498.5

Apple Acquisition, Internet of Things, and Online Attribution

SANS

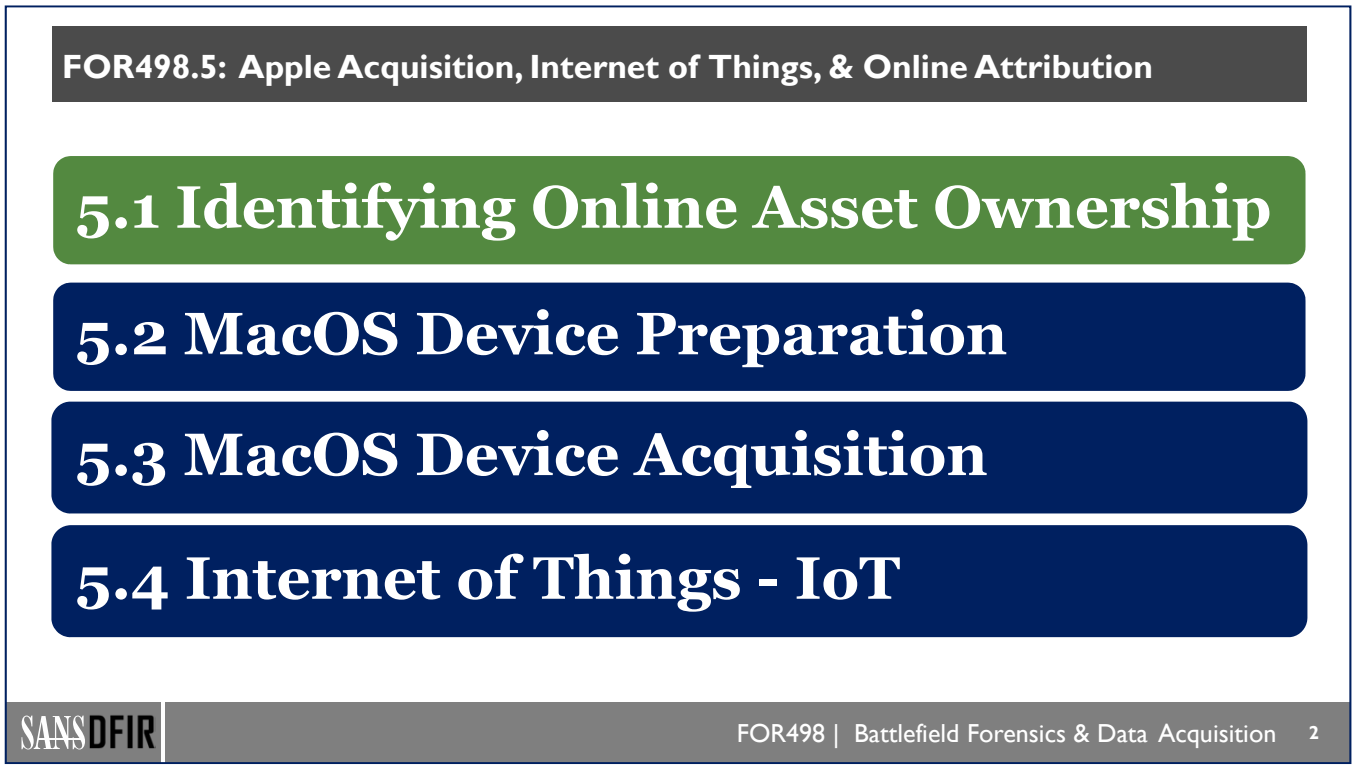
THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

FOR498.5 | Battlefield Forensics & Data Acquisition

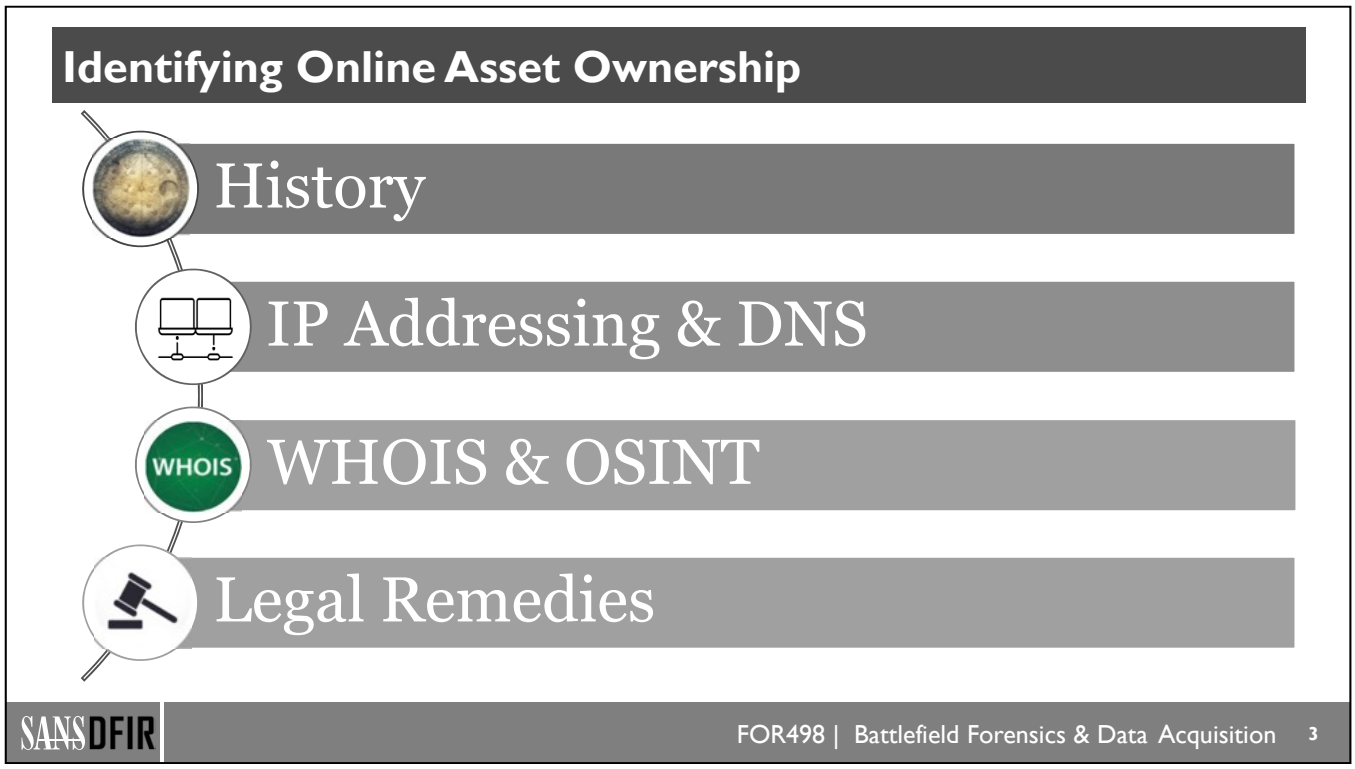
Apple Acquisition, Internet of Things, and Online Attribution

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

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



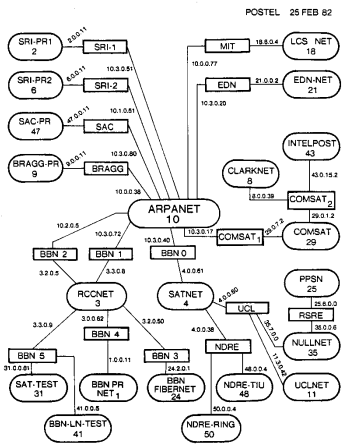
This page intentionally left blank.



This page intentionally left blank.

History

Jonathan Bruce Postel, aka the "god of the Internet"



Jon's map of the Internet from 1982

No "history of the Internet" story would be complete without first introducing you to Jonathan Bruce Postel.[1] Jon was widely regarded as the god of the Internet, and the self-proclaimed "czar of socket numbers". This was no exaggeration. Jon was the editor of the Request for Comment (RFC) series for many years. RFCs are essentially the rules for the Internet, and how its communication works. For example, RFC 1591 is the documentation for "Domain Name System Structure and Delegation" [2], and RFC 791 is the documentation for "Internet Protocol". [3] In all, Jon wrote or co-authored over 200 RFCs.

Email would not exist had it not been for Jon. In 1980, Postel came up with the idea of Mail Transfer Protocol, and then in 1981 he formalized Simple Mail Transfer Protocol (SMTP). [4]

As if this was not enough for any one person, Postel was the Internet Assigned Numbers Authority (IANA) from the very beginning. IANA was the very genesis of what we know today as IP addressing and domain names. The "Authority" portion of the title was not meant as a title. It was originally meant that Postel was the Authority. He held this position until shortly before his death in 1998.

A very enlightening RFC 2468 was written specifically for Jon, and is entitled "I Remember IANA" [5]. This RFC was written by his close friend, and 'father of the Internet', Vint Cerf [6]; a legend in his own right.

[1] Who was Jon Postel | `https://for498.com/8k5b4`

[2] RFC 1591 | `https://for498.com/rfc1591`

[3] RFC 791 | `https://for498.com/rfc791`

[4] Simple Mail Transfer Protocol | `https://for498.com/6-cl7`

[5] RFC 2468 | `https://for498.com/rfc2468`

[6] Who is Vint Cerf | `https://for498.com/kwrfs`

Key Dates in History

1972-1991	InterNIC run by Stanford Research Institute
1979	Network Solutions created
1983	Domain Name System (DNS) created
1985	First domain name registered. Symbolics.com
1988	Internet Assigned Numbers Authority (IANA) created
1991-1998	Network Solutions manages DNS and sales of domain names under contract from DISA
1998+	IANA becomes division of Internet Corporation of Assigned Names & Numbers (ICANN) IANA takes over from Network Solutions, IANA tasked w/ managing Regional Internet Registries

It has been quite a long road in a very short period of time, as concerns who had the authority over Internet Numbers, and later, names.

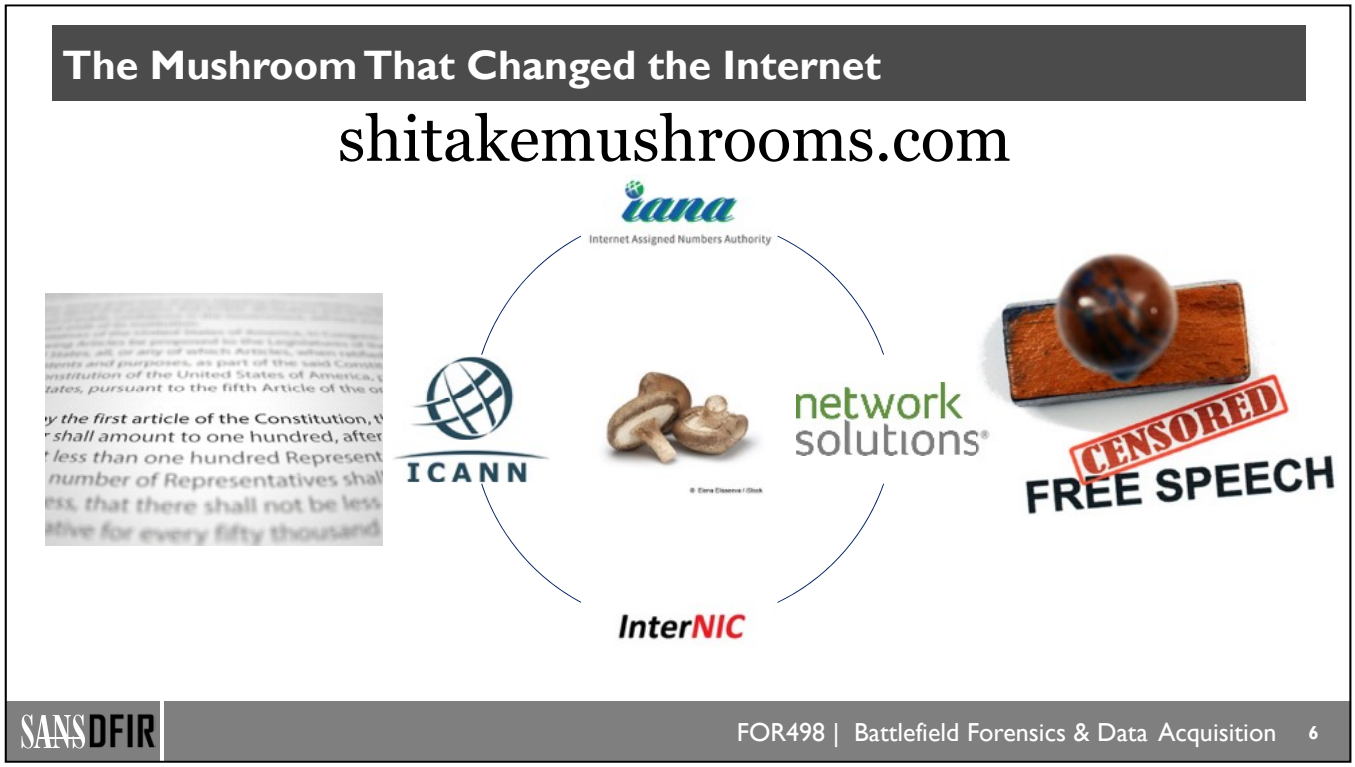
When the Internet was in its infancy, computer host names and their corresponding IP addresses were maintained on a hosts.txt file that resided at the Stanford Research Institute. [1] As the network got larger, it became very difficult to maintain a centralized hostname registry, so in 1983, the Domain Name System (DNS) was created. As the Internet evolved, the Internet Assigned Numbers Authority (IANA) was officially created and placed under the control of Jon Postel in 1988. At that time, it governed all things to do with Internet names and numbers. In 1991, Network Solutions [2] was granted authority to manage the DNS, as well as serve as the first commercial vendor of domain names. This was not the beginning of issuance of domain names. This had already started in 1985. On March 15, 1985, a computer systems company named Symbolics Inc. registered the domain name of symbolics.com.

In 1998, partly due to issues with Network Solutions and partly to remove US government control, the Internet Corporation of Assigned Names & Numbers (ICANN) [3] was created. It is the umbrella under which IANA now operates. All Internet names and numbers administration, as well as ports and protocols.

This new, centralized organization was/is tasked with the following:
1. Responsibility for managing parameters
2. Making sure that everyone uses the same protocols and parameters
3. Coordinating the assignment of identifiers
4. Ensuring that the creation and allocation of addresses and domain names is done accurately

IIn short, IANA manages Top Level Domains as well as dealing with the assignment of IP addresses, ranges, ports, etc., and ICANN runs IANA.

[1] First Hosts.txt file | `https://for498.com/heq65`
[2] Network Solutions DNS control | `https://for498.com/07aqu`
[3] ICANN | `https://for498.com/6e-hg`



During the early days of domain name registration, if you wanted to purchase a domain name, your only option was to purchase it from Network Solutions, Inc (NSI). NSI was a company long before it was awarded this role in 1991, but its owners were highly critical of some of the names they saw people trying to register. In 1996 NSI started automatic name filtering. [1] They would block the registration of any pornography related names, as well as any profane or distasteful names.

In 1998, Jeff Gold attempted to register the domain name of shitakemushrooms.com, but he was unable to do so. NSI attempted to argue that it was their First Amendment right to censor terms they found offensive. They lost this argument because they were operating under license from a government agency, and not autonomously. Incidentally, the website shit.com was already registered at this time, although it had been registered prior to NSI being awarded sole control of domain name assignments.

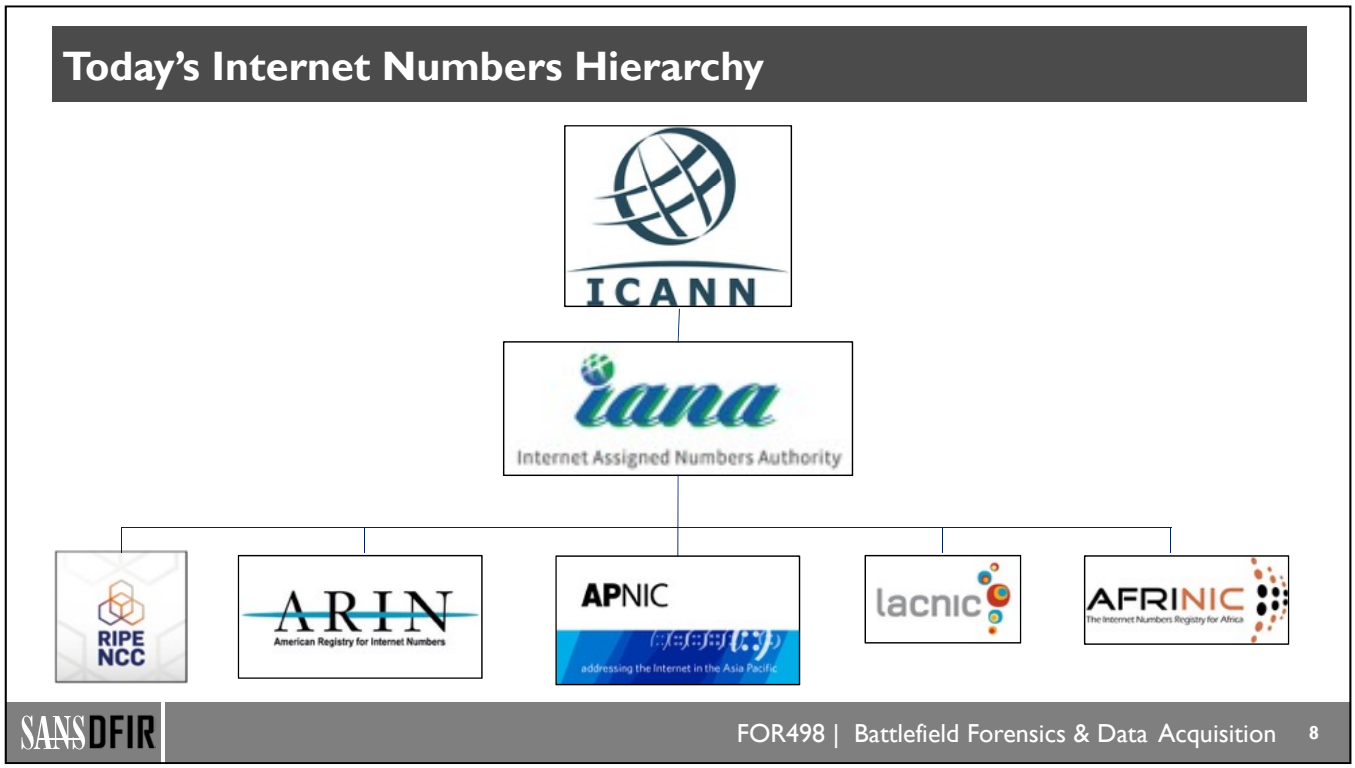
Shortly after this, the Internet Corporation of Assigned Names & Numbers (ICANN) was formed by the United States Department of Commerce to take over control of these duties and others. It is now an international organization, independent of any government.

According to the Memorandum of Understanding [2] between the two parties, it was agreed that ICANN would carry out the following duties:

1. Establishment of policy for and direction of the allocation of IP number blocks;
2. Oversight of the operation of the authoritative root server system;
3. Oversight of the policy for determining the circumstances under which new top-level domains would be added to the root system;
4. Coordination of the assignment of other Internet technical parameters as needed to maintain universal connectivity on the Internet; and
5. Other activities necessary to coordinate the specified DNS management functions, as agreed by the Parties.

ICANN assumed responsibility for IANA, and tasked it with performing much of the duties. In 2014, the contract over IANA expired, and in 2016 it was transferred to the Post Transition IANA. PTI now oversees the duties of IANA.

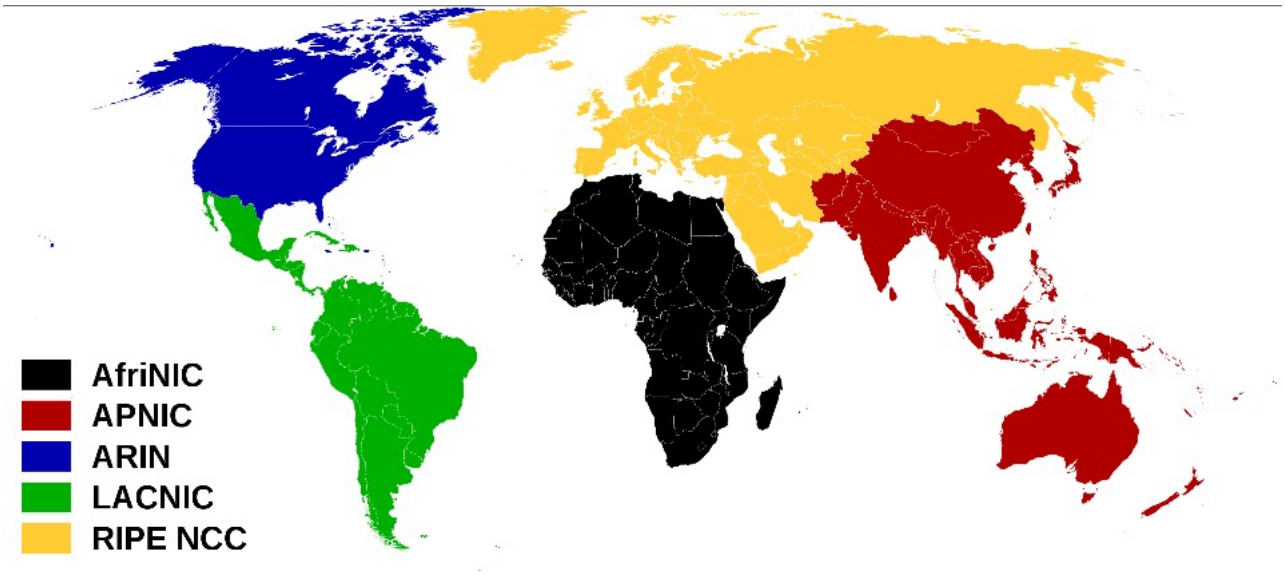
[1] NSI Name Filtering | `https://for498.com/07aqu`
[2] ICANN MoU | `https://for498.com/kf86y`



There are quite a few convoluted stories, not to mention happenings that have brought the control of Internet Assigned Names and Numbers to where it is today. Suffice to say that Regional Internet Registries are governed by IANA, now administered by Public Technical Identifiers (PTI) [1], which is a department of ICANN.

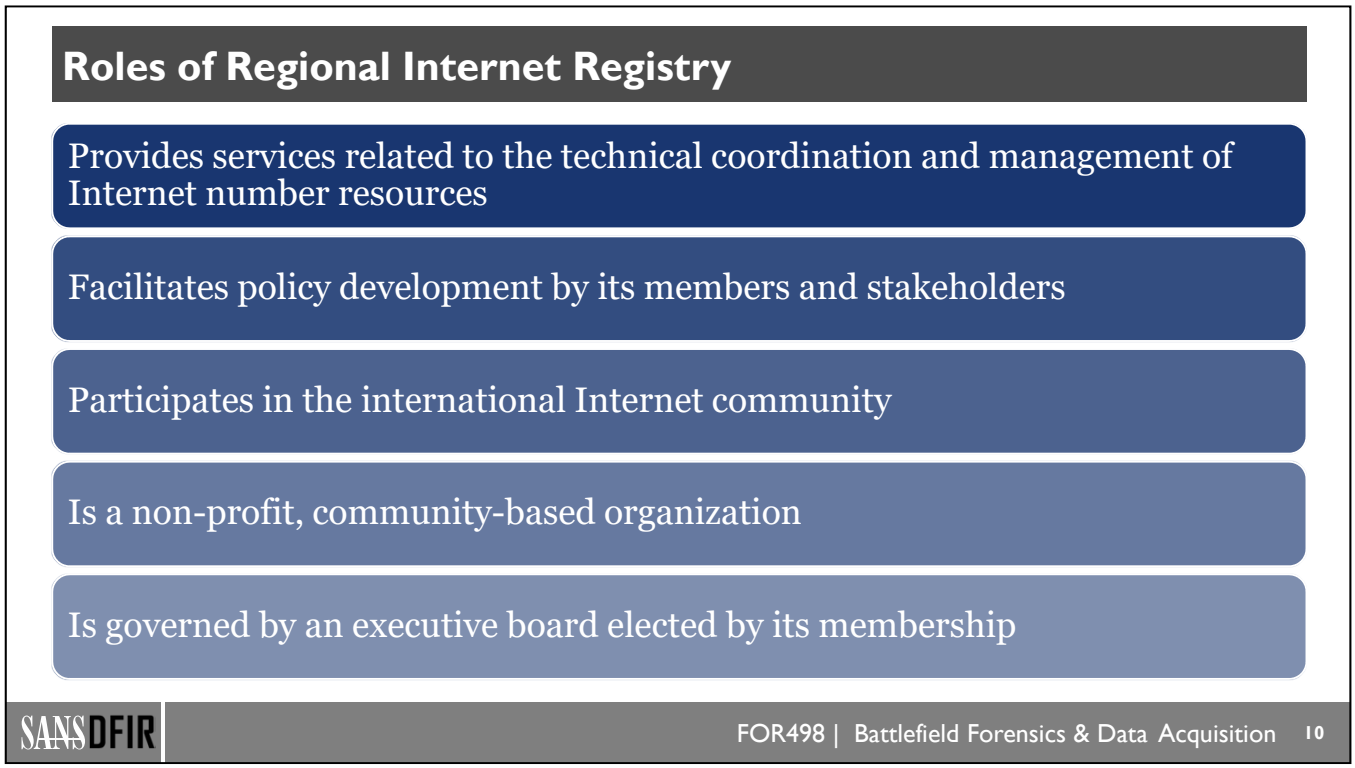
[1] Public Technical Identifiers | `https://for498.com/h0vek`

Regional Internet Registries



AfriNIC – African Network Information Center
APNIC – Asia-Pacific Network Information Center
ARIN – American Registry of Internet Numbers
LACNIC – Latin America & Caribbean Network Information Center
RIPE NCC - Réseaux IP Européens Network Coordination Center – (European IP Networks)

Regional Internet Registries | `https://for498.com/idnjk`



A Regional Internet Registry (RIR) is an organization that manages the allocation and registration of things like IP addresses and Autonomous System numbers within a particular region of the world. Their roles are as outlined above.

Each of the previously discussed RIRs use these guidelines for operations. Each of these RIRs is responsible for operations on their defined territory, but each coordinates with each other via the Number Resource Organization (NRO). [1]

[1] Number Resource Organization | `https://for498.com/9k03m`

IP Addresses: IPv4

IPV4

- 32-bit length address divided into four sections called octets
- Each octet is one byte in size
- Can range from 0.0.0.0 – 255.255.255.255
- Total of 4,294,967,296 IP addresses
- Represented in decimal form

123.52.65.45

The Internet Protocol (IP) Address [1] is a unique address issued to a router by the Internet Service Provider (ISP) for the period that the device is connected to the Internet. They are also assigned to domain names. No two devices or domains can have the same address at the same time. An example is 123.124.125.126. This IP address is as unique as a home address. No two houses can have the same address.

When a device attempts to connect to the Internet, it first has to communicate with the ISP. When a subscriber (in this example Comcast customer) attempts to go online, their device will communicate with a server at Comcast. The device will ask for an IP address so that it can access the Internet. Comcast verifies the device's right to have the access, and will then issue the IP address, thereby connecting the device to the Internet. It is possible to get the same IP address on different occasions, but not very likely. It is also possible to hold an IP address for very long periods of time.

An IP address is attached to every email that is sent from a device. When an email is sent, it passes through a minimum of two, and more typically, at least four, servers. Each of these servers tags the email with its IP address, allowing an examiner to trace the exact path of the email. In some cases, the examiner can then employ Open Source Intelligence (OSINT) gathering techniques to more clearly identify the sender. When this profiling is ineffective in establishing more conclusive proof, a subpoena or warrant can be issued to the ISP, compelling them to provide subscriber information for the device connected to the IP address at the specified date and time.

The IP version 4 address is comprised of four groups of numbers or octets, and each octet number can be a number from 0-255. IPv4 is represented in decimal form. An example is 123.123.123.123.

This constitutes a total of 4,294,967,296 IP addresses, but believe it or not, we ran out of IP addresses in 1996! IANA had to come up with a solution, and the assignment of Private Address Space [2] was born. Prior to then, every computer connected to the Internet had a public IP address (because every connected computer

needs an IP address). If your company had 10,000 endpoints, you needed 10,000 public IP addresses. IANA decreed that there would be three ranges of IP addresses set aside that could never be used on the public Internet, and could only be used internally. This meant that IP addresses could be re-used as long as they were not within the same sub-network. Companies that had been leasing millions of addresses immediately gave back most of them, and virtually overnight, 35-40% of public IP addresses were returned to IANA.

The private address spaces are defined as follows:

10.0.0.0 - 10.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 - 192.168.255.255

[1] IP Address | `https://for498.com/iab3d`
[2] Private Address Space | `https://for498.com/rfc1918`

Notice From RIPE NCC

- On November 25, 2019, RIPE NCC made the following announcement:

"Today, at 15:35 (UTC+1) on 25 November 2019, we made our final /22 IPv4 allocation from the last remaining addresses in our available pool. We have now run out of IPv4 addresses."

On November 25, 2019, RIPE NCC[1] made the following announcement:

"Today, at 15:35 (UTC+1) on 25 November 2019, we made our final /22 IPv4 allocation from the last remaining addresses in our available pool. We have now run out of IPv4 addresses."

[1] RIPE NCC Announcement | https://for498.com/3tq5z

IP Addresses: IPv6

IPV6 128-bit length address divided into eight sections called hextets

Each hextet is two bytes in size

Can range from 0000:0000:0000:0000:0000:0000:0000:0000 – FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

Total of 340,282,366,920,938,463,463,374,607,431,768,211,456 IP addresses

Represented in hexadecimal form

2015:0F44:19FA:0000:0000:F4F4:1683:ABCD

The IP version 6 address [1] is comprised of eight groups of numbers or hextets, and each hextet number can be a hexadecimal number from 0000-FFFF. Represented in decimal, this would be 0000 – 64435, although IPv6 is referenced in hexadecimal, and not decimal. An example is:

2015:0F44:19FA:0000:0000:F4F4:1683:ABCD

This constitutes a total of just over 340 undecillion IP addresses, or about seven IP addresses per atom in all the bodies of all the people in all the world. In technical terms, that is a lot!

[1] IPv6 Address | `https://for498.com/v5g4h`

Sample IPv4 Address Distribution

IPV4	IANA	0.0.0.0 – 255.255.255.255
	ARIN	63.0.0.0 – 76.255.255.255
	Lessee (Comcast, GoDaddy, Ford)	69.136.0.0 – 69.143.255.255
	Subscriber (Home, business, website)	69.138.18.123

Internet Assigned Numbers Authority (IANA) controls all IP addresses in use. The entirety of all IP version 4 addresses range from 0.0.0.0 to 255.255.255.255. This equals 4,294,967,296 possible IP addresses. This sounds like a lot, but consider that every website (for the most part) gets its own IP address, every Internet subscriber gets an IP address, and approximately 600,000,000 IP addresses are reserved and not available for use, and it becomes easy to see that almost 4.3 billion addresses is not so many at all. [1] For example, the United States alone is assigned 1,585,642,333 of these addresses [2], or almost 43% of all available IP addresses! By comparison, China has 344,286,611 IP addresses, and Russia has 45,418,774 addresses. North Korea has 1028 addresses!

ARIN – IANA issues blocks of IP addresses to the 5 Regional Internet Registries around the world, including the RIR of ARIN. ARIN is the RIR that administers IP addresses for North America. The IP addresses under its control are not contiguous, but an example of a large span of IP addresses that are controlled by ARIN is the range between 63.0.0.0 and 76.255.255.255. No other RIR can use these IP addresses.

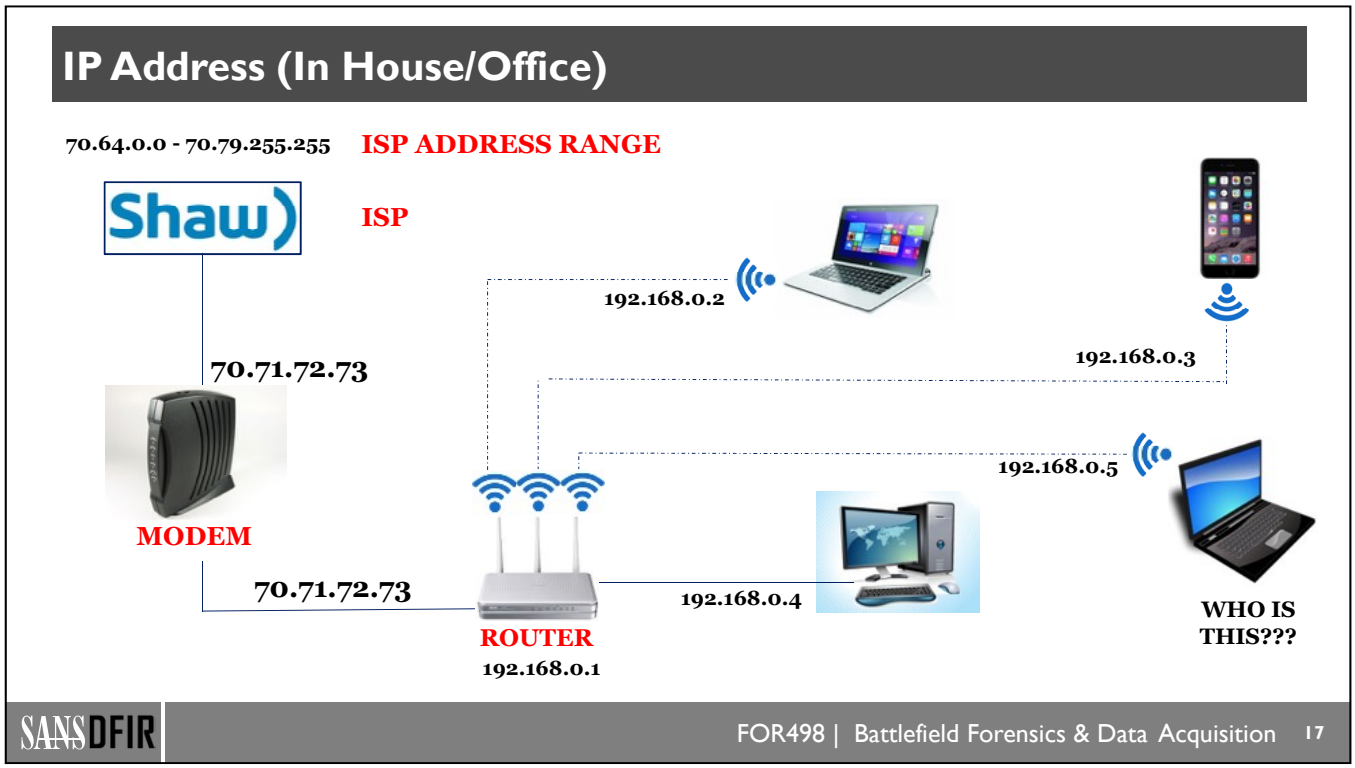
Lessee – A company will lease IP addresses from ARIN. An individual will not typically do this (although it is possible). One example is a lessee that wants to set up an Internet Service Provider (ISP) company. Comcast would be an example. They lease a range of IP addresses from ARIN. One of their IP address ranges spans from 69.136.0.0 to 69.143.255.255. This is almost half a million IP addresses that they would then lease out to their subscribers. Nobody but Comcast can use the IP addresses in this range. In another example, a company like GoDaddy wants to set up a domain name/hosting company. They will lease a range of IP addresses from ARIN, and then use them to assign to domain hosting (where a website sits). For example, the website firstcry.com is a website hosted on GoDaddy, and has an IP address issued to it by GoDaddy. Nobody but GoDaddy can use IP addresses within their range.

The cost for leasing IP addresses from ARIN is not expensive, relatively speaking. A /24 (256 IP addresses) costs $250.00 USD per year. On the other end of the spectrum, a /8 (16,777,216 IP addresses) costs $64,000.00 USD per year.

Subscriber – The subscriber is an end user. For example your residence or office. You as a subscriber would pay Comcast (for example) a fee per month to access the Internet. In return, they issue you a single IP address from their range of IP addresses. In the example above, it is 69.138.18.123. While you have/use this IP address, nobody else can use it. This makes you very unique, as the ISP (in this case Comcast) keeps logs of who it issues IP addresses to, and the dates and times that they had exclusive use of a given IP address. Since your IP address is attached to everything you do, anyone getting your IP address can perform a WHOIS function on it to determine who the lessee is, and with the proper authority they can go to the lessee and get your information.

[1] Global IP range assignments and reservations | `https://for498.com/einsx`

[2] IP Address assignments by country | `https://for498.com/8jyce`



In this example, we have used the public IP address of 70.71.72.73. this IP address is one of thousands owned/leased/assigned to Shaw Communications in Calgary, Canada. Shaw then issues it (or anyone within its IP range) to its customers upon request.

When you attempt to access the internet, whether to visit a website or to check email, etc., your equipment (specifically the modem) needs to first authenticate you with the Internet Service Provider (ISP) Once the ISP authorizes you, it issues your modem an IP address from its ISP Address Range that it was assigned by the Regional Internet Registry. The vast majority of customers have a Dynamic IP address, and so the assignment of this IP address is entirely random and can potentially change every time the modem tries to authenticate with the ISP.

Once the modem has received an IP address from the ISP, it passes it on to the router. (In many cases today when you order Internet for your home, the ISP provides a router that has the modem built into it.) This IP address is called the Public IP address. Since no two devices can have the same IP address at the same time on the same network, the IP address given to the router by the ISP can only be used by that router (and nobody else's) for the period that it holds the lease of that IP address from the ISP. This can be minutes, hours, days, or months. It is important to note that this IP address has NOT been issued the devices being used inside the home/office. This IP address has been issued to the modem/router ONLY.

The router then, through the use of Network Address Translation (NAT), converts the public IP address to a Gateway address. In our example, the public IP address of 70.71.72.73 has been converted to a Gateway Address of 192.168.0.1. This IP address is part of a range of IP addresses specifically reserved for private network use and cannot be used on the public side of the router. This Gateway Address is configurable by the user. The router than uses its internal routing table to assign Private IP addresses to all of the devices connected to it internally, with no care as to whether the devices are connected wirelessly, or via hard wire.

Generally speaking, these private IP addresses are invisible to the outside world. In other words, when the laptop inside the home network that carries the IP address of 192.168.0.2 goes out on the Internet to Google, to Amazon, or to any site, the site does not see the IP address of 192.68.0.2. It sees the IP address of 70.71.72.73. In fact, if all four internally attached devices in the example all visited google.com at different times of the day, Google would not know they were four different devices simply based on IP address, because they would all be carrying the same public IP address.

In the example, a computer bearing IP address 192.168.0.5 is connected to the router. Notwithstanding any security that may be on the router, any wireless device can connect to it, as long as the device is within wireless range. For all we know, this device could be in use by a stranger in a car near the building, or in the apartment next door. If they can authenticate their device on your router, the router will allow it, and now when they go out on the Internet, they look to the entire world like they are 70.71.72.73. This is an incredibly important fact that must be considered when requesting warrants or subpoenas.

In certain circumstances, Google (and others like it, Apple, Facebook, Amazon etc.), can, based on user agent string data, discern and discriminate between different types of devices coming from the same public IP address. In other words they can provide (if the request is worded properly) the device name including MAC address that is/was using the IP address of interest on the date and time (relative to the time zone) specified in the legal order.

Domain Names

Top level domain

- example.com

Sub domain

- an.example.com

Second level domain

- example.co.uk

A Domain Name[1] is a string of characters that identify a location on the Internet. A domain name resides inside a domain, and there are a large amount of domains on the Internet. An example of a domain would be .com. An example of a domain name would be google.com.

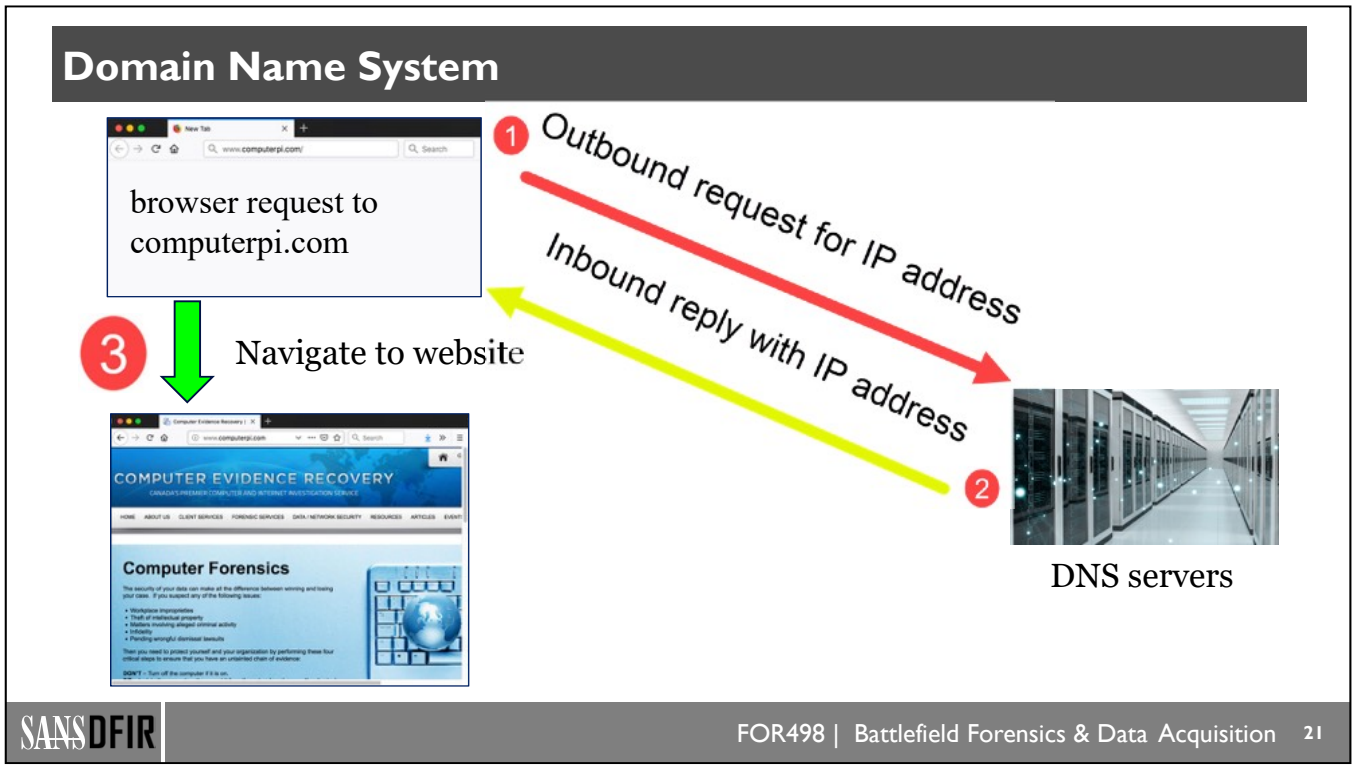
A domain name is merely a 'human friendly' representation of an IP address.

[1] Domain Name | `https://for498.com/gfdjo`

Domain Names: Some Rules

- No more than 63 characters long
- Everything after first slash is case sensitive
- Not sure which is a slash and a backslash?

A domain name field (the characters before the first slash) cannot exceed 63 characters, and everything after the first slash in a URL is case sensitive.



As previously indicated, everything on the Internet must have an IP address in order to communicate. Having said that, humans have a difficult time remembering numbers of any kind, let alone really long and complex ones. Humans have a much better time remembering words and names. This is why we give our computers names. This is also why websites have names. It is much easier to remember google.com than 74.125.127.99. It is much easier to remember ns1.mail.com than it is to remember 67.22.34.12.

