. Whether or not an examiner feels that this is forensically sound or allowable in a given situation is a judgment call that has to be made at the time. The point is that for as much as an examiner may not want to root a device, most forensic tools are doing this anyway.

A great free tool for extracting data from Android devices is a tool called Android Debug Bridge (ADB). [1] This program will allow you to interact with an Android device in a number of different ways, from creating a shell so that you can interact with the file system on the device, to creating a complete download on your system, ADB allows for great granularity. With the ability to create a shell on the device, it allows the examiner to target very specific information in very specific places, if these locations are known. For example, if the examiner knows exactly where the photos and screenshots of the device are, she can target that area immediately, and in a matter of a couple of commands, she can copy them over to the examination machine.

The drawback to this tool is that it is command line, and not enough examiners are comfortable at the command line. As well, this tool is not as simple as “double-click and install”. When it is working, it works very well, but when it's not working, troubleshooting can be challenging. There are some great tutorials [2][3] online that cover the basics of ADB and there are also websites [3] that help troubleshoot some of the more common issues. A very common issue when trying to install ADB onto a windows system, is conflicts with Android drivers. You may have to uninstall the drivers that are on your system and install generic Android drivers in order for ADB to work.

[1] Downloading of ADB | `https://for498.com/9lwjf`
[2] Basics of using ADB(1) | `https://for498.com/6zabe`
[3] Basics of using ADB(2) | `https://for498.com/8xp-o`
[3] Driver troubleshooting | `https://for498.com/fg1bk`

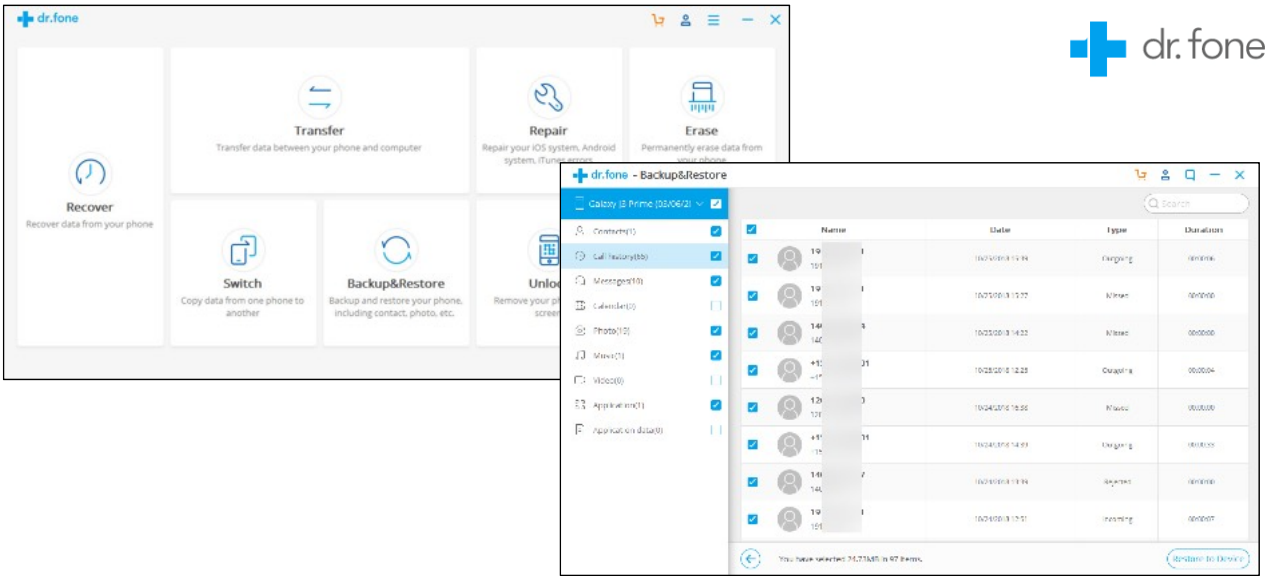
Andriller



Andriller [1] is another tool that gives a great deal of granularity, not to mention the ability to parse various types of data.

[1] Andriller | `https://for498.com/pbf2o`

dr.fone



dr.fone [1] is a "try before you buy" program from Wondershare. This software is incredibly versatile and allows the forensicator to acquire and analyze both iPhones and Android devices. It can analyze resident data and it can also root an Android device to extract a physical data dump. It is also quite proficient at recovering deleted data.

It is not uncommon for dr.fone to recover thousands of deleted text messages that far more expensive pieces of software are unable to recover. It is also typically the first tool to be able to access a new file system for a given device, often days if not weeks sooner than many forensic products.

[1] dr.fone | `https://for498.com/hgte1`

While a great many of these tools and others will attempt to parse all apps on the device, each one will do a slightly better or worse job than other. Given the plethora of messaging and communication apps, this can become quite challenging. It can be very difficult to stay abreast of the changes within the cellular device forensics market space.

There are a number of people within the forensic market space that have contributed greatly to the topic of smart phone forensics. In the iPhone space, Sarah Edwards has done magnificent work to further the abilities that exist today. She created a tool that can analyze health related data from iPhones. It is called Apple Pattern of Life Lazy Output'er (APOLLO). [1] This tool analyzes iPhone data to parse out a user's daily habits based on various "Pattern of Life" data points. Sarah is also the author and lead instructor of the SANS FOR518 Mac and iOS Forensic Analysis and Incident Response course. [2] Her blog [3] is certainly one of the top smart phone forensics blogs that every forensicator should monitor regularly.

Another incredible leader in the smart phone forensics market space is Heather Mahalik, the co-author of the SANS FOR585 Smartphone Forensic Analysis In-Depth course. [4] Heather maintains a blog [5] that every forensicator should be following.

[1] Apple Pattern of Life Lazy Output'er (APOLLO) | `https://for498.com/-3fki`

[2] FOR518 Mac and iOS Forensic Analysis and Incident Response | `https://for498.com/ofiym`

[3] Sarah Edwards blog | `https://for498.com/f837a`

[4] FOR585 Smartphone Forensic Analysis In-Depth | `https://for498.com/fvk7q`

[5]Heather Mahalik blog | `https://for498.com/z36of`

Email on Portable Devices

In digital forensics, a massive treasure trove of data can often be found within email. Given that smart phones and other portable devices are some of our most personal possessions, it stands to reason then that these would contain email. Not only would these contain normal email that we would see through normal day-to-day activities on computers, but if someone is trying to communicate in a clandestine manner, this will typically be done on a smart phone. There are any number of apps for both iPhones and Android devices that will allow for sending and receiving of emails outside of the normal email programs that would typically be found. These are then destroyed automatically at some point in time.

Unfortunately from a forensic perspective, manufacturers of smart phones have gone to great lengths to ensure that at least this part of the data store is not something that is recoverable in a normal forensic operation. In fact, the only way that emails can be retrieved from smart phones is in the case of a full physical dump of the data. In the case of iPhones, this would mean that the device needs to be jailbroken first, and in the case of Androids, the device would need to be rooted. Although in many cases this is not an insurmountable issue, it all depends on the matter at hand. At least as regards many civil cases, most courts would be loath to issue instruction to allow for that level of change on someone's device.

The takeaway of all of this is that emails in their normal format are typically not recoverable from smart phones in the majority of examinations. Having said that, this does not mean that there is no relevant email information. As has been seen through a number of the different pieces of software available to us, at the very least we can get email account information that can possibly lead us to other sources for this email.

Summary

- Once again, no single tool is best
- Determine what is important to move your case forward, and target this data
- Understand the differences between Apple and Android
- Tools and capabilities are constantly changing
- Don’t test theories or practice ideas on the evidence

This page intentionally left blank.

Exercise 2.2A-B

Portable Device Analysis

Synopsis: In this exercise, you will first use Physical Analyzer to analyze and extract quick win data from a data dump of an iPhone. You will then use Axiom to do the same with an Android data dump.

Average Time: 50 Minutes

This page intentionally left blank.

Exercise 2.2A-B Takeaway

- There are many “quick wins” in the easily available data from a portable device.
- You must be able to explain what you see and give reasons why.
- Vendors pack more and more features into their software every day. Software that looks easy and intuitive could have many intricate surprises, and you must become familiar with them.
- You cannot take the results from one tool for granted. Different tools present data in different ways, and often show different results.

This page intentionally left blank.

FOR498 Section 2 — Agenda

2.1 Portable Device Acquisition

2.2 Portable Device Analysis

2.3 Acquisition Hardware & Software

2.4 Acquisition Methodology

2.5 Discovering & Interacting with Data

This page intentionally left blank.

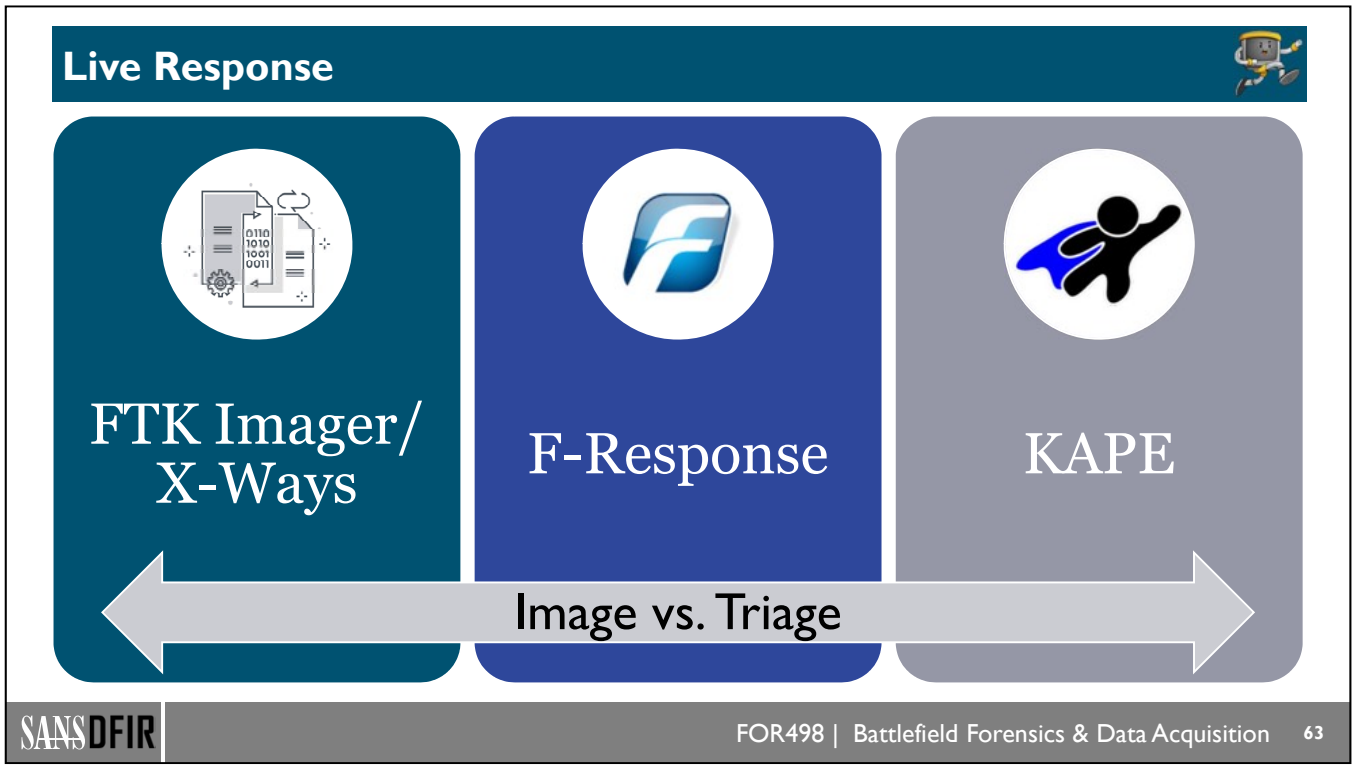
Acquisition Hardware & Software

Live Response

Dead Box: Write Blocking Techniques

Preparing Destination Media

This page intentionally left blank.



In the early days of forensic response, it was a given that the storage media would be removed from a computer and connected to a special device that would write block it, and then present it to a computer from which the imaging process would be performed. RAM was not a consideration, and there was no live response. If the subject machine was on at the time of seizure, it was simply turned off in some manner. If the computer was a standalone machine, it was customary to pull the plug on the box, so as to not cause any further writing to the drive from the shut down operations. This caused its own issues with examiners who were not conversant with what a proper shut down operation looked like from a forensic perspective. If the machine was a server style product, it was customary to do a proper shut down through the Start button, because the risk of overwriting data during shutdown was significantly outweighed by the risk of corrupting the server and having a situation where it would not restart properly. It goes without saying that standalone computers are and were far more resilient than servers during non-standard operating conditions.

There really weren't many things going on with the "on the wire" investigative space other than tap capturing and analysis. Due to bandwidth issues both internally and publicly, the idea of pulling data from a remote computer was reserved for only the smallest of artifacts. Some of the forensic software suites had network capable acquisition tools, but they were prohibitively expensive to any but the largest organizations. The notion of cloud computing hadn't been explored yet, at least certainly not from the perspective of forensic collection implications.

One of the biggest challenges that the forensic community faced was the ability to adequately target the most useful data from a hard drive. Data sets were typically small enough that full disk acquisition was not such a time-consuming event, nor was the data density and type so complex. Even then, only a small percentage of forensicators truly understood the data therein and how to analyze it. The rest were relying on one of the two large forensic analysis suites of the day to massage the data out for them. These tools were considered "magic buttons". If the buttons in the program couldn't get the data an examiner needed, it must not be available. This was the perception, and critical information relevant to many investigations went undiscovered, or improperly interpreted. In fact, although there are a great many more tools today than ever before, many of them are surgical in their focus. Up until recently, accessing the data on a hard drive for rapid triage purpose, although faster than full acquisition and parsing, was anything but rapid. Nothing existed that could be considered the much sought-after Holy Grail forensic magic button. Until today.

Live Response: FTK Imager/X-Ways

FTK Imager/
X-Ways

Live acquisition from USB

Targeted file extraction

Triage imaging

RAM acquisition

In today's forensic landscape, computers are left on more than ever before, and this is how we are finding them. Whole disk encryption brings a new dynamic to the playing field as well. If we don't deal with the computer in a live manner, we may never be able to access the data. RAM has become quite large and holds vastly more data than in the past. Couple that with the fact that when you turn the computer off, data in memory is forever lost. This is a recipe for potential disaster. Technology and a more knowledgeable forensicator have given us the ability to surgically extract only the data we need, or at least the data that we quickly need.

FTK Imager is a tool by Access Data that allows us to collect forensic images of computer hard drives. It allows for much more, but much of its capability went unused for other than this purpose. It also allows for the imaging of a live system when the 'lite' version is run from an external storage media. An examiner can plug in a USB device to the subject system as long as they are logging the activity, and activate the FTK Imager software without having to install it. This would allow for a number of options including full live imaging to an external storage media, partial targeted data extraction such as logical file extraction, and something called Custom Content Imaging, or what we call triage imaging, for the relatively rapid extraction of time sensitive data. This tool is extensively used in the e-Discovery world for targeted extraction, usually of specific file types like email containers. A very important capability of FTK Imager is also the acquisition of RAM from a live system.

X-Ways Forensics is an integrated computer forensics program that runs on Windows. From the X-Ways web page:

"X-Ways Forensics is an advanced work environment for computer forensic examiners and our flagship product. Runs under Windows XP/2003/Vista/2008/7/8/8.1/2012/10*, 32 Bit/64 Bit, standard/PE/FE. Compared to its competitors, X-Ways Forensics is more efficient to use after a while, by far not as resource-hungry, often runs much faster, finds deleted files and search hits that the competitors will miss, offers many features that the others lack, as a German product is potentially more trustworthy, comes at a fraction of the cost, does not have any ridiculous hardware requirements, does not depend on setting up a complex database, etc.! X-Ways Forensics is fully portable and runs off a USB stick on any given Windows system without installation if you want. Downloads and installs within seconds (just a few MB in size, not GB).

X-Ways Forensics is based on the WinHex hex and disk editor and part of an efficient workflow model where computer forensic examiners share data and collaborate with investigators that use X-Ways Investigator." [1]

[1] X-Ways Forensics | `https://for498.com/guy1o`

Live Response: F-Response

F-Response

‘Over the wire’ acquisition

Accessing computers across a network

Cloud data acquisition

F-Response is a very capable tool that has been in use for a few years and gained a solid reputation in the industry for its capabilities. It can perform several functions using servlets (on Windows, this is in the form of a service). The servlets are little beacons that can be transferred quite easily by email. They are then placed on a machine of interest, and the host being operated by the forensicator can connect to it across the internal network, or even across the Internet to cloud space to collect artifacts as needed. Bandwidth is still an impedance to full disk acquisition, when you consider trying to image a full 2 TB drive and pull it across a network. Even a 16 GB RAM extraction would be slow enough to cause RAM smear and render the RAM unusable. Normally a forensicator would use F-Response to cause the RAM acquisition to happen but save the image to an external drive on the subject machine placed there by a trusted source at the subject end.

A great feature of F-Response is its ‘covert mode’, where it leaves no indication that the servlet is running on the subject machine, save for artifacts that would take a forensicator to find.

Live Response: KAPE

Target specific data, including that in volume shadow copies

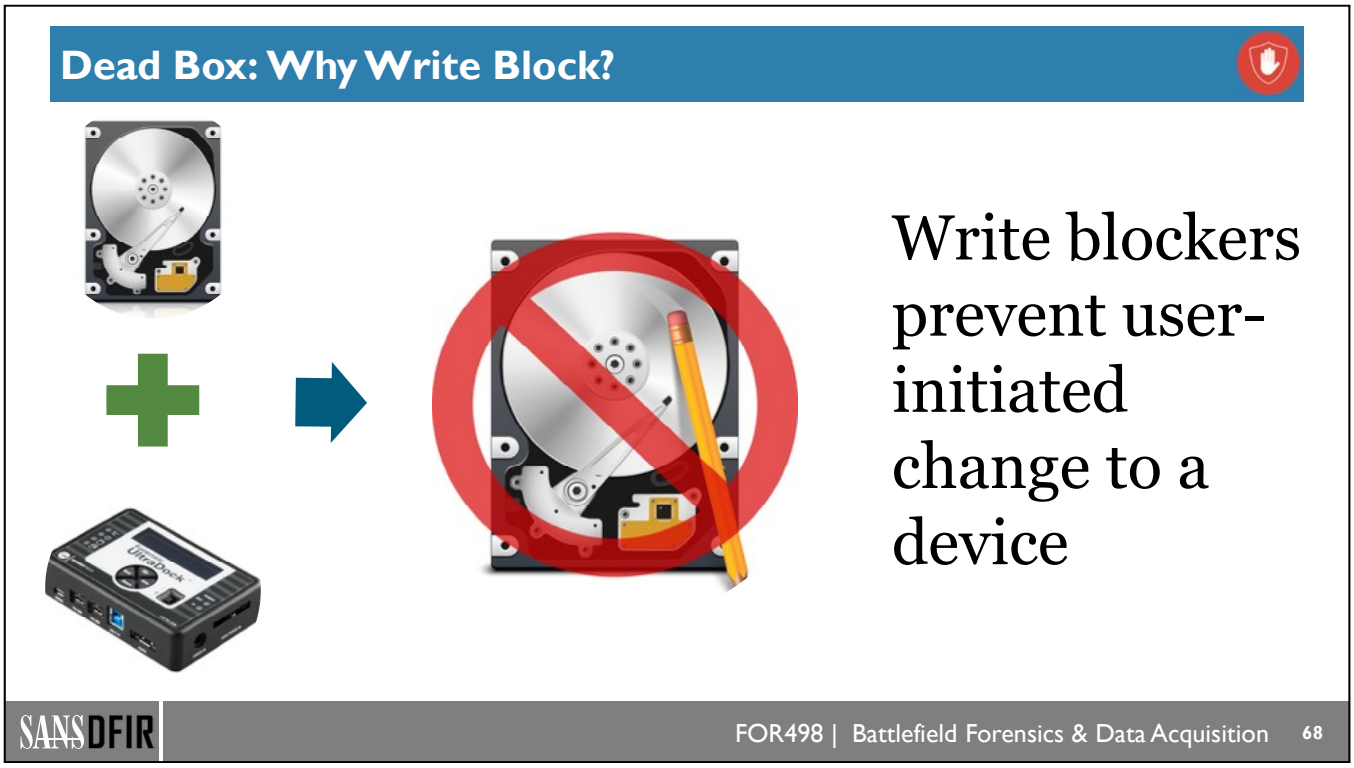
Forensically sound extraction

Run programs against extracted data

KAPE is an easy to use and flexible triage tool designed to quickly find the most important data to your investigations and extract it in a forensically sound manner. This includes automatic mounting and processing of volume shadow copies, deduplication of files based on SHA-1, and a rich audit log of all actions performed.

