FOR518 | MAC AND IOS FORENSIC ANALYSIS AND INCIDENT RESPONSE

Workbook

SANS

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

FOR518 | MAC AND IOS FORENSIC ANALYSIS AND INCIDENT RESPONSE

Workbook

SANS

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

Lab 0: Lab Setup (Pre-Class)

Objectives

- Install required software for FOR518: Mac and iOS Forensic Analysis and Incident Response

Class Preparation

This process should take approximately 1 hour, including download time. Xcode is **very** large will take a long time to download; depending on your connection, this process could take longer.

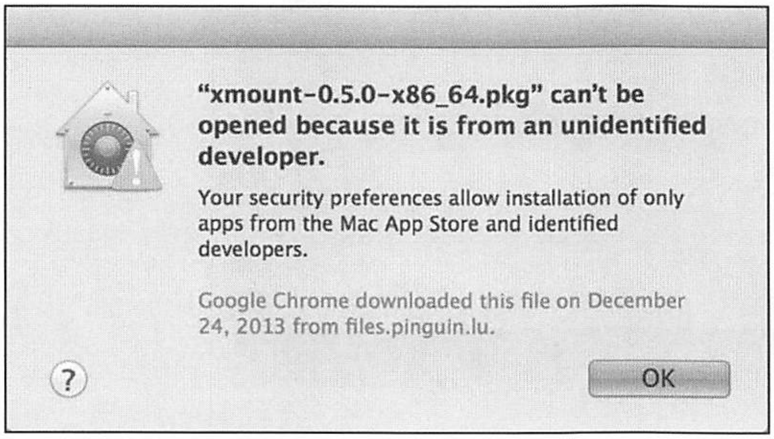
*You may use your host system **or** a virtual machine; however, this setup has not been fully tested in a VM. If you choose to go this route, please be aware that not all tools may work as intended.*

*****NOTE**: It is **very** important that steps 1–5 are followed in order to ensure proper software installation.****

You may download the files at their respective websites listed **or** you may download an archive of these files here: http://for518.com/tools (excludes tools that are too large or needs to be done online). **If you are in class**, the Tools directory on your thumb drives will provide these tools. Please use the application "The Unarchiver" to extract the 7zip files (included on thumb drive).

Gatekeeper Settings:

- Some installer files are from "Unidentified Developers" or "Not from the App Store."

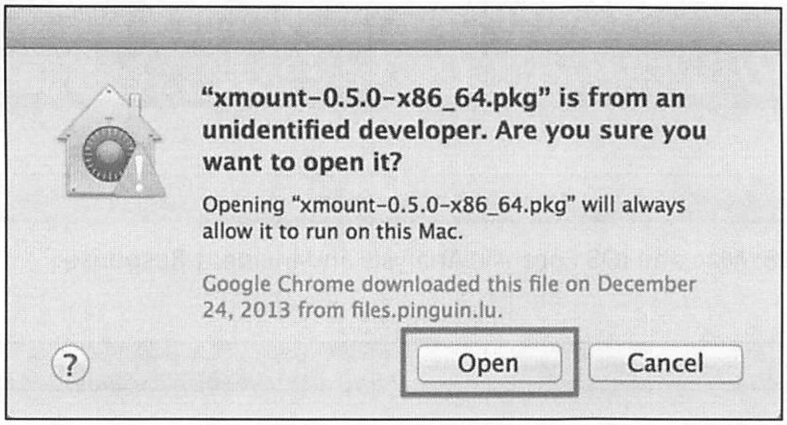


"xmount-0.5.0-x86_64.pkg" can't be opened because it is from an unidentified developer.

Your security preferences allow installation of only apps from the Mac App Store and identified developers.

Google Chrome downloaded this file on December 24, 2013 from files.pinguin.lu.

- Users may allow these files to be installed by Control+clicking the installer file and choosing "Open." A window will pop-up; select "Open."

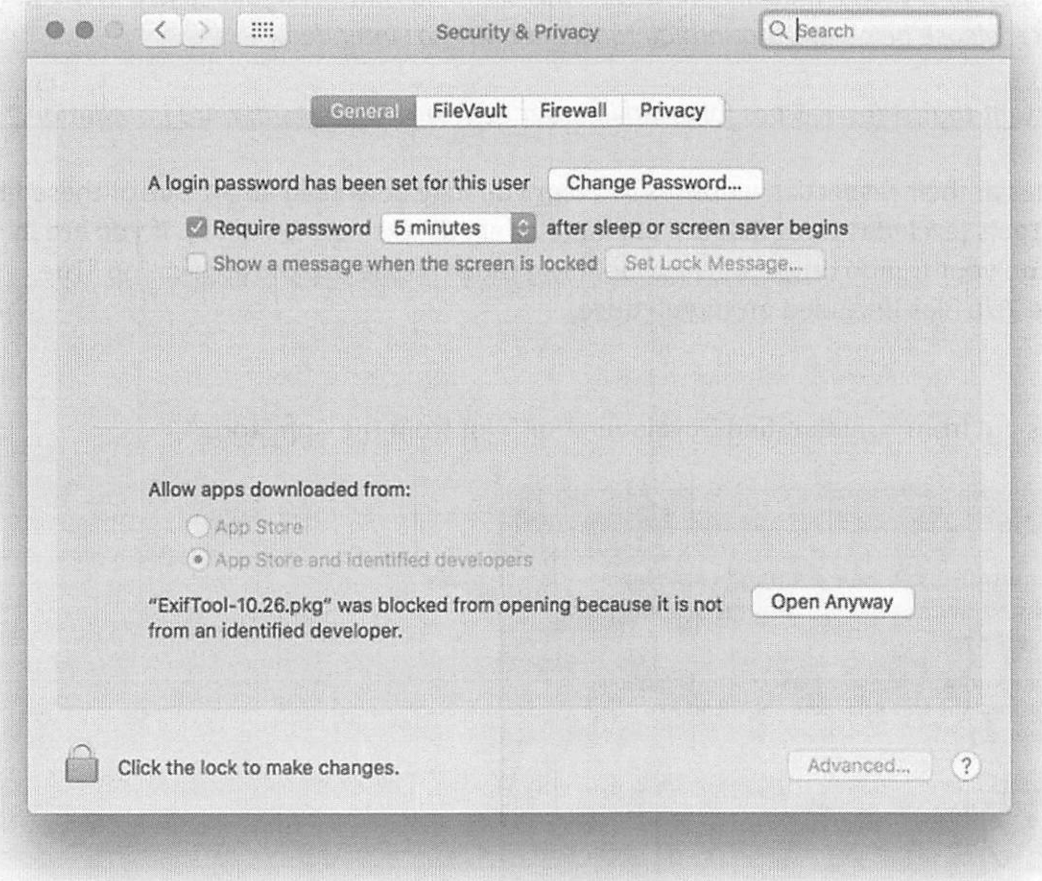


"xmount-0.5.0-x86_64.pkg" is from an unidentified developer. Are you sure you want to open it?

Opening "xmount-0.5.0-x86_64.pkg" will always allow it to run on this Mac.

Google Chrome downloaded this file on December 24, 2013 from files.pinguin.lu.

Another option is to use the "Open Anyway," shown below, each time they get the "Unidentified Developer" or "Not from the App Store" error.



A login password has been set for this user

☑ Require password 5 minutes after sleep or screen saver begins

□ Show a message when the screen is locked

Allow apps downloaded from:

○ App Store

● App Store and identified developers

"ExifTool-10.26.pkg" was blocked from opening because it is not from an identified developer.

Click the lock to make changes.

1. **Xcode and Xcode Command Line Tools**
   - If you have not already done so, register for an Apple Developer Account here. It requires an Apple ID; if you do not have one, you may also register for one at `https://developer.apple.com/register/`
   - Determine your OS version by going to Apple Menu | About This Mac; you will need to download Xcode and Command Line Tools specific for this OS version. This chart may help determine this: `https://en.wikipedia.org/wiki/Xcode#Version_comparison_table`

    1. Please download the latest **Xcode** available for your operating system from the App Store or `https://developer.apple.com/downloads/`
        i. You may have to click “More Downloads” to access older versions.

    2. Please **also** download the latest **Command Line Tools** (for your version of the OS) from `https://developer.apple.com/downloads/`
        i. You may have to go click “More Downloads” to access older versions.

    3. Install **Xcode** (**Note**: This will take a while; grab some coffee.)
        i. If installing via App Store, installation will be done for you.
        ii. If installing via DMG file, open the DMG file and drag the application to the `/Applications` directory.

    4. Install **Command Line Tools**
        i. Open the DMG file, double-click the package installer and follow the default prompts.

2. **OS X FUSE**
    1. Download OSXFUSE from `http://osxfuse.github.io/`
    2. Open the DMG file, double-click the package installer, and follow the default prompts.

3. **xmount 64-bit Package**
    1. Download xmount-0.7.6.pkg (or newer) from `http://www.pinguin.lu/`
        a. Click the `XMOUNT` link on the right side under “Projects.”
        b. Download the package labeled “`Mac OS X 64bit package.`”
    2. Open the DMG file, double-click the package installer, and follow the default prompts.
        a. **If you get the error “OS X Fuse Not Installed Error,” please run the following "mkdir" command below, then rerun the xmount package installer. *(Make sure you type out “osxfusefs.fs” in each case when using tab completion.)***

```
$ mkdir -p /Library/Filesystems/osxfusefs.fs/Support/osxfusefs.kext
```

4. **The Sleuth Kit**
    1. Download `sleuthkit-4.#.#.tar.gz` from `https://www.sleuthkit.org/sleuthkit/download.php.`
    2. Locate and open the `Terminal.app` from `/Applications/Utilities/.`
    3. Use the `cd` command to open the default `Downloads` directory.
    4. Use the `tar` command to unpack the `sleuthkit-4.#.#.tar.gz` file.
    5. Once unpacked, `cd` into the `sleuthkit-4.#.#.tar.gz` directory.
    6. Configure and install `sleuthkit` using the commands:
        a. `./configure --disable-java`
        b. `make`
        c. `sudo make install`

```
$ cd ~/Downloads

$ tar -xvf sleuthkit-4.#.#.tar.gz

$ cd sleuthkit-#.#.#

$ ./configure --disable-java

$ make

$ sudo make install

$ mmls -i list
```

5. **exiftool**
    1. Download `ExifTool-9.48.dmg` (or newer) from `http://www.sno.phy.queensu.ca/~phil/exiftool/`.
    2. Open the DMG file, double-click the package installer, and follow the default prompts.

6. **Synalyze It!**
    1. NOTE: Please do not install the trial version of Synalyze It! Pro until you are ready to start Lab 2.1: HFS+.
    2. Download Synalyze It! Pro Trial from `http://www.synalysis.net/downloads/`. This is a 30-day trial; you may also purchase the non-pro version for from the App Store.
    3. If purchased from the App Store, it will install automatically.
    4. If you downloaded the trial, unzip the file and move the application to your `/Applications` directory.

7. **SQLite Database Browser**
    1. Download the latest version of SQLite Database Browser from `http://sqlitebrowser.org/`.
    2. Open the DMG file and drag the SQLite Database Browser application to the `/Applications` directory.

8. **Hex Editors**
    - You may choose your favorite; these are recommended:
        - i. Hex Fiend
            1. Download from `http://ridiculousfish.com/hexfiend/`.
            2. Unzip and move the application to the `/Applications` directory.
        - ii. 0xED
            1. Download from `http://www.suavetech.com/0xed/`.
            2. Open the BZip2 archive by double clicking, then move the application to the `/Applications` directory.

9. **The Unarchiver**
    1. Download The Unarchiver from the Mac App Store or from `http://unarchiver.c3.cx/unarchiver`, under the “Other Links” heading.
    2. Double-click to unzip.
    3. Drag the Unarchiver.app file to the `/Applications` directory.

10. **Homebrew**
    1. Download the Mac package manager Homebrew from `https://brew.sh/`.
    2. This web page will contain a script that you need to copy and paste into your Terminal window.

11. **Volatility**
    1. Change directory back to your home directory using the ‘`cd`’ command.
    2. Download and install Volatility using Homebrew.
    3. Use the brew install command to do this.
        1. `brew install volatility`

```
$ cd ~

$ brew install volatility
```

12. **John the Ripper**
    1. Change the directory back to your home directory using the "`cd`" command.
    2. Download and install John the Ripper using Homebrew.
    3. Use the brew install command to do this.
        1. `brew install john-jumbo`

```
$ cd ~

$ brew install john-jumbo
```

This page intentionally left blank.

Lab 1.0: Lab Setup (In-Class)

Objectives

1. Introduction to FOR518 thumb drives.
2. Unarchive and copy files to analysis system.

Lab Preparation

1. **Software Preparation**: The following tools may be used in this Lab:
   - The Unarchiver.app
     i. Locate "The Unarchiver.app" from /Applications/; if you have not installed this yet, you may find it in the Tools directory on your thumb drive.

Lab

1. **Create a FOR518 directory**
   - The labs for this class will reference an FOR518 folder in the user's home directory to dump various files for use in other labs (`~/FOR518`).
   - Please create a directory named FOR518. You do not have to create it in your home directory but be sure to remember where it is. The workbook used in class will reference this directory in your home directory.
   - The command below shows how to create this folder in your home directory. You may also use the GUI interface to do this.
   - To make this directory more accessible, you can drag and drop it to your Finder sidebar. Using the "open" command, you can open it from Terminal into Finder.
     i. Select the folder icon for the FOR518 directory and drop it into the Finder sidebar in "Favorites."

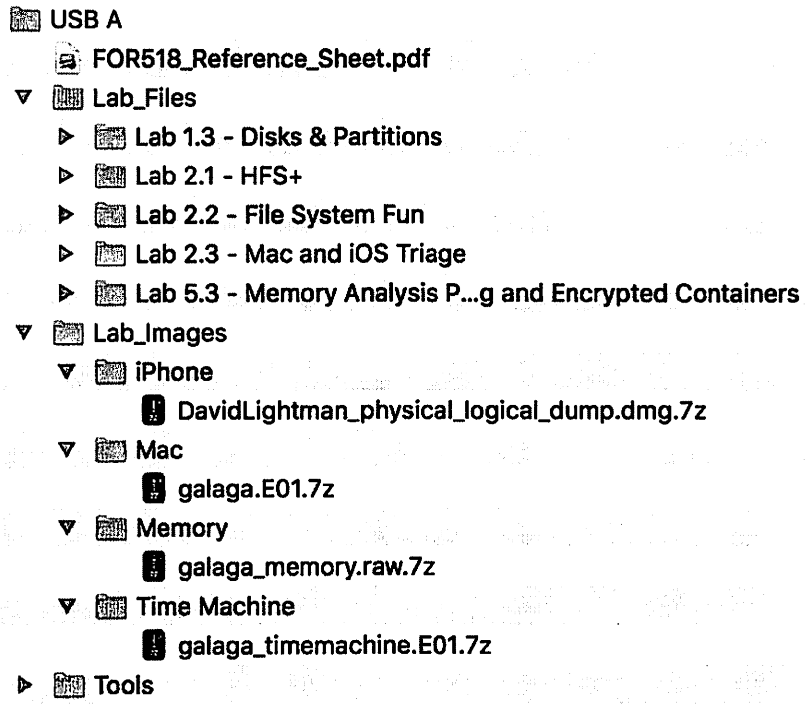
```
$ mkdir ~/FOR518

$ open ~
```

1. **Introduction to the [FOR518 – A] Thumb Drive**
   1. Insert the FOR518 - A thumb drive into your laptop.
   2. View the mounted thumb drive using the Finder application.
   3. The thumb drive has the following directory structure:
      i. **Lab Files**: This directory contains files and software that you will need for the class Labs, listed for each Lab.
      ii. **Lab Images**: This directory contains the forensic images that you will be working with on the Labs. You will need to unarchive these files for this class.

iii. **FOR518 HFS+ Reference Sheet and Command-line Reference PDF (FOR518_Reference_Sheet.pdf)**: This file contains a command line cheat sheet as well as a reference for HFS+ for the class.

iv. **Tools**: This directory contains many of the tools you have already installed plus some extras that can be installed later in the class.

v. **VERSION-FOR518-##-##.txt**: This file contains the MD5 hashes for the 7zip archives as well as for the image files used in this class.



```
USB A
    FOR518_Reference_Sheet.pdf
▽   Lab_Files
    ▷   Lab 1.3 - Disks & Partitions
    ▷   Lab 2.1 - HFS+
    ▷   Lab 2.2 - File System Fun
    ▷   Lab 2.3 - Mac and iOS Triage
    ▷   Lab 5.3 - Memory Analysis P...g and Encrypted Containers
▽   Lab_Images
    ▽   iPhone
            DavidLightman_physical_logical_dump.dmg.7z
    ▽   Mac
            galaga.E01.7z
    ▽   Memory
            galaga_memory.raw.7z
    ▽   Time Machine
            galaga_timemachine.E01.7z
▷   Tools
```

2. **Unarchive**

   1. Unarchive the following items to your host system (or external hard drive) from the Lab_Images directory. You should have installed "The Unarchiver.app" application prior to coming to class in Lab 0. If you have not yet installed it, please do so now. This zip file containing this application can be found in the Tools directory on the FOR518 - A thumb drive. The iPhone and Mac images are needed first, you will not be using the Memory file until Day 5 if you want to wait to unarchive these files. The Time Machine image is part of a bonus lab and is not required to unarchive at this time.

      i. DavidLightman_physical_logical_dump.dmg.7z (Unarchived Size: 13.98GB)
      ii. galaga.E01.7z (Unarchived Size: 14.71GB)
      iii. galaga_memory.raw.7z (Unarchived Size: 19.05GB)
      iv. galaga_timemachine.E01.7z (Unarchived Size: 55.64GB)

3. **Copy out the Lab Files**

   1. Copy the Lab_Files directories to your ~/FOR518 directory.

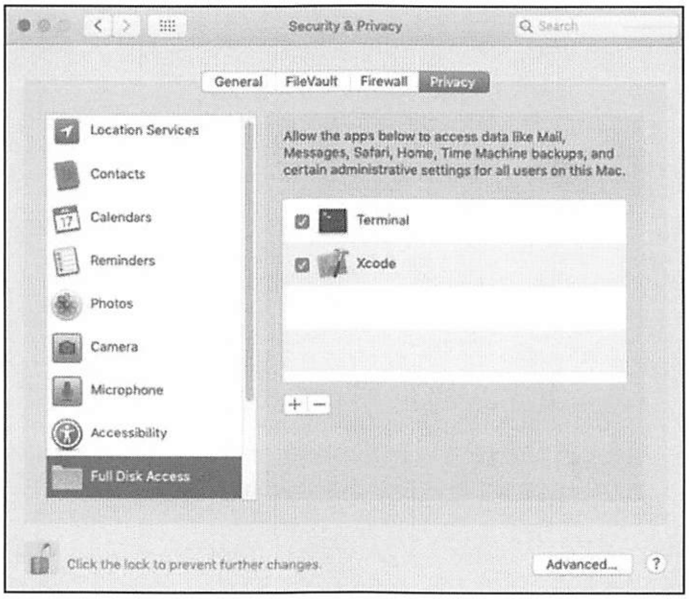
4. **Add license to BlackLight**

   1. Open BlackLight from your `/Applications/BlackLight/BlackLight #### Release #/` directory. You should be presented with a window allowing you to `"Enter`

Demo Key…", your instructor will provide you with a license name and key. Please enter this information where appropriate.

5. **Install BlackBag Epoch Converter**
    1. Open the `epoch_converter.app_.zip` file in the Tools directory on your FOR518 thumb drive.
    2. Copy this app to the `/Applications` directory.

6. **Additional Setting Changes for 10.14+ Users**
    - Users who are using macOS 10.14 and higher will need to configure additional items to allow full disk access for mounted images for labs and the final challenge for this course. The final settings should look like the example below. You may now close the System Preferences application. You may choose to revert these actions at the end of class.
    - Change Privacy Settings:
        1. Open "System Preferences" from the Dock or the Apple Menu at the top-left of the menu bar.
        2. Select the "Security & Privacy" Preferences Panel
        3. Select the "Privacy" tab.
        4. Click the lock icon at the bottom-left of the window, provide the pop-up window Administrator credentials. Select "Unlock"
        5. On the left, select "Full Disk Access"
        6. In the right panel, select the "+" icon to add two Applications. (Adding these Applications may require those applications to be exited, please allow them to be closed.)
            a. /Applications/Utilities/Terminal.app. & /Applications/Xcode.app



Allow the apps below to access data like Mail, Messages, Safari, Home, Time Machine backups, and certain administrative settings for all users on this Mac.

☑ Terminal

☑ Xcode

Click the lock to prevent further changes.

This page intentionally left blank.

Lab 1.1: BlackLight Case Setup and Image Mounting

Objectives

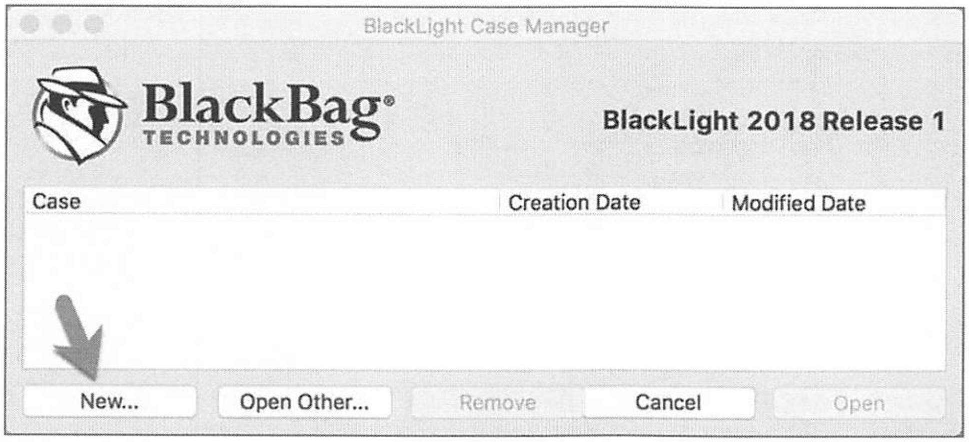
- Import exercise images to BlackLight
- Practice mounting lab images on the command line
- Introduction to BlackLight

Lab Preparation

1. Locate the `galaga.E01` file that was extracted from your FOR518 thumb drive from the `Lab_Images/Mac/` directory.
2. **Software Preparation**: The following tools will be used in this exercise:
   - BlackLight.app
     i. Locate and open `/Applications/BlackLight/BlackLight Release YYYY Release #/BlackLight.app`
   - BlackLight Key: Your instructor should provide this to you.
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from `/Applications/Utilities/`

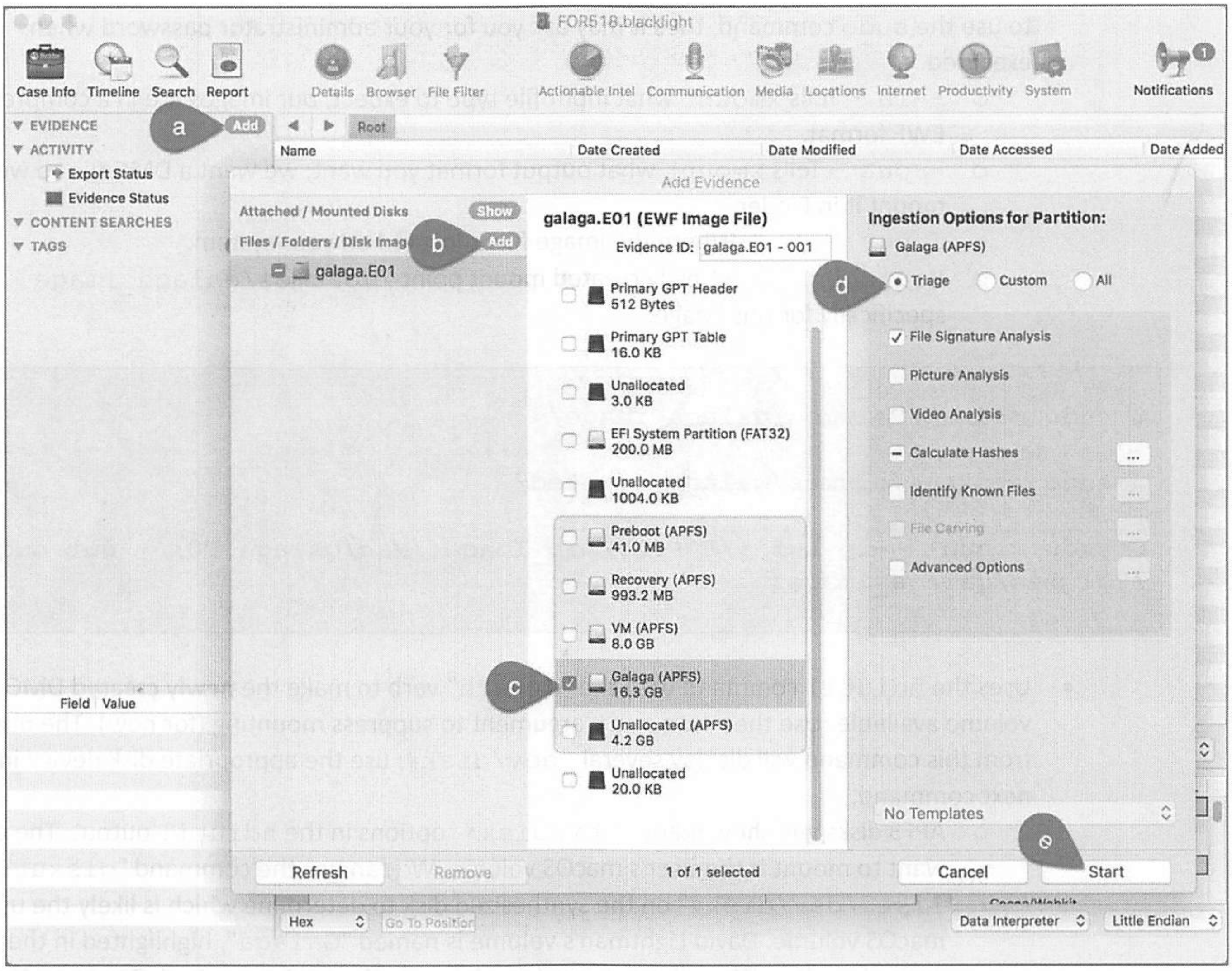
Lab

1. **Load Lab Image in BlackLight**
   1. The first window presented to a user is the Case Manager window. This window will show all your recent cases and allow you to create new ones.
      a. Create a new case. Select the "New..." button at the bottom-left of the window.
      b. Save the case in a directory of your choice. You may want to create a `FOR518` directory for this class, as we will be saving a variety of files for analysis.
      c. Save the case file as `FOR518.BlackLight` in your `FOR518` directory.



2. This should open up the BlackLight Case Info window.
    a. The Case Info tab of BlackLight allows an analyst to input case specifics and change the time zone display. The defaults are fine.

3. Open the Disk Image:
    a. In the upper-left corner near "EVIDENCE," you should see a small green "Add" button. Select this button.
    b. In the "Add Evidence" window, select the green "Add" button to select the image file. Locate the `Lab_Images` directory where you extracted your files, select the `galaga.E01` image and click Open.
    c. This will open the Evidence Selection window. Please de-select the following disks so that only the "Galaga (APFS)" disk is checked (shown below).
        i. EFI System Partition (FAT32)
        ii. Preboot (APFS)
        iii. Recovery (APFS)
        iv. VM (APFS)
    d. Keep the Triage button selected; you can run any additional tasks later if required.
    e. Select "Start." This will start the image processing; this may take a few minutes. The Evidence Status window will show the disk processing progression.



galaga.E01 (EWF Image File)

Evidence ID: galaga.E01 - 001

□	Primary GPT Header 512 Bytes
□	Primary GPT Table 16.0 KB
□	Unallocated 3.0 KB
□	EFI System Partition (FAT32) 200.0 MB
□	Unallocated 1004.0 KB
□	Preboot (APFS) 41.0 MB
□	Recovery (APFS) 993.2 MB
□	VM (APFS) 8.0 GB
☑	Galaga (APFS) 16.3 GB
□	Unallocated (APFS) 4.2 GB
□	Unallocated 20.0 KB

1 of 1 selected

Ingestion Options for Partition:

Galaga (APFS)

● Triage ○ Custom ○ All

- ☑ File Signature Analysis
- □ Picture Analysis
- □ Video Analysis
- ⊟ Calculate Hashes ...
- □ Identify Known Files ...
- □ File Carving ...
- □ Advanced Options ...

No Templates

4. While BlackLight is processing, please move on to mounting the image via the command line below.
   a. Once processing has finished, feel free to browse the disk at your leisure!

2. **Practice Mounting David Lightman's Mac forensic image (`galaga.E01`)**

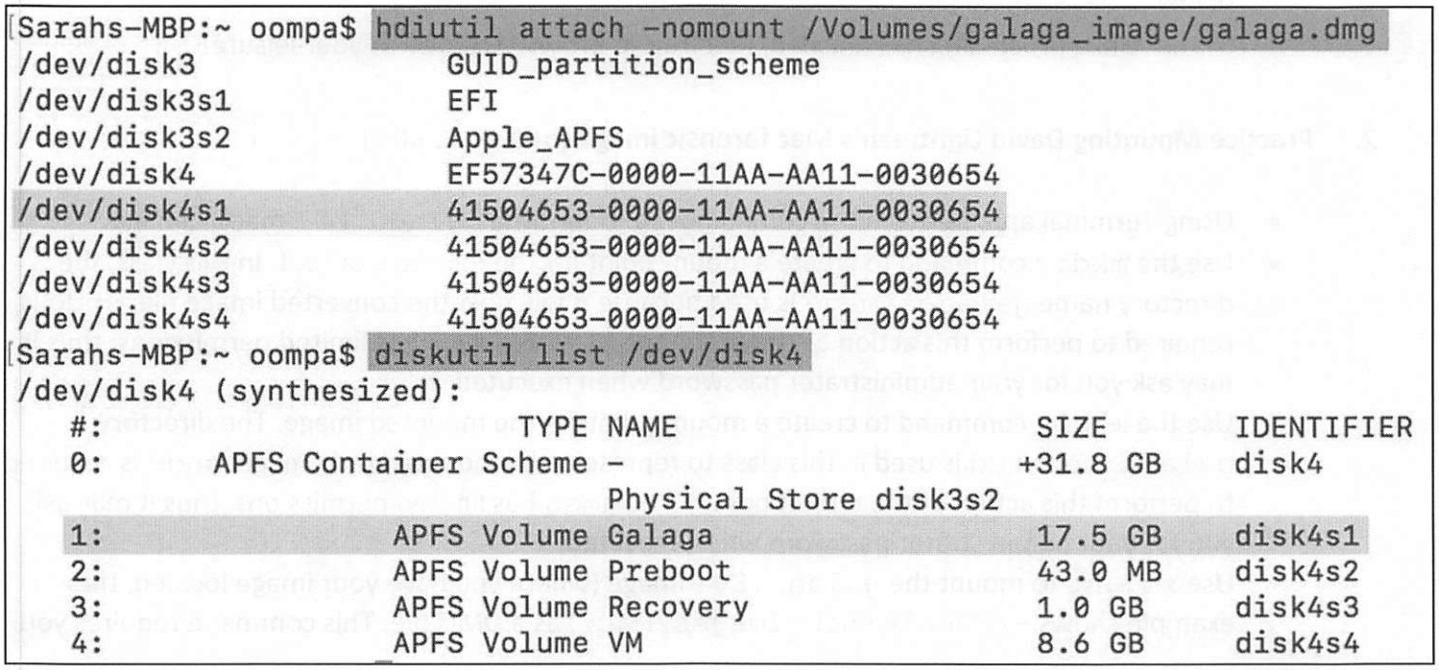
- Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
- Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
- Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
- Use `xmount` to mount the `galaga.E01` image (where you have your image located, the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you

to use the sudo command, thus it may ask you for your administrator password when executed.

  - --in – Tells xmount what input file type to expect; our images are in a compressed EWF format.
  - --out – Tells xmount what output format you want; we want a DMG file so we can mount it in Finder.
  - Input File – Where the image file is located on your system.
  - Mount Point – Newly created mount point /Volumes/galaga_image specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Uses the hdiutil command with the "attach" verb to make the newly created DMG volume available. Use the -nomount argument to suppress mounting (for now). The output from this command will display several /dev/disk#; use the appropriate disk device in the next command.
  - APFS disks will show many /dev/disk* options in the hdiutil output. The one we want to mount is the user's macOS volume. We can use the command "diskutil list /dev/disk4" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "Galaga", highlighted in the example below. We will use /dev/disk4s1 in the next command. **Be aware that yours may be mounted on a different disk number!**



```
[Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3              GUID_partition_scheme
/dev/disk3s1            EFI
/dev/disk3s2            Apple_APFS
/dev/disk4              EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1            41504653-0000-11AA-AA11-0030654
/dev/disk4s2            41504653-0000-11AA-AA11-0030654
/dev/disk4s3            41504653-0000-11AA-AA11-0030654
/dev/disk4s4            41504653-0000-11AA-AA11-0030654
[Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the `mount_apfs` command with the following parameters to mount the `/dev/disk#s#` (from the previous command) to the `/Volumes/galaga_mounted/` mount point. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
  - o `-o` – Options:
    - `rdonly` – Mount in read-only mode.
    - `noexec` – Do not allow execution of binaries on the mounted system.
    - `noowners` – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg


$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

3. **Sanity Check**

- You can access this newly created mounted drive on `/Volumes/galaga_mounted/`, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
- Use the `ls -l` command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "`dlightman`" in the `Users` directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

4. **Unmount and Eject the Exercise Image**

- Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`". In my example, it would be `/dev/disk3`.
- Use the `diskutil eject` command on the disk you would like to eject.
- Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/` if you are following the naming scheme from the examples).
- Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command.

- *****WARNING*****: If you are in the mounted image in Terminal, or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

*****OPTIONAL*****

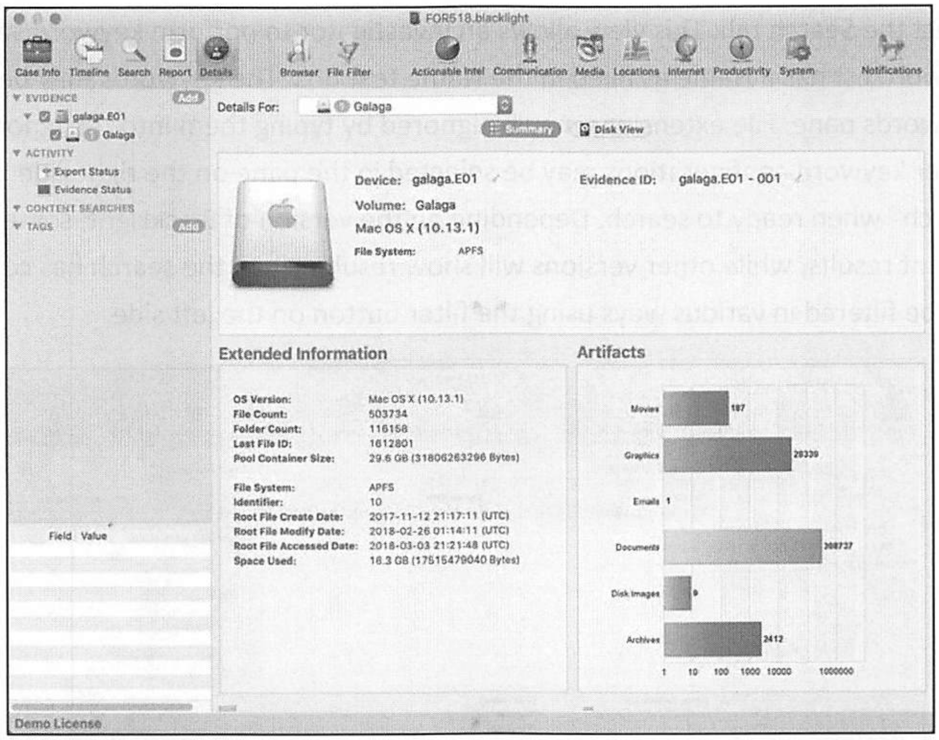
5. **BlackLight 101**

- If you have never used BlackLight before, this section of the lab will give you a beginner overview of the tool.
- Using the BlackLight Case file you just created in the beginning of this lab, let's take a look at some of the features of BlackLight.

6. **Details Tab**

- Check the box next to the `galaga.E01` drive under the EVIDENCE section on the left and select the Details tab at the top. This view will show basic triage information for the disk and the Galaga volume.

- Use the dropdown to switch between the drive and the volume to view different details.



Details For: Galaga

Device: galaga.E01

Evidence ID: galaga.E01 - 001

Volume: Galaga

Mac OS X (10.13.1)

File System: APFS

Extended Information

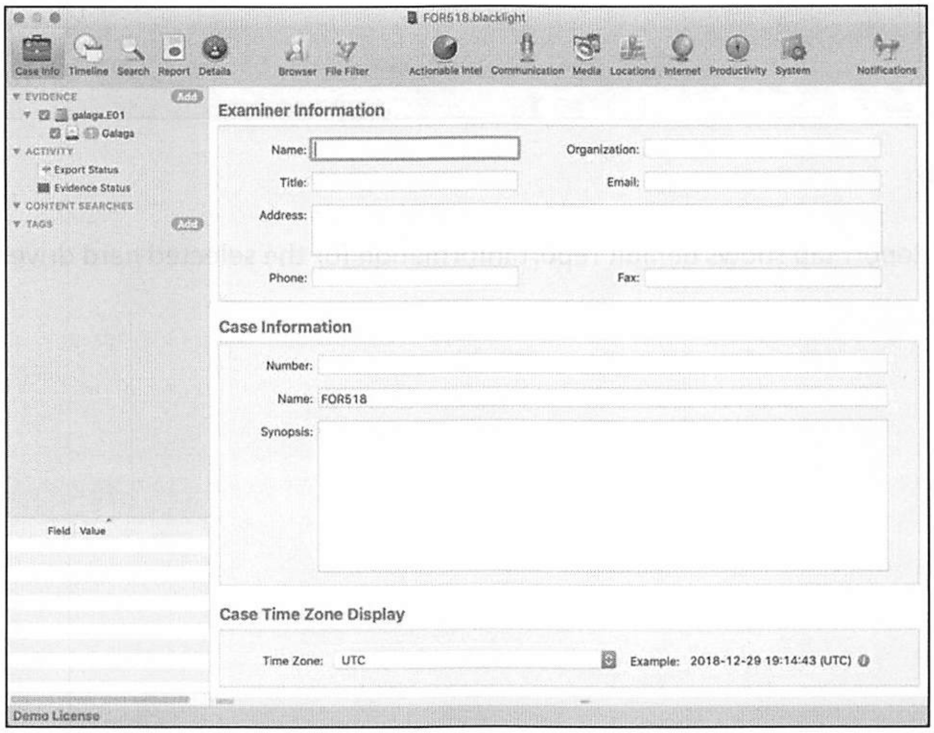
OS Version: Mac OS X (10.13.1)
File Count: 503734
Folder Count: 116158
Last File ID: 1612801
Pool Container Size: 29.6 GB (31806263296 Bytes)

File System: APFS
Identifier: 10
Root File Create Date: 2017-11-12 21:17:11 (UTC)
Root File Modify Date: 2018-02-26 01:14:11 (UTC)
Root File Accessed Date: 2018-03-03 21:21:48 (UTC)
Space Used: 16.3 GB (17515479040 Bytes)

Artifacts

1. **Case Info Tab**

   - Select the Case Info tab, this shows an area where you may fill in case-specific information and change the time zone.



Examiner Information

Name:

Organization:

Title:

Email:

Address:

Phone:

Fax:

Case Information

Number:

Name: FOR518

Synopsis:

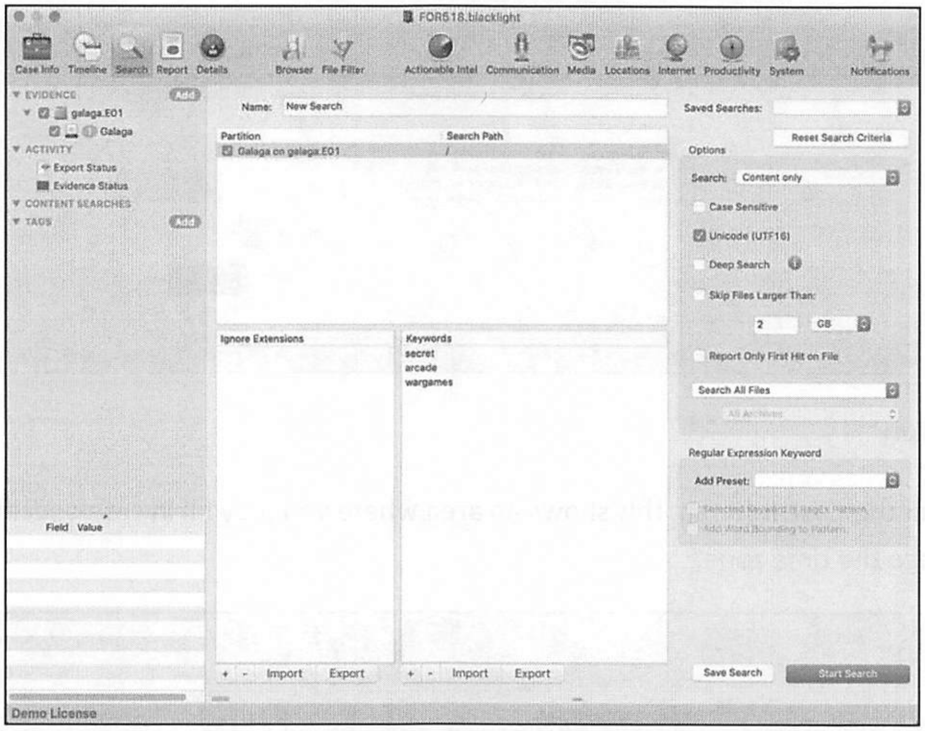
Case Time Zone Display

Time Zone: UTC

Example: 2018-12-29 19:14:43 (UTC)

2. **Search Tab**

- Select the Search tab. This view allows an investigator to perform keyword searches. Each keyword list has a Name as filled in the Name text box. The keywords may be typed in the Keywords pane. File extensions may be ignored by typing them into the Ignore Extensions pane. Other keyword configurations may be selected in the pane on the right side. Select "Start Search" when ready to search. Depending on the version of BlackLight, some searches will show instant results, while other versions will show results when the search has completed. Results can be filtered in various ways using the filter button on the left side.



☑ galaga.E01
☑ Galaga

Name: New Search

Partition	Search Path
☑ Galaga on galaga.E01	/

Ignore Extensions

Keywords
secret
arcade
wargames

Saved Searches:

Options

Search: Content only

☐ Case Sensitive

☑ Unicode (UTF16)

☐ Deep Search

☐ Skip Files Larger Than: 2 GB

☐ Report Only First Hit on File

Search All Files

All Archives

Regular Expression Keyword

Add Preset:

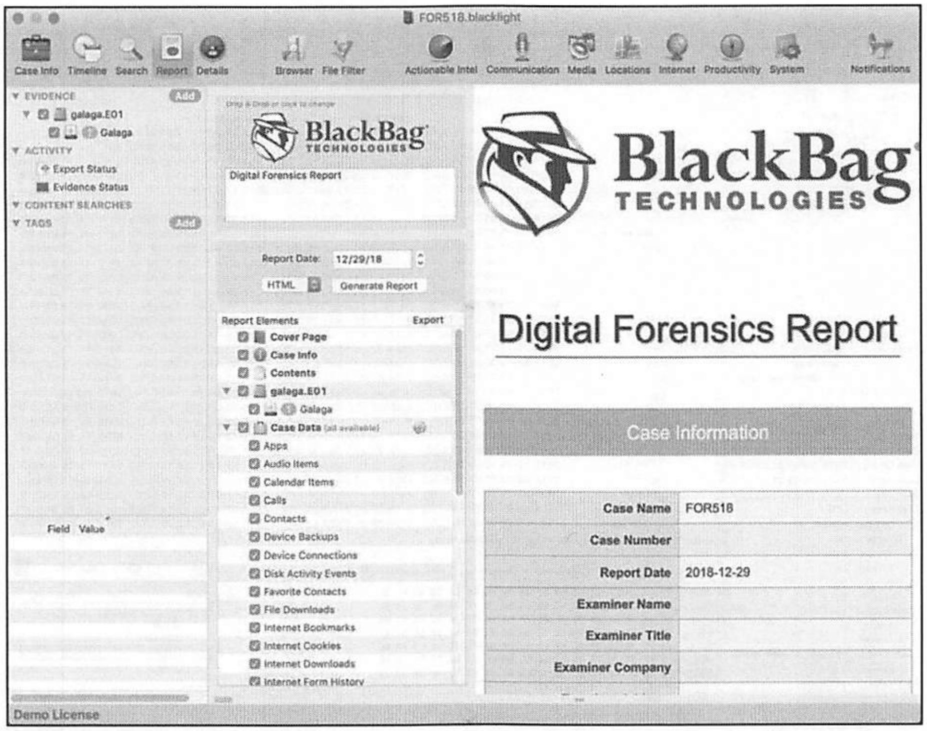
☐ Selected Keyword is RegEx Pattern

☐ Add Word Bounding to Pattern

Field Value

3. **Report Tab**

- The Report tab shows default report information for the selected hard drive.



Digital Forensics Report

Report Date: 12/29/18

HTML | Generate Report

Report Elements — Export

- ☑ Cover Page
- ☑ Case Info
- ☑ Contents
- ☑ galaga.E01
  - ☑ Galaga
- ☑ Case Data (all available)
  - ☑ Apps
  - ☑ Audio Items
  - ☑ Calendar Items
  - ☑ Calls
  - ☑ Contacts
  - ☑ Device Backups
  - ☑ Device Connections
  - ☑ Disk Activity Events
  - ☑ Favorite Contacts
  - ☑ File Downloads
  - ☑ Internet Bookmarks
  - ☑ Internet Cookies
  - ☑ Internet Downloads
  - ☑ Internet Form History

BlackBag TECHNOLOGIES

Digital Forensics Report

Case Information

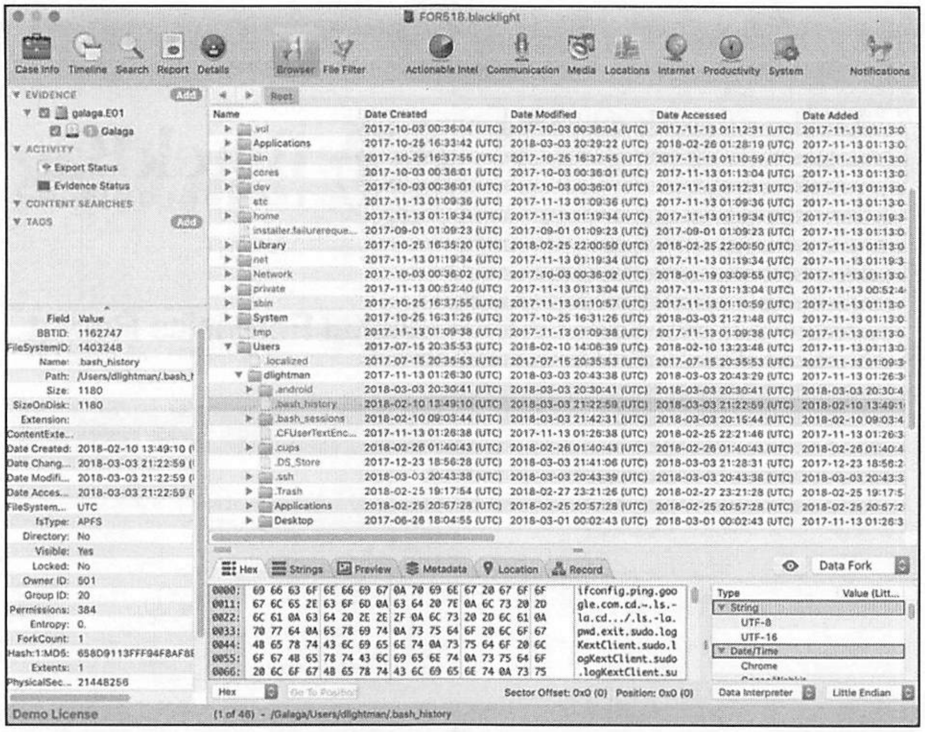
Case Name	FOR518
Case Number	
Report Date	2018-12-29
Examiner Name	
Examiner Title	
Examiner Company	

4. **Browser Tab**

- The Browser tab shows the file system as an investigator is most likely used to seeing it. This view shows the file system in a tree format with hidden files shown in gray, and other file metadata including timestamps and file size.

- The lower pane (the bar may have to be moved up from the bottom of the window) shows the file. The views available include Hex, Strings, Preview, Metadata, Location, and Record. An analyst may also select the "eye"-shaped button to do a "Quick Look" on the file. The Data and Resource fork may also be chosen. In the Hex view, the data-type window on the right will be shown for the analyst to select various data types, if conversion is needed.

- In the lower-left pane, the file metadata and extended attributes are shown. Everything from file size, filename, timestamps, Finder data, and disk location, to extended attributes are available in this window. Lots of good information may be found here!



Name	Date Created	Date Modified	Date Accessed	Date Added
.vol	2017-10-03 00:36:04 (UTC)	2017-10-03 00:36:04 (UTC)	2017-11-13 01:12:31 (UTC)	2017-11-13 01:13:0
Applications	2017-10-25 16:33:42 (UTC)	2018-03-03 20:29:22 (UTC)	2018-02-26 01:28:19 (UTC)	2017-11-13 01:13:0
bin	2017-10-25 16:37:55 (UTC)	2017-10-25 16:37:55 (UTC)	2017-11-13 01:10:59 (UTC)	2017-11-13 01:13:0
cores	2017-10-03 00:36:01 (UTC)	2017-10-03 00:36:01 (UTC)	2017-11-13 01:13:04 (UTC)	2017-11-13 01:13:0
dev	2017-10-03 00:36:01 (UTC)	2017-10-03 00:36:01 (UTC)	2017-11-13 01:12:31 (UTC)	2017-11-13 01:13:0
etc	2017-11-13 01:09:36 (UTC)	2017-11-13 01:09:36 (UTC)	2017-11-13 01:09:36 (UTC)	2017-11-13 01:13:0
home	2017-11-13 01:19:34 (UTC)	2017-11-13 01:19:34 (UTC)	2017-11-13 01:19:34 (UTC)	2017-11-13 01:19:3
installer.failurereque...	2017-09-01 01:09:23 (UTC)	2017-09-01 01:09:23 (UTC)	2017-09-01 01:09:23 (UTC)	2017-11-13 01:13:0
Library	2017-10-25 16:35:20 (UTC)	2018-02-25 22:00:50 (UTC)	2018-02-25 22:00:50 (UTC)	2017-11-13 01:13:0
net	2017-11-13 01:19:34 (UTC)	2017-11-13 01:19:34 (UTC)	2017-11-13 01:19:34 (UTC)	2017-11-13 01:19:3
Network	2017-10-03 00:36:02 (UTC)	2017-10-03 00:36:02 (UTC)	2018-01-19 03:09:55 (UTC)	2017-11-13 01:13:0
private	2017-11-13 00:52:40 (UTC)	2017-11-13 01:13:04 (UTC)	2017-11-13 01:13:04 (UTC)	2017-11-13 00:52:4
sbin	2017-10-25 16:37:55 (UTC)	2017-11-13 01:10:57 (UTC)	2017-11-13 01:10:59 (UTC)	2017-11-13 01:13:0
System	2017-10-25 16:31:26 (UTC)	2017-10-25 16:31:26 (UTC)	2018-03-03 21:21:48 (UTC)	2017-11-13 01:13:0
tmp	2017-11-13 01:09:38 (UTC)	2017-11-13 01:09:38 (UTC)	2017-11-13 01:09:38 (UTC)	2017-11-13 01:13:0
Users	2017-07-15 20:35:53 (UTC)	2018-02-10 14:06:39 (UTC)	2018-02-10 13:23:48 (UTC)	2017-11-13 01:13:0
.localized	2017-07-15 20:35:53 (UTC)	2017-07-15 20:35:53 (UTC)	2017-07-15 20:35:53 (UTC)	2017-11-13 01:09:3
dlightman	2017-11-13 01:26:30 (UTC)	2018-03-03 20:43:38 (UTC)	2018-03-03 20:43:29 (UTC)	2017-11-13 01:26:3
.android	2018-03-03 20:30:41 (UTC)	2018-03-03 20:30:41 (UTC)	2018-03-03 20:30:41 (UTC)	2018-03-03 20:30:4
.bash_history	2018-02-10 13:49:10 (UTC)	2018-03-03 21:22:59 (UTC)	2018-03-03 21:22:59 (UTC)	2018-02-10 13:49:1
.bash_sessions	2018-02-10 09:03:44 (UTC)	2018-03-03 21:42:31 (UTC)	2018-03-03 20:15:44 (UTC)	2018-02-10 09:03:4
.CFUserTextEnc...	2017-11-13 01:26:38 (UTC)	2017-11-13 01:26:38 (UTC)	2018-02-25 22:21:46 (UTC)	2017-11-13 01:26:3
.cups	2018-02-26 01:40:43 (UTC)	2018-02-26 01:40:43 (UTC)	2018-02-26 01:40:43 (UTC)	2018-02-26 01:40:4
.DS_Store	2017-12-23 18:56:28 (UTC)	2018-03-03 21:41:06 (UTC)	2018-03-03 21:28:31 (UTC)	2017-12-23 18:56:2
.ssh	2018-03-03 20:43:38 (UTC)	2018-03-03 20:43:39 (UTC)	2018-03-03 20:43:38 (UTC)	2018-03-03 20:43:3
.Trash	2018-02-25 19:17:54 (UTC)	2018-02-27 23:21:26 (UTC)	2018-02-27 23:21:28 (UTC)	2018-02-25 19:17:5
Applications	2018-02-25 20:57:28 (UTC)	2018-02-25 20:57:28 (UTC)	2018-02-25 20:57:28 (UTC)	2018-02-25 20:57:2
Desktop	2017-06-28 18:04:55 (UTC)	2018-03-01 00:02:43 (UTC)	2018-03-01 00:02:43 (UTC)	2017-11-13 01:26:3

Field	Value
BBTID:	1162747
FileSystemID:	1403248
Name:	.bash_history
Path:	/Users/dlightman/.bash_h
Size:	1180
SizeOnDisk:	1180
Extension:	
ContentExte...	
Date Created:	2018-02-10 13:49:10 (
Date Chang...	2018-03-03 21:22:59 (
Date Modifi...	2018-03-03 21:22:59 (
Date Acces...	2018-03-03 21:22:59 (
FileSystem...	UTC
fsType:	APFS
Directory:	No
Visible:	Yes
Locked:	No
Owner ID:	501
Group ID:	20
Permissions:	384
Entropy:	0.
ForkCount:	1
Hash:1:MD5:	658D9113FFF94F8AF8E
Extents:	1
PhysicalSec...	21448256