The problem with this is that computers don't know anything about names. They only know numbers. Think back to a day when telephones were devices that only made and received actual telephone calls. They were connected to the wall via wire, and without that wire, there was no communication! Because it was difficult to remember numbers, a book existed called a telephone book. It kept an alphabetical list of people by name, and then showed the telephone number beside the name. In fact there was a telephone book for people, and another telephone book for businesses.

This model was adopted for the Internet. How great would it be if we had a book that listed people and/or businesses by their name, and then had the IP address beside it? How big would this book be, since the Internet knows no geography? It would be too massive to print and have on your desk. So a system called the Domain Name System (DNS) [1] was created.

DNS is widely referred to as the phonebook of the Internet. It is an electronic lookup table that keeps track of the names that go with specific IP addresses, and vice versa. For example, the author's website is computerpi.com, but a computer has no idea what that means. A computer DOES know what 108.160.156.126 means. DNS essentially keeps track of the fact that computerpi.com = 108.160.156.126.

This translation, or resolution starts at the lowest level, on a user's computer. For example Joe wants to go to the website amazon.com. When Joe enters the name into the browser bar, the computer looks first at the hosts file resident on the computer. Every computer has one, and it will have at least one entry, but possibly more.

If the browser does not find amazon.com as an entry with a corresponding IP address, it reaches out to a resolver further upstream, for example from the ISP. Once a resolver has been found that knows the answer, it provides this back to Joe's computer, which can then ask for amazon.com by using its IP address.

Without these DNS resolvers sitting on the Internet providing translation services, we would not be able to communicate as easily as we do. When a new entity such as a website is introduced to the Internet, Its name and IP address get populated into the DNS resolvers, usually in minutes.

[1] Domain Name System (DNS) | `https://for498.com/chkd4`

WHOIS

- Mechanism in place to determine ownership of a website
- WHOIS, along with reverse lookup are instrumental in investigations
- DomainTools & ICANN both excellent places to run WHOIS

```
Registrant Contact Information:
        Name                    Consulting Investigation Services
        Organization            Consulting Investigation Services
        Address                 PO BOX 2097
        City                    WAXAHACHIE
        State / Province        TX
        Postal Code             75168-2097
        Country                 US
        Phone                   +1.9729373938
        Fax                     +1.9999999999
        Email                   info@cispi.net

Administrative Contact Information:
        Name                    Consulting Investigation Services
        Organization            Consulting Investigation Services
        Address                 PO BOX 2097
        City                    WAXAHACHIE
        State / Province        TX
        Postal Code             75168-2097
        Country                 US
        Phone                   +1.9729373938
        Fax                     +1.9999999999
        Email                   info@cispi.net

Technical Contact Information:
        Name                    Consulting Investigation Services
        Organization            Consulting Investigation Services
        Address                 PO BOX 2097
        City                    WAXAHACHIE
        State / Province        TX
        Postal Code             75168-2097
        Country                 US
        Phone                   +1.9729373938
        Fax                     +1.9999999999
        Email                   info@cispi.net
```


WHOIS is both a noun and a verb. The noun iteration is the name of a function that queries a database for information. (Please query the WHOIS database).The database query itself is the verb version of WHOIS. (Please run a WHOIS on that domain name).

When an entity registers a domain name, there is certain information that is requested by the Domain Name Registrar. Registrant name, address, telephone number, and email address for each of the Registrant, Administrative, and Technical contacts. This information is then made available publicly to anyone who knows how to conduct a search. A sample of this data for the domain name “cispi.net” is shown above.

There is currently no mandatory verification of this information, and the registrant can enter any information they choose. The only information that is truly verified is the email address at the time of registration. This is notable in that this information is commonly queried and used as a method to contact persons believed to be associated with a website. In many instances today, registrants are using privacy blocks to hide information from people conducting WHOIS searches.

The first WHOIS directory was created in the early 1970s and was managed by the Defense Advanced Research Projects Agency (DARPA). This continued up until 1993 when the management was handed over to Network Solutions. Up until 1998, if someone wanted to register a domain or conduct a WHOIS query, they were doing it through Network Solutions. Upon ICANN being formed and taking over, the registration process was decentralized, and many new domain name registrars were formed, like GoDaddy, eNom, and Tucows. Today there are hundreds of companies in this market space, all with different database rules and WHOIS functionality.

```
Registrant Contact Information:
        Name                            Consulting Investigation Services
        Organization                    Consulting Investigation Services
        Address                         PO BOX 2097
        City                            WAXAHACHIE
        State / Province                TX
        Postal Code                     75168-2097
        Country                         US
        Phone                           +1.9729373938
        Fax                             +1.9999999999
        Email                           info@cispi.net

Administrative Contact Information:
        Name                            Consulting Investigation Services
        Organization                    Consulting Investigation Services
        Address                         PO BOX 2097
        City                            WAXAHACHIE
        State / Province                TX
        Postal Code                     75168-2097
        Country                         US
        Phone                           +1.9729373938
        Fax                             +1.9999999999
        Email                           info@cispi.net

Technical Contact Information:
        Name                            Consulting Investigation Services
        Organization                    Consulting Investigation Services
        Address                         PO BOX 2097
        City                            WAXAHACHIE
        State / Province                TX
        Postal Code                     75168-2097
        Country                         US
        Phone                           +1.9729373938
        Fax                             +1.9999999999
        Email                           info@cispi.net
```

When Network Solutions was the only registrar, the entire WHOIS database was public, and completely searchable (if you knew how) by any parameter in the WHOIS entry. Using special programs like Sam Spade, a person could query Network Solutions for all domain names with a given telephone number in the registration information, for example. It was a very robust investigative tool in its day. When other registrants entered the market, their WHOIS databases were also searchable in this way.

In the very early 2000s, people within the advertising and data collection worlds created scripts that would do wholesale harvesting of all data in the WHOIS servers. This caused a number of issues within the Domain Registry community, not the least of which was availability to humans. The harvesters were so aggressive that the average person experienced significant delays in getting their information. Because the data was deemed to be public information though, the databases couldn't just be shut down. In an attempt to curtail this activity, as well as assist the Domain Name Registers to create more robust systems, it was decided to start charging for the service of being able to query these databases in any other way than by domain name. As a result, a registrar like GoDaddy could charge up to $10,000.00 USD per year (but didn't have to) for access to their database. With the significant proliferation of registrars, the cost became prohibitive to most.

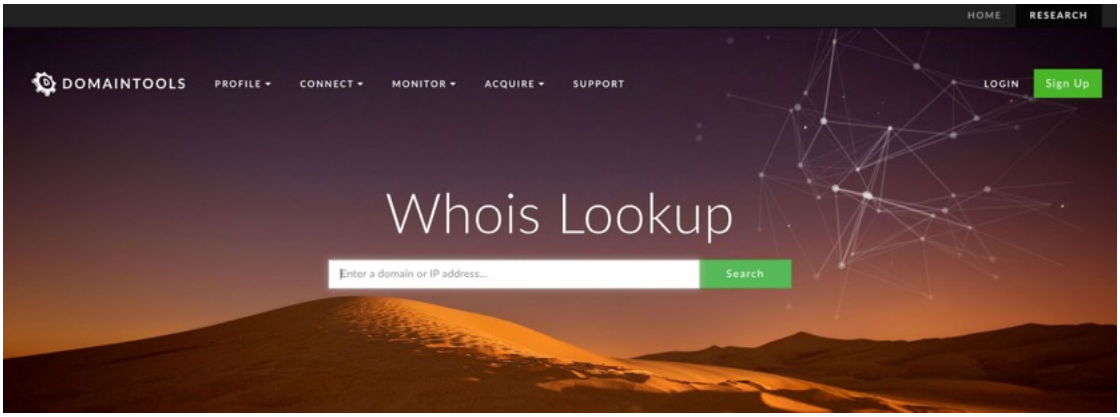
In the mid to late 2000s, domain registrars created a service whereby a registrant could keep their information private, and it would not be released during a WHOIS. Depending on the registrar, a subpoena or warrant is necessary to get to the information. Oddly, there are a number of different rules applied for different domains when it comes to privacy. For example, any website in the .US domain cannot make their information private.

ICANN has an excellent WHOIS search function at whois.icann.org, however it only does WHOIS lookups, in other words, you can only search for registration data on a domain name.

DomainTools also has a very robust WHOIS capability at **whois.domaintools.com**, but it also does much more.

DomainTools

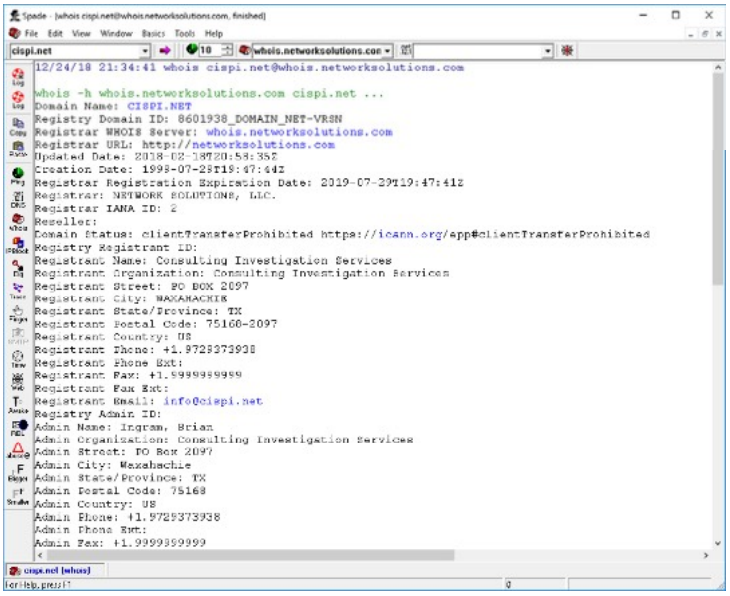
- Domaintools.com
- Most comprehensive historical reverse and WHOIS databases available



DomainTools is the last word in WHOIS investigations. Not only does it do the traditional WHOIS lookup, but it will automatically perform a Referral WHOIS, if it detects that secondary level of information. Most entities have just the one set of WHOIS data, however some larger companies also maintain a secondary database. Comcast and Cox Cable (to name two) maintain an rWHOIS database that allows for pinpointing a particular city, rather than just the national registration address of the company.

DomainTools will also reverse an IP address to its registrant, but its real value lies in its aggregation and historical address capabilities. With the paid subscription, a customer can order things like a historical WHOIS (everyone that EVER was a registrant on a domain), IP address tenants (list all websites hosted at a particular IP address), and many of the functions that used to be available in the early days of Network Solutions. It truly is a tool that should be in every examiner's toolbox.

Sam Spade



Sam Spade is a very "old school" open source tool that was created by Steven Atkins in 1997. It has never been updated. Having said that, it is still in use today, still very effective, provides information not found anywhere else, and runs on Windows 10. World renowned email tracing expert Brian Ingram built on the program, and still does improvements to this day, adding WHOIS servers to the program as they become available.

Sam Spade has an incredibly steep learning curve in order to get it to do anything out of the ordinary, and with domain registrars changing backend access seemingly at will, the Sam Spade abilities change with them.

Some of the extensive abilities of Sam Spade include:

WHOIS on a domain name

rDNS on an IP address

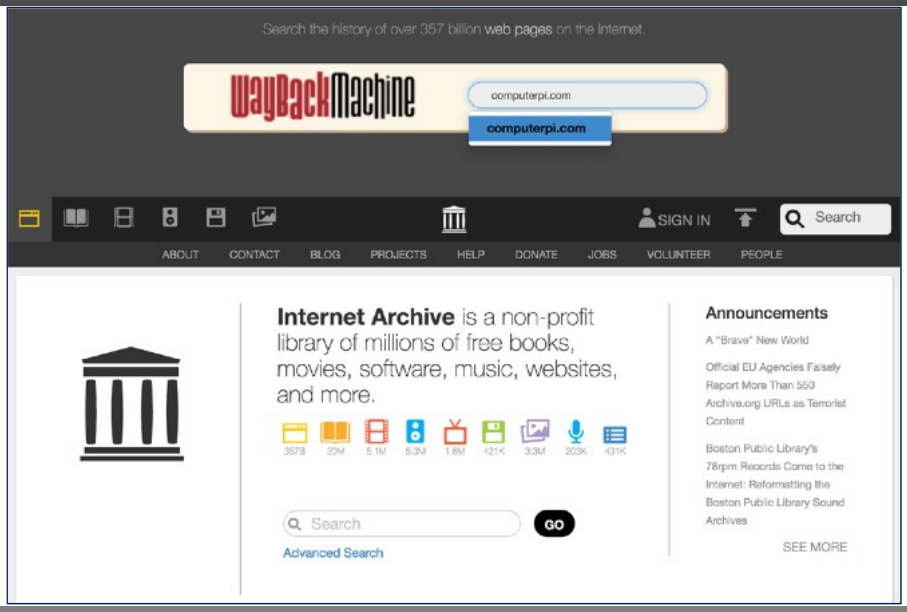
SMTP server probe to see if a particular mail server will accept/send email anonymously

'Dig' function, to poll standard records of a query (AName, MX records, Server of Authority, etc.)

Finger command displays information about users currently logged into a host

-a switch before a name will list all people within a particular WHOIS server with that name. (not available on most servers)

Wayback Machine



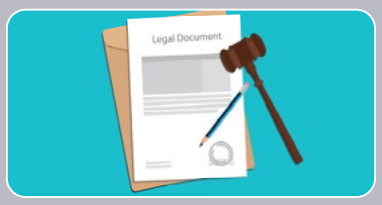
In 1996, something called the Internet Archive was started. The idea behind this was to create an archive or preservation of the World Wide Web for historical purpose. The Internet Archive created something called the WayBack Machine to allow anyone to access all of the information that was being archived through this project.

Internet Archive has created digital versions of webpages books, audios, videos, software and in some cases, social media. For example, many Twitter posts are preserved in the Internet Archive, as are Blogspot articles, the blogging medium provided by Google.

Never forget or underestimate the amount of data that you may find on the WayBack Machine. One of the best uses for this platform is to verify the content of a website today against a copy from a previous time. In fact even if the website does not exist anymore today, it is possible to extract a full copy of the website from a previous moment in time when it was active. Consider the amount of documents that may have been available on a website that is no longer available on the Internet. In many cases, they are available at the WayBack Machine.

[1] WayBack Machine | `https://archive.org`

When All Else Fails



Preservation Letters

- Stop someone from destroying potential evidence
- Best to serve as early as possible in a case (when possible)

Production Orders

- Legally compel someone to turn potential evidence over that is in their possession

Subpoena

- Legally compel a company to turn over data they hold that can assist in an investigation, such as email information from Gmail

Although this is not a legal course, there are certain aspects of the legal system that an examiner should be aware of, both in the civil law realm as well as the criminal realm. This is not to be taken as legal advice, and you must check the laws in your jurisdiction. However, this information is generally relevant in most judicial systems around the world.

Depending on the investigation and how it unfolds, you may have several situations you must react to. If you do not respond appropriately, you may miss evidence, or allow it to be destroyed.

What if you know where digital evidence might exist, but it is not someplace you control?
What if you know where evidence might be, but if you ask for it, it might get destroyed first?
What if you know an entity has information you need, but for privacy reasons, won't give it to you?
What if you are conducting an investigation and come across potentially illegal, unrelated activity?

Let's explore each of these using a specific scenario. In this scenario, you are tasked to conduct a forensic examination of a computer used by an employee. The belief is that they have potentially exfiltrated intellectual property (IP) belonging to the company and shared it with a competitor. It is your job to determine if this has occurred.

You have determined through your analysis that the employee has exfiltrated data to an external hard drive. The external hard drive has not been found, and the belief is that the employee has the hard drive at home. A Production Order is a legal document that can be served upon the employee. It orders the employee to produce the device.

A Production Order is usually accompanied by a Preservation Letter. A Preservation Letter tells the employee that they are not to delete any information or otherwise destroy anything that contains potential evidence, pending any further instructions or investigation.

Suppose during your investigation you have determined that the employee had emailed the IP to a Gmail address, where the address gives no indication of who it belongs to. How would you go about determining who owns the email address? You can accomplish this through the issuing of a subpoena to the entity that you believe has the information you seek. Of course you need legal counsel to issue the subpoena.

Suppose during your investigation, you found potential evidence of criminal activity on the computer. In many jurisdictions, you have a duty to report the activity under penalty of law. This does not mean you immediately run to the police with your findings, although it may. Potentially informing your company's legal counsel could be enough. The point is that you should know the laws as they affect your jurisdiction.

It is also important to be careful what you call evidence. Evidence may be data that is held by the custodian, and the serving of a valid and jurisdictionally lawful preservation order or production order provides a legal medium of compliance for the custodian to provide this data. The use of the term evidence is subjective as this is ascribed by the investigator/examiner. The custodian of the 'evidence' just views this as data versus evidence.

As stated earlier, this is not to be construed as legal advice, and it is not designed to turn you into a lawyer. It has very often been the experience of the author though, that many in the legal field are not aware of how laws apply to digital evidence, or the process of dealing with it. You become a much more valuable examiner if you can provide guidance or assist with such matters.

It is quite common in investigations to need information from online entities. Here are just two few examples:

You have found a received email on your subject's computer and need to identify the sender.
You have found a post or message online for whom you need to identify the sender.

The first step in determining the answers is to identify who might have the information you seek, and then ask them for it. In our first example, we will be using a Gmail account.

Our subject received an email from 'mysansclass@gmail.com'. If we parse the headers of the email, we determine that the originating IP address traces back to Google. How did we do that? We identified our originating IP address and ran a WHOIS on it! We can perform some OSINT on the email address and hope it shows up somewhere. Even if it does, and we get a name, we may still have a problem. Just because we see a name with an email address somewhere online, does not mean it is legally admissible. In other words, we need hard evidence. The next step then, would be to issue a subpoena to Google.

Before a subpoena can be issued, it must have a legal action backing it. In other words, someone has to have filed legal action against someone. But how do you file action against someone when you don't yet know who they are? The whole idea of the subpoena is to find out who they are, but you don't know who they are, so you can't file an action against them to get the power of subpoena so that you can determine who they…..Ok. I think the point is made. In most jurisdictions around the world, there exists in civil law a mechanism called a "John Doe" action. It allows for the registration of a lawsuit without yet knowing who you will be filing action against. Once the action is filed you know have the authority to serve subpoena, in this case, on Google.

Once you can serve a subpoena, you must determine who specifically to serve it to. Search.org [1] is a great website that allows someone to determine the Custodian of Record for most every ISP and large online entity. In looking for the Custodian of Record for Gmail, it is determined to be thus:

Google LLC
Contact Name: Google Legal Investigations Support
Online Service: Google LLC
Online Service Address: 1600 Amphitheatre Parkway, Mountain View, CA 94043
Phone Number: (844)383-8524

It must be noted that the major online email providers do a terribly inadequate job of providing assistance in the civil realm. You will see that most all entries at search.org are geared towards law enforcement requests. It often takes a phone call or email (and hope for a response) to determine the appropriate method of submission. If you submit a subpoena to them that is not created in a way that they are amenable to, they will return it. At the end of the day though, if the request is done properly, you will now be provided with whatever you asked for.

This is not without its pitfalls. You must ask for everything you need in the proper way, because you will not get anything more than you ask for. For example, if you ask for subscriber name, address, and telephone number, you will get it. But when was the last time anyone verified that what a new account holder entered was correct? That doesn't mean you don't ask for it, but you want to also ask for date of account creation, IP address used for account creation, and connection logs. It is customary to ask just for connection logs that are in relation to the date and time of the offending communication.

At this point, you might be asking why the offending email traced back to Gmail in the first place? When an email is sent, it should have the originating IP address in the mail headers. When you send an email from your service provider account, for example joebloggins@comcast.net, your computer's IP address as issued by your ISP will be in the email headers. When someone uses an online email provider such as Gmail, Hotmail, Yahoo, etc., their actual IP address is not placed in the email header. Rather, an IP address that belongs to the email provider will be used. The provider does keep logs of the user's originating IP address though, and this is what you are hoping to get from them with your subpoena.

As if this isn't confusing enough, there is a very important caveat to the above. The above only applies if the email was sent via the online interface. If the sender has integrated their online email account with their local email program such as Outlook or Apple Mail, then even though it is being sent from a Gmail account, it will have the host computer's IP address in the email header. Many people will say that you don't even bother parsing headers for a Gmail email, but you should ALWAYS parse the headers. If the email was sent from inside a local mail program, you may not need a subpoena to Google now.

In the case of an email that has traced back to Gmail, once they respond to the subpoena, you will have possibly some usable data. Generally though, if someone set up a Gmail account to send an offending email, they didn't give Gmail their real name!. What you do have is the sender's actual IP address from where they logged in to send that email. This is what you were after. At this point, you are now in the same situation as if the email had been sent from a local ISP subscriber account.

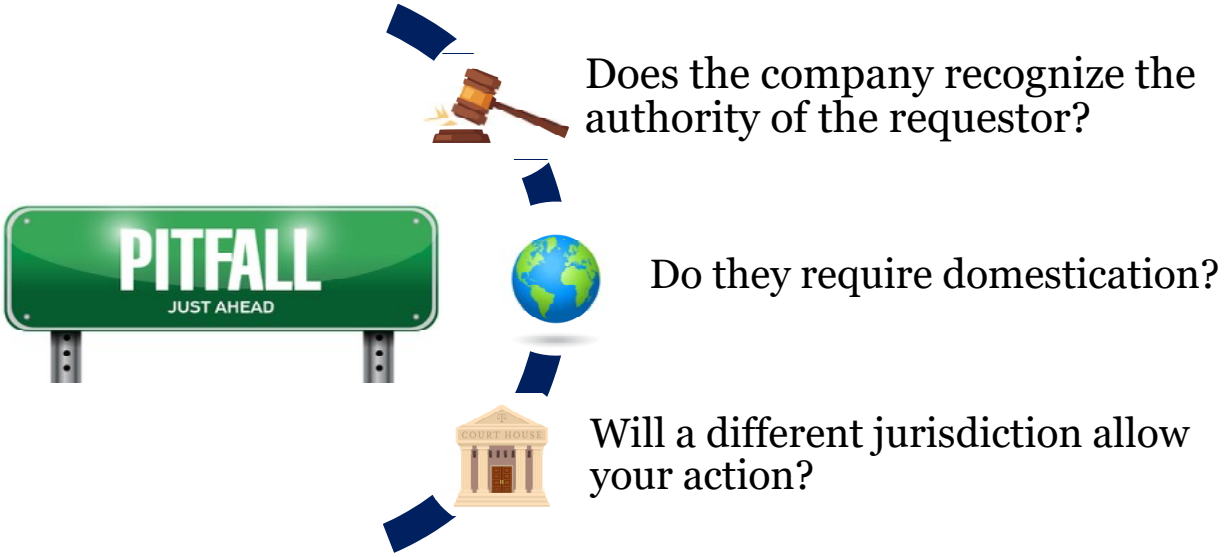
In the case of an email sent from a local subscriber account, the subpoena will be sent to the ISP, using all the same processes as with Gmail. Identify the ISP Custodian of Record and issue the subpoena to them.

If you are trying to identify the poster of a message on something like Twitter, or a blog post on a website, you will first attempt to identify the entity behind the website by doing a WHOIS on the domain name. This will be who the subpoena goes to.

Again a reminder that this is not legal advice. This is merely designed to assist you and/or counsel on how to get the data they need. This process can become incredibly complicated very fast, and there are many potential outcomes to the above.

[1] Search.org | `https://for498.com/vufra`

Subpoena Pitfalls



There are a number of potential pitfalls when attempting to issue subpoenas, and some of them are outlined here.

It is possible that the entity you serve the subpoena on will reject it because they do not recognize the authority of the requestor. An example of this would be an entity not accepting a simple subpoena because it is not approved by anyone beforehand. The entity would request something like a court order instead.

Larger entities such as Facebook and Google will almost always demand that a subpoena be "domesticated" in their jurisdiction. This means that no matter where you are from, you must hire counsel in the jurisdiction that the receiving entity is in, and have them register your action in a court there. For example, Facebook is headquartered in Menlo Park, California, and may require domestication in their jurisdiction.

Laws, both civil and criminal, can vary from jurisdiction to jurisdiction, meaning differences between counties, states, provinces, countries, etc. This differences can have significant impact on your ability to support legal process.

Summary

- Understanding a bit of history can give insight into why things are the way they are
- We are out of IPv4, and IPv6 is the wave of the future
- DNS converts numbers to names & vice versa
- A significant amount of information can be derived from various online resources
- Sometimes the only recourse left is legal

This page intentionally left blank.

Exercise 5.1A-B

Online Attribution

Synopsis: In this exercise, you will use Sam Spade to gather online identification information. You will then use DomainTools to identify useful information about a domain or IP address. Finally, you will use Archive.org to find information that no longer resides online.

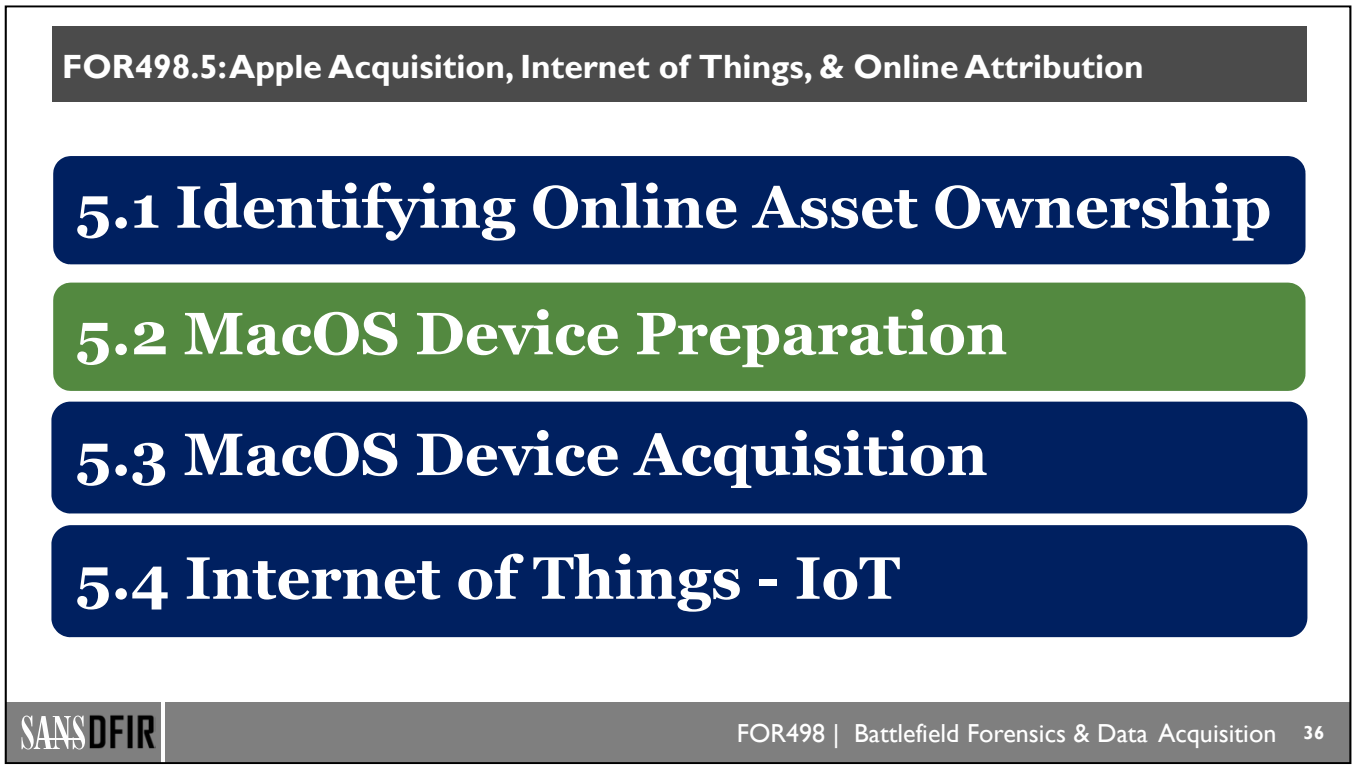
Average Time: 40 Minutes

This page intentionally left blank.

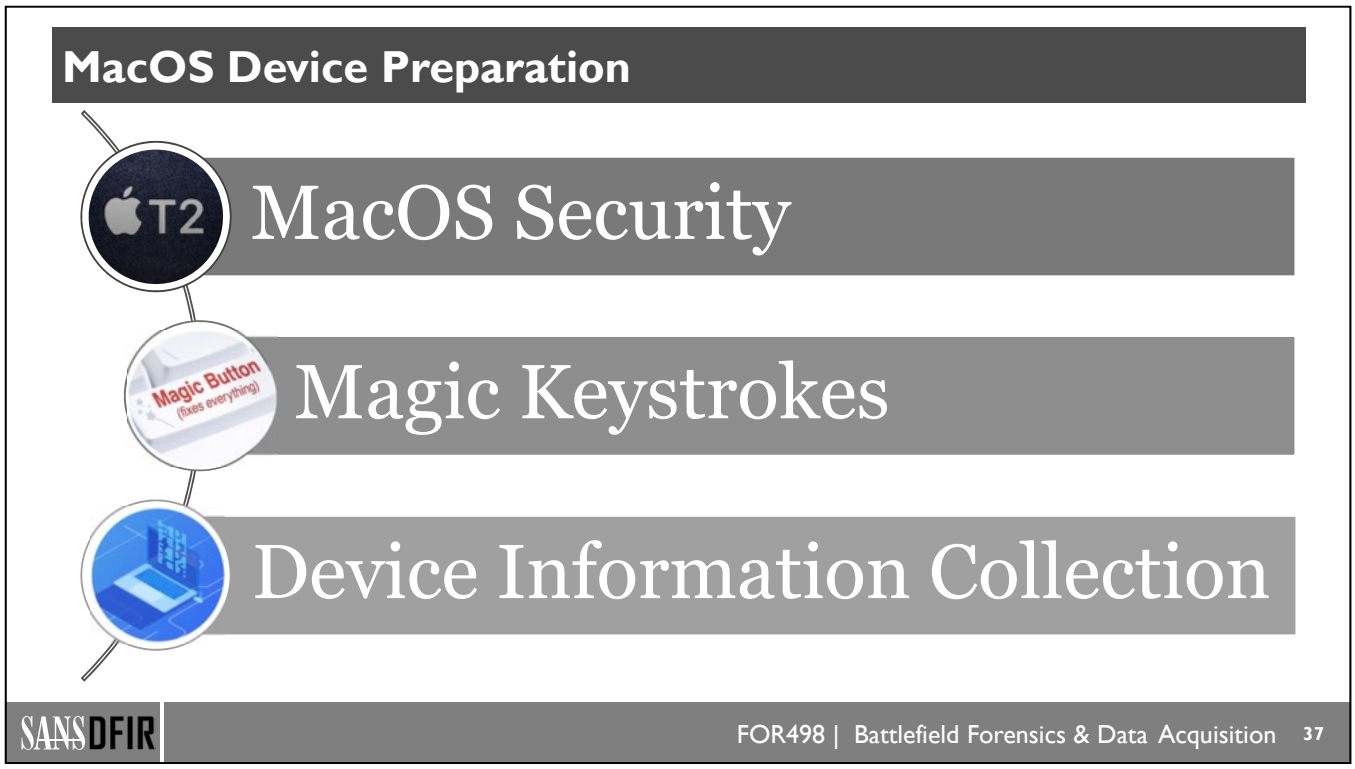
Exercise 5.1A-B Takeaway

- A simple Whois search almost always contains less information than you can find by performing a more in-depth analysis.
- Old tools like Sam Spade can do things that no other tool can do.
- Whois information is public information (for the time being).
- Whois Registrars are mandated to provide a certain amount of information for free.
- DomainTools is one of the most robust tools for IP and domain information. The subscription version allows for a great deal of aggregation and historical information.
- Half the battle in gathering information is knowing where to look and how to ask.
- Archive.org will maintain information, including websites, long after information has been changed or removed.
- Websites like search.org assist in finding the entities responsible for Internet infrastructure.

This page intentionally left blank.

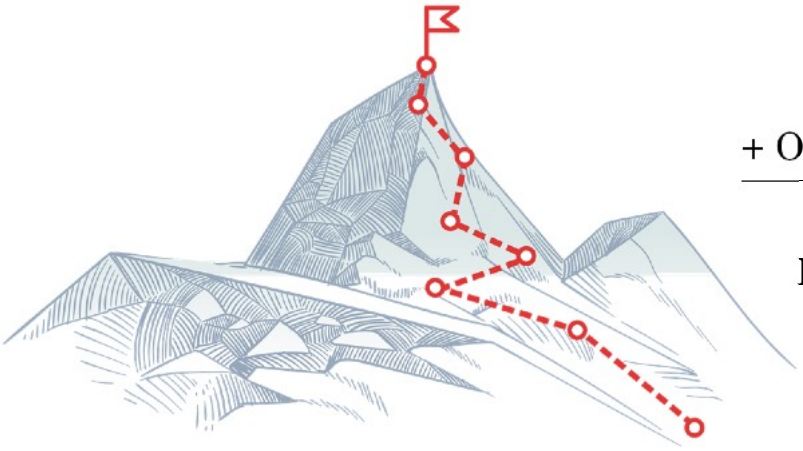


This page intentionally left blank.



This page intentionally left blank.

Apple Acquisition Challenges



Physical format
Storage format
Encryption format
File system format
+ Old school thought process

NEED FOR ENTIRELY DIFFERENT APPROACH

There is no question that Mac computers are gaining market share, and as forensic examiners, we see more of them in the lab. Many labs that have been doing forensics on Windows computers think they can just pull a hard drive from a Mac, image it the way they always have, and then examine it. This is simply not true. This is akin to saying that because you can drive a car, you must be able to fly a plane, because they are both modes of travel.

Apple computers, most likely due to their smaller market share, have a smaller amount of tools that a digital forensics examiner can work with. When there is little competition, (and little demand), cost of these tools can be extremely prohibitive, and examiners look for workarounds. In some cases there are workarounds, but in many cases there is not.

Many complaints are made about the difficulties faced with data collection from Apple devices, however we must always remember that these devices were not designed with forensic examination in mind. They were designed to be sleek, slim, fast, and beautiful. The design choices used to achieve these goals are the same things that lead to frustration in the collection process.

A lab must decide whether it will handle Apple devices or not. It is not a prudent position to assume that a hard drive is a hard drive. Too many labs examine Apple devices just as they would Windows devices. It is true that the large forensic suites typically in use by many examiners will acquire a hard drive from an Apple device, but what is being acquired? And how is it being analyzed? If you believe that everything being presented to you is all that is available, you have failed before you have started.

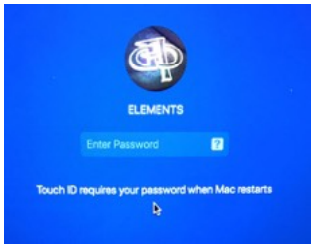
If you expect to conduct a proper examination on an Apple device, you are best served doing it on an Apple device. Even where a Linux distribution may allow you to image the drive, the examination tools are another matter. Challenges exist with the HFS+ file system, not to mention the newer Apple File System (APFS). CoreStorage and FileVault are just two of a number of other complications that cause acquisition of Apple devices to be very challenging.

Although Apple devices make up a much smaller market share than their Windows counterparts, they have developed enough of a following that they cannot be ignored in today's forensic world. The problem that has arisen for anyone that has been performing forensics for any length of time is that we cannot approach the Apple products in the same manner, and too often, the investigation world is very resistant to change. Many mistakes have been made by examiners who endeavored to approach an Apple forensics case in the same manner as a Windows computer. These devices are very difficult to take apart, and in at least one case, the first tool needed is a suction cup of all things.

You have the choice with Apple to acquire it either live (if the machine is on when you get to it) or dead (removing the hard drive to image it separately from the device).

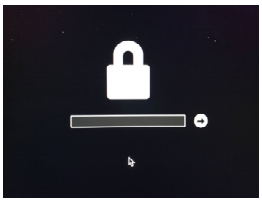
MacOS Security

User logon password



FileVault whole disk encryption

Firmware password



There are different places within the MacOS architecture that a user can implement security. The simplest level is usually a User's logon password, and this is what most users have as the only security on their device. The next level of security would be enabling the FileVault [1] found under **System Preferences -> Security & Privacy -> FileVault.** This enables whole disk encryption of any HFS+ or APFS volumes. The third, and least known, security application on a MacOS device is the Firmware Password [2].

This is not an Apple security course; however examiners need to be aware of the different levels because each and every one of them affects the acquisition process in different ways that will be explained as the module progresses.

In addition to these software based security methods, a new layer of security was introduced in late 2018 in the form of the Apple T2 security chip [3]. This hardware based security layer has changed some of the processes that the examiner used to have access to, and we will explore this later in the module.

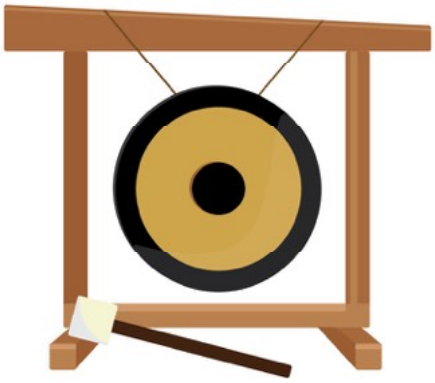
[1] File Vault whole disk encryption | `https://for498.com/f6aj9`

[2] MacOS Firmware Password | `https://for498.com/zljv-`

[3] T2 security chip | `https://for498.com/4mxyd`

Apple POST Codes

- GONG
- GOOOOOOOOOONG
- GONG GONG GONG
- GOOOOOOONG GOOOOOOONG GOOOOOOONG
- This is not just any gong show!



Although Apple products do not have a BIOS like Windows computers, they do perform a Power-On Self-Test (POST) at startup, like Windows. Any Apple user will be familiar with the gong sound once you press the power button.

Prior to the computer issuing the gong sound, it has run checks to determine its functionality, and although you may have only ever heard the gong, there are other startup sounds that would occur based on the detection of problematic hardware.

Apple startup sounds | `https://for498.com/ie6-n`

"MUST KNOW" Apple Key Combinations

- alt/option key – For booting options
- ⌘ + S – Enter Single User Mode
- ⌘ + R – Enter Recovery Mode
- T – Target Disk Mode

During various phases of the acquisition process, it may be necessary to use special keys or combinations of keys to access areas of the Mac or allow it to be booted in modes other than "normal". When performing these keystrokes, it is assumed that you will be holding the keys down and then pressing the power button while continuing to hold the buttons down until certain things happen as outlined in the following slides.

In a normal acquisition situation, you may be turning off the device and using various key combinations to restart it as many as 4 or 5 times before the acquisition is done. This is by no means an exhaustive startup key list [1][2], but it covers the combinations that we will use for most acquisition functions.

NO startup key combinations will work when the system hard drive is removed. This is of critical importance, as the takeaway is that the system hard drive is still in the machine and susceptible to write activity if the examiner does not understand what they are doing, or the proper order to do it in.

The only time that these startup key combinations will be used is in the case of acquisition of an Apple device that is powered off. None of these will be used in the case of imaging a live system.

[1] Apple startup keys | `https://for498.com/7jsd1`

[2] More Apple startup keys | `https://for498.com/s9beg`

alt/option Key (1)

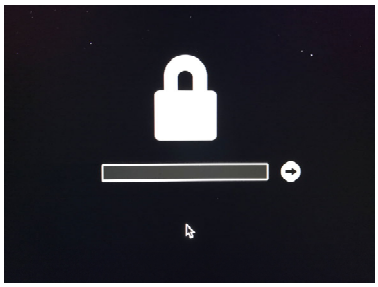
alt/option key – For booting options

The alt/option key is used to start the computer to the point where it shows any bootable partitions that might be available to the system. An examiner will do this for potentially a number of reasons. The first reason would be to determine whether or not there is a Firmware password on the computer. Another reason would be if the examiner was using a tool like MacQuisition and booting from a thumb drive that contains it. We will be discussing MacQuisition later.

In virtually every scenario of acquisition of an Apple machine that is off, this would be the first startup key used.

alt/option Key (2)

Hopefully you see something like this...



...and not something like this

The initial application of this startup key is when beginning the acquisition process. The first thing to watch for is whether or not you are presented with a field with a padlock beside it. If you see this, it indicates that a Firmware Password is enabled on the system. If you do not have this Firmware Password, the acquisition process is over, or at least from the perspective of using the subject system as an acquisition host.

There is one caveat to this. Depending on the circumstances of your investigation or need for acquisition, you could boot the computer up as a normal user. In most cases of the presence of a Firmware Password, the machine will boot to its default storage without having to enter the Firmware Password. Once booted, you would perform a live acquisition, but ensure that you have documented clearly and completely what you did and why.

Hopefully you will have the Firmware Password if you see the Firmware Password screen!

In the best of scenarios, you will start the device and be presented with the bootable device selection screen. You will have done this for a couple of reasons. The first will be to simply see what you are presented with, and document it. The boot storage in the device will be visible by name here. You will also see whatever bootable media you have connected to the device for your acquisition, and it is from here that you will boot to that media to begin the acquisition process.

You can see in the slide that there are five different partitions being presented. It is important to understand that although some of them look like internal hard drives, and some look like external hard drives, they are not actually representing hard drives at all. They are representing bootable partitions. We could have 3 other external hard drives connected to the computer and 2 more partitions on the internal storage, but if they are not bootable, they will not show up here. In the example in the slide, there is a drive named Windows. This is for a Bootcamp partition on the computer. There is a drive named Macintosh HD. This is the MacOS

partition on the computer. It is important to note that you must name any destination hard drives a very unique name, so as to not get them mixed up with any drives in the actual computer. There are three other partitions being presented. These are all from a MacQuisition USB drive that is connected to the computer. In fact, the MacQuisition USB drive has 2 more partitions on it, but you cannot see them in the slide because they are not bootable.

⌘ + S Keys

⌘ + S – Enter Single User Mode

⌘ + S – This key combination is used to start the computer in Single User Mode in all MacOS computers EXCEPT devices with the Apple T2 security chip. In the case of devices that have the Apple T2 security chip, this method is not available, and you will not be able to collect hardware device information. By following instructions on the slide titled "⌘ + R keys", you will at least be able to verify date/time of the device.

Single User Mode is used to gather information about the computer and storage media, typically for troubleshooting purpose, but we use it for our own purpose. Although it is pulling data from the storage media itself, this data is akin to the BIOS/UEFI information in a Windows computer. The date and time of the machine is the most critical of data. As explained on day one, without the date/time that the machine is reporting, you have little to no context of when something happened.