KAPE can also, if you choose to, run one or more programs against collected data (or on the live machine) to parse artifacts and generate CSVs or other more consumable data. This “collect and process” approach makes getting answers from computers much easier than it has been in the past. By using KAPE before a full image (or in lieu of it entirely), you can start the analytical process against key artifacts in minutes, not hours or days.

We will be taking a much deeper look at KAPE in a future section.



This truth should be self-evident! We always want to avoid changing any data we interact with as much as possible.

We can avoid changes to a hard drive by using a write blocker. This is in the case of spinning media. As we saw previously however, things change when dealing with an SSD. Does this mean we should not write block an SSD when we need to interact with it? Certainly not! It is always a best practice to take every step we can to avoid being able to change things.

In short, when dealing with a loose storage device, always connect it to a write blocker (that has been tested of course!) and document this fact in your report. This way you will be able to show you took every step possible to prevent changes to the source device. Once the device is connected to a write blocker, the write blocker is powered on and the device behind the write blocker will be presented to the operating system.

Another useful feature of most write blockers is getting information about the source device from the write blocker via an LCD screen and various menus. Information such as capacity, make, and model are generally available on the write blocker and can be documented and compared to what is on the drive label. Any discrepancies should be noted in your report.

Write blockers also offer a wide range of connections such as FireWire, USB2/3, eSATA, and so on that provide you flexibility depending on the system you are connecting the source drive to. You should choose the fastest interface available, or at least one that is as fast as the source device, to ensure you can image the drive at the fastest speeds possible. This will typically be either eSATA or USB3.

Once the source device is recognized by the host operating system, acquisition can begin.

Dead Box: Write Blocking Techniques

Software

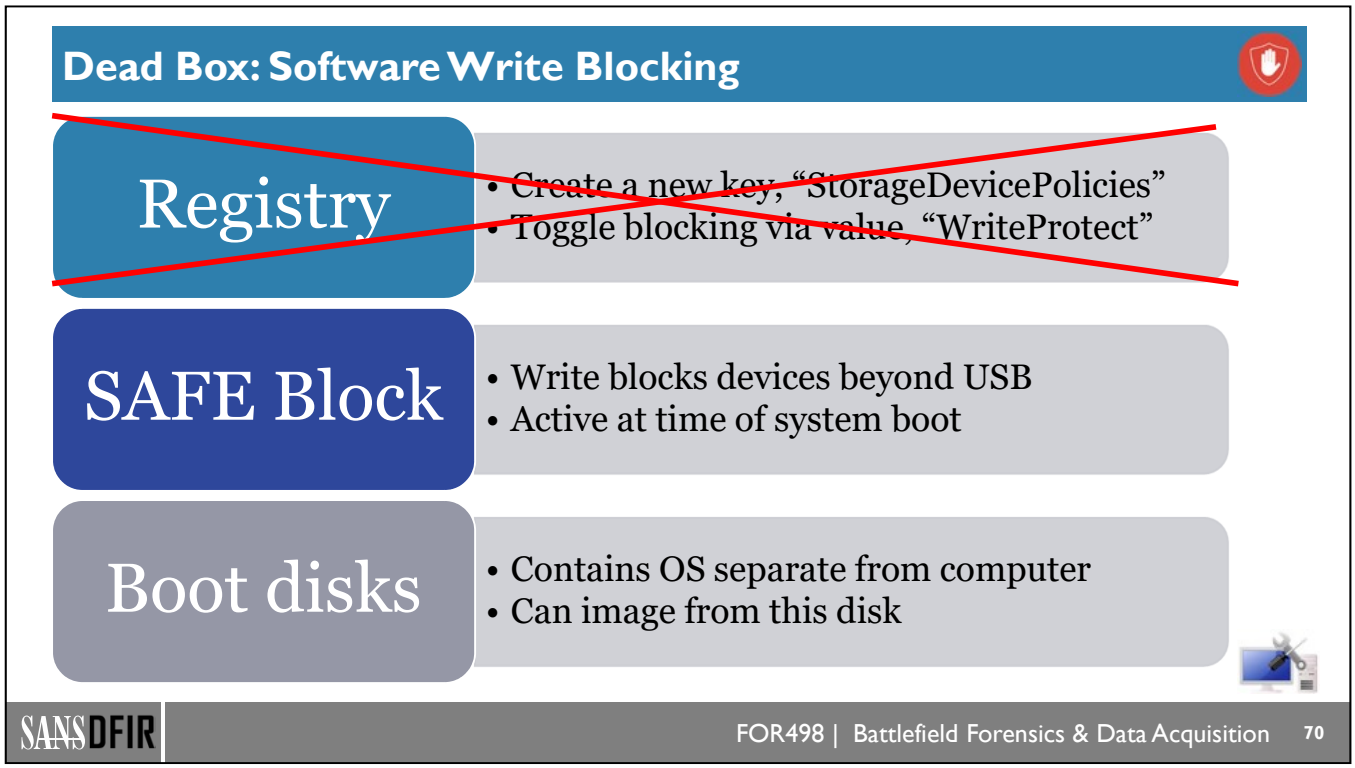
Hardware

Which one

to choose?

When a device is not powered on, we have additional choices to make. While it is certainly an option to power on a device to image it (and in some cases, this may be the best choice, as with exotic storage for example), we always want to be as forensically sound as possible. Another way to think about this (and perhaps the better way) is that we want to be as minimally intrusive as possible. Working with a device in its powered off state vs turning it on to interact with it is an example of something being less intrusive vs. more intrusive, respectfully.

While the result is essentially the same, how we go about achieving this will be decided by which path we take. So what are the two paths we can take? The first is software write blocking and the second is hardware write blocking. Let's take a closer look at each of these techniques to get a better understanding of how they work and their pros and cons.



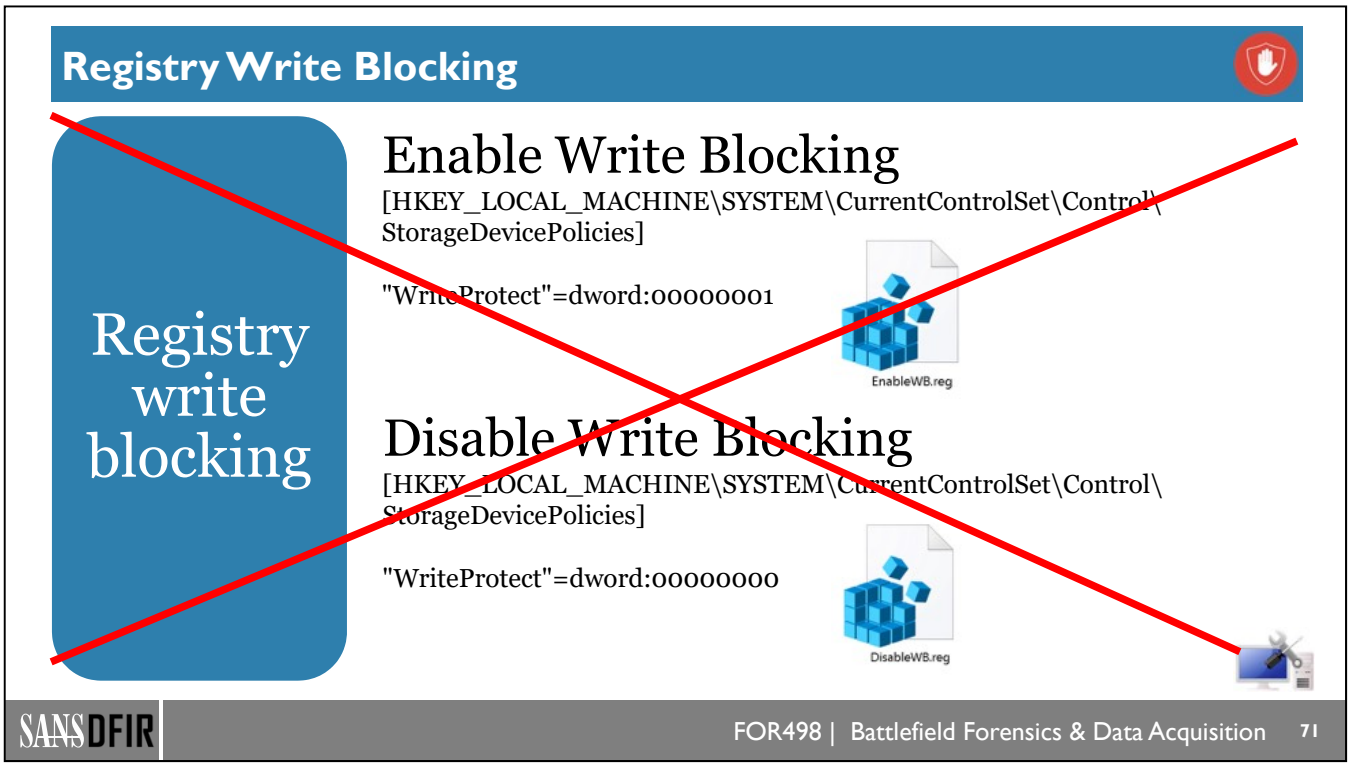
There are relatively few techniques at our disposal for write blocking at a software level. We used to be able to use our computer's Registry to control data transfer across our USB ports so that data could be read from an attached device, but not written to the device. Today with changes to the Windows 10 Registry, the original port blocking keys have been moved, and committing the known write-blocking registry keys will no longer work. As well, with the benefit of experience and testing, we now know that it is very possible to write to a drive that has been write blocked using the Registry, and as a result, this method is no longer used or recommended.

We can use a product called SAFE Block to essentially write block anything connected to a computer through any interface.

Finally, we have boot disks that we can use for the purpose. A boot disk variously refers to a CD, DVD, or USB drive that has an operating system on it. When connected to a subject machine, we can start the machine using the operating system on the boot disk, rather than the operating system of the subject computer.

When would you ever use software based write blocking? Should you ever use it? We would say that you should not use Registry based software write blocking unless you do not have any hardware-based means to perform write blocking, or software specifically designed for the purpose. It is important to understand the risks specifically of software write blocking, and what is meant by the literature that abounds, professing that it is too risky to use.

With that said, let's look at what our options are as it relates to software write blocking.



Back in the days of early Windows 7 and older, it was a perfectly viable way to write block a hard drive by using the Windows Registry[1]. A user could create a Registry Key to **ENABLE** write blocking, and then another to **DISABLE** write blocking. A shortcut was created to both keys, and then write blocking could be controlled simply by double clicking on the appropriate icon. Unfortunately, many examiners have/had no idea how this worked, and believe that it is simply write blocking all ports on the computer. This is incorrect. The write blocking is occurring outbound on USB ports only. And then, it only works for specific types of media.

Many of the technical details of USB are beyond the scope of this class, however it should be known that different USB devices are detected by a computer in different ways. For example, some will identify as "Removable Disk", and some will identify as "Fixed Disk Drive". Fixed disk drives used to be reserved for ordinary hard drives that were connected to a system internally.

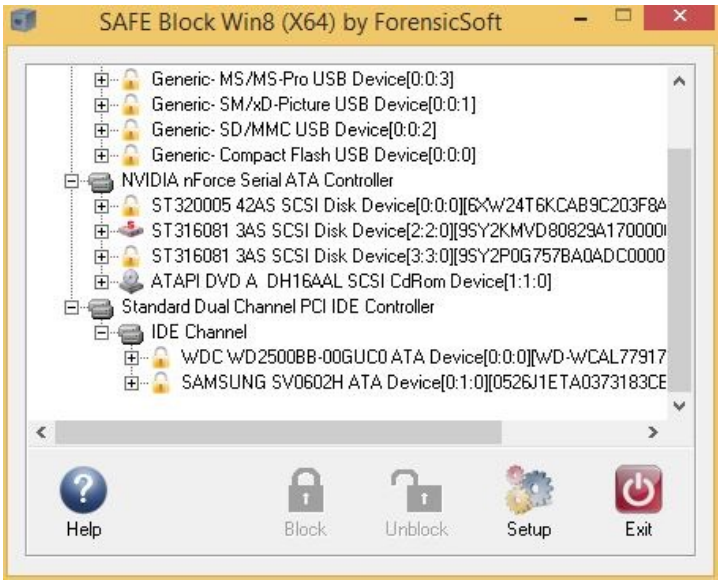
In recent years, many external hard drives, and some flash drives will be detected as Fixed Disk. This is attributable to the fact that these drives are USB Attached SCSI Protocol (UASP)[2] enabled. UASP enabled drives are controlled by the Windows driver uaspstor.sys instead of the traditional usbstor.sys driver. This led to the discovery that UASP SSD drives like the popular Samsung T5 SSD and others are completely unprotected by the "WriteProtect" registry key and do not attach read-only. As a result, Registry write blocking will NOT block writes to these types of drives. For this and other reasons, we do NOT recommend using Registry write blocking.

[1] Write Blocking Using the Windows Registry | `https://for498.com/djahi`

[2] USB Attached SCSI Protocol | `https://for498.com/rvlcj`

SAFE Block Write Blocking

SAFE Block



SAFE Block is a tool created by ForensicSoft [1]. It is a software write blocker that currently has no equal. It is the last word in software write blocking because of its capability. Once installed, it functions from the moment of Windows boot to automatically write block any connected device except the boot drive. While Registry write blocking only write blocks devices connected to USB ports, SAFE Block write blocks everything connected to *any* interface. In the case of desktop computers, it will even write block any storage connected via PCI-e cards on the motherboard. It will write block SATA, IDE, RAID volumes, SAS, SCSI, USB, Firewire, eSATA, Fibre Channel, and more. As well, it has been tested by NIST Test Suite, and verified to be forensically sound when properly deployed. Another feature is the acquisition speed. In many cases, it can match or exceed speeds delivered by expensive hardware write blockers; and that is for the limited media that hardware write blockers can address. It has the added bonus of having customer support that is unchallenged in the industry.

Once the system is booted, SAFE Block displays the drives that it can see, and their status. A user can disable or enable write blocking with a simple mouse click and confirmation.

In one particular case, there was a need to image a number of fibre-channel hard drives. The challenge beyond interface type was that these drives came from an EMC setup and were created at the factory with 520-byte sectors, as opposed to the normal 512-byte sector size. This created an environment where Windows could not see the devices in any way. Due to the usage and configuration of these drives, imaging them without being write blocked was simply not an option for any reason. SAFE Block was able to read the device's existence and write block it, which then allowed it to be imaged.

The only drawback to this tool is the marketing surrounding it. This appears to be a classic case of long time forensicators developing a tool to fill a void. They are extremely good at what they do, yet their marketing is not where it should be, as evidenced by the fact that many in the industry for a long time have never heard of them.

[1] SAFE Block by ForensicSoft | `https://for498.com/safe`

A boot disk is a disk (in various formats) that carries an operating system. This OS will allow a forensicator to connect it to a subject device and boot to the OS of the boot disk. This allows access to the subject media in a write blocked manner, both for triage, and for disk imaging.

There are a number of great choices for this including a relative newcomer from ForensicSoft called SAFE Block To Go. It is not a standalone product, and actually needs to be installed to a device that is already carrying a Windows To Go installation. This means it is not the easiest method to get up and running. Windows To Go itself is meant to be a Windows 10 Enterprise license and can only be installed on certain USB drives. However if you are potentially dealing with media types that cannot be accessed by anything else, this may be the answer.

Another great choice is Paladin, by Sumuri. This boot disk (USB interface) is free but donations are suggested (and recommended), and a USB drive with Paladin installed and ready to go is also available. Paladin is a great choice in certain circumstances for its ability to interact with certain laptops that cannot be accessed by other boot disk solutions.

Any good boot disk solution will offer the ability to format destination drives in whatever format is necessary, such as FAT, NTFS, exFat, EXT4, HFS+, and others. It should also afford the ability to create multiple different evidence file output formats, such as .Exx, RAW, dd, DMG, and others. In some cases, you even have the ability to image out to two destinations simultaneously. In the case of Paladin, you also have tools onboard that will allow for accessing data and analysis through the use of the respected forensic platform Autopsy.

Boot disks are not without risks, however. In some cases, forensic examiners have shown that subject drives can execute code when accessed via common forensic boot disks[1]. With that said, and just like most every other aspect of forensics, you must weigh the risk as it relates to who the subject drive belongs to and when you may actually use a boot disk.

[1] Live boot disk executing malicious code from a suspect drive | `https://for498.com/m-15p`

Dead Box: Hardware Write Blocking

All-in-one

- WiebeTech Ditto
- Logicube Talon
- Tableau TD3

Write blocker

- WiebeTech UltraDock
- Tableau T8U

Under the category of hardware write blocking, devices exist that only write block, and devices that will write block and perform the acquisition process all from the same device, with no need for a host computer.

One example of an all-in-one write blocker/imager is the Ditto DX [1] from CRU. This 'field station' as it is referred to, is an extremely versatile device. It performs write blocking, as well as imaging of the drive. Some of the standout features are the ability to do a multitude of different interfaces, as well as the ability to image the logical partitions only.

Another very useful feature is its network acquisition ability. The device can be shipped anywhere in the world, where a non-technical person can plug it into the network, and a technician can connect to it, and perform the acquisition remotely via a web interface that provides for incredible granularity of the acquisition process. CRU also professes that the Ditto and the Ditto DX have unparalleled device life, owing to their ability to be upgraded for years to come. More importantly, very unlike many devices on the market, they actually release updates on a regular basis.

All-in-one devices come with a price however and are often in the $1600.00 – $3000.00 USD price range and higher. A more economical option would be a simple write blocker. These devices perform write blocking only and rely on a host computer to perform the actual imaging process, while they block any write attempts to the subject drive.

You have been provided with a CRU Forensic UltraDock, which performs this function. As an added bonus, it provides a number of other information gathering features including the operating temperature of the drive, identification and notification of hidden Host Protected Area (HPA) and Device Configuration Overlay (DCO) [3]. We will be discovering more features of this device in a lab later in the course.

[1] Ditto DX | https://for498.com/9hp10

Dead Box: Write Blocker Testing

Regular testing and documentation protects you from allegations of tampering

Testing and validation can be time consuming and boring processes, but they are some of the most critical to do on a regular basis. By performing regular testing and keeping detailed notes about the results, you can easily head off any accusations of not following best practices when it comes to using a write blocker.

Testing of a write blocker is straightforward. It can be as simple as connecting a storage device to the write blocker and attempting to alter data by deleting a file, editing a file, etc. Depending on the type of write blocker, this simple test may fail outright, which means the change was committed. Other write blockers may cache the changes, so that it seems like the source data was altered, but when the device is disconnected and reconnected, you can see the data has not been changed. This can be verified by acquiring the device through the write blocker and then comparing the hashes for the source drive and destination image as well. Of course, when performing these tests, it should be done against one of your own hard drives and not original evidence!