```
0000:  69 66 63 6F 6E 66 69 67 0A 70 69 6E 67 20 67 6F 6F   ifconfig.ping.goo
0011:  67 6C 65 2E 63 6F 6D 0A 63 64 20 7E 0A 6C 73 20 2D   gle.com.cd.~.ls.-
0022:  6C 61 0A 63 64 20 2E 2E 2F 0A 6C 73 20 2D 6C 61 0A   la.cd.../.ls.-la.
0033:  70 77 64 0A 65 78 69 74 0A 73 75 64 6F 20 6C 6F 67   pwd.exit.sudo.log
0044:  4B 65 78 74 43 6C 69 65 6E 74 0A 73 75 64 6F 20 6C   KextClient.sudo.l
0055:  6F 67 4B 65 78 74 43 6C 69 65 6E 74 0A 73 75 64 6F   ogKextClient.sudo
0066:  20 6C 6F 67 4B 65 78 74 43 6C 69 65 6E 74 0A 73 75   .logKextClient.su
```

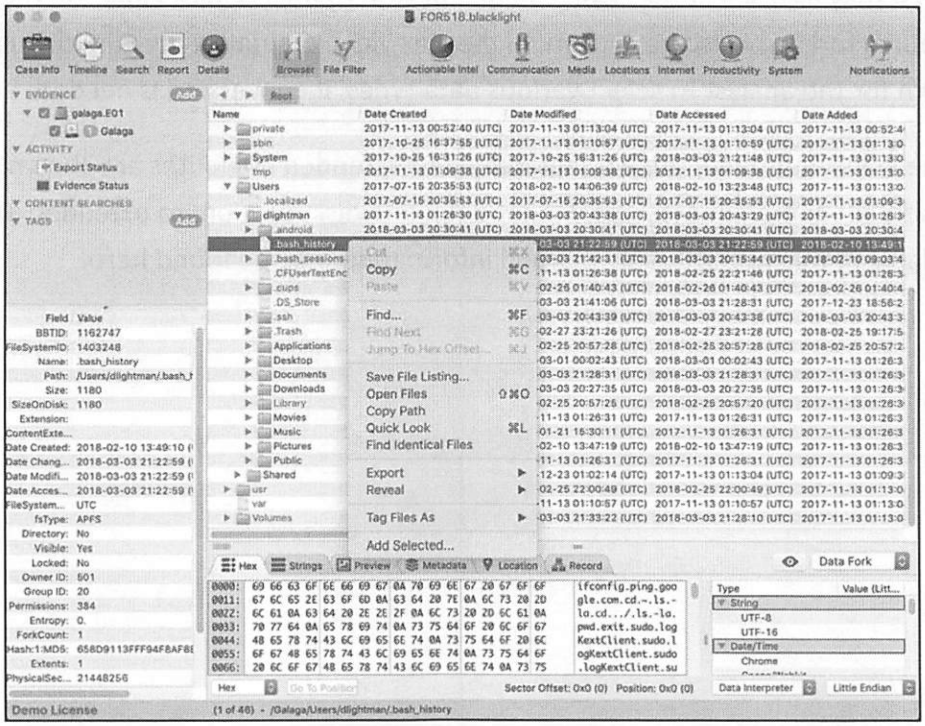
Hex | Sector Offset: 0x0 (0) Position: 0x0 (0)

(1 of 46) - /Galaga/Users/dlightman/.bash_history

5. **Context Menu**

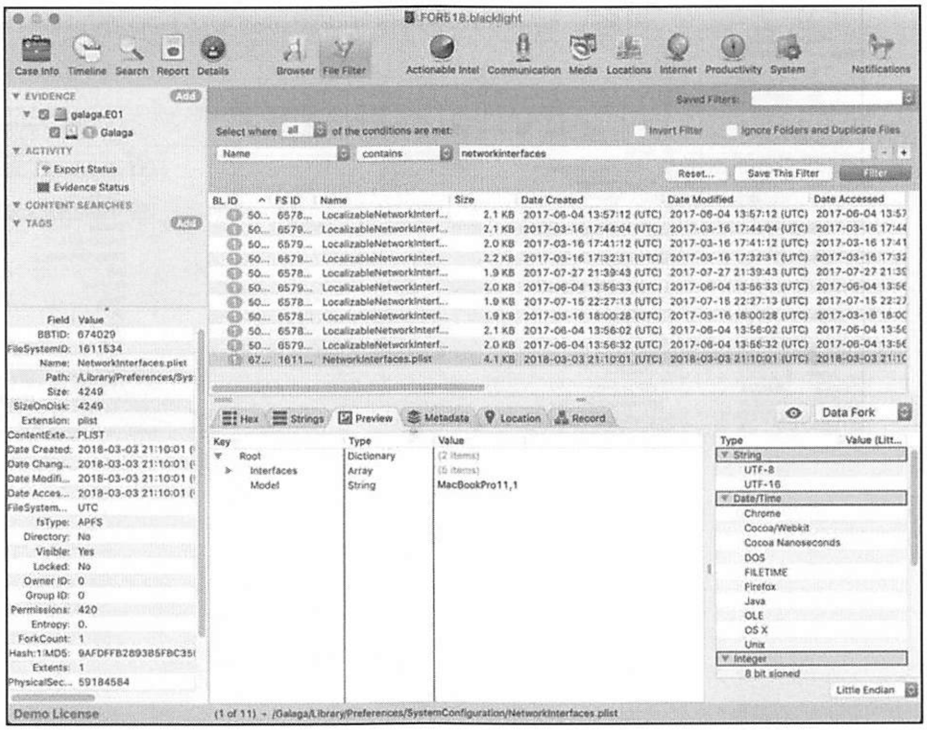
- A "right-click" (or two-finger click on a track pad, or control-click) will bring up a context menu as shown below. This menu allows the analyst to perform certain actions such as export information, jump to a file location, or tag a file.

- File tagging is similar to bookmarking as seen in other forensic suites. These tags will show up in the left pane under "TAGS".



6. **File Filter Tab**

- The File Filter tab allows an investigator to select certain files based on some type of data, whether it is size, file type, or by creation date. Many different combinations can be played with. The course author highly recommends spending some time with this feature. It can help you pinpoint specific files quickly.



Select where all of the conditions are met: □ Invert Filter   □ Ignore Folders and Duplicate Files

Name | contains | networkinterfaces

BL ID	FS ID	Name	Size	Date Created	Date Modified	Date Accessed
50...	6578...	LocalizableNetworkInterf...	2.1 KB	2017-06-04 13:57:12 (UTC)	2017-06-04 13:57:12 (UTC)	2017-06-04 13:57
50...	6579...	LocalizableNetworkInterf...	2.1 KB	2017-03-16 17:44:04 (UTC)	2017-03-16 17:44:04 (UTC)	2017-03-16 17:44
50...	6579...	LocalizableNetworkInterf...	2.0 KB	2017-03-16 17:41:12 (UTC)	2017-03-16 17:41:12 (UTC)	2017-03-16 17:41
50...	6579...	LocalizableNetworkInterf...	2.2 KB	2017-03-16 17:32:31 (UTC)	2017-03-16 17:32:31 (UTC)	2017-03-16 17:32
50...	6578...	LocalizableNetworkInterf...	1.9 KB	2017-07-27 21:39:43 (UTC)	2017-07-27 21:39:43 (UTC)	2017-07-27 21:39
50...	6579...	LocalizableNetworkInterf...	2.0 KB	2017-06-04 13:56:33 (UTC)	2017-06-04 13:56:33 (UTC)	2017-06-04 13:56
50...	6578...	LocalizableNetworkInterf...	1.9 KB	2017-07-15 22:27:13 (UTC)	2017-07-15 22:27:13 (UTC)	2017-07-15 22:27
50...	6578...	LocalizableNetworkInterf...	1.9 KB	2017-03-16 18:00:28 (UTC)	2017-03-16 18:00:28 (UTC)	2017-03-16 18:00
50...	6578...	LocalizableNetworkInterf...	2.1 KB	2017-06-04 13:56:02 (UTC)	2017-06-04 13:56:02 (UTC)	2017-06-04 13:56
50...	6579...	LocalizableNetworkInterf...	2.0 KB	2017-06-04 13:56:32 (UTC)	2017-06-04 13:56:32 (UTC)	2017-06-04 13:56
67...	1611...	NetworkInterfaces.plist	4.1 KB	2018-03-03 21:10:01 (UTC)	2018-03-03 21:10:01 (UTC)	2018-03-03 21:10

Field	Value
BBTID:	674029
FileSystemID:	1611534
Name:	NetworkInterfaces.plist
Path:	/Library/Preferences/Sys
Size:	4249
SizeOnDisk:	4249
Extension:	plist
ContentExte...	PLIST
Date Created:	2018-03-03 21:10:01 (
Date Chang...	2018-03-03 21:10:01 (
Date Modifi...	2018-03-03 21:10:01 (
Date Acces...	2018-03-03 21:10:01 (
FileSystem...	UTC
fsType:	APFS
Directory:	No
Visible:	Yes
Locked:	No
Owner ID:	0
Group ID:	0
Permissions:	420
Entropy:	0.
ForkCount:	1
Hash:1:MD5:	9AFDFFB289385FBC35(
Extents:	1
PhysicalSec...	59184584

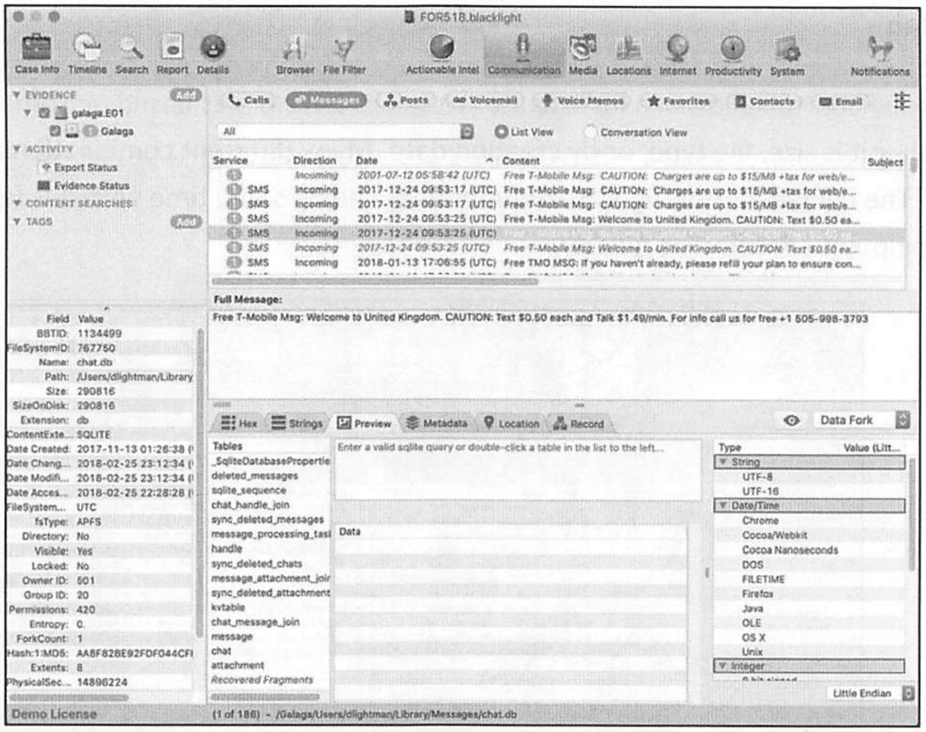
Key	Type	Value
Root	Dictionary	(2 items)
Interfaces	Array	(5 items)
Model	String	MacBookPro11,1

Type	Value (Litt...
String	
UTF-8	
UTF-16	
Date/Time	
Chrome	
Cocoa/Webkit	
Cocoa Nanoseconds	
DOS	
FILETIME	
Firefox	
Java	
OLE	
OS X	
Unix	
Integer	
8 bit signed	

(1 of 11) → /Galaga/Library/Preferences/SystemConfiguration/NetworkInterfaces.plist

7. **Artifacts Tabs**

- BlackLight also does some pre-processing when it comes to various popular artifacts. We will be reviewing some of these more in depth during the course labs, but you may start reviewing the contents when you get a free moment.



Service	Direction	Date	Content	Subject
	Incoming	*2001-07-12 05:58:42 (UTC)*	*Free T-Mobile Msg: CAUTION: Charges are up to $15/MB +tax for web/e...*	
SMS	Incoming	2017-12-24 09:53:17 (UTC)	Free T-Mobile Msg: CAUTION: Charges are up to $15/MB +tax for web/e...	
SMS	Incoming	2017-12-24 09:53:17 (UTC)	Free T-Mobile Msg: CAUTION: Charges are up to $15/MB +tax for web/e...	
SMS	Incoming	2017-12-24 09:53:25 (UTC)	Free T-Mobile Msg: Welcome to United Kingdom. CAUTION: Text $0.50 ea...	
SMS	Incoming	2017-12-24 09:53:25 (UTC)		
SMS	*Incoming*	*2017-12-24 09:53:25 (UTC)*	*Free T-Mobile Msg: Welcome to United Kingdom. CAUTION: Text $0.50 ea...*	
SMS	Incoming	2018-01-13 17:06:55 (UTC)	Free TMO MSG: If you haven't already, please refill your plan to ensure con...	

Full Message:

Free T-Mobile Msg: Welcome to United Kingdom. CAUTION: Text $0.50 each and Talk $1.49/min. For info call us for free +1 505-998-3793

Field	Value
BBTID:	1134499
FileSystemID:	767750
Name:	chat.db
Path:	/Users/dlightman/Library
Size:	290816
SizeOnDisk:	290816
Extension:	db
ContentExte...	SQLITE
Date Created:	2017-11-13 01:26:38 (
Date Chang...	2018-02-25 23:12:34 (
Date Modifi...	2018-02-25 23:12:34 (
Date Acces...	2018-02-25 22:28:28 (
FileSystem...	UTC
fsType:	APFS
Directory:	No
Visible:	Yes
Locked:	No
Owner ID:	501
Group ID:	20
Permissions:	420
Entropy:	0.
ForkCount:	1
Hash:1:MD5:	AA8F828E92FDF044CFI
Extents:	8
PhysicalSec...	14896224

Tables
_SqliteDatabasePropertie
deleted_messages
sqlite_sequence
chat_handle_join
sync_deleted_messages
message_processing_tasl
handle
sync_deleted_chats
message_attachment_joir
sync_deleted_attachment
kvtable
chat_message_join
message
chat
attachment
Recovered Fragments

Enter a valid sqlite query or double-click a table in the list to the left...

Data

Type	Value (Litt...
▼ String	
UTF-8	
UTF-16	
▼ Date/Time	
Chrome	
Cocoa/Webkit	
Cocoa Nanoseconds	
DOS	
FILETIME	
Firefox	
Java	
OLE	
OS X	
Unix	
▼ Integer	

(1 of 186) - /Galaga/Users/dlightman/Library/Messages/chat.db

Lab 1.2: Exploring iOS Acquisitions

Objectives

- Review the different types of iOS acquisitions, including backups and physical/logical acquisition
- Perform an initial triage analysis

Lab Preparation

1. **Software Preparation**: The following tools will be used in this exercise:
    - Terminal.app
        1. You will be using the native OS X Terminal application for this lab.
        2. Locate and open the Terminal.app from /Applications/Utilities/
    - BlackLight.app

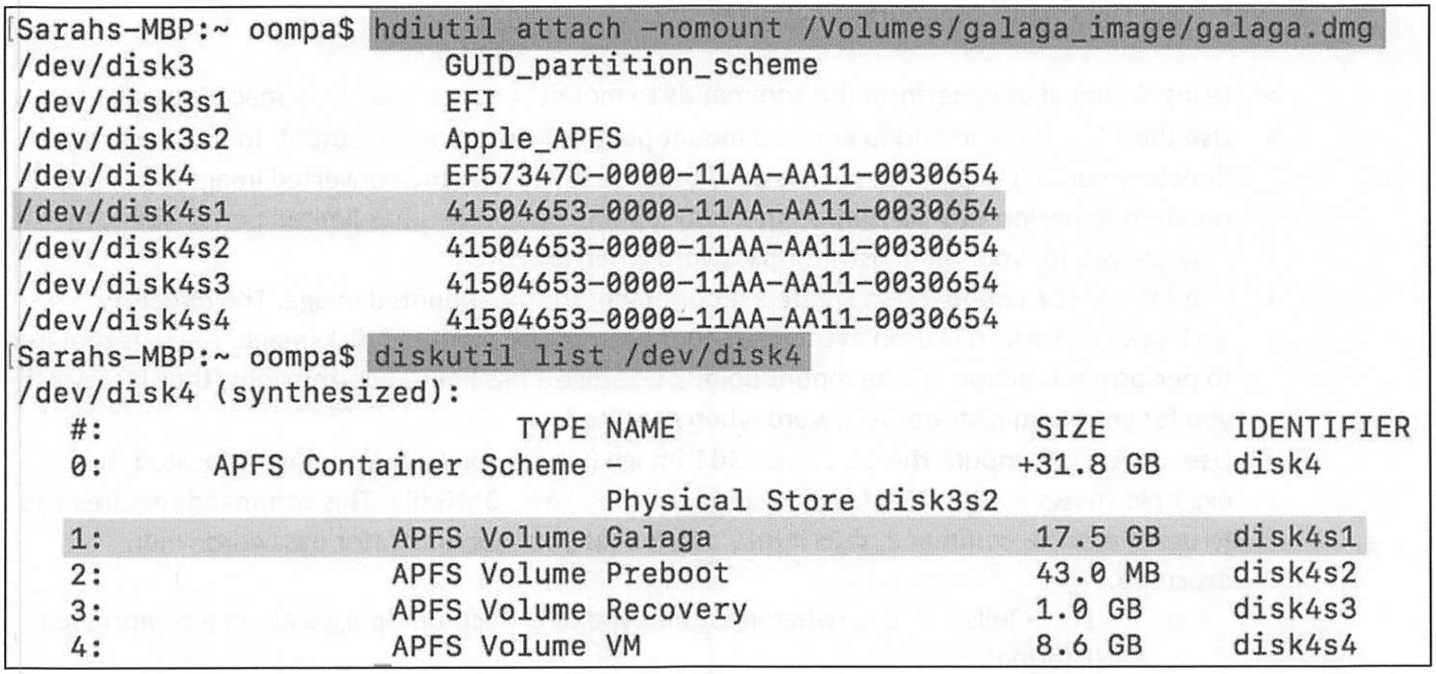
2. **Exercise File Preparation**:
    - Locate the files located in the `Lab_Images/iPhone/` directory on your FOR518 USB drive.

3. **Mount David Lightman's Mac Forensic Image (`galaga.E01`)**

    - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
    - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
    - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
    - Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.
        - `--in` – Tells `xmount` what input file type to expect; our images are in a compressed EWF format.
        - `--out` – Tells `xmount` what output format you want; we want a DMG file so we can mount it in Finder.
        - `Input File` – Where the image file is located on your system.
        - `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Use the `hdiutil` command with the "`attach`" verb to make the newly created DMG volume available. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display several `/dev/disk#` entries; use the appropriate disk device in the next command.
  - APFS disks will show many `/dev/disk*` options in the `hdiutil` output. The one we want to mount is the user's macOS volume. We can use the command "`diskutil list /dev/disk4`" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "`Galaga`," highlighted in the example below. We will use `/dev/disk4s1` in the next command. **Be aware that yours may be mounted on a different disk number!**



```
Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3              GUID_partition_scheme
/dev/disk3s1            EFI
/dev/disk3s2            Apple_APFS
/dev/disk4              EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1            41504653-0000-11AA-AA11-0030654
/dev/disk4s2            41504653-0000-11AA-AA11-0030654
/dev/disk4s3            41504653-0000-11AA-AA11-0030654
/dev/disk4s4            41504653-0000-11AA-AA11-0030654
Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the `mount_apfs` command with the following parameters to mount the `/dev/disk#s#` (from the previous command) to the `/Volumes/galaga_mounted/` mount point. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
  - `-o` – Options:
    - `rdonly` – Mount in read-only mode.
    - `noexec` – Do not allow execution of binaries on mounted system.

        - noowners – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

4. **Sanity Check**
    - You can access this newly created mounted drive on /Volumes/galaga_mounted/, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
    - Use the ls -l command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "dlightman" in the Users directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

5. *****When Needed***: Image Unmount Instructions**
    - Use the diskutil list command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "(disk image)" versus the one labeled "(synthesized)". In my example, it would be /dev/disk3.
    - Use the diskutil eject command on the disk you would like to eject.
    - Use the mount command to view the list of mounted disks. Find the disk that you want to unmount (likely /Volumes/galaga_image/, if you are following the naming scheme from the examples).
    - Use the umount command with the mount point to unmount the disk. You will have to use the sudo command.
    - *****WARNING***: If you are in the mounted image in Terminal or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

Lab: Questions

1. **Review iOS Files from David's macOS Disk Image**

   - In David's Mac mounted image, navigate to and open the file `/Volumes/galaga_mounted/Users/dlightman/Library/Preferences/com.apple.iPod.plist` and answer the following questions. Note the identifiers: IMEI and Serial Number.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Preferences/

$ open com.apple.iPod.plist
```

1. How many times was an iPhone connected to this system while using the "dlightman" user account?

   ____

2. When was this iPhone last connected (UTC)?

   ____

3. What was the iOS version of the iPhone when it was last connected?

   ____

4. What is the "human-conversion" make and model of the iPhone (i.e., iPhone X, iPhone 6S+)?

   ____

- Navigate to the MobileSync backup directory on David's Mac, located here: `/Volumes/galaga_mounted/Users/dlightman/Library/Application\ Support/MobileSync/Backup/`.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Application\
Support/MobileSync/Backup/

$ ls -la
```

5. What are the first few characters of the Universal Device Identifier (UDID) for this backup?

   ____

- Navigate into this iOS backup, review the backup structure, and open the plist metadata files.

```
$ cd 01bdc468ee1e1f0bc186d7992314dbe7fdb168ac

$ ls -la

$ open *.plist
```

- Review the contents of the `Status.plist` file.
- Review the contents of the `Info.plist` file.

6. What is the date of this backup (UTC)?

7. What is the name of this iPhone?

8. What was the phone number of this device when it was backed up?

- Review the contents of the `Manifest.plist` file.

9. Was there a passcode set on the device at the time of backup?

10. Is this backup encrypted or not?

- Navigate to the lockdown directory for this system, `/Volumes/galaga_mounted/private/var/db/lockdown/`. Open the lockdown file/pairing certificate for the connected iPhone.

```
$ cd /Volumes/galaga_mounted/private/var/db/lockdown/

$ ls -la

$ open 01bdc468ee1e1f0bc186d7992314dbe7fdb168ac.plist
```

11. What is the Wi-Fi MAC Address for the connected iPhone?

2. Extract and Analyze iOS Backup Files from David's macOS Disk Image

- Navigate to David Lightman's `Documents` directory. David was smart enough to back up his backups before he jailbroke his iPhone. He created an encrypted and unencrypted version of his iPhone and stored them in the iPhone_Backups directory (/Volumes/galaga_mounted/Users/dlightman/Documents/iPhone_Backups/).

- In the unencrypted backup (/Volumes/galaga_mounted/Users/dlightman/Documents/iPhone_Backups/unencrypted_iPhone_backup_01bdc468ee1e1f0bc186d7992314dbe7fdb168ac), Copy the Manifest.db file to your ~/FOR518 directory and review the contents of the Manifest.db file. If you look for this file in the encrypted backups, you'll find that it is… encrypted!
- We are copying out this file because sqlite3 (via command line) and DB Browser for SQLite application cannot open the file on a read-only volume. This is something you will have to do quite often to access the SQLite databases outside of another application like BlackLight.
- Use whichever tool you prefer to open the Manifest.db database and review its contents.

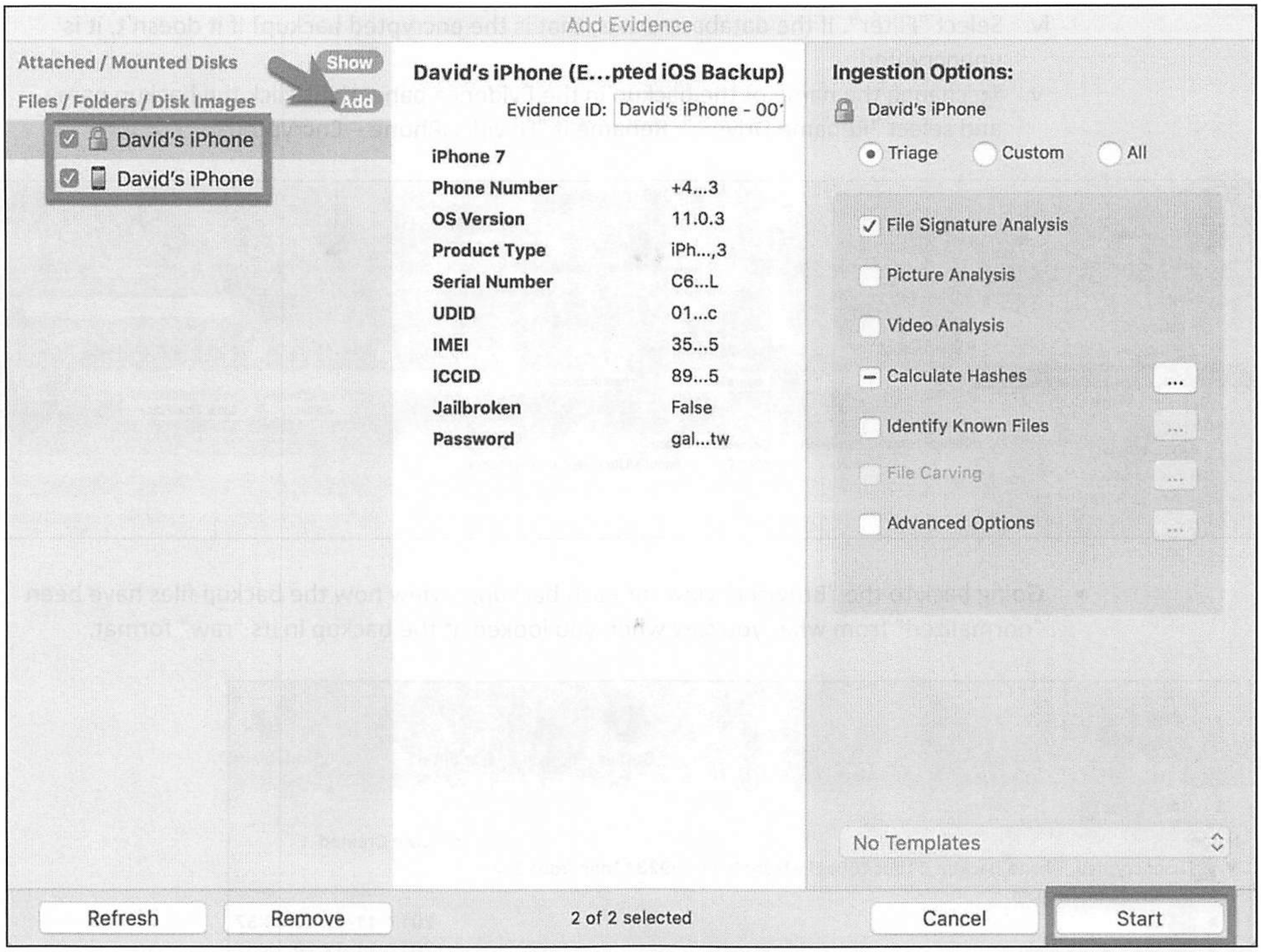
```
$ cd /Volumes/galaga_mounted/Users/dlightman/Documents/iPhone_Backups/
unencrypted_iPhone_backup_01bdc468ee1e1f0bc186d7992314dbe7fdb168ac/

$ cp Manifest.db ~/FOR518
```

1. In the "Files" table, what is the fileID hash of the sms.db database?

- Extract both backups to your ~/FOR518 directory.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Documents/

$ cp -R iPhone_Backups/ ~/FOR518
```

- Import these backups into your FOR518 BlackLight Case File.
  - i. Select the "Add" Evidence button.
  - ii. Select the "Add" button under "Files/Folders/Disk Images".
  - iii. Navigate to where you saved the backups (~/FOR518).
  - iv. Select each backup directory (separately). For the backup labeled "encrypted," you'll have to input the backup password by double-clicking the lock icon in the middle pane. The backup password is "galagaftw". Please be patient; as stated in the pop-up window, this will take a bit of time.
  - v. Once each backup is imported into the "Add Evidence" window, select both backups, keep the default "Ingestion Options" and select "Start" to add them to the Case File.



Add Evidence

Attached / Mounted Disks — Show

Files / Folders / Disk Images — Add

- ☑ David's iPhone
- ☑ David's iPhone

David's iPhone (E…pted iOS Backup)

Evidence ID: David's iPhone - 00'

iPhone 7	
Phone Number	+4...3
OS Version	11.0.3
Product Type	iPh...,3
Serial Number	C6...L
UDID	01...c
IMEI	35...5
ICCID	89...5
Jailbroken	False
Password	gal...tw

Ingestion Options:

David's iPhone

● Triage ○ Custom ○ All

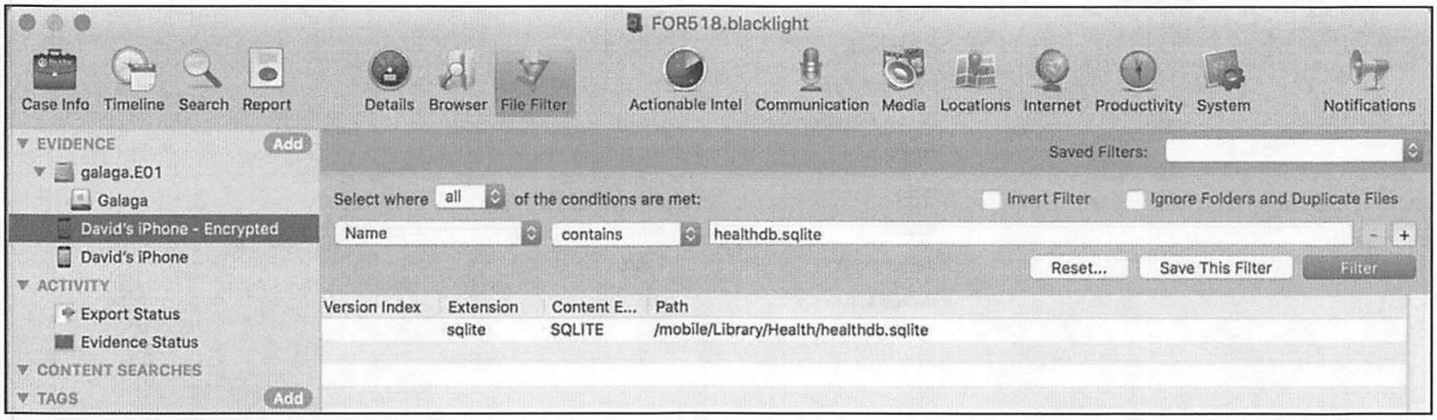
- ☑ File Signature Analysis
- □ Picture Analysis
- □ Video Analysis
- ⊟ Calculate Hashes ...
- □ Identify Known Files ...
- □ File Carving ...
- □ Advanced Options ...

No Templates

- Once imported, both backups will be labeled "David's iPhone"; lets figure out which one is encrypted and label it as such.
- Select the "Browser" view within BlackLight; it should show a folder name as the root folder with the label "`encrypted_iPhone_backup`…" or "`unencrypted_iPhone_backup`…" because David was nice enough to label them for you. Perhaps he labeled them incorrectly; let's verify which is which.
- BlackLight does not show you the backup metadata plist files (`Info.plist`, `Status.plist`, and `Manifest.plist`) where it is easy to verify which is the backup file. We can do this a few ways: We can go to these backup files in Finder or a Terminal window and look at the plists like we did in the previous section and find the "`IsEncrypted`" within the `Manifest.plist` file or we can search for files that are only backed up in the encrypted backup.
- Select any backup and go to the "File Filter" section of BlackLight. Look for the `healthdb.sqlite` database file (FYI: This file keeps track of all the health-related data for the user; however, it is only backed up when the user performs an encrypted backup).
  i. In the "File Filter," select the "List all Files" dropdown and select "Name".
  ii. In the next dropdown, select "contains".
  iii. In the text field, type "`healthdb.sqlite`".

iv. Select "Filter". If the database exists, that is the encrypted backup! If it doesn't, it is unencrypted.

v. To change the name of the backup in the Evidence pane, right-click the backup name and select "Rename Drive...". Rename it "David's iPhone – Encrypted".

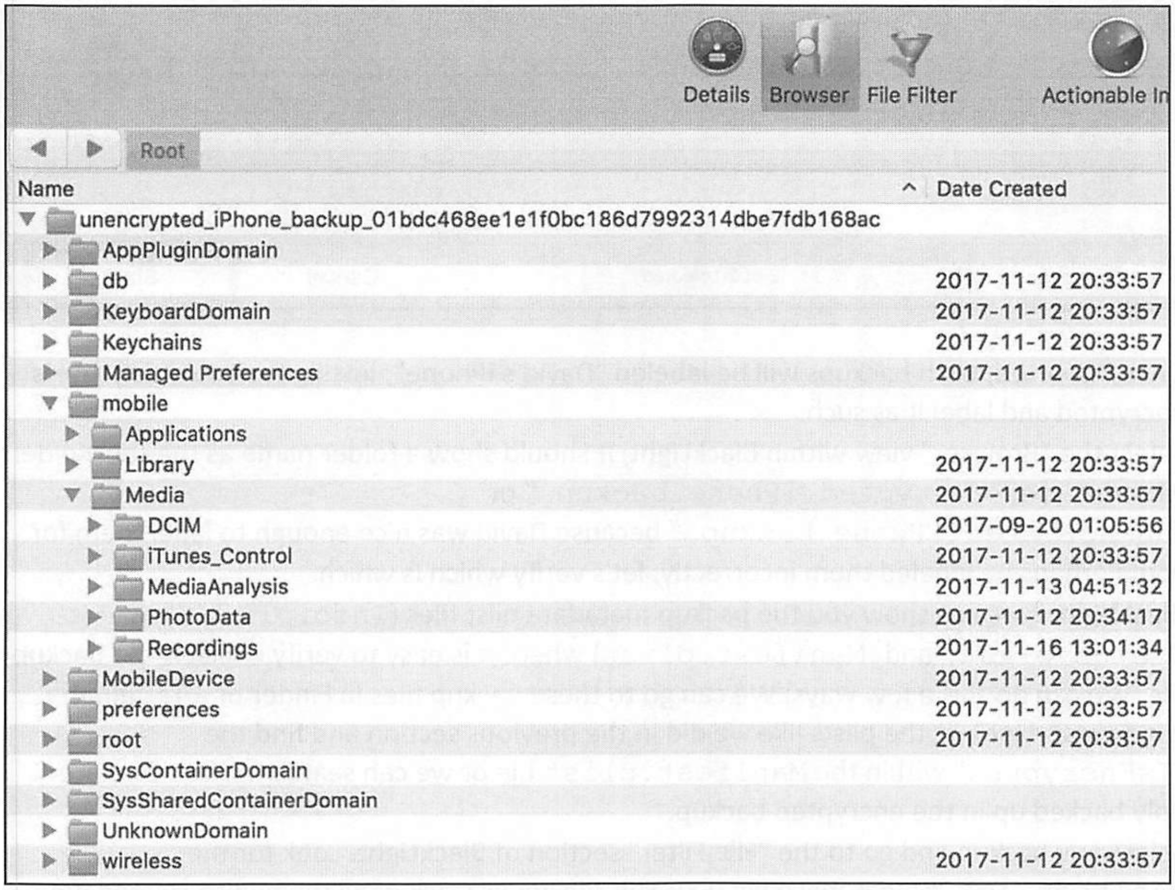


Select where all of the conditions are met:

Name contains healthdb.sqlite

Version Index	Extension	Content E...	Path
	sqlite	SQLITE	/mobile/Library/Health/healthdb.sqlite

- Going back to the "Browser" view for each backup, review how the backup files have been "normalized" from what you saw when you looked at the backup in its "raw" format.



Name	Date Created
unencrypted_iPhone_backup_01bdc468ee1e1f0bc186d7992314dbe7fdb168ac	
AppPluginDomain	
db	2017-11-12 20:33:57
KeyboardDomain	2017-11-12 20:33:57
Keychains	2017-11-12 20:33:57
Managed Preferences	2017-11-12 20:33:57
mobile	
Applications	
Library	2017-11-12 20:33:57
Media	2017-11-12 20:33:57
DCIM	2017-09-20 01:05:56
iTunes_Control	2017-11-12 20:33:57
MediaAnalysis	2017-11-13 04:51:32
PhotoData	2017-11-12 20:34:17
Recordings	2017-11-16 13:01:34
MobileDevice	2017-11-12 20:33:57
preferences	2017-11-12 20:33:57
root	2017-11-12 20:33:57
SysContainerDomain	
SysSharedContainerDomain	
UnknownDomain	
wireless	2017-11-12 20:33:57

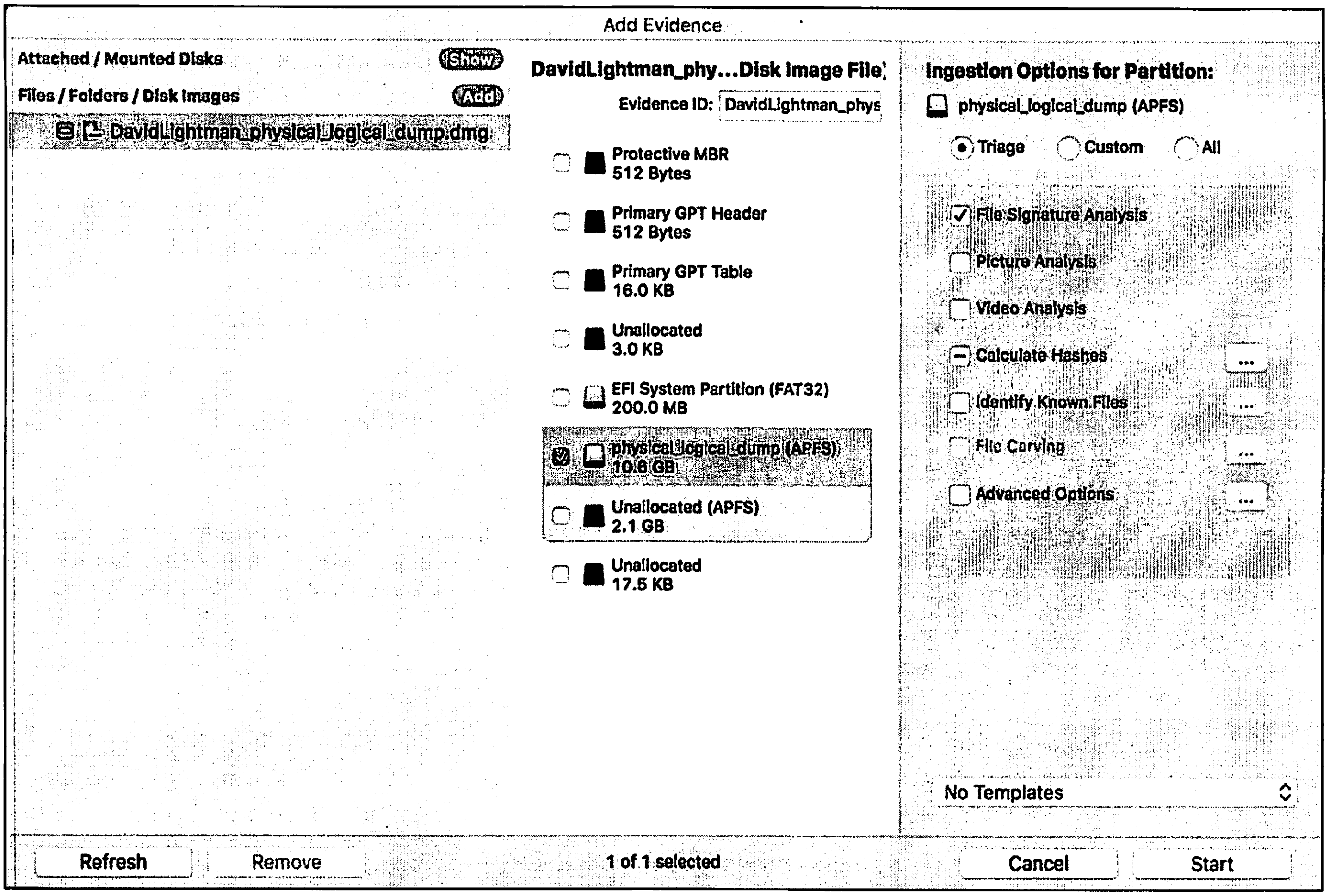
3. Analyze the "Logical Physical" iOS Acquisition

- In your FOR518 /Lab_Images/iPhone/ directory, there is a "Physical/Logical" dump of David's jailbroken iPhone (`DavidLightman_physical_logical_dump.dmg`).

- This dump was created from the jailbroken iPhone, using the SSH/TAR combination to acquire all the physical files in an “unlocked” state. This TAR bundle was then uncompressed and stored in a DMG file, using the Disk Utility.app application for easy transport and mounting.
- You may choose to import this DMG into BlackLight, using or mounting it within the Terminal (or both, your choice!). Follow the instructions for at least one method below:
- **Terminal Mount:** Follow nearly the same procedure as the disks mounted previously. Select the partition labeled “41504653-0000-11AA-AA11-0030654” for the `mount_apfs` command. If you perform a `diskutil list`, it will show up as an APFS volume named “`physical_logical_dump`”.

```
$ sudo mkdir /Volumes/davids_iphone/

$ hdiutil attach -nomount DavidLightman_physical_logical_dump.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/davids_iphone/
```

- **Import into BlackLight:** Follow the steps above when importing the iOS backup directories. For the DMG, de-select “`EFI System Partition (FAT32)`”. We will only be looking at the volume labeled “`physical_logical_dump`”. Default “Ingestion Options” are ok to keep.



DavidLightman_physical_logical_dump.dmg

DavidLightman_phy...Disk Image File

Evidence ID: DavidLightman_phys

- □ Protective MBR 512 Bytes
- □ Primary GPT Header 512 Bytes
- □ Primary GPT Table 16.0 KB
- □ Unallocated 3.0 KB
- □ EFI System Partition (FAT32) 200.0 MB
- ☑ physical_logical_dump (APFS) 10.6 GB
- □ Unallocated (APFS) 2.1 GB
- □ Unallocated 17.5 KB

Ingestion Options for Partition:

physical_logical_dump (APFS)

● Triage ○ Custom ○ All

- ☑ File Signature Analysis
- □ Picture Analysis
- □ Video Analysis
- ⊟ Calculate Hashes ...
- □ Identify Known Files ...
- □ File Carving ...
- □ Advanced Options ...

No Templates

1 of 1 selected

- Take a moment to peruse the structure of this "Physical/Logical Acquisition."
  - i. Note the contents of `/jb/` on the root of the file system, this is one of the artifacts left behind for the LiberiOS jailbreak.
- Take a look at the matching Lockdown records located in `/private/var/root/Library/Lockdown/`

1. What was the name of the computer that last backed up this device (data_ark.plist)?

   ____

2. What type of system was it? (Mac or Windows?)

   ____

4. Review the following files and their locations in all the different iOS acquisitions—some are there; some aren't!

- **Health Database**
  - i. Physical Logical: /private/var/mobile/Library/Health/healthdb.sqlite
  - ii. Encrypted Backup: /mobile/Library/Health/healthdb.sqlite
  - iii. Unencrypted Backup: Does not exist!
- **Keychain**
  - i. Physical Logical: /private/var/Keychains/
  - ii. iOS Backups: /Keychains/

- **Location Data**
  1. Physical Logical: /private/var/root/Library/Caches/locationd/cache_encrypted*
  2. iOS Backups: /root/Library/Caches/locationd/

Lab: Step-by-Step

1. Review iOS Files from David’s macOS Disk Image

- In David’s Mac mounted image, navigate to and open the file `/Volumes/galaga_mounted/Users/dlightman/Library/Preferences/com.apple.iPod.plist` and answer the following questions. Note the identifiers: IMEI and Serial Number.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Preferences/

$ open com.apple.iPod.plist
```

1. How many times was an iPhone connected to this system while using the "dlightman" user account?
   a.“Use Count” Key: 14 times
2. When was this iPhone last connected (UTC)?
   a. “Connected” Key: 03/03/2018 21:10:00 UTC
3. What was the iOS version of the iPhone when it was last connected?
   a. “Firmware Version String”: iOS 11.0.3
4. What is the “human-conversion” make and model of the iPhone (i.e., iPhone X, iPhone 6S+)?
   a. “Product Type”: iPhone9,3 = iPhone 7
   b. Search for it; this website is good: `https://www.theiphonewiki.com/wiki/Models`

- Navigate to the MobileSync backup directory on David’s Mac, located here: `/Volumes/galaga_mounted/Users/dlightman/Library/Application\ Support/MobileSync/Backup/`.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Application\
Support/MobileSync/Backup/

$ ls -la
```

5. What are the first few characters of the Universal Device Identifier (UDID) for this backup?
   a. Each backup is stored in a directory named with its UDID: 01bdc468ee1e1f0bc186d7992314dbe7fdb168ac

- Navigate into this iOS backup, review the backup structure, and open the plist metadata files.

```
$ cd 01bdc468ee1e1f0bc186d7992314dbe7fdb168ac

$ ls -la

$ open *.plist
```

- Review the contents of the `Status.plist` file.
- Review the contents of the `Info.plist` file.

6. What is the date of this backup (UTC)?
   a. "Last Backup Date" Key: 03/03/2018 20:28:06 UTC
7. What is the name of this iPhone?
   a. "Device Name" or "Display Name": David's iPhone
8. What was the phone number of this device when it was backed up?
   a. "Phone Number": +44 7848 916073

- Review the contents of the `Manifest.plist` file.

9. Was there a passcode set on the device at the time of backup?
   a. "WasPasscodeSet" Key: Yes, it had a passcode.
10. Is this backup encrypted or not?
    a. "IsEncrypted" Key: Yes, this backup is encrypted.

- Navigate to the lockdown directory for this system, `/Volumes/galaga_mounted/private/var/db/lockdown/`. Open the lockdown file/pairing certificate for the connected iPhone.

```
$ cd /Volumes/galaga_mounted/private/var/db/lockdown/

$ ls -la

$ open 01bdc468ee1e1f0bc186d7992314dbe7fdb168ac.plist
```

11. What is the Wi-Fi MAC Address for the connected iPhone?
    a. "WiFiMACAddress" Key: b8:53:ac:09:cc:86

2. Extract and Analyze iOS Backup Files from David's macOS Disk Image

- Navigate to David Lightman's `Documents` directory. David was smart enough to back up his backups before he jailbroke his iPhone. He created an encrypted and unencrypted version of his iPhone and stored them in the `iPhone_Backups` directory (`/Volumes/galaga_mounted/Users/dlightman/Documents/iPhone_Backups/`).
- In the unencrypted backup (`/Volumes/galaga_mounted/Users/dlightman/Documents/iPhone_Backups/unencrypted_iPhone_backup_01bdc468ee1e1f0bc186d7992314dbe7fdb168ac`), Copy the `Manifest.db` file to your `~/FOR518` directory and review the contents of the `Manifest.db` file. If you look for this file in the encrypted backups, you'll find that it is... encrypted!
- We are copying out this file because `sqlite3` (via command line) and DB Browser for SQLite application cannot open the file on a read-only volume. This is something you will

have to do quite often to access the SQLite databases outside of another application like BlackLight.

- Use whichever tool you prefer to open the `Manifest.db` database and review its contents.

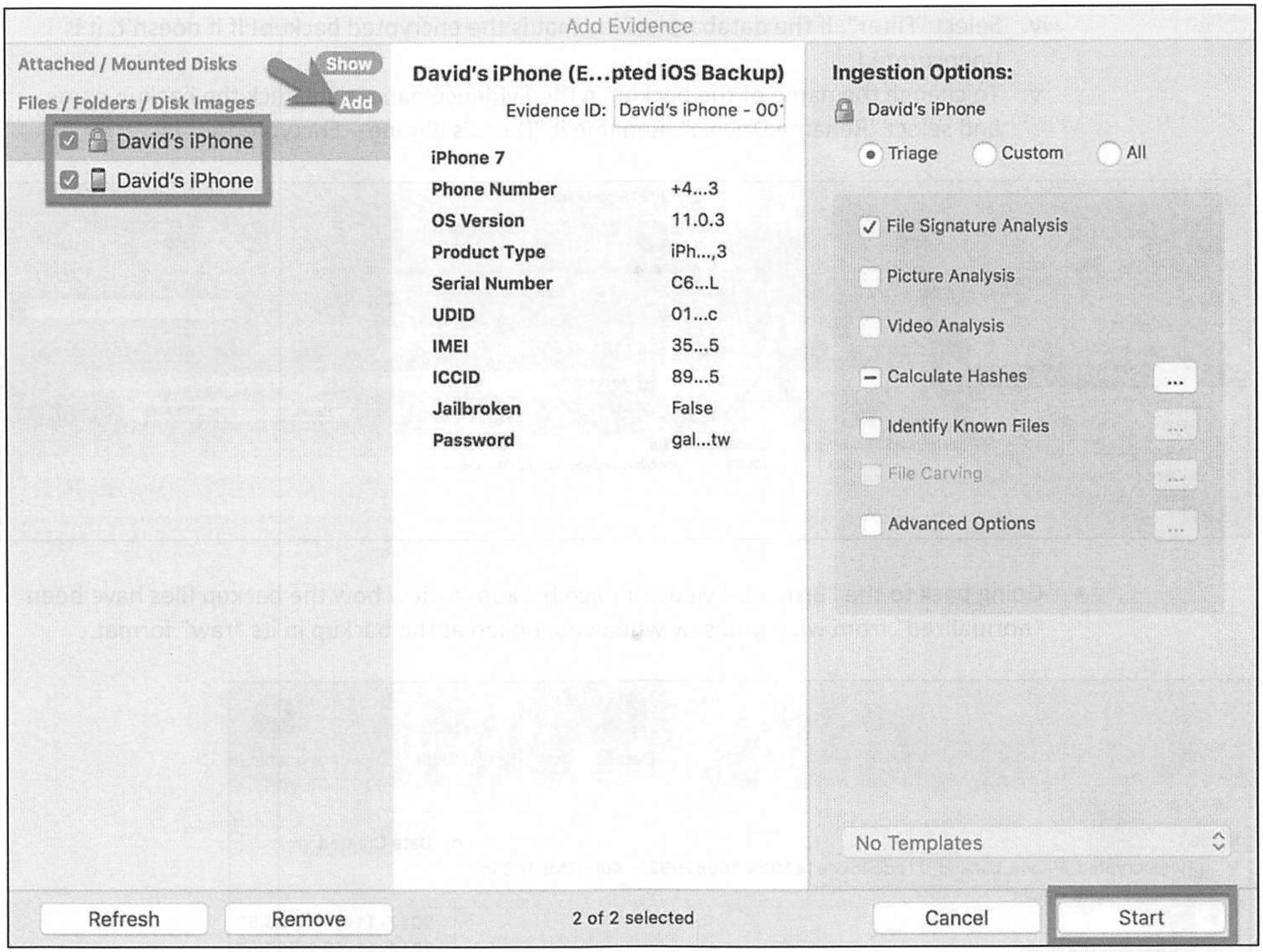
```
$ cd /Volumes/galaga_mounted/Users/dlightman/Documents/iPhone_Backups/
unencrypted_iPhone_backup_01bdc468ee1e1f0bc186d7992314dbe7fdb168ac/

$ cp Manifest.db ~/FOR518
```

1. In the "Files" table, what is the `fileID` hash of the `sms.db` database?
    a. 3d0d7e5fb2ce288813306e4d4636395e047a3d28
    b. You can run a query on the database such as:
        1. `select * from Files where relativePath like "%sms.db%"`
    c. In DB Browser for SQLite, you can filter for “`sms.db`” in the `relativePath` column area.

- Extract both backups to your ~/FOR518 directory.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Documents/

$ cp -R iPhone_Backups/ ~/FOR518
```

- Import these backups into your FOR518 BlackLight Case File.
    i. Select the "Add" Evidence button.
    ii. Select the "Add" button under "Files/Folders/Disk Images".
    iii. Navigate to where you saved the backups (~/FOR518).
    iv. Select each backup directory (separately). For the backup labeled “encrypted,” you’ll have to input the backup password by double-clicking the lock icon in the middle pane. The backup password is "`galagaftw`". Please be patient; as stated in the pop-up window, this will take a bit of time.
    v. Once each backup is imported into the "Add Evidence" window, select both backups, keep the default "Ingestion Options," and select "Start" to add them to the Case File.



☑ David's iPhone
☑ David's iPhone

David's iPhone (E...pted iOS Backup)

Evidence ID: David's iPhone - 00'

iPhone 7

Phone Number	+4...3
OS Version	11.0.3
Product Type	iPh...,3
Serial Number	C6...L
UDID	01...c
IMEI	35...5
ICCID	89...5
Jailbroken	False
Password	gal...tw

Ingestion Options:

David's iPhone

● Triage ○ Custom ○ All

- ☑ File Signature Analysis
- □ Picture Analysis
- □ Video Analysis
- ⊟ Calculate Hashes ...
- □ Identify Known Files ...
- □ File Carving ...
- □ Advanced Options ...

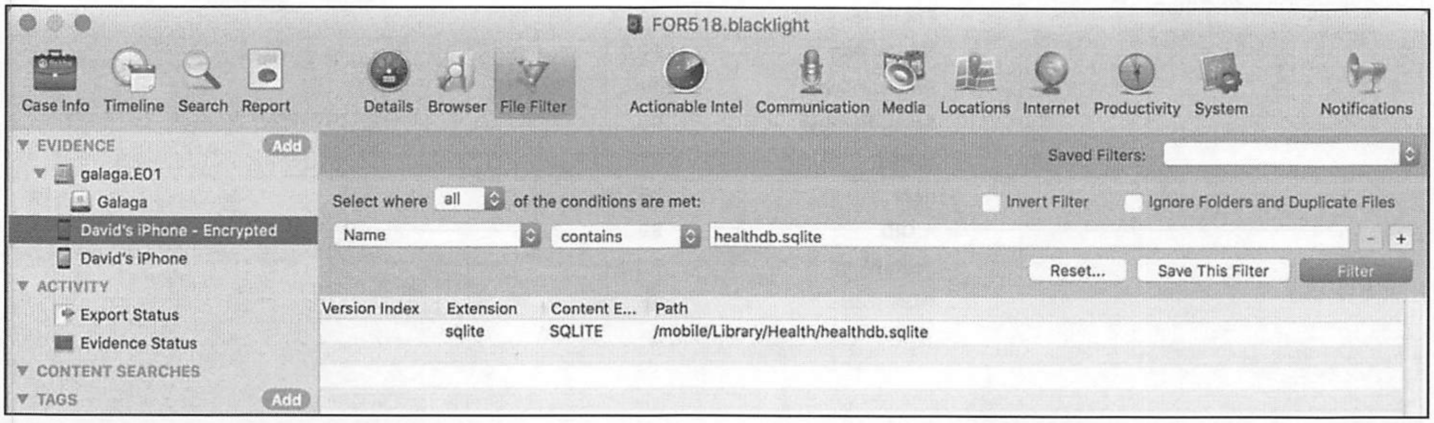
No Templates

2 of 2 selected

- Once imported, both backups will be labeled "David's iPhone"; let's figure out which one is encrypted and label it as such.
- Select the "Browser" view within BlackLight; it should show a folder name as the root folder with the label "`encrypted_iPhone_backup`…" or "`unencrypted_iPhone_backup`…" because David was nice enough to label them for you. Perhaps he labeled them incorrectly; let's verify which is which.
- BlackLight does not show you the backup metadata plist files (`Info.plist`, `Status.plist`, and `Manifest.plist`) where it is easy to verify which is the backup file. We can do this a few ways: We can go to these backup files in Finder or a Terminal window and look at the plists like we did in the previous section and find the "`IsEncrypted`" within `Manifest.plist` file or we can search for files that are only backed up in the encrypted backup.
- Select any backup and go to the "File Filter" section of BlackLight. Look for the `healthdb.sqlite` database file (FYI: This file keeps track of all the health-related data for the user; however it is only backed up when the user performs an encrypted backup).
  i. In the "File Filter," select the "List all Files" dropdown and select "Name".
  ii. In the next dropdown, select "contains".
  iii. In the text field, type "`healthdb.sqlite`".

iv. Select "Filter". If the database exists, that is the encrypted backup! If it doesn't, it is unencrypted.

v. To change the name of the backup in the Evidence pane, right-click the backup name and select "Rename Drive...". Rename it "David's iPhone – Encrypted".

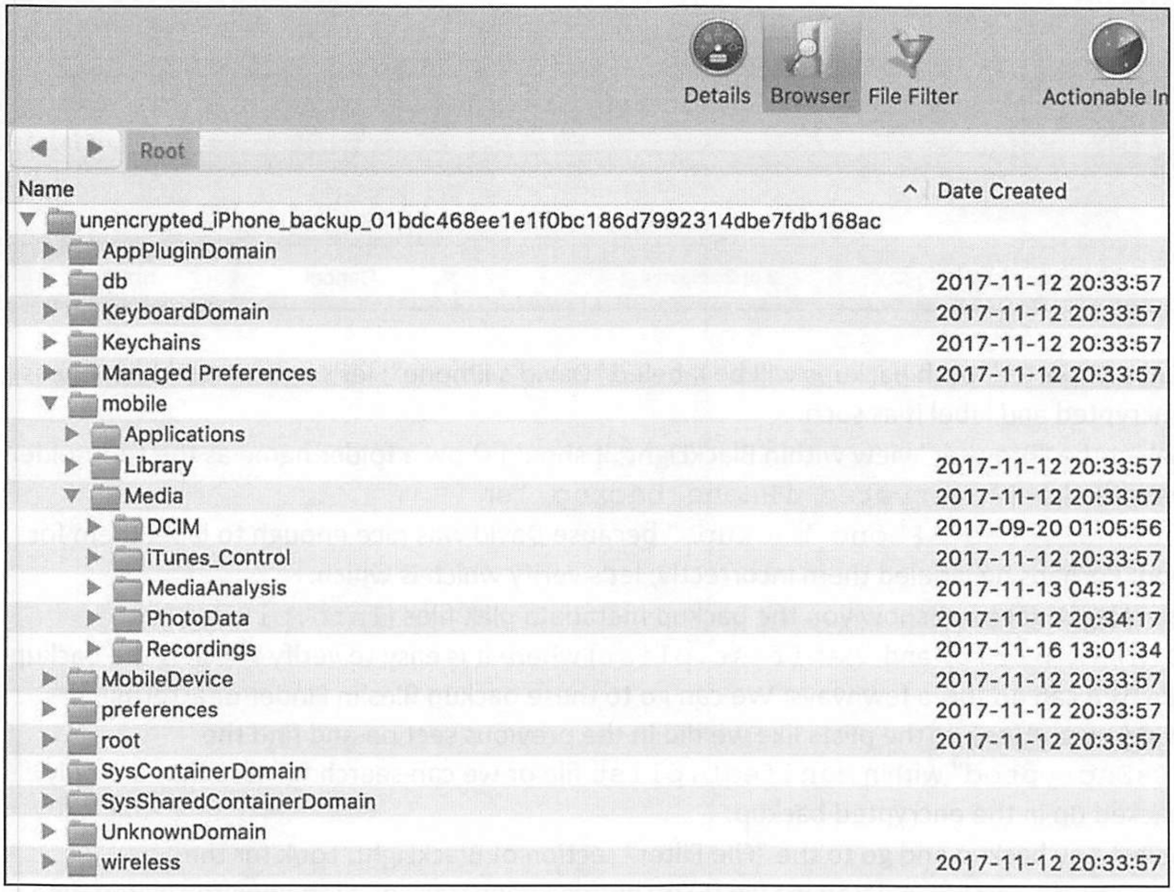


Select where all of the conditions are met:

Name contains healthdb.sqlite

Version Index	Extension	Content E...	Path
	sqlite	SQLITE	/mobile/Library/Health/healthdb.sqlite

- Going back to the "Browser" view for each backup, review how the backup files have been "normalized" from what you saw when you looked at the backup in its "raw" format.



```
Name                                                          Date Created
▼ unencrypted_iPhone_backup_01bdc468ee1e1f0bc186d7992314dbe7fdb168ac
  ▶ AppPluginDomain
  ▶ db                                                        2017-11-12 20:33:57
  ▶ KeyboardDomain                                            2017-11-12 20:33:57
  ▶ Keychains                                                 2017-11-12 20:33:57
  ▶ Managed Preferences                                       2017-11-12 20:33:57
  ▼ mobile
    ▶ Applications
    ▶ Library                                                 2017-11-12 20:33:57
    ▼ Media                                                   2017-11-12 20:33:57
      ▶ DCIM                                                  2017-09-20 01:05:56
      ▶ iTunes_Control                                        2017-11-12 20:33:57
      ▶ MediaAnalysis                                         2017-11-13 04:51:32
      ▶ PhotoData                                             2017-11-12 20:34:17
      ▶ Recordings                                            2017-11-16 13:01:34
  ▶ MobileDevice                                              2017-11-12 20:33:57
  ▶ preferences                                               2017-11-12 20:33:57
  ▶ root                                                      2017-11-12 20:33:57
  ▶ SysContainerDomain
  ▶ SysSharedContainerDomain
  ▶ UnknownDomain
  ▶ wireless                                                  2017-11-12 20:33:57
```

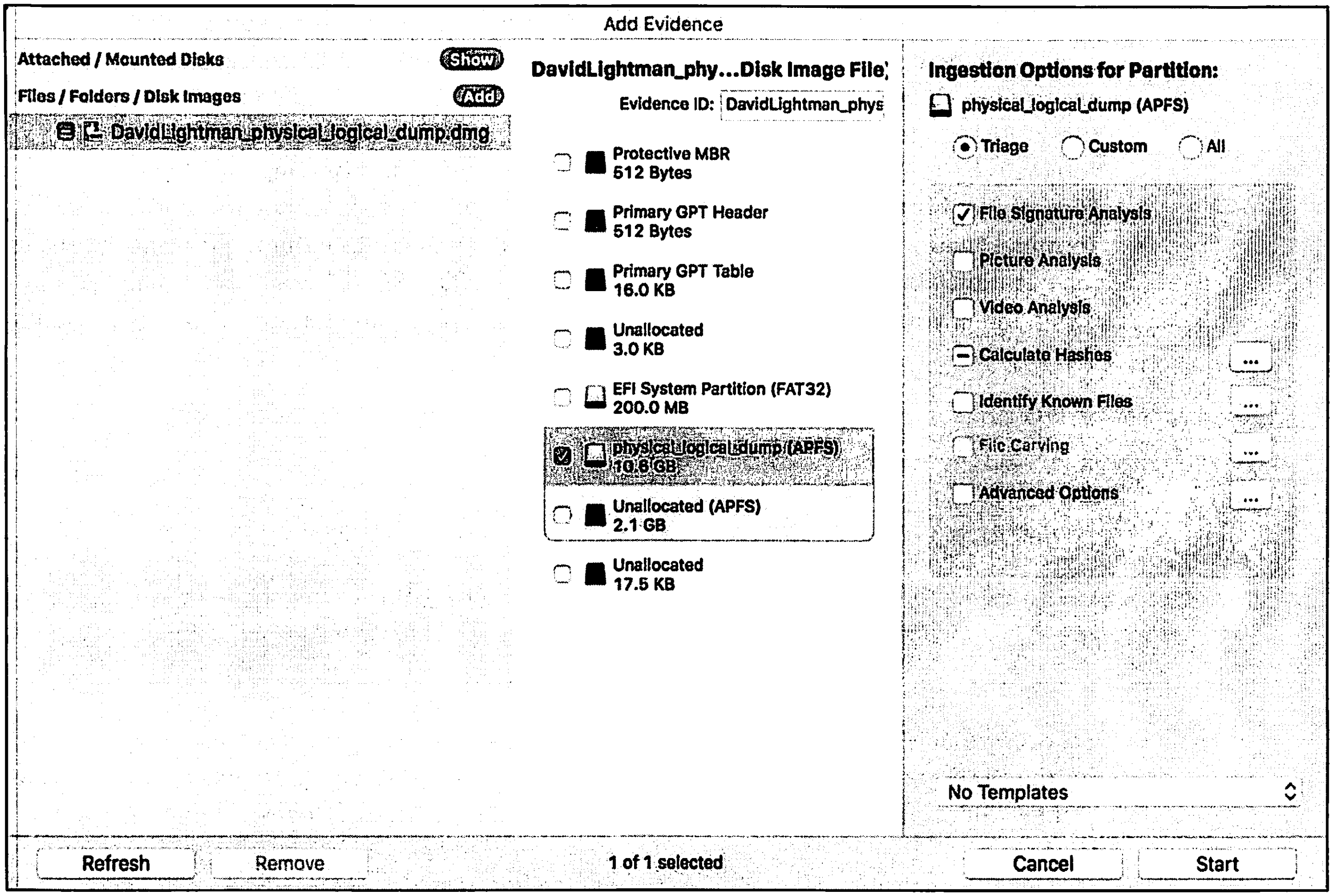
3. Analyze the "Logical Physical" iOS Acquisition

- In your FOR518 /Lab_Images/iPhone/ directory, there is a "Physical/Logical" dump of David's jailbroken iPhone (`DavidLightman_physical_logical_dump.dmg`).

- This dump was created from the jailbroken iPhone using the SSH/TAR combination to acquire all the physical files in an "unlocked" state. This TAR bundle was then uncompressed and stored in a DMG file, using the Disk Utility.app application for easy transport and mounting.
- You may choose to import this DMG into BlackLight, using or mounting it within the Terminal (or both, your choice!). Follow the instructions for at least one method below:
- **Terminal Mount:** Follow nearly the same procedure as the disks mounted previously. Select the partition labeled "41504653-0000-11AA-AA11-0030654" for the `mount_apfs` command. If you perform a `diskutil list`, it will show up as an APFS volume named "`physical_logical_dump`".

```
$ sudo mkdir /Volumes/davids_iphone/

$ hdiutil attach -nomount DavidLightman_physical_logical_dump.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/davids_iphone/
```

- **Import into BlackLight:** Follow the steps above when importing the iOS backup directories. For the DMG, de-select "`EFI System Partition (FAT32)`". We will only be looking at the volume labeled "`physical_logical_dump`". Default "Ingestion Options" are ok to keep.



Add Evidence

Attached / Mounted Disks (Show)

Files / Folders / Disk Images (Add)

DavidLightman_physical_logical_dump.dmg

DavidLightman_phy...Disk Image File

Evidence ID: DavidLightman_phys

- ☐ Protective MBR 512 Bytes
- ☐ Primary GPT Header 512 Bytes
- ☐ Primary GPT Table 16.0 KB
- ☐ Unallocated 3.0 KB
- ☐ EFI System Partition (FAT32) 200.0 MB
- ☑ physical_logical_dump (APFS) 10.6 GB
- ☐ Unallocated (APFS) 2.1 GB
- ☐ Unallocated 17.5 KB

Ingestion Options for Partition:

physical_logical_dump (APFS)

● Triage ○ Custom ○ All

- ☑ File Signature Analysis
- ☐ Picture Analysis
- ☐ Video Analysis
- ⊟ Calculate Hashes ...
- ☐ Identify Known Files ...
- ☐ File Carving ...
- ☐ Advanced Options ...

No Templates

Refresh | Remove | 1 of 1 selected | Cancel | Start

- Take a moment to peruse the structure of this "Physical/Logical Acquisition."
   i. Note the contents of `/jb/` on the root of the file system, this is one of the artifacts left behind for the LiberiOS jailbreak.
- Take a look at the matching Lockdown records located in `/private/var/root/Library/Lockdown/`

1. What was the name of the computer that last backed up this device (data_ark.plist)?
   a. "com.apple.iTunes.backup-LastBackupComputerName" Key: "David's MacBook Pro"
2. What type of system was it? (Mac or Windows?)
   a. "com.apple.iTunes.backup-LastBackupComputerType" Key: Surprise! It was a Mac! (Couldn't see that one coming!)

4. Review the following files and their locations in all the different iOS acquisitions—some are there; some aren't!

- **Health Database**
   i. Physical Logical: /private/var/mobile/Library/Health/healthdb.sqlite
   ii. Encrypted Backup: /mobile/Library/Health/healthdb.sqlite
   iii. Unencrypted Backup: Does not exist!
- **Keychain**
   i. Physical Logical: /private/var/Keychains/
   ii. iOS Backups: /Keychains/
- **Location Data**

i. Physical Logical: /private/var/root/Library/Caches/locationd/cache_encrypted*
ii. iOS Backups: /root/Library/Caches/locationd/

Lab: Key Takeaways

- **Notice the differences between each type of iOS acquisition.**

- **Get comfortable with the key areas in which to find initial triage data for the device being analyzed.**

This page intentionally left blank.