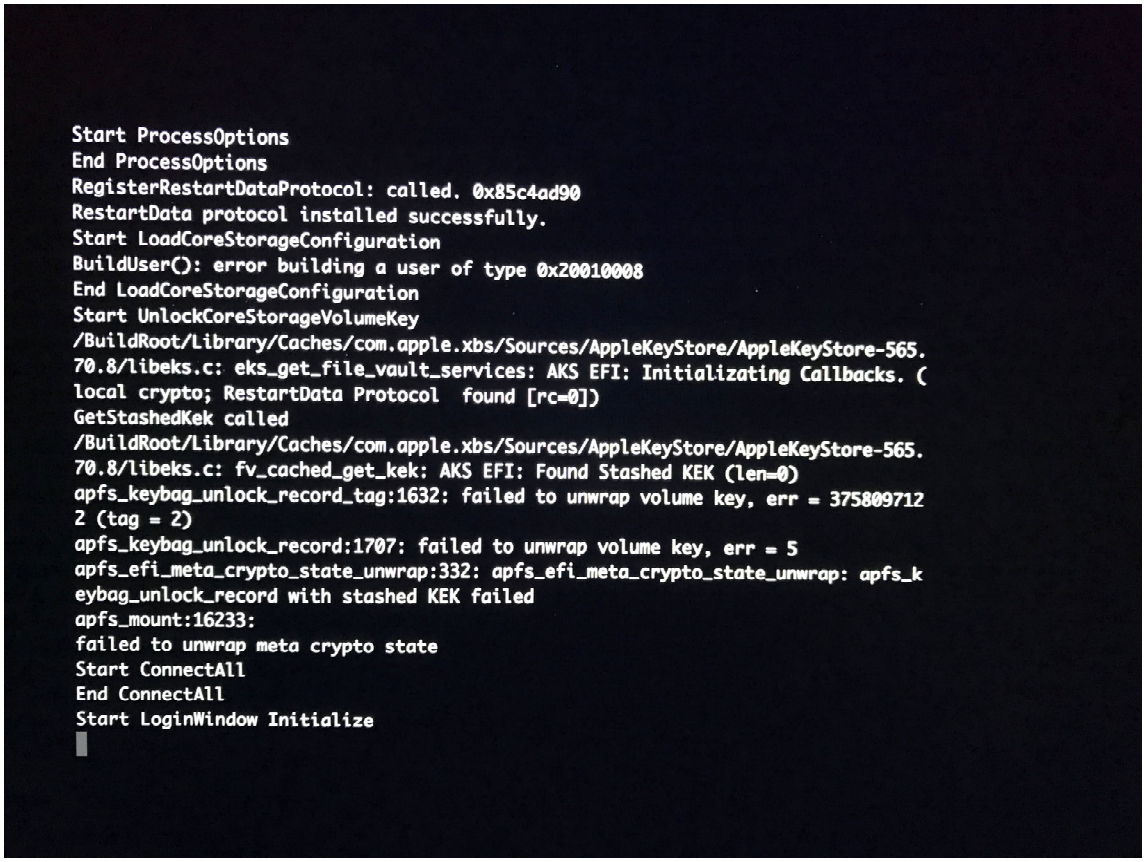
Because the data being populated in Single User Mode is coming from the subject storage media, it can only be collected when the media is in the computer. This is unlike Windows computers, where it is recommended to gather BIOS/UEFI data when the hard drive is removed. As well on Windows computers, this data exists on a chip completely separate from the storage media, unlike Apple.

As a result, the examiner must understand the potential volatility of performing such a task. If everything works as planned (and does it ever?), when the computer boots into Single User Mode, the storage media is mounted in a "read-only" manner. This is good for us. What is bad for us is not getting the keystrokes exactly right at bootup. The only realization of this is when the system boots straight to the normal login screen.

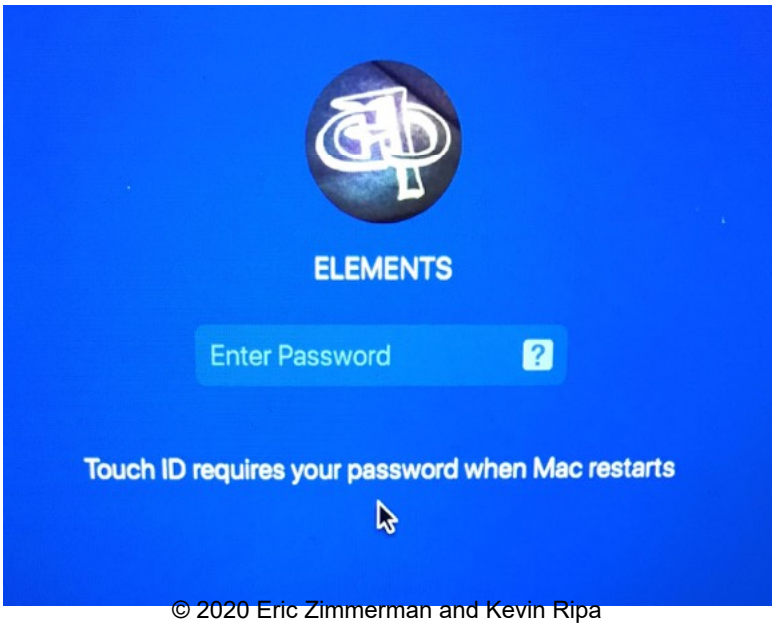
Another consideration is if a user has FileVault enabled. This is Apple's whole disk encryption. If a user has FileVault enabled, you will not be able to enter Single User Mode without knowing the user's password.

The resources that are being drawn from to provide this information are from the System Profiler. This is the program/service that is being accessed when, on a normally booted MacOS, the user clicks on the Apple logo at the top left corner of a screen and selects “About This Mac”.

Once the ⌘ + **S** key combination is invoked at startup, you must continue to hold these keys down until you see a black screen with writing on it, as seen below.



Once you see this, you can release the keys and you will see large amounts of text rapidly scrolling by. After a few seconds of this scrolling, it will either stop, or you will be presented with a login screen as seen below, that looks suspiciously like a normal login screen.

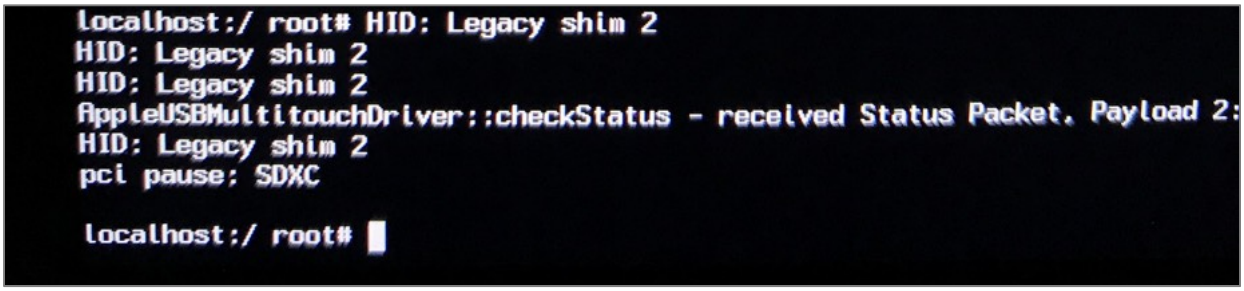


You may be worried that the process did not work. Seeing this screen merely means that FileVault encryption is enabled, so you must authenticate with the user password to proceed further. If all worked as it should, once you enter the password, it will go back to a black screen with more text scrolling. If it does not, and you see the system desktop, something has gone wrong and you have missed Single User Mode. If so, you will now understand why we image first and do this later!

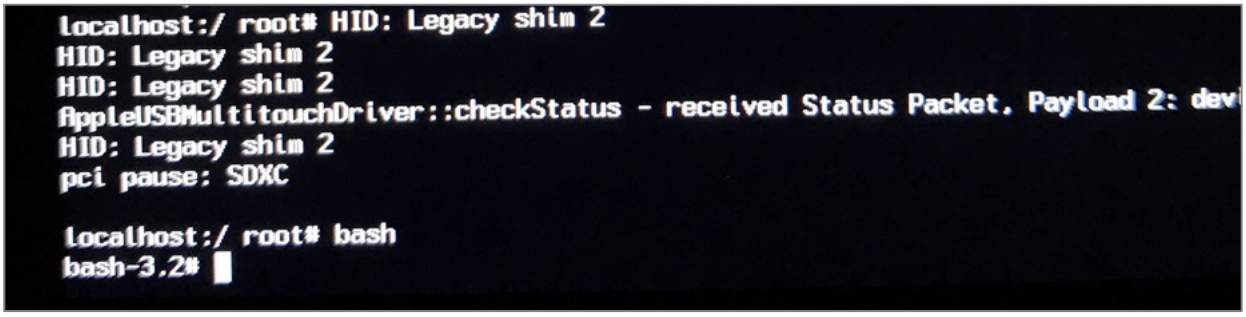
There are generally two reasons for this. The first is that you possibly didn't get the key combination right, or at the right time. The second is that the Firmware Password is set. If it is set, you cannot boot into Single User Mode unless you disable the Password. Disabling the Firmware Password is explained shortly.

Suggesting it has all worked properly, and you are in Single User Mode, you will see a curser at the end of all of the text, and it may seem as though the loading is not finished, as seen below.

At this point, just press the "Enter" key, and you will be presented with a normal looking cursor line as shown below.



We now have to get into the bash shell. Do this by typing **"bash"** and then pressing "Enter". You will now see the screen below, and we are ready to start extracting data about the system, which will be covered later in the module.



In Depth Single User Mode | `https://for498.com/cz4jp`

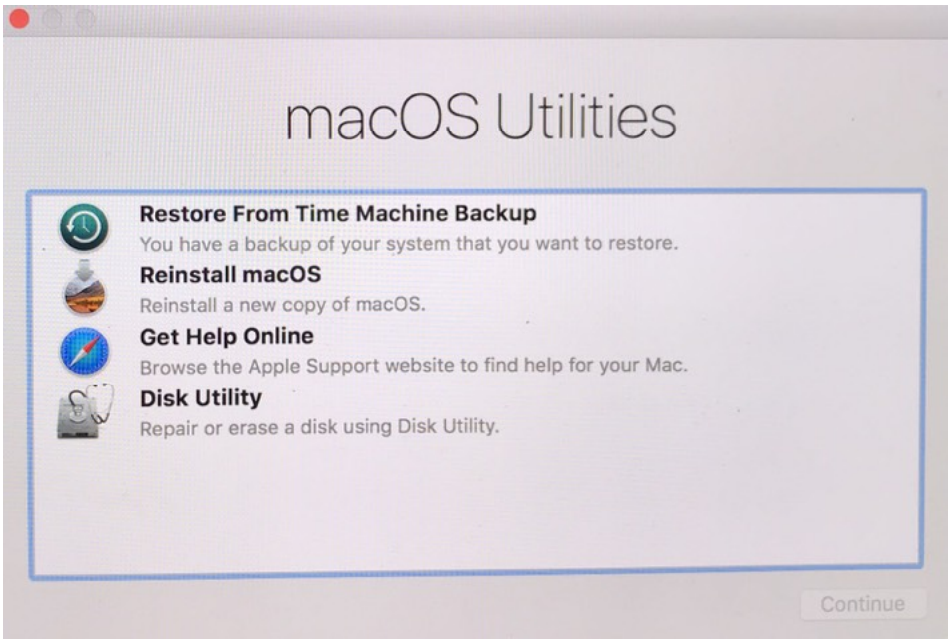
⌘ + R Keys

⌘ + R – Enter Recovery Mode

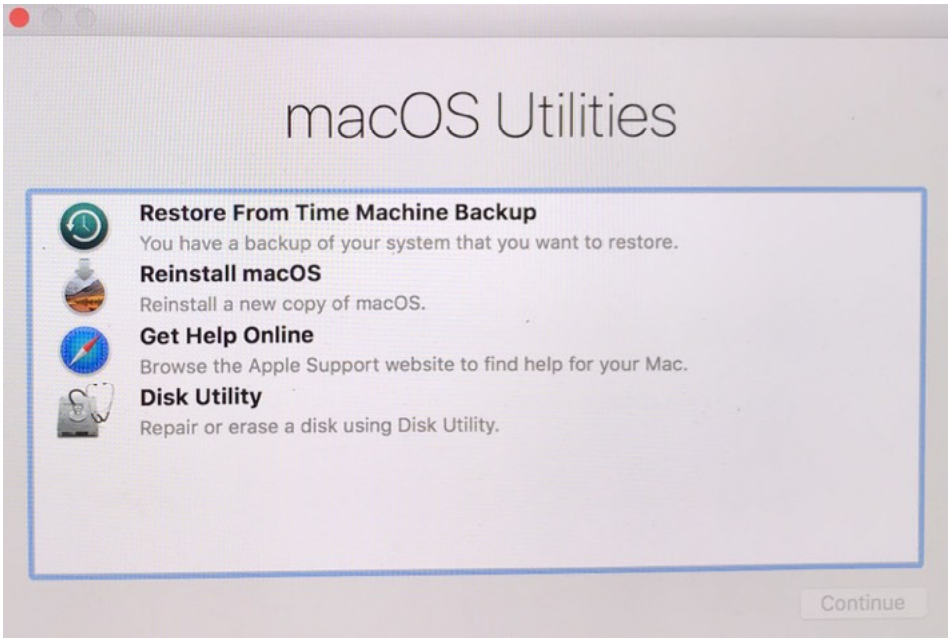
⌘ + R – This key combination is used to start the computer in Recovery Mode. There are generally two reasons we would enter this mode for our purposes, but there are many more reasons for this mode.

The first reason we would use this is if the system you are imaging contains the Apple T2 security chip. You would use this key combination rather than the ⌘ + S combination in order to collect the date/time of the device. Unfortunately, when accessing the Terminal from Recovery Mode, the system_profiler utility is not available for any model of Apple.

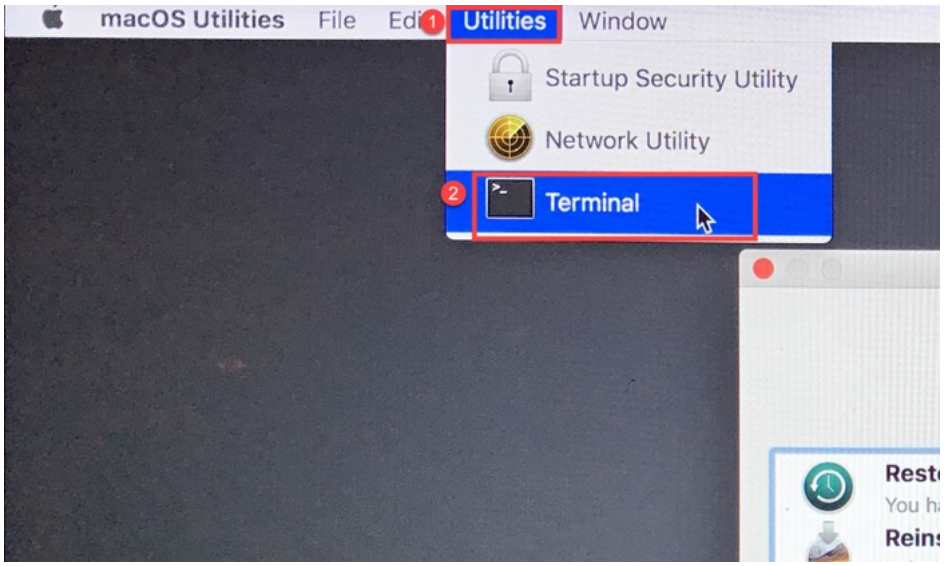
The second reason would be if the computer has a Firmware Password set. The examiner would have to enter this mode to disable the Password. We will cover this further in the module.



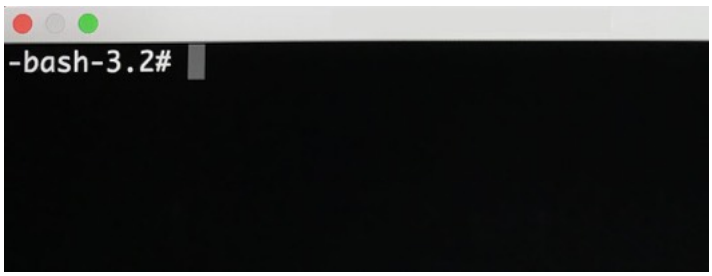
Hold the key combination down while you press and release the power button on the machine. Continue holding the keys down until you see the Apple logo and a progress bar. You can then release the keys and wait for the machine to load the Recovery mode as seen below.



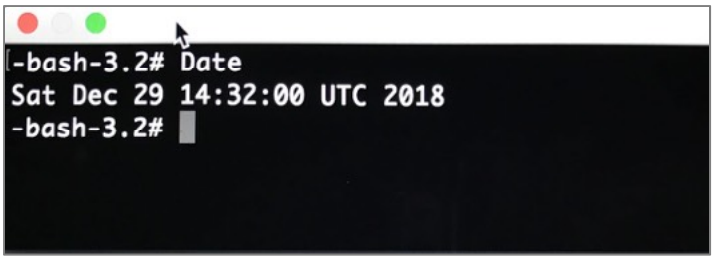
For the purposes of collecting the date/time in the case of a device with a T2 security chip, at the top of the screen, click on Utilities, and then on Terminal.



You will now see a terminal window open.



At the prompt, type **"Date"** and press Enter. You will be shown the date and time that the machine is reporting. Record this with a photo of the screen.



"T" Key – Target Disk Mode

T – Target Disk Mode

SANS DFIR FOR498 | Battlefield Forensics & Data Acquisition 52

"T" – Holding down the T key while powering up the computer places it into Target Disk Mode (TDM) [1]. This is the mode necessary for forensic acquisition without other tools. This mode does not bypass any security measures but offers the ability for an examiner to create a forensic image of the primary drive in the device, when the examiner has no other tools at their disposal except another Apple computer. TDM will also work when the computer is connected to a Windows computer, however in order for this to work, the Windows computer would require third party drivers and software. We will not be discussing this here.

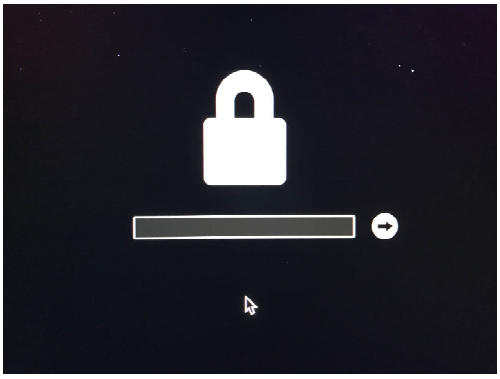
TDM allows for the subject device to be connected to a separate device for various reasons. The reason that matters to us is for the purpose of forensic acquisition of its storage media. The subject device gets booted into TDM, then a cable is used to connect the subject device to the machine that will perform the acquisition. Standard USB is not an acceptable connection method. The only interfaces that support TDM are Thunderbolt, USB-C, and Firewire, so these interfaces have to be on both devices. It is important to note that the USB-C power cable that ships with Macs cannot be used for TDM.

When a device is placed in TDM, it can be best be described as the Mac now being addressed as an external storage device. Having said that, there are two very important distinctions. The first is that TDM will only allow for addressing of the primary storage on a device. If there are 2 hard drives, or the machine has a fusion drive setup, TDM will only see the first drive and no others. The second important distinction is that TDM does NOT write block the storage within it. In other words, it is NOT "read only". Plugging it into another computer in TDM without first taking some important precautions will cause write activity to the subject drive. We will discuss this further during the TDM acquisition phase of the module.

It is a very simple matter to place a device in TDM. Simple hold down the “T” key and press and release the power button. Keep holding the “T” key down until you see a lightning bolt appear. You are now in TDM.

[1] Target Disk Mode | `https://for498.com/loe7p`

Firmware Password

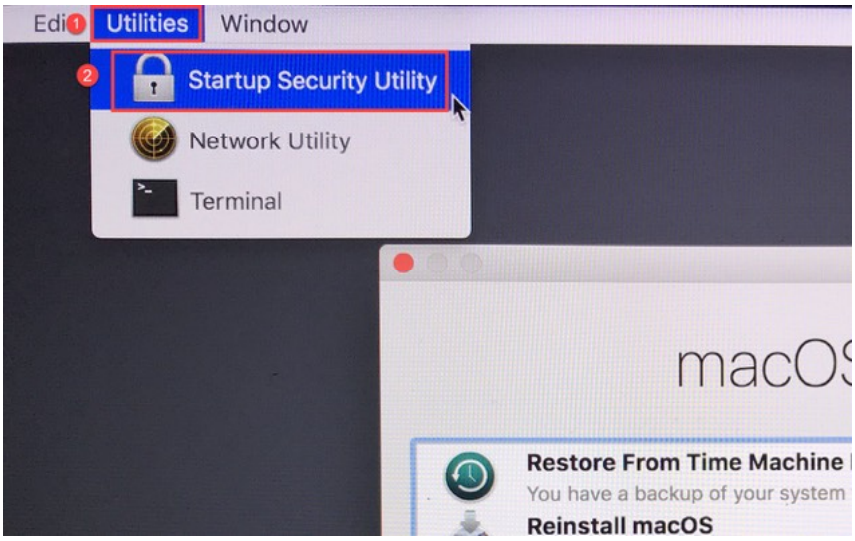


- Set from within Recovery Mode
- Cannot boot device without it
- Cannot enter Recovery Mode without it
- Cannot enter Single User Mode unless disabled
- Cannot reset computer without it

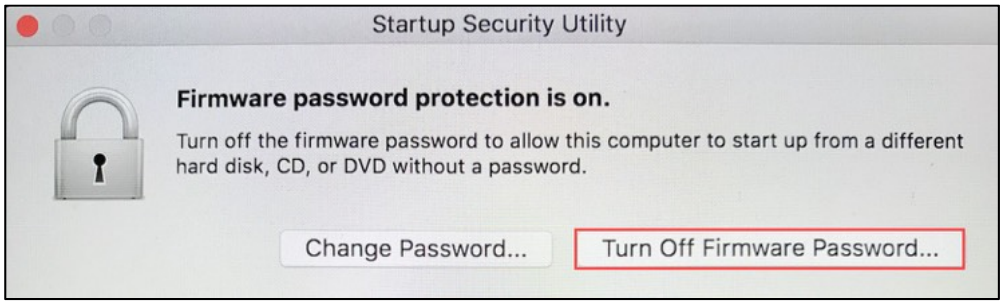
A Firmware Password is a password that can be created on a MacOS device to secure it from unauthorized entry. This password is enabled and disabled from a computer's Recovery Mode. When it is set, it will restrict access to any storage media, as well as Recovery Mode itself. If a user has Bootcamp for Windows installed as a separate partition on the hard drive, they will have to enter the firmware password to change to that partition. This does not need to be the same as a user's logon password, and it can also be set so that it will not prompt for a password as long as the computer is logging into a default boot volume. This can create issues because a user could set it and forget about it, and then an examiner is significantly limited to what they can do from an acquisition standpoint. This is yet another reason why we may want to perform a live acquisition if the machine is already on.

As previously indicated (and notwithstanding machines with a T2 security chip), the firmware password must be removed in order to use Single User Mode. Alternatively, you would be limited to gathering only the date/time in the Recovery Mode terminal.

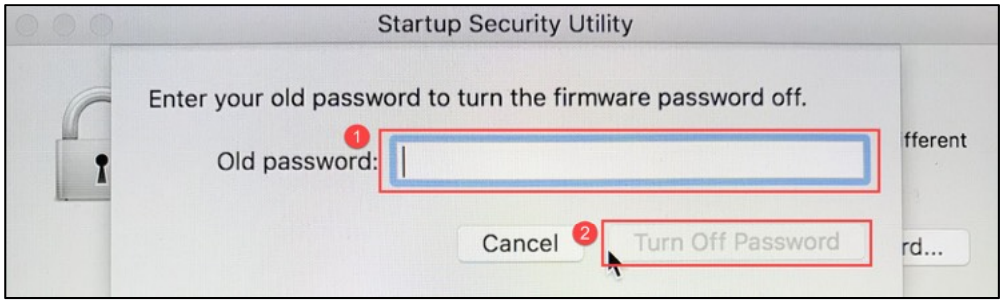
To turn off or disable the firmware password, enter Recovery Mode and click on Utilities -> Startup Security Utility.



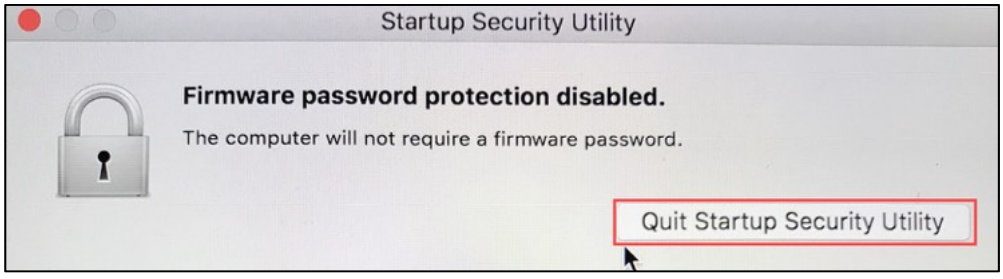
The utility will open, indicating that Firmware password protection is on. Click on **"Turn Off Firmware Password…"**



A new box will appear and prompt for the password. Enter it, and then click on **"Turn Off Password".**



Once done, a confirmation will appear. Click **"Quit Startup Security Utility"**.

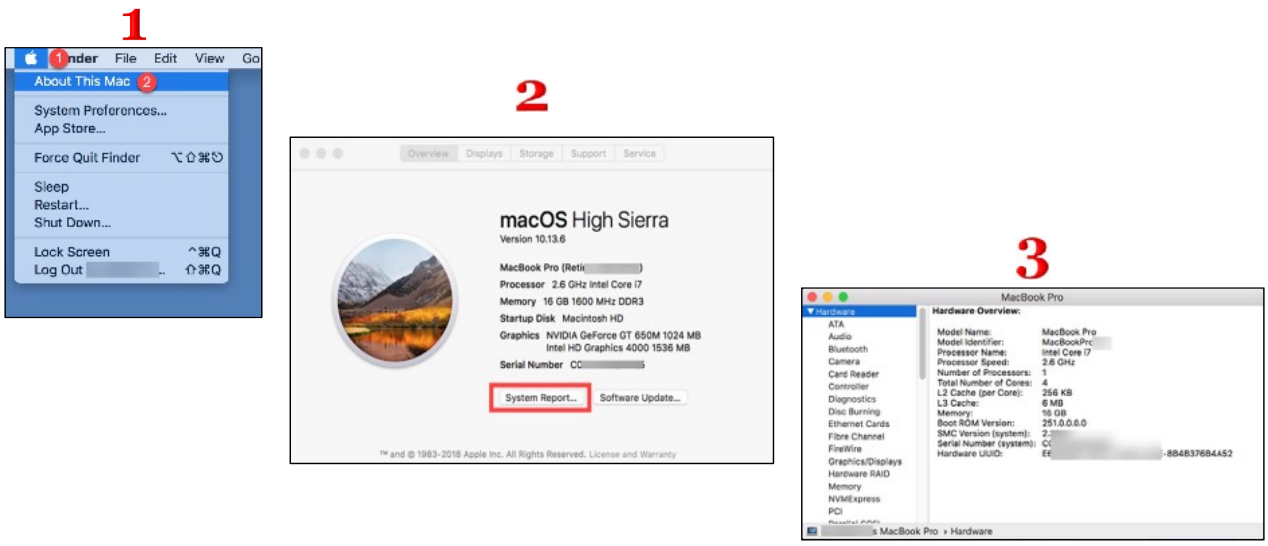


You can now click on the Apple icon and shut down the device. Then you can start it into Single User Mode.

It is widely known that the resale market for stolen Apple products far exceeds anything on the Windows side of things. Apple devices are targeted for theft above all others, because they hold their value so well. In any case of an Apple product being stolen by opportunists trying to make money, the thieves are not interested in the data the computer holds. They also don't care if FileVault encryption is enabled. They simply enter Recovery Mode, reset the computer to factory settings, and have the device sold in no time.

While the firmware password cannot stop the theft of a device, it can deny the thief any chance of resale. The device cannot be reset to factory without entering Recovery Mode, and Recovery Mode cannot be accessed without entering the firmware password first.

System Profiler GUI



The system profiler is a utility that is present on Apple computers. Just as the name suggests, it allows for the profiling of a system.

Most anyone that has used an Apple computer for more than 5 minutes has used the system profiler, but probably didn't know its name. When logged in to an Apple computer, a user can click on the Apple symbol on the top left corner of the screen, and then click on "About This Mac".

An overview screen will appear with various tabs and buttons. Click on "System Report".

The next screen to appear will likely be the hardware overview, but will also list all of the things that the program can provide information about. We utilize system profiler in this manner while performing live acquisitions, to collect system information like the processor, RAM, and hard drive information, as well as date/time.

System Profiler CLI

- How do we get the date and time on a Mac system?
- There is no BIOS
- Type “date” in the CLI
- Take a photo

When we are faced with doing an acquisition of a drive from a computer that is off at the time of seizure, we don’t have the luxury of always being able to boot the system. We want to do as little damage as possible, so even after the forensic image is acquired, we may want to think twice about just booting the image.

System Profiler [1] gives us the ability to extract data about the system hardware and settings at the command line in Single User Mode. Entering Single User mode has already been covered previously, so will not be covered again. Rather, we will assume that you have already booted to Single User Mode based on the instructions earlier in this module and are currently at the command prompt.

Remember that we treat everything as though it is (or may become) evidence. You may think at the time that you don’t need to worry about it. If you are ever in a situation where you are sure that you don’t need to collect something as potential evidence, you need to ask yourself this question. “Am I willing and able to support my decision today in a court of law tomorrow?” It is always better to have it and not need it, than need it and not have it.

The first thing to collect is the date/time. We do this by typing “date” at the command prompt and pressing “Enter”. Record this with a photo.

[1] System Profiler | `https://for498.com/xr4qd`

System Profiler Options

We want to use this option...

```
bash-3.2$ system_profiler -listDataTypes
Available Datatypes:
SPParallelATADataType
SPUniversalAccessDataType
SPApplicationsDataType
SPAudioDataType
SPBluetoothDataType
SPCameraDataType
SPCardReaderDataType
SPComponentDataType
SPiBridgeDataType
SPDeveloperToolsDataType
SPDiagnosticsDataType
SPDisabledSoftwareDataType
SPDiscBurningDataType
SPEthernetDataType
SPExtensionsDataType
SPFibreChannelDataType
SPFireWireDataType
SPFirewallDataType
SPFontsDataType
SPFrameworksDataType
SPDisplaysDataType
SPHardwareDataType
SPHardwareRAIDDataType
SPInstallHistoryDataType
SPNetworkLocationDataType
SPLogsDataType
SPManagedClientDataType
SPMemoryDataType
SPNVMeDataType
SPNetworkDataType
SPPCIDataType
SPParallelSCSIDataType
SPPowerDataType
SPPrefPaneDataType
SPPrintersSoftwareDataType
SPPrintersDataType
SPConfigurationProfileDataType
SPRawCameraDataType
SPSASDataType
SPSerialATADataType
SPSPIDataType
SPSmartCardsDataType
SPSoftwareDataType
SPStartupItemDataType
SPStorageDataType
SPSyncServicesDataType
SPThunderboltDataType
SPUSBDataType
SPNetworkVolumeDataType
SPWWANDataType
SPAirPortDataType
```

...and this one

SANS DFIR | FOR498 | Battlefield Forensics & Data Acquisition 59

System Profiler will now be used to collect the data that we need. The MAN or Manual page is shown below. The MAN page shows all of the various features of a program and how to run it to achieve your goal. You can see this MAN page, by typing “man” before any program you want to run. In this example, you would type “man system_profiler” and press “Enter”.

DESCRIPTION

system_profiler reports on the hardware and software configuration of the system. It can generate plain text reports or XML reports which can be opened with System Information.app

Progress and error messages are printed to stderr while actual report data is printed to stdout. Redirect stderr to /dev/null to suppress progress and error messages.

The following options are available:

-xml	Generates a report in XML format. If the XML report is redirected to a file with a ".spx" suffix that file can be opened with System Information.app.
-listDataTypes	Lists the available datatypes.

-detailLevel level Specifies the level of detail for the report:

mini report with no personal information

basic basic hardware and network information

full all available information

-timeout Specifies the maximum time to wait in seconds for results. If some information is not available within the specified time limit then an incomplete or partial report will be generated. The default timeout is 180 seconds. Specifying a timeout of 0 means no timeout.

-usage Prints usage info and examples.

EXAMPLES

system_profiler
Generates a text report with the standard detail level.

system_profiler -detailLevel mini
Generates a short report containing no personal information.

system_profiler -listDataTypes
Shows a list of the available data types.

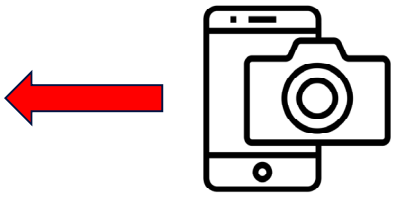
system_profiler SPSoftwareDataType SPNetworkDataType
Generates a text report containing only software and network data.

system_profiler -xml > MyReport.spx
Creates a XML file which can be opened by System Profiler.app

We could simply run the command **"system_profiler"** at the command prompt and press **"Enter"**, but this would not be efficient. Without giving system_profiler some parameters to work with, it will simply grab everything. Not only is this time consuming, but it generates approximately 1580 printed pages of data; most of it useless for our purpose. System Profiler has approximately 50 different options you could use at the command prompt.

System Profiler Output (1)

```
bash-3.2$ system_profiler SPHardwareDataType
Hardware:

    Hardware Overview:

      Model Name: MacBook Pro
      Model Identifier: MacBookPr
      Processor Name: Intel Core i7
      Processor Speed: 2.6 GHz
      Number of Processors: 1
      Total Number of Cores: 4
      L2 Cache (per Core): 256 KB
      L3 Cache: 6 MB
      Memory: 16 GB
      Boot ROM Version: 251.0.0.0.0
      SMC Version (system): 2.3
      Serial Number (system): C02JH
      Hardware UUID: E6E0650F-8CFA-
                                    376B4A52
```



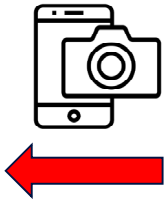
There are two commands from the System Profiler Data Type that we will use for our purpose, but you can use any of them as the situation dictates. The first one will show us the device information and serial number, as well as information regarding the processor and the RAM.

At the command prompt, type "**`system_profiler SPHardwareDataType`**" and press **"Enter"**.

System Profiler Output (2)

```
bash-3.2$ system_profiler SPSerialATADataType
SATA/SATA Express:

    Intel 7 Series Chipset:

      Vendor: Intel
      Product: 7 Series Chipset
      Link Speed: 6 Gigabit
      Negotiated Link Speed: 6 Gigabit
      Physical Interconnect: SATA
      Description: AHCI Version 1.30 Supported

        OWC Aura Pro SSD:

          Capacity: 1.92 TB (1,920,383,410,176 bytes)
          Model: OWC Aura Pro SSD
          Revision: P1225C
          Serial Number: OW1805[redacted]
          Native Command Queuing: Yes
          Queue Depth: 32
          Removable Media: No
          Detachable Drive: No
          BSD Name: disk0
          Medium Type: Solid State
          TRIM Support: No
          Partition Map Type: GPT (GUID Partition Table)
          S.M.A.R.T. status: Verified
```

```
Volumes:
  EFI:
    Capacity: 209.7 MB (209,715,200 bytes)
    File System: MS-DOS FAT32
    BSD Name: disk0s1
    Content: EFI
    Volume UUID: 0E239BC6-F960-3107-89CF-1C97F78BB46B
  disk0s2:
    Capacity: 1.22 TB (1,216,999,981,056 bytes)
    BSD Name: disk0s2
    Content: Apple_APFS
  BOOTCAMP:
    Capacity: 703.17 GB (703,172,968,448 bytes)
    Available: 348.5 GB (348,500,201,472 bytes)
    Writable: Yes
    File System: UFSD_NTFS
    BSD Name: disk0s3
    Mount Point: /Volumes/BOOTCAMP
    Content: Microsoft Basic Data
    Volume UUID: 8E39399C-2FFB-4B78-8113-ED1537EE55E4
```



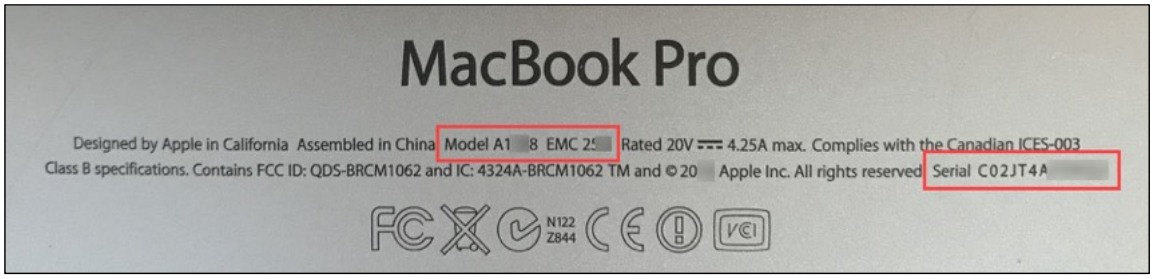
The second command will show us the hard drive information and serial number, as well as information regarding the Volumes on the drive.

At the command prompt, type "**`system_profiler SPSerialATADataType`**" and press **"Enter"**.

Once this data is catalogued, you are almost at the point of creating your forensic image.

Gather Machine Details (1)

- Model Number
- EMC Number
- Serial Number



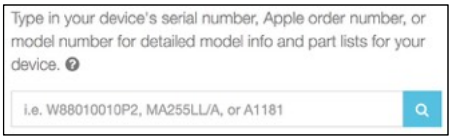
Mac computers present many challenges for forensic collection that are both easier, and harder than their Windows counterparts.

Although Macs provide extensive information about their inner workings via the Apple System Profiler feature, what happens in a forensic seizure situation when the computer is not on? Maybe you need to know what type of hard drive, and its location in the computer? Much like a Service Tag on a Dell, Mac computers provide a surprising amount of data without ever turning them on.

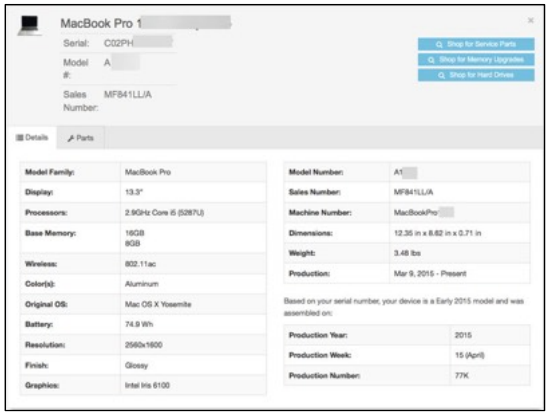
Let's look at the seeming minimalist information provided in the writing on an actual device. In our example, we are looking at the data on the back of a Mac laptop. The important information from the label consists of the Serial Number, the Model Number, and the ElectroMagnetic Compatibility (EMC) number. [1]

[1] Apple device numbers | `https://for498.com/w4yng`

Gather Machine Details (2)



From PowerBookMedic website



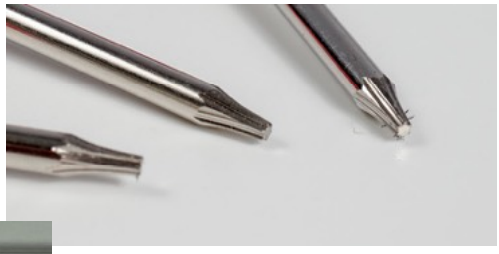
With this information, go to PowerBookMedic website at https://for498.com/s76iq and plug the serial number in to the search field. You will find all of the specs displayed about this device.

The only information it doesn't give is the hard drive specification. Scroll down the page and you will see a listing of all model numbers, and this will have the hard drive size.

One further important component of the website is that on a tab beside the information regarding the computer, there are all the details about replacement parts for the computer.

Tear Downs

Biggest challenge with Apple is teardown and unique fasteners



The first tool needed in the teardown of an older iMac...

It is not the easiest project trying to take an Apple product apart. Probably the best site on the Internet to assist with taking Macs apart would be at www.ifixit.com.

Tools have been specifically designed for Apple products, and using anything else can damage the fasteners. Any examiner needs a complete set of Pentalobe [1] screwdrivers and a suction cup, among other tools. Besides the above-mentioned website, there are hard drive removal instructions on other sites. [2]

[1] Pentalobe Screw | `https://for498.com/cpek4`

[2] Apple hard drive removal | `https://for498.com/bpur2`

Imaging Statistics

Drive	Connection	Hashed	Duration (minutes)
HDD 5400 RPM	Thunderbolt	No	27
HDD 7200 RPM	Thunderbolt	No	24
SSD	Thunderbolt	No	7.5
SSD (m-SATA)	USB 3	No	7.5
SSD	USB 3	No	8.2
HDD 7200 RPM	USB 3	No	20
HDD 7200 RPM (FVE)	USB 3	No	16
HDD 7200 RPM	USB 3	MD5	21
HDD 7200 RPM	USB 3	SHA-1	21
HDD 7200 RPM	USB 3	SHA-256	32

Imaged in Target Disk Mode to SSD at 64KB block size, No hash = 53 minutes
Imaged in Target Disk Mode to SSD at 512KB block size, No hash = 20 minutes

There are a considerable number of variations in imaging parameters generally, and many considerations when imaging Apple devices specifically. In most cases, typically due to Apple design, an examiner will be imaging a device using its own hardware, and or the hardware of another machine. Without the luxury of actually removing a hard drive and connecting it to a hardware imaging device, the examiner is already at a speed disadvantage. With smaller solid-state hard drives, the difference isn't that great across different methods, but if you are trying to image a 2 TB device, the chosen settings will have significant impact.

Four significant choices make up the setup process (among others). They are (1) destination media; (2) connection type; (3) hashing choice; (4) and block size. Let's review each of these.

Destination Media

Very little is understood about destination media and hard drives in general. Although it is commonly known that solid state hard drives are faster than spinning hard drives, it is not so commonly known that within these two types of drives there can also be many differences. The one difference that matters most to us for imaging purposes is the speed of the drive.

In the case of spinning hard drives, the most common speeds are 5400 RPM and 7200 RPM. The speed difference between the two is approximately 33%. All other things being equal, if a 7200 RPM destination drive will cause an acquisition to take 2 hours, the 5400 RPM will take 3 hours. In the case of a 2 TB hard drive, the difference in imaging speed is potentially 3 hours longer with a 5400 RPM drive, and then an additional 3 hours for the verification. Input and output operations (read/write) depend on a number of things such as if the reading and writing are contiguous or not, but in comparison, 7200 RPM hard drives will average about 150 MB/s and 5400 RPM hard drives operate around 100 MB/s.

Solid state hard drives have a much wider variation in speeds depending on interface type, but cheap SSDs will operate at approximately 250 MB/s, and expensive ones operate at approximately 3 GB/s. Given the very limited ability to harness this speed via USB 3.1, it becomes irrelevant for this application. A respectable (if slightly more expensive) happy medium sees SSDs operating at 540 MB/s.

An SSD would seem to be the best choice, but it is not that simple. If the drive you are acquiring is a spinning hard drive, then the read/write speeds of an SSD will be of no value because if the HDD can only be read at 150 MB/s, there is no gain having a destination drive that writes faster than that. Another consideration is size. A 4 TB destination hard drive is in the order of 20 times cheaper than a 4 TB SSD!

Connection Type

The next consideration is how you are connecting your destination media. Is it connected directly to the machine you are imaging? Or are you slaving the subject drive through your machine in target disk mode? Is the interface USB 2.0? USB 3.0? USB 3.1? Thunderbolt? Firewire? eSATA? USB-C? So many options. Thunderbolt would be the fastest of all, but it is not as common as other options. USB 3.1 and USB-C are the fastest today, with USB-C being preferential. Given that our discussion regards Apple devices, anything relatively new will have this interface.

Hashing Choice

When a forensic image is gathered, it is almost mandatory to ensure you are hashing the data at the time of acquisition, and then verifying immediately after. Of course there are situations where this may not be an option, however if you do not collect at the time of acquisition, you had better have a good explanation.

The hashing algorithm takes operational overhead. In other words, the imaging process will take longer if you are hashing during acquisition (which you should be). Bit length of the hash output will have an impact on imaging speed. The MD5 hashing algorithm output is 128 bits long. SHA-1 is 160 bits, and SHA256 is 256 bits. The greater the bit length, the longer the time to calculate it. If you elect to hash with two algorithms rather than one, this will also take longer. There is very little argument to support hashing with more than one algorithm when imaging, although it is seen frequently.

Block Size

Blocks can mean many things when referring to data. In the context of forensic imaging (acquisition), data is read from the subject drive to the destination drive in chunks called blocks. Each of these blocks is hashed and verified as part of the overall hashing function. It would stand to reason then, that the larger the blocks, the fewer the calculations, and that is true.