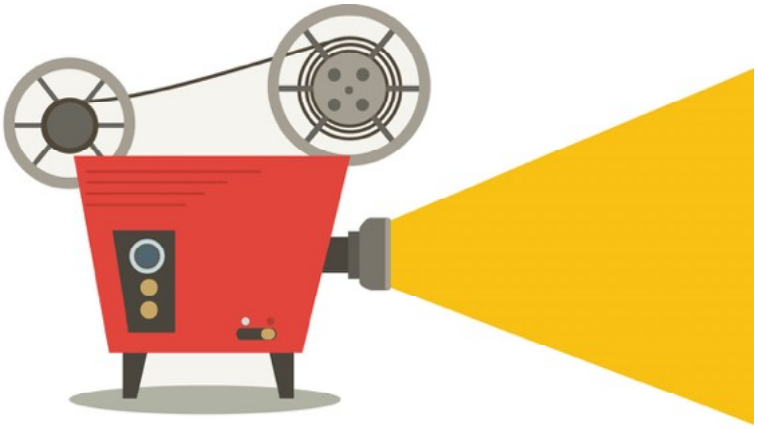
Another aspect related to testing is keeping up with firmware updates for your write blockers. Over time, manufacturers release updates to firmware that add features, and more importantly, address any bugs that may be present in the device's software. Signing up to the vendor's mailing list, especially related to security matters and notifications, is often a good idea to ensure you are notified about any critical updates.

If you make firmware updates a regular part of your write blocker validation, both needs can be addressed at the same time. This also makes documenting things much easier because you can update the firmware to the latest version and then test the updated firmware to ensure it still meets your requirements.

One example of a great, NIST validated write blocking validation tool is The CRU WriteBlocking Validation Utility[1].

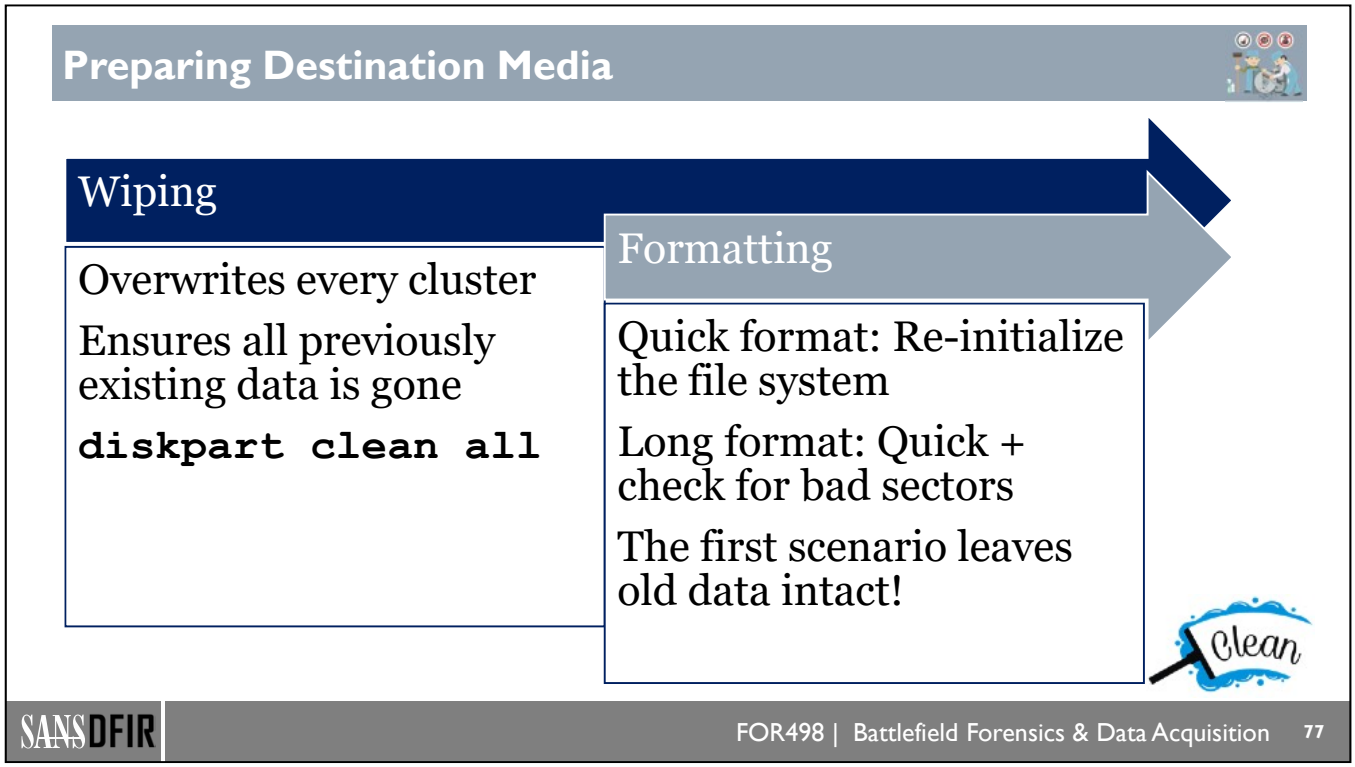
[1] Write Blocker Testing | `https://for498.com/zpnky`

Movie Time!



- 2_1: Write Blocking UltraDock

This page intentionally left blank.



Mac vs. PC, Cubs vs. White Sox, Manchester United vs. Liverpool, Capitals vs. Penguins (GO PENS!) and of course, wiping vs. formatting of hard drives as it relates to forensics. Let's start with a scenario.

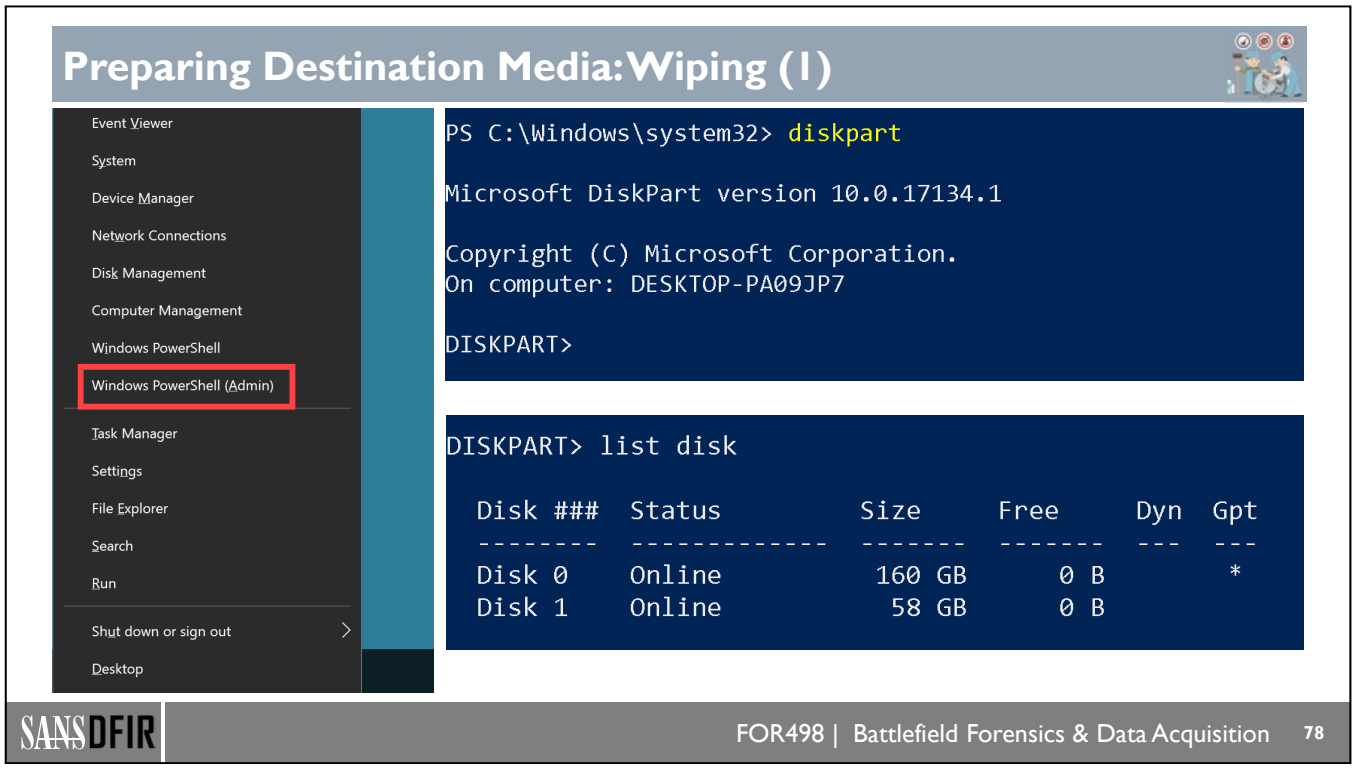
Back in the day, when a copy of a hard drive was needed, it was often "cloned" from the original to another drive of equal or larger size. Back then, container files like an E01 or RAW files were not used like they are today. So consider the case where you have a 500 GB drive and you clone it to a 500 GB hard drive. At the 400 GB mark is 50 GB of illegal images and videos. When the case finishes and you move on to the next, you once again clone a target drive, but this time it is only 100 GB. The cloning process in this case would only overwrite the first 100 GB of space, leaving the other 400 GB as they were.

If you were to then give this drive to someone outside your organization, do you see what the potential problem is? What if someone, in the course of their job, decided to carve for deleted files? What would they find at the 400 GB mark? Nothing good!

This situation is where the notion of always wiping a drive comes from, because it ensures that every cluster is overwritten with zeros before the drive is reused. But what about more modern times and practices? We generally do not (ever?) clone hard drives anymore, but rather, we image a drive to E01 or raw format. Doing this, we end up with a few large files vs. all of the files and data structures from the original device. It should be noted however, that when dealing with a raw image, you essentially have the same thing as a cloned device, but all the data is stored as a single file.

So what do we recommend? When a hard drive may leave your control, wipe it, but if a hard drive will not be going to another person, it is just more time consuming to wipe a drive vs format it. After all, when was the last time you carved a drive you use to store E01 and/or RAW files?

When you need to either wipe or format a device, using a tool like **DiskPart** on Windows makes short work of both.



In order to completely destroy the data on a drive, you will need to perform a proper wipe function. Certain forensic suites such as EnCase have the functionality to wipe storage media, however this is an expensive wiping tool!

The Windows operating system comes with a formatting/partitioning/wiping tool built right in for free. That tool is called **DiskPart** [1]. The **DiskPart** tool can perform a myriad of functions including formatting and wiping. Most people will use the Windows Disk Management tool for formatting drives, and between that tool and **DiskPart**, there is no right or wrong. The tools both perform the same formatting function, but it is a personal choice between command line function and GUI function. The command line function requires a higher level of attention to detail, as it is quite easy to pick the wrong drive to format. Also, at the command line, you must identify and execute every piece of the format function, from identifying the drive initially, to selecting the correct one, cleaning it, applying partition type, name, letter, volume, and file system, whereas the GUI tool manages most of this for you by way of default settings. Again, there is no right or wrong, and one is not better than the other for formatting. It is a matter of preference.

[1] DiskPart commands | `https://for498.com/1ig9b`

Preparing Destination Media: Wiping (2)

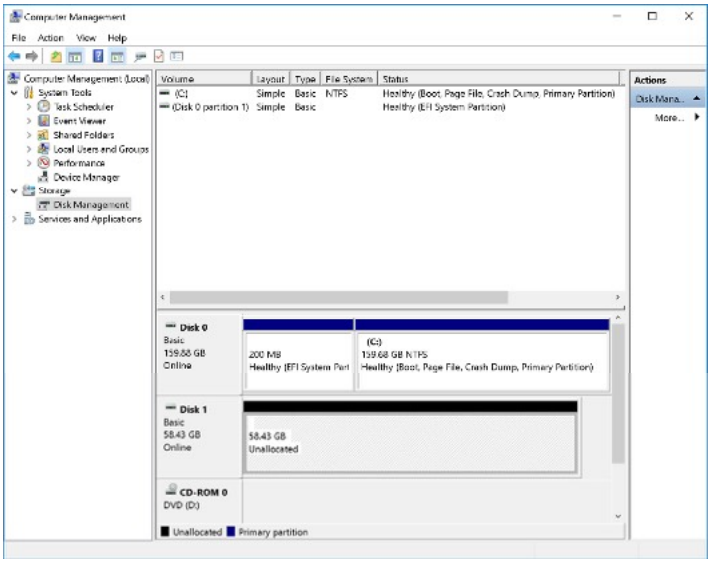
```
DISKPART> select Disk 1

Disk 1 is now the selected disk.
```

```
DISKPART> clean all

DiskPart succeeded in cleaning the disk.
```

Now you can format using either the Windows Disk Management Tool, or DiskPart



The wipe function is not fast. As a point of reference it took approximately 10 minutes to wipe a 64 GB USB drive (even though you shouldn't be wiping USB flash storage). A spinning 500 GB hard drive would take approximately 2-3 hours.

If you were to just type **`clean`** rather than **`clean all`** in the instruction above, this merely removes any existing formatting, but wipes nothing.

To show how easy it is to confuse the use of the term "wipe", even Microsoft's own page [1] showing example scripts for **DiskPart** carry a technically erroneous instruction. The instruction is thus:

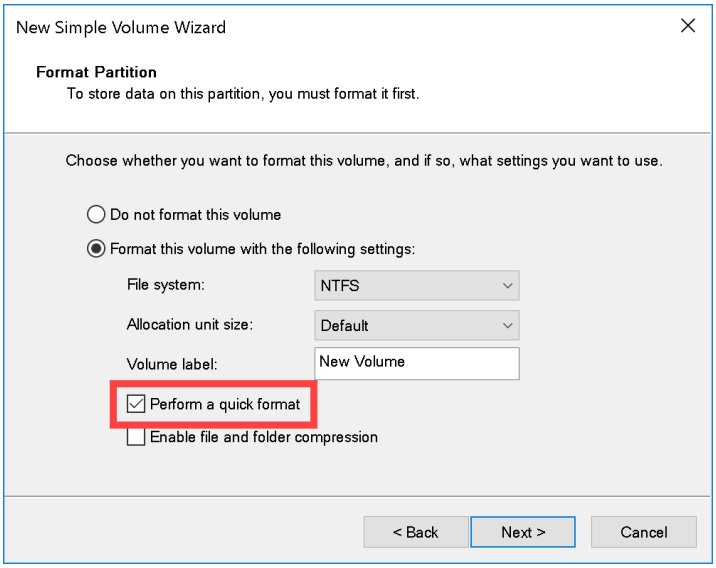
"…for example, here is a script that ***wipes*** *(emphasis mine)* a disk and then creates a 300 MB partition for the Windows recovery Environment:"

```
select disk 0
clean
convert gpt
create partition primary size=300
format quick fs=ntfs label="Windows RE tools"
assign letter="T"
```

In fact nothing in the commands above will **wipe** the drive. It will merely format. This is why we VERIFY, VERIFY, VERIFY.

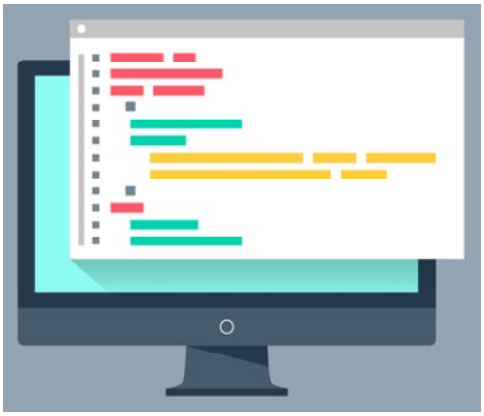
[1] DiskPart Scripts and Examples | `https://for498.com/uxq4w`

Preparing Destination Media: Formatting (1)



During the formatting process through Disk Management, you reach a point in the process where you are asked if you want to 'Perform a quick format' or not. The only difference between having this box checked and not is the length of time it will take to format your drive. If the box is checked, it only takes moments, as the OS is merely creating new boundaries and mapping. If the box is not checked, the OS will inspect and wipe each and every sector on the drive you are formatting, looking for bad sectors and reassigning them. Depending on the drive, this happens at anywhere from 4-6 GB per minute. As you can see, on a 2 TB hard drive, this will take hours.

Preparing Destination Media: Formatting (2)



Do you really want to script the formatting function?

```
select disk 0
clean
convert gpt
create partition primary
size=300
format quick fs=ntfs
label="Windows RE tools"
assign letter="T"
```

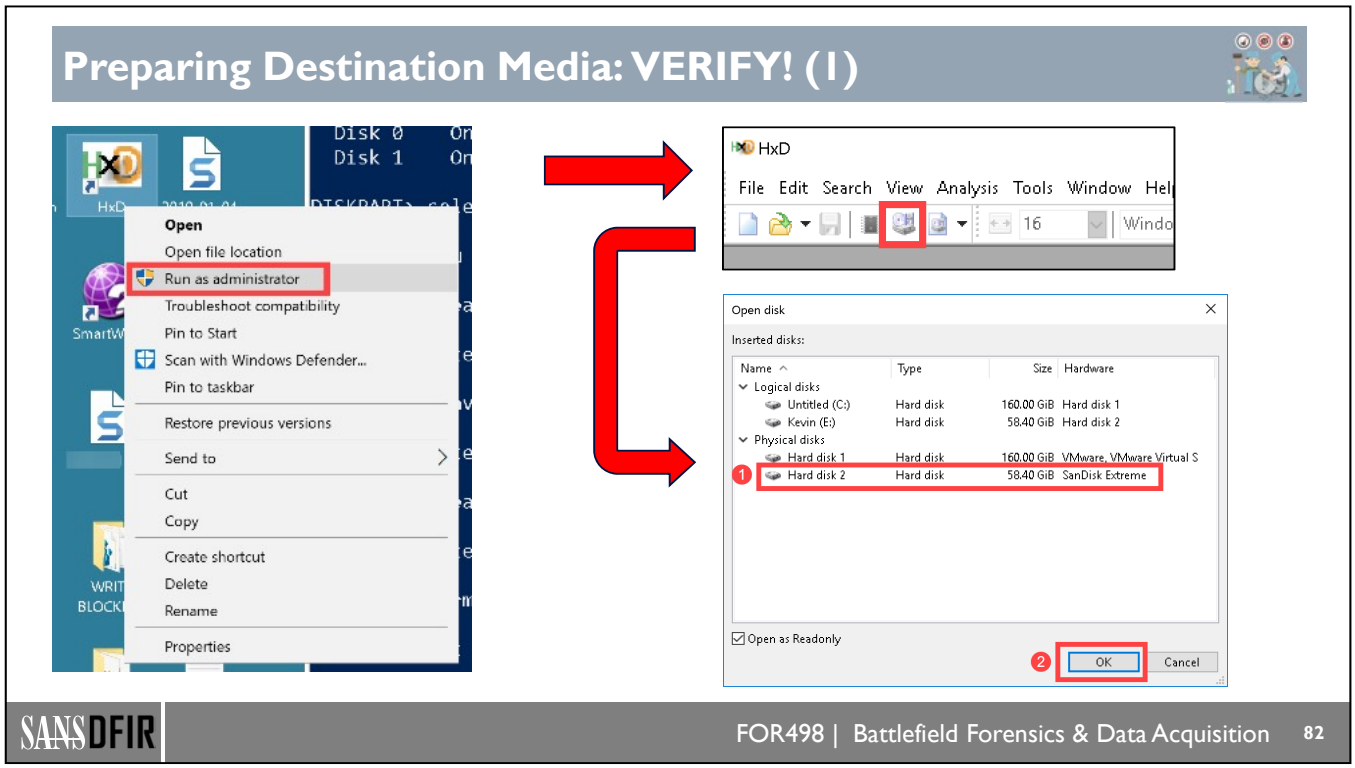
If it is your choice to format using the **DiskPart** tool, as mentioned before you must be careful about your selections.

