SANS

www.sans.org

**FORENSICS 518**

# Mac Forensic Analysis

# Workbook

*The right security training for your staff, at the right time, in the right location.*

For518_Wkbk_A05_01

# Exercise 0 – Exercise Setup (Pre-Class)

## Objectives

- Install required software for FOR518 – Mac Forensic Analysis

## Class Preparation

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

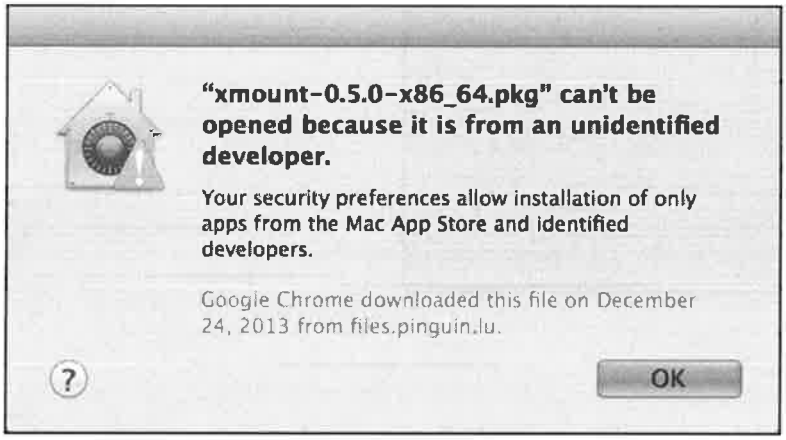
*You may use your host system **or** a virtual machine, however this setup has not been fully tested in a VM. If you choose to go this route, please be aware that not all tools may work as intended.*

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

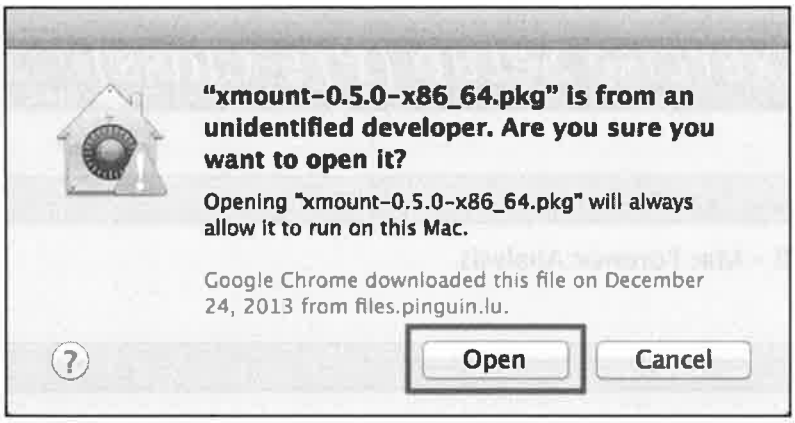
You may download the files at their respective websites listed or you may download an archive of these files at the provided SANS link.

Gatekeeper Settings:

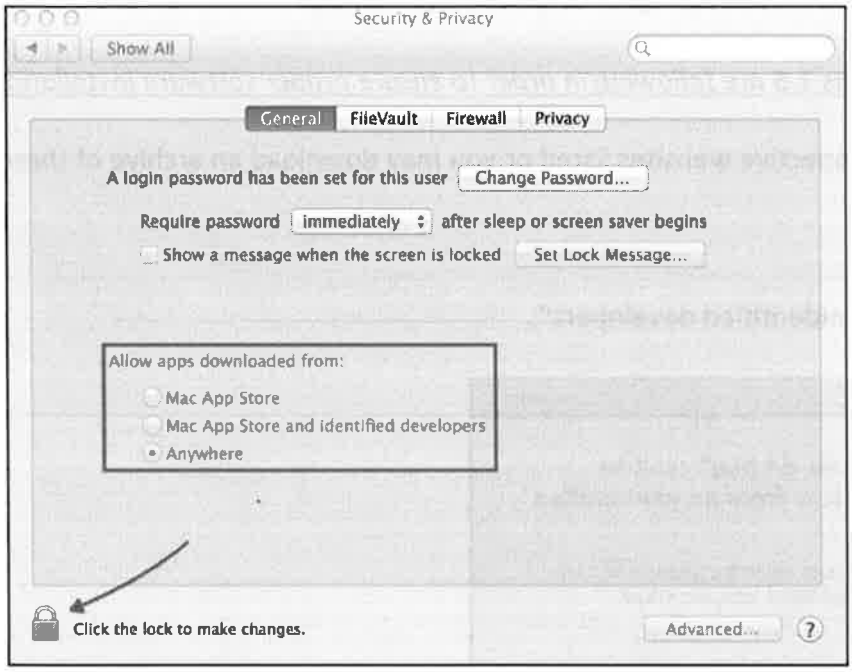
- Some installer files are from "unidentified developers".




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




- To permanently change this setting, navigate to the Security & Privacy Preferences Panel (`Apple Menu | System Preferences | Security & Privacy | General Tab`). Select "`Anywhere`" under "`Allow apps downloaded from:`".
    - This may require Administrative privileges.



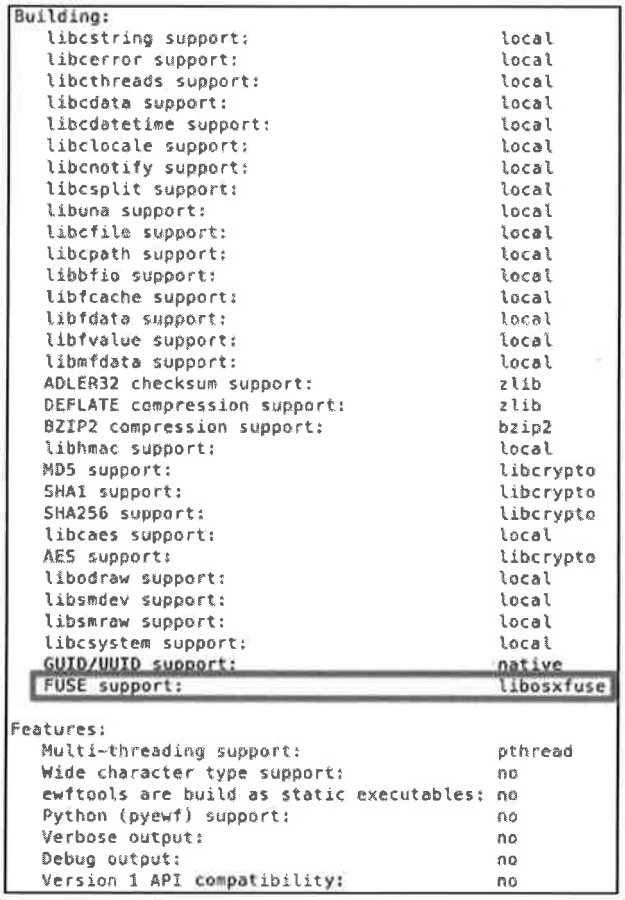

1. **Xcode**
    - If you have not already done so, register for an Apple Developer Account here. It requires an Apple ID, if you do not have one you may also register for one at `https://developer.apple.com/register/`

    1. Please download the latest Xcode available for your system from the App Store or `https://developer.apple.com/downloads/`
    2. Please download the latest Command Line Tools (for your version of OS X) from `https://developer.apple.com/downloads/`

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. **libewf**
    1. Download `libewf-########.tar.gz` from `https://53efc0a7187d0baa489ee347026b8278fe4020f6.googledrive.com/host/0B3fBvzttpiiSMTdoaVExWWNsRjg/`.
    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 `libewf-20131230.tar.gz` file.
    5. Once unpacked, cd into the `libewf-20131230` directory.
    6. Configure and install `libewf` using the commands: `./configure; make; sudo make install`.
    7. **A summary screen will be shown to you once complete. Ensure the "Fuse support" equals "libosxfuse".**



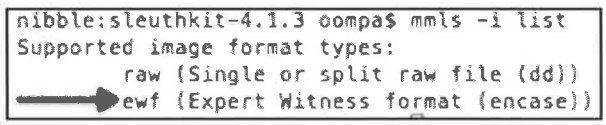
```
Building:
   libcstring support:                    local
   libcerror support:                     local
   libcthreads support:                   local
   libcdata support:                      local
   libcdatetime support:                  local
   libclocale support:                    local
   libcnotify support:                    local
   libcsplit support:                     local
   libuna support:                        local
   libcfile support:                      local
   libcpath support:                      local
   libbfio support:                       local
   libfcache support:                     local
   libfdata support:                      local
   libfvalue support:                     local
   libmfdata support:                     local
   ADLER32 checksum support:              zlib
   DEFLATE compression support:           zlib
   BZIP2 compression support:             bzip2
   libhmac support:                       local
   MD5 support:                           libcrypto
   SHA1 support:                          libcrypto
   SHA256 support:                        libcrypto
   libcaes support:                       local
   AES support:                           libcrypto
   libodraw support:                      local
   libsmdev support:                      local
   libsmraw support:                      local
   libcsystem support:                    local
   GUID/UUID support:                     native
   FUSE support:                          libosxfuse

Features:
   Multi-threading support:               pthread
   Wide character type support:           no
   ewftools are build as static executables: no
   Python (pyewf) support:                no
   Verbose output:                        no
   Debug output:                          no
   Version 1 API compatibility:           no
```

i. Like in screenshot above, your output should say "libosxfuse" rather than "no". If yours does not, try the following troubleshooting:
   1. Find your `fuse.h` file. In a terminal type `sudo find / -name fuse.h`
   2. Take note of where your fuse.h file is located. For example, the author's is installed in `/usr/local/include/osxfuse/fuse.h`
   3. Try the command `./configure --with-libfuse=/usr/local/`
   4. If all else fails, send the path to your `fuse.h` file and the `config.log` file created in the `libewf-YYYYMMDD` directory to your instructor.

```
$ cd ~/Downloads

$ tar xvf libewf-########.tar.gz

$ cd libewf-########

$ ./configure

$ make

$ sudo make install
```

4. **xmount 64-bit Package**
   1. Download xmount-0.5.0-x86_64.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.

5. **The Sleuth Kit**
   1. Download `sleuthkit-4.1.3.tar.gz` (or newer) from `http://sourceforge.net/projects/sleuthkit/`
   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-4.1.3.tar.gz` file.
   5. Once unpacked, `cd` into the `sleuthkit-4.1.3.tar.gz` directory.
   6. Configure and install `sleuthkit` using the commands: `./configure; make; sudo make install`
      a. You may need to install the Java JRE and JDK, please visit the following to download and install using default prompts.
         i. `https://www.java.com/en/download/mac_download.jsp`
         ii. `http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html`
   7. Ensure the install was successful and the `libewf` package was recognized by executing the `mmls -i list` command.



a. If you do not see the string "`ewf (Expert Witness format (encase))`" in this list something went wrong. Please attempt to reinstall `libewf` and The Sleuth Kit.

```
$ cd ~/Downloads

$ tar -xvf sleuthkit-4.1.3.tar.gz

$ cd sleuthkit-4.1.3

$ ./configure

$ make

$ sudo make install

$ mmls -i list
```

6. **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.

7. **Synalyze It!**
   1. 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.
   2. If purchased from the App Store, it will install automatically.
   3. If you downloaded the trial, unzip the file and move the application to your `/Applications` directory.

8. **SQLite Browsers:**
   - You may choose your favorite, these are recommended:
   - **Firefox & SQLite Manager Add-on**

   1. Firefox
      a. Download the latest version of Firefox from `firefox.com`.
      b. Open the DMG file, drag the Firefox application to the `/Applications` directory.
   2. SQLite Manager Add-on
      a. From Firefox, download from `http://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/`
      b. Click the green "`+ Add to Firefox`" button, press the "`Install Now`" button, and restart Firefox.
         i. To ensure successful installation, go to Tools in the menu – you should see "SQLite Manager".

   - **SQLite Database Browser**

   3. Download the latest version of SQLite Database Browser from `http://sqlitebrowser.org/`.

4. Open the DMG file, drag the SQLite Database Browser application to the `/Applications` directory.

9. **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.

10. **iBackupBot**
    1. Download "`iBackupBot for Mac`" Trial from `http://www.icopybot.com/download.htm`
    2. Open the DMG file, drag the iBackupBot application to the `/Applications` directory.

11. **Spotlight Inspector**
    1. Download the OSX version (select the appropriate checkbox) of Spotlight Inspector from `http://www.504ensics.com/tools/spotlight-inspector-digital-forensics-app-for-mac-osx/`
       a. This will require you to input and name and email.
    2. Double-click to unzip the archive.
    3. Unzip and move the application to the `/Applications` directory.

12. **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.

13. **Create a FOR518 directory**
    - The exercises for this class will reference a FOR518 folder in the user's home directory to dump various files for use in other exercises. (`~/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 command below shows how to create this folder in your home directory. You may also use the GUI interface to do this.

```
$ mkdir ~/FOR518
```

# Exercise 1.0 – Exercise Setup

## *Objectives*

- Introduction to FOR518 thumb drive.
- Copy files to host system.

## *Exercise Preparation*

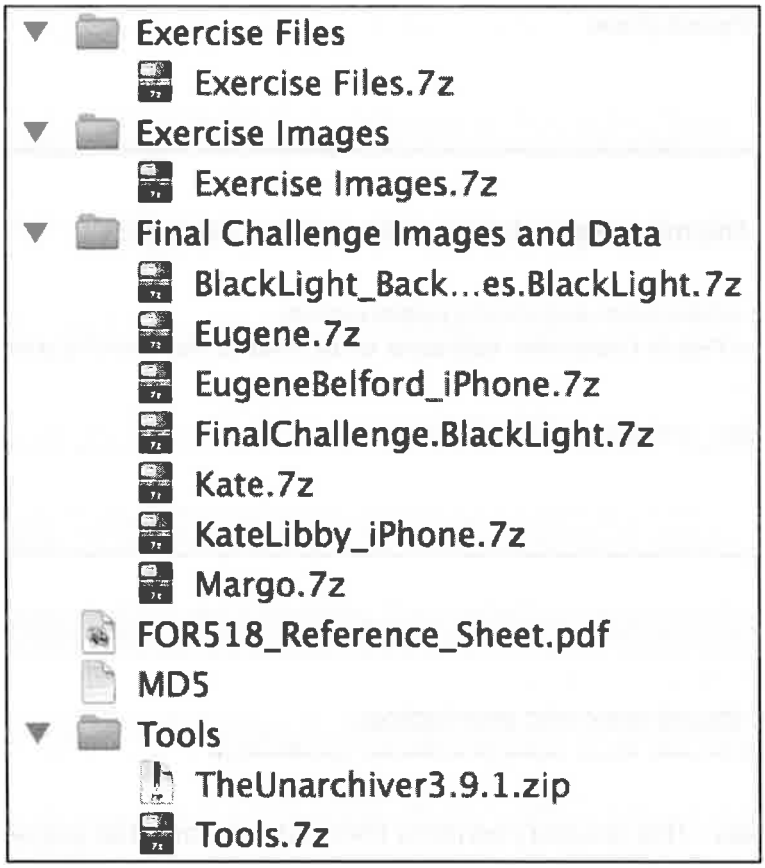
1. **Software Preparation** – The following tools may be used in this exercise:
   - The Unarchiver.app
     i. Locate The Unarchiver.app from /Applications/.
2. Locate the BlackLight and Epoch Converter software setup files in the `Tools` directory located on your FOR518 thumb drive.
   - `blacklight_mac_setup_201#r#.pkg`
   - `epoch_converter.app.zip`

## *Exercise*

1. **Introduction to the FOR518 Thumb Drive**
   1. Insert the FOR518 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. **Exercise Files** - This directory contains files and software that you will need for the following exercises:
         1. System Disk Image - dademurphy.E01
         2. Memory Image - dademurphy_memory.001
         3. Time Machine Image - dade_timemachine.E01
      ii. **Exercise Images** – This directory contains the forensic images that you will be working with on the exercises:
         1. Exercise 1.2 – Disks & Partitions
         2. Exercise 1.3 – HFS+
         3. Exercise 2.2 – Safari
         4. Exercise 4.1 – Time Machine & Spotlight
         5. Exercise 4.2 – Password Cracking & Encrypted Containers
         6. Exercise 4.4 – Memory Analysis
         7. Exercise 5.1 – Decoding iOS Artifacts
         8. Exercise 5.2 – iOS Artifacts from 3rd-party Apps
      iii. **Final Challenge Images** - This directory contains images and files needed to complete the FOR518 Mac Forensic Challenge on Day 6.
      iv. **Tools** - This directory contains many of the tools you have already installed plus some extras that will be installed later in the class. Most of the files are archived using 7zip. The Unarchiver utility has been provided so you may unarchive these files.

v. **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.

vi. **MD5** – This file contains the MD5 hashes for the 7zip archives as well as for the image files used in this class.




## 2. Copy & Unarchive

1. Copy the following items to your host system (or external hard drive):
   i. Exercise Files.7z
   ii. Exercise Images.7z
   iii. FOR518_Reference_Sheet.pdf
   iv. Tools.7z
2. Unarchive these items.
   i. You should have installed The Unarchiver.app application prior to coming to class in Exercise 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 thumb drive.

## 3. Install Epoch Converter

- If you already have a BlackBag software license, you may login to your BlackBag account on `www.blackbagtech.com` and download this tool by selecting `Resources | Free Tools` in the menu.
- Find the Epoch converter archive `(epoch_converter.app.zip)` from the Tool archive.
  1. Double-click to unzip the archive.
  2. Unzip and move the application to the `/Applications` directory.

4. **Install BlackLight**
    - Find the BlackLight Package installer file (`blacklight_mac_setup_201#r#.pkg`) from the Tool archive.
        1. Double-click the `blacklight_mac_setup_201#r#.pkg` file.
        2. Follow the default prompts.
        3. Once installed, open BlackLight. You should be presented with a window allowing you to "`Enter Demo Key...`", your instructor will provide you with a 90-day license name and key. Please enter this information where appropriate.

This page intentionally left blank.

# Exercise 1.1 – Mac Live Response

## Objectives

- Get familiar with the Mac OS X command line.
- Gather and analyze live response data.

## Exercise Preparation

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/
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.

## Exercise – Questions

**This exercise may be completed by using the provided screenshots or by using your own system. Please feel free to choose which option you prefer. (Option 1 starts below, Option 2 starts at question 9.)**

**Option 1 – Review the screenshots of the provided live response data.**

1. **Review the output of these system information commands:**

```
bit:~ oompa$ date
Sun Aug 12 18:13:26 EDT 2012
bit:~ oompa$ hostname
bit
bit:~ oompa$ uname -a
Darwin bit 12.0.0 Darwin Kernel Version 12.0.0: Sun Jun 24 23:00:16 PDT 2012; ro
ot:xnu-2050.7.9~1/RELEASE_X86_64 x86_64
bit:~ oompa$ sw_vers
ProductName:    Mac OS X
ProductVersion: 10.8
BuildVersion:   12A269
```

1. What time zone is this system using?

   _____

2. What is the hostname for this system?

   _____

3. What kernel version is this system using?

   _____

4. What version of OS X is it running?

_____

**2. Review the output of the `netstat -an` command:**

```
bit:~ oompa$ netstat -an | more
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  192.168.1.101.61354    74.125.228.105.443     ESTABLISHED
tcp4       0      0  192.168.1.101.61353    74.125.137.93.443      ESTABLISHED
tcp4       0      0  192.168.1.101.61345    192.168.1.1.445        ESTABLISHED
tcp4       0      0  192.168.1.101.61316    17.172.232.106.5223    ESTABLISHED
tcp4      37      0  192.168.1.101.61139    199.47.216.172.443     CLOSE_WAIT
tcp4      37      0  192.168.1.101.61045    199.47.217.173.443     CLOSE_WAIT
tcp4       0      0  192.168.1.101.61044    74.125.228.103.443     ESTABLISHED
tcp4      37      0  192.168.1.101.61035    107.22.245.91.443      CLOSE_WAIT
tcp4       0      0  192.168.1.101.61033    173.194.68.108.993     ESTABLISHED
tcp4       0      0  192.168.1.101.61027    205.188.0.231.443      ESTABLISHED
tcp4       0      0  192.168.1.101.61026    129.21.49.169.993      ESTABLISHED
tcp4       0      0  192.168.1.101.61025    173.194.68.108.993     ESTABLISHED
tcp4       0      0  192.168.1.101.61022    129.21.49.169.993      ESTABLISHED
tcp4       0      0  192.168.1.101.61021    173.194.68.109.993     ESTABLISHED
```

- Perform a `whois` on 74.125.228.105.

```
$ whois 74.125.228.105
```

1. Who is 74.125.228.105 registered to?

_____

2. What ports are these connections running on?

_____

_____

3. What IP address does this system have?

_____

3. **Review the output of the `lsof -i` command:**

```
bit:~ oompa$ lsof -i | more
COMMAND     PID  USER   FD   TYPE             DEVICE SIZE/OFF NODE NAME
SystemUIS   236 oompa    8u  IPv4 0x9ed2b1a706515075      0t0  UDP *:*
NetworkBr   241 oompa    5u  IPv4 0x9ed2b1a7065179cd      0t0  UDP *:*
imagent     258 oompa   12u  IPv4 0x9ed2b1a706763535      0t0  UDP localhost:58160->localhost:58160
imagent     258 oompa   13u  IPv4 0x9ed2b1a71bc9bc65      0t0  TCP bit:61027->bos-d005b-rdr4.blue.aol.com:https (ESTABLISHED)
imagent     258 oompa   16u  IPv4 0x9ed2b1a706eeb84d      0t0  TCP bit:61011->qc-in-f125.1e100.net:5223 (ESTABLISHED)
Dropbox     279 oompa   10u  IPv4 0x9ed2b1a71bca0c65      0t0  TCP bit:60999->sjc-not18.sjc.dropbox.com:http (ESTABLISHED)
Dropbox     279 oompa   16u  IPv4 0x9ed2b1a7065151fd      0t0  UDP *:17500
Dropbox     279 oompa   19u  IPv4 0x9ed2b1a70cace20d      0t0  TCP *:17500 (LISTEN)
Dropbox     279 oompa   21u  IPv4 0x9ed2b1a710367df5      0t0  TCP bit:61139->v-client-1a.sjc.dropbox.com:https (CLOSE_WAIT)
Dropbox     279 oompa   25u  IPv4 0x9ed2b1a709419c65      0t0  TCP localhost:26164 (LISTEN)
Dropbox     279 oompa   28u  IPv4 0x9ed2b1a710259ad5      0t0  TCP bit:61035->ec2-107-22-245-91.compute-1.amazonaws.com:https
 (CLOSE_WAIT)
Dropbox     279 oompa   29u  IPv4 0x9ed2b1a706eebf85      0t0  TCP bit:61045->v-client-2b.sjc.dropbox.com:https (CLOSE_WAIT)
Mail        824 oompa   34u  IPv4 0x9ed2b1a70fed8df5      0t0  TCP bit:61017->mail.csh.rit.edu:imaps (ESTABLISHED)
```

1. List three of the processes, their associated user, network type, and network connection information.

| Process | User | Connection Type | Network Connection Host, Protocol, Status |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

4. **Review the output of the `netstat -rn` and `arp -an` commands:**

```
bit:~ oompa$ netstat -rn | more
Routing tables

Internet:
Destination        Gateway            Flags        Refs      Use   Netif Expire
default            192.168.1.254      UGSc           22        0     en1
127                127.0.0.1          UCS             0        0     lo0
127.0.0.1          127.0.0.1          UH              5    23616     lo0
169.254            link#6             UCS             1        0     en1
169.254.204.125    b8:c7:5d:cc:5:80   UHLSW           0        1     en1
192.168.1          link#6             UCS             5        0     en1
192.168.1.1        c0:3f:e:8c:59:59   UHLWIi          1      104     en1    850
192.168.1.101      127.0.0.1          UHS             0        0     lo0
192.168.1.133      3c:7:54:3:65:20    UHLWIi          0     4502     en1    295
192.168.1.209      d0:23:db:72:91:10  UHLWIi          0       45     en1   1163
192.168.1.254      e0:69:95:50:4c:6   UHLWIir        23      577     en1   1188
192.168.1.255      ff:ff:ff:ff:ff:ff  UHLWbI          0        1     en1
```

```
bit:~ oompa$ arp -an
? (169.254.204.125) at b8:c7:5d:cc:5:80 on en1 [ethernet]
? (192.168.1.1) at c0:3f:e:8c:59:59 on en1 ifscope [ethernet]
? (192.168.1.133) at 3c:7:54:3:65:20 on en1 ifscope [ethernet]
? (192.168.1.209) at d0:23:db:72:91:10 on en1 ifscope [ethernet]
? (192.168.1.254) at e0:69:95:50:4c:6 on en1 ifscope [ethernet]
? (192.168.1.255) at ff:ff:ff:ff:ff:ff on en1 ifscope [ethernet]
```

1. What IP addresses did this system connect to recently?

_____

_____

_____

2. Look up the MAC Address for these IP addresses to see what kind of system it may be. *(Hint: Google the first three octets)*

_____

_____

_____

**5. Review the output of the `ifconfig` command:**

```
bit:~ oompa$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=2b<RXCSUM,TXCSUM,VLAN_HWTAGGING,TSO4>
        ether c4:2c:03:09:ca:fd
        media: autoselect (none)
        status: inactive
fw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 4078
        lladdr e8:06:88:ff:fe:d5:5d:08
        media: autoselect <full-duplex>
        status: inactive
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 90:27:e4:f8:e6:5f
        inet6 fe80::9227:e4ff:fef8:e65f%en1 prefixlen 64 scopeid 0x6
        inet 192.168.1.101 netmask 0xffffff00 broadcast 192.168.1.255
        media: autoselect
        status: active
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
        ether 02:27:e4:f8:e6:5f
        media: autoselect
        status: inactive
```

1. What is the IP address of this system?

_____

6. **Review the output of the `who -a` and `w` commands:**

```
bit:~ oompa$ who -a
reboot   ~        Aug  4 11:24 00:24         1
oompa    console  Aug  4 11:26  old         57
oompa    ttys000  Aug 12 18:13   .       13949
oompa    ttys001  Aug 12 18:18 00:09     13976
   .       run-level 3
bit:~ oompa$ w
18:55  up 8 days,  7:31, 3 users, load averages: 1.07 0.99 0.88
USER     TTY      FROM              LOGIN@  IDLE WHAT
oompa    console  -                04Aug12 8days -
oompa    s000     -                18:13       - w
oompa    s001     -                18:18       9 /usr/bin/less -is
```

1. What users are currently logged on?

   _____

2. What type of logins are the users using?

   _____

   _____

3. When was the last system reboot?

   _____

4. How long has the system been up?

   _____

5. What was the local system time when the `w` command was run?

   _____

6. Review the commands the users were running. (Shown in the output of the '`w`' command.)

7. **Review the output of the `last` command:**

```
byte:~ oompa$ last
oompa     ttys002                  Fri Jan  3 16:05   still logged in
testuser  ttys001                  Fri Jan  3 15:49   still logged in
testuser  console                  Fri Jan  3 15:48   still logged in
oompa     ttys000                  Thu Jan  2 19:51   still logged in
oompa     ttys003                  Thu Jan  2 19:51 - 19:51  (00:00)
oompa     ttys002                  Thu Jan  2 19:50 - 19:51  (00:00)
oompa     ttys001                  Thu Jan  2 19:50 - 19:51  (00:00)
oompa     ttys000                  Thu Jan  2 19:50 - 19:51  (00:00)
oompa     console                  Thu Jan  2 19:50   still logged in
reboot    ~                        Thu Jan  2 19:49
shutdown  ~                        Thu Jan  2 19:22
reboot    ~                        Thu Jan  2 19:22
shutdown  ~                        Thu Jan  2 19:21
reboot    ~                        Thu Jan  2 19:21
shutdown  ~                        Thu Jan  2 19:20
reboot    ~                        Thu Jan  2 19:20
shutdown  ~                        Thu Jan  2 19:19
reboot    ~                        Thu Jan  2 19:19
shutdown  ~                        Thu Jan  2 19:19
reboot    ~                        Thu Jan  2 19:18
shutdown  ~                        Thu Jan  2 19:17
oompa     ttys001                  Thu Jan  2 14:14 - 19:17  (05:02)
oompa     ttys000                  Thu Jan  2 13:55 - 19:17  (05:22)
oompa     console                  Thu Jan  2 13:55 - 19:17  (05:22)
reboot    ~                        Thu Jan  2 13:54
shutdown  ~                        Thu Jan  2 13:54
oompa     ttys000                  Thu Jan  2 13:39 - 13:54  (00:14)
oompa     ttys000                  Thu Jan  2 12:02 - 13:38  (01:36)
oompa     console                  Wed Jan  1 19:00 - 13:54  (18:54)
reboot    ~                        Wed Jan  1 18:59

wtmp begins Wed Jan  1 18:59
```

1. How many Terminals does the user '`testuser`' currently have open?

   _____

2. When did this log start?

   _____

8. **Review the output of the `ps aux` command:**

```
oompa      8814   0.3  1.2  1003732  98268   ??  S    Wed06PM   1:56.18 /Applications/Microsoft Office 2011/M
kelly     14583   0.2  0.5  2525132  45848   ??  S     7:09PM   0:02.11 /System/Library/PrivateFrameworks/He
oompa      8586   0.2  0.3  2572748  22840   ??  S    Tue08PM   4:49.34 /Applications/Utilities/Activity Mon
oompa       237   0.2  1.6  3847120 137032   ??  S     4Aug12  10:45.67 /System/Library/CoreServices/Finder.
root       8589   0.1  0.0  2445112   3248   ??  Ss   Tue08PM   3:30.86 /usr/libexec/activitymonitord
oompa       254   0.1  0.2  2548272  20176   ??  SN    4Aug12   0:12.17 /System/Library/CoreServices/Notifica
oompa       279   0.1  0.8   758272  65752   ??  S     4Aug12  14:27.73 /Applications/Dropbox.app/Contents/Ma
charlie   14677   0.0  0.0  2484312   3532   ??  SN    7:10PM   0:00.12 /usr/sbin/usernoted
charlie   14676   0.0  0.0  2485232   3000   ??  S     7:10PM   0:00.07 /System/Library/CoreServices/NetworkB
charlie   14673   0.0  0.2  2531632  16512   ??  S     7:10PM   0:00.48 /System/Library/CoreServices/Finder.a
charlie   14672   0.0  0.6  2580296  51196   ??  S     7:10PM   0:04.97 /System/Library/CoreServices/Dock.app
charlie   14662   0.0  0.0  2488108   2472   ??  S     7:09PM   0:00.03 /System/Library/Services/AppleSpell.s
charlie   14652   0.0  0.0  2467404   2920   ??  S     7:09PM   0:00.07 /System/Library/CoreServices/pbs
charlie   14651   0.0  0.3  2599168  24472   ??  S     7:09PM   0:01.43 /System/Library/CoreServices/SystemUI
charlie   14649   0.0  0.1  2501732  10120   ??  S     7:09PM   0:01.18 /System/Library/Frameworks/Applicatio
charlie   14648   0.0  0.1  2504224   7332   ??  S     7:09PM   0:00.11 /System/Library/CoreServices/talagent
charlie   14642   0.0  0.0  2433976   1592   ??  S     7:09PM   0:00.08 /usr/sbin/pboard
charlie   14633   0.0  0.0  2483820   2456   ??  S     7:09PM   0:00.13 /usr/sbin/distnoted agent
charlie   14631   0.0  0.0  2465740   1500   ??  S     7:09PM   0:00.25 /usr/sbin/cfprefsd agent
charlie   14629   0.0  0.0  2471500   1576   ??  Ss    7:09PM   0:00.14 /sbin/launchd
charlie   14616   0.0  0.3  2572380  25728   ??  Ss    7:09PM   0:01.50 /System/Library/CoreServices/loginwi
kelly     14608   0.0  0.0  2465392   2036   ??  S     7:09PM   0:00.02 /System/Library/CoreServices/AirPort
kelly     14603   0.0  0.2  2516072  14220   ??  Ss    7:09PM   0:00.31 com.apple.dock.extra
```

1. List the users on the system and a process they are running.

_____

_____

_____

_____

**Option 2 – Gather Information from your own system**

9. **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.
     ii. Run the `hostname` command.
     iii. Run the `uname -a` command.
     iv. Run the `sw_vers` command.

```
$ date

$ hostname

$ uname -a

$ sw_vers
```

10. **What are the active network connections <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. **Note**: The option "`-b`" shows the number of bytes transferred/received for each IP address.

```
$ netstat -an
```

11. **What are the active network connections <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
```

12. **Review <u>your</u> Routing and ARP tables**
    - Run and review the following commands as if you were responding to your analysis system.
    - Run the `netstat -rn` and `arp -an` commands

```
$ netstat -rn

$ arp -an
```

13. **Review the network configuration data <u>your</u> system**
    - Run and review the following commands as if you were responding to your analysis system.
    - Run the `ifconfig` command

```
$ ifconfig
```

14. **What are the open files <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.

```
$ lsof
```

15. **What users are logged onto your system?**
    - Run and review the following commands as if you were responding to your analysis system.
    - Run the `who -a` and `w` commands

```
$ who -a

$ w
```

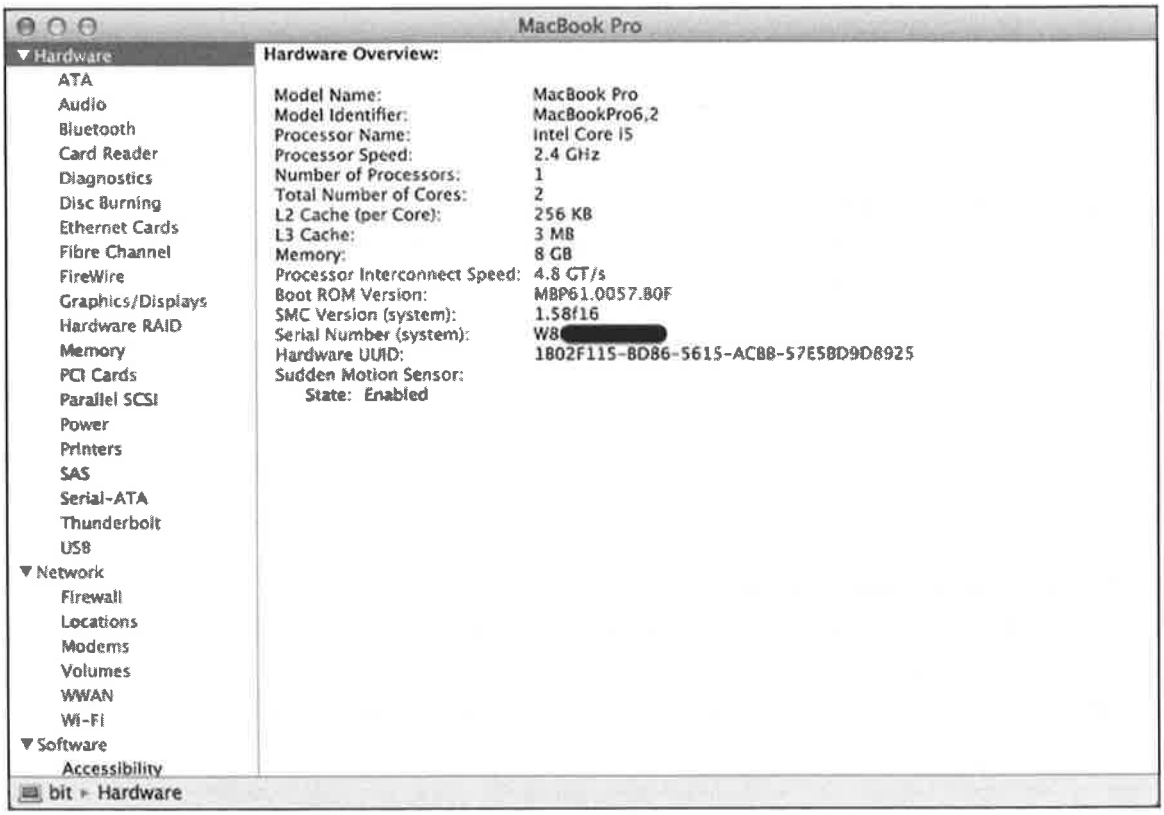
16. **What are the running processes your system?**
    - Run and review the following commands as if you were responding to your analysis system.
    - Run the `ps aux` command
        i. **Note**: The "`ps -ef`" command gives a different output that you may find preferable.

```
$ ps aux
```

17. **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`

```
$ system_profiler -xml -detailLevel full > system-profiler-data.spx
```

18. **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 `File | Open` to open the file you have just created.
    - Review the various data components.


MacBook Pro
Hardware
ATA
Audio
Bluetooth
Card Reader
Diagnostics
Disc Burning
Ethernet Cards
Fibre Channel
FireWire
Graphics/Displays
Hardware RAID
Memory
PCI Cards
Parallel SCSI
Power
Printers
SAS
Serial-ATA
Thunderbolt
USB
Network
Firewall
Locations
Modems
Volumes
WWAN
Wi-Fi
Software
Accessibility
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: 1B02F115-8D86-5615-AC88-57E5BD9D8925
Sudden Motion Sensor:
State: Enabled
bit ▸ Hardware

**Option 1:**

1. **Review the output of these system information commands:**

```
bit:~ oompa$ date
Sun Aug 12 18:13:26 EDT 2012
bit:~ oompa$ hostname
bit
bit:~ oompa$ uname -a
Darwin bit 12.0.0 Darwin Kernel Version 12.0.0: Sun Jun 24 23:00:16 PDT 2012; ro
ot:xnu-2050.7.9~1/RELEASE_X86_64 x86_64
bit:~ oompa$ sw_vers
ProductName:    Mac OS X
ProductVersion: 10.8
BuildVersion:   12A269
```

1. What time zone is this system using?
   - Eastern Daylight Time (EDT)
2. What is the hostname for this system?
   - 'bit'
3. What kernel version is this system using?
   - 12.0.0
   - You can review the Wikipedia page for Darwin to see what Kernel Version is mapped to which OS X version. (12.0.0 is Mountain Lion, 10.8)
   - [http://en.wikipedia.org/wiki/Darwin_(operating_system)]
4. What version of OS X is it running?
   - 10.8 – Mountain Lion

2. **Review the output of the `netstat -an` command:**

```
bit:~ oompa$ netstat -an | more
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  192.168.1.101.61354    74.125.228.105.443     ESTABLISHED
tcp4       0      0  192.168.1.101.61353    74.125.137.93.443      ESTABLISHED
tcp4       0      0  192.168.1.101.61345    192.168.1.1.445        ESTABLISHED
tcp4       0      0  192.168.1.101.61316    17.172.232.106.5223    ESTABLISHED
tcp4      37      0  192.168.1.101.61139    199.47.216.172.443     CLOSE_WAIT
tcp4      37      0  192.168.1.101.61045    199.47.217.173.443     CLOSE_WAIT
tcp4       0      0  192.168.1.101.61044    74.125.228.103.443     ESTABLISHED
tcp4      37      0  192.168.1.101.61035    107.22.245.91.443      CLOSE_WAIT
tcp4       0      0  192.168.1.101.61033    173.194.68.108.993     ESTABLISHED
tcp4       0      0  192.168.1.101.61027    205.188.0.231.443      ESTABLISHED
tcp4       0      0  192.168.1.101.61026    129.21.49.169.993      ESTABLISHED
tcp4       0      0  192.168.1.101.61025    173.194.68.108.993     ESTABLISHED
tcp4       0      0  192.168.1.101.61022    129.21.49.169.993      ESTABLISHED
tcp4       0      0  192.168.1.101.61021    173.194.68.109.993     ESTABLISHED
```

- Perform a `whois` on 74.125.228.105.

```
$ whois 74.125.228.105
```

1. Who is 74.125.228.105 registered to?
   a. Google, Inc.
2. What ports are these connections running on?
   a. 443
   b. 445
   c. 993
   d. 5223
3. What IP address does this system have?
   a. 192.168.1.101

### 3. Review the output of the `lsof -i` command:

```
bit:~ oompa$ lsof -i | more
COMMAND    PID  USER   FD   TYPE             DEVICE SIZE/OFF NODE NAME
SystemUIS  236 oompa    8u  IPv4 0x9ed2b1a706515075      0t0  UDP *:*
NetworkBr  241 oompa    5u  IPv4 0x9ed2b1a7065179cd      0t0  UDP *:*
imagent    258 oompa   12u  IPv4 0x9ed2b1a706763535      0t0  UDP localhost:58160->localhost:58160
imagent    258 oompa   13u  IPv4 0x9ed2b1a71bc9bc65      0t0  TCP bit:61027->bos-d005b-rdr4.blue.aol.com:https (ESTABLISHED)
imagent    258 oompa   16u  IPv4 0x9ed2b1a706eeb84d      0t0  TCP bit:61011->qc-in-f125.1e100.net:5223 (ESTABLISHED)
Dropbox    279 oompa   10u  IPv4 0x9ed2b1a71bca0c65      0t0  TCP bit:60999->sjc-not18.sjc.dropbox.com:http (ESTABLISHED)
Dropbox    279 oompa   16u  IPv4 0x9ed2b1a7065151fd      0t0  UDP *:17500
Dropbox    279 oompa   19u  IPv4 0x9ed2b1a70cace20d      0t0  TCP *:17500 (LISTEN)
Dropbox    279 oompa   21u  IPv4 0x9ed2b1a710367df5      0t0  TCP bit:61139->v-client-1a.sjc.dropbox.com:https (CLOSE_WAIT)
Dropbox    279 oompa   25u  IPv4 0x9ed2b1a709419c65      0t0  TCP localhost:26164 (LISTEN)
Dropbox    279 oompa   28u  IPv4 0x9ed2b1a710259ad5      0t0  TCP bit:61035->ec2-107-22-245-91.compute-1.amazonaws.com:https
 (CLOSE_WAIT)
Dropbox    279 oompa   29u  IPv4 0x9ed2b1a706eebf85      0t0  TCP bit:61045->v-client-2b.sjc.dropbox.com:https (CLOSE_WAIT)
Mail       824 oompa   34u  IPv4 0x9ed2b1a70fed8df5      0t0  TCP bit:61017->mail.csh.rit.edu:imaps (ESTABLISHED)
```

1. List three of the processes, their associated user, network type, and network connection information.

| Process | User | Connection Type | Network Connection Host, Protocol, Status |
|---|---|---|---|
| imagent | oompa | IPv4 | bos-d005b-rdr4.blue.aol.com<br>HTTPS<br>ESTABLISHED |
| Dropbox | oompa | IPv4 | v-client-1a.sjc.dropbox.com<br>HTTPS<br>CLOSE_WAIT |
| Mail | oompa | IPv4 | mail.csh.rit.edu<br>IMAPS<br>ESTABLISHED |

### 4. Review the output of the `netstat -rn` and `arp -an` commands:

```
bit:~ oompa$ netstat -rn | more
Routing tables

Internet:
Destination        Gateway            Flags        Refs      Use   Netif Expire
default            192.168.1.254      UGSc           22        0     en1
127                127.0.0.1          UCS             0        0     lo0
127.0.0.1          127.0.0.1          UH              5    23616     lo0
169.254            link#6             UCS             1        0     en1
169.254.204.125    b8:c7:5d:cc:5:80   UHLSW           0        1     en1
192.168.1          link#6             UCS             5        0     en1
192.168.1.1        c0:3f:e:8c:59:59   UHLWIi          1      104     en1    850
192.168.1.101      127.0.0.1          UHS             0        0     lo0
192.168.1.133      3c:7:54:3:65:20    UHLWIi          0     4502     en1    295
192.168.1.209      d0:23:db:72:91:10  UHLWIi          0       45     en1   1163
192.168.1.254      e0:69:95:50:4c:6   UHLWIir        23      577     en1   1188
192.168.1.255      ff:ff:ff:ff:ff:ff  UHLWbI          0        1     en1
```

```
bit:~ oompa$ arp -an
? (169.254.204.125) at b8:c7:5d:cc:5:80 on en1 [ethernet]
? (192.168.1.1) at c0:3f:e:8c:59:59 on en1 ifscope [ethernet]
? (192.168.1.133) at 3c:7:54:3:65:20 on en1 ifscope [ethernet]
? (192.168.1.209) at d0:23:db:72:91:10 on en1 ifscope [ethernet]
? (192.168.1.254) at e0:69:95:50:4c:6 on en1 ifscope [ethernet]
? (192.168.1.255) at ff:ff:ff:ff:ff:ff on en1 ifscope [ethernet]
```

1. What IP addresses did this system connect to recently?
    a. 192.168.1.1
    b. 192.168.1.133
    c. 192.168.1.209
2. Look up the MAC Address for these IP addresses to see what kind of system it may be. *(Hint: Google the first three octets)*
    a. Netgear (c0:3f:0e, remember if it is a single digit/character – it has a leading zero)
    b. Apple (b8:c7:5d)
    c. Apple (3c:07:54)

5. **Review the output of the `ifconfig` command:**

```
bit:~ oompa$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=2b<RXCSUM,TXCSUM,VLAN_HWTAGGING,TSO4>
        ether c4:2c:03:09:ca:fd
        media: autoselect (none)
        status: inactive
fw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 4078
        lladdr e8:06:88:ff:fe:d5:5d:08
        media: autoselect <full-duplex>
        status: inactive
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 90:27:e4:f8:e6:5f
        inet6 fe80::9227:e4ff:fef8:e65f%en1 prefixlen 64 scopeid 0x6
        inet 192.168.1.101 netmask 0xffffff00 broadcast 192.168.1.255
        media: autoselect
        status: active
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
        ether 02:27:e4:f8:e6:5f
        media: autoselect
        status: inactive
```

1. What is the IP address of this system?
    a. 192.168.1.101

6. **Review the output of the `who -a` and `w` commands:**

```
bit:~ oompa$ who -a
reboot   ~        Aug  4 11:24 00:24         1
oompa    console  Aug  4 11:26  old         57
oompa    ttys000  Aug 12 18:13   .       13949
oompa    ttys001  Aug 12 18:18 00:09     13976
   .       run-level 3
bit:~ oompa$ w
18:55  up 8 days,  7:31, 3 users, load averages: 1.07 0.99 0.88
USER     TTY      FROM              LOGIN@  IDLE WHAT
oompa    console  -                04Aug12 8days -
oompa    s000     -                18:13       - w
oompa    s001     -                18:18       9 /usr/bin/less -is
```

1. What users are currently logged on?
    a. oompa
2. What type of logins are the users using?
    a. GUI (console)
    b. Terminal (ttys###)

3. When was the last system reboot?
    a. August 4 at 11:24
4. How long has the system been up?
    a. 8 days, 7 hours and 31 minutes
5. What was the local system time when the `w` command was run?
    a. 18:55
6. Review the commands the users were running. (Shown in the output of the '`w`' command.)

**7. Review the output of the `last` command:**

```
byte:~ oompa$ last
oompa     ttys002                  Fri Jan  3 16:05   still logged in
testuser  ttys001                  Fri Jan  3 15:49   still logged in
testuser  console                  Fri Jan  3 15:48   still logged in
oompa     ttys000                  Thu Jan  2 19:51   still logged in
oompa     ttys003                  Thu Jan  2 19:51 - 19:51  (00:00)
oompa     ttys002                  Thu Jan  2 19:50 - 19:51  (00:00)
oompa     ttys001                  Thu Jan  2 19:50 - 19:51  (00:00)
oompa     ttys000                  Thu Jan  2 19:50 - 19:51  (00:00)
oompa     console                  Thu Jan  2 19:50   still logged in
reboot    ~                        Thu Jan  2 19:49
shutdown  ~                        Thu Jan  2 19:22
reboot    ~                        Thu Jan  2 19:22
shutdown  ~                        Thu Jan  2 19:21
reboot    ~                        Thu Jan  2 19:21
shutdown  ~                        Thu Jan  2 19:20
reboot    ~                        Thu Jan  2 19:20
shutdown  ~                        Thu Jan  2 19:19
reboot    ~                        Thu Jan  2 19:19
shutdown  ~                        Thu Jan  2 19:19
reboot    ~                        Thu Jan  2 19:18
shutdown  ~                        Thu Jan  2 19:17
oompa     ttys001                  Thu Jan  2 14:14 - 19:17  (05:02)
oompa     ttys000                  Thu Jan  2 13:55 - 19:17  (05:22)
oompa     console                  Thu Jan  2 13:55 - 19:17  (05:22)
reboot    ~                        Thu Jan  2 13:54
shutdown  ~                        Thu Jan  2 13:54
oompa     ttys000                  Thu Jan  2 13:39 - 13:54  (00:14)
oompa     ttys000                  Thu Jan  2 12:02 - 13:38  (01:36)
oompa     console                  Wed Jan  1 19:00 - 13:54  (18:54)
reboot    ~                        Wed Jan  1 18:59

wtmp begins Wed Jan  1 18:59
```

1. How many Terminals does the user '`testuser`' currently have open?
    a. One
2. When did this log start?
    a. January 1st at 18:59

8. **Review the output of the `ps aux` command:**

```
oompa     8814  0.3  1.2  1003732  98268  ??  S     Wed06PM   1:56.18 /Applications/Microsoft Office 2011/
kelly    14583  0.2  0.5  2525132  45848  ??  S      7:09PM   0:02.11 /System/Library/PrivateFrameworks/He
oompa     8586  0.2  0.3  2572748  22840  ??  S     Tue08PM   4:49.34 /Applications/Utilities/Activity Moni
oompa      237  0.2  1.6  3847120 137032  ??  S      4Aug12  10:45.67 /System/Library/CoreServices/Finder.a
root      8589  0.1  0.0  2445112   3248  ??  Ss    Tue08PM   3:30.86 /usr/libexec/activitymonitord
oompa      254  0.1  0.2  2548272  20176  ??  SN     4Aug12   0:12.17 /System/Library/CoreServices/Notifica
oompa      279  0.1  0.8   758272  65752  ??  S      4Aug12  14:27.73 /Applications/Dropbox.app/Contents/Ma
charlie  14677  0.0  0.0  2484312   3532  ??  SN     7:10PM   0:00.12 /usr/sbin/usernoted
charlie  14676  0.0  0.0  2485232   3000  ??  S      7:10PM   0:00.07 /System/Library/CoreServices/NetworkB
charlie  14673  0.0  0.2  2531632  16512  ??  S      7:10PM   0:00.48 /System/Library/CoreServices/Finder.a
charlie  14672  0.0  0.6  2580296  51196  ??  S      7:10PM   0:04.97 /System/Library/CoreServices/Dock.app
charlie  14662  0.0  0.0  2488108   2472  ??  S      7:09PM   0:00.03 /System/Library/Services/AppleSpell.s
charlie  14652  0.0  0.0  2467404   2920  ??  S      7:09PM   0:00.07 /System/Library/CoreServices/pbs
charlie  14651  0.0  0.3  2599168  24472  ??  S      7:09PM   0:01.43 /System/Library/CoreServices/SystemUI
charlie  14649  0.0  0.1  2501732  10120  ??  S      7:09PM   0:01.18 /System/Library/Frameworks/Applicatio
charlie  14648  0.0  0.1  2504224   7332  ??  S      7:09PM   0:00.11 /System/Library/CoreServices/talagent
charlie  14642  0.0  0.0  2433976   1592  ??  S      7:09PM   0:00.08 /usr/sbin/pboard
charlie  14633  0.0  0.0  2483820   2456  ??  S      7:09PM   0:00.13 /usr/sbin/distnoted agent
charlie  14631  0.0  0.0  2465740   1500  ??  S      7:09PM   0:00.25 /usr/sbin/cfprefsd agent
charlie  14629  0.0  0.0  2471500   1576  ??  Ss     7:09PM   0:00.14 /sbin/launchd
charlie  14616  0.0  0.3  2572380  25728  ??  Ss     7:09PM   0:01.50 /System/Library/CoreServices/loginwin
kelly    14608  0.0  0.0  2465392   2036  ??  S      7:09PM   0:00.02 /System/Library/CoreServices/AirPort
kelly    14603  0.0  0.2  2516072  14220  ??  Ss     7:09PM   0:00.31 com.apple.dock.extra
```

1. List the users on the system and a process they are running.
    a. oompa – Microsoft Office
    b. kelly – com.apple.dock.extra
    c. root - activitymonitord
    d. charlie – pboard

**Option 2:**

9. **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.
        ii. Run the `hostname` command.
        iii. Run the `uname -a` command.
        iv. Run the `sw_vers` command.

```
$ date

$ hostname

$ uname -a

$ sw_vers
```

10. **What are the active network connections 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. **Note**: The option "`-b`" shows the number of bytes transferred/received for each IP address.

```
$ netstat -an
```

**11. What are the active network connections <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
```

**12. Review <u>your</u> Routing and ARP tables**

- Run and review the following commands as if you were responding to your analysis system.
- Run the `netstat -rn` and `arp -an` commands

```
$ netstat -rn

$ arp -an
```

**13. Review the network configuration data <u>your</u> system**

- Run and review the following commands as if you were responding to your analysis system.
- Run the `ifconfig` command

```
$ ifconfig
```

**14. What are the open files <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.

```
$ lsof
```

15. **What users are logged onto your system?**
    - Run and review the following commands as if you were responding to your analysis system.
    - Run the `who -a` and `w` commands

```
$ who -a

$ w
```

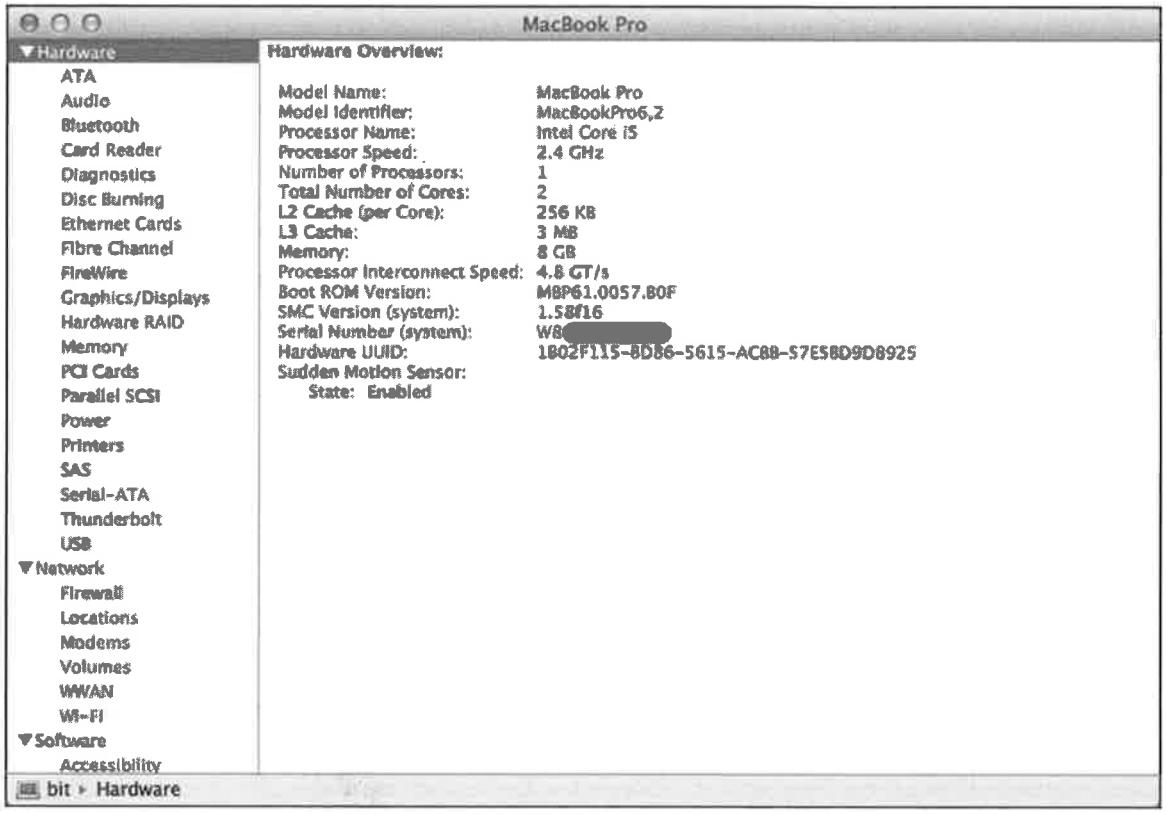
16. **What are the running processes your system?**
    - Run and review the following commands as if you were responding to your analysis system.
    - Run the `ps aux` command
        i. **Note**: The "`ps -ef`" command gives a different output that you may find preferable.

```
$ ps aux
```

17. **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`

```
$ system_profiler -xml -detailLevel full > system-profiler-data.spx
```

18. **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 `File | Open` to open the file you have just created.
    - Review the various data components.




