OSForensics Collection

By Catherine Nguyen and Matthew Calderon

12/9/2020

Table of Contents

Background information:

OSForensics

OSForensics is a paid service that can cost $79 to $799 depending on the monthly or yearly subscription. This service provides four main features, the first being an entire suite for a forensic investigation. This includes properties such as password recovery, extensive file support, extremely fast file searching, deleted evidence recovery, and many more. The second main feature is the ability to identify suspicious activity and files, by verifying/matching their hash values with MD5, SHA-1, and SHA-256 with a clear and intuitive collection of file analysis tools. The third feature of OSForensics is an all-in-one digital investigation manager that allows its users to create cases, generate reports, audit tracing, and storage device management. The final, and perhaps most unique feature is the bootable edition of OSForensics that contains all of the same properties as the professional edition described above. However, it also allows this program to be run on systems without a valid operating system.

ImageUSB

ImageUSB is free software that is included in the OSF collection of tools, with a simple UI and clear instructions. This service gives users the ability to create an exact bit-level copy of a USB drive and write said image to multiple other USBs concurrently. Not only can ImageUSB zero-out and reformat a USB drive extremely quickly, but it can preserve all slack and unused space leftover on a drive during the cloning process. The GUI for this software tracks the percentage progress of all actions and simplifies the overall data acquisition process rather than relying on command line commands.

OSFClone

OSFClone is a free utility whose main feature is to provide a self-booting solution to create, clone, and verify raw disk images. In addition to the normal raw disk format, this software provides options for AFF, Advanced Forensics Format and EWF, Witness Compression Format to meet the needs of every case situation. By default this tool uses “dd” to create disk images, but can also perform in “dc3dd” format for an increased level of reporting for progress and errors.

OSFMount

OSFMount is another free software in the OSF collection which allows you to mount local disk image files in Windows as a physical disk or a logical drive letter. It can be used to mount image files that were created using a disk cloning application such as OSFClone. Also, the following images are mounted as a virtual drive on Windows and can then be analyzed using OSForensics. Additionally, OSFMount can also be used to create RAM disks and mount CD/DVD-ROMs as RAM disks.

Statement of the problem (or project goal):

To explore the installation of forensic tools not covered in class. Test them to develop various methodologies provided by those tools. Implement the chosen tools and highlight their usage. In this case, we aim to apply our knowledge and broaden our scope of the tools available for an investigation.

Sensitive Information:

To simulate the criminal investigation scenario we had to create a suspicious drive that could be hiding criminal activity that may look invisible/confusing to the naked eye.

- DEAL.TXT
    - This file was given the hidden attribute to hide important data regarding the products and services available for a criminal deal.
- Don’t be Late.png
    - This file had its data hidden, by having its hex header changed from its .xlsx default: 50 4B 03 04 14 00 06 00 to a PNG header :89 50 4E 47 0D 0A 1A 0A. Therefore, it cannot be opened to reveal the hidden info without changing the header back.
- Gibberish.docx
    - This file contained numerous lines of “gibberish” or fake data hiding the true contents of the message that was invisible due to the white font.
- inconspicuous.txt
    - This file’s data containing the list of attendees for the deal was hidden by encrypting the file’s contents.

Proposed solution:

As a demonstration of our research, we've developed a scenario in which a suspect has techniques to hide sensitive information from investigators. We will cover in detail what the suspect has done, and then play the role of an investigator trying to find sensitive information by displaying the usage of these tools to examine its contents.

Implementation:

Hiding Information and Performing Data Acquisition:

We began the project by hiding information in files through processes such as hidden file attributes, changing hex headers to appear as a different format (excel to PNG), hiding messages with white/small font, and encrypting a text file as shown below:

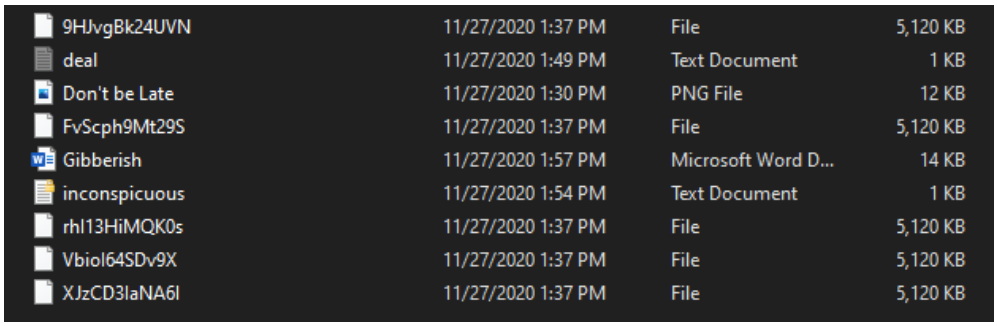

Figure 1: Criminal USB drive containing hidden data.