Although it is possible to script the format functionality, it would only be recommended if you are always performing the same functions on the same computer with the same intended result every time. Why? Let's look at a simple script:

```
select disk 0
clean
convert gpt
create partition primary size=300
format quick fs=ntfs label="Windows RE tools"
assign letter="T"
```

The script above would apply its parameters in that manner every time. What if you didn't mean to select Disk 0? What if your script said Disk 1, but you really meant to select Disk 2? There is one word for this: Disaster.

[1] MBR vs GPT | `https://for498.com/hwt0y`

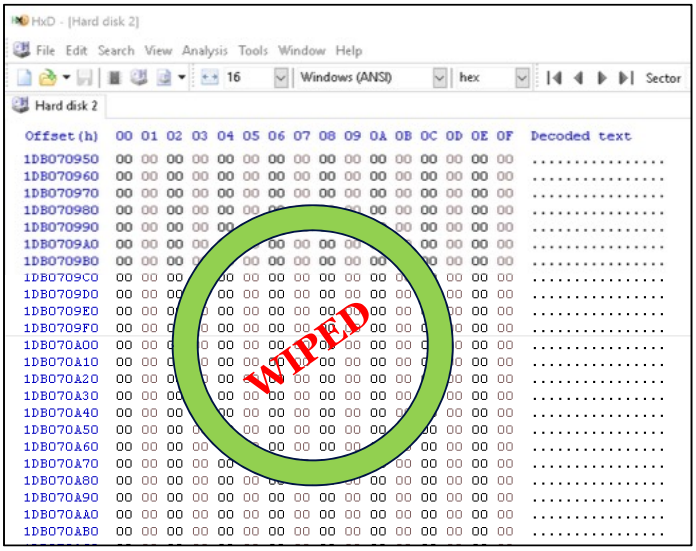


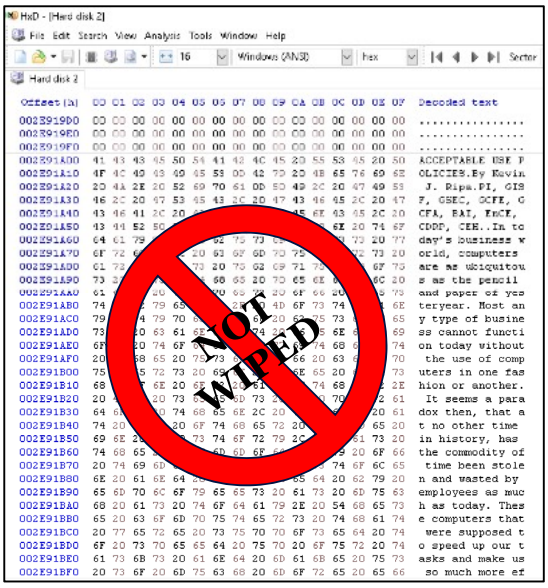
If the drive you have been dealing with was worthy of wiping, it is worthy of verifying whether your commands were correct, and that the wipe actually happened. The best tool to use for this is a hex editor. Not all hex editors are built the same.

An important factor in picking a hex editor is whether it can deal with drives at the disk level or not. **HxD**[1] is a great free tool from Maël Hörz, is included on your VM, and has the ability to see the drive at the physical level, rather than just the logical. You must run it with administrative privilege though.

[1] HxD Hex Editor | `https://for498.com/bq4a8`

Preparing Destination Media: VERIFY! (2)

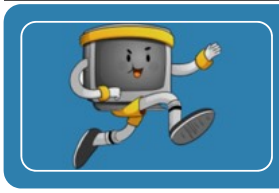




Scrolling through the contents of the disk, if the wipe was performed properly, there should be nothing but 0x00 in the hex field, and nothing but dots in the Decoded text field. If you see anything else, the wipe was not performed, or not performed properly.

In the example on the right, we see a drive that was formatted, but not wiped. You can clearly see that old data still exists and is readable. This falls under the previously mentioned category of disaster!

Summary

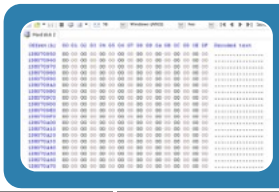


Live response options

- Full image vs selective imaging of a running device
- Triage collection can save considerable time

Write blockers

- Software based blockers allow for flexibility when hardware blockers fail or do not exist for a device
- Hardware blockers cannot write to devices, but are often interface specific



Formatting vs. wiping

- They are not the same!
- If you have a reason to wipe, you have a reason to verify

This page intentionally left blank.

Exercise 2.3

Hard Drive Wiping and Formatting

Synopsis: In this exercise, you will perform a wipe of a hard drive, as well as a formatting of a hard drive.

Average Time: 30 Minutes

This page intentionally left blank.

Exercise 2.3 Takeaway

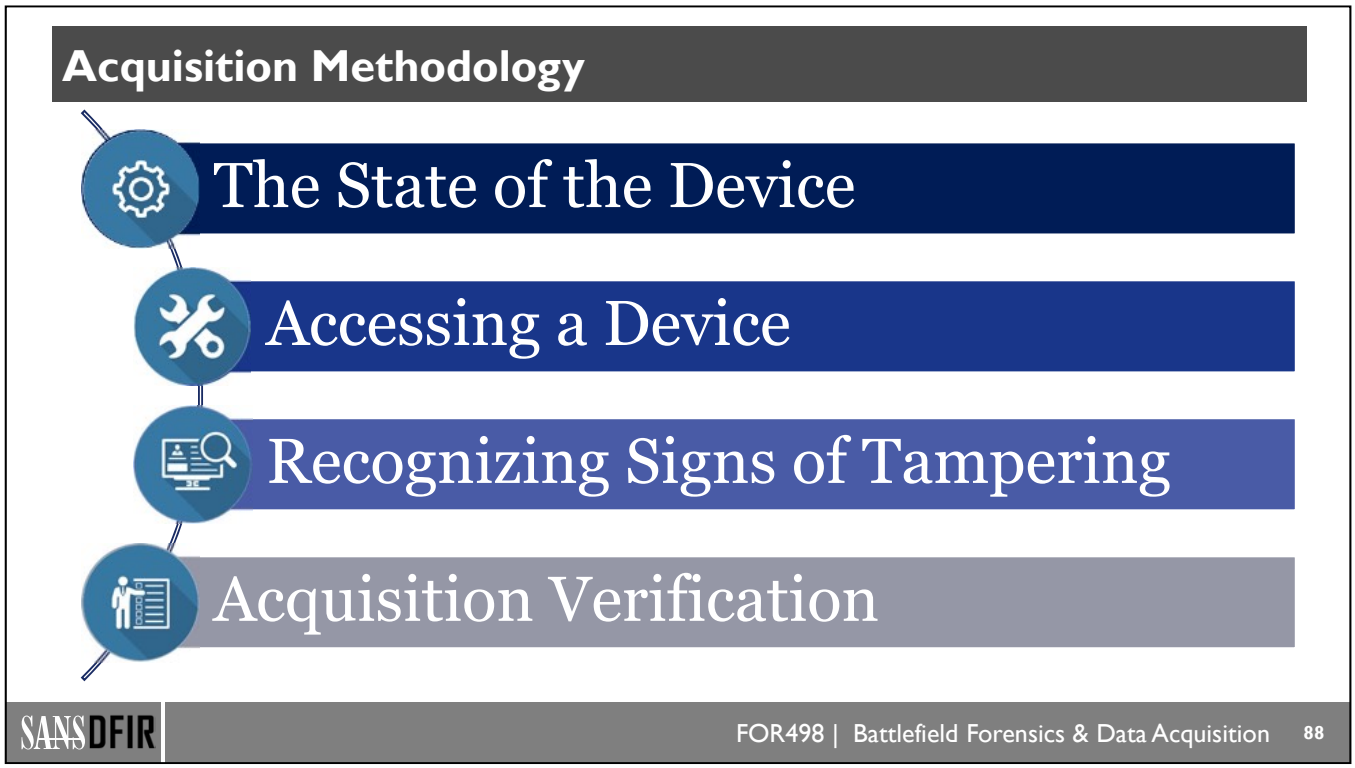
- There is no need for third party tools to forensically wipe a hard drive. The diskpart utility is built into every Windows operating system.
- There is a significant difference between formatting a hard drive and wiping it.
- Don't automatically trust that your software is performing a wipe. Verify, verify, verify.

This page intentionally left blank.

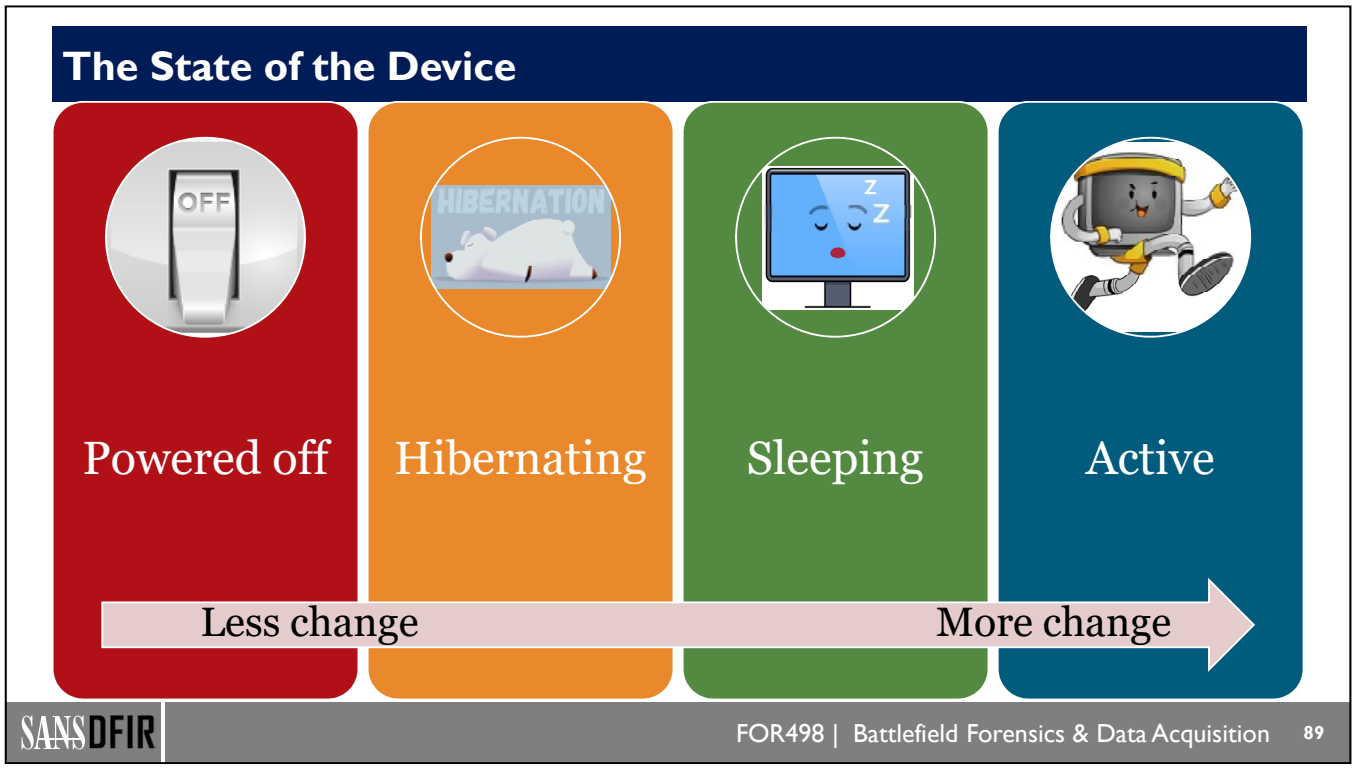
FOR498.2: Portable Devices & Evidence Acquisition Agenda

- 2.1 Portable Device Acquisition
- 2.2 Portable Device Analysis
- 2.3 Acquisition Hardware & Software
- 2.4 Acquisition Methodology
- 2.5 Discovering & Interacting with Data

This page intentionally left blank.



This page intentionally left blank.



When encountering a device, be it a desktop, laptop, or phone, one of, if not the first questions that must be asked, is whether the device is actively running, fully shut down, or a state in between these two extremes. At first glance, it can be difficult to determine this based solely on looking at the device. In most cases, interaction will be necessary to determine the state of the machine. For our purposes, consider the following states a device may be in:

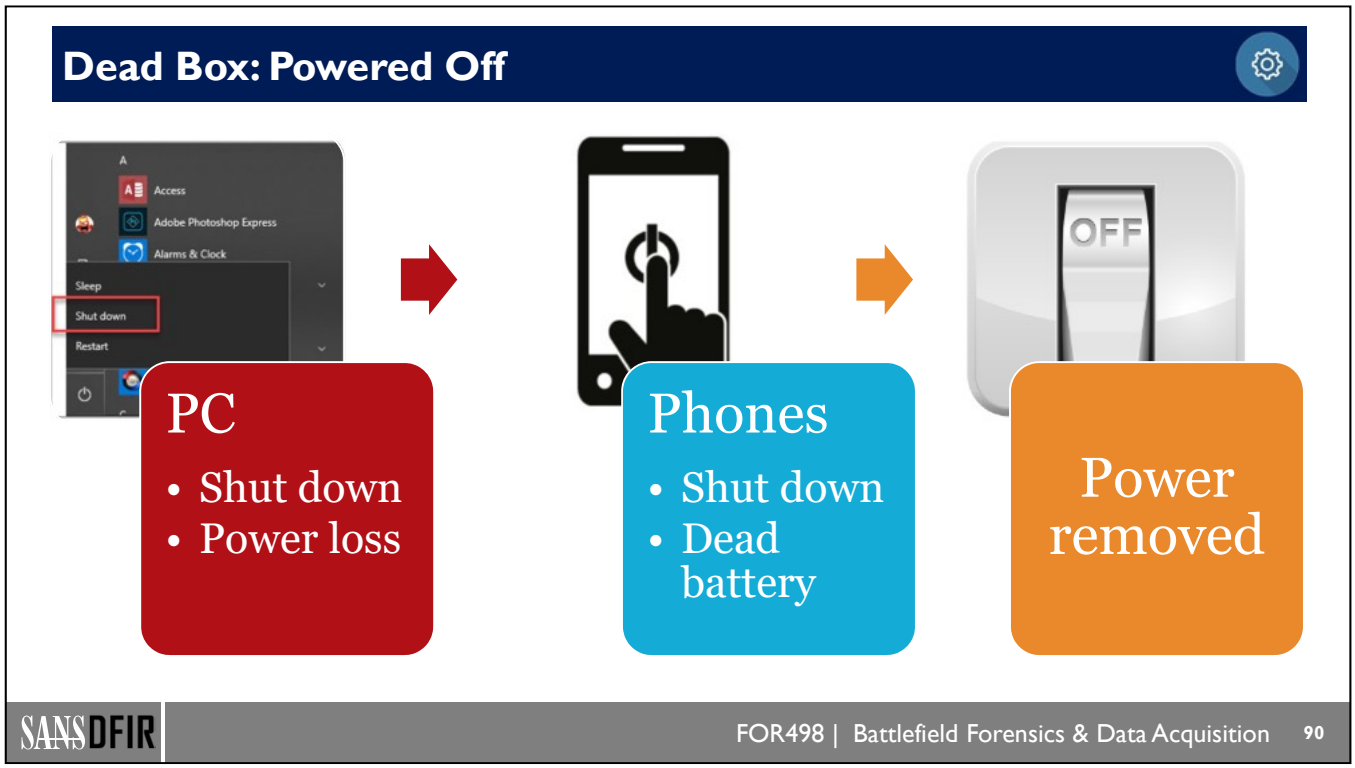
- Powered off
- Hibernating
- Sleeping
- Active

The topmost item, powered off, is the lowest level of a device you may encounter, whereas on the other end, active is the highest level (by "level" we mean your ability to interact with the device in a meaningful way). In both middle cases, a higher level can be achieved by "waking up" the device, which usually involves moving the mouse, pressing a key on the keyboard, opening the lid on a laptop, and so on.

The arrow across the bottom serves as a reminder that, in the last two cases, change can be happening on a device, simply from it being powered on. This includes things like software updates, remote connections, anti-virus scans, and so on. Depending on the type of investigation you are conducting, you may need to take additional steps to isolate active devices from the network to limit tools such as TeamViewer, RDP, VNC, etc. from being able to interact with devices. [1] While these are more traditional remote access tools, you should also consider remote access trojans (RAT) such as Dark Comet, Sakula, and so on. [2]

Each of these states requires different tactics in handling and interacting with the device in order to successfully acquire the device. We will look at each of these states in detail next.

[1] 10 Best Free Remote Desktop Tools You Should Know | `https://for498.com/fzh3y`
[2] The 7 'Most Common' RATS In Use Today | `https://for498.com/lyhpb`



Perhaps the easiest case to deal with, a device is powered off if it is completely shut down. This is generally done via an explicit command on the device, such as using the Start menu in Windows to shut down the computer or using a phone's power button to shut off the phone. When a device is in a powered off state, it generally does not have anything persisting in random access memory (RAM).

When a powered off device is turned on, it will begin its boot sequence. This would be considered a "cold" boot since it was initiated from a powered off state. A "warm" boot on the other hand, would happen as a result of a device being restarted vs. completely shut down and then powered on again. In the case of a warm boot, previously existing data may still be present in RAM since power was never completely removed from RAM. Rather, when the device was warm booted, parts of RAM were simply reinitialized by the operating system[1].

In both cases, the operating system is being completely shut down and restarted, but in the case of a warm boot, the underlying hardware is not completely shut down, in that it continues to run while the software component of the device reinitializes.

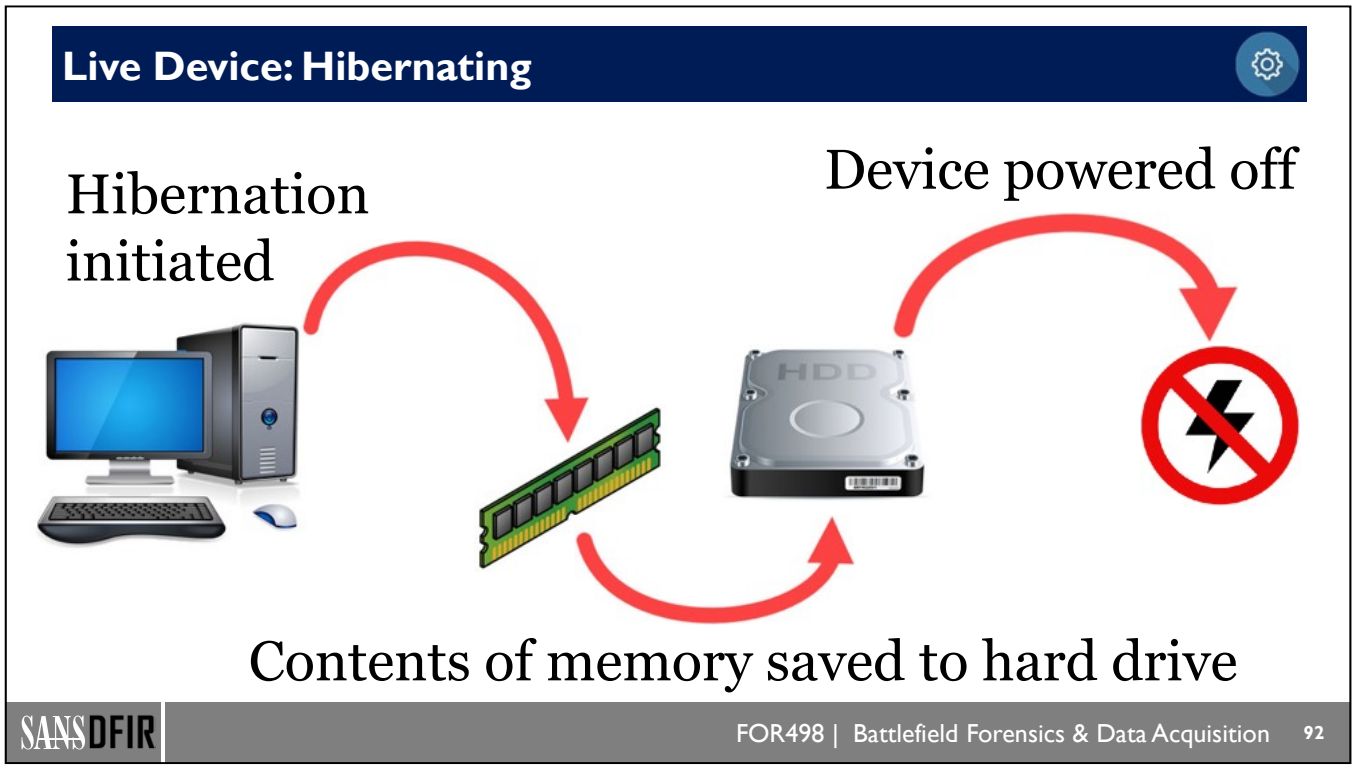
Excluding removing storage from a device and accessing it via forensic software, there is only one way to interact with a powered off device, and that is powering it on. In many situations you would not want to power on a device because it will, during starting up, be changed as a result. Some items that would change include timestamps, file contents, configuration data, etc.