## *Exercise – Key Takeaways*

- **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.

# Exercise 1.2 – Disks & Partitions

## Objectives

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

## 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/
   - 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 Exercise 0, please review this exercise 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. **Exercise File Preparation** – Locate the `GPT.dmg` file located in the `Exercise Files/Exercise 1.2 - Disks & Partitions` directory on your FOR518 USB drive. This file should have the MD5: 9e36e2a9e4fc9d6a04a1f13aad8c9e75. This can be checked by executing the command: `md5 GPT.dmg`.
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.

## Exercise – 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 five disks

| Disk Identifier | Partition Scheme | Number of Partitions | Volume Names | Volume Disk Format | Disk Size |
|---|---|---|---|---|---|
| /dev/disk0 | | | | | |
| /dev/disk1 | | | | | |
| /dev/disk2 | | | | | |
| /dev/disk3 | | | | | |
| /dev/disk4 | | | | | |

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

```
$ diskutil info disk0
```

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 disk0s2
```

1. What is the volume name?

   _____

2. What is the partition type?

   _____

3. What is the size of this volume?

   _____

5. **Use the `sudo gpt -r show` command, with and without the '`-l`' option**
   - Use your system's boot drive, likely `disk0`.

```
$ sudo gpt -r show disk0

$ sudo gpt -r show -l disk0
```

- Take note of the start sectors for the Protective MBR, Primary GPT Partition Header, Primary GPT Table and partitions. Also notice the difference between the GUIDs and Partition Labels. Refer back to the slide listing the Partition Type GUIDs. Fill in the GPT and partition information below.

- **GPT Information**

| | |
|---|---|
| Sector containing Protective MBR | |
| Sector containing Primary GPT Header | |
| Starting sector of Primary GPT Table | |
| Starting sector containing Secondary GPT Table | |
| Sector containing Secondary GPT Header | |

- **Partition Information**

| Partition Number | GUID Partition Type | Start Sector | Length (in Sectors) |
|---|---|---|---|
| 1 | | | |
| 2 | | | |
| 3 | | | |

6. **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 | | | |

7. **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>`
     ii. You may want to change the offsets from hex to decimal to use the offsets in this exercise, **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.

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

1. Is this volume bootable (Offset 446)?

_____

2. What is the partition type (Offset 450)?

3. What is the starting LBA (Offset 454-457)?

4. What is the size of the volume in sectors (Offset 458-461)?

**8. 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 size of the header in bytes (Offset 12-15)?

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

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

5. What is the GUID of the partition (Offset 56-71)?

6. Starting LBA of the GPT Partition Table (Offset 72-79)?

7. Number of partition table entries available (Offset 80-83)?

**9. 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+)?

_____

**10. Use the `hdiutil imageinfo GPT.dmg` command, 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 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 File Vaulted disks, try using a different disk or volume.

*Exercise – 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 | Volume Disk Format | Disk Size |
|---|---|---|---|---|---|
| /dev/disk0 | GUID Partition Scheme | 3 | EFI (Unamed)<br>Macintosh HD<br>Recovery HD | EFI (FAT)<br>HFS+<br>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 |
| /dev/disk4 | FDisk Partition Scheme (MBR) | 1 | ORANGE | FAT16 | 1 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 partitions 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 is 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 `sudo gpt -r show` command, with and without the '`-l`' option**
   - Use your system's boot drive, likely `disk0`.

```
$ sudo gpt -r show disk0

$ sudo gpt -r show -l disk0
```

The output of the `gpt show -r` command on the `/dev/disk0` device shows `disk0` has three partitions.

```
byte:~ oompa$ sudo gpt -r show disk0
      start       size  index  contents
          0          1         PMBR
          1          1         Pri GPT header
          2         32         Pri GPT table
         34          6
         40     409600      1  GPT part - C12A7328-F81F-11D2-BA4B-00A0C93EC93B
     409640  975093952      2  GPT part - 48465300-0000-11AA-AA11-00306543ECAC
  975503592    1269536      3  GPT part - 426F6F74-0000-11AA-AA11-00306543ECAC
  976773128          7
  976773135         32         Sec GPT table
  976773167          1         Sec GPT header
byte:~ oompa$ sudo gpt -r show -l disk0
      start       size  index  contents
          0          1         PMBR
          1          1         Pri GPT header
          2         32         Pri GPT table
         34          6
         40     409600      1  GPT part - "EFI System Partition"
     409640  975093952      2  GPT part - "Customer"
  975503592    1269536      3  GPT part - "Recovery HD"
  976773128          7
  976773135         32         Sec GPT table
  976773167          1         Sec GPT header
```

- **GPT Information**

| | |
|---|---|
| Sector containing Protective MBR | 0 |
| Sector containing Primary GPT Header | 1 |
| Starting sector of Primary GPT Table | 2 |
| Starting sector containing Secondary GPT Table | 976773135 |
| Sector containing Secondary GPT Header | 976773167 |

- **Partition Information**

| Partition Number | GUID Partition Type | Start Sector | Length (in Sectors) |
|---|---|---|---|
| 1 | EFI System Partition | 40 | 409600 |
| 2 | HFS+ Partition | 409640 | 975093952 |
| 3 | Apple Boot Partition | 975503592 | 1269536 |

6. **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 |

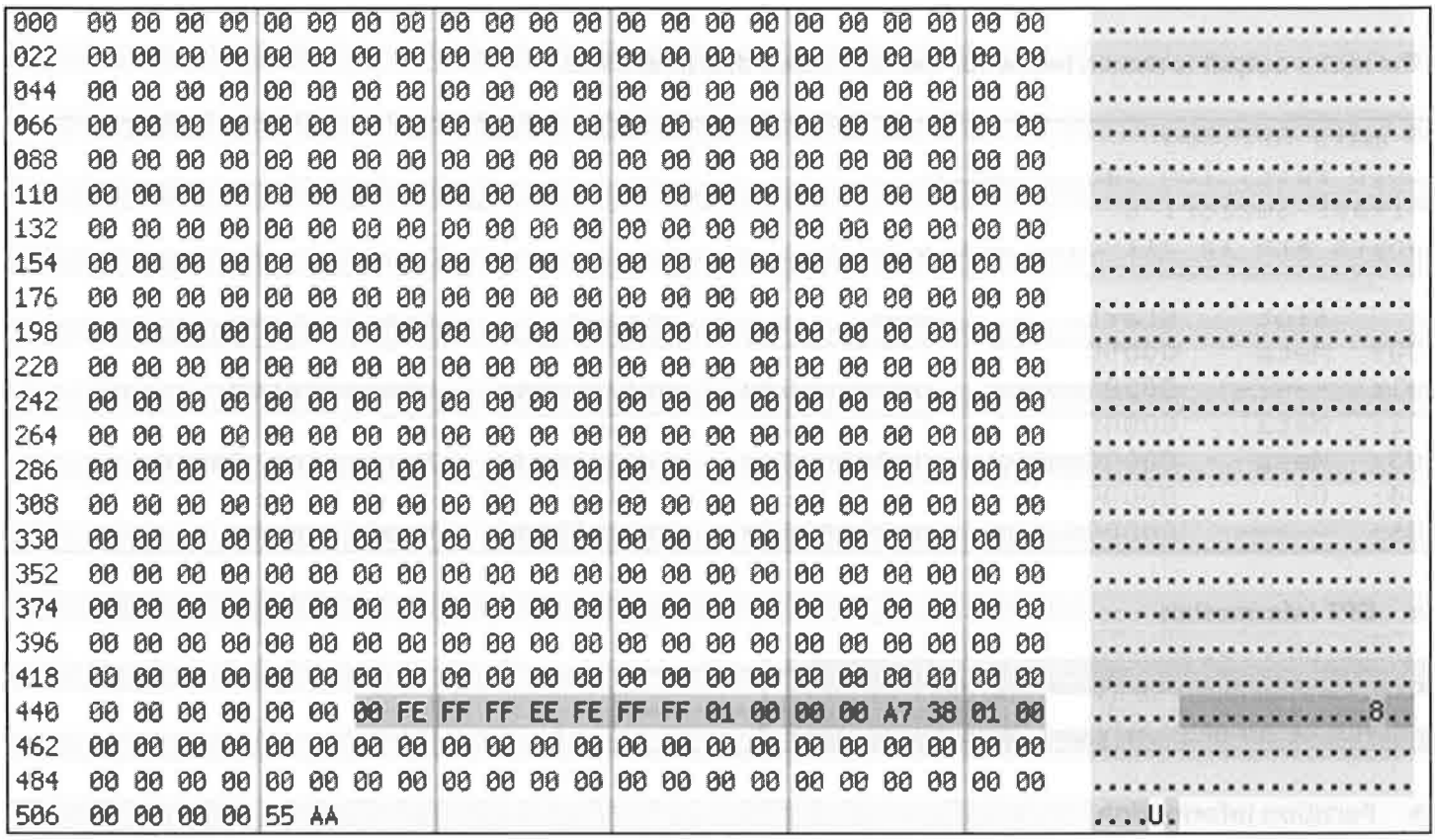
- **Partition Information**

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

**7. 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 exercise, **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.

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

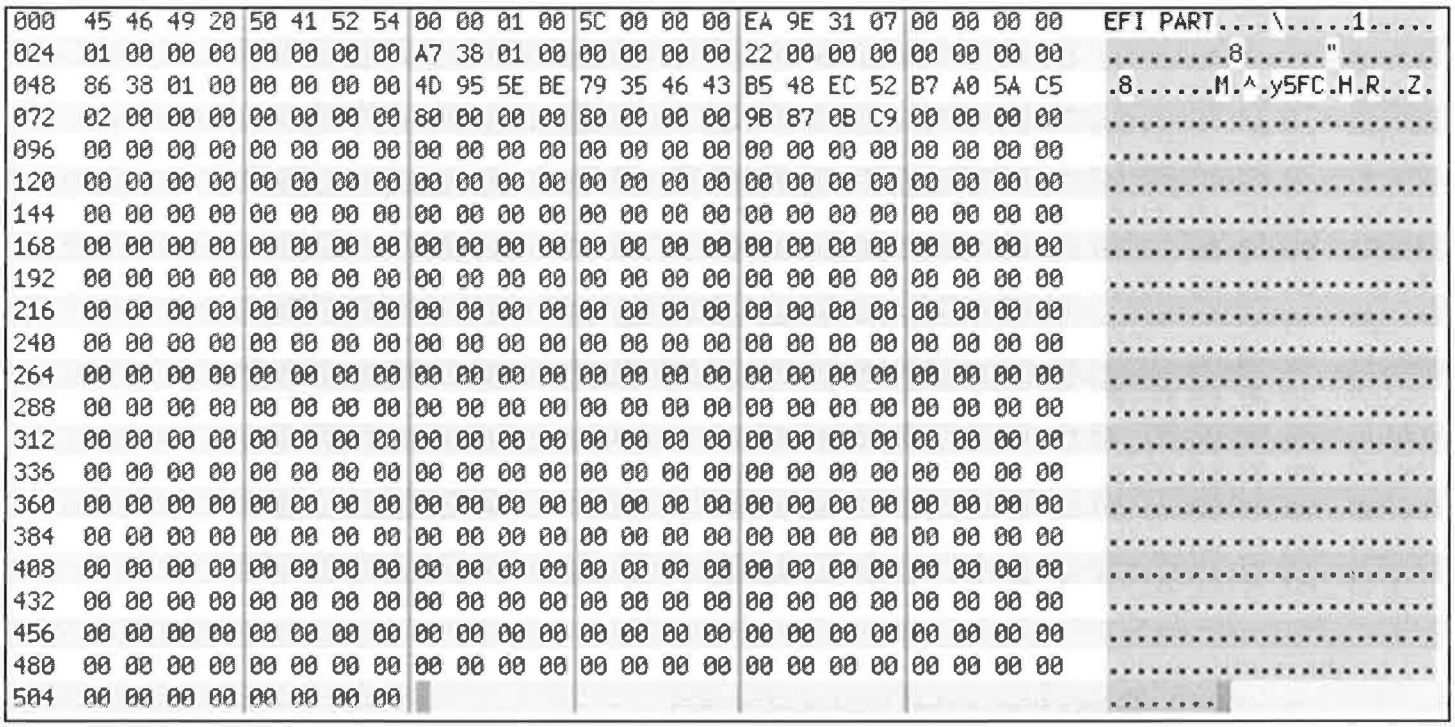


```
000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ......................
022  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ......................
044  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ......................
066  00 00 00 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)?
    a. 0x00 - No
2. What is the partition type (Offset 450)?
    a. 0xEE – EFI GPT Disk (EFI Protective MBR)
3. What is the starting LBA (Offset 454-457)?
    a. 0x01000000 = 1 (Little Endian)
4. What is the size of the volume in sectors (Offset 458-461)?
    a. 0x000138A7 = 0xA7380100 (Little Endian) = 80039

**8. Extract the GPT Header using dd**

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

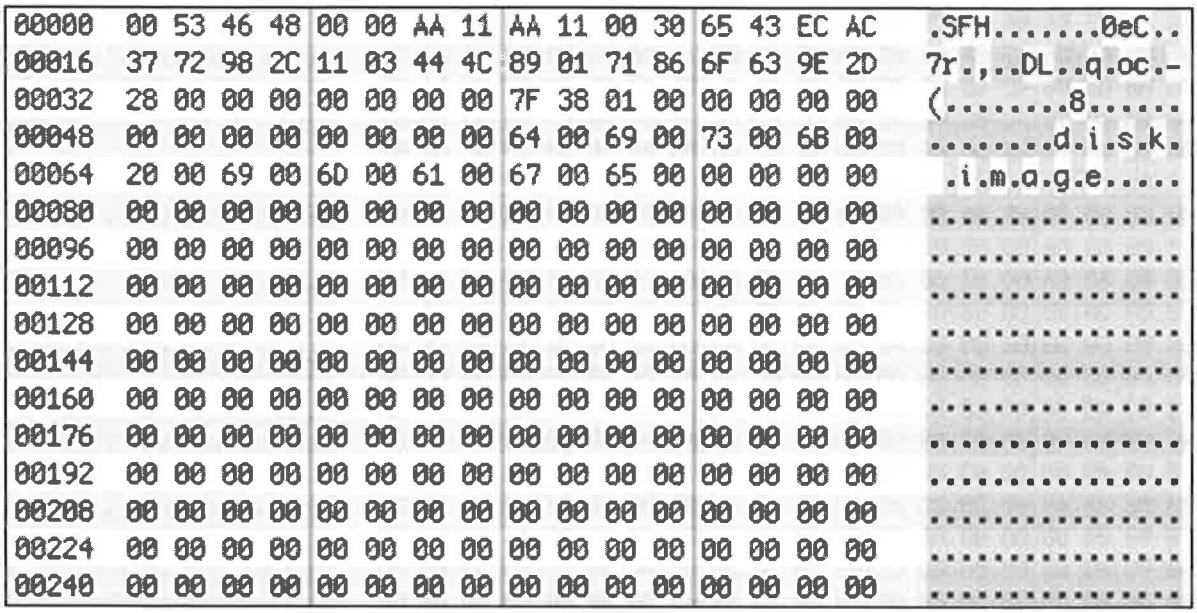


| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 000 | 45 | 46 | 49 | 20 | 50 | 41 | 52 | 54 | 00 | 00 | 01 | 00 | 5C | 00 | 00 | 00 | EFI PART....\.....1..... |
| 024 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | A7 | 38 | 01 | 00 | 00 | 00 | 00 | 00 | .........8....." ..... |
| 048 | 86 | 38 | 01 | 00 | 00 | 00 | 00 | 00 | 4D | 95 | 5E | BE | 79 | 35 | 46 | 43 | .8.....M.^.y5FC.H.R..Z. |
| 072 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 00 | 00 | 00 | 80 | 00 | 00 | 00 | ..... |
| 096 | 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 | ..... |
| 144 | 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 | ..... |
| 192 | 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 | ..... |
| 240 | 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 | ..... |
| 288 | 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 | ..... |
| 336 | 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 | ..... |
| 384 | 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 | ..... |
| 432 | 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 | ..... |
| 480 | 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 size of the header in bytes (Offset 12-15)?
    a. 0x5C – 92 bytes
3. What is the LBA of the GPT Header (this file) (Offset 24-31)?
    a. 0x0100000000000000 = 1 (Little Endian)
4. What is the LBA of the Backup/Secondary GPT Header (Offset 32-39)?
    a. 0xA738010000000000 = 80039 (Little Endian)
5. 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.
6. Starting LBA of the GPT Partition Table (Offset 72-79)?
    a. 0x0200000000000000 = 2 (Little Endian)
7. Number of partition table entries available (Offset 80-83)?
    a. 0x80000000 – 128 entries available (Little Endian)

**9. 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)

**10. Use the `hdiutil imageinfo GPT.dmg` command, 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 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 File Vaulted disks, try using a different disk or volume.

## *Exercise – 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.

# Exercise 1.3 – HFS+

## Objectives

- Review the HFS+ Volume Header and compare with native utilities.
- Review the HFS+ Special Files (Catalog, Extents Overflow and Attributes Files.)

## 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/
   - 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 Exercise 0, please review this exercise if needed.
   - Synalyze It!.app
     i. Locate and open the Synalyze It!.app from /Applications/Synalyze It!.app
     ii. This tool is available on your USB drive in the Tools directory.
     iii. This tool is available at http://www.synalysis.net/downloads/
   - BBT Epoch Converter
     i. Locate and open the BlackBag Epoch Converter from /Applications/Epoch Converter.app
2. **Exercise File Preparation** – Locate the `GPT.dmg` file located in the `Exercise Files/Exercise 1.3 - HFS+` directory on your FOR518 USB drive. This file should have the MD5: 9e36e2a9e4fc9d6a04a1f13aad8c9e75. This can be checked by executing the command: `md5 GPT.dmg`.
3. **FOR518 HFS+ Reference Sheet** – Locate the FOR518 HFS+ 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 exercise

## *Exercise – Questions*

### 1. Open another Terminal Window

- Use the `hdiutil fsid` command to view the volume header for the '`disk image`' volume. Use the Primary Volume Header.

```
$ hdiutil fsid GPT.dmg
```

1. What file system does Partition 4 '`disk image`' use?

   _____

2. How large is the volume?

   _____

3. What is the volume block size?

   _____

4. What is the signature of the volume?

   _____

5. When was this volume created?

   _____

### 2. Extract the Volume Header

- Use the `dd` command to extract the volume header for the '`disk image`' volume to a file.
- The volume header is located 1024 bytes from the beginning of the volume. In the last exercise we found that the 'disk image' volume starts at sector 40 and each sector is 512 bytes. Use the `skip=` parameter to go to this part of the disk.
- The volume header is always 512 bytes in size, use the `count=` parameter to only copy out one sector containing the volume header.
- Save it to a file named `volume_header`.

```
$ dd if=GPT.dmg skip=42 count=1 > volume_header
```

### 3. Parse the Volume Header

- Open the `volume_header` file in the hex editor of your choice.
- You can also use the Synalyze It! application for a colorized version of the volume header.
  - Open the Synalyze It! application.
  - Open the extracted volume header file from the previous step, `volume_header`.
  - Also open the included `volume_header_grammar` file, this file contains the colorized offsets to help translate what is seen in the `volume_header` file.

1. When was this volume last modified?

   _____

2. How many files does this volume contain?

   _____

3. How many folders does this volume contain?

   _____

4. How large is the Catalog file?

   _____

5. What is the start block of the first extent of the Catalog file?

   _____

6. How many blocks does this extent run?

   _____

**4. Extract the HFS+ Special Files from the Disk Image.**

- Use the TSK command `icat` extract the HFS+ special files. We know that each HFS+ special file uses a specific Catalog Node ID for each file.
  - CNID 4 points to the Catalog File
  - CNID 8 points to the Attributes File

```
$ icat -f hfs -o 40 GPT.dmg 4 > catalog_file
$ icat -f hfs -o 40 GPT.dmg 8 > attributes_file
```

**5. Parse the Catalog File**

- Open the `catalog_file` file in the hex editor of your choice.
- You can also use the Synalyze It! application for a colorized version of the catalog file.
  - Open the Synalyze It! application.
  - Open the extracted catalog file from the extraction step, `catalog_file`. Use the "`Open Other`" button.
  - Next, open the included `hfs+_catalog_node_grammar` file; this file contains the colorized offsets to help translate what is seen in the `catalog_file` file. Use `File | Open...`
  - In the window containing the `catalog_file` file, select the `HFS_Catalog_File` grammar in the `Grammar` dropdown menu located at the top of the window.
  - You should notice a colorization effect.

**6. Parse the Catalog File – Header Node**

- The Header Node is always the first node in an HFS+ special file. The header node contains four components:
  - Node Descriptor
  - Header Record
  - User Data Record - Unused

- Map Record – Same format as the Attributes File

1. Review the Node Descriptor, 14 bytes at offset 0 in the `catalog_file`.
   a. What are the values for the Forward Link and Backward Link?

   _____

   b. What is the node type in the Node Descriptor?

   _____

   c. How many records does this node contain?

   _____

2. Review the Header Record, 46 bytes at offset 14: in the `catalog_file`.
   a. How many leaf records are contained in the catalog file?

   _____

   b. What is the Node Size?

   _____

   c. How many nodes are in this `catalog_file` file?

   _____

**7. Catalog Record –Leaf Nodes & Node Descriptor**

- Leaf Nodes hold the data that most forensic analysts will be interested in: files and folders.
- Each node starts at multiples of the Node Size found in the Header Record (found in the Header Node).

1. Review the Node Descriptor for each of the three nodes after the Header Node. Which type of node are they?
   a. 14 bytes at offset 4096 (0x1000)

   _____

   b. 14 bytes at offset 8192 (0x2000)

   _____

   c. 14 bytes at offset 12288 (0x3000)

   _____

2. Review the Node Descriptor, 14 bytes at offset 4096 in the `catalog_file` for the first leaf node.
   a. What are the forward and backward link nodes for this node?

      i. Forward Link: _____

      ii. Backward Link: _____

   b. How many records does this node contain?

   _____

**8. Catalog Record – Node Keys & Records**

- Each node can have multiple keys and associated records. Review the first leaf node and its keys and records.

- Fill in the table below with the information for the first record in first node (offset 4096).
- **EXTRA CREDIT: Fill in the table below with the information for the second record in first node (offset 4096).**

| Key.... | 1st (Offset 4110 for 342 bytes) | 2nd (Offset 4452 for 348 bytes) **EXTRA CREDIT ONLY** |
|---|---|---|
| Key Length | | |
| Parent CNID | | |
| Key Length Name | | |
| Key Name (Converted Unicode) | | |
| **Record....** | | |
| Record Type | | |
| File ID | | |
| Create Date (in hex) | | |
| Content Modification Date (in hex) | | |
| Attribute Modification Date (in hex) | | |
| Access Date (in hex) | | |
| Backup Date (in hex) | | |
| Logical File Size | | |
| Total Allocated Blocks | | |
| Extent [1] Start Block | | |
| Extent [1] Block Count | | |
| Extents [2-8] are empty | | |

**9. Attributes File**

- Open the `attributes_file` file in the hex editor of your choice.
- You can also use the Synalyze It! application for a colorized version of the attributes file.
  - Open the Synalyze It! application.
  - Open the extracted attributes file from the extraction step, `attributes_file`. Use the "`Open Other`" button.
  - Next, open the included `hfs+_attributes_node_grammar` file; this file contains the colorized offsets to help translate what is seen in the `attributes_file` file. Use `File | Open...`
  - In the window containing the `attributes_file` file, select the `HFS_Attributes_File` grammar in the `Grammar` dropdown menu located at the top of the window.

- You should notice a colorization effect.

**10. Attributes File – Header Node, Node Descriptor & Header Record**

1. How many leaf records are there?

_____

2. What is the Node Size?

_____

3. How many nodes are in this `attributes_file` file?

_____

**11. Attributes File – File Attributes**

- Each node can have multiple keys and associated records. Review the first leaf node and its keys and records.
- Fill in the table below with the information for the first record in first node (offset 8192).
- **EXTRA CREDIT: Fill in the table below with the information for the second record in first node (offset 8192).**

| Key... | 1st (Offset 8206 for 290 bytes) | 2nd (Offset 8496 for 160 bytes) **EXTRA CREDIT ONLY** |
|---|---|---|
| Key Length | | |
| File CNID | | |
| Key Name Length (in bytes) | | |
| Key Name (Converted Unicode) | | |
| **Record...** | | |
| Record Type | | |
| Attribute Size | | |
| Attribute Data (Describe Contents) | | |

**Extra Credit –**

- **Keep parsing various keys and records in the Catalog and Attributes files.**

1. **Open another Terminal Window**
   - Use the `hdiutil fsid` command to view the volume header for the '`disk image`' volume. Use the Primary Volume Header.

```
$ hdiutil fsid GPT.dmg

Analyzing partition 0: Protective Master Boot Record MBR
----------------------------------------
Analyzing partition 1: GPT Header Primary GPT Header
----------------------------------------
Analyzing partition 2: GPT Partition Data Primary GPT Table
----------------------------------------
Analyzing partition 3:  Apple_Free
----------------------------------------
Analyzing partition 4: disk image Apple_HFS
HFS+
volume size                             0x0270B000 (40939520) bytes [39.0 MB]
min stretch size                        0x0087D000 (8900608) bytes [8.5 MB]
max stretch size                        0x08000000 (134217728) bytes [128 MB]
current free space                      0x024B3000 (38481920) bytes [36.7 MB]
allocation blocks                       0x0000270B (9995)
block size                              0x00001000 (4096) bytes [4 KB]
post-al-block space                     0x00000000 (0) sectors
VH (sector 2)
   signature                    H+
   version                      0x0004
   attributes                   0x80002100
   lastMountedVersion           0x4846534A (HFSJ)
   journalInfoBlock             0x00000002
   createDate                   0xCC1FA2C6 7/8/12, 8:49:10 PM EDT
   modifyDate                   0xCEB82129 11/24/13, 9:33:29 PM GMT
   backupDate                   0x00000000 1/1/04, 12:00:00 AM GMT
   checkedDate                  0xCC1FDB06 7/9/12, 12:49:10 AM GMT
   fileCount                    0x0000000D
   folderCount                  0x00000007
   blockSize                    0x00001000
   totalBlocks                  0x0000270B
   freeBlocks                   0x000024B3
   nextAllocation               0x00000124
   rsrcClumpSize                0x00010000
   dataClumpSize                0x00010000
   nextCatalogID                0x0000003E
   writeCount                   0x00000039
   encodingsBitmap              0x0000000000000001
   finderInfo
     0                          0 Blessed folder directory ID
     1                          0
     2                          0 Open folder directory ID
     3                          0 Mac OS 9 blessed folder directory ID
     4                          0
     5                          0 Mac OS X blessed folder directory ID
     VSDB Volume ID             0x679CB905565CF3D2
   allocationFile
     logicalSize          0x0000000000001000
     clumpSize            0x00001000
     totalBlocks          0x00000001
     extents              startBlock blockCount
                          0x00000001 0x00000001
   extentsFile
     logicalSize          0x000000000004E000
     clumpSize            0x0004E000
     totalBlocks          0x0000004E
     extents              startBlock blockCount
                          0x00000083 0x0000004E
   catalogFile
     logicalSize          0x000000000004E000
     clumpSize            0x0004E000
     totalBlocks          0x0000004E
```

```
    extents             startBlock blockCount
                        0x0000042B 0x0000004E
  attributesFile
    logicalSize         0x000000000004E000
    clumpSize           0x0004E000
    totalBlocks         0x0000004E
    extents             startBlock blockCount
                        0x000000D1 0x0000004E
  startupFile
    logicalSize         0x0000000000000000
    clumpSize           0x00000000
    totalBlocks         0x00000000
    extents             startBlock blockCount
Alternate VH (2 from end of HFS+ volume, sector 79958)
  signature                     H+
  version                       0x0004
  attributes                    0x80002100
  lastMountedVersion            0x31302E30 (10.0)
  journalInfoBlock              0x00000002
  createDate                    0xCC1FA2C6 7/8/12, 8:49:10 PM EDT
  modifyDate                    0xCC1FDB06 7/9/12, 12:49:10 AM GMT
  backupDate                    0x00000000 1/1/04, 12:00:00 AM GMT
  checkedDate                   0xCC1FDB06 7/9/12, 12:49:10 AM GMT
  fileCount                     0x00000002
  folderCount                   0x00000000
  blockSize                     0x00001000
  totalBlocks                   0x0000270B
  freeBlocks                    0x0000259D
  nextAllocation                0x00000785
  rsrcClumpSize                 0x00010000
  dataClumpSize                 0x00010000
  nextCatalogID                 0x00000012
  writeCount                    0x00000000
  encodingsBitmap               0x0000000000000001
  finderInfo
    0                           0 Blessed folder directory ID
    1                           0
    2                           0 Open folder directory ID
    3                           0 Mac OS 9 blessed folder directory ID
    4                           0
    5                           0 Mac OS X blessed folder directory ID
    VSDB Volume ID              0x679CB905565CF3D2
  allocationFile
    logicalSize         0x0000000000001000
    clumpSize           0x00001000
    totalBlocks         0x00000001
    extents             startBlock blockCount
                        0x00000001 0x00000001
  extentsFile
    logicalSize         0x000000000004E000
    clumpSize           0x0004E000
    totalBlocks         0x0000004E
    extents             startBlock blockCount
                        0x00000083 0x0000004E
  catalogFile
    logicalSize         0x000000000004E000
    clumpSize           0x0004E000
    totalBlocks         0x0000004E
    extents             startBlock blockCount
                        0x0000042B 0x0000004E
  attributesFile
    logicalSize         0x000000000004E000
    clumpSize           0x0004E000
    totalBlocks         0x0000004E
    extents             startBlock blockCount
                        0x000000D1 0x0000004E
  startupFile
    logicalSize         0x0000000000000000
    clumpSize           0x00000000
    totalBlocks         0x00000000
    extents             startBlock blockCount
-----------------------------------------
Analyzing partition 5:  Apple_Free
-----------------------------------------
Analyzing partition 6: GPT Partition Data Backup GPT Table
-----------------------------------------
Analyzing partition 7: GPT Header Backup GPT Header
```

1. What file system does Partition 4 'disk image' use?
    a. HFS+
2. How large is the volume?
    a. 39.0 MB
3. What is the volume block size?
    a. 4096
4. What is the signature of the volume?
    a. H+
5. When was this volume created?
    a. 7/8/12, 8:49:10 PM EDT

**2. Extract the Volume Header**

- Use the `dd` command to extract the volume header for the 'disk image' volume to a file.
- The volume header is located 1024 bytes from the beginning of the volume. In the last exercise we found that the 'disk image' volume starts at sector 40 and each sector is 512 bytes. Use the `skip=` parameter to go to this part of the disk.
- The volume header is always 512 bytes in size, use the `count=` parameter to only copy out one sector containing the volume header.
- Save it to a file named `volume_header`.

```
$ dd if=GPT.dmg skip=42 count=1 > volume_header
```

**3. Parse the Volume Header**

- Open the `volume_header` file in the hex editor of your choice.
- You can also use the Synalyze It! application for a colorized version of the volume header.
    - Open the Synalyze It! application.
    - Open the extracted volume header file from the previous step, `volume_header`.
    - Also open the included `volume_header_grammar` file, this file contains the colorized offsets to help translate what is seen in the `volume_header` file.

1. When was this volume last modified?
    a. 3468173609 or 2013-11-24 16:33:29 Sun EST
        i. 4 bytes at offset 20
        ii. Use Epoch Converter to convert timestamp
2. How many files does this volume contain?
    a. 13
        i. 4 bytes at offset 20
3. How many folders does this volume contain?
    a. 7
        i. 4 bytes at offset 32
4. How large is the Catalog file?
    a. 319488 bytes
        i. 4 bytes at offset 272

5. What is the start block of the first extent of the Catalog file?
    a. 1067
        i. 4 bytes at offset 288
6. How many blocks does this extent run?
    a. 78
        i. 4 bytes at offset 292

*Figure 1 – Volume Header Using a Basic Hex Editor (0xED Application is shown)*

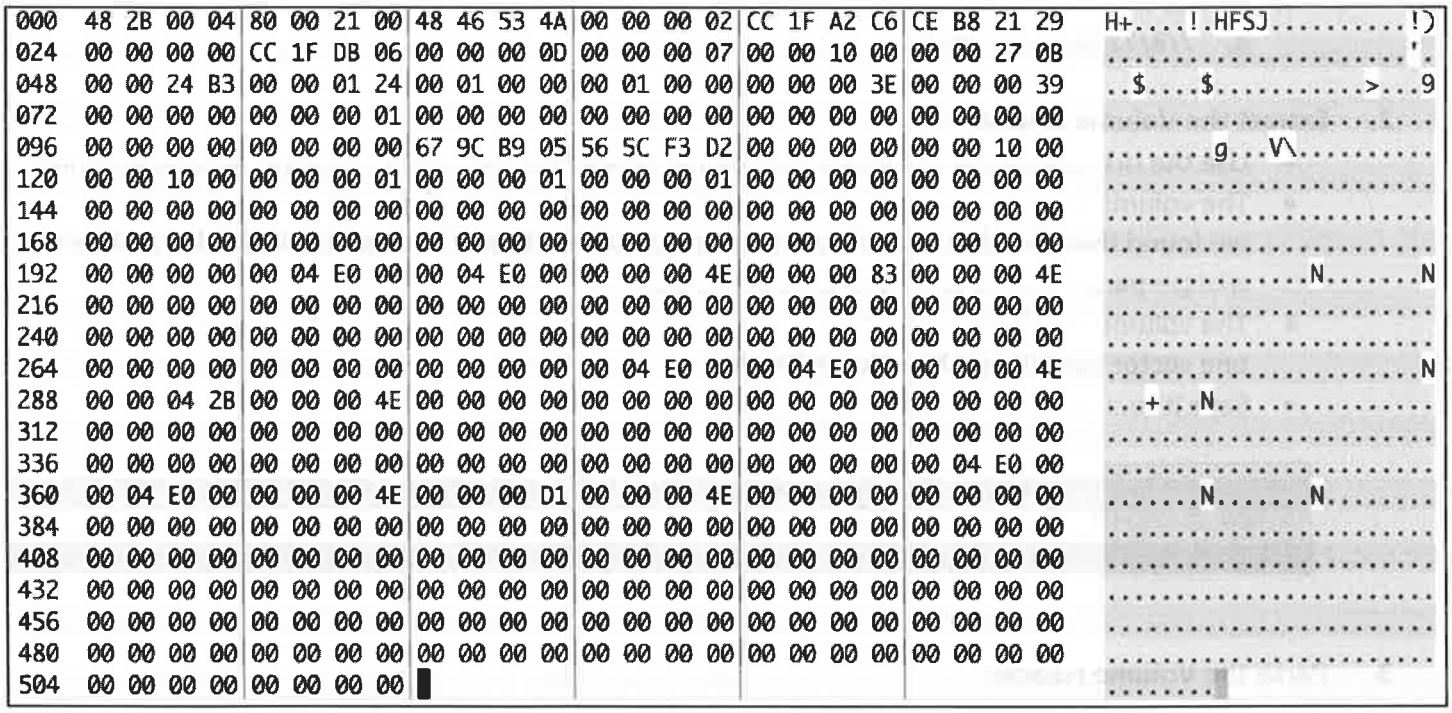
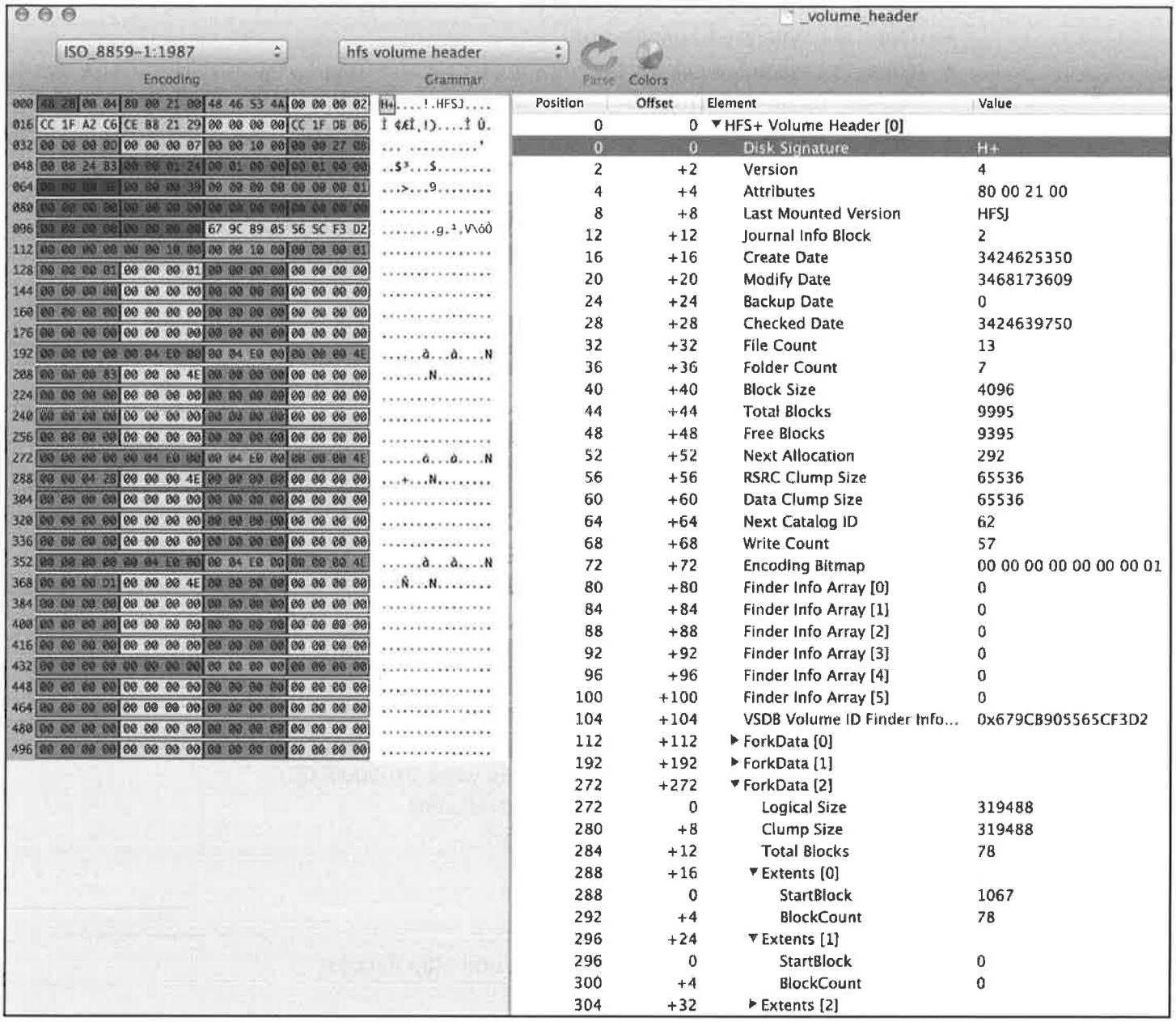


*Figure 2 - Volume Header using Synalyze It! Application & Custom Grammar*




| Position | Offset | Element | Value |
|---|---|---|---|
| 0 | 0 | ▼HFS+ Volume Header [0] | |
| 0 | 0 | Disk Signature | H+ |
| 2 | +2 | Version | 4 |
| 4 | +4 | Attributes | 80 00 21 00 |
| 8 | +8 | Last Mounted Version | HFSJ |
| 12 | +12 | Journal Info Block | 2 |
| 16 | +16 | Create Date | 3424625350 |
| 20 | +20 | Modify Date | 3468173609 |
| 24 | +24 | Backup Date | 0 |
| 28 | +28 | Checked Date | 3424639750 |
| 32 | +32 | File Count | 13 |
| 36 | +36 | Folder Count | 7 |
| 40 | +40 | Block Size | 4096 |
| 44 | +44 | Total Blocks | 9995 |
| 48 | +48 | Free Blocks | 9395 |
| 52 | +52 | Next Allocation | 292 |
| 56 | +56 | RSRC Clump Size | 65536 |
| 60 | +60 | Data Clump Size | 65536 |
| 64 | +64 | Next Catalog ID | 62 |
| 68 | +68 | Write Count | 57 |
| 72 | +72 | Encoding Bitmap | 00 00 00 00 00 00 00 01 |
| 80 | +80 | Finder Info Array [0] | 0 |
| 84 | +84 | Finder Info Array [1] | 0 |
| 88 | +88 | Finder Info Array [2] | 0 |
| 92 | +92 | Finder Info Array [3] | 0 |
| 96 | +96 | Finder Info Array [4] | 0 |
| 100 | +100 | Finder Info Array [5] | 0 |
| 104 | +104 | VSDB Volume ID Finder Info... | 0x679CB905565CF3D2 |
| 112 | +112 | ▶ForkData [0] | |
| 192 | +192 | ▶ForkData [1] | |
| 272 | +272 | ▼ForkData [2] | |
| 272 | 0 | Logical Size | 319488 |
| 280 | +8 | Clump Size | 319488 |
| 284 | +12 | Total Blocks | 78 |
| 288 | +16 | ▼Extents [0] | |
| 288 | 0 | StartBlock | 1067 |
| 292 | +4 | BlockCount | 78 |
| 296 | +24 | ▼Extents [1] | |
| 296 | 0 | StartBlock | 0 |
| 300 | +4 | BlockCount | 0 |
| 304 | +32 | ▶Extents [2] | |

**4. Extract the HFS+ Special Files from the Disk Image.**

- Use the TSK command `icat` to extract the HFS+ special files. We know that each HFS+ special file uses a specific Catalog Node ID for each file.
  - CNID 4 points to the Catalog File
  - CNID 8 points to the Attributes File

```
$ icat -f hfs -o 40 GPT.dmg 4 > catalog_file
$ icat -f hfs -o 40 GPT.dmg 8 > attributes_file
```

**5. Parse the Catalog File**

- Open the `catalog_file` file in the hex editor of your choice.
- You can also use the Synalyze It! application for a colorized version of the catalog file.
    - Open the Synalyze It! application.
    - Open the extracted catalog file from the extraction step, `catalog_file`. Use the "`Open Other`" button.
    - Next, open the included `hfs+_catalog_node_grammar` file; this file contains the colorized offsets to help translate what is seen in the `catalog_file` file. Use `File | Open...`
    - In the window containing the `catalog_file` file, select the `HFS_Catalog_File` grammar in the `Grammar` dropdown menu located at the top of the window.
    - You should notice a colorization effect.

**6. Parse the Catalog File – Header Node**

- The Header Node is always the first node in an HFS+ special file. The header node contains four components:
    - Node Descriptor
    - Header Record
    - User Data Record - Unused
    - Map Record – Same format as the Attributes File

1. Review the Node Descriptor, 14 bytes at offset 0 in the `catalog_file`.
    a. What are the values for the Forward Link and Backward Link?
        i. 0x00000000
        ii. 0x00000000
            1. Another way to check if this is a header node is to see if the backward and forward links are both have the value 0x00000000.
            2. 4 bytes at offsets 0 (0x00) and 4 (0x04)
    b. What is the node type in the Node Descriptor?
        i. Header Node (Type 1, 0x0001)
            1. One byte at offset 8 (0x08)
    c. How many records does this node contain?
        i. 3 (Header Record, User Data Record, and Map Record)
            1. 2 bytes at offset 10 (0x0A)
2. Review the Header Record, 46 bytes at offset 14: in the `catalog_file`.
    a. How many leaf records are contained in the catalog file?
        i. 42
            1. 4 bytes at offset 20 (0x14)
    b. What is the Node Size?
        i. 4096 bytes
            1. 2 bytes at offset 32 (0x20)
    c. How many nodes are in this `catalog_file` file?
        i. 78
            1. 4 bytes at offset 36 (0x24)

**7. Catalog Record –Leaf Nodes & Node Descriptor**

- Leaf Nodes hold the data that most forensic analysts will be interested in: files and folders.

- Each node starts at multiples of the Node Size found in the Header Record (found in the Header Node).

1. Review the Node Descriptor for each of the three nodes after the Header Node. Which type of node are they?
    a. 14 bytes at offset 4096 (0x1000)
        i. Leaf Node (One byte at offset 8 (0x08) of the Node Descriptor)
    b. 14 bytes at offset 8192 (0x2000)
        i. Leaf Node (One byte at offset 8 (0x08) of the Node Descriptor)
    c. 14 bytes at offset 12288 (0x3000)
        i. Index Node (One byte at offset 8 (0x08) of the Node Descriptor)
2. Review the Node Descriptor, 14 bytes at offset 4096 in the `catalog_file` for the first leaf node.
    a. What are the forward and backward link nodes for this node?
        i. Forward Link: 0
        ii. Backward Link: 2
            1. 4 bytes at offsets 4096 (0x1000) and 4100 (0x1004)
    b. How many records does this node contain?
        i. 16 Records
            1. 2 bytes at offset 4106 (0x100A)

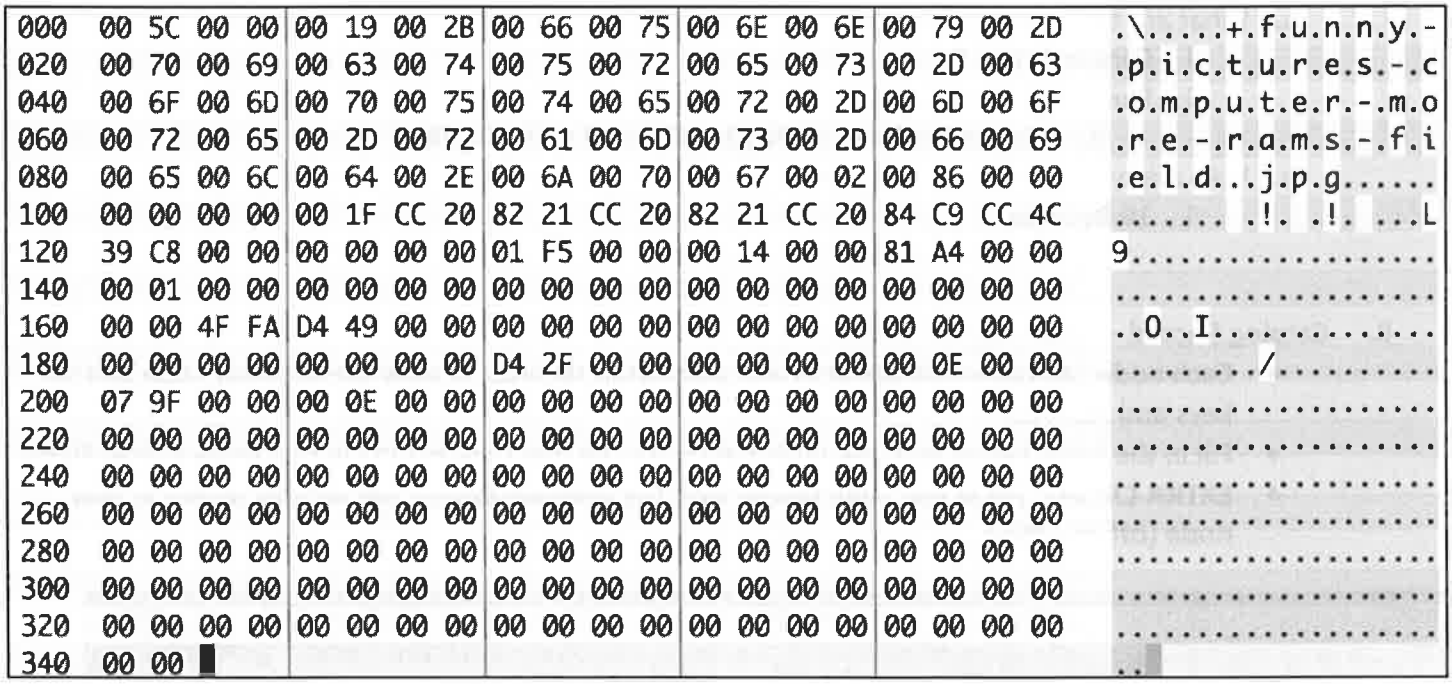
**8. Catalog Record – Node Keys & Records**

- Each node can have multiple keys and associated records. Review the first leaf node and its keys and records.
- Fill in the table below with the information for the first two records in first node (offset 4096).
- **EXTRA CREDIT: Fill in the table below with the information for the second record in first node (offset 4096).**

| Key... | 1st (Offset 4110 for 342 bytes) | 2nd (Offset 4452 for 348 bytes) **EXTRA CREDIT ONLY** |
|---|---|---|
| Key Length | 92 bytes<br>0x005C | 98 bytes<br>0x0062 |
| Parent CNID | 25<br>0x00000019 | 25<br>0x00000019 |
| Key Length Name | 43 bytes<br>0x002B<br>(86 bytes<br>UTF-16) | 46 bytes<br>0x002E<br>(92 bytes<br>UTF-16) |
| Key Name<br>(Converted Unicode) | funny-pictures-computer-more-rams-field.jpg | funny-pictures-firefox-in-on-your-computer.jpg |
| **Record...** | | |
| Record Type | File Record<br>0x0002 | File Record<br>0x0002 |
| File ID | 31<br>0x0000001F | 32<br>0x00000020 |
| Create Date (in hex) | 0xCC208221 | 0xCC208171 |
| Content Modification Date (in hex) | 0xCC208221 | 0xCC208171 |
| Attribute Modification Date (in hex) | 0xCC2084C9 | 0xCC2084C9 |

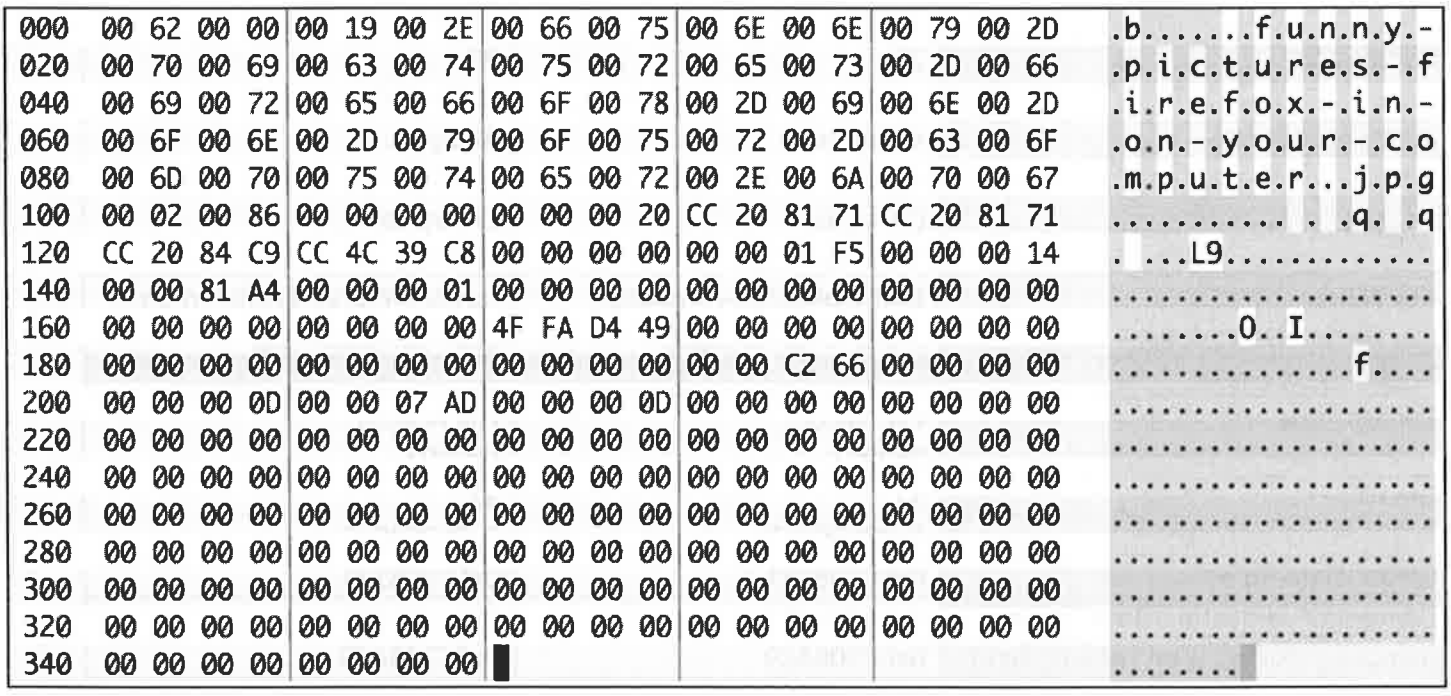
| | | |
|---|---|---|
| Access Date (in hex) | 0xCC4C39C8 | 0xCC4C39C8 |
| Backup Date (in hex) | 0x00000000 | 0x00000000 |
| Logical File Size | 54319 bytes<br>0x000000000000D42F | 49766 bytes<br>0x000000000000C266 |
| Total Allocated Blocks | 14<br>0x0000000E | 13<br>0x0000000D |
| Extent [1] Start Block | 1951<br>0x0000079F | 1965<br>0x000007AD |
| Extent [1] Block Count | 14<br>0x0000000E | 13<br>0x0000000D |
| Extents [2-8] are empty | | |

*Figure 3 - First File Record*



```
000  00 5C 00 00 00 19 00 2B 00 66 00 75 00 6E 00 6E 00 79 00 2D  .\.....+.f.u.n.n.y.-
020  00 70 00 69 00 63 00 74 00 75 00 72 00 65 00 73 00 2D 00 63  .p.i.c.t.u.r.e.s.-.c
040  00 6F 00 6D 00 70 00 75 00 74 00 65 00 72 00 2D 00 6D 00 6F  .o.m.p.u.t.e.r.-.m.o
060  00 72 00 65 00 2D 00 72 00 61 00 6D 00 73 00 2D 00 66 00 69  .r.e.-.r.a.m.s.-.f.i
080  00 65 00 6C 00 64 00 2E 00 6A 00 70 00 67 00 02 00 86 00 00  .e.l.d...j.p.g......
100  00 00 00 00 00 1F CC 20 82 21 CC 20 82 21 CC 20 84 C9 CC 4C  ....... .!. .!. ...L
120  39 C8 00 00 00 00 00 00 01 F5 00 00 00 14 00 00 81 A4 00 00  9...................
140  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
160  00 00 4F FA D4 49 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..O..I..............
180  00 00 00 00 00 00 00 00 D4 2F 00 00 00 00 00 00 00 0E 00 00  ........./..........
200  07 9F 00 00 00 0E 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  ....................
240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
260  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
280  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
300  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
320  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
340  00 00 ▮                                                      ..
```

*Figure 4 - Second File Record*



```
000  00 62 00 00 00 19 00 2E 00 66 00 75 00 6E 00 6E 00 79 00 2D  .b.......f.u.n.n.y.-
020  00 70 00 69 00 63 00 74 00 75 00 72 00 65 00 73 00 2D 00 66  .p.i.c.t.u.r.e.s.-.f
040  00 69 00 72 00 65 00 66 00 6F 00 78 00 2D 00 69 00 6E 00 2D  .i.r.e.f.o.x.-.i.n.-
060  00 6F 00 6E 00 2D 00 79 00 6F 00 75 00 72 00 2D 00 63 00 6F  .o.n.-.y.o.u.r.-.c.o
080  00 6D 00 70 00 75 00 74 00 65 00 72 00 2E 00 6A 00 70 00 67  .m.p.u.t.e.r...j.p.g
100  00 02 00 86 00 00 00 00 00 00 00 20 CC 20 81 71 CC 20 81 71  ........... . .q. .q
120  CC 20 84 C9 CC 4C 39 C8 00 00 00 00 00 00 01 F5 00 00 00 14  . ...L9.............
140  00 00 81 A4 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00  ....................
160  00 00 00 00 00 00 00 00 4F FA D4 49 00 00 00 00 00 00 00 00  ........O..I........
180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 C2 66 00 00 00 00  ..............f.....
200  00 00 00 0D 00 00 07 AD 00 00 00 0D 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  ....................
240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
260  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
280  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
300  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
320  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
340  00 00 00 00 00 00 00 00 ▮                                    ........
```

**9. Attributes File**

- Open the `attributes_file` file in the hex editor of your choice.
- You can also use the Synalyze It! application for a colorized version of the attributes file.
    - Open the Synalyze It! application.
    - Open the extracted attributes file from the extraction step, `attributes_file`. Use the "`Open Other`" button.
    - Next, open the included `hfs+_attributes_node_grammar` file; this file contains the colorized offsets to help translate what is seen in the `attributes_file` file. Use `File | Open...`
    - In the window containing the `attributes_file` file, select the `HFS_Attributes_File` grammar in the `Grammar` dropdown menu located at the top of the window.
    - You should notice a colorization effect.