Lab 1.3: Disks and Partitions

Objectives

- Review the disks and partitions on your analysis system
- Parse, by hand, the Protective MBR, GPT Header, and GPT Table

Lab Preparation

(Note: Some of this might already be accomplished via earlier labs, but this is the state that we hope your system is in prior to the start of this lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this lab.)

1. **Software Preparation**: The following tools will be used in this lab:
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from /Applications/Utilities/.
   - Hex Editor
     i. Locate and open the hex editor of your choice.
     ii. I like these:
        1. 0xED: `http://www.suavetech.com/0xed/0xed.html`
           a. /Applications/0xED.app
        2. Hex Fiend: `http://ridiculousfish.com/hexfiend/`
           a. /Applications/Hex Fiend.app
        3. `xxd` Command: Native command-line utility on OS X
   - The Sleuth Kit
     i. TSK utilities should have been installed in Lab 0; please review this lab if needed.
   - Calculator.app
     i. Locate and open the Calculator.app application in /Applications/.
     ii. Use the `View | Programmer` setting to perform the hex conversions.
2. **Lab File Preparation**: Locate the `GPT.dmg` file located in the `Lab_Files/Lab 1.3 - Disks & Partitions` directory on your FOR518 USB drive. **DO NOT DOUBLE-CLICK/OPEN THIS FILE**. This file should have the MD5: `9e36e2a9e4fc9d6a04a1f13aad8c9e75`. This can be checked by executing the command: `md5 GPT.dmg`. If you do happen to open this file, just be aware the timestamps answers will show different timestamps.
3. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

Lab: Questions

*****Remember that GPT is Little Endian*****

1. `Use the diskutil list` command

- Use the `diskutil list` command to view the disks and partitions on your analysis system.

```
$ diskutil list
```

2. Review the output of the `diskutil list` command

1. How many disks does your system have connected?

 ____

2. Fill out the table below with the information for up to four disks

Disk Identifier	Partition Scheme	Number of Partitions	Volume Names	Disk Formats or File Systems	Disk Size
/dev/disk0					
/dev/disk1					
/dev/disk2					
/dev/disk3					

3. Use the `diskutil info` command on a couple of disks

```
$ diskutil info disk#
```

1. What is the Device/Media name?

 ____

2. What partition scheme does it use?

 ____

3. How large is this drive?

4. Use the `diskutil info` command on a couple of partition slices

```
$ diskutil info disk#s#
```

1. What is the volume name?

2. What is the partition type?

3. What is the size of this volume?

5. Use The Sleuth Kit command `mmls` to view the GUID Partition Table on the `GPT.dmg` file.

- The MD5 hash for the `GPT.dmg` file should be `9e36e2a9e4fc9d6a04a1f13aad8c9e75`; if it is not, please extract the file from the FOR518 thumb drive again—otherwise the answers may not match those in the Step-by-Step section.

```
$ mmls GPT.dmg
```

- Fill in the GPT and partition table information below.

- **GPT Information (You can find the Partition Type GUIDs on your FOR518 Reference Sheet)**

Sector containing Protective MBR (Safety Table)	
Sector containing Primary GPT Header	
Starting sector of Primary GPT Table	

- **Partition Information**

Partition Number	Partition Name	Start Sector	Length (in Sectors)
1			

6. Extract/View the Protective MBR using `dd`

- You may use the `xxd` command for output rather than redirecting it to a file for a GUI hex editor if you prefer the command-line interface:
  i. `dd if=GPT.dmg count=1 | xxd`

- You can use the "`open`" command to open this file in a GUI hex editor from the command line. To open the output file in 0xED, use this command:
    i. `open -a 0xED <output_filename>`
    ii. You may want to change the offsets from hex to decimal to use the offsets in this lab; **all offsets in this course are in decimal**:
        1. 0xED: Double-click the "`Dec`" in the lower-left corner of the application. You may also go into the application preferences and change the "Number Mode" (`0xED | Preferences`).
        2. Hex Fiend: Single-click the offset column to switch between hex and decimal offsets.
- In the command below, the ">" character is used to redirect the output of a command to a file. The file in this instance is GPT-DMG-PMBR. You may name your file anything you want, as long as you remember what it is and where you put it.

```
$ dd if=GPT.dmg count=1 > GPT-DMG-PMBR
```

1. Is this volume bootable (Offset 446, 0x00 = No, 0x80 = Yes)?

   ____

2. What is the size of the volume in sectors (Offset 458–461)?

   ____

7. Extract the GPT Header using `dd`

```
$ dd if=GPT.dmg skip=1 count=1 > GPT-DMG-GPTHeader
```

1. What is the signature (Offset 0–7)?

   ____

2. What is the LBA of the GPT Header (this file) (Offset 24–31)?

   ____

3. What is the LBA of the Backup/Secondary GPT Header (Offset 32–39)?

   ____

4. What is the GUID of the partition (Offset 56–71)?

   ____

5. Starting LBA of the GPT Partition Table (Offset 72–79)?

   ____

8. Extract the GPT Table using `dd`

```
$ dd if=GPT.dmg skip=2 count=1 > GPT-DMG-GPTTable
```

1. What is the partition type GUID, and what type of partition is it (Offset 0–15)?

2. What is the unique GUID for the partition (Offset 16–31)?

3. What is the starting LBA for the partition (Offset 32–39)?

4. What is the ending LBA for the partition (Offset 40–47)?

5. What is the name of the partition (Offset 56+)?

9. **Use the `hdiutil imageinfo GPT.dmg` command and review the output.**
    - Check your answers with this command or make your life much easier in the future!

```
$ hdiutil imageinfo GPT.dmg
```

Extra Credit:

Parse your own GPT Header and Table of your analysis system. Use `dd` to extract files; you will need to use `sudo` with the command or you will get a "Permission denied” error. This may not work with FileVault disks; try using a different disk or volume.

Lab: Step-by-Step

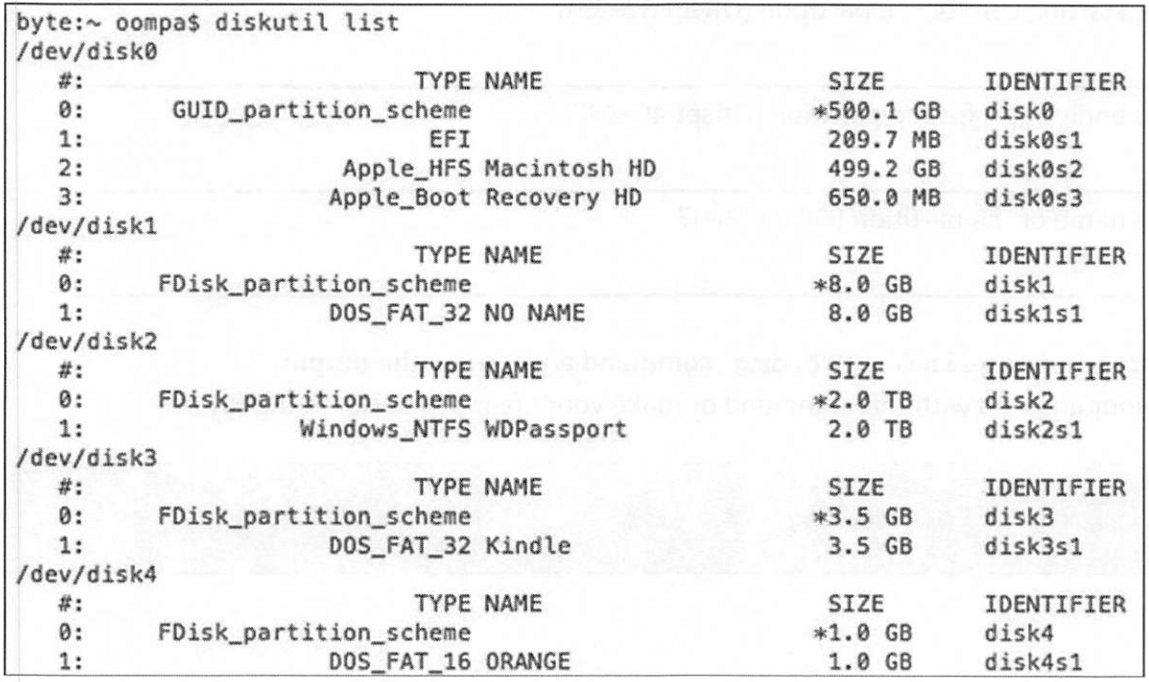
1. Use the `diskutil list` command

- Use the `diskutil list` command to view the disks and partitions on your analysis system.

```
$ diskutil list
```

2. Review the output of the `diskutil list` command

This system output has five disks:



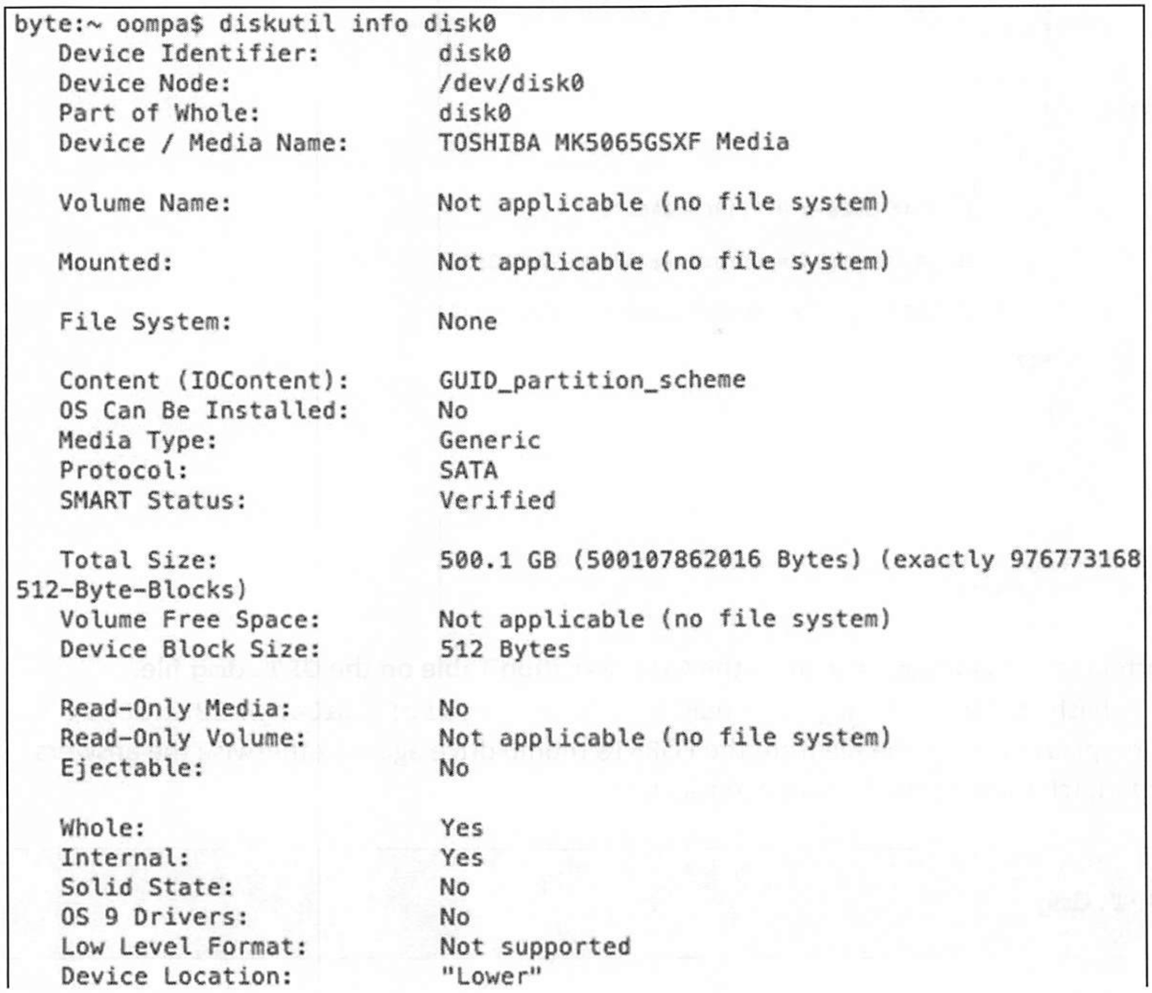
```
byte:~ oompa$ diskutil list
/dev/disk0
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      GUID_partition_scheme                        *500.1 GB   disk0
   1:                        EFI                         209.7 MB   disk0s1
   2:                  Apple_HFS Macintosh HD            499.2 GB   disk0s2
   3:                 Apple_Boot Recovery HD             650.0 MB   disk0s3
/dev/disk1
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     FDisk_partition_scheme                        *8.0 GB     disk1
   1:                 DOS_FAT_32 NO NAME                 8.0 GB     disk1s1
/dev/disk2
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     FDisk_partition_scheme                        *2.0 TB     disk2
   1:               Windows_NTFS WDPassport              2.0 TB     disk2s1
/dev/disk3
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     FDisk_partition_scheme                        *3.5 GB     disk3
   1:                 DOS_FAT_32 Kindle                  3.5 GB     disk3s1
/dev/disk4
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     FDisk_partition_scheme                        *1.0 GB     disk4
   1:                 DOS_FAT_16 ORANGE                  1.0 GB     disk4s1
```

Disk Identifier	Partition Scheme	Number of Partitions	Volume Names	Disk Formats or File Systems	Disk Size
/dev/disk0	GUID Partition Scheme	3	EFI (Unamed) Macintosh HD Recovery HD	EFI (FAT) HFS+ HFS+	500 GB
/dev/disk1	FDisk Partition Scheme (MBR)	1	"NO NAME"	FAT32	8 GB
/dev/disk2	FDisk Partition Scheme (MBR)	1	WDPassport	NTFS	2 TB
/dev/disk3	FDisk Partition Scheme (MBR)	1	Kindle	FAT32	3.5 GB

3. Use the `diskutil info` command on a couple of disks

```
$ diskutil info disk0
```

The output for this `/dev/disk0` using the `diskutil info` command shows that I am using a 500GB Toshiba hard drive. This disk uses the GUID Partitioning scheme.

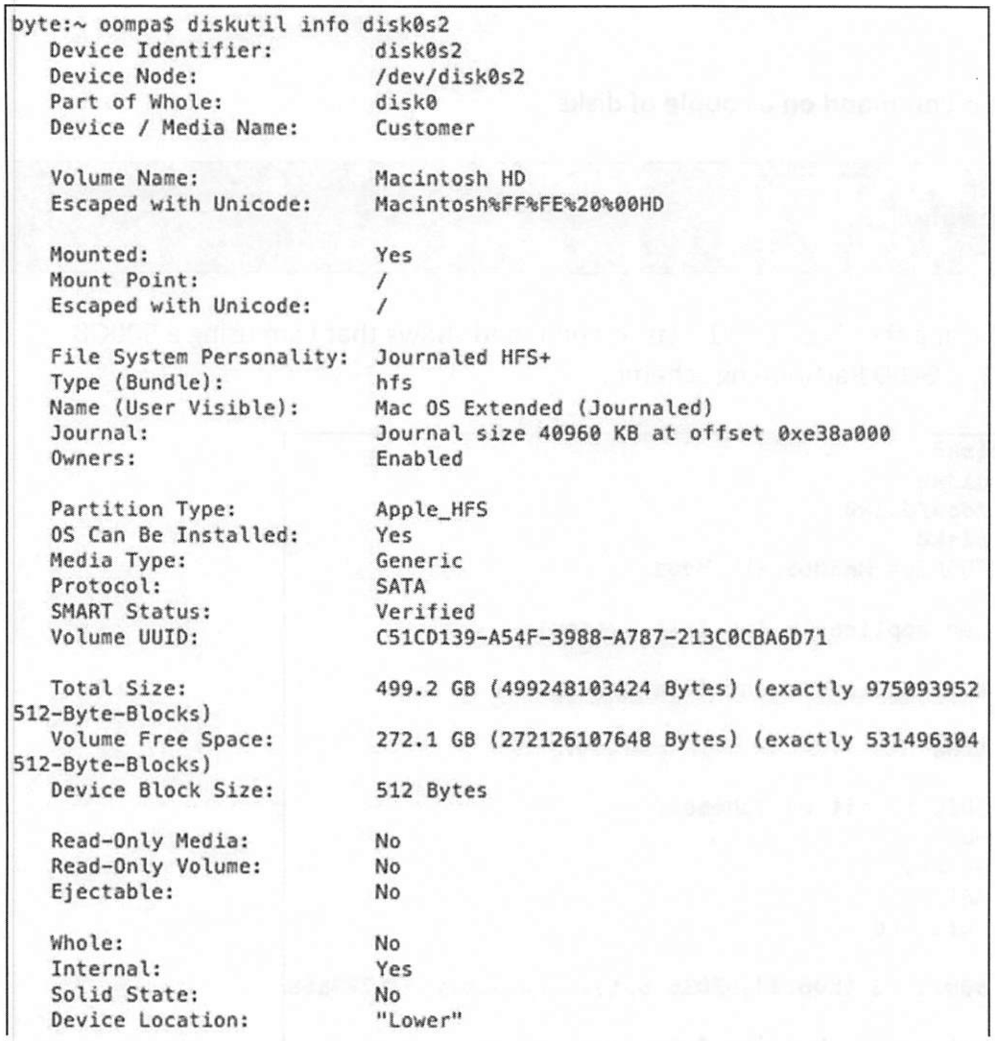


```
byte:~ oompa$ diskutil info disk0
   Device Identifier:        disk0
   Device Node:              /dev/disk0
   Part of Whole:            disk0
   Device / Media Name:      TOSHIBA MK5065GSXF Media

   Volume Name:              Not applicable (no file system)

   Mounted:                  Not applicable (no file system)

   File System:              None

   Content (IOContent):      GUID_partition_scheme
   OS Can Be Installed:      No
   Media Type:               Generic
   Protocol:                 SATA
   SMART Status:             Verified

   Total Size:               500.1 GB (500107862016 Bytes) (exactly 976773168
512-Byte-Blocks)
   Volume Free Space:        Not applicable (no file system)
   Device Block Size:        512 Bytes

   Read-Only Media:          No
   Read-Only Volume:         Not applicable (no file system)
   Ejectable:                No

   Whole:                    Yes
   Internal:                 Yes
   Solid State:              No
   OS 9 Drivers:             No
   Low Level Format:         Not supported
   Device Location:          "Lower"
```

4. Use the `diskutil info` command on a couple of partition slices

```
$ diskutil info disk0s2
```

The output for this disk identified as `disk0s2`; the boot disk shows that it is named "Macintosh HD." This partition uses HFS+ and is 499.2GB in size.



```
byte:~ oompa$ diskutil info disk0s2
   Device Identifier:        disk0s2
   Device Node:              /dev/disk0s2
   Part of Whole:            disk0
   Device / Media Name:      Customer

   Volume Name:              Macintosh HD
   Escaped with Unicode:     Macintosh%FF%FE%20%00HD

   Mounted:                  Yes
   Mount Point:              /
   Escaped with Unicode:     /

   File System Personality:  Journaled HFS+
   Type (Bundle):            hfs
   Name (User Visible):      Mac OS Extended (Journaled)
   Journal:                  Journal size 40960 KB at offset 0xe38a000
   Owners:                   Enabled

   Partition Type:           Apple_HFS
   OS Can Be Installed:      Yes
   Media Type:               Generic
   Protocol:                 SATA
   SMART Status:             Verified
   Volume UUID:              C51CD139-A54F-3988-A787-213C0CBA6D71

   Total Size:               499.2 GB (499248103424 Bytes) (exactly 975093952
512-Byte-Blocks)
   Volume Free Space:        272.1 GB (272126107648 Bytes) (exactly 531496304
512-Byte-Blocks)
   Device Block Size:        512 Bytes

   Read-Only Media:          No
   Read-Only Volume:         No
   Ejectable:                No

   Whole:                    No
   Internal:                 Yes
   Solid State:              No
   Device Location:          "Lower"
```

5. **Use The Sleuth Kit command `mmls` to view the GUID Partition Table on the `GPT.dmg` file.**
   - The MD5 hash for the `GPT.dmg` file should be `9e36e2a9e4fc9d6a04a1f13aad8c9e75`; if it is not, please extract the file from the FOR518 thumb drive again—otherwise the answers may not match those in the Step-by-Step section.

```
$ mmls GPT.dmg
```

The `mmls` output is shown below for the `GPT.dmg` disk image file.

```
$ mmls GPT.dmg
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
00:  Meta    0000000000   0000000000   0000000001   Safety Table
01:  -----   0000000000   0000000039   0000000040   Unallocated
02:  Meta    0000000001   0000000001   0000000001   GPT Header
03:  Meta    0000000002   0000000033   0000000032   Partition Table
04:  00      0000000040   0000079999   0000079960   disk image
05:  -----   0000080000   0000080039   0000000040   Unallocated
```

- **GPT Information**

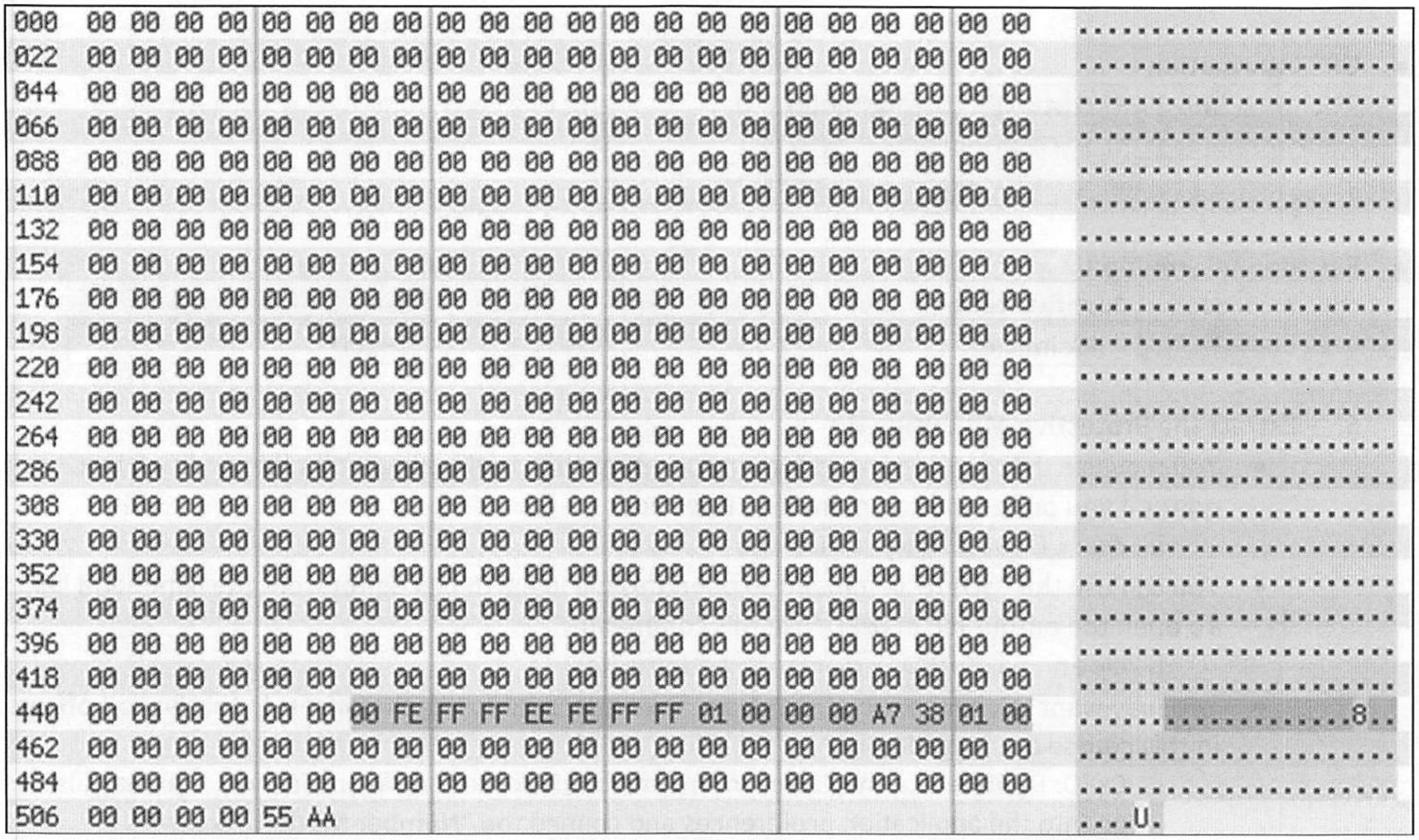
Sector containing Protective MBR (Safety Table)	0
Sector containing Primary GPT Header	1
Starting sector of Primary GPT Table	2

- **Partition Information**

Partition Number	Partition Name	Start Sector	Length (in Sectors)
1	"disk image"	40	79960

6. **Extract the Protective MBR using dd**
   - You may use the `xxd` command for output rather than redirecting it to a file for a GUI hex editor if you prefer the command-line interface:
     i. `dd if=GPT.dmg count=1 | xxd`
   - You can use the "`open`" command to open this file in a GUI hex editor from the command line. To open the output file in 0xED, use this command:
     i. `open -a 0xED` *`<output_filename>`*
   - You may want to change the offsets from hex to decimal to use the offsets in this lab; **all offsets in this course are in decimal**:
     i. 0xED: Double-click the "`Dec`" in the lower-left corner of the application. You may also go into the application preferences and change the "Number Mode" (`0xED | Preferences`).
     ii. Hex Fiend: Single-click the offset column to switch between hex and decimal offsets.
   - In the command below, the ">" character is used to redirect the output of a command to a file. The file in this instance is GPT-DMG-PMBR. You may name your file anything you want, as long as you remember what it is and where you put it.

```
$ dd if=GPT.dmg count=1 > GPT-DMG-PMBR
```

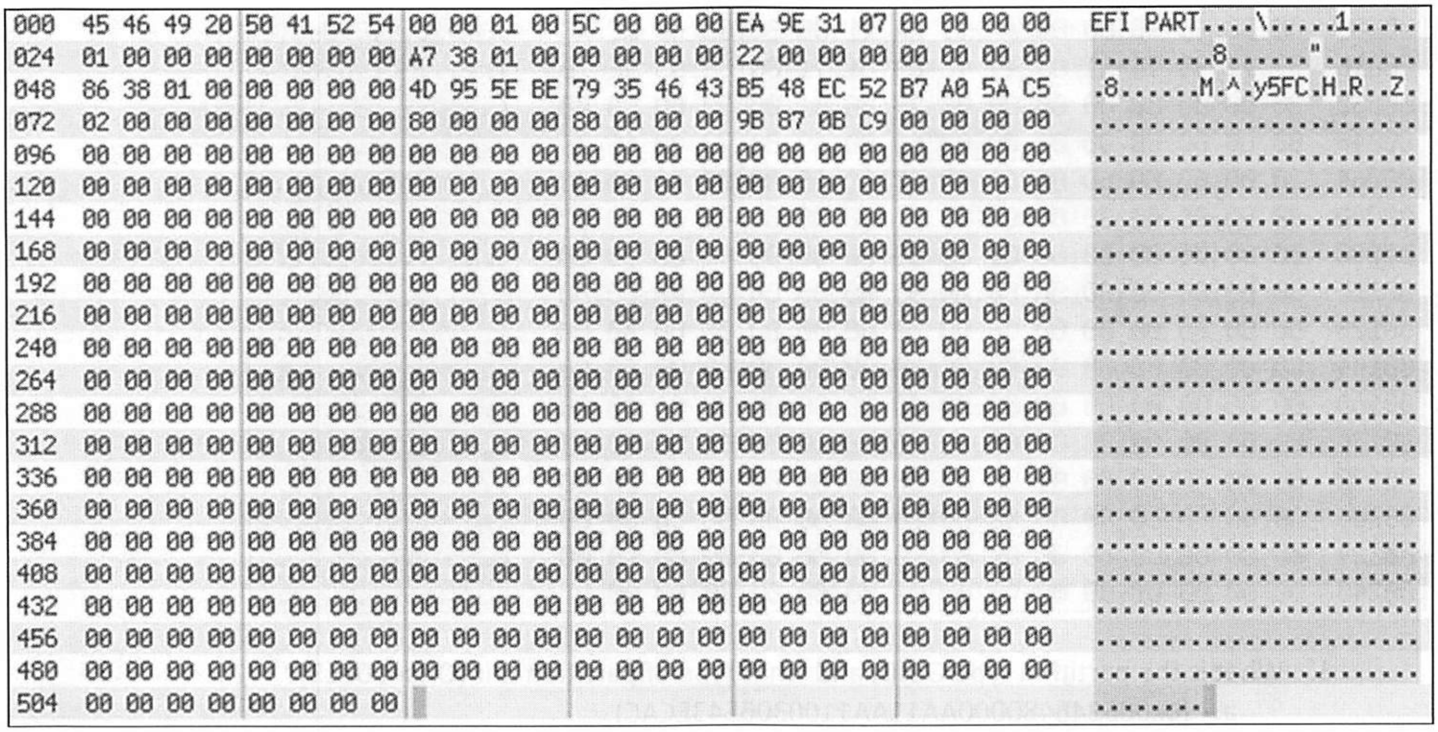


```
000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
022  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
044  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
066  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
088  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
132  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
154  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
176  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
198  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
242  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
264  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
286  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
308  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
330  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
352  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
374  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
396  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
418  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
440  00 00 00 00 00 00 00 FE FF FF EE FE FF FF 01 00 00 00 A7 38 01 00   ...................8..
462  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
484  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
506  00 00 00 00 55 AA                                                   ....U.
```

1. Is this volume bootable (Offset 446, 0x00 = No, 0x80 = Yes)?
    a. 0x00: No
2. What is the size of the volume in sectors (Offset 458–461)?
    a. 0x000138A7 = 0xA7380100 (Little Endian) = 80039

7. Extract the GPT Header using dd

```
$ dd if=GPT.dmg skip=1 count=1 > GPT-DMG-GPTHeader
```

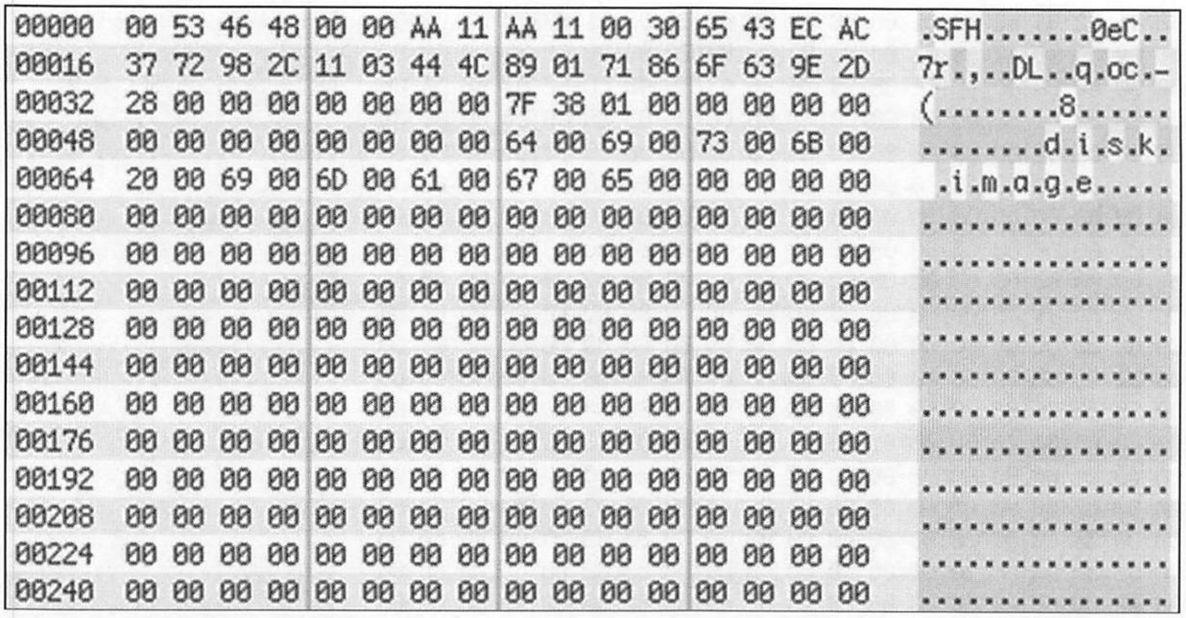


```
000  45 46 49 20 50 41 52 54 00 00 01 00 5C 00 00 00 EA 9E 31 07 00 00 00 00   EFI PART....\.....1.....
024  01 00 00 00 00 00 00 00 A7 38 01 00 00 00 00 00 22 00 00 00 00 00 00 00   .........8......".......
048  86 38 01 00 00 00 00 00 4D 95 5E BE 79 35 46 43 B5 48 EC 52 B7 A0 5A C5   .8......M.^.y5FC.H.R..Z.
072  02 00 00 00 00 00 00 00 80 00 00 00 80 00 00 00 9B 87 0B C9 00 00 00 00   ........................
096  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
144  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
168  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
192  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
216  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
264  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
288  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
312  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
336  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
360  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
384  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
408  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
432  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
456  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
480  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ........................
504  00 00 00 00 00 00 00 00                                                   ........
```

1. What is the signature (Offset 0–7)?
   a. “EFI PART”
2. What is the LBA of the GPT Header (this file) (Offset 24–31)?
   a. 0x0100000000000000 = 1 (Little Endian)
3. What is the LBA of the Backup/Secondary GPT Header (Offset 32–39)?
   a. 0xA738010000000000 = 80039 (Little Endian)
4. What is the GUID of the partition (Offset 56–71)?
   a. (0x4D955EBE79354643B548EC52B7A05AC5)
   b. BE5E954D-3579-4346-B548-EC52B7A05AC5
   c. The first three parts of each GUID are little endian; the last two are big endian.
5. Starting LBA of the GPT Partition Table (Offset 72–79)?
   a. 0x0200000000000000 = 2 (Little Endian)

8. Extract the GPT Table using dd

```
$ dd if=GPT.dmg skip=2 count=1 > GPT-DMG-GPTTable
```



```
00000  00 53 46 48 00 00 AA 11 AA 11 00 30 65 43 EC AC  .SFH.......0eC..
00016  37 72 98 2C 11 03 44 4C 89 01 71 86 6F 63 9E 2D  7r.,..DL..q.oc.-
00032  28 00 00 00 00 00 00 00 7F 38 01 00 00 00 00 00  (.......8.......
00048  00 00 00 00 00 00 00 00 64 00 69 00 73 00 6B 00  ........d.i.s.k.
00064  20 00 69 00 6D 00 61 00 67 00 65 00 00 00 00 00   .i.m.a.g.e.....
00080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00096  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00112  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00128  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00144  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00176  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00192  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00208  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00224  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

1. What is the partition type GUID, and what type of partition is it (Offset 0–15)?
    a. (0x005346480000AA11AA1100306543ECAC)
    b. 48465300-0000-11AA-AA11-00306543ECAC
    c. HFS+ Partition
2. What is the unique GUID for the partition (Offset 16–31)?
    a. (0x3772982C1103444C890171866F639E2D)
    b. 2C987237-0311-4C44-8901-71866F639E2D
3. What is the starting LBA for the partition (Offset 32–39)?
    a. 0x2800000000000000 = 40 (Little Endian)
4. What is the ending LBA for the partition (Offset 40–47)?
    a. 0x7F38010000000000 = 79999 (Little Endian)
5. What is the name of the partition (Offset 56+)?
    a. "`disk image`" (Unicode)

9. Use the `hdiutil imageinfo GPT.dmg` command and review the output.

- Check your answers with this command or make your life much easier in the future!

```
$ hdiutil imageinfo GPT.dmg
```

Extra Credit:

Parse your own GPT Header and Table of your analysis system. Use `dd` to extract files; you will need to use `sudo` with the command or you will get a "Permission denied" error. This may not work with FileVault disks; try using a different disk or volume.

Lab: Key Takeaways

- **Mac OS X uses the GUID Partition Table.**

- **There are many native and open-source command-line utilities on Mac OS X to view and parse disks and partitions.**

- **Command-line tools will easily parse what you can do by hand, but you can learn by doing it the hard way (also more fun?).**

This page intentionally left blank.

Lab 2.1: Mac and iOS Triage

Objectives

- Review files that are can provide triage information.
- Get familiar with the MacOS command line and Blacklight.

Exercise Preparation

(Note: Some of this might already be accomplished via earlier exercises, but this is the state that we hope your system is in prior to the start of this exercise. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this exercise.)

1. **Software Preparation**: The following tools will be used in this exercise:
    - Terminal.app
        i. Locate and open the native OS X Terminal.app from /Applications/Utilities/.
    - Xcode.app
        i. Locate and open the Xcode.app from /Applications/.
    - SQLite Database Browser
        i. You will be using the SQLite Database Browser (Applications/sqlitebrowser.app)
        ii. This tool is available on your USB drive in the Tools directory.
        iii. The SQLite Manager is available at http://sqlitebrowser.org/
    - Blacklight.app
        i. Locate and open the Blacklight.app from /Applications/Blacklight 201# Release #/Blacklight.app
        ii. This tool is available on your USB drive in the Tools directory.

2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Open the FOR518.blacklight BlackLight Case file.**

4. **Mount David Lightman's Mac forensic image (`galaga.E01`).**

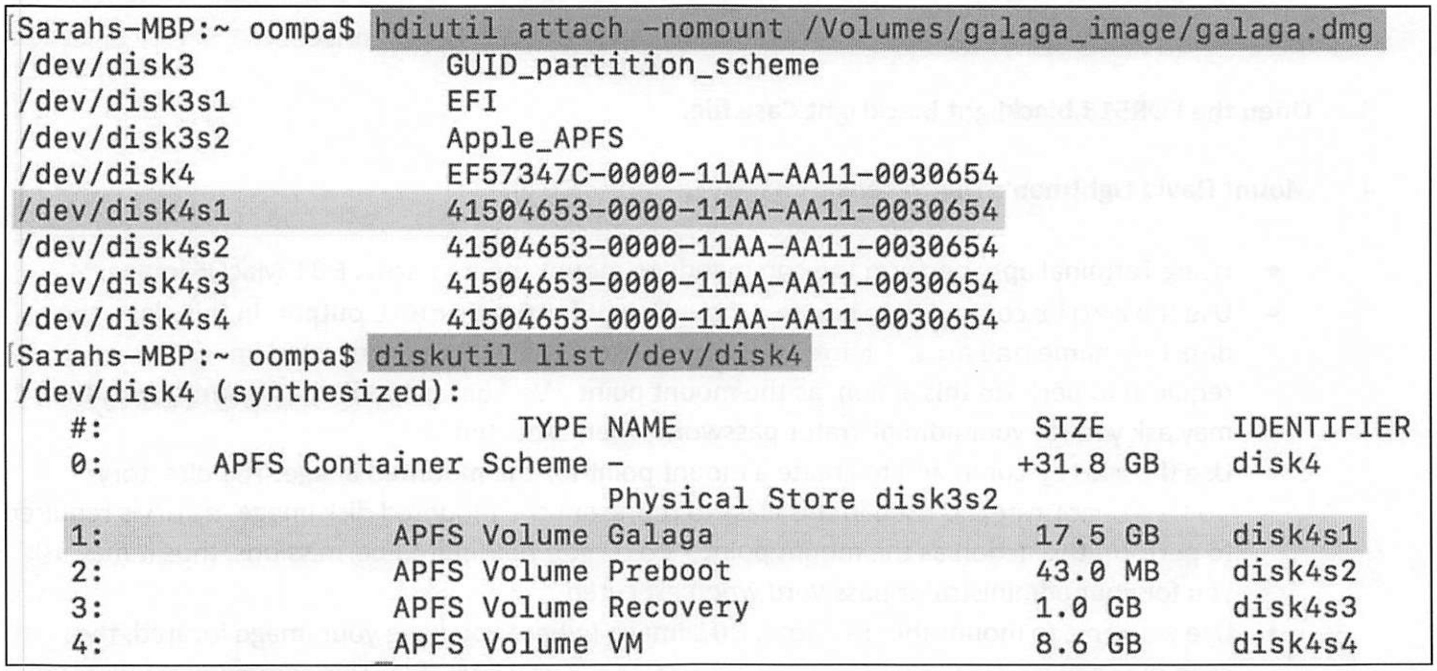
    - Using Terminal.app, perform the commands to mount the `galaga.E01` MacOS image.
    - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
    - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
    - Use `xmount` to mount the `galaga.E01` image (where you have your image located, the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you

to use the sudo command, thus it may ask you for your administrator password when executed.

  - --in – Tells xmount what input file type to expect; our images are in a compressed EWF format.
  - --out – Tells xmount what output format you want; we want a DMG file so we can mount it in Finder.
  - Input File – Where the image file is located on your system.
  - Mount Point – Newly created mount point /Volumes/galaga_image specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Uses the hdiutil command with the "attach" verb to make the newly created DMG volume available. Use the -nomount argument to suppress mounting (for now). The output from this command will display several /dev/disk#; use the appropriate disk device in the next command.
  - APFS disks will show many /dev/disk* options in the hdiutil output. The one we want to mount is the user's MacOS volume. We can use the command "diskutil list /dev/disk4" on the synthesized disk to determine which is likely the user's MacOS volume. David Lightman's volume is named "Galaga," highlighted in the example below. We will use /dev/disk4s1 in the next command. **Be aware that yours may be mounted on a different disk number!**



```
Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3              GUID_partition_scheme
/dev/disk3s1            EFI
/dev/disk3s2            Apple_APFS
/dev/disk4              EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1            41504653-0000-11AA-AA11-0030654
/dev/disk4s2            41504653-0000-11AA-AA11-0030654
/dev/disk4s3            41504653-0000-11AA-AA11-0030654
/dev/disk4s4            41504653-0000-11AA-AA11-0030654
Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the mount_apfs command with the following parameters to mount the /dev/disk#s# (from the previous command) to the /Volumes/galaga_mounted/ mount point. This command requires you to use the sudo command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
    - -o – Options:
        - rdonly: Mount in read-only mode.
        - noexec: Do not allow execution of binaries on mounted system.
        - noowners: Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

5. **Sanity Check**
    - You can access this newly created mounted drive on /Volumes/galaga_mounted/, thus all command line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
    - Use the ls -l command to view the contents in the terminal to (hopefully) view the macOS directory structure. You should see an account for "dlightman" in the Users directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

6. *****When Needed***: Image Unmount Instructions**
    - Use the diskutil list command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "(disk image)" versus the one labeled "(synthesized)." In my example, it would be /dev/disk3.
    - Use the diskutil eject command on the disk you would like to eject.
    - Use the mount command to view the list of mounted disks. Find the disk that you want to unmount (likely /Volumes/galaga_image/ if you are following the naming scheme from the examples).
    - Use the umount command with the mount point to unmount the disk. You will have to use the sudo command.
    - *****WARNING***: If you are in the mounted image in Terminal, or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list
$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

Exercise: Questions

Mac Triage

Perform the following in David's mounted image on the command line:

1. Mac Version Information

- Use the `cd` command to navigate to the `CoreServices` directory.
- Use the `open` command to open the `SystemVersion.plist` file.

```
$ cd /Volumes/galaga_mounted/System/Library/CoreServices

$ open SystemVersion.plist
```

1. What version of macOS is this system running?

____

2. System Installation Date

- Use the `cd` command to navigate to the `/private/var/db` directory.
- Use the `ls -la` command to view all files in this directory.

```
$ cd /Volumes/galaga_mounted/private/var/db/

$ ls -la
```

1. What is the likely date of system installation?

____

3. System Time Zone and Language Settings

- Use the `cd` command to navigate to the `/etc` directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory
- Use the `ls -l` command on the `localtime` file.

```
$ cd /Volumes/galaga_mounted/etc/

$ ls -l

$ ls -l localtime
```

1. What time zone is in use on this system?

____

4. **Review the user property list for user `dlightman`.**
   - Get a root shell (`sudo -s`).
   - Change directory to view the user property lists on the system.
     i. `/Volumes/galaga_mounted/private/var/db/dslocal/nodes/Default/users`
   - Using the `cp` command, copy the `dlightman.plist` property list to a directory of your choice.
   - Use the `chown` command to change the ownership to your user account name.
   - **VERY IMPORTANT: Exit the root shell.**
   - Use the `open` command to open and view the `dlightman.plist` property list in Xcode.

```
$ sudo -s

# cd
/Volumes/galaga_mounted/private/var/db/dslocal/nodes/Default/users

# cp dlightman.plist ~/FOR518

# chown <your username> ~/FOR518/dlightman.plist

# exit

$ open -a Xcode ~/FOR518/dlightman.plist
```

5. **Review the `dlightman`'s user property list.**
   1. What is the user's "Real Name"?

      ____

   2. What is the path to the user's home directory?

      ____

   3. What is the user's UID (User ID)?

      ____

   4. What is the user's linked iCloud identity?

      ____

   5. When was this user account created (Hint: `accountPolicyData` Key)?
      - 10.15 users can use PlistBuddy instead in a couple different ways.
        i. XML Output: `/usr/libexec/PlistBuddy -c Print:accountPolicyData dlightman.plist`

ii. Plutil Output: `/usr/libexec/PlistBuddy -c Print:accountPolicyData:0 dlightman.plist | plutil -p -`

____

6. **Review the time zone and language settings for `dlightman`.**
   - Use the `cd` command to navigate to the system preferences directory.
     i. `/Volumes/galaga_mounted/Library/Preferences`
   - Use the `ls -la` command to view all files in this directory. Note the contents of this directory.
   - Use the `open` command to open the `.GlobalPreferences.plist` file.

```
$ cd /Volumes/galaga_mounted/Library/Preferences

$ ls -la

$ open .GlobalPreferences.plist
```

1. What city is used to determine the time zone used?

____

2. What are the location coordinates? Do they match up with the city listed?

____

3. What is the primary language setting used?

____

7. **Network Settings**
   - Use the `cd` command to navigate to the `SystemConfiguration` directory.
   - Use the `ls -la` command to view all files in this directory. Note the contents of this directory.
   - Use the `open` command to open all the plist files in this directory.

```
$ cd SystemConfiguration/

$ ls -la

$ open *.plist
```

- Review the `NetworkInterfaces.plist` file.

1. What model system is this?

____

2. What is the MAC address for the Wi-Fi interface?

____

- Review the `com.apple.airport.preferences.plist` file.

3. How many "remembered" Wi-Fi networks are there?

   ____

4. Provided all Wi-Fi networks are available, which is the name of the first access point to be connected to via user configuration?

   ____

5. What is the name of the network that was last accessed on January 21, 2018 (UTC)?

   ____

6. Which access point has WPA2 Personal security implemented?

   ____

- Determine what IP Address this system had at the time of collection. Navigate to the `leases` directory.

```
$ cd /Volumes/galaga_mounted/private/var/db/dhcpclient/leases/

$ ls -la

$ plutil -p en0-1\,b8\:e8\:56\:37\:ec\:6
```

7. When this system was last connected to `CrystalPalace`, what was its IP address?

   ____

8. **MRUs: Open and Review the contents of the `dlightman`'s /Library/Preferences directory.**
   - Use the `cd` command to change the directory to `dlightman`'s /Library/Preferences/ directory.
   - Open the Finder plist file in Xcode. While default settings should open it in Xcode, to explicitly open it in Xcode, use the command `open -a Xcode com.apple.finder.plist`.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Preferences/

$ open -a Xcode com.apple.finder.plist
```

9. **Review the Recent Folders.**
   - Review the contents of the `com.apple.finder.plist` file.

1. Under the FXRecentFolders key, where did the folder "iPhone" exist? (10.15 users will need to extract the BLOB using another tool, try PlistBuddy.)

- Review the newer SFL MRU files (/Users/dlightman/Library/Application\ Support/com.apple.sharedfilelist).
- Attempt to open the com.apple.LSSharedFileList.RecentApplications.sfl2 plist with Xcode. These will fail because of the file extension.
- Use plutil to make a readable copy of this plist file and save it as recentapps.txt.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/
Application\ Support/com.apple.sharedfilelist

$ open com.apple.LSSharedFileList.RecentApplications.sfl2

$ open -a Xcode com.apple.LSSharedFileList.RecentApplications.sfl2

$ plutil -p com.apple.LSSharedFileList.RecentApplications.sfl2 >
~/FOR518/recentapps.txt

$ open ~/FOR518/recentapps.txt
```

- Determine the most recent application in the "list" manually; follow the steps from the slides.

2. What is the name of the most recent app used (the first app in the list)?

3. What directory was this application run from?

10. Review the recent documents for the TextEdit application.

- cd into com.apple.LSSharedFileList.ApplicationRecentDocuments
- Review the contents of the com.apple.textedit.sfl2 file in the same method used above.

```
$ cd com.apple.LSSharedFileList.ApplicationRecentDocuments

$ plutil -p com.apple.textedit.sfl2 > ~/FOR518/recenttextedit.txt

$ open ~/FOR518/recenttextedit.txt
```

1. How many documents are in this list?

11. **Try running the MacMRU python script.**
    - Find the script in the exercise folder for this exercise. **Be sure to check your file paths; the location of the MacMRU script will likely be different depending on where you unarchived your files. Make sure the file `ccl_bplist.py` is also in the directory.**
    - Run it on the `dlightman`'s directory mounted image and output it to a file called `galaga_mrus.txt.`
    - Run it again, using the "`—blob_parse_human`" option and save it to a text file called `galaga_mrus_blobs.txt.`
    - Review each file using the open command.
    - For `com.apple.LSSharedFileList.RecentApplications.sfl2` and `com.apple.textedit.sfl2`, answer the following:

1. From what location was the Home Printer application run?

2. What is the filename and path of the document that was most recently opened with the TextEdit application?

```
$ python macMRU.py /Volumes/galaga_mounted/Users/dlightman/ >
~/FOR518/galaga_mrus.txt

$ python macMRU.py --blob_parse_human
/Volumes/galaga_mounted/Users/dlightman >
~/FOR518/galaga_mrus_blobs.txt

$ open ~/FOR518/galaga_mrus*
```

- *****OPTIONAL*****: Try running this script on your own system. It might take a while, so feel free to continue to the next section.

iOS Triage

Perform the following in David's iPhone images in BlackLight:

12. **iOS Device Information**
    - **In David's iPhone, select the "physical_logical" acquisition.**
    - Review the `general.log` file located in either of the following paths:
        i. `/private/var/logs/AppleSupport/general.log`
        ii. `/private/var/mobile/Library/Logs/AppleSupport/general.log`

1. What version of iOS is this phone running?

2. What is the serial number of this phone (last four digits)?

3. Model of the Phone: translate "comma'ed" make/model into a commercially known model.

   - Review the file at: `/private/var/containers/Data/System/BB422B72-4829-4993-ABC7-3D6E54E01FBE/Library/activation_records/activation_record.plist`

4. What are the last four digits of the IMEI?

   - **Now select <u>any</u> of David's iPhone acquisitions.**
   - Review the file at `[/private/var] /wireless/Library/Preferences/com.apple.commcenter.plist`

5. What was the phone number of this device when it was imaged?

6. What is the ICCID number for the device (last four digits)?

7. Who was the provider of the device at the time of acquisition?

   - **Now select any of David's iPhone acquisitions.**
   - Review the file at `[/private/var] /mobile/Library/Preferences/com.apple.purplebuddy.plist`

8. On what day was this device likely setup?

13. Network Settings

- **Select any of David's iPhone acquisitions.**
- Review the `[/private/var] /preferences/SystemConfiguration/com.apple.wifi.plist` file.

1. How many "known" Wi-Fi networks are there?

2. On what day was "`FlyDulles`" last potentially used (local system time)?

____

14. **Accounts**
    - **Select any of David's iPhone acquisitions.**
    - Review the `[/private/var]` `/preferences/SystemConfiguration/com.apple.accounts.exists.plist` file.

1. How many Google accounts are set up on this device?

____

- Review the `[/private/var]/mobile/Library/Accounts/Accounts3.sqlite` database.

2. What is the username for the Gmail account set up on this device?

____

15. **iOS MRU: Recent Applications**
    - **Select David's Physical iPhone acquisition.**
    - Review the `[/private/var]` `/mobile/Library/Preferences/com.apple.springboard.plist` file.

1. What are the three most recently used applications (assuming the user did not clear running applications)?

____

____

____

Exercise: Step-By-Step

Mac Triage

Perform the following in David's mounted image on the command line:

1. **Mac Version Information**
   - Use the `cd` command to navigate to the `CoreServices` directory.
   - Use the `open` command to open the `SystemVersion.plist` file.

```
$ cd /Volumes/galaga_mounted/System/Library/CoreServices

$ open SystemVersion.plist
```

   1. What version of macOS is this system running?
      a. `ProductVersion` Key = 10.13.1

2. **System Installation Date**
   - Use the `cd` command to navigate to the `/private/var/db` directory.
   - Use the `ls -la` command to view all files in this directory.

```
$ cd /Volumes/galaga_mounted/private/var/db/

$ ls -la
```

   1. What is the likely date of system installation?
      a. November 13, 2017 (UTC)

i. Use the `stat -x` command to determine MAC times for `.AppleSetupDone` and `.AppleInstallType.plist` files.
ii. To view the UTC/Unix Epoch timestamps, use `-r` instead of `-x`.
iii. Change your terminal time zone to UTC temporarily using this command:
    1. `export "TZ=UTC"`

3. **System Time Zone and Language Settings**
    - Use the `cd` command to navigate to the `/etc` directory.
    - Use the `ls -l` command to view all files in this directory. Note the contents of this directory.
    - Use the `ls -l` command on the `localtime` file.

```
$ cd /Volumes/galaga_mounted/etc/

$ ls -l

$ ls -l localtime
```

1. What time zone is in use on this system?
    a. America/New_York

4. **Review the user property list for user `dlightman`.**
    - Get a root shell (`sudo -s`).
    - Change the directory to view the user property lists on the system.
        i. `/Volumes/galaga_mounted/private/var/db/dslocal/nodes/Default/users`
    - Using the `cp` command, copy the `dlightman.plist` property list to a directory of your choice.
    - Use the `chown` command to change the ownership to your user account name.
    - **VERY IMPORTANT: Exit the root shell.**
    - Use the `open` command to open and view the `dlightman.plist` property list in Xcode.

```
$ sudo -s

# cd
/Volumes/galaga_mounted/private/var/db/dslocal/nodes/Default/users

# cp dlightman.plist ~/FOR518

# chown <your username> ~/FOR518/dlightman.plist

# exit

$ open -a Xcode ~/FOR518/dlightman.plist
```

5. Review the `dlightman`'s user property list.

1. What is the user's "Real Name"?
   - `realname` Key = David Lightman
2. What is the path to the user's home directory?
   - /Users/dlightman
3. What is the user's UID (User ID)?
   - `uid` Key = 501
4. What is the user's linked iCloud identity?
   - d.l1ghtm4n@gmail.com
   - Extract the contents of the `LinkedIdentity` Key into a text viewer. Review the contents of the "`full name`" key of this embedded XML plist file.
5. When was this user account created (Hint: `accountPolicyData` Key)?
   - 2017-11-13 01:26:28 Mon UTC
   - Extract the contents of the `accountPolicyData` Key, input into a hex editor, and save as a .plist file. Open the plist file in Xcode.
   - 10.15 users can use PlistBuddy instead in a couple different ways.
     i. XML Output: `/usr/libexec/PlistBuddy -c Print:accountPolicyData dlightman.plist`
     ii. Plutil Output: `/usr/libexec/PlistBuddy -c Print:accountPolicyData:0 dlightman.plist | plutil -p -`
   - The `creationTime` key holds the time the account was created: copy the first 10 digits (1510536388, remove the commas), and convert it in the Terminal with **`date -r 1510536388`**.

6. Review the time zone and language settings for `dlightman`.

- Use the `cd` command to navigate to the system preferences directory.
  i. `/Volumes/galaga_mounted/Library/Preferences`
- Use the `ls -la` command to view all files in this directory. Note the contents of this directory.
- Use the `open` command to open the `.GlobalPreferences.plist` file.

```
$ cd /Volumes/galaga_mounted/Library/Preferences

$ ls -la

$ open .GlobalPreferences.plist
```

1. What city is used to determine the time zone used?
   a. Arlington
      i. `com.apple.TimeZonePref.Last_Selected_City` Key
2. What are the location coordinates? Do they match up with the city listed?
   a. Yes, they do. 38.89076, -77.08475 = Arlington, VA
3. What is the primary language setting used?
   a. en_US: US English
      i. `AppleLocale` or `AppleLanguages` Keys

7. Network Settings

- Use the `cd` command to navigate to the `SystemConfiguration` directory.
- Use the `ls -la` command to view all files in this directory. Note the contents of this directory.
- Use the `open` command to open all the plist files in this directory.

```
$ cd SystemConfiguration/

$ ls -la

$ open *.plist
```

- Review the `NetworkInterfaces.plist` file.

1. What model system is this?
   a. MacBookPro11,1
2. What is the MAC address for the Wi-Fi interface?
   a. <b8e85637 ec06> = b8:e8:56:37:ec:06
   b. `Item 0` | `IOMACAddress` Key for the interface labeled `en0`, `IEEE80211` and/or `Wi-Fi`.

- Review the `com.apple.airport.preferences.plist` file.

3. How many "remembered" Wi-Fi networks are there?
   a. Three
   b. Number of items under `KnownNetworks` Key
4. Provided all Wi-Fi networks are available, which is the name of the first access point to be connected to via user configuration?
   a. `CrystalPalace`
   b. The `PreferredOrder` key contains the order in which each will be connected. Item 0 is the most preferred.
   c. Take the key "`wifi.ssid.<43727973 74616c50 616c6163 65>`" and match it with the entry under `KnownNetworks` to find `CrystalPalce` in the key `SSIDString`.
5. What is the name of the network that was last accessed on January 21, 2018 (UTC)?
   a. `shmoocon`
   b. Look for the `LastConnected` Key timestamp under each access point.
6. Which access point has WPA2 Personal security implemented?
   a. `CrystalPalace`
   b. `SecurityType` Key contains WPA2 Personal (versus Open)

- Determine what IP Address this system had at the time of collection. Navigate to the `leases` directory.

```
$ cd /Volumes/galaga_mounted/private/var/db/dhcpclient/leases/
$ ls -la

$ plutil -p en0-1\,b8\:e8\:56\:37\:ec\:6
```

7. When this system was last connected to `CrystalPalace`, what was its IP address?
    a. 192.168.101.138
    b. `IPAddress` Key

8. MRUs: Open and Review the contents of the `dlightman`'s /Library/Preferences directory.

- Use the `cd` command to change the directory to `dlightman`'s /Library/Preferences/ directory.
- Open the Finder plist file in Xcode. While default settings should open it in Xcode, to explicitly open it in Xcode, use the command `open -a Xcode com.apple.finder.plist`.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Preferences/

$ open -a Xcode com.apple.finder.plist
```

9. Review the Recent Folders.

- Review the contents of the `com.apple.finder.plist` file.

1. Under the `FXRecentFolders` key, where did the folder "`iPhone`" exist? (10.15 users will need to extract the BLOB using another tool, try PlistBuddy.)
    a. `/Volumes/WDPassport/MyBackups/iPhone`
    b. Extract the `file-bookmark` data for the folder `iPhone` and view it in a hex editor. If you are on 10.15, use `PlistBuddy` and `xxd`.
        i. `/usr/libexec/PlistBuddy -c Print:FXRecentFolders:0:file-bookmark com.apple.finder.plist | xxd`

- Review the newer SFL MRU files (`/Users/dlightman/Library/Application\ Support/com.apple.sharedfilelist`).
- Attempt to open the `com.apple.LSSharedFileList.RecentApplications.sfl2` plist with Xcode. These will fail because of the file extension.
- Use `plutil` to make a readable copy of this plist file and save it as `recentapps.txt`.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/
Application\ Support/com.apple.sharedfilelist

$ open com.apple.LSSharedFileList.RecentApplications.sfl2

$ open -a Xcode com.apple.LSSharedFileList.RecentApplications.sfl2

$ plutil -p com.apple.LSSharedFileList.RecentApplications.sfl2 >
~/FOR518/recentapps.txt
```

```
$ open ~/FOR518/recentapps.txt
```

- Determine the most recent application in the "list" manually; follow the steps from the slides.

2. What is the name of the most recent app used (the first app in the list)?
   a. Home Printer

10. Review the recent documents for the TextEdit application.

- cd into `com.apple.LSSharedFileList.ApplicationRecentDocuments`
- Review the contents of the `com.apple.textedit.sfl2` file in the same method used above.

```
$ cd com.apple.LSSharedFileList.ApplicationRecentDocuments

$ plutil -p com.apple.textedit.sfl2 > ~/FOR518/recenttextedit.txt

$ open ~/FOR518/recenttextedit.txt
```

1. How many documents are in this list?
   a. Two

11. Try running the MacMRU python script.

- Find the script in the exercise folder for this exercise. **Be sure to check your file paths; the location of the MacMRU script will likely be different depending on where you unarchived your files. Make sure the file `ccl_bplist.py` is also in the directory.**
- Run it on the `dlightman`'s directory mounted image and output it to a file called `galaga_mrus.txt.`
- Run it again, using the "`—blob_parse_human`" option and save it to a text file called `galaga_mrus_blobs.txt.`
- Review each file using the open command.
- For `com.apple.LSSharedFileList.RecentApplications.sfl2` and `com.apple.textedit.sfl2`, answer the following:

3. From what location was the Home Printer application run?
   a. `"/Users/dlightman/Library/Printers/Home Printer.app"`
   b.
4. What is the filename and path of the document that was most recently opened with the TextEdit application?
   a. `/Users/dlightman/Desktop/out_logfile.txt`

```
$ python macMRU.py /Volumes/galaga_mounted/Users/dlightman/ >
~/FOR518/galaga_mrus.txt

$ python macMRU.py --blob_parse_human
/Volumes/galaga_mounted/Users/dlightman >
~/FOR518/galaga_mrus_blobs.txt

$ open ~/FOR518/galaga_mrus*
```

- *****OPTIONAL*****: Try running this script on your own system. It might take a while, so feel free to continue to the next section.

iOS Triage

Perform the following in David's iPhone images in BlackLight:

12. **iOS Device Information**
    - **In David's iPhone, select the "physical_logical" acquisition.**
    - Review the `general.log` file located in either of the following paths:
        i. `/private/var/logs/AppleSupport/general.log`
        ii. `/private/var/mobile/Library/Logs/AppleSupport/general.log`

1. What version of iOS is this phone running?
    a. 11.0.3
2. What is the serial number of this phone (last four digits)?
    a. C6KSC32B**HG7L**
3. Model of the Phone: translate "comma'ed" make/model into a commercially known model.
    a. iPhone9,3 = iPhone 7

- Review the file at: `/private/var/containers/Data/System/BB422B72-4829-4993-ABC7-3D6E54E01FBE/Library/activation_records/activation_record.plist`

4. What are the last four digits of the IMEI?
    a. 35920407080**8295** (extract the `AccountToken` Key)

- **Now select <u>any</u> of David's iPhone acquisitions.**
- Review the file at `[/private/var] /wireless/Library/Preferences/com.apple.commcenter.plist`

5. What was the phone number of this device when it was imaged?
    a. `NetworkPhoneNumber` Key = `+447848916073`
6. What is the ICCID number for the device (last four digits)?
    a. `ICCID` Key = `8944200116623054965`
7. Who was the provider of the device at the time of acquisition?
    a. `CarrierBundleName` Key = 23420 = 3 Network (Look this up on http://www.imei.info/operator-codes/)

- **Now select any of David's iPhone acquisitions.**
- Review the file at `[/private/var]` `/mobile/Library/Preferences/com.apple.purplebuddy.plist`

8. On what day was this device likely setup?
    a. November 12, 2017 (Review the `GuessedCountry` "at" time, or `SetupLastExit` Key.)