But is there ever a situation where you would turn on a device? Consider the case where you discover dozens (or more) devices at a scene. Are all the devices of interest? If not, do you want to, or should you, take them all? In this situation, it may make sense, and be 100% justified, in powering on a device and looking at it in order to determine if it is a device of interest. For example, take for instance a residence with five people living in the

home, all of which have their own computer. As part of the investigation, the people who live there are interviewed and three of the five do not seem to be involved in any way. Their computers are only used by them, they have passwords, and are in a location only they would have access to, like their bedroom. In this situation, it would be fine to power on their devices, triage them to some degree to confirm their story and rule out that device as being of interest, and leaving the device. Of course, any time you do something like this, it is always a good idea to thoroughly document your actions.

With that said, devices that are suspected to be in the control of a suspect should not be powered on without justification (threat of death, missing person, etc.) and even in these cases, accessing the information on the device another way may still be the better solution.

[1] Difference between Cold and Warm Booting | `https://for498.com/16g8c`



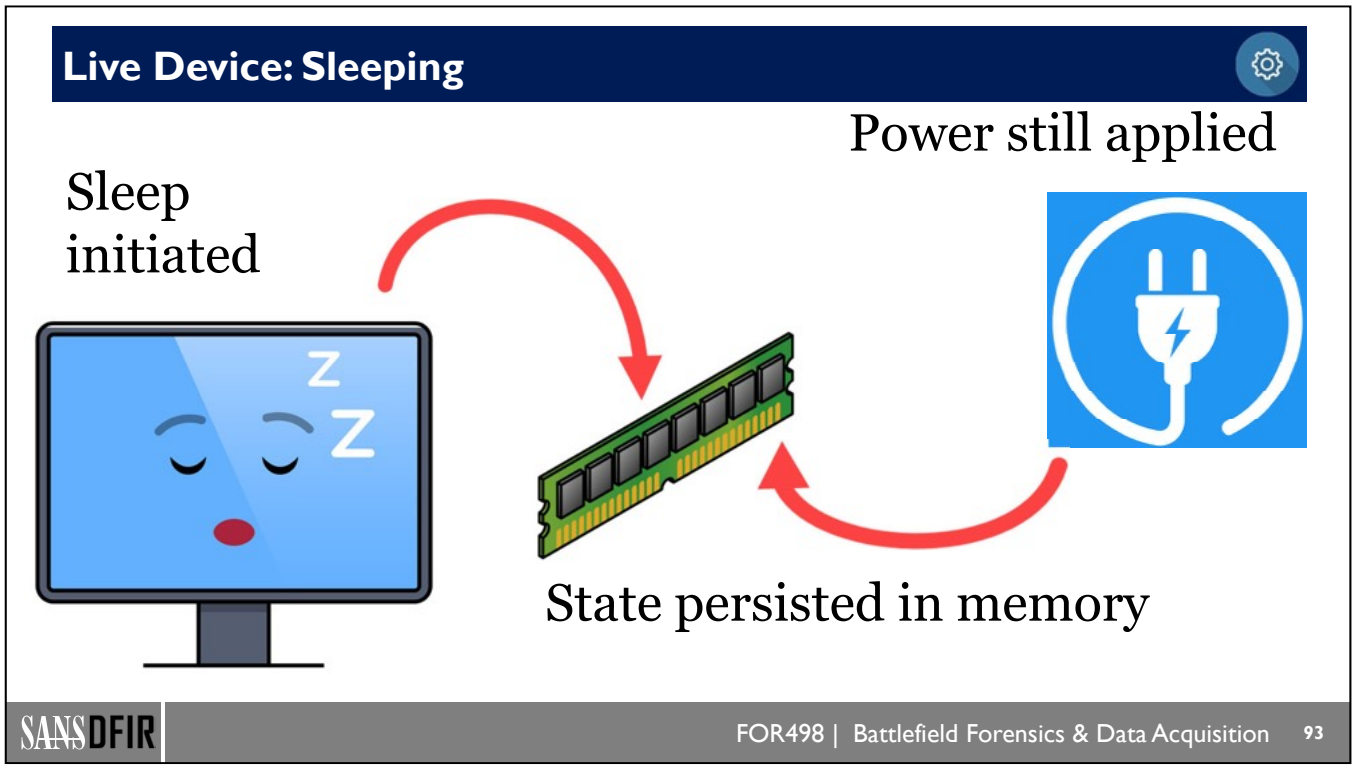
Hibernating allows a device to essentially turn off after saving its state to a file on permanent storage such as a hard drive. When a computer is running, most of its state (i.e. the running configuration of all the software) is kept in RAM. The goal of hibernation is to enter a powered off state while retaining this current state when the device is resumed. To accomplish this, when a device enters hibernation, the contents of RAM are copied to a file and then the device is powered off. When power is reapplied, the device knows it was hibernated and the operating system then takes steps to read the saved state file, then copies its contents back into RAM. Once this is complete, the computer is in the same exact state it was when it was hibernated. [1]

This is essentially a seamless process to the end user, but it has implications to the digital forensics practitioner. Since the state of the computer must be preserved in a file on something like a hard drive, the file used to save the state can, in many cases, be used to recover data that previously existed! In this regard, it is almost like a time machine that looks into the past. As to how far it can look, that depends on when the device in question was last hibernated. Another thing to keep in mind is that, at least as it relates to Windows, Microsoft has, in more recent versions of Windows, changed the way hibernation works in that when a Windows device wakes up from hibernation, much of the file that contains the copy of memory is zeroed out. While it is still possible to recover some useful data via slack space using tools like Arsenal Hibernation Recon, this change on Microsoft's part has made exploiting the hibernation file more difficult.

While we haven't talked about sleep yet, another possibility you may encounter is hybrid sleep. This is a combination of sleep (discussed next) and hibernation in that it keeps a copy of the state of the machine in RAM and saves a copy to your hard drive. This mode is enabled by default on desktop computers but is disabled on laptops. This is useful because it can protect against power outages in that should a power outage occur (which means what is in RAM would be lost), the state of the computer can still be recovered from the hibernation file. [2]

[1] Microsoft Windows shutdown and sleep | `https://for498.com/nj47a`

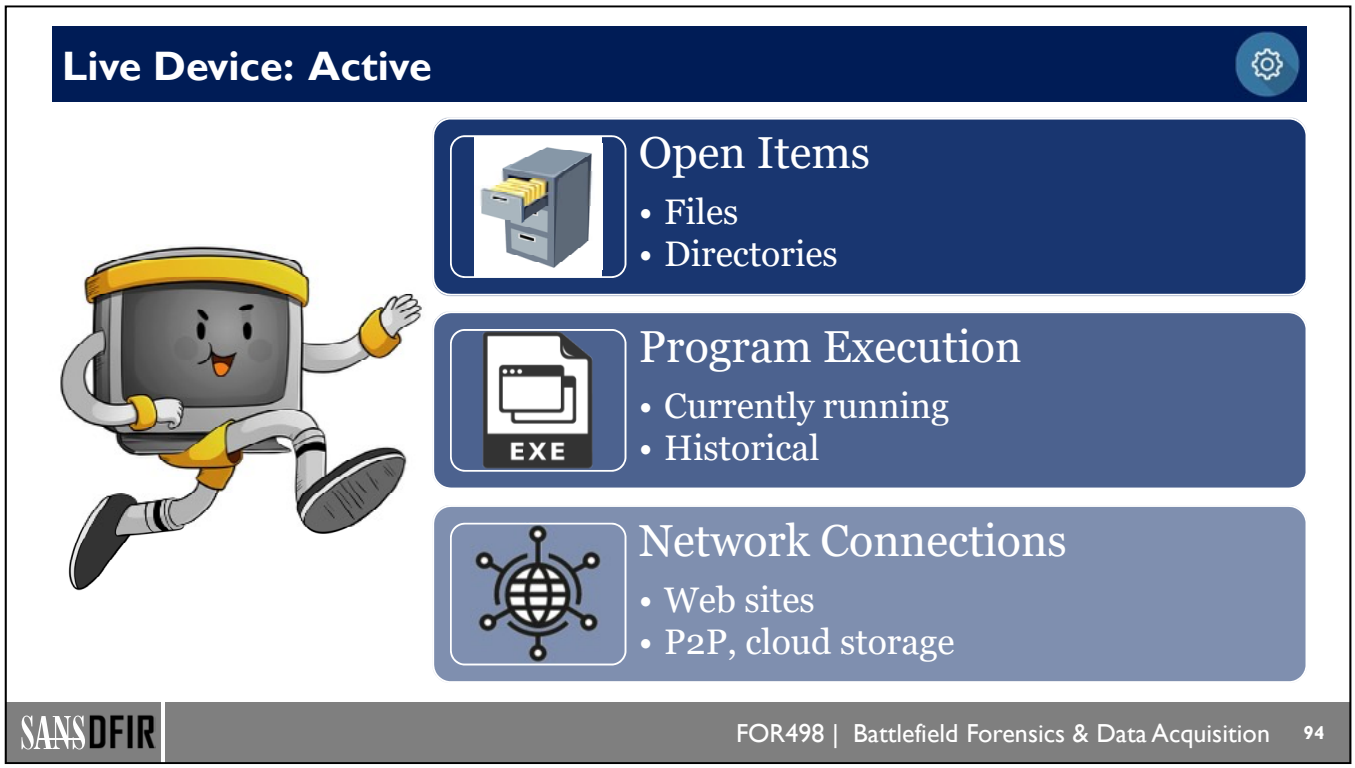
[2] Difference between hibernate and sleep | `https://for498.com/29sc6`



Like hibernation, the end goal here is to preserve the state of the computer to allow for it to quickly resume. One difference with sleep, however, is the requirement for some level of power to always be available to the device. This is necessary because the state of the device is kept in memory only and will only exist while power is applied to the memory. If the device's power is interrupted, the state of the device will be lost, resulting in the need for the machine to fully reboot.

Most laptops automatically go to sleep when the lid is closed. When the lid is opened again, the device wakes up and resumes exactly where you left off. On desktops, pushing the power button quickly puts a machine to sleep in most cases. You can also achieve this via the Start button in Windows by choosing the Sleep option under the Shutdown menu.

The thing to remember here is that a sleeping device is still technically powered on, but it just uses a small amount of power vs. an active device. This is why a device coming out of sleep resumes as fast as it does. Since the device's state is kept in memory, and memory can be accessed very quickly, the device can be back in its active state (discussed next) in just a few seconds.



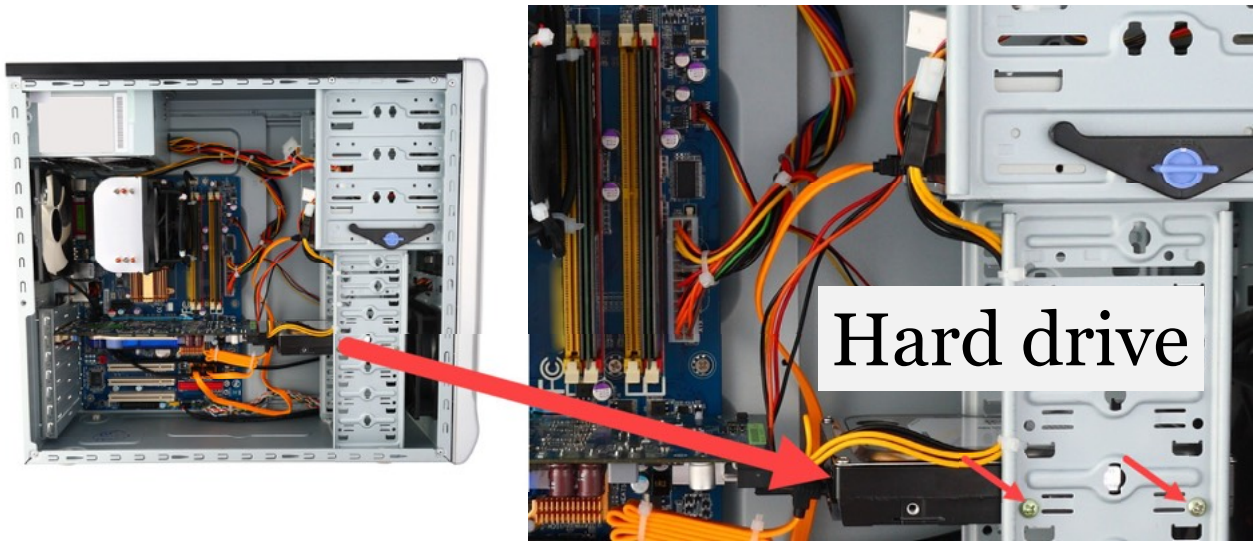
When a device is in its active state, it is fully ready for use. Programs can be started, files can be edited, and so on without having to resume from a previous state. When a device is in this state, the following information can be ascertained (this is certainly not all inclusive):

- Running programs
- Open files
- Active network connections
- File system activity
- Historical evidence related to program execution, files opened, directories visited, etc.

Of course, to interact with a device in this state, a user would have to be logged in and the desktop visible. When encountering a device in this stage, nothing is stopping you from browsing the file system, running programs, opening files, and so on, but keep in mind that doing any of these actions will result in change happening that you must be able to account for. Because of this, it is better to ***not*** interact with a system manually, but instead perform automated triage that is repeatable, verifiable, and minimizes the amount of change on the system. In future sections we will see techniques that fall into both categories.

The last thing to consider when dealing with an active device is how to determine if any encryption is present. This is critical, because if encryption is present and the device is powered down for collection, access to the data will be lost unless the password is known. We will discuss this topic at length in another section and show techniques to detect encryption on a device and, if any encryption is found, how to deal with it to avoid it becoming a problem for your case.

Accessing a Device: Desktops



For many devices, accessing the internal components will be as simple as unscrewing a panel that allows for access to the device's internal components. The example above is a typical desktop computer tower. Most cases have either a set of thumb screws or more traditional screws that hold the sides of the case in place. Unscrewing these fasteners allows for the side of the case to be removed. Some cases also have a button that releases the side of the case.

In most instances, it is obvious how to gain access to the inside of the computer, but when in doubt, look up the make and model of the case online in order to locate a user manual which explains how to access the internals of the case. Once access is gained and the panel removed, inspection of the interior can begin.

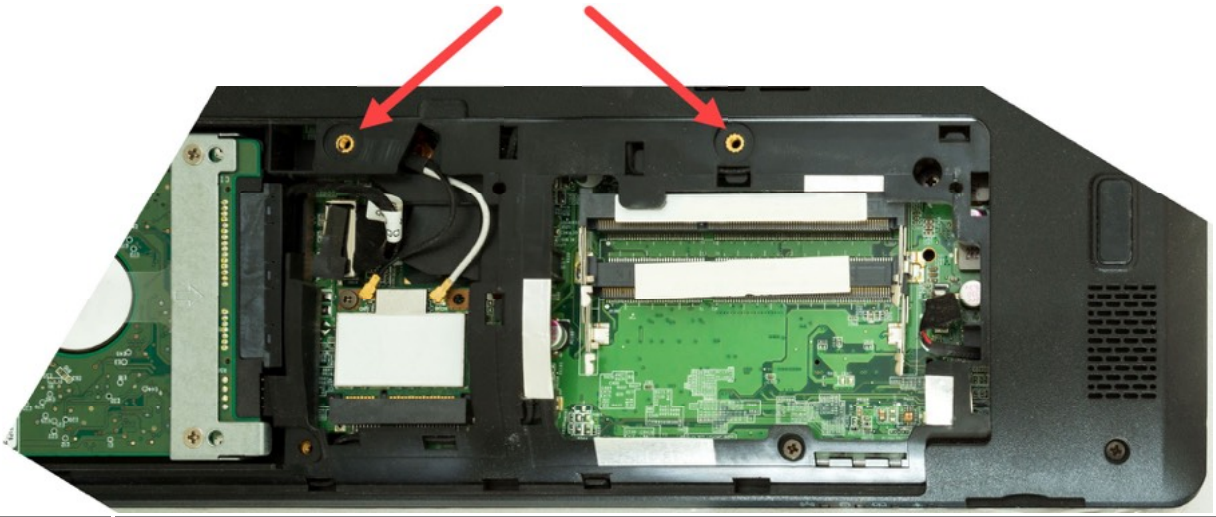
We will primarily be interested in storage devices inside the computer, which will typically come in two forms: IDE and SATA drives. IDE devices have a flat ribbon cable whereas SATA devices have a much smaller cable. The hard drives themselves look the same with respect to the interface used. In the above example, a single SATA hard drive is present. While another SATA cable can be seen, this cable is going to a DVD or CDROM drive located in the lower slot of the topmost cage.

Once the storage devices are located, determine how to remove them from the case. In the example above, two screws are seen holding the hard drive in place. There may, or may not be, an additional two screws on the other side as well. In some cases, screws are not used to secure the hard drive in place. These cases typically have a caddy system that can be removed from the case without needing to remove several screws.

It should be emphasized that you fully document the inside of every device before removing anything from the case. This includes written documentation as well as a photograph or video showing the state of the device's internals.

Accessing a Device: Laptops (1)

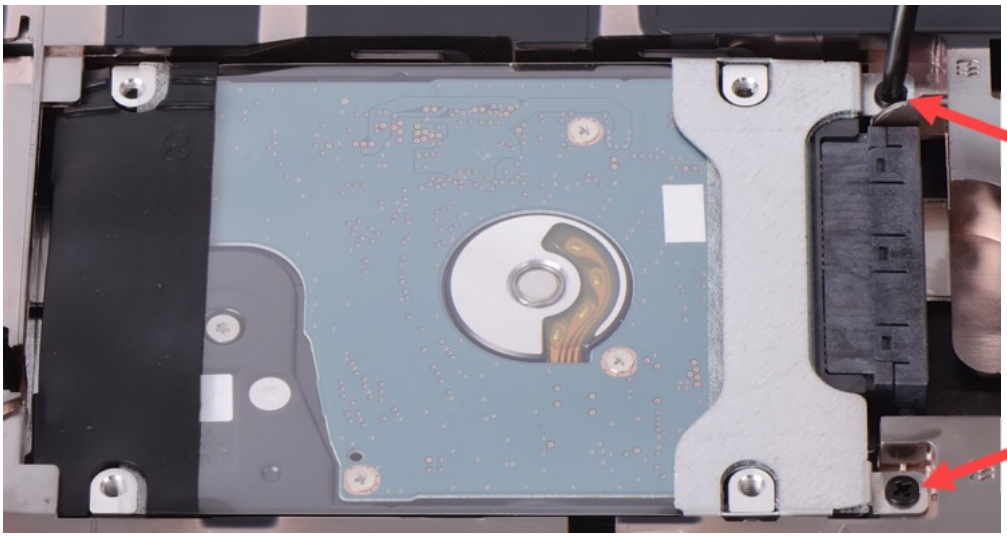
Remove one or more screws, and then the panel



When it comes to laptops, things can become more complicated very quickly. For most laptops that allow for accessing internal components, the components are typically located under panels on the bottom of the device, or in some cases, under the keyboard.