**10. Attributes File – Header Node, Node Descriptor & Header Record**

1. How many leaf records are there?
    a. 9
        i. 4 bytes at offset 20 (0x14)
2. What is the Node Size?
    a. 8192
        i. 2 bytes at offset 32 (0x20)
3. How many nodes are in this `attributes_file` file?
    a. 39
        i. 4 bytes at offset 36 (0x24)

**11. Attributes File – File Attributes**

- Each node can have multiple keys and associated records. Review the first leaf node and its keys and records.
- Fill in the table below with the information for the first two records in the first node (offset 8192).
- **EXTRA CREDIT: Fill in the table below with the information for the second record in first node (offset 8192).**

| Key... | 1st (Offset 8206 for 290 bytes) | 2nd (Offset 8496 for 160 bytes) **EXTRA CREDIT ONLY** |
|---|---|---|
| Key Length | 84 bytes<br>0x0054 | 52 bytes<br>0x0034 |
| File CNID | 29<br>0x0000001D | 29<br>0x0000001D |
| Key Name Length (in bytes) | 36 bytes<br>0x0024<br>(72 bytes<br>UTF-16) | 20 bytes<br>0x0014<br>(40 bytes<br>UTF-16) |
| Key Name (Converted | com.apple.metadata:kMDItemWhereFroms | com.apple.quarantine |

| Unicode) | | |
|---|---|---|
| **Record...** | | |
| Record Type | Inline Attribute<br>0x00000010 | Inline Attribute<br>0x00000010 |
| Attribute Size | 187 bytes<br>0x000000BB | 89 bytes<br>0x00000059 |
| Attribute Data (Describe Contents) | Binary Property List containing URL http://icanhascheezburger.files.wordpress.com/2011/03/949bcd7a-6afc-4b41-a1be-c74f64815e77.jpg | Quarantine Attribute Data: 0001;4ffad124;Google\x20Chrome.app;4F68601F-85D9-4DB7-A061-244CDD2C650E\|com.google.Chrome |

*Figure 5 - First Attribute Record*

```
000  00 54 00 00 00 00 00 1D 00 00 00 00 00 24 00 63 00 6F 00 6D   .T..........$.c.o.m
020  00 2E 00 61 00 70 00 70 00 6C 00 65 00 2E 00 6D 00 65 00 74   ...a.p.p.l.e...m.e.t
040  00 61 00 64 00 61 00 74 00 61 00 3A 00 6B 00 4D 00 44 00 49   .a.d.a.t.a.:.k.M.D.I
060  00 74 00 65 00 6D 00 57 00 68 00 65 00 72 00 65 00 46 00 72   .t.e.m.W.h.e.r.e.F.r
080  00 6F 00 6D 00 73 00 00 00 10 00 00 00 00 00 00 00 00 00 00   .o.m.s..............
100  00 BB 62 70 6C 69 73 74 30 30 A2 01 02 5F 10 5E 68 74 74 70   ..bplist00..._.^http
120  3A 2F 2F 69 63 61 6E 68 61 73 63 68 65 65 7A 62 75 72 67 65   ://icanhascheezburge
140  72 2E 66 69 6C 65 73 2E 77 6F 72 64 70 72 65 73 73 2E 63 6F   r.files.wordpress.co
160  6D 2F 32 30 31 31 2F 30 33 2F 39 34 39 62 63 64 37 61 2D 36   m/2011/03/949bcd7a-6
180  61 66 63 2D 34 62 34 31 2D 61 31 62 65 2D 63 37 34 66 36 34   afc-4b41-a1be-c74f64
200  38 31 35 65 37 37 2E 6A 70 67 5F 10 29 68 74 74 70 3A 2F 2F   815e77.jpg_.)http://
220  69 63 61 6E 68 61 73 63 68 65 65 7A 62 75 72 67 65 72 2E 63   icanhascheezburger.c
240  6F 6D 2F 3F 73 3D 63 6F 6D 70 75 74 65 72 08 0B 6C 00 00 00   om/?s=computer..l...
260  00 00 00 01 01 00 00 00 00 00 00 00 03 00 00 00 00 00 00 00   ....................
280  00 00 00 00 00 00 00 00 98 00 [redacted]                      ..........
```

*Figure 6 - Second Attribute Record*

```
000  00 34 00 00 00 00 00 1D 00 00 00 00 00 14 00 63 00 6F 00 6D   .4.............c.o.m
020  00 2E 00 61 00 70 00 70 00 6C 00 65 00 2E 00 71 00 75 00 61   ...a.p.p.l.e...q.u.a
040  00 72 00 61 00 6E 00 74 00 69 00 6E 00 65 00 00 00 10 00 00   .r.a.n.t.i.n.e......
060  00 00 00 00 00 00 00 00 00 59 30 30 30 31 3B 34 66 66 61 64   .........Y0001;4ffad
080  31 32 34 3B 47 6F 6F 67 6C 65 5C 78 32 30 43 68 72 6F 6D 65   124;Google\x20Chrome
100  2E 61 70 70 3B 34 46 36 38 36 30 31 46 2D 38 35 44 39 2D 34   .app;4F68601F-85D9-4
120  44 42 37 2D 41 30 36 31 2D 32 34 34 43 44 44 32 43 36 35 30   DB7-A061-244CDD2C650
140  45 7C 63 6F 6D 2E 67 6F 6F 67 6C 65 2E 43 68 72 6F 6D 65 00   E|com.google.Chrome.
```

**Extra Credit –**

- **Keep parsing various keys and records in the Catalog and Attributes files.**

## *Exercise – Key Takeaways*

- **Knowing the various data structures of HFS+ can help determine certain file and folder attributes and extended attributes.**
- **The HFS+ file system can easily be parsed by hand (using the provided cheat sheet, of course!).**
- **Being able to parse a file system by hand can help when automated tools fail or if tool validation is necessary.**

This page intentionally left blank.

# Exercise 1.4 – BlackLight and Image Mounting

## Objectives

- Install required software for FOR518 – Mac Forensic Analysis
- Import exercise images to BlackLight
- Practice mounting lab images

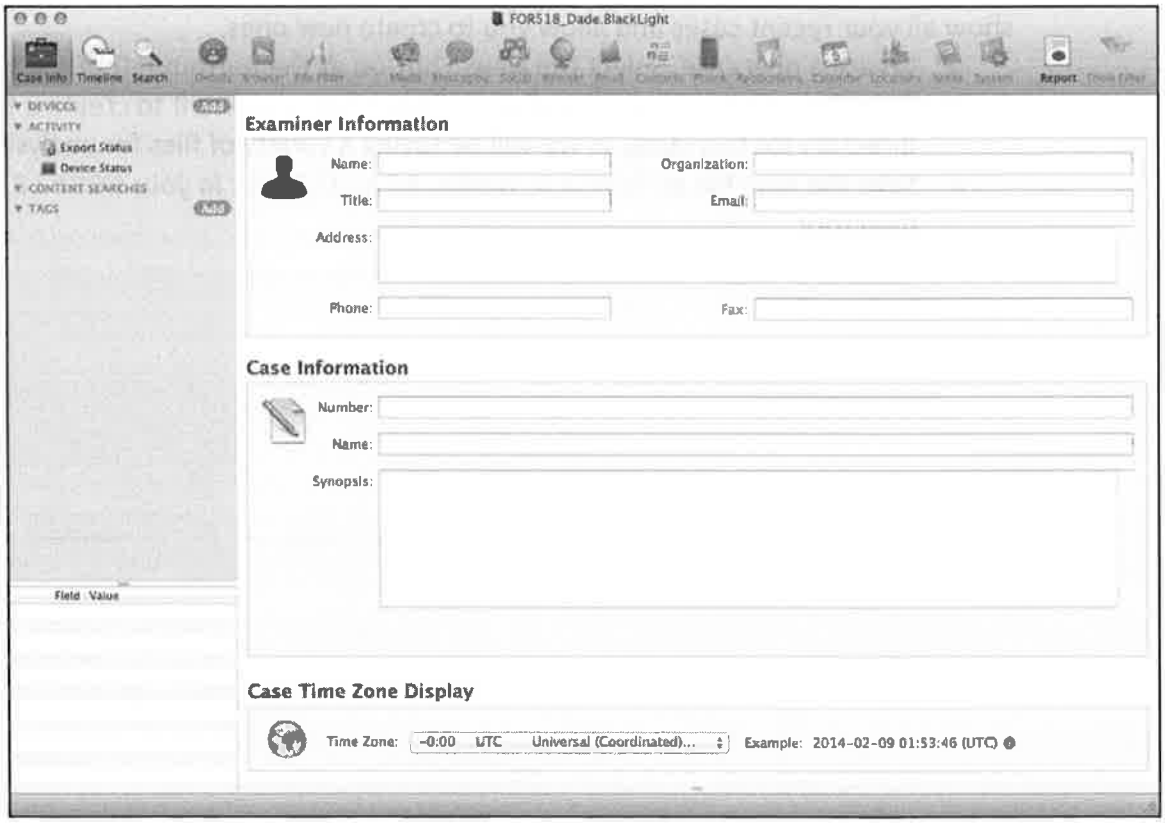
## Exercise Preparation

1. Locate the `dademurphy.e01` file on your FOR518 thumb drive in the `Exercise Images` directory.
2. **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/

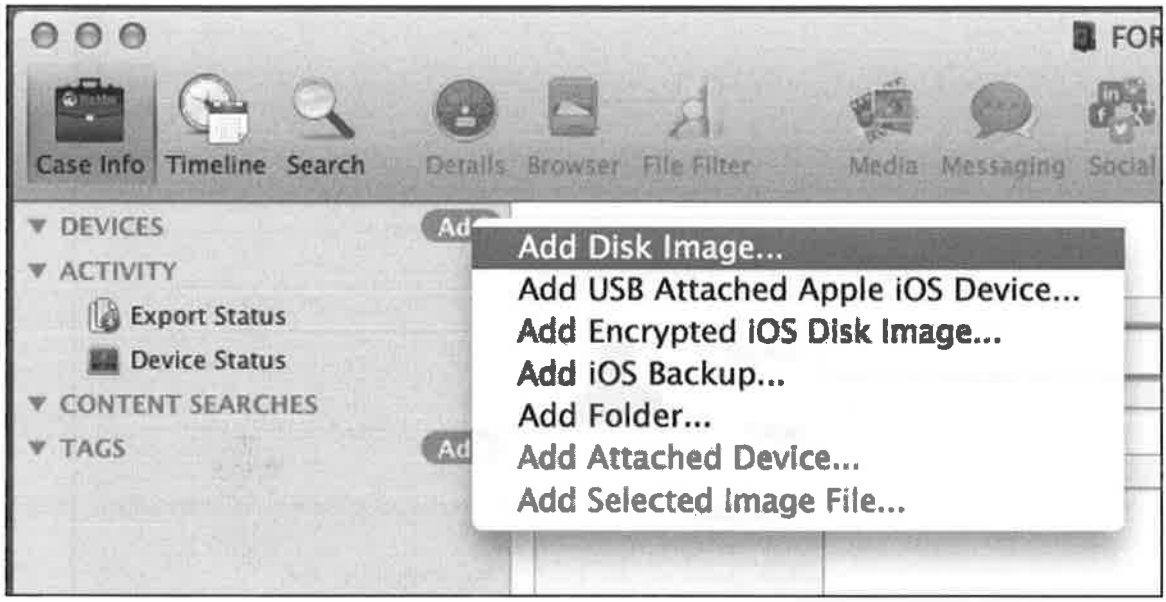
## Exercise

1. **Load Lab Image in BlackLight**
    - Locate the `dademurphy.e01` file on you FOR518 thumb drive.
        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_dade.BlackLight` in your `FOR518` directory.

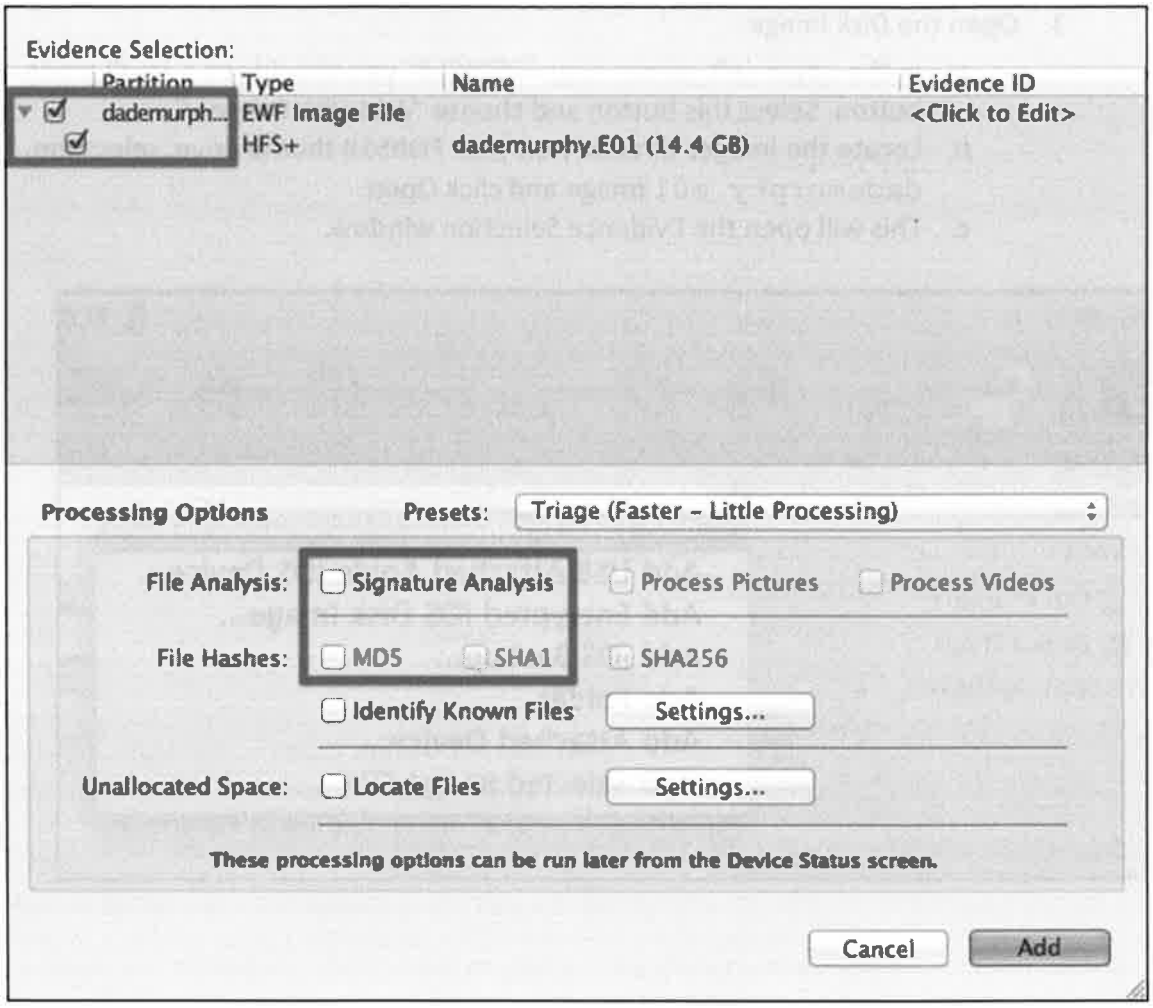

2. This should open up the BlackLight Case Info window.
    a. The Case Info tab of BlackLight allows an analyst to input case specifics and change the time zone display.



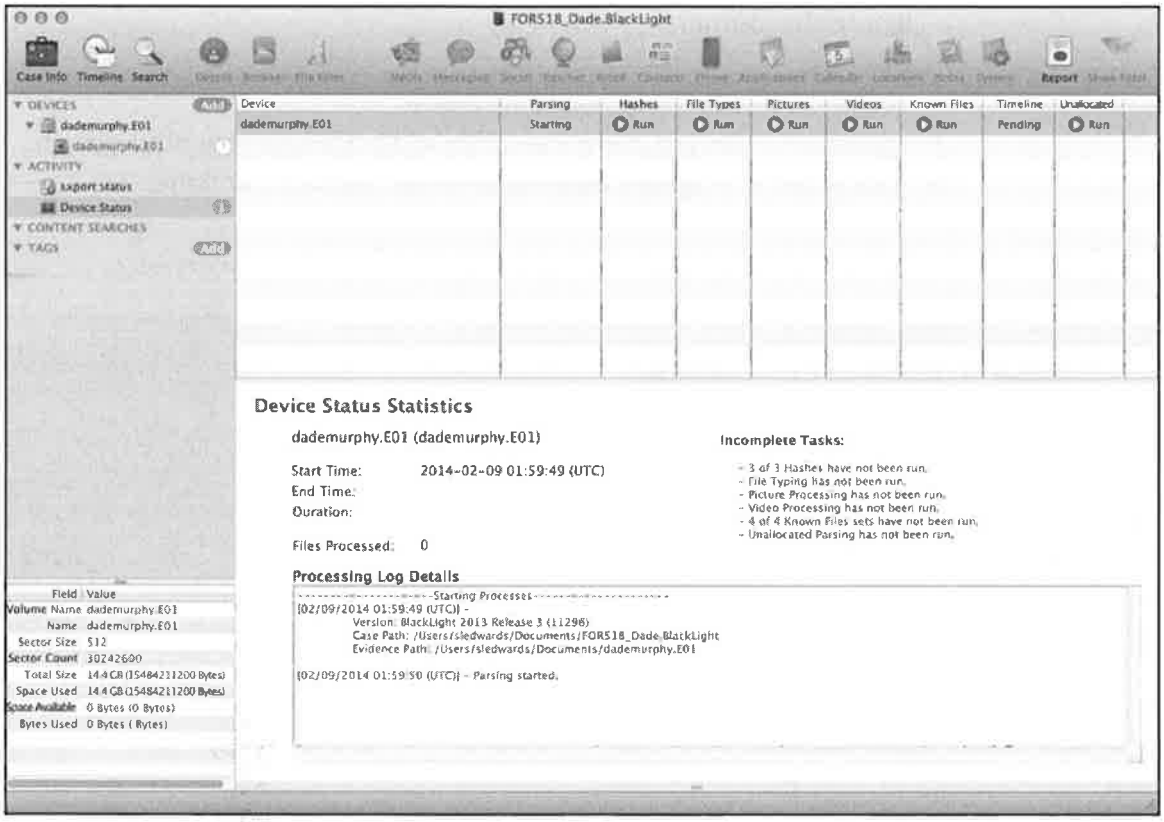
3. Open the Disk Image:
    a. In the upper-left corner near "DEVICES" you should see a small green Add button. Select this button and choose "Add Disk Image...".
    b. Locate the Images directory on your FOR518 thumb drive, select the `dademurphy.e01` image and click Open.
    c. This will open the Evidence Selection window.




4. Select your evidence:
    a. Please checkbox the HFS+ partition.
    b. In the processing options section below, please de-select Signature Analysis and MD5 hashing. You are welcome to select any of these options for later processing, but it will take some time to process.
    c. Press Add.
    d. This will start processing the disk image.




5. The Disk Status window will show how the disk processing progression. Given the selections we made in the Evidence Selection window, this should take approximately six minutes.
    a. This window is where you may run the other processes (some of which were de-selected earlier) such as Hashing, File Types, Pictures, Videos, Known Files, and Unallocated.




2. **Practice Mounting the Exercise Images from the Command-Line**
   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
     - Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
       - `Mount Point` – Newly created specifically for this image.
     - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
     - Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.

    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
    - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_image` is used in the example.
    - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_mounted` in used the example above.
    - Use `ewfmount` to mount the `dademurphy.e01` image to the `/Volumes/dadmurphy_image/` mount point.
    - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadeimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)
    - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
    - Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
        - `-j` – Ignore the journal
        - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

3. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

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.
   - Use the `diskutil eject` command on the disk you would like to eject. (This may also be done by pressing the eject button in the Finder application.)
   - Use the `mount` command to view the list of mounted disks. Find the disk that you want to unmounts (likely `/Volumes/dademurphy_image/`, if you following the naming scheme from the examples.)
   - Use the `umount` command with the mount point to unmount the disk. You may have to use the `sudo` command.)
   - ***WARNING*** - If you are in the Dade Murphy mounted image in terminal, or have a program using, you will get an error that it cannot be ejected or unmounted.

```
$ diskutil list

$ diskutil eject /dev/disk#

$ mount

$ sudo umount /Volumes/dademurphy_image
```

# Exercise 2.1 – User Data & Preferences

## Objectives

- Learn how to mount a forensic image so that it is accessible using the Terminal application.
- Get familiar with the Mac OS X user preferences and data files.
- Get familiar with property lists using Xcode.
- Get more comfortable with the OS X command line using Terminal

## 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/.
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 the Dade Murphy forensic (`dademurphy.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
     - Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
       - `Mount Point` – Newly created specifically for this image.
     - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to

suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.

- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
  - `-j` – Ignore the journal
  - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
  - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_image` is used in the example.
  - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_mounted` in used the example above.
  - Use `ewfmount` to mount the `dademurphy.e01` image to the `/Volumes/dadmurphy_image/` mount point.
  - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadeimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)
  - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
  - Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

4. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

***Exercise – Questions***

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

```
$ sudo -s

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

# cp zerocool.plist ~/FOR518

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

# open -a Xcode ~/FOR518/zerocool.plist

# exit
```

**2. Review the `zerocool`'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?

_____

**3. Open and Review the contents of the `zerocool`'s /Library/Preferences directory.**

- Use the `cd` command to change the directory to `zerocool`'s /Library/Preferences/ directory.
- Open all the property list files in Xcode. While default settings should open these in Xcode, to explicitly open them in Xcode, use the command `open -a Xcode *.plist`.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open -a Xcode *.plist
```

**4. Review the recent documents for the Preview application.**

- Review the contents of the `com.apple.Preview.LSSharedFileList.plist` file.

1. How many documents are shown as Recent Documents list?

_____

2. What file extension does the file named '`2JZcpuVh`' have? (Hint: Use a Hex Editor)

_____

3. Where is the file named '`2JZcpuVh`' located in the file system? (Hint: Use a Hex Editor)

_____

4. Where is the file 'Preview of "Mining hardware comparison - Bitcoin"' stored on the file system?

---

5. **Review the users Login Items.**
   - Review the contents of the `com.apple.loginitems.plist` file.

   1. What is the only login item listed?

   ---

   2. What is the file path for this login item?

   ---

6. **Review the Finder Preferences.**
   - Review the contents of the `com.apple.finder.plist` file.

   1. Under the `FXDesktopVolumePositions` key, how many unique volumes appear to have been mounted?

   ---

   2. Under the `FXRecentFolders` key, what is the last folder that was accessed?

   ---

7. **Review the Finder Sidebar Preferences.**
   - Review the contents of the `com.apple.sidebarlists.plist` file.

| |
|---|
| 8 - Time Machine (AFPFS), AFP File Shares, OSXFUSE Volumes |
| 16 - Network Hard Drive, iDisk, "Computer" |
| 128 - "iDisk" |
| 261 - Hard Drive, Boot Hard Drive |
| 515 - USB Flash, Time Machine Backups, Disk Image (HFS, MBR) |
| 517 - USB Hard Drive (FAT/ExFAT/HFS+) |
| 1024 - "Remote Disk" |
| 1027 - Disk Image (Bzip, VAX COFF Executable), DVD |
| 1029 - External HDD (NTFS) |

   1. What type of volume was "`Adobe Flash Player Installer`" as shown by the `favorites/VolumesList` key?

   ---

8. **Review the contents of Recent Items property list.**
   - Review the contents of the `com.apple.recentitems.plist` file.

1. What was the most recently used application?

_____

2. What is the second most recent document named?

_____

3. Where was this file stored on the file system?

_____

4. What kind of file is this?

_____

**9. Open and Review the contents of the `zerocool`'s /Library/Preferences/ByHost directory.**

- Use the `cd` command to change the directory to `zerocool`'s `/Library/Preferences/ByHost` directory.
- Open all the property list files in Xcode. To explicitly open them in Xcode, use the command `open -a Xcode *.plist`.

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/ByHost

$ open -a Xcode *.plist
```

**10. Review the Bluetooth Settings**

- Review the contents of the `com.apple.Bluetooth.40A90B07-FC53-52C8-A774-6F1A5E659E9C.plist` file.

1. What is the MAC address of the only device setup with this system?

_____

2. When was this device last connected?

_____

**11. Review the `zerocool` user's `.bash_history` file.**

- Use the `cd` command to change directory to the `zerocool`'s home directory.
- Use the `cat` command to view the contents of the `.bash_history` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/

$ cat .bash_history
```

1. What command showed the IP address of the system to the user?

_____

12. **Review the `zerocool`'s `Trash`.**
    - Use the `cd` command to change directory to the `zerocool`'s .Trash directory.
    - Use the `ls -l` command to view the contents of this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/.Trash

$ ls -l
```

1. What two files are in the trash?

_____

_____

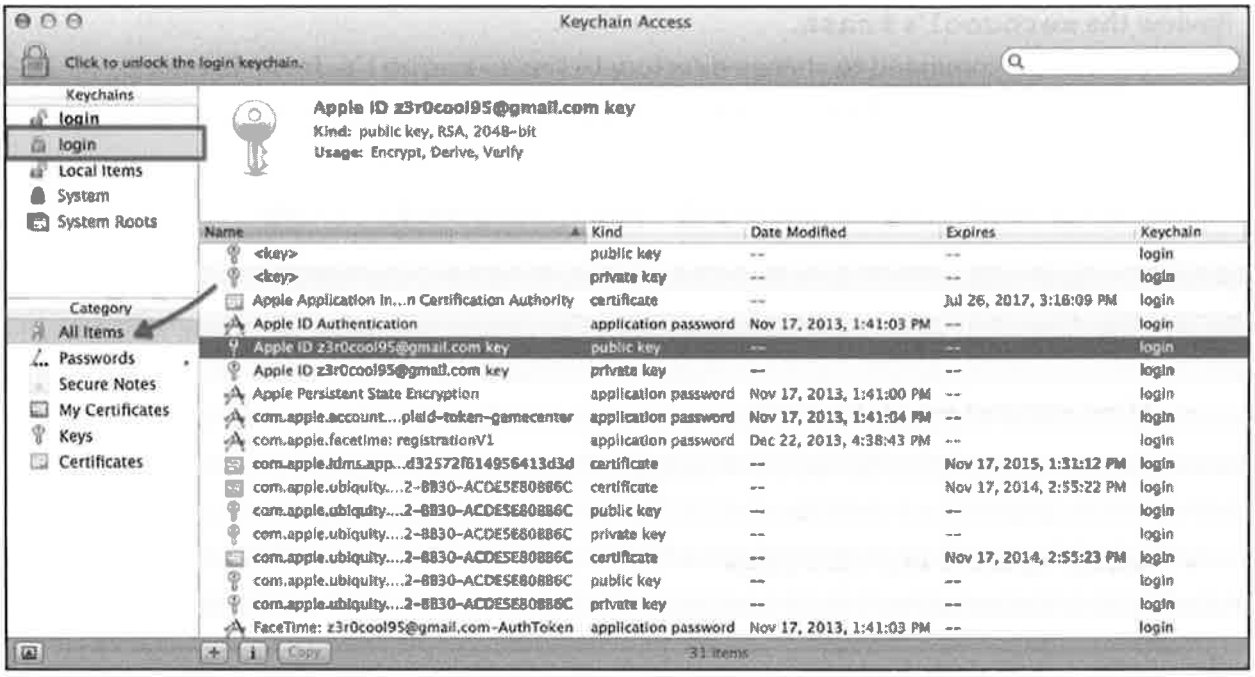
2. Where did these files once exist?

_____

13. **Review the `zerocool`'s `Keychains`.**
    - Use the `cd` command to change directory to the `zerocool`'s Keychain directory.
    - Use the `file` command to view the file type for the files in this directory.
    - Use the open command to view the contents of the `login.keychain` using `Keychain Access.app`

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Keychains

$ file *

$ open login.keychain
```

- Once Keychain Access.app has been opened, view the login.keychain. On the left-hand side choose the correct login keychain under "Keychains". The `login.keychain` in bold is <u>your</u> keychain – choose the non-bold login keychain.
- In the "Category" section choose "All Items". This will display all keychain items in the main viewing pane.




1. What email address is used for the iTunes store?

_____

2. If this entry is double-clicked, and the "Show password" checkbox is checked – are you able to see the password?

_____

**14. Review the `zerocool`'s Downloads directory.**

- Use the `cd` command to change directory to the `zerocool`'s Downloads directory

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Downloads/

$ ls -l
```

1. What file was downloaded December 15th (UTC)? *(Hint: Extended Attributes)*

_____

2. What application downloaded this file? *(Hint: Extended Attributes)*

_____

3. What kind of file is this?

_____

**Extra Credit –**
**Keep reviewing various property lists in the Preferences directories, there might be some interesting data in there!**

*Exercise – Step-By-Step*

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

```
$ sudo -s

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

# cp zerocool.plist ~/FOR518

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

# open -a Xcode ~/FOR518/zerocool.plist

# exit
```

2. **Review the `zerocool`'s user property list.**
   1. What is the user's "Real Name"?
      - Dade Murphy
   2. What is the path to the user's home directory?
      - /Users/zerocool
   3. What is the user's UID?
      - 501

3. **Open and Review the contents of the `zerocool`'s /Library/Preferences directory.**
   - Use the `cd` command to change the directory to `zerocool`'s /Library/Preferences/ directory.
   - Open all the property list files in Xcode. While default settings should open these in Xcode, to explicitly open them in Xcode, use the command `open -a Xcode *.plist`.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open -a Xcode *.plist
```

4. **Review the recent documents for the Preview application.**

- Review the contents of the `com.apple.Preview.LSSharedFileList.plist` file.

1. How many documents are shown as Recent Documents list?
    a. Nine (Items 0-8)
2. What file extension does the file named '`2JZcpuVh`' have? (Hint: Use a Hex Editor)
    a. JPG
        i. Extract the Bookmark data blog and view with a hex editor
3. Where is the file named '`2JZcpuVh`' located in the file system? (Hint: Use a Hex Editor)
    a. "/users/zerocool/pictures/"
        i. Extract the Bookmark data blog and view with a hex editor
4. Where is the file '`Preview of "Mining hardware comparison - Bitcoin"`' stored on the file system?
    a. "/users/zerocool/library/mobile documents/com~apple~preview/documents/"
        i. This is an iCloud mobile document.
        ii. Extract the Bookmark data blog and view with a hex editor

**5. Review the users Login Items.**

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

1. What is the only login item listed?
    a. iTunesHelper.app
2. What is the file path for this login item?
    a. /Applications/iTunes.app/Contents/MacOS/iTunesHelper.app
        i. Extract the Alias data blob and view in a hex editor.

**6. Review the Finder Preferences.**

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

1. Under the `FXDesktopVolumePositions` key, how many unique volumes appear to have been mounted?
    a. Eight
2. Under the `FXRecentFolders` key, what is the last folder that was accessed?
    a. The DATA folder, we can tell it is a volume by extracting the `file-bookmark` data and viewing it in a hex editor.

**7. Review the Finder Sidebar Preferences.**

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

| |
|---|
| 8 - Time Machine (AFPFS), AFP File Shares, OSXFUSE Volumes |
| 16 - Network Hard Drive, iDisk, "Computer" |
| 128 - "iDisk" |
| 261 - Hard Drive, Boot Hard Drive |
| 515 - USB Flash, Time Machine Backups, Disk Image (HFS, MBR) |
| 517 - USB Hard Drive (FAT/ExFAT/HFS+) |
| 1024 - "Remote Disk" |
| 1027 - Disk Image (Bzip, VAX COFF Executable), DVD |
| 1029 - External HDD (NTFS) |

1. What type of volume was "Adobe Flash Player Installer" as shown by the favorites/VolumesList key?
    a. EntryType = 1,027 = Disk Image

**8. Review the contents of Recent Items property list.**

- Review the contents of the com.apple.recentitems.plist file.

1. What was the most recently used application?
    a. Macquisition (Item 0)
2. What is the second most recent document named?
    a. "Bitcoin shopping list!"
3. Where was this file stored on the file system?
    a. /users/zerocool/documents/
4. What kind of file is this?
    a. Pages Document

**9. Open and Review the contents of the zerocool's /Library/Preferences/ByHost directory.**

- Use the cd command to change the directory to zerocool's /Library/Preferences/ByHost directory.
- Open all the property list files in Xcode. To explicitly open them in Xcode, use the command open -a Xcode *.plist.

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/ByHost

$ open -a Xcode *.plist
```

**10. Review the Bluetooth Settings**

- Review the contents of the com.apple.Bluetooth.40A90B07-FC53-52C8-A774-6F1A5E659E9C.plist file.

1. What is the MAC address of the only device setup with this system?
    a. 00-24-ef-be-dc-07 (RecentDevices or FavoriteDevices key)
2. When was this device last connected?
    a. Dec 22, 2013, 5:05:58 PM (EST/EDT) (RecentDevices key)

**11. Review the zerocool user's .bash_history file.**

- Use the cd command to change directory to the zerocool's home directory.
- Use the cat command to view the contents of the .bash_history file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/

$ cat .bash_history
```

1. What command showed the IP address of the system to the user?
    a. `ifconfig` (`ipconfig` is a different command on OS X systems)

**12. Review the `zerocool`'s `Trash`.**

- Use the `cd` command to change directory to the `zerocool`'s .Trash directory.
- Use the `ls -l` command to view the contents of this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/.Trash

$ ls -l
```

1. What two files are in the trash?
    a. Peercoin Logo Files.zip
    b. pooler-cpuminer-2.3.2-osx64.zip

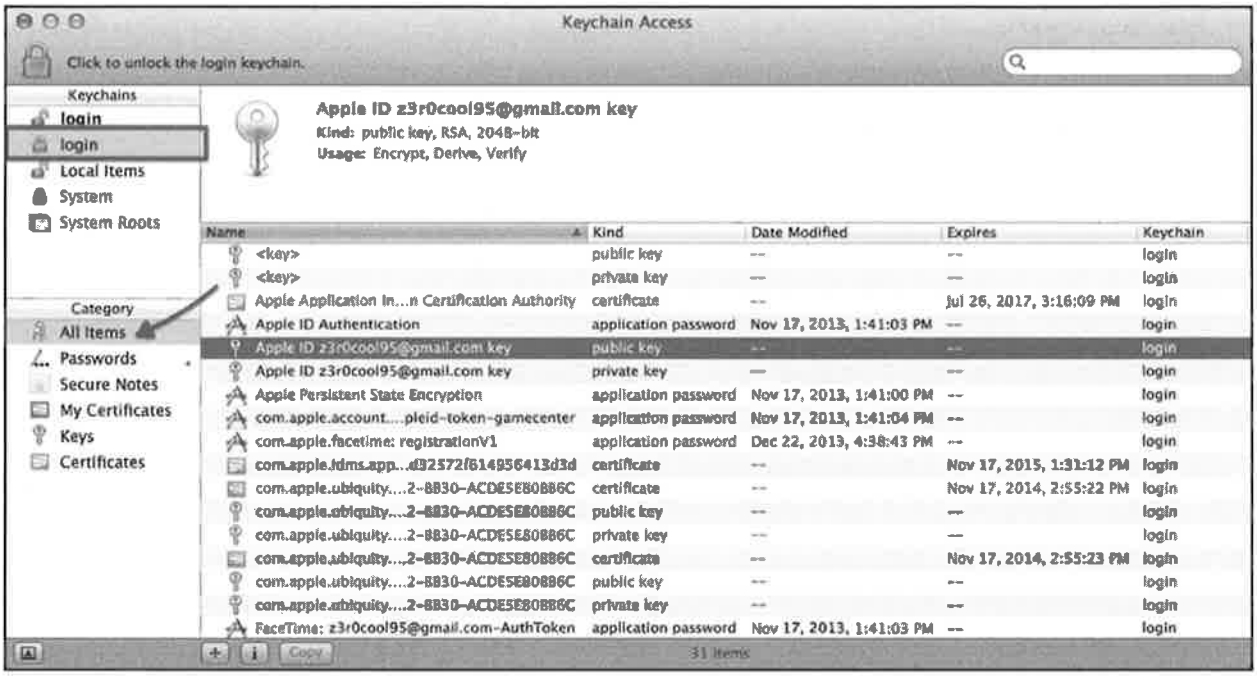
2. Where did these files once exist?
    a. /Users/zerocool/Downloads directory (view the `.DS_store` file in a Hex Editor)

**13. Review the `zerocool`'s `Keychains`.**

- Use the `cd` command to change directory to the `zerocool`'s Keychain directory.
- Use the `file` command to view the file type for the files in this directory.
- Use the open command to view the contents of the `login.keychain` using `Keychain Access.app`

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Keychains

$ file *

$ open login.keychain
```

- Once Keychain Access.app has been opened, view the login.keychain. On the left-hand side choose the correct login keychain under "Keychains". The `login.keychain` in bold is <u>your</u> keychain – choose the non-bold login keychain.
- In the "Category" section choose "All Items". This will display all keychain items in the main viewing pane.




1. What email address is used for the iTunes store?
    a. z3r0cool95@gmail.com
2. If this entry 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 to see these passwords.

**14. Review the `zerocool`'s Downloads directory.**

- Use the `cd` command to change directory to the `zerocool`'s Downloads directory

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Downloads/

$ ls -l
```

1. What file was downloaded December 15th (UTC)? *(Hint: Extended Attributes)*
    a. Firefox 26.0.dmg
2. What application downloaded this file? *(Hint: Extended Attributes)*
    a. Safari Web Browser
        i. Check the file metadata by using the `xattr -xl` command.
3. What kind of file is this?
    a. Bzip2 Compressed Data
        i. Use the `file` command to determine file type.

**Extra Credit –**
**Keep reviewing various property lists in the Preferences directories, there might be some interesting data in there!**

## *Exercise – Key Takeaways*

- **Review the contents of user preferences property lists and data files.**
- **Know the differences between what data is available for users versus the system.**

# Exercise 2.2 – Safari

## Objectives

- Introduce the key data files associated with the Safari Web Browser.
- Parse these data files using native, free, and commercial toolsets.
- Recognize differences in tool output versus raw data.

## 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/.
   - Safari Binary Cookie Parser
     i. Find the Safari Binary Cookie Parser (`bc_parser_v1_2.py`), available in the `Exercise Files/Exercise 2.2` directory on your FOR518 USB drive.
     ii. This program is also available at http://az4n6.blogspot.com/p/downloads.html.
   - SQLite Database Browser
     i. You will be using the SQLite Database Browser (Applications/sqlitebrowser.app)
     ii. This 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. **Exercise File Preparation** – Locate the files located in the `Exercise Files/Exercise 2.2 - Safari` directory on your FOR518 USB drive.
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. **Mount the Dade Murphy forensic (`dademurphy.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.

- Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
    - `Mount Point` – Newly created specifically for this image.
- Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
    - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_image` is used in the example.
    - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_mounted` in used the example above.
    - Use `ewfmount` to mount the `dademurphy.e01` image to the `/Volumes/dadmurphy_image/` mount point.
    - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadeimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)

- Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

5. **Sanity Check**
    - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

## *Exercise – Questions*

1. **Review the Safari Preferences**
   - Use the `cd` command to change the directory to `zerocool`'s /Library/Preferences/ directory.
   - Use the `open` command to open the `com.apple.Safari.plist` property list file in Xcode.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

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

   1. What was the last item searched for? (Hint: Item 0 is the newest item.)

      _____

2. **Review the Safari Web Browser Files**
   - Use the `cd` command to change the directory to `zerocool`'s /Library/Safari/ directory.
   - Use the `open` command to open all the *.plist files in this directory using Xcode.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Safari/

$ open -a Xcode *.plist
```

3. **Review the Safari Bookmarks**
   - Review the `Bookmarks.plist` file.

4. **Review the Safari Downloads**
   - Review the `Downloads.plist` file.

   1. How many items are listed in the Safari download list?

      _____

   2. How many bytes in size is the `bitcoin-0.8.6-macosx.dmg` file?

      _____

   3. What URL was the file `python-2.7.6.msi` downloaded from?

      _____

5. **Review the Safari Internet History**
   - Review the `History.plist` file.

1. When was the page titled "Captain Ridley's Shooting Party" visited?

_____

2. What is the domain of this visited web page?

_____

3. How many times was this page visited?

_____

6. **Review the Safari Last Session Settings**
   - Review the `LastSession.plist` file.

1. How many tabs were kept open when this system was imaged?

_____

2. What was being searched for on this tab?

_____

3. What was the page displayed before a search was performed?

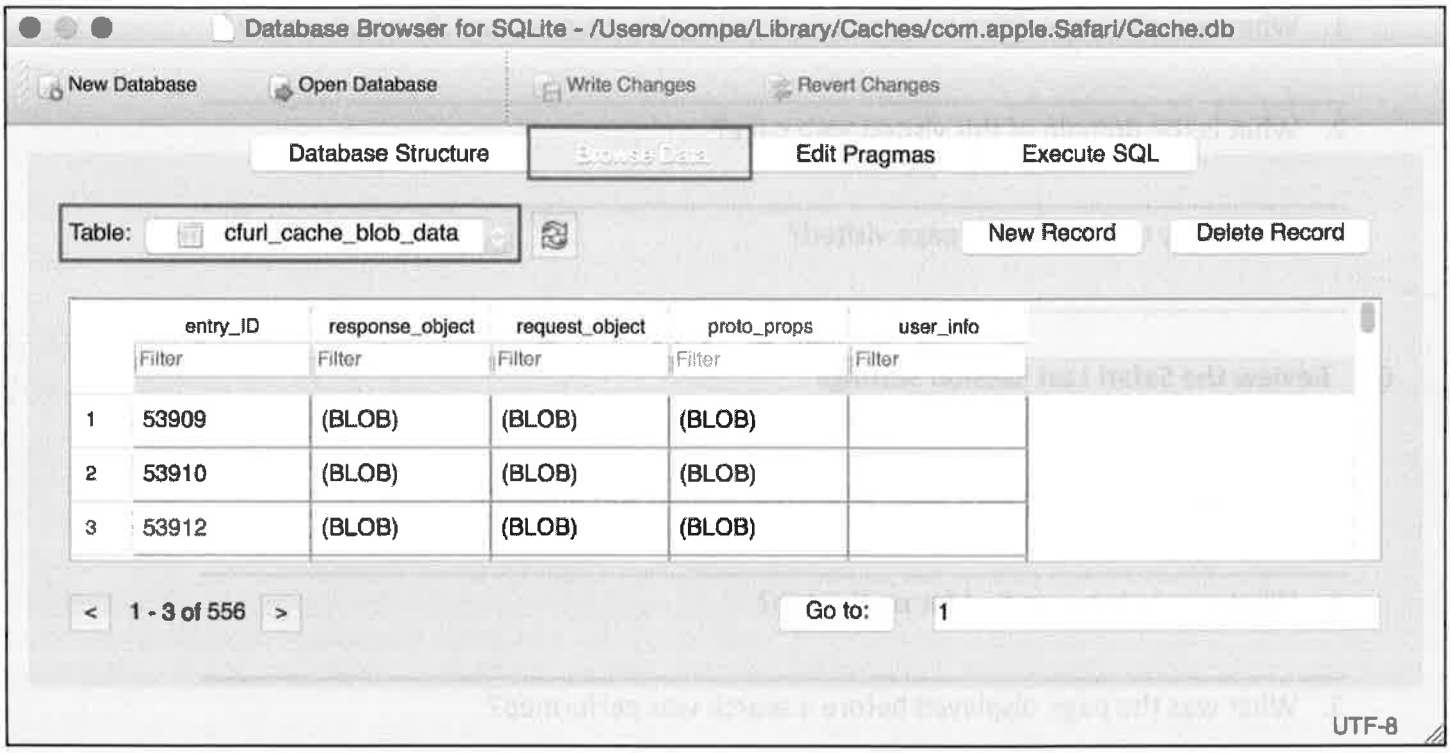
_____

7. **Open the Safari Cache Database**
   - Use the `cd` command to change the directory to `zerocool`'s `/Library/Caches/com.apple.Safari/` directory.
   - Copy the Cache.db database files to your FOR518 directory (the *-shm and *-wal files are part of the SQLite database and may contain additional database transactions):
     - `Cache.db`
     - `Cache.db-shm`
     - `Cache.db-wal`

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Caches/com.apple.Safa
ri/

$ cp Cache* ~/FOR518
```

   - Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
   - Go to the "`File`" menu and open the "`Open Database`".

8. **Review the Safari Cache Database**
   - Select "Browse Data".
   - Review the SQLite tables in Table dropdown menu.




- Select the "`cfurl_cache_response`" table.
- Find the record with the "`entry_ID`" of 335. You may have to use the arrows at the bottom to find this entry.

1. What type of file does this entry contain?

_____

2. When was this file cached?

_____

3. What URL domain was this file found on?

_____

- "Cancel" this pop-up window, and select the table named "`cfurl_cache_receiver_data`".
- Find and Select the record with the "`entry_ID`" of 335. You may have to use the arrows at the bottom to find this entry.
- Double-click the "`receiver_data`" to open the BLOB data.

4. Open this file; what is this picture of?

_____

9. **Open the Safari Cookies**
   - Use the `cd` command to change the directory to `zerocool`'s /Library/Cookies/ directory.
   - Use the Safari Binary Cookie Parser (`bc_parser_v1_2.py`), available in the `Exercise Files/Exercise 2.2 - Safari` directory.

i. This should create four files, one for the cookies and three others for Google Analytics cookies.

- Use the `open` command to open the output from the parser, this should be default open it in the TextEdit application, otherwise use `open -a TextEdit bc_output.tsv`.
- Review the contents of this output.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Cookies/

$ python bc_parser_v1_2.py -f Cookies.binarycookies -o
~/FOR518/bc_output.tsv

$ open ~/FOR518/bc_output.tsv
```

10. *****EXTRA CREDIT*** - Safari Analysis with BlackLight**
    - Review these same files under the "Internet" section.


i. **Top Sites**

ii. **History**

1. What URL has the most visits?

_____

2. What URL was visited on 11/20/13 at 12:03:02 PM (UTC)

_____

iii. **Last Session**

1. Note the absence of the last session data; always know your tool limitations.

iv. **Downloads**

1. What is the filename of the largest file downloaded?

_____

2. Were all files downloaded to the Downloads directory?

_____

v. **Bookmarks**

1. What two bookmarks are not shown in this output that were found in the `Bookmarks.plist` file. Note, you can view the property list file in the viewer below using the "Preview" tab.

_____

_____

vi. **Cache**

1. Review the Cache, try to find the file we viewed before from the Cache.db SQLite database (entry_ID 335).
2. Note the timestamps of the file, what might cause this slight difference?

_____

_____

_____

vii. **Preferences**

1. Note the parsed contents of the `com.apple.Safari.plist`, it only shows the recent search strings. Remember not all data items are parsed by tools, going to the raw data might be required in your investigations.

**Extra Credit –**

- **Keep exploring the property lists and SQLite databases associated with the Safari browser.**
- **Explore the nuances with these files in BlackLight.**
- **Explore your own Safari files!**

## *Exercise – Step-By-Step*

1. **Review the Safari Preferences**
   a. Use the `cd` command to change the directory to `zerocool`'s /Library/Preferences/ directory.
   b. Use the `open` command to open the `com.apple.Safari.plist` property list file in Xcode.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

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

1. What was the last item searched for? (Hint: Item 0 is the newest item.)
   a. "bitcoin rig" (`RecentSearchStrings` key)

2. **Review the Safari Web Browser Files**
   a. Use the `cd` command to change the directory to `zerocool`'s /Library/Safari/ directory.
   b. Use the `open` command to open all the *.plist files in this directory using Xcode.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Safari/

$ open -a Xcode *.plist
```

3. **Review the Safari Bookmarks**
   - Review the `Bookmarks.plist` file.

4. **Review the Safari Downloads**
   - Review the `Downloads.plist` file.

1. How many items are listed in the Safari download list?
   a. Ten (Note: This is just the current list, it may have been cleared)
2. How many bytes in size is the `bitcoin-0.8.6-macosx.dmg` file?
   a. 13,432,650 bytes
3. What URL was the file `python-2.7.6.msi` downloaded from?
   a. http://www.python.org/ftp/python/2.7.6/python-2.7.6.msi

5. **Review the Safari Internet History**
   - Review the `History.plist` file.

1. When was the page titled "`Captain Ridley's Shooting Party`" visited?
   a. 2013-12-22 22:24:11 Sun UTC (use Epoch Converter to convert WebKit time (409443851.2) to a human-readable timestamp)
2. What is the domain of this visited web page?
   a. www.bletchleypark.org.uk

3. How many times was this page visited?
    a. Once

**6. Review the Safari Last Session Settings**

- Review the `LastSession.plist` file.

1. How many tabs were kept open when this system was imaged?
    a. One
2. What was being searched for on this tab?
    a. "bitcoin rig"
3. What was the page displayed before a search was performed?
    a. The default front page, Top Sites (topsites://)
    b. Extract the "`SessionState`" data and view in a hex editor. Be sure to remove the first four bytes so that it is formatted as a proper plist file.

**7. Open the Safari Cache Database**

a. Use the `cd` command to change the directory to `zerocool`'s `/Library/Caches/com.apple.Safari/` directory.

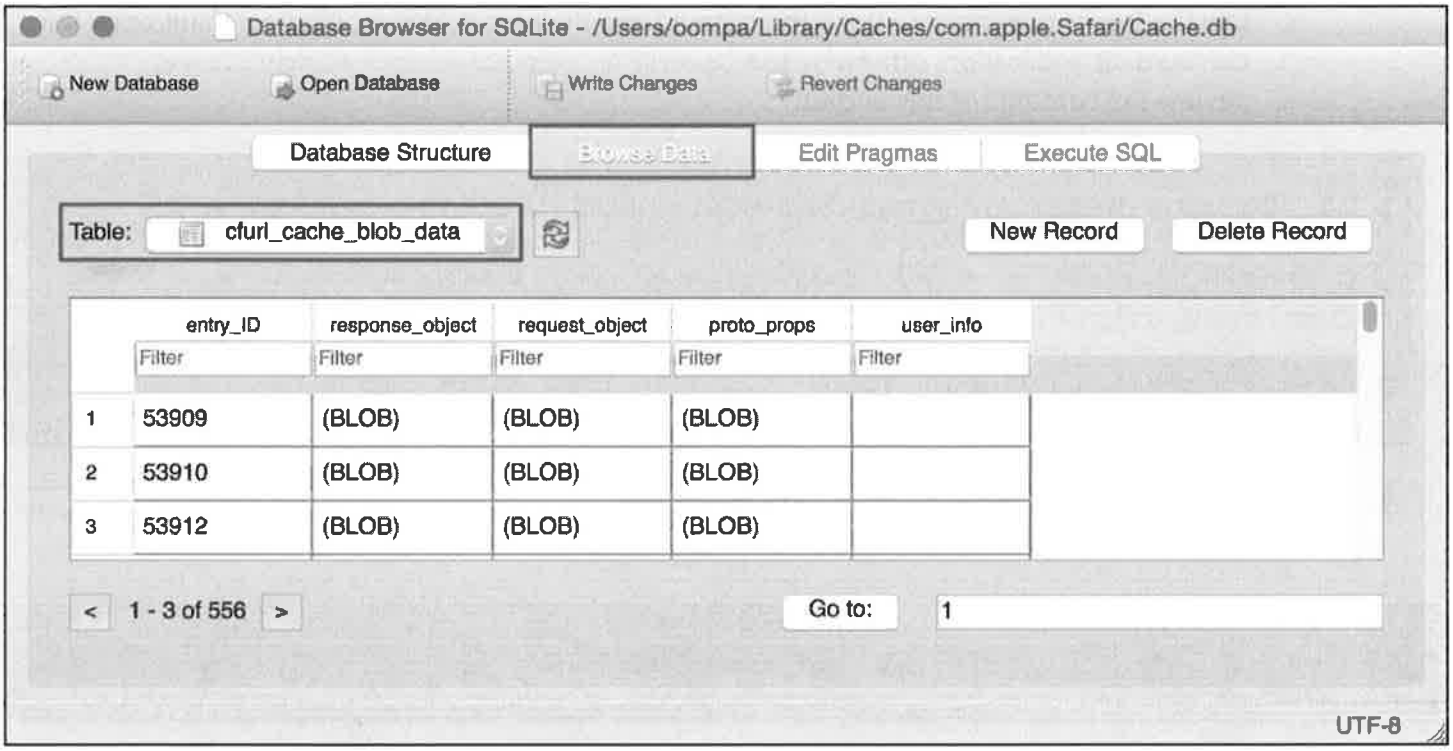
- Copy the Cache.db database files to your FOR518 directory (the *-shm and *-wal files are part of the SQLite database and may contain additional database transactions):
    - `Cache.db`
    - `Cache.db-shm`
    - `Cache.db-wal`

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Caches/com.apple.Safa
ri/

$ cp Cache* ~/FOR518
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".

**11. Review the Safari Cache Database**

- Select "Browse Data".
- Review the SQLite tables in Table dropdown menu.




- Select the "`cfurl_cache_response`" table.

b. Find the record with the "`entry_ID`" of 335. You may have to use the arrows at the bottom to find this entry.

1. What type of file does this entry contain?
    a. JPG
2. When was this file cached?
    a. 2013-12-22 21:37:49 (EST/EDT)
3. What URL domain was this file found on?
    a. teslamotors.com

c. "Cancel" this pop-up window, and select the table named "`cfurl_cache_receiver_data`".
d. Find and Select the record with the "`entry_ID`" of 335. You may have to use the arrows at the bottom to find this entry.
e. Double-click the "`receiver_data`" to open the BLOB data.