13. Network Settings

- **Select any of David's iPhone acquisitions.**
- Review the `[/private/var]` `/preferences/SystemConfiguration/com.apple.wifi.plist` file.

1. How many "known" Wi-Fi networks are there?
    a. 18
    b. `"List of known networks"` Key
2. On what day was `"FlyDulles"` last potentially used (local system time)?
    a. February 11, 2018 (Check the `lastJoined` and `LastAutoJoined` keys.)

14. Accounts

- **Select any of David's iPhone acquisitions.**
- Review the `[/private/var]` `/preferences/SystemConfiguration/com.apple.accounts.exists.plist` file.

1. How many Google accounts are set up on this device?
    a. 1 – There is a "1" in the "`exists`" key and a "1" in the related "`count`" key.

- Review the `[/private/var]/mobile/Library/Accounts/Accounts3.sqlite` database.

2. What is the username for the Gmail account set up on this device?
    a. Find the type number for Gmail in the "`ZACCOUNTTYPE`" table. It is a "36" (Z_PK). Match that up with the information found in the "`ZACCOUNT`" table. Look for a "36" in the "`ZACCOUNTTYPE`" column. (The entry should be for Z_PK = 19.) The email is `d.l1ghtm4n@gmail.com`.

15. iOS MRU: Recent Applications

- **Select David's Physical iPhone acquisition.**
- Review the `[/private/var]` `/mobile/Library/Preferences/com.apple.springboard.plist` file.

1. What are the three most recently used applications (assuming the user did not clear running applications)?
    a. `SBRecentAppLayoutsPlistRepresentation` Key (First three)

b. Most Recent = Safari: com.apple.MobileSafari
c. Settings: com.apple.Preferences
d. Messages: com.apple.MobileSMS

Exercise: Key Takeaways

- **Determine where triage information is stored for Mac and iOS devices.**
- **Get comfortable with some MacOS command lines.**
- **Get comfortable with the BlackLight application interface and nuances.**

This page intentionally left blank.

Lab 2.2: File System Fun!

Objectives

- Learn how the file system metadata can be found in different files and databases.
- Find various ways to look for forensic artifacts that may be useful in an investigation that are not common to other systems other than Mac and iOS.

Lab Preparation

(Note: Some of this might already be accomplished via earlier Labs, but this is the state that we hope your system is in prior to the start of this Lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this Lab.)

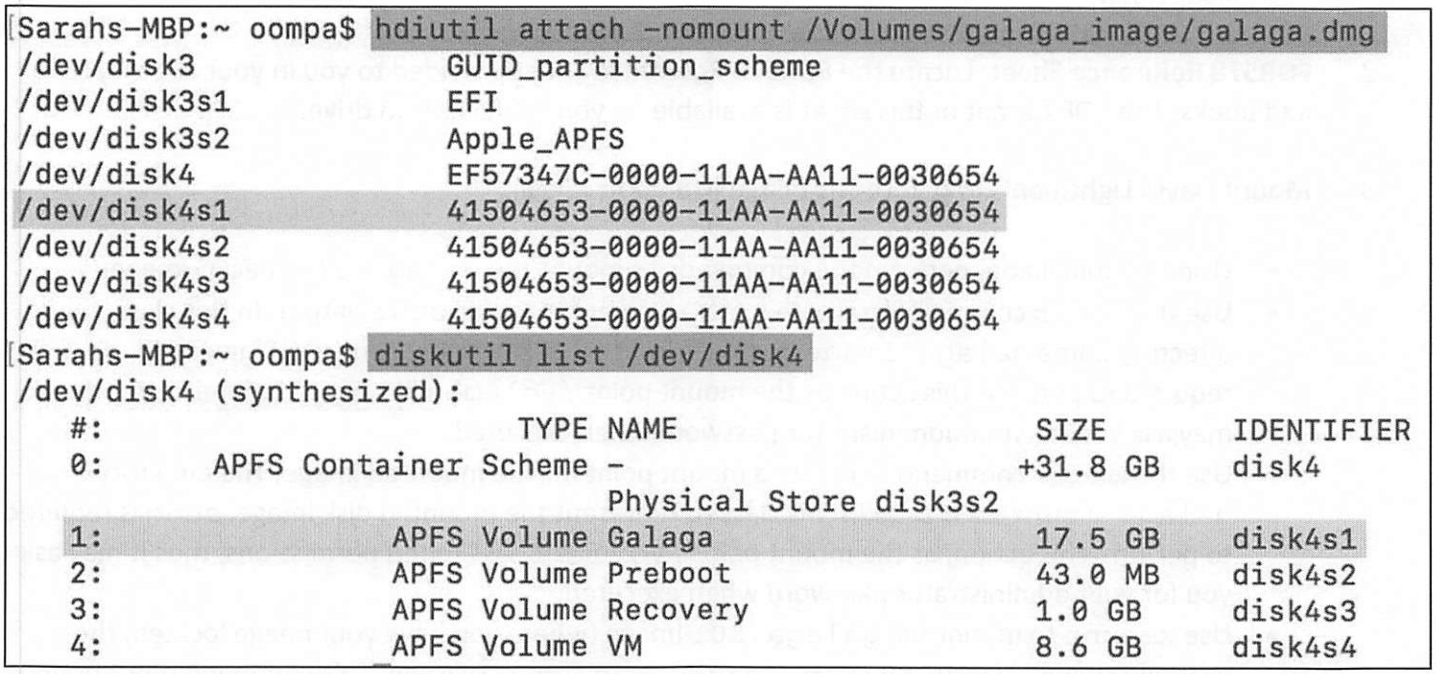
1. **Software Preparation**: The following tools will be used in this Lab:
    - Terminal.app
        i. Locate and open the native OS X Terminal.app from /Applications/Utilities/
    - Xcode.app
        i. Locate and open the Xcode.app from /Applications/.
2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Mount David Lightman's Mac forensic image (`galaga.E01`).**

    - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
    - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
    - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
    - Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.
        - `--in` – Tells `xmount` what input file type to expect, our images are in a compressed EWF format.
        - `--out` – Tells `xmount` what output format you want; we want a DMG file so we can mount it in Finder.
        - `Input File` – Where the image file is located on your system.
        - `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Uses the `hdiutil` command with the "`attach`" verb to make the newly created DMG volume available. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display several `/dev/disk#`, use the appropriate disk device in the next command.
    - APFS disks will show many `/dev/disk*` options in the `hdiutil` output. The one we want to mount is the user's macOS volume. We can use the command "`diskutil list /dev/disk4`" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "`Galaga`," highlighted in the example below. We will use `/dev/disk4s1` in the next command. **Be aware that yours may be mounted on a different disk number!**



```
Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3              GUID_partition_scheme
/dev/disk3s1            EFI
/dev/disk3s2            Apple_APFS
/dev/disk4              EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1            41504653-0000-11AA-AA11-0030654
/dev/disk4s2            41504653-0000-11AA-AA11-0030654
/dev/disk4s3            41504653-0000-11AA-AA11-0030654
/dev/disk4s4            41504653-0000-11AA-AA11-0030654
Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the `mount_apfs` command with the following parameters to mount the `/dev/disk#s#` (from the previous command) to the `/Volumes/galaga_mounted/` mount point. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
    - `-o` – Options:
        - `rdonly` – Mount in read-only mode.
        - `noexec` – Do not allow execution of binaries on the mounted system.
        - `noowners` – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

4. **Sanity Check**
   - You can access this newly created mounted drive on `/Volumes/galaga_mounted/`, thus all command line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
   - Use the `ls -l` command to view the contents in the terminal to (hopefully) view the macOS directory structure. You should see an account for "`dlightman`" in the `Users` directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

5. *****When Needed***: Image Unmount Instructions**
   - Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`." In my example, it would be `/dev/disk3`.
   - Use the `diskutil eject` command on the disk you would like to eject.
   - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/` if you are following the naming scheme from the examples).
   - Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command.
   - *****WARNING***: If you are in the mounted image in Terminal or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

Lab: Questions

Perform the following in David's mounted image on the command line:

1. **Review the `dlightmans`'s Downloads directory for Extended Attributes.**
   - Use the `cd` command to change the directory to the `dlightman`'s Downloads directory.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Downloads/

$ ls -l
```

   1. When was the file "`Firefox 58.0.2.dmg`" downloaded? (UTC)?

      ____

   2. What browser application downloaded the file "`Firefox 58.0.2.dmg`"?

      ____

   3. Which file was transferred to the system via a Messages File Transfer?

      ____

   4. Which DMG file in the Downloads directory was the only one NOT double-clicked and opened?

      ____

2. **Review the `dlightmans`'s File System Events Store Database.**
   - Use the `cd` command to change directory to the `dlightmans`'s `.fseventsd` directory.
   - List the Files with the `ls` command.
   - Determine the file types with the `file` command.

```
$ cd /Volumes/galaga_mounted/.fseventsd

$ ls -l

$ file *
```

   - Locate the FSEParser python script in your lab files for this Lab. Use it to parse these files. **Be sure to check your file paths; the location of the FSEParser script will likely be different depending on where you unarchived your lab files. Note: There is no space before ".fsevents".**
     i. "-t" is for what type of evidence, we will be using 'folder' here.
     ii. "-s" is for the source directory (the directory you are currently in).
     iii. "-o" is for the output directory, your FOR518 directory.
   - Move into the ~/FOR518 directory and review the file output from the script. The files are in a directory name `FSE_Reports/`. Find the one starting with the file name, "FSEvents.sqlite". You should see a text file, a TSV file, and one SQLite database.

- Open the database for analysis using a SQLite viewer. The SQLite database browser is being used as an example below.

```
$ python FSEParser_v4.0.py -t folder -s
/Volumes/galaga_mounted/.fseventsd  -o ~/FOR518/

$ cd ~/FOR518/FSE_Reports/

$ ls -l ~/FOR518/

$ open -a “DB Browser for SQLite” FSEvents.sqlite
```

- Use the filters in SQLite Browser to search for mounted volumes. In the “`Browse Data`” tab, type in “`/Volumes`” in the “`fullpath`” column. Review the mounted Volumes.
- Now search for DMG files on the system; focus on `dlightman`’s Desktop directory.

1. What two DMG files were located on `dlightman`’s Desktop (not inside of a sub-directory of the Desktop)?

   ____

2. Are these two separate files or one file that was renamed (hint: look at the CNID in “`node_id`” column)?

   ____

- Search for the file IMG_0030.JPG using an SQLite query in the “Execute SQL” tab.

```
select * from fsevents where fullpath like '%IMG_0030.JPG%'
```

3. How do you think this file ended up on `dlightman`’s system (staring in February 2018)?

   ____

4. This picture was later edited by Dave Lightman; what software did he use to edit it?

   ____

- Search activity for a file using the iNode/CNID 1417428.

```
select * from fsevents where node_id == 1417428
```

5. What browser downloaded this file?

   ____

6. This file was downloaded to the default downloads directory (~/Downloads); where did it move later?

- Search for a file ms-nAphDJ.gif.

```
select * from fsevents where filename == "ms-nAphDJ.gif" order by id
```

7. Where did this file come from?

3. Review dlightman's Spotlight Directory

- Use the cd command to enter the Spotlight directory.
- Use the ls -la command to view the contents of this directory. Review the contents of this directory.
- Use the open command to open the VolumeConfiguration.plist file and review the contents of this file. Note that the first time that Xcode's plist reader runs, it may prompt to add more features. If prompted, please do so.

```
$ cd /Volumes/galaga_mounted/.Spotlight-V100

$ ls -la

$ open VolumeConfiguration.plist
```

1. Are there any files or directories excluded from Spotlight indexing?

4. Spotlight: Review the Spotlight Metadata

- Use the cd command to explore the dlightman's ~/Downloads directory.
- Use the mdls command to view the files in this directory. Answer the following questions.
    i. Some students may need to use "sudo" with the mdls command.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Downloads/

$ mdls LiberiOS11.0.3.ipa                ←Repeat as necessary for each file
```

1. Where was the file LiberiOS11.0.3.ipa downloaded from?

2. On what day did the file `Impactor_0.9.44.dmg` get used last?

   ____

3. Please answer the following on the file `IMG_0007.JPG`:
   a. How did the file get transferred to this system?

      ____

   b. From whom?

      ____

   c. When?

      ____

   d. What is the Make/Model of the phone?

      ____

   e. What version of iOS was it running?

      ____

   f. Does this photo have location coordinates?

      ____

- Find photos that have locational data in them.
- Use the "`mdfind`" command to search “`-onlyin`” in the mounted volume for Dave Lightman.
- Search for items containing the metadata item for latitude.
- Find the path for the photo `IMG_0042.JPG` and perform an `mdls` on it.
  i. Some students may need to use “`sudo`” with the `mdls` command.

```
mdfind -onlyin /Volumes/galaga_mounted/ -name "kMDItemLatitude == *"

mdls “/Volumes/galaga_mounted/Users/dlightman/Library/Containers/
com.apple.cloudphotosd/Data/Library/Application Support/
com.apple.cloudphotosd/services/com.apple.photo.icloud.sharedstreams/
assets/66B292A9-9F24-4889-913C-1A90395F2338/
E17A868C-9AF1-4DED-802A-A9F7655F4065/IMG_0042.JPG”
```

1. What are the coordinates for `IMG_0042.JPG`?

   ____

2. In what major landmark was this photo taken?

   ____

5. Review the `dlightman’s Trash`.

- Use the `cd` command to change directory to the `dlightman`’s `.Trash` directory.
- Use the `ls -la` command to view the contents of this directory.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/.Trash

$ ls -la
```

1. What three files are in the trash?

   ____

   ____

2. Where did some of these files once exist?

   ____

   - Note that the file `Spectacle+1.2.zip` did not exist in the `.DS_Store` file—it's not a perfect system.

Lab: Step-by-Step

Perform the following in David's mounted image on the command line:

1. **Review the `dlightmans`'s Downloads directory for Extended Attributes.**
   - Use the `cd` command to change the directory to the `dlightman`'s Downloads directory

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Downloads/

$ ls -l
```

1. When was the file "`Firefox 58.0.2.dmg`" downloaded? (UTC)?
   a. Use "`xattr -xl`" on the files to get the extended attributes.
   b. The timestamps are located in the following attributes:
      i. `com.apple.metadata:kMDItemDownloadedDate` in the binary plist file
      ii. `com.apple.quarantine` (type **`date -r 0x5a931512`** in the Terminal to see the result of 2018-02-25 19:57:06 Sun UTC)
   c. It is far easier to get the date from `com.apple.quarantine` than it is to extract the binary plist from an extended attribute—but it's good to have other options when needed.
2. What browser application downloaded the file "`Firefox 58.0.2.dmg`"?
   a. `com.apple.quarantine` attribute: Safari
3. Which file was transferred to the system via a Messages File Transfer?
   a. `ms-nAphDJ.gif`
   b. Using "`xattr -xl *`" on all the files, look for the following attributes:
      i. `com.apple.metadata:kMDItemWhereFroms` – This contains a binary plist that shows the file was transferred from 1337jmack@gmail.com via Messages file transfer.
      ii. `com.apple.quarantine` – This contains the application that downloaded the file Messages.app.
4. Which DMG file in the Downloads directory was the only one NOT double-clicked and opened?
   a. Using "`xattr -xl *.dmg`", look for the attributes `com.apple.diskimages.fsck` and `com.apple.diskimages.recentcksum`, which indicate that a DMG was opened.
      i. The only DMG file that was not opened was `Firefox 58.0.2.dmg`. `Impactor_0.9.44.dmg` and `googlechrome.dmg` were both opened.

2. **Review the `dlightmans`'s File System Events Store Database.**
   - Use the `cd` command to change the directory to the `dlightmans`'s `.fseventsd` directory.
   - List the Files with the `ls` command.
   - Determine the file types with the `file` command.

```
$ cd /Volumes/galaga_mounted/.fseventsd

$ ls -l
$ file *
```

i. "-t" is for what type of evidence, we will be using 'folder' here.
ii. "-s" is for the source directory (the directory you are currently in).
iii. "-o" is for the output directory, your FOR518 directory.

- Move into the ~/FOR518 directory and review the file output from the script. The files are in a directory name `FSE_Reports/`. Find the one starting with the file name, "FSEvents.sqlite". You should see a text file, a TSV file, and one SQLite database.
- Open the database for analysis using a SQLite viewer. The SQLite database browser is being used as an example below.

```
$ python FSEParser_v4.0.py -t folder -s
/Volumes/galaga_mounted/.fseventsd  -o ~/FOR518/

$ cd ~/FOR518/FSE_Reports/

$ ls -l ~/FOR518/

$ open -a "DB Browser for SQLite" FSEvents.sqlite
```

- Use the filters in SQLite Browser to search for mounted volumes. In the "`Browse Data`" tab, type in "`/Volumes`" in the "`fullpath`" column. Review the mounted Volumes.
- Now search for DMG files on the system; focus on `dlightman`'s Desktop directory.

1. What two DMG files were located on `dlightman`'s Desktop (not inside of a sub-directory of the Desktop)?
    a. Filter on "`/dlightman/Desktop/`" in the "`fullpath`" column. You can filter on file extension by typing "`dmg`" in the "`filename`" column.
    b. `kl.dmg` and `kl2.dmg`
2. Are these two separate files or one file that was renamed (hint: look at the CNID in "`node_id`" column)?
    a. The CNIDs for these files are different; therefore, they are two separate DMG files that are similarly named.
        i. `kl.dmg` = 1529172
        ii. `kl2.dmg` = 1529237

- Search for the file IMG_0030.JPG using an SQLite query in the "Execute SQL" tab.

```
select * from fsevents where fullpath like '%IMG_0030.JPG%'
```

3. How do you think this file ended up on `dlightman`'s system (staring in February 2018)?
    a. Looking at entry #2349120, it shows that it was "shared" via the `sharingd` process.

   b. This file was shared via AirDrop from Jen Mack's iPhone; take a look at the extended attributes for this file, "`xattr -xl /Volumes/galaga_mounted/Users/dlightman/Documents/IMG_0030.jpeg`".
   c. This file was originally downloaded into the user's Downloads directory, then opened with Preview App (a couple of times), edited, and finally moved/saved into the users Documents directory.

4. This picture was later edited by Dave Lightman; what software did he use to edit it?
   a. There are multiple entries that suggest that this file was edited by "`Preview.app`".
   b. "`(A Document Being Saved By Preview)`"
      i. 2434304
      ii. 2435655
      iii. 2436512
      iv. 2437481
      v. 2438373
      vi. 2438398

- Search activity for a file using the iNode/CNID 1417428.

```
select * from fsevents where node_id == 1417428
```

5. What browser downloaded this file?
   a. Safari: Looking at entry # 1093597, it shows that it was "`(A Document Being Saved By Safari)`"; this can be validated by extended attributes.

6. This file was downloaded to the default downloads directory (`~/Downloads`); where did it move later?
   a. `/Users/dlightman/Documents/games/asteroids_1b.pdf`, Entry #2881049

- Search for a file ms-nAphDJ.gif.

```
select * from fsevents where filename == "ms-nAphDJ.gif" order by id
```

7. Where did this file come from?
   a. It was an attachment in Messages; it was sent in a chat. It was later downloaded to the default downloads directory.
   b. See records:
      a. 2344842
      b. 2882654
      c. 2882668

3. Review `dlightman's` Spotlight Directory

- Use the `cd` command to enter the Spotlight directory.

- Use the `ls -la` command to view the contents of this directory. Review the contents of this directory.
- Use the `open` command to open the `VolumeConfiguration.plist` file and review the contents of this file. Note that the first time that Xcode's plist reader runs, it may prompt to add more features. If prompted, please do so.

```
$ cd /Volumes/galaga_mounted/.Spotlight-V100

$ ls -la

$ open VolumeConfiguration.plist
```

1. Are there any files or directories excluded from Spotlight indexing?
   a. No, the `Exclusions` key is blank.

4. Spotlight: Review the Spotlight Metadata

- Use the `cd` command to explore the `dlightman`'s `~/Downloads` directory.
- Use the `mdls` command to view the files in this directory. Answer the following questions.
  i. Some students may need to use "`sudo`" with the `mdls` command.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Downloads/

$ mdls LiberiOS11.0.3.ipa                ←Repeat as necessary for each file
```

1. Where was the file `LiberiOS11.0.3.ipa` downloaded from?
   a. `kMDItemWhereFroms` = http://newosxbook.com/liberios/
   b. The same information is found in the Quarantine Extended Attribute.
2. On what day did the file `Impactor_0.9.44.dmg` get used last?
   a. `kMDItemUsedDates` (more general) = `03/03/2018`
   b. `kMDItemLastUsedDate` (more specific) = `2018-03-03 20:27:35 +0000`
3. Please answer the following on the file `IMG_0007.JPG`:
   a. How did the file get transferred to this system?
      i. `kMDItemUserSharedReceivedTransport` = AirDrop
   b. From whom?
      i. `kMDItemUserSharedReceivedSender` = Jen Mack
      ii. `kMDItemUserSharedReceivedSenderHandle` = 1337jmack@gmail.com
      iii. `kMDItemWhereFroms` = Jen Mack's iPhone
   c. When?
      i. `kMDItemUserSharedReceivedDate` = 2018-02-25 22:31:57 +0000
   d. What is the Make/Model of the phone?
      i. `kMDItemAcquisitionMake` = Apple
      ii. `kMDItemAcquisitionModel` = iPhone 6
   e. What version of iOS was it running?
      i. `kMDItemCreator` = 11.0.1
   f. Does this photo have location coordinates?
      i. No.

- Find photos that have locational data in them.
- Use the "mdfind" command to search “-onlyin” in the mounted volume for Dave Lightman.
- Search for items containing the metadata item for latitude.
- Find the path for the photo IMG_0042.JPG and perform an mdls on it.
    - i. Some students may need to use “sudo” with the mdls command.

```
mdfind -onlyin /Volumes/galaga_mounted/ -name "kMDItemLatitude == *"

mdls “/Volumes/galaga_mounted/Users/dlightman/Library/Containers/
com.apple.cloudphotosd/Data/Library/Application Support/
com.apple.cloudphotosd/services/com.apple.photo.icloud.sharedstreams/
assets/66B292A9-9F24-4889-913C-1A90395F2338/
E17A868C-9AF1-4DED-802A-A9F7655F4065/IMG_0042.JPG”
```

4. What are the coordinates for IMG_0042.JPG?
    - a. Latitude: kMDItemLatitude = 51.51343
    - b. Longitude: kMDItemLongitude = -0.099358
5. In what major landmark was this photo taken?
    - a. St. Paul’s Cathedral
    - b. Plug these coordinates into Google Maps, Apple Maps, etc. or…
    - c. Open the photo in the Preview application and open the Inspector [Tools | Show Inspector]
        - i. Select the GPS tab and zoom in.

5. Review the dlightman’s Trash.

- Use the cd command to change the directory to the dlightman’s .Trash directory.
- Use the ls -la command to view the contents of this directory.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/.Trash

$ ls -la
```

1. What three files are in the trash?
    - a. ApplePi-Baker.zip
    - b. Spectacle+1.2.zip
    - c. logKext-master 2

2. Where did some of these files once exist?
    - a. /Users/dlightman/Downloads directory
    - b. View the .DS_store file in a Hex Editor or use xxd on the command line.
        - i. xxd .DS_Store | less
3. Note that the file Spectacle+1.2.zip did not exist in the .DS_Store file; it’s not a perfect system.

Lab: Key Takeaways

- **Review the contents of files and databases that contain data that use the file system.**

- **Find that different files may contain metadata that may not be easy to find at first glance, but that you might have to go digging for it.**

Lab 2.3: Parsing APFS

Objectives

- Parse out important APFS structures; Container Super Block, Volume Super Block, and a file entry.

Lab Preparation

(Note: Some of this might already be accomplished via earlier Labs, but this is the state that we hope your system is in prior to the start of this Lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this Lab.)

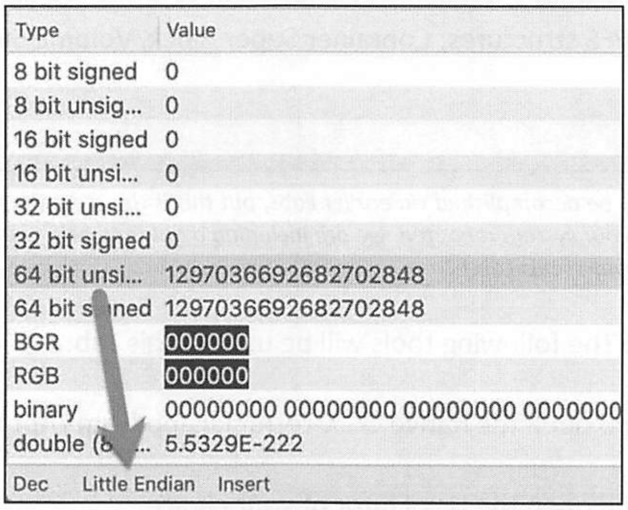
1. **Software Preparation**: The following tools will be used in this Lab:
   - Terminal.app
     - i. Locate and open the native OS X Terminal.app from /Applications/Utilities/.
   - Hex Editor
     - i. Locate and open the Hex Editor of your choice.
     - ii. I like these:
       1. 0xED: http://www.suavetech.com/0xed/0xed.html
          - a. /Applications/0xED.app
       2. Hex Fiend: http://ridiculousfish.com/hexfiend/
          - a. /Applications/Hex Fiend.app

2. **Lab File Preparation**: Locate the `APFS.dmg` file located in the `Lab Files/Lab 2.3 – Parsing APFS` directory on your FOR518 USB drive. This file should have the MD5: `f1234a31feb2ddd4a57a61dc540cacc5`. This can be checked by executing the command: `md5 APFS.dmg`.

3. **FOR518 APFS Reference Sheet**: Locate the FOR518 APFS Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive. This reference is **HIGHLY** recommended for this Lab.

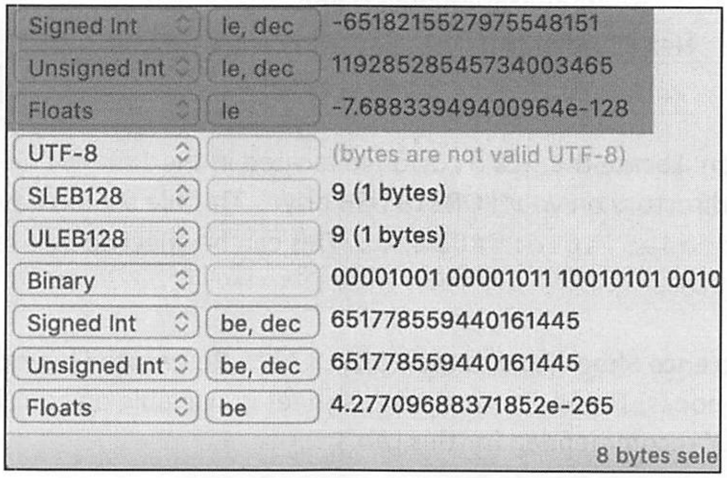
Lab: Questions

1. **Determine how to view little endian values in your hex editor.**
    a. **0xED** – Ensure it says 'Little Endian' in the bottom. Click it if it says 'Big Endian'



Type	Value
8 bit signed	0
8 bit unsig...	0
16 bit signed	0
16 bit unsi...	0
32 bit unsi...	0
32 bit signed	0
64 bit unsi...	1297036692682702848
64 bit signed	1297036692682702848
BGR	000000
RGB	000000
binary	00000000 00000000 00000000 0000000
double (8...	5.5329E-222

    b. **Hex Fiend –** Ensure you have at least the highlighted entries shown.



Signed Int	le, dec	-6518215527975548151
Unsigned Int	le, dec	11928528545734003465
Floats	le	-7.68833949400964e-128
UTF-8		(bytes are not valid UTF-8)
SLEB128		9 (1 bytes)
ULEB128		9 (1 bytes)
Binary		00001001 00001011 10010101 0010
Signed Int	be, dec	651778559440161445
Unsigned Int	be, dec	651778559440161445
Floats	be	4.27709688371852e-265

1. **Extract structures to parse from the APFS DMG image.**
    a. Use `dd` to extract each APFS structure. Each block is 4096 bytes. The offsets were provided to you as, these have the most recent transaction ID (XID) values for each object structure. The input block size is set to 1 (`ibs=1`) so these values can be seen in the command line (The default block size for `dd` is 512).

    b. **Container Super Block - 4096 bytes at offset 53248**

```
$ dd if=APFS.dmg ibs=1 skip=53248 count=4096 >
~/FOR518/container_super_block
```

    c. **Volume Super Block – 4096 bytes at offset 921600**

```
$ dd if=APFS.dmg ibs=1 skip=921600 count=4096 >
~/FOR518/volume_super_block
```

d. **B-Tree Node – 4096 bytes at offset 905216**

```
$ dd if=APFS.dmg ibs=1 skip=905216 count=4096 >
~/FOR518/btree_node
```

2. **Parse the Container Super Block**
   a. Open the `container_super_block` file you just created in the hex editor of your choice. Fill in the blanks.

Object Header (obj_phys_t) [32 bytes, offset 0]

B-tree Offset	Size (in bytes)	Field	Value & Notes
0	8	o_cksum	______________________________
8	8	o_oid	______________________________
16	8	o_xid	______________________________
24	2	o_type.type	Type ______________________________
	2	o_type.flags	Flags 0x0080 = Non-persistent
28	4	o_subtype	0x00000000 = None

Container Super Block Object (nx_superblock) [4064 bytes, Offset 32]

B-tree Offset	Size (in bytes)	Field	Value & Notes
32	4	nx_magic	______________________________
36	4	nx_block_size	______________________________
40	8	nx_block_count	______________________________
48	8	nx_features	0x00000000 00000000
56	8	nx_read_only_ compatible_features	0x00000000 00000000
64	8	nx_incompatible_features	0x02000000 00000000 = NX_INCOMPAT_VERSION2
72	16	nx_uuid	0x65EC907FCF8C4869AD342F2E02C59E02 = 65EC907F-CF8C-4869-AD34-2F2E02C59E02 (verify with `diskutil info /dev/disk#` [Container])
88	8	nx_next_oid	0x0804000000000000 = 1032
96	8	nx_next_xid	0x0D00000000000000 = 13

104	4	nx_xp_desc_blocks	0x08000000 = 8
108	4	nx_xp_data_blocks	0x34000000 = 52
112	8	nx_xp_desc_base	0x01000000 00000000 = 1
120	8	nx_xp_data_base	0x9D000000 00000000 = 9
128	4	nx_xp_desc_next	0x00000000 = 0
132	4	nx_xp_data_next	0x2E000000 = 46
136	4	nx_xp_desc_index	0x06000000 = 6
140	4	nx_xp_desc_len	0x02000000 = 2
144	4	nx_xp_data_index	0x2A000000 = 42
148	4	nx_xp_data_len	0x04000000 = 4
152	8	nx_spaceman_oid	0x00040000 00000000 = 1024
160	8	nx_omap_oid	0xDD00000000000000 = 221
168	8	nx_reaper_oid	0x01040000 00000000 = 1025
176	4	nx_test_type	0x00000000
180	4	nx_max_file_systems	______________________________
184	8	nx_fs_oid[0]	0x02040000 00000000 = 1026 (oid for LetsParseAPFS Volume)

3. **Parse the Volume Super Block**
    b. Open the `volume_super_block` file you just created in the hex editor of your choice. Fill in the blanks.

Object Header (obj_phys_t) [32 bytes, offset 0]

Offset	Size (in bytes)	Field	Value & Notes
0	8	o_cksum	0xE0345B935464182B
8	8	o_oid	______________________________
16	8	o_xid	______________________________
24	2	o_type.type	Type ______________________________
	2	o_type.flags	Flags 0x0000 = None
28	4	o_subtype	0x00000000 = None

Volume Super Block Object (apfs_superblock) [4064 bytes, Offset 32]

Offset	Size (in bytes)	Field	Value/Notes
32	4	apfs_magic	______________________________
36	4	apfs_fs_index	0x00000000 = 0 (First volume…only one volume)
40	8	apfs_features	0x02000000 00000000 = APFS_FEATURE_HARDLINK_MAP_RECORDS
48	8	apfs_readonly_compatible_features	0x00000000 00000000

56	8	apfs_incompatible_features	0x01000000 00000000 = APFS_INCOMPAT_CASE_INSENSITIVE
64	8	apfs_unmount_time	______________________ ______________________
72	8	apfs_fs_reserve_block_count	0x00000000 00000000 = 0
80	8	apfs_fs_quota_block_count	0x00000000 00000000 = 0
88	8	apfs_fs_alloc_count	0x3800000000000000 = 56
96	2	wrapped_crypto_state_t. wrapped_crypto_state.major_version	0x0500
98	2	wrapped_crypto_state_t. wrapped_crypto_state.minor_version	0x0000
100	4	wrapped_crypto_state_t. wrapped_crypto_state.cpflags	0x00000000
104	4	wrapped_crypto_state_t. wrapped_crypto_state.persistent_class	0x06000000
108	4	wrapped_crypto_state_t. wrapped_crypto_state.key_os_version	0x39004313 19 C 57 – 19C57 – Catalina 10.15.2
112	2	wrapped_crypto_state_t. wrapped_crypto_state.key_revision	0x0100
114	2	wrapped_crypto_state_t. wrapped_crypto_state.key_len	0x0000
N/A	0	wrapped_crypto_state_t. wrapped_crypto_state.persistent_key	Null – No Key, see key_len above
116	4	apfs_root_tree_oid_type	0x02000000 = B-Tree
120	4	apfs_extentref_tree_oid_type	0x02000040 = B-Tree, Physical
124	4	apfs_snap_meta_tree_oid_type	0x02000040 = B-Tree, Physical
128	8	apfs_omap_oid	0xD900000000000000 = 217
136	8	apfs_root_tree_oid	0x0404000000000000 = 1028
144	8	apfs_extentref_tree_oid	0xD400000000000000 = 212
152	8	apfs_snap_meta_tree_oid	0x5800000000000000 = 88
160	8	apfs_revert_to_xid	0x0000000000000000 = 0
168	8	apfs_revert_to_sblock_oid	0x0000000000000000 = 0
176	8	apfs_next_obj_id	0x1A00000000000000 = 26
184	8	apfs_num_files	______________________
192	8	apfs_num_directories	______________________
200	8	apfs_num_symlinks	0x00000000 00000000 = 0
208	8	apfs_num_other_fsobjects	0x00000000 00000000 = 0
216	8	apfs_num_snapshots	0x00000000 00000000 = 0
224	8	apfs_total_blocks_alloced	0x4100000000000000 = 65
232	8	apfs_total_blocks_freed	0x1000000000000000 = 16
240	16	apfs_vol_uuid	0xED919A5F81114AA5B88A5D34316C7EE9 = ED919A5F-8111-4AA5-B88A-5D34316C7EE9

			(verify with `diskutil info /dev/disk#s#` [Volume])
256	8	apfs_last_mod_time	
264	8	apfs_fs_flags	0x0100000000000000
272	32	apfs_modified_by_t.formatted_by.id[]	
304	8	apfs_modified_by_t.formatted_by.timestamp	
312	8	apfs_modified_by_t.formatted_by.last_xid	0x0200000000000000
320	32	apfs_modified_by_t.modified_by.id[]	
352	8	apfs_modified_by_t.modified_by.timestamp	
360	8	apfs_modified_by_t.modified_by.last_xid	0x0900000000000000
368	336	apfs_modified_by_t.modified_by[1-7]	apfs_modified_by_t[8] 48x8 = 384
704	256	apfs_volname	
960	4	apfs_next_doc_id	0x03000000 = 3
964	2	apfs_role	0x0000 = None
966	2	apfs_reserved	0x0000
976	8	apfs_root_to_xid	0x00000000 00000000 = 0
984	8	apfs_er_state_oid	0x00000000 00000000 = 0

4. **Parse a B-Tree Node**
    c. Open the `btree_node` file you just created in the hex editor of your choice. Fill in the blanks.

Object Header (obj_phys_t) [32 bytes, offset 0]

Btree Offset	Size (in bytes)	Field	Value & Notes
0	8	o_cksum	0x77B4DE6C812048DE
8	8	o_oid	
16	8	o_xid	
24	2	o_type.type	Type
	2	o_type.flags	Flags 0x0000 = None
28	4	o_subtype	

B-Tree Node (btree_node_phys_t) [24 bytes, offset 32]

Btree Offset	Size (in bytes)	Field	Value & Notes
32	2	btn_flags	0x0200 – Leaf Node
34	2	btn_level	0x0000
36	4	btn_nkeys	______________________________ (keys stored in this node)
40	2	btn_table_space.off	______________________________ (TOC)
42	2	btn_table_space.len	______________________________
44	2	btn_freespace.off	0xA303 = 931 (Free Space)
46	2	btn_freespace.len	0x2300 = 35
48	2	btn_key_free_list.off	0x9303 = 915 (Free Key Space)
50	2	btn_key_free_list.len	0x1000 = 16
52	2	btn_val_free_list.off	0xCA09 = 2506 (Free Value Space)
54	2	btn_val_free_list.len	0x3000 = 48

Table of Contents – Fill in the missing pieces of the TOC fields.
[376 bytes (47 entries * 8 bytes), offset 56]

B-tree Offset	TOC Entry	key_offset (2 bytes)	key_length (2 bytes)	value_offset (2 bytes)	value_length (2 bytes)	Object ID (Inode # in Hex)
56	1	0x0000 = 0	0x1800 = 24	0x1200 = 18	0x1200 = 18	01 private-dir
64	2	0x1800 = 24	0x1100 = 17	0x2400 = 36	0x1200 = 18	01 root
72	3	0x2900 = 41	0x0800 = 8	0x9000 = 144	0x6C00 = 108	02
80	4	49	22	162	18	02
88	5	71	22	180	18	02
96	6	93	23	198	18	02
104	7	116	22	216	18	02
112	8	138	8	332	116	03
120	9	146	8	2666	160	10
128	10	154	31	368	36	10
136	11	185	8	372	4	10
144	12	193	16	396	24	10
152	13	209	8	512	116	11
160	14	217	28	542	30	11
168	15	245	36	560	18	11
176	**16**	**281**	**8**	**728**	**168**	**12**
184	**17**	**289**	**36**	**748**	**20**	**12**
192	**18**	**325**	**47**	**936**	**188**	**12**
200	**19**	**372**	**31**	**997**	**61**	**12**
208	**20**	______	______	______	______	**12**
216	**21**	______	______	______	______	**12**
224	**22**	______	______	______	______	**12**
232	23	455	8	1171	116	13

240	24	463	28	1201	30	13
248	25	491	35	1219	18	13
256	26	526	29	1237	18	13
264	27	555	8	1405	168	14
272	28	563	36	1425	20	14
280	29	599	31	1486	61	14
288	30	630	28	1516	30	14
296	31	658	8	1520	4	14
304	32	666	16	1544	24	14
312	33	682	8	1660	116	15
320	34	690	27	1678	18	15
328	35	717	29	1696	18	15
336	36	746	29	1714	18	15
344	37	775	8	1874	160	16
352	38	783	8	1878	4	16
360	39	791	16	1902	24	16
368	40	807	8	2070	168	17
376	41	815	8	2074	4	17
384	42	823	16	2098	24	17
392	43	839	8	2266	168	18
400	44	847	8	2270	4	18
408	45	855	16	2294	24	18
416	46	871	8	2462	168	19
424	47	879	36	2482	20	19
432	8	Extra 8 bytes, table space value is 384 while TOC contents is 376 bytes				

File System Keys – Fill in the missing pieces of for TOC Entries 16 – 22. These are the File System Keys for the `smudge_transformer.jpeg` file. File System Keys in B-tree File: Bytes 440 – 915 (475 total bytes).

File System Keys - Inode Keys for `smudge_transformer.jpeg` file

B-tree Offset	Entry	Offset	Size (in bytes)	Object ID (Inode #)	Entry Kind [Highest byte in first 8 bytes]	Entry Type	Value & Notes
721	16	281	**8**	0x120000000000 = 12	0x30	Inode	N/A
729	17	289	**36**	0x120000000000 = 12	0x40	Xattr	`com.apple.lastuseddate#PS` [2 byte size before, 1 byte padding after]
765	18	325	**47**	0x120000000000 = 12	0x40	Xattr	____________________ ____________________ [2 byte size before, 1 byte padding after]

812	19	372	**31**	0x12000000000000 = 12	0x40	Xattr	______________________ ______________________ [2 byte size before, 1 byte padding after]
843	20	______	______	0x12000000000000 = 12	______	______	com.dropbox.attrs [2 byte size before, 1 byte padding after]
871	21	______	______	0x12000000000000 = 12	______	______	N/A
879	22	______	______	0x12000000000000 = 12	______	______	0x0000000000000000

File System Values – Fill in the missing pieces of for TOC Entries 16 – 22. These are the File System Values for the smudge_transformer.jpeg file. File System Values in B-tree File: Bytes 1614 – 4096 (2482 total bytes).

File System Values - Inode Values for smudge_transformer.jpeg File

B-tree Offset	Entry	Offset	Size (in bytes)	Entry Type	Value & Notes
3368	16	728	**168**	Inode	File Metadata for smudge_transformer.jpeg **[See below]** 0x110000000000000012000000000000000C8B8E50243ED1500C8B8E50243ED1570D76C933743ED15002E5F812D43ED150080000000000000001000000000000000200000000000000F501000014000000A4810000000000000000000020040000402180008202800736D756467655F7472616E73666F726D65722E6A70656700A08900000000000000090000000000000000000000000000A0890000000000000000000000000000
3348	17	748	**20**	Xattr	0x0200100028C72C5E00000000AED5671300000000 = com.dropbox.attrs
3160	18	936	**188**	Xattr	**Question: Where was this photo downloaded from?** ______________________________
3099	19	997	**61**	Xattr	**Question: How was this photo downloaded?** ______________________________
3069	20	1027	**30**	Xattr	0x02001A000A120A1059C45688BCFCFFB4000000000007C9FD1099BD92B608 = com.dropbox.attrs
3065	21	1031	**4**	Data Stream	0x01000000 = Number of References
3041	22	1055	**24**	File Extent	File Size: ______________________ Physical Block Location:

					Physical Block Number from start of container (add 5 (20,480) blocks for start of disk) (# * 4096) + 20,480 = start of file location in bytes	_______________________
					crypto_id	0x0000000000000000 – No Key

Inode Entry/File Metadata for smudge_transformer.jpeg

B-tree Offset	Inode Entry Offset	Size (in bytes)	Field	Value & Notes
3368	0	8	parent_id	_______________________
3376	8	8	private_id	_______________________
3384	16	8	create_time	0x00C8B8E50243ED15 = 1579992724000000000 = 2020-01-25 22:52:04 UTC
3392	24	8	mod_time	0x00C8B8E50243ED15 = 1579992724000000000 = 2020-01-25 22:52:04 UTC
3400	32	8	change_time	0x70D76C933743ED15 = 1579992950252558192 = 2020-01-25 22:55:50.252558 UTC
3408	40	8	access_time	0x002E5F812D43ED15 = 1579992907000000000 = 2020-01-25 22:55:07 UTC
3416	48	8	internal_flags	0x0080000000000000
3424	56	4	nchildren or nlink	0x01000000 = 1
3428	60	4	default_protection_class	0x00000000
3432	64	4	write_generation_counter	0x02000000
3426	68	4	bsd_flags	0x00000000
3440	72	4	owner	_______________________
3444	76	4	group	_______________________

3448	80	2	mode	1000 = 8 (Regular File)
				000 = SetUID, SetGID, Sticky bits
				___ = ___ User Permissions
				___ = ___ Group Permissions
				___ = ___ Other Permissions

3450	82	2	pad1	0x0000
3452	84	8	pad2	0x0000000000000000
3460	92	2	xf_num_exts	Number of Extended Fields = 0x0200 = 2
3462	94	2	xf_used_data	Extended Fields Data Used = 0x4000 = 64 bytes

<table>
<tr><td><b>3464</b></td><td>96</td><td>x_field_t 8</td><td colspan="2"><table><tr><td>x_type [1 byte]</td><td>x_flags [1 byte]</td><td>x_size [2 byte]</td></tr><tr><td>0x04 = String Name</td><td>0x02 = Do not copy</td><td>0x1800 = 24</td></tr><tr><td>0x08 = Data Stream</td><td>0x20 = System Field</td><td>0x2800 = 40</td></tr></table></td></tr>
<tr><td><b>3472</b></td><td>104</td><td>{24}</td><td>File Name</td><td>________________________
(w/1 padding bytes 0x00), 24 total bytes</td></tr>
<tr><td><b>3496</b></td><td>120</td><td>{40}</td><td>Data Stream
(Size: First 8 bytes)</td><td>File Size: ________________________
Allocated: 0090000000000000 = 36864</td></tr>
</table>

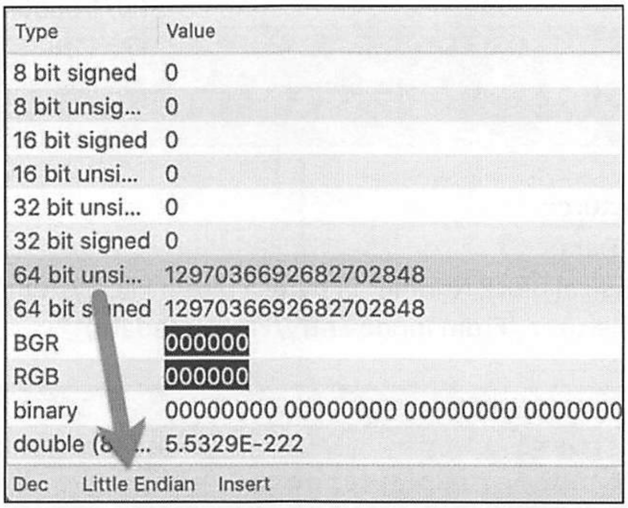
Use dd to extract the picture:

- From File Extent Data:
  - skip=<Physical Block Number in bytes> (From File System Values - Inode Values
  - count=<file size> (From Inode Entry/File Metadata)

```
$ dd if=APFS.dmg ibs=1 skip=___________ count=___________ >
~/FOR518/smudge_transformer_extracted.jpeg
```

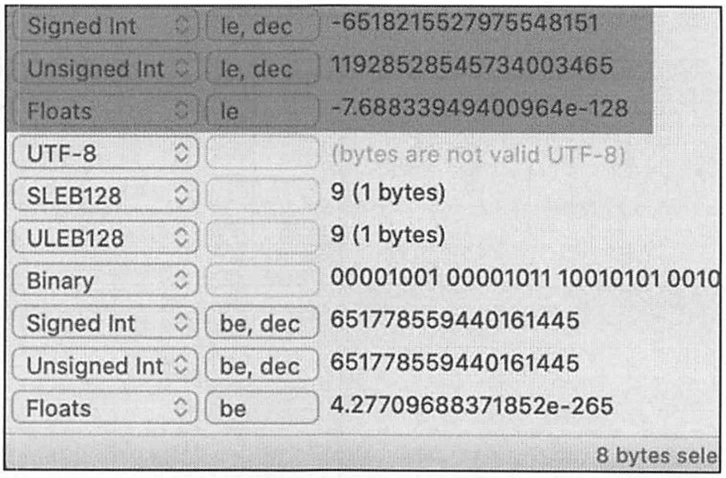
Lab: Step-by-Step

1. **Determine how to view little endian values in your hex editor.**
   a. **0xED** – Ensure it says 'Little Endian' in the bottom. Click it if it says 'Big Endian'



Type	Value
8 bit signed	0
8 bit unsig...	0
16 bit signed	0
16 bit unsi...	0
32 bit unsi...	0
32 bit signed	0
64 bit unsi...	1297036692682702848
64 bit signed	1297036692682702848
BGR	000000
RGB	000000
binary	00000000 00000000 00000000 0000000
double (8...	5.5329E-222

   b. **Hex Fiend –** Ensure you have at least the highlighted entries shown.



Signed Int	le, dec	-6518215527975548151
Unsigned Int	le, dec	11928528545734003465
Floats	le	-7.68833949400964e-128
UTF-8		(bytes are not valid UTF-8)
SLEB128		9 (1 bytes)
ULEB128		9 (1 bytes)
Binary		00001001 00001011 10010101 0010
Signed Int	be, dec	651778559440161445
Unsigned Int	be, dec	651778559440161445
Floats	be	4.27709688371852e-265

2. **Extract structures to parse from the APFS DMG image.**
   a. Use dd to extract each APFS structure. Each block is 4096 bytes. The offsets were provided to you as, these have the most recent transaction ID (XID) values for each object structure. The input block size is set to 1 (ibs=1) so these values can be seen in the command line (The default block size for dd is 512).

   b. **Container Super Block - 4096 bytes at offset 53248**

```
$ dd if=APFS.dmg ibs=1 skip=53248 count=4096 >
~/FOR518/container_super_block
```

   c. **Volume Super Block – 4096 bytes at offset 921600**

```
$ dd if=APFS.dmg ibs=1 skip=921600 count=4096 >
~/FOR518/volume_super_block
```

d. B-Tree Node – 4096 bytes at offset 905216

```
$ dd if=APFS.dmg ibs=1 skip=905216 count=4096 >
~/FOR518/btree_node
```

3. Parse the Container Super Block

d. Open the `container_super_block` file you just created in the hex editor of your choice. Fill in the blanks.

Object Header (obj_phys_t) [32 bytes, offset 0]

B-tree Offset	Size (in bytes)	Field	Value & Notes
0	8	o_cksum	**0x4E90821780CF1BFA**
8	8	o_oid	**0x0100000000000000 = 1**
16	8	o_xid	**0x0C00000000000000 = 12**
24	2	o_type.type	**Type** **0x0100 = Container Super Block**
	2	o_type.flags	Flags 0x0080 = Non-persistent
28	4	o_subtype	0x00000000 = None

Container Super Block Object (nx_superblock) [4064 bytes, Offset 32]

B-tree Offset	Size (in bytes)	Field	Value & Notes
32	4	nx_magic "NXSB"	**0x4E585342 = "NXSB"**
36	4	nx_block_size	**0x00100000 = 4096**
40	8	nx_block_count	**0x330A000000000000 = 2611 (verify with `diskutil info /dev/disk#` [Container])** **2611*4096 = 10694656 Bytes**
48	8	nx_features	0x00000000 00000000
56	8	nx_read_only_ compatible_features	0x00000000 00000000
64	8	nx_incompatible_features	0x02000000 00000000 = NX_INCOMPAT_VERSION2
72	16	nx_uuid	0x65EC907FCF8C4869AD342F2E02C59E02 = 65EC907F-CF8C-4869-AD34-2F2E02C59E02 (verify with `diskutil info /dev/disk#` [Container])

88	8	nx_next_oid	0x0804000000000000 = 1032
96	8	nx_next_xid	0x0D00000000000000 = 13
104	4	nx_xp_desc_blocks	0x08000000 = 8
108	4	nx_xp_data_blocks	0x34000000 = 52
112	8	nx_xp_desc_base	0x01000000 00000000 = 1
120	8	nx_xp_data_base	0x9D000000 00000000 = 9
128	4	nx_xp_desc_next	0x00000000 = 0
132	4	nx_xp_data_next	0x2E000000 = 46
136	4	nx_xp_desc_index	0x06000000 = 6
140	4	nx_xp_desc_len	0x02000000 = 2
144	4	nx_xp_data_index	0x2A000000 = 42
148	4	nx_xp_data_len	0x04000000 = 4
152	8	nx_spaceman_oid	0x00040000 00000000 = 1024
160	8	nx_omap_oid	0xDD00000000000000 = 221
168	8	nx_reaper_oid	0x01040000 00000000 = 1025
176	4	nx_test_type	0x00000000
180	4	nx_max_file_systems	0x01000000 = 1
184	8	nx_fs_oid[0]	0x02040000 00000000 = 1026 (oid for LetsParseAPFS Volume)

4. **Parse the Volume Super Block**
    e. Open the `volume_super_block` file you just created in the hex editor of your choice. Fill in the blanks.

Object Header (obj_phys_t) [32 bytes, offset 0]

Offset	Size (in bytes)	Field	Value & Notes
0	8	o_cksum	0xE0345B935464182B
8	8	o_oid	**0x0204000000000000 = 1026**
16	8	o_xid	**0x0C00000000000000 = 12**
24	2	o_type.type	**Type** **0x0D00 = Volume Super Block**
	2	o_type.flags	Flags 0x0000 = None
28	4	o_subtype	0x00000000 = None

Volume Super Block Object (apfs_superblock) [4064 bytes, Offset 32]

Offset	Size (in bytes)	Field	Value/Notes
32	4	apfs_magic "APSB"	**0x41505342 = "APSB"**
36	4	apfs_fs_index	0x00000000 = 0 (First volume…only one volume)
40	8	apfs_features	0x02000000 00000000 = APFS_FEATURE_HARDLINK_MAP_RECORDS

48	8	apfs_readonly_compatible_features	0x00000000 00000000
56	8	apfs_incompatible_features	0x01000000 00000000 = APFS_INCOMPAT_CASE_INSENSITIVE
64	8	apfs_unmount_time	**0xCF29D2975443ED15 = 1579993074880358863 = 2020-01-25 22:57:54.880359 UTC**
72	8	apfs_fs_reserve_block_count	0x00000000 00000000 = 0
80	8	apfs_fs_quota_block_count	0x00000000 00000000 = 0
88	8	apfs_fs_alloc_count	0x3800000000000000 = 56
96	2	wrapped_crypto_state_t. wrapped_crypto_state.major_version	0x0500
98	2	wrapped_crypto_state_t. wrapped_crypto_state.minor_version	0x0000
100	4	wrapped_crypto_state_t. wrapped_crypto_state.cpflags	0x00000000
104	4	wrapped_crypto_state_t. wrapped_crypto_state.persistent_class	0x06000000
108	4	wrapped_crypto_state_t. wrapped_crypto_state.key_os_version	0x39004313 19 C 57 – 19C57 – Catalina 10.15.2
112	2	wrapped_crypto_state_t. wrapped_crypto_state.key_revision	0x0100
114	2	wrapped_crypto_state_t. wrapped_crypto_state.key_len	0x0000
N/A	0	wrapped_crypto_state_t. wrapped_crypto_state.persistent_key	Null – No Key, see key_len above
116	4	apfs_root_tree_oid_type	0x02000000 = B-Tree
120	4	apfs_extentref_tree_oid_type	0x02000040 = B-Tree, Physical
124	4	apfs_snap_meta_tree_oid_type	0x02000040 = B-Tree, Physical
128	8	apfs_omap_oid	0xD900000000000000 = 217
136	8	apfs_root_tree_oid	0x0404000000000000 = 1028
144	8	apfs_extentref_tree_oid	0xD400000000000000 = 212
152	8	apfs_snap_meta_tree_oid	0x5800000000000000 = 88
160	8	apfs_revert_to_xid	0x0000000000000000 = 0
168	8	apfs_revert_to_sblock_oid	0x0000000000000000 = 0
176	8	apfs_next_obj_id	0x1A00000000000000 = 26
184	8	apfs_num_files	**0x0700000000000000 = 7**
192	8	apfs_num_directories	**0x0300000000000000 = 3**
200	8	apfs_num_symlinks	0x00000000 00000000 = 0
208	8	apfs_num_other_fsobjects	0x00000000 00000000 = 0
216	8	apfs_num_snapshots	0x00000000 00000000 = 0
224	8	apfs_total_blocks_alloced	0x4100000000000000 = 65
232	8	apfs_total_blocks_freed	0x1000000000000000 = 16
240	16	apfs_vol_uuid	0xED919A5F81114AA5B88A5D34316C7EE9 = ED919A5F-8111-4AA5-B88A-5D34316C7EE9

			(verify with `diskutil info /dev/disk#s#` [Volume])
256	8	apfs_last_mod_time	**0xA933888C6943ED15 = 1579993164885275561 = 2020-01-25 22:59:24.885276 UTC**
264	8	apfs_fs_flags	0x0100000000000000
272	32	apfs_modified_by_t.formatted_by.id[]	**0x6E657766735F617066732028313431322E36312E3129000000000000000000000 = "newfs_apfs (1412.61.1)"**
304	8	apfs_modified_by_t.formatted_by. timestamp	**0xD8B96C2C3743ED15 = 1579992948524497368 2020-01-25 22:55:48.524498 UTC**
312	8	apfs_modified_by_t.formatted_by. last_xid	0x0200000000000000
320	32	apfs_modified_by_t.modified_by.id[]	**0x617066735F6B6578742028313431322E36312E31290000000000000000000000000 = apfs_kext (1412.61.1)**
352	8	apfs_modified_by_t.modified_by. timestamp	**0x5919D2975443ED15 = 1579993074880354649 = 2020-01-25 22:57:54.880355 UTC**
360	8	apfs_modified_by_t.modified_by. last_xid	0x0900000000000000
368	336	apfs_modified_by_t.modified_by[1-7]	apfs_modified_by_t[8] 48x8 = 384
704	256	apfs_volname	**0x4C657473506172736541504653 = LetsParseAPFS**
960	4	apfs_next_doc_id	0x03000000 = 3
964	2	apfs_role	0x0000 = None
966	2	apfs_reserved	0x0000
976	8	apfs_root_to_xid	0x00000000 00000000 = 0
984	8	apfs_er_state_oid	0x00000000 00000000 = 0

5. **Parse a B-Tree Node**
   - **f.** Open the `btree_node` file you just created in the hex editor of your choice. Fill in the blanks.

Object Header (obj_phys_t) [32 bytes, offset 0]

Btree Offset	Size (in bytes)	Field	Value & Notes
0	8	o_cksum	0x77B4DE6C812048DE
8	8	o_oid	**0x0704000000000000 = 1031**
16	8	o_xid	**0x0C00000000000000 = 12**
24	2	o_type.type	**Type 0x0300 = B-Tree Node**
	2	o_type.flags	Flags 0x0000 = None
28	4	o_subtype	**0x0E000000 = File System Tree**

B-Tree Node (btree_node_phys_t) [24 bytes, offset 32]

Btree Offset	Size (in bytes)	Field	Value & Notes
32	2	btn_flags	0x0200 – Leaf Node
34	2	btn_level	0x0000
36	4	btn_nkeys	**0x2F000000 = 47 (keys stored in this node)**
40	2	btn_table_space.off	**0x0000 = 0 (TOC)**
42	2	btn_table_space.len	**0x8001 = 384**
44	2	btn_freespace.off	0xA303 = 931 (Free Space)
46	2	btn_freespace.len	0x2300 = 35
48	2	btn_key_free_list.off	0x9303 = 915 (Free Key Space)
50	2	btn_key_free_list.len	0x1000 = 16
52	2	btn_val_free_list.off	0xCA09 = 2506 (Free Value Space)
54	2	btn_val_free_list.len	0x3000 = 48

Table of Contents – Fill in the missing pieces of the TOC fields.
[376 bytes (47 entries * 8 bytes), offset 56]

B-tree Offset	TOC Entry	key_offset (2 bytes)	key_length (2 bytes)	value_offset (2 bytes)	value_length (2 bytes)	Object ID (Inode # in Hex)
56	1	0x0000 = 0	0x1800 = 24	0x1200 = 18	0x1200 = 18	01 private-dir
64	2	0x1800 = 24	0x1100 = 17	0x2400 = 36	0x1200 = 18	01 root
72	3	0x2900 = 41	0x0800 = 8	0x9000 = 144	0x6C00 = 108	02
80	4	49	22	162	18	02
88	5	71	22	180	18	02
96	6	93	23	198	18	02
104	7	116	22	216	18	02
112	8	138	8	332	116	03
120	9	146	8	2666	160	10
128	10	154	31	368	36	10
136	11	185	8	372	4	10
144	12	193	16	396	24	10
152	13	209	8	512	116	11
160	14	217	28	542	30	11
168	15	245	36	560	18	11
176	16	281	8	728	168	**12**
184	17	289	36	748	20	**12**
192	18	325	47	936	188	**12**
200	19	372	31	997	61	**12**
208	20	**403**	**28**	**1027**	**30**	**12**
216	21	**431**	**8**	**1031**	**4**	**12**
224	22	**439**	**16**	**1055**	**24**	**12**

232	23	455	8	1171	116	13
240	24	463	28	1201	30	13
248	25	491	35	1219	18	13
256	26	526	29	1237	18	13
264	27	555	8	1405	168	14
272	28	563	36	1425	20	14
280	29	599	31	1486	61	14
288	30	630	28	1516	30	14
296	31	658	8	1520	4	14
304	32	666	16	1544	24	14
312	33	682	8	1660	116	15
320	34	690	27	1678	18	15
328	35	717	29	1696	18	15
336	36	746	29	1714	18	15
344	37	775	8	1874	160	16
352	38	783	8	1878	4	16
360	39	791	16	1902	24	16
368	40	807	8	2070	168	17
376	41	815	8	2074	4	17
384	42	823	16	2098	24	17
392	43	839	8	2266	168	18
400	44	847	8	2270	4	18
408	45	855	16	2294	24	18
416	46	871	8	2462	168	19
424	47	879	36	2482	20	19
432	8	Extra 8 bytes, table space value is 384 while TOC contents is 376 bytes				

File System Keys – Fill in the missing pieces of for TOC Entries 16 – 22. These are the File System Keys for the `smudge_transformer.jpeg` file. File System Keys in B-tree File: Bytes 440 – 915 (475 total bytes).

File System Keys - Inode Keys for `smudge_transformer.jpeg` file

B-tree Offset	Entry	Offset	Size (in bytes)	Object ID (Inode #)	Entry Kind [Highest byte in first 8 bytes]	Entry Type	Value & Notes
721	16	281	8	0x12000000000000 = 12	0x30	Inode	N/A
729	17	289	36	0x12000000000000 = 12	0x40	Xattr	`com.apple.lastuseddate#PS` [2 byte size before, 1 byte padding after]
765	18	325	47	0x12000000000000 = 12	0x40	Xattr	**`com.apple.metadata:kMDItemWhereFroms` [2 byte size before, 1 byte padding after]**

812	19	372	31	0x12000 0000000 00 = 12	0x40	Xattr	**com.apple.quarantine [2 byte size before, 1 byte padding after]**
843	20	**403**	**28**	0x12000 0000000 00 = 12	**0x40**	**Xattr**	com.dropbox.attrs [2 byte size before, 1 byte padding after]
871	21	**431**	**8**	0x12000 0000000 00 = 12	**0x60**	**Data Stream**	N/A
879	22	**439**	**16**	0x12000 0000000 00 = 12	**0x80**	**File Extent**	0x0000000000000000

File System Values – Fill in the missing pieces of for TOC Entries 16 – 22. These are the File System Values for the smudge_transformer.jpeg file. File System Values in B-tree File: Bytes 1614 – 4096 (2482 total bytes).

File System Values - Inode Values for smudge_transformer.jpeg File

B-tree Offset	Entry	Offset	Size (in bytes)	Entry Type	Value & Notes
3368	16	728	**168**	Inode	File Metadata for smudge_transformer.jpeg **[See below]** 0x11000000000000001200000000000000C8B8E50243ED1500C8B8E50243ED1570D76C933743ED15002E5F812D43ED150080000000000000001000000000000000200000000000000F501000014000000A48100000000000000000000200400004021800082028007 36D756467655F7472616E73666F726D65722E6A70656700A089000000000000000900000000000000000000000000000000A089000000000000000000000000000000
3348	17	748	**20**	Xattr	0x0200100028C72C5E00000000AED5671300000000 = com.dropbox.attrs
3160	18	936	**188**	Xattr	**Answer: Twitter** **0x0200B80062706C6973743030A201025F104368747470733A2F2F7062732E7477696D672E636F6D2F6D656469612F454F716C7269635655415565383744A3F666F726D61743D6A7067266E616D653D3930307839303 05F10416874747470733A2F2F747769747465722E636F6D2F536F6E6F66647696767616E2F7374617475732F3132313839363838323032373630353139363 82F70686F746F2F31080B510000000000000001010000000000000003000000000000000000000000000000095 = Binary Plist containing Spotlight WhereFroms Data**

<table>
<tr><th></th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td><b>“bplist00¢_Chttps://pbs.twimg.com/media/EOqlricVUAUe87J?format=jpg&name=900x900_Ahttps://twitter.com/SonofGigan/status/1218968820276051968/photo/1”</b></td></tr>
<tr><td><b>3099</b></td><td>19</td><td>997</td><td><b>61</b></td><td>Xattr</td><td><b>Answer: Chrome</b>
<b>0x0200390030303833 3B356532636336613 3B4368726F6D653B32313139353145332D313741432D344333362D383545302D3839353533333834434539 = File Quarantine Attribute Data</b>
<b>0083;5e2cc6a3;Chrome;211951E3-17AC-4C36-85E0-89553383DCE9</b></td></tr>
<tr><td><b>3069</b></td><td>20</td><td>1027</td><td><b>30</b></td><td>Xattr</td><td>0x02001A000A120A1059C45688BCFCFFB4000000000007C9FD1099BD92B608 = com.dropbox.attrs</td></tr>
<tr><td><b>3065</b></td><td>21</td><td>1031</td><td><b>4</b></td><td>Data Stream</td><td>0x01000000 = Number of References</td></tr>
<tr><td><b>3041</b></td><td>22</td><td>1055</td><td><b>24</b></td><td>File Extent</td><td><b>0x0090000000000000600000000000000000000000000000000</b>
<table>
<tr><td>File Size</td><td><b>0x0090000000000000 = 36864</b></td></tr>
<tr><td>Physical Block Location:
Physical Block Number from start of container (add 5 (20,480) blocks for start of disk)
(# * 4096) + 20,480 = start of file location in bytes</td><td><b>0x6000000000000000 = 96 = Physical Block Number</b>
<b>(96 * 4096) + 20,480 = 413696 bytes</b></td></tr>
<tr><td>crypto_id</td><td><b>0x0000000000000000 – No Key</b></td></tr>
</table></td></tr>
</table>

Inode Entry/File Metadata for smudge_transformer.jpeg

B-tree Offset	Inode Entry Offset	Size (in bytes)	Field	Value & Notes
3368	0	8	parent_id	**0x1100000000000000 = 17**
3376	8	8	private_id	**0x12000000 00000000 = 18**
3384	16	8	create_time	0x00C8B8E50243ED15 = 1579992724000000000 = 2020-01-25 22:52:04 UTC
3392	24	8	mod_time	0x00C8B8E50243ED15 = 1579992724000000000 = 2020-01-25 22:52:04 UTC

3400	32	8	change_time	0x70D76C933743ED15 = 1579992950252558192 = 2020-01-25 22:55:50.252558 UTC
3408	40	8	access_time	0x002E5F812D43ED15 = 1579992907000000000 = 2020-01-25 22:55:07 UTC
3416	48	8	internal_flags	0x0080000000000000
3424	56	4	nchildren or nlink	0x01000000 = 1
3428	60	4	default_protection_class	0x00000000
3432	64	4	write_generation_counter	0x02000000
3426	68	4	bsd_flags	0x00000000
3440	72	4	owner	**0xF5010000 = 501**
3444	76	4	group	**0x14000000 = 20**
3448	80	2	mode	0xA481 = 1010010010000001 Byte Flip = 1000 000 110 100 100 1000 = 8 (Regular File) 000 = SetUID, SetGID, Sticky bits **110 = 6 (rw-) User Permissions** **100 = 4 (r--) Group Permissions** **100 = 4 (r--) Other Permissions** (See tables 15.11-15.13 in File System Forensic Analysis by Brian Carrier)
3450	82	2	pad1	0x0000
3452	84	8	pad2	0x0000000000000000
3460	92	2	xf_num_exts	Number of Extended Fields = 0x0200 = 2
3462	94	2	xf_used_data	Extended Fields Data Used = 0x4000 = 64 bytes
3464	96	x_field_t 8	x_type [1 byte] 0x04 = String Name 0x08 = Data Stream	x_flags [1 byte] 0x02 = Do not copy 0x20 = System Field x_size [2 byte] 0x1800 = 24 0x2800 = 40
3472	104	{24}	File Name	**0x736D756467655F7472616E73666F726D65722E6A70656700 =** **smudge_transformer.jpeg** **(w/1 padding bytes 0x00), 24 total bytes**
3496	120	{40}	Data Stream (Size: First 8 bytes)	**0x0A089000000000000000900000000000000000000000000000A08900000000000000000000000000000** **File Size: 0xA0890000000000 = 35232 bytes** **Allocated: 0090000000000000 = 36864**

Use dd to extract the picture:

- From File Extent Data:
  - skip=<Physical Block Number in bytes> (From File System Values - Inode Values
  - count=<file size> (From Inode Entry/File Metadata)