What happens if there is a bad sector or bad data area on the subject drive? The acquisition software will skip the entire block of data, writing zeros to the entire block to account for the space, and then moving on to the next block. In other words, if you have a 64 KB block size (common default size) and 1 sector (512 bytes) reads bad, the other 63.5 KB of the block will be lost and overwritten. 64 KB is 60 emails without attachments, or potentially crucial Internet activity.

Now it comes down to a tradeoff. Smaller block size to lose potentially the least amount of data? Or huge block size to get the fastest image. Again, the default is 64 KB block size, but factors on the scene may dictate otherwise.

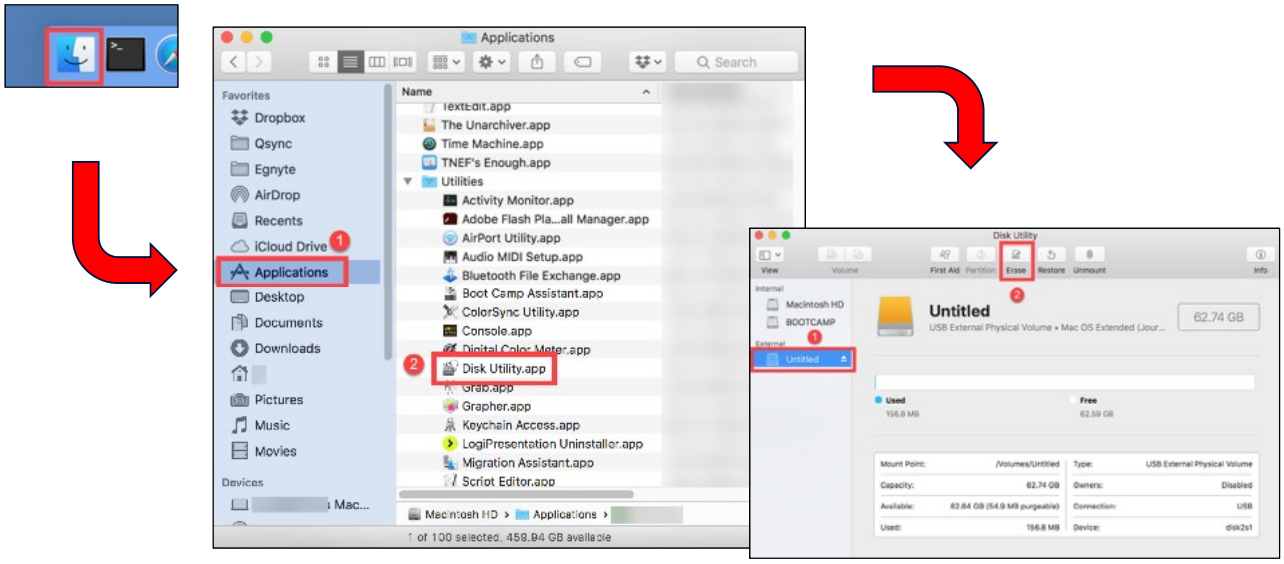
The slide shows an acquisition speed comparison using the same computer and hard drive, with various different destination parameters. The subject machine whose drive was imaged was a MacBook Air with a 128 GB SSD and 4 GB of RAM. The imaging was performed in every case by booting to a MacQuisition USB drive and using MacQuisition to control the imaging process.

Destination media was plugged directly into the subject machine in various ways as outlined in the slide. Hashing was not performed on any images except as indicated to show the time tax due to hashing.

Observations

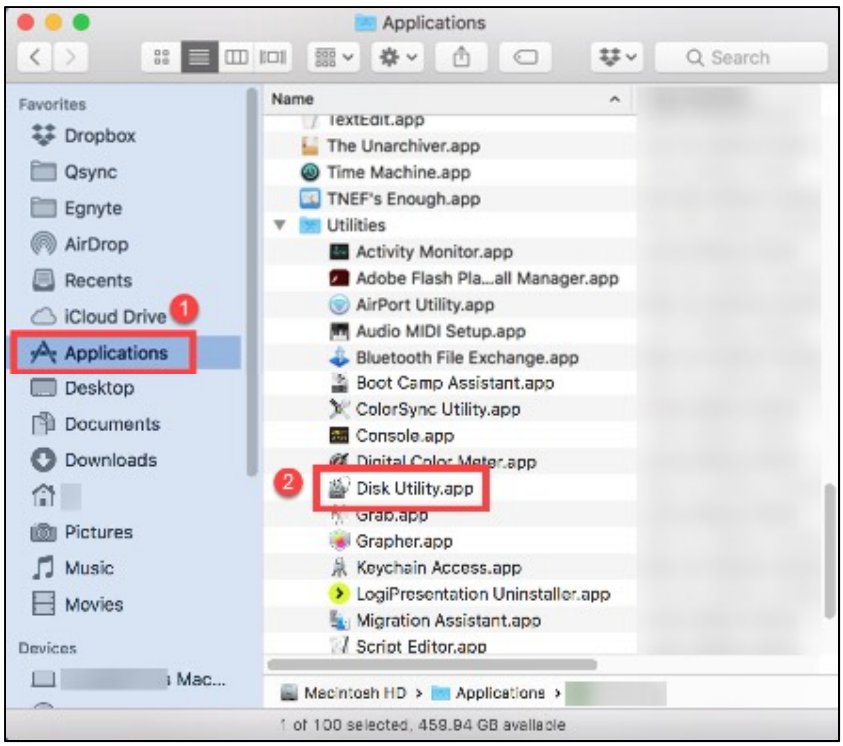
Surprisingly, USB 3 performed better than Thunderbolt. FileVault imaged faster than no encryption. Clearly, block size has considerable impact on acquisition speed.

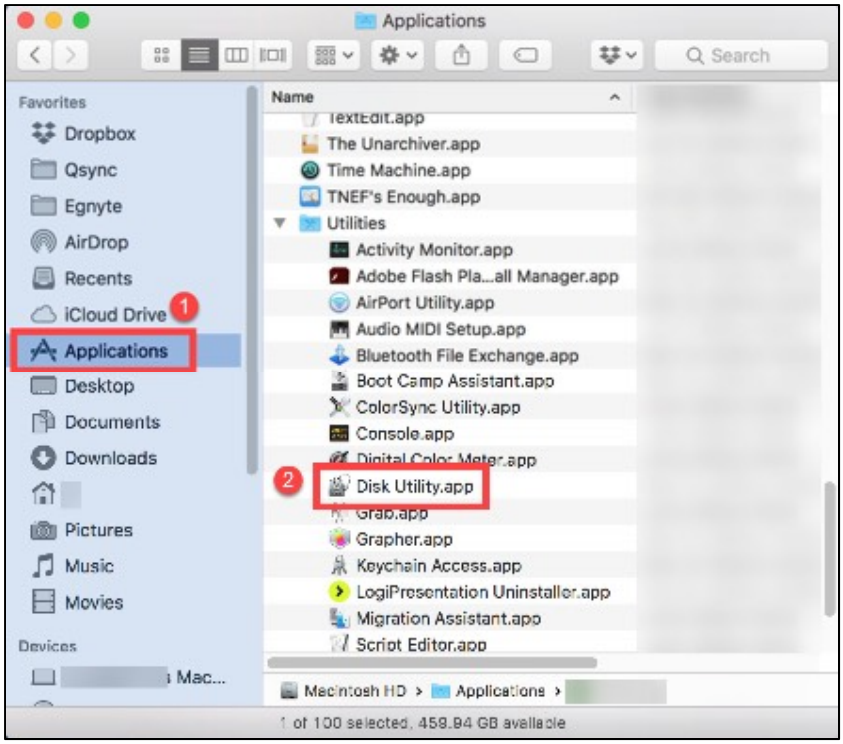
Formatting Destination Drive (1)



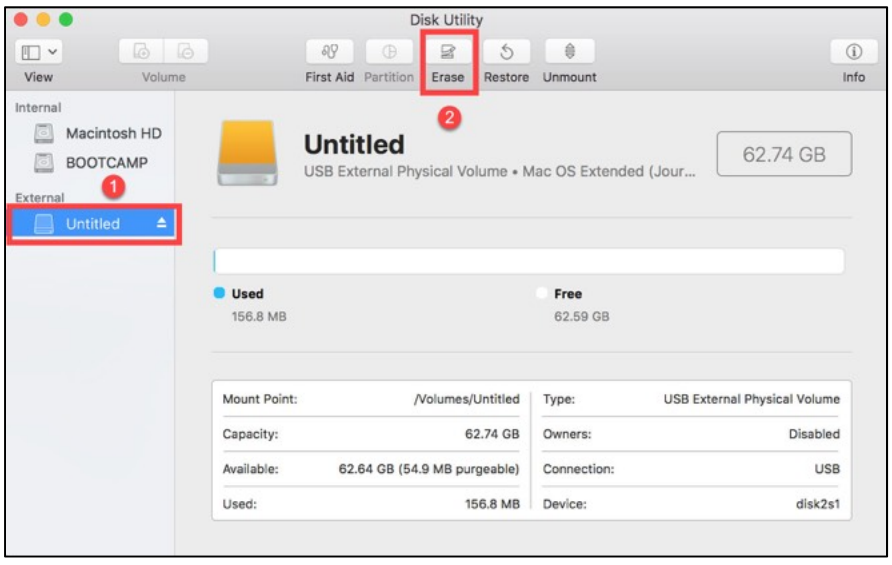
In the case of Mac acquisitions, it is best to have the destination drive formatted to OS X Extended (Journaled) if you are imaging directly from the subject's machine, or if you are imaging using Target Disk Mode. The steps shown here will look slightly different on older operating systems.

Take your destination hard drive of choice and plug it in to your computer. By "your computer", this does not mean the machine you are imaging. It means the forensic examiner's machine.



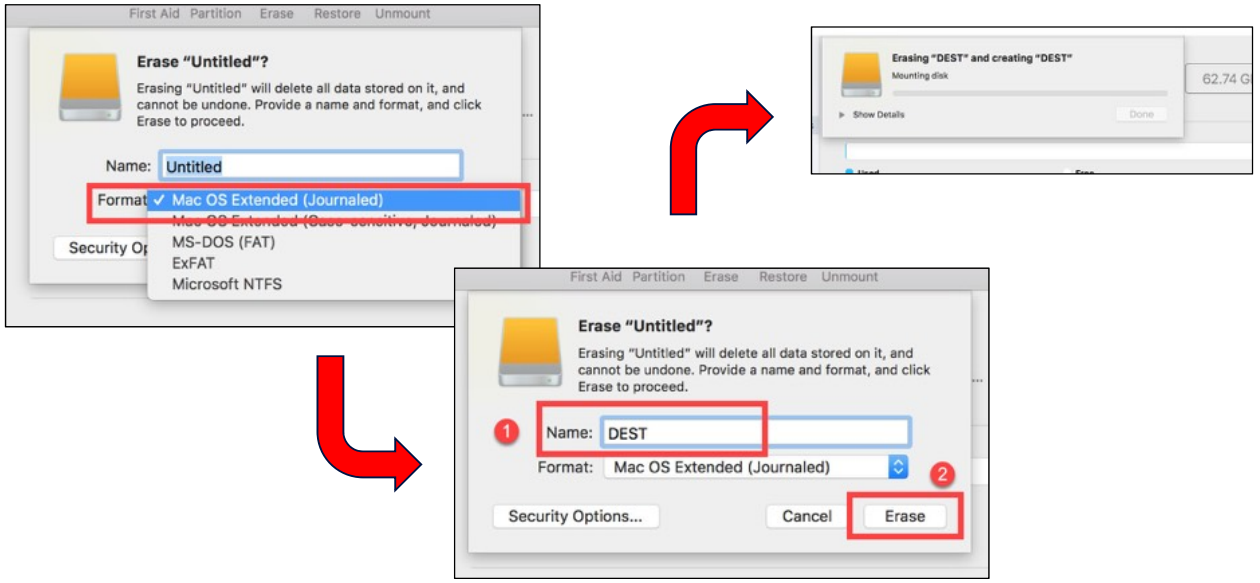


Open a Finder window, and in the left column, select the Applications folder, and in the right pane, expand the Utilities folder and locate the Disk Utility App, as shown below.

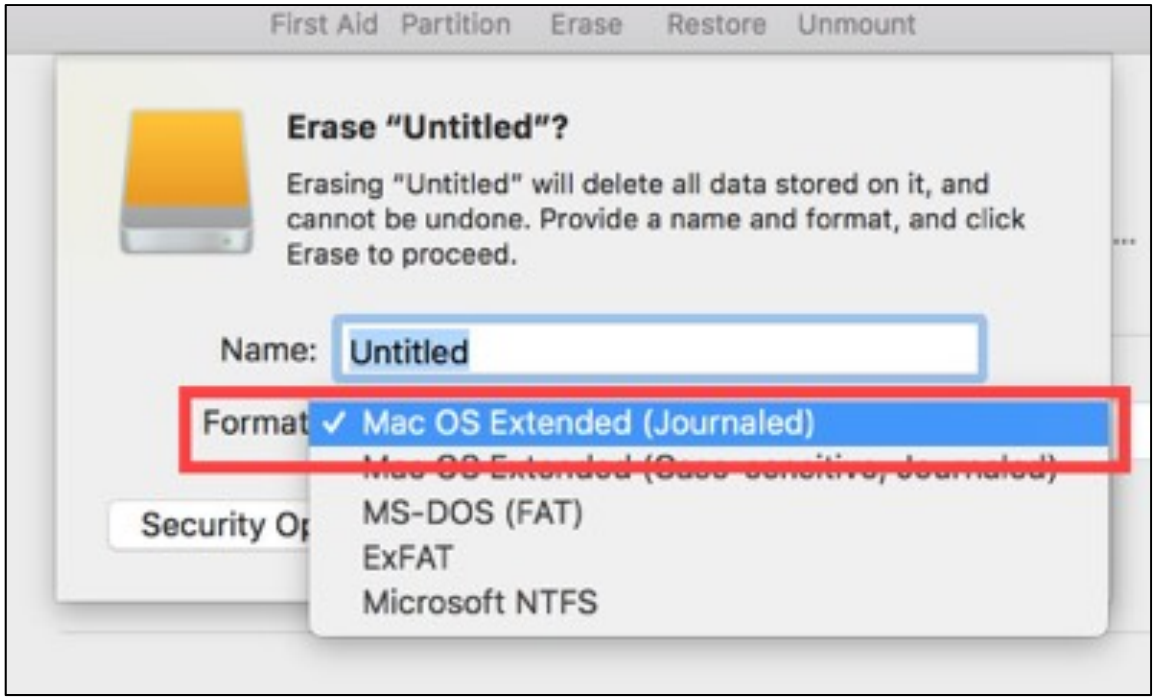


Double click to open the Disk Utility, and you will see the Disk Utility window. As you can see, the physical hard drives appear in the left column. For preparing the destination hard drive, select the physical drive in the left column, then click on the Erase button as indicated.

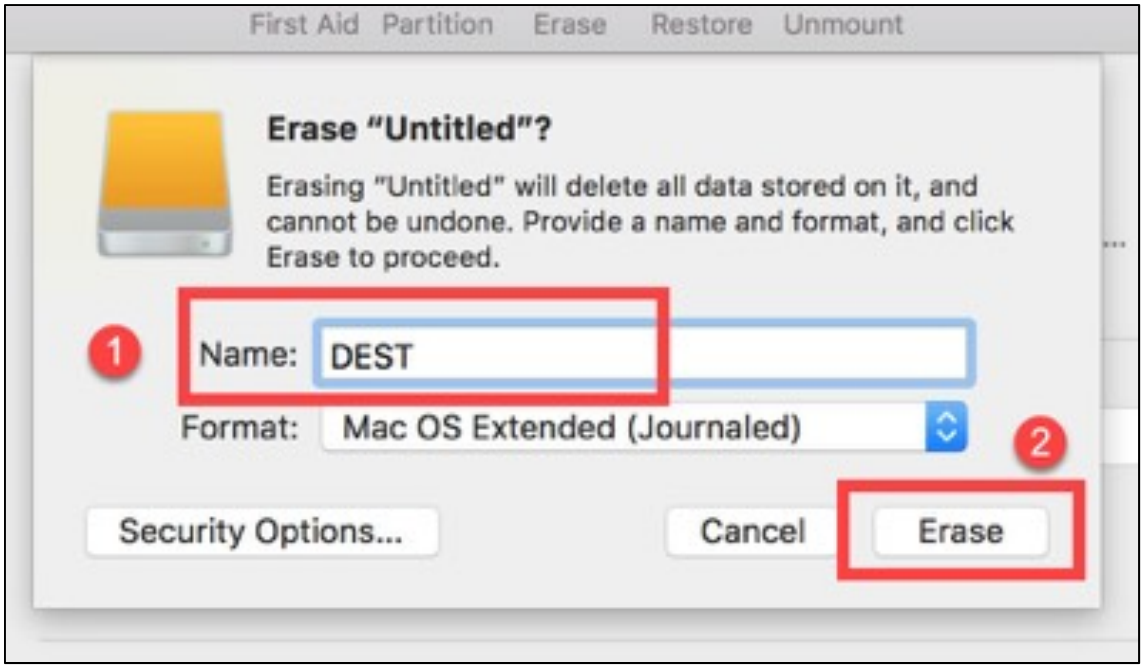
Formatting Destination Drive (2)



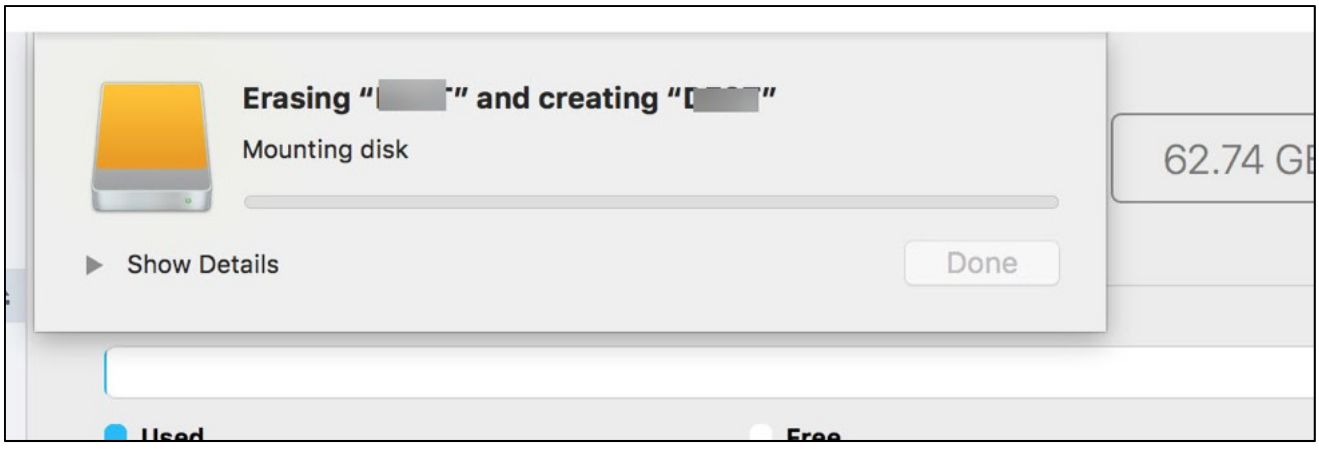
You will now see a box appear with some options. In the **Format** field, select **OS X Extended (Journaled)**, unless you know of a reason to pick something else.



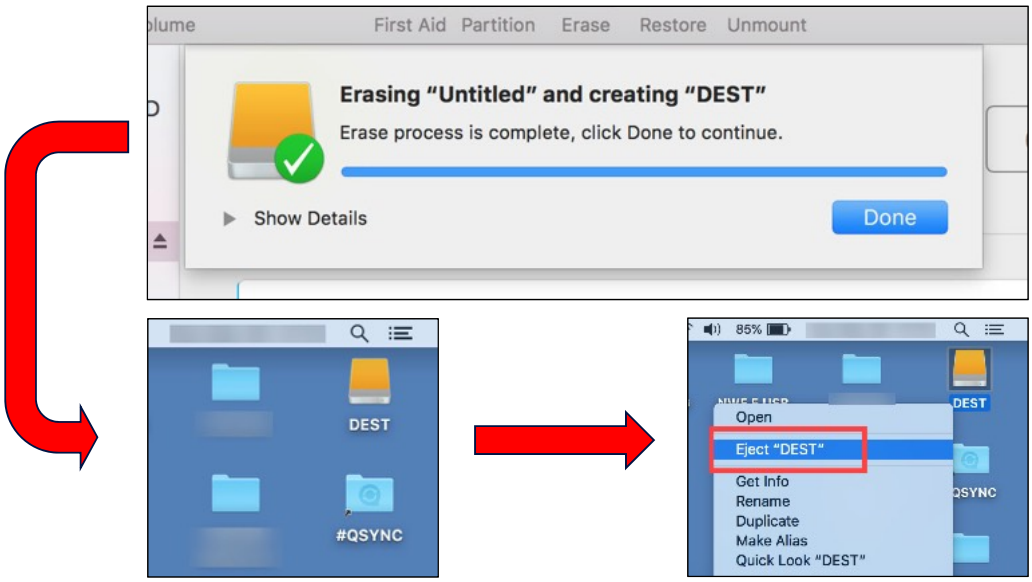
The **Name** field can be anything you wish it to be. For reasons that only experience can teach you, it is preferential to select a short, single word for a name. Now click on **Erase**.



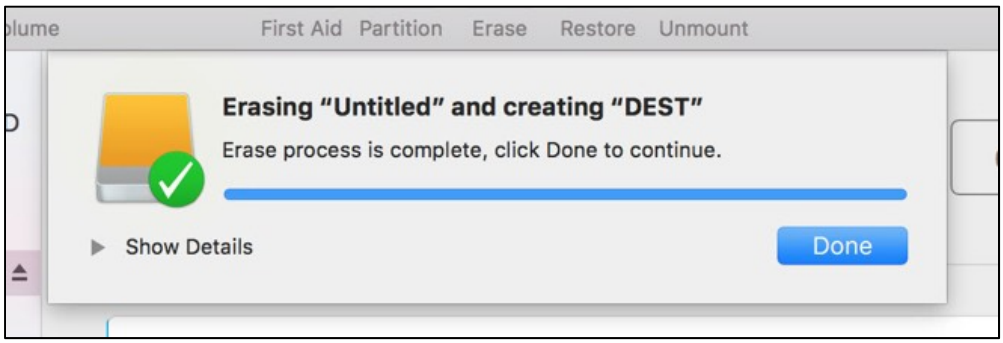
The formatting process will start, and you will see the progress as shown.



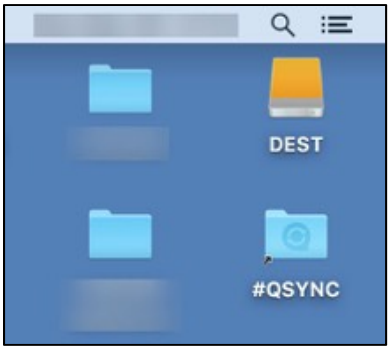
Formatting Destination Drive (3)



After a brief period of time, typically measured in seconds, the formatting will be complete, and you will see the box below.



Now that the formatting is complete, click on **Done**, and you will be back at the original screen, but the new drive name and partition will be visible. You can close the utility above. Back on your Desktop, you should now see an icon indicating the new destination drive, as shown below.

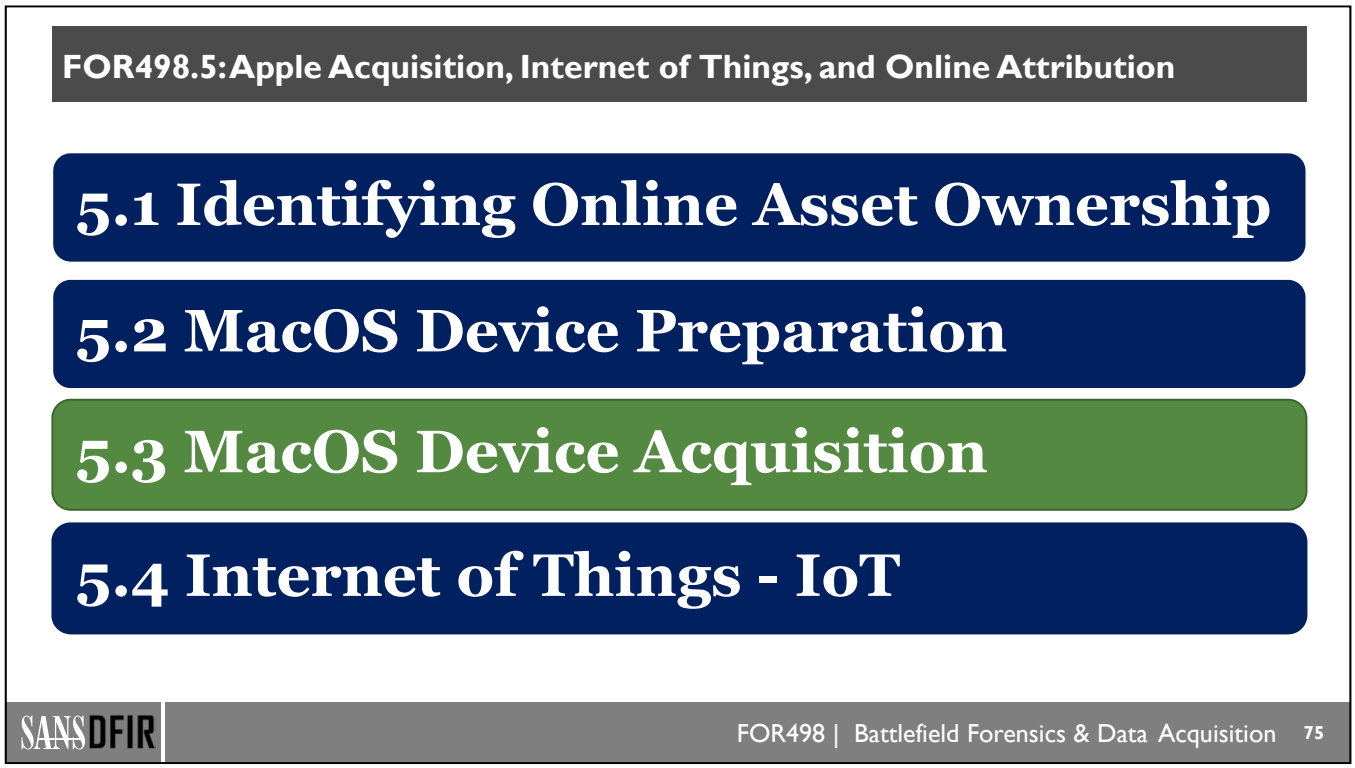


Do not simply unplug the new hard drive, or you risking damaging the partition. You should always right click on the icon, and select **Eject**, and then you can unplug your drive AFTER the icon disappears. Your new drive is now ready.

Summary

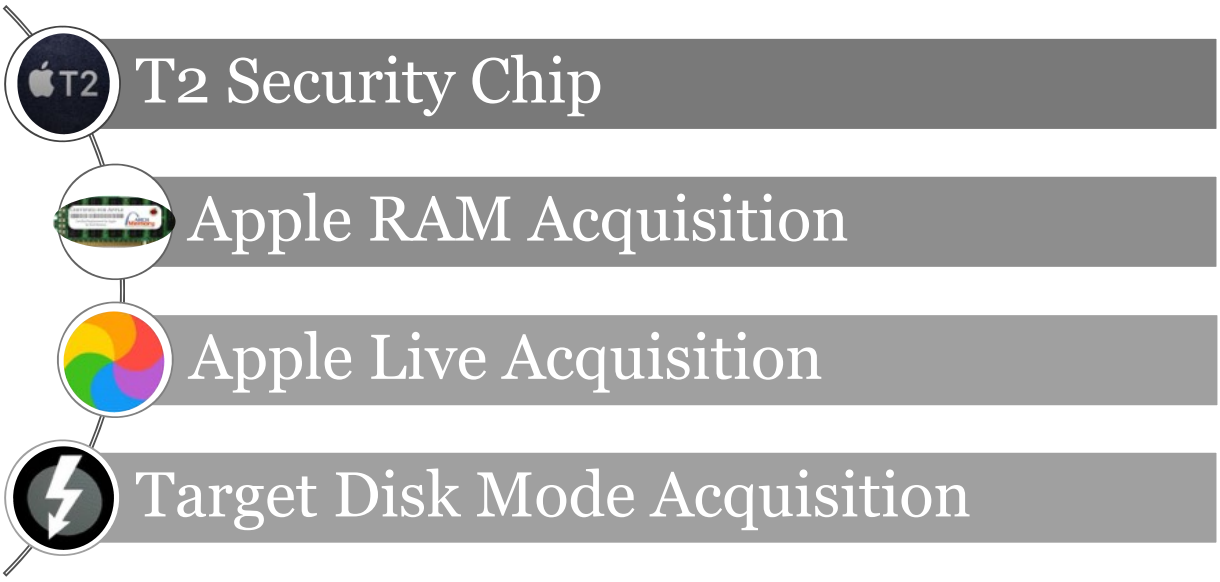
- Apple devices have far more variables in their acquisition approach than anything with the Windows OS
- By far, the most difficult thing to deal with are the different layers and applications of security
- There are a number of different keystroke combinations to access various data points
- System Profiler allows us to collect BIOS style data, since Apple does not have a BIOS in the way we are used to
- Removing the storage media from an Apple device (when it is even possible) is quite a different process than with Windows computers

This page intentionally left blank.



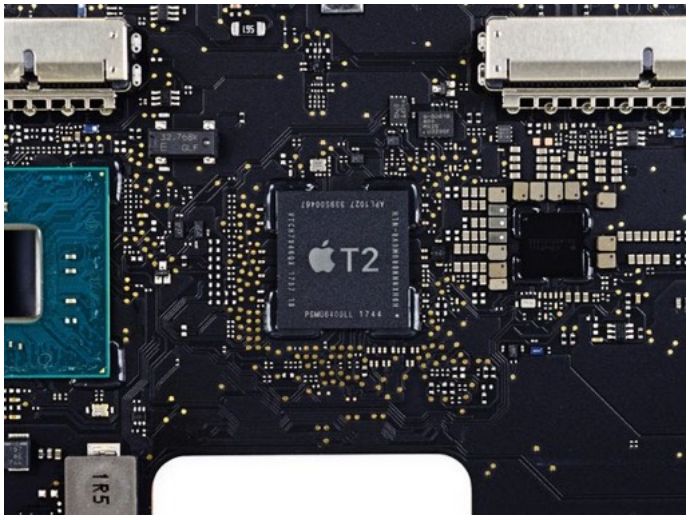
This page intentionally left blank.

MacOS Device Acquisition



This page intentionally left blank.

T2 Security Chip



Apple's commitment to security comes as no surprise to anyone. We have been hearing for years the acrimony between law enforcement and Apple because of Apple's security, and its reluctance to provide any assistance in that regard. The argument rages on with law enforcement saying that Apple should assist them in decrypting systems to “catch the bad guys". Aspersions are then cast when Apple refuses to do so.

In fairness to Apple it is not a situation where they don't want to help. And up until very recently they had the capability to help. The reluctance to assist though, comes when the technology behind their assistance may be compromised. Rules of evidence state that any software, program, tool, or technology that is used in collecting or testing or analyzing evidence must be available or made available to both sides of an action in order to verify an outcome. Apple's position is that if they bypass their own security to extract evidence, and the matter ends up in court, they may be forced through rules of evidence to provide any tool or technology that they may have used, to the team on the other side.

In the past, forensicators had to deal with core storage as well as FileVault. FileVault is Apple’s whole disk encryption solution. The Apple File System (APFS) has created its own set of challenges for examiners as well as the new security layers inherent in Apple's OS Mojave. As of late 2018 a new security dynamic has entered the landscape from Apple. This has come in the form of a new security chip that is present with every Apple computer product. This chip is called the T2 security chip. [1] It works quite similar to the security on Apple portable devices. The T2 security chip contains something called a security enclave. In this security enclave, an encryption key is housed. All encrypted data on the drive gets decrypted with this key and passed to the processor in an unencrypted state. In this manner the only data that is unencrypted is data in use at a given moment.

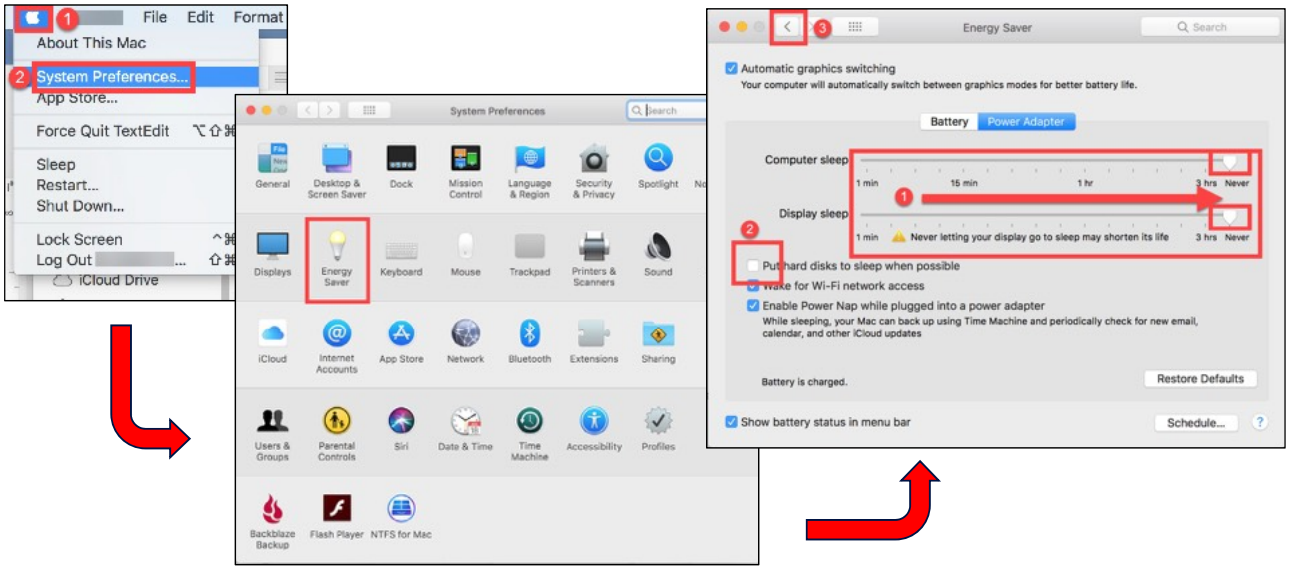
The T2 security chip has a number of security functions that severely limit an examiner’s ability to collect a forensic image from an Apple computer. This applies not only to RAM, but live system acquisitions as well as dead box acquisitions. In previous days, even with security in place, an examiner had options. For example, turning off system integrity protection (SIP) [2] would allow us the ability to collect a forensic image using Target Disk Mode or the command line.

It is extremely important to the forensicator, or at the very least the first responder if they are not the same person, to understand the new security features inherent with any device that contains the T2 security chip. [3] Very much like Apple portable devices, there is now a limit to the amount of passwords someone can attempt while trying to access the device. The device will allow 15 attempts before time delays start to become enforced between attempts. After 30 attempts at trying to login with the user interface or through target disk mode, the device will be locked out from further access through this interface. There are further allowances for recovery attempt through iCloud and FileVault recovery keys, however, after a total of 180 additional attempts, the security enclave will no longer process access attempts and the data on the device will be inaccessible forever.

Also like Apple portable devices, there is no longer the need to completely wipe a hard drive prior to sale or return. Committing a factory reset will merely destroy the encryption key in the T2 security module and generate a new one. Without the old key the data is inaccessible.

[1] Apple T2 Security Chip | `https://for498.com/tk4xe`
[2] System Integrity Protection | `https://for498.com/as4r7`
[3] Apple T2 Security Chip Overview | `https://for498.com/r9hne`

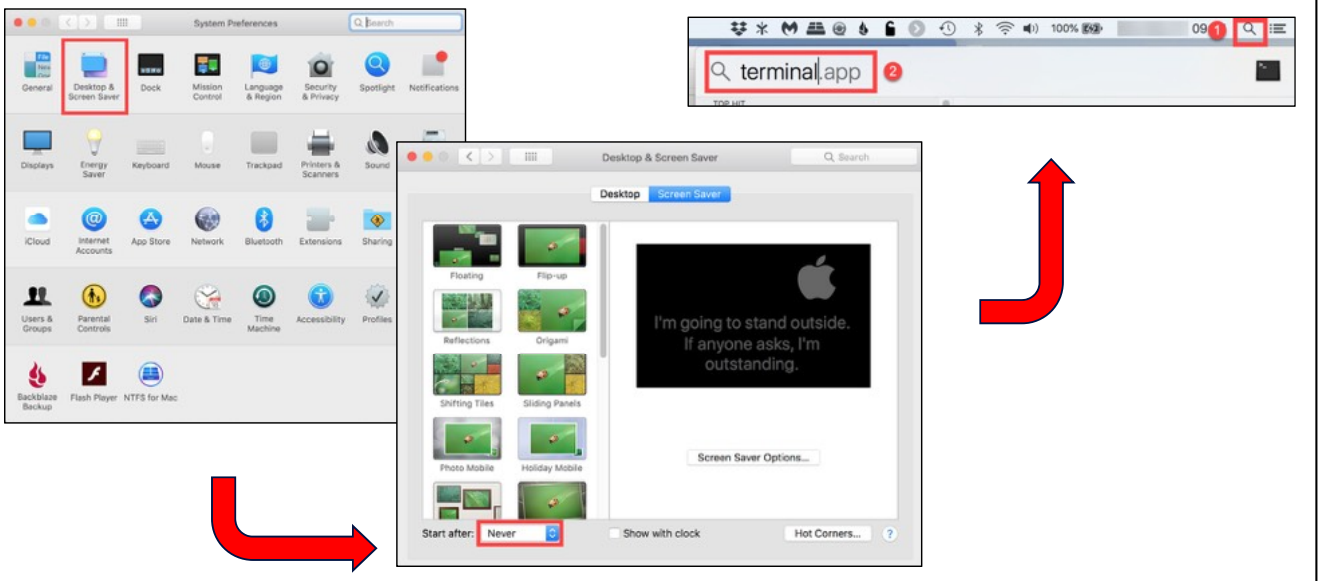
RAM Acquisition Process (1)



There are some necessary steps to perform prior to actually collecting the RAM. It is assumed that you have already done a proper evidentiary collection of the SUBJECT device and prepared an external drive to save the acquired data to. If you are only collecting RAM, anything larger than the RAM dump will be big enough. If you will also be collecting a live acquisition of the Mac computer, you will need an external drive large enough to hold that too.

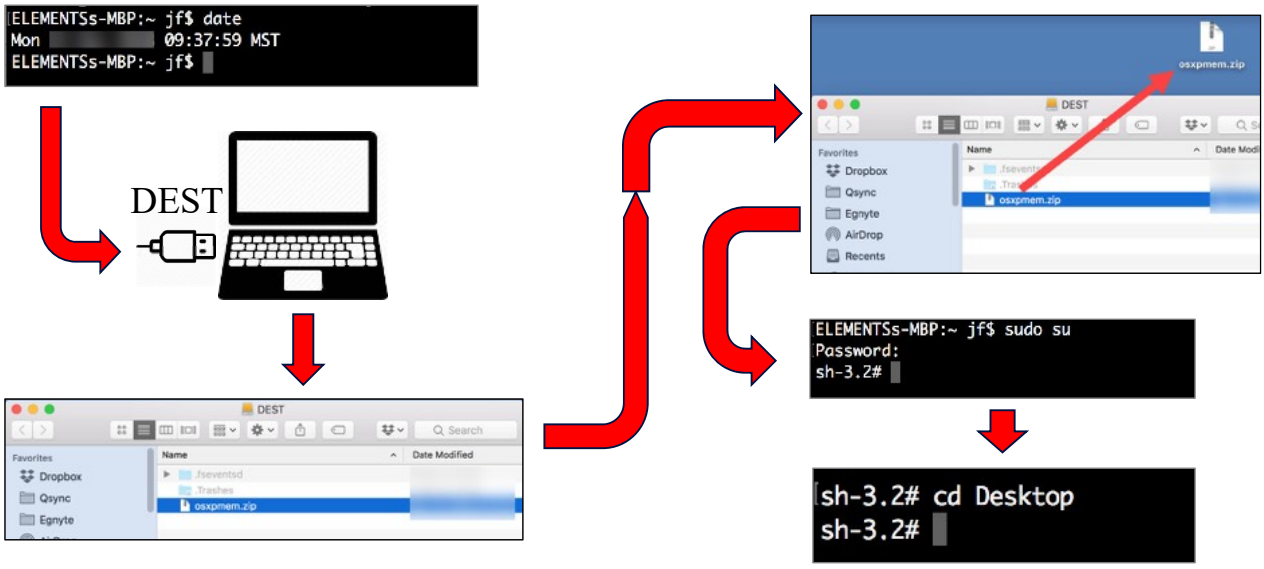
1. Ensure that OSXPmem is saved to the DESTINATION device, but do not plug it into the SUBJECT machine at this time.
2. Let's turn our attention to the SUBJECT computer. Ensure the SUBJECT machine is connected to a power cord. Do not attempt this on battery!
3. From the desktop, click on the Apple symbol in the top left corner, then click on **'System Preferences'**.
4. With the **'System Preferences'** window open, select the **'Energy Saver'** option.
5. You will see two sliders. One is for **'Computer sleep'**, and one is for **'Display sleep'**. Slide them both all the way to the right to **'Never'**.
6. Uncheck the box beside **'Put hard disks to sleep when possible'**.
7. At the top left of the **'Energy Saver'** window, click on the back arrow to return to the **'System Preferences'** window.

RAM Acquisition Process (2)

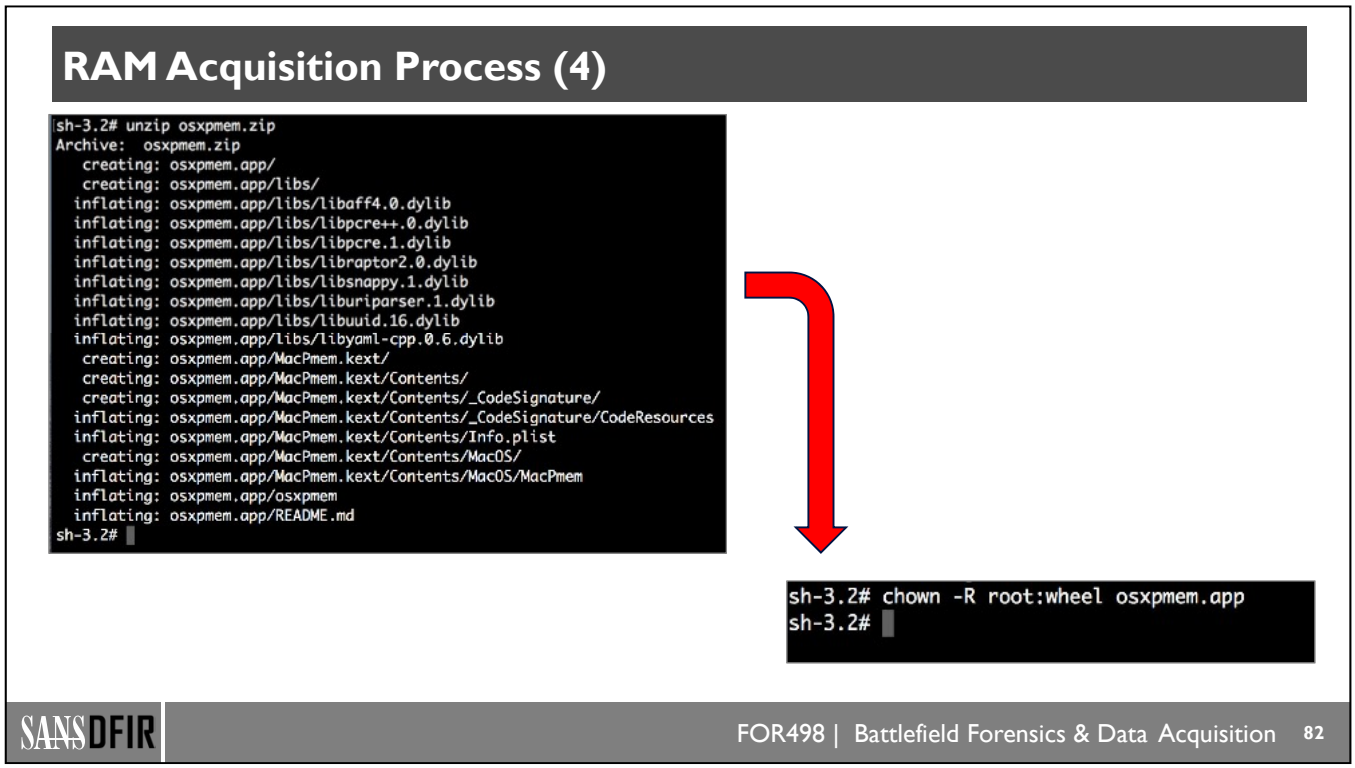


8. Click on the **'Desktop & Screen Saver'** option.
9. On the bottom left of the window, use the dropdown menu beside **'Start after:'** to select the option of **'Never'**, and then you can close the window, returning to the desktop.
10. Access the computer Terminal by clicking on the magnifying glass at the top right of the desktop. When the Search box opens, type **"terminal"** and press **'enter'**. A Terminal window should open.