4. Open this file; what is this picture of?
    a. Black Model S Tesla Car

**8. Open the Safari Cookies**

a. Use the `cd` command to change the directory to `zerocool`'s /Library/Cookies/ directory.
b. Use the Safari Binary Cookie Parser (`bc_parser_v1_2.py`), available in the `Exercise Files/Exercise 2.2 - Safari` directory.
    i. This should create four files, one for the cookies and three others for Google Analytics cookies.

c. Use the 'open' command to open the output from the parser, this should be default open it in the TextEdit application, otherwise use `open -a TextEdit bc_output.tsv`.
d. Review the contents of this output.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Cookies/

$ python bc_parser_v1_2.py -f Cookies.binarycookies -o
~/FOR518/bc_output.tsv

$ open ~/FOR518/bc_output.tsv
```

9. ***EXTRA CREDIT*** - Safari Analysis with BlackLight**
   a. Review these same files under the "Internet" section.


i. **Top Sites**
ii. **History**
   1. What URL has the most visits?
      a. https://www.facebook.com/BBCFuture
   2. What URL was visited on 11/20/13 at 12:03:02 PM (UTC)
      a. topgear.com/uk/
iii. **Last Session**
iv. **Downloads**
   1. What is the filename of the largest file downloaded?
      a. Pages.pdf
   2. Were all files downloaded to the Downloads directory?
      a. No, one file was downloaded to the Pictures directory.
v. **Bookmarks**
   1. What two bookmarks are not shown in this output that were found in the `Bookmarks.plist` file. Note, you can view the property list file in the viewer below using the "Preview" tab.
      a. https://en.bitcoin.it/wiki/Mining_rig
      b. http://www.microcenter.com/site/content/CatalogB.aspx?rd=1
vi. **Cache**
   1. Review the Cache; try to find the file we viewed before from the Cache.db SQLite database (entry_ID 335).
   2. Note the timestamps of the file, what might cause this slight difference?
      a. The timestamps are being extracted from the `response_object` data BLOB, rather than the timestamp associated with the `cfurl_cache_response` table.
      b. It is good to know where a tool pulls the data from so these nuances can be explained later.

vii. **Preferences**

1. Note the parsed contents of the `com.apple.Safari.plist`, it only shows the recent search strings. Remember not all data items are parsed by tools, going to the raw data might be required in your investigations.

**Extra Credit –**

- **Keep exploring the property lists and SQLite databases associated with the Safari browser.**
- **Explore the nuances with these files in BlackLight.**
- **Explore your own Safari files!**

*Exercise – Key Takeaways*

- **Know the key files associated with the Safari web browser and where to find them in the file system.**
- **Know what you tools are parsing and what they are showing (or not showing) you.**
- **Get comfortable using the BlackLight application.**

This page intentionally left blank.

# Exercise 2.3 – Apple Mail

## Objectives

- Review the key files associated with Apple Mail.
- Get familiar with the Mac OS X command line.

## 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 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. **Mount the Dade Murphy forensic (`dademurphy.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
     - Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
    - `Mount Point` – Newly created specifically for this image.
- Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
    - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_image` is used in the example.
    - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_mounted` in used the example above.
    - Use `ewfmount` to mount the `dademurphy.e01` image to the `/Volumes/dadmurphy_image/` mount point.
    - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadeimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)
    - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.

- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
  - `-j` – Ignore the journal
  - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

4. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

***Exercise – Questions***

1. **Apple Mail Directory**
   - Use the cd command to change directory to the `zerocool`'s Apple Mail directory `/Users/zerocool/Library/Mail/V2/`.
   - Use the '`ls -l`' command to view the files in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Mail/V2/

$ ls -l
```

1. How many email accounts appear to be associated with this user?

---

2. **Mail Accounts**
   - Use the `cd` command to change directory to the `zerocool`'s Apple Mail MailData directory /Users/zerocool/Library/Mail/V2/MailData/.
   - Use the '`open`' command to open the `Accounts.plist` property list in Xcode.

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Mail/V2/MailData/

$ open Accounts.plist
```

   - Review the `Accounts.plist` property list.

1. What email addresses are associated with these accounts?

---

3. **Apple Mail – Envelope Index SQLite Database**
   - Using the same directory, /Users/zerocool/Library/Mail/V2/MailData/.

   1. Use the `file` command on the file "Envelope Index". What kind of file is this?

---

   - Copy the Envelope Index database files to your FOR518 directory (the *-shm and *-wal files are part of the SQLite database and may contain additional database transactions):
     - `Envelope Index`
     - `Envelope Index-shm`
     - `Envelope Index-wal`

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Mail/V2/MailData/

$ file Envelope\ Index

$ cp Envelope\ Index* ~/FOR518
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".

- Select the `Envelope Index` file that you just copied to your FOR518 directory.

- Review the SQLite tables in the left pane.
- Select the '`addresses`' table.

2. How many email addresses have been indexed (non-unique)?

---

- Select the '`attachments`' table.

3. How many unique attachments are there (by filename)?

---

- Select the '`mailboxes`' table.

4. What mailbox URL has the most email messages?

---

5. How many unread messages are there in the mailbox labeled `imap://z3r0cool95@imap.gmail.com/INBOX`?

---

- Select the '`messages`' table. Find entry with ROWID of 71

6. Was this message read?

---

- Find entry with ROWID of 74.

7. What is the epoch time this message was last viewed?

---

## 4. Mail Messages & Attachments

- Use the `cd` command to change directory to the `zerocool`'s Gmail email directory /Users/zerocool/Library/Mail/V2/IMAP-z3r0cool95@imap.gmail.com.
- Use the `ls` command to view the email MBOX directories.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Mail/V2/IMAP-
z3r0cool95@imap.gmail.com

$ ls
```

1. How many *.mbox directories are in this directory?

_____

- Use the `cd` command to enter the INBOX.mbox/21C7CD98-9DCB-4CE5-B19E-931FBE825451/Data/ directory.
- Use the `ls` command to list the contents of this directory.

```
$ cd INBOX.mbox/21C7CD98-9DCB-4CE5-B19E-931FBE825451/Data

$ ls
```

2. What two directories are found here?

_____

_____

- Use the `cd` command to enter the Messages directory.
- Use the `ls` command to list the contents of this directory.

```
$ cd Messages/

$ ls
```

- Open the email message `71.emlx` in TextEdit, using the '`open`' command.

```
$ open -a TextEdit 71.emlx
```

3. What email address was this message delivered to?

_____

4. Who is the sender of this message?

_____

5. When was this email message received?

_____

- Scroll to the bottom of this file. Review the property list at the end of the message.

6. What is the subject of this email?

_____

- Use the cd command to go back to the Attachments directory.
- Review the attachments in this directory. Use the '`ls -lR`' to perform a recursive directory listing.

```
$ cd ../Attachments/

$ ls -lR
```

7. How many attachments are in these directories?

_____

## 5. Mail Downloads

1. Were any attachments opened on this system?

_____

2. How do you know?

_____

_____

## 6. Apple Mail Analysis with BlackLight

- Review the output under the "Email" section.


- Find the email message associated with 71.emlx in the BlackLight email view.
- Review the email metadata shown in the middle pane.
- View the email message using different content types; change the dropdown "Show Content Type" to "Raw Source" and "HTML".

**Extra Credit –**

- **Get comfortable using the BlackLight interface - browse other email messages.**
- **Review the Envelope Index more to find more information pertaining to various email messages.**

## *Exercise – Step-By-Step*

### 1. Apple Mail Directory

- Use the cd command to change directory to the `zerocool`'s Apple Mail directory `/Users/zerocool/Library/Mail/V2/`.
- Use the '`ls -l`' command to view the files in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Mail/V2/

$ ls -l
```

1. How many email accounts appear to be associated with this user?
    a. Two IMAP accounts
        i. AosIMAP-z3r0cool95
        ii. IMAP-z3r0cool95@imap.gmail.com

### 2. Mail Accounts

- Use the `cd` command to change directory to the `zerocool`'s Apple Mail MailData directory /Users/zerocool/Library/Mail/V2/MailData/.
- Use the '`open`' command to open the `Accounts.plist` property list in Xcode.

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Mail/V2/MailData/

$ open Accounts.plist
```

- Review the `Accounts.plist` property list.

1. What email addresses are associated with these accounts?
    a. z3r0cool95@icloud.com
    b. z3r0cool95@gmail.com

### 3. Apple Mail – Envelope Index SQLite Database

- Using the same directory, /Users/zerocool/Library/Mail/V2/MailData/.

1. Use the `file` command on the file "Envelope Index". What kind of file is this?
    a. SQLite 3.x Database

- Copy the Envelope Index database files to your FOR518 directory (the *-shm and *-wal files are part of the SQLite database and may contain additional database transactions):
    - `Envelope Index`
    - `Envelope Index-shm`
    - `Envelope Index-wal`

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Mail/V2/MailData/

$ file Envelope\ Index

$ cp Envelope\ Index* ~/FOR518
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".

- Select the `Envelope Index` file that you just copied to your FOR518 directory.

- Review the SQLite tables in the left pane.
- Select the '`addresses`' table.

2. How many email addresses have been indexed (non-unique)?
    a. 23

- Select the '`attachments`' table.

3. How many unique attachments are there (by filename)?
    a. Three

- Select the '`mailboxes`' table.

4. What mailbox URL has the most email messages?
    a. imap://z3r0cool95@imap.gmail.com/%5BGmail%5D/All%20Mail
        i. 99 email messages as shown under `total_count` column.
5. How many unread messages are there in the mailbox labeled `imap://z3r0cool95@imap.gmail.com/INBOX`?
    a. 69 (unseen_count column)

- Select the '`messages`' table. Find entry with ROWID of 71.

6. Was this message read?
    a. No, the '`read`' column shows a 0.
        i. The `date_last_viewed` value is also *Null*.

- Find entry with ROWID of 74.
7. What is the epoch time this message was last viewed?
    a. 1387069958

**4. Mail Messages & Attachments**

- Use the `cd` command to change directory to the `zerocool`'s Gmail email directory /Users/zerocool/Library/Mail/V2/IMAP-z3r0cool95@imap.gmail.com.

- Use the `ls` command to view the email MBOX directories.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Mail/V2/IMAP-
z3r0cool95@imap.gmail.com

$ ls
```

1. How many *.mbox directories are in this directory?
    a. Four:
        i. Drafts.mbox
        ii. INBOX.mbox
        iii. Sent Messages.mbox
        iv. [Gmail].mbox

- Use the `cd` command to enter the INBOX.mbox/21C7CD98-9DCB-4CE5-B19E-931FBE825451/Data/ directory.
- Use the `ls` command to list the contents of this directory.

```
$ cd INBOX.mbox/21C7CD98-9DCB-4CE5-B19E-931FBE825451/Data

$ ls
```

2. What two directories are found here?
    a. Attachments
    b. Messages

- Use the `cd` command to enter the Messages directory.
- Use the `ls` command to list the contents of this directory.

```
$ cd Messages/

$ ls
```

- Open the email message `71.emlx` in TextEdit, using the '`open`' command.

```
$ open -a TextEdit 71.emlx
```

3. What email address was this message delivered to?
    a. z3r0cool95@gmail.com
4. Who is the sender of this message?

a. Google+ / noreply-203795f6@plus.google.com

5. When was this email message received?
   a. 12 Nov 2013 21:45:10 -0800 (PST)

- Scroll to the bottom of this file. Review the property list at the end of the message.

6. What is the subject of this email?
   a. "Dade, do you know Kate Libby?"

- Use the cd command to go back to the Attachments directory.
- Review the attachments in this directory. Use the '`ls -lR`' to perform a recursive directory listing.

```
$ cd ../Attachments/

$ ls -lR
```

7. How many attachments are in these directories?
   a. Two:
      i. google.png
      ii. profilephoto.png

**5. Mail Downloads**

1. Were any attachments opened on this system?
   i. No
2. How do you know?
   i. The file OpenedAttachmentsV2.plist was never created in the ~/Library/Mail/V2/MailData/ directory.
   ii. The directory ~/Library/Mail Downloads/ also was not created.

**6. Apple Mail Analysis with BlackLight**

- Review the output under the "Email" section.


- Find the email message associated with 71.emlx in the BlackLight email view.
- Review the email metadata shown in the middle pane.
- View the email message using different content types; change the dropdown "Show Content Type" to "Raw Source" and "HTML".

**Extra Credit –**

- **Get comfortable using the BlackLight interface - browse other email messages.**
- **Review the Envelope Index more to find more information pertaining to various email messages.**

## *Exercise – Key Takeaways*

- **Know where the Apple Mail key files are stored and their importance.**
- **Get comfortable with some Mac OS X command line utilities.**
- **Get comfortable with the BlackLight application interface and nuances.**

This page intentionally left blank.

# Exercise 2.4 – Mac Applications

## Objectives

- Get familiar with the key files for a variety of Mac applications including Messages, FaceTime, Calendar, Address Book, iPhoto, iWork and Airdrop.
- Get familiar with the Mac OS X command line.

## 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/.
   - A SQLite Database viewer.
   - 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. **Mount the Dade Murphy forensic (`dademurphy.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
     - Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
    - Mount Point – Newly created specifically for this image.
- Uses the hdiutil command with the "attach" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the -nomount argument to suppress mounting (for now). The output from this command will display a /dev/disk#, use the appropriate disk device in the next command.
- Use the mount_hfs command with the following parameters to mount the /dev/disk# (from the previous command) to the /Volumes/dademurphy_mounted/ mount point. This drive will now be available in the Finder or Terminal applications.
    - -j – Ignore the journal
    - -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.

You can access this newly created mounted drive on /Volumes/dademurphy_mounted/.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
    - Use the mkdir command to create a mount point for the ewfmount output. The directory name dademurphy_image is used in the example.
    - Use the mkdir command again to create a mount point for the mounted disk image, the example dademurphy_mounted in used the example above.
    - Use ewfmount to mount the dademurphy.e01 image to the /Volumes/dadmurphy_image/ mount point.
    - Use the ln -s command to create a symbolic link for the ewf1 file, name the link dadeimage.dmg. (A DMG file is needed for hdiutil to recognize the file.)
    - Uses the hdiutil command with the "attach" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the -nomount argument to suppress mounting (for now). The output from this command will display a /dev/disk#, use the appropriate disk device in the next command.
    - Use the mount_hfs command with the following parameters to mount the /dev/disk# (from the previous command) to the /Volumes/dademurphy_mounted/ mount point. This drive will now be available in the Finder or Terminal applications.
        - -j – Ignore the journal

- `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

4. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

**CHOOSE YOUR OWN ADVENTURE!**

**There are many Mac applications and each investigation is different. We do not have enough time into all of them. Choose <u>two</u> or <u>three</u> that interest you the most – the rest are extra credit!**

1. **Instant Messaging & FaceTime**
2. **Calendar**
3. **Address Book**
4. **iPhoto**
5. **iWork**
6. **AirDrop**

1. **Instant Messaging & FaceTime.**
   - Use the `cd` command to go to `zerocool`'s Preferences directory.
   - Use the `open` command to open all the property list files in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open *.plist
```

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

1. Whom did this account last chat with?

   a. _____

2. What is the `accountID` (GUID) used to chat with this person?

   a. _____

   - Review the contents of the `com.apple.iChat.Jabber.plist` file.

3. What is the GUID associated with this user account?

   _____

4. What email address is used for this Jabber account?

   _____

5. What type of chat account is this?

   _____

   - Review the contents of the `com.apple.imservice.FaceTime.plist` file.

6. What email accounts are associated with this FaceTime account?

_____

_____

- Use the `cd` command to go to `zerocool`'s Messages Archive, ~/Library/Messages/Archive/.
- Use the `ls` command to list the contents of this directory.

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Messages/Archive/

$ ls -l
```

7. What two dates did communication occur?

_____

_____

- Use the `cd` command to go to `zerocool`'s Messages Archive, ~/Library/Messages/Archive/2013-12-16/.
- Use the `ls` command to list the contents of this directory.

8. Using the file command, what kind of file is this?

_____

- Use the `open` command to open this file, review the contents of this window.
  - This command should open the contents in the Messages or iChat application.

```
$ cd 2013-12-16/

$ ls -l

$ file Kate\ Libby\ on\ 2013-12-16\ at\ 20.10.ichat

$ open Kate\ Libby\ on\ 2013-12-16\ at\ 20.10.ichat
```

9. Just by looking at this chat window not including chat context – can you tell if there had been file transfers or transfer attempts?

_____

10. Try renaming this file with a .plist extension and reviewing the contents in Xcode.

- Review the contents of the `/Users/zerocool/Library/Preferences/ByHost/com.apple.Messages.FileTransfers.40A90B07-FC53-52C8-A774-6F1A5E659E9C.plist` file. Review the contents under Item 0.

11. What is the filename of the file transferred?

    _____

12. How many bytes were transferred? (Ideally this is the file size)

    _____

13. Who was this file transferred to?

    _____

14. When was this file transfer started in UTC?

    _____

**2. Calendar**

- Use the `cd` command to go to `zerocool's Calendars` directory.
- Use the `open` command to open the `com.apple.iCal.plist` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open com.apple.iCal.plist
```

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

1. What is the GUID of the default principle calendar?

   _____

2. What is the GUID of the default calendar?

   _____

- Use the `cd` command to go to `zerocool's Calendars` directory.
- Use the `ls -l` command to list the contents of the directory.
- Use the `cd` command to go to the default principle calendar directory.
- Use the `ls -l` command to list the contents of this directory.
- Use the `open` command to open the `Info.plist` file for this calendar.

```
$ cd ../Calendars/

$ ls -l

$ cd 68CD24C5-55EE-43CA-BF47-2EE61C3DD648.caldav/
```

```
$ ls -l

$ open Info.plist
```

3. What kind of calendar is this?

_____

4. What is the title for this calendar?

_____

5. What is the login for this calendar?

_____

- Use the `cd` command to go to the default calendar.
- Use the `ls -l` command to list the contents of this directory.
- Use the `open` command to open the `Info.plist` file for this calendar.

```
$ cd E629B7FD-FEAC-407B-805E-25478F232AA5.calendar/

$ ls -l

$ open Info.plist
```

6. What is the title for this calendar?

_____

7. What is the time zone this calendar?

_____

- Use the `cd` command to go to the default calendar /Events directory.
- Use the `ls -l` command to list the contents of this directory.

```
$ cd Events/

$ ls -l
```

8. How many events are in this calendar (*.ics files)?

_____

- Use the `cat` command to view the contents of the `7646AAD2-3185-4CB8-911F-27663354B004.ics` file.

```
$ cat 7646AAD2-3185-4CB8-911F-27663354B004.ics
```

9. When was this calendar item created?

_____

10. When did this calendar item occur?

_____

11. What are the contents of this calendar item as a user would see it?

_____

- Use the `cd` command to go back to the /Calendars directory (three directories up).
- Use the open command to open this directory.

```
$ cd ../../../

$ open .
```

- Using the same methods we've used in the previous exercises, open the `Calendar Cache` SQLite database file in the SQLite Manager plugin for Firefox.
- Find tuple where "`Z_PK`" = 11 in the '`ZICSELEMENT`' table. Double-click this entry. Take a moment to review its contents...look familiar?
- **Extra Credit:** Continue reviewing this database.

**3. Address Book**

- Use the `cd` command to go to `zerocool`'s Preferences directory.
- Use the `open` command to open the `com.apple.AddressBook.plist` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open com.apple.AddressBook.plist
```

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

1. What is the GUID of the last selected Address Book Contact?

_____

2. What is the Address Book Default Source GUID?

_____

- Use the `cd` command to go to `zerocool`'s Address Book directory.
- Use the `ls -l` command to list the contents of this directory. Note the contents.
- Use the `cd` command to enter the directory for the default source for Address Book using the GUID noted above.
- Use the `ls -l` command to list the contents of this directory. Note the contents.
- Use the `cd` command to enter Metadata/ the directory.
- Use the `ls -l` command to list the contents of this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Application\
Support/AddressBook/

$ ls -l

$ cd Sources/F1E178A9-19A4-44F3-9730-9FA5FD0EA049

$ ls -l

$ cd Metadata/
```

3. How many groups are in this Address Book?

   _____

4. How many persons are in this Address Book?

   _____

- Use the `file` command to show the file type of these files.

```
$ file *
```

5. What kind of files are these?

   _____

- **Students using 10.8+ (Students using 10.7, see below)**
- Use the `plutil -p` command to view the `CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp` file, piping the output to the `less` utility.
  - Why this command? Try the following – sometimes it is good to have a backup command.
    i. `open CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp`
    ii. `open -a Xcode CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp`
- This `plutil -p` command outputs the contents of a binary property list to the Terminal in a human-readable format without changing the format of the property list.

```
$ plutil -p CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp | less
```

- **Students using 10.7 (Students using 10.8+, please skip)**
- Copy the file using the `cp` command `CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp` to your FOR518 directory, and change the file extension to ".plist" using the `mv` command.
- Open the file using the `open` command.

```
$ cp CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp ~/FOR518/

$ mv ~/FOR518/CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp
~/FOR518/CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.plist

$ open ~/FOR518/CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.plist
```

6. What is the full name of this contact?

   _____

7. What is her email address?

   _____

8. What is her Jabber chat account ID?

   _____

9. When was this contact created?

   _____

- Using the `cd` command, traverse back to the previous directory and into the `Images` directory.
- List the contents of this folder using the `ls -l` command.
- Using the open command, open all the files in this directory.

```
$ cd ../Images/

$ ls -l

$ open *
```

10. What buddy icon does Kate Libby use?

    _____

- Using the `cd` command, traverse back to the main `AddressBook` directory.

- List the contents of this folder using the `ls -l` command.
- Using the open command, open this directory in finder.

```
$ cd ../../../

$ ls -l

$ open .
```

- Open the `MailRecents-v4.abcdmr` SQLite database in the SQLite Manager plug-in for Firefox. Please see previous labs for process.
- Review the contents of the '`ZABCDMAILRECENT`'.

11. What email has `zerocool` recently emailed?

---

- Review the contents of the '`ZABCDLASTEMAILDATE`'.

12. What is the earliest date recorded that `zerocool` emailed Kate Libby (`ZMAILRECENT` = 1)?

---

**4. iPhoto**

- Use the `cd` command to go to `zerocool`'s Preferences directory.
- Use the `open` command to open the `com.apple.iPhoto.plist` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open com.apple.iPhoto.plist
```

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

1. Where is the iPhoto Library stored?

---

- Use the `cd` command to go to `zerocool`'s iPhoto Library directory.
- Use the `ls -l` command to list the contents of this directory. Note the contents.
- Use the open command to open the `AlbumData.xml` file.
  - This file may be copied and renamed with a .plist file extension for easier viewing if preferred.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Pictures/iPhoto\
Library.photolibrary

$ ls -l

$ open AlbumData.xml
```

2. How many photo albums are in this iPhoto library?

   _____

3. What is the album name of AlbumID = 18?

   _____

4. What is Image 21 a photo of?

   _____

5. **iWork**
   - Use the `cd` command to go to `zerocool`'s Preferences directory.
   - Use the `open` command to open the `com.apple.iWork*.plist` files.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open com.apple.iWork*
```

   - Review the contents of these files.

1. What is the name of the last Keynote document opened?

   _____

2. Why are there two documents named "`bitcoin shopping list!`" in the recent documents for Pages?

   _____

   - Use the `cd` command to go to `zerocool`'s Documents directory.
   - Use the `ls -l` command to list the contents of this directory.
   - Use the file command on the `bitcoin shopping list!.pages` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Documents/

$ ls -l

$ file bitcoin shopping list!.pages
```

3. What kind of file is `bitcoin shopping list!.pages`?

_____

- Use the `cp` command to copy this file to your FOR518 directory.

**6. AirDrop**

- Use the `cd` command to go to `zerocool`'s AirDrop Cache directory.
- Use the `ls -l` command to list the contents of this directory.

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Caches/com.apple.AirD
rop/

$ ls -l
```

1. How many users did `zerocool` communicate with over AirDrop?

_____

2. What date is the likely date that communication was established?

_____

3. What icon did this user use?

_____

**Extra Credit –**

- **Review these same files and databases with BlackLight, find what tool is easier for you to use.**

1. **Instant Messaging & FaceTime**
   - Use the `cd` command to go to `zerocool`'s Preferences directory.
   - Use the `open` command to open all the property list files in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open *.plist
```

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

1. Whom did this account last chat with?
   a. katelibby11@gmail.com (`RecentChats` & `Participants` keys)
2. What is the `accountID` (GUID) used to chat with this person?
   a. 5C49D451-BA4D-45AD-9710-386E0AFA7928

   - Review the contents of the `com.apple.iChat.Jabber.plist` file.

3. What is the GUID associated with this user account?
   a. 5C49D451-BA4D-45AD-9710-386E0AFA7928
4. What email address is used for this Jabber account?
   a. z3r0cool95@gmail.com (`LoginAs` key)
5. What type of chat account is this?
   a. Gmail Chat (`Description` key)

   - Review the contents of the `com.apple.imservice.FaceTime.plist` file.

6. What email accounts are associated with this FaceTime account?
   a. z3r0cool95@icloud.com
   b. z3r0cool95@gmail.com
      i. Check under `Aliases` or `VettedAliases` keys.

   - Use the `cd` command to go to `zerocool`'s Messages Archive, ~/Library/Messages/Archive/.
   - Use the `ls` command to list the contents of this directory.

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Messages/Archive/

$ ls -l
```

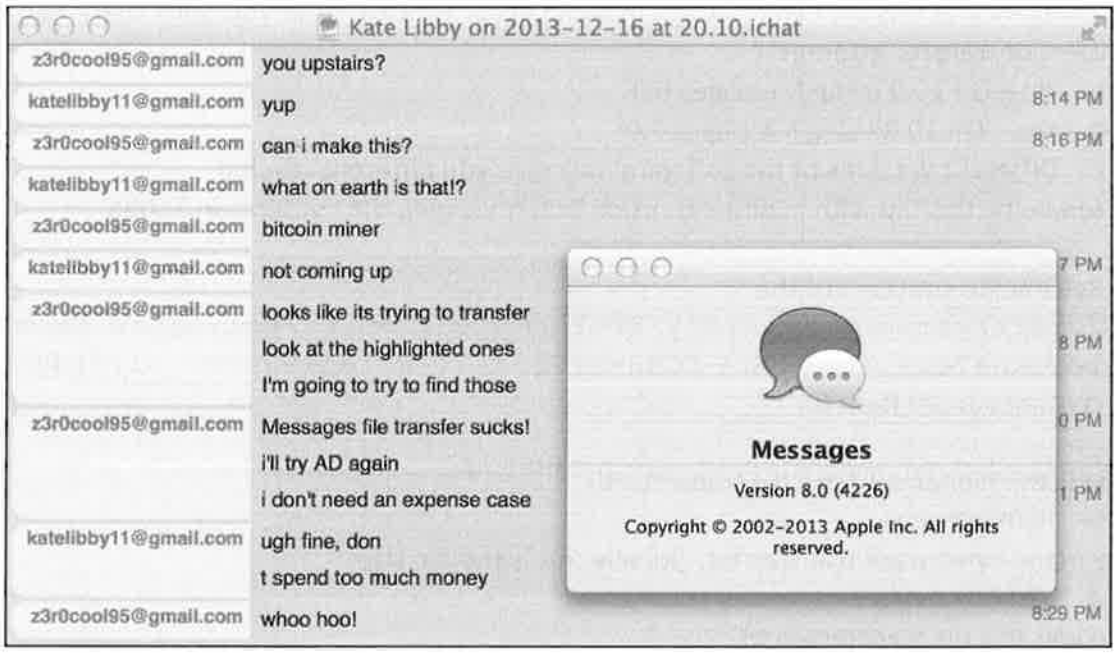
7. What two dates did communication occur?
   a. 2013-12-16
   b. 2013-12-22

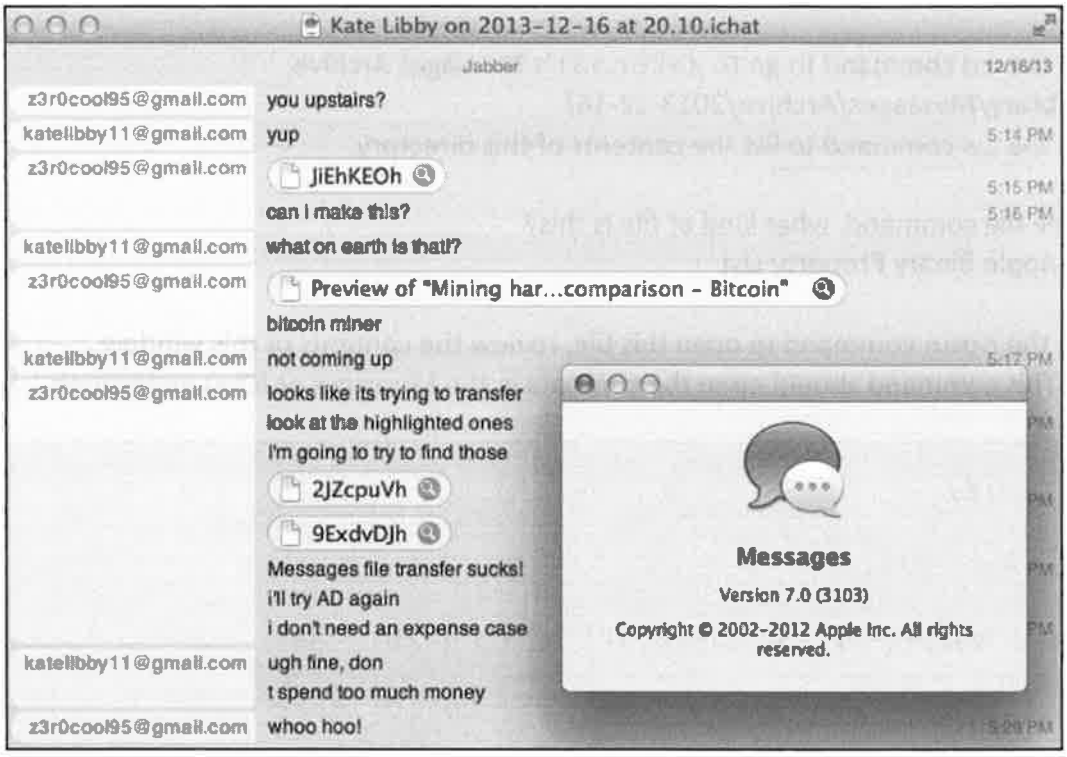
- Use the `cd` command to go to `zerocool`'s Messages Archive, ~/Library/Messages/Archive/2013-12-16/.
- Use the `ls` command to list the contents of this directory.

8. Using the file command, what kind of file is this?
   a. Apple Binary Property List

- Use the `open` command to open this file, review the contents of this window.
  - This command should open the contents in the Messages or iChat application.

```
$ cd 2013-12-16/

$ ls -l

$ file Kate\ Libby\ on\ 2013-12-16\ at\ 20.10.ichat

$ open Kate\ Libby\ on\ 2013-12-16\ at\ 20.10.ichat
```






9. Just by looking at this chat window not including chat context – can you tell if there had been file transfers or transfer attempts?
    a. No – On 10.9 using Messages 8.0
    b. Yes – On 10.8 using Messages 7.0
    c. Different versions of the software may give you different results!
10. Try renaming this file with a .plist extension and reviewing the contents in Xcode.

- Review the contents of the `/Users/zerocool/Library/Preferences/ByHost/com.apple.Messages.FileTransfers.40A90B07-FC53-52C8-A774-6F1A5E659E9C.plist` file. Review the contents under Item 0.

11. What is the filename of the file transferred?
    a. JiEhKEOh.jpg
12. How many bytes were transferred? (Ideally this is the file size)
    a. 120,507 bytes
13. Who was this file transferred to?
    a. katelibby11@gmail.com
14. When was this file transfer started in UTC?
    a. Tue Dec 17 01:16:05 UTC 2013
        i. `date -ur 1387242965` from first 10 digits from `IMFileTransferStartDate` key.

**2. Calendar**

- Use the `cd` command to go to `zerocool`'s Preferences directory.
- Use the `open` command to open the `com.apple.iCal.plist` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open com.apple.iCal.plist
```

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

1. What is the GUID of the default principle calendar?
    a. 68CD24C5-55EE-43CA-BF47-2EE61C3DD648 (`CalDefaultPrincipal` Key)
2. What is the GUID of the default calendar?
    a. E629B7FD-FEAC-407B-805E-25478F232AA5 (`CalDefaultCalendar` Key)

- Use the `cd` command to go to `zerocool`'s Preferences directory.
- Use the `ls -l` command to list the contents of the directory.
- Use the `cd` command to go to the default principle calendar directory.
- Use the `ls -l` command to list the contents of this directory.
- Use the `open` command to open the `Info.plist` file for this calendar.

```
$ cd ../Calendars/

$ ls -l

$ cd 68CD24C5-55EE-43CA-BF47-2EE61C3DD648.caldav/

$ ls -l

$ open Info.plist
```

3. What kind of calendar is this?
    a. CalDAV (`Type` Key)
4. What is the title for this calendar?
    a. iCloud (`Title` Key)
5. What is the login for this calendar?
    a. z3r0cool95@gmail.com (`Login` Key)

- Use the `cd` command to go to the default calendar.
- Use the `ls -l` command to list the contents of this directory.
- Use the `open` command to open the `Info.plist` file for this calendar.

```
$ cd E629B7FD-FEAC-407B-805E-25478F232AA5.calendar/

$ ls -l

$ open Info.plist
```

6. What is the title for this calendar?
    a. Home (`Title` Key)
7. What is the time zone this calendar?
    a. America/New_York (`TimeZone` Key)

- Use the `cd` command to go to the default calendar /Events directory.
- Use the `ls -l` command to list the contents of this directory.

```
$ cd Events/

$ ls -l
```

8. How many events are in this calendar (*.ics files)?
    a. Six

- Use the `cat` command to view the contents of the `7646AAD2-3185-4CB8-911F-27663354B004.ics` file.

```
$ cat 7646AAD2-3185-4CB8-911F-27663354B004.ics
```

9. When was this calendar item created?
    a. 20131120T121200Z, 11/20/13 at 12:12:00 UTC (`CREATED`)
10. When did this calendar item occur?
    a. 20131121T100000, 11/21/13 at 00:00:00 UTC (`DTSTART`)
11. What are the contents of this calendar item as a user would see it?
    a. "Meet with Frank"

- Use the `cd` command to go back to the /Calendars directory (three directories up).
- Use the open command to open this directory.

```
$ cd ../../../

$ open .
```

- Using the same methods we've used in the previous exercises, open the `Calendar Cache` SQLite database file in the SQLite Manager plugin for Firefox.
- Find tuple where "`Z_PK`" = 11 in the '`ZICSELEMENT`' table. Double-click this entry. Take a moment to review its contents...look familiar?
- **Extra Credit:** Continue reviewing this database.

### 3. Address Book

- Use the `cd` command to go to `zerocool`'s Preferences directory.
- Use the `open` command to open the `com.apple.AddressBook.plist` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open com.apple.AddressBook.plist
```

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

1. What is the GUID of the last selected Address Book Contact?
    a. FC8B50AA-83D2-468B-9E30-53775EF61769
        i. `ABBookWindowController-MainBookWindow-personListController/selectedUIDs/Item 0` Key
2. What is the Address Book Default Source GUID?
    a. F1E178A9-19A4-44F3-9730-9FA5FD0EA049

- Use the `cd` command to go to `zerocool`'s Address Book directory.
- Use the `ls -l` command to list the contents of this directory. Note the contents.
- Use the `cd` command to enter the directory for the default source for Address Book using the GUID noted above.
- Use the `ls -l` command to list the contents of this directory. Note the contents.
- Use the `cd` command to enter Metadata/ the directory.
- Use the `ls -l` command to list the contents of this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Application\
Support/AddressBook/

$ ls -l

$ cd Sources/F1E178A9-19A4-44F3-9730-9FA5FD0EA049

$ ls -l

$ cd Metadata/
```

3. How many groups are in this Address Book?
    a. One
        i. F70A6B13-B247-4398-AF8B-AFFAAB400416:ABGroup.abcdg

4. How many persons are in this Address Book?
    a. Four
        i. 701A34E7-3773-4DBF-8C07-C0260E07151F:ABPerson.abcdp
        ii. CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp
        iii. FAEECBCF-8580-4B53-830D-6BA0B57BE3EB:ABPerson.abcdp
        iv. FC8B50AA-83D2-468B-9E30-53775EF61769:ABPerson.abcdp

- Use the `file` command to show the file type of these files.

```
$ file *
```

5. What kind of files are these?
    a. Apple Binary Property Lists

- **Students using 10.8+ (Students using 10.7, see below)**
- Use the `plutil -p` command to view the `CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp` file, piping the output to the `less` utility.
    - Why this command? Try the following – sometimes it is good to have a backup command.
        i. `open CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp`
        ii. `open -a Xcode CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp`
- This `plutil -p` command outputs the contents of a binary property list to the Terminal in a human-readable format without changing the format of the property list.

```
$ plutil -p CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp | less
```

- **Students using 10.7 (Students using 10.8+, please skip)**
- Copy the file using the `cp` command `CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp` to your FOR518 directory, and change the file extension to ".plist" using the `mv` command.
- Open the file using the `open` command.

```
$ cp CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp ~/FOR518/

$ mv ~/FOR518/CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.abcdp
~/FOR518/CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.plist

$ open ~/FOR518/CC4D39B3-F3C5-4D2B-9F82-8234477505A4:ABPerson.plist
```

6. What is the full name of this contact?
    a. Kate Libby (`First` & `Last` Keys)

7. What is her email address?
    a. katelibby11@gmail.com (`Email` Key)
8. What is her Jabber chat account ID?
    a. katelibby11@gmail.com (`JabberInstant` or `InstantMessage` Keys)
9. When was this contact created?
    a. 2013-11-17 18:41:28 +0000 (`Creation` Key)

- Using the `cd` command, traverse back to the previous directory and into the `Images` directory.
- List the contents of this folder using the `ls -l` command.
- Using the open command, open all the files in this directory.

```
$ cd ../Images/

$ ls -l

$ open *
```

10. What buddy icon does Kate Libby use?
    a. A penguin
    b. The GUID of the images matches the contact GUID previously seen.

- Using the `cd` command, traverse back to the main `AddressBook` directory.
- List the contents of this folder using the `ls -l` command.
- Using the open command, open this directory in finder.

```
$ cd ../../../

$ ls -l

$ open .
```

- Open the `MailRecents-v4.abcdmr` SQLite database in the SQLite Manager plug-in for Firefox. Please see previous labs for process.
- Review the contents of the '`ZABCDMAILRECENT`'.

11. What email has `zerocool` recently emailed?
    a. Katelibby11@gmail.com

- Review the contents of the '`ZABCDLASTEMAILDATE`'.

12. What is the earliest date recorded that `zerocool` emailed Kate Libby (`ZMAILRECENT` = 1)?
    a. 390939788 (WebKit)
    b. Wed May 22 18:23:08 UTC 2013

4. **iPhoto**
    - Use the `cd` command to go to `zerocool`'s Preferences directory.
    - Use the `open` command to open the `com.apple.iPhoto.plist` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open com.apple.iPhoto.plist
```

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

1. Where is the iPhoto Library stored?
    a. /Users/zerocool/Pictures/iPhoto Library.photolibrary

    - Use the `cd` command to go to `zerocool`'s iPhoto Library directory.
    - Use the `ls -l` command to list the contents of this directory. Note the contents.
    - Use the open command to open the `AlbumData.xml` file.
        - This file may be copied and renamed with a .plist file extension for easier viewing if preferred.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Pictures/iPhoto\
Library.photolibrary

$ ls -l

$ open AlbumData.xml
```

2. How many photo albums are in this iPhoto library?
    a. Five (`List of Albums` key)
3. What is the album name of AlbumID = 18?
    a. "Dec 2013 Photo Stream" (`List of Albums/Item 4/AlbumName` Keys)
4. What is Image 21 a photo of?
    a. A New Orleans themed sign.
    b. Follow the file path in the `AlbumData.xml` for Image 21 and view with the open command or via Finder.

5. **iWork**
    - Use the `cd` command to go to `zerocool`'s Preferences directory.
    - Use the `open` command to open the `com.apple.iWork*.plist` files.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences/

$ open com.apple.iWork*
```

- Review the contents of these files.

1. What is the name of the last Keynote document opened?
    a. "Untitled"
2. Why are there two documents named "`bitcoin shopping list!`" in the recent documents for Pages?
    a. Same document name, stored in two locations.
    b. Extract the Bookmark Data BLOB and view in a hex editor (com.apple.iWork.Pages.LSSharedFileList.plist)

- Use the `cd` command to go to `zerocool`'s Documents directory.
- Use the `ls -l` command to list the contents of this directory.
- Use the file command on the `bitcoin shopping list!.pages` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Documents/

$ ls -l

$ file bitcoin shopping list!.pages
```

3. What kind of file is `bitcoin shopping list!.pages`?
    a. Zip Archive

**6. AirDrop**

- Use the `cd` command to go to `zerocool`'s AirDrop Cache directory.
- Use the `ls -l` command to list the contents of this directory.

```
$ cd
/Volumes/dademurphy_mounted/Users/zerocool/Library/Caches/com.apple.AirDrop/

$ ls -l
```

1. How many users did `zerocool` communicate with over AirDrop?
    a. One, only one base64-encoded filename exists in this directory.
2. What date is the likely date that communication was established?
    a. December 17, 2013 (UTC) – When this file was created.
3. What icon did this user use?
    a. Water Lily, use the `open` command to view the image.

**Extra Credit –**

- **Review these same files and databases with BlackLight, find what tool is easier for you to use.**

## *Exercise – Key Takeaways*

- **Get comfortable with some Mac OS X command line utilities.**
- **Get familiar with the key files associated with Mac specific applications.**

# Exercise 3.1 – System Data & Preferences

## Objectives

- Review the key files associated with the system preferences.
- Get familiar with the Mac OS X command line.

## 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/.
   - A SQLite database viewer
   - 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. **Mount the Dade Murphy forensic (`dademurphy.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
     - Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.

    - Mount Point – Newly created specifically for this image.
- Uses the hdiutil command with the "attach" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the -nomount argument to suppress mounting (for now). The output from this command will display a /dev/disk#, use the appropriate disk device in the next command.
- Use the mount_hfs command with the following parameters to mount the /dev/disk# (from the previous command) to the /Volumes/dademurphy_mounted/ mount point. This drive will now be available in the Finder or Terminal applications.
    - -j – Ignore the journal
    - -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.

You can access this newly created mounted drive on /Volumes/dademurphy_mounted/.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
    - Use the mkdir command to create a mount point for the ewfmount output. The directory name dademurphy_image is used in the example.
    - Use the mkdir command again to create a mount point for the mounted disk image, the example dademurphy_mounted in used the example above.
    - Use ewfmount to mount the dademurphy.e01 image to the /Volumes/dadmurphy_image/ mount point.
    - Use the ln -s command to create a symbolic link for the ewf1 file, name the link dadeimage.dmg. (A DMG file is needed for hdiutil to recognize the file.)
    - Uses the hdiutil command with the "attach" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the -nomount argument to suppress mounting (for now). The output from this command will display a /dev/disk#, use the appropriate disk device in the next command.
    - Use the mount_hfs command with the following parameters to mount the /dev/disk# (from the previous command) to the /Volumes/dademurphy_mounted/ mount point. This drive will now be available in the Finder or Terminal applications.
        - -j – Ignore the journal
        - -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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

4. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

## *Exercise – Questions*

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

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

$ open SystemVersion.plist
```

What version of OS X 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/dademurphy_mounted/private/var/db/
$ ls -la
```

1. What is the likely date of system installation?

_____

3. **System Time Zone & 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 ../../../etc/

$ ls -l

$ ls -l localtime
```

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

_____

- Use the `cd` command to navigate to the system preferences directory.

- Use the `pwd` command to verify you are in the system preferences directory.
  i. `/Volumes/dademurphy_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 ../Library/Preferences

$ pwd

$ ls -la

$ open .GlobalPreferences.plist
```

2. More specifically, what city is used to determine the time zone used?

_____

3. What is the primary language setting used?

_____

**4. Network Settings**

- Use the `cd` command to navigate to the `SystemConfiguration` 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 all the plist files in this directory.

```
$ cd SystemConfiguration/

$ ls -l

$ open *.plist
```

- Review the `NetworkInterfaces.plist` file.

1. What model system is this?

_____

2. How many network interfaces does this system have?

_____

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

_____

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

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

5. What is the name of the network that was last accessed on December 12, 2013 (EST/EDT) (local system time)?

6. Is this network "open" or are network credentials open?

7. What is the name of the network that is likely the system's "home" network?

8. Why do you think it is the "home" network?

9. When was this system last connected to its "home" network?

10. **Extra Credit:** What brand is this Wi-Fi Access Point?

- Review the `preferences.plist` file.

11. What is the hostname of this system?

## 5. 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/dademurphy_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 Log directory.
- Use the `open` command to open the `page_log` file.

```
$ cd /Volumes/dademurphy_mounted/var/log/cups/
```

```
$ open page_log
```

3. What two documents were printed?

_____

_____

4. How many pages long were these documents?

_____

5. What user did the printing?

_____

- Get a root shell.
- Use the `cd` command to navigate to the Printer Spool directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory.
- Use the `file` command on the printer data files.
- Using the `cp` command, copy the `d000*` data files property list to a directory of your choice.
- Use the `chown` command to change the ownership to your user account name.
- Using the `open` command to view the PDF printer data files.

```
$ sudo -s
# cd /Volumes/dademurphy_mounted/private/var/spool/cups/
# ls -l
# file d0000*
# cp d0000* ~/FOR518
# chown <your username> ~/FOR518/d0000*
# open ~/FOR518/d0000*
```

6. Using the printer control files (`c0000#`), can you determine what application those two documents were printed from (hint: Look for an `ApplicationName`, try using the `strings` program)?

_____

**6. 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/dademurphy_mounted/Library/Preferences/
$ open com.apple.SoftwareUpdate.plist
```

1. When was the last successful update?

_____

2. System version when last updated?

_____

3. When was the software update last attempted?

_____

4. What are the names of the three available software updates?

_____

_____

_____

- 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/dademurphy_mounted/Library/Receipts/
$ ls -l
$ open InstallHistory.plist
```

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

_____

6. Review Item 0, what do you think this entry is indicative of?

_____

7. What application was used to install OS X?

_____

8. On what date was the "LanScan" application installed?

_____

9. What application was used to install this application?

_____

10. What application was installed on 11/20/13 (EST/EDT)?

11. What application was used to install this application?

12. What application installed the Brother Printer Software Update?

- Use the `cd` command to navigate to 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 that 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 LanScan application.

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

$ ls -lt

$ open <anyfile>.plist

$ lsbom -s com.iwaxx.LanScan.bom
```

13. How many files are associated with the LanScan application?

**Extra Credit –**

- **Review the same files using the BlackLight application.**

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

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

$ open SystemVersion.plist
```

1. What version of OS X is this system running?
   a. 10.8.5

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/dademurphy_mounted/private/var/db/

$ ls -la
```

1. What is the likely date of system installation?
   a. November 17, 2013 (EST/EDT)
      i. Use the `stat -x` command to determine MAC times for `.AppleSetupDone` and `.AppleInstallType.plist` files

3. **System Time Zone & 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 ../../../etc/

$ ls -l

$ ls -l localtime
```

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

- Use the `cd` command to navigate to the system preferences directory.
- Use the `pwd` command to verify you are in the system preferences directory.
  i. `/Volumes/dademurphy_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 ../Library/Preferences

$ pwd

$ ls -la

$ open .GlobalPreferences.plist
```

2. More specifically, what city is used to determine the time zone used?
   a. Washington, D.C.
      i. `com.apple.TimeZonePref.Last_Selected_City` or `com.apple.preferences.timezone.selected_city` Keys
3. What is the primary language setting used?
   a. en_US – US English
      i. `AppleLocale` or `AppleLanguages` Keys

**4. Network Settings**

- Use the `cd` command to navigate to the `SystemConfiguration` 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 all the plist files in this directory.

```
$ cd SystemConfiguration/

$ ls -l

$ open *.plist
```

- Review the `NetworkInterfaces.plist` file.

1. What model system is this?
   a. MacBookAir5,1
2. How many network interfaces does this system have?
   a. Two; en0 and en1
3. What is the MAC address for the Wi-Fi interface?
   a. 7c:d1:c3:df:64:67

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

4. How many "remembered" Wi-Fi networks are there?
    a. Four
5. What is the name of the network that was last accessed on December 12, 2013 (local system time)?
    a. Hhonors
6. Is this network "open" or are network credentials open?
    a. Open (`SecurityType` Key)
7. What is the name of the network that is likely the system's "home" network?
    a. Cyberdelia
8. Why do you think it is the "home" network?
    a. It is Item 0 in the remembered networks list and has a SecurityType of "WPA2 Personal".
9. When was this system last connected to its "home" network?
    a. December 23, 2013 (EST/EDT) (local system time)
10. **Extra Credit:** What brand is this Wi-Fi Access Point?
    a. Tenda
        i. Extract the BSSID MAC address from the "`CachedScanRecord`" key, and Google the first three octets of the MAC address.

- Review the `preferences.plist` file.

11. What is the hostname of this system?
    a. Dades-MacBook-Air
        i. `System/Network/Hostnames/LocalHostName` or `/System/System/ComputerName` Keys

**5. 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/dademurphy_mounted/Library/Preferences/

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

1. What kind of printer was used with this system?
    a. "Brother HL-2170W series" (`printer-name` or `printer-info` Key)
2. How was this printer accessed?
    a. Via USB (`device-url` Key)

- Use the `cd` command to navigate to the Printer Log directory.
- Use the `open` command to open the `page_log` file.

```
$ cd /Volumes/dademurphy_mounted/var/log/cups/

$ open page_log
```

3. What two documents were printed?
    a. "scrypt - Wikipedia, the free encyclopedia"
    b. "Litecoin Charts - Litecoin Cryptocurrency Blockchain Explorer"
4. How many pages long were these documents?
    a. Three & six pages
5. What user did the printing?
    a. Zerocool

- Get a root shell.
- Use the `cd` command to navigate to the Printer Spool directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory.
- Use the `file` command on the printer data files.
- Using the `cp` command, copy the `d000*` data files property list to a directory of your choice.
- Use the `chown` command to change the ownership to your user account name.
- Using the `open` command to view the PDF printer data files.

```
$ sudo -s

# cd /Volumes/dademurphy_mounted/private/var/spool/cups/

# ls -l

# file d0000*

# cp d0000* ~/FOR518

# chown <your username> ~/FOR518/d0000*

# open ~/FOR518/d0000*
```

6. What user did the printing?

Using the printer control files (`c0000#`), can you determine what application those two documents were printed from (hint: Look for an `ApplicationName`, try using the `strings` program)?

    a. Yes, Safari.app
        i. The control files contain a "`JobInfo.PMApplicationName`" string.

**6. 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/dademurphy_mounted/Library/Preferences/

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

1. When was the last successful update?
    a. Dec 22, 2013, 7:44:45 PM (EST/EDT) (local system time, `LastSuccessfulDate` Key)
2. System version when last updated?
    a. 10.8.5
3. When was the software update last attempted?
    a. Dec 22, 2013, 7:44:45 PM (EST/EDT) (local system time, `LastAttemptDate Key`)
        i. The same time is shown in both the `LastSuccessfulDate` and `LastAttemptDate` keys because not all updates were installed at this time.
4. What are the names of the three available software updates?
    a. Digital Camera RAW Compatibility Update
    b. Remote Desktop Client Update
    c. Safari
        i. 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/dademurphy_mounted/Library/Receipts/

$ ls -l

$ open InstallHistory.plist
```

5. How many updates are shown in the `InstallHistory.plist` file?
    a. 13 Items
6. Review Item 0, what do you think this entry is indicative of?
    b. System installation. The user appears to have installed this system on 11/17/13 installing OS X 10.8.5.
7. What application was used to install OS X?
    c. The OS X Installer
8. On what date was the "LanScan" application installed?
    d. 12/14/13 (EST/EDT) (Item 10)
9. What application was used to install this application?
    e. "storeagent" = The App Store
10. What application was installed on 11/20/13 (EST/EDT)?
    f. Adobe Flash Player
11. What application was used to install this application?
    g. "Installer"
        i. An external installer was used rather than the app store.
12. What application installed the Brother Printer Software Update?
    h. Software Update

- Use the `cd` command to navigate to 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 that 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 LanScan application.

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

$ ls -lt

$ open <anyfile>.plist

$ lsbom -s com.iwaxx.LanScan.bom
```

13. How many files are associated with the LanScan application?
    i. 19 files (Use the `wc` command against the output. Subtract one for the current directory '.' file.
       i. `lsbom -s com.iwaxx.LanScan.bom | wc`

**Extra Credit –**

- **Review the same files using the BlackLight application.**

***Exercise – Key Takeaways***

- **Know where the key files associated with the system are stored. These files are system-wide preferences as opposed to the previous labs where they were user-focused preferences.**
- **Get comfortable with some Mac OS X command line utilities.**

This page intentionally left blank.

# Exercise 3.2 – Log Analysis

## Objectives

- Know where the key log files are stored and how to parse the Apple System Logs and Basic Security Module Audit logs.
- Get familiar with the Mac OS X command line.

## 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/
   - 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 the Dade Murphy forensic (`dademurphy.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
     - Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
       - `Mount Point` – Newly created specifically for this image.
     - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to

suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.

- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
  - `-j` – Ignore the journal
  - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
  - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_image` is used in the example.
  - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_mounted` in used the example above.
  - Use `ewfmount` to mount the `dademurphy.e01` image to the `/Volumes/dadmurphy_image/` mount point.
  - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadeimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)
  - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
  - Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

4. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

***Exercise – Questions***

1. **System Log Directory & 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/dademurphy_mounted/private/var/log/

$ ls -l

$ file *
```

1. What two log files have been archived using BZip2 compression?

_____

_____

- Use the `bzcat` 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`.

```
$ bzcat system.log.1.bz2 system.log.0.bz2 >> ~/FOR518/system_all.log

$ cat system.log >> ~/FOR518/system_all.log
```

2. Use the `wc` command to determine how many records are now in the `system_all.log` file.

_____

3. Do the same for the wifi.log and wifi.log BZip2 compressed log files. How many records are found in these logs?

_____

## 2. Introduction to the Console Application

- Locate and open the native OS X Console.app from /Applications/Utilities/.
- This will show you the log contents of your host system.

- Review the left pane.
- **Briefly** review the log files under "FILES".
    i. Note the different locations where logs may be found.
- Select "All Messages" under "SYSTEM LOG QUERIES"; review the syslog events on <u>your</u> system.
    i. Select an event, and press the blue "i" Inspector button on the top the application.
        1. Review the contents in this popup window.
        2. You may select another event; the window should stay open for you to view its contents.