Because of how densely populated most laptops are (because space is at a premium), take special care to avoid damaging or disconnecting ribbon cables or other connections between components. This is usually more of a concern when removing a laptop's keyboard vs. removing panels on the bottom of the laptop case.

Accessing a Device: Laptops (2)

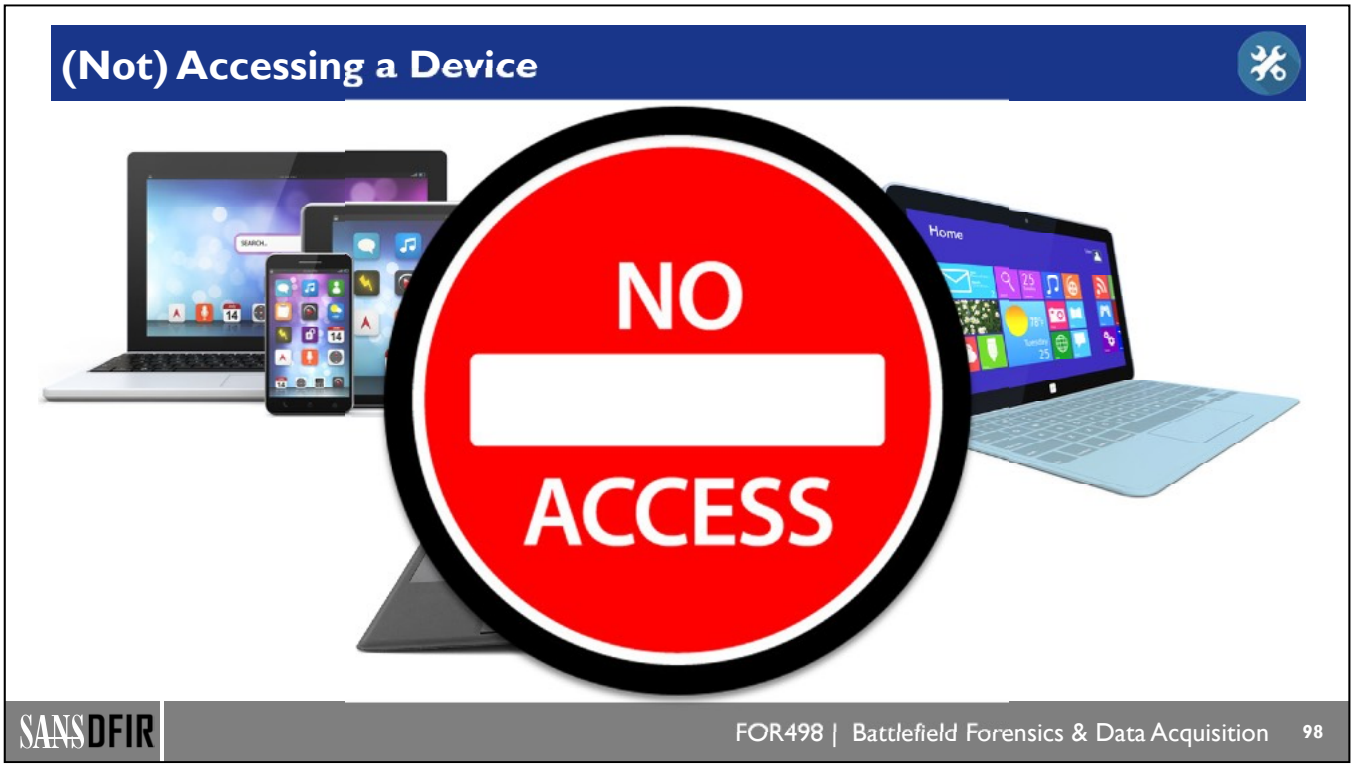


Some laptops, typically more powerful ones like those used for gaming or a portable workstation, have multiple peripheral slots that can accommodate multiple hard drives. As we saw with gaining access to a desktop case, it is often a good idea to locate a user manual for the device you are working on to ensure you have accounted for all possible storage devices.

In the example above we see a SATA hard drive on the bottom of the device. On the right are two screws that hold the hard drive in place. To remove the drive from the laptop, unscrew both screws and carefully slide the entire hard drive to the left. This disengages the drive from the interface. Most of the time there is a tab of some sort to aid in moving the drive away from the connector.

In a number of recent examples, our lab has been tasked to do forensics on laptops that didn't appear to have any hard drives in them, as the spaces that housed them were empty. Instead, the laptops contained solid state drives of the NVMe type, so they were on the motherboard itself. Be aware of this, lest you allege that someone has removed the hard drive from the machine.

Once access has been gained, we once again need to document the state of the device via notes and photographs just like we did with desktops.

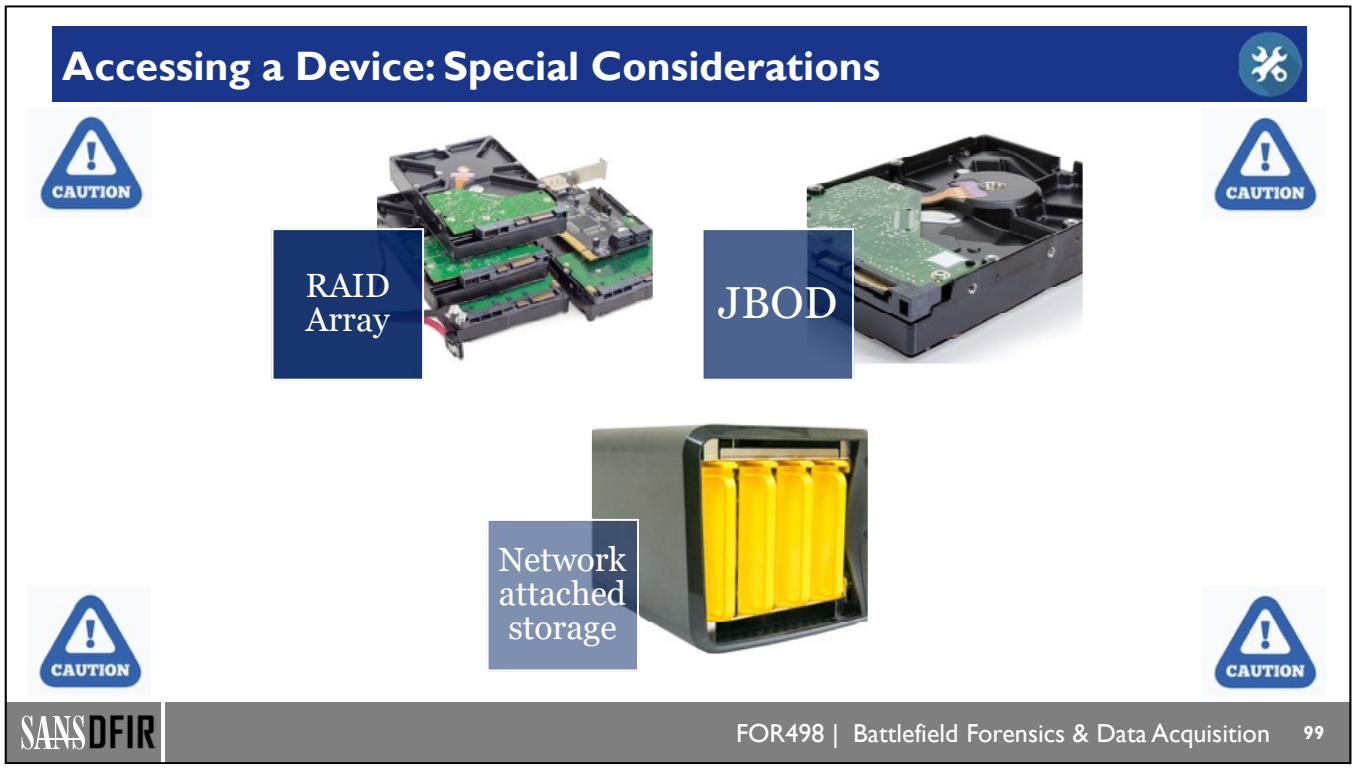


In some situations, it will not be possible to access a device's internal components. Many newer devices are not end user serviceable and are not intended to be open. In many cases, the devices are not upgradable at all. Another reason is that the goal of a lot of newer devices is to make the devices as thin and light as possible. In some cases this means certain kinds of devices may not even be present, such as DVD drives or a network port.

Even in the cases where you do manage to access the internals of a device, directly accessing the storage is not possible because the manufacturer uses epoxy (or equivalent) to secure the storage drive inside the device. In these cases, attempting to remove the drive from the device may irreparably damage the drive.

When we encounter this situation, is all hope lost? Certainly not, but our traditional approach to this problem will no longer suffice. Each device will require a slightly different technique, many of which we will specifically cover in this course. These techniques may involve booting a device into target disk mode or in some cases, starting the device and imaging it while it is running (or perhaps imaging it even before it is shut down for the first time).

Regardless of whether we are dealing with a laptop, desktop, or some other device, there is at least one other step to take before anything is physically removed from the system This will be covered in the next section. For now, we just want to have an understanding as it relates to gaining access to where storage devices are located.



Of course, like just about everything in digital forensics, there are exceptions to things. In some cases, even if you gain access to the internals of a device, some interesting things may be going on, including some level of Redundant Array of Independent Disks (RAID) [1] or Just a Bunch Of Disks (JBOD) [2] storage. A RAID array differs from a JBOD array in that each disk in the JBOD array is simply combined to create one large storage volume without any redundancy. Compare this with RAID arrays which involves some level of striping (using the same size disks) and parity, or a combination of the two.

How can you go about determining when one of these special circumstances is in place? Let's look at RAID and JBOD first. A RAID array is going to involve at least two hard drives but could involve many more. These hard drives will be connected to either a dedicated controller usually found as an add-on that goes into one of the peripheral slots on the motherboard. Many higher end motherboards also have some level of RAID support built right in, but these implementations are usually handled via software rather than hardware, which is what a dedicated controller would provide.

In many cases, a simple visual inspection may tip you off to the fact that one of these situations may be present, but to confirm this, it is often necessary to boot the computer into at least the BIOS to see how things are set up. When a dedicated controller is present, it will present some kind of management interface on bootup and offer a hot key to access its configuration data. Going into the BIOS or the controller's configuration software is the best way to determine exactly what you are dealing with.

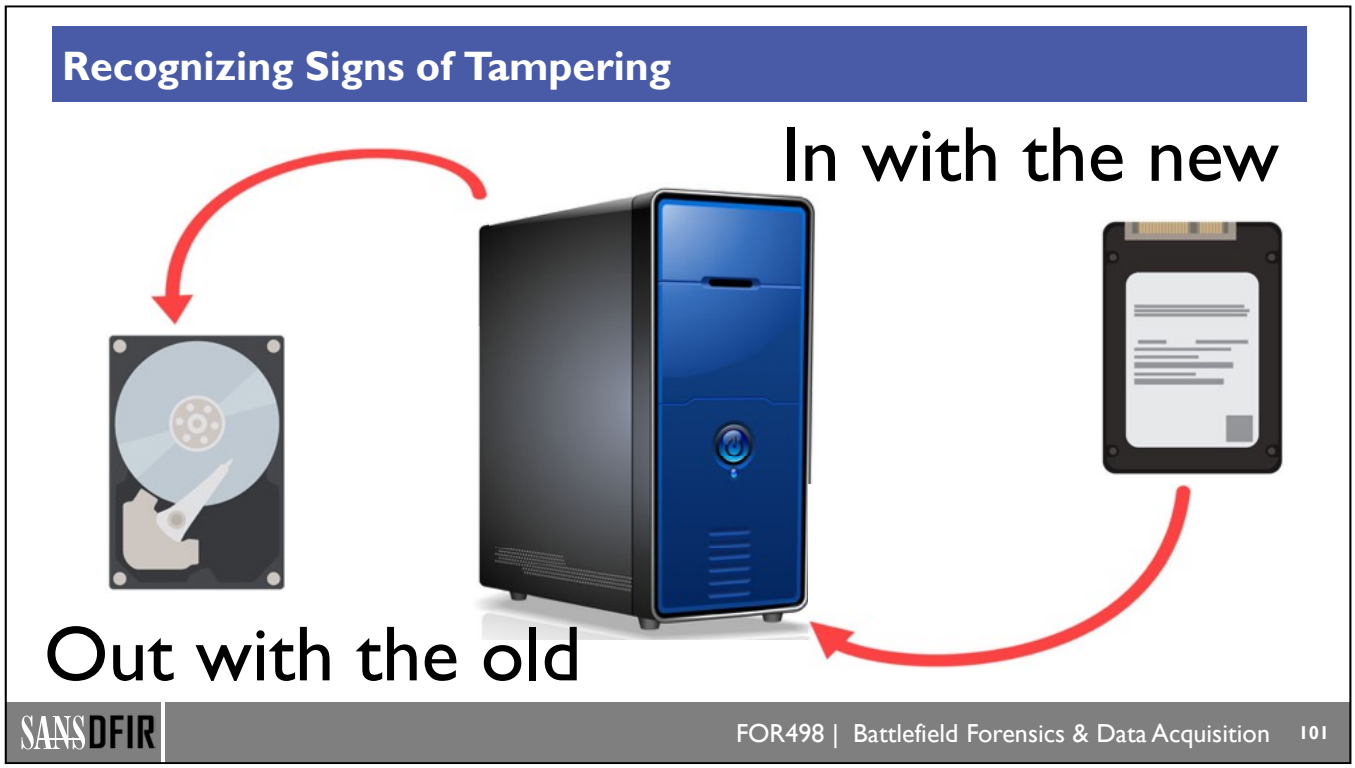
Either of these situations can also be present in commercial Network Attached Storage (NAS) devices as well. In some cases, the implementation of RAID is proprietary to the vendor and may not be known outside of that vendor.

So what does all this mean to us from an investigative perspective? When we encounter things such as a RAID/JBOD array or a NAS device, it may make more sense to image the storage presented by these items

directly vs. trying to image each separate piece that goes into them. If this approach isn't done, you must now concern yourself with parity, stripe size, and a host of other variables that would all be necessary to understand in order to rebuild the storage volume as its presented by whatever is controlling the individual disks. This is no easy feat!

[1] RAID Overview | `https://for498.com/j8ozx`

[2] What is JBOD | `https://for498.com/ivn-r`

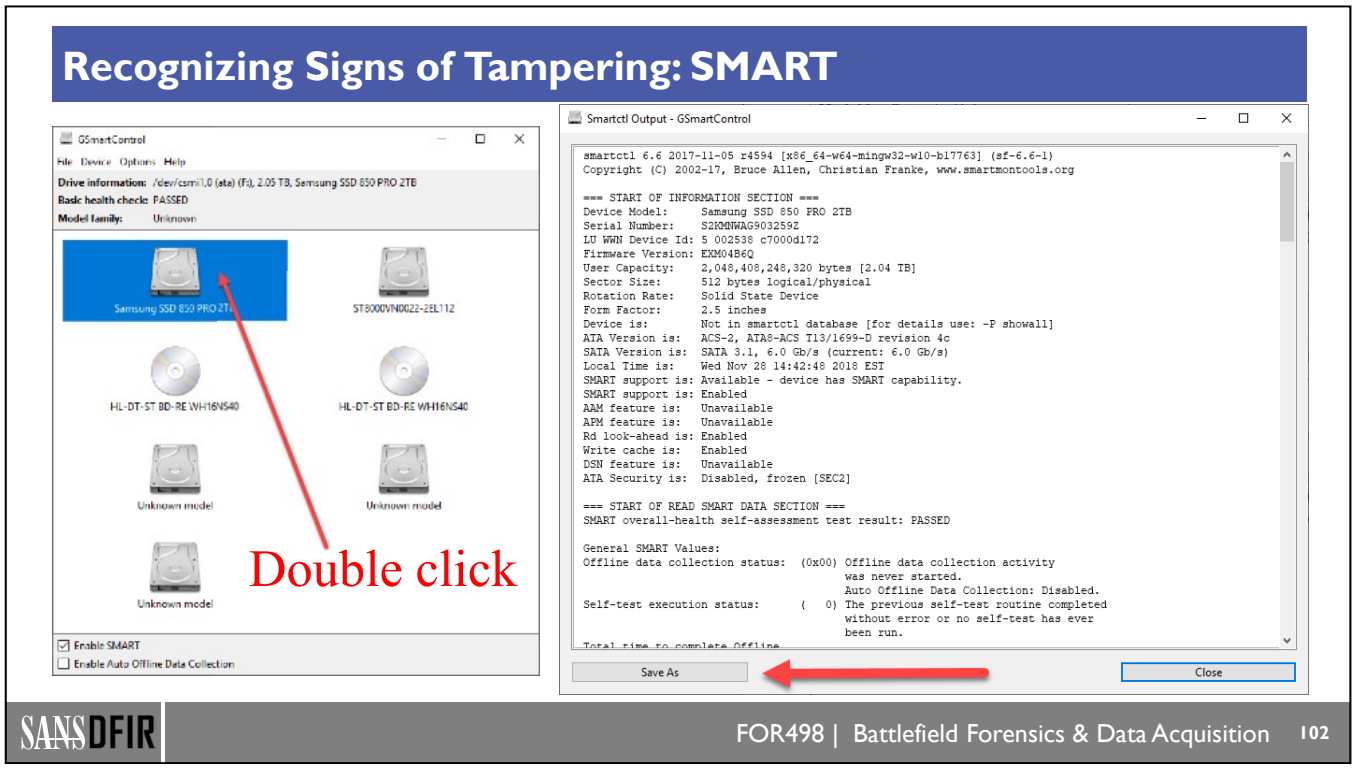


Another aspect to consider during an investigation as it relates to hardware has to do with tampering. In many cases when we think about tampering, we think of it happening regarding the operating system where files may be deleted or altered. In this situation however, we are more concerned with hardware tampering, specifically as it relates to the storage in a device.

For example, upon inspecting the hard drives inside of the case, there is a relatively newer hard drive installed from the last few months, yet the machine is three years old. In other cases, what if the data on the hard drive is so pristine it looks like the computer is barely used at all? In some cases this may be perfectly legitimate; in others it may be a deliberate attempt to hide meaningful data from investigators.

During a search warrant, it was determined that the hard drive in the computer as found during the search was not the hard drive used by the subject in the house. The hard drive in the computer tower had nothing of interest on it. During the search, a loose hard drive was found in the room where the computer was used. When this hard drive was previewed, it contained a wealth of information related to the investigation. A follow-up interview with the subject revealed he was swapping the two hard drives in and out of the computer to hide his activities from his family members.

In the above case, a much newer hard drive was being swapped in and out of the computer and, by just looking at the computer itself, this would be difficult to determine unless the newer drive was found in the old machine. The interview was used to corroborate investigator's suspicions, so this serves as a reminder to not completely rely on what a device is telling when there may be more information available from other sources.



Another investigative avenue beyond physical inspection is looking at a storage device's Self-Monitoring, Analysis and Reporting Technology (SMART) [1] data. SMART is a monitoring system found in hard drives and SSDs that keeps track of metadata about the drive, including read error rate, performance, reallocated sectors, seek time, power-on hours, and others.