RAM Acquisition Process (3)

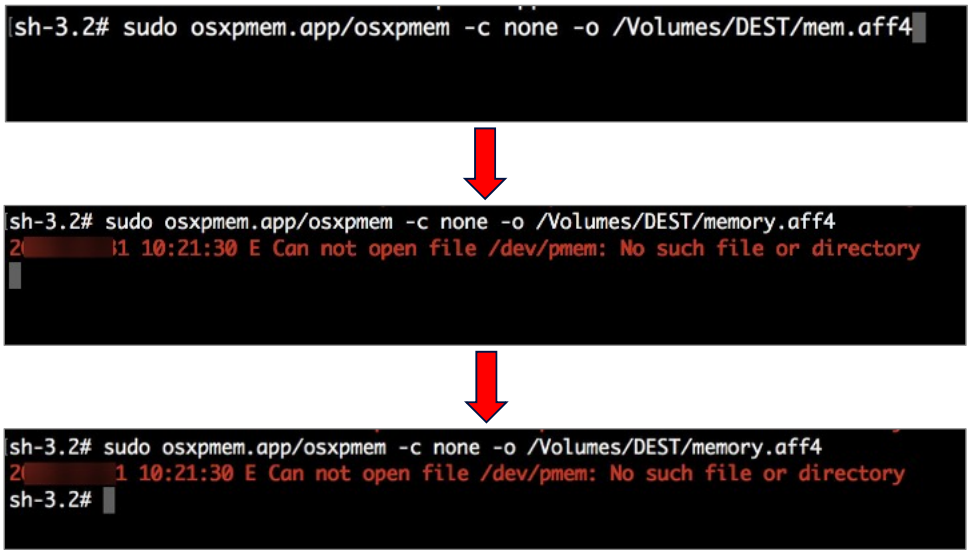


11. Type **"date"** and then press **'enter'**. Take a photo and catalogue this information.
12. Connect the DESTINATION external drive to the SUBJECT machine.
13. Access it and move the program **'osxpmem.zip'** to the desktop of the SUBJECT machine. All of the rest of the steps are assumed to be on the SUBJECT machine.
14. Go back to the Terminal window. When typing the instructions below, only type what is inside the quotes. Don't type the quotes themselves. Anything placed inside { } is a variable that will be determined by you. Don't type the { }.
15. Type **"sudo su"** and press **'Enter'**. This command allows you to go from a restricted user to having root level privilege.
16. You will be prompted for the SUBJECT computer user password. Enter it. If you don't have it, you are done unless you can find it. When entering a password at the command prompt, the cursor does NOT move. Just enter the password and press **'Enter'**. If the password was correct, you will be given a shell command prompt. You now have root level privilege.
17. Type **"cd Desktop"** and press **'Enter'**. Although you will not see a change in the directory, the necessary changes have been made.



18. Type **"`unzip osxpmem.zip`"** and press **'Enter'**. You will see the zip file become uncompressed, and you will be returned to the prompt.
19. Now we need to change the ownership of the osxpmem file so that root has ownership of it. Do this by typing **"`chown -R root:wheel osxpmem.app`"** and pressing **'Enter'**.
20. Type **"`date`"** and then press **'Enter'**. Take a photo and catalogue this information.

RAM Acquisition Process (5)



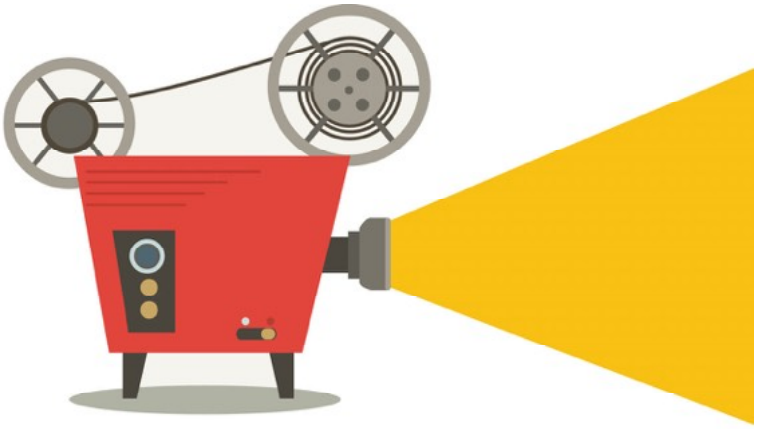
21. Type "**sudo osxpmem.app/osxpmem –c none –o /Volumes/*{name of your DESTINATION}*/*{name you want to call your RAM dump}*.aff4**" and then press **'Enter'**. What we have done is told the osxpmem application to create an image of the RAM. The –c indicates compression level of the RAM we are acquiring, and we have selected **'none'**. The –o indicates the output file and its directory path.
22. You may see an error message as indicated in the slide. You will also note that the cursor is situated on the next line. You can ignore the error message. As long as the cursor is on the next line with no text, work is happening in the background. In this case, the memory image is now writing to your DESTINATION storage.
23. Depending on a number of variables including type and speed of DESTINATION drive, as well as size of RAM, it will take a few to many minutes for the acquisition to complete. Again, speed of DESTINATION media is incredibly important, especially with larger RAM sizes. If the RAM cannot be imaged fast enough, it will cause something called 'smear'. RAM is in use at all times on a computer, including while the acquisition is happening. If the data takes too long to get off of the computer and on to the DESTINATION media, it is changing from one location to another. If it takes too long to get to the DESTINATION media, it will potentially be unusable. In the example above, the imaging process was done from the SUBJECT computer to a SanDisk Extreme USB device, and took 7 minutes for 16 GB of RAM.
24. Once the imaging process is complete, you will see a new command prompt appear. Type "**date**" and then press **'Enter'**. Take a photo and catalogue this information. The reason you have done this is to show that you have not had time to alter data.

RAM Acquisition Process (6)

25. Immediately hash the RAM dump by typing "**md5 /Volumes/{*name of your DESTINATION*}/{*name you called your RAM dump*}.aff4 > {*name you called your RAM dump*}.txt**" and pressing **'Enter'**. Again you will see a cursor at the beginning of a new line with no text. The hash calculation is occurring in the background. In our example, the hashing took 6 minutes.
26. Once the hashing process is complete, you will be returned to a command prompt, and it will appear that nothing has happened. In fact, the command you typed has created a text file on the DESTINATION media with the hash value in it.
27. For the final time in this process, type "**date**" and then press **'Enter'**, take a photo and catalogue this information.
28. Go to the computer desktop and open the DESTINATION media.

If you have reason for a different hashing algorithm than MD5, instead of typing "**md5**", you can type "**shasum**" in its place to get a SHA1 hash, or you can type "**shasum -a 256**" to get a SHA256 hash.

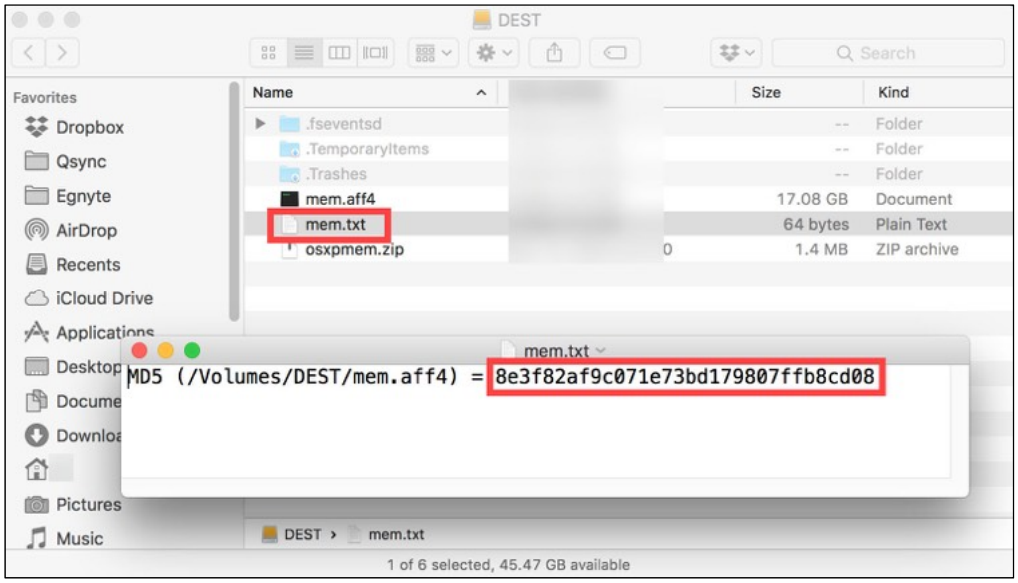
Movie Time!



- 5_1: Apple RAM Acquisition
- 5_2: MacQuisition RAM HFS+ CS Acquisition

This page intentionally left blank.

RAM Acquisition Process (7)



29. On the DESTINATION media, you will see a text file with the same name as your RAM image. When you open the file, you will see the hash of the RAM image.
30. If all you were doing was collecting RAM, you are done. Close the terminal window, eject your DESTINATION drive from the desktop BEFORE you unplug it, and unplug.

dd

- dd – Convert and Copy
  - Program name is dd because cc was already in use by C Compiler
- AKA
  - DataSet Definition
  - Data Definition
  - Disk Dump
  - Data Dump


```
root@      :~# dd if=/dev/sda2 of=/tmp/raw.img bs=65536 conv=noerror,sync
16384+0 records in
16384+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 2.68972 s, 399 MB/s
```

For many years the most ubiquitous tool used in acquisition was a command line tool called dd. [1] dd has been around since before digital forensics had a name. There is no shortage of dialogue on the Internet about what dd stands for, however the most intensive research seems to trace it back to the name Convert and Copy. It would stand to reason then that the initials should be cc and not dd. Apparently though, cc was already being used by a program called C compiler, so dd was the next available double letter set, and was assigned. Many old school Unix/Linux users will say that it stands for Data Definition, and this is also very popular as to the belief of what dd stands for. Dataset Definition Disc Dump, and Data Dump, are also in use. As well, data deletion and data destruction are commonly used, tongue-in-cheek names.

dd is used for an overwhelmingly diverse amount of functions within the Unix/Linux community, although there are many in the forensics community who would believe that dd was designed specifically for creating disk acquisitions. Although dd provides for the best disk acquisition environment outside of true digital forensics, it does have its shortcomings. It does not create hashes, it does not allow for splitting of output, it does not do compression, and by itself does not allow for progression status of the acquisition. It also does not provide any logging other than error logging.

Even with its inherent shortcomings, if applied properly, it can still be used as a verifiable tool. Some would argue that it would be difficult if not impossible to introduce a dd image into court as evidence, however this simply comes back to what we have been saying since the first day of training. As long as you can explain the process and explain why a dd image is acceptable, and as long as you can verify the hash of the image, it should be allowed. It is crucially important that because of the lack of a hashing capability, you run the **date** command immediately prior to and immediately following image creation. This will show that no one would have had time to alter the data on the drive from the moment acquisition started, through to the moment that acquisition was completed. If the examiner then creates a hash of the resulting dd image immediately following the acquisition, it would be extremely difficult for anyone to make an argument to not allow this image as evidence.

There are two other free command line utilities available to the examiner, however they require download and installation on the system. dd on the other hand, is a default component of a Unix/Linux operating system.

[1] dd program | `https://for498.com/bydu5`

dcfldd

- dcfldd – Defense Computer Forensics Laboratory dd
  - Logging
  - Splitting of output
  - Hashing in various formats + verification
  - Progress indicator
- Has not been updated since 2006


```
root@      :~# dcfldd if=/dev/sda2 hash=md5,sha256 hashwindow=10G md5log=md5.txt sha256log=sha256.txt hashconv=after \
> bs=512 conv=noerror,sync split=10G splitformat=aa of=driveimage.dd
2097152 blocks (1024Mb) written.
2097152+0 records in
2097152+0 records out
```

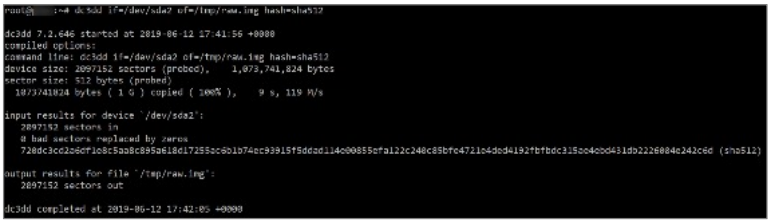
A gentleman by the name of Nicholas Harbour from the Defense Computer Forensics Laboratory saw the shortcomings of dd as a forensics tool and decided to make it better. He developed a tool called dcfldd [1] to address these shortcomings.

It performed essentially the same functions as dd, however he added components that would allow for logging, splitting of output, hashing on the fly, and verification of the image once it was completed. It also has a progress function that allows the examiner the ability to monitor the progress of the acquisition.

[1] dcfldd | `https://for498.com/rhm-c`

dc3dd

- dc3dd – Department of Defense Cyber Crime Center dd
  - Logging
  - Splitting of output
  - Hashing in various formats + verification
  - Progress indicator
- Superior to both dcfldd and dd
- Built on 'patched' version of dd
- Gets updated with dd



dc3dd is a program that was developed by the DoD CyberCrime Center. This program is essentially a patched version of dd, which means that whenever dd gets updated dc3dd also gets updated.

It maintains all the same functionality as dcfldd, however, being that it is constantly updated when dd is updated, there is much less concern with bugs.

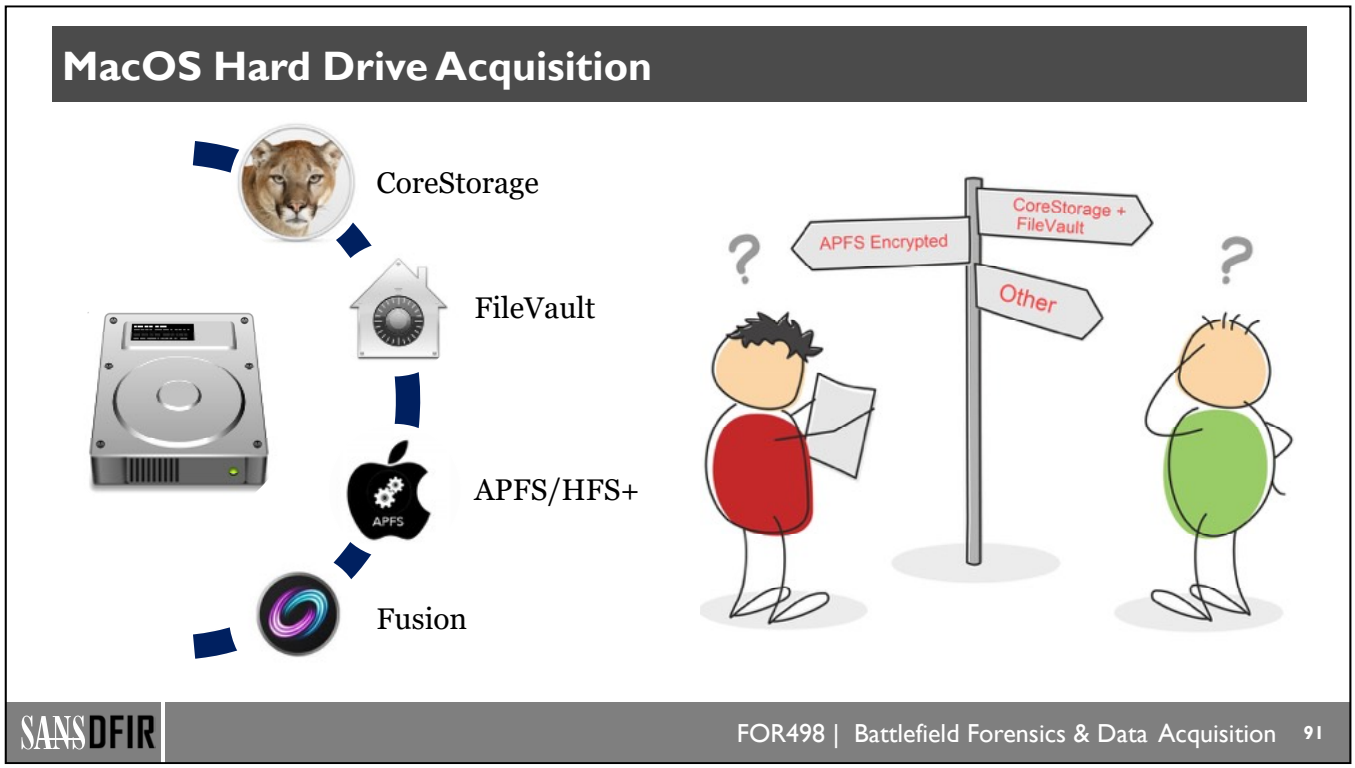
dc3dd does need to be downloaded and installed, unlike dd.

If an examiner chooses to use the command line for their acquisition methodology, it is imperative that they use dc3dd over dd. Although in a prior slide it is noted that from an evidence perspective dd is sufficient, dc3dd will ensure that no protracted explanations and verifications are necessary.

For the purposes of this module we will be using dd and not dc3dd, simply for ease of use.

[1] dc3dd | `https://for498.com/kbvu-`

[2] Excellent comparison and primer on dd, dcfldd, dc3dd | `https://for498.com/gmven`



There are several various different configurations of a MacOS hard drive, and each of them brings its own set of rules and processes for acquisition. A list of just some of the possibilities is listed below:

HFS+
No CoreStorage, no FileVault
CoreStorage, no FileVault
CoreStorage, FileVault

APFS
CoreStorage, FileVault
APFS Encrypted

Add Fusion drive setups, as well as live or dead box acquisition to the above and you can see that things get confusing very fast! Imaging each and every one of the above is different in some way. Sometimes they are small but important differences, and sometimes the differences are considerable. In Windows, we don't have these issues. Live box is done one way, and dead box another.

This module does not get into evidence intake procedures, as these were covered in a previous module. It is assumed that you are already aware of them and will follow them in every case. This module also assumes that you have the necessary credentials to access the device. With FileVault enabled, you cannot extract a usable image without them.

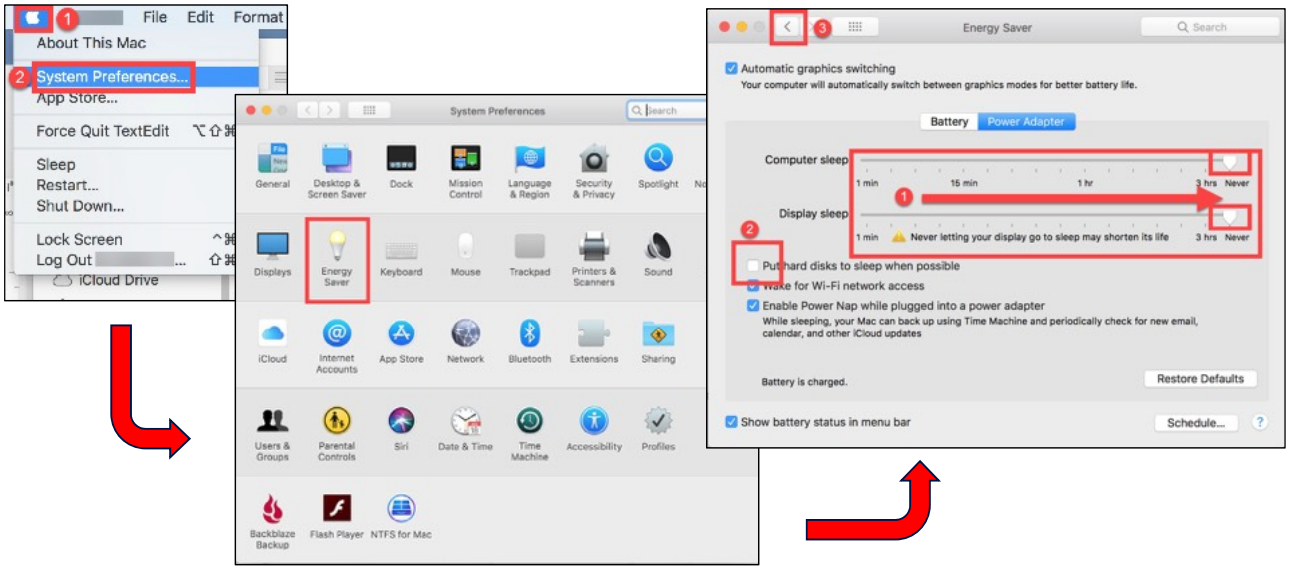
FileVault & CoreStorage
An important distinction to be made is that CoreStorage and FileVault are NOT the same thing. This is a popular misconception. FileVault is the process that encrypts the data on the hard drive. CoreStorage is

a technology that is basically a window into the encrypted data. In other words, CoreStorage allows that the entire drive does not have to be decrypted in order to see data. Only the data in use at the time is actually decrypted on the fly, and viewable through the use of CoreStorage. When CoreStorage exists, it will be seen in Terminal as a separate physical drive, called a virtual disk. This is what you will be imaging. The important takeaway is that Mac computers are shipping with CoreStorage enabled, but FileVault is off. Understanding CoreStorage, FileVault, and the various file systems is beyond the scope of this class, but they are covered in depth in the FOR518 class.

We will not be covering every imaging possibility for MacOS[1], but we will be covering three. We will cover imaging a MacOS HFS+ with FileVault enabled as a live acquisition; we will cover a MacOS HFS+ with FileVault enabled as a dead box acquisition through Target Disk Mode; and we will cover an APFS dead box acquisition using MacQuisition.

[1] Various scenarios (with step by step instruction) for Mac acquisition | `https://for498.com/vuh02`

Live Acquisition Process (1)



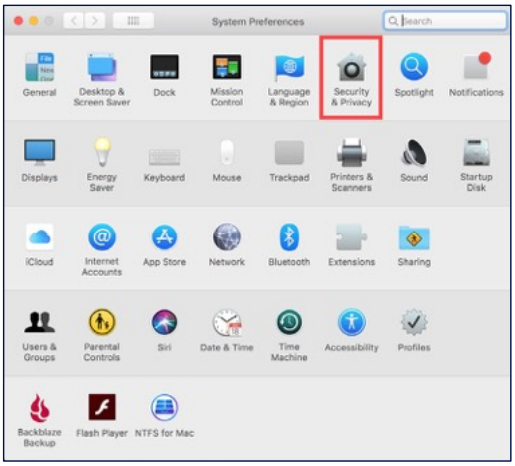
The next set of slides will cover Live Acquisition of an HFS+ system that has FileVault enabled and is running OS X El Capitan. There are some necessary steps to perform prior to actually collecting the image. It is assumed that you have already done a proper evidentiary collection of the SUBJECT device and prepared an external drive to save the acquired data to.

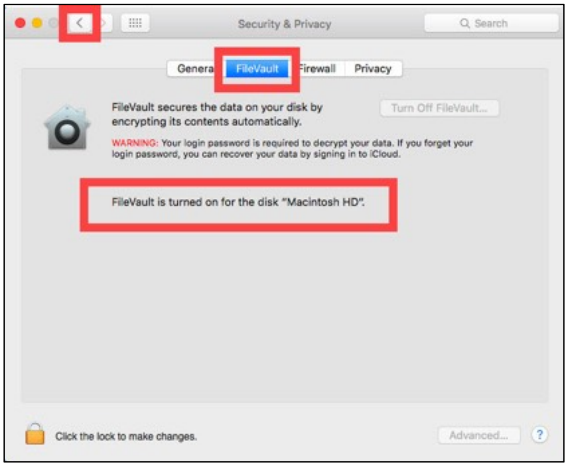
It goes without saying that if you are doing a live acquisition, the computer is ON and LIVE. As such, you need to be extra careful about your processes and steps. The number one rule to live by is RECORD, RECORD, RECORD. Everything you do on the live machine needs to be recorded either via video/pictures,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

Ideally, evidentiary intake is going to make the least amount of changes possible to the data on the drive. To this end, it is important to know where to get system and hardware information directly, rather than hunting around.

1. Ensure the SUBJECT machine is connected to a power cord. Do not attempt this on battery!
2. From the desktop, click on the Apple symbol in the top left corner, then click on **'System Preferences'**.
3. With the **'System Preferences'** window open, select the **'Energy Saver'** option.
4. You will see two sliders. One is for **'Computer sleep'**, and one is for **'Display sleep'**. Slide them both all the way to the right to **'Never'**.
5. Uncheck the box beside **'Put hard disks to sleep when possible'**.
6. At the top left of the **'Energy Saver'** window, click on the back arrow to return to the **'System Preferences'** window.

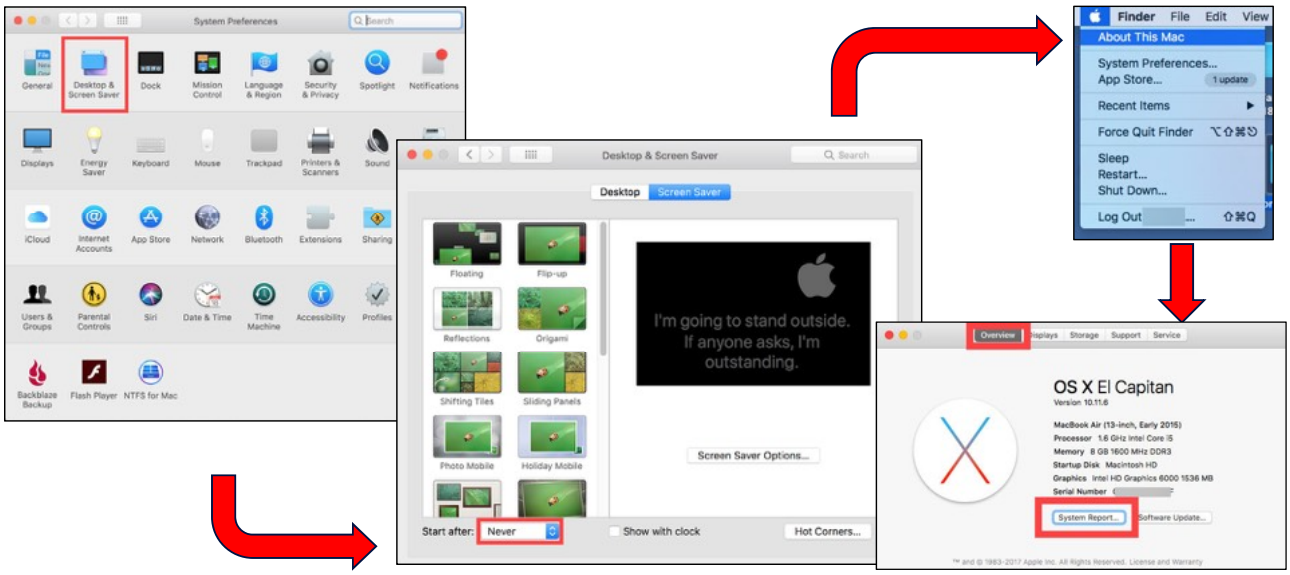
Live Acquisition Process (2)





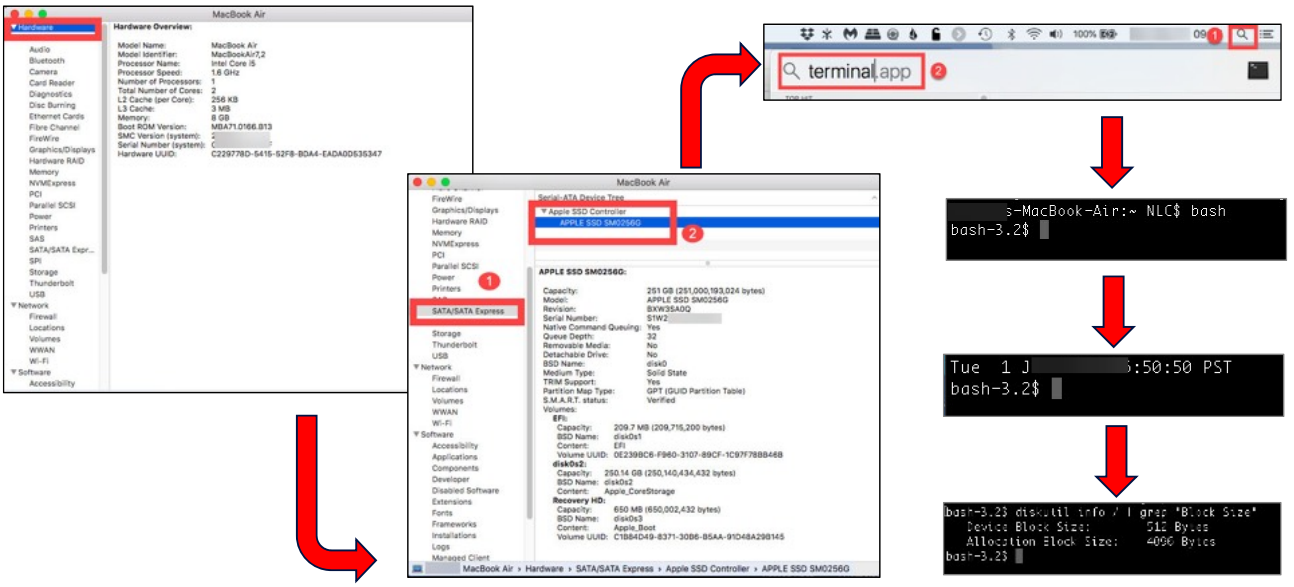
7. Click on the **'Security & Privacy'** option.
8. Ensure that the **'FileVault'** tab is selected.
9. Determine if the message at the bottom says FileVault is turned on or if it is turned off.
10. It is a good idea to photograph this. This information helps determine how an imaging process will proceed.
11. An information screen will appear. Depending on the OS X version, you may see a different name, and certainly the data in your screen will be different than the slide. Take a picture of this.
12. At the top left of the window, click on the back arrow to return to the **'System Preferences'** window.

Live Acquisition Process (3)



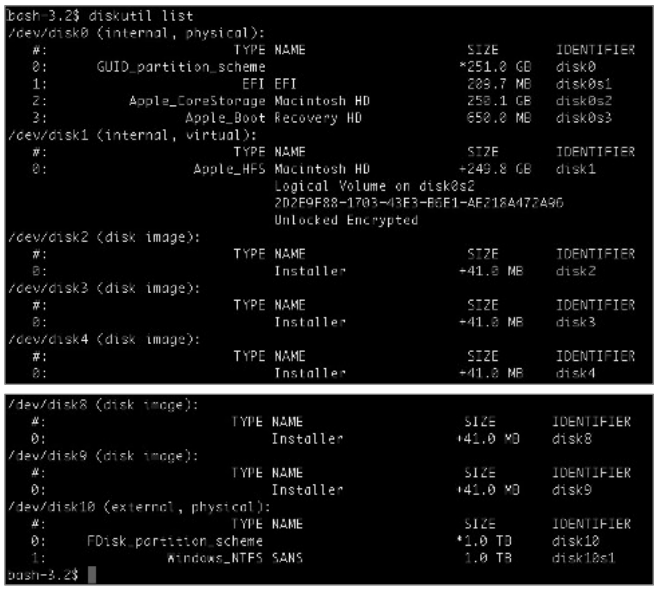
13. Click on the **'Desktop & Screen Saver'** option.
14. On the bottom left of the window, use the drop down menu beside **'Start after:'** to select the option of **'Never'**, and then you can close the window, returning to the desktop.
15. Click on the Apple symbol again at top left of screen, and select **'About This Mac'**.
16. An information screen will appear. Depending on the OS X version, you may see a different name, and certainly the data in your screen will be different than the slide. Take a picture of this.
17. Click on the **'System Report'** button.

Live Acquisition Process (4)



18. In the left column, you see that Hardware is highlighted, which causes the data on the right to appear. Take a photo of this.
19. Now scroll down the column on the left and select SATA. The window on the right will show the physical particulars of the hard drive, along with the logical particulars. Take a photo of this and then you can close the window, returning to the desktop.
20. Access the computer Terminal by clicking on the magnifying glass at the top right of the desktop. When the Search box opens, type **"terminal"** and press **'enter'**. A Terminal window should open.
21. Type **"date"** and then press **'enter'**. Take a photo and catalogue this information.
22. Connect the DESTINATION external drive to the SUBJECT machine.
23. Go back to the Terminal window. When typing the instructions below, only type what is inside the quotes. Don't type the quotes themselves. Anything placed inside { } is a variable that will be determined by you. Don't type the { }.
24. Type **"bash"** and press **'Enter'**. This command allows you to go from a restricted user to having root level privilege.
25. You may be prompted for the SUBJECT computer user password. Enter it. If you don't have it, you are done unless you can find it. When entering a password at the command prompt, the cursor does NOT move. Just enter the password and press **'Enter'**. If the password was correct, you will be given a shell command prompt. You now have root level privilege.
26. Next, type **"diskutil info / | grep "Block Size""**. Yes in this case you need to type quotes around Block Size when you type it in. Case also matters. It must be a capital B and capital S. Device Block Size is akin to Sector Size in the NTFS world, and Allocation Block Size is akin to Cluster Size in the NTFS world. Device block size of 512 bytes will probably be seen on systems using MacOS X El Capitan and older. Anything newer is probably using a Device block size of 4096 bytes. This format was introduced by Apple in 2015. This is not an issue unless you plan on mounting your acquired 'dd' image on another computer. If the block size from the imaged computer and the host computer don't match, the image will not mount.

Live Acquisition Process (5)



27. Type **"diskutil list"** and press **Enter**. This will now give you the list of hard drives, as well as the architecture of each hard drive. It will also list various elements such as if the drive is encrypted or not (CoreStorage), and if there is a Fusion drive in use. You MUST understand this layout in order to identify exactly what to image, and how.
28. In our example, CoreStorage is enabled, FileVault is enabled, and the drive is not a Fusion drive, so we will proceed with the instructions based on that. No encryption, and Fusion drives bring an entirely different dynamic to the playing field and change the way you would image them.
29. In the picture above, we see /dev/disk0, /dev/disk1, and /dev/disk10 at the bottom. We will be ignoring /dev/disk2 thru 9.
30. /dev/disk0 is the physical hard drive in the SUBJECT machine; /dev/disk1 is the virtual CoreStorage drive on the SUBJECT machine; and /dev/disk10 is the DESTINATION hard drive. In the right most column, you see disk0s1, disk0s2, disk0s3, etc. These are the partitions or Volumes of the hard drive. On Macs (and Linux/Unix flavors), they are called Slices, hence the s1, s2, s3 following the disk0. On Slice 2 of disk0, or disk0s2, you can see the words "Apple CoreStorage". When you see this, it does NOT automatically mean FileVault is enabled.
31. The existence of disk1 in the above is the first sign that FileVault is enabled. If it was not enabled, this disk would not exist. You can also see that immediately following /dev/disk1, it states (internal, virtual). Another indication of FileVault being enabled. One last pointer is the bottom of the section under /dev/disk1. It says Unlocked Encrypted. In other words, because FileVault is enabled, it says Encrypted, but because we are logged into the machine with the proper credentials, it is Unlocked. Take note of the physical disks above, as you will need them later in the process.
32. In this demonstration, we are imaging disk0 and saving the image to disk10.
33. Next, type **"date"**, press **'Enter'**, and take a picture. Don't waste any time from this point forward, as any time unaccounted for will be difficult to explain.

Live Acquisition Process (6)

```
bash-3.2$ sudo dd if=/dev/rdisk0 of=/Volumes/SANS/image.dd bs=512k conv=noerror,sync

WARNING: Improper use of the sudo command could lead to data loss
or the deletion of important system files. Please double-check your
typing when using sudo. Type "man sudo" for more information.

To proceed, enter your password, or type Ctrl-C to abort.

Password:
Sorry, try again.
Password:

478744+1 records in
478745+0 records out
251000258560 bytes transferred in 1724.582459 secs (145542625 bytes/sec)
bash-3.2$
```

(Breakdown of instruction is in notes below, and also on next slide)

34. Type the instruction to start the imaging, as seen above. The line to type is **"sudo dd if=/dev/rdisk0 of=/Volumes/{*name of your DESTINATION Volume*}/{*name of your image file*}.dd bs=64k conv=noerror,sync"** In the example above, a block size of 512k was used merely for demonstration purpose.
35. Let's break down what is happening in that line. "sudo" means "Switch User & Do". In other words, run the following command as "Root". "dd" is the name of the program we are using to perform the acquisition. "if=" means "Input File equals". In other words, what are you imaging? This is the file path to the SUBJECT drive. You will note that we have used "rdisk0" instead of "disk0". The instruction of rdisk vs disk as you saw in the previous slide makes the acquisition perform much faster. Anything labelled as disk is actually a buffered device, and the data undergoes extra processing. When we relabel it during acquisition to rdisk, this indicates raw path, and the data does not undergo any extra processing, thus speeding things up by anywhere from 20% to 20X.
36. The next part of the command is "of=", or "Output File equals". This is the file that will be created on the DESTINATION drive, so we have typed the path to the DESTINATION drive, and given our acquisition a name, and .dd on the end. Next is "bs=512k". This is the block size that the program will use as it is imaging. In other words, in this case, it will process the data in 512 kb chunks. Why does this matter? When the chunk of data is being read, if there is a problem with the media, it will just fill the rest of the block with zeros. If the block size is small, you will not have lost much data, but if the block size is large, you may very well lose vast amounts of data that you otherwise would have gotten. So you might think that making it really small is better. Block size will also dictate how long the imaging process will take. The same drive that takes 1 hour with a 64k block size will take a dozen hours or more at 4k. So we need a happy medium. 64k is that happy medium. I have used 512k during the example just for time constraints. You would not use a block size that large except in very specific circumstances.
37. The next part of the command is conv=noerror,sync. This means that if, when reading a block, there is a problem, don't stop the imaging process. Just skip over the rest of the block to the next one and pad the space on the DESTINATION drive with zeros. It is also worth noting that if there are any issues during the imaging process, you will be notified at the end, of any blocks that had problems.

38. Once you type the instruction and press **'Enter'**, you will be prompted for the password of the machine. Enter it here and press **'Enter'**, and the imaging will start. It will look like nothing is happening. Nothing will appear on the next line until the image is complete, at which time you will see something like the lower screenshot.
39. Again, type **`"date"`**, and immediately take a photo. The reason you have done this is to show that you have not had time to alter data in the dump. You can see that the total time for the image is listed above in seconds.

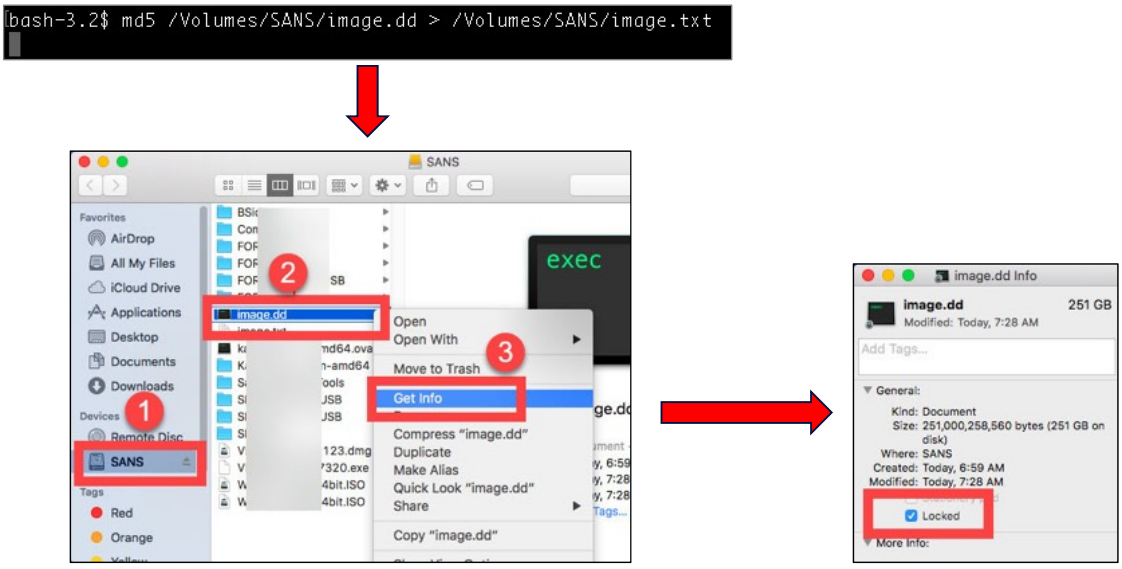
Live Acquisition Process (Imaging Command)

```
[ELEMENTS:~ JF$ sudo dd if=/dev/rdisk5 of=/Volumes/Passport/imagefile.dd bs=2048k
conv=noerror,sync
[Password:
```

- **sudo**: switch user & do (change to root and do the following)
- **dd**: (just the name of the program)
- **if=**: input file equals (what are you acquiring?)
- **of=**: output file equals (where are you putting your output, and what are you calling it?)
- **bs=**: block size (how big you want your data chunks to be)
- **conv=**: convert (when you type noerror and sync, you are telling dd not to stop if it finds an error, and to sync everything to the destination drive. It will do this by filling the error block with zeros, rather than skipping it)

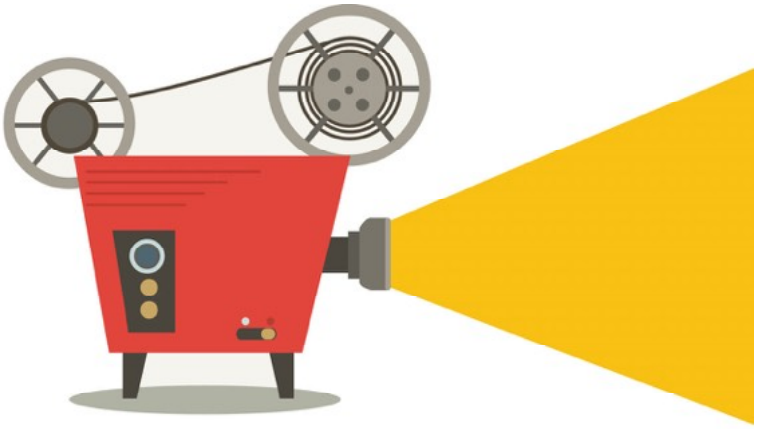
This page intentionally left blank.

Live Acquisition Process (7)



40. Immediately hash the image by typing "**md5 /Volumes/{*name of your DESTINATION*}/{*name you called your image*}.dd > /Volumes/{*name of your DESTINATION*}/{*name you called your image*}.txt**". This process will take some time. In fact, almost as much time as the imaging process itself.
41. Once done, the hash of the file will be available in a text file at the same DESTINATION as your image file.
42. You are now done. Close the terminal window and navigate to your DESTINATION drive.
43. Right click on the .dd file you just created and select **'Get Info'**.
44. In the screen that appears, click in the box beside the word **'Locked'**. This will lock the file and protect from inadvertent writing later.