1. What is the Facility for this event?

_____

2. What is the User ID (UID) for this event?

_____

3. What is the Group ID (GID) for this event?

_____

4. What is the hostname of the system?

_____

5. What is the Sender for this event?

_____

6. When was this event logged (use the epoch time)?

_____

### 3. Apple System Log (ASL) Directory

- Get a root shell.
- Use the `cd` command to navigate to the Apple System Log directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory.

```
$ sudo -s

$ cd /Volumes/dademurphy_mounted/private/var/log/asl/

$ ls -l
```

1. What is the date of the oldest ASL log file (not including auxiliary or "best before" ASL files)?

_____

2. How many days in the past are events recorded as shown by the ASL filenames (not including auxiliary or "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.
- Use the `ls -l` command to view all files in this directory.

```
$ man syslog
```

- Use the `syslog` command to view the contents of the ASL log file containing data for UID 501 (`zerocool`) for December 22, 2013.

```
# syslog -f 2013.12.22.U501.asl
```

1. What is the date/time range of the timestamps in this syslog output?

_____

_____

- Use the `syslog` command to view the contents of the ASL log file containing data for UID 501 (`zerocool`) for December 22, 2013 using the UTC timestamp.
    i. Note the differences in the timestamp.

```
$ syslog -T utc -f 2013.12.22.U501.asl
```

- Use the `syslog` command to view the contents of the ASL log file containing data for UID 501 (`zerocool`) for December 22, 2013 using the UTC timestamp in RAW format.
    i. Note the differences in the log output. Recall the information you saw in the Console.app Inspector window.

```
$ syslog -F raw -T utc -f 2013.12.22.U501.asl
```

- Use the `syslog` command to output all the ASL logs in this directory using the UTC timestamp in RAW format.
    i. Redirect the output to a file `ASL.out` in your FOR518 directory.
- Open the `ASL.out` log in Console.app using the `open` command.
    i. Review this output.

```
$ syslog -F raw -T utc -d . > ~/FOR518/ASL.out
$ open -a Console ~/FOR518/ASL.out
```

2. What is the date (UTC) of the first message?

_____

3. What is the message level?

_____

4. What is the UID and GID of this message?

_____

5. What is the hostname used in this message?

_____

6. What is the facility?

_____

7. When does this message expire?

_____

8. How long is this message kept for?

_____

9. What is the date (UTC) of the last message?

_____

10. What is the message level?

_____

11. What is the hostname used in this message?

_____

12. What is the facility?

_____

13. When does this message expire?

_____

**5. Basic Security Module Audit Logs**

- Temporarily change the time zone of your Terminal window using the "`export TZ="EST5EDT""` command. The system time for the `dademurphy.e01` image is EST/EDT. (Those users who are already using EST/EDT on their hosts system will not have to run this command, however it should not hinder if you do.)
- Use the `cd` command to navigate to the Audit Log directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory.

```
$ export TZ="EST5EDT"
$ cd /Volumes/dademurphy_mounted/private/var/audit/
$ ls -l
```

1. What is the start timestamp of the oldest audit log file?

_____

2. What is the end timestamp of this same file?

_____

- Use the `praudit` command to output the contents of the 20131117181352.20131117200025 audit log.
    i. Use the `less` command to control the output.
- Review the output of this command.

```
$ praudit 20131117181352.20131117200025 | less
```

- Use the `praudit` command to output the contents of the 20131117181352.20131117200025 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 20131117181352.20131117200025 | 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. **A username on your system** 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.)
    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 20131117181352.20131117200025 | 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.out`.
    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.
- Do not forget to exit out of your root shell by typing exit.

```
$ praudit -xn *.* > ~/FOR518/audit.out

$ open -a Console ~/FOR518/audit.out

$ exit
```

- 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.

3. When was the user `zerocool` created (Search "`create user`")?

_____

4. Find the one "user authentication" event recorded on Sun Nov 17 14:54:41 2013. What user authenticated to the system?

_____

5. What is the UID associated with this event?

_____

6. What is the GID associated with this event?

_____

7. Looking at the next two records, what was the user likely authenticating to do?

_____

***** If you have changed your time zone in this terminal window, be sure to either change it back or exit and open a new Terminal window to reset the time zone. *****

**1. System Log Directory & 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/dademurphy_mounted/private/var/log/

$ ls -l

$ file *
```

1. What two log files have been archived using BZip2 compression?
   a. system.log
   b. wifi.log

- Use the `bzcat` 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`.

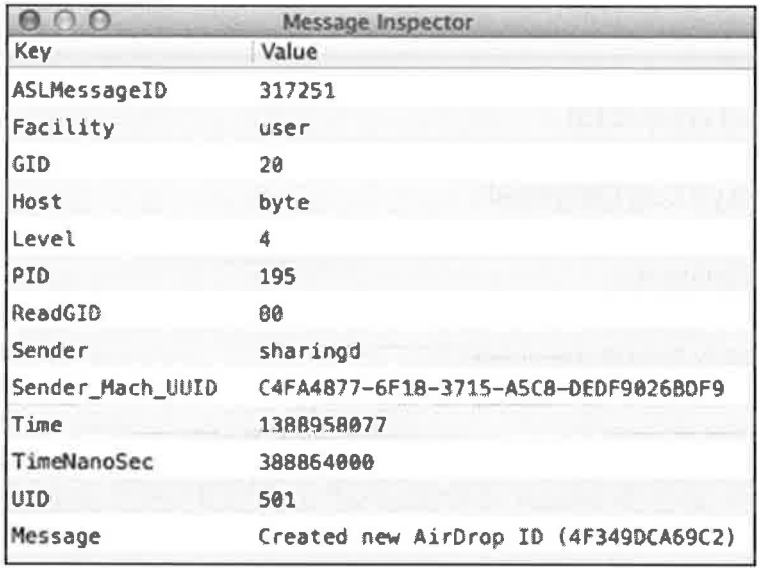
```
$ bzcat system.log.1.bz2 system.log.0.bz2 >> ~/FOR518/system_all.log

$ cat system.log >> ~/FOR518/system_all.log
```

2. Use the `wc` command to determine how many records are now in the `system_all.log` file.
   a. 9422 records (`wc ~/FOR518/system_all.log`)
3. Do the same for the wifi.log and wifi.log BZip2 compressed log files. How many records are found in these logs?
   a. 247 records
      i. Use the same commands above; change the filename system.log to wifi.log.

**2. Introduction to the Console Application**

- Locate and open the native OS X Console.app from /Applications/Utilities/.
- This will show you the log contents of your host system.

- Review the left pane.
- **Briefly** review the log files under "FILES".
    i. Note the different locations where logs may be found.
- Select "All Messages" under "SYSTEM LOG QUERIES"; review the syslog events on <u>your</u> system.
    i. Select an event, and press the blue "i" Inspector button on the top the application.
        1. Review the contents in this popup window.
        2. You may select another event; the window should stay open for you to view its contents.




1. What is the Facility for this event?
    a. "user"
2. What is the User ID (UID) for this event?
    a. 501
3. What is the Group ID (GID) for this event?
    a. 20
4. What is the hostname of the system?
    a. "byte"
5. What is the Sender for this event?
    a. "sharingd"
6. When was this event logged (use the epoch time)?
    a. 1388958077
        i. Use the date –ur command or Epoch Converter to convert it to human-readable UTC format: Sun Jan 5 21:41:17 UTC 2014

### 3. Apple System Log (ASL) Directory

- Get a root shell.
- Use the `cd` command to navigate to the Apple System Log directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory.

```
$ sudo -s

$ cd /Volumes/dademurphy_mounted/private/var/log/asl/

$ ls -l
```

1. What is the date of the oldest ASL log file (not including auxiliary or "best before" ASL files)?
    a. 12/17/2013
2. How many days in the past are events recorded as shown by the ASL filenames (not including auxiliary or "best before" ASL files)?
    a. Seven (12/17/2013 – 12/23/2013)

**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.
- Use the `ls -l` command to view all files in this directory.

```
$ man syslog
```

- Use the `syslog` command to view the contents of the ASL log file containing data for UID 501 (`zerocool`) for December 22, 2013.

```
$ syslog -f 2013.12.22.U501.asl
```

1. What is the date/time range of the timestamps in this syslog output?
    a. Dec 22 16:36:41
    b. Dec 22 17:36:36

- Use the `syslog` command to view the contents of the ASL log file containing data for UID 501 (`zerocool`) for December 22, 2013 using the UTC timestamp.
    i. Note the differences in the timestamp.

```
$ syslog -T utc -f 2013.12.22.U501.asl
```

- Use the `syslog` command to view the contents of the ASL log file containing data for UID 501 (`zerocool`) for December 22, 2013 using the UTC timestamp in RAW format.
    i. Note the differences in the log output. Recall the information you saw in the Console.app Inspector window.

```
$ syslog -F raw -T utc -f 2013.12.22.U501.asl
```

- Use the `syslog` command to output all the ASL logs in this directory using the UTC timestamp in RAW format.
    i. Redirect the output to a file `ASL.out` in your FOR518 directory.
- Open the `ASL.out` log in Console.app using the `open` command.
    i. Review this output.

```
$ syslog -F raw -T utc -d . > ~/FOR518/ASL.out

$ open -a Console ~/FOR518/ASL.out
```

2. What is the date (UTC) of the first message?
    a. 2013-11-17 18:13:52Z
3. What is the message level?
    a. 5 (Notice)
4. What is the UID and GID of this message?
    a. 0
5. What is the hostname used in this message?
    a. "localhost"
6. What is the facility?
    a. com.apple.system.utmpx
7. When does this message expire?
    a. 1416334432 = Tue Nov 18 18:13:52 UTC 2014
        i. ASLExpireTime Field
8. How long is this message kept for?
    a. One Year + 1 Day (366 days or 31622400 seconds via `man asl.conf`)
9. What is the date (UTC) of the last message?
    a. 2013-12-23 18:23:55Z
10. What is the message level?
    a. 7 (Debug)
11. What is the hostname used in this message?
    a. "Dades-MacBook-Air"
12. What is the facility?
    a. "kern"
13. When does this message expire?
    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

- Temporarily change the time zone of your Terminal window using the "`export TZ="EST5EDT""` command. The system time for the `dademurphy.e01` image is EST/EDT.

(Those users who are already using EST/EDT on their hosts system will not have to run this command, however it should not hinder if you do.)

- Use the `cd` command to navigate to the Audit Log directory.
- Use the `ls -l` command to view all files in this directory. Note the contents of this directory.

```
$ export TZ="EST5EDT"

$ cd /Volumes/dademurphy_mounted/private/var/audit/

$ ls -l
```

1. What is the start timestamp of the oldest audit log file?
   a. 20131117181352 = 11/17/2013 18:13:52
2. What is the end timestamp of this same file?
   a. 20131117200025 = 11/17/2013 20:00:25

- Use the `praudit` command to output the contents of the 20131117181352.20131117200025 audit log.
  i. Use the `less` command to control the output.
- Review the output of this command.

```
$ praudit 20131117181352.20131117200025 | less
```

- Use the `praudit` command to output the contents of the 20131117181352.20131117200025 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 20131117181352.20131117200025 | 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. **A username on your system** 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.)
  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 20131117181352.20131117200025 | 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.out`.
    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.
- Do not forget to exit out of your root shell by typing exit.

```
$ praudit -xn *.* > ~/FOR518/audit.out

$ open -a Console ~/FOR518/audit.out

$ exit
```

- 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.

3. When was the user `zerocool` created (Search "`create user`")?
    a. Sun Nov 17 13:40:37 2013
4. Find the one "user authentication" event recorded on Sun Nov 17 14:54:41 2013. What user authenticated to the system?
    a. zerocool
5. What is the UID associated with this event?

    a. 501
6. What is the GID associated with this event?
    a. 20
7. Looking at the next two records, what was the user likely authenticating to do?
    a. Software Update to the App Store

***** If you have changed your time zone in this terminal window, be sure to either change it back or exit and open a new Terminal window to reset the time zone. *****

### *Exercise – Key Takeaways*

- **Know how to parse these log files by hand; most tools do not parse these automatically.**
- **Get comfortable with some Mac OS X command line utilities.**

# Exercise 3.3 – Timeline Analysis & Data Correlation

## Objectives

- Get familiar with correlating events in a time line using log analysis and data correlation of key OS X data files.

## 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/
   - 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 the Dade Murphy forensic (`dademurphy.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
     - Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
       - `Mount Point` – Newly created specifically for this image.
     - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.

- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
    - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_image` is used in the example.
    - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_mounted` in used the example above.
    - Use `ewfmount` to mount the `dademurphy.e01` image to the `/Volumes/dadmurphy_image/` mount point.
    - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadeimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)
    - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
    - Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
        - `-j` – Ignore the journal
        - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

4. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

## *Exercise – Questions*

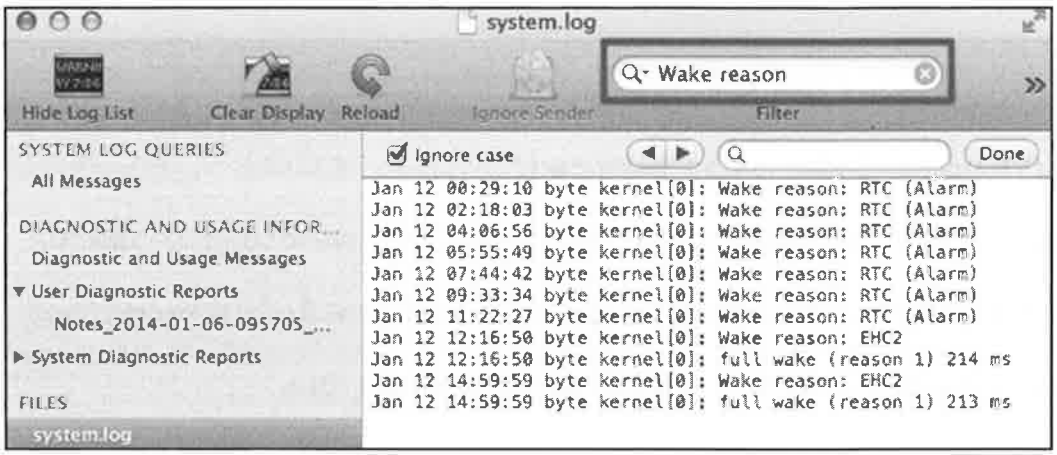
1. **Choose Your Own Adventure Log Analysis!**

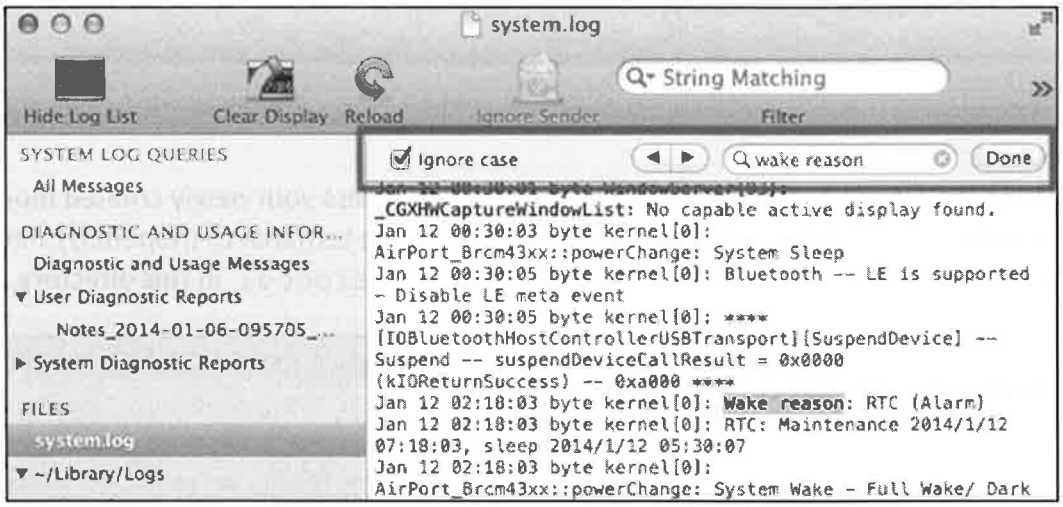
   **Choose one of two choices:**
   - **Choice A – Use Console.app**
   - **Choice B – Use the Command-line**

   - **Choice A – Console.app.**
     i. Use the `open` command to open the log file of interest in Console.app.

```
$ open -a Console <example>.log
```

     ii. Use the search functions:
       1. Filter textbox at the top-right
       2. "Find" Function - Edit | Find (Command-F)




- **Choice B – Via the Command-line using `grep`**
  i. 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. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
  - ii. /Volumes/dademurphy_mounted /private/var/log/daily.out
  - iii. /Volumes/dademurphy_mounted /Users/zerocool/Library/Preferences/com.apple.finder.plist
  - iv. /Volumes/dademurphy_mounted /Users/zerocool/Library/Preferences/com.apple.sidebarlists.plist

1. What is the USBMSC identifier for the device mounted on December 22, 2013?

_____

2. Was this device used previously, if so when?

_____

_____

3. What is the volume name of this device?

_____

4. What format is this volume?

| |
|---|
| BDxF - ExFAT |
| BDIS – FAT32 |
| BDcu – UDF (DVD) |
| NTcu – Unknown |
| H+ - HFS+ |
| KG - FTP |

_____

5. What type of disk is this volume located on (review the list in your 518.3 book)?

_____

## 3. System Information & State

- Review these files.
  - i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
  - ii. /Volumes/dademurphy_mounted /private/var/log/daily.out

1. How many times was this system booted?

_____

2. When did the user wake the system November 20, 2013?

_____

_____

3. What kind of hard drive was the system booted from?

_____

4. What percentage of the boot hard drive was allocated on 12/15/2013?

_____

5. What percentage of the boot drive was allocated on 12/23/2013?

_____

**4. Network Analysis**

- Review these files.
  i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
  ii. /Volumes/dademurphy_mounted/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist
  iii. Internet History (use BlackLight, or files noted in Exercise 2.2)

1. What four Wi-Fi networks did this system associate to?

_____

_____

_____

_____

2. What two Wi-Fi networks did this system attempt to associate to?

_____

_____

3. Create a timeline of travel activity:

| Timestamp | WiFi SSID | IP | Likely Location |
|---|---|---|---|
| **11/17/2013 10:38:39** | | | |
| **11/19/2013 21:36:34** | | | |
| **11/20/2013 06:39:01** | | | |
| **11/20/2013 15:43:09** | | | |
| **12/12/2013 17:42:13** | | | |
| **12/14/2013 20:01:01** | | | |
| **12/23/2013 13:20:05** | | | |

**5. User Access**

- Review these files.
  i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
  ii. /Volumes/dademurphy_mounted/private/var/log/asl (remember you should have the full log created in Exercise 3.2)
  iii. /Volumes/dademurphy_mounted/private/var/audit (remember you should have the full log created in Exercise 3.2)

1. What two methods did users use to log onto this system?

_____

_____

2. What is the start time and end time of the logon session associated with PID 1685?

_____

3. What user account logged on at this time?

_____

**6. Software Installation**

- Review these files.
  i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
  ii. /Volumes/dademurphy_mounted/private/var/log/install.log

1. When was the Skitch application installed?

_____

2. When was this system upgraded from 10.8?

_____

7. **Backup Activity**
   - Review these files.
     i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)

1. When was the first Time Machine backup started?

_____

2. What type of backups were started on December 16th?

_____

3. What type of backup was started on December 22nd?

_____

**Extra Credit –**

- **Keep reviewing log files, including those not included in this exercise – get comfortable with the different types of events in each.**
- **Review these files in the BlackLight application.**

1. **Choose Your Own Adventure Log Analysis**

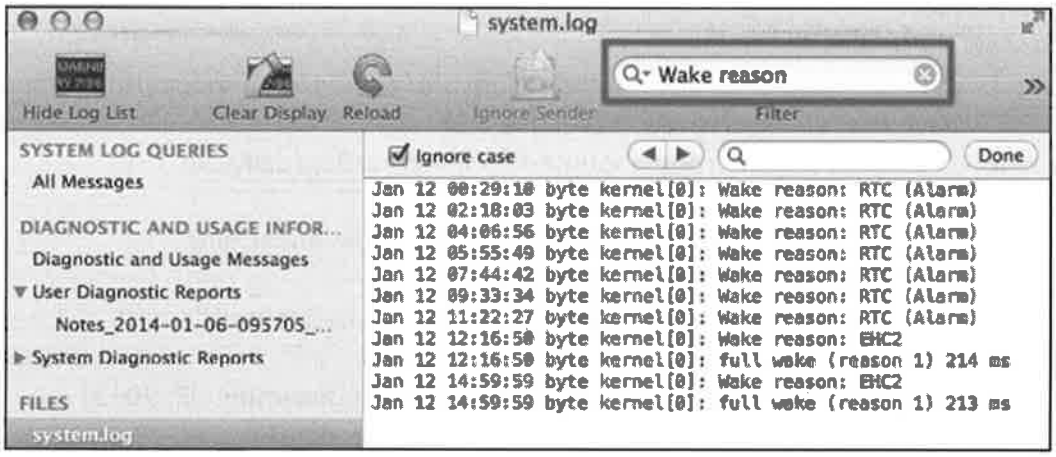
**Choose one of two choices:**

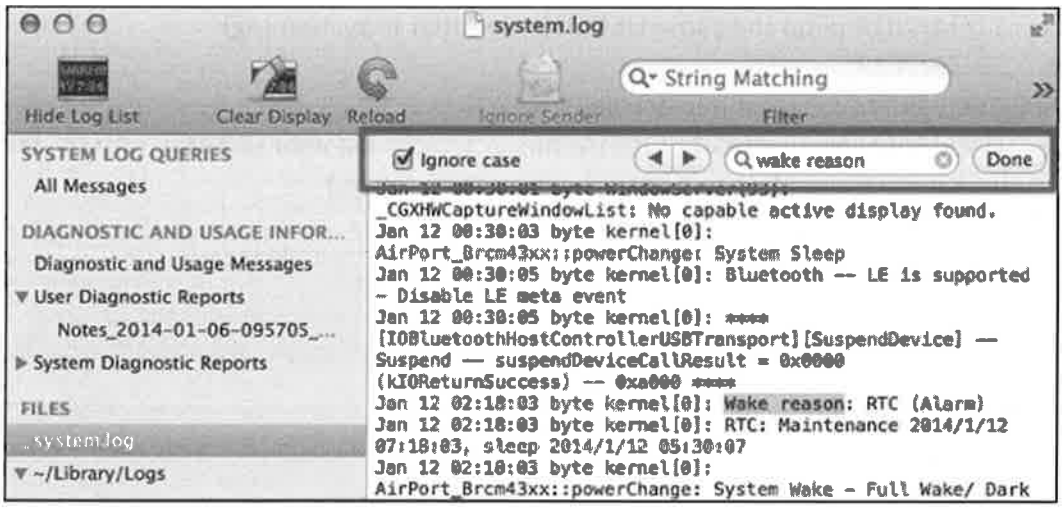
- **Choice A – Use Console.app**
- **Choice B – Use the Command-line**

- **Choice A – Console.app.**
  i. Use the `open` command to open the log file of interest in Console.app.

```
$ open -a Console <example>.log
```

ii. Use the search functions:
  1. Filter textbox at the top-right
  2. "Find" Function - Edit | Find (Command-F)



- **Choice B – Via the Command-line using `grep`**
    i. 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. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
    ii. /Volumes/dademurphy_mounted /private/var/log/daily.out
    iii. /Volumes/dademurphy_mounted /Users/zerocool/Library/Preferences/com.apple.finder.plist
    iv. /Volumes/dademurphy_mounted /Users/zerocool/Library/Preferences/com.apple.sidebarlists.plist

1. What is the USBMSC identifier for the device mounted on December 22, 2013?
    a. DEF10BEC448D (Search for USBMSC entries on this date)
2. Was this device used previously, if so when?
    a. 12/14/2013 (Find the same USBMSC identifier in system.log)
    b. 12/16/2013
3. What is the volume name of this device?
    a. TIMEMACHINE (Correlate timestamps in system.log with USBMSC entries, you may have to look before and after this entry to get a better idea.)
4. What format is this volume?

| |
|---|
| BDxF - ExFAT |
| BDIS – FAT32 |
| BDcu – UDF (DVD) |
| NTcu – Unknown |
| H+ - HFS+ |
| KG - FTP |

    a. HFS+ (Extract the Alias BLOB data from com.apple.sidebarlists.plist)
5. What type of disk is this volume located on (review the list in your 518.3 book)?

a. USB Hard Drive (Entry Type 517 found in com.apple.sidebarlists.plist)

**3. System Information & State**

- Review these files.
  i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
  ii. /Volumes/dademurphy_mounted /private/var/log/daily.out

1. How many times was this system booted?
   a. Seven (Search "boot" in system.log)
2. When did the user wake the system November 20, 2013?
   a. 06:38:49 (Search "wake reason" in system.log)
   b. 15:40:35
3. What kind of hard drive was the system booted from?
   a. Apple SSD TS128E (Search "boot-uuid", search for boot device information within same context in system.log)
4. What percentage of the boot hard drive was allocated on 12/15/2013?
   a. 49% (daily.out - /dev/disk0s2)
5. What percentage of the boot drive was allocated on 12/23/2013?
   a. 79% (daily.out - /dev/disk0s2)

**4. Network Analysis**

- Review these files.
  i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
  ii. /Volumes/dademurphy_mounted/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist
  iii. Internet History (use BlackLight, or files noted in Exercise 2.2)

1. What four Wi-Fi networks did this system associate to?
   a. cyberdelia (Search "associated" in system.log)
   b. Washington Dulles Wi-Fi
   c. #SFO FREE WIFI
   d. hhonors
2. What two Wi-Fi networks did this system attempt to associate to?
   a. 0xDEADBEEF (Search "airportd" in system.log)
   b. Sophie

3. Create a timeline of travel activity:

| Timestamp | WiFi SSID<br>Search "airportd" in | IP<br>Search | Likely Location |
|---|---|---|---|

| | system_all.log | "configd" in system_all.log | |
|---|---|---|---|
| **11/17/2013 10:38:39** | cyberdelia | 192.168.2.101 | Home |
| **11/19/2013 21:36:34** | cyberdelia | 192.168.2.101 | Home |
| **11/20/2013 06:39:01** | Washington Dulles WiFi | 10.59.3.48 | Washington Dulles Airport (IAD) |
| **11/20/2013 15:43:09** | #SFO FREE WIFI | 172.31.44.108 | San Francisco Airport (SFO) |
| **12/12/2013 17:42:13** | hhonors | 192.168.9.70 | Hilton Hotel WiFi |
| **12/14/2013 20:01:01** | cyberdelia | 192.168.2.101 | Home |
| **12/23/2013 13:20:05** | cyberdelia | 192.168.2.102 | Home |

5. **User Access**
   - Review these files.
     i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
     ii. /Volumes/dademurphy_mounted/private/var/log/asl (remember you should have the full log created in Exercise 3.2)
     iii. /Volumes/dademurphy_mounted/private/var/audit (remember you should have the full log created in Exercise 3.2)

1. What two methods did users use to log onto this system?
   a. Login Window (Search "_PROCESS:" in system.log)
   b. Terminal
2. What is the start time and end time of the logon session associated with PID 1685?
   a. Dec 15 13:54:41 -> Dec 15 13:56:18
3. What user account logged on at this time?
   a. zerocool (Search for "1685" or timestamps in ASL.out or audit.out)

6. **Software Installation**
   - Review these files.
     i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)
     ii. /Volumes/dademurphy_mounted/private/var/log/install.log

1. When was the Skitch application installed?
   a. Dec 14 20:03:57 (Search "Installed" in install.log)
2. When was this system upgraded from 10.8?
   a. Nov 17 09:59:29 (10.8.5) (Search "Build:" in install.log)

7. **Backup Activity**
   - Review these files.
     i. /Volumes/dademurphy_mounted/private/var/log/system.log (remember you should have the full log created in Exercise 3.2)

1. When was the first Time Machine backup started?
   a. Dec 14 21:08:50 (Search "backupd" in system.log)
2. What type of backups were started on December 16th?
   a. Automatic (Search "backupd" and "starting" in system.log)

3. What type of backup was started on December 22nd?
    a. Manual

**Extra Credit –**

- **Keep reviewing log files, including those not included in this exercise – get comfortable with the different types of events in each.**
- **Review these files in the BlackLight application.**

## *Exercise – Key Takeaways*

- **Using various log files and data files, correlation can be done to prove or disprove different activities.**

This page intentionally left blank.

# Exercise 4.1 – Time Machine & Spotlight

## Objectives

- **Review and analyze files associated with Time Machine backups.**
- **Review a Time Machine backup volume.**
- **Review and analyze the Spotlight Index database.**

## 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/
   - Spotlight Inspector
     i. Locate and open the Spotlight Inspector application that was installed in Exercise 0.
     ii. This file can be downloaded from http://www.504ensics.com/tools/digital-forensics-tool-spotlight-inspector/
   - The Sleuth Kit
     i. The Sleuth Kit application was installed in Exercise 0.
     ii. TSK can be downloaded from http://www.sleuthkit.org/sleuthkit/download.php

2. **Exercise File Preparation** – Locate the files located in the `Exercise Files/Exercise 4.1 - Time Machine & Spotlight` directory on your FOR518 USB drive.

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. **Mount the Dade Murphy forensic (`dademurphy.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
     - Uses `xmount` to mount the `dademurphy.E01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
    - `Mount Point` – Newly created specifically for this image.
- Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
    - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_image` is used in the example.
    - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_mounted` in used the example above.
    - Use `ewfmount` to mount the `dademurphy.E01` image to the `/Volumes/dadmurphy_image/` mount point.
    - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadeimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)
    - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.

- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
  - `-j` – Ignore the journal
  - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

5. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

6. **Mount the Dade Murphy Time Machine forensic (`dade_timemachine.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_tm_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_tm_mounted` is used in this class to represent the mounted disk image.

- Uses `xmount` to mount the `dade_tm.E01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
  - `Mount Point` – Newly created specifically for this image.
- Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command. Use the "`Apple_HFS`" partition.
- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_tm_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
  - `-j` – Ignore the journal
  - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_tm_mounted/`.

```
$ mkdir /Volumes/dademurphy_tm_image/

$ mkdir /Volumes/dademurphy_tm_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dade_timemachine.E01
/Volumes/dademurphy_tm_image/

$ hdiutil attach -nomount
/Volumes/dademurphy_tm_image/dade_timemachine.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_tm_mounted/
```

- **Method 2:**
  - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_tm_image` is used in the example.
  - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_tm_mounted` in used the example above.
  - Use `ewfmount` to mount the `dade_timemachine.E01` image to the `/Volumes/dadmurphy_tm_image/` mount point.
  - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadetmimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)

- Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command. Use the "`Apple_HFS`" partition.
- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_tm_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
  - `-j` – Ignore the journal
  - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_tm_mounted/`.

```
$ mkdir /Volumes/dademurphy_tm_image/

$ mkdir /Volumes/dademurphy_tm_mounted/

$ ewfmount ~/FOR518/dade_timemachine.E01 /Volumes/dademurphy_tm_image/

$ ln -s /Volumes/dademurphy_tm_image/ewf1 ~/FOR518/dadetmimage.dmg

$ hdiutil attach -nomount ~/FOR518/dade_tmimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_tm_mounted/
```

7. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see the "`Backsups.backupdb`" directory.

```
$ cd /Volumes/dademurphy_tm_mounted/

$ ls -l
```

***Exercise – Questions***

1. **Time Machine - 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/dademurphy_mounted/Library/Preferences/

$ open com.apple.TimeMachine.plist
```

1. When was the last backup performed?

_____

2. What directory is excluded from the Time Machine backup?

_____

3. How many snapshots have been created?

_____

4. When was the first complete backup performed?

_____

5. Are the Time Machine backups stored on a network or external hard drive?

_____

6. What is the first section of the Destination UUID for these snapshots?

_____

**2. Time Machine - Review the Time Machine Snapshot Data**

- Use the `cd` command to explore the /var/db/ (/private/var/db/) directory.
- Use the `open` command to open the `com.apple.TimeMachine.SnapshotDates.plist` file.

```
$ cd /Volumes/dademurphy_mounted/var/db/

$ open com.apple.TimeMachine.SnapshotDates.plist
```

1. How many snapshots have been created?

_____

2. What is the first section of the Destination UUID for these snapshots?

_____

- Use the `open` command to open the `.TimeMachine.Results.plist` file and review the contents of this file.

```
$ open .TimeMachine.Results.plist
```

3. **Time Machine – Volume UUID (Connect a Time Machine backup disk to a specific OS X System)**
   - Use the `cd` command to explore the root of the Time Machine volume.
   - Use the `ls -la` command to view the contents of this directory. Note the contents of this directory.
   - Use the `cat` command to view the contents of the `.apdisk` file.

```
$ cd /Volumes/dademurphy_tm_mounted/

$ ls -la

$ cat .apdisk
```

1. What is the first section of the Disk UUID of this volume?

   _____

2. What is the name of this volume?

   _____

4. **Time Machine - 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 Backups.backupdb/

$ xattr -xl Dade's\ MacBook\ Air/
```

1. What is the MAC address of the backed up system?

   _____

2. What is the make and model of the backed up system?

   _____

5. **Time Machine - Review the Time Machine – Snapshot Metadata**
   - Use the `cd` command to explore the "Dade's MacBook Air" directory.
   - Use the `xattr -xl` command to view the extended attributes of the 2013-12-23-133929 snapshot.

```
$ cd /Volumes/dademurphy_tm_mounted/Backups.backupdb/Dade's\ MacBook\
Air/

$ xattr -xl 2013-12-23-133929/
```

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, Daily, Monthly)?

_____

5. How many bytes were copied in this snapshot?

_____

6. **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.

```
$ sudo 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 "Dade's MacBook Air/" for the current directory.

```
$ tmutil calculatedrift .
```

2. Which snapshots had the most data added?

_____

   - Use the `tmutil compare` command to compare two snapshots:
     i. `2013-12-22-164115`
     ii. `2013-12-23-133929`
   - Output this to a file in your `FOR518` directory named, `tm_compare.txt`.
   - Use the open command to view this file.

```
$ tmutil compare 2013-12-22-164115 2013-12-23-133929 >
~/FOR518/tm_compare.txt

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

| | |
|---|---|
| ! | Metadata Changed |
| + | File Added |
| - | File Removed |

3. What two files were created in zerocool's `Downloads` directory?

_____

_____

4. What attributes were changed on the file `Macintosh HD/Users/zerocool/Library/Preferences/com.apple.Safari.plist`?

_____

_____

5. How much data was removed in this snapshot?

_____

**7. Spotlight – Review the Spotlight Metadata**

- Use the `cd` command to explore the zerocool's Downloads directory.
- Use the `mdls` command to view the Spotlight metadata associated with the `Firefox 26.0.dmg` file.
    i. **Optional**: Use the `mdimport -A` command to show more information about the Spotlight metadata tags.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Downloads/

$ mdls Firefox\ 26.0.dmg
```

1. When was this file downloaded?

_____

2. What domain was this file downloaded from?

_____

_____

3. What link did the user click to download this file?

8. **Spotlight – Review the Spotlight Directory**
   - Use the `cd` command to explore the `Macintosh HD` directory for the `dademurphy.E01` image.
   - Use the `ls -la` command to view the contents of this directory. Review the contents of this 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.

```
$ cd /Volumes/dademurphy_mounted/
$ ls -la
$ cd .Spotlight-V100
$ ls -la
$ open VolumeConfiguration.plist
```

1. What is the first part of the GUID for the Spotlight Store for the root volume?

---

   - Use the `cd` command to enter the Spotlight store for the root volume.
   - Use the `ls -la` command to view the contents of this directory. Review the contents of this directory.

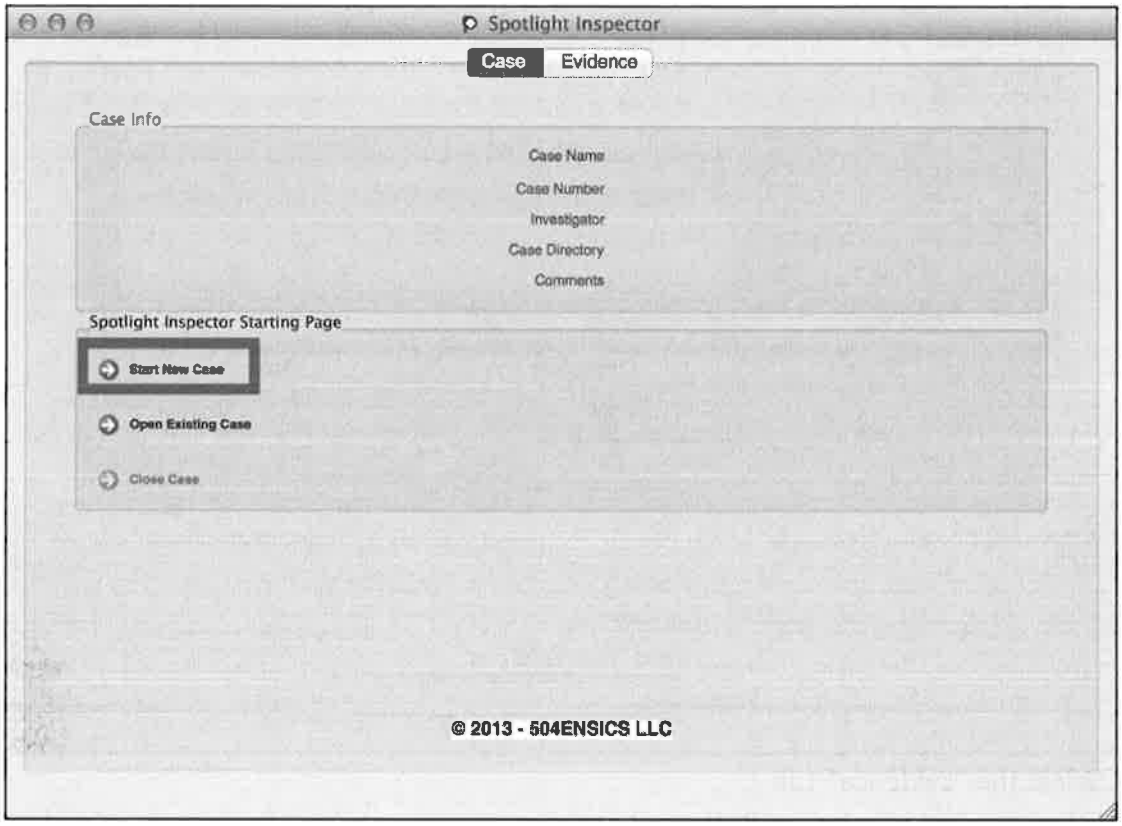
```
$ cd Store-V2/0A00840B-5B8C-44DC-9337-B6B58F5D5607
$ ls -la
```

   - Use TSK's `icat` command to extract the store.db file from the dademurphy.E01 image. The inode number for this file is 325602. Copy the file into your FOR518 directory and name it `store.db`.

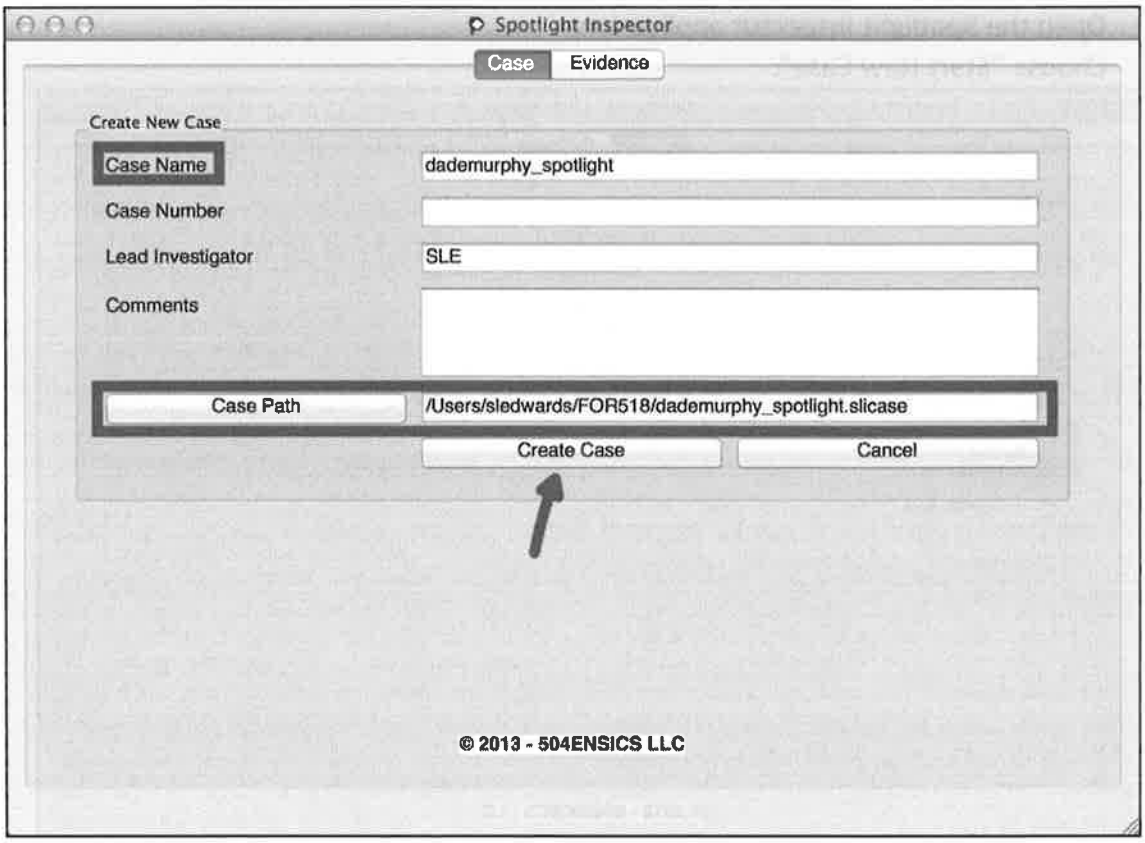
```
$ icat ~/FOR518/dademurphy.E01 325602 > ~/FOR518/store.db
```

9. **Spotlight – Review the Spotlight with Spotlight Inspector**

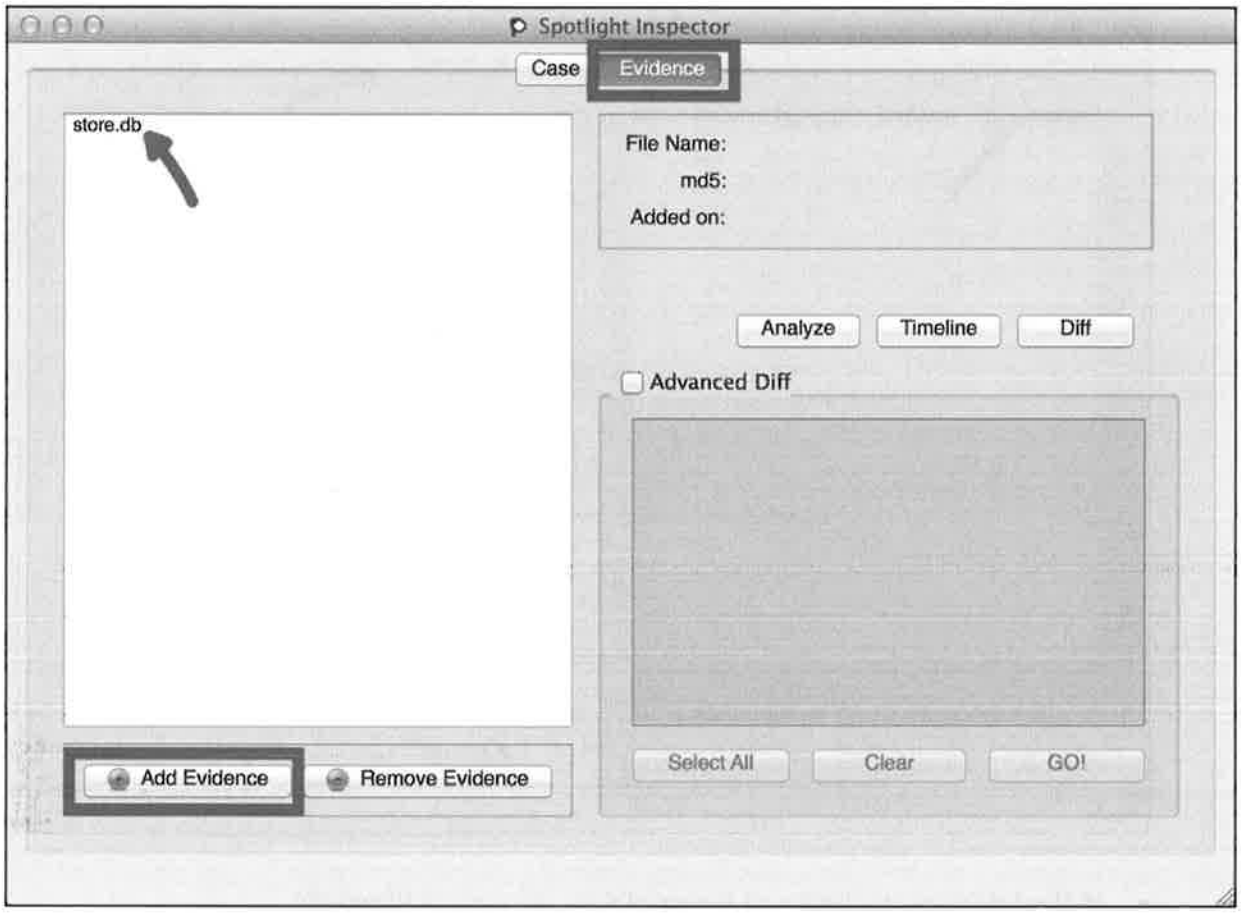
- Open the Spotlight Inspector application.
- Choose "Start New Case".



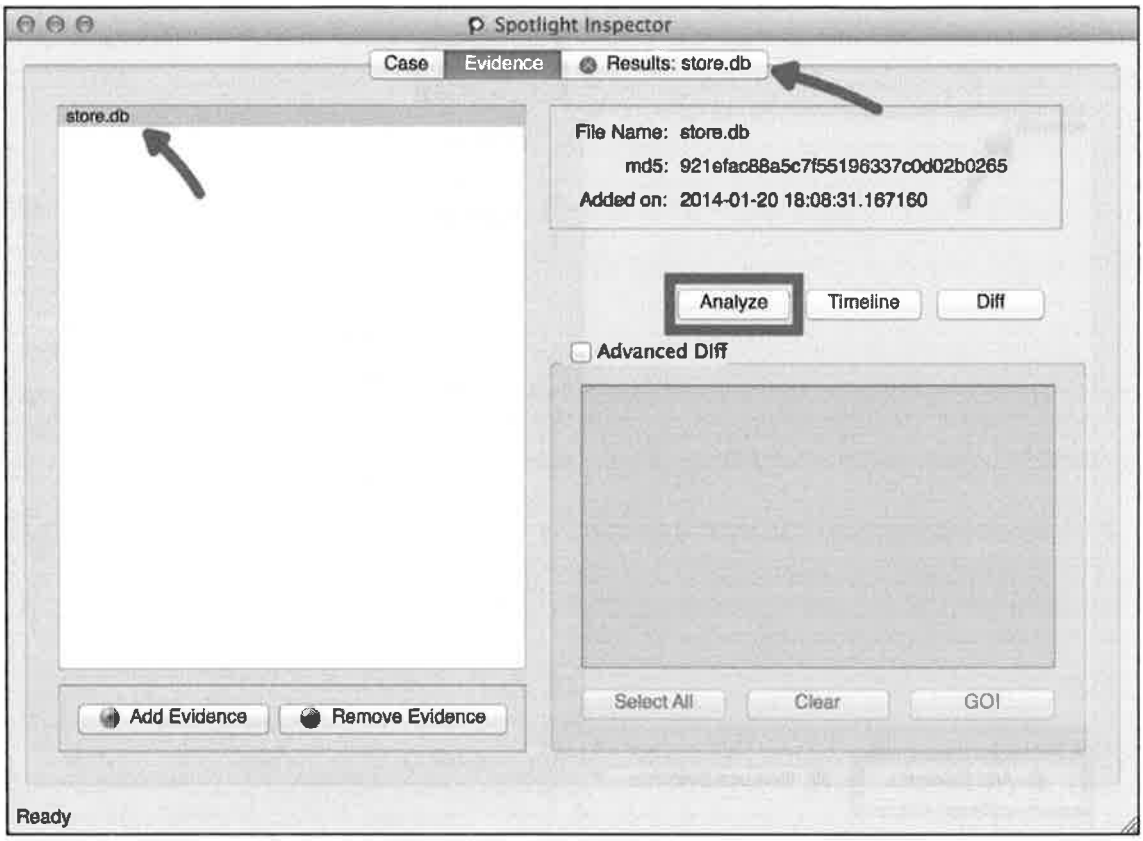

- Input a Case Name.
- Input a Case Path (You may choose to put it in your FOR518 directory).
- Select "Create Case".



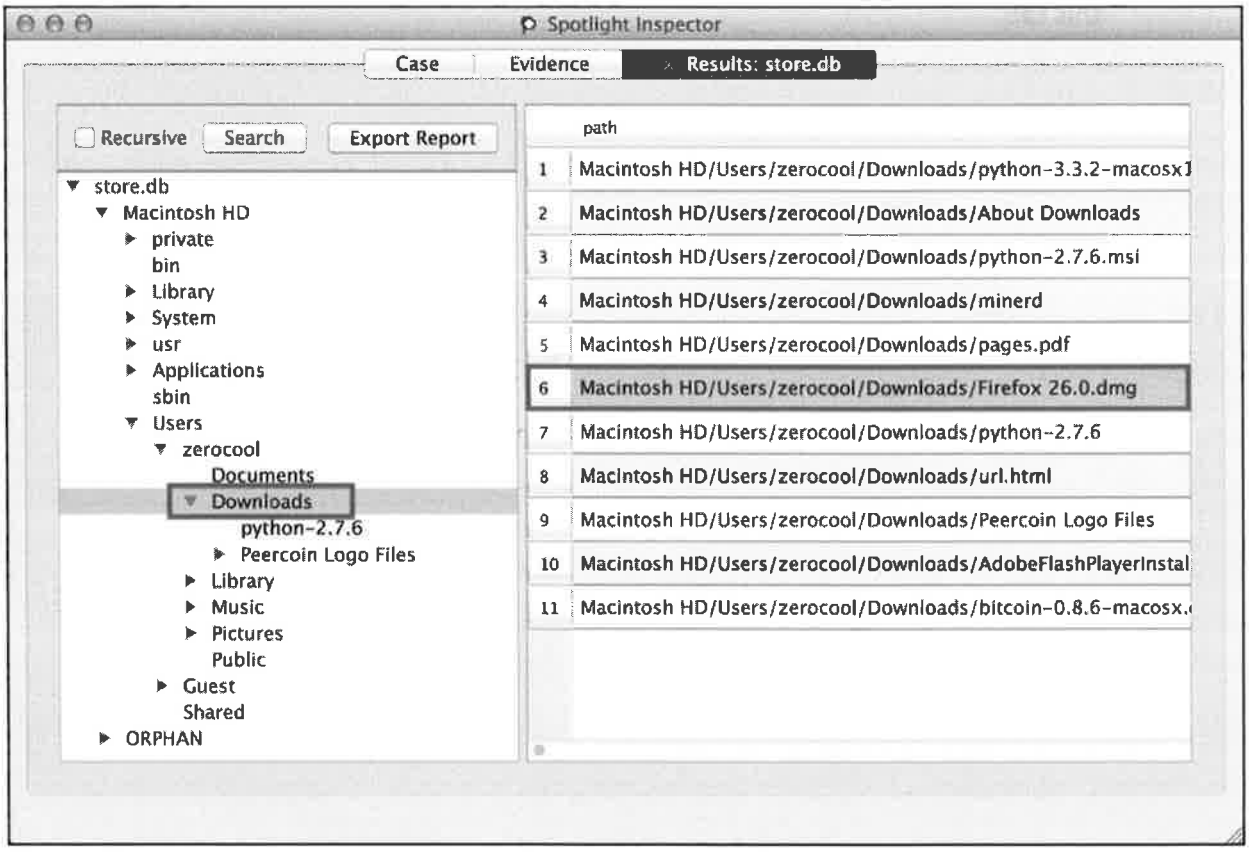

- Select the "Evidence" Tab.
- Select the "Add Evidence" Button; choose the `store.db` file you just extracted from the `dademurphy.E01` image.
- You should now see the `store.db` file in the window.




- Select the `store.db` file and click the "Analyze" Button.
- Once analysis is finished, you should see a new tab called "`Results: store.db`" - Select this tab.




- In the left pane, navigate to zerocool's `Downloads` directory.
- Find the `Firefox 26.0.dmg` file and review the contents of this tuple. Note the similarities from the `mdls` output.



- Find the photo in the file path:
  i. `Macintosh HD/Users/zerocool/Pictures/iPhoto Library/Masters/2013/12/16/20131216-192257/IMG_0003.JPG`
  ii. Review the contents of this tuple.

**Extra Credit**

- **Get familiar with the Spotlight Inspector application.**
  - **Use the Timeline function on the main Evidence Tab.**

***Exercise – Step-By-Step***

1. **Time Machine - 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/dademurphy_mounted/Library/Preferences/

$ open com.apple.TimeMachine.plist
```

1. When was the last backup performed?
   a. Dec 22, 2013, 4:41:15 PM (EST)
2. What directory is excluded from the Time Machine backup?
   a. /Users/Shared/adi
3. How many snapshots have been created?
   a. Five
4. When was the first complete backup performed?
   a. Dec 14, 2013, 9:26:29 PM (EST)
5. Are the Time Machine backups stored on a network or external hard drive?
   a. External
6. What is the first section of the Destination UUID for these snapshots?
   a. 652BBBB8

2. **Time Machine - Review the Time Machine Snapshot Data**
   - Use the `cd` command to explore the /var/db/ (/private/var/db/) directory.
   - Use the `open` command to open the `com.apple.TimeMachine.SnapshotDates.plist` file.

```
$ cd /Volumes/dademurphy_tm_mounted/var/db/

$ open com.apple.TimeMachine.SnapshotDates.plist
```

1. How many snapshots have been created?
    a. Five
2. What is the first section of the Destination UUID for these snapshots?
    a. 652BBBB8

- Use the `open` command to open the `.TimeMachine.Results.plist` file and review the contents of this file.

```
$ open .TimeMachine.Results.plist
```

**3. Time Machine – Volume UUID (Connect a Time Machine backup disk to a specific OS X System)**

- Use the `cd` command to explore the root of the Time Machine volume.
- Use the `ls -la` command to view the contents of this directory. Note the contents of this directory.
- Use the `cat` command to view the contents of the `.apdisk` file.

```
$ cd /Volumes/dademurphy_tm_mounted/

$ ls -la

$ cat .apdisk
```

1. What is the first section of the Disk UUID of this volume?
    a. 652BBBB8
2. What is the name of this volume?
    a. TIMEMACHINE

**4. Time Machine - 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 Backups.backupdb/

$ xattr -xl Dade's\ MacBook\ Air/
```

1. What is the MAC address of the backed up system?
    a. 7c:d1:c3:df:64:67
2. What is the make and model of the backed up system?
    a. MacBookAir5,1

**5. Time Machine - Review the Time Machine – Snapshot Metadata**

- Use the `cd` command to explore the "Dade's MacBook Air" directory.