```
$ dd if=APFS.dmg ibs=1 skip=413696 count=35232 >
~/FOR518/smudge_transformer_extracted.jpeg
```

Lab: Key Takeaways

- **Review and manually parse the contents the file system.**

Lab 3.1: User Data and System Configuration – Part I

Objectives

- Get familiar with the macOS user preferences and system configuration data files.
- Get familiar with property lists using Xcode.
- Get more comfortable with the macOS command line using Terminal.

Lab Preparation

(Note: Some of this might already be accomplished via earlier labs, but this is the state that we hope your system is in prior to the start of this lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this lab.)

1. **Software Preparation**: The following tools will be used in this lab:
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from /Applications/Utilities/.
   - Xcode.app
     i. Locate and open the Xcode.app from /Applications/.
   - BlackLight.app
     i. Locate and open the BlackLight.app from /Applications/BlackLight/BlackLight YYYY Release #/.

2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Open the FOR518.blacklight BlackLight Case File.**

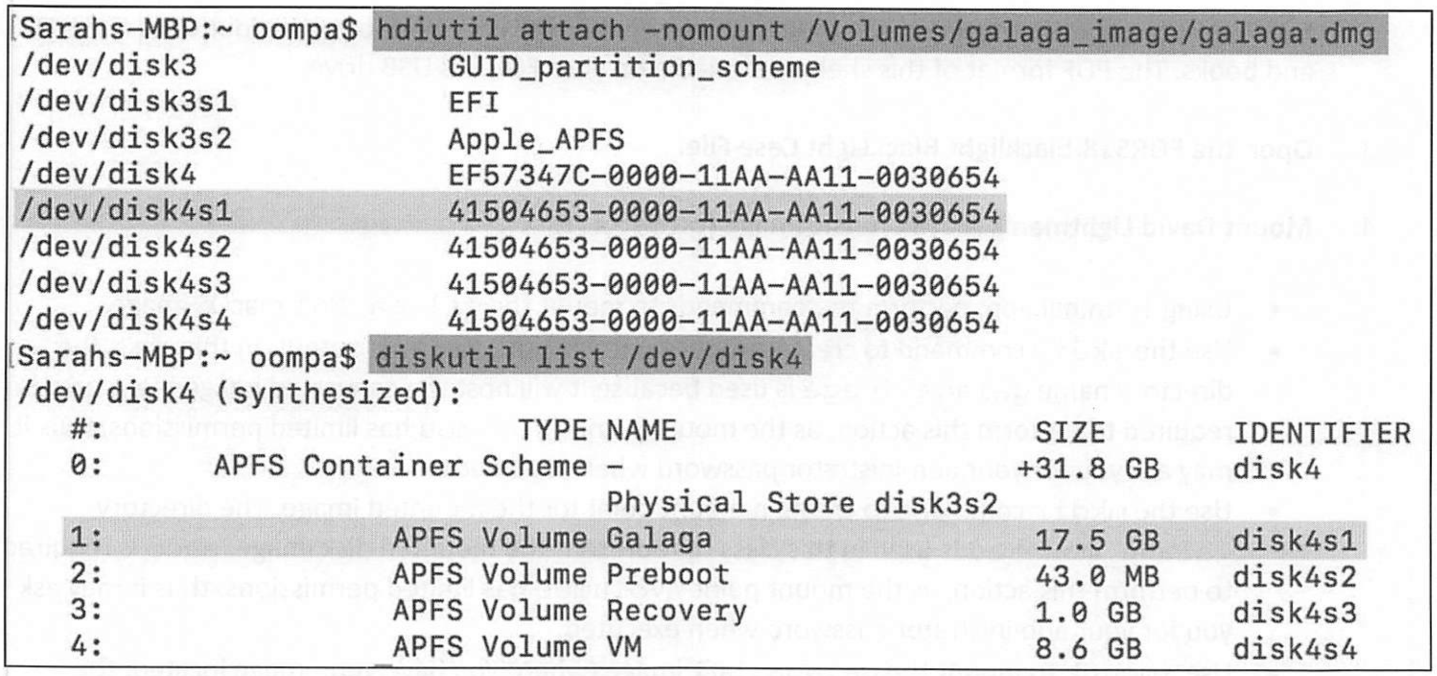
4. **Mount David Lightman's Mac Forensic Image (`galaga.E01`).**

   - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
   - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.

- `--in` – Tells `xmount` what input file type to expect; our images are in a compressed EWF format.
- `--out` – Tells `xmount` what output format you want; we want a DMG file so we can mount it in Finder.
- `Input File` – Where the image file is located on your system.
- `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Use the `hdiutil` command with the "`attach`" verb to make the newly created DMG volume available. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display several `/dev/disk#` entries; use the appropriate disk device in the next command.
  - APFS disks will show many `/dev/disk*` options in the `hdiutil` output. The one we want to mount is the user's macOS volume. We can use the command "`diskutil list /dev/disk4`" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "`Galaga`," highlighted in the example below. We will use `/dev/disk4s1` in the next command. **Be aware that yours may be mounted on a different disk number!**



```
Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3           GUID_partition_scheme
/dev/disk3s1         EFI
/dev/disk3s2         Apple_APFS
/dev/disk4           EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1         41504653-0000-11AA-AA11-0030654
/dev/disk4s2         41504653-0000-11AA-AA11-0030654
/dev/disk4s3         41504653-0000-11AA-AA11-0030654
/dev/disk4s4         41504653-0000-11AA-AA11-0030654
Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the `mount_apfs` command with the following parameters to mount the `/dev/disk#s#` (from the previous command) to the `/Volumes/galaga_mounted/`

mount point. This command requires you to use the sudo command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.

- -o – Options:
    - rdonly: Mount in read-only mode.
    - noexec: Do not allow execution of binaries on a mounted system.
    - noowners: Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

5. **Sanity Check**
    - You can access this newly created mounted drive on /Volumes/galaga_mounted/, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
    - Use the ls -l command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "dlightman" in the Users directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

6. *****When Needed***: Image Unmount Instructions**
    - Use the diskutil list command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "(disk image)" versus the one labeled "(synthesized)". In my example, it would be /dev/disk3.
    - Use the diskutil eject command on the disk you would like to eject.
    - Use the mount command to view the list of mounted disks. Find the disk that you want to unmount (likely /Volumes/galaga_image/, if you are following the naming scheme from the examples.)
    - Use the umount command with the mount point to unmount the disk. You will have to use the sudo command.)
    - *****WARNING***: If you are in the mounted image in Terminal, or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount
$ sudo umount /Volumes/galaga_image
```

macOS User Data

1. **Review the `dlightman` user's `.bash_history` and `.bash_session` files.**
   - Use the `cd` command to change directory to the `dlightman`'s home directory.
   - Use the `cat` command to view the contents of the `.bash_history` file.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/

$ cat .bash_history
```

   1. What command/s were run to check the `dlightman' s` network status?

      ____

   2. What command software did the `dlightman` install via the brew command?

      ____

   - Review the contents of the `.bash_session` files.

```
$ cd .bash_sessions

$ ls -la
```

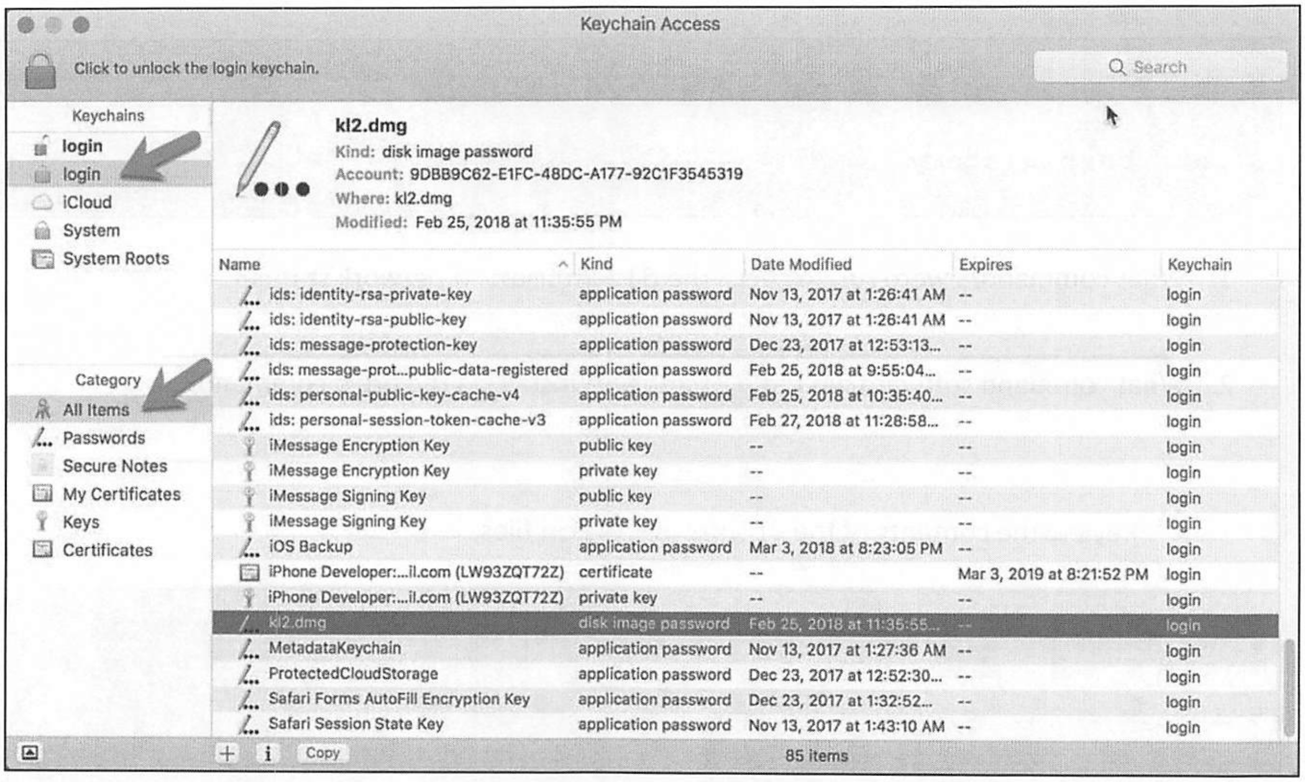
   3. When was the `libimobiledevice` potentially installed?

      ____

2. **Review the `dlightman`'s `Keychains`.**
   - Use the `cd` command to change directory to the `dlightman`'s Keychain directory.
   - Use the `file` command to view the file type for the files in this directory.
   - Use `strings` on the `login.keychain-db` to get an idea of what is contained in the file.
   - Use the open command to view the contents of the `login.keychain-db` using `Keychain Access.app`

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Keychains

$ file *

$ strings login.keychain-db | less

$ open login.keychain-db
```

- Once `Keychain Access.app` has been opened, view the `login.keychain-db`. On the left-hand side choose the correct login keychain under "`Keychains`". The `login.keychain-db` in bold is your keychain—choose the non-bold login keychain. You may have to click back and forth to get it to show in the main pane.
- In the "`Category`" section, choose "`All Items`". This will display all keychain items in the main viewing pane.



kl2.dmg
Kind: disk image password
Account: 9DBB9C62-E1FC-48DC-A177-92C1F3545319
Where: kl2.dmg
Modified: Feb 25, 2018 at 11:35:55 PM

Name	Kind	Date Modified	Expires	Keychain
ids: identity-rsa-private-key	application password	Nov 13, 2017 at 1:26:41 AM	--	login
ids: identity-rsa-public-key	application password	Nov 13, 2017 at 1:26:41 AM	--	login
ids: message-protection-key	application password	Dec 23, 2017 at 12:53:13...	--	login
ids: message-prot...public-data-registered	application password	Feb 25, 2018 at 9:55:04...	--	login
ids: personal-public-key-cache-v4	application password	Feb 25, 2018 at 10:35:40...	--	login
ids: personal-session-token-cache-v3	application password	Feb 27, 2018 at 11:28:58...	--	login
iMessage Encryption Key	public key	--	--	login
iMessage Encryption Key	private key	--	--	login
iMessage Signing Key	public key	--	--	login
iMessage Signing Key	private key	--	--	login
iOS Backup	application password	Mar 3, 2018 at 8:23:05 PM	--	login
iPhone Developer:...il.com (LW93ZQT72Z)	certificate	--	Mar 3, 2019 at 8:21:52 PM	login
iPhone Developer:...il.com (LW93ZQT72Z)	private key	--	--	login
kl2.dmg	disk image password	Feb 25, 2018 at 11:35:55...	--	login
MetadataKeychain	application password	Nov 13, 2017 at 1:27:36 AM	--	login
ProtectedCloudStorage	application password	Dec 23, 2017 at 12:52:30...	--	login
Safari Forms AutoFill Encryption Key	application password	Dec 23, 2017 at 1:32:52...	--	login
Safari Session State Key	application password	Nov 13, 2017 at 1:43:10 AM	--	login

1. What email address appears to be used for many of the credentials?

2. If one of these entries is double-clicked, and the "Show password" checkbox is checked, are you able to see the password?

3. Does this keychain hold the credentials for an iTunes backup?

4. What DMG file's password is stored in this keychain?

- You may remove David's Keychain by right-clicking and selecting "Delete Keychain". Select "Delete References".

3. **Review the `dlightman's Saved Application State` directory.**
   - Use the `cd` command to change the directory to the `dlightman`'s Saved Application State directory.

- Use the `ls -la` command to list the files in this directory; note how some are symbolic links to `Container` directories.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Saved\
Application\ State

$ ls -la

$ cd com.apple.Safari.savedState

$ ls -la

$ open windows.plist
```

1. What website was open in Safari?

iOS User Data

- Review iOS User Data in your FOR518 BlackLight Case File.
- Use the “File Filter” to find and review these files.

4. **Review the Keyboard Dynamic Text**
   - Select the “`physical_logical`” acquisition.
   - Review the contents of the `dynamic-lexicon.dat` file.

macOS System Configuration

- Go back to the mounted macOS Galaga image.

5. **Autoruns**
   - Use the `cd` command to navigate to one of the launch daemons directories.
   - Use the `plutil -p` command to open each launch daemon in this directory.

```
$ cd /Volumes/galaga_mounted/Library/LaunchDaemons/

$ plutil -p keylogger.plist

$ plutil -p logKext.plist
```

1. What path and binary are run for the keylogger named “keylogger”?

2. What is the bundle ID for `logKext`?

macOS User Data

1. **Review the `dlightman` user's `.bash_history` and `.bash_session` files.**
   - Use the `cd` command to change directory to the `dlightman`'s home directory.
   - Use the `cat` command to view the contents of the `.bash_history` file.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/

$ cat .bash_history
```

   1. What command/s were run to check the `dlightman`'s network status?
      a. `ifconfig`
      b. `ping google.com`
   2. What command software did the `dlightman` install via the brew command?
      a. `libimobiledevice` (`brew install libimobiledevice`)

   - Review the contents of the `.bash_session` files.

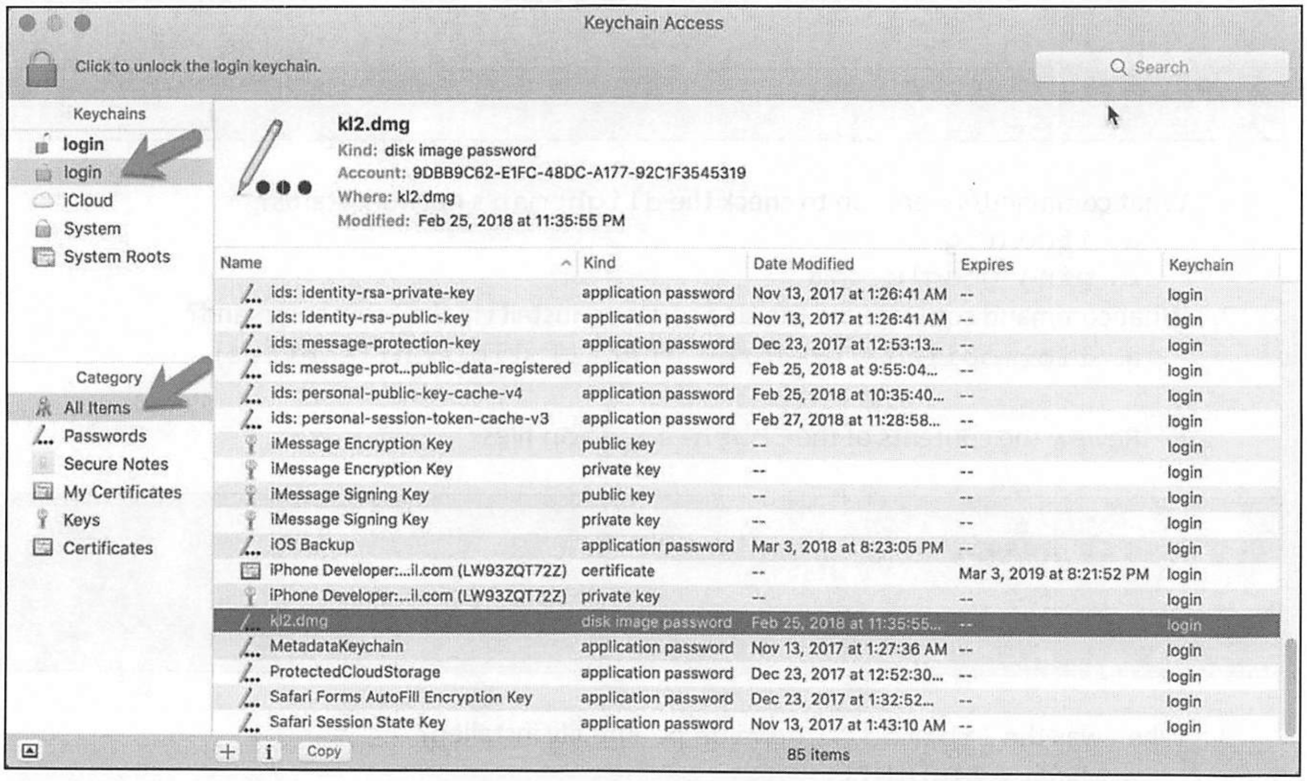
```
$ cd .bash_sessions

$ ls -la
```

   3. When was the `libimobiledevice` potentially installed?
      a. March 3, 2018 (UTC)
      b. Use a grep command to determine the "brew install" entry is located in the file: `B7341ECB-98BB-4863-8220-A965CF7DB9C3.history`
      c. Use `stat -x` command on that file to review MAC timestamps.

2. **Review the `dlightman's Keychains`.**
   - Use the `cd` command to change directory to the `dlightman`'s Keychain directory.
   - Use the `file` command to view the file type for the files in this directory.
   - Use `strings` on the `login.keychain-db` to get an idea of what is contained in the file.
   - Use the open command to view the contents of the `login.keychain-db` using `Keychain Access.app`

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Keychains

$ file *

$ strings login.keychain-db | less
$ open login.keychain-db
```

- Once `Keychain Access.app` has been opened, view the `login.keychain-db`. On the left-hand side, choose the correct login keychain under "`Keychains`". The `login.keychain-db` in bold is <u>your</u> keychain—choose the non-bold login keychain. You may have to click back and forth to get it to show in the main pane.
- In the "`Category`" section, choose "`All Items`". This will display all keychain items in the main viewing pane.



kl2.dmg
Kind: disk image password
Account: 9DBB9C62-E1FC-48DC-A177-92C1F3545319
Where: kl2.dmg
Modified: Feb 25, 2018 at 11:35:55 PM

Name	Kind	Date Modified	Expires	Keychain
ids: identity-rsa-private-key	application password	Nov 13, 2017 at 1:26:41 AM	--	login
ids: identity-rsa-public-key	application password	Nov 13, 2017 at 1:26:41 AM	--	login
ids: message-protection-key	application password	Dec 23, 2017 at 12:53:13...	--	login
ids: message-prot...public-data-registered	application password	Feb 25, 2018 at 9:55:04...	--	login
ids: personal-public-key-cache-v4	application password	Feb 25, 2018 at 10:35:40...	--	login
ids: personal-session-token-cache-v3	application password	Feb 27, 2018 at 11:28:58...	--	login
iMessage Encryption Key	public key	--	--	login
iMessage Encryption Key	private key	--	--	login
iMessage Signing Key	public key	--	--	login
iMessage Signing Key	private key	--	--	login
iOS Backup	application password	Mar 3, 2018 at 8:23:05 PM	--	login
iPhone Developer:...il.com (LW93ZQT72Z)	certificate	--	Mar 3, 2019 at 8:21:52 PM	login
iPhone Developer:...il.com (LW93ZQT72Z)	private key	--	--	login
kl2.dmg	disk image password	Feb 25, 2018 at 11:35:55...	--	login
MetadataKeychain	application password	Nov 13, 2017 at 1:27:36 AM	--	login
ProtectedCloudStorage	application password	Dec 23, 2017 at 12:52:30...	--	login
Safari Forms AutoFill Encryption Key	application password	Dec 23, 2017 at 1:32:52...	--	login
Safari Session State Key	application password	Nov 13, 2017 at 1:43:10 AM	--	login

1. What email address appears to be used for many of the credentials?
    a. `d.l1ghtm4n@gmail.com`
2. If one of these entries is double-clicked, and the "Show password" checkbox is checked, are you able to see the password?
    a. No, a password entry box is opened. You'll need the user's password (by default) to see these passwords.
3. Does this keychain hold the credentials for an iTunes backup?
    a. Yes, the entry labeled "`iOS Backup`" holds these credentials.
4. What DMG file's password is stored in this keychain?
    a. `kl2.dmg`

- You may remove David's Keychain by right-clicking and selecting "Delete Keychain". Select "Delete References".

3. **Review the `dlightman`'s `Saved Application State` directory.**
    - Use the `cd` command to change the directory to the `dlightman`'s Saved Application State directory.

- Use the `ls -la` command to list the files in this directory; note how some are symbolic links to `Container` directories.

```
$ cd /Volumes/galaga_mounted/Users/dlightman/Library/Saved\
Application\ State

$ ls -la

$ cd com.apple.Safari.savedState

$ ls -la

$ open windows.plist
```

1. What website was open in Safari?
   a. Wikipedia(Spider Monkey)
   b. Look for the `NSTitle` keys.

iOS User Data

- Review iOS User Data in your FOR518 BlackLight Case File.
- Use the "File Filter" to find and review these files.

4. **Review the Keyboard Dynamic Text**
   - Select the "`physical_logical`" acquisition.
   - Review the contents of the `dynamic-lexicon.dat` file.

macOS System Configuration

- Go back to the mounted macOS Galaga image.

5. **Autoruns**
   - Use the `cd` command to navigate to one of the launch daemons directories.
   - Use the `plutil -p` command to open each launch daemon in this directory.

```
$ cd /Volumes/galaga_mounted/Library/LaunchDaemons/

$ plutil -p keylogger.plist

$ plutil -p logKext.plist
```

1. What path and binary are run for the keylogger named "keylogger"?
   a. /usr/local/bin/keylogger

    b. `ProgramArguments` Key

2. What is the bundle ID for `logKext`?
    a. `com.fsb.logKext` (`Label` Key)

Lab: Key Takeaways

- **Review the contents of user data and system configuration files.**
- **Start to determine how the system was used.**

Lab 3.2: User Data and System Configuration – Part II

Objectives

- Get familiar with the macOS user preferences and system configuration data files.
- Get familiar with property lists using Xcode.
- Get more comfortable with the macOS command line using Terminal.

Lab Preparation

(Note: Some of this might already be accomplished via earlier labs, but this is the state that we hope your system is in prior to the start of this lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this lab.)

1. **Software Preparation**: The following tools will be used in this lab:
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from /Applications/Utilities/.
   - Xcode.app
     i. Locate and open the Xcode.app from /Applications/.
   - BlackLight.app
     i. Locate and open the BlackLight.app from /Applications/BlackLight/BlackLight YYYY Release #/.

2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Open the FOR518.blacklight BlackLight Case File.**

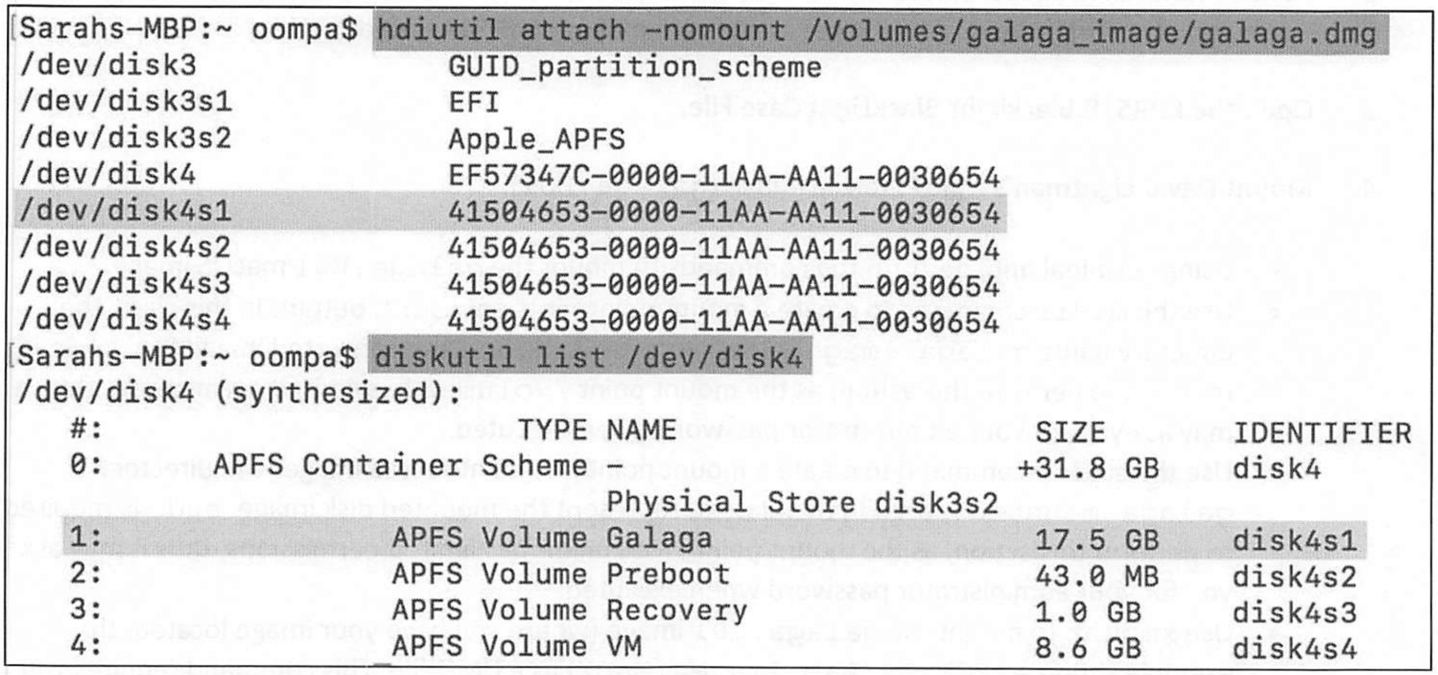
4. **Mount David Lightman's Mac Forensic Image (`galaga.E01`).**

   - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
   - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.

- `--in` – Tells `xmount` what input file type to expect; our images are in a compressed EWF format.
- `--out` – Tells `xmount` what output format you want; we want a DMG file so we can mount it in Finder.
- `Input File` – Where the image file is located on your system.
- `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Use the `hdiutil` command with the "`attach`" verb to make the newly created DMG volume available. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display several `/dev/disk#` entries; use the appropriate disk device in the next command.
  - APFS disks will show many `/dev/disk*` options in the `hdiutil` output. The one we want to mount is the user's macOS volume. We can use the command "`diskutil list /dev/disk4`" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "`Galaga`," highlighted in the example below. We will use `/dev/disk4s1` in the next command. **Be aware that yours may be mounted on a different disk number!**



```
[Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3              GUID_partition_scheme
/dev/disk3s1            EFI
/dev/disk3s2            Apple_APFS
/dev/disk4              EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1            41504653-0000-11AA-AA11-0030654
/dev/disk4s2            41504653-0000-11AA-AA11-0030654
/dev/disk4s3            41504653-0000-11AA-AA11-0030654
/dev/disk4s4            41504653-0000-11AA-AA11-0030654
[Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the `mount_apfs` command with the following parameters to mount the `/dev/disk#s#` (from the previous command) to the `/Volumes/galaga_mounted/`

mount point. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.

- `-o` – Options:
  - `rdonly`: Mount in read-only mode.
  - `noexec`: Do not allow execution of binaries on a mounted system.
  - `noowners`: Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

5. **Sanity Check**
   - You can access this newly created mounted drive on `/Volumes/galaga_mounted/`, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
   - Use the `ls -l` command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "`dlightman`" in the `Users` directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

6. *****When Needed***: Image Unmount Instructions**
   - Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`". In my example, it would be `/dev/disk3`.
   - Use the `diskutil eject` command on the disk you would like to eject.
   - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/`, if you are following the naming scheme from the examples.)
   - Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command.)
   - *****WARNING***: If you are in the mounted image in Terminal, or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount
$ sudo umount /Volumes/galaga_image
```

iOS User Data

- Review iOS User Data in your FOR518 BlackLight Case File.
- Use the “File Filter” to find and review these files.

1. Review the Bluetooth Settings

- Pick any iOS acquisition.
- Review the contents of the `com.apple.MobileBluetooth.ledevices.paired.db` file.

1. Was there an Apple Watch associated with this iPhone?

   ____

2. Review the Icon Settings

- Pick any iOS acquisition.
- Review the contents of the `IconState.plist` file.

1. What is the top left application on David’s iPhone on the second screen?

   ____

macOS System Configuration

- Go back to the mounted macOS Galaga image.

3. Kernel Extensions

- Use the `cd` command to navigate to the kernel extensions directory.
- Use `ls -la` to view the contents of this directory; note the timestamp on `logKext.kext`.
- Use `ls -laR` on `logKext.kext` to view the recursive contents of this kernel extension.
- Use the `plutil -p` command to open the `Info.plist` file for this extension.
- Use `xxd` to view the file signature on the `logKext` binary. (Use "q" to quit out of the `less` command.)

```
$ cd /Volumes/galaga_mounted/System/Library/Extensions/

$ ls -la

$ ls -laR logKext.kext

$ plutil -p logKext.kext/Contents/Info.plist

$ xxd logKext.kext/Contents/MacOS/logKext | less
$ file logKext.kext/Contents/MacOS/logKext
```

1. What is the file signature on the `logKext` binary?

2. What type of file is this (via `file` command)?

4. Printing

- Use the `cd` command to navigate to the system preferences directory.
- Use the `open` command to open the `org.cups.printers.plist` file.

```
$ cd /Volumes/galaga_mounted/Library/Preferences/

$ open org.cups.printers.plist
```

1. What kind of printer was used with this system?

2. How was this printer accessed?

- Use the `cd` command to navigate to the Printer Spool directory.
- Use the `ls -la` command to view all files in this directory. Note the contents of this directory.
- Use `strings` to view the contents of the third print job, `c00003`.

```
$ cd /Volumes/galaga_mounted/private/var/spool/cups/

$ ls -la

$ strings c00003
```

3. Provide the following information for this print job.
   a. What user printed this?

   b. What application did they print from?

   c. What is the name of the print job?

- Use the `file` command on the printer data files.
- Use the `open` command to view the PDF printer data files.

```
$ file d0000*

$ open d0000*
```

4. What did the user print at 3/3/18 4:31 (their system time)?

5. Software Updates

- Use the `cd` command to navigate to the system preferences directory.
- Use the `open` command to open the `com.apple.SoftwareUpdate.plist` file.

```
$ cd /Volumes/galaga_mounted/Library/Preferences/

$ open com.apple.SoftwareUpdate.plist
```

1. What is the name of the recommended update that has yet to install?

- Use the `cd` command to navigate to the software receipts directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory.
- Use the `open` command to open the `InstallHistory.plist` file.

```
$ cd /Volumes/galaga_mounted/Library/Receipts/

$ ls -l

$ open InstallHistory.plist
```

2. How many updates are shown in the `InstallHistory.plist` file?

3. What native application was updated on February 10, 2018?

4. How many times was `logKext` installed?

- Use the cd command to navigate to the software receipts directory where the receipts are stored.
- Use the ls -lt command to view all files in this directory. The "t" option allows us to sort by last modified time. Note how each receipt *.plist and *.bom file modified time matches those found in the InstallHistory.plist file.
- Use the open command to open a plist file. Note the similar data found in the InstallHistory.plist file.
- Use the lsbom -s command to view the files for the Text Wrangler application.

```
$ cd /Volumes/galaga_mounted/var/db/receipts/

$ ls -lt

$ open <anyfile>.plist

$ lsbom -s com.barebones.textwrangler.bom
```

iOS User Data

- Review iOS User Data in your FOR518 BlackLight Case File.
- Use the "File Filter" to find and review these files.

1. Review the Bluetooth Settings

- Pick any iOS acquisition.
- Review the contents of the `com.apple.MobileBluetooth.ledevices.paired.db` file.

1. Was there an Apple Watch associated with this iPhone?
    a. Yes, David's Apple Watch (`PairedDevices` key)

2. Review the Icon Settings

- Pick any iOS acquisition.
- Review the contents of the `IconState.plist` file.

1. What is the top left application on David's iPhone on the second screen?
    a. FaceTime
    b. `Item 1` is the second screen; `com.apple.facetime` is the bundle ID shown first (it goes top to bottom, left to right).

macOS System Configuration

- Go back to the mounted macOS Galaga image.

3. Kernel Extensions

- Use the `cd` command to navigate to the kernel extensions directory.
- Use `ls -la` to view the contents of this directory; note the timestamp on `logKext.kext`.
- Use `ls -laR` on `logKext.kext` to view the recursive contents of this kernel extension.
- Use the `plutil -p` command to open the `Info.plist` file for this extension.
- Use `xxd` to view the file signature on the `logKext` binary. (Use "`q`" to quit out of the `less` command.)

```
$ cd /Volumes/galaga_mounted/System/Library/Extensions/

$ ls -la

$ ls -laR logKext.kext

$ plutil -p logKext.kext/Contents/Info.plist

$ xxd logKext.kext/Contents/MacOS/logKext | less
$ file logKext.kext/Contents/MacOS/logKext
```

1. What is the file signature on the `logKext` binary?
    a. `cffa edfe`
    b. The first four bytes of the file.
2. What type of file is this (via `file` command)?
    a. `Mach-O 64-bit kext bundle x86_64`

4. Printing

- Use the `cd` command to navigate to the system preferences directory.
- Use the `open` command to open the `org.cups.printers.plist` file.

```
$ cd /Volumes/galaga_mounted/Library/Preferences/

$ open org.cups.printers.plist
```

1. What kind of printer was used with this system?
    a. "Brother HL-L2380DW series" (`printer-make-and-model` Key)
2. How was this printer accessed?
    a. Via the network (`device-uri` Key)

- Use the `cd` command to navigate to the Printer Spool directory.
- Use the `ls -la` command to view all files in this directory. Note the contents of this directory.
- Use `strings` to view the contents of the third print job, `c00003`.

```
$ cd /Volumes/galaga_mounted/private/var/spool/cups/

$ ls -la

$ strings c00003
```

3. Provide the following information for this print job.
    a. What user printed this?
        i. David Lightman
        ii. Search around the term "`com.apple.print.JobInfo.PMJobOwner`"
    b. What application did they print from?
        i. Safari
        ii. Search around the term "`com.apple.print.JobInfo.PMApplicationName`"
    c. What is the name of the print job?
        i. Red panda – Wikipedia

ii. Search around the term “job-name” or “com.apple.print.JobInfo.PMJobName”

- Use the file command on the printer data files.
- Use the open command to view the PDF printer data files.

```
$ file d0000*

$ open d0000*
```

4. What did the user print at 3/3/18 4:31 (their system time)?
   a. The first page of the Wikipedia article for Spider Monkey (d00004-001)

5. Software Updates

- Use the cd command to navigate to the system preferences directory.
- Use the open command to open the com.apple.SoftwareUpdate.plist file.

```
$ cd /Volumes/galaga_mounted/Library/Preferences/

$ open com.apple.SoftwareUpdate.plist
```

1. What is the name of the recommended update that has yet to install?
   a. macOS High Sierra 10.13.3 Update Combo
   b. Use the “Display Name” Key.

- Use the cd command to navigate to the software receipts directory.
- Use the ls -l command to view all files in this directory. Note the contents of this directory.
- Use the open command to open the InstallHistory.plist file.

```
$ cd /Volumes/galaga_mounted/Library/Receipts/

$ ls -l

$ open InstallHistory.plist
```

2. How many updates are shown in the InstallHistory.plist file?
   a. 14 Items
   b. Look at the Root Key.
3. What native application was updated on February 10, 2018?
   a. iTunes
   b. Item 9
4. How many times was logKext installed?

a. Twice
b. `Item 11` and `12`

- Use the `cd` command to navigate to the software receipts directory where the receipts are stored.
- Use the `ls -lt` command to view all files in this directory. The "`t`" option allows us to sort by last modified time. Note how each receipt `*.plist` and `*.bom` file modified time matches those found in the `InstallHistory.plist` file.
- Use the `open` command to open a plist file. Note the similar data found in the `InstallHistory.plist` file.
- Use the `lsbom -s` command to view the files for the Text Wrangler application.

```
$ cd /Volumes/galaga_mounted/var/db/receipts/

$ ls -lt

$ open <anyfile>.plist

$ lsbom -s com.barebones.textwrangler.bom
```

Exercise: Key Takeaways

- **Review the contents of user data and system configuration files.**

- **Start to determine how the system was used.**

Lab 3.3: Log Parsing and Analysis

Objectives

- Know where the key log files are stored and how to parse the Apple System Logs, Basic Security Module Audit logs, and Unified logs.
- Get familiar with the macOS command line.

Lab Preparation

(Note: Some of this might already be accomplished via earlier labs, but this is the state that we hope your system is in prior to the start of this lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this lab.)

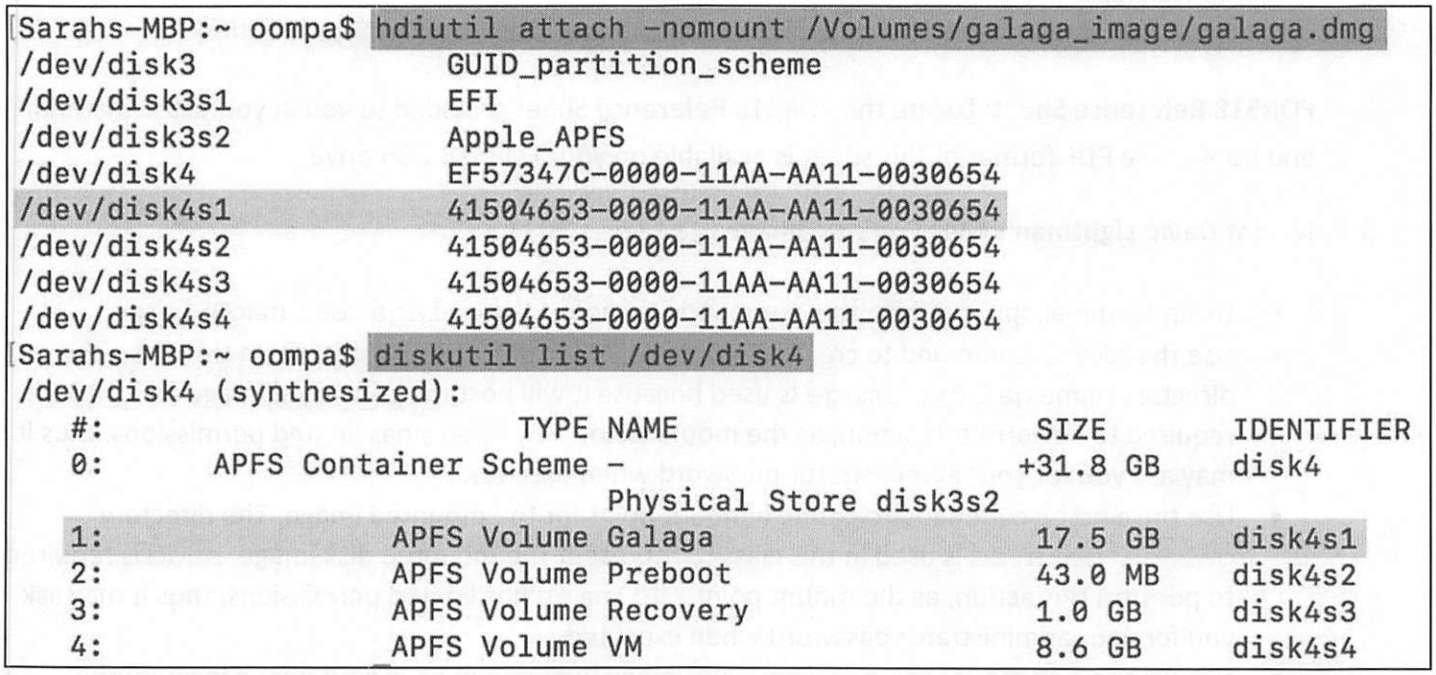
1. **Software Preparation**: The following tools will be used in this lab:
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from /Applications/Utilities/
   - Console.app
     i. Locate and open the native OS X Console.app from /Applications/Utilities/

2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Mount David Lightman's Mac Forensic Image (**`galaga.E01`**).**
   - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
   - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows ~`/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.
     - `--in` – Tells `xmount` what input file type to expect; our images are in a compressed EWF format.
     - `--out` – Tells `xmount` what output format you want; we want a DMG file so we can mount it in Finder.
     - `Input File` – Where the image file is located on your system.
     - `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Use the hdiutil command with the “attach” verb to make the newly created DMG volume available. Use the -nomount argument to suppress mounting (for now). The output from this command will display several /dev/disk# entries; use the appropriate disk device in the next command.
  - o APFS disks will show many /dev/disk* options in the hdiutil output. The one we want to mount is the user’s macOS volume. We can use the command "diskutil list /dev/disk4" on the synthesized disk to determine which is likely the user’s macOS volume. David Lightman’s volume is named "Galaga," highlighted in the example below. We will use /dev/disk4s1 in the next command. **Be aware that yours may be mounted on a different disk number!**



```
[Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3              GUID_partition_scheme
/dev/disk3s1            EFI
/dev/disk3s2            Apple_APFS
/dev/disk4              EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1            41504653-0000-11AA-AA11-0030654
/dev/disk4s2            41504653-0000-11AA-AA11-0030654
/dev/disk4s3            41504653-0000-11AA-AA11-0030654
/dev/disk4s4            41504653-0000-11AA-AA11-0030654
[Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the mount_apfs command with the following parameters to mount the /dev/disk#s# (from the previous command) to the /Volumes/galaga_mounted/ mount point. This command requires you to use the sudo command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
  - o -o – Options:
    - ▪ rdonly – Mount in read-only mode.
    - ▪ noexec – Do not allow execution of binaries on the mounted system.
    - ▪ noowners – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

4. **Sanity Check**
   - You can access this newly created mounted drive on `/Volumes/galaga_mounted/`, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
   - Use the `ls -l` command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "`dlightman`" in the `Users` directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

5. *****When Needed***: Image Unmount Instructions**
   - Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`". In my example, it would be `/dev/disk3`.
   - Use the `diskutil eject` command on the disk you would like to eject.
   - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/` if you are following the naming scheme from the examples).
   - Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command.
   - *****WARNING***: If you are in the mounted image in Terminal or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

Lab: Questions

1. Introduction to the Console Application

- Locate and open the native macOS `Console.app` from `/Applications/Utilities/`.
- This will show you the log contents of your host system.
- **Briefly**, review the log files in the sidebar on the left.
    i. Note the different locations where logs may be found.

2. System Log Directory and Bzip2 Compression

- Use the `cd` command to navigate to the System Log directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory.
- Use the `file` command to view the file types listed for these files. Note the files labeled as "`bzip2 compressed data`".

```
$ cd /Volumes/galaga_mounted/private/var/log/

$ ls -l

$ file *
```

1. What set of log files has been archived using Bzip2 compression?

2. What set of log files has been archived using Gzip compression?

- Use the `gzcat` and `cat` commands to decompress and create a comprehensive log file of the `system.log`. Output this log file to your FOR518 directory as `system_all.log`.

```
$ gzcat system.log.{5..0}.gz > ~/FOR518/system_all.log

$ cat system.log >> ~/FOR518/system_all.log
```

3. Use the `wc -l` command to determine how many records are now in the `system_all.log` file.

3. Apple System Log (ASL) Directory

- Use the `cd` command to navigate to the Apple System Log directory.
- Use the `ls -la` command to view all files in this directory. Note the contents of this directory.

```
$ cd /Volumes/galaga_mounted/private/var/log/asl/

$ ls -la
```

1. What is the date of the oldest ASL log file (not including "best before" ASL files)?

   ____

2. How many days in the past are events recorded, as shown by the ASL filenames (not including "best before" ASL files)?

   ____

4. ASL Log Conversion Using the `syslog` Command

- View the man page for the `syslog` command using the `man` command.
  - i. **Briefly**, review its contents.
  - ii. Use the spacebar to page down.
  - iii. Press "q" when ready to quit the viewer.

```
$ man syslog
```

- Use the `syslog` command to view the contents of any ASL log file.
- Note the contents and format of this output.

```
$ syslog -f 2018.02.28.U501.asl
```

- Use the `syslog` command to view the contents of the same ASL log file in RAW format.
- Note the differences in the log output.

```
$ syslog -F raw -f 2018.02.28.U501.asl
```

- Use the `syslog` command to output all the ASL logs in this directory using the UTC timestamp in RAW format.
  - i. Your terminal should be set with the UTC time zone; if not, use the "`export TZ=UTC`".
  - ii. Redirect the output to a file `ASL.log` in your FOR518 directory.
  - iii. You can check the time zone of the terminal window by using the `date` command and looking at the time zone.
- Open the `ASL.log` log in Console.app using the `open` command.
  - i. Review this output.

```
$ export TZ=UTC

$ syslog -F raw -T utc -d . > ~/FOR518/ASL.log

$ open -a Console ~/FOR518/ASL.log
```

1. How many records are there? (Hint: Use `wc -l` command.)

2. What is the date (UTC) of the first message?

3. What is the facility of the first message?

4. When does the first message expire?

5. How long is this message kept for?

6. What is the date (UTC) of the last message?

7. What is the hostname used in this message?

8. How long until this message expires?

5. Basic Security Module Audit Logs

- Ensure the time zone of your Terminal window is UTC using the `export TZ=UTC` command.
- Use the `cd` command to navigate to the Audit Log Directory.
- Use the `ls -la` command to view all files in this directory. Note the contents of this directory.

```
$ export TZ=UTC

$ cd /Volumes/galaga_mounted/private/var/audit/

$ ls -la
```

1. What is the start timestamp of the oldest audit log file?

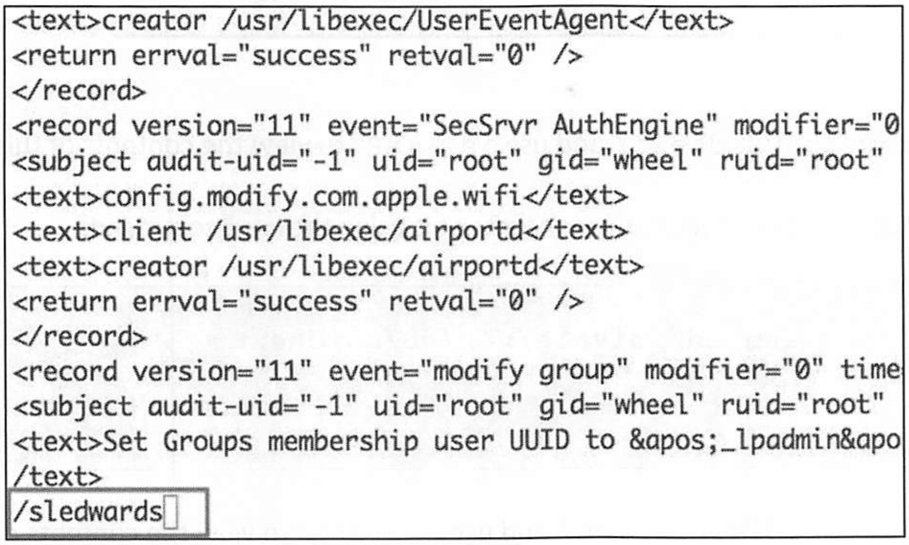
- Use the `praudit` command to output the contents of any single audit log file.
    i. Use the `less` command to control the output.
- Review the output of this command.

```
$ praudit 20171113011901.crash_recovery | less
```

- Use the praudit command to output the contents of the audit log in XML format.
    i. Use the less command to control the output.
- Review the output of this command. Note how the data pieces are now labeled.

```
$ praudit -x 20171113011901.crash_recovery | less
```

- Perform a search for a username.
    i. While in the less output from the previous command, type a "/", then type the username for user 501 on your system. (i.e., /sledwards). Hit [return]. This will search the output for this username.
    ii. <u>**A username on your system**</u> should not be showing up in someone else's logs! (Hint: **This will only work if you have a user 501**; some systems that are network-logon-based may not have one.) **If you are not user 501 on your system, please skip this demo.**
    iii. The praudit command is translating the current users of the system into the output of these logs – **not good for forensics!**



```
<text>creator /usr/libexec/UserEventAgent</text>
<return errval="success" retval="0" />
</record>
<record version="11" event="SecSrvr AuthEngine" modifier="0
<subject audit-uid="-1" uid="root" gid="wheel" ruid="root"
<text>config.modify.com.apple.wifi</text>
<text>client /usr/libexec/airportd</text>
<text>creator /usr/libexec/airportd</text>
<return errval="success" retval="0" />
</record>
<record version="11" event="modify group" modifier="0" time
<subject audit-uid="-1" uid="root" gid="wheel" ruid="root"
<text>Set Groups membership user UUID to &apos;_lpadmin&apo
/text>
/sledwards
```

- Use the "-n" option to stop the UID and GID translation.
- Perform the same search—does your username show up now?

```
$ praudit -xn 20171113011901.crash_recovery | less
```

- Use the praudit command to output the contents of the audit logs in this directory to a file in your FOR518 directory named audit.log.
    i. The "*.*" notation is used so as not to include the "current" link. (This file is already included, and the link is pointing to your own file system.)
- Review the contents in Console.app.

```
$ praudit -xn *.* > ~/FOR518/audit.log
$ open -a Console ~/FOR518/audit.log
```

- To search, press Command+F—this will allow you to search the contents while still viewing all the contents.
- The search box located in the top-right of the application will filter contents based on a search string. While convenient for records using one line, this causes issues when records are multi-line, much like these XML-based records.

2. When was the user `dlightman` created? (Search "`create user`".)

   ____

3. Find the "user authentication" event recorded Mon Nov 13 01:26:35 2017. What user authenticated to the system?

   ____

4. Did the default `Guest` user ever log in successfully?

   ____

6. Unified Logs

- Navigate to `/var/db/uuidtext/` and use `ls -laR` to view the contents of this directory recursively.
- Feel free to select a file and review the contents of it using the `xxd` command.

```
$ cd /Volumes/galaga_mounted/private/var/db/uuidtext

$ ls -laR
```

- Navigate to `/var/db/diagnostics/` and use `ls -laR` to view the contents of this directory recursively.

```
$ cd /Volumes/galaga_mounted/private/var/db/diagnostics

$ ls -laR
```

- View the man page for the `log` command using the `man` command.
  i. **Briefly**, review its contents.
  ii. Use the spacebar to page down.
  iii. Press "q" when ready to quit the viewer.

```
$ man log
```

- Use the `log stream` command on your own system.
- Wow! Lots of logs! **You can quit this by using Ctrl+C.**
- This command is useful for research and testing of different scenarios to see what the logs may look like.

```
$ log stream
```

- Navigate up one directory to `/var/db/`
- Use `mkdir` to make a log archive for this system named `galaga.logarchive`
- Use `cp` to copy the `uuidtext` and `diagnostics` directories to this log archive.

```
$ cd ../

$ mkdir ~/FOR518/galaga.logarchive

$ cp -R diagnostics/ uuidtext/ ~/FOR518/galaga.logarchive
```

- Use `log show` on this newly created log archive file.
- This gives you an "`Archive format needs updating…`" message. Go ahead and re-run the command with `--force` to update it.
- Run the log show command again without `--force`, but pipe it to `less`.
- Expect to get some errors, although the log file appears to be ok.
- Note the "Skipping info and debug messages" message; let's get the `info` messages too!
- Re-run with `--info`; let's also change the time zone to UTC with `--timezone`
- Briefly browse the content and format of this output.

```
$ log show ~/FOR518/galaga.logarchive

$ log show --force ~/FOR518/galaga.logarchive

$ log show ~/FOR518/galaga.logarchive | less

$ log show --info --timezone utc ~/FOR518/galaga.logarchive | less
```

1. What is the timestamp of the first record?

____

- Give the `log stats` command a try. This may take a few moments to run. Review the output.

```
$ log stats --overview --archive ~/FOR518/galaga.logarchive/
```

1. Introduction to the Console Application

- Locate and open the native macOS `Console.app` from `/Applications/Utilities/`.
- This will show you the log contents of your host system.
- **Briefly**, review the log files in the sidebar on the left.
    i. Note the different locations where logs may be found.

2. System Log Directory and BZip2 Compression

- Use the `cd` command to navigate to the System Log directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory.
- Use the `file` command to view the file types listed for these files. Note the files labeled as "`bzip2 compressed data`".

```
$ cd /Volumes/galaga_mounted/private/var/log/

$ ls -l

$ file *
```

1. What set of log files has been archived using BZip2 compression?
    a. wifi.log.bz2*
2. What set of log files has been archived using GZip compression?
    a. system.log.gz*

- Use the `gzcat` and `cat` commands to decompress and create a comprehensive log file of the `system.log`. Output this log file to your FOR518 directory as `system_all.log`.

```
$ gzcat system.log.{5..0}.gz > ~/FOR518/system_all.log

$ cat system.log >> ~/FOR518/system_all.log
```

3. Use the `wc -l` command to determine how many records are now in the `system_all.log` file.
    a. 18,254 records (`wc -l ~/FOR518/system_all.log`)

3. Apple System Log (ASL) Directory

- Use the `cd` command to navigate to the Apple System Log directory.
- Use the `ls -la` command to view all files in this directory. Note the contents of this directory.