Movie Time!

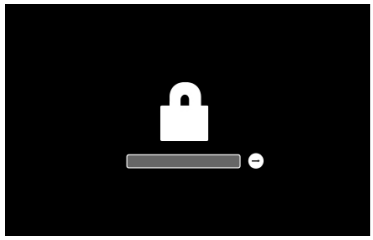


- 5_3: Apple HFS+ Live Acquisition

This page intentionally left blank.

Target Disk Mode Acquisition (1)

Hold down option key and press power key. Continue holding option key until you see one of the two screens. If you see anything different, something has gone wrong.



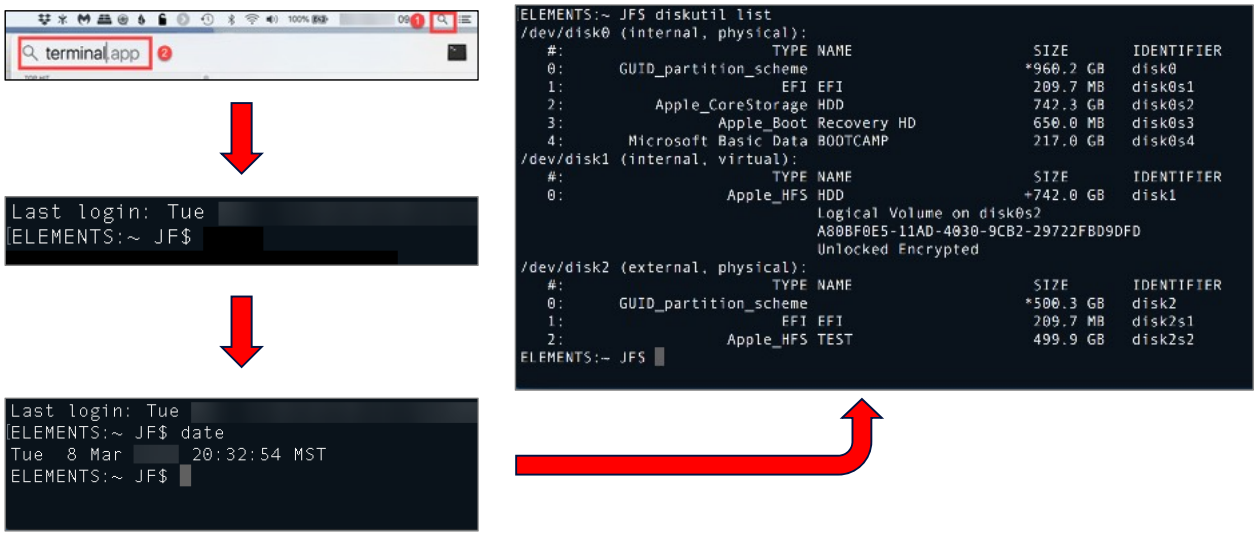
(The names of your drives may be different, and there may be a different number)

The instructions below are designed to create a forensic image of a Mac Computer with FileVault enabled, via the command line and Target Disk Mode. In this case, the computer was found in the 'off' state. It will work on any Intel based Mac. Instructions and screen shots are from El Capitan. Your system may vary slightly. Read all instructions FIRST, before attempting. If, after reading this, there are still things you don't understand, **STOP before you START**. If this is your first time dealing with acquisition of Mac computers, now is not the time to practice on a real case.

These instructions also assume that you have the necessary credentials to access the device. With FileVault enabled, you cannot extract a usable image without them.

1. You must first determine if imaging the drive via this method is even possible. You must check for a number of things in a particular order. a) is there a Firmware Password; b) is FileVault enabled; c) is there a Fusion drive; and d) what is the Block Size. The first two will be determined in early steps below, and the last two will be determined as the imaging process progresses.
2. The first step will be to turn on the SUBJECT machine while holding down the **option** key. Keep holding down the **option** key until you see one of the two screens indicated above. If you see anything other than one of these two (like a login screen), you have done something wrong. Catalogue everything you did leading up to this moment. (The names of your drives may be different, and there may be a different number. This is OK.)
3. The picture on the left is what you will most likely see, and that is a good thing. Now shut the computer off by holding down the power button for 4 seconds. The picture on the right is evidence that a Firmware Password is in use. This is pretty much bulletproof, and without having the password, your interaction with this computer is done, unless you are able to remove the hard drive and image it separately. (If it has FileVault enabled, or a T2 chip, and you have no passwords, this is useless). As before, shut the computer off by holding down the power button for 4 seconds. This Firmware Password is at the hardware level of the computer, and not on the hard drive, so hard drive removal bypasses this.

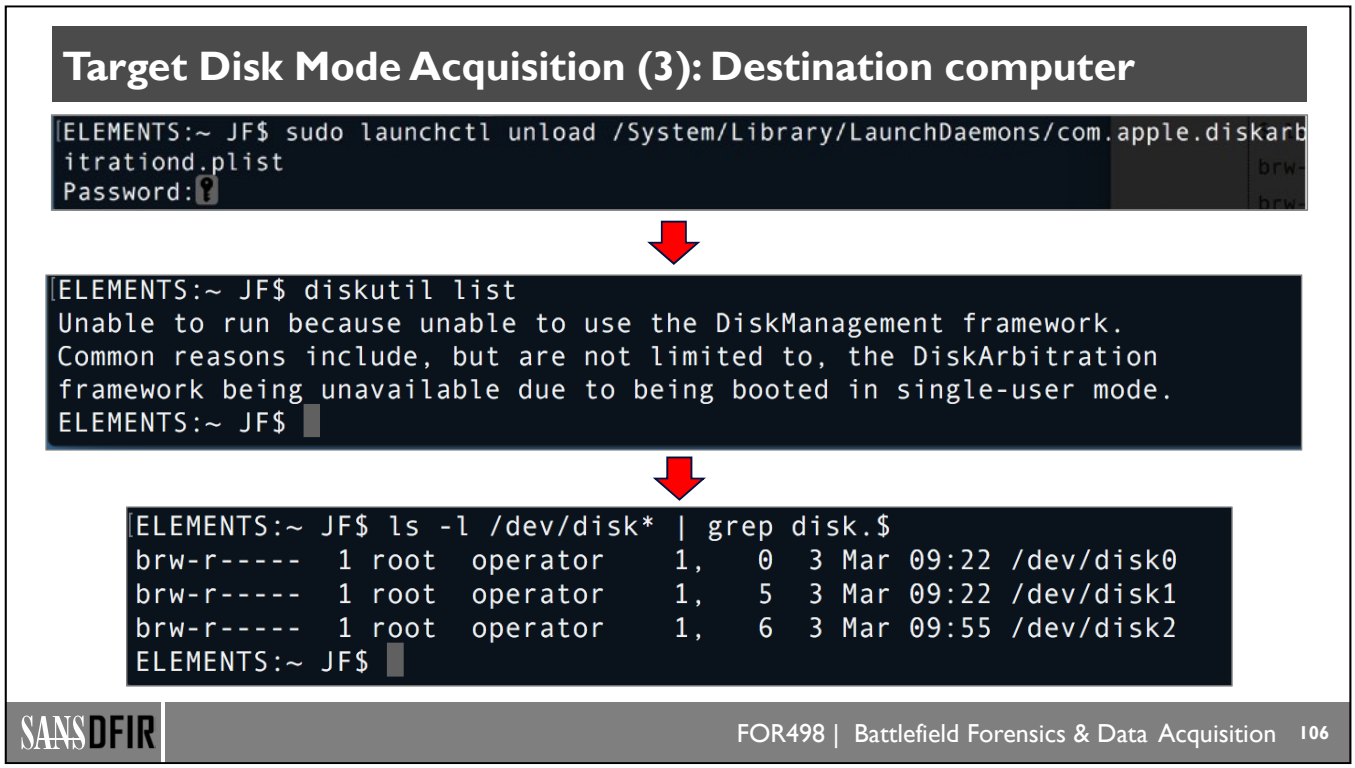
Target Disk Mode Acquisition (2)



4. Proceed under the assumption that you already know the drive you are imaging has FileVault enabled, as determined in steps previously learned.
5. Ensure that you are performing the functions outlined, on the right computer. From this point forward, we will refer to the computers as follows: the computer you are imaging is the SUBJECT computer; the computer that performs the acquisition and where most of your typing will be, is the CONTROL computer, and the external hard drive where the forensic image is going to be written to is the DESTINATION drive or media.
6. First prepare the CONTROL computer. Ensure that it is turned on and that you are logged in to it.
7. Access the computer terminal by clicking on the magnifying glass at the top right of the desktop. When the Search box opens, type **"terminal"** and press **'Enter'**. A Terminal window should open.
8. Go to the Terminal window. When typing the instructions in the following steps, only type what is inside the quotes. Don't type the quotes themselves, unless specifically instructed to. Anything placed inside { } is a variable that will be determined by you. Don't type the { }.

Acquisition

9. Connect the DESTINATION external drive to the CONTROL machine.
10. Unless otherwise stated, the following steps are assumed to be on the CONTROL machine.
11. Type **"`date`"** and press **'Enter'** to get the CONTROL computer system date, time, and time offset, and take a photo.
12. Type **"`diskutil list`"** and press **'Enter'**. This will give you the list of hard drives connected to the CONTROL computer, as well as the architecture of each hard drive. It will also list various elements such as if the drive is encrypted or not, and whether or not CoreStorage is enabled, and if there is a Fusion drive in use. You MUST understand this layout in order to identify exactly what to image, and how. Understand that the above screenshots are from the CONTROL computer. We have not yet involved our SUBJECT computer.
13. In the picture on the slide, we see **/dev/disk0, /dev/disk1**, and **/dev/disk2**. This is quite a simple layout. **/dev/disk0** is the physical hard drive in the CONTROL machine, **/dev/disk1** is the virtual CoreStorage drive on the CONTROL machine, and **/dev/disk2** is the DESTINATION hard drive. In the right most column, you see **disk0s1, disk0s2, disk0s3,** etc. These are the partitions or volumes of a given hard drive. On Macs (and Linux/Unix flavors), they are called slices, hence the s1, s2, s3 following the disk0. On Slice 2 of disk0, or disk0s2, you can see the words **Apple_CoreStorage**. When you see this, it does NOT automatically mean FileVault is enabled. The existence of disk1 in the above is the first sign that FileVault is enabled. If it was not enabled, this disk would not exist. You can also see that immediately following **/dev/disk1**, it states **(internal, virtual)**. Dead giveaway of FileVault. One last pointer is the bottom of the section under **/dev/disk1**. It says **Unlocked Encrypted**. In other words, because FileVault is enabled, it says **Encrypted**, but because the examiner is logged into the machine with the proper credentials, it is **Unlocked**. Remember that this step is on the CONTROL machine. We have not yet attached our SUBJECT computer.
14. Take note of the physical disks displayed, as you will need them later in the process.



15. You need to now disable DiskArbitration on the CONTROL computer. This function mounts the drives connected to it. Obviously, the moment you plug in the SUBJECT computer, the drives will mount, and we have now just written to our SUBJECT drive. This would not be good. Remember that you have already plugged in your DESTINATION drive first. Disabling DiskArbitration only applies to any drives connected after the disabling process.
16. To disable DiskArbitration, type "**`sudo launchctl unload /System/Library/LaunchDaemons/com.apple.diskarbitrationd.plist`**" and press **'Enter'**.
17. You will be prompted for the password on your CONTROL computer. Enter it and press **'Enter'**. Remember that you will not see the characters of your password being entered. You will be returned to a command prompt.
18. Now it is best to test if it worked. Type "**`diskutil list`**" again and press **'Enter'**. If DiskArbitration is off, you will see the message below.

    Unable to run because unable to use the DiskManagement framework. Common reasons include, but are not limited to, the DiskArbitration framework being unavailable due to being booted in single-user mode.

19. With DiskArbitration off, you can no longer see the drive names and volumes. This could make things very confusing if you are not prepared. That is why logging the drive numbers was important. We can use a different tool to at least see what drive numbers we have. So type "**`ls -l /dev/disk* | grep disk.$`**" and press **'Enter'**.
20. The same three physical drives are being shown, even though **/dev/disk1** is not a real physical drive, but a virtual CoreStorage drive that is seen as a physical drive.

Target Disk Mode Acquisition (4): Destination computer

- After plugging in the SUBJECT computer, we can see 3 new /dev/disk items
- /dev/disk3, /dev/disk4, and /dev/disk5 are all from the SUBJECT computer

```
ELEMENTS:~ JF$ ls -l /dev/disk* | grep disk.$
brw-r-----  1 root  operator    1,   0  3 Mar 09:22 /dev/disk0
brw-r-----  1 root  operator    1,   5  3 Mar 09:22 /dev/disk1
brw-r-----  1 root  operator    1,   6  3 Mar 09:55 /dev/disk2
brw-r-----  1 root  operator    1,   9  3 Mar 10:00 /dev/disk3
brw-r-----  1 root  operator    1,  13  3 Mar 10:00 /dev/disk4
brw-r-----  1 root  operator    1,  17  3 Mar 10:00 /dev/disk5
ELEMENTS:~ JF$
```


21. At this point, turn your attention to the SUBJECT machine. You need to place it into Target Disk Mode (TDM).
22. This is done by holding down the **'T'** key and then turning on the SUBJECT computer. Continue holding down the **'T'** key until you see the lightning bolt image, then you can let go. (If the SUBJECT computer has a firmware password set, it will NOT enter TDM, and will go directly to either the firmware password screen, or the user logon screen.)
23. For acquisition in TDM, you have 3 choices. You can use traditional Thunderbolt, FireWire, or USB-C. But of course, both the SUBJECT computer and the CONTROL computer must have the same connection. Most every Apple computer since 2011 has traditional Thunderbolt. Anything prior to that would not have any of the security in today's systems, and thus could be imaged using normal tools after removing the hard drive.
24. Plug the chosen cable into the SUBJECT computer, then plug the other end into the CONTROL computer. You should be prompted on the CONTROL computer, to enter the SUBJECT machine password. This is an indicator that FileVault is enabled on the SUBJECT computer. Without the password, you cannot perform the imaging task.
25. Once you have entered the password, type **"ls –l /dev/disk* | grep disk.$"** again, and press **'Enter'**. You will now see the listing you saw previously, but in addition, you will see the drives from the SUBJECT computer as well.
26. You can see that there are an additional 3 disks now. These are the SUBJECT drive MBR (disk3), the 'physical' drive (disk4), and the CoreStorage, or virtual drive (disk5) being seen. This will always happen in sequential order, so you know that disk3, disk4, and disk5 are the newest additions, or the SUBJECT computer that you just plugged in. You want to image the virtual CoreStorage drive, and NOT the physical drive.
27. Don't forget that Block Size matters, as referenced in the previous process of imaging a system live. You cannot image a system and view it on a system with a different Block Size.

Target Disk Mode Acquisition (5): Destination computer

```
ELEMENTS:~ JF$ hdiutil partition /dev/disk3
scheme:     fdisk
block size: 512
_ ## Type_____________________ Name_____________________ Start___ Size____
+    MBR                     Master Boot Record           0         1
   1 Type EE                                             1   5859999
+    Apple_Free                                      5860000 41020000

+ synthesized
ELEMENTS:~ JF$
```

```
ELEMENTS:~ JF$ hdiutil partition /dev/disk5
scheme:     none
block size: 512
_ ## Type_____________________ Name_____________________ Start___ Size____
+    Apple_HFS               whole disk                   0 1996082176

+ synthesized
ELEMENTS:~ JF$
```


28. Type **"hdiutil partition /dev/disk{?}"**, where the **{?}** is the number of the disk you want information on, and press **Enter**. You will get a warning message asking for your CONTROL computer password.
29. Once you enter it, you will get the information on the screen as shown above.
30. You can see that **/dev/disk3** is the MBR, and not the /dev/disk that we want to image. Let's look at **/dev/disk5** with the same instruction.
31. We see that this is **whole disk**. This is the disk we want to image. We also see **block size**, and in this case, it is 512.
32. An indicator that you may be looking at a 4096 byte block size SUBJECT computer, from a 512 byte block size CONTROL computer, is that when you run the command to see all the /dev/disks, a 4096 byte block size will not show anything other than one extra /dev/disk beyond what was shown before you plugged the SUBJECT computer in. So if you don't see all of the disks that you were expecting, this may be why.
33. It is imperative that you understand that Target Disk Mode will NOT show all the physical drives in the SUBJECT computer. It will only show the Primary drive, and no Secondary drives. In the case of Fusion drives, this means you will not see the small SSD that forms part of the fusion, and you will be left with a useless, incomplete forensic image. You must check the computer physically to determine if there are more drives.
34. Now, type **"date"** press **'Enter'**, and take a picture. Don't waste any time from this point forward, as any time unaccounted for will be difficult to explain.

Target Disk Mode Acquisition (6): Destination computer

```
[ELEMENTS:~ JF$ sudo dd if=/dev/rdisk5 of=/Volumes/Passport/imagefile.dd bs=2048k
conv=noerror,sync
[Password:
```

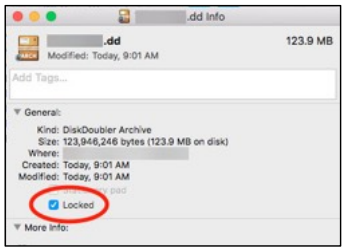
- **sudo**: switch user & do (change to root and do the following)
- **dd**: (just the name of the program)
- **if=**: input file equals (what are you acquiring?)
- **of=**: output file equals (where are you putting your output, and what are you calling it?)
- **bs=**: block size (how big you want your data chunks to be)
- **conv=**: convert (when you type noerror and sync, you are telling dd not to stop if it finds an error, and to sync everything to the destination drive. It will do this by filling the error block with zeros, rather than skipping it)

35. Type the instruction to start the imaging, as seen below and in the slide. The actual line you type will be different based on the variables in the command. The line to type is **"sudo dd if=/dev/rdisk{*drive you are imaging*} of=/Volumes/{*name of your DESTINATION Volume*}/{*name of your image file*}.dd bs=64k conv=noerror,sync"**
36. Let's break down what is happening in that line. "sudo" means "Switch User & Do". In other words, run the following command as "Root". "dd" is the name of the program we are using to perform the acquisition. "if=" means "Input File equals". In other words, what are you imaging? This is the file path to the SUBJECT drive. You will note that we have used "rdisk0" instead of "disk0". The instruction of rdisk vs disk as you saw in the previous slide makes the acquisition perform much faster. Anything labelled as disk is actually a buffered device, and the data undergoes extra processing. When we relabel it during acquisition to rdisk, this indicates raw path, and the data does not undergo any extra processing, thus speeding things up by anywhere from 20% to 20X.
37. The next part of the command is "of=", or "Output File equals". This is the file that will be created on the DESTINATION drive, so we have typed the path to the DESTINATION drive, and given our acquisition a name, and .dd on the end. Next is "bs=64k". This is the block size that the program will use as it is imaging. In other words, in this case, it will process the data in 64 KB chunks. Why does this matter? When the chunk of data is being read, if there is a problem with the media, it will just fill the rest of the block with zeros. If the block size is small, you will not have lost much data, but if the block size is large, you may very well lose vast amounts of data that you otherwise would have gotten. So you might think that making it really small is better. Block size will also dictate how long the imaging process will take. The same drive that takes 1 hour with a 64k block size will take a dozen hours or more at 4k. So we need a happy medium. 64k is that happy medium. I have used 2048k during the example just for time constraints. You would not use a block size that large except in very specific circumstances.
38. The next part of the command is "conv=noerror,sync". This means that if, when reading a block, there is a problem, don't stop the imaging process. Just skip over the rest of the block to the next one and pad the space on the DESTINATION drive with zeros. It is also worth noting that if there are any issues during the imaging process, you will be notified at the end, of any blocks that had problems.
39. Once you type the instruction and press **Enter**, you will be prompted for the password of the CONTROL machine. Enter it here and press **Enter**, and the imaging will start. It will look like nothing is happening. Nothing will appear on the next line until the image is complete.

Target Disk Mode Acquisition (7): Destination computer

```
ELEMENTS:~ JF$ sudo dd if=/dev/rdisk5 of=/Volumes/Passport/imagefile.dd bs=2048k
conv=noerror,sync
Password:
^C40550+0 records in
40549+0 records out
85037416448 bytes transferred in 953.604201 secs (89174750 bytes/sec)
ELEMENTS:~ JF$
```

- ...then date as you did at the beginning of the process
- ...then hash as you did at the end of the live image process
- Then navigate to image file, right click on it, and lock it



40. When the acquisition process is complete, you will see something like the above. Depending on the size of the SUBJECT drive, this could be hours.
41. Again, type "**`date`**", press **Enter**, and immediately take a photo. The reason you have done this is to show that you have not had time to alter data in the image. You can see that the total time for the image is listed in seconds.
42. Immediately hash the image by typing "**`md5 /Volumes/{name of your DESTINATION drive}/{name you called your image}.dd > /Volumes/{name of your DESTINATION drive}/{name you called your image}.txt`**". This process will again take some time. In fact almost as much time as the imaging process itself.
43. Once done, the hash of the file will be available in a text file at the same place as your image file.
44. You are now done. Close the terminal window and navigate to your DESTINATION drive.
45. Right click on the .dd file you just created and select **Get Info**.
46. In the screen that appears, click in the box beside the word **Locked**. This will lock the file and protect from inadvertent writing later.
47. Now power down the CONTROL computer. Because you previously turned off DiskArbitration, you cannot properly eject the DESTINATION drive.
48. Once the CONTROL computer is powered down, unplug the SUBJECT computer, and the DESTINATION drive. Power down the SUBJECT computer by holding down the Power button for 4 seconds. You are now done the acquisition portion, but you are not done with the collection.
49. Remember that because FileVault was enabled, we could not start the SUBJECT computer in Single User Mode to get the information regarding the components in the SUBJECT computer, or the baseline Date/Time.
50. You could perform this task now, depending on circumstances. You would do this by booting the SUBJECT computer live. Once you have recorded this information, you can shut the system down in a normal fashion, and you are done.

APFS Acquisition

- Apple File System
- Generally the same process as for HFS+
- T2 security chip on Apple devices makes command line acquisition impossible
- MacQuisition and SAFE Block To Go software are currently the only methods to create a physical image

Conducting an acquisition on Mac APFS is not significantly different than a normal acquisition. The only thing that changes is that you must collect the actual physical disk as opposed to a synthesized slice.

The most significant issue to affect APFS acquisition is the introduction of the T2 security chip. Acquisition is unsuccessful at the command line, even after disabling System Integrity Protection. Although the acquisition process functions, the resulting image data is not readable.

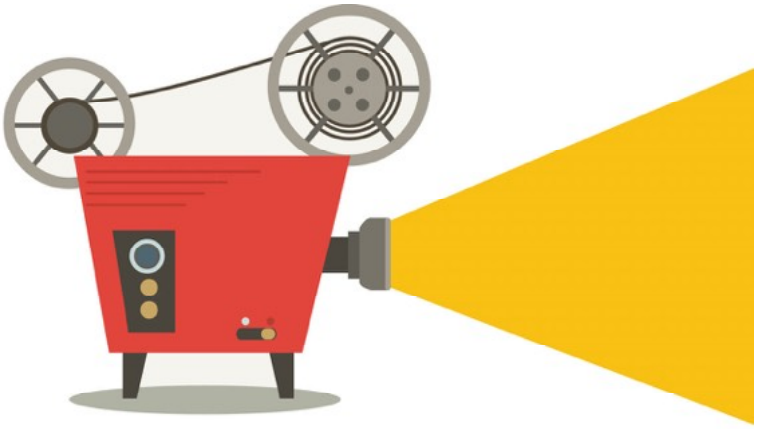
As of this writing the only methods possible to acquire a hard drive in a computer containing the T2 security chip are by using MacQuisition [1] software by BlackBag Technologies, SAFE Block To Go by ForensicSoft, Inc. [2], Recon Imager by Sumuri [3], or Target Disk Mode.

[1] MacQuisition by BlackBag Technologies | `https://for498.com/ytxkb`

[2] SAFE Block To Go by ForensicSoft, Inc. | `https://for498.com/q9z76`

[3] Recon Imager by Sumuri | `https://for498.com/yat73`

Movie Time!



- 5_4: MacQuisition APFS Acquisition
- 5_5a: MacQuisition Live HFS+ CS Acquisition
- 5_5b: Hashing at End of Acquisition
- 5_6: MacQuisition Deadbox Acquisition

This page intentionally left blank.

Summary

- The T2 security chip has radically changed security as we know it
- Just as in Windows, we can (and should) acquire RAM from a Mac
- There are a number of different acquisition methodologies that need to be deployed, dependant on OS, FS, and hardware
- Sometimes the acquisition method of last resort is the TDM

This page intentionally left blank.

5.1 Identifying Online Asset Ownership

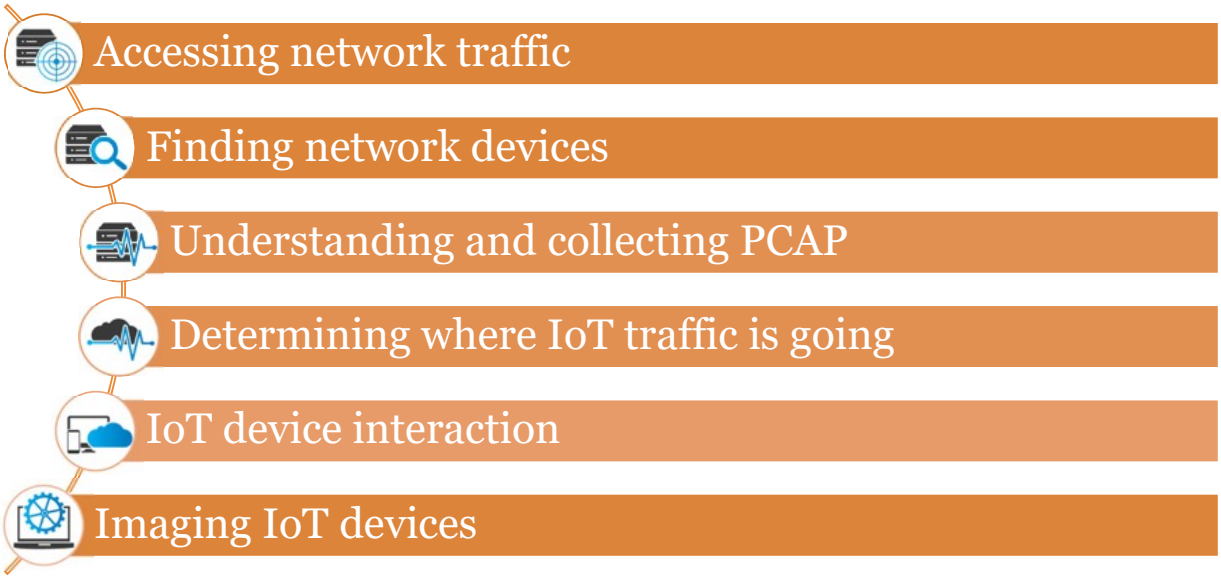
5.2 MacOS Device Preparation

5.3 MacOS Device Acquisition

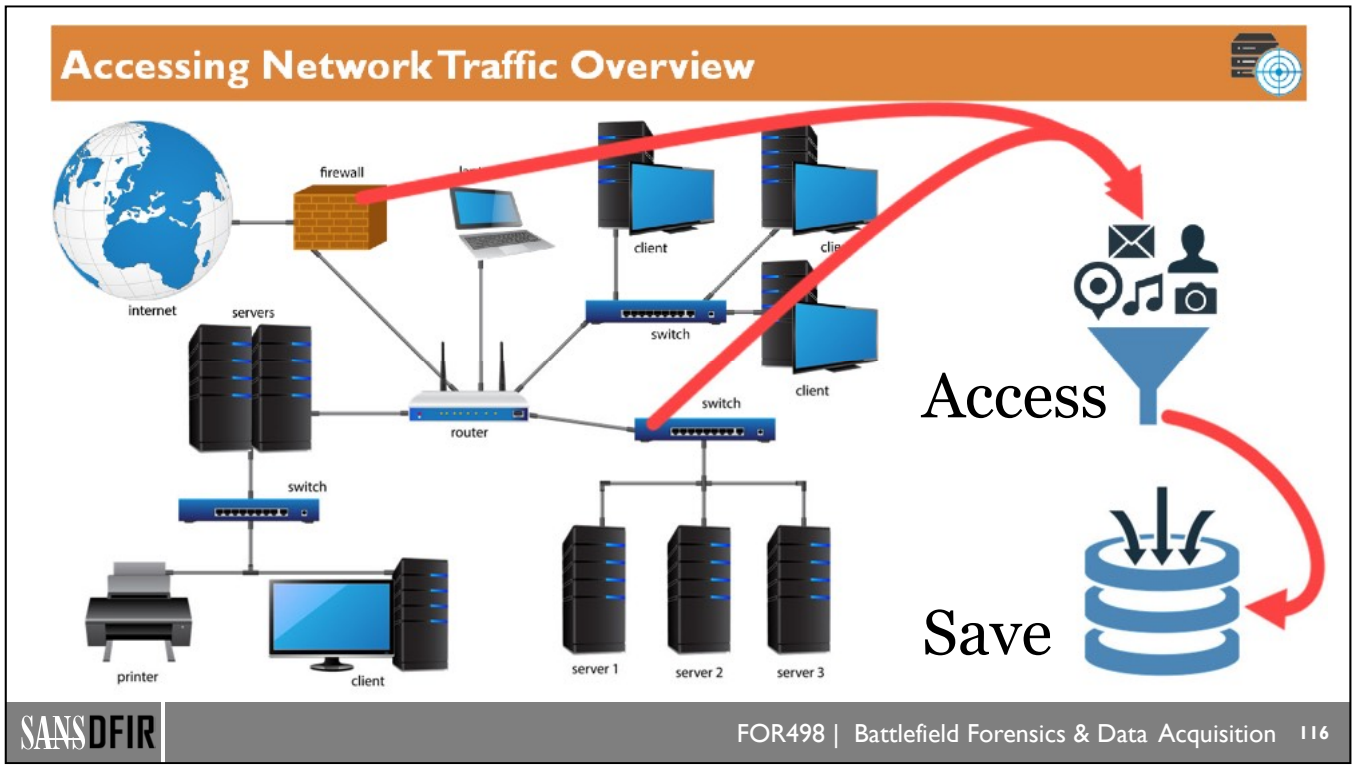
5.4 Internet of Things - IoT

This page intentionally left blank.

Internet of Things



This page intentionally left blank.



Networks range from the very simple to the very complex. In many cases, it will be necessary to interact with a network to answer questions, such as where relevant data is located, which clients hold it, and which clients are communicating with each other. In this section we will discuss how we can go about accessing network traffic and how to save this traffic for later analysis. In the image above we see a typical network arranged in a few different sections. Each section is connected via switches which are all connected to a router. From an overview perspective, we want to access the traffic at some point on the network (which point depends on your investigation), take a copy of data, and save it to a file. These concepts are what we will be unpacking in more detail.

Once we have accessed and collected network traffic, we can then look through the collected network traffic for devices that are communicating with each other. But this begs the question, how do we go about collecting traffic on a network? What does that look like, both from accessing the network to even see the traffic, to saving it?

In our example above we see two scenarios. In the first, we access the network via the switch in the lower right. This would give us access to traffic flowing to and from server 1, server 2, and server 3. In the second, we access the network at the firewall, which means it is possible to get a copy of all data flowing in and out of the network. With our connection point established, we can then save the network traffic to a file.

Let's look at how we save the traffic first. Like many other things, there are standard ways to collect network data, one of which is commonly referred to as "pcap" format. [1] There is also a newer specification, PCAP-ng, which is an extension of the PCAP format, but there is limited support for it in various tools, so it is recommended to always use the PCAP format to save network traffic. This will either be the default format used by network collection tools, or an option will exist that allows the collection tool to use the PCAP format. By using PCAP format, we can be sure the widest range of post collection analytical tools can be used. If you do encounter PCAP-ng formatted collections, tools like tcpdump (to be discussed soon) can convert to PCAP format.

Before looking at full content collection, let's take a moment to discuss other, less intrusive options. One technique to get a more limited set of information about what is transpiring on a network is by using what is known as a "pen register", otherwise known as a trap and trace. This technique does not collect the content of network traffic, but rather, the metadata related to it, like where the communication is going (hostnames for example) and from. This can be thought of as only being able to observe people walking in and out of a room and talking to each other (the "hosts" that are communicating), but not being able to hear what was said (the actual traffic between the two hosts). With that picture in your mind, now imagine that the people are wearing jackets with company logos on them. When someone with a jacket walks into the room from the electric company and starts a conversation, it seems reasonable that the conversation is most likely about electricity in some form or fashion, even though you are not privy to the actual conversation that took place. In much the same way, a pen register would record the fact that a certain host on the network communicated with gmail.com or dropbox.com, which in turn you could deduce that these conversations were related to email and cloud storage, respectively.

Another option when it comes to collection is NetFlow [2]. This can be thought of as more of a summary of the data in that we do not get full packet capture, but rather, source and destination IP addresses, ports, how much data went across a connection, and so on. NetFlow collection results in smaller files when compared with PCAP due to how much less data is stored, but that comes with a price in that we cannot extract out transferred files, look at web pages visited, and so on.

Finally, a pen register is typically something that is authorized via legal proceedings in a criminal investigation. It would require less of a legal burden to use than a full wiretap (i.e. listening to and recording voice communications), which is much more difficult to be given permission to perform. In an internal investigation, or one where you have the consent of the network owners, a pen register and NetFlow would allow for the same kinds of questions to be answered, whereas a full packet capture would be synonymous with a wiretap. While a pen register can be useful, the shortcomings are obvious. What we really want to get is not only an overview of who is talking to whom, but a recording of the conversations that took place. This is what a full packet capture gets us.

With an understanding of how we can save network traffic, we can now look at how we access the network traffic in the first place. We have several options available, but some of the methods we will discuss may not be available depending on the type of hardware the network is based on. Next, we will look at a few ways we can get access to a network to perform a full capture.

[1] Development/LibpcapFileFormat | `https://for498.com/-0xi7`
[2] What is NetFlow? | `https://for498.com/0u6jn`

Accessing Network Traffic: Taps and Mirrors

Port mirroring

- Switch based
- Copy traffic from one port to another
- May not always be possible

Network tap

- Dedicated hardware
- Placed inline between two network segments
- Always works, more reliable

The techniques that can be used to gain access to network traffic include network taps and port mirroring. What makes these two options different though?

In short, port mirroring uses the capability of a switch to copy all the traffic from one switch port to another [1]. As such, no additional hardware (where hardware allows for accessing network traffic, not necessarily recording it) is introduced to the environment. Using a port mirror to collect traffic typically involves a network administrator configuring an unused switch port to receive a copy of all the traffic flowing over another, in-use switch port. For example, if all the traffic flowing in and out of a network is going over switch port one, the switch can be configured to make a copy of all traffic on port one and send it to port 35. Which port number the copy of traffic ends up on does not matter. Once a copy of the traffic is flowing to the additional port, it can be collected to a storage device, such as a laptop or other dedicated piece of recording equipment.

A network tap [2] on the other hand, is a physical device that is placed in between two pieces of networking equipment, such as between a router and a switch, for example. In normal use, a single network cable would go from the router to the switch, but when a tap is introduced, there is a network cable from the router to the tap, and from the tap to the switch. Because of this, installing a tap can lead to a small outage as the connection is reestablished through the tap. Using a network tap should always work, whereas depending on the type of switch and its capabilities, you may not be able to use port mirroring at all. A network tap, in addition to the two ports used to access network traffic, must also have one or more output ports where a copy of the traffic seen is sent. This is the point where a laptop or other device would be connected in order to "see" the traffic and collect it. Aside from the small outage when installing a tap, they are undetectable and do not interfere with traffic going across the tap.

So why would you want to use port mirroring vs. a tap and vice versa? Well, for one, a tap, being an additional piece of hardware, adds cost as one must be purchased. Taps also support a wide range of network speeds and can more reliably capture network traffic than port mirroring can. Mirroring can be thought of as using "best effort" when it comes to providing a copy of the data from one port to another. If the switch gets busy, or too much traffic is generated, the port mirror may not get a copy of everything sent to it. A tap on the

other hand, because it is physically sitting on the wire, always gets a copy of the data. Another concern with port mirroring is aggregating multiple ports to a single port, because the aggregate traffic can exceed the capabilities of the single port.

Later we will look at methods for collecting the traffic to one or more files, which can then be used for analysis.

[1] Understanding Port Mirroring | `https://for498.com/zgeau`

[2] What is Network Tap? | `https://for498.com/zgm13`

Accessing Network Traffic: Legalities/Logistics

Asset coordination

SANS DFIR FOR498 | Battlefield Forensics & Data Acquisition 120

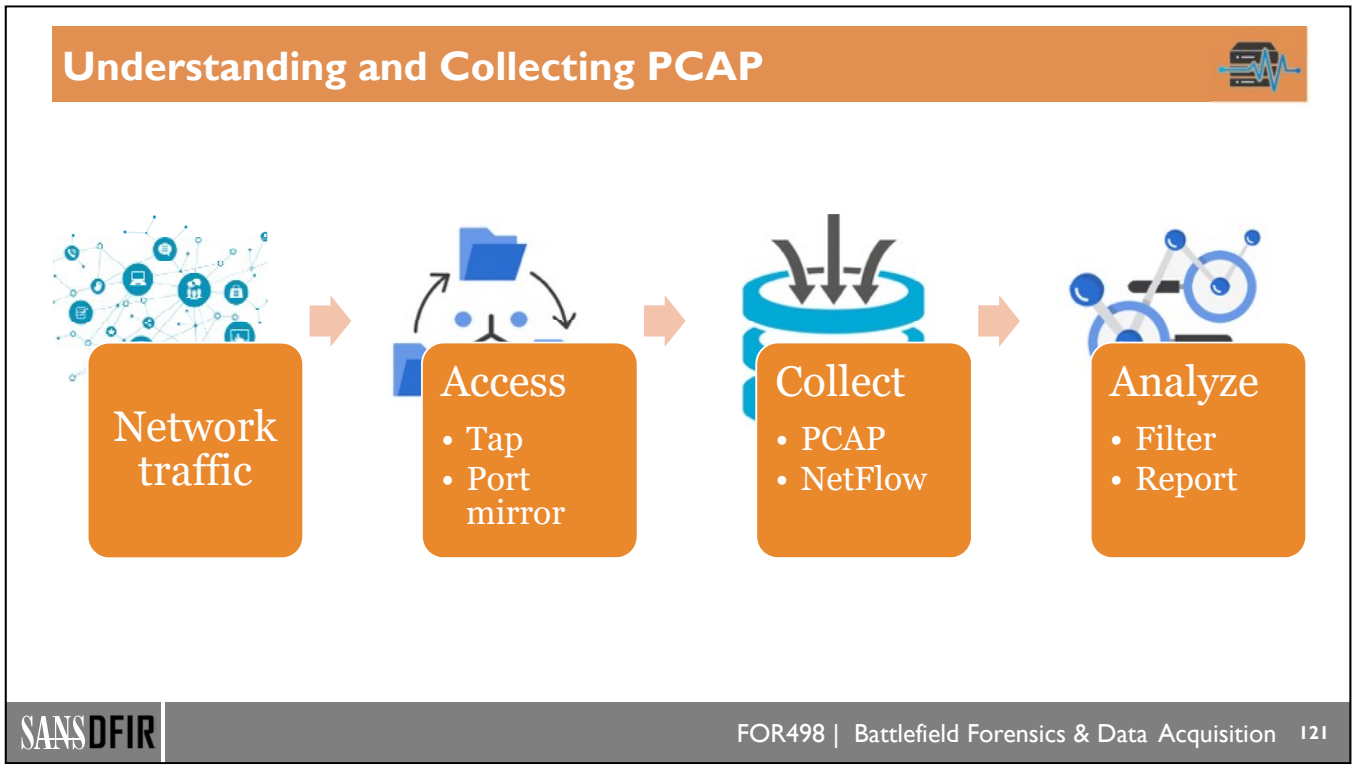
With an understanding of how we can go about accessing and collecting network traffic, we must also be sure we have permission to do so, either in the form of legal process, like a search warrant or Title III, or permission from the owners of a network.

In cases involving legal process, this usually involves an affidavit of some sort, articulating the facts as to why you want to collect network traffic, and the probable cause for doing so. A judge would then review this material and if the legal burden is met, the judge would authorize the collection of the data for a certain amount of time (30 days for example). At the end of the time period, the judge would review an order to extend coverage if collection is needed to continue, to ensure that an overly broad collection does not occur.

On the other hand we have cases where the consent of the network owners is available, and it has been agreed to allow collection to take place. This too usually involves some limiting scope vs "collect all the things" just as we saw with the legal example. How long access remains open is more flexible here as well, as consent can typically be withdrawn as easily as it was given. In some cases you may be a third party operating on someone else's network, and in other cases you may be operating on your own network.

Regardless of how you found yourself needing to monitor a network, be sure to have proper documentation as it relates to intercepting network traffic. Failure to do so is, in many cases, a felony and carries a hefty penalty for unauthorized interception of communication. When in doubt, get counsel involved to be sure you are covered legally and always have permission in writing before beginning monitoring and collection of data.

Other possible pain points relate to logistics issues where the physical network is in one place (or many!) while network administrators are in another. It may take a good deal of logistical coordination to get monitoring in place and ensure successful collection happens, so measure twice, and monitor once!



Before looking at techniques to analyze network traffic, it is helpful to understand what the overall process looks like at a higher level. The first step is of course locating a network whose traffic you want to collect. The next step is using one of the techniques we saw earlier to gain access to the network, like port mirroring or using a network tap. With access in place, traffic is collected to PCAP files. Once the collection is finished, the PCAP files are analyzed to generate leads, answer questions, and so on. This generally involves filtering for data that can help move your case forward such as web sites visited, file transfers, DNS lookups, email, and so on.

Focusing on the analysis piece a bit closer, once PCAP has been collected (or otherwise available, should you run across an organization that has full PCAP available by keeping it around for a period of time), we need a way to look at the contents in order to extract the information it contains.

There are many tools out there that understand PCAP files and can extract a wide range of information from them, but we will spend our time looking at two easy to use tools: Wireshark, a GUI program, and Tshark, a CLI program. These tools allow for filtering based on IP address, protocol, and ports, as well as searching for certain strings. By leveraging the tool's capabilities when it comes to drilling down into the PCAP data, a lot of the noise is removed which allows you to focus on what is important to your case. With that said, what is important to you now may not be in another case, which is one reason we like having PCAP files available. In much the same way we can go back to a full disk image and reprocess certain files, when we have PCAP we can go back and reanalyze it for different protocols or connections as our investigation progresses.

While tools like Wireshark CAN collect PCAP data, it is recommended to never use it for such purposes. As we saw earlier, the preferred tool for collecting PCAP data is tcpdump, and this is the tool to use for network collection because of its speed, flexibility, and stability in collecting raw network data efficiently. On Windows, Wireshark installs additional tools that can be used to capture network traffic from the command

line. The dumpcap tool is generally equivalent to tcpdump. Usage is available via **`dumpcap -h`**. Wireshark can make use of either Npcap, a more modern Windows packet capture library, [1] or WinPcap, which it will offer to install as a part of the Wireshark installation process.

[1] Npcap: Windows Packet Capture Library & Driver | https://for498.com/wd96x

PCAP Tools: tcpdump

```
root@siftworkstation:~# tcpdump -D
1.eth0
2.bluetooth0 (Bluetooth adapter number 0)
3.any (Pseudo-device that captures on all interfaces)
4.lo
root@siftworkstation:~# tcpdump -i eth0 -w /tmp/test.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
^C4770 packets captured
4770 packets received by filter
0 packets dropped by kernel
```

Capture a specific interface to a specific file

tcpdump is the de facto tool for collecting network traffic. It is available on most platforms but may have a different executable name on Windows (wincap.exe). Once installed, usage is relatively straightforward for simple things. As with most powerful tools, there are many options available that allow for filtering traffic before collection, writing out packets to a file (vs. outputting information to the console), limiting capture to only certain interfaces, etc. See the man page for full details for all available options, examples, and more [1].