- Use the `xattr -xl` command to view the extended attributes of the 2013-12-23-133929 snapshot.

```
$ cd /Volumes/dademurphy_tm_mounted/Backups.backupdb/Dade's\ MacBook\
Air/

$ xattr -xl 2013-12-23-133929/
```

1. What is the Snapshot number?
    a. 451
2. When did the backup start (in UTC)?
    a. 1387823964413215 = 2013-12-23 18:39:24 Mon UTC
    b. Take the first 10 digits and use Epoch Converter or `date -ur` to convert.
    c. com.apple.backupd.SnapshotCompletionDate
3. When did the backup complete (in UTC)?
    a. 1387823969876474 = 2013-12-23 18:39:29 Mon UTC
    b. Take the first 10 digits and use Epoch Converter or `date -ur` to convert.
    c. com.apple.backupd.SnapshotStartDate
4. What type of snapshot is it (Hourly, Daily, Monthly)?
    a. com.apple.backupd.SnapshotType is 3, which is an daily snapshot.
5. How many bytes were copied in this snapshot?
    a. com.apple.backupd.SnapshotTotalBytesCopied = 31,751,544 bytes

**6. 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.

```
$ sudo tmutil uniquesize *
```

1. Which snapshot is the largest?
    a. 2013-12-16-193058, 10.9M

- Use the `tmutil calculatedrift` command to view the differences between snapshots in this directory.
    i. Use the period "." Instead of "Dade's MacBook Air/" for the current directory.

```
$ tmutil calculatedrift .
```

2. Which snapshots had the most data added?
    a. 2013-12-15-102408 - 2013-12-16-193058 (3.1G)

- Use the `tmutil compare` command to compare two snapshots:
  i. `2013-12-22-164115`
  ii. `2013-12-23-133929`
- Output this to a file in your `FOR518` directory named, `tm_compare.txt`.
- Use the open command to view this file.

```
$ tmutil compare 2013-12-22-164115 2013-12-23-133929 >
~/FOR518/tm_compare.txt

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

| | |
|---|---|
| ! | **Metadata Changed** |
| + | **File Added** |
| - | **File Removed** |

3. What two files were created in zerocool's `Downloads` directory?
   a. 306px-Bitcoin_logo.svg.png
   b. 799px-Official_Litecoin_Logo_With_Text.png
4. What attributes were changed on the file `Macintosh HD/Users/zerocool/Library/Preferences/com.apple.Safari.plist`?
   a. Modified Time
   b. Size
5. How much data was removed in this snapshot?
   a. 122.4K

**7. Spotlight – Review the Spotlight Metadata**

- Use the `cd` command to explore the zerocool's Downloads directory.
- Use the `mdls` command to view the Spotlight metadata associated with the `Firefox 26.0.dmg` file.
  i. **Optional**: Use the `mdimport -A` command to show more information about the Spotlight metadata tags.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Downloads/

$ mdls Firefox\ 26.0.dmg
```

1. When was this file downloaded?
   a. kMDItemDownloadedDate = 2013-12-15 03:01:01 +0000
2. What domain was this file downloaded from?
   a. http://download-installer.cdn.mozilla.net
   b. kMDItemWhereFroms
3. What link did the user click to download this file?
   a. http://www.mozilla.org/en-US/firefox/new/
   b. kMDItemWhereFroms

8. **Spotlight – Review the Spotlight Directory**
   - Use the `cd` command to explore the `Macintosh HD` directory for the `dademurphy.E01` image.
   - Use the `ls -la` command to view the contents of this directory. Review the contents of this 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.

```
$ cd /Volumes/dademurphy_mounted/
$ ls -la
$ cd .Spotlight-V100
$ ls -la
$ open VolumeConfiguration.plist
```

1. What is the first part of the GUID for the Spotlight Store for the root volume?
   a. 0A00840B

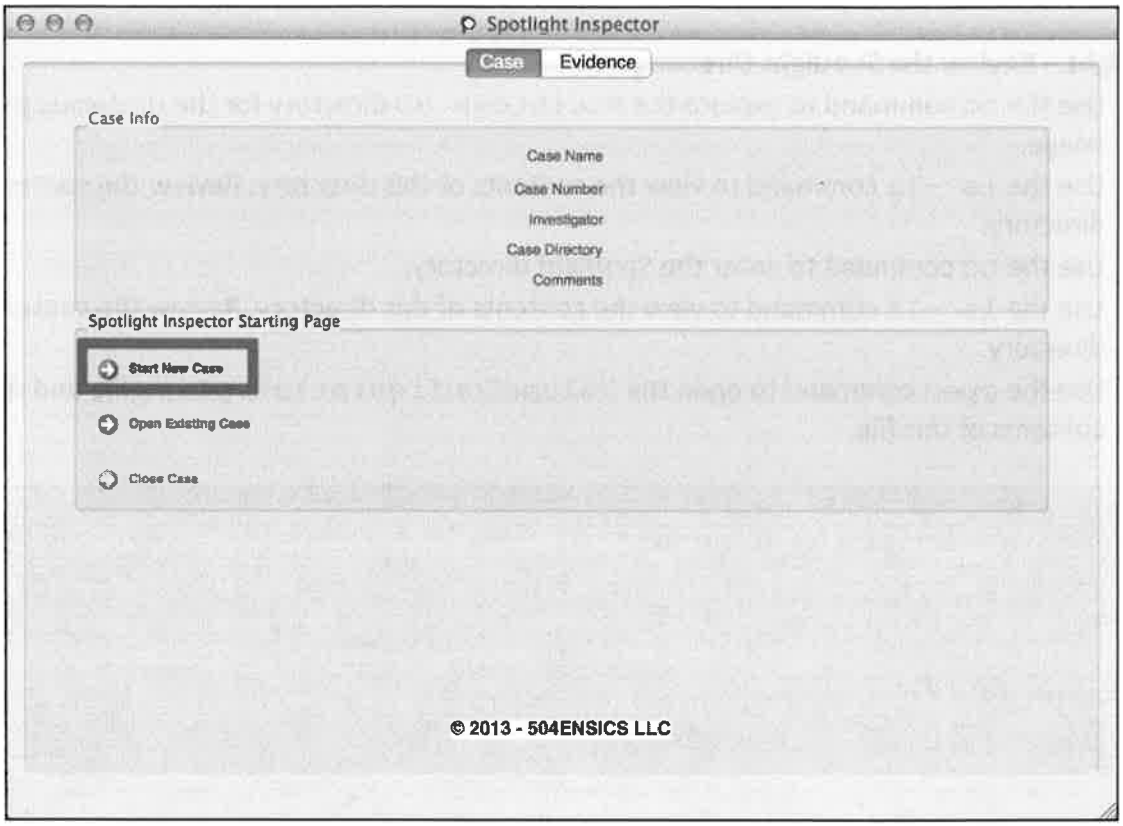
   - Use the `cd` command to enter the Spotlight store for the root volume.
   - Use the `ls -la` command to view the contents of this directory. Review the contents of this directory.

```
$ cd Store-V2/0A00840B-5B8C-44DC-9337-B6B58F5D5607
$ ls -la
```

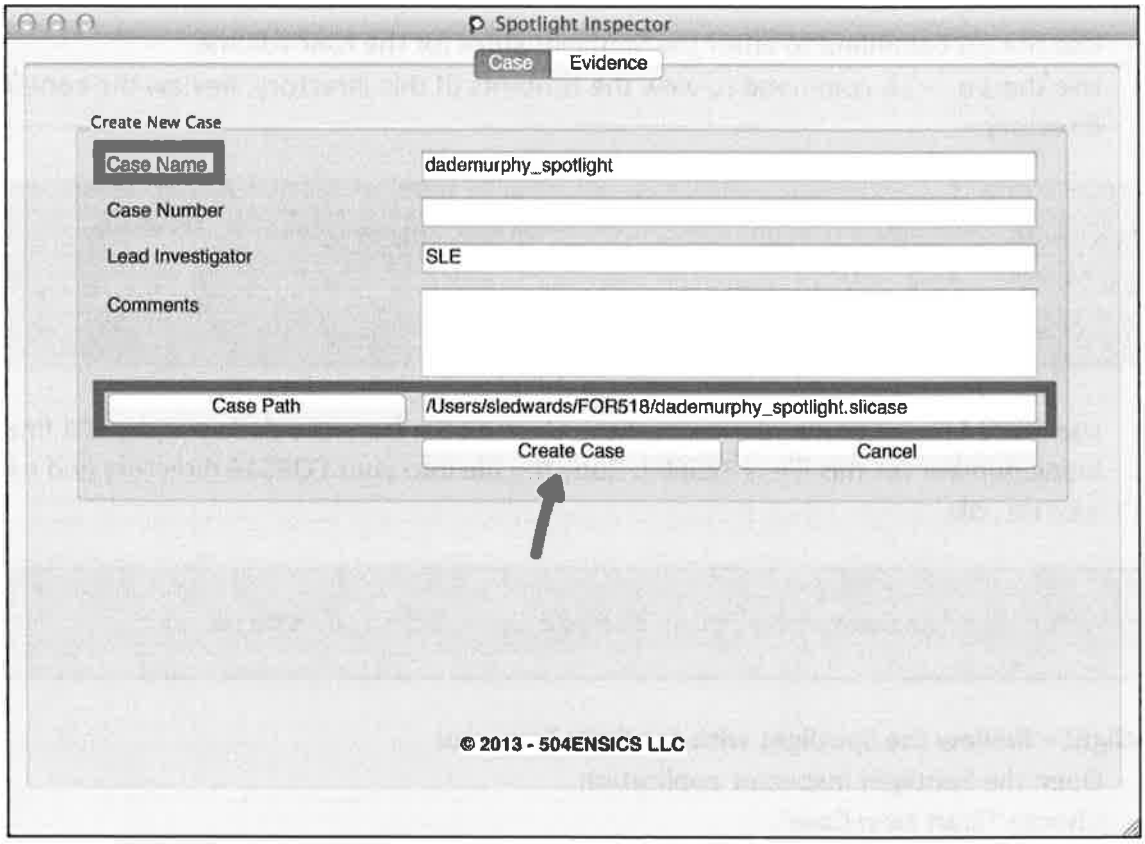
   - Use TSK's `icat` command to extract the store.db file from the dademurphy.E01 image. The inode number for this file is 325602. Copy the file into your FOR518 directory and name it `store.db`.

```
$ icat ~/FOR518/dademurphy.E01 325602 > ~/FOR518/store.db
```

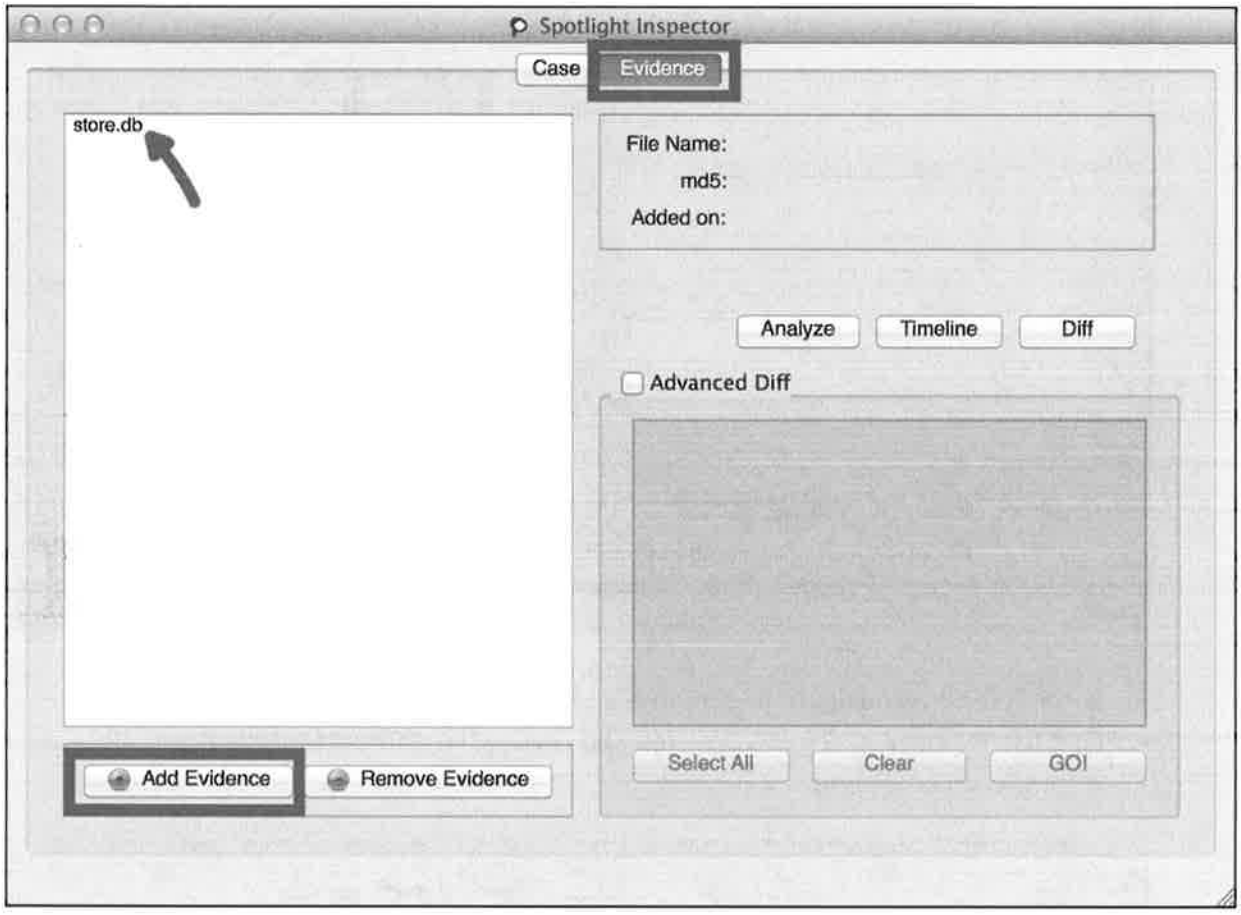
9. **Spotlight – Review the Spotlight with Spotlight Inspector**
   - Open the Spotlight Inspector application.
   - Choose "Start New Case".



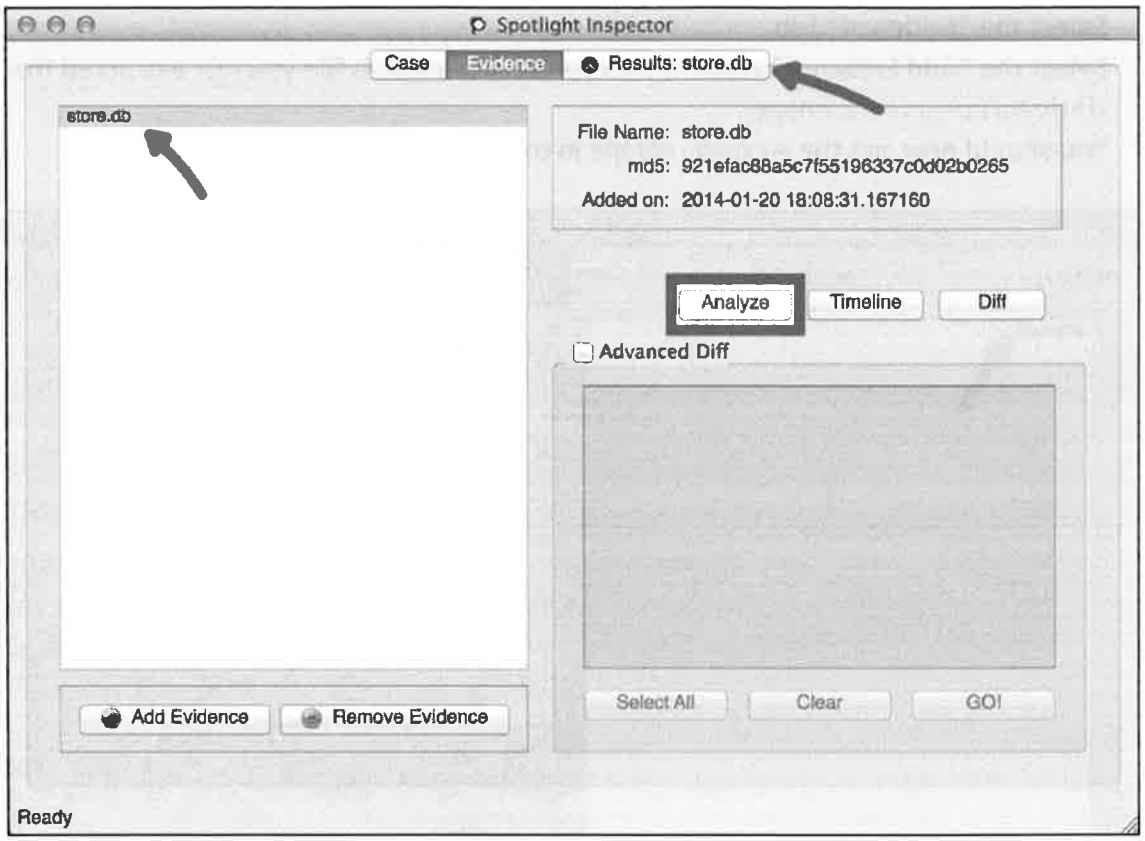

- Input a Case Name.
- Input a Case Path (You may choose to put it in your FOR518 directory).
- Select "Create Case".



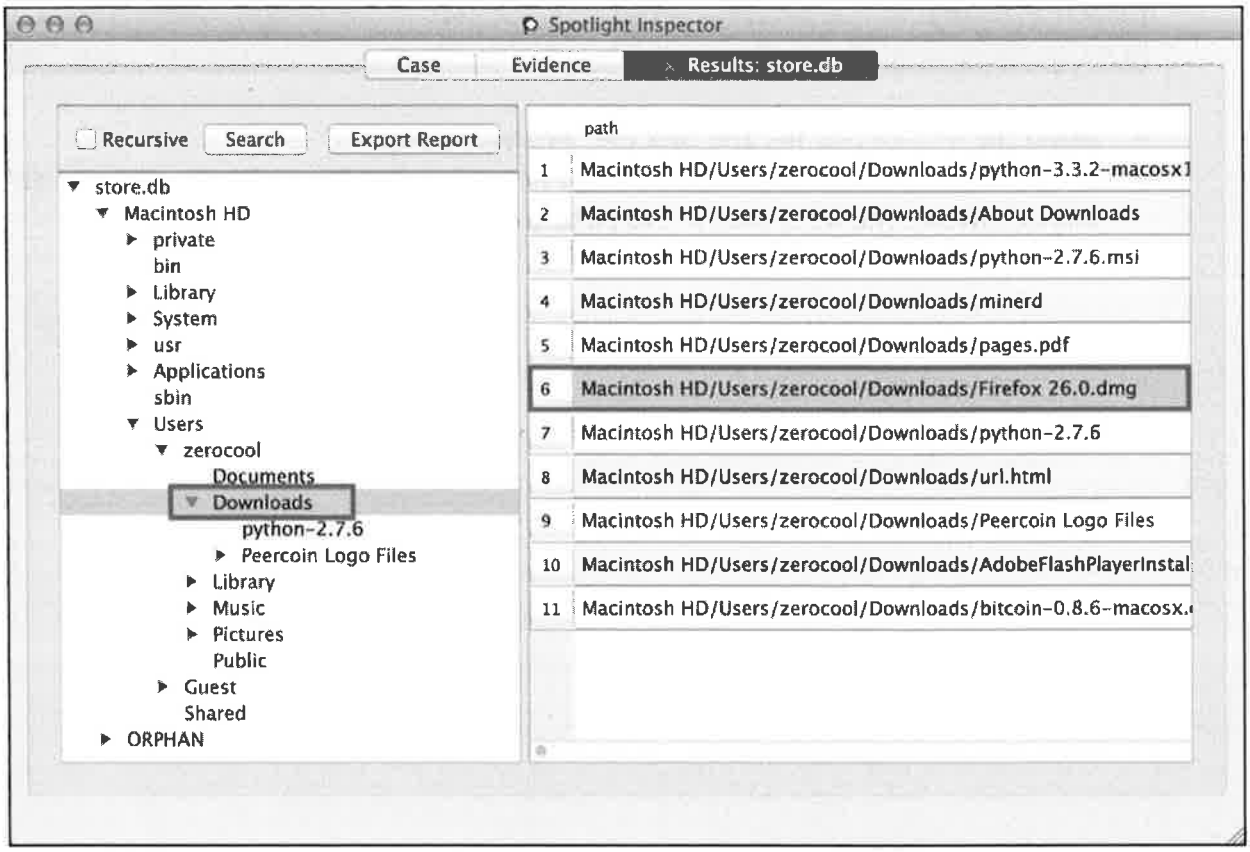
- Select the "Evidence" Tab.
- Select the "Add Evidence" Button; choose the `store.db` file you just extracted from the `dademurphy.E01` image.
- You should now see the `store.db` file in the window.




- Select the `store.db` file and click the "Analyze" Button.
- Once analysis is finished, you should see a new tab called "`Results: store.db`" - Select this tab. (Note: The MD5 hash in this picture is **not** accurate.)




- In the left pane, navigate to zerocool's `Downloads` directory.
- Find the `Firefox 26.0.dmg` file and review the contents of this tuple. Note the similarities from the `mdls` output.



- Find the photo in the file path:
  i. `Macintosh HD/Users/zerocool/Pictures/iPhoto Library/Masters/2013/12/16/20131216-192257/IMG_0003.JPG`
  ii. Review the contents of this tuple.

**Extra Credit**

- **Get familiar with the Spotlight Inspector application.**
  - **Use the Timeline function on the main Evidence Tab.**

*Exercise – Key Takeaways*

- **Understand how to analyze a Time Machine volume and its snapshots.**
- **Understand what file metadata the Spotlight index database contains.**

This page intentionally left blank.

# Exercise 4.2 – iCloud & Document Versions

## Objectives

- Get familiar with iCloud preference and data files.
- Get familiar with Document Version storage data and databases.

## Exercise Preparation

1. **Software Preparation** – The following tools will be used in this exercise:
   - 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/
   - A SQLite database browser
   - 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

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 the Dade Murphy forensic (`dademurphy.E01`) image**

   - Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

   - **Method 1:**
     - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
     - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
     - Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
       - `Mount Point` – Newly created specifically for this image.

- Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
- Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
    - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_image` is used in the example.
    - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_mounted` in used the example above.
    - Use `ewfmount` to mount the `dademurphy.e01` image to the `/Volumes/dadmurphy_image/` mount point.
    - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadeimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)
    - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
    - Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
        - `-j` – Ignore the journal
        - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

4. **Sanity Check**
   - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/

$ ls -l
```

*Exercise – Questions*

1. **iCloud - iCloud Accounts**
   - Use the `cd` command to explore the iCloud accounts directory.
   - Use the `ls -l` command to view the associated iCloud accounts.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Application\
Support/iCloud/Accounts/

$ ls -l
```

1. What is zerocool's iCloud Person ID?

_____

2. What two email accounts are associated with this account number?

_____

_____

**2. iCloud - Review the iCloud Preference Settings**

- Use the `cd` command to explore `zerocool`'s Preferences directory.
- Use the `open` command to review the contents of the `MobileMeAccounts.plist` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences
$ open MobileMeAccounts.plist
```

1. What email address is setup with this iCloud account?

_____

2. What is the numeric iCloud account ID?

_____

3. Is the Photo Stream iCloud service enabled?

_____

4. Are documents synced to iCloud?

_____

**3. iCloud - Review the iCloud Mobile Documents**

- Use the `cd` command to explore `zerocool`'s Mobile Documents directory.
- Use the `ls -la` command to view the contents of this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Mobile\
Documents/
$ ls -la
$ cd com~apple~TextEdit/Documents
$ cat Shopping\ List.txt
```

1. How many items are on `zerocool`'s shopping list?

_____

4. **iCloud - Review the iCloud Ubiquity Directory**
   - Use the `cd` command to explore `zerocool`'s Ubiquity directory.
   - Use the `ls -l` command to view the contents of this directory.
   - Use the `cd` command to explore the `peer-D623D7FE-793F-CE66-30A9-5CB6C01FBE61-v23` directory.
   - Use the `ls -l` command to view the contents of this directory. Note the contents of this directory.
   - Open the current directory using the `open` command.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Application\
Support/Ubiquity/

$ ls -l

$ cd peer-D623D7FE-793F-CE66-30A9-5CB6C01FBE61-v23/

$ ls -l

$ open .
```

   - Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
   - Go to the "`File`" menu and open the "`Open Database`".

   - This will open a selection window. Due to the Library being a hidden directory, we can select a file in the `peer-D623D7FE-793F-CE66-30A9-5CB6C01FBE61-v23` directory that you have just opened and drag it to this selection window to select the `item-info.db` file.
   - Review the `peer_names` table.

1. What iDevice does iCloud appear to sync with?

_____

   - Review the `item_table` table.
   - Look for the item named "The Cuckoo's Egg.pdf".

2. What is the size of this file?

_____

3. When was the file last modified?

_____

5. **iCloud - Review the iCloud Photo Stream**
   - Use the `cd` command to explore `zerocool`'s `iLifeAssetManagement` directory.

- Use the `ls -l` command to view the contents of this directory. Note the contents of this directory.
- Open the current directory using the `open` command.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Application\
Support/iLifeAssetManagement/

$ ls -l

$ open .
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".
- This will open a selection window. Due to the Library being a hidden directory, we can select a file in the `iLifeAssetManagement` directory that you have just opened and drag it to this selection window to select the `iLifeAssetManagement.db` file.
- Review the `AMAsset` table.

1. How large is IMG_0001.jpg?

_____

2. What is the height and width of this photo?

_____

3. When was this file downloaded?

_____

4. What is the iCloud Person ID?

_____

5. What are the first few characters of the photo UUID?

_____

6. What are the first few characters of the Device ID that this photo was taken with?

_____

- Use the `cd` command to explore `zerocool`'s `assets` directory.
- Use the `ls -l` command to view the contents of this directory. Note the sub and sub-shared directories.
- Use the `cd` command to explore `the sub/` directory. This directory contains the user's own photo stream.

- Use the open command to view the photo we saw the metadata for in the iLifeAssetManagement.db database file.
    i. In Preview, click Tools | Show Inspector (or Command+I) to view the EXIF data.
    ii. You may also use `exiftool`.

```
$ cd /Volumes/dademurphy_mounted/ Users/zerocool/Library/Application\
Support/iLifeAssetManagement/assets/

$ ls -l

$ cd sub/

$ open 017f5adf70881a3a98c973b6a84b3d3419743e815c/IMG_0001.JPG

$ cd ../sub-shared/
```

7. What kind of phone was this photo taken with?

_____

- Use the `cd` command to explore the `sub-shared/` directory.
- Review the photos from a shared photo stream.

```
$ cd ../sub-shared/
```

**6. Versions - Review the Document Versions Directory**

- Use the `cd` command to explore the system's Document Versions directory.
- Use the sudo –s command to get a privileged shell.
- Use the `ls -la` command to view the contents of this directory.
- Use the `cd` command to explore `the PerUID/501/3/com.apple.documentVersions` directory.
- Use the `ls -lTrt` command to view the contents of this directory. Note the contents of this directory.
- Use the open command to view all these files in TextEdit.app. (Note: 10.10 users may not be able to open these files due to permissions issues, please skip the following questions.)

```
$ cd /Volumes/dademurphy_mounted/.DocumentRevisions-V100
$ sudo -s
$ ls -la
$ cd PerUID/501/3/com.apple.documentVersions/
$ ls -lTrt
$ open *
```

1. What item was added to the list from Dec 14 22:07 to Dec 14 22:08 (EST/EDT)?

_____

2. What happened to this file around Dec 14 22:11 (EST/EDT)?

_____

3. What is the original name of this file?

_____

**7. Versions - Review the Document Versions Database**

- Use the `cd` command to explore the systems' Document Versions database directory.
- Use the `ls -l` command to view the contents of this directory.
- Use the `open` to this directory.

```
$ cd /Volumes/dademurphy_mounted/.DocumentRevisions-V100/db-V1/
$ ls -l
$ open .
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".
- This will open a selection window. Due to the Library being a hidden directory, we can select a file in the `db-V1` directory that you have just opened and drag it to this selection window to select the `db.sqlite` file.
    i. NOTE: IF you get an error, copy out the `db.sqlite`, and `db.sqlite-wal` database files to your FOR518 directory, and then open the file. (`cp db.sqlite db.sqlite-wal ~/FOR518/`)
- Review the `generations` table.

- Find the "`generations_path`" that looks familiar – you should see the three files we just saw.
- Review the `files` table.
- Find our "Shopping List" file; review the information in this tuple.

**8. Versions - Review the Versions Chunk Store Database**

- Use the `cd` command to explore the systems Versions Chunk Store database directory.
- Use the `ls -l` command to view the contents of this directory.
- Use the `open` to this directory.

```
$ cd /Volumes/dademurphy_mounted/.DocumentRevisions-V100/.cs/

$ ls -l

$ open .
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".
- This will open a selection window. Due to the Library being a hidden directory, we can select a file in the `db-V1` directory that you have just opened and drag it to this selection window to select the `ChunkStoreDatabase` file.
    i. NOTE: If you get an error, copy out the `ChunkStoreDatabase` database file to your FOR518 directory, and then open the file.
    ii. (`cp ChunkStoreDatabase ~/FOR518/`)
- Review the `CSChunkTable` table.
- Keeping the `ChunkStoreDatabase` open for reference, change directories to the Chunk Storage file – "3".
- Using the open command, open this file in your favorite hex editor.

```
$ cd /Volumes/dademurphy_mounted/.DocumentRevisions-
V100/.cs/ChunkStorage/0/0/0/

$ open -a 0xED 3
```

| Chuck Storage Record Format | |
|---|---|
| **4 bytes** | Size of chunk record |
| **21 bytes** | Chunk ID |
| **Remaining** | Chunk Contents |

1. At offset 0, calculate the size of the first chuck record. What is the size of this record?

---

2. At offset 4, what is the chunk ID?

---

3. At offset 25, review the data contained in the next 510 bytes.

*Exercise – Step-By-Step*

1. **iCloud - iCloud Accounts**
   - Use the `cd` command to explore the iCloud accounts directory.
   - Use the `ls -l` command to view the associated iCloud accounts.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Application\
Support/iCloud/Accounts/

$ ls -l
```

1. What is zerocool's iCloud Person ID?
   a. 1488584776
2. What two email accounts are associated with this account number?
   a. z3r0cool95@gmail.com
   b. z3r0cool95@icloud.com

2. **iCloud - Review the iCloud Preference Settings**
   - Use the `cd` command to explore `zerocool`'s Preferences directory.
   - Use the `open` command to review the contents of the `MobileMeAccounts.plist` file.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Preferences

$ open MobileMeAccounts.plist
```

1. What email address is setup with this iCloud account?
   a. z3r0cool95@gmail.com
2. What is the numeric iCloud account ID?
   a. 1488584776
3. Is the Photo Stream iCloud service enabled?
   a. Yes – the "`Enabled`" key under the `PHOTO_STREAM` item is set to "YES".
4. Are documents synced to iCloud?
   a. Yes – the "`Enabled`" key under the `MOBILE DOCUMENTS` item is set to "YES".

3. **iCloud - Review the iCloud Mobile Documents**
   - Use the `cd` command to explore `zerocool`'s Mobile Documents directory.
   - Use the `ls -la` command to view the contents of this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Mobile\
Documents/

$ ls -la

$ cd com~apple~TextEdit/Documents

$ cat Shopping\ List.txt
```

1. How many items are on `zerocool`'s shopping list?
    a. Nine items

**4. iCloud - Review the iCloud Ubiquity Directory**

- Use the `cd` command to explore `zerocool`'s Ubiquity directory.
- Use the `ls -l` command to view the contents of this directory.
- Use the `cd` command to explore the `peer-D623D7FE-793F-CE66-30A9-5CB6C01FBE61-v23` directory.
- Use the `ls -l` command to view the contents of this directory. Note the contents of this directory.
- Open the current directory using the `open` command.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Application\
Support/Ubiquity/

$ ls -l

$ cd peer-D623D7FE-793F-CE66-30A9-5CB6C01FBE61-v23/

$ ls -l

$ open .
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".

- This will open a selection window. Due to the Library being a hidden directory, we can select a file in the `peer-D623D7FE-793F-CE66-30A9-5CB6C01FBE61-v23` directory that you have just opened and drag it to this selection window to select the `item-info.db` file.
- Review the `peer_names` table.

1. What iDevice does iCloud appear to sync with?
    a. Dade's iPhone

- Review the `item_table` table.
- Look for the item named "The Cuckoo's Egg.pdf".

2. What is the size of this file?
    a. 925,853 bytes
3. When was the file last modified?
    a. 1387077280 = 2013-12-15 03:14:40 Sun UTC

**5. iCloud - Review the iCloud Photo Stream**

- Use the `cd` command to explore `zerocool`'s `iLifeAssetManagement` directory.
- Use the `ls -l` command to view the contents of this directory. Note the contents of this directory.
- Open the current directory using the `open` command.

```
$ cd /Volumes/dademurphy_mounted/Users/zerocool/Library/Application\
Support/ iLifeAssetManagement/

$ ls -l

$ open .
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".
- This will open a selection window. Due to the Library being a hidden directory, we can select a file in the `iLifeAssetManagement` directory that you have just opened and drag it to this selection window to select the `iLifeAssetManagement.db` file.
- Review the `AMAsset` table.

1. How large is IMG_0001.jpg?
    a. 2881470 bytes
2. What is the height and width of this photo?
    a. 3264,2448
3. When was this file downloaded?
    a. 408849634 = 2013-12-16 01:20:34 Mon UTC
4. What is the iCloud Person ID?
    a. 1488584776
5. What are the first few characters of the photo UUID?
    a. 017f5adf70881a3a98c973b6a84b3d3419743e815c
6. What are the first few characters of the Device ID that this photo was taken with?
    a. aeef53ba55a329288af8d5513a077c913a12a1d0

- Use the `cd` command to explore `zerocool`'s `assets` directory.

- Use the `ls -l` command to view the contents of this directory. Note the sub and sub-shared directories.
- Use the `cd` command to explore `the sub/` directory. This directory contains the user's own photo stream.
- Use the open command to view the photo we saw the metadata for in the iLifeAssetManagement.db database file.
    i. In Preview, click Tools | Show Inspector (or Command+I) to view the EXIF data.
    ii. You may also use `exiftool`.

```
$ cd /Volumes/dademurphy_mounted/ Users/zerocool/Library/Application\
Support/ iLifeAssetManagement/assets/

$ ls -l

$ cd sub/

$ open 017f5adf70881a3a98c973b6a84b3d3419743e815c/IMG_0001.JPG

$ cd ../sub-shared/
```

7. What kind of phone was this photo taken with?
    a. iPhone 4S

- Use the `cd` command to explore the `sub-shared/` directory.
- Review the photos from a shared photo stream.

```
$ cd ../sub-shared/
```

**6. Versions - Review the Document Versions Directory**

- Use the `cd` command to explore the system's Document Versions directory.
- Use the sudo –s command to get a privileged shell.
- Use the `ls -la` command to view the contents of this directory.
- Use the `cd` command to explore `the PerUID/501/3/com.apple.documentVersions` directory.
- Use the `ls -lTrt` command to view the contents of this directory. Note the contents of this directory.
- Use the open command to view all these files in TextEdit.app. (Note: 10.10 users may not be able to open these files due to permissions issues, please skip the following questions.)

```
$ cd /Volumes/dademurphy_mounted/.DocumentRevisions-V100

$ sudo -s

$ ls -la

$ cd PerUID/501/3/com.apple.documentVersions/

$ ls -lTrt

$ open *
```

1. What item was added to the list from Dec 14 22:07 to Dec 14 22:08 (EST/EDT)?
    a. Ultra Duster 10oz.
2. What happened to this file around Dec 14 22:11 (EST/EDT)?
    a. It was changed to a plaintext file from an RTF file.
3. What is the original name of this file?
    a. "Shopping List"
        i. View the extended attributes using `xattr -xl` command, view the com.apple.genstore.origdisplayname attribute.

**7. Versions - Review the Document Versions Database**

- Use the `cd` command to explore the systems' Document Versions database directory.
- Use the `ls -l` command to view the contents of this directory.
- Use the `open` to this directory.

```
$ cd /Volumes/dademurphy_mounted/.DocumentRevisions-V100/db-V1/

$ ls -l

$ open .
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".
- This will open a selection window. Due to the Library being a hidden directory, we can select a file in the `db-V1` directory that you have just opened and drag it to this selection window to select the `db.sqlite` file.
    i. NOTE: IF you get an error, copy out the `db.sqlite`, and `db.sqlite-wal` database files to your FOR518 directory, and then open the file. (`cp db.sqlite db.sqlite-wal ~/FOR518/`)
- Review the `generations` table.
- Find the "`generations_path`" that looks familiar – you should see the three files we just saw.

- Review the `files` table.
- Find our "Shopping List" file; review the information in this tuple.

**8. Versions - Review the Versions Chunk Store Database**

- Use the `cd` command to explore the systems' Versions Chunk Store database directory.
- Use the `ls -l` command to view the contents of this directory.
- Use the `open` to this directory.

```
$ cd /Volumes/dademurphy_mounted/.DocumentRevisions-V100/.cs/

$ ls -l

$ open .
```

- Open the SQLite Database Browser file located in `/Applications/sqlitebrowser.app`.
- Go to the "`File`" menu and open the "`Open Database`".
- This will open a selection window. Due to the Library being a hidden directory, we can select a file in the `db-V1` directory that you have just opened and drag it to this selection window to select the `ChunkStoreDatabase` file.
    i. NOTE: If you get an error, copy out the `ChunkStoreDatabase` database file to your FOR518 directory, and then open the file.
    ii. (`cp ChunkStoreDatabase ~/FOR518/`)
- Review the `CSChunkTable` table.
- Keeping the `ChunkStoreDatabase` open for reference, change directories to the Chunk Storage file – "3".
- Using the open command, open this file in your favorite hex editor.

```
$ cd /Volumes/dademurphy_mounted/.DocumentRevisions-
V100/.cs/ChunkStorage/0/0/0/

$ open -a 0xED 3
```

| Chuck Storage Record Format | |
|---|---|
| **4 bytes** | Size of chunk record |
| **21 bytes** | Chunk ID |
| **Remaining** | Chunk Contents |

1. At offset 0, calculate the size of the first chuck record. What is the size of this record?
    a. 0x00000217 (535 bytes)
2. At offset 4, what is the chunk ID?
    a. 0x012D1449115117051F0A1CAE83493F58C93A133BFF7B
3. At offset 25, review the data contained in the next 510 bytes.

## *Exercise – Key Takeaways*

- **Understand how iCloud documents, photos and preferences are stored.**
- **Understand how Chunk Storage is implemented in Document Versions.**

This page intentionally left blank.

# Exercise 4.3 – Memory Analysis

## *Objectives*

- Understand the various tools used to conduct memory analysis on Mac memory.
- Understand the capability of these tools and what you can expect to extract from Mac memory.

## *Exercise Preparation*

1. **Software Preparation** – The following tools will be used in this exercise:
   - 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. You will be using the `Volatility 2.4` directory from the Exercise Files directory on your FOR518 USB drive. This directory already has the Mac overlays required for this class in the `/volatility/plugins/overlays/mac/` directory.
        1. Specifically `MountainLion_10.8.5_AMD.zip` and `MacLion_10_7_5_Intelx86`.
2. **Memory Image** – Copy the `dademurphy_memory.001` 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 exercise.
3. **Exercise File Preparation** – Locate the `Exercise Files/Exercise 4.4 – Memory Analysis` directory on your FOR518 USB drive.
4. **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.

*Exercise – Questions*

1. **Volatility - Documentation**
   - In the Terminal, use the `cd` command to enter the `volatility-2.4` directory where you have unpacked your tools from the FOR518 USB drive.
   - Run the `vol.py` with the `--info` parameter to view the tool documentation.

```
$ python vol.py --info | less
```

1. Review the "`Profiles`" Section. What two profiles will we likely be using in this class? These profiles were pre-loaded in the overlays directory for you. You may add more profiles when needed, but for processing speed considerations, only the profiles needed for this class were placed in this directory (`/volatility/plugins/overlays/mac`).

_____

_____

   - Review the "Plugins" Section. Note the plugins named with the "`mac_*`". We will be using these in this class.

```
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
```

- Run the `vol.py` with the `-h` parameter to view the tool usage documentation.

```
$ python 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
```

**2. Volatility – 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 exercise.)**

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_version
```

1. What Kernel version does this system use?

_____

- Run the `vol.py` with the `mac_machine_info` parameter to view the system information.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_machine_info
```

2. How much RAM does this system have?

_____

3. How many physical CPUs does this system have?

_____

- Run the `vol.py` with the `mac_dmesg` parameter to view the kernel message buffer.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_dmesg | less
```

- Review the output of this command.

- Run the `vol.py` with the `mac_mount` parameter to view mounted volumes on this system.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_mount
```

4. How many volumes are mounted on `/Volumes`?

_____

5. What format is `/dev/disk2s1`?

_____

**3. Volatility – Network Information**

- Run the `vol.py` with the `mac_ifconfig` parameter to view the system network configuration.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_ifconfig
```

1. What IP does this system have?

_____

2. What is the MAC address of the primary network card?

---

- Run the `vol.py` with the `mac_arp` and `mac_route` parameters to view ARP and routing tables. Review the output of these commands.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_arp

$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_route
```

- Run the `vol.py` with the `mac_netstat` parameter to view the system network connections and UNIX sockets.
    i. To filter only TCP and UDP connections, pipe the output to the `egrep -w 'TCP|UDP'` command.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_netstat
```

3. What two remote IP addresses does the Mail program have connections to?

---

---

4. Perform a `whois` on these IP addresses. Whom do they belong to?

---

---

**4. Volatility - Processes**

- Run the `vol.py` with the `mac_pslist` parameter to view system processes by walking the process list.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_pslist
```

- Note the start time for PIDs 0 and 1.

1. How many processes are shown in this output? (Pipe the output to the "`grep -c 0x`" command to count the lines containing each process)

_____

- Run the `vol.py` with the `mac_pgrp_hash_table` parameter to view the system processes by enumerating the process group hash table.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_pgrp_hash_table
```

2. How many processes are shown in this output? (Pipe the output to the "`grep -c 0x`" command to count the lines containing each process)

_____

- Run the `vol.py` with the `mac_pstree` parameter to view system processes in a tree formation.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_pstree
```

3. What processes were performed using the `sudo` command?

_____

- Run the `vol.py` with the `mac_lsof` parameter to view the open file handles for each process.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_lsof
```

4. What process has the `db.sqlite` file open?

_____

**5. Volatility – Kernel Extensions**

- Run the `vol.py` with the `mac_lsmod` parameter to view kernel extensions.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_lsmod
```

1. How many kernel extensions were found? (Pipe the output to the "`grep -c 0x`" command to count the lines containing each process)

_____

**1. Volatility - Documentation**

- In the Terminal, use the `cd` command to enter the `volatility-2.4` directory where you have unpacked your tools from the FOR518 USB drive.
- Run the `vol.py` with the `--info` parameter to view the tool documentation.

```
$ python vol.py --info | less
```

1. Review the "`Profiles`" Section. What two profiles will we likely be using in this class? These profiles were pre-loaded in the overlays directory for you. You may add more profiles when needed, but for processing speed considerations, only the profiles needed for this class were placed in this directory (`/volatility/plugins/overlays/mac`).
   a. MacLion_10_7_5_Intelx86
   b. MacMountainLion_10_8_5_AMDx64

- Review the "Plugins" Section. Note the plugins named with the "`mac_*`". We will be using these in this class.

```
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
```

- Run the `vol.py` with the `-h` parameter to view the tool usage documentation.

```
$ python 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
```

**2. Volatility – 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 exercise.)**

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_version
```

1. What Kernel version does this system use?
   a. 12.5.0

- Run the `vol.py` with the `mac_machine_info` parameter to view the system information.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_machine_info
```

2. How much RAM does this system have?
   a. 8589934592 (bytes) or 8GB
3. How many physical CPUs does this system have?
   a. Two

- Run the `vol.py` with the `mac_dmesg` parameter to view the kernel message buffer.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_dmesg | less
```

- Review the output of this command.

- Run the `vol.py` with the `mac_mount` parameter to view mounted volumes on this system.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_mount
```

4. How many volumes are mounted on `/Volumes`?
   a. Seven
5. What format is `/dev/disk2s1`?
   a. ExFAT

**3. Volatility – Network Information**

- Run the `vol.py` with the `mac_ifconfig` parameter to view the system network configuration.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_ifconfig
```

1. What IP does this system have?
   a. 192.168.2.102
2. What is the MAC address of the primary network card?
   a. 7c:d1:c3:df:64:67 (en0)

- Run the `vol.py` with the `mac_arp` and `mac_route` parameters to view ARP and routing tables. Review the output of these commands.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_arp

$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_route
```

- Run the `vol.py` with the `mac_netstat` parameter to view the system network connections and UNIX sockets.
    i. To filter only TCP and UDP connections, pipe the output to the `egrep -w 'TCP|UDP'` command.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_netstat
```

3. What two remote IP addresses does the Mail program have connections to?
    a. 17.151.237.9
    b. 173.194.68.108
4. Perform a `whois` on these IP addresses. Whom do they belong to?
    a. 17.151.237.9 (Apple)
    b. 173.194.68.108 (Google)

**4. Volatility - Processes**

- Run the `vol.py` with the `mac_pslist` parameter to view system processes by walking the process list.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_pslist
```

- Note the start time for PIDs 0 and 1.

1. How many processes are shown in this output? (Pipe the output to the "`grep -c 0x`" command to count the lines containing each process)
    a. 112

- Run the `vol.py` with the `mac_pgrp_hash_table` parameter to view the system processes by enumerating the process group hash table.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_pgrp_hash_table
```

2. How many processes are shown in this output? (Pipe the output to the "`grep -c 0x`" command to count the lines containing each process)
   a. 114

- Run the `vol.py` with the `mac_pstree` parameter to view system processes in a tree formation.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_pstree
```

3. What processes were performed using the `sudo` command?
   b. MacQuisition -> bash -> dc3dd

- Run the `vol.py` with the `mac_lsof` parameter to view the open file handles for each process.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_lsof
```

4. What process has the `db.sqlite` file open?
   c. 51 = revisiond (Document Revision Daemon)

**5. Volatility – Kernel Extensions**

- Run the `vol.py` with the `mac_lsmod` parameter to view kernel extensions.

```
$ python vol.py -f dademurphy_memory.001
--profile=MacMountainLion_10_8_5_AMDx64 mac_lsmod
```

1. How many kernel extensions were found? (Pipe the output to the "`grep -c 0x`" command to count the lines containing each process)
   a. 104

***Exercise – Key Takeaways***

- **Get comfortable with Volatility and Mac Memoryze command-line utilities for Mac memory analysis.**
- **Understand what each tool is capable of and the differences of output between each. Multiple tools may be needed to accomplish analysis.**

# Exercise 4.4 – Password Cracking & Encrypted Containers

## *Objectives*

- Use Dave Grohl and John the Ripper to crack passwords for a user account, a keychain, and an encrypted DMG file.

## *Exercise Preparation*

1. **Software Preparation** – The following tools will be used in this exercise:
   - 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/
   - Dave Grohl
     i. Use the files in the `Exercise 4.2 Files` directory on your USB drive.
     ii. This file can also be downloaded from: http://davegrohl.org/downloads.html

2. **Exercise File Preparation** – Locate the files located in the `Exercise Files/Exercise 4.2 - Password Cracking & Encrypted Containers` directory on your FOR518 USB drive.
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. **Compile John the Ripper**
   - Use the files in the `Exercise 4.2 Files` directory on your USB drive.
   - This file can also be downloaded from: https://github.com/magnumripper/JohnTheRipper

   1. Locate and open the `Terminal.app` from `/Applications/Utilities/`
   2. Use the `unzip` command to unzip the `JohnTheRipper-bleeding-jumbo.zip` archive.
   3. Use the `cd` command to enter the `src` directory of the `JohnTheRipper-bleeding-jumbo/` directory.
   4. Use the `make` command below to compile JTR. This will take a minute or two.
   5. Use the following commands to test your JTR installation.
      a. Use the `cd` command to go back up to the `run` directory.
      b. Run the command, `./john -test`
         i. You should see "Benchmarking" tests being run. Only let this run for about 10 seconds.
      c. Use Control+C to exit the test.

```
$ unzip JohnTheRipper-bleeding-jumbo.zip

$ cd JohnTheRipper-bleeding-jumbo/src/

$ make clean macosx-x86-64
```

```
$ cd ../run/

$ ./john –test
```

*Exercise – Questions*

1. **User Password Cracking – Dave Grohl**
   - A sample user property list is located on your FOR518 USB drive. This file has been extracted from a OS X Lion (10.7) system from the `/Macintosh HD/private/var/db/dslocal/nodes/Default/users/` directory. The user account property list is named "`emmanuelgoldstein.plist`".
   - Use the `cd` command to enter the `DaveGrohl` directory.
   - Run the `dave` program by itself and review the documentation.
   - Use the `dave` program to crack the password for this user account.
     i. The `-d` parameter allows the program to intake a dictionary file for use in cracking the password. A dictionary was created using the `strings` program against a memory image of the system. This dictionary is located in the Dave Grohl "`wordlists`" directory as `emmanualgoldstein_memory_strings.txt`.
        1. This file is not a requirement, however it may make password cracking faster.
     ii. The `-p` parameter allows the program to intake a property list (which contains the password hash) to crack the password.

```
$ cd DaveGrohl/

$ ./dave

$ ./dave -d -p ../FILES\ TO\ CRACK/emmanuelgoldstein.plist
```

1. What kind of hash is being cracked here?

   _____

2. What is the password for the `emmanuelgoldstein` user account?

   _____

2. **User Password Cracking – John the Ripper (JTR)**
   - Using the same property list as the previous example, we'll crack it again using another program – John the Ripper.
   - Use the `cd` command to enter the `JohnTheRipper-bleeding-jumbo/run/` directory.
   - Using the `perl` utility, and JTR's `lion2john.pl` script to extract the hash from the user's property list file in the input file format that JTR requires. Save this file to your FOR518 directory using the name `jtr_EM_hash.txt`.
   - Run the `john` program by itself and review the documentation.
   - Use the `john` program to crack the password for this user account.

i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password. We'll use the same dictionary located in the Dave Grohl "`wordlists`" directory as `emmanuelgoldstein_memory_strings.txt`.
   1. This file is not a requirement; however, it may make password cracking faster.

```
$ cd ../JohnTheRipper-bleeding-jumbo/run

$ perl lion2john.pl ../../FILES\ TO\ CRACK/emmanuelgoldstein.plist >
~/FOR518/jtr_EM_hash.txt

$ ./john

$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt
~/FOR518/jtr_EM_hash.txt
```

```
nibble:run sledwards$ perl lion2john.pl ../../FILES\ TO\ CRACK/emmanuelgoldstein.plist > ~/FOR518/jtr_EM_hash.txt
nibble:run sledwards$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt ~/FOR518/jtr_EM
_hash.txt
Loaded 1 password hash (Mac OS X 10.7+ salted SHA-512 [64/64 CommonCrypto])
hacker1984       (emmanuelgoldstein)
guesses: 1  time: 0:00:00:05 DONE (Wed Jan 22 20:26:05 2014)  c/s: 1264K  trying: AVSH - LWNSWhite
Use the "--show" option to display all of the cracked passwords reliably
```

- **Note**: If you would like to re-run this hash crack again, remove the `john.pot` file from the `run/` directory.

**3. User Keychain Cracking – John the Ripper (JTR)**

- Next, we'll crack (or at least start) Emmanuel's login keychain using John the Ripper.
- Use JTR's `keychain2john` script to extract the hash from the user's login keychain file in the input file format that JTR requires. Save this file to your FOR518 directory using the name `jtr_EM_keychain.txt`.
- Use the `john` program to crack the password for this user account.
   i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password. We'll use the same dictionary located in the Dave Grohl "`wordlists`" directory as `emmanuelgoldstein_memory_strings.txt`.
      1. This file is not a requirement, however it may make password cracking faster.

```
$ ./keychain2john ../../FILES\ TO\ CRACK/login.keychain >
~/FOR518/jtr_EM_keychain.txt

$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt
~/FOR518/jtr_EM_keychain.txt
```

1. What kind of hash is being cracked here?

_____

- **This will take a while to run, even with the dictionary file.** This is due to the PBKDF2 encryption.
- Press the Spacebar to show the status of the password cracking process.
- You may use Control+C to kill the process. **You do not need to continue crack this password.**
    i. This process took about an hour and a half on a Macbook Pro laptop.
- **Note**: If you would like to re-run this hash crack again, remove the `john.pot` file from the `run/` directory.

- You can now try to open the login.keychain file from the "FILES TO CRACK" directory using Keychain Access.app located in Applications/Utilities/ directory. Double-click the keychain file or using File | Add Keychain.
- Unlock the keychain using the password that you cracked (Hint: **hacker1984**) by selecting the locked login keychain and clicking "Click to unlock the login keychain." lock above.

2. What email address was used to login to accounts.google.com?

_____

3. What is the password for this account?

_____

4. **Encrypted DMG Cracking – John the Ripper (JTR)**
    - Next, we'll crack (or at least start) an encrypted DMG file.
    - Use JTR's `dmg2john` script to extract the hash from the encryption key from the file in the input file format that JTR requires. Save this file to your FOR518 directory using the name `jtr_EM_dmg.txt`.
    - Use the `john` program to crack the password for this user account.
        i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password. We'll use the same dictionary located in the Dave Grohl "`wordlists`" directory as `emmanuelgoldstein_memory_strings.txt`.
            1. This file is not a requirement, however it may make password cracking faster.

```
$ ./dmg2john ../../FILES\ TO\ CRACK/protected.dmg >
~/FOR518/jtr_EM_dmg.txt

$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt
~/FOR518/jtr_EM_dmg.txt
```

```
nibble:run sledwards$ ./dmg2john ../../FILES\ TO\ CRACK/protected.dmg > ~/FOR518/jtr_EM_dmg.txt
protected.dmg (DMG v2) successfully parsed, iterations count 196078
nibble:run sledwards$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt  ~/FOR518/jtr_E
M_dmg.txt
Loaded 1 password hash (Apple DMG PBKDF2-HMAC-SHA-1 3DES / AES [32/64])
guesses: 0  time: 1:14:36:40 15.02% (ETA: Sat Feb  1 11:25:06 2014)  c/s: 7.81  trying: ion" id="prevent-prompting-for
-location" onclick="PrivacyView.al
Session aborted
```

- This will take a while to run, even with the dictionary file. This is due to the PBKDF2 encryption. This screenshot example estimates it would have taken approximately 12-13 days to crack using the dictionary file provided.
- Press the Spacebar to show the status of the password cracking process.
- You may use Control+C to kill the process. This cracking process will take much longer than the keychain. This is due to the "iterations" count. To learn more about why this takes much longer from a humorous article - read is this one: blog.whitehatsec.com/cracking-aes-256-dmgs-and-epic-self-pwnage/.
- **Note**: If you would like to re-run this hash crack again, you may need to remove the `john.pot` file from the `run/` directory.

1. **User Password Cracking – Dave Grohl**
   - A sample user property list is located on your FOR518 USB drive. This file has been extracted from a OS X Lion (10.7) system from the `/Macintosh HD/private/var/db/dslocal/nodes/Default/users/` directory. The user account property list is named "`emmanuelgoldstein.plist`".
   - Use the `cd` command to enter the `DaveGrohl` directory.
   - Run the `dave` program by itself and review the documentation.
   - Use the `dave` program to crack the password for this user account.
     i. The `-d` parameter allows the program to intake a dictionary file for use in cracking the password. A dictionary was created using the `strings` program against a memory image of the system. This dictionary is located in the Dave Grohl "`wordlists`" directory as `emmanualgoldstein_memory_strings.txt`.
        1. This file is not a requirement, however it may make password cracking faster.
     ii. The `-p` parameter allows the program to intake a property list (which contains the password hash) to crack the password.

```
$ cd DaveGrohl/

$ ./dave

$ ./dave -d -p ../FILES\ TO\ CRACK/emmanuelgoldstein.plist
```

1. What kind of hash is being cracked here?
   a. Salted SHA512, default user account hash for 10.7 systems.
2. What is the password for the `emmanuelgoldstein` user account?
   a. hacker1984

```
nibble:DaveGrohl sledwards$ ./dave -d -p ../FILES\ TO\ CRACK/emmanuelgoldstein.plist
-- Loaded Salted SHA512 hash...
-- Starting attack

-- Found password : 'hacker1984'
-- (dictionary attack)

Finished in 7.672 seconds / 9,394,258 guesses...
1,224,482 guesses per second.
```

2. **User Password Cracking – John the Ripper (JTR)**
   - Using the same property list as the previous example, we'll crack it again using another program – John the Ripper.
   - Use the `cd` command to enter the `JohnTheRipper-bleeding-jumbo/run/` directory.
   - Using the `perl` utility, and JTR's `lion2john.pl` script to extract the hash from the user's property list file in the input file format that JTR requires. Save this file to your FOR518 directory using the name `jtr_EM_hash.txt`.

- Run the `john` program by itself and review the documentation.
- Use the `john` program to crack the password for this user account.
    i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password. We'll use the same dictionary located in the Dave Grohl "`wordlists`" directory as `emmanuelgoldstein_memory_strings.txt`.
        1. This file is not a requirement; however, it may make password cracking faster.