```
$ cd /Volumes/galaga_mounted/private/var/log/asl/

$ ls -la
```

1. What is the date of the oldest ASL log file (not including "best before" ASL files)?
   a. 02/25/2018
2. How many days in the past are events recorded as shown by the ASL filenames (not including "best before" ASL files)?
   a. Seven (2/25/2018–3/3/2018)

4. ASL Log Conversion Using the `syslog` Command

- View the man page for the `syslog` command using the `man` command.
  i. **Briefly**, review its contents.
  ii. Use the spacebar to page down.
  iii. Press "q" when ready to quit the viewer.

```
$ man syslog
```

- Use the `syslog` command to view the contents of any ASL log file.
- Note the contents and format of this output.

```
$ syslog -f 2018.02.28.U501.asl
```

- Use the `syslog` command to view the contents of the same ASL log file in RAW format.
- Note the differences in the log output.

```
$ syslog -F raw -f 2018.02.28.U501.asl
```

- Use the `syslog` command to output all the ASL logs in this directory using the UTC timestamp in RAW format.
  i. Your terminal should be set with the UTC time zone; if not, use the "`export TZ=UTC`".
  ii. Redirect the output to a file `ASL.log` in your FOR518 directory.
  iii. You can check the time zone of the terminal window by using the `date` command and looking at the time zone.
- Open the `ASL.log` log in Console.app using the `open` command.
  i. Review this output.

```
$ export TZ=UTC

$ syslog -F raw -T utc -d . > ~/FOR518/ASL.log

$ open -a Console ~/FOR518/ASL.log
```

1. How many records are there? (Hint: Use `wc  -l` command.)

a. 14,006 Records

2. What is the date (UTC) of the first message?
    a. 2017-11-13 01:18:25Z
3. What is the facility of the first message?
    a. `com.apple.system.utmpx`
4. When does the first message expire?
    a. 1542158305 = 2018-11-14 01:18:25 Wed UTC
        i. `ASLExpireTime` Field
5. How long is this message kept for?
    a. One Year + 1 Day (366 days or 31,622,400 seconds via `man asl.conf`)
6. What is the date (UTC) of the last message?
    a. 2018-03-03 21:43:59Z
7. What is the hostname used in this message?
    a. "Davids-MBP"
    b. `Host` Field
8. How long until this message expires?
    a. Seven days
    b. If no `ASLExpireTime` field is present, default expire time is seven days from the date of the message.

5. Basic Security Module Audit Logs

- Ensure the time zone of your Terminal window is UTC using the `export TZ=UTC` command.
- Use the `cd` command to navigate to the Audit Log Directory.
- Use the `ls -la` command to view all files in this directory. Note the contents of this directory.

```
$ export TZ=UTC

$ cd /Volumes/galaga_mounted/private/var/audit/

$ ls -la
```

1. What is the start timestamp of the oldest audit log file?
    a. 20171113011901 = November 13, 2017 01:19:01

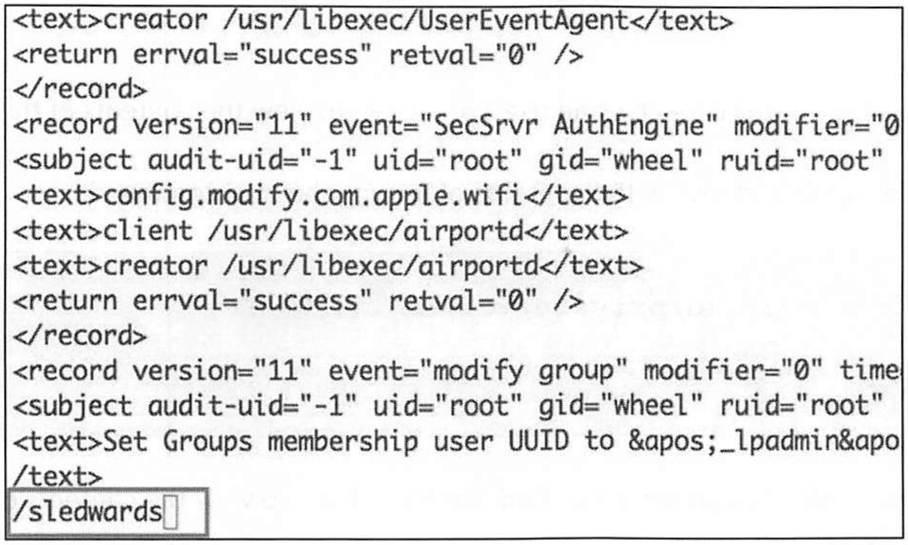
- Use the `praudit` command to output the contents of any single audit log file.
    i. Use the `less` command to control the output.
- Review the output of this command.

```
$ praudit 20171113011901.crash_recovery | less
```

- Use the `praudit` command to output the contents of the audit log in XML format.
    i. Use the `less` command to control the output.
- Review the output of this command. Note how the data pieces are now labeled.

```
$ praudit -x 20171113011901.crash_recovery | less
```

- Perform a search for a username.
    i. While in the `less` output from the previous command, type a "/", then type the username for user 501 on your system. (i.e., /sledwards). Hit [return]. This will search the output for this username.
    ii. <u>**A username on your system**</u> should not be showing up in someone else's logs! (Hint: **This will only work if you have a user 501**; some systems that are network-logon-based may not have one.) **If you are not user 501 on your system, please skip this demo.**
    iii. The `praudit` command is translating the current users of the system into the output of these logs – **not good for forensics**!



```
<text>creator /usr/libexec/UserEventAgent</text>
<return errval="success" retval="0" />
</record>
<record version="11" event="SecSrvr AuthEngine" modifier="0
<subject audit-uid="-1" uid="root" gid="wheel" ruid="root"
<text>config.modify.com.apple.wifi</text>
<text>client /usr/libexec/airportd</text>
<text>creator /usr/libexec/airportd</text>
<return errval="success" retval="0" />
</record>
<record version="11" event="modify group" modifier="0" time
<subject audit-uid="-1" uid="root" gid="wheel" ruid="root"
<text>Set Groups membership user UUID to &apos;_lpadmin&apo
/text>
/sledwards
```

- Use the "`-n`" option to stop the UID and GID translation.
- Perform the same search—does your username show up now?

```
$ praudit -xn 20171113011901.crash_recovery | less
```

- Use the `praudit` command to output the contents of the audit logs in this directory to a file in your FOR518 directory named `audit.log`.
    i. The "`*.*`" notation is used so as not to include the "`current`" link. (This file is already included, and the link is pointing to your own file system.)
- Review the contents in Console.app.

```
$ praudit -xn *.* > ~/FOR518/audit.log

$ open -a Console ~/FOR518/audit.log
```

- To search, press Command+F—this will allow you to search the contents while still viewing all the contents.
- The search box located in the top-right of the application will filter contents based on a search string. While convenient for records using one line, this causes issues when records are multi-line, much like these XML-based records.

2. When was the user `dlightman` created? (Search "`create user`".)
    a. Mon Nov 13 01:26:28 2017
3. Find the "user authentication" event recorded Mon Nov 13 01:26:35 2017. What user authenticated to the system?
    b. `dlightman`
4. Did the default `Guest` user ever log in successfully?
    c. Sure did, on Sat Feb 10 13:54:19 2018.

6. Unified Logs

- Navigate to `/var/db/uuidtext/` and use `ls -laR` to view the contents of this directory recursively.
- Feel free to select a file and review the contents of it using the `xxd` command.

```
$ cd /Volumes/galaga_mounted/private/var/db/uuidtext

$ ls -laR
```

- Navigate to `/var/db/diagnostics/` and use `ls -laR` to view the contents of this directory recursively.

```
$ cd /Volumes/galaga_mounted/private/var/db/diagnostics

$ ls -laR
```

- View the man page for the `log` command using the `man` command.
    i. **Briefly**, review its contents.
    ii. Use the spacebar to page down.
    iii. Press "q" when ready to quit the viewer.

```
$ man log
```

- Use the `log stream` command on your own system.
- Wow! Lots of logs! **You can quit this by using Ctrl+C**.
- This command is useful for research and testing of different scenarios to see what the logs may look like.

```
$ log stream
```

- Navigate up one directory to /var/db/
- Use mkdir to make a log archive for this system named galaga.logarchive
- Use cp to copy the uuidtext and diagnostics directories to this log archive.

```
$ cd ../

$ mkdir ~/FOR518/galaga.logarchive

$ cp -R diagnostics/ uuidtext/ ~/FOR518/galaga.logarchive
```

- Use log show on this newly created log archive file.
- This gives you an "Archive format needs updating…" message. Go ahead and re-run the command with --force to update it.
- Run the log show command again without --force, but pipe it to less.
- Expect to get some errors, although the log file appears to be ok.
- Note the "Skipping info and debug messages" message; let's get the info messages too!
- Re-run with --info; let's also change the time zone to UTC with --timezone
- Briefly browse the content and format of this output.

```
$ log show ~/FOR518/galaga.logarchive

$ log show --force ~/FOR518/galaga.logarchive

$ log show ~/FOR518/galaga.logarchive | less

$ log show --info --timezone utc ~/FOR518/galaga.logarchive | less
```

1. What is the timestamp of the first record?
   a. 2018-02-07 08:49:48.946180+0000
   b. (2018-02-07 03:49:48.946180-0500 if you didn't change the time zone)
   c. Note the time zone and the microseconds!

- Give the log stats command a try. This may take a few moments to run. Review the output.

```
$ log stats --overview --archive ~/FOR518/galaga.logarchive/
```

Exercise: Key Takeaways

- **Know how to parse these log files by hand; most tools do not parse these automatically.**
- **Get comfortable with some macOS command-line utilities.**

Lab 3.4: Timeline Analysis and Data Correlation

Objectives

- Get familiar with correlating events in a timeline using log analysis and data correlation of key macOS data files.

Lab Preparation

(Note: Some of this might already be accomplished via earlier labs, but this is the state that we hope your system is in prior to the start of this lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this lab.)

1. **Software Preparation**: The following tools will be used in this lab:
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from /Applications/Utilities/
   - Console.app
     i. Locate and open the native OS X Console.app from /Applications/Utilities/

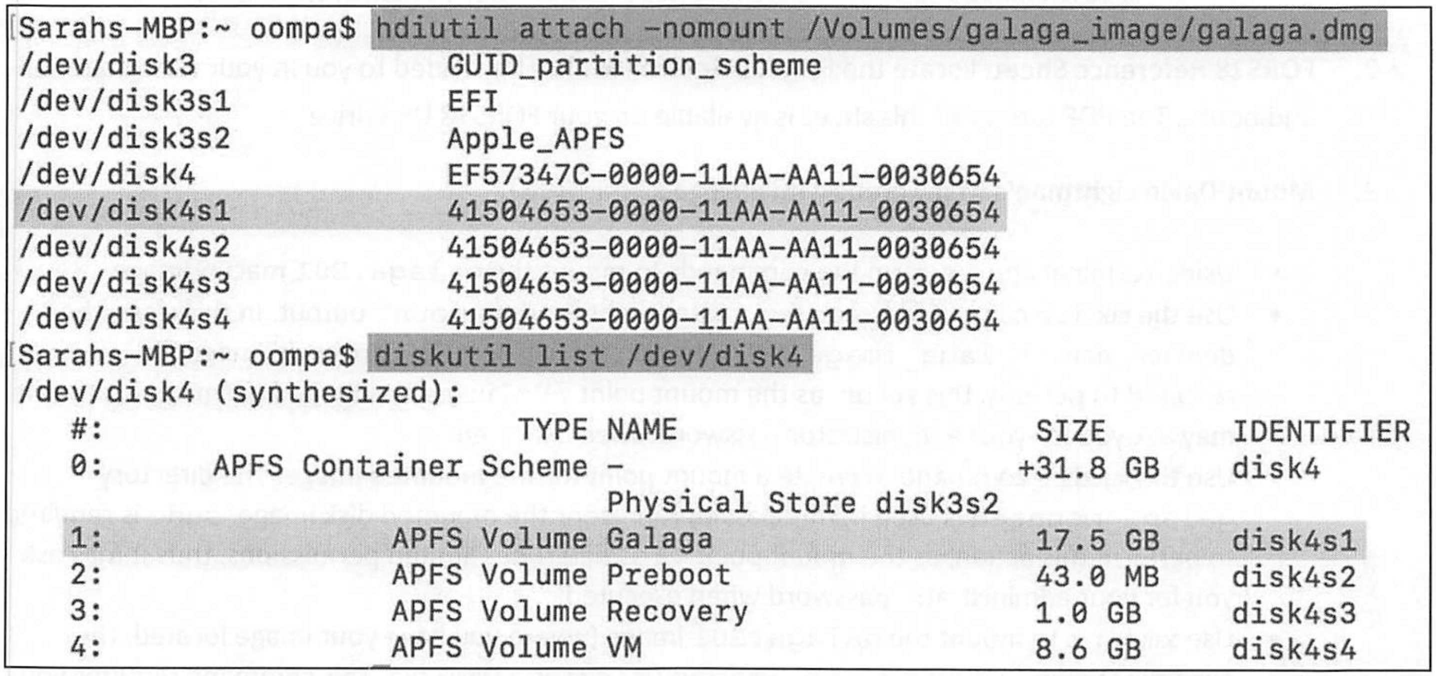
2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Mount David Lightman's Mac Forensic Image (`galaga.E01`).**

   - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
   - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows ~`/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.
     - `--in` – Tells `xmount` what input file type to expect, our images are in a compressed EWF format.
     - `--out` – Tells `xmount` what output format you want, we want a DMG file so we can mount it in Finder.
     - `Input File` – Where the image file is located on your system.
     - `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Use the `hdiutil` command with the "`attach`" verb to make the newly created DMG volume available. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display several `/dev/disk#` entries; use the appropriate disk device in the next command.
  - APFS disks will show many `/dev/disk*` options in the `hdiutil` output. The one we want to mount is the user's macOS volume. We can use the command "`diskutil list /dev/disk4`" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "`Galaga`," highlighted in the example below. We will use `/dev/disk4s1` in the next command. **Be aware that yours may be mounted on a different disk number!**



```
Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3          	GUID_partition_scheme
/dev/disk3s1        	EFI
/dev/disk3s2        	Apple_APFS
/dev/disk4          	EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1        	41504653-0000-11AA-AA11-0030654
/dev/disk4s2        	41504653-0000-11AA-AA11-0030654
/dev/disk4s3        	41504653-0000-11AA-AA11-0030654
/dev/disk4s4        	41504653-0000-11AA-AA11-0030654
Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the `mount_apfs` command with the following parameters to mount the `/dev/disk#s#` (from the previous command) to the `/Volumes/galaga_mounted/` mount point. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
  - `-o` – Options:
    - `rdonly` – Mount in read-only mode.
    - `noexec` – Do not allow execution of binaries on mounted system.
    - `noowners` – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

4. **Sanity Check**
   - You can access this newly created mounted drive on `/Volumes/galaga_mounted/`, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
   - Use the `ls -l` command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "`dlightman`" in the `Users` directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

5. *****When Needed***: Image Unmount Instructions**
   - Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`". In my example, it would be `/dev/disk3`.
   - Use the `diskutil eject` command on the disk you would like to eject.
   - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/` if you are following the naming scheme from the examples.)
   - Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command.
   - *****WARNING***: If you are in the mounted image in Terminal or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

Lab: Questions

1. **Choose Your Own Adventure Log Analysis**

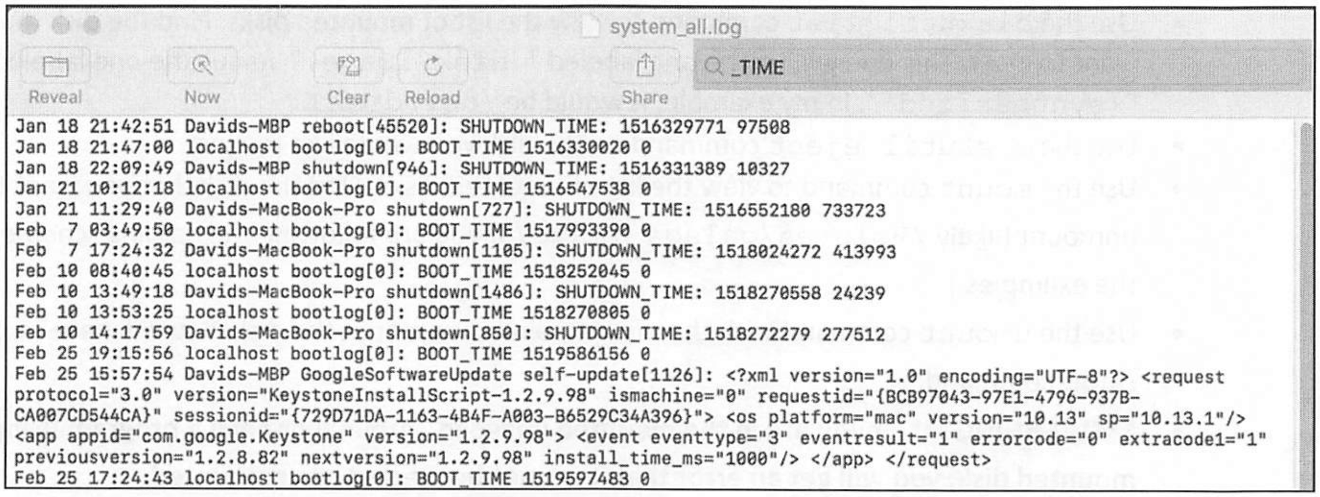
Choose one of two choices:

- **Choice A: Use Console.app.**
- **Choice B: Use the command line.**

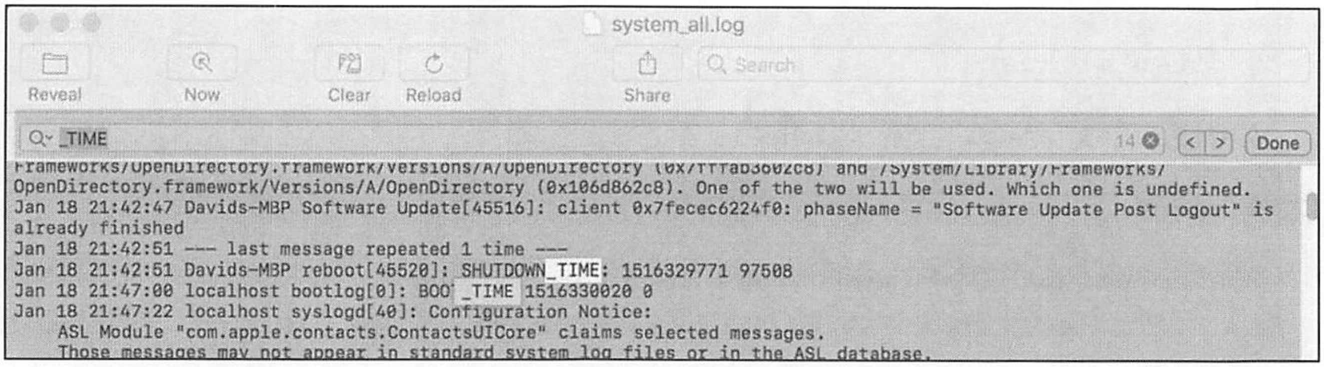
- **Choice A: Console.app.**
  1. Use the `open` command to open the log file of interest in Console.app. You do not necessarily have to open a specific log file; remove the <example>.log section and you will open your logs in Console.

```
$ open -a Console <example>.log
```

  2. Use the search functions:
     1. Filter text box at the top right
     2. "Find" Function - Edit | Find (Command+F)

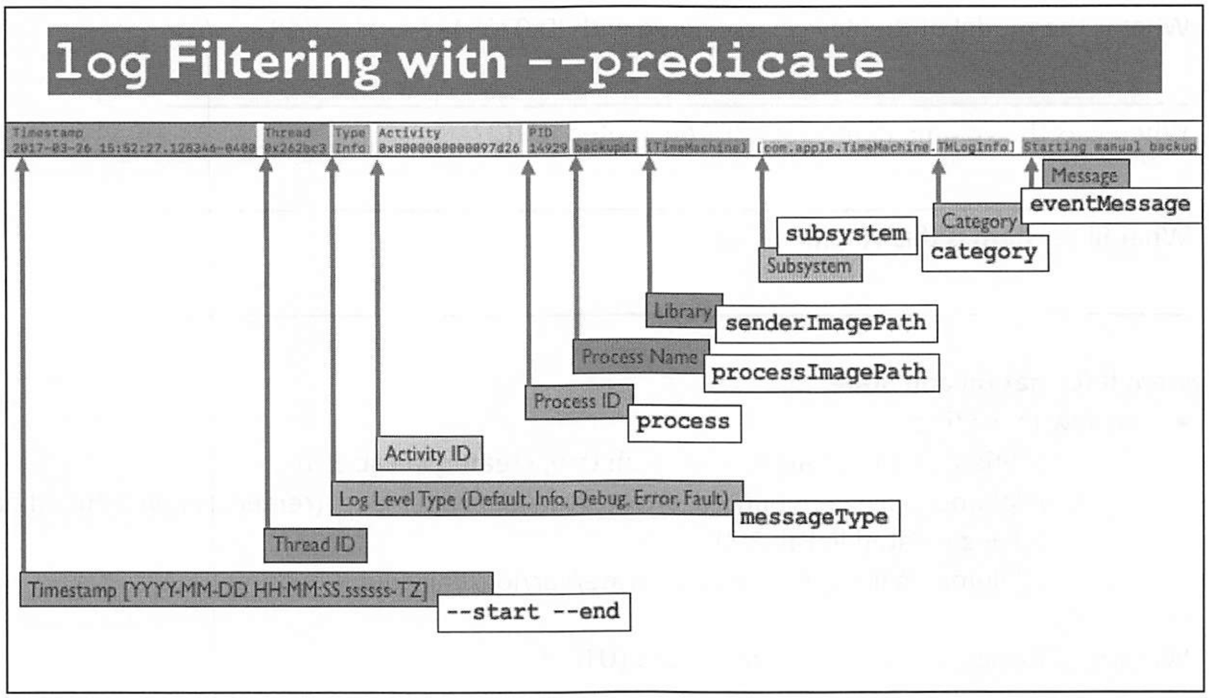


```
Jan 18 21:42:51 Davids-MBP reboot[45520]: SHUTDOWN_TIME: 1516329771 97508
Jan 18 21:47:00 localhost bootlog[0]: BOOT_TIME 1516330020 0
Jan 18 22:09:49 Davids-MBP shutdown[946]: SHUTDOWN_TIME: 1516331389 10327
Jan 21 10:12:18 localhost bootlog[0]: BOOT_TIME 1516547538 0
Jan 21 11:29:40 Davids-MacBook-Pro shutdown[727]: SHUTDOWN_TIME: 1516552180 733723
Feb  7 03:49:50 localhost bootlog[0]: BOOT_TIME 1517993390 0
Feb  7 17:24:32 Davids-MacBook-Pro shutdown[1105]: SHUTDOWN_TIME: 1518024272 413993
Feb 10 08:40:45 localhost bootlog[0]: BOOT_TIME 1518252045 0
Feb 10 13:49:18 Davids-MacBook-Pro shutdown[1486]: SHUTDOWN_TIME: 1518270558 24239
Feb 10 13:53:25 localhost bootlog[0]: BOOT_TIME 1518270805 0
Feb 10 14:17:59 Davids-MacBook-Pro shutdown[850]: SHUTDOWN_TIME: 1518272279 277512
Feb 25 19:15:56 localhost bootlog[0]: BOOT_TIME 1519586156 0
Feb 25 15:57:54 Davids-MBP GoogleSoftwareUpdate self-update[1126]: <?xml version="1.0" encoding="UTF-8"?> <request
protocol="3.0" version="KeystoneInstallScript-1.2.9.98" ismachine="0" requestid="{BCB97043-97E1-4796-937B-
CA007CD544CA}" sessionid="{729D71DA-1163-4B4F-A003-B6529C34A396}"> <os platform="mac" version="10.13" sp="10.13.1"/>
<app appid="com.google.Keystone" version="1.2.9.98"> <event eventtype="3" eventresult="1" errorcode="0" extracode1="1"
previousversion="1.2.8.82" nextversion="1.2.9.98" install_time_ms="1000"/> </app> </request>
Feb 25 17:24:43 localhost bootlog[0]: BOOT_TIME 1519597483 0
```



```
Frameworks/OpenDirectory.framework/Versions/A/OpenDirectory (0x7fffab3602c8) and /System/Library/Frameworks/
OpenDirectory.framework/Versions/A/OpenDirectory (0x106d862c8). One of the two will be used. Which one is undefined.
Jan 18 21:42:47 Davids-MBP Software Update[45516]: client 0x7fecec6224f0: phaseName = "Software Update Post Logout" is
already finished
Jan 18 21:42:51 --- last message repeated 1 time ---
Jan 18 21:42:51 Davids-MBP reboot[45520]: SHUTDOWN_TIME: 1516329771 97508
Jan 18 21:47:00 localhost bootlog[0]: BOOT_TIME 1516330020 0
Jan 18 21:47:22 localhost syslogd[40]: Configuration Notice:
    ASL Module "com.apple.contacts.ContactsUICore" claims selected messages.
    Those messages may not appear in standard system log files or in the ASL database.
```

- **Choice B: Via the command line using `grep` and/or `log` commands**
  1. Use the `log` command with `--predicate` filtering.

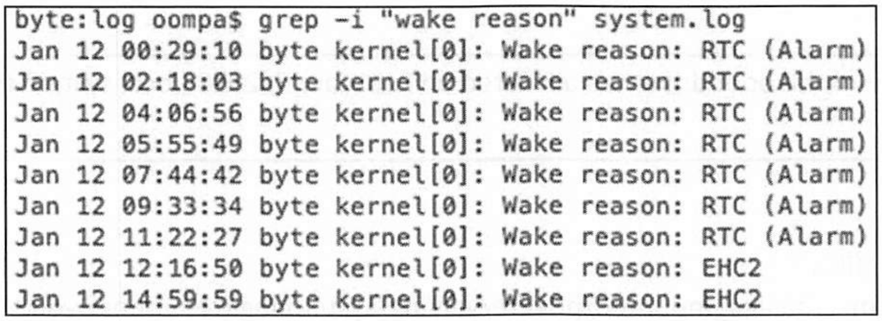


log Filtering with --predicate

```
Timestamp                        Thread   Type  Activity            PID
2017-03-26 15:52:27.128346-0400  0x262bc3 Info  0x8000000000097d26  14929 backupd: (TimeMachine) [com.apple.TimeMachine.TMLogInfo] Starting manual backup
```

ii. Use the grep command to search for items of interest.

1. Recommended for students with previous grep experience.
2. Use the man grep command for options to this utility.



```
byte:log oompa$ grep -i "wake reason" system.log
Jan 12 00:29:10 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 02:18:03 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 04:06:56 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 05:55:49 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 07:44:42 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 09:33:34 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 11:22:27 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 12:16:50 byte kernel[0]: Wake reason: EHC2
Jan 12 14:59:59 byte kernel[0]: Wake reason: EHC2
```

```
$ grep -i "wake reason" <example>.log
```

2. Volume Analysis

- Review these files.
  - i. Unified Logs (galaga.logarchive, created in Lab 3.3)

1. What is the USBMSC identifier for the actual USB device mounted most often in the Unified Logs?

____

2. Looking at the Vendor ID, what company makes the device inserted into the system on March 3, 2018, at 16:11 (UTC)?

____

3. What is the model of the device associated with ID 070843790D1DDF61?

   ____

4. When was the volume named SEKRET encrypted (UTC)?

   ____

5. What file system is this volume using?

   ____

3. System Information and State

- Review these files.
  - i. Unified Logs (`galaga.logarchive`, created in Lab 3.3)
  - ii. /Volumes/galaga_mounted/private/var/log/system.log (remember you should have the full log created in Lab 3.3)
  - iii. /Volumes/galaga_mounted /private/var/log/daily.out

1. When was the last time this system booted (UTC)?

   ____

2. Was this system ever hard powered down?

   ____

3. On February 10, 2018 (local system time), what time zone was this system in?

   ____

4. What percentage of the boot drive was allocated on February 7, 2018 (local system time)?

   ____

4. Network Analysis

- Review these files.
  - i. /Volumes/galaga_mounted/private/var/log/system.log (remember you should have the full log created in Lab 3.3)
  - ii. /Volumes/galaga_mounted/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist

1. What four Wi-Fi networks did this system associate to?

   ____

2. Create a timeline of travel activity (UTC):

Time Frame	Wi-Fi SSID(s)	Possible IP(s)	Possible Location/Country
Feb 07–Feb 10, 2018			
Feb 25–March 3, 2018			

5. **User Access**
   - Review these files.
      i. /Volumes/galaga_mounted/private/var/log/system.log (remember you should have the full log created in Lab 3.3)
      ii. /Volumes/galaga_mounted/private/var/log/asl (remember you should have the full log created in Lab 3.3)
      iii. /Volumes/galaga_mounted/private/var/audit (remember you should have the full log created in Lab 3.3)

1. What two methods did users use to log on to this system?

   ____

2. What are the start time and end time of the logon session associated with PID 612 (local system time)?

   ____

3. What user account logged on at this time?

   ____

6. **Software Installation**
   - Review these files.
      i. /Volumes/galaga_mounted/private/var/log/install.log

1. What software was installed with administrative rights?

   ____

2. When was this system first installed (local system time)?

   ____

Extra Credit:

- **Keep reviewing log files, including those not included in this lab—get comfortable with the different types of events in each.**
- **Review these files in the BlackLight application.**

Lab: Step-by-Step

1. Choose Your Own Adventure Log Analysis

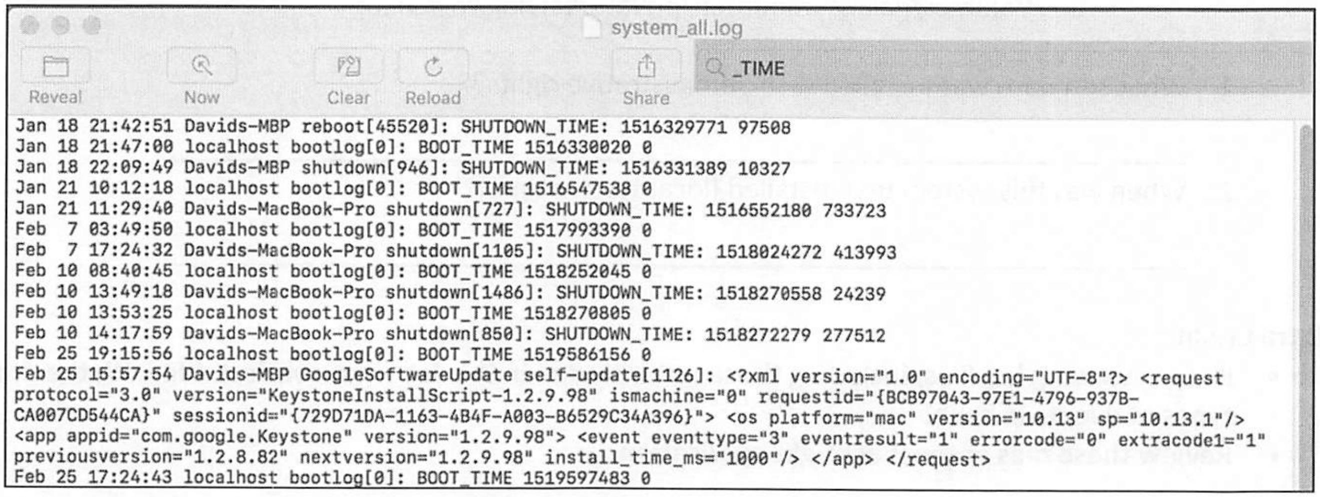
Choose one of two choices:

- **Choice A: Use Console.app.**
- **Choice B: Use the command line.**

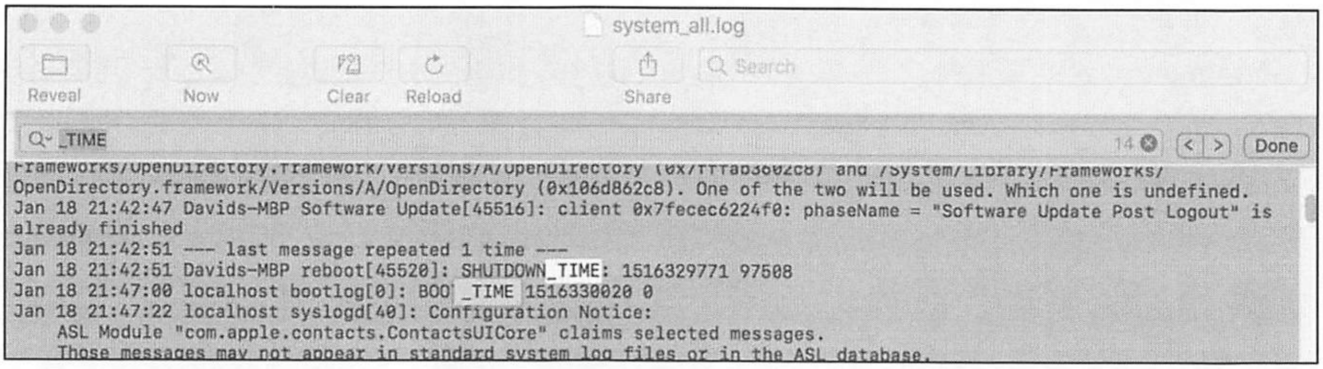
- **Choice A: Console.app.**
  i. Use the `open` command to open the log file of interest in Console.app. You do not necessarily have to open a specific log file; remove the <example>.log section and you will open your logs in Console.

```
$ open -a Console <example>.log
```

  ii. Use the search functions:
    1. Filter text box at the top right
    2. "Find" Function - Edit | Find (Command+F)

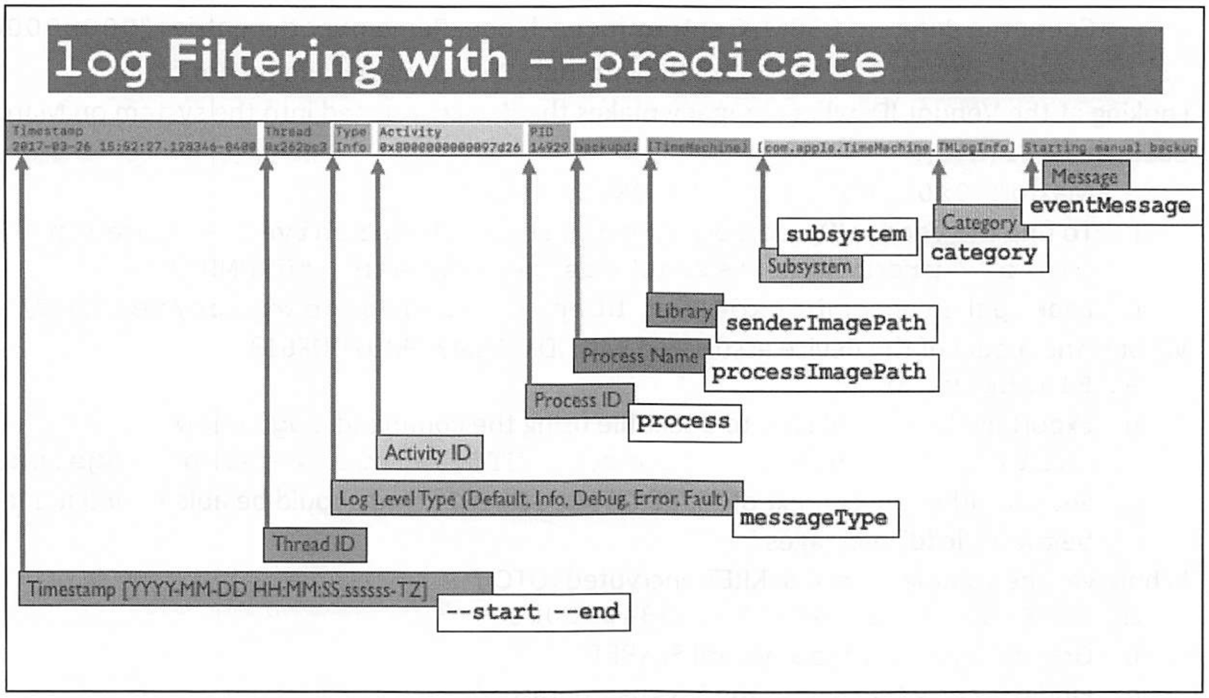


```
Jan 18 21:42:51 Davids-MBP reboot[45520]: SHUTDOWN_TIME: 1516329771 97508
Jan 18 21:47:00 localhost bootlog[0]: BOOT_TIME 1516330020 0
Jan 18 22:09:49 Davids-MBP shutdown[946]: SHUTDOWN_TIME: 1516331389 10327
Jan 21 10:12:18 localhost bootlog[0]: BOOT_TIME 1516547538 0
Jan 21 11:29:40 Davids-MacBook-Pro shutdown[727]: SHUTDOWN_TIME: 1516552180 733723
Feb  7 03:49:50 localhost bootlog[0]: BOOT_TIME 1517993390 0
Feb  7 17:24:32 Davids-MacBook-Pro shutdown[1105]: SHUTDOWN_TIME: 1518024272 413993
Feb 10 08:40:45 localhost bootlog[0]: BOOT_TIME 1518252045 0
Feb 10 13:49:18 Davids-MacBook-Pro shutdown[1486]: SHUTDOWN_TIME: 1518270558 24239
Feb 10 13:53:25 localhost bootlog[0]: BOOT_TIME 1518270805 0
Feb 10 14:17:59 Davids-MacBook-Pro shutdown[850]: SHUTDOWN_TIME: 1518272279 277512
Feb 25 19:15:56 localhost bootlog[0]: BOOT_TIME 1519586156 0
Feb 25 15:57:54 Davids-MBP GoogleSoftwareUpdate self-update[1126]: <?xml version="1.0" encoding="UTF-8"?> <request
protocol="3.0" version="KeystoneInstallScript-1.2.9.98" ismachine="0" requestid="{BCB97043-97E1-4796-937B-
CA007CD544CA}" sessionid="{729D71DA-1163-4B4F-A003-B6529C34A396}"> <os platform="mac" version="10.13" sp="10.13.1"/>
<app appid="com.google.Keystone" version="1.2.9.98"> <event eventtype="3" eventresult="1" errorcode="0" extracode1="1"
previousversion="1.2.8.82" nextversion="1.2.9.98" install_time_ms="1000"/> </app> </request>
Feb 25 17:24:43 localhost bootlog[0]: BOOT_TIME 1519597483 0
```



```
Frameworks/OpenDirectory.framework/Versions/A/OpenDirectory (0x7fffab3602c8) and /System/Library/Frameworks/
OpenDirectory.framework/Versions/A/OpenDirectory (0x106d862c8). One of the two will be used. Which one is undefined.
Jan 18 21:42:47 Davids-MBP Software Update[45516]: client 0x7fecec6224f0: phaseName = "Software Update Post Logout" is
already finished
Jan 18 21:42:51 --- last message repeated 1 time ---
Jan 18 21:42:51 Davids-MBP reboot[45520]: SHUTDOWN_TIME: 1516329771 97508
Jan 18 21:47:00 localhost bootlog[0]: BOO _TIME 1516330020 0
Jan 18 21:47:22 localhost syslogd[40]: Configuration Notice:
    ASL Module "com.apple.contacts.ContactsUICore" claims selected messages.
    Those messages may not appear in standard system log files or in the ASL database.
```

- **Choice B: Via the command line using `grep` and/or `log` commands**
  i. Use the `log` command with `--predicate` filtering.

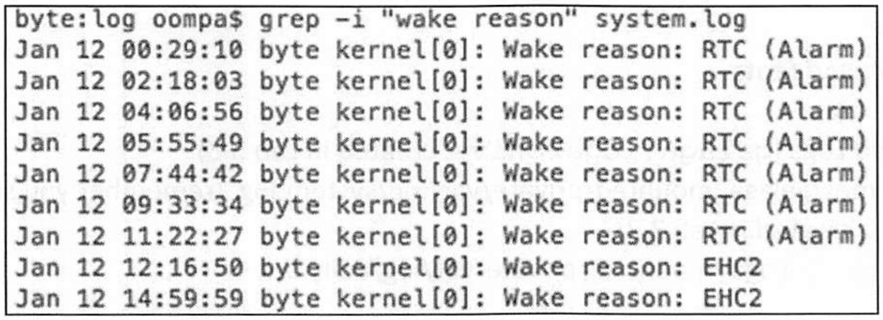


log Filtering with --predicate

```
Timestamp                     Thread   Type Activity           PID
2017-03-26 15:52:27.128346-0400 0x262bc3 Info 0x8000000000097d26 14929 backupd: [TimeMachine] (com.apple.TimeMachine.TMLogInfo) Starting manual backup
```

ii. Use the `grep` command to search for items of interest.

1. Recommended for students with previous `grep` experience.
2. Use the man `grep` command for options to this utility.



```
byte:log oompa$ grep -i "wake reason" system.log
Jan 12 00:29:10 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 02:18:03 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 04:06:56 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 05:55:49 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 07:44:42 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 09:33:34 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 11:22:27 byte kernel[0]: Wake reason: RTC (Alarm)
Jan 12 12:16:50 byte kernel[0]: Wake reason: EHC2
Jan 12 14:59:59 byte kernel[0]: Wake reason: EHC2
```

```
$ grep -i "wake reason" <example>.log
```

2. Volume Analysis

- Review these files.
  i. Unified Logs (`galaga.logarchive`, created in Lab 3.3)

1. What is the USBMSC identifier for the actual USB device mounted most often in the Unified Logs?
   a. AA01102412155309 3678
   b. Use `log show galaga.logarchive/ --timezone UTC --info --predicate 'eventMessage contains "USBMSC"'| awk '{print $13}' | sort | uniq -c`
      i. This command filters the USBMSC entries using `log` and pipes the results to an `awk` command to print out only the 13th column (the ID). This output gets piped to `sort` to sort them so it can finally be piped to `uniq -c` to uniquely count each entry.

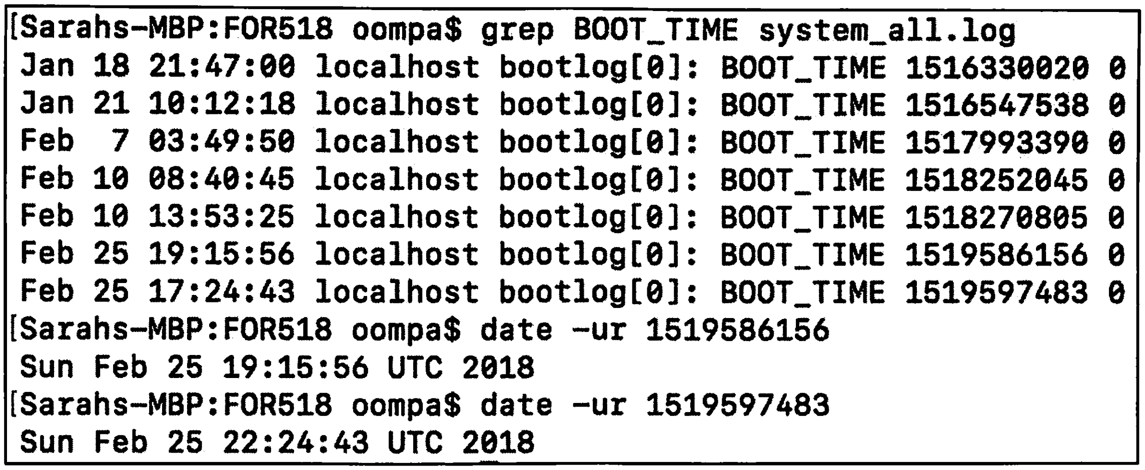
   c. Count the different USBMSC entries for each one. Remember the entries “`000000000820 0x5ac 0x8406 0x820`” are the internal SD card reader and do not count.

2. Looking at the Vendor ID, what company makes the device inserted into the system on March 3, 2018, at 16:11 (UTC)?
   a. Maxtor: `0x00000000 0xd49 0x7250 0x1`
   b. To find the Vendor ID, use `log show galaga.logarchive/ --timezone UTC --info --predicate 'eventMessage contains "USBMSC"'`
   c. Look up the Vendor ID “`0xd49`” at `http://usb-ids.gowdy.us/read/UD/`
3. What is the model of the device associated with ID 070843790D1DDF61?
   a. `FlashBlu 30`
   b. Export all the Unified Logs to a text file using the command `log show galaga.logarchive/ --timezone UTC --info > galaga_logs.txt`
   c. Search within the context of “070843790D1DDF61”; you should be able to find it a few lines below in “icdd” messages.
4. When was the volume named SEKRET encrypted (UTC)?
   a. `2018-02-26 01:49:08.719840+0000`
   b. Grep or search for the keyword SEKRET.
   c. Find the entry that shows the “-S <passphrase>”.
   d. `diskmanagementd: diskmanagement: execve(2) pid=1276 /System/Library/Filesystems/apfs.fs/Contents/Resources/newfs_apfs -A -i -E -S frogger13 -v SEKRET disk5 .`
5. What file system is this volume using?
   a. APFS (note the use of `newfs_apfs` command).

3. System Information and State

- Review these files.
  i. Unified Logs (`galaga.logarchive`, created in Lab 3.3)
  ii. /Volumes/galaga_mounted/private/var/log/system.log (remember you should have the full log created in Lab 3.3)
  iii. /Volumes/galaga_mounted /private/var/log/daily.out

1. When was the last time this system booted (UTC)?
   a. Sun Feb 25 22:24:43 UTC 2018
   b. Grep or search for `BOOT_TIME` in the system.log.
   c. Even though the Feb 25 19:15:56 timestamped entry is later, in reality, the record timestamp was recorded in local system time, using the Unix epoch timestamp; the last entry is the last startup time, as shown below with the date command.



```
[Sarahs-MBP:FOR518 oompa$ grep BOOT_TIME system_all.log
 Jan 18 21:47:00 localhost bootlog[0]: BOOT_TIME 1516330020 0
 Jan 21 10:12:18 localhost bootlog[0]: BOOT_TIME 1516547538 0
 Feb  7 03:49:50 localhost bootlog[0]: BOOT_TIME 1517993390 0
 Feb 10 08:40:45 localhost bootlog[0]: BOOT_TIME 1518252045 0
 Feb 10 13:53:25 localhost bootlog[0]: BOOT_TIME 1518270805 0
 Feb 25 19:15:56 localhost bootlog[0]: BOOT_TIME 1519586156 0
 Feb 25 17:24:43 localhost bootlog[0]: BOOT_TIME 1519597483 0
[Sarahs-MBP:FOR518 oompa$ date -ur 1519586156
 Sun Feb 25 19:15:56 UTC 2018
[Sarahs-MBP:FOR518 oompa$ date -ur 1519597483
 Sun Feb 25 22:24:43 UTC 2018
```

2. Was this system ever hard powered down?
   a. Yes, on 2018-02-07 08:49:49.735856+0000
   b. Search for "`shutdown cause`" in the Unified Logs. look for entries with a "3".
   c. `log show galaga.logarchive/ --timezone UTC --info -predicate 'eventMessage contains[c] "shutdown cause"'`
3. On February 10, 2018 (local system time), what time zone was this system in?
   a. GMT
   b. Do a search for the timestamps in daily.out with "2018"; look for Feb 10.
   c. Sat Feb 10 08:49:08 GMT 2018
4. What percentage of the boot drive was allocated on February 7, 2018 (local system time)?
   a. 75%
   b. daily.out log: Search for the day then look in the Disk Status area for the percentage for root disk "/".

4. Network Analysis

- Review these files.
  i. /Volumes/galaga_mounted/private/var/log/system.log (remember you should have the full log created in Lab 3.3)
  ii. /Volumes/galaga_mounted/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist

1. What four Wi-Fi networks did this system associate to?
   a. CrystalPalace (Unified/Airport plist)
   b. De Vere Grand Connaught Rooms (Unified/Airport plist)
   c. shmoocon (Airport plist)
   d. acetomato (Unified only—it was removed by the user in the preferences panel, therefore it was not in the plist file.)
   e. `log show galaga.logarchive/ --timezone UTC --info --predicate 'eventMessage contains "BSSID"' | grep configd`

2. Create a timeline of travel activity (UTC):
   a. `log show galaga.logarchive/ --timezone UTC --info --predicate '(senderImagePath contains[cd] "IPConfiguration" and eventMessage contains[cd] "SSID") or eventMessage contains[cd] "network changed" or eventMessage contains "country code set"'`

Time Frame	Wi-Fi SSID(s) Search "SSID" in Unified Logs	Possible IP(s) Search "network changed" in Unified Logs	Possible Location/Country
Feb 07–Feb 10 2018	De Vere Grand Connaught Rooms	10.5.48.38 10.5.49.169	De Vere Grand Connaught Rooms in United Kingdom (GMT)
Feb 25–March 3 2018	CrystalPalace acetomato	192.168.101.138 (Crystal Palace) 192.168.8.133 (acetomato)	Home (Crystal Palace) in US

5. **User Access**

- Review these files.
  i. /Volumes/galaga_mounted/private/var/log/system.log (remember you should have the full log created in Lab 3.3)
  ii. /Volumes/galaga_mounted/private/var/log/asl (remember you should have the full log created in Lab 3.3)
  iii. /Volumes/galaga_mounted/private/var/audit (remember you should have the full log created in Lab 3.3)

1. What two methods did users use to log on to this system?
   a. Login Window
   b. Terminal
   c. Search "_PROCESS:" in system.log (use the full log created).
2. What are the start time and end time of the logon session associated with PID 612 (local system time)?
   a. Feb 25 17:53:12 -> Mar 3 15:27:39 (2018)
   b. `Feb 25 17:53:12 Davids-MBP login[612]: USER_PROCESS: 612 ttys000`
   c. `Mar  3 15:27:39 Davids-MBP login[612]: DEAD_PROCESS: 612 ttys000`
3. What user account logged on at this time?
   a. dlightman (Search for "612" or timestamps in ASL.log or audit.log)
   b. Audit File:

```
<record version="11" event="logout - local" modifier="0" time="Sat Mar
3 20:27:39 2018" msec=" + 633 msec" >
<subject audit-uid="501" uid="0" gid="20" ruid="501" rgid="20"
pid="612" sid="612" tid="2684354560.0.0.0" />
<return errval="success" retval="0" />
</record>
```

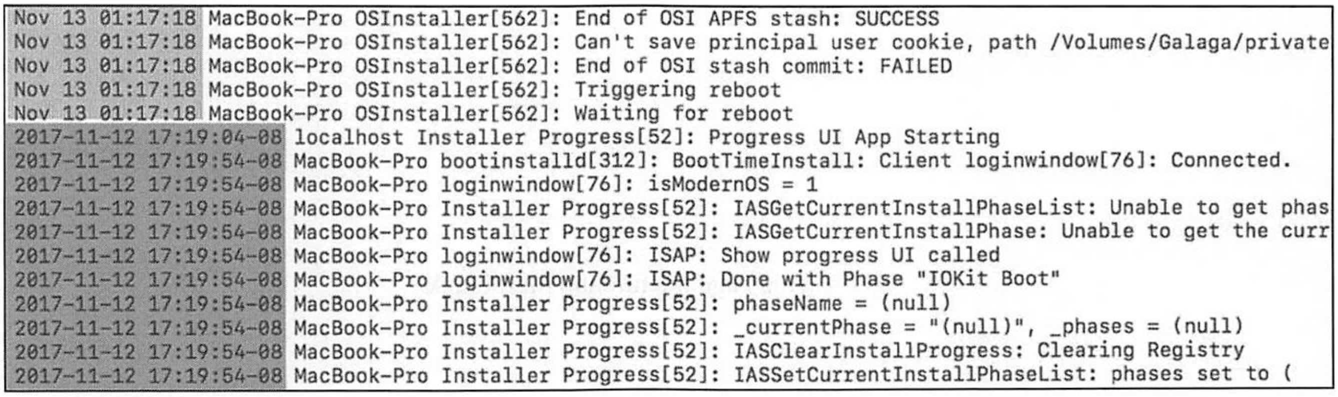
   c. ASL File:

```
[ASLMessageID 24869] [Time 2018-02-25 22:53:12Z] [TimeNanoSec
433279000] [Level 5] [PID 612] [UID 0] [GID 20] [ReadGID 80] [Host
Davids-MBP] [Sender login] [Facility com.apple.system.lastlog]
[Message USER_PROCESS: 612 ttys000] [ut_user dlightman] [ut_id s000]
[ut_line ttys000] [ut_pid 612] [ut_type 7] [ut_tv.tv_sec 1519599192]
```

```
[ut_tv.tv_usec 433191] [SenderMachUUID 9015BFF2-0D5C-34E3-BE7E-
15DA6FC115C6] [ASLExpireTime 1551221592]
```

6. **Software Installation**
   - Review these files.
     i. /Volumes/galaga_mounted/private/var/log/install.log

1. What software was installed with administrative rights?
   a. logKext (twice)
   b. Search "Administrator authorization granted"
2. When was this system first installed (local system time)?
   a. November 12, 2017
   b. Watch the timestamps jump from Nov 13 back to Nov 12 (it is adjusting from Cupertino time to local system time Eastern as shown by the timestamp "time zone" of "-08").



```
Nov 13 01:17:18 MacBook-Pro OSInstaller[562]: End of OSI APFS stash: SUCCESS
Nov 13 01:17:18 MacBook-Pro OSInstaller[562]: Can't save principal user cookie, path /Volumes/Galaga/private
Nov 13 01:17:18 MacBook-Pro OSInstaller[562]: End of OSI stash commit: FAILED
Nov 13 01:17:18 MacBook-Pro OSInstaller[562]: Triggering reboot
Nov 13 01:17:18 MacBook-Pro OSInstaller[562]: Waiting for reboot
2017-11-12 17:19:04-08 localhost Installer Progress[52]: Progress UI App Starting
2017-11-12 17:19:54-08 MacBook-Pro bootinstalld[312]: BootTimeInstall: Client loginwindow[76]: Connected.
2017-11-12 17:19:54-08 MacBook-Pro loginwindow[76]: isModernOS = 1
2017-11-12 17:19:54-08 MacBook-Pro Installer Progress[52]: IASGetCurrentInstallPhaseList: Unable to get phas
2017-11-12 17:19:54-08 MacBook-Pro Installer Progress[52]: IASGetCurrentInstallPhase: Unable to get the curr
2017-11-12 17:19:54-08 MacBook-Pro loginwindow[76]: ISAP: Show progress UI called
2017-11-12 17:19:54-08 MacBook-Pro loginwindow[76]: ISAP: Done with Phase "IOKit Boot"
2017-11-12 17:19:54-08 MacBook-Pro Installer Progress[52]: phaseName = (null)
2017-11-12 17:19:54-08 MacBook-Pro Installer Progress[52]: _currentPhase = "(null)", _phases = (null)
2017-11-12 17:19:54-08 MacBook-Pro Installer Progress[52]: IASClearInstallProgress: Clearing Registry
2017-11-12 17:19:54-08 MacBook-Pro Installer Progress[52]: IASSetCurrentInstallPhaseList: phases set to (
```

Extra Credit:

- **Keep reviewing log files, including those not included in this lab—get comfortable with the different types of events in each.**
- **Review these files in the BlackLight application.**

Lab: Key Takeaways

- **Using various log files and data files, correlation can be done to prove or disprove different activities.**

This page intentionally left blank.

Lab 4.1: Safari and Mail

Objectives

- Introduce the key data files associated with the Safari web browser and Apple Mail (with some extras thrown in!).
- Parse these data files using native, free, and commercial toolsets.
- Recognize differences in tool output versus raw data.

Lab Preparation

(Note: Some of this might already be accomplished via earlier labs, but this is the state that we hope your system is in prior to the start of this lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this lab.)

1. **Software Preparation**: The following tools will be used in this lab:
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from /Applications/Utilities/.
   - Xcode.app
     i. Locate and open the Xcode.app from /Applications/.
   - SQLite Database Browser
     i. You will be using the SQLite Database Browser (Applications/sqlitebrowser.app).
     ii. These tools are available on your USB drive in the Tools directory.
     iii. The SQLite Manager is available at `http://sqlitebrowser.org/`.
   - BlackLight.app
     i. Locate and open the Blacklight.app from /Applications/Blacklight 201# Release #/Blacklight.app.
     ii. This tool is available on your USB drive in the Tools directory.

2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Open the FOR518.blacklight BlackLight Case File.**

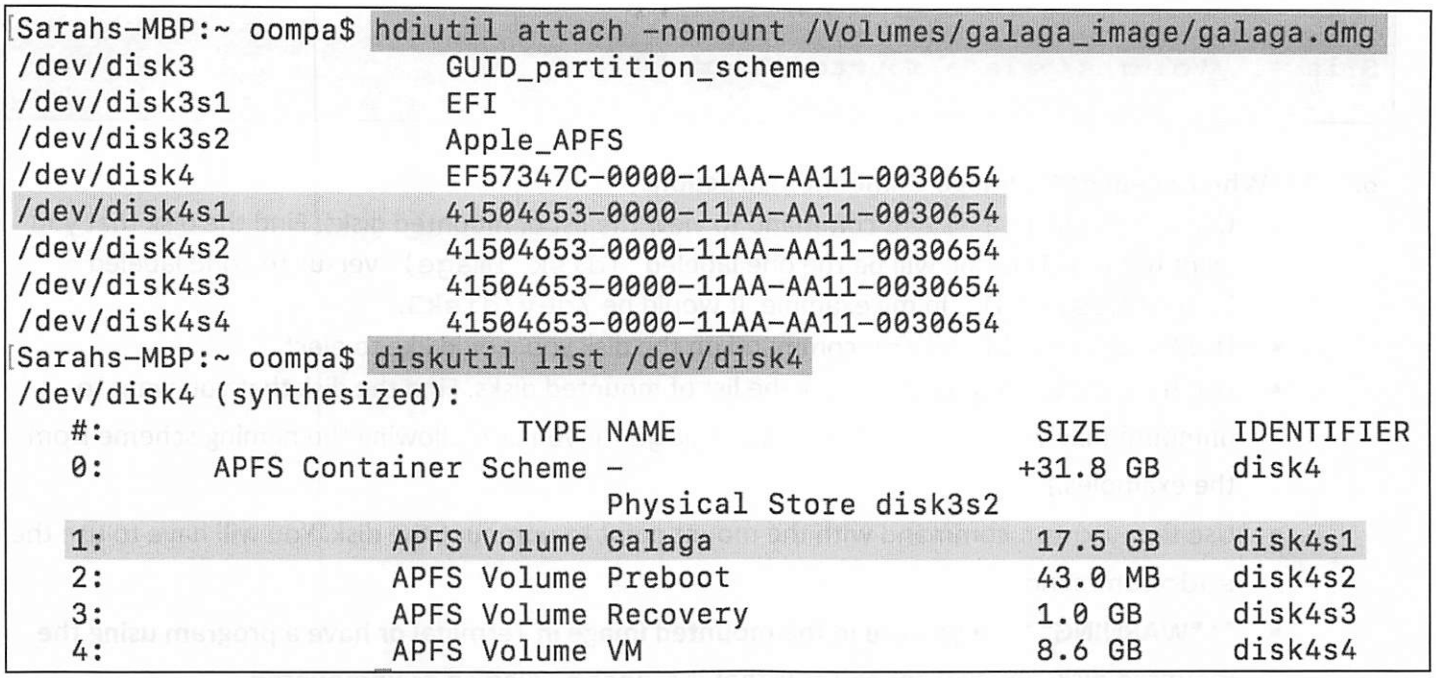
4. **Mount David Lightman's Mac Forensic Image (`galaga.E01`).**

   - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
   - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.

- Use xmount to mount the galaga.E01 image (where you have your image located; the example shows ~/FOR518/Lab_Images/Mac/) as a DMG file. This command requires you to use the sudo command, thus it may ask you for your administrator password when executed.
  - --in – Tells xmount what input file type to expect; our images are in a compressed EWF format.
  - --out – Tells xmount what output format you want; we want a DMG file so we can mount it in Finder.
  - Input File – Where the image file is located on your system.
  - Mount Point – Newly created mount point /Volumes/galaga_image specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Use the hdiutil command with the "attach" verb to make the newly created DMG volume available. Use the -nomount argument to suppress mounting (for now). The output from this command will display several /dev/disk# entries; use the appropriate disk device in the next command.
  - APFS disks will show many /dev/disk* options in the hdiutil output. The one we want to mount is the user's MacOS volume. We can use the command "diskutil list /dev/disk" on the synthesized disk to determine which is likely the user's MacOS volume. David Lightman's volume is named "Galaga," highlighted in the example below. We will use /dev/disk4s1 in the next command. **Be aware that yours may be mounted on a different disk number!**



```
[Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3               GUID_partition_scheme
/dev/disk3s1             EFI
/dev/disk3s2             Apple_APFS
/dev/disk4               EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1             41504653-0000-11AA-AA11-0030654
/dev/disk4s2             41504653-0000-11AA-AA11-0030654
/dev/disk4s3             41504653-0000-11AA-AA11-0030654
/dev/disk4s4             41504653-0000-11AA-AA11-0030654
[Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the `mount_apfs` command with the following parameters to mount the `/dev/disk#s#` (from the previous command) to the `/Volumes/galaga_mounted/` mount point. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
    - `-o` – Options:
        - `rdonly` – Mount in read-only mode.
        - `noexec` – Do not allow execution of binaries on mounted system.
        - `noowners` – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg


$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

5. **Sanity Check**
    - You can access this newly created mounted drive on `/Volumes/galaga_mounted/`, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
    - Use the `ls -l` command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "`dlightman`" in the `Users` directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

6. *****When Needed*****: Image Unmount Instructions**
   - Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`". In my example, it would be `/dev/disk3`.
   - Use the `diskutil eject` command on the disk you would like to eject.
   - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/` if you are following the naming scheme from the examples.)
   - Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command).
   - *****WARNING*****: If you are in the mounted image in Terminal or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

```
   DATETIME(ZOBJECT.ZCREATIONDATE + 978307200, 'UNIXEPOCH') AS "ENTRY
CREATION",
CASE ZOBJECT.ZSTARTDAYOFWEEK
WHEN "1" THEN "Sunday"
WHEN "2" THEN "Monday"
WHEN "3" THEN "Tuesday"
WHEN "4" THEN "Wednesday"
WHEN "5" THEN "Thursday"
WHEN "6" THEN "Friday"
WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
ZOBJECT.ZVALUESTRING AS "BUNDLE ID",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__ACTIVITYTYPE
AS "ACTIVITY TYPE",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__TITLE AS
"TITLE",
DATETIME(ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__EXPIR
ATIONDATE + 978307200, 'UNIXEPOCH') AS "EXPIRATION DATE",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__ITEMRELATEDCON
TENTURL AS "CONTENT URL",
ZOBJECT.ZSTREAMNAME AS "STREAM NAME",
ZOBJECT.Z_PK AS "ZOBJECT TABLE ID"
FROM
   ZOBJECT
   LEFT JOIN
      ZSTRUCTUREDMETADATA
      ON ZOBJECT.ZSTRUCTUREDMETADATA = ZSTRUCTUREDMETADATA.Z_PK
   LEFT JOIN
      ZSOURCE
      ON ZOBJECT.ZSOURCE = ZSOURCE.Z_PK
WHERE
   ZSTREAMNAME IS "/app/activity"
ORDER BY "ENTRY CREATION"
```

1. What "tour" was looked at in the Viator App (com.viator)?

   ____

2. What two locations were likely researched using Apple Maps (com.apple.Maps) on February 26, 2018?

   ____

6. Review the iOS CurrentPowerlog.PLSQL Database for Battery Level

- Get into a root shell.
- Navigate to the `BatteryLife` directory below in your mounted Physical/Logical iPhone image.
- Use `cp` to copy out the `CurrentPowerlog.PLSQL` database files to your FOR518 directory.
- Exit the root shell.
- Using `chown` and your username change the ownership of these files.

- Use the open command to view these files in DB Browser for SQLite.