So how exactly can we leverage SMART to know when something suspicious has happened? In the case of an older computer, say, from three years ago, one would expect the hard drive that came with the device to have been used for many hours over the course of those three years. By looking at the power-on hours, which counts the total number of hours the drive has had power, an investigator can gauge whether a newer device (with just a few dozen hours) is in a computer vs. one with 25,000 thousand hours on it.

Since the devices themselves store the SMART data, we as investigators need a way to query the data. There are several programs (some free, some commercial, including forensic suites) that can read the SMART data on devices. The tool we will look at is called GSmartControl [2], which is a graphical interface to **smartctrl,** often found on Linux systems, to monitor disk health. It allows you to inspect the drive's SMART data to determine its health, as well as run various tests on it.

Usage of the tool is simple. GSmartControl comes in portable versions for both 32-bit and 64-bit systems. Once the zip file is downloaded and extracted, simply double click on **gsmartcontrol.exe** and the interface will load. Once loaded, all drives connected to the system will be interrogated and displayed on the main interface. To drill into a specific device, double click on one of the icons and a new window is shown that contains complete SMART details for the device.

[1] S.M.A.R.T. information | `https://for498.com/hsduf`

[2] GSmartControl | `https://for498.com/olre9`

Hashing (1)

- AKA – one-way transformation
- Variable input = fixed length output
- Key length = hash length
- Cannot be reversed
- Used for message integrity

Common hash algorithms

MD5	- 128 bit
SHA1	- 160 bit
SHA2	- 256 bit
	- 512 bit
SHA3	- 256 bit
	- 512 bit

2af505efd72694140b32d0a84070fa31

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 103

Hashing, otherwise known as a one-way transformation, is the de facto standard of verifying the integrity of electronic evidence. A hash is a hexadecimal value created through the use of a specific mathematical algorithm in conjunction with arbitrary data input. The idea is that if a specific piece of data has a specific hash algorithm applied to it, this will result in an irreversible hash output value. No matter how many times the specific piece of data is moved from medium to medium, and no matter how much time passes, and no matter who is subsequently analyzing it, if the same hash algorithm or math is used, even many years later, the hash output will be the same, as long as the input data has not changed.

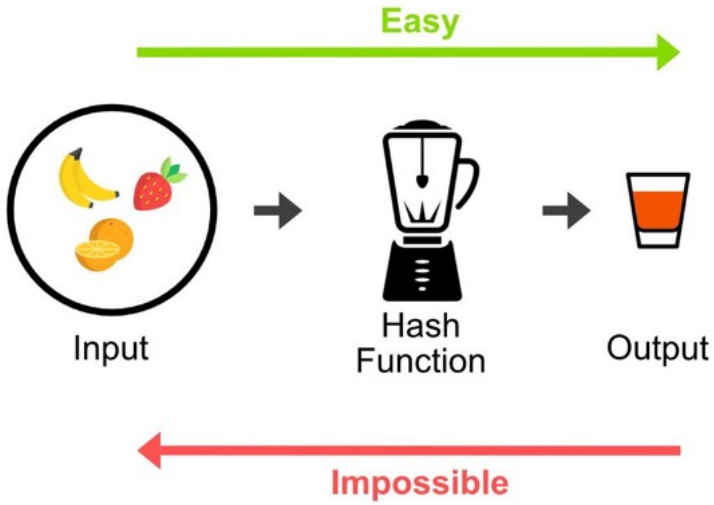
Two very common hash algorithms in use today are MD5 and SHA1. MD5 stands for message Digest 5th version, and SHA stands for Secure Hash Algorithm. MD5 is a mathematical algorithm that can take any size of input data from one bit to many terabytes of data and create a 128 bit output value. This value is represented in hexadecimal characters. Although this output displays on the screen as 32 characters, because they are hexadecimal characters, it takes two characters to represent one computer byte. Thus the MD5 output is actually 16 bytes in length.

It has commonly been seen by the author in court that many examiners have no idea how to describe a hash algorithm other than the most basic description. That being, if I apply my hash to this data and it doesn't change from the original hash then the data hasn't changed. This is not a clear nor court acceptable explanation. The examiner must understand what 128 bits of data looks like and how it translates to 16 bytes on the screen. To the uninitiated, the output would absolutely look like 32 bytes on the screen.

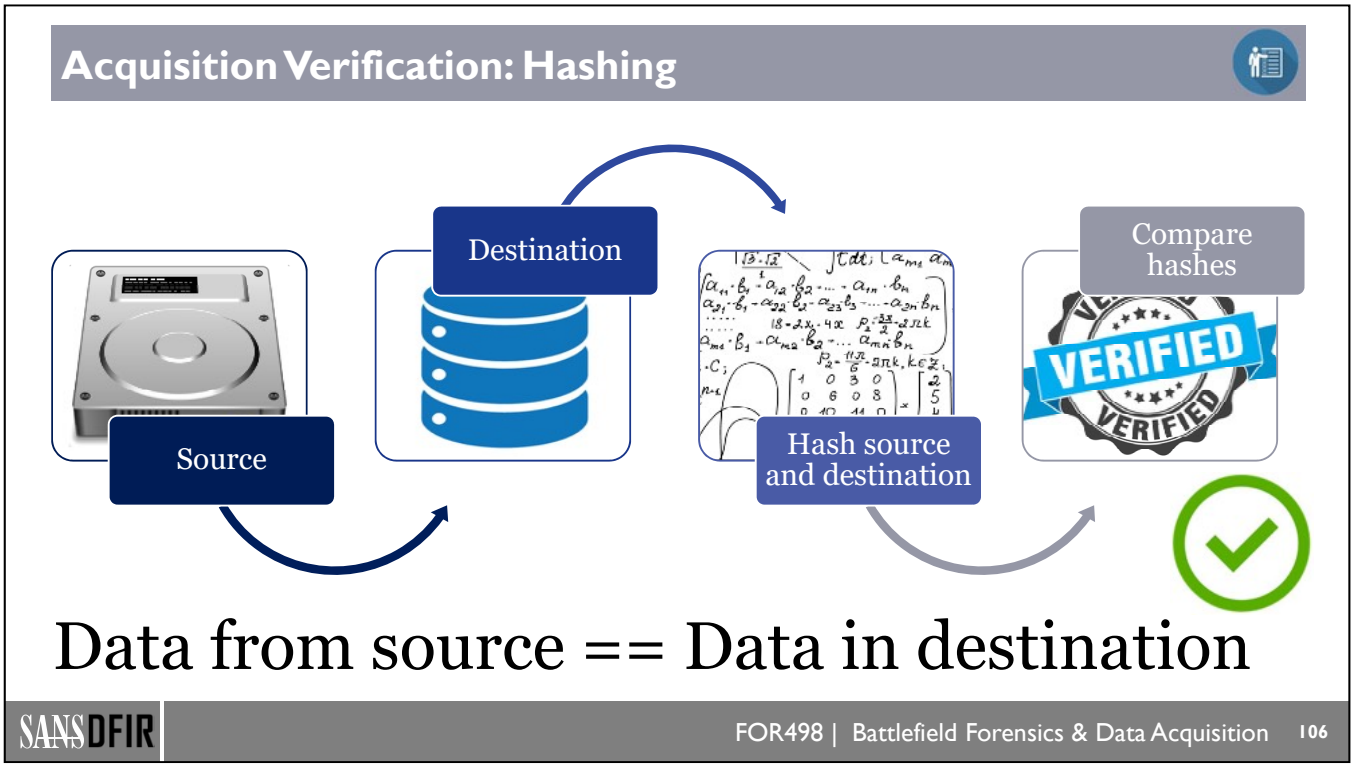
Although much has been made in recent years about MD5 hash collision, it simply is not something we need to be concerned about. The notion that we can no longer use MD5 hashing because of collisions is simply allowing our lack of understanding to get in the way. Is a hash collision possible? Absolutely. Let's take a moment to understand what a hash collision is.

In the case of MD5 being a 128-bit algorithm, we have 340 undecillion possible combinations of hash output. Is it possible that there are more than 340 undecillion possible combinations of characters, words, digits, phrases, compounded with every language in the world? Again the answer is absolutely. The notion of a hash collision is that two pieces of dissimilar data will randomly and coincidentally end up with the same hash value. Although this is possible, it certainly is not probable. You stand a significantly higher probability in matching the DNA in two unrelated human beings than you do of matching hashes with two dissimilar pieces of data. You also stand a much higher probability of matching fingerprints on two unrelated human beings than you do of matching hashes against two dissimilar pieces of data. In a court of law, we still rely very heavily on DNA and fingerprints. Although I would not use MD5 or SHA1 hashing as security for passwords, it is perfectly acceptable to use it as a basis of integrity checking for evidence.

Hashing (2)



This page intentionally left blank.



In a DFIR sense, what does verification mean, and why is it important? First, let's talk about why it's important. Verification is important because it tells us definitively that the copy of data we made is the same as the original. This is critical because we generally want to avoid working on the original device (for many reasons), and making a copy allows us to do this. As to how we do verification, the most common way is via cryptographic hash [1], like MD5 or SHA-1. A cryptographic hash takes some input, performs some math on it, and returns a consistently sized output string after processing all the input. The string that comes out of the hash algorithm is usually referred to as the hash or digest value.

While we will not be getting into the mathematical properties of exactly how the hashes are calculated, what is important to understand is that any change in the input will radically alter the output hash (called the "avalanche effect"). Even something as simple as changing one bit on a source drive will cause a very different looking hash value to be generated. An example will help.

Consider a text file that contains the string **This is a hash test.** Calculating its MD5 and SHA-1 produces the following:

SHA-1	92CEFE9E0742D585154401CAF36EE2AE83FD1498
MD5	B8882B86C15FEF3E410997F48F97105C

However, if we change it to **This is b hash test.** (increasing the hexadecimal value for 'a' by a single digit, from 61 to 62, the following hashes are generated:

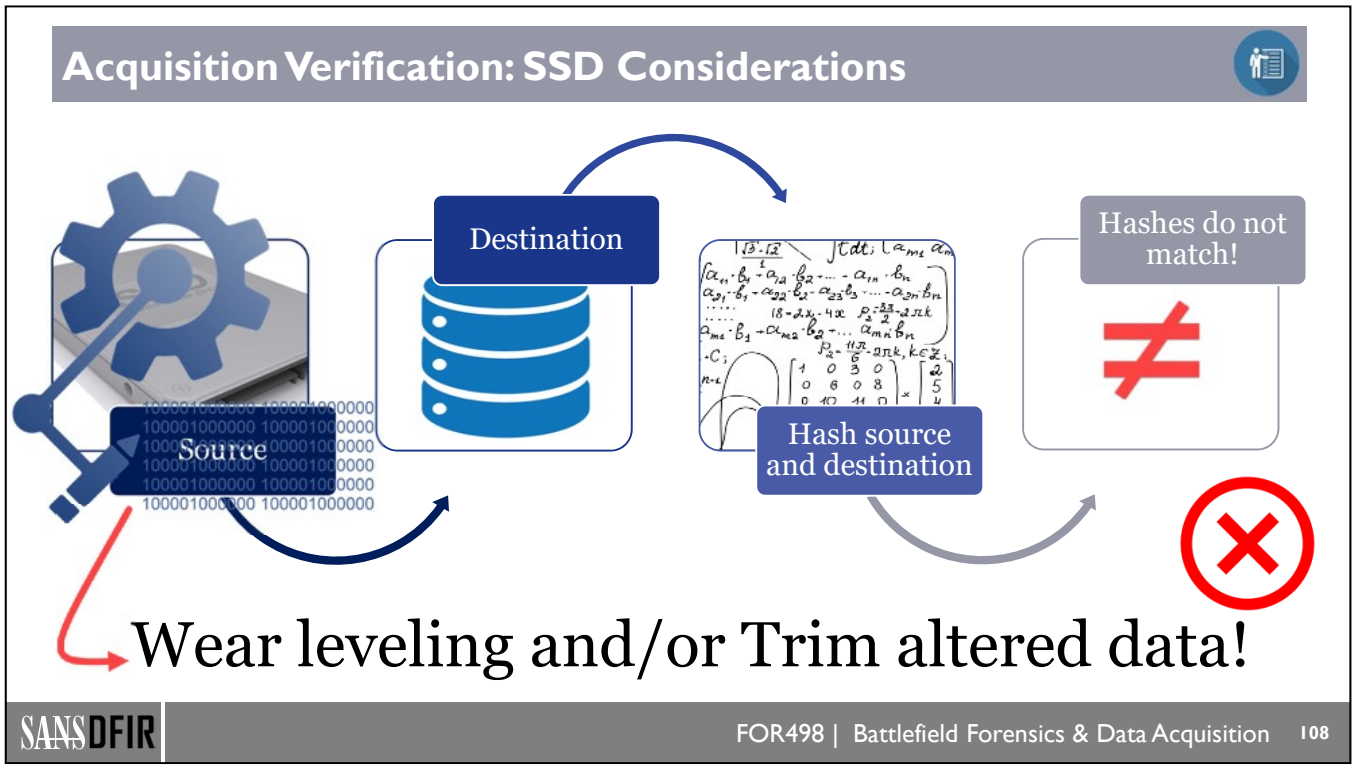
SHA-1	B8BE0332311881189901DE311514A95836098D7D
MD5	7D6D8E6B4F31499029CB71037FC8A945

The key thing to notice here is how the hashes are entirely different. In other words, changing a single byte in the source resulted not in a single character in the hash changing, but a radically different hash being generated.

Once the source is copied to the destination, the hash of the source is calculated, and the hash for the copy is calculated. Both hashes are then compared and, if they match, the data has been copied accurately. Once the hashes match, we can work with our copy of the data without worrying about the source anymore.

But is this always the case? Generally speaking, yes. Most of the time the source hash will match the destination. Next, let's look at a situation that is becoming more common where a mismatch may occur, and talk about what is happening.

[1] Cryptographic Hash Function | `https://for498.com/aevm4`



Here we have a very similar situation in that we have a source drive, copy it, and calculate hashes, but when it comes to verifying them, we do not get a match! If you look closely you will notice in this case the source drive is a Solid-State Drive (SSD) which is based on flash memory vs. a traditional spinning platter we previously looked at.

Before we get into what can cause a hash mismatch, we need to understand a few key features of SSDs, namely, wear leveling and trim. Trim effectively clears the free space (i.e. not currently being used by a file tracked by the operating system) of an SSD whereas wear leveling serves to balance the flash memory cells where data is written so as not to "fatigue" them, since flash memory cells only have a certain number of write cycles before they start failing.

Wear leveling and trim operations are not solely handled by the operating system, but rather, they are handled by the firmware of the SSD itself. This means that when power is applied, the SSD may be making decisions to move things around, wipe free space, and so on, due to its maintenance operations.

What do you do if you image a solid-state drive and the hash does not match? This is a question many in the field have wrestled with, and some have even made the mistake of deleting the first image and conducting a second image in hopes that the hashes would match. The way most imaging tools work is by hashing the drive to be imaged, then copying the data into a corresponding image file format (e.g. DD, E01). Once this is done, the image is hashed and compared to the original hash. If, during the imaging process, the SSD performed trim or wear leveling commands on sectors that were already read, this would result in a different hash being calculated, if the drive is imaged again.

Should this occur, the recommended practice is to keep the original image and reimage the drive again. By conducting a comparison of the first and second image you will likely see the deleted files and unallocated space in the first image that were destroyed by trim or wear leveling that are no longer in the second image. With the comparison of the two images, and by understanding and explaining wear leveling and trim to the courts or general counsel's office, the concern over unmatched hashes is easily mitigated.

Pulling the Plug and SSDs

Many discussions about proper acquisition techniques have discussed whether to pull the power from a running system to "freeze" the state of the hard drive from accidentally erasing data. Depending on the circumstance, this might be a good idea– except when it comes to solid state drives. This action could cause some serious problems.[1] SSDs are not meant to immediately cease functioning. In fact, there is a good chance that you could brick your SSDs with a power failure[2]. Most drives can repair themselves automatically if reconnected to a power source, but it leads to another thought. If the SSD can self-write and self-repair its own data, then an investigator has very little control over the actual data while it is stored on an SSD.

In fact, cutting the power could be the worst option for trying to ensure proper collection of data on the drive. The repair operation could be doing many things, including performing trimming operations and wear leveling while the drive is self-repairing after a power loss. Some thinking has recommended that the best options might include imaging the system live. The longer you leave the solid-state drive running in any form might corrupt the data. Powering off the system using a normal shutdown can also engage drive trimming/optimization or additional wear leveling as a result of data being closed and written as a result of the shutdown process. The only option that might result in the best evidence would be like imaging memory—doing it on a live system through live acquisition.

There are no firm recommendations as to what best practices are so far; however, some in the forensics community might discover that their procedures need updating when they would deal with drive acquisition of solid-state drives.

The usual process of write blocking drives using a standard write blocker will protect the drive from only accidental writes from the connected operating system. However, the drive's controller itself is fairly robust and is likely to perform wear leveling and trimming operations when the drive is powered on. Given that so much is unknown on each vendor's implementation of their own SSDs, it is likely that there is a good risk associated with assuming that the SSD integrity will be achieved with a write blocker for prolonged periods.

As a result, it is recommended that a write blocker is used to ***image*** the hard drive only. It is not recommended to perform analysis on an SSD connected via a write blocker as it might be plugged in for an extended period, increasing the chances that the controller-initiated SSD management could occur, resulting in a loss of integrity.

References

[1] Understanding the Robustness of SSDs under Power Fault | `https://for498.com/zv-0j`
[2] Are power outages killing your SSDs? | `https://for498.com/zd-p0`

Summary

- There can be a number of different states that you may find a computer in
- System hibernation and sleep state are not the same thing
- Accessing storage devices physically can sometimes be a challenge
- An examiner must be able to recognize signs of tampering
- Acquisition verification is how you prove that nothing has changed during the acquisition process

This page intentionally left blank.

Exercise 2.4

Write Blocking Methodologies

Synopsis: In this exercise, you will use the Forensic UltraDock to perform hardware write blocking, and then you will use SAFE Block to perform software write blocking.

Average Time: 30 Minutes

This page intentionally left blank.