```
$ cd ../JohnTheRipper-bleeding-jumbo/run

$ perl lion2john.pl ../../FILES\ TO\ CRACK/emmanuelgoldstein.plist >
~/FOR518/jtr_EM_hash.txt

$ ./john

$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt
~/FOR518/jtr_EM_hash.txt
```

```
nibble:run sledwards$ perl lion2john.pl ../../FILES\ TO\ CRACK/emmanuelgoldstein.plist > ~/FOR518/jtr_EM_hash.txt
nibble:run sledwards$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt ~/FOR518/jtr_EM
_hash.txt
Loaded 1 password hash (Mac OS X 10.7+ salted SHA-512 [64/64 CommonCrypto])
hacker1984       (emmanuelgoldstein)
guesses: 1  time: 0:00:00:05 DONE (Wed Jan 22 20:26:05 2014)  c/s: 1264K  trying: AVSH - LWNSWhite
Use the "--show" option to display all of the cracked passwords reliably
```

- **Note**: If you would like to re-run this hash crack again, remove the `john.pot` file from the `run/` directory.

### 3. User Keychain Cracking – John the Ripper (JTR)

- Next, we'll crack (or at least start) Emmanuel's login keychain using John the Ripper.
- Use JTR's `keychain2john` script to extract the hash from the user's login keychain file in the input file format that JTR requires. Save this file to your FOR518 directory using the name `jtr_EM_keychain.txt`.
- Use the `john` program to crack the password for this user account.
    i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password. We'll use the same dictionary located in the Dave Grohl "`wordlists`" directory as `emmanuelgoldstein_memory_strings.txt`.
        1. This file is not a requirement, however it may make password cracking faster.

```
$ ./keychain2john ../../FILES\ TO\ CRACK/login.keychain >
~/FOR518/jtr_EM_keychain.txt

$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt
~/FOR518/jtr_EM_keychain.txt
```

1. What kind of hash is being cracked here?
    a. "PBKDF2-HMAC-SHA-1 3DES"

```
nibble:run sledwards$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt ~/FOR518/jtr_EM
_keychain.txt
Loaded 1 password hash (Mac OS X Keychain PBKDF2-HMAC-SHA-1 3DES [32/64])
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
hacker1984       (login.keychain)
guesses: 1  time: 0:01:31:50 DONE (Tue Jan 21 19:52:36 2014)  c/s: 1528  trying: [^_]
```

- **This will take a while to run, even with the dictionary file.** This is due to the PBKDF2 encryption.
- Press the Spacebar to show the status of the password cracking process.
- You may use Control+C to kill the process. **You do not need to continue crack this password.**
    i. This process took about an hour and a half on a Macbook Pro laptop.
- **Note**: If you would like to re-run this hash crack again, remove the `john.pot` file from the `run/` directory.

- You can now try to open the login.keychain file from the "FILES TO CRACK" directory using Keychain Access.app located in Applications/Utilities/ directory. Double-click the keychain file or using File | Add Keychain.
- Unlock the keychain using the password that you cracked (Hint: **hacker1984**) by selecting the locked login keychain and clicking "Click to unlock the login keychain." lock above.

2. What email address was used to login to accounts.google.com?
    a. emmanualgoldstein2600@gmail.com
3. What is the password for this account?
    a. Cerealkiller1337
        1. Double-click the item of interest and check the box to "show password" and input the password.

**4. Encrypted DMG Cracking – John the Ripper (JTR)**

- Next, we'll crack (or at least start) an encrypted DMG file.
- Use JTR's `dmg2john` script to extract the hash from the encryption key from the file in the input file format that JTR requires. Save this file to your FOR518 directory using the name `jtr_EM_dmg.txt`.
- Use the `john` program to crack the password for this user account.
    i. The `--wordlist` parameter allows the program to intake a dictionary file for use in cracking the password. We'll use the same dictionary located in the Dave Grohl "`wordlists`" directory as `emmanuelgoldstein_memory_strings.txt`.
        1. This file is not a requirement, however it may make password cracking faster.

```
$ ./dmg2john ../../FILES\ TO\ CRACK/protected.dmg >
~/FOR518/jtr_EM_dmg.txt

$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt
~/FOR518/jtr_EM_dmg.txt
```