```
$ sudo -s

# cd
/Volumes/davids_iphone/private/var/containers/Shared/SystemGroup/A6BC0
D08-2B73-431D-872B-71C6DDE3B162/Library/BatteryLife/

# cp CurrentPowerlog.PLSQL* ~/FOR518

# exit

$ sudo chown yourusername ~/FOR518/CurrentPowerlog.PLSQL*

$ open -a "DB Browser for SQLite" ~/FOR518/CurrentPowerlog.PLSQL
```

- Copy from the FOR518 notebook the following query and execute it on the `CurrentPowerlog.PLSQL` database.

```
SELECT
  DATETIME(TIMESTAMP, 'unixepoch') AS TIMESTAMP,
  LEVEL,
  ID AS "PLBATTERYAGENT_EVENTBACKWARD_BATTERYUI TABLE ID"
FROM
  PLBATTERYAGENT_EVENTBACKWARD_BATTERYUI
```

1. When was the battery level at its lowest point?

7. Review the iOS healthdb_secure.sqlite for Step Count

- Navigate to the `Health` directory below in your mounted Physical/Logical iPhone image.
- Use `cp` to copy out the `healthdb_secure.sqlite` database files to your `FOR518` directory.
- Use the open command to view these files in DB Browser for SQLite.

```
$ cd /Volumes/davids_iphone/private/var/mobile/Library/Health/

$ cp healthdb_secure.sqlite* ~/FOR518

$ open -a "DB Browser for SQLite" ~/FOR518/healthdb_secure.sqlite
```

- Copy from the FOR518 notebook the following query and execute it on the `healthdb_secure.sqlite` database.

```
SELECT
   DATETIME(SAMPLES.START_DATE + 978307200, 'unixepoch') AS "START
DATE",
   DATETIME(SAMPLES.END_DATE + 978307200, 'unixepoch') AS "END DATE",
   SAMPLES.DATA_TYPE AS "DATA TYPE",
   QUANTITY AS "STEPS",
   SAMPLES.DATA_ID AS "SAMPLES TABLE ID"
FROM
   SAMPLES
   LEFT OUTER JOIN
                QUANTITY_SAMPLES
                ON SAMPLES.DATA_ID = QUANTITY_SAMPLES.DATA_ID
   LEFT OUTER JOIN
                UNIT_STRINGS
                ON QUANTITY_SAMPLES.ORIGINAL_UNIT = UNIT_STRINGS.ROWID
   LEFT OUTER JOIN
                CORRELATIONS
                ON SAMPLES.DATA_ID = CORRELATIONS.OBJECT
   LEFT OUTER JOIN
                METADATA_VALUES
                ON METADATA_VALUES.OBJECT_ID = SAMPLES.DATA_ID
   LEFT OUTER JOIN
                METADATA_KEYS
                ON METADATA_KEYS.ROWID = METADATA_VALUES.KEY_ID
WHERE
   SAMPLES.DATA_TYPE = 7
   AND KEY IS NULL
   ORDER BY "START DATE"
```

1. What is the date range of the recorded steps?

   ____

2. Were any steps recorded on February 12, 2018?

   ____

8. iOS Cellular/Wi-Fi Locations

- On the Physical/Logical image, navigate to the /private/var/root/Library/Caches/locationd/ directory.
- Extract the cache_encryptedB.db files (all of them: *-shm and *-wal) to a "location" directory in your FOR518 directory.
- Open this database using SQLite Database Browser.
- Browse the contents of the following tables:
  - CellLocation
  - LteCellLocation
  - WifiLocation

9. iOS Routined/Significant Locations

- On the Physical/Logical image, navigate to the /private/var/mobile/Library/Caches/com.apple.routined/ directory.
- Extract all the database files (including: *-shm and *-wal) to the same "location" directory in your FOR518 directory.
- Open these database files using SQLite Database Browser.
- Browse the contents of the following tables:
    - Cloud.sqlite
        - ZRTLearnedPlaceMO
        - ZRTLearnedTransitionMO
        - ZRTLearnedVisitMO
    - Cache.sqlite
        - ZRTCLLocationMO
        - ZRTHintMO
    - Local.sqlite
        - ZRTLearnedLocationOfInterestMO
            - Specifically, ZPLACEMAPITEMGEOMAPTITEMHANDLE BLOB data
        - ZRTLearnedLocationOfInterestTransitionMO
        - ZRTLearnedLocationOfInterestVisitMO
        - ZRTPredictionItemMO
        - ZRTVehicleEventHistoryMO
        - ZRTVehicleEventMO

macOS

1. Review the macOS knowledgeC.db Database for Application Usage

- Navigate to the `Knowledge` directory below in your mounted macOS image.
- Use `cp` to copy out the `knowledgeC.db` database files to your `FOR518` directory.
- Use the open command to view these files in DB Browser for SQLite.
- Browse the contents in the tables.

```
$ cd /Volumes/galaga_mounted/private/var/db/CoreDuet/Knowledge/

$ cp knowledgeC.db* ~/FOR518

$ open ~/FOR518/knowledgeC.db
```

- Copy from the FOR518 notebook the following query and execute it on the `knowledgeC.db` database.

```
SELECT
datetime(ZOBJECT.ZCREATIONDATE+978307200,'UNIXEPOCH') as "ENTRY
CREATION",
ZOBJECT.ZVALUESTRING AS "BUNDLE ID",
CASE ZOBJECT.ZSTARTDAYOFWEEK
    WHEN "1" THEN "Sunday"
    WHEN "2" THEN "Monday"
    WHEN "3" THEN "Tuesday"
    WHEN "4" THEN "Wednesday"
    WHEN "5" THEN "Thursday"
    WHEN "6" THEN "Friday"
    WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
ZOBJECT.ZSECONDSFROMGMT/3600 AS "GMT OFFSET",
datetime(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH') as "START",
datetime(ZOBJECT.ZENDDATE+978307200,'UNIXEPOCH') as "END",
(ZOBJECT.ZENDDATE-ZOBJECT.ZSTARTDATE) as "USAGE IN SECONDS"
FROM ZOBJECT
WHERE ZSTREAMNAME IS "/app/inFocus"
ORDER BY "START"
```

1. How many time zones was this device likely in using the active records in this database?
   a. Two, GMT (0) and -5 (East Coast of US)
   b. Look at the 'GMT offset' column.

2. On what days was Google Chrome used (Bundle ID: com.google.Chrome)?
   a. 2018-02-25

    b. 2018-03-01
    c. 2018-03-03
    d. Add “and "bundle id" like '%chrome%'” to the end of the WHERE clause.

3. What was the most used “application” in a single session?
    a. com.apple.loginwindow – The laptop was sitting at the login screen for a good amount of time.
    b. Change the “ORDER BY” line from “START” to “USAGE IN SECONDS”

2. Review the macOS knowledgeC.db Database for Application Activities

- Copy from the FOR518 notebook the following query and execute it on the `knowledgeC.db` database.

```
SELECT
   DATETIME(ZOBJECT.ZCREATIONDATE + 978307200, 'UNIXEPOCH') AS "ENTRY
CREATION",
CASE ZOBJECT.ZSTARTDAYOFWEEK
WHEN "1" THEN "Sunday"
WHEN "2" THEN "Monday"
WHEN "3" THEN "Tuesday"
WHEN "4" THEN "Wednesday"
WHEN "5" THEN "Thursday"
WHEN "6" THEN "Friday"
WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
ZOBJECT.ZVALUESTRING AS "BUNDLE ID",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__ACTIVITYTYPE
AS "ACTIVITY TYPE",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__TITLE AS
"TITLE",
DATETIME(ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__EXPIR
ATIONDATE + 978307200, 'UNIXEPOCH') AS "EXPIRATION DATE",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__ITEMRELATEDCON
TENTURL AS "CONTENT URL",
ZOBJECT.ZSTREAMNAME AS "STREAM NAME",
ZOBJECT.Z_PK AS "ZOBJECT TABLE ID"
FROM
   ZOBJECT
   LEFT JOIN
      ZSTRUCTUREDMETADATA
      ON ZOBJECT.ZSTRUCTUREDMETADATA = ZSTRUCTUREDMETADATA.Z_PK
   LEFT JOIN
      ZSOURCE
      ON ZOBJECT.ZSOURCE = ZSOURCE.Z_PK
WHERE
   ZSTREAMNAME IS "/app/activity"
ORDER BY "ENTRY CREATION"
```

1. What two applications have activities recorded in this database?

a. com.apple.Maps = Apple Maps
b. com.apple.Mail = Apple Mail
c. Look at the ‘BUNDLE ID’ column.

3. Review the macOS knowledgeC.db Database for Safari Browsing

- Copy from the FOR518 notebook the following query and execute it on the `knowledgeC.db` database.

```
SELECT
  DATETIME(ZOBJECT.ZCREATIONDATE + 978307200, 'UNIXEPOCH') AS "ENTRY
CREATION",
CASE ZOBJECT.ZSTARTDAYOFWEEK
WHEN "1" THEN "Sunday"
WHEN "2" THEN "Monday"
WHEN "3" THEN "Tuesday"
WHEN "4" THEN "Wednesday"
WHEN "5" THEN "Thursday"
WHEN "6" THEN "Friday"
WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
ZOBJECT.ZVALUESTRING AS "URL",
ZSOURCE.ZBUNDLEID AS "BUNDLE ID",
ZOBJECT.ZSTREAMNAME AS "STREAM NAME",
ZOBJECT.Z_PK AS "ZOBJECT TABLE ID"
FROM
  ZOBJECT
  LEFT JOIN
     ZSTRUCTUREDMETADATA
     ON ZOBJECT.ZSTRUCTUREDMETADATA = ZSTRUCTUREDMETADATA.Z_PK
  LEFT JOIN
     ZSOURCE
     ON ZOBJECT.ZSOURCE = ZSOURCE.Z_PK
WHERE
  ZSTREAMNAME IS "/safari/history"
 ORDER BY "ENTRY CREATION"
```

1. What was searched for in Safari on February 27th?
   a. A Google search was performed for “`ars technica`”
   b. “`https://www.google.com/search?client=safari&rls=en&q=ars+technica&ie=UTF-8&oe=UTF-8`”

iOS

4. Review the iOS knowledgeC.db Database for Application Usage

- Navigate to the `Knowledge` directory below in your mounted Physical/Logical iPhone image.
- Use `cp` to copy out the `knowledgeC.db` database files to your FOR518 directory.
  i. Note: These will overwrite your macOS database files from the previous part of the lab.

- Use the open command to view these files in DB Browser for SQLite.
- Browse the contents in the tables.

```
$ cd
/Volumes/davids_iphone/private/var/mobile/Library/CoreDuet/Knowledge/

$ cp knowledgeC.db* ~/FOR518

$ open ~/FOR518/knowledgeC.db
```

- Copy from the FOR518 notebook the following query and execute it on the `knowledgeC.db` database.

```
SELECT
datetime(ZOBJECT.ZCREATIONDATE+978307200,'UNIXEPOCH') as "ENTRY
CREATION",
ZOBJECT.ZVALUESTRING AS "BUNDLE ID",
CASE ZOBJECT.ZSTARTDAYOFWEEK
    WHEN "1" THEN "Sunday"
    WHEN "2" THEN "Monday"
    WHEN "3" THEN "Tuesday"
    WHEN "4" THEN "Wednesday"
    WHEN "5" THEN "Thursday"
    WHEN "6" THEN "Friday"
    WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
ZOBJECT.ZSECONDSFROMGMT/3600 AS "GMT OFFSET",
datetime(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH') as "START",
datetime(ZOBJECT.ZENDDATE+978307200,'UNIXEPOCH') as "END",
(ZOBJECT.ZENDDATE-ZOBJECT.ZSTARTDATE) as "USAGE IN SECONDS"
FROM ZOBJECT
WHERE ZSTREAMNAME IS "/app/inFocus"
ORDER BY "START"
```

1. When was WhatsApp used the longest?
   a. 2018-02-25 20:56:56, for 68 seconds.
   b. Add “AND "BUNDLE ID" LIKE '%WHATSAPP%'” to the WHERE clause to search for the WhatsApp bundle ID (net.whatsapp.WhatsApp).

2. In what time zone was the Starbucks app used?
   a. GMT
   b. Add “AND "BUNDLE ID" LIKE '%starbucks%'” to the WHERE clause to search for the Starbucks bundle ID (com.starbucks.mystarbucks).

3. What was the most used “application” in a single session?
   a. com.apple.mobileslideshow – The “Photos” App, 940 seconds
   b. Change the “ORDER BY” line from “START” to “USAGE IN SECONDS”

5. Review the iOS knowledgeC.db Database for Application Activities

- Copy from the FOR518 notebook the following query and execute it on the `knowledgeC.db` database.

```
SELECT
   DATETIME(ZOBJECT.ZCREATIONDATE + 978307200, 'UNIXEPOCH') AS "ENTRY
CREATION",
CASE ZOBJECT.ZSTARTDAYOFWEEK
WHEN "1" THEN "Sunday"
WHEN "2" THEN "Monday"
WHEN "3" THEN "Tuesday"
WHEN "4" THEN "Wednesday"
WHEN "5" THEN "Thursday"
WHEN "6" THEN "Friday"
WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
ZOBJECT.ZVALUESTRING AS "BUNDLE ID",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__ACTIVITYTYPE
AS "ACTIVITY TYPE",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__TITLE AS
"TITLE",
DATETIME(ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__EXPIR
ATIONDATE + 978307200, 'UNIXEPOCH') AS "EXPIRATION DATE",
ZSTRUCTUREDMETADATA.Z_DKAPPLICATIONACTIVITYMETADATAKEY__ITEMRELATEDCON
TENTURL AS "CONTENT URL",
ZOBJECT.ZSTREAMNAME AS "STREAM NAME",
ZOBJECT.Z_PK AS "ZOBJECT TABLE ID"
FROM
   ZOBJECT
   LEFT JOIN
      ZSTRUCTUREDMETADATA
      ON ZOBJECT.ZSTRUCTUREDMETADATA = ZSTRUCTUREDMETADATA.Z_PK
   LEFT JOIN
      ZSOURCE
      ON ZOBJECT.ZSOURCE = ZSOURCE.Z_PK
WHERE
   ZSTREAMNAME IS "/app/activity"
ORDER BY "ENTRY CREATION"
```

1. What “tour” was looked at in the Viator App (com.viator)?
   a. Jack the Ripper Tour with 'Ripper-Vision' in London
   b. Look for entries with the bundle ID of com.viator. The activity type is ‘com.viator.viatorApp.product’

2. What two locations were likely researched using Apple Maps (com.apple.Maps) on February 26, 2018?
   a. Central Library in Arlington
   b. Gaijin Ramen Shop in Arlington

c.

6. Review the iOS CurrentPowerlog.PLSQL Database for Battery Level

- Get into a root shell.
- Navigate to the `BatteryLife` directory below in your mounted Physical/Logical iPhone image.
- Use `cp` to copy out the `CurrentPowerlog.PLSQL` database files to your FOR518 directory.
- Exit the root shell.
- Using `chown` and your username change the ownership of these files.
- Use the `open` command to view these files in DB Browser for SQLite.

```
$ sudo -s

# cd
/Volumes/davids_iphone/private/var/containers/Shared/SystemGroup/A6BC0
D08-2B73-431D-872B-71C6DDE3B162/Library/BatteryLife/

# cp CurrentPowerlog.PLSQL* ~/FOR518

# exit

$ sudo chown yourusername ~/FOR518/CurrentPowerlog.PLSQL*

$ open -a "DB Browser for SQLite" ~/FOR518/CurrentPowerlog.PLSQL
```

- Copy from the FOR518 notebook the following query and execute it on the `CurrentPowerlog.PLSQL` database.

```
SELECT
  DATETIME(TIMESTAMP, 'unixepoch') AS TIMESTAMP,
  LEVEL,
  ID AS "PLBATTERYAGENT_EVENTBACKWARD_BATTERYUI TABLE ID"
FROM
  PLBATTERYAGENT_EVENTBACKWARD_BATTERYUI
```

1. When was the battery level at its lowest point?
    a. 2018-02-26 01:41:15, for level 59.
    b. Add “ORDER BY ‘Level’” at the end of the query.

7. Review the iOS healthdb_secure.sqlite for Step Count

- Navigate to the `Health` directory below in your mounted Physical/Logical iPhone image.
- Use `cp` to copy out the `healthdb_secure.sqlite` database files to your FOR518 directory.
- Use the open command to view these files in DB Browser for SQLite.

```
$ cd /Volumes/davids_iphone/private/var/mobile/Library/Health/
$ cp healthdb_secure.sqlite* ~/FOR518

$ open -a "DB Browser for SQLite" ~/FOR518/healthdb_secure.sqlite
```

- Copy from the FOR518 notebook the following query and execute it on the `healthdb_secure.sqlite` database.

```
SELECT
   DATETIME(SAMPLES.START_DATE + 978307200, 'unixepoch') AS "START
DATE",
   DATETIME(SAMPLES.END_DATE + 978307200, 'unixepoch') AS "END DATE",
   SAMPLES.DATA_TYPE AS "DATA TYPE",
   QUANTITY AS "STEPS",
   SAMPLES.DATA_ID AS "SAMPLES TABLE ID"
FROM
   SAMPLES
   LEFT OUTER JOIN
               QUANTITY_SAMPLES
               ON SAMPLES.DATA_ID = QUANTITY_SAMPLES.DATA_ID
   LEFT OUTER JOIN
               UNIT_STRINGS
               ON QUANTITY_SAMPLES.ORIGINAL_UNIT = UNIT_STRINGS.ROWID
   LEFT OUTER JOIN
               CORRELATIONS
               ON SAMPLES.DATA_ID = CORRELATIONS.OBJECT
   LEFT OUTER JOIN
               METADATA_VALUES
               ON METADATA_VALUES.OBJECT_ID = SAMPLES.DATA_ID
   LEFT OUTER JOIN
               METADATA_KEYS
               ON METADATA_KEYS.ROWID = METADATA_VALUES.KEY_ID
WHERE
   SAMPLES.DATA_TYPE = 7
   AND KEY IS NULL
   ORDER BY "START DATE"
```

1. What is the date range of the recorded steps?
   a. 2017-11-12 – 2018-03-03

2. Were any steps recorded on February 12, 2018?
   a. None; on the day before and after yes.
   b. The watch was not being worn and likely the phone was not being used, thus not recording steps.

8. iOS Cellular/Wi-Fi Locations

- On the Physical/Logical image, navigate to the /private/var/root/Library/Caches/locationd/ directory.
- Extract the cache_encryptedB.db files (all of them: *-shm and *-wal) to a "location" directory in your FOR518 directory.

- Open this database using SQLite Database Browser.
- Browse the contents of the following tables:
  - CellLocation
  - LteCellLocation
  - WifiLocation

9. iOS Routined/Significant Locations

- On the Physical/Logical image, navigate to the /private/var/mobile/Library/Caches/com.apple.routined/ directory.
- Extract all the database files (including: *-shm and *-wal) to the same "location" directory in your FOR518 directory.
- Open these database files using SQLite Database Browser.
- Browse the contents of the following tables:
  - Cloud.sqlite
    - ZRTLearnedPlaceMO
    - ZRTLearnedTransitionMO
    - ZRTLearnedVisitMO
  - Cache.sqlite
    - ZRTCLLocationMO
    - ZRTHintMO
  - Local.sqlite
    - ZRTLearnedLocationOfInterestMO
      - Specifically, ZPLACEMAPITEMGEOMAPTITEMHANDLE BLOB data
    - ZRTLearnedLocationOfInterestTransitionMO
    - ZRTLearnedLocationOfInterestVisitMO
    - ZRTPredictionItemMO
    - ZRTVehicleEventHistoryMO
    - ZRTVehicleEventMO

Lab: Key Takeaways

- **Review some of the pattern-of-life artifacts in databases from macOS and iOS.**

Lab 5.2: Document Versions

Objectives

- **Get familiar with Document Version storage data and databases.**

Lab Preparation

(Note: Some of this might already be accomplished via earlier labs, but this is the state that we hope your system is in prior to the start of this lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this lab.)

1. **Software Preparation**: The following tools will be used in this lab:
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from /Applications/Utilities/

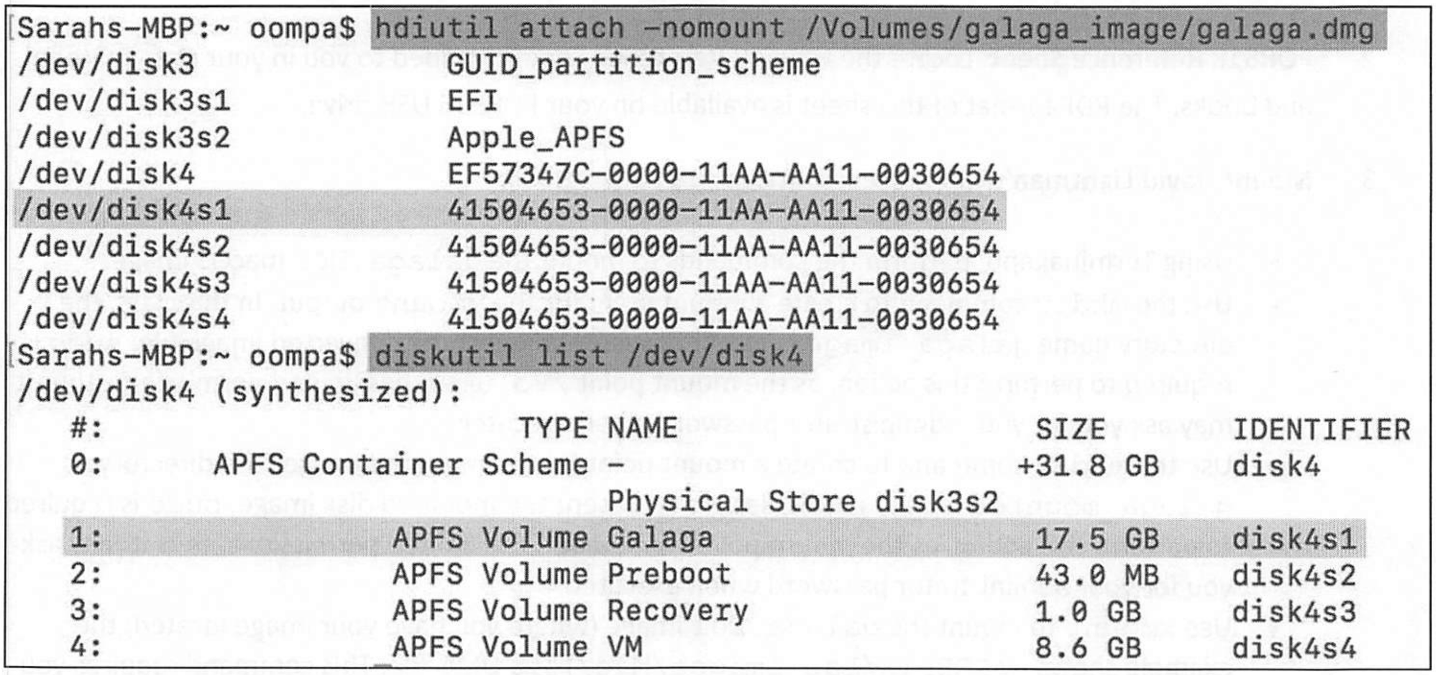
2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Mount David Lightman's Mac Forensic Image (`galaga.E01`).**

   - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
   - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.
     - o `--in` – Tells `xmount` what input file type to expect; our images are in a compressed EWF format.
     - o `--out` – Tells `xmount` what output format you want; we want a DMG file so we can mount it in Finder.
     - o `Input File` – Where the image file is located on your system.
     - o `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/
$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Use the hdiutil command with the "attach" verb to make the newly created DMG volume available. Use the -nomount argument to suppress mounting (for now). The output from this command will display several /dev/disk# entries; use the appropriate disk device in the next command.
  - APFS disks will show many /dev/disk* options in the hdiutil output. The one we want to mount is the user's macOS volume. We can use the command "diskutil list /dev/disk4" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "Galaga," highlighted in the example below. We will use /dev/disk4s1 in the next command. **Be aware that yours may be mounted on a different disk number!**



```
[Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3              GUID_partition_scheme
/dev/disk3s1            EFI
/dev/disk3s2            Apple_APFS
/dev/disk4              EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1            41504653-0000-11AA-AA11-0030654
/dev/disk4s2            41504653-0000-11AA-AA11-0030654
/dev/disk4s3            41504653-0000-11AA-AA11-0030654
/dev/disk4s4            41504653-0000-11AA-AA11-0030654
[Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the mount_apfs command with the following parameters to mount the /dev/disk#s# (from the previous command) to the /Volumes/galaga_mounted/ mount point. This command requires you to use the sudo command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
  - -o – Options:
    - rdonly – Mount in read-only mode.
    - noexec – Do not allow execution of binaries on mounted system.
    - noowners – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

4. **Sanity Check**
   - You can access this newly created mounted drive on `/Volumes/galaga_mounted/`, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
   - Use the `ls -l` command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "`dlightman`" in the `Users` directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

5. *****When Needed***: Image Unmount Instructions**
   - Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`". In my example, it would be `/dev/disk3`.
   - Use the `diskutil eject` command on the disk you would like to eject.
   - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/` if are you following the naming scheme from the examples).
   - Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command.
   - *****WARNING***: If you are in the mounted image in Terminal or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

Lab: Questions

1. Versions: Review the Document Versions Directory

- Use the `sudo -s` command to get a privileged shell.
- Use the `cd` command to explore the system's Document Versions directory.
- Use the `ls -la` command to view the contents of this directory.
- Use the `ls -laR` command to recursively view the contents of the `PerUID` directory. Note the contents of these directories.
- Use the `cd` command to explore the `PerUID/501/c/com.apple.documentVersions` directory.
- Use the `ls -litr` command to view the contents of this directory reverse sorted by time. It will also print the inode numbers for these files. Note the contents of this directory.

```
$ sudo -s

# cd /Volumes/galaga_mounted/.DocumentRevisions-V100

# ls -la

# ls -laR PerUID

# cd PerUID/501/c/com.apple.documentVersions/

# ls -litr
```

1. Did this file grow or shrink in size over time?

   ____

2. What are the inode numbers for these files?

   ____

   ____

- Use `xattr -xl` to review the contents of the extended attributes of these files. Feel free to run it all at once (`xattr -xl *`) or on a per- file basis (`xattr -xl <file>`).

3. What was the original filename and the final filename? Also take note of the filenames of the first and last file generation (generally, just the first section of the GUID and the last few characters of the filename will work).

   ____

2. Versions: Review the Document Versions Database

- Use the `cd` command to explore the systems' Document Versions database directory.

- Copy out the `db.sqlite` files to your FOR518 directory and open these files in SQLite Database Browser.

```
# cd /Volumes/galaga_mounted/.DocumentRevisions-V100/db-V1/

# cp db.sqlite* ~/FOR518/
```

- Review the `generations` table.
- Find the "`generations_name`" column. That looks familiar—you should see the files we just saw.
    1. Generation IDs = 3, 11, 13, 18
    2. Also note the `generation_size` column.
- Review the `files` table.
- Find our "`games.rtf`" file; review the information in this tuple.

3. Versions: Review the Versions Chunk Store Database

- Use the `cd` command to explore the systems' Versions Chunk Store database directory.
- Use the `ls -la` command to view the contents of this directory.
- Copy the ChunkStoreDatabase to your FOR518 directory and open the database with SQLite Database Browser.

```
# cd /Volumes/galaga_mounted/.DocumentRevisions-V100/.cs/

# ls -la

# cp ChunkStoreDatabase* ~/FOR518/
```

- Review the `CSStorageChunkListTable`; note the items for `clt_rowid` and `clt_inode` listed in the table below:
- Review the `CSChunkTable` table.
    1. Note the number in the column `ft_rowid = 5`; this is the ChunkStorage file we will look at.

1. Fill in the table below with the offset and data length for items 3 and 13 in the `CSChunkTable`. The other generations do not appear to store metadata in this table, perhaps because they are iCloud documents.

clt_rowid	clt_inode	offset	dataLen
3	1422797		
11	1529625	N/A	N/A
13	1531497		
18	1532140	N/A	N/A

- Keeping the ChunkStoreDatabase open for reference, change directories to the Chunk Storage file – “5”.
- Using the open command, open this file in your favorite hex editor (0xED [shown] or Hex Fiend).
- Exit the root shell.

```
# cd /Volumes/galaga_mounted/.DocumentRevisions-
V100/.cs/ChunkStorage/0/0/0/

# open -a 0xED 5

# exit
```

- If you get an error while using the "open" command, please copy ("cp") the file to your ~/FOR518 directory, change ownership of the file ("chown"), and open it directly in any hex editor (0xED [shown] or Hex Fiend).
- Using the offsets and data lengths above, find the two generations of the games.rtf file.
- Use the Chunk Storage Record Format below to review the contents of these chunks.

Chunk Storage Record Format	
4 bytes	Size of chunk record
21 bytes	Chunk ID
Remaining	Chunk Contents

2. What changed between these two document versions?

____

____

1. **Versions: Review the Document Versions Directory**
   - Use the `sudo -s` command to get a privileged shell.
   - Use the `cd` command to explore the system's Document Versions directory.
   - Use the `ls -la` command to view the contents of this directory.
   - Use the `ls -laR` command to recursively view the contents of the `PerUID` directory. Note the contents of these directories.
   - Use the `cd` command to explore the `PerUID/501/c/com.apple.documentVersions` directory.
   - Use the `ls -litr` command to view the contents of this directory reverse sorted by time. It will also print the inode numbers for these files. Note the contents of this directory.

```
$ sudo -s

# cd /Volumes/galaga_mounted/.DocumentRevisions-V100

# ls -la

# ls -laR PerUID

# cd PerUID/501/c/com.apple.documentVersions/

# ls -litr
```

1. Did this file grow or shrink in size over time?
   a. It grew, 178b -> 421b -> 502b -> 546b
   b. Look at the file size column.

2. What are the inode numbers for these files?
   a. 1529625 = com~apple~TextEdit_93AC3DEB-4D6A-4AB9-8293-EAAA824334E5_6.rtf
   b. 1422797 = F54F8520-A9BB-4E7F-9C23-0B41C11DC720.rtf
   c. 1532140 = com~apple~TextEdit_93AC3DEB-4D6A-4AB9-8293-EAAA824334E5_w.rtf
   d. 1531497 = CE93CFC8-A53F-40C9-8A22-A932BBB9DFF5.rtf

- Use `xattr -xl` to review the contents of the extended attributes of these files. Feel free to run it all at once (`xattr -xl *`) or on a per-file basis (`xattr -xl <file>`).

3. What was the original filename and the final filename? Also take note of the filenames of the first and last file generation (generally, just the first section of the GUID and the last few characters of the filename will work).
   a. Original = Untitled.rtf
      i. (com~apple~TextEdit_93AC3DEB-4D6A-4AB9-8293-EAAA824334E5_6.rtf)
   b. Final = games.rtf
      i. (CE93CFC8-A53F-40C9-8A22-A932BBB9DFF5.rtf)

c. com.apple.genstore.origposixname

2. Versions: Review the Document Versions Database

- Use the `cd` command to explore the systems' Document Versions database directory.
- Copy out the `db.sqlite` files to your FOR518 directory and open these files in SQLite Database Browser.

```
# cd /Volumes/galaga_mounted/.DocumentRevisions-V100/db-V1/

# cp db.sqlite* ~/FOR518/
```

- Review the `generations` table.
- Find the "`generations_name`" column. That looks familiar—you should see the files we just saw.
  - i. Generation IDs = 3, 11, 13, 18
  - ii. Also note the `generation_size` column.
- Review the `files` table.
- Find our "`games.rtf`" file; review the information in this tuple.

3. Versions: Review the Versions Chunk Store Database

- Use the `cd` command to explore the systems' Versions Chunk Store database directory.
- Use the `ls -la` command to view the contents of this directory.
- Copy the ChunkStoreDatabase to your FOR518 directory and open the database with SQLite Database Browser.

```
# cd /Volumes/galaga_mounted/.DocumentRevisions-V100/.cs/

# ls -la

# cp ChunkStoreDatabase* ~/FOR518/
```

- Review the `CSStorageChunkListTable`; note the items for `clt_rowid` and `clt_inode` listed in the table below:
- Review the `CSChunkTable` table.
  - i. Note the number in the column `ft_rowid` = 5; this is the ChunkStorage file we will look at.

1. Fill in the table below with the offset and data length for items 3 and 13 in the `CSChunkTable`. The other generations do not appear to store metadata in this table, perhaps because they are iCloud documents.

clt_rowid	clt_inode	offset	dataLen
3	1422797	3193764	446

11	1529625	N/A	N/A
13	1531497	3417208	527
18	1532140	N/A	N/A

- Keeping the `ChunkStoreDatabase` open for reference, change directories to the Chunk Storage file – "`5`".
- Using the open command, open this file in your favorite hex editor (0xED [shown] or Hex Fiend).
- Exit the root shell.

```
# cd /Volumes/galaga_mounted/.DocumentRevisions-
V100/.cs/ChunkStorage/0/0/0/

# open -a 0xED 5

# exit
```

- If you get an error while using the "`open`" command, please copy ("`cp`") the file to your ~/FOR518 directory, change ownership of the file ("`chown`"), and open it directly in any hex editor (0xED [shown] or Hex Fiend).
- Using the offsets and data lengths above, find the two generations of the `games.rtf` file.
- Use the Chunk Storage Record Format below to review the contents of these chunks.

Chunk Storage Record Format	
4 bytes	Size of chunk record
21 bytes	Chunk ID
Remaining	Chunk Contents

2. What changed between these two document versions?
    a. Added two games (Centipede and Frogger)
    b. Added a link for SEGA games and retropie.
    c. This can be seen in the RTF files. To make it easier, you can extract the RTF files starting with the curly bracket "{" and ending with the opposite curly bracket "}", saving these chunks into two separate files and opening them.

Lab: Key Takeaways

- **Understand how Chunk Storage is implemented in Document Versions.**

This page intentionally left blank.

Lab 5.3: Malware and Live Response

Objectives

- Review the contents of security-related files and databases.
- Get familiar with the macOS command-line utilities.
- Gather and analyze live response data.

Lab Preparation

(Note: Some of this might already be accomplished via earlier labs, but this is the state that we hope your system is in prior to the start of this lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this lab.)

1. **Software Preparation**: The following tools will be used in this lab:
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from /Applications/Utilities/

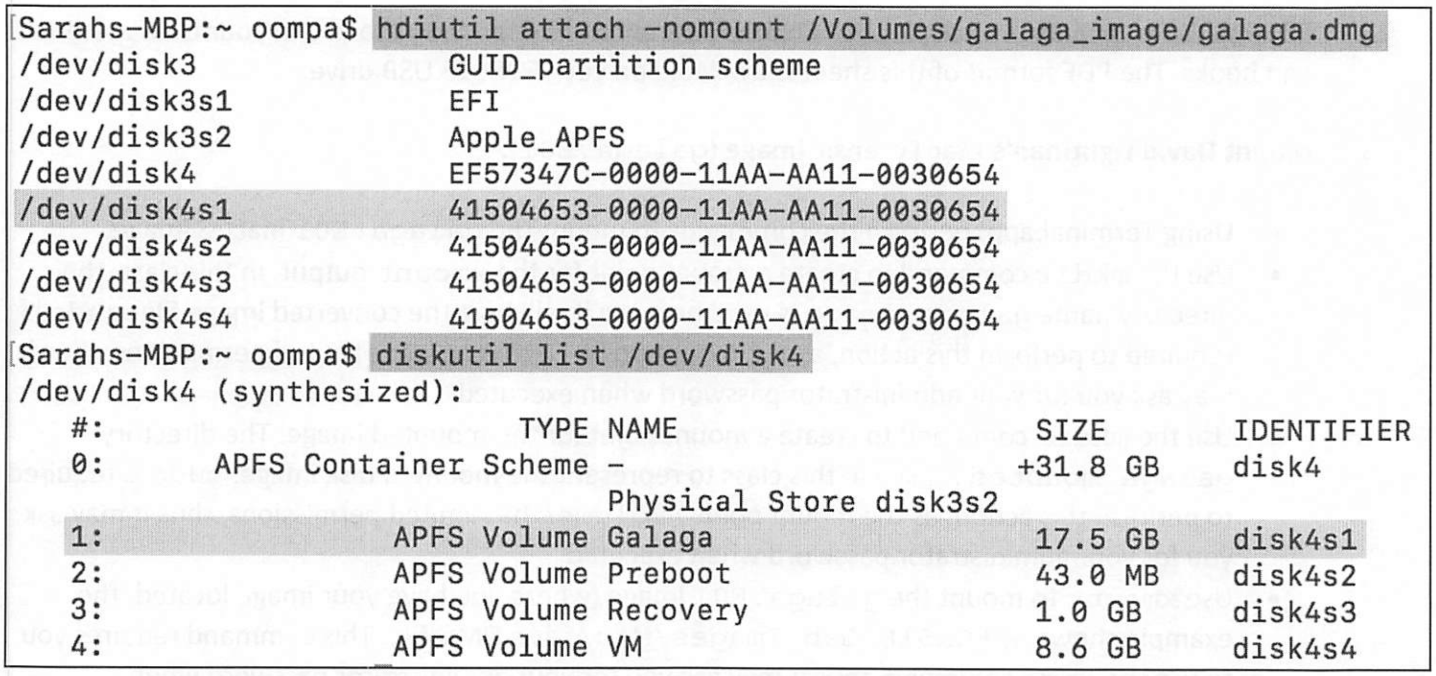
2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Mount David Lightman's Mac Forensic Image (`galaga.E01`).**

   - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
   - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.
     - `--in` – Tells `xmount` what input file type to expect; our images are in a compressed EWF format.
     - `--out` – Tells `xmount` what output format you want; we want a DMG file so we can mount it in Finder.
     - `Input File` – Where the image file is located on your system.
     - `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Use the `hdiutil` command with the "`attach`" verb to make the newly created DMG volume available. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display several `/dev/disk#` entries; use the appropriate disk device in the next command.
  - APFS disks will show many `/dev/disk*` options in the `hdiutil` output. The one we want to mount is the user's macOS volume. We can use the command "`diskutil list /dev/disk4`" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "`Galaga`," highlighted in the example below. We will use `/dev/disk4s1` in the next command. **Be aware that yours may be mounted on a different disk number!**



```
Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3              GUID_partition_scheme
/dev/disk3s1            EFI
/dev/disk3s2            Apple_APFS
/dev/disk4              EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1            41504653-0000-11AA-AA11-0030654
/dev/disk4s2            41504653-0000-11AA-AA11-0030654
/dev/disk4s3            41504653-0000-11AA-AA11-0030654
/dev/disk4s4            41504653-0000-11AA-AA11-0030654
Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the `mount_apfs` command with the following parameters to mount the `/dev/disk#s#` (from the previous command) to the `/Volumes/galaga_mounted/` mount point. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
  - `-o` – Options:
    - `rdonly` – Mount in read-only mode.
    - `noexec` – Do not allow execution of binaries on mounted system.
    - `noowners` – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg


$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

4. **Sanity Check**
   - You can access this newly created mounted drive on `/Volumes/galaga_mounted/`, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
   - Use the `ls -l` command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "`dlightman`" in the `Users` directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

5. *****When Needed***: Image Unmount Instructions**
   - Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`". In my example, it would be `/dev/disk3`.
   - Use the `diskutil eject` command on the disk you would like to eject.
   - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/` if you are following the naming scheme from the examples).
   - Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command.
   - *****WARNING***: If you are in the mounted image in Terminal or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

Lab: Questions

1. Review the File Quarantine Database

- Copy and review the David's `com.apple.LaunchServices.QuarantineEventsV2` database using SQLite Database Browser.

```
$ cp /Volumes/galaga_mounted/Users/dlightman/Library/Preferences/
com.apple.LaunchServices.QuarantineEventsV2 ~/FOR518/
```

1. Who sent files via the "sharing" process?

2. What action might have caused this sharing?

3. How many items were downloaded with Safari as it pertains to this database?

- Run the following `xattr` command on David's Downloads directory. Were there really only two downloads? It's not a perfect system. This is why it is good to look at multiple sources for the same information.

```
$ xattr -xlp com.apple.quarantine
/Volumes/galaga_mounted/Users/dlightman/Downloads/*
```

2. Review the XProtect Signatures

- Navigate to the XProtect files in CoreServices.
- Use the `ls -la` command to view the contents of this directory.
- Use the `less` command to take a peek at the YARA rules. (Use "`q`" to quit `less`.)
- Use `plutil -p` to review the contents of `XProtect.meta.plist`.
- Use `open` to view the contents of `XProtect.plist`. Take a moment to review it.

```
$ cd
/Volumes/galaga_mounted/System/Library/CoreServices/XProtect.bundle/Cont
ents/Resources/

$ ls -la

$ less XProtect.yara

$ plutil -p XProtect.meta.plist
$ open XProtect.plist
```

3. Gather the System Information of <u>Your</u> Analysis System

- Run and review the following commands as if you were responding to your analysis system.
  - i. Run the `date` command.
    1. What time zone is your system set to?
    2. Is your time current?
  - ii. Run the `hostname` command.
  - iii. Run the `uname -a` command.
    1. What is your kernel version?
  - iv. Run the `sw_vers` command.
    1. What macOS version and build are you running?

```
$ date

$ hostname

$ uname -a

$ sw_vers
```

4. What Are the Active Network Connections of <u>Your</u> System?

- Run and review the following commands as if you were responding to your analysis system.
  - i. Run the `netstat -an` command.
    1. **Note**: The option "`-f inet`" or "`-f inet6`" may be used to limit the output to just IPv4 or IPv6 addresses.
    2. Try the same command without the "`-n`".
    3. Try performing a "`whois`" on some of these IP addresses.
    4. **Note**: The option "`-b`" shows the number of bytes transferred/received for each IP address.

```
$ netstat -an
```

5. What Are the Active Network Connections of <u>Your</u> System, by Process?

- Run and review the following commands as if you were responding to your analysis system.
  - i. Run the `lsof -i` command.

```
$ lsof -i
```

6. **Review the Network Configuration Data of <u>Your</u> System**
   - Run and review the following commands as if you were responding to your analysis system.
   - Run the `ifconfig` command.
     i. What is the IP of your system?

```
$ ifconfig
```

7. **What Are the Open Files in <u>Your</u> System?**
   - Run and review the following commands as if you were responding to your analysis system.
   - Run the `lsof` command.
   - Review the Command, Process ID, User, and Name fields.
     i. Note: Pipe the output to the less command "`lsof | less`" for easier viewing. (Use "`q`" to exit `less`.)

```
$ lsof
```

8. **What Users Are Logged on to <u>Your</u> System?**
   - Run and review the following commands as if you were responding to your analysis system.
   - Run the `who -a` and `w` commands.
   - Run the `last` command to get a historical overview of logins, system shutdowns, and reboots.

```
$ who -a

$ w

$ last
```

9. **What Are the Running Processes on <u>Your</u> System?**
   - Run and review the following commands as if you were responding to your analysis system.
   - Run the `ps aux` command.
     i. **Note**: The "`ps -ef`" command gives a different output that you may find preferable.

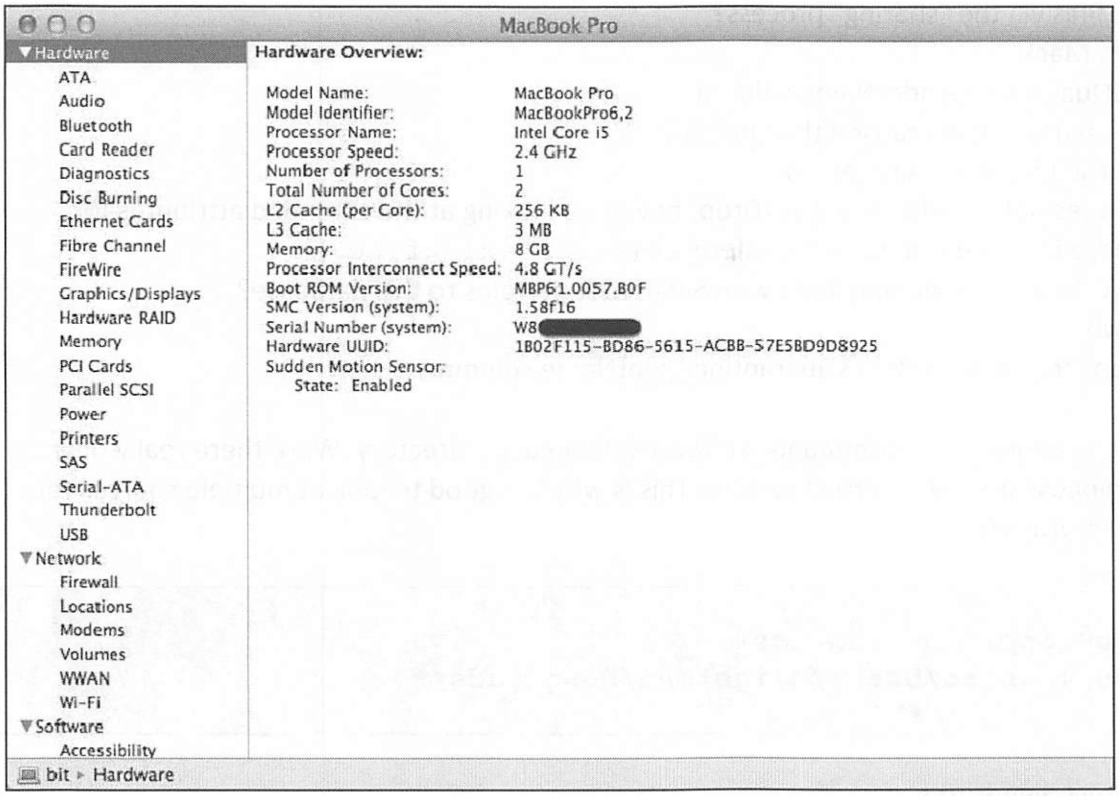
```
$ ps aux
```

10. **Extract <u>Your</u> System Information Using the `system_profiler` Command-Line Utility**
    - Run the `system_profiler` command; output to a file named `system-profiler-data.spx` in your FOR518 directory.

```
$ system_profiler -xml -detailLevel full > ~/FOR518/system-profiler-
data.spx

$ open ~/FOR518/system-profiler-data.spx
```

11. Review the Output of the `system_profiler` Command Using System Information.app

- Open the file `system-profiler-data.spx` file you just created in the System Information.app. This application is located in /Applications/Utilities/.
- Use the `open` command to open the file you have just created.
- Review the various data components.



```
Hardware Overview:

  Model Name:                      MacBook Pro
  Model Identifier:                MacBookPro6,2
  Processor Name:                  Intel Core i5
  Processor Speed:                 2.4 GHz
  Number of Processors:            1
  Total Number of Cores:           2
  L2 Cache (per Core):             256 KB
  L3 Cache:                        3 MB
  Memory:                          8 GB
  Processor Interconnect Speed:    4.8 GT/s
  Boot ROM Version:                MBP61.0057.B0F
  SMC Version (system):            1.58f16
  Serial Number (system):          W8[redacted]
  Hardware UUID:                   1B02F115-BD86-5615-ACBB-57E5BD9D8925
  Sudden Motion Sensor:
      State:  Enabled
```

1. Review the File Quarantine Database

- Copy and review the David's `com.apple.LaunchServices.QuarantineEventsV2` database using SQLite Database Browser.

```
$ cp /Volumes/galaga_mounted/Users/dlightman/Library/Preferences/
com.apple.LaunchServices.QuarantineEventsV2 ~/FOR518/
```

1. Who sent files via the "sharing" process?
   - Jen Mack
   - LSQuarantineSenderName column
2. What action might have caused this sharing?
   - These files were AirDrop'ed.
   - It does not specifically say AirDrop; however, looking at the extended attributes for these files, we can make the inference (`xattr -xl <file>`).
3. How many items were downloaded with Safari as it pertains to this database?
   - Two
   - Look for Safari in the LSQuarantineAgentName column.

- Run the following `xattr` command on David's Downloads directory. Were there really only two downloads? It's not a perfect system. This is why it is good to look at multiple sources for the same information.

```
$ xattr -xlp com.apple.quarantine
/Volumes/galaga_mounted/Users/dlightman/Downloads/*
```

2. Review the XProtect Signatures

- Navigate to the XProtect files in CoreServices.
- Use the `ls -la` command to view the contents of this directory.
- Use the `less` command to take a peek at the YARA rules. (Use "`q`" to quit `less`.)
- Use `plutil -p` to review the contents of `XProtect.meta.plist`.
- Use `open` to view the contents of `XProtect.plist`. Take a moment to review it.

```
$ cd
/Volumes/galaga_mounted/System/Library/CoreServices/XProtect.bundle/Cont
ents/Resources/

$ ls -la

$ less XProtect.yara
$ plutil -p XProtect.meta.plist

$ open XProtect.plist
```

3. **Gather the System Information of Your Analysis System**
   - Run and review the following commands as if you were responding to your analysis system.
     - i. Run the `date` command.
       1. What time zone is your system set to?
       2. Is your time current?
     - ii. Run the `hostname` command.
     - iii. Run the `uname -a` command.
       1. What is your kernel version?
     - iv. Run the `sw_vers` command.
       1. What macOS version and build are you running?

```
$ date

$ hostname

$ uname -a

$ sw_vers
```

4. **What Are the Active Network Connections of Your System?**
   - Run and review the following commands as if you were responding to your analysis system.
     - i. Run the `netstat -an` command.
       1. **Note**: The option "`-f inet`" or "`-f inet6`" may be used to limit the output to just IPv4 or IPv6 addresses.
       2. Try the same command without the "`-n`".
       3. Try performing a "`whois`" on some of these IP addresses.
       4. **Note**: The option "`-b`" shows the number of bytes transferred/received for each IP address.

```
$ netstat -an
```

5. **What Are the Active Network Connections of Your System, by Process?**
   - Run and review the following commands as if you were responding to your analysis system.
     - i. Run the `lsof -i` command.

```
$ lsof -i
```

6. Review the Network Configuration Data of Your System

- Run and review the following commands as if you were responding to your analysis system.
- Run the `ifconfig` command.
    i. What is the IP of your system?

```
$ ifconfig
```

7. What Are the Open Files in Your System?

- Run and review the following commands as if you were responding to your analysis system.
- Run the `lsof` command.
- Review the Command, Process ID, User, and Name fields.
    i. Note: Pipe the output to the less command "`lsof | less`" for easier viewing. (Use "`q`" to exit `less`.)

```
$ lsof
```

8. What Users Are Logged on to Your System?

- Run and review the following commands as if you were responding to your analysis system.
- Run the `who -a` and `w` commands.
- Run the `last` command to get a historical overview of logins, system shutdowns, and reboots.

```
$ who -a

$ w

$ last
```

9. What Are the Running Processes on Your System?

- Run and review the following commands as if you were responding to your analysis system.
- Run the `ps aux` command.
    i. **Note**: The "`ps -ef`" command gives a different output that you may find preferable.

```
$ ps aux
```

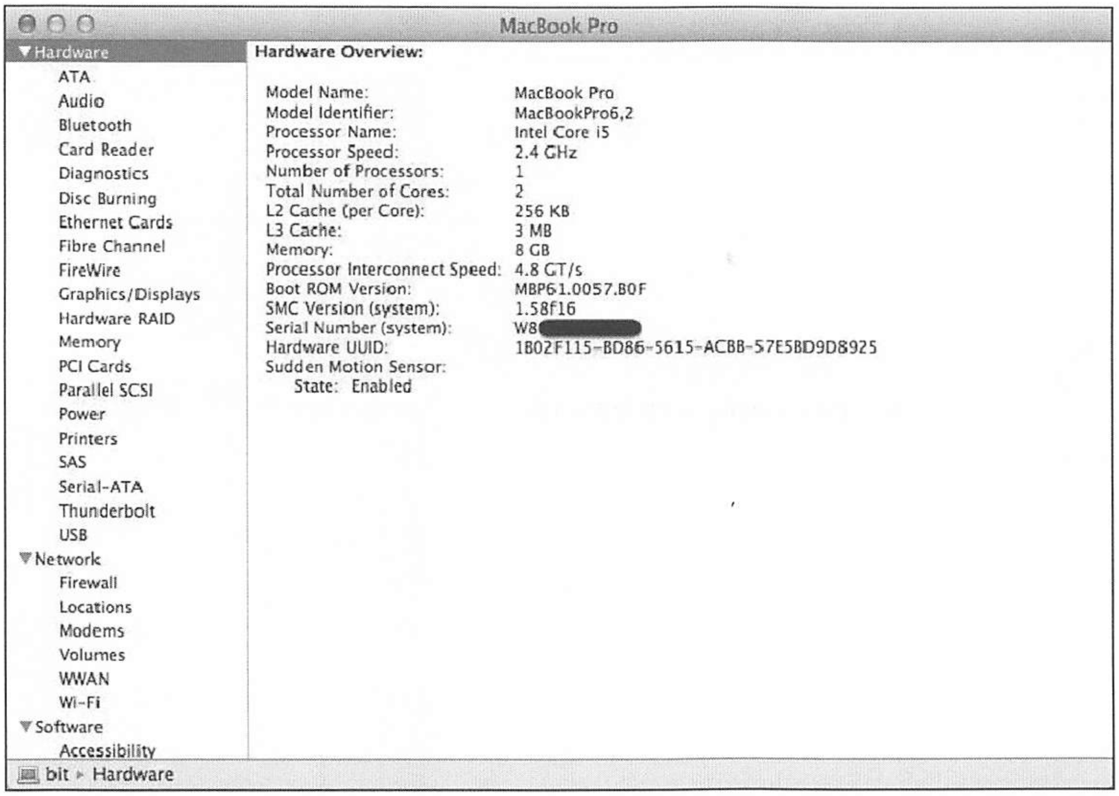
10. Extract Your System Information Using the `system_profiler` Command-Line Utility

- Run the `system_profiler` command; output to a file named `system-profiler-data.spx` in your FOR518 directory.