Exercise 2.4 Takeaway

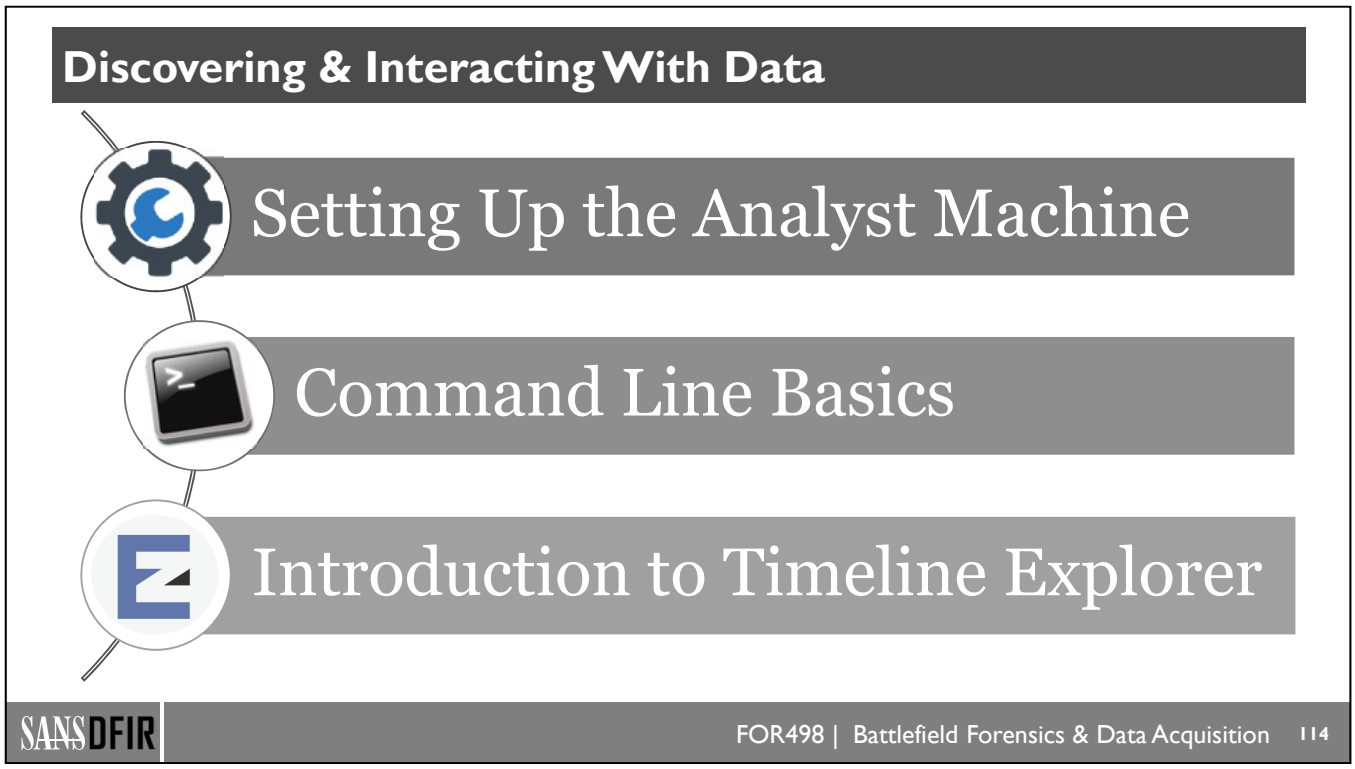
- Any time you are acquiring media that has been removed from a machine, it must be write blocked.
- Until recently, hardware write blocking was the accepted standard.
- There is no tool on the market like SAFE Block for managing write blocking of non-standard devices.
- SAFE Block can write block virtually anything that can be connected to a computer via any interface.

This page intentionally left blank.

FOR498.2: Portable Devices & Evidence Acquisition Agenda

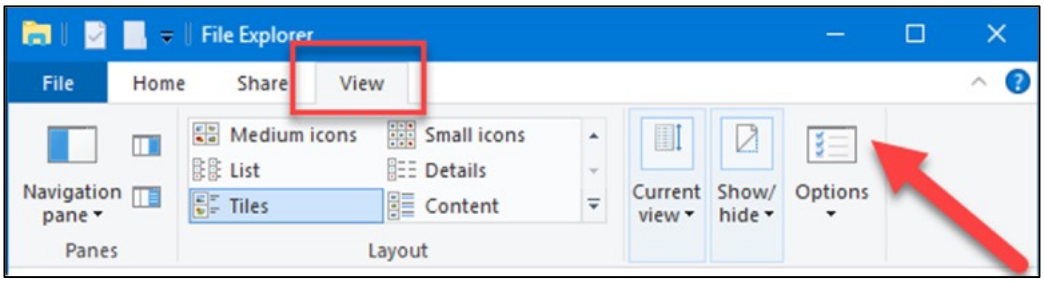
- 2.1 Portable Device Acquisition
- 2.2 Portable Device Analysis
- 2.3 Acquisition Hardware & Software
- 2.4 Acquisition Methodology
- 2.5 Discovering & Interacting with Data

This page intentionally left blank.



This page intentionally left blank.

Windows Configuration (1)

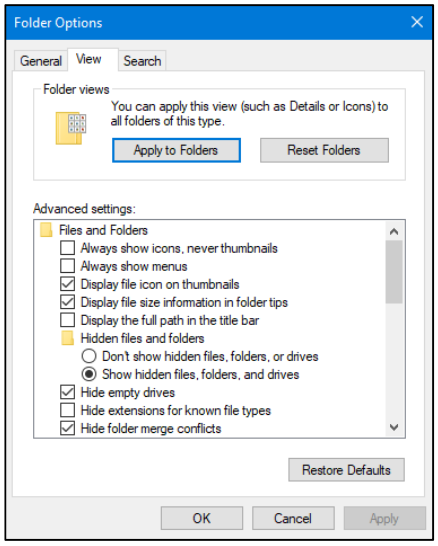


While many options can be adjusted on the View tab, using the Options dialog allows for configuring all folders to use the same settings

Before we begin looking at data, some basic configuration is needed. Because we will often be dealing with files and directories that Windows considers to be related to the operating system, we will want to configure Windows to show us these files and directories. The next few slides will provide an overview of these changes. At the end of this section, we will be configuring our virtual machine to use these settings.

The configuration is done using File Explorer's Options dialog. Once File Explorer is started, clicking the View tab shows several more groups of options. Here, the Details Layout can be selected, and other various tweaks made. Most of the options we will want to change will be found in the Options Dialogue box.

Windows Configuration (2)



Change settings for:

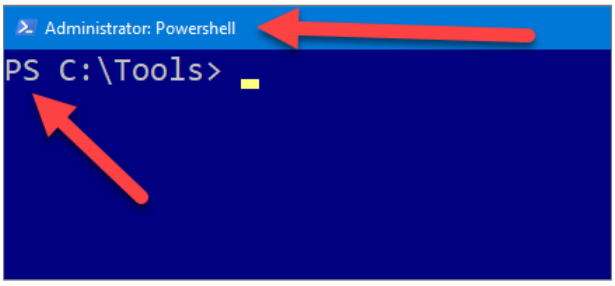
- Show hidden files, folders, and drives
- Hide extensions for known file types
- Hide protected operating system files
- Launch folder windows in separate process

Once in the Folder Options dialog, there are several settings we want to change. Essentially, we want to ensure that we can see and have access to as many different locations in Windows as possible to include seeing Registry files, normally hidden folders, and so on. There are also other configuration options related to starting File Explorer in a separate thread as well.

Since we will be running through an exercise soon for all these settings, we will not show every single option we will be configuring, but in general we want to have a consistent interface to the files which means we will want to view the file listings in details mode, see all file extensions, and see hidden/system files. Once we have the options configured the way we want them, clicking the **Apply to Folders** button will ensure that all the directories we visit using File Explorer will appear the same.

Command Line Basics

Command prompt
- Legacy interface
- Text based
- Still required for some commands



PowerShell
- Preferred over CMD
- Powerful scripting
- Object oriented

There are two primary command line interfaces in Windows: CMD and PowerShell. Both have their uses, but for most things, PowerShell is the preferred command line interface in Windows. In the virtual machine we are using, shortcuts for each of these command line interfaces exist on the Desktop. Both are configured to start the interface as an Administrator. This ensures our commands will work when higher privileges over a regular user are needed.

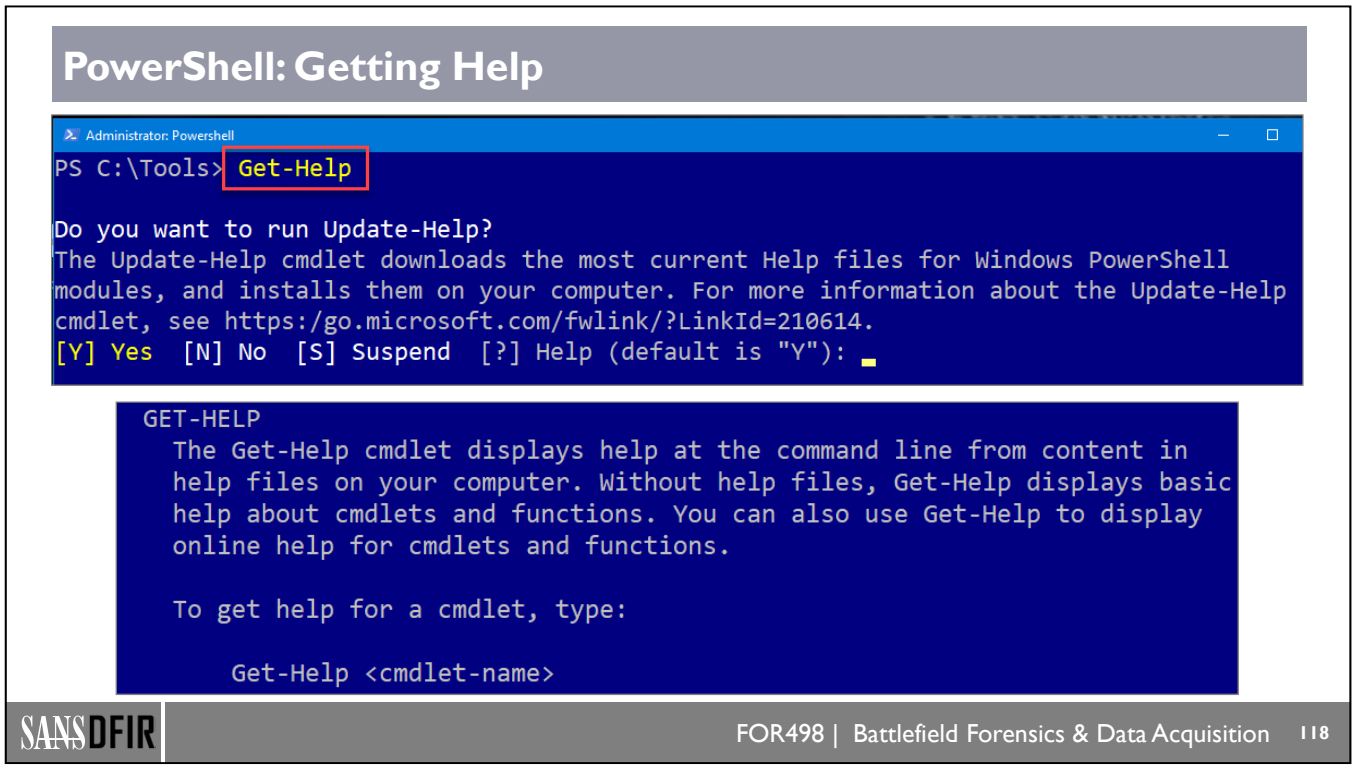
Using both shells to do basic things like running commands, repeating commands, and so on works the same. Both also offer nice features like tab completion of file and directory names, maintaining a history of previously entered commands that can be reviewed or rerun, and so on. Each interface has its caveats to be aware of and we will see some of these when we do exercises. For example, in PowerShell, the $ symbol has special meaning in that it is used to declare and later reference a variable. There are, however, several files related to NTFS that use $ in the file name ($MFT for example). Because we do not want to PowerShell to think $MFT is a variable, we have to enclose the value in single quotes. This "disables" PowerShell treating the $ as the beginning of a variable and things will work as expected. This same type of thing is not required in CMD, but as we will see, PowerShell offers us a lot more in terms of scripting, getting help, and many other things that CMD simply cannot do.

As you use either shell and start to type a command or directory name, pressing the <TAB> key will automatically complete any matching files or directories up to the point where there is more than one option found. For example, if a directory has the following files in it…:

Text 1.txt
Text 12.txt
Text 13.txt

…and you typed **`Tex`** and then pressed **`<TAB>`** in PowerShell, PowerShell would complete the value to **`Text 1`**, since that is the extent of the similarities between the three files. If you continue to press the **`<TAB>`** key, you will scroll through all of the similar names.

To see previous commands, that you have typed, press the **UP** arrow.



Another nice change from CMD to PowerShell is PowerShell's built in (and very extensive) help. To see the help for a command, simply type **`Get-Help <cmdlet>`** where **`<cmdlet>`** is the PowerShell command you want help for. PowerShell commands are called "cmdlets" and are essentially Verb-Noun combinations such as **`Get-ChildItem`** or **`New-Item`**. The first time you use **`Get-Help`**, you may be prompted to update the local help files, as shown above. It is generally a good idea to do this at least once, so just hitting **`Enter`** here will use the default value of **`Y`**, or **`Yes`**, which will update things from Microsoft. Of course, if you are not on the Internet, this will not work. Additionally, it does take a bit of time to finish.

For example, if you run the command: **`Get-Help Get-History`** , the following information would be displayed (partial output shown):

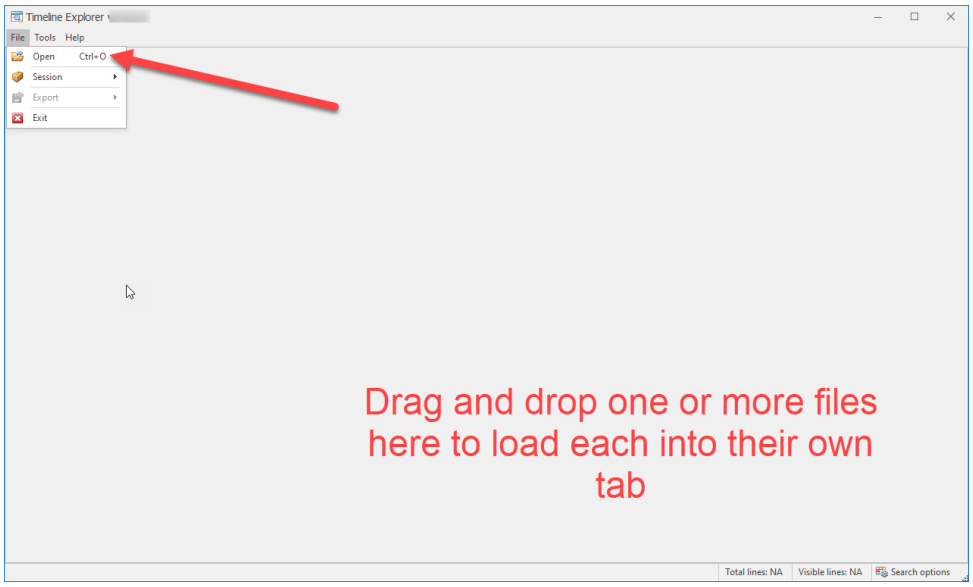
```
NAME
    Get-History

SYNOPSIS
    Gets a list of the commands entered during the current session.

SYNTAX
    Get-History [[-Id] <Int64[]>] [[-Count] <Int32>] [<CommonParameters>]
```

PowerShell provides robust help as well as examples of using available commands, so getting up to speed becomes much easier.

Data Review Techniques: Timeline Explorer (1)

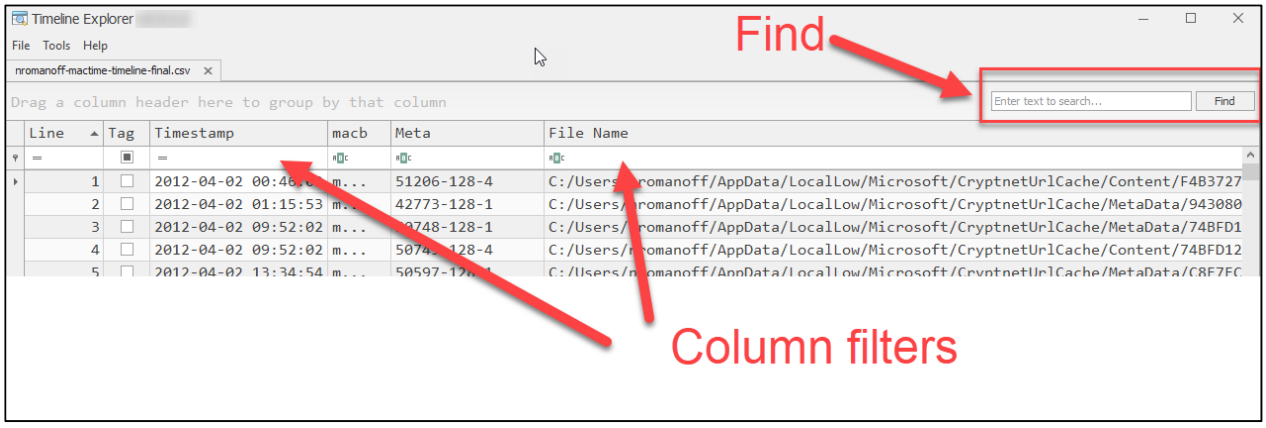


As we locate and process various artifacts, we will of course need a way to review the processed data. For many situations, Timeline Explorer will be our tool of choice for this. Timeline Explorer can load and display just about any Excel or CSV file without needing Excel to be installed. It can open several files at once, with each loaded file being displayed in its own tab. There are powerful searching, sorting, and filtering commands available that make finding data quick and easy. We will be using Timeline Explorer throughout the course, but the following few slides indicate its general use.

To open a file, use the File | Open menu, or press CTRL-O as a shortcut. Alternatively, you can select one or more files in File Explorer and drag and drop them into the main interface.

Note: If running Timeline Explorer as an administrator (which is not necessary), Windows will prevent drag and drop from working.

Data Review Techniques: Timeline Explorer (2)



Once the file is loaded, the data is presented in a read-only fashion. There are options on top of each tab that allow for searching and filtering. Additionally, the top of each column allows for filtering data in a manner that is specific to that column. Clicking on a column header allows for sorting on that column.

Throughout the course, your instructor will continue to demonstrate different aspects of Timeline Explorer that will help you find the most important data efficiently.

Finally, the Help menu contains an option that shows many additional features of Timeline Explorer.

Timeline Explorer can open just about any CSV or Excel document (first workbook only) and display the contents in a grid.

NOTE: All times are assumed to be UTC!!! Even if source file contains a time zone (like Plaso file), it is IGNORED. Always use UTC.

Support for a wide variety of formats is built in, including Autoruns, Mactime generated timelines, Plaso timelines, all of Eric Zimmerman's command line tools (CSV output), and more.

Shortcuts
CTRL-T: Tag or untag selected rows
CTRL-N: Select the next tagged row (loops around when at the last tagged row)
CTRL-D: Bring up Details (for use with super timelines)
CTRL-C: Copy selected cells (and headers) to clipboard

With focus in the grid
CTRL-Left: Select first column in current row
CTRL-Right: Select last column in current row
CTRL-Up: Select first row
CTRL-Down: Select last row

Find
Allows for very specific filter criteria to be entered in a single place. Click the question mark icon in the upper right of the Search options dialog for more details.

Wildcards
Wildcards are supported in column filters when using the **LIKE** operator. **%** matches anything and _ matches a single character. **CONTAINS** implies wildcards and is the default but using **LIKE** along with wildcard operators can often be more precise.

Tagging allows for selecting a subset of data, filtering on it, then exporting to Excel using the File menu. Tagging can also be used to notate interesting results and go back to them as needed.

The Details window is used to review super timelines generated by Plaso. It allows for easier inspection of data in a non-horizontal format. Double clicking a row in a super timeline file will also bring up the Details window. There are options to make the Details window the topmost window, and buttons to move to the next or previous record (allows for not having to click on each row in the main window to change the selected row).

Layouts
For all supported file types, layout files are saved and loaded as needed. The layout files contain settings for that file type, such as which columns are shown, the order of columns, conditional formatting rules, and so on.

Summary

- Microsoft Windows hides a significant amount of directories and files from the view of normal computer users
- An analyst machine must be able to see and access all resident data on the computer
- Command line skills in cmd or PowerShell can greatly enhance analyst effectiveness
- Timeline Explorer is a very useful tool in finding the needle in the haystack

This page intentionally left blank.

Exercise 2.5

Preparing the Analyst Machine

Synopsis: In this exercise, you will take the necessary steps to set up your VM in the recommended manner of a forensic analyst's machine. You will also use a command line utility to familiarize yourself with working at the command line. Finally, you will review .csv files of extracted metadata in Timeline Explorer.

Average Time: 30 Minutes

This page intentionally left blank.

Exercise 2.5 Takeaway

- Having a properly configured environment allows for consistency and being able to see all available files, especially those that are of forensic interest (and are hidden by default).
- Many forensic tools are command line driven, so being comfortable with both Windows and Linux command line interfaces is necessary.
- Many computer forensic tools output data into CSV format. By leveraging the capabilities of Timeline Explorer, interacting with this data becomes much easier to do.

This page intentionally left blank.