```
nibble:run sledwards$ ./dmg2john ../../FILES\ TO\ CRACK/protected.dmg > ~/FOR518/jtr_EM_dmg.txt
protected.dmg (DMG v2) successfully parsed, iterations count 196078
nibble:run sledwards$ ./john --wordlist=../../DaveGrohl/wordlists/emmanuelgoldstein_memory_strings.txt  ~/FOR518/jtr_E
M_dmg.txt
Loaded 1 password hash (Apple DMG PBKDF2-HMAC-SHA-1 3DES / AES [32/64])
guesses: 0  time: 1:14:36:40 15.02% (ETA: Sat Feb  1 11:25:06 2014)  c/s: 7.81  trying: ion" id="prevent-prompting-for
-location" onclick="PrivacyView.al
Session aborted
```

- This will take a while to run, even with the dictionary file. This is due to the PBKDF2 encryption. This screenshot example estimates it would have taken approximately 12-13 days to crack using the dictionary file provided.
- Press the Spacebar to show the status of the password cracking process.
- You may use Control+C to kill the process. This cracking process will take much longer than the keychain. This is due to the "iterations" count. To learn more about why this takes much longer from a humorous article - read is this one: blog.whitehatsec.com/cracking-aes-256-dmgs-and-epic-self-pwnage/.
- **Note**: If you would like to re-run this hash crack again, you may need to remove the `john.pot` file from the `run/` directory.

## *Exercise – Key Takeaways*

- **Get familiar with Dave Grohl and John the Ripper password cracking utilities.**
- **Understand the speed differences when using a dictionary file as well as speed differences of different encryption methods.**

This page intentionally left blank.

# Exercise 5.1 – Decoding iOS Artifacts

## *Objectives*

- **Examine and decode an iOS backup file recovered from a Mac.**
- **Gain experience with examining PLists and database files from iOS backup files.**
- **Learn how to utilize BlackLight and iBackupBot for forensic examinations.**
- **Learn how to utilize keyword searching to aid in your investigation.**

## *Exercise Preparation*

**Exercise File Preparation** – Locate the files located in the `Exercise Files/Exercise 5.1 - Decoding iOS Artifacts` directory on your FOR518 USB drive. Unarchive the Lab5.1.7z file prior to starting the lab exercise.

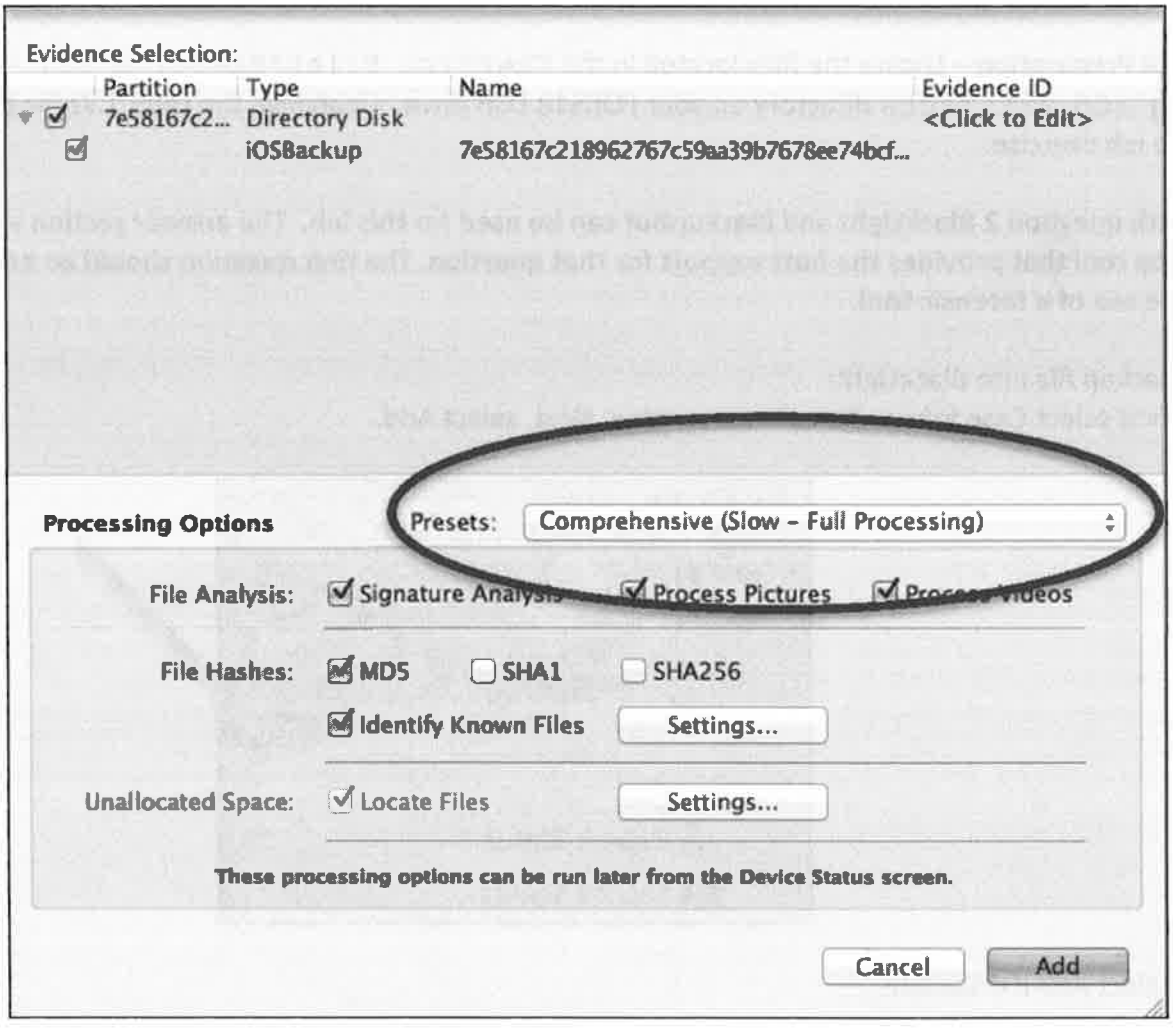
**Starting with question 2 BlackLight and iBackupBot can be used for this lab. The answer section will highlight the tool that provides the best support for that question. The first question should be answered without the use of a forensic tool.**

To load a backup file into BlackLight:

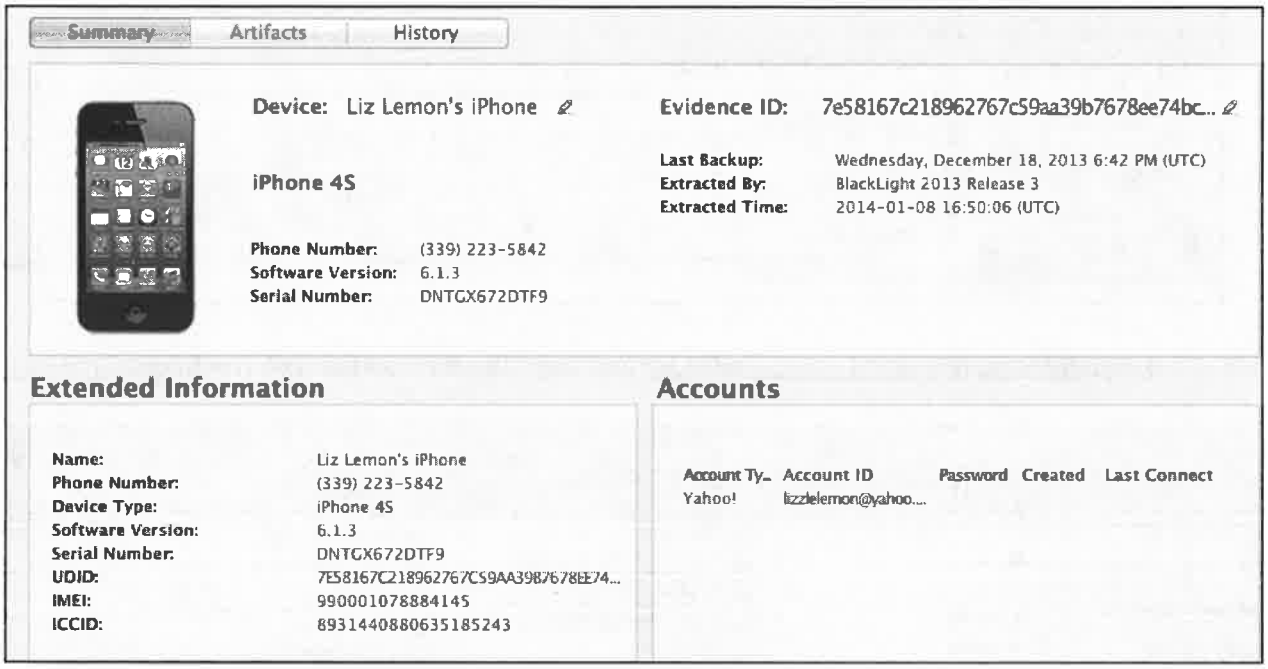
1. First select Case Info and create a new case. Next, select Add.


2. Select Add iOS Backup...


3. Check the box for the highest level of the Directory Disk. Change the Presets to "Comprehensive (Slow – Full Processing). Determine which hashing you would like to have enabled and select Add.



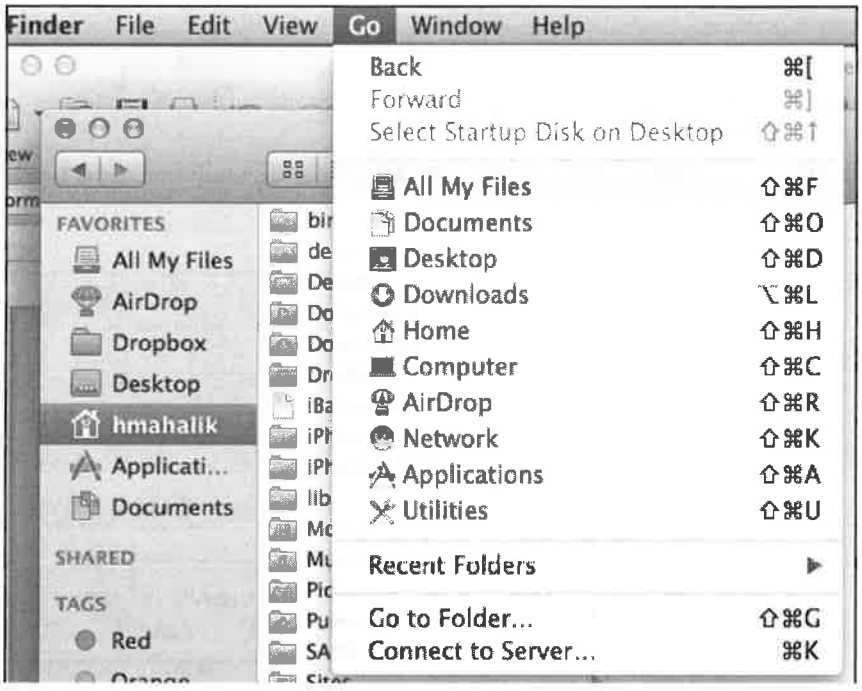

4. The iOS backup file will parse and be loaded into BlackLight for your examination.



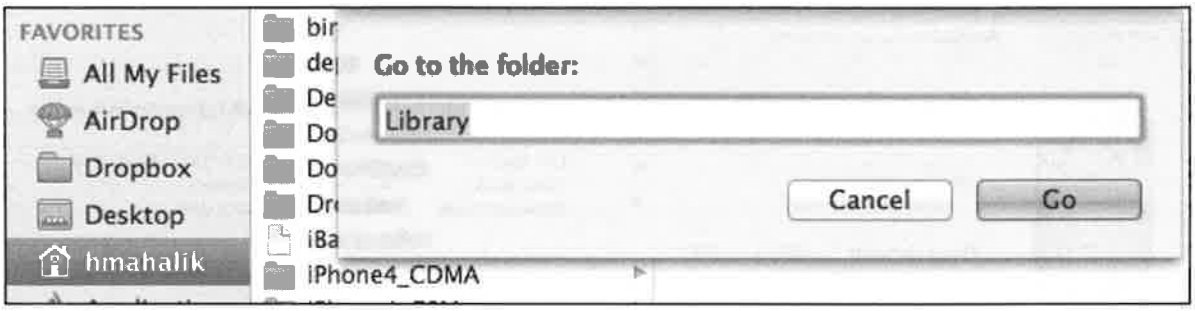

**To load into iBackupBot go to File>Open>Backup Directory and navigate to your backup file in Lab 5.1.**

**To move your backup to the correct location follow the steps below.**

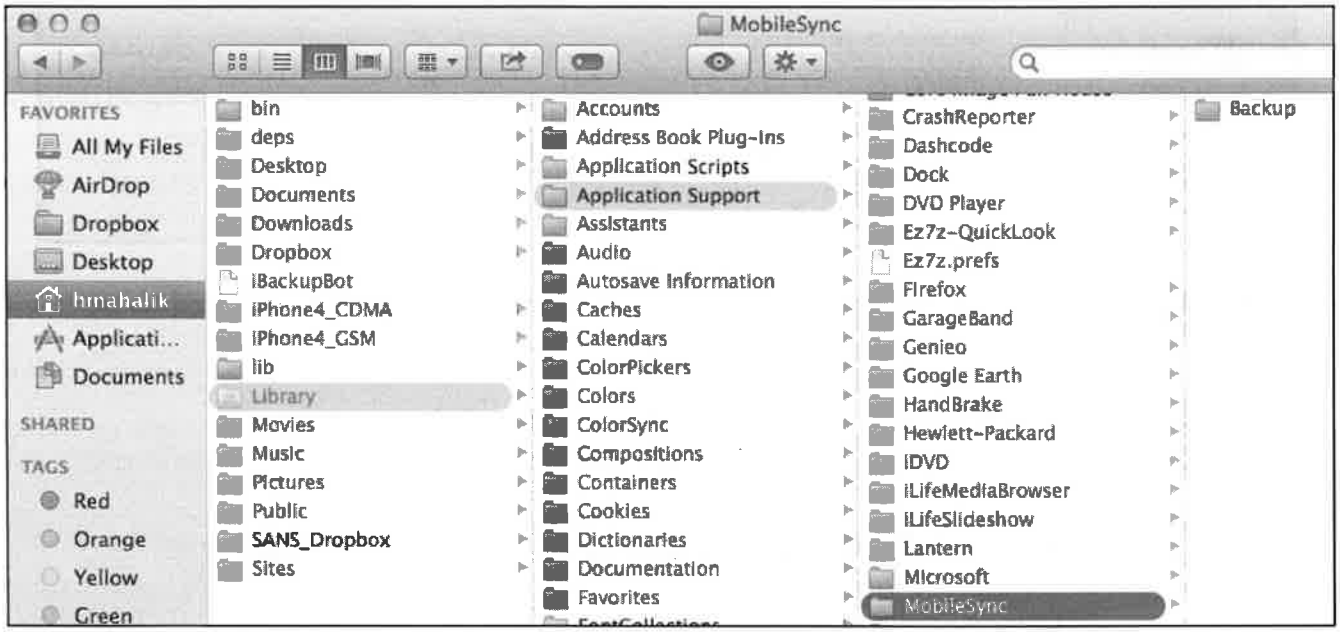
1. Click on your user profile and select Go in the Finder.



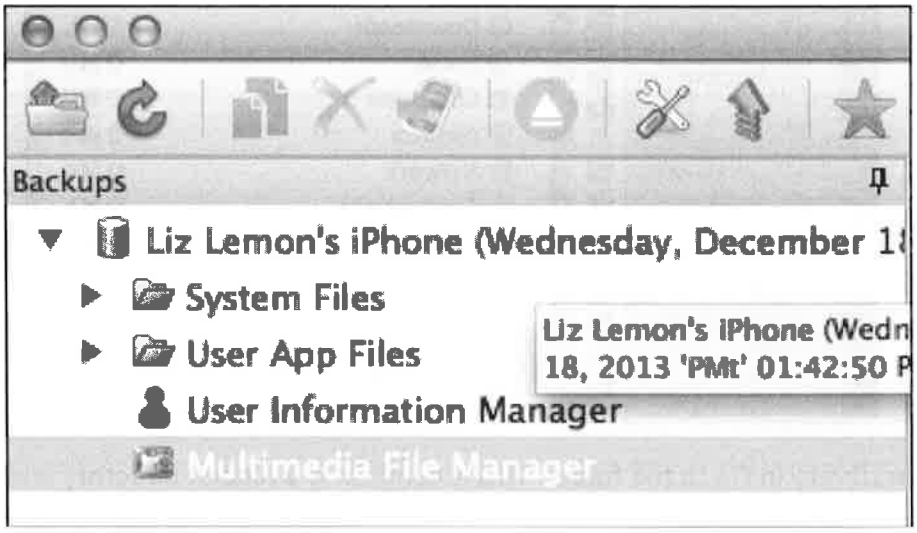

2. Type Library in Go to the folder. This is commonly a hidden directory and will not be shown to the user by default.




3. Navigate to the `MobileSync` folder and copy the backup file into the Backup folder.




4. Launch iBackupBot and the backup files in the folder will be added automatically. Note, all backups will be added!



## *Exercise – Questions*

1. **Prior to loading the backup file into a tool, answer the question below. Then, load the backup into your tool of choice and verify that all of the questions were answered by your tool!**
   - **A.** What is the device name?

     _____

   - **B.** What are three applications installed on the device?

     _____

   - **C.** What type of iOS device is this backup from?

     _____

   - **D.** Was the backup encrypted?

     _____

   - **E.** Did the device contain a passcode?

     _____

2. **Were location coordinates found for `IMG_0012.JPG`? If so, where was this photo taken?**

   _____

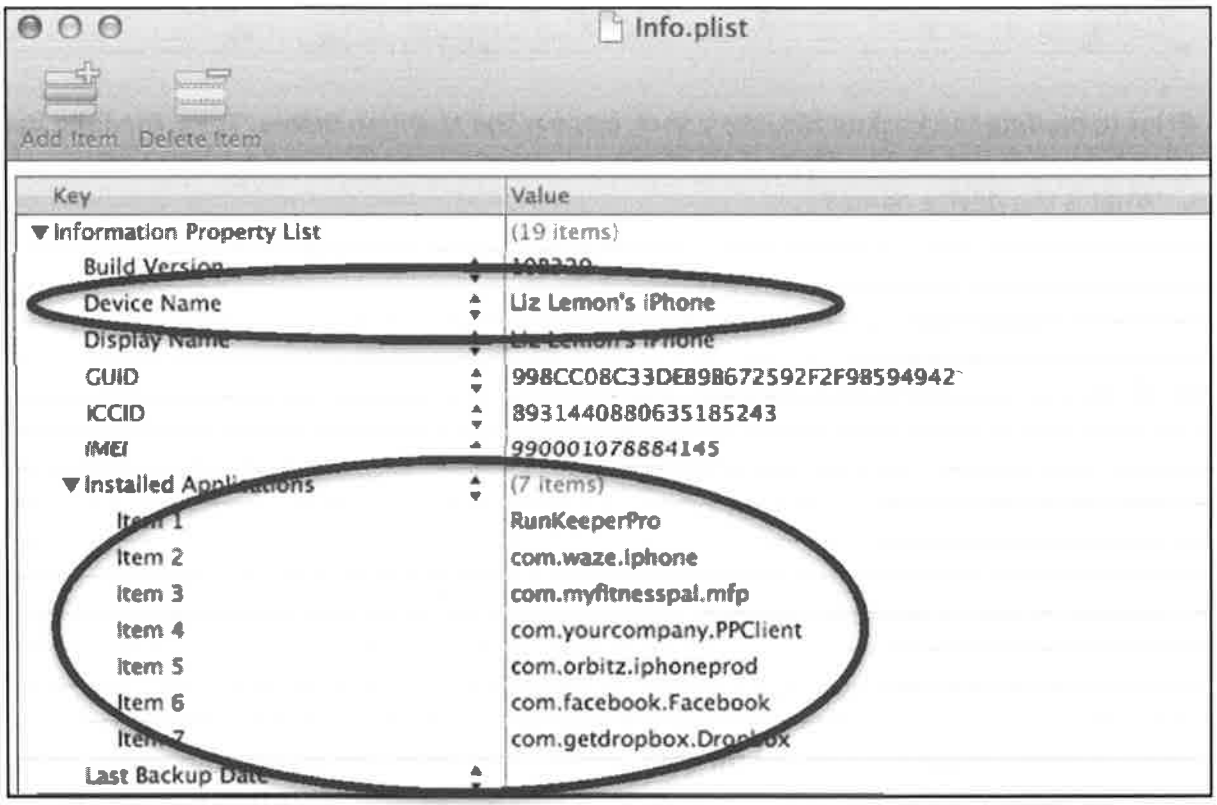
3. **What was the last Email address used for the Facebook account in this backup file? How can you manually verify the tool results?**

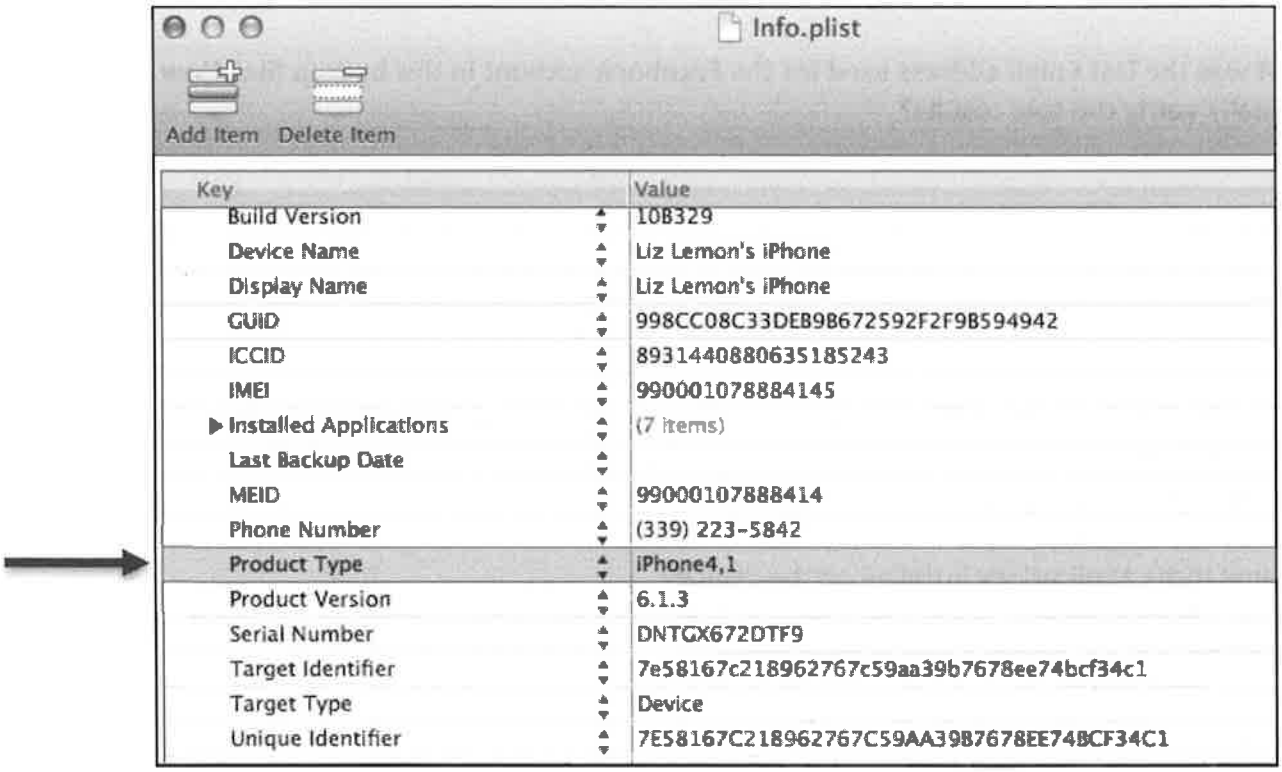
   _____

## *Exercise – Answers*

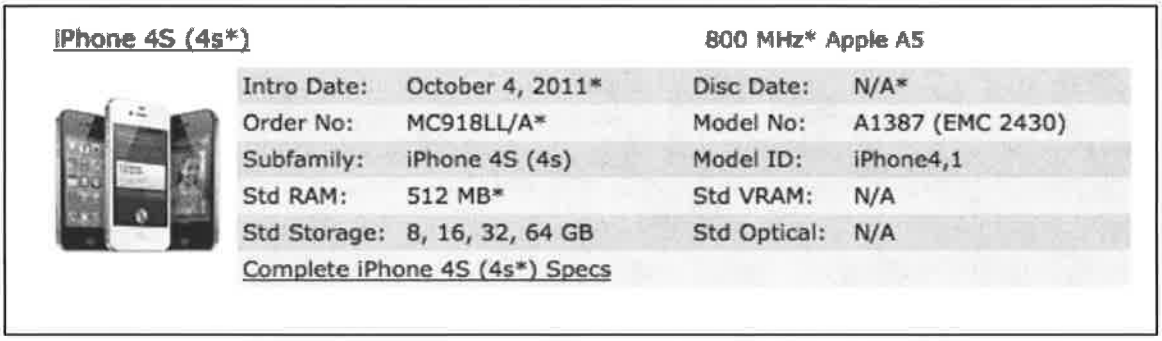
1. **Prior to loading the backup file into a tool, answer the question below. Then, load the backup into your tool of choice and verify that all of the questions were answered by your tool!**
   - A. What is the device name?
     - a. Liz Lemon's iPhone
   - B. Name three applications installed on the device?
     - a. RunKeeperPro, Waze, Myfitnesspal, Yourcompany, Orbitz, Facebook, Dropbox
   - C. What type of iOS device is this backup from?
     - a. iPhone 4s
   - D. Was the backup encrypted?
     - a. No
   - E. Did the device contain a passcode?
     - a. No

   The `Info.plist` file can be examined to answer some of the questions above. The device name, installed apps, and device type can be determined from examining this property list.




The Product Type above can be searched on the Internet using the keyphrase iPhone 4,1 represents or iPhone 4s.

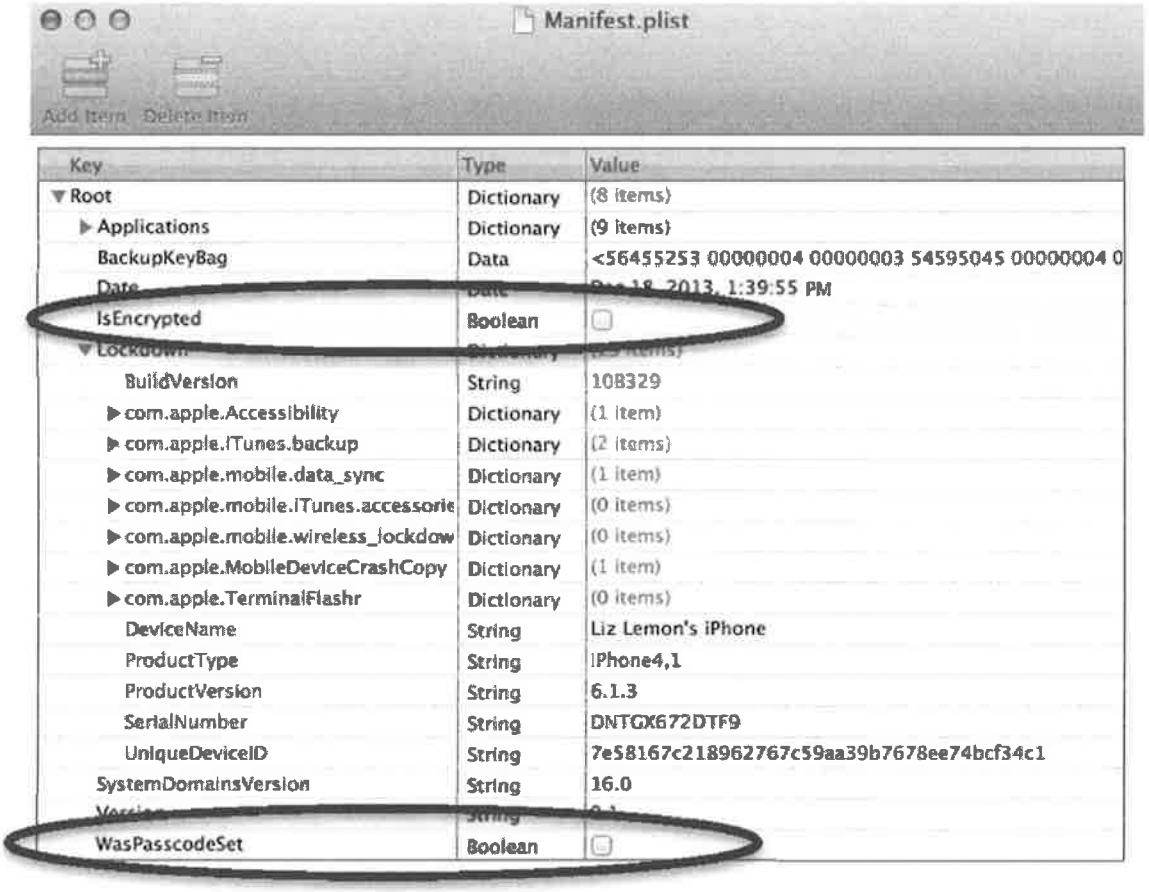


iPhone 4S (4s*) 800 MHz* Apple A5

| | | | |
|---|---|---|---|
| Intro Date: | October 4, 2011* | Disc Date: | N/A* |
| Order No: | MC918LL/A* | Model No: | A1387 (EMC 2430) |
| Subfamily: | iPhone 4S (4s) | Model ID: | iPhone4,1 |
| Std RAM: | 512 MB* | Std VRAM: | N/A |
| Std Storage: | 8, 16, 32, 64 GB | Std Optical: | N/A |

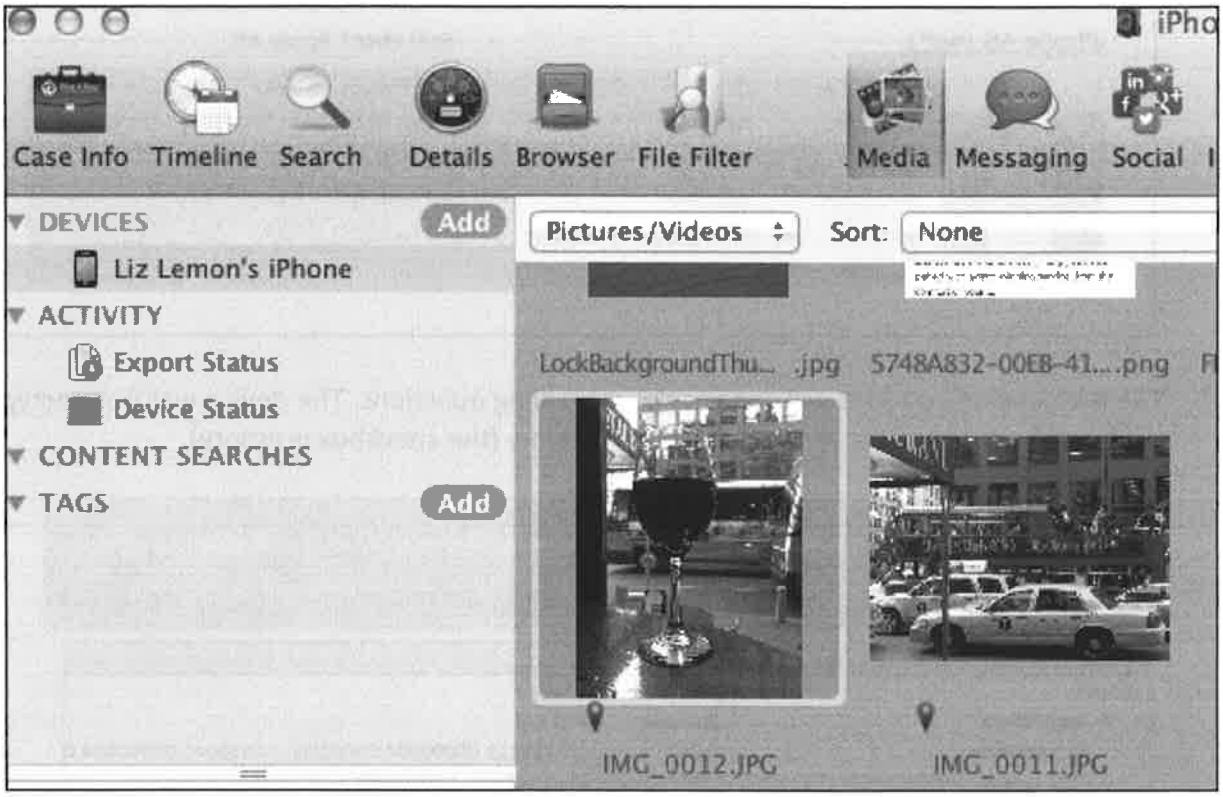
Complete iPhone 4S (4s*) Specs

The `Manifest.plist` will answer the remaining questions. The device was not encrypted and the device did not contain a passcode as shown below (the checkbox is empty).

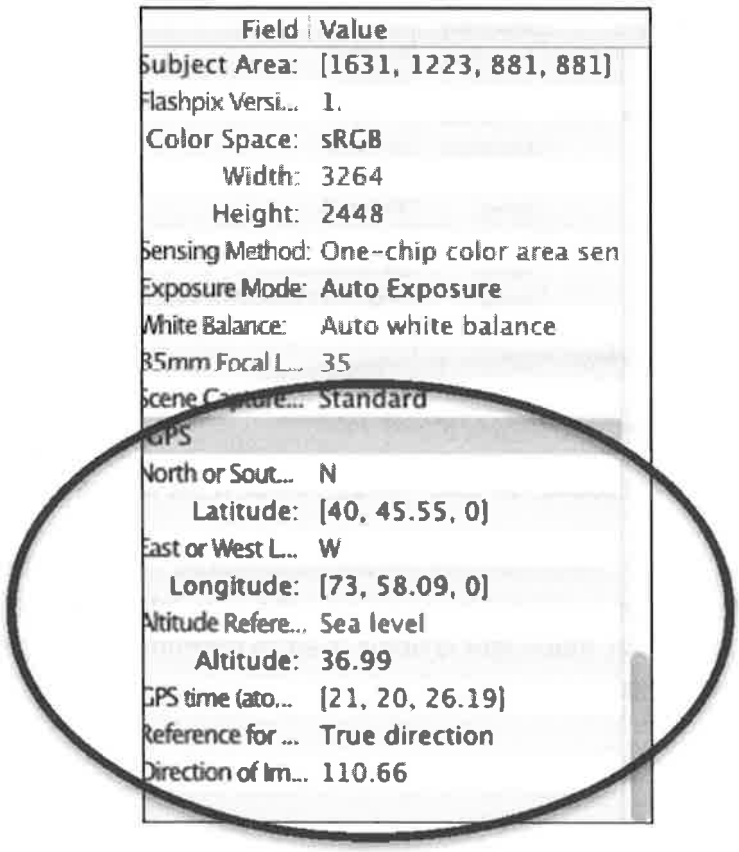



**2. Were location coordinates found for `IMG_0012.JPG`? If so, where was this photo taken?**

A quick look at `IMG_0012.JPG` does not reveal coordinates, however, they are present in the metadata of the file and can be found by examining the raw data in BlackLight. This data cannot be found in iBackupBot. Below, BlackLight is being used to examine the image file. Click on Media in the toolbar to access the images.

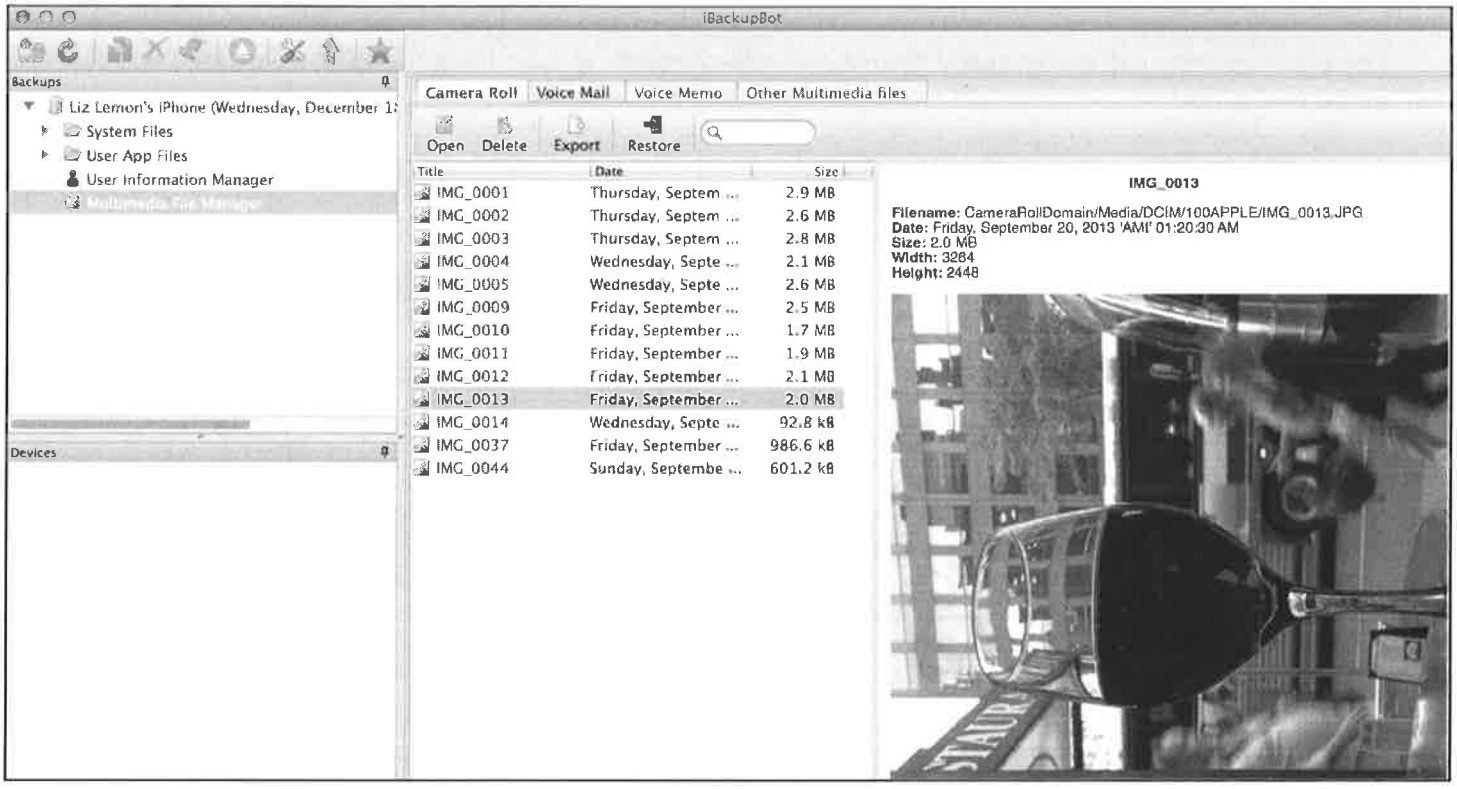



The bottom left pane will show the Field and Value for the metadata. Scrolling down on this file will reveal coordinates.




The coordinates can be mapped using the Internet as shown below. This image was taken on 115 5th Ave, NY, NY.

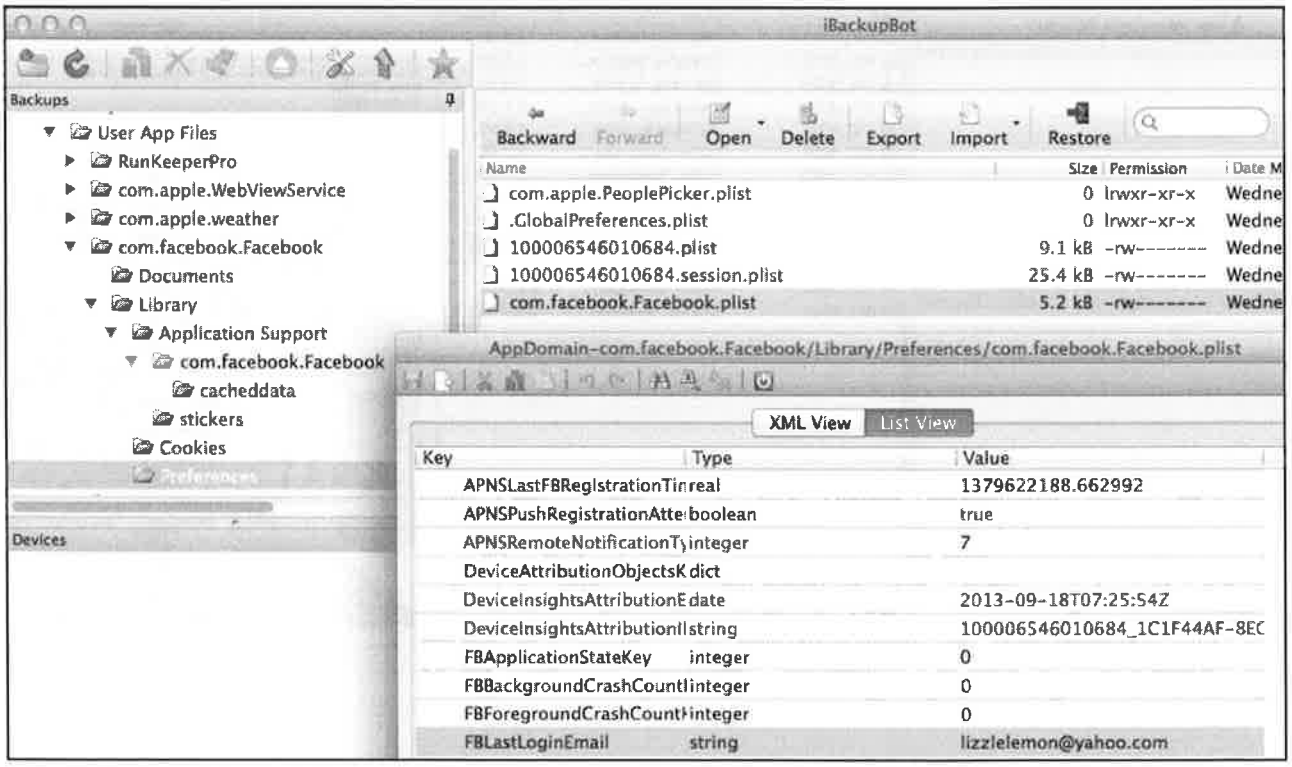

As shown below, iBackupBot does not provide access to this metadata.



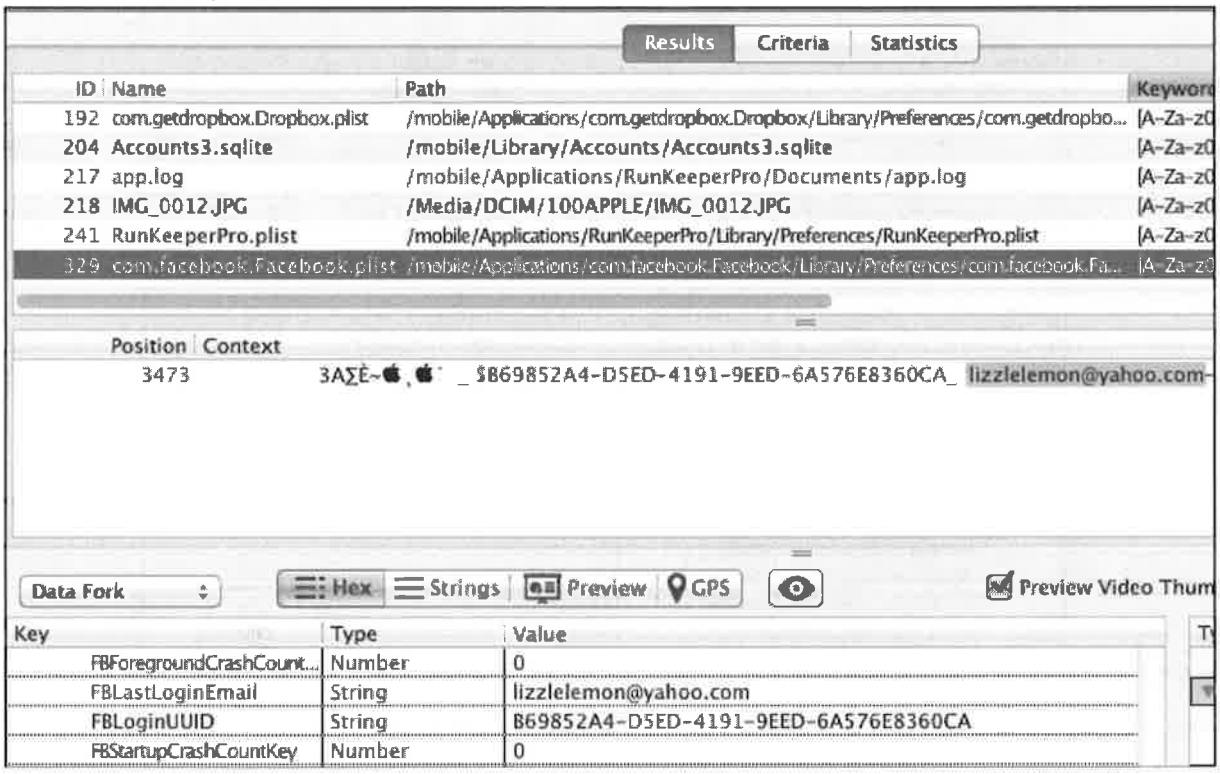
**3. What was the last Email address used for the Facebook account in this backup file? How can you manually verify the tool results?**

The Email address last used to access Facebook in this iOS backup file was "lizzlelemon@yahoo.com." This Email address can easily be recovered using BlackLight and examining the profile under Social.

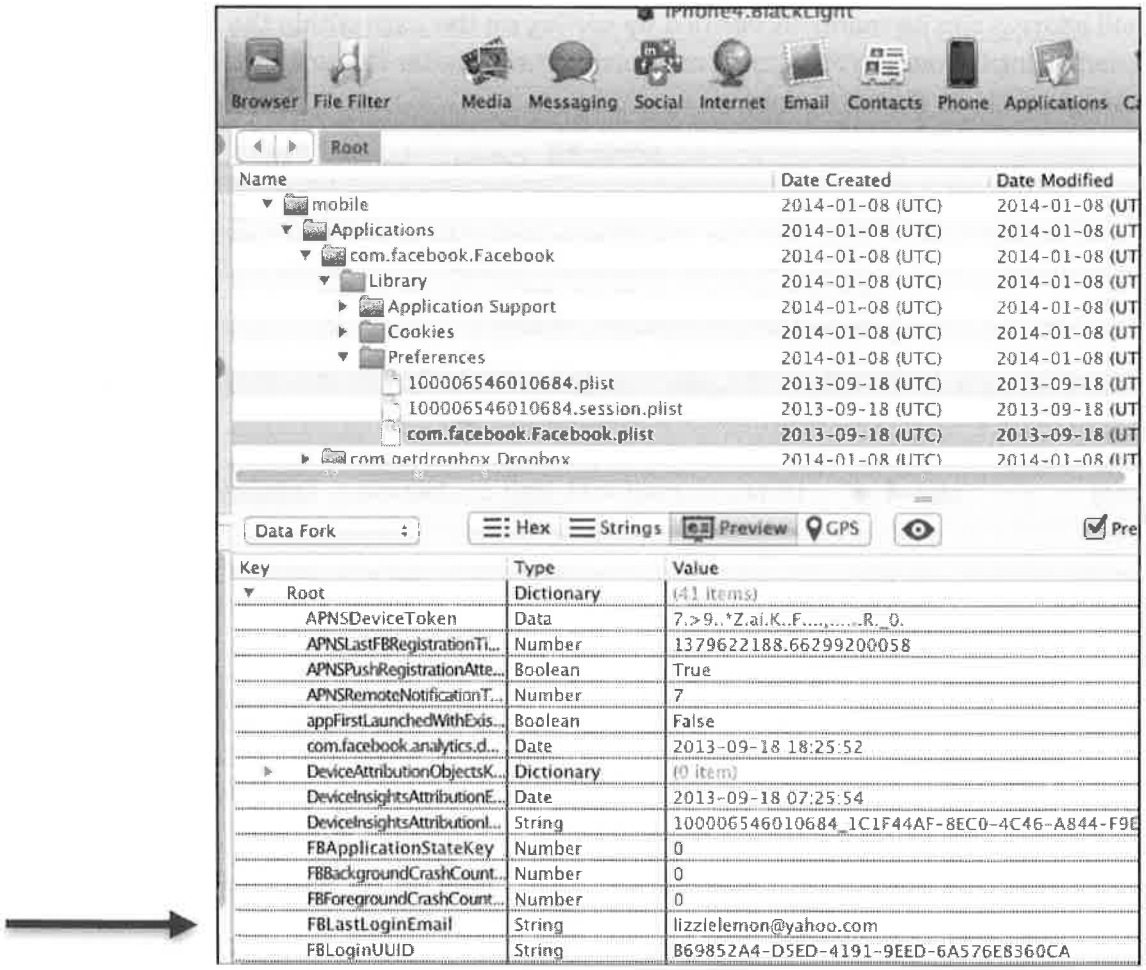

The Email address can be manually recovered in iBackupBot by examining the PList associated with Facebook as shown below:



In BlackLight, the Email address can be manually verified by sorting on the path within the Email address search hits and then narrowing it down to Facebook, or by using the Browser feature and navigating to the com.facebook.Facebook.plist file as shown below.




Navigate to the plist file to for manually verify the Email address.




***Exercise – Key Takeaways***

- **You must manually examine the raw metadata associated with image files to uncover data relevant to the investigation. The tools are not created equally and different results may be attained. It is best to manually examine the raw data when possible.**

- **Being able to parse a file system by hand can help when automated tools fail, or if tool validation is necessary.**

# Exercise 5.2 – iOS Artifacts and Third-party Applications

## *Objectives*

- Examine the iOS backup image that was created using iTunes.
- Review Native iOS applications and Third-party applications for user-related data.
- Re-enforce the use of BlackLight as a complementary toolset to existing tools on the Mac to aid in forensic investigations.
- Compare the data contained in the iTunes backup to the File System dump of the same device and determine differences in the content available from different acquisition methods.

## *Exercise Preparation*

**Exercise File Preparation** – Locate the files located in the `Exercise Files/Exercise 5.2 - iOS Artifacts and Third-party Applications` directory on your FOR518 USB drive.

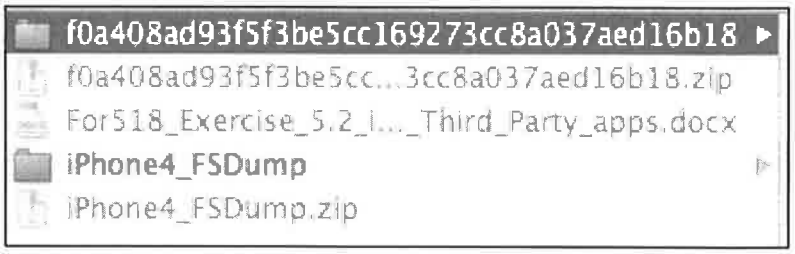
**An iTunes backup of an iPhone 4 has been provided. First, expand the archive file, f0a408ad93f5f3be5cc169273cc8a037aed16b18.zip *[The Mac archiving utility will work fine for this file].* Load the uncompressed backup file into BlackLight and answer the questions below. In addition to using BlackLight, iBackupBot can be used to examine the backup.**

To load the iTunes backup file into BlackLight:

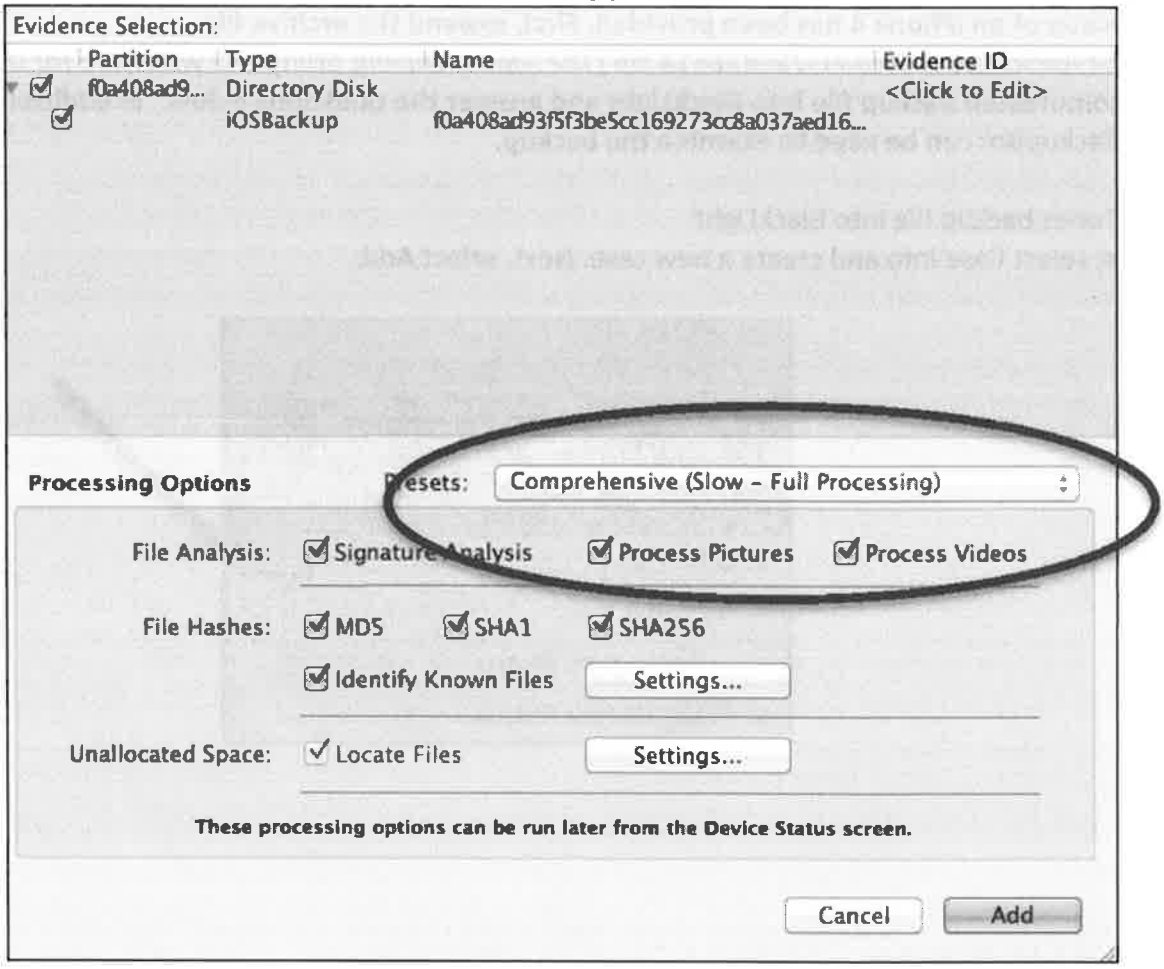
1. First select Case Info and create a new case. Next, select Add.


2. Select Add iOS Backup...

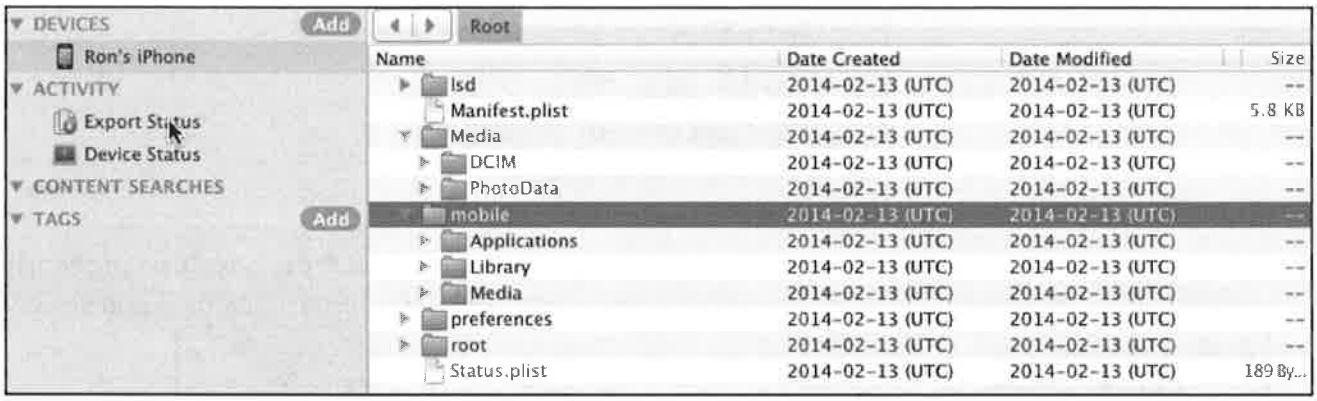

3. Navigate to the **uncompressed** iTunes backup file labeled: "f0a408ad93f5f3be5cc169273cc8a037aed16b18".



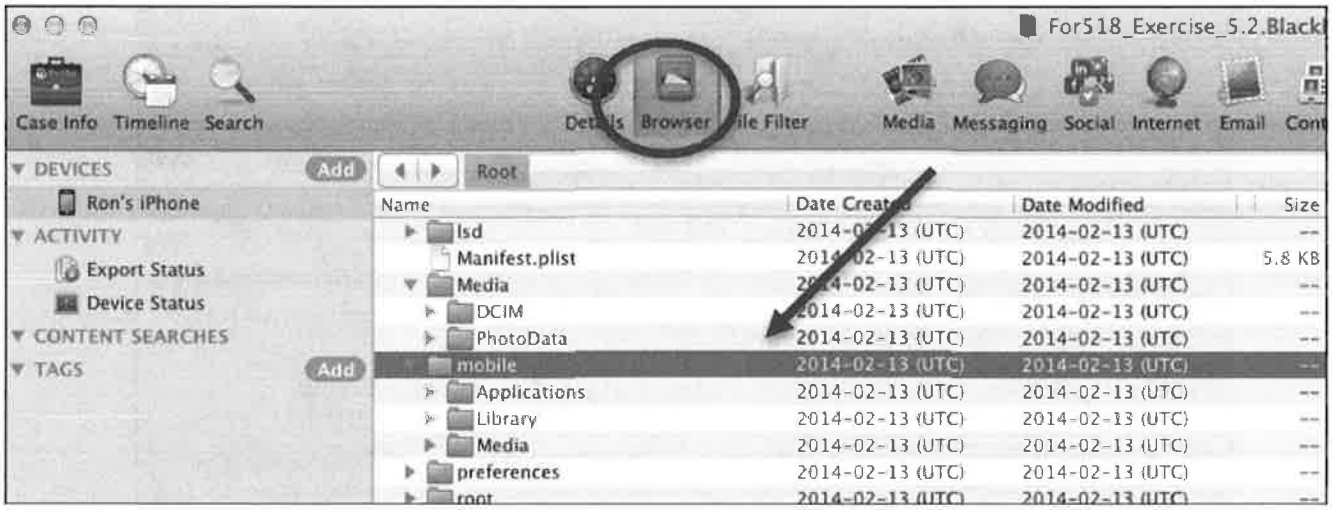

4. Check the box for the highest level of the Directory Disk. Change the Presets to "Comprehensive (Slow – Full Processing)". Determine which hashing you would like to have enabled and select Add.



4. The image will parse and be loaded into BlackLight for your examination.




5. To begin viewing the file system of the device, click on "Browser". Expand the sub-directories nested under the Data directory until you reach the "mobile" directory.




**In addition to the iTunes backup, a File System acquisition of the same device has been provided for comparison. To Load the folder containing the File System image, follow the steps outlined below.**

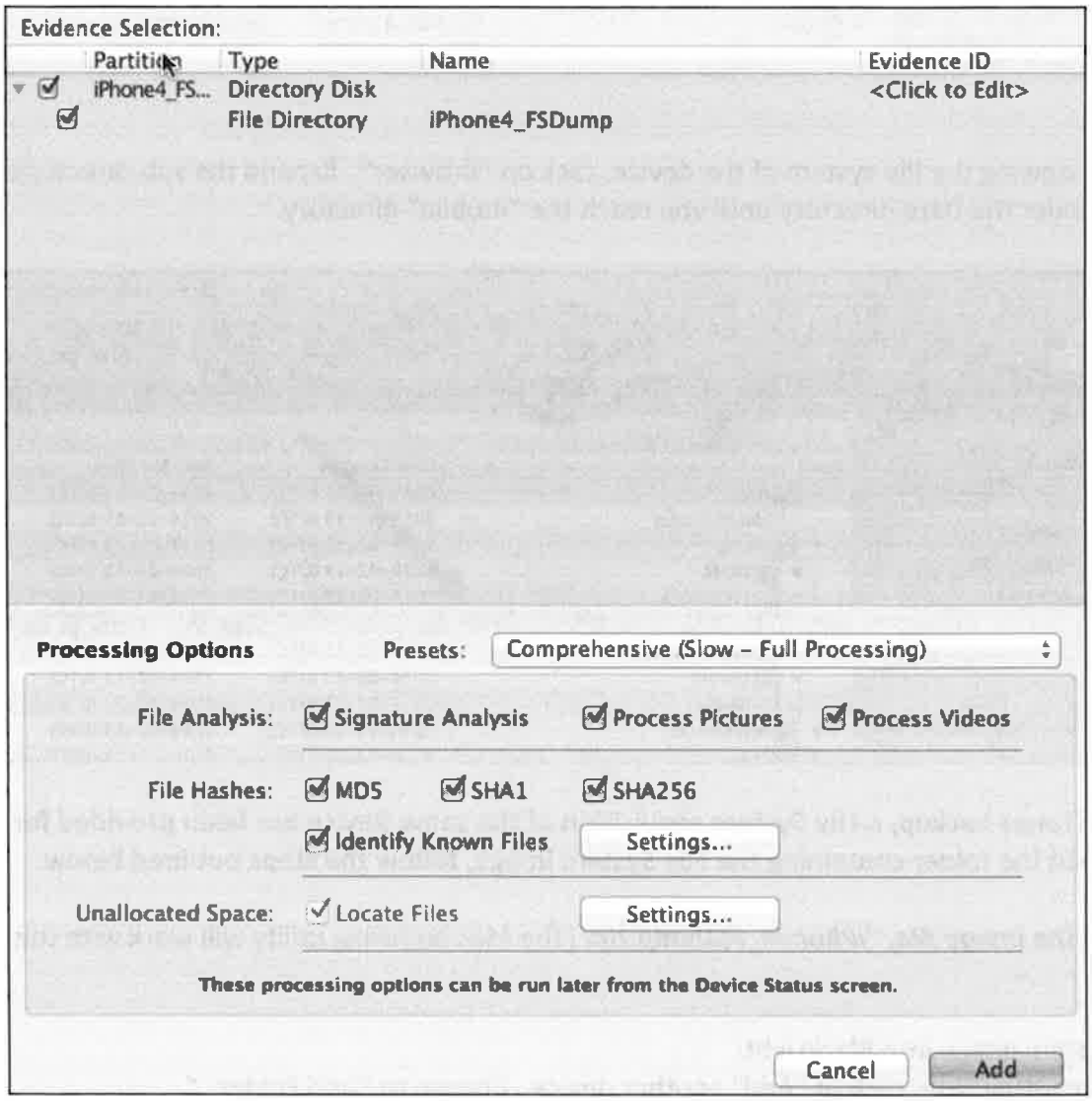
***Start by unzipping the image file, "iPhone4_FSDump.zip.*** [The Mac archiving utility will work with this file.]

To load the File System image into BlackLight:

1. In your existing case, click to "Add" another device. Choose to "Add Folder..."

2. Navigate to the uncompressed File System acquisition labeled, "iPhone4_FSDump".


3. Check the box for the highest level of the Directory Disk. Change the Presets to "Comprehensive (Slow – Full Processing)". Determine which hashing you would like to have enabled and select Add.




You will need to view the files from both the backup as well as the File System Acquisition in order to answer all of the questions below. (Hint: Some tool features will work on the File System Dump and not the iTunes Backup.

## *Exercise – Questions*

1. **Did the user configure any email accounts set up on the device? If yes:**
   A. In which file did you find this information?

   _____

   B. Can you identify the email address/addresses configured?

   _____

   C. What protocol was used?

   _____

2. **Were GEO location services enabled for images taken with the iPhone camera? If yes, how do you know? Was the device setup for Photo Stream data?**

   _____

3. **Were any movies created using the iPhone camera? How can you verify?**

   _____

**This section focuses on the use of third-party applications and how they store data.**

4. **The following questions are based on the third-party application, Kik:**
   A. What is the Kik username on the device?

   _____

   B. What email address and phone number was used to register?

   _____

   C. Did the user setup a profile picture? If yes, what is it?

   _____

5. **Can you locate any Facebook messages on the device?**

_____

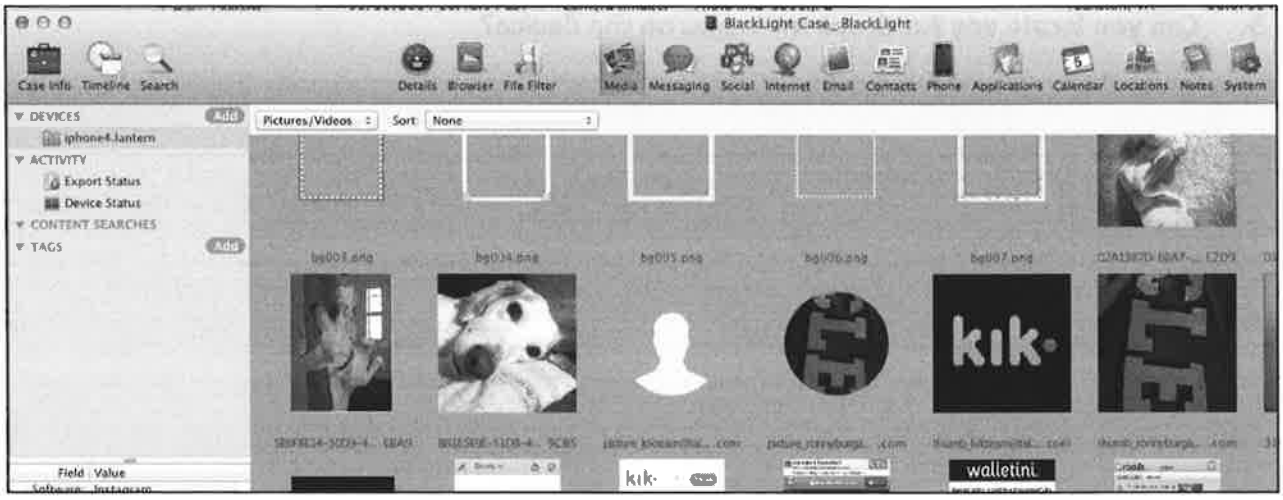
1. **Did the user configure any email accounts set up on the device? If yes:**

One email account was set up to PUSH to the iPhone. This information can be located in the path: private/var/mobile/Library/DataAccess/IMAP-ron.burgandy20@yahoo.com@apple.imap.mail.yahoo.com

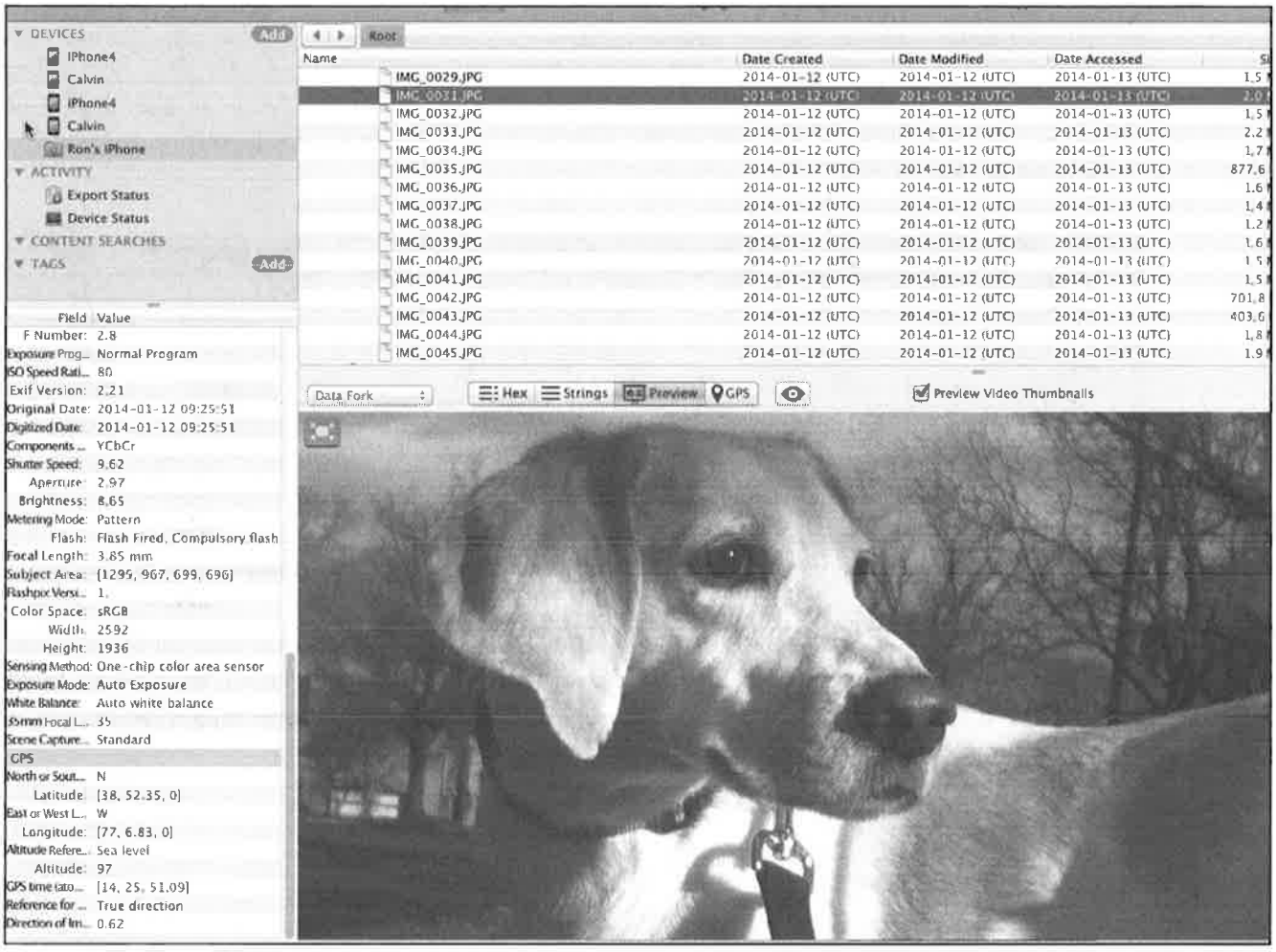
The email address configured is ron.burgandy20@yahoo.com and this was configured using the Internet Message Access Protocol (IMAP).


2. **Were GEO location services enabled for images taken with the iPhone camera? If yes, how do you know? Was the device setup for Photo Stream?**

You could click on the Media Tab in BlackLight to view all of the images captured from the iOS device. This view will supply many images for review.



Navigating to the folder where images taken with the device are stored, Media/DCIM/100APPLE, shows that there are a total of 45 files available for review. First, sort by the file extension to filter out those images that were saved to the device (PNG) and not taken using the device camera. Next, examine the JPG images for EXIF data indicating the GPS was enabled.




By reviewing the EXIF data in the image above, you can see that coordinates have been stored in the metadata of the JPG. In total, six images contain GPS coordinates. These files are:

IMG_0011.JPG
IMG_0012.JPG
IMG_0013.JPG
IMG_0014.JPG
IMG_0029.JPG
IMG_0031.JPG

It is important to note that many times the tool that was used to acquire the device does a better job of analyzing the data. The screenshot below is of the "Breadcrumbs" feature in Lantern 3.0.6.

Breadcrumbs

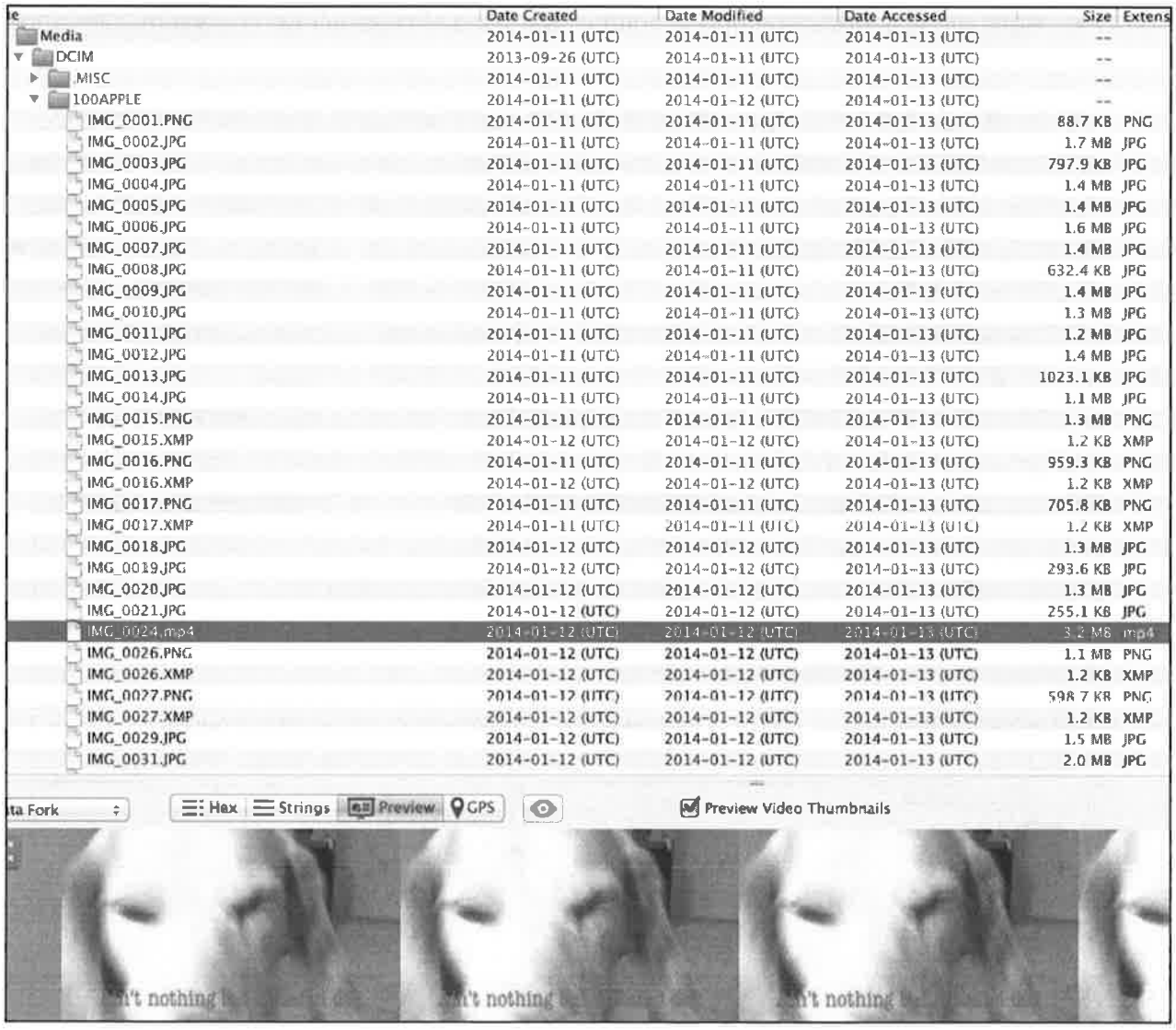
| Time | Type | Summary | Where | Latitude | Longitude |
|---|---|---|---|---|---|
| 01/11/2014 16:48:04 EST | Camera (image) | Photo IMG_0011.JPG | Ballston, VA | 38.8798 N | -77.1062 |
| 01/11/2014 16:51:21 EST | Camera (image) | Photo IMG_0012.JPG | Ballston, VA | 38.8797 N | -77.1062 |
| 01/11/2014 16:53:08 EST | Camera (image) | Photo IMG_0013.JPG | Buckingham, VA | 38.8763 N | -77.1082 |
| 01/11/2014 16:53:17 EST | Camera (image) | Photo IMG_0014.JPG | Buckingham, VA | 38.8763 N | -77.1082 |
| 01/11/2014 20:55:41 EST | Facebook (message) | Facebook msg to Ronny Burgandy: I missed you soo... | Arlington Forest... | 38.8719 N | -77.1115 |
| 01/11/2014 20:57:01 EST | Facebook (message) | Facebook msg to Ronny Burgandy: I was drinking na... | Arlington Forest... | 38.8719 N | -77.1114 |
| 01/12/2014 09:25:39 EST | Camera (image) | Photo IMG_0029.JPG | Brandon Village,... | 38.8725 N | -77.1138 |
| 01/12/2014 09:25:52 EST | Camera (image) | Photo IMG_0031.JPG | Brandon Village,... | 38.8725 N | -77.1138 |
| 01/12/2014 10:55:47 EST | Facebook (message) | Facebook msg to Lee Crognale: It was nice to get aw... | Arlington Forest... | 38.8719 N | -77.1115 |

To verify whether or not the device was configured for Photo Stream, you can examine the Media Directory. This directory is lacking the PhotoStream sub-directory and therefore it was not configured on this device.

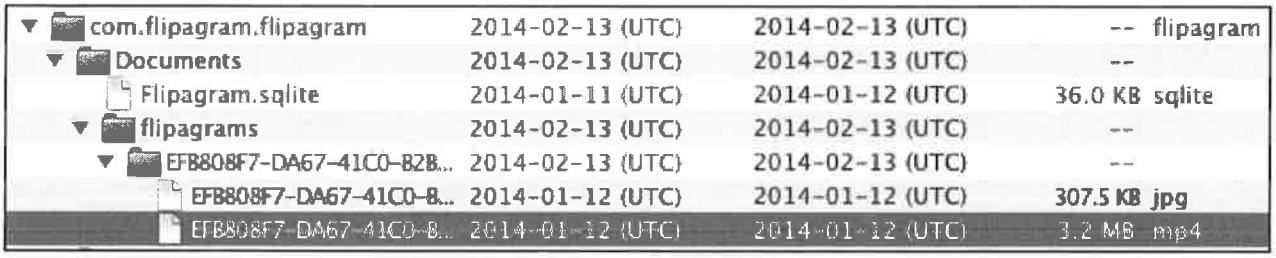
| | | | |
|---|---|---|---|
| Media | 2014-02-13 (UTC) | 2014-02-13 (UTC) | 2014-02-13 |
| DCIM | 2014-02-13 (UTC) | 2014-02-13 (UTC) | 2014-02-13 |
| PhotoData | 2014-02-13 (UTC) | 2014-02-13 (UTC) | 2014-02-13 |
| mobile | 2014-02-13 (UTC) | 2014-02-13 (UTC) | 2014-02-13 |
| Applications | 2014-02-13 (UTC) | 2014-02-13 (UTC) | 2014-02-13 |
| Library | 2014-02-13 (UTC) | 2014-02-13 (UTC) | 2014-02-13 |

**3. Were any movies created using the iPhone camera? How can you verify?**

The same directory was examined for the evidence of videos created with the iPhone camera, MEDIA/DCIM/100APPLE. This directory was examined for video files, which will utilize the extension MOV. There are no MOV files in this directory. In your examination, you may have noticed another file extension indicative of a movie file in this folder.



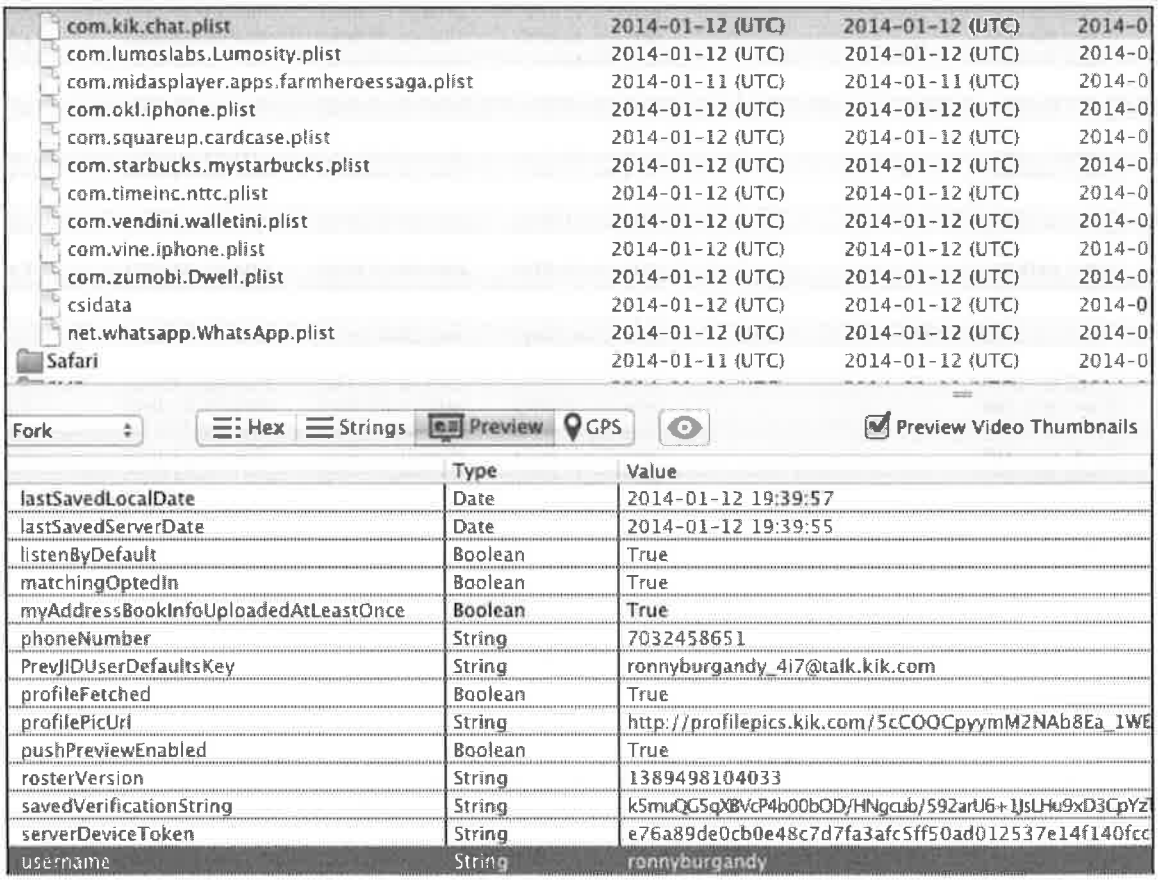

Additional review of this MP4 file reveals that it was created using a third-party application named Flipagram, not with the built-in iOS camera.




**This section focuses on the use of third-party applications and how they store data.**

4. **The following questions are based on the third-party application, Kik:**
   - Data was found in mobile/Applications/com.kik.chat/Library/Preferences/com.kik.chat.plist
   - What is the Kik username on the device? Ronnyburgandy

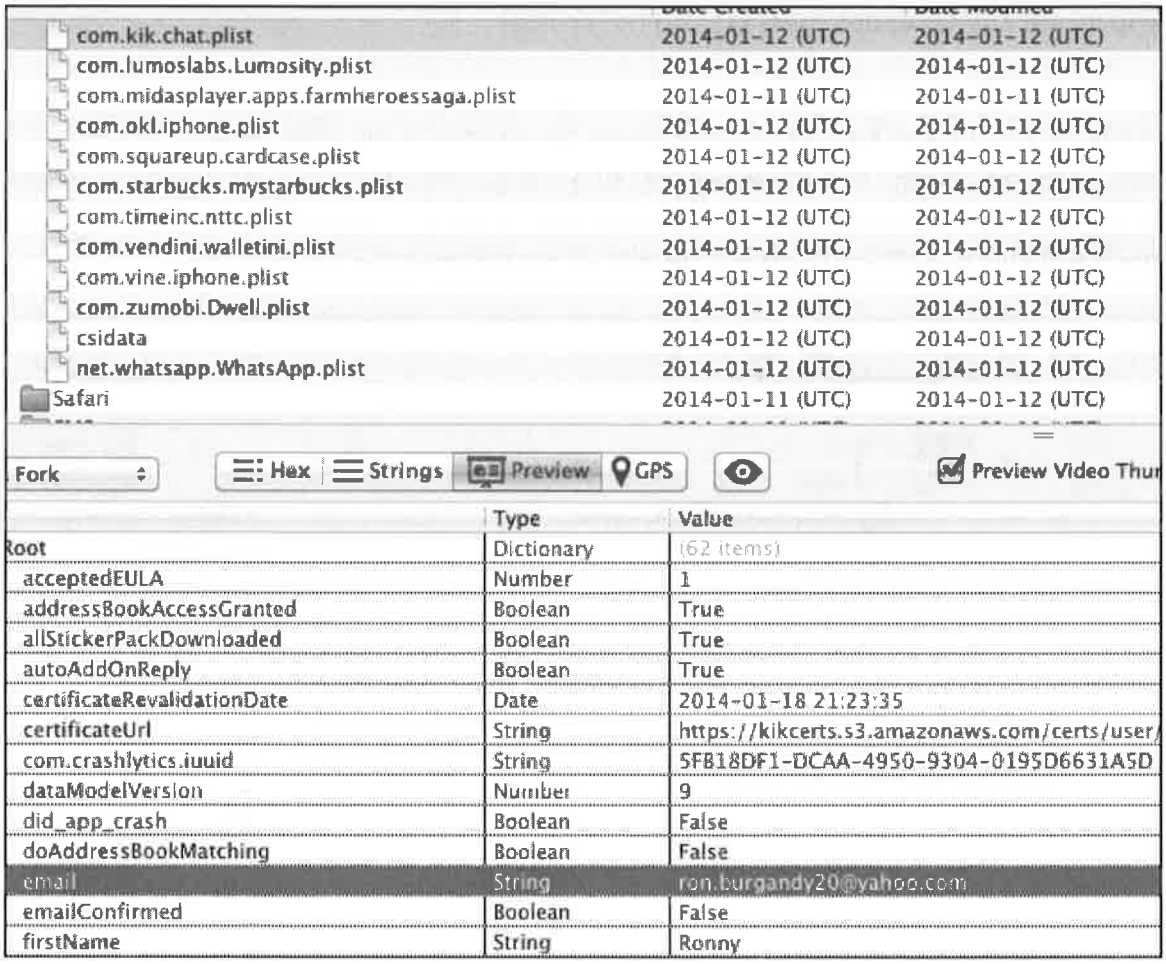
- What email address and phone number was used to register? Ron.burgandy20@yahoo.com, 7032458651



| | | |
|---|---|---|
| com.kik.chat.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| com.lumoslabs.Lumosity.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| com.midasplayer.apps.farmheroessaga.plist | 2014-01-11 (UTC) | 2014-01-11 (UTC) |
| com.okl.iphone.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| com.squareup.cardcase.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| com.starbucks.mystarbucks.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| com.timeinc.nttc.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| com.vendini.walletini.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| com.vine.iphone.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| com.zumobi.Dwell.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| csidata | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| net.whatsapp.WhatsApp.plist | 2014-01-12 (UTC) | 2014-01-12 (UTC) |
| Safari | 2014-01-11 (UTC) | 2014-01-12 (UTC) |

Fork | Hex | Strings | Preview | GPS | Preview Video Thumbnails

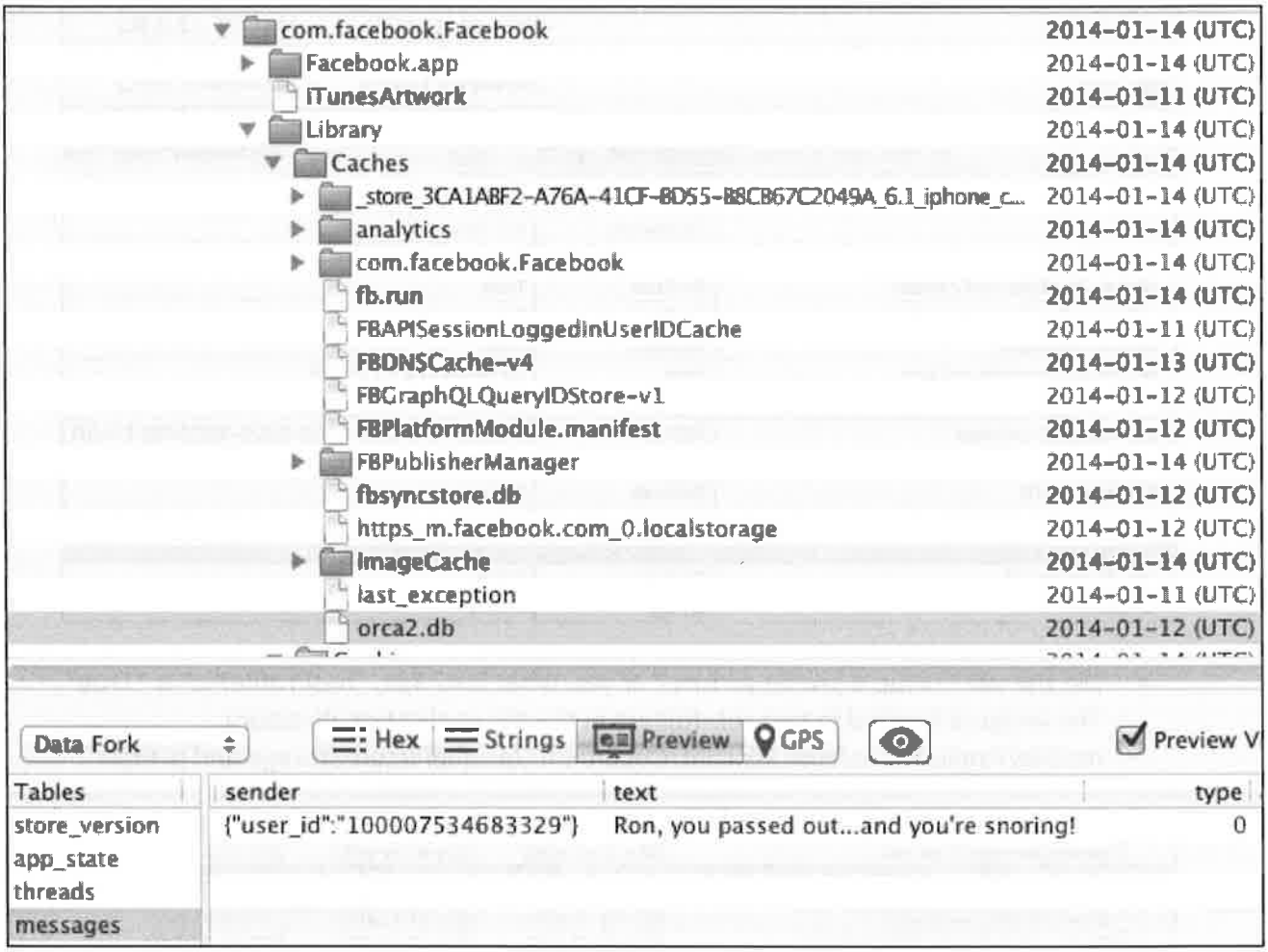
| | Type | Value |
|---|---|---|
| lastSavedLocalDate | Date | 2014-01-12 19:39:57 |
| lastSavedServerDate | Date | 2014-01-12 19:39:55 |
| listenByDefault | Boolean | True |
| matchingOptedIn | Boolean | True |
| myAddressBookInfoUploadedAtLeastOnce | Boolean | True |
| phoneNumber | String | 7032458651 |
| PrevJIDUserDefaultsKey | String | ronnyburgandy_4i7@talk.kik.com |
| profileFetched | Boolean | True |
| profilePicUrl | String | http://profilepics.kik.com/5cCOOCpyymM2NAb8Ea_1WE |
| pushPreviewEnabled | Boolean | True |
| rosterVersion | String | 1389498104033 |
| savedVerificationString | String | k5muQG5qXBvcP4b00bOD/HNgcub/592arU6+1JsLHu9xD3CpYzT |
| serverDeviceToken | String | e76a89de0cb0e48c7d7fa3afc5ff50ad012537e14f140fcc |
| username | String | ronnyburgandy |



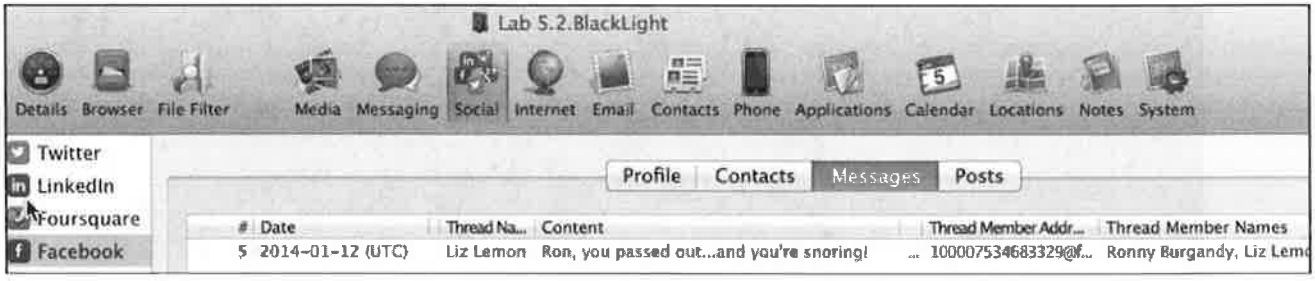

- Did the user setup a profile picture? If yes, what is it? Yes, "hasProfilePic" is "True". The image is located in two sub-folders in the Kik application directory, mobile/Applications/com.kik.chat/Documents/profilePictureStorage and profpix

**5. Can you locate any Facebook messages on the device?**

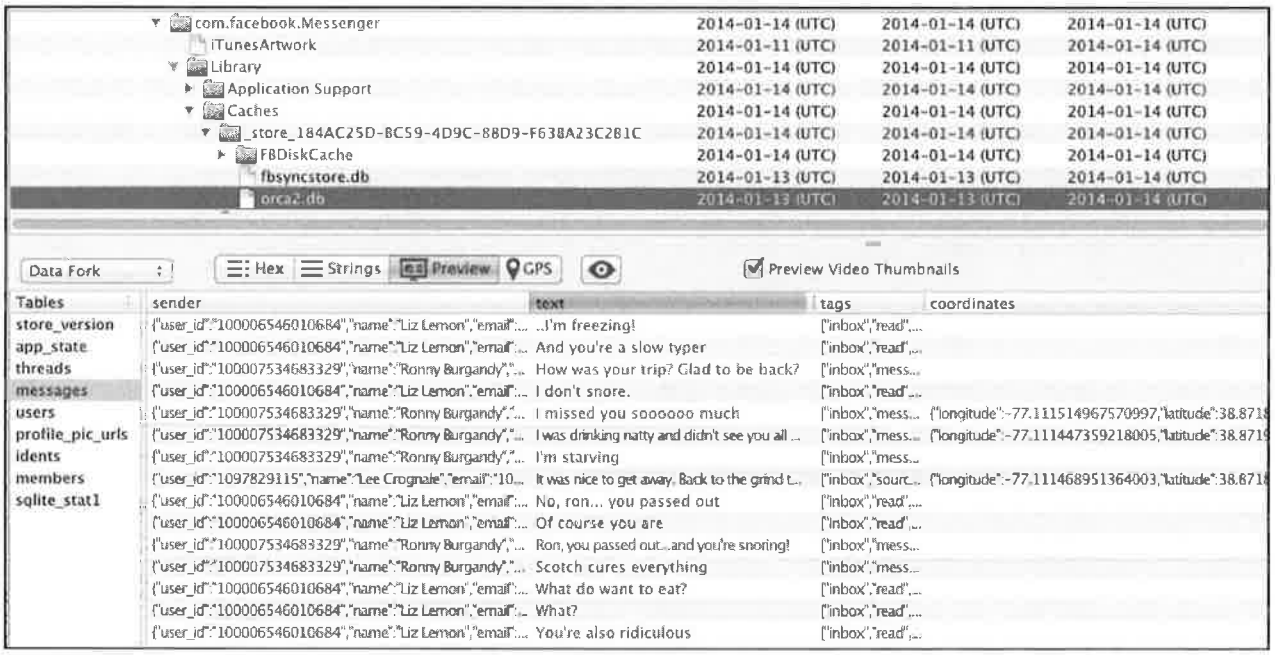
Yes, multiple Facebook messages were recovered but you must be reviewing the file system acquisition of the iPhone 4 in order to find them. Locate the application folder for Facebook in the iPhone4_FSDump. Within this folder you will find a large amount of user data, but the messages will be stored in a SQLite database in the path:
mobile/Applications/com.facebook.Facebook/Library/Caches/orca2.db




You can also utilize the "Social" Tab in BlackLight. This will take you directly to the Facebook Application and provide easy access to Profiles, Contacts, Messages, and Posts.
*(Hint: You will not see the Facebook messages because you are only examining the iOS backup. The image below is from the File System Dump of the same iPhone 4.)*



If you only searched the path indicated above, you would have missed the fifteen messages located in the SQLite database associated with the Facebook Messenger application.
mobile/Applications/com.facebook.Messenger/Library/Caches/_store_184AC25D-BC59-4D9C-88D9-F63BA23C2B1C/orca2.db




It is important to note that some of the Facebook messages include Geo-location data. These coordinates can be viewed by examining the SQLite database.

## *Exercise – Key Takeaways*

- Knowing where to look for application data can greatly expedite your examination.
- Know how third-party applications store their user-related content.
- Use the tools built in "evidence finders", but continue to validate your results by manually examining the file system.
- Compare and contrast the data that is provided by different acquisition methods.

This page intentionally left blank.

# Exercise 6.1
# SANS FOR518 - Mac Forensic Analysis Challenge Preparation

## Objectives

- Create and organize your group.
- Decompress the images and data used for the Mac Forensic Challenge.
- Start preparing your data.
- Get ready to put your new Mac forensic analysis skills to the test!

## Challenge Preparation

1. Challenge Rules
   - This exercise is intended to help you **PREP ONLY**, please do not start conduct analysis.
     - If a group decides to start analysis before the start of Day 6 the group will be penalized.
     - You may strategize with your group, but **DO NOT** start analyzing the images.
   - You will receive an additional exercise with specific tasks for you to accomplish at the beginning of Day 6.
   - You **may** ask the instructor about the tools and techniques you learned this week.
   - You **may not** ask the instructor for hints or answers.
   - Your team is expected to draft a presentation; your instructor will let you know what time the challenge will end.
     i. The most complete, innovative, and accurate presentation will win the challenge
     ii. Voting: Each team will vote for another team (not themselves).
        1. In the case of a tie, the instructor will be the tiebreaker.

2. Form a group of four, you and three of your fellow students.
   - Students leaving early should be on one team.
   - If there are an odd number of students, one team may have a group of five.
   - Depending on the class size this group may be smaller or larger; your instructor will advise you if this is the case.

3. Determine who in your group is going to work on what evidentiary items:
   - You may want to break this up by suspect (i.e.: Dade, Kate, Eugene, or Margo)

   - **Dade Murphy (Washington, DC Metro Area)**
     - FOR518 flash drive - `Exercise Images/Exercise Images.7z` – The same images you've seen all week!
       - System Image
       - RAM Capture

        - Time Machine Backup
- **Kate Libby (Washington, DC Metro Area)**
    - FOR518 flash drive – `Final Challenge Images/ Kate.7z`
        - System Image
        - RAM Capture
    - iPhone (Physical) - FOR518 flash drive - `Final Challenge Images/KateLibby_iPhone.7z`
    - iPhone (BlackLight Logical) - FOR518 flash drive - `Final Challenge Images/BlackLight_Backup_iPhones.BlackLight.7z`
- **Eugene Belford (Las Vegas, NV)**
    - FOR518 flash drive - `Final Challenge Images/Eugene.7z`
        - System Image
        - RAM Capture
        - Incident Response Data
    - iPhone (Physical) - FOR518 flash drive - `Final Challenge Images/EugeneBelford_iPhone.7z`
    - iPhone (BlackLight Logical) - FOR518 flash drive - `Final Challenge Images/BlackLight_Backup_iPhones.BlackLight.7z`
- **Margo Wallace (Las Vegas, NV)**
    - FOR518 flash drive - `Final Challenge Images/Margo.7z`
        - System Image
        - RAM Capture
        - Incident Response Data
        - Time Machine Backup

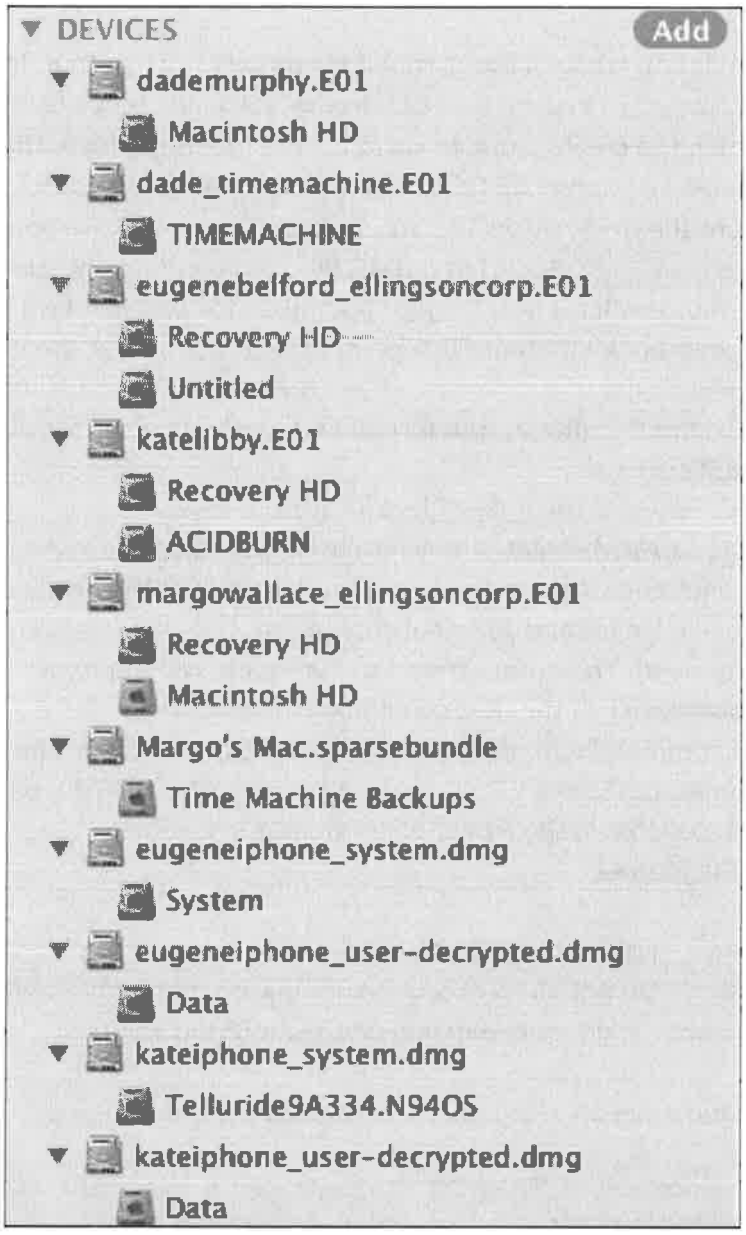
4. Copy and un-archive evidence archives.
    - On your FOR518 flash drive, locate the "Final Challenge Images and Data".
    - Determine which evidentiary items need to be copied and/or un-archived. I would not recommend copying and un-archive everything. This could take a while and will take up ~137GB of disk space.
    - Your instructor will provide you with the password for the archives.
    - Be patient, this will take a while.

5. Create a method of communication and finding documentation for your team:
    - Online documentation (Google Docs) and chat applications have been used successfully in the past to silently communicate your findings to the rest of your team.
    - Find a conference room or other private space where you can speak freely with your team.
    - Be sure to document your findings, you will need to present these to the class.

6. Software Preparation – Please prepare and install any tools you think may help you in your analysis. Please feel free to use tools, scripts, etc., NOT used in this class. Creativity is a plus!

7. Two BlackLight case files have been created and provided to you:
    - `BlackLight_Backup_iPhones.BlackLight` – This BlackLight Case File includes logical backups of Kate and Eugene's iPhones. These backups were acquired by BlackLight. (Full physical, system and data partitions have also been included, separately.)

- `FinalChallenge.BlackLight` – This BlackLight Case File includes all system, time machine backup, and physical phone images. This backup has been pre-processed for you.
    i. You may need to "relocate" forensic images. You can do this by "right-clicking" the evidentiary item and choosing "Relocate Drive Image". The correct mapping is shown in the screenshot below.




8. You may also choose to mount them using the same techniques you used in class.
    - *** Be sure to name your mount points unique to the volume you are reviewing. For example:
        i. `/Volumes/dademurphy_image, /Volumes/dademurphy_mounted`
        ii. `/Volumes/katelibby_image, /Volumes/katelibby_mounted`
        iii. `/Volumes/eugenebelford_image, /Volumes/eugenebelford_mounted/`
        iv. `/Volumes/margowallace_image, /Volumes/margowallace_mounted`
    - You may also choose to rename multiple volumes, once mounted using Finder. You may also use the command: `diskutil rename /Volumes/Untitled/ EUGENE.`

9. **Mount the forensic image; remember to create unique mount points for each image!**

- Using Terminal.app, follow the commands below for Method 1 and Method 2. Choose which method you prefer.

- **Method 1:**
  - Use the `mkdir` command to create a mount point for the `xmount` output. In this class the directory name `dademurphy_image` is used because it will just host the image file.
  - Use the `mkdir` command to create a mount point for the mounted drive. The directory `dademurphy_mounted` is used in this class to represent the mounted disk image.
  - Uses `xmount` to mount the `dademurphy.e01` image (where you have your image located, the example shows `~/FOR518`) as a DMG file. This command requires you to use the `sudo` command, thus it will 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.
    - `Mount Point` – Newly created specifically for this image.
  - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
  - Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ sudo xmount --in ewf --out dmg ~/FOR518/dademurphy.E01
/Volumes/dademurphy_image/

$ hdiutil attach -nomount /Volumes/dademurphy_image/dademurphy.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

- **Method 2:**
  - Use the `mkdir` command to create a mount point for the `ewfmount` output. The directory name `dademurphy_image` is used in the example.
  - Use the `mkdir` command again to create a mount point for the mounted disk image, the example `dademurphy_mounted` in used the example above.
  - Use `ewfmount` to mount the `dademurphy.e01` image to the `/Volumes/dadmurphy_image/` mount point.
  - Use the `ln -s` command to create a symbolic link for the `ewf1` file, name the link `dadeimage.dmg`. (A DMG file is needed for `hdiutil` to recognize the file.)
  - Uses the `hdiutil` command with the "`attach`" verb to mount the newly created DMG volume so it is available in Finder and Terminal application. Use the `-nomount` argument to suppress mounting (for now). The output from this command will display a `/dev/disk#`, use the appropriate disk device in the next command.
  - Use the `mount_hfs` command with the following parameters to mount the `/dev/disk#` (from the previous command) to the `/Volumes/dademurphy_mounted/` mount point. This drive will now be available in the Finder or Terminal applications.
    - `-j` – Ignore the journal
    - `-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.

You can access this newly created mounted drive on `/Volumes/dademurphy_mounted/`.

```
$ mkdir /Volumes/dademurphy_image/

$ mkdir /Volumes/dademurphy_mounted/

$ ewfmount ~/FOR518/dademurphy.E01 /Volumes/dademurphy_image/

$ ln -s /Volumes/dademurphy_image/ewf1 ~/FOR518/dadeimage.dmg

$ hdiutil attach -nomount ~/FOR518/dadeimage.dmg

$ mount_hfs -j -o rdonly,noexec,noowners /dev/disk#
/Volumes/dademurphy_mounted/
```

10. **Sanity Check**
    - Using the Finder or the `cd` command in Terminal, access your newly created mounted volume. Use the `ls -l` command to view the contents in the terminal to (hopefully) view the OS X directory structure. You should see an account for '`zerocool`' in this directory.

```
$ cd /Volumes/dademurphy_mounted/Users/
$ ls -l
```