```
$ system_profiler -xml -detailLevel full > ~/FOR518/system-profiler-data.spx

$ open ~/FOR518/system-profiler-data.spx
```

11. Review the Output of the `system_profiler` Command Using System Information.app

- Open the file `system-profiler-data.spx` file you just created in the System Information.app. This application is located in /Applications/Utilities/.
- Use the `open` command to open the file you have just created.
- Review the various data components.



Hardware Overview:

```
  Model Name:                     MacBook Pro
  Model Identifier:               MacBookPro6,2
  Processor Name:                 Intel Core i5
  Processor Speed:                2.4 GHz
  Number of Processors:           1
  Total Number of Cores:          2
  L2 Cache (per Core):            256 KB
  L3 Cache:                       3 MB
  Memory:                         8 GB
  Processor Interconnect Speed:   4.8 GT/s
  Boot ROM Version:               MBP61.0057.B0F
  SMC Version (system):           1.58f16
  Serial Number (system):         W8[redacted]
  Hardware UUID:                  1B02F115-BD86-5615-ACBB-57E5BD9D8925
  Sudden Motion Sensor:
      State:   Enabled
```

Lab: Key Takeaways

- **Review some of the security-related files and databases.**
- **Get comfortable with some Mac OS X command-line utilities.**
- **Many of the same commands you may have used with other systems may be different on Mac OS X, such as the `ps  aux` command.**

This page intentionally left blank.

Lab 5.4: Memory Analysis, Password Cracking, and Encrypted Containers

Objectives

- Understand the capability of Volatility, how it is used, and what you can expect to extract from Mac memory.
- Create a dictionary file using the memory image.
- Use John the Ripper to crack (or attempt to crack) passwords for a keychain file, an encrypted DMG, and a user account.

Lab Preparation

(Note: Some of this might already be accomplished via earlier labs, but this is the state that we hope your system is in prior to the start of this lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this lab.)

1. **Software Preparation**: The following tools will be used in this lab:
   - Terminal.app
      i. You will be using the native OS X Terminal application for this lab.
      ii. Locate and open the Terminal.app from /Applications/Utilities/
   - Volatility
      i. Ensure you have Volatility installed via Homebrew from `https://brew.sh/` (see Lab 0).
   - John the Ripper
      i. Ensure you have John the Ripper (john-jumbo) installed via Homebrew (see Lab 0).
   - Keychain Access.app
      i. You will be opening David Lightman's keychain file.
      ii. Locate and open the Keychain Access.app from /Applications/Utilities/
2. **Lab File Preparation**: Locate the `Lab Files/Lab 5.4 - Memory Analysis, Password Cracking & Encrypted Containers` directory.
3. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.
4. **Memory Image**: Copy the `galaga_memory.raw` memory image to your local host system to make some of the memory commands run faster. Remember where you put this file; you need to point to that file and path in this lab. While the icon for this file may look like an archive to "The Unarchiver", this file **DOES NOT** need to be unarchived. This file should have been unarchived from the 7-Zip archive on Day 1 from the original file on USB FOR518-A: `galaga_memory.raw.7z`
   - This memory dump was created with OSXPMem. The default output for OSXPMem is AFF format, which is compressed but not compatible with Volatility. We will also be creating a dictionary file to use with John the Ripper, so we need the RAW format. The following commands were used to convert this memory image from AFF format to RAW format. The first command is used to determine which data stream to output (/dev/pmem). The second command is used for the format conversion.
      i. `-V` – View AFF Metadata

ii. `-e` – Export a data stream (/dev/pmem)

iii. `-o` – Output file (RAW memory image)

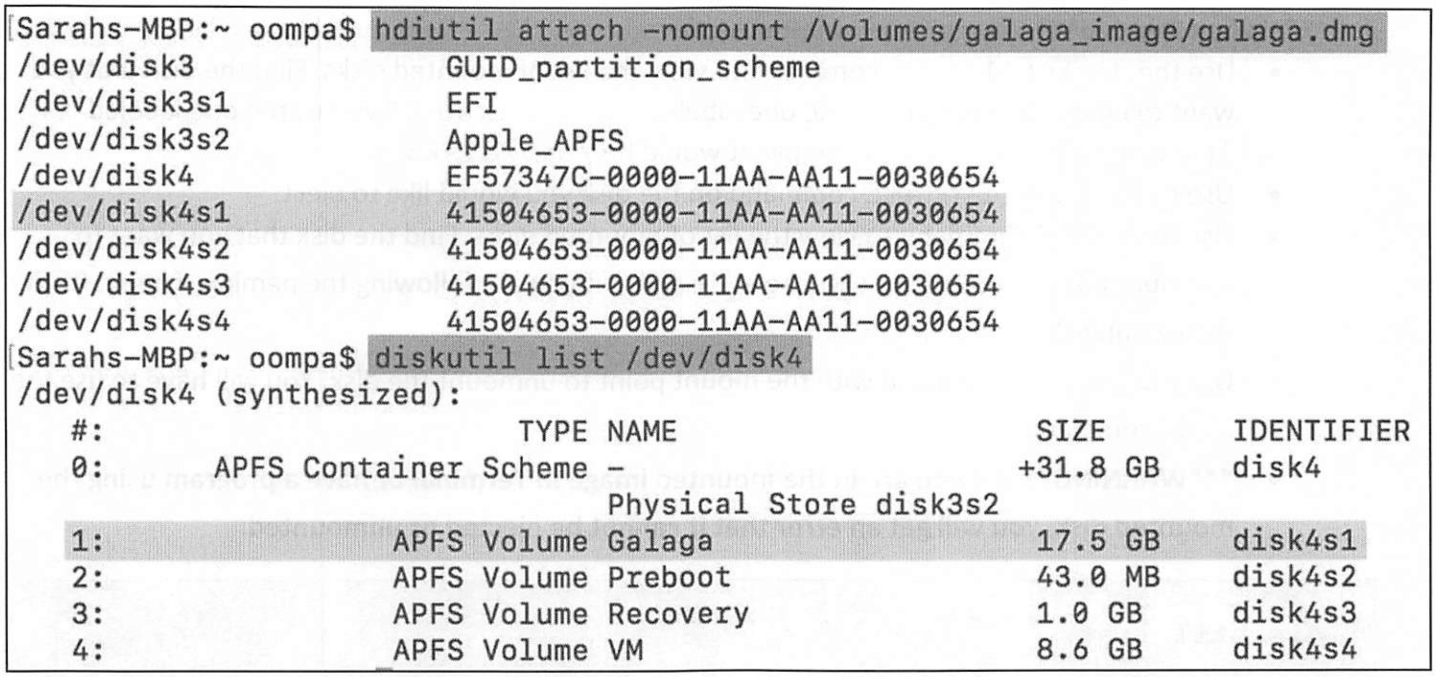
- `./osxpmem -V galaga_memory.aff`
- `./osxpmem -e /dev/pmem -o galaga_memory.raw galaga_memory.aff`

5. **Mount David Lightman's Mac Forensic Image (`galaga.E01`).**

- Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
- Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
- Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
- Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.
  - `--in` – Tells `xmount` what input file type to expect; our images are in a compressed EWF format.
  - `--out` – Tells `xmount` what output format you want; we want a DMG file so we can mount it in Finder.
  - `Input File` – Where the image file is located on your system.
  - `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Use the `hdiutil` command with the "`attach`" verb to make the newly created DMG volume available. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display several `/dev/disk#` entries; use the appropriate disk device in the next command.
  - APFS disks will show many `/dev/disk*` options in the `hdiutil` output. The one we want to mount is the user's macOS volume. We can use the command "`diskutil list /dev/disk4`" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "`Galaga`," highlighted in the example below. We will use `/dev/disk4s1` in the next command. **Be aware that yours may be mounted on a different disk number!**



```
[Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3          GUID_partition_scheme
/dev/disk3s1        EFI
/dev/disk3s2        Apple_APFS
/dev/disk4          EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1        41504653-0000-11AA-AA11-0030654
/dev/disk4s2        41504653-0000-11AA-AA11-0030654
/dev/disk4s3        41504653-0000-11AA-AA11-0030654
/dev/disk4s4        41504653-0000-11AA-AA11-0030654
[Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the mount_apfs command with the following parameters to mount the /dev/disk#s# (from the previous command) to the /Volumes/galaga_mounted/ mount point. This command requires you to use the sudo command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
    - -o – Options:
        - rdonly – Mount in read-only mode.
        - noexec – Do not allow execution of binaries on mounted system.
        - noowners – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

6. **Sanity Check**
    - You can access this newly created mounted drive on /Volumes/galaga_mounted/, thus all command-line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
    - Use the ls -l command to view the contents in the Terminal to (hopefully) view the macOS directory structure. You should see an account for "dlightman" in the Users directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

7. ***When Needed***: Image Unmount Instructions
    - Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`". In my example, it would be `/dev/disk3`.
    - Use the `diskutil eject` command on the disk you would like to eject.
    - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/` if you are following the naming scheme from the examples).
    - Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command.
        - *****WARNING***: If you are in the mounted image in Terminal or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

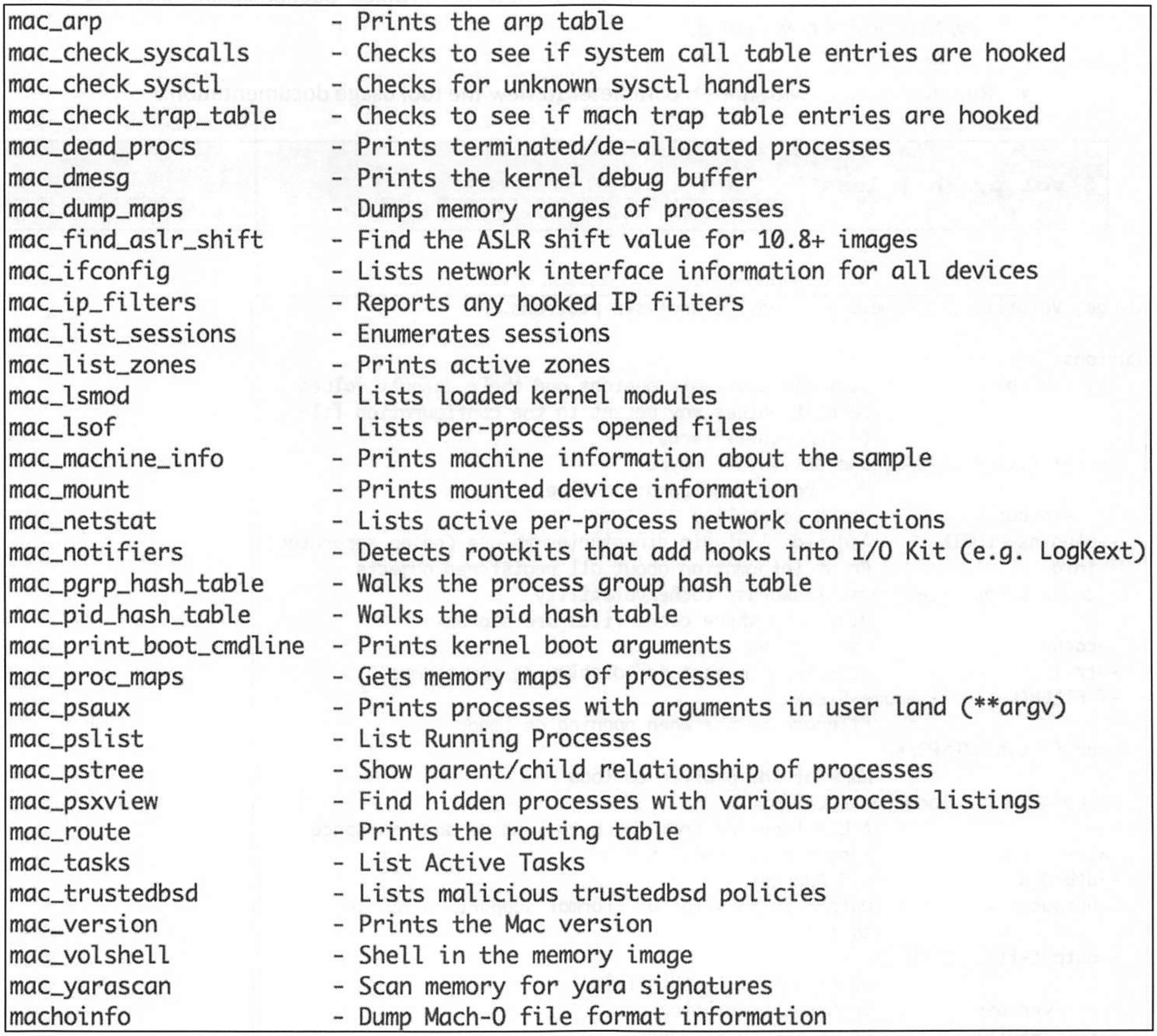
Lab: Questions

1. Volatility: Documentation

- Run the `vol.py` with the `--info` parameter to view the tool documentation.

```
$ vol.py --info | less
```

- Review the "Plugins" Section. Note the plugins named with the "`mac_*`". We will be using some of these in this lab.



```
mac_arp                  - Prints the arp table
mac_check_syscalls       - Checks to see if system call table entries are hooked
mac_check_sysctl         - Checks for unknown sysctl handlers
mac_check_trap_table     - Checks to see if mach trap table entries are hooked
mac_dead_procs           - Prints terminated/de-allocated processes
mac_dmesg                - Prints the kernel debug buffer
mac_dump_maps            - Dumps memory ranges of processes
mac_find_aslr_shift      - Find the ASLR shift value for 10.8+ images
mac_ifconfig             - Lists network interface information for all devices
mac_ip_filters           - Reports any hooked IP filters
mac_list_sessions        - Enumerates sessions
mac_list_zones           - Prints active zones
mac_lsmod                - Lists loaded kernel modules
mac_lsof                 - Lists per-process opened files
mac_machine_info         - Prints machine information about the sample
mac_mount                - Prints mounted device information
mac_netstat              - Lists active per-process network connections
mac_notifiers            - Detects rootkits that add hooks into I/O Kit (e.g. LogKext)
mac_pgrp_hash_table      - Walks the process group hash table
mac_pid_hash_table       - Walks the pid hash table
mac_print_boot_cmdline   - Prints kernel boot arguments
mac_proc_maps            - Gets memory maps of processes
mac_psaux                - Prints processes with arguments in user land (**argv)
mac_pslist               - List Running Processes
mac_pstree               - Show parent/child relationship of processes
mac_psxview              - Find hidden processes with various process listings
mac_route                - Prints the routing table
mac_tasks                - List Active Tasks
mac_trustedbsd           - Lists malicious trustedbsd policies
mac_version              - Prints the Mac version
mac_volshell             - Shell in the memory image
mac_yarascan             - Scan memory for yara signatures
machoinfo                - Dump Mach-O file format information
```

- Review the "`Profiles`" Section. By default, there are no Mac profiles loaded with Volatility. We will need to install them.

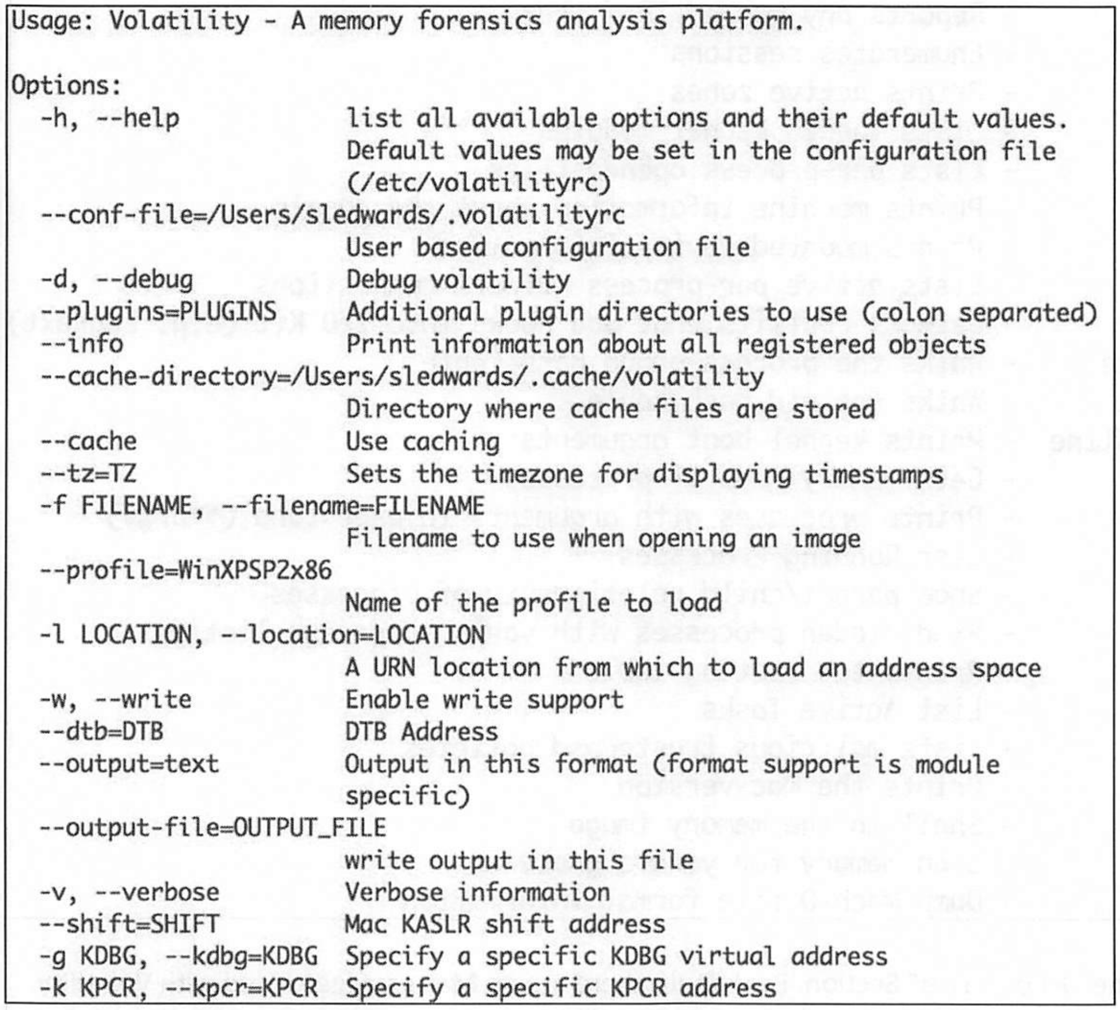
- In your Lab 5.4 directory, please copy the profile ZIP (*HighSierra_10.13.1_17B35a.zip*) archive to the Volatility profile directory. This is the default installation area for items installed with Homebrew. **Your directory path may be slightly different**; use tab completion to ensure you have the correct path. You do not need to unzip them.

```
$ cp HighSierra_10.13.1_17B35a.zip
/usr/local/Cellar/volatility/<#.#_#>/libexec/lib/python2.7/site-
packages/volatility/plugins/overlays/mac/
```

- Re-run the `vol.py -info` command. Review the “Profiles” Section again. Take note of the new Mac profile now loaded.

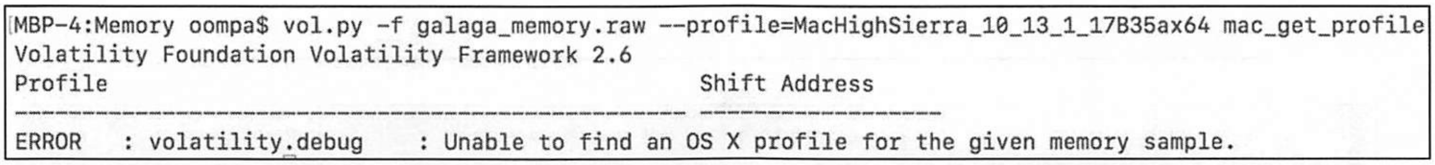
- Run the `vol.py` with the `-h` parameter to view the tool usage documentation.

```
$ vol.py -h | less
```



```
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/Users/sledwards/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --info                Print information about all registered objects
  --cache-directory=/Users/sledwards/.cache/volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the timezone for displaying timestamps
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --output=text         Output in this format (format support is module
                        specific)
  --output-file=OUTPUT_FILE
                        write output in this file
  -v, --verbose         Verbose information
  --shift=SHIFT         Mac KASLR shift address
  -g KDBG, --kdbg=KDBG  Specify a specific KDBG virtual address
  -k KPCR, --kpcr=KPCR  Specify a specific KPCR address
```

- Run the `vol.py` with the `-f` (filename) parameter on the `galaga_memory.raw` image. We will use the `mac_get_profile` plugin first to determine which profile we need to use.
  - i. **This will error out—unfortunately, it does not recognize the profile required**.
  - ii. Trial and error with various 10.13.1 profiles shows that the installed profile needed is the "`MacHighSierra_10_13_1_17B35ax64`" profile that was just installed.



```
MBP-4:Memory oompa$ vol.py -f galaga_memory.raw --profile=MacHighSierra_10_13_1_17B35ax64 mac_get_profile
Volatility Foundation Volatility Framework 2.6
Profile                                                Shift Address
------------------------------------------------------ ------------------
ERROR   : volatility.debug    : Unable to find an OS X profile for the given memory sample.
```

```
$ vol.py -f galaga_memory.raw mac_get_profile
```

2. Volatility Analysis: System Information

- Run the `vol.py` with the `mac_version` parameter to view the system kernel information. *****NOTE: These Volatility command lines can be long. These commands are meant to be executed as a single line. (They appear as two lines in this lab.)**

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_version
```

1. What kernel version does this system use?

____

- Run the `vol.py` with the `mac_mount` parameter to view mounted volumes on this system.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_mount
```

2. What external disk is mounted on this system?

____

3. What format is `/dev/disk5s2`?

____

3. Volatility: Network Information

- Run the `vol.py` with the `mac_ifconfig` parameter to view the system network configuration.

```
$ vol.py -f galaga_memory.raw
```

```
--profile=MacHighSierra_10_13_1_17B35ax64 mac_ifconfig
```

1. What IPv4 did this system have at the time of acquisition?

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_netstat
```

2. Whose email servers are the Mail applications calling out to (use `whois`)?

4. Volatility: Processes

- Run the `vol.py` with the `mac_pslist` parameter to view system processes by walking the process list.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_pslist
```

1. What are the process names for PID 0 and 1?

2. Find the "keylogger" process—who owns this process?

3. What is the process ID for the "`logKextDaemon`" process?

- Run the `vol.py` with the `mac_pstree` parameter to view system processes in a tree formation.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_pstree
```

4. What process was performed using the `sudo` command?

- Run the `vol.py` with the `mac_lsof` parameter to view the open file handles for each process.
- The output from this command can be quite verbose; you can choose to redirect the output to a file for easier analysis.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_lsof >
~/FOR518/mac_lsof.txt

$ open ~/FOR518/mac_lsof.txt
```

5. What file (that would be of investigative value) does "`logKextDaemon`" have open?

____

5. Volatility: Kernel Extensions

- Run the `vol.py` with the `mac_lsmod` parameter to view kernel extensions.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_lsmod
```

1. What two kernel extensions are loaded that are not from Apple?

____

6. Create a Dictionary File for Password Cracking

- Use the `strings` command below to create a dictionary file from the memory image to crack some passwords.
    - i. The `-n` flag specifies the minimum string length of 8 characters.
    - ii. This output will be piped to the "sort -u" command to filter out only unique strings.
    - iii. This output will then be piped to two `awk` commands.
        1. The first `awk` command will filter for strings that contain only lower and uppercase characters (no special characters).
        2. The second `awk` command will filter for string lengths of less than 12 characters.
    - iv. Finally, the output of these commands will be outputted to a file named `galaga_dictionary.txt` in your FOR518 directory.
    - v. This should output a dictionary that is of reasonable size (416k) for relatively quick brute force password cracking. This should take just over a minute or so.

```
$ strings -n 8 galaga_memory.raw | sort -u | awk '$0 ~ /^[a-zA-
Z]{1,}$/'| awk 'length($0)<12' > ~/FOR518/galaga_dictionary.txt
```

7. Crack a Keychain File with John the Ripper

- Extract the `login.keychain-db` file for `dlightman` to your FOR518 directory.
   i. `/Volumes/galaga_mounted/Users/dlightman/Library/Keychains/`
- Extract the password hash from the `login.keychain-db` file using the `keychain2john.py` Python script. **Your directory path may be slightly different**; use tab completion to ensure you have the correct path.
- Ensure you extracted the password hash by using `cat` to view the contents of the newly created file.
- Using the `john` utility and the created dictionary file, crack the keychain password.
   i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password.
   ii. Press the "Enter" key a few times to see the status.
   iii. This should not take too long (~40 seconds depending on your Mac hardware—example time was performed on a 2016 MacBook Pro, 2.9Ghz, Core i5)
   iv. Once you get the password, use Control+C to quit John.
- Open the `login.keychain` file using "Keychain Access.app"; you may need to switch back and forth from your login keychain (the bolded one) to `dlightman`'s keychain file (the unbolded one) to get it to read correctly.
- **Note**: If you would like to re-run this hash crack again, remove the `john.pot` file from your home directory using the command "`rm ~/.john/john.pot`".

```
$ cp login.keychain-db ~/FOR518/

$ python /usr/local/Cellar/john-
jumbo/<#.#.#>/share/john/keychain2john.py ~/FOR518/login.keychain-db >
~/FOR518/dlightman_keychain.txt

$ cat ~/FOR518/dlightman_keychain.txt

$ john --wordlist=~/FOR518/galaga_dictionary.txt
~/FOR518/dlightman_keychain.txt
```

1. What kind of password hash is detected in the keychain?

2. Approximately how many passwords per second is john brute forcing (sixth column from the left with the numbers labeled p/s)?

3. What is `dlightman`'s keychain password?

4. Open the `login.keychain-db` file using Keychain Access.app. What is the password for the iOS Backup?

8. Crack a Login Password with John the Ripper

- Using the already extracted (Lab 2.3) user plist for `dlightman`, extract the password hash from the `dlightman.plist` file using the `mac2john.py` Python script.
    1. `/Volumes/galaga_mounted/private/var/db/dslocal/nodes/Default/users/dlightman.plist`
- Ensure you extracted the password hash by using `cat` to view the contents of the newly created file.
- Using the `john` utility and the created dictionary file, crack the login password.
    1. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password.
    2. Press the "Enter" key a few times to see the status.
    3. **YOU WILL NOT CONTINUE TO CRACK THIS PASSWORD; use Control+C to quit John.**
        1. This took approximately 90 minutes on a 2016 MacBook Pro, 2.9Ghz, Core i5.

```
$ python /usr/local/Cellar/john-jumbo/<#.#.#>/share/john/mac2john.py
~/FOR518/dlightman.plist > ~/FOR518/dlightman_loginpassword.txt

$ cat ~/FOR518/dlightman_loginpassword.txt

$ john --wordlist=~/FOR518/galaga_dictionary.txt
~/FOR518/dlightman_loginpassword.txt
```

1. What kind of password hash is detected in the login password?

    ____

2. Approximately how many passwords per second is john brute forcing (sixth column from the left with the numbers labeled p/s)?

    ____

3. If you had to take a guess, what is the user's login password?

    ____

9. Crack a DMG Password with John the Ripper

- Extract the `k12.dmg` file from `dlightman`'s system to your FOR518 directory.
- Extract the password hash from the `k12.dmg` file using the `dmg2john.py` Python script.
- Ensure you extracted the password hash by using `cat` to view the contents of the newly created file.
- Using the `john` utility and the created dictionary file, crack the DMG password.
    1. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password.
    2. Press the "Enter" key a few times to see the status.
    3. **YOU WILL NOT CONTINUE TO CRACK THIS PASSWORD; use Control+C to quit John.**
- Password cracking using a dictionary file is not perfect, as it turns out the password is not in the dictionary file we created, because it is six characters in length. Even if we filtered for passwords that were shorter, our dictionary file would have strings that included the password text but not as a string itself. Sometimes you win, sometimes you do not. A better dictionary file could have been created, but that takes a bit more work to do.

```
$ cp /Volumes/galaga_mounted/Users/dlightman/Documents/Stuff/kl2.dmg
~/FOR518/

$ python /usr/local/Cellar/john-jumbo/<#.#.#>/share/john/dmg2john.py
~/FOR518/kl2.dmg > ~/FOR518/dlightman_dmg.txt

$ cat ~/FOR518/dlightman_dmg.txt

$ john --wordlist=~/FOR518/galaga_dictionary.txt
~/FOR518/dlightman_dmg.txt
```

1. What kind of password hash is detected in the encrypted DMG file?

   ____

2. Approximately how many passwords per second is john brute forcing (sixth column from the left with the numbers labeled p/s)?

   ____

3. Using the keychain password, what is the password for this DMG file?

   ____

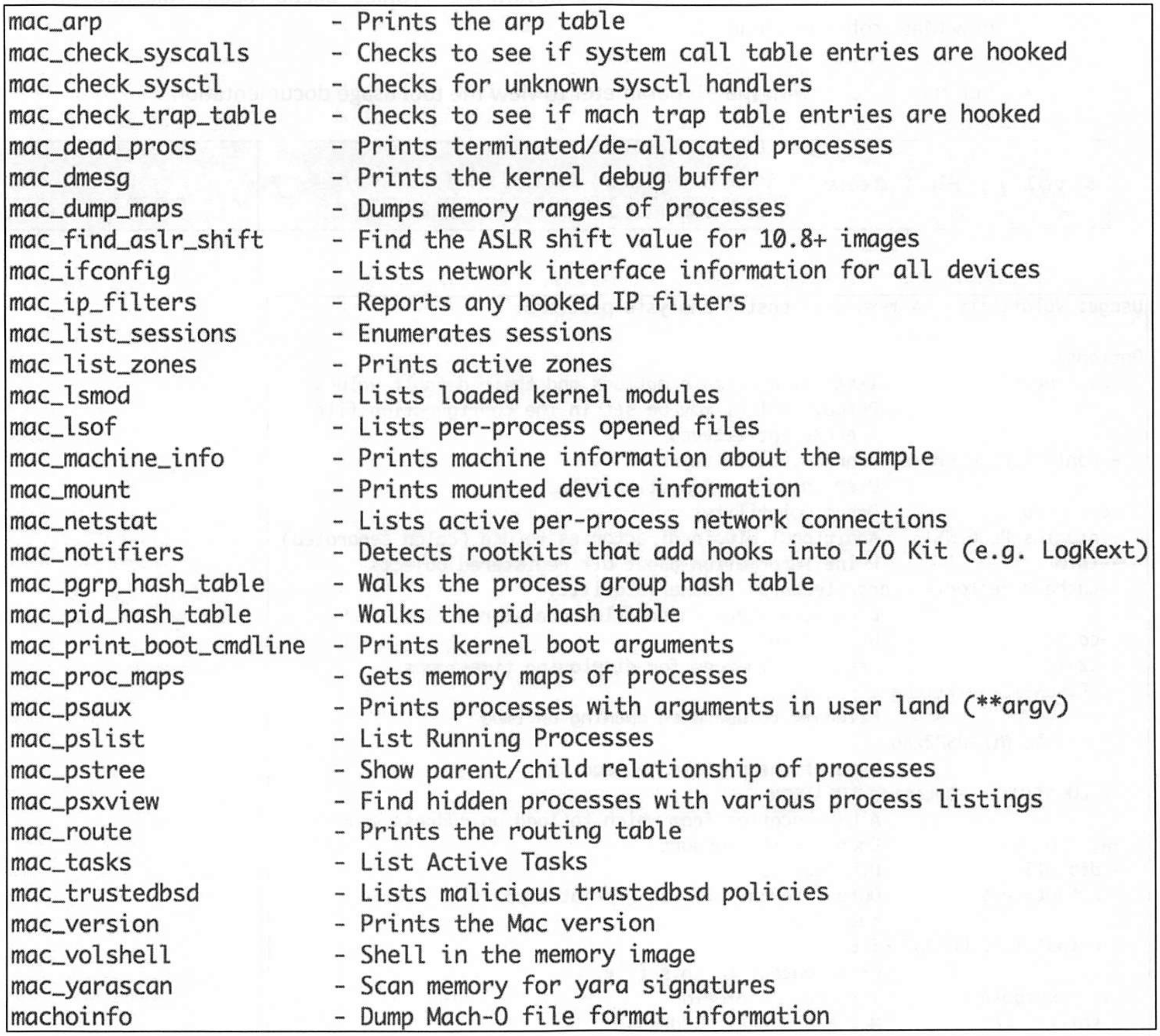
Lab: Step-by-Step

1. Volatility: Documentation

- Run the `vol.py` with the `--info` parameter to view the tool documentation.

```
$ vol.py --info | less
```

- Review the "Plugins" Section. Note the plugins named with the "`mac_*`". We will be using some of these in this lab.



```
mac_arp                  - Prints the arp table
mac_check_syscalls       - Checks to see if system call table entries are hooked
mac_check_sysctl         - Checks for unknown sysctl handlers
mac_check_trap_table     - Checks to see if mach trap table entries are hooked
mac_dead_procs           - Prints terminated/de-allocated processes
mac_dmesg                - Prints the kernel debug buffer
mac_dump_maps            - Dumps memory ranges of processes
mac_find_aslr_shift      - Find the ASLR shift value for 10.8+ images
mac_ifconfig             - Lists network interface information for all devices
mac_ip_filters           - Reports any hooked IP filters
mac_list_sessions        - Enumerates sessions
mac_list_zones           - Prints active zones
mac_lsmod                - Lists loaded kernel modules
mac_lsof                 - Lists per-process opened files
mac_machine_info         - Prints machine information about the sample
mac_mount                - Prints mounted device information
mac_netstat              - Lists active per-process network connections
mac_notifiers            - Detects rootkits that add hooks into I/O Kit (e.g. LogKext)
mac_pgrp_hash_table      - Walks the process group hash table
mac_pid_hash_table       - Walks the pid hash table
mac_print_boot_cmdline   - Prints kernel boot arguments
mac_proc_maps            - Gets memory maps of processes
mac_psaux                - Prints processes with arguments in user land (**argv)
mac_pslist               - List Running Processes
mac_pstree               - Show parent/child relationship of processes
mac_psxview              - Find hidden processes with various process listings
mac_route                - Prints the routing table
mac_tasks                - List Active Tasks
mac_trustedbsd           - Lists malicious trustedbsd policies
mac_version              - Prints the Mac version
mac_volshell             - Shell in the memory image
mac_yarascan             - Scan memory for yara signatures
machoinfo                - Dump Mach-O file format information
```

- Review the "`Profiles`" Section. By default, there are no Mac profiles loaded with Volatility. We will need to install them.

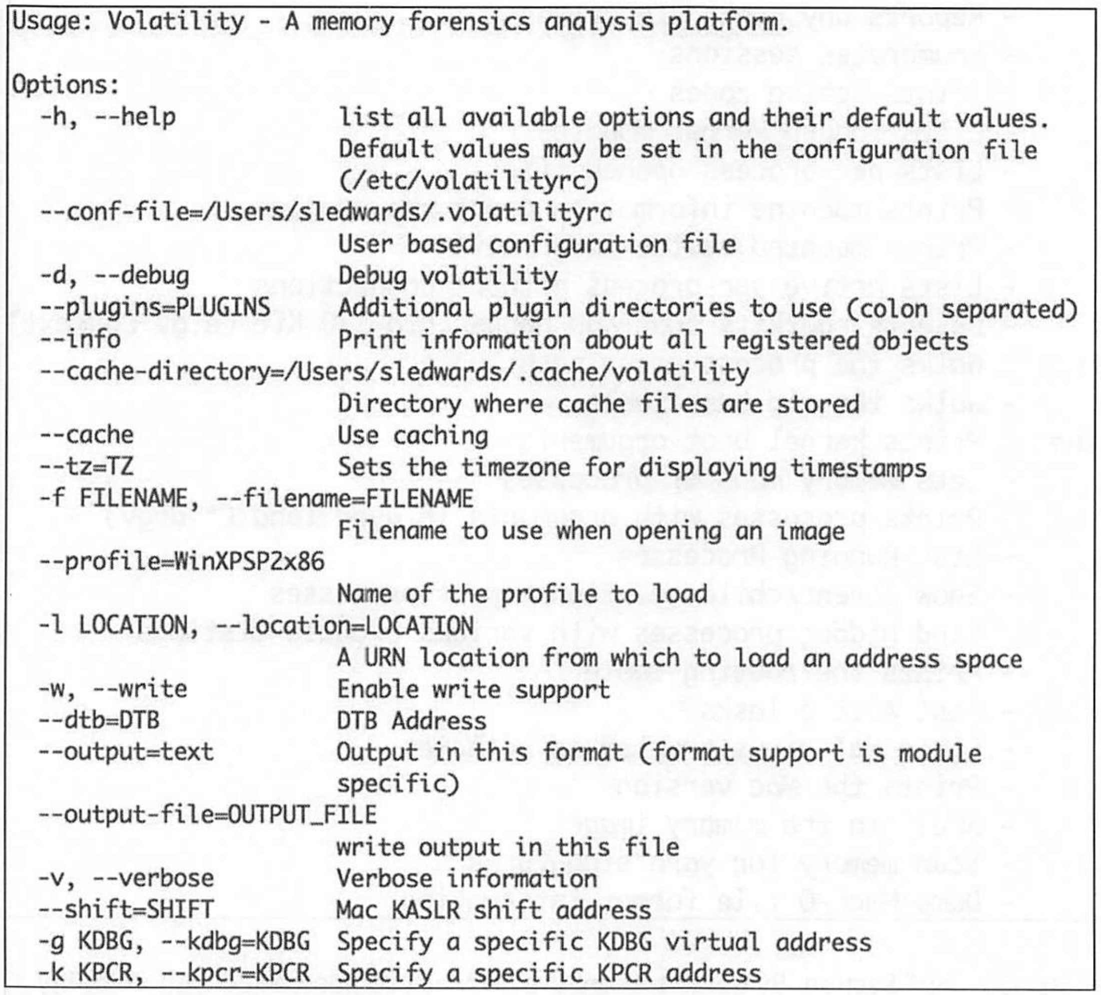
- In your Lab 5.4 directory, please copy the profile ZIP (*HighSierra_10.13.1_17B35a.zip*) archive to the Volatility profile directory. This is the default installation area for items installed with Homebrew. **Your directory path may be slightly different**; use tab completion to ensure you have the correct path. You do not need to unzip them.

```
$ cp HighSierra_10.13.1_17B35a.zip
/usr/local/Cellar/volatility/<#.#_#>/libexec/lib/python2.7/site-
packages/volatility/plugins/overlays/mac/
```

- Re-run the `vol.py -info` command. Review the "Profiles" Section again. Take note of the new Mac profile now loaded.

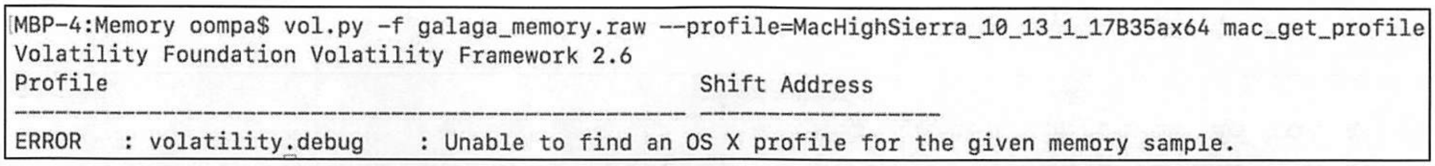
- Run the `vol.py` with the `-h` parameter to view the tool usage documentation.

```
$ vol.py -h | less
```



```
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/Users/sledwards/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --info                Print information about all registered objects
  --cache-directory=/Users/sledwards/.cache/volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the timezone for displaying timestamps
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --output=text         Output in this format (format support is module
                        specific)
  --output-file=OUTPUT_FILE
                        write output in this file
  -v, --verbose         Verbose information
  --shift=SHIFT         Mac KASLR shift address
  -g KDBG, --kdbg=KDBG  Specify a specific KDBG virtual address
  -k KPCR, --kpcr=KPCR  Specify a specific KPCR address
```

- Run the `vol.py` with the `-f` (filename) parameter on the `galaga_memory.raw` image. We will use the `mac_get_profile` plugin first to determine which profile we need to use.
    i. **This will error out—unfortunately, it does not recognize the profile required.**
    ii. Trial and error with various 10.13.1 profiles shows that the installed profile needed is the "`MacHighSierra_10_13_1_17B35ax64`" profile that was just installed.



```
MBP-4:Memory oompa$ vol.py -f galaga_memory.raw --profile=MacHighSierra_10_13_1_17B35ax64 mac_get_profile
Volatility Foundation Volatility Framework 2.6
Profile                                                   Shift Address
--------------------------------------------------------- ------------------
ERROR    : volatility.debug    : Unable to find an OS X profile for the given memory sample.
```

```
$ vol.py -f galaga_memory.raw mac_get_profile
```

2. Volatility Analysis: System Information

- Run the `vol.py` with the `mac_version` parameter to view the system kernel information. *****NOTE: These Volatility command lines can be long. These commands are meant to be executed as a single line. (They appear as two lines in this lab.)**

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_version
```

1. What kernel version does this system use?
    a. 17.2.0
    b. "Darwin Kernel Version **17.2.0**: Fri Sep 29 18:27:05 PDT 2017; root:xnu-4570.20.62~3/RELEASE_X86_64"

- Run the `vol.py` with the `mac_mount` parameter to view mounted volumes on this system.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_mount
```

2. What external disk is mounted on this system?
    a. /Volumes/WDPassport
3. What format is `/dev/disk5s2`?
    a. HFS+

3. Volatility: Network Information

- Run the `vol.py` with the `mac_ifconfig` parameter to view the system network configuration.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_ifconfig
```

1. What IPv4 did this system have at the time of acquisition?
   a. 192.168.101.138

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_netstat
```

2. Whose email servers are the Mail applications calling out to (use `whois`)?
   a. 173.194.206.109 = Google
   b. 17.36.205.4 = Apple
   c. Look for entries in the output that have "Mail/1093" in the Process column.
   d. Perform a `whois` on these IP addresses to find out the company associated with the IP address (`whois <IP Address>`).

4. Volatility: Processes

- Run the `vol.py` with the `mac_pslist` parameter to view system processes by walking the process list.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_pslist
```

1. What are the process names for PID 0 and 1?
   a. Kernel_task (0), launchd (1)
2. Find the "keylogger" process—who owns this process?
   a. UID/GID is 0 = root
3. What is the process ID for the "`logKextDaemon`" process?
   a. 96

- Run the `vol.py` with the `mac_pstree` parameter to view system processes in a tree formation.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_pstree
```

4. What process was performed using the `sudo` command?
   a. `osxpmem` (Capturing this memory image)

- Run the `vol.py` with the `mac_lsof` parameter to view the open file handles for each process.
- The output from this command can be quite verbose; you can choose to redirect the output to a file for easier analysis.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_lsof >
~/FOR518/mac_lsof.txt

$ open ~/FOR518/mac_lsof.txt
```

5. What file (that would be of investigative value) does "`logKextDaemon`" have open?
   b. `/Galaga/Library/Preferences/com.fsb.logKext`
   c. Look for items opened by process 96 that we found in a previous question.

5. Volatility: Kernel Extensions

- Run the `vol.py` with the `mac_lsmod` parameter to view kernel extensions.

```
$ vol.py -f galaga_memory.raw
--profile=MacHighSierra_10_13_1_17B35ax64 mac_lsmod
```

1. What two kernel extensions are loaded that are not from Apple?
   a. `com.google.MacPmem`
   b. `com.fsb.kext.logKext`
   c. Look for items that do not have the "com.apple.*" naming scheme. Yes, someone could name their malware as `com.apple.somethingevil` and hide as an Apple kernel extension.

6. Create a Dictionary File for Password Cracking

- Use the `strings` command below to create a dictionary file from the memory image to crack some passwords.
  i. The `-n` flag specifies the minimum string length of 8 characters.
  ii. This output will be piped to the "sort -u" command to filter out only unique strings.
  iii. This output will then be piped to two `awk` commands.
    1. The first `awk` command will filter for strings that contain only lower and uppercase characters (no special characters).
    2. The second `awk` command will filter for string lengths of less than 12 characters.
  iv. Finally, the output of these commands will be outputted to a file named `galaga_dictionary.txt` in your FOR518 directory.
  v. This should output a dictionary that is of reasonable size (416k) for relatively quick brute force password cracking. This should take just over a minute or so.

```
$ strings -n 8 galaga_memory.raw | sort -u | awk '$0 ~ /^[a-zA-
Z]{1,}$/'| awk 'length($0)<12' > ~/FOR518/galaga_dictionary.txt
```

7. Crack a Keychain File with John the Ripper

- Extract the `login.keychain-db` file for `dlightman` to your FOR518 directory.
    i. `/Volumes/galaga_mounted/Users/dlightman/Library/Keychains/`
- Extract the password hash from the `login.keychain-db` file using the `keychain2john.py` Python script. **Your directory path may be slightly different**; use tab completion to ensure you have the correct path.
- Ensure you extracted the password hash by using `cat` to view the contents of the newly created file.
- Using the `john` utility and the created dictionary file, crack the keychain password.
    i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password.
    ii. Press the "Enter" key a few times to see the status.
    iii. This should not take too long (~40 seconds depending on your Mac hardware—example time was performed on a 2016 MacBook Pro, 2.9Ghz, Core i5)
    iv. Once you get the password, use Control+C to quit John.
- Open the `login.keychain` file using "Keychain Access.app"; you may need to switch back and forth from your login keychain (the bolded one) to `dlightman`'s keychain file (the unbolded one) to get it to read correctly.
- **Note**: If you would like to re-run this hash crack again, remove the `john.pot` file from your home directory using the command "`rm ~/.john/john.pot`".

```
$ cp login.keychain-db ~/FOR518/

$ python /usr/local/Cellar/john-
jumbo/<#.#.#>/share/john/keychain2john.py ~/FOR518/login.keychain-db >
~/FOR518/dlightman_keychain.txt

$ cat ~/FOR518/dlightman_keychain.txt

$ john --wordlist=~/FOR518/galaga_dictionary.txt
~/FOR518/dlightman_keychain.txt
```

1. What kind of password hash is detected in the keychain?
    a. `(keychain, Mac OS X Keychain [PBKDF2-SHA1 3DES 8x SSE2])`
2. Approximately how many passwords per second is john brute forcing (sixth column from the left with the numbers labeled p/s)?
    a. ~900–1,000 (2016 MacBook Pro, 2.9Ghz, Core i5)
3. What is `dlightman`'s keychain password?
    a. `galagarocks`
4. Open the `login.keychain-db` file using Keychain Access.app. What is the password for the iOS Backup?
    d. `galagaftw`

8. Crack a Login Password with John the Ripper

- Using the already extracted (Lab 2.3) user plist for `dlightman`, extract the password hash from the `dlightman.plist` file using the `mac2john.py` Python script.

    i. `/Volumes/galaga_mounted/private/var/db/dslocal/nodes/Default/users/dlightman.plist`

- Ensure you extracted the password hash by using `cat` to view the contents of the newly created file.
- Using the `john` utility and the created dictionary file, crack the login password.
    - i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password.
    - ii. Press the "Enter" key a few times to see the status.
    - **iii. YOU WILL NOT CONTINUE TO CRACK THIS PASSWORD; use Control+C to quit John.**
        1. This took approximately 90 minutes on a 2016 MacBook Pro, 2.9Ghz, Core i5.

```
$ python /usr/local/Cellar/john-jumbo/<#.#.#>/share/john/mac2john.py
~/FOR518/dlightman.plist > ~/FOR518/dlightman_loginpassword.txt

$ cat ~/FOR518/dlightman_loginpassword.txt

$ john --wordlist=~/FOR518/galaga_dictionary.txt
~/FOR518/dlightman_loginpassword.txt
```

1. What kind of password hash is detected in the login password?
    a. `PBKDF2-HMAC-SHA512, GRUB2 / OS X 10.8+ [PBKDF2-SHA512 128/128 SSSE3 2x]`
2. Approximately how many passwords per second is john brute forcing (sixth column from the left with the numbers labeled p/s)?
    a. ~29 (2016 MacBook Pro, 2.9Ghz, Core i5)
3. If you had to take a guess, what is the user's login password?
    a. Same as keychain file, galagarocks.

9. Crack a DMG Password with John the Ripper

- Extract the `k12.dmg` file from `dlightman`'s system to your FOR518 directory.
- Extract the password hash from the `k12.dmg` file using the `dmg2john.py` Python script.
- Ensure you extracted the password hash by using `cat` to view the contents of the newly created file.
- Using the `john` utility and the created dictionary file, crack the DMG password.
    - i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password.
    - ii. Press the "Enter" key a few times to see the status.
    - **iii. YOU WILL NOT CONTINUE TO CRACK THIS PASSWORD; use Control+C to quit John.**
- Password cracking using a dictionary file is not perfect, as it turns out the password is not in the dictionary file we created, because it is six characters in length. Even if we filtered for passwords that were shorter, our dictionary file would have strings that included the password text but not as a string itself. Sometimes you win, sometimes you do not. A better dictionary file could have been created, but that takes a bit more work to do.

```
$ cp /Volumes/galaga_mounted/Users/dlightman/Documents/Stuff/k12.dmg
~/FOR518/

$ python /usr/local/Cellar/john-jumbo/<#.#.#>/share/john/dmg2john.py
~/FOR518/k12.dmg > ~/FOR518/dlightman_dmg.txt

$ cat ~/FOR518/dlightman_dmg.txt

$ john --wordlist=~/FOR518/galaga_dictionary.txt
~/FOR518/dlightman_dmg.txt
```

1. What kind of password hash is detected in the encrypted DMG file?
   a. `(dmg, Apple DMG [PBKDF2-SHA1 3DES/AES 8x SSE2])`
2. Approximately how many passwords per second is john brute forcing (sixth column from the left with the numbers labeled p/s)?
   a. ~3–4 (2016 MacBook Pro, 2.9Ghz, Core i5)
3. Using the keychain password, what is the password for this DMG file?
   a. tetris

Lab: Key Takeaways

- **Get comfortable with the Volatility command-line utilities for Mac memory analysis.**
- **Get familiar with John the Ripper's password cracking utilities.**
- **Understand the speed differences when using a dictionary file as well as speed differences of different encryption methods.**

Bonus Lab 5.5: Time Machine

Objectives

- **Review and analyze files associated with Time Machine backups.**
- **Review a Time Machine backup volume.**

Lab Preparation

(Note: Some of this might already be accomplished via earlier Labs, but this is the state that we hope your system is in prior to the start of this Lab. Just in case your system rebooted, we are including a guide to help you get back to the proper analysis starting point prior to the beginning of this Lab.)

1. **Software Preparation**: The following tools will be used in this Lab:
   - Terminal.app
     i. Locate and open the native OS X Terminal.app from /Applications/Utilities/

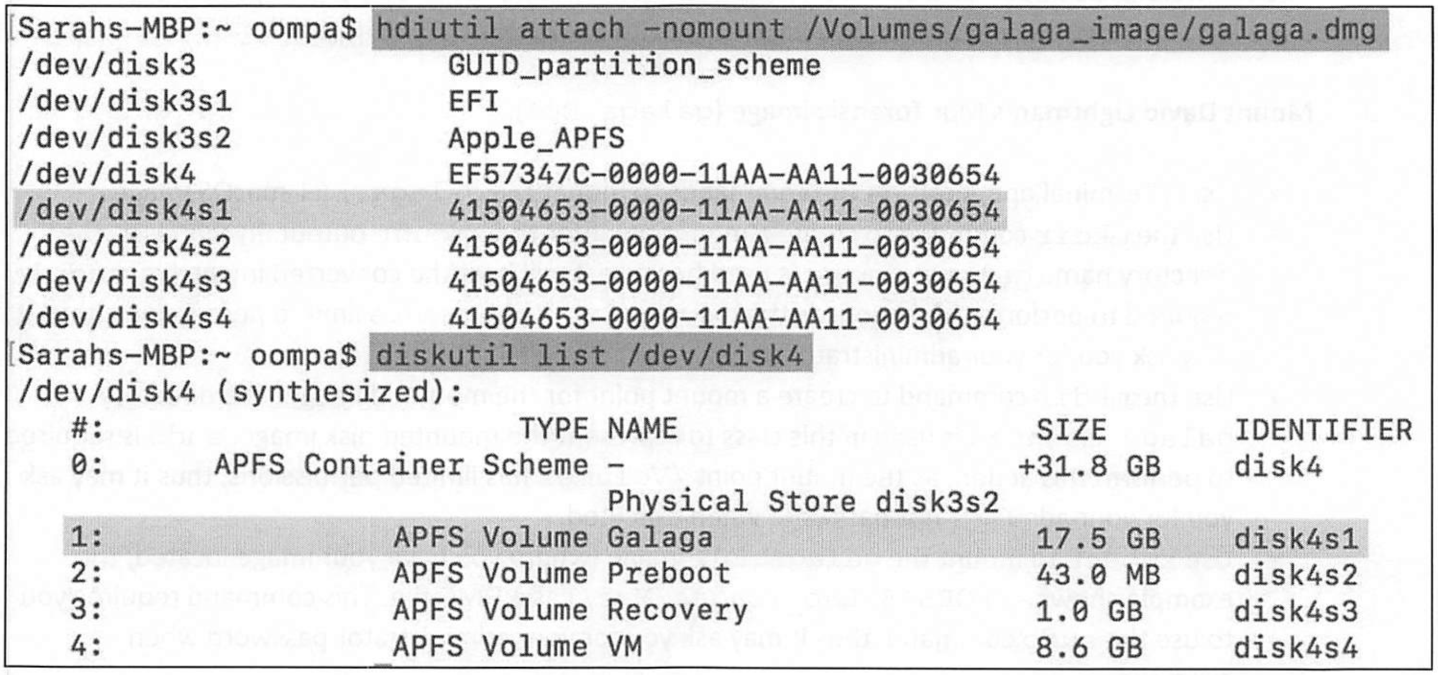
2. **FOR518 Reference Sheet**: Locate the FOR518 Reference Sheet provided to you in your class material and books. The PDF format of this sheet is available on your FOR518 USB drive.

3. **Mount David Lightman's Mac forensic image (`galaga.E01`).**

   - Using Terminal.app, perform the commands to mount the `galaga.E01` macOS image.
   - Use the `mkdir` command to create a mount point for the `xmount` output. In this class, the directory name `galaga_image` is used because it will host the converted image file. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use the `mkdir` command to create a mount point for the mounted image. The directory `galaga_mounted` is used in this class to represent the mounted disk image. `sudo` is required to perform this action, as the mount point `/Volumes` has limited permissions, thus it may ask you for your administrator password when executed.
   - Use `xmount` to mount the `galaga.E01` image (where you have your image located; the example shows `~/FOR518/Lab_Images/Mac/`) as a DMG file. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed.
     - `--in` – Tells `xmount` what input file type to expect; our images are in a compressed EWF format.
     - `--out` – Tells `xmount` what output format you want; we want a DMG file so we can mount it in Finder.
     - `Input File` – Where the image file is located on your system.
     - `Mount Point` – Newly created mount point `/Volumes/galaga_image` specifically for this image.

```
$ sudo mkdir /Volumes/galaga_image/

$ sudo mkdir /Volumes/galaga_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Mac/galaga.E01 --out dmg
/Volumes/galaga_image/
```

- Uses the `hdiutil` command with the "`attach`" verb to make the newly created DMG volume available. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display several `/dev/disk#`; use the appropriate disk device in the next command.
  - APFS disks will show many `/dev/disk*` options in the `hdiutil` output. The one we want to mount is the user's macOS volume. We can use the command "`diskutil list /dev/disk4`" on the synthesized disk to determine which is likely the user's macOS volume. David Lightman's volume is named "`Galaga`," highlighted in the example below. We will use `/dev/disk4s1` in the next command. **Be aware that yours may be mounted on a different disk number!**



```
[Sarahs-MBP:~ oompa$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg
/dev/disk3          	GUID_partition_scheme
/dev/disk3s1        	EFI
/dev/disk3s2        	Apple_APFS
/dev/disk4          	EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1        	41504653-0000-11AA-AA11-0030654
/dev/disk4s2        	41504653-0000-11AA-AA11-0030654
/dev/disk4s3        	41504653-0000-11AA-AA11-0030654
/dev/disk4s4        	41504653-0000-11AA-AA11-0030654
[Sarahs-MBP:~ oompa$ diskutil list /dev/disk4
/dev/disk4 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +31.8 GB    disk4
                                 Physical Store disk3s2
   1:                APFS Volume Galaga                  17.5 GB    disk4s1
   2:                APFS Volume Preboot                 43.0 MB    disk4s2
   3:                APFS Volume Recovery                1.0 GB     disk4s3
   4:                APFS Volume VM                      8.6 GB     disk4s4
```

- Use the `mount_apfs` command with the following parameters to mount the `/dev/disk#s#` (from the previous command) to the `/Volumes/galaga_mounted/` mount point. This command requires you to use the `sudo` command, thus it may ask you for your administrator password when executed. This drive will now be available in the Finder or Terminal applications.
  - `-o` – Options:
    - `rdonly` – Mount in read-only mode.
    - `noexec` – Do not allow execution of binaries on mounted system.
    - `noowners` – Ignore ownership on the mounted volume.

```
$ hdiutil attach -nomount /Volumes/galaga_image/galaga.dmg

$ sudo mount_apfs -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_mounted/
```

4. **Sanity Check**
   - You can access this newly created mounted drive on `/Volumes/galaga_mounted/`, thus all command line references in the workbook will be using this path. Using the Finder or the Terminal, access your newly created mounted volume.
   - Use the `ls -l` command to view the contents in the terminal to (hopefully) view the macOS directory structure. You should see an account for "`dlightman`" in the `Users` directory, hopefully not yours!

```
$ ls -l /Volumes/galaga_mounted/Users/
```

5. **Mount David Lightman's Mac Time Machine image (`galaga_timemachine.E01`).**
   - Same instructions as mounting the system image, however to mount the disk `mount_hfs` is used instead of `mount_apfs` since the Time Machine is an HFS+ formatted disk.
   - Select the `Apple_HFS` volume to provide the correct disk in the `mount_hfs` command for `/dev/disk*s*`

```
$ sudo mkdir /Volumes/galaga_tm_image/

$ sudo mkdir /Volumes/galaga_tm_mounted/

$ sudo xmount --in ewf ~/FOR518/Lab_Images/Time\  Machine/
galaga_timemachine.E01 --out dmg /Volumes/galaga_tm_image/

$ hdiutil attach -nomount
/Volumes/galaga_tm_image/galaga_timemachine.dmg

$ sudo mount_hfs -j -o rdonly,noexec,noowners /dev/disk#s#
/Volumes/galaga_tm_mounted/
```

6. *****When Needed***: Image Unmount Instructions**
   - Use the `diskutil list` command to view the list of mounted disks. Find the disk that you want to eject. This one will be the one labeled "`(disk image)`" versus the one labeled "`(synthesized)`." In my example, it would be `/dev/disk3`.
   - Use the `diskutil eject` command on the disk you would like to eject.
   - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmount (likely `/Volumes/galaga_image/` if are you following the naming scheme from the examples).

- Use the `umount` command with the mount point to unmount the disk. You will have to use the `sudo` command.
- *****WARNING***: If you are in the mounted image in Terminal or have a program using the mounted disk, you will get an error that it cannot be ejected or unmounted.**

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/galaga_image
```

Lab: Questions

1. **On David's System Image: Review the Time Machine Preferences**
   - Use the `cd` command to explore the System Preferences directory.
   - Use the `open` command to open the `com.apple.TimeMachine.plist` file.

```
$ cd /Volumes/galaga_mounted/Library/Preferences/

$ open com.apple.TimeMachine.plist
```

   1. What directory is excluded from the Time Machine backup?

      ____

   2. How many snapshots have been created?

      ____

   3. When was the last backup performed?

      ____

   4. Is this backup disk encrypted?

      ____

   5. Are the Time Machine backups stored on a network or external hard drive?

      ____

2. **On David's Time Machine Image: Review the Time Machine—Machine Directory**
   - Use the `cd` command to explore the `Backups.backupdb` directory.
   - Use the `xattr -xl` command to view the extended attributes of the machine directory.

```
$ cd /Volumes/galaga_tm_mounted/Backups.backupdb/

$ xattr -xl David's\ MacBook\ Pro/
```

   1. What is the MAC address of the backed-up system?

      ____

   2. What is the make and model of the backed-up system?

      ____

3. **Time Machine: Review the Time Machine—Snapshot Metadata**
   - Use the `cd` command to explore the "David's MacBook Pro" directory.
   - Use '`ls -la`' to list the contents of this directory.
   - Use the `xattr -xl` command to view the extended attributes of the 2018-01-18-071124 snapshot.

```
$ cd /Volumes/galaga_tm_mounted/Backups.backupdb/David's\ MacBook\
Pro/

$ ls -la

$ xattr -xl 2018-01-18-071124/
```

1. What is the Snapshot number?

   ____

2. When did the backup start (in UTC)?

   ____

3. When did the backup complete (in UTC)?

   ____

4. What type of snapshot is it (Hourly = 2, Daily = 3, Monthly = 1)?

   ____

5. How many bytes were copied in this snapshot?

   ____

4. Time Machine: Review the Time Machine—`tmutil`

- Use the `tmutil uniquesize` command to view the unique size of all snapshots in this directory.
  i. Sudo may be needed in case of the "`Error calculating unique size.`" error.

```
$ tmutil uniquesize *
```

1. Which snapshot is the largest?

   ____

- Use the `tmutil calculatedrift` command to view the differences between snapshots in this directory.
  i. Use the period "." instead of "David's MacBook Pro/" for the current directory.

```
$ tmutil calculatedrift .
```

2. Which snapshots had the most data added?

   ____

- Use the tmutil compare command to compare two snapshots:
    i. 2018-01-18-100229
    ii. 2018-03-03-112237
- Output this to a file in your FOR518 directory named tm_compare.txt.
- Use the open command to view this file.

```
$ tmutil compare 2018-01-18-100229/ 2018-03-03-112237/ >
~/FOR518/tm_compare.txt

$ open ~/FOR518/tm_compare.txt
```

!	Metadata Changed
+	File Added
-	File Removed

3. How many files (non-hidden) were added into dlightman's Downloads directory?

____

4. How many Launch Daemons were added to the system?

____

5. How much data was removed in this snapshot?

____

Lab: Step-by-Step

1. On David's System Image: Review the Time Machine Preferences

- Use the `cd` command to explore the System Preferences directory.
- Use the `open` command to open the `com.apple.TimeMachine.plist` file.

```
$ cd /Volumes/galaga_mounted/Library/Preferences/

$ open com.apple.TimeMachine.plist
```

1. What directory is excluded from the Time Machine backup?
   a. /Users/Shared/adi
   b. SkipPaths Key
2. How many snapshots have been created?
   a. 25
   b. SnapshotDates Key
3. When was the last backup performed?
   a. March 3, 2018 at 1:50:09 PM (Local system time, EST)
   b. Look for the last timestamp in Destination/SnapshotDates Key
4. Is this backup disk encrypted?
   a. No
   b. LastKnownEncryptionState Key
5. Are the Time Machine backups stored on a network or external hard drive?
   a. External Drive
   b. Extract the BackupAlias Key and review in a hex editor: there are no indications of an AFP File Share mount.

2. On David's Time Machine Image: Review the Time Machine—Machine Directory

- Use the `cd` command to explore the `Backups.backupdb` directory.
- Use the `xattr -xl` command to view the extended attributes of the machine directory.

```
$ cd /Volumes/galaga_tm_mounted/Backups.backupdb/

$ xattr -xl David's\ MacBook\ Pro/
```

1. What is the MAC address of the backed-up system?
   a. b8:e8:56:37:ec:06
   b. com.apple.backupd.BackupMachineAddress
   c. You can verify that it matches the system by looking for the MAC address in the file: /Volumes/galaga_mounted/Library/Preferences/SystemConfiguration/NetworkInterfaces.plist
2. What is the make and model of the backed-up system?
   a. MacBookPro11,1
   b. com.apple.backupd.ModelID

3. Time Machine: Review the Time Machine—Snapshot Metadata

- Use the `cd` command to explore the "David's MacBook Pro" directory.
- Use "`ls -la`" to list the contents of this directory.
- Use the `xattr -xl` command to view the extended attributes of the 2018-01-18-071124 snapshot.

```
$ cd /Volumes/galaga_tm_mounted/Backups.backupdb/David's\ MacBook\
Pro/

$ ls -la

$ xattr -xl 2018-01-18-071124/
```

1. What is the Snapshot number?
   a. 19801
2. When did the backup start (in UTC)?
   a. 1516277479599482 = 2018-01-18 12:11:19 Thu UTC
   b. Take the first 10 digits and use `date -ur` to convert.
   c. com.apple.backupd.SnapshotStartDate
3. When did the backup complete (in UTC)?
   a. 1516277484119615 = 2018-01-18 12:11:24 Thu UTC
   b. Take the first 10 digits and use `date -ur` to convert.
   c. com.apple.backupd.SnapshotCompletionDate
4. What type of snapshot is it (Hourly = 2, Daily = 3, Monthly = 1)?
   a. com.apple.backupd.SnapshotType is 2, which is an hourly snapshot.
5. How many bytes were copied in this snapshot?
   a. com.apple.backupd.SnapshotTotalBytesCopied = 9,864,474 bytes

4. Time Machine: Review the Time Machine—`tmutil`

- Use the `tmutil uniquesize` command to view the unique size of all snapshots in this directory.
  i. Sudo may be needed in case of the "`Error calculating unique size.`" error.

```
$ tmutil uniquesize *
```

1. Which snapshot is the largest?
   a. 2018-03-03-112237, 32.1M

- Use the `tmutil calculatedrift` command to view the differences between snapshots in this directory.
  i. Use the period "." instead of "David's MacBook Pro/" for the current directory.

```
$ tmutil calculatedrift .
```

2. Which snapshots had the most data added?
    a. 2018-01-18-100229 - 2018-03-03-112237 (2.2G)

- Use the `tmutil compare` command to compare two snapshots:
    i. `2018-01-18-100229`
    ii. `2018-03-03-112237`
- Output this to a file in your `FOR518` directory named, `tm_compare.txt`.
- Use the `open` command to view this file.

```
$ tmutil compare 2018-01-18-100229/ 2018-03-03-112237/ >
~/FOR518/tm_compare.txt

$ open ~/FOR518/tm_compare.txt
```

!	**Metadata Changed**
+	**File Added**
-	**File Removed**

3. How many files (non-hidden) were added into dlightman's `Downloads` directory?
    a. 11
    b. Search for /Downloads/
4. How many Launch Daemons were added to the system?
    a. Two, keylogger.plist and logKext.plist
    b. Search for LaunchDaemon/; only the ones with the "+" were added.
5. How much data was removed in this snapshot?
    a. 499.2M
    b. Look in the stats at the bottom of the listing.

Lab: Key Takeaways

- **Understand how to analyze a Time Machine volume and its snapshots.**

"As usual, SANS courses pay for themselves by Day 2. By Day 3, you are itching to get back to the office to use what you've learned."
Ken Evans, Hewlett Packard Enterprise - Digital Investigation Services

SANS Programs
sans.org/programs

GIAC Certifications
Graduate Degree Programs
NetWars & CyberCity Ranges
Cyber Guardian
Security Awareness Training
CyberTalent Management
Group/Enterprise Purchase Arrangements
DoDD 8140
Community of Interest for NetSec
Cybersecurity Innovation Awards

Search SANSInstitute

SANS Free Resources
sans.org/security-resources

- E-Newsletters
  NewsBites: Bi-weekly digest of top news
  OUCH!: Monthly security awareness newsletter
  @RISK: Weekly summary of threats & mitigations
- Internet Storm Center
- CIS Critical Security Controls
- Blogs
- Security Posters
- Webcasts
- InfoSec Reading Room
- Top 25 Software Errors
- Security Policies
- Intrusion Detection FAQ
- Tip of the Day
- 20 Coolest Careers
- Security Glossary

SANS Institute
8120 Woodmont Avenue | Suite 310
Bethesda, MD 20814
301.654.SANS(7267)
info@sans.org