In the above example, the **`-D`** option is used to get a listing of all available network interfaces. Notice that four interfaces were returned. When referring to an interface, you can use the number of the name of the device, and **tcpdump** will monitor only that device for packets. The second command above uses the **`-i`** switch along with **eth0** to limit the traffic captured to that which is using the **eth0** interface. Finally, the **`-w`** option is used to capture all packets to a PCAP file. This allows for analyzing the PCAP file once the capture is complete using either GUI or other command line tools (including **tcpdump** itself via the **`-r`** switch).

There are a vast amount of references online related to **tcpdump** usage that can show you how you can find HTTP, DNS, or FTP traffic, user agent strings, and more[2]. As you get more familiar with common use cases and specific options, you will of course be able to craft your own commands from scratch. Until then, leverage the expertise of others to learn by example.

[1] Man page of tcpdump | `https://for498.com/sin2v`

[2] tcpdump Examples: 50 Practical Recipes for Everyday Tasks | `https://for498.com/1q6up`

PCAP Tools: tcpdump Basic Options

-D: List interfaces
-i: Interface to monitor
-w: capture to named file
<filter>: Rules for limiting traffic collected

(Many more options available. See man page for more)

```
tcpdump src 192.168.1.25 -i eth0 -w /tmp/test.pcap
```

For its most basic usage, we want to identify which interface to capture. This can be a wired ethernet connection or a wireless network interface. Depending on your case, and where the traffic of interest is flowing, you may have to choose one interface vs. another. We saw this option in use in the previous slide, along with **`-w`** which lets us specify a file to save the captured packets to for later analysis. The other option shown above, <filter> is really a series of things that can be combined to tell **tcpdump** (and similar programs) exactly which traffic to collect based on host, port, protocol, connection state, and so on.

Verb	Meaning	Example
host	Show traffic coming in or out of host	host 192.168.1.52
src	Show only traffic originating from source address	src 192.168.1.158
dst	Like src, but only traffic destined for an address	dst 192.168.1.158
port	Packets must be using the provided port	port 80
portrange	Like port, but for a range of ports	portrange 80-88
http	See only http traffic	http

These can be combined too, so something like **`dst port 8080`** would capture traffic headed to a destination when the port being connected to is 8080.

You can also combine filter criteria using Boolean expressions ('and' or '&&', 'or' or '||', etc.) to do things like **`src 192.168.1.52 and dst port 8080`** for example. Leveraging filters can greatly simplify investigations when you are already aware of some hosts, ports, or protocols of interest, because by capturing less data, you will have less data to analyze down the road.

Just keep in mind that by pre-filtering out certain traffic it will not be available later. For this reason, we generally recommend capturing all traffic up front and then filtering it out in the analysis stage. This ensures you have all the data you may need depending on which way your investigation goes.

A cheat sheet for **tcpdump** can be downloaded from `https://for498.com/ce7yl`. This contains the most commonly used flags, filter criteria, protocols, and more.

PCAP Tools: dumpcap (1)

```
PS C:\Program Files\Wireshark> .\dumpcap.exe -h
Dumpcap (Wireshark) 2.6.5 (v2.6.5-0-gf766965a)
Capture network packets and dump them into a pcapng or pcap file.
See https://www.wireshark.org for more information.

Usage: dumpcap [options] ...

Capture interface:
  -i <interface>           name or idx of interface (def: first non-loopback),
                           or for remote capturing, use one of these formats:
                               rpcap://<host>/<interface>
                               TCP@<host>:<port>
  -f <capture filter>      packet filter in libpcap filter syntax
  -s <snaplen>             packet snapshot length (def: appropriate maximum)
  -p                       don't capture in promiscuous mode
  -I                       capture in monitor mode, if available
  -B <buffer size>         size of kernel buffer in MiB (def: 2MiB)
  -y <link type>           link layer type (def: first appropriate)
  --time-stamp-type <type> timestamp method for interface
  -D                       print list of interfaces and exit
  -L                       print list of link-layer types of iface and exit
  --list-time-stamp-types  print list of timestamp types for iface and exit
  -d                       print generated BPF code for capture filter
  -k                       set channel on wifi interface:
                           <freq>,[<type>],[<center_freq1>],[<center_freq2>]
  -S                       print statistics for each interface once per second
  -M                       for -D, -L, and -S, produce machine-readable output
```

dumpcap is another tool available both on Windows and Linux. It is installed with Wireshark and provides many of the same options that **tcpdump** does. In some cases, the switches are the same, but for other options, **dumpcap** uses different switches entirely. When using **dumpcap** on Windows, packet capture software is also required, such as Npcap which we discussed earlier.

PCAP Tools: dumpcap (2)

```
PS C:\Program Files\Wireshark> .\dumpcap.exe -w C:\Tools\cap.pcap -a duration:300 -P
Capturing on 'Ethernet0'
File: C:\Tools\cap.pcap
Packets captured: 32129
Packets received/dropped on interface 'Ethernet0': 32129/0 (pcap:0/dumpcap:0/flushed:0/ps_ifdrop:0) (100.0%)
```

Options are similar to tcpdump, but verify options via help to confirm switches

```
root@siftworkstation:~# tcpdump -i eth0 -w /tmp/test.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
^C4770 packets captured
4770 packets received by filter
0 packets dropped by kernel
```

Above we see examples of a capture started with **dumpcap** as well as **tcpdump**. Both are targeting a specific interface and are saving captured packets out to a file via the **`-w`** switch. Notice however, that **dumpcap** has other options in use to limit the duration to 300 seconds. The **`-P`** switch used with **dumpcap** tells **dumpcap** to save the packet capture in PCAP format vs. the newer PCAP-ng format. In some cases PCAP-ng can confuse other tools, so we recommend sticking to the tried and true PCAP format when using **dumpcap**.

To see all the command line arguments available, use **`-h`**.

Exercise 5.2A

PCAP Collection

Synopsis: In this exercise, you will use dumpcap to collect network traffic and save it to to PCAP files. You will then use filters to limit the traffic collected by dumpcap.

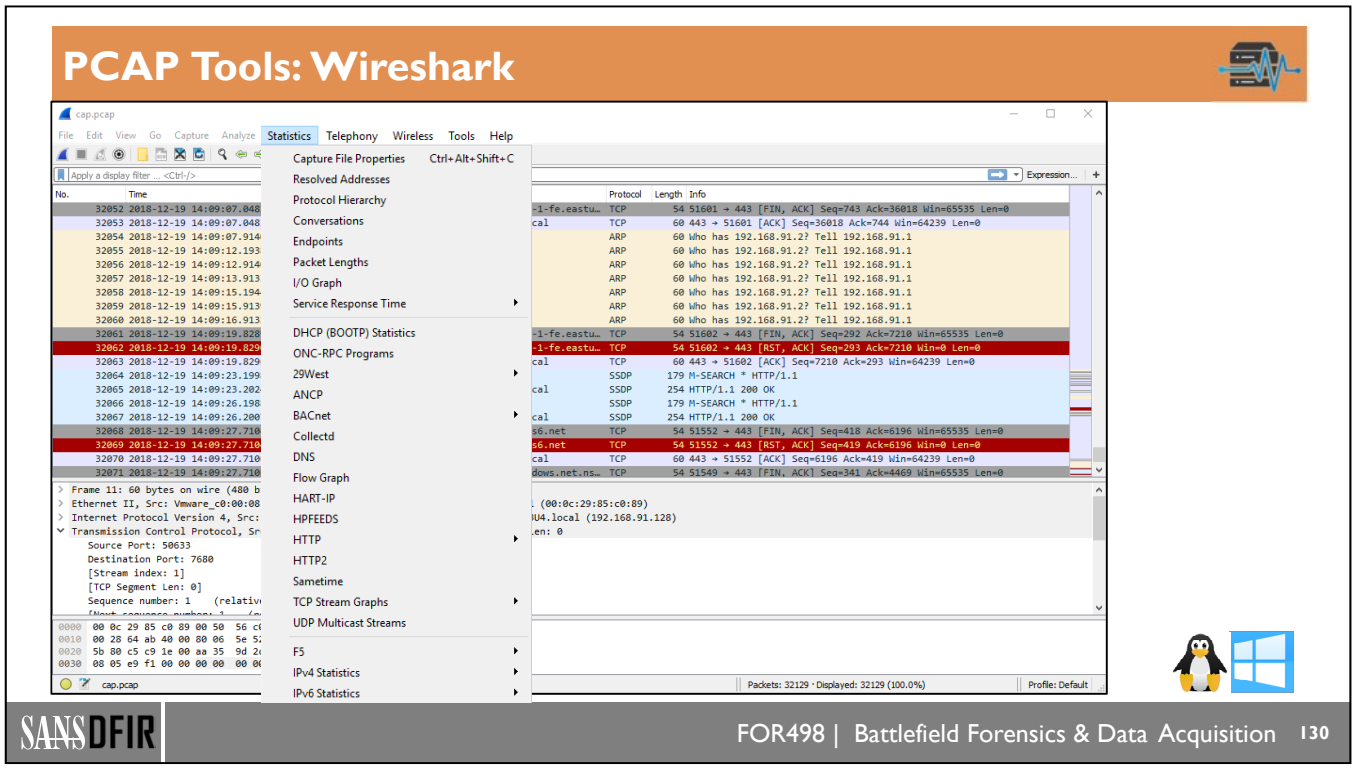
Average Time: 25 Minutes

This page intentionally left blank.

Exercise 5.2A Takeaway

- Tools like dumpcap and tcpdump allow for the collection of network traffic for later analysis.
- The collection of traffic can be filtered based on a wide range of criteria which makes it very flexible.
- Encryption of data on the wire may prevent you from getting all the answers you would like to get.

This page intentionally left blank.

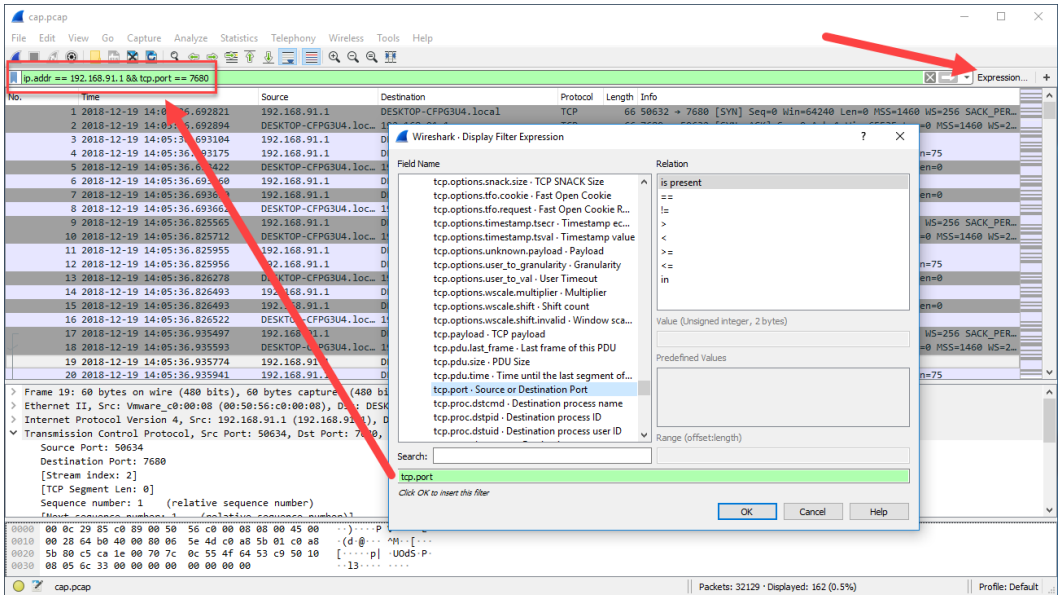


Wireshark is a program that allows for loading PCAP files and interacting with network traffic using a graphical user interface. It includes features like filtering and searching, as well as built in reports related to the parsed network traffic. The screen shot above shows what a typical Wireshark session may look like, but with a few tweaks to the display format. Notice that the Time column has a full date/time for each line. This allows you to see exactly when a packet was encountered vs seeing the time since the beginning of the packet, and so on. The other change is that IP addresses have been resolved to host names. This is also optional but can make it easier to find interesting things based on host name. Each of these settings can be adjusted under the View menu.

The Statistics tab is also a great resource in Wireshark because it contains a wide range of reports and summaries for common questions, such as DNS traffic, HTTP conversations, IPv4 statistics, and so on. The options under the Statistics menu enable you to quickly answer such questions as “Which hosts sent or received the largest volume of traffic?” and “What HTTP requests were made?”. While this information could be gathered and aggregated manually, by leveraging the built-in capabilities of the tool you will save time, and because the software will do things the same way every time, will have more repeatable and consistent results.

The Statistics tab is a great way to get an overall feel for what kind of traffic you have collected, so it should be one of the first places you check out once you have a PCAP file collected.

PCAP Tools: Wireshark Filtering (1)



Wireshark also includes extensive filtering capabilities that can dissect packets based on IP addresses, ports, protocol, TCP flags, and just about anything else you have ever heard of as it relates to network traffic. At the top of the packet list is a text field that allows for free form entry of filter criteria that can be used to quickly enter preconfigured filters.

To assist you as you are learning about filter expressions, to the far right of the text box is an Expression button that, when clicked, brings up an interface that contains all available filter criteria, including a Search box at the bottom. As criteria is found, clicking the OK button will insert it into the current filter. This allows you to build complex filters quickly. Other things to notice are the ability to do comparisons like equals, less than, not equal to, and so on, which let you drill down into specific network data even faster.

The Analyze | Display Filters menu contains many excellent, preconfigured filters you can use, both as examples, and to quickly locate network packets that match the description under the Name column. You can also add new filters to this list, making it even easier to reference filters that help in network traffic analysis. A cheat sheet of the more commonly used filter criteria can be found at the link below[1].

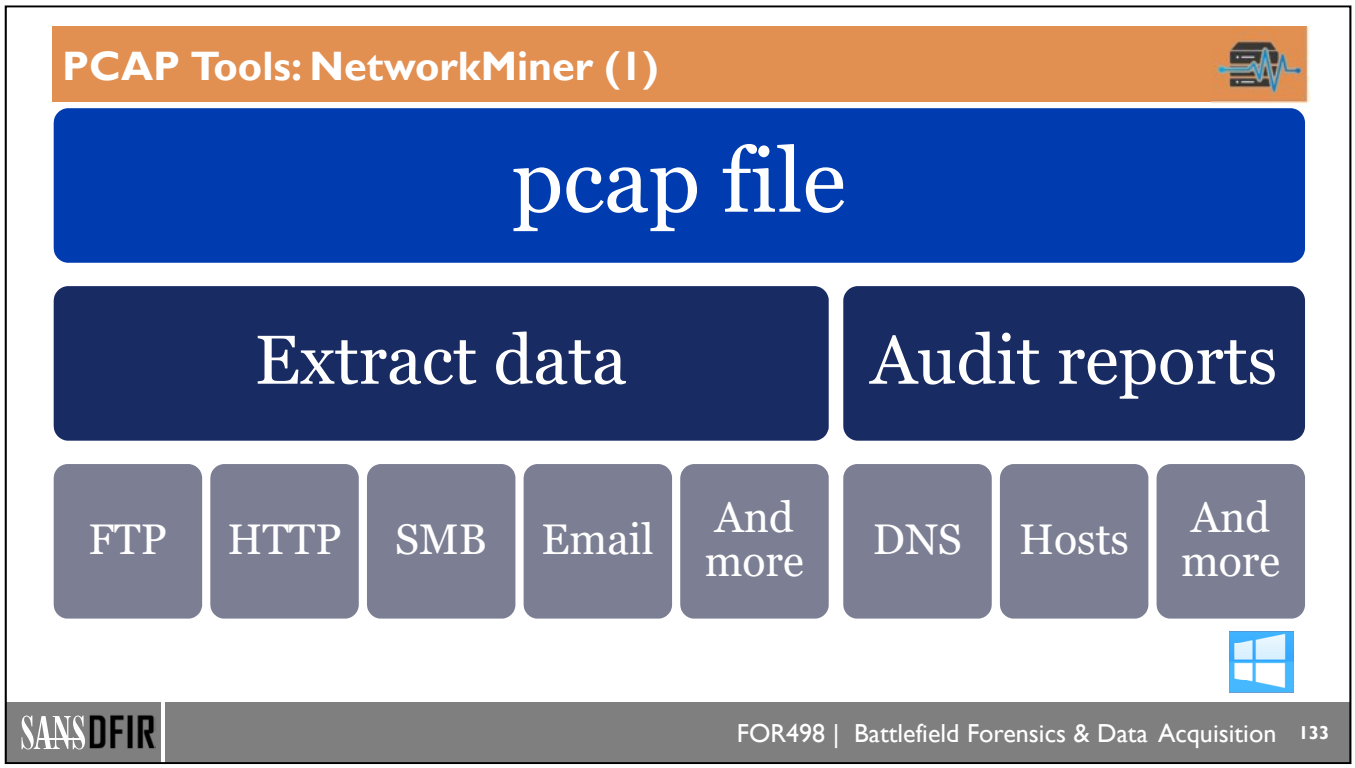
[1] Cheat sheets: tcpdump and Wireshark | `https://for498.com/6fcmg`

PCAP Tools: Wireshark Filtering (2)

Wireshark · Display Filters

Name	Filter
Non-HTTP and non-SMTP to/from 192.0.2.1	ip.addr == 192.0.2.1 and not tcp.port in {80 25}
No ARP and no DNS	not arp and !(udp.port == 53)
HTTP	http
TCP or UDP port is 80 (HTTP)	tcp.port == 80 || udp.port == 80
Non-DNS	!(udp.port == 53 || tcp.port == 53)
UDP only	udp
TCP only	tcp
IPX only	ipx
IPv6 address 2001:db8::1	ipv6.addr == 2001:db8::1
IPv6 only	ipv6
IPv4 address isn't 192.0.2.1 (don't use != for this!)	!(ip.addr == 192.0.2.1)
IPv4 address 192.0.2.1	ip.addr == 192.0.2.1
IPv4 only	ip
No ARP	not arp
Ethernet broadcast	eth.addr == ff:ff:ff:ff:ff:ff
Ethernet type 0x0806 (ARP)	eth.type == 0x0806
Ethernet address 00:00:5e:00:53:00	eth.addr == 00:00:5e:00:53:00

This page intentionally left blank.



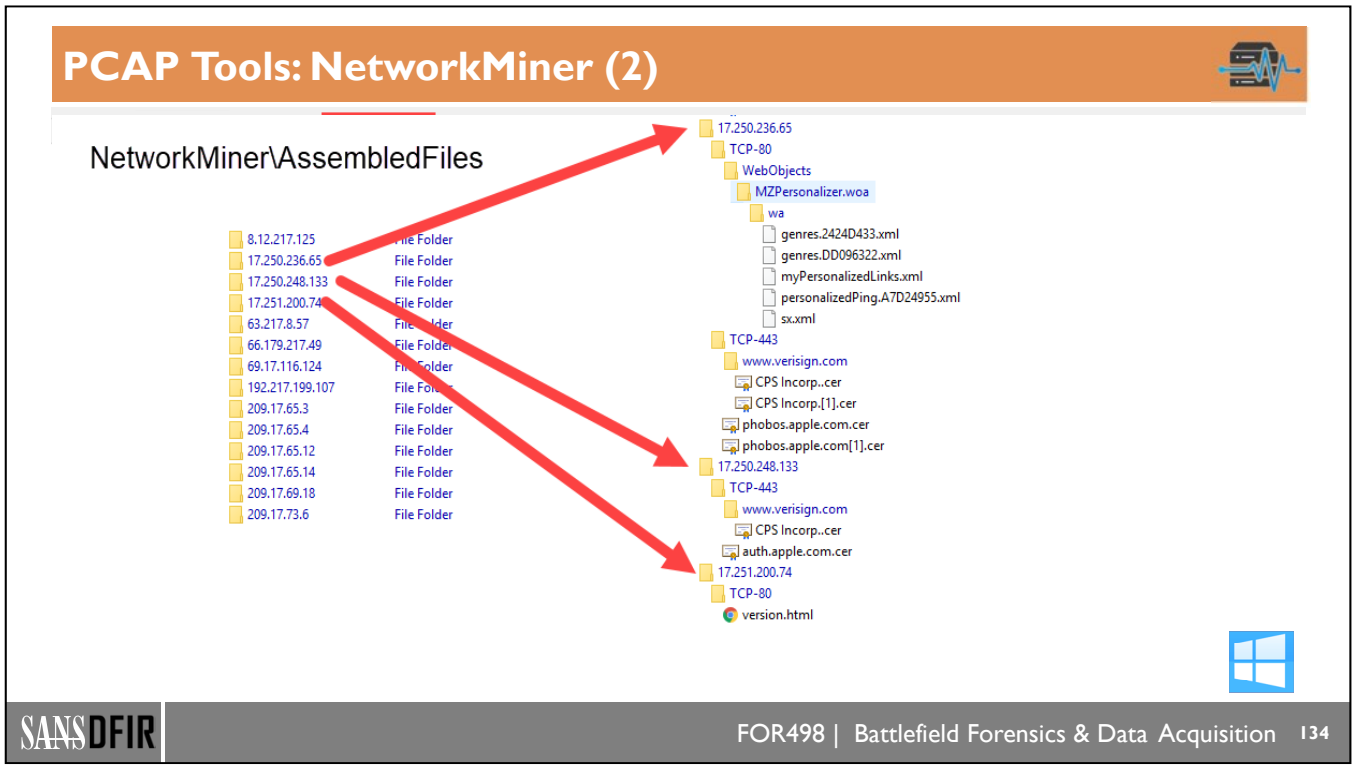
NetworkMiner[1] is an easy to use GUI based program that can do both live traffic capture and analysis as well as reading a PCAP file. In either case, NetworkMiner disassembles network packets into groups of data, such as the computers engaged in conversations, transferred files, pictures, email messages, DNS history, and more. NetworkMiner comes in two versions, a freeware version and a commercial version, which adds even more features such as IP geolocation, time zone support, and whitelisting.

After starting NetworkMiner and loading a PCAP file, a tabbed interface is presented. Clicking on any of the tabs loads the data relevant to that tab. Some tabs offer a filter to look for keywords which is useful when looking for a particular file that was transferred, or email that contained a certain phrase, and so on.

As NetworkMiner processes a PCAP file, it follows the TCP streams in the network traffic and reassembles file transfers. During this process, NetworkMiner actually saves out a copy of all the files that went across the wire to subdirectories where NetworkMiner ran from. This means you can review all the files NetworkMiner was able to find using the tool of your choice. Since the files are extracted from the PCAP file, it allows for hashing or any other processing you deem necessary based on your investigation.

Depending on the size of the PCAP file being analyzed, it can take a while for results to be available, but NetworkMiner is a great tool to get an idea of the kinds of information that exist in a PCAP file.

[1] NetworkMiner - The NSM and Network Forensics Analysis Tool | `https://for498.com/i098c`



Once a PCAP file is loaded, each tab can be clicked on to look at the different buckets of data. Depending on which tab is selected, other options are available via filters at the top, context menus via right clicking on an entry, and so on.

One great feature of NetworkMiner is reassembling packets related to file transfers, such as web traffic, certificates, etc. All this data is broken down by IP address and port, which makes drilling down into a specific conversation very easy. Because the data is saved to disk, it allows for keyword searching, hashing, carving, and any other technique that can be leveraged against files. As hosts of interest are located, it also allows for quickly jumping into that particular host's directory which then leads you to the traffic broken down by the protocol and port used. Nice!

Exercise 5.2B-C

PCAP Graphical Tools

Synopsis: In this exercise, you will use Wireshark to analyze pre-collected PCAP files and extract relevant data. You will then use Network Miner to review and extract conversations, file transferred, etc.

Average Time: 60 Minutes

This page intentionally left blank.

Exercise 5.2B-C Takeaway

- Wireshark provides detailed and full insight into network packets and allows for robust filtering to find items of interest.
- Leveraging one or more tools to automate network analysis can significantly speed up getting to answers.
- NetworkMiner provides an easy way to inspect the contents of a PCAP file.
- Being able to see and interact with files transferred over the network can provide additional context and leads in an investigation.

This page intentionally left blank.

PCAP Tools: tshark (1)

```
PS C:\Program Files\Wireshark> .\tshark.exe -r C:\Tools\cap.pcap -Y http.request
 5776  30.551649 192.168.91.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
 5777  31.553377 192.168.91.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
 5778  32.554295 192.168.91.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
 5783  33.554722 192.168.91.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
11775 143.107444 192.168.91.128 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11796 146.119530 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11803 149.125645 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11806 150.559067 192.168.91.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
11808 151.560222 192.168.91.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
11809 152.132372 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11810 152.132615 192.168.91.1 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11812 152.561233 192.168.91.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
11813 153.562709 192.168.91.1 → 239.255.255.250 SSDP 215 M-SEARCH * HTTP/1.1
11815 155.140574 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11816 155.140826 192.168.91.1 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11824 158.148207 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11825 158.148381 192.168.91.1 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11853 162.905442 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11854 162.905941 192.168.91.1 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11939 165.904652 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11940 165.906929 192.168.91.1 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11959 168.909688 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11960 168.909858 192.168.91.1 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11975 171.932902 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
11976 171.933114 192.168.91.1 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
12566 174.943893 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
12567 174.944090 192.168.91.1 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
12653 177.946505 DESKTOP-CFPG3U4.local → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
12654 177.946772 192.168.91.1 → 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
14143 180.000954 DESKTOP-CFPG3U4.local → www3.l.google.com HTTP 279 GET /GTSGIAG3/MEk
14432 180.150134 DESKTOP-CFPG3U4.local → www3.l.google.com HTTP 279 GET /GTSGIAG3/MEk
16465 181.436144 DESKTOP-CFPG3U4.local → www3.l.google.com HTTP 279 GET /GTSGIAG3/MEk
```

Tshark provides command level access to essentially all of Wireshark's functionality. **Tshark** is a command line tool, and because of this, it can be scripted and automated against packet capture files, which saves the analyst time, and allows for scanning many capture files much faster than Wireshark can do. **Tshark** is flexible in that it can be used for ad-hoc queries on the fly or wrapped into one or more scripts that are purpose made for finding and even extracting things out of packet capture files.

Tshark usage is as follows:

```
Tshark –r <input.pcap> <other options> -Y "<Wireshark display filter>"
```

Depending on your needs, other common switches include **–n** (do not do DNS resolution, i.e. IP addresses to host names), and **–T** fields, which, when used in conjunction with **–e** <Wireshark display field>, allows you to selectively choose which fields to extract. An example using all these options might look like:

```
Tshark –r foo.pcap -n -Y 'http.user_agent contains "Chrome"' -T fields -e
ip.src -e http.host -e http.user_agent
```

The above says to read PCAP data from foo.pcap, do not resolve IP addresses to host names, only show packets that contain the string "Chrome" in the user agent, and, when a match is found, display the source IP address (i.e. who made the request), the destination IP and port, and the full user agent.

Compare this with the first example, which finds all packets with User Agent fields (because the command does not contain the 'contains "Chrome"' piece). When designing commands, we recommend you start overly broad and get more specific, so you can be sure your queries are returning the data you are expecting them to.

PCAP Tools: tshark (2)

```
PS C:\Program Files\Wireshark> .\tshark.exe -r C:\Tools\cap.pcap -Y 'http.user_agent'
 -T fields -e ip.src -e http.host -e http.user_agent
192.168.91.1    239.255.255.250:1900    Google Chrome/71.0.3578.98 Windows
192.168.91.1    239.255.255.250:1900    Google Chrome/71.0.3578.98 Windows
192.168.91.1    239.255.255.250:1900    Google Chrome/71.0.3578.98 Windows
192.168.91.1    239.255.255.250:1900    Google Chrome/71.0.3578.98 Windows
192.168.91.1    239.255.255.250:1900    Google Chrome/71.0.3578.98 Windows
192.168.91.1    239.255.255.250:1900    Google Chrome/71.0.3578.98 Windows
192.168.91.1    239.255.255.250:1900    Google Chrome/71.0.3578.98 Windows
192.168.91.1    239.255.255.250:1900    Google Chrome/71.0.3578.98 Windows
192.168.91.128  ocsp.pki.goog   Microsoft-CryptoAPI/10.0
192.168.91.128  ocsp.pki.goog   Microsoft-CryptoAPI/10.0
192.168.91.128  ocsp.pki.goog   Microsoft-CryptoAPI/10.0
```


Once you review your initial command, look for data you either want to keep, or inversely, want to drop, and update the filter accordingly. Over time you will produce recipes you can reuse for a wide range of investigations.

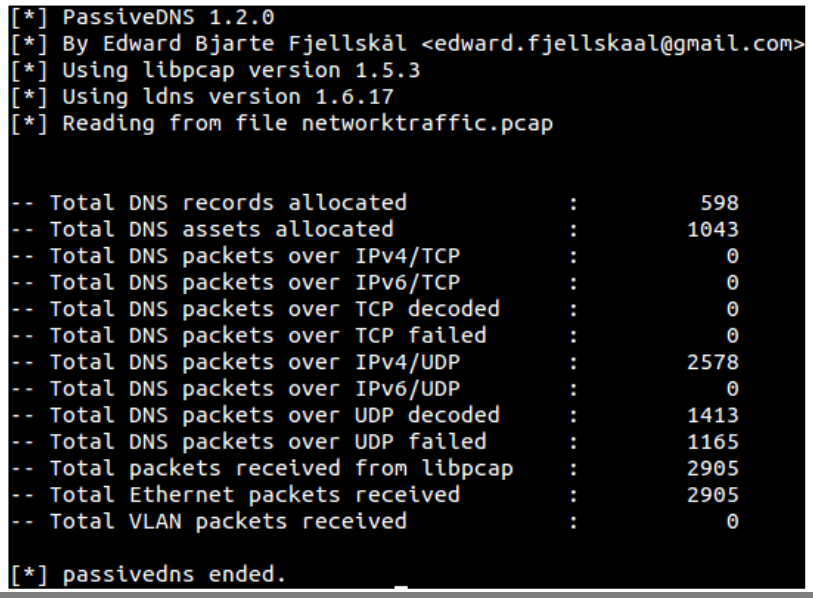
As you get more familiar with Wireshark and its filters, you will automatically gain this same level of ability in **Tshark**, along with its other advantages. **Tshark** is a powerful tool that can do a lot of the tedious work for you when you need to look at packet captures.

Wireshark, **Tshark**, and **tcpdump** are by no means a comprehensive list of available tools. For example, if you need to find and save out files (such as downloads or network file copies), tools like **tcpxtract** and NetworkMiner can be used for such purposes. For examples on these and many other programs, see the SANS Network Forensics poster[1].

Finally, since **Tshark** uses the same display filters as Wireshark does, any resources and references pertaining to Wireshark can be leveraged with **Tshark**.

[1] SANS Network Forensics poster `https://for498.com/h2p0o`

PCAP Tools: passivedns (1)



```
[*] PassiveDNS 1.2.0
[*] By Edward Bjarte Fjellskål <edward.fjellskaal@gmail.com>
[*] Using libpcap version 1.5.3
[*] Using ldns version 1.6.17
[*] Reading from file networktraffic.pcap


-- Total DNS records allocated            :         598
-- Total DNS assets allocated             :        1043
-- Total DNS packets over IPv4/TCP        :           0
-- Total DNS packets over IPv6/TCP        :           0
-- Total DNS packets over TCP decoded     :           0
-- Total DNS packets over TCP failed      :           0
-- Total DNS packets over IPv4/UDP        :        2578
-- Total DNS packets over IPv6/UDP        :           0
-- Total DNS packets over UDP decoded     :        1413
-- Total DNS packets over UDP failed      :        1165
-- Total packets received from libpcap    :        2905
-- Total Ethernet packets received        :        2905
-- Total VLAN packets received            :           0

[*] passivedns ended.
```

passivedns [1] is a command line program that ingests a PCAP file and summarizes the DNS related information contained within it. **passivedns** allows for the distilling of large amounts of PCAP down to the hosts that are making DNS requests, and so for our purposes, this can be very useful to locate hosts communicating both internally and externally with other computers. Using **passivedns** is simple in that we just need to supply a PCAP file to read, and two output files to write the results to, like this:

```
passivedns -r networktraffic.pcap -l dnslog.txt -L nxdomain.txt
```

passivedns runs very quickly and when its finished, displays a summary of what it did. When reviewing the two output files, we may only see data in one of them. This is because nothing was found that should end up in the other file. In other words, there was no DNS traffic that matched the kind of information that would be saved.

The **-r** option is the input file to read, the **-l** option (lower case 'ell') file will contain successful queries, and the **-L** option file will contain any errors. If you are not interested in errors, the **-L** option can be dropped.

[1] passivedns | `https://for498.com/tfa5o`

PCAP Tools: passivedns (2)

Timestamp || Client IP || Server IP || Class || Query || Query Type || Answer || TTL || Count

```
1216691468.768180||192.168.1.64||192.168.1.254||IN||www.blogger.com.||CNAME||blogger.l.google.com.||80||1
1216691468.768180||192.168.1.64||192.168.1.254||IN||blogger.l.google.com.||A||72.14.223.191||112||1
1216691479.141145||192.168.1.64||192.168.1.254||IN||f.e.drugstore.com.||CNAME||f.chtah.com.||3600||1
1216691479.141145||192.168.1.64||192.168.1.254||IN||f.chtah.com.||A||209.3.183.2||511||1
1216691479.141145||192.168.1.64||192.168.1.254||IN||f.chtah.com.||A||216.15.189.52||511||1
1216691479.141145||192.168.1.64||192.168.1.254||IN||f.chtah.com.||A||216.15.189.53||511||1
1216691479.141145||192.168.1.64||192.168.1.254||IN||f.chtah.com.||A||216.15.189.54||511||1
1216691479.141145||192.168.1.64||192.168.1.254||IN||f.chtah.com.||A||208.49.63.20||511||1
1216691479.141145||192.168.1.64||192.168.1.254||IN||f.chtah.com.||A||209.3.183.4||511||1
1216691479.200991||192.168.1.64||192.168.1.254||IN||e.drugstore.com.||A||65.175.87.70||600||1
1216691479.200991||192.168.1.64||192.168.1.254||IN||e.drugstore.com.||A||74.127.1.71||600||1
1216691484.304304||192.168.1.64||192.168.1.254||IN||fxfeeds.mozilla.com.||CNAME||fxfeeds.mozilla.org.||484||1
1216691484.304304||192.168.1.64||192.168.1.254||IN||fxfeeds.mozilla.org.||A||63.245.209.121||32||1
1216691484.404047||192.168.1.64||192.168.1.254||IN||newsrss.bbc.co.uk.||CNAME||newsrss.bbc.net.uk.||871||1
1216691484.404047||192.168.1.64||192.168.1.254||IN||newsrss.bbc.net.uk.||A||212.58.226.75||198||1
1216691515.397011||192.168.1.64||192.168.1.254||IN||www.phdcomics.com.||A||69.17.116.124||372||1
```

date -d @1216691467.389299 → Tue Jul 22 01:51:07 UTC 2008

The DNS log file contains several columns that break down as follows (**passivedns** uses a double pipe for its delimiter):

Timestamp || Client IP || Server IP || Class || Query || Query Type || Answer || TTL || Count

Which looks like this:

Column	Meaning
Timestamp	UNIX (epoch) timestamp with milliseconds
Client IP	The IP address of the client making the request
Server IP	The IP address of the DNS server answering the request
Class	The class of the request
Query	The host name being resolved
Query Type	The record type
Answer	The response sent by the DNS server
TTL	Time to live (how long to cache this answer before asking again)
Count	The number of responses since last request

To convert the timestamp to a human readable format, use the **`date`** command, like this:

```
date -d @1216691467.389299
```

…which converts to '**`Tue Jul 22 01:51:07 UTC 2008`**'

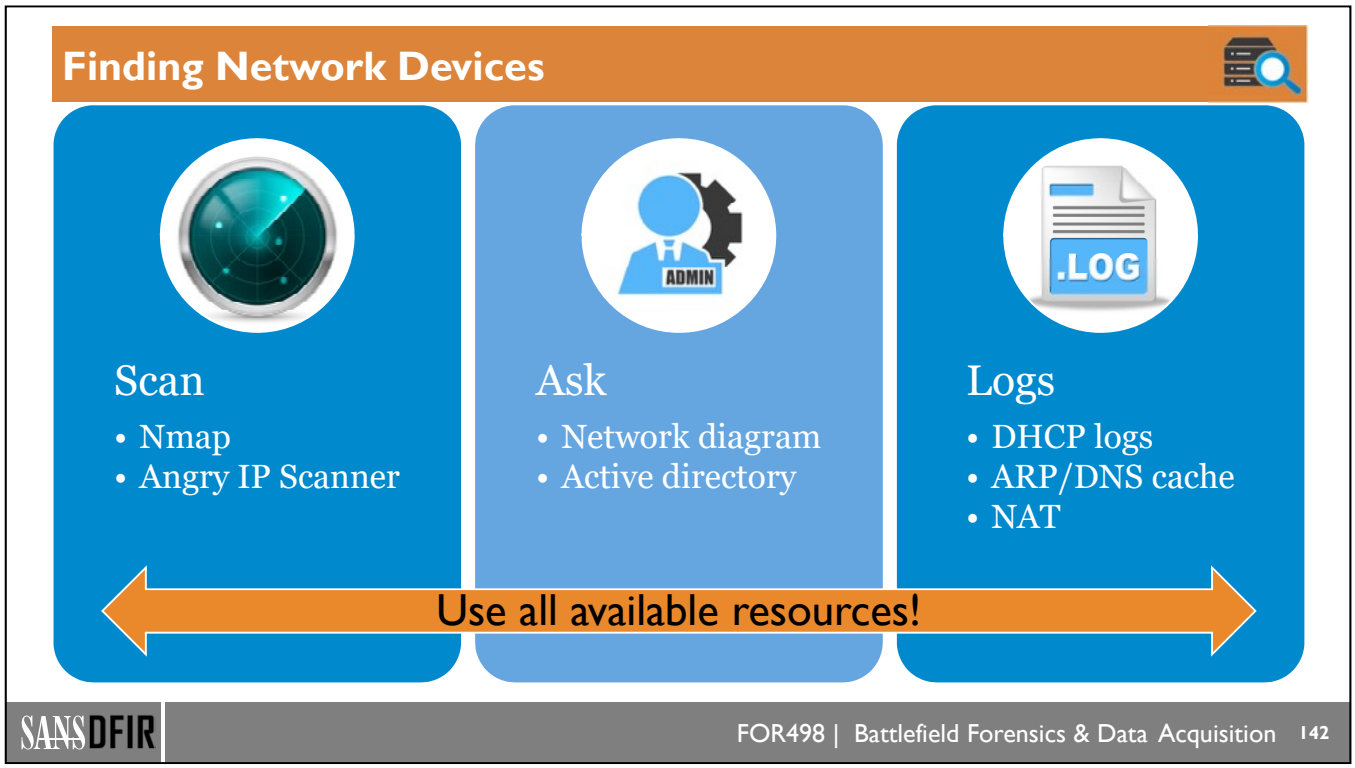
Exercise 5.2D

PCAP Command Line Tools

OPTIONAL, OUT OF CLASS EXERCISE

Synopsis: In this exercise, you will use Tshark from the command line to inspect pre-collected PCAP data. You will then use passivedns to extract out all DNS related information.

This page intentionally left blank.



Internet of Things (IoT) devices can be just about anywhere these days, from doorbells and light bulbs, to cameras and major appliances like microwaves or refrigerators. This is because of the low cost and availability of the hardware that provides network connectivity, as well as consumer demand for such connectivity. Whether this connectivity for everything is a good idea is a matter that is constantly debated. Regardless of which side of the issue you find yourself on, what will always be true is the need to account for such devices in your investigations.

There are two "areas" where you can detect such devices: inside a local area network (LAN), and outside a LAN. Depending on the kind of case you have, you may have access to a LAN and everything that comes with it; DHCP logs, network traffic, a friendly administrator to ask questions to, and so on. But in other cases you may not have access to the LAN, and as such, you can only see traffic coming out of the LAN to the Internet for example. How does being on this side of the LAN affect what you can do when it comes to finding devices? Let's explore being outside a LAN first.

In most cases, Internet connectivity will use some form of Network Address Translation (NAT)[1] to allow multiple devices inside a network to communicate with devices on the Internet. Internally, these devices are using non-routable (or internal, private) IP addresses. NAT keeps track of a map between the non-routable and the public IP address for a network and makes the communication between devices seamless. From our perspective though, this means that even if there are 50 devices inside the LAN, we will see traffic as if its originating from a single IP. In this situation we must look at things like where the traffic is going to and coming from as it relates to hostnames and/or IP addresses. This in turn can provide clues as to what kinds of devices exist inside the network that we do not have visibility of for whatever reason. We will talk about specific techniques to collect traffic soon that can aid us when we find ourselves in this situation.

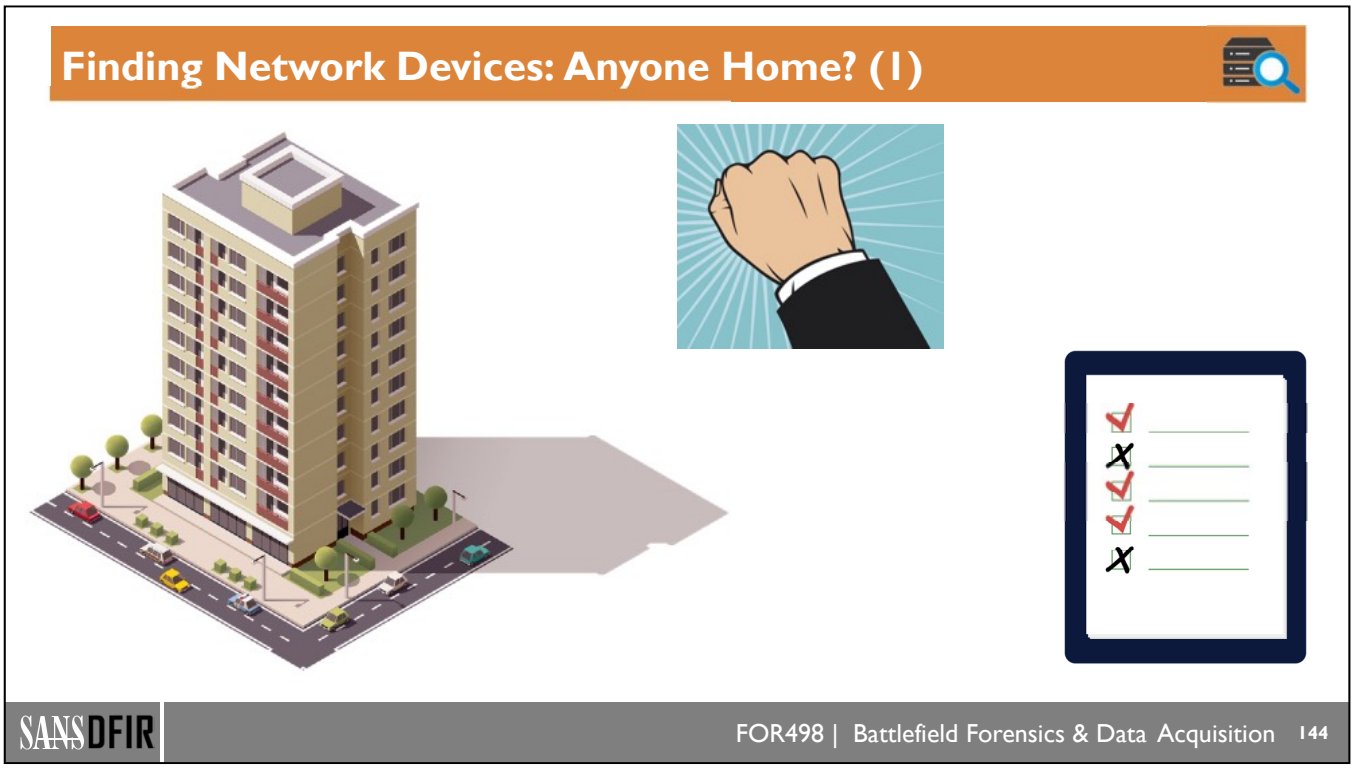
On the other hand, when we are inside a LAN, we have more opportunity to track down individual devices using several techniques including port scanning, DHCP logs, and so on. If we have access to LAN and can connect our own device to it, we can use several different pieces of software to look for devices with network

connectivity (or at least those devices that respond to our search!). Tools like Nmap (command line based) or Angry IP Scanner (GUI) allow for sweeping network subnets to look for IP addresses with open ports, responding to pings, etc. If full packet capture is available, this can be analyzed as well for network devices.

In some situations, looking at the Address Resolution Protocol (ARP)[2] cache can be helpful as it will give you a list of Media Access Control (MAC)[3] addresses that have been resolved on the network. This is often more useful when looking at another device that has been connected to a network for longer periods of time and have communicated with other devices on the LAN. A good technique, whether dealing with ARP or other sources of data when a MAC address is available is to resolve the MAC address to the hardware vendor that manufactured the network interface. This can often lead to clues about the kinds of devices on a network as many times the MAC address will be tied to a specific vendor, like Dell, Synology, Apple, etc.

Finally, when possible, leverage existing sources of data, including log files (DHCP and DNS logs for example) as well as simply asking an administrator for the details about their network.

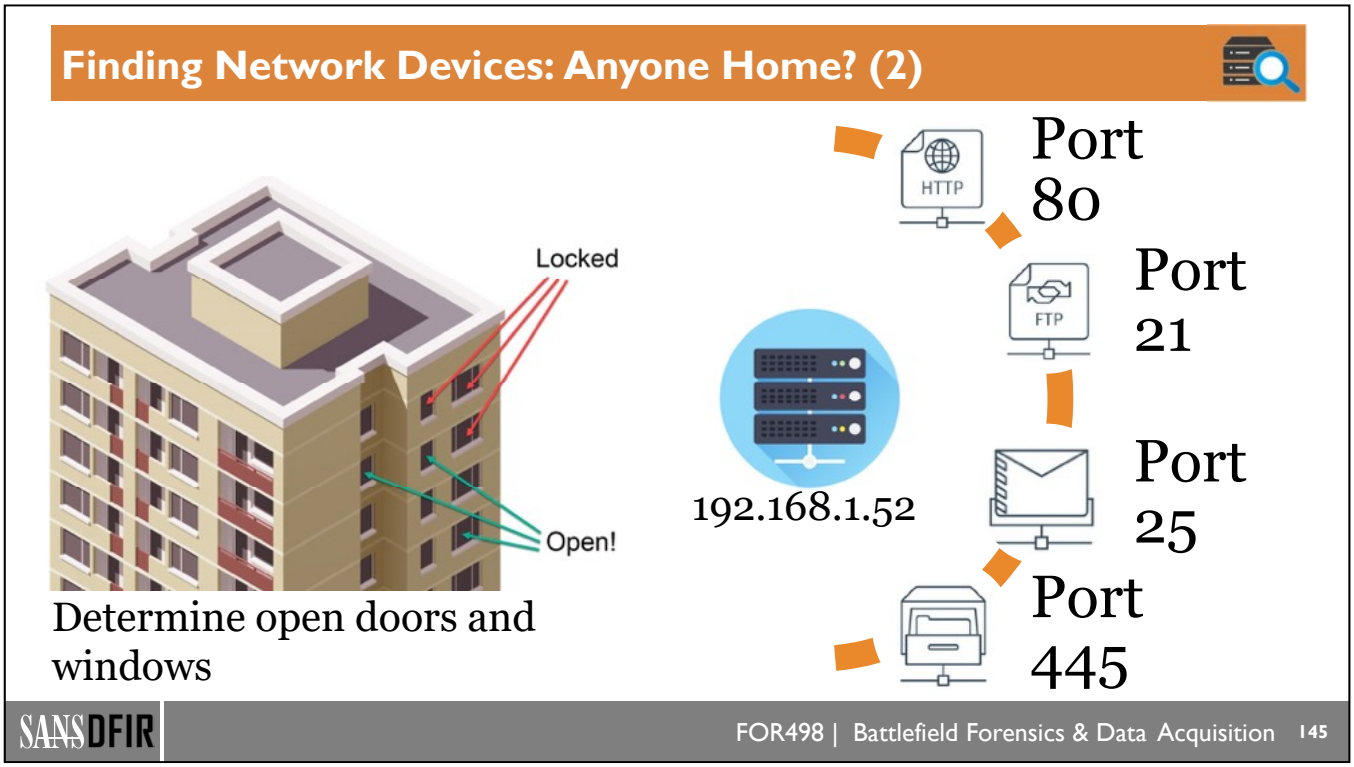
[1] Network Address Translation (NAC) | `https://for498.com/n8w-f`
[2] Address Resolution Protocol (ARP) | `https://for498.com/rbls6`
[3] Media Access Control (MAC) | `https://for498.com/4f182`



Imagine being tasked with finding every person who is home in a large apartment building. How would you go about determining who is home, and who is away? Perhaps the most direct method would be to simply walk around on each floor, knocking on every door. If someone answers the door, you note the apartment number in your report, along with the time the door was answered. When you are done visiting every floor, and knocking on every door, you would know who is home, and who is not.

In much the same way, we often want to know about computers on a network, but in this case, the apartment number can be thought of as an IP address, and the concept of knocking on the door is sending an Internet Control Message Protocol (ICMP)[1] ping request to an IP address. For any given network, couldn't we create a list of all possible IP addresses, send a ping request to each IP address, and record which IP addresses respond with a ping reply? When the process is done, the results would be displayed, and we would know who is home (the machine responded) and who is away (the machine did not respond).

[1] Internet Control Message Protocol (ICMP) | `https://for498.com/k68tr`



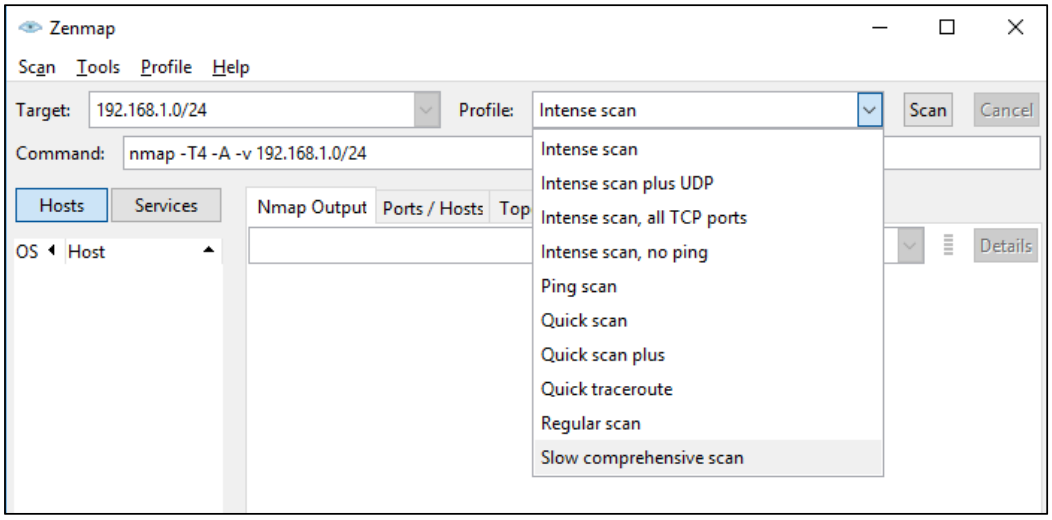
To take the analogy further, what if we, when knocking on a door and having a person answer the door, then took things a step further and asked the resident which, if any, of their doors and windows were unlocked. Perhaps they would say that their bedroom window is unlocked, but everything else is locked. This is analogous on the network side to finding an IP address that answers us, and then looking at each port (or a list or range of ports) and seeing which ports responds to a connection attempt. You can see, in our apartment example, how this would take longer than a simple knock would (because we must ask many more additional questions about doors and windows, which ones are locked, etc.). The same is true when it comes to scanning a network. A simple ping sweep may complete very quickly, but a ping and port scan may take much longer.

It is clear however, that when we check each door and window, we get back a lot more information about any given apartment. In the same way, we get back a lot more detail when looking at IP addresses and listening ports, so the extra time it takes is offset by the insights we gain about a given IP address. Port numbers range from 0 to 65535 , but we typically only scan ports 1-1024 as these are, generally, the ports that services listen on [1]. This is not always the case though, so if you have reason to, expand the port range you scan, just to be sure. Understanding ports can be quite confusing, as there are TCP ports and UDP ports. Ports are further referred to via their port ranges. These consist of Well Known, Registered, and Private or Dynamic.

In our example above, IP address 192.168.1.52 responded to our ping request (someone answered the door when we knocked). Once we got back a ping reply, we did a port scan between 1 and 1024 and found that four ports were listening (when we asked the owner of the apartment which of the 10 windows in their apartment were unlocked, they pointed out four windows). But now what? What good does it do us to know that an IP address has four ports listening on it? This is the kind of detail we will be diving into next.

[1] List of Well-Known TCP Port Numbers | `https://for498.com/62lhe`

Finding Network Devices: Nmap (1)

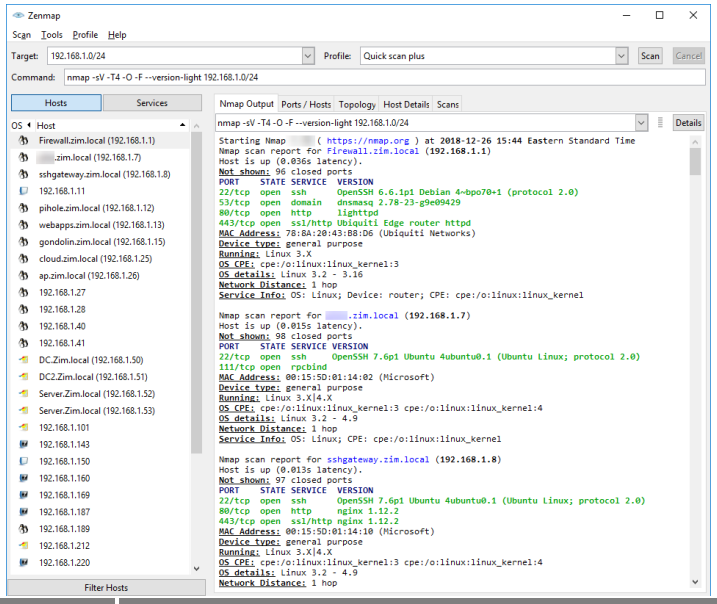


Nmap is a command line network scanning tool that runs on Linux and Windows. There is also a GUI front end available for it called Zenmap, that can assist with making it a bit easier to start using Nmap. One nice thing about Zenmap is that, in addition to it running nmap.exe for you, it shows you the exact command that will be executed. This is useful for automation once you determine what kind of scan you will typically use.

Zenmap also breaks down the text-based output from the Nmap program into various sections, such as hosts, services, ports, host details, graphical network map (nice for reports), etc. Each group of information is held within a tab or list and the details from some areas (like Nmap output) can be selected and copied as needed.

Zenmap serves as a great introduction to help get comfortable with various options in Nmap, as well as making the large amount of information that Nmap generates a bit easier to digest and understand. If you do not already have experience with the Nmap command line interface, we recommend trying out Zenmap a few times to get a feel for what the different types of scans can do.

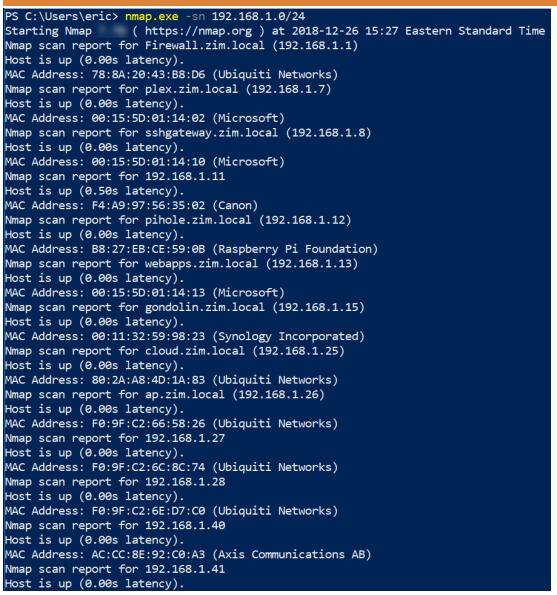
Finding Network Devices: Nmap (2)



- Color coding
- OS icons
- Open ports
- Service info
- Traceroute info

This page intentionally left blank.

Finding Network Devices: Nmap (3)



```
PS C:\Users\eric> nmap.exe -sn 192.168.1.0/24
Starting Nmap [redacted] ( https://nmap.org ) at 2018-12-26 15:27 Eastern Standard Time
Nmap scan report for Firewall.zim.local (192.168.1.1)
Host is up (0.00s latency).
MAC Address: 78:8A:20:43:B8:D6 (Ubiquiti Networks)
Nmap scan report for plex.zim.local (192.168.1.7)
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:14:02 (Microsoft)
Nmap scan report for sshgateway.zim.local (192.168.1.8)
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:14:10 (Microsoft)
Nmap scan report for 192.168.1.11
Host is up (0.50s latency).
MAC Address: F4:A9:97:56:35:02 (Canon)
Nmap scan report for pihole.zim.local (192.168.1.12)
Host is up (0.00s latency).
MAC Address: B8:27:EB:CE:59:0B (Raspberry Pi Foundation)
Nmap scan report for webapps.zim.local (192.168.1.13)
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:14:13 (Microsoft)
Nmap scan report for gondolin.zim.local (192.168.1.15)
Host is up (0.00s latency).
MAC Address: 00:11:32:59:98:23 (Synology Incorporated)
Nmap scan report for cloud.zim.local (192.168.1.25)
Host is up (0.00s latency).
MAC Address: 80:2A:A8:4D:1A:83 (Ubiquiti Networks)
Nmap scan report for ap.zim.local (192.168.1.26)
Host is up (0.00s latency).
MAC Address: F0:9F:C2:66:58:26 (Ubiquiti Networks)
Nmap scan report for 192.168.1.27
Host is up (0.00s latency).
MAC Address: F0:9F:C2:6C:8C:74 (Ubiquiti Networks)
Nmap scan report for 192.168.1.28
Host is up (0.00s latency).
MAC Address: F0:9F:C2:6E:D7:C0 (Ubiquiti Networks)
Nmap scan report for 192.168.1.40
Host is up (0.00s latency).
MAC Address: AC:CC:8E:92:C0:A3 (Axis Communications AB)
Nmap scan report for 192.168.1.41
Host is up (0.00s latency).
```

```
Nmap scan report for sshgateway.zim.local (192.168.1.8)
Host is up (0.0041s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 58:d4:36:34:bc:d3:89:69:ca:0c:8d:9d:1c:e1:69:38 (RSA)
|   256 39:52:1b:c8:57:9a:92:06:0e:6f:af:8f:c3:c6:20:e7 (ECDSA)
|_  256 b5:b2:11:22:94:c4:b7:f3:61:38:41:73:da:f1:72:7f (ED25519)
80/tcp  open  http     nginx 1.12.2
|_http-server-header: nginx/1.12.2
|_http-title: Did not follow redirect to https://sshgateway.zim.local:443/
443/tcp open  ssl/http nginx 1.12.2
|_http-server-header: nginx/1.12.2
|_http-title: UNMS 0.11.3
| ssl-cert: Subject: commonName=192.168.1.8
| Subject Alternative Name: IP Address:192.168.1.8
| Not valid before: 2018-01-01T04:00:01
|_Not valid after:  2018-04-11T04:00:01
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
| tls-nextprotoneg:
|_  http/1.1
MAC Address: 00:15:5D:01:14:10 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   4.06 ms sshgateway.zim.local (192.168.1.8)
```


When using Zenmap, the command line version of Nmap is still doing all the heavy lifting. While it is very handy that Zenmap cuts up the output generated by Nmap, in some instances this is not what you want. Another downside of Zenmap is its inability to automate via scripting. While just about any Nmap command should be usable in Zenmap, it is another layer in the process and on some computers, and you may not have access to a GUI. For these reasons, getting direct experience with the command line version of Nmap is recommended.

Running Nmap without any arguments generates a long and comprehensive list of options as well as a few example use cases. Many of the options in Nmap are overkill for what we want to do, so we will focus on two particular scans using Nmap: finding hosts, and finding hosts along with other details, such as listening ports and operating system information.

Nmap accepts host names, IP addresses, and networks when defining a scope to search. This includes 'slash notation' [1] (192.168.1.0/24) as well as range notation (192.168.1.1-18)

Locating hosts (quick)

Command: **`nmap.exe -sn 192.168.1.0/24`**

The **`-sn`** option is a ping scan only. This option disables checking a host to see what ports are listening. As a result, this type of scan runs very quickly. The scope is every IP address from 192.168.1.1 to 192.168.1.255.

An example of this is shown above on the left side. Notice that Nmap shows each IP address that responded to the ping, along with the MAC address of the machine. Additionally, the vendor for the MAC address is shown, giving you an idea of exactly what kind of devices you have on a network. This saves you manual look ups for each unique MAC address (or rather, the first three bytes of the MAC address).

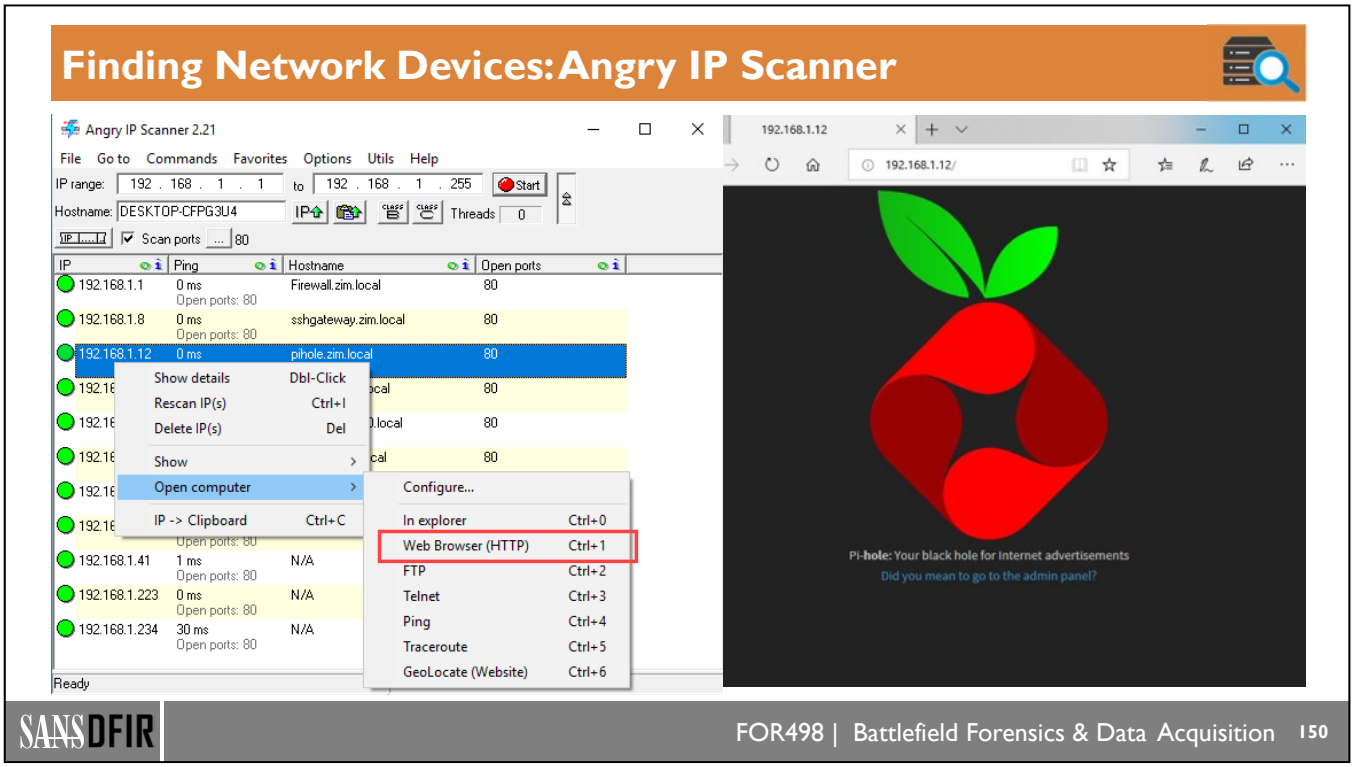
Locating hosts with listening ports (not quick)
Command: **`nmap.exe -A 192.168.1.0/24 --open`**

The **`-A`** option tells Nmap to "enable OS detection, version detection, script scanning, and traceroute" as it searches. The **`--open`** option tells Nmap not to show any details for hosts that it didn't find.

An example of this is shown on the right side of the slide. For this example, notice that we still get some of the same details as in our previous example (IP, host name, and MAC information), but we also get a lot more details, including which ports are open AND what is running on each of the ports. This can be very useful in determining what is running on a device, from printer software, to a database, and everything in between. This search can take MUCH longer to perform because it is not only checking each IP address in the defined range, but for each IP address it finds, it checks a wide range of ports to see what is available. This is in addition to the other options the **`-A`** option provides.

These two examples provide a good starting point you can use and adjust as necessary, depending on the type of investigation, and the depth of answers you are seeking.

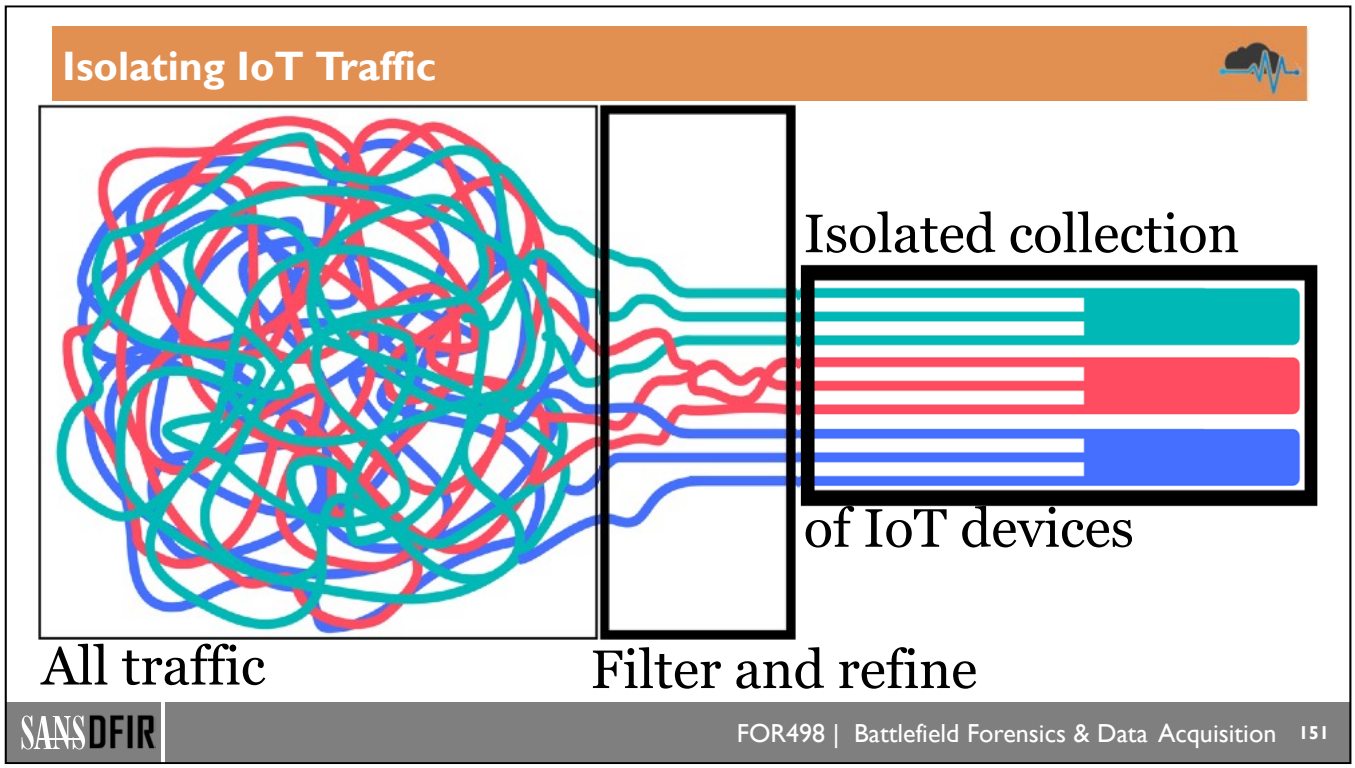
[1] Subnetting, netmasks and slash notation | `https://for498.com/ryl8c`



Angry IP Scanner is a simple program that allows for sweeping a range of IP addresses and optionally, ports on each IP address that is found. While Angry IP Scanner can look for computers on a network in a similar fashion as Nmap, it does not have the advanced features Nmap does, such as determining the operating system on the host and MAC address manufacturer resolution for example. One advantage Angry IP Scanner has over Nmap is that it has a portable version, and does not need to be installed, which makes it easy to get going on a Windows computer that is connected to a network.

Usage is straightforward. Enter a starting and ending IP address in the IP range boxes and, if needed, select which ports to scan. Once the options are configured, click the Start button to begin. In the example above, you can see Angry IP Scanner report the list of IP addresses and the corresponding hostname (if one could be determined). Since ports were scanned (only port 80 was selected in the example above), we can also see the list of ports that were listening on each host.

Finally, each entry has a context menu that allows for getting additional details, copying address information to the clipboard, and the ability to connect to the computer using different programs via the Open computer menu. Each option invokes the proper program, ranging from File Explorer to a web browser. After clicking on the menu entry, or using the shortcut shown, the program is launched, and the connection is made (Edge connected to port 80 on host 192.168.1.12, for example).



Once you have located network devices using one or more of the techniques discussed above, what should you do next? With a list of internal IP addresses in hand related to Internet of Things (IoT) devices, you can now set up additional monitoring of the network that specifically targets only these IPs. Using the tools and techniques discussed earlier related to accessing the network and collecting traffic (using a tap and **tcpdump** for example), we can slightly adjust the collection piece in that we can limit the traffic collected to only the devices we are interested in. This of course would be in addition to any existing network captures (assuming you use PCAP collection and analysis to discover them). In either case, a filter can be applied to **tcpdump** when collecting new traffic and when using **tcpdump** to replay back an already existing PCAP file.

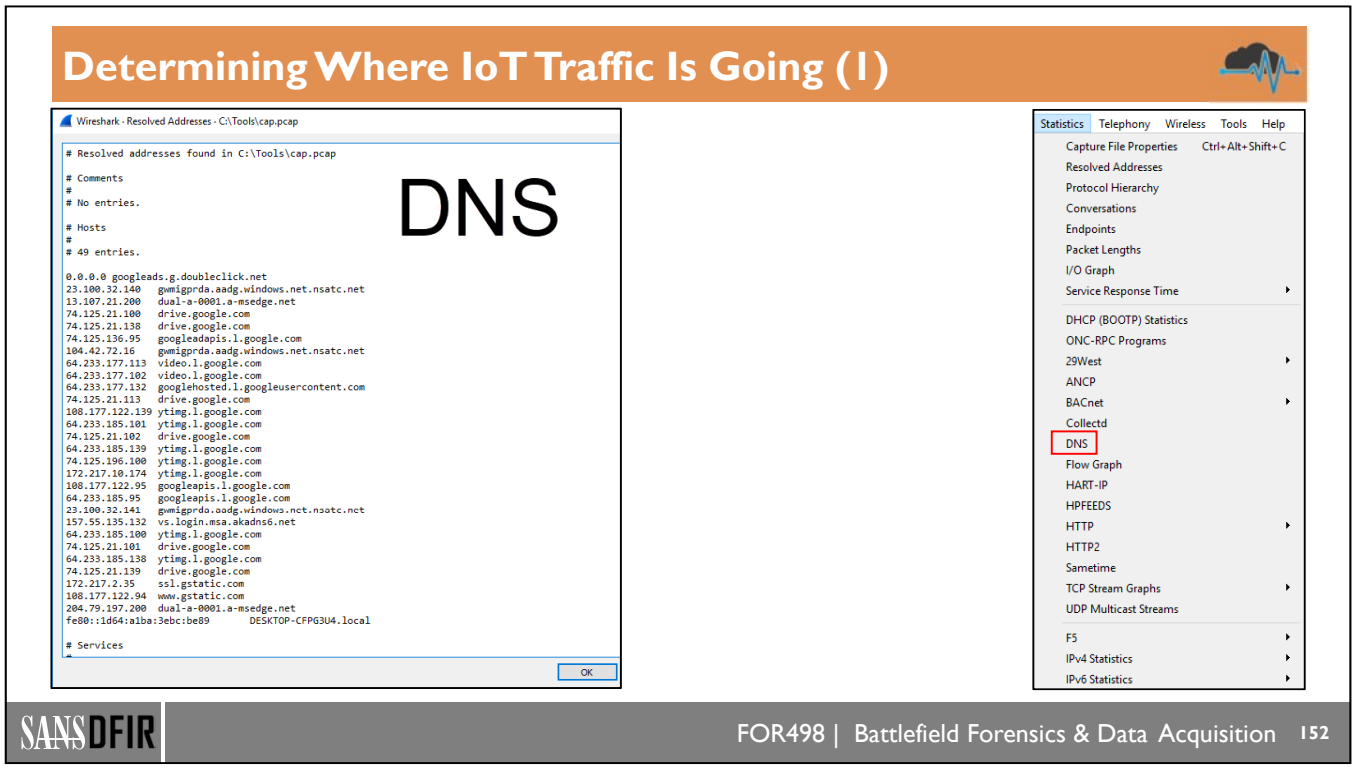
While **tcpdump** (and related programs) supports many different methods to filter, we will look at a few specific keywords: **`host`**, **`src`** and **`dst`**. Each of these works in slightly different ways that you can use based on your needs.

host: looks for traffic going TO or FROM an IP address
src: looks for traffic ORIGINATING from an IP address
dst: looks for traffic DESTINED for an IP address

This would then simply be combined with the IP address you wish to look for. Consider the case where we want to only see traffic going to or coming from IP address 192.168.1.52. In this case, the **`host`** keyword can be used like this:

```
tcpdump host 192.168.1.52
```

To keep the example simple, we are only showing the filter and not the rest of the options we previously discussed. With network traffic filtered based on the IP addresses of interest and traffic collected, you can now focus in on the next stage of the investigation: determining where the traffic from the IoT devices is going, or where traffic coming into IoT devices originated from. We will look at this next.



With more focused PCAP in hand, we can now pivot back to what we saw earlier when using tools like Wireshark and **Tshark** to extract out interesting traffic such as DNS lookups, HTTP/HTTPS requests, file transfers, and more. Using these techniques will yield you another set of data; namely hostnames and/or IP addresses that IoT devices communicated with.

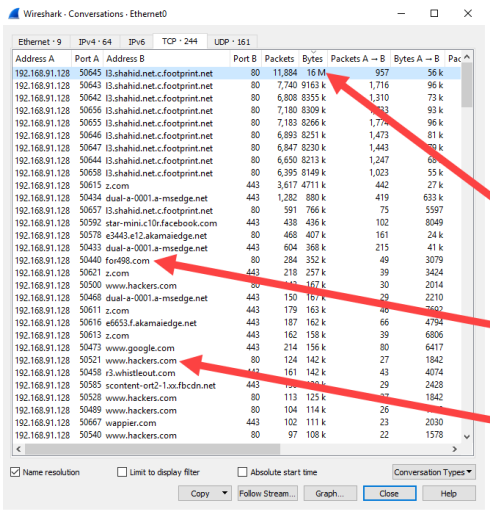
With this information in hand, you can now start creating a picture of where additional data may be, and more importantly, who controls this data. Of course our end goal here is to get this data in a useable form, and while we did capture (in many cases) full PCAP to get to this point, consider the case when HTTPS is being used. Because the connection is fully encrypted, you will have no way to know exactly what went over the wire to an external channel (without things like SSL man-in-the-middle, but that is another story). The data left your network encrypted and ended up on someone else's network, but if you think about it, little could be done with the data in an encrypted form on the other end. In just about every case, the data from outbound connections will interact with a range of other computers and services determined by the IoT device manufacturer, such as hardware configuration, usage patterns, registered mobile devices, and so on.

What we now need is a way to get this data from who holds it. This is where either consent or legal process again comes into play. In some cases you may be able to simply ask or use an administrator level account to get more details as to the particulars regarding IoT devices. Services such as Ring doorbells and Nest cameras can provide web interfaces that allow for some level of reviewing the data held by the provider, but in many cases, the *really* useful stuff will not be available via a web page. Things like a list of all IP addresses and connection times to the devices, billing history, remote interaction with IoT devices (viewing a video for example) history, and so on may only be available by the provider pulling and then making available the more detailed logs in their possession.

Determining things such as who owns an IP address, a domain name, and so on can be done using a wide range of things such as IP geolocation services like MaxMind, to dedicated services such as DomainTools that contain a wealth of information related to IP address ownership, domain WHOIS history, DNS history, and more.

In some cases, traffic may be going into cloud resources, such as Microsoft Azure or Amazon AWS. In these cases, additional legal process or open source research may be required to find out the company or persons who are using Azure or AWS.

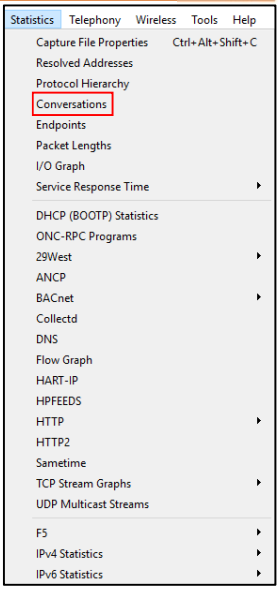
Determining Where IoT Traffic Is Going (2)



Conversations

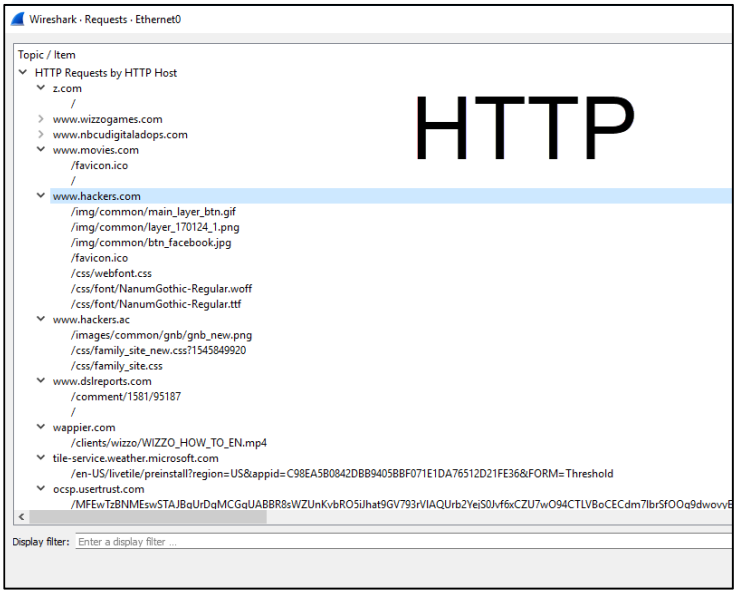
Sort by bytes transfered

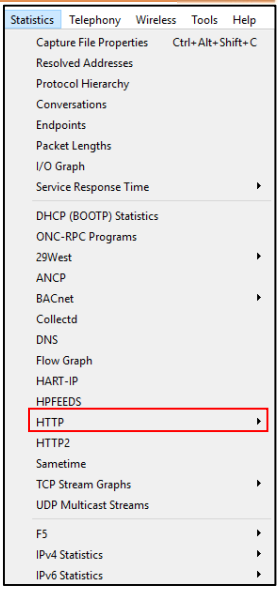
Host names



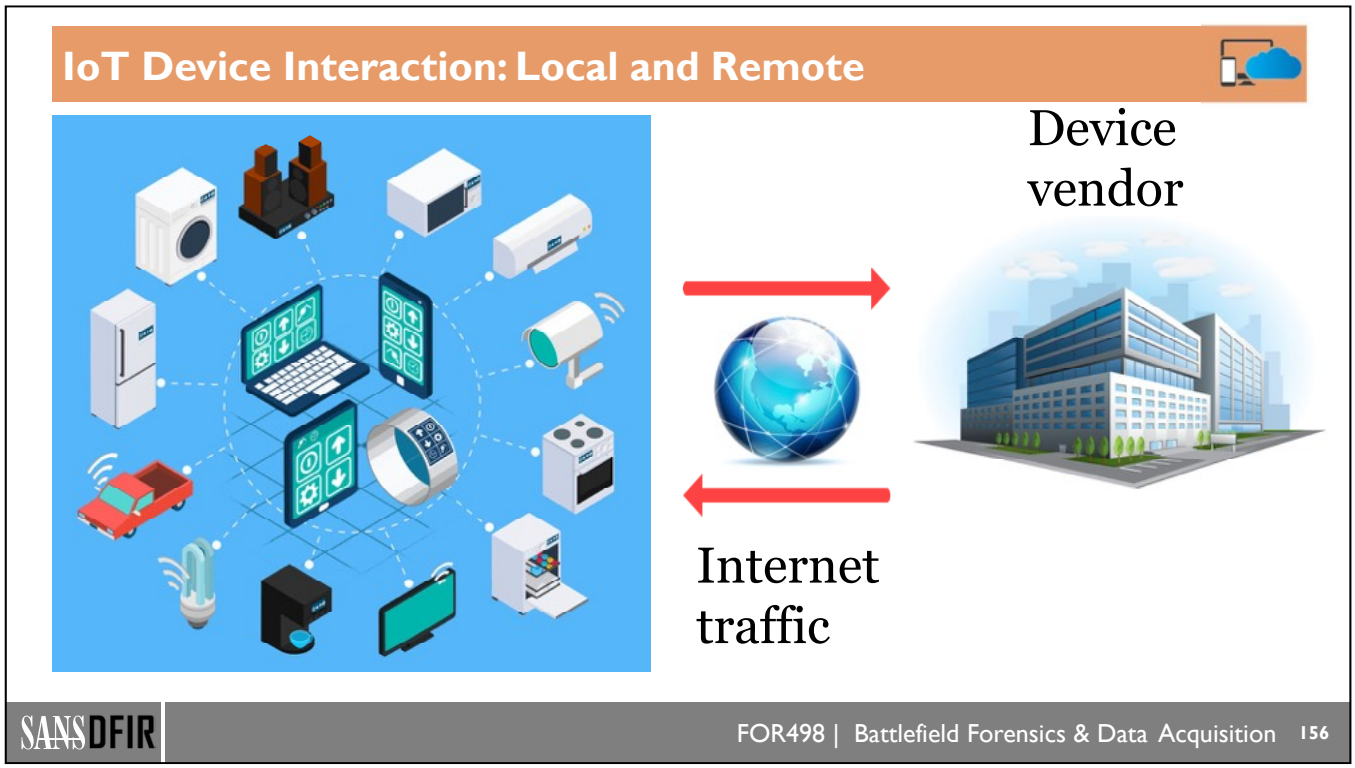
This page intentionally left blank.

Determining Where IoT Traffic Is Going (3)





This page intentionally left blank.



Once you have determined who has the data and you have acquired it via legal means or otherwise, the next stage can be started, which is figuring out if any portable devices have accessed the IoT devices remotely. In some investigations, this information may just fall in your lap while you are working on the previous step of determining traffic to and from LAN devices for example. Recall that we were interested in, and recording traffic TO and FROM, IoT devices of interest. This would include both devices on the LAN AND to external services. Let's talk about LAN devices first.

While monitoring for IoT traffic, if any mobile phones or tablets were also on the same network and interacted with the IoT device, its information would be recorded, including such things as its MAC and IP address. This can then be used to determine the types of devices that are interacting with IoT devices (iPhone or Android phone, a laptop or tablet, and so on).

In the cases where incoming traffic was observed that originated from outside the LAN, the specifics of the device in most cases would not be present unless the remote connection also included the device that initiated the request in the traffic, and the traffic was not encrypted. For most new IoT devices, this will not be the case and you will not have the luxury of seeing details about remote clients, but only the fact that a remote connection was made. This is where you must rely on cloud providers/IoT infrastructure owners to provide you a list, via legal process or some other means, to see which devices have connected to THEIR infrastructure, which then resulted in a connection going to the IoT device in question.

But why do we care about all this detail? Well, with IoT devices, we want to know, at the end of the day, WHO interacted with the IoT device to initiate such things as unlocking doors, viewing a security camera, deleting an archived video, and so on. This is the real story we are trying to tell. We want to show a human being taking action, via some kind of device, to interact with an IoT device. While all the pieces and parts in between are interesting to some degree, always keep in mind we want to find out who is behind the actions that have taken place.

Imaging IoT Devices

Traditional
- Least invasive
- Device may not have any accessible storage

JTAG
- Not always possible

Chip-off
- Most invasive
- Repairing to original state may not be possible

JTAG and chip-off are specialized skills! Do not attempt without proper training!

With IoT devices located, their interactions mapped to mobile devices or other remote management, and so on, it may also become necessary to access any storage available on the IoT device itself. In some cases you may know about what a device can store soon after you identify it (a simple Google search helps here to tell you want a device's capabilities are). The catch here is that a lot of times IoT devices may not allow for traditional collection methodologies to be used (like imaging a traditional file system for example) due to the more embedded nature of IoT devices.

In many cases, logs, access history, and other useful data may not even be stored on the device at all. In this situation we would get this information from the steps previously discussed. But in other cases where there is a bit of flash storage or memory present on the device that stores logs or configuration data, how can we go about accessing that data? If you determine that an IoT device is pertinent to your case and you do need to get any and all available data from it, it may be necessary to go to more invasive procedures such as using a Joint Test Action Group (JTAG)[1] interface or chip-off techniques.

JTAG is a standard used to verify and test printed circuit boards after they have been manufactured. This includes using debug information over a communications interface to interact with the circuit board. In many cases, vendor specific extensions are available, which adds to the amount of information available via this technique. There are many aspects to using JTAG and it is considered a specialized area of expertise. Should this level of interaction with a device be required, track down a JTAG expert to see what is possible.

Another advanced extraction technique is chip-off. This involves actually desoldering chips from the circuit board in order to interact with them via additional, specialized hardware. From here, the raw data can be acquired using this specialized equipment. As with JTAG, chip-off is also a specialized field. When in

doubt, contact someone who specializes in chip-off forensics. Because it involves physically removing chips from circuit boards, this can be a destructive process and is usually undertaken as a last resort. In many cases, repairing a device back to its original state will not be possible [2].

[1] What is JTAG? | `https://for498.com/d19vp`
[2] Chip-Off Forensics | `https://for498.com/903mg`

Summary

- Network taps are an ideal way of collecting data as it travels along the wire
- We collect this data in PCAP files
- In more and more cases today, this is the best way to collect data
- In order to understand communications, we must understand addressing and ports
- There are a number of traffic analysis tools that can assist in rebuilding entire sessions, file transfers, and even phone calls
- The Internet of Things will simply continue to become more and more ingrained in everything we do, so we must learn how to deal with it

This page intentionally left blank.