This was to simulate finding a suspect's drive in which we performed a data acquisition using ImageUSB to do the following:

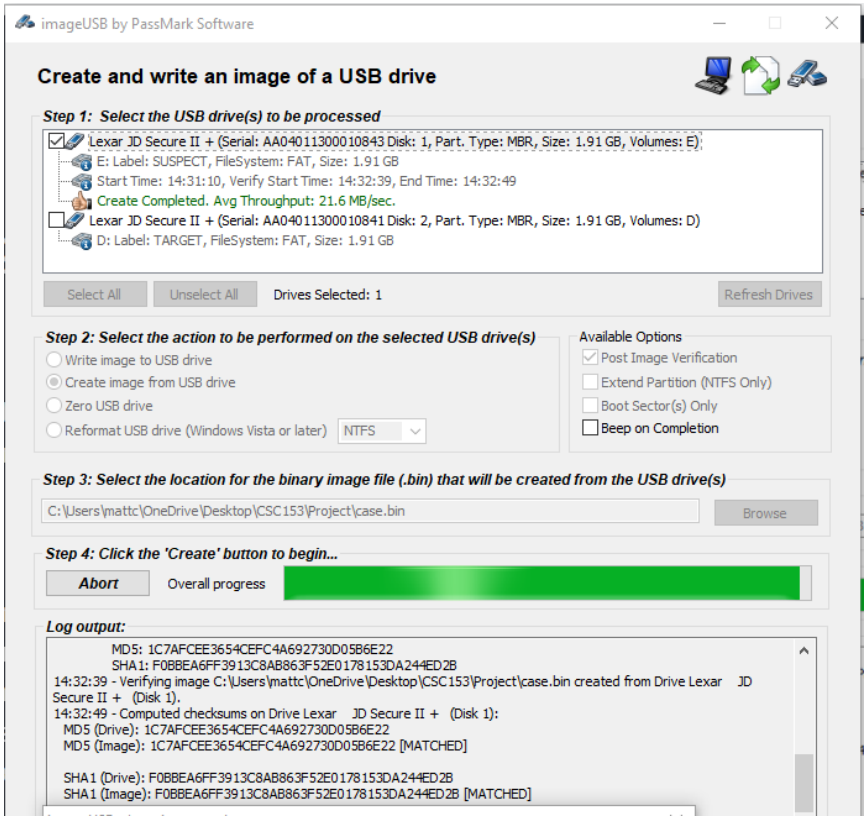

Figure 2: Created a "case.bin" image file from the suspect's drive and verified MD5 values.

OSFClone also provides an alternative to ImageUSB when completing the data acquisition. In this particular experiment, we utilized the USB flash drive installation, however, they offer instructions for CD/DVDs. ImageUSB is the preferred method to install this software and they listed clear instructions:

The installation of OSFClone requires an UFD which is at least 2 GB in size.

1. Download the *osfclone.zip* file and extract it to a directory of your choosing on your local hard disk drive.
   In this example, we extracted the files to a folder in the program files directory at *C:\Program Files (x86)\OSFClone*.
2. To reduce the likelihood of mistakes, remove all other USB drives or devices which you may have connected to your system.
3. Plug the UFD you'd like to use for booting OSFClone into your system and make a note of its drive letter. The UFD must be at least 2 GB in size for installation to be successful.
4. Start **ImageUSB** by double-clicking the *ImageUSB.exe* application.
5. From the **ImageUSB** window, first select the drive you would like to use by checking the box next to the appropriate drive letter.
6. Ensure that the *"Write to UFD"* radio button is selected in the next section. This option is selected by default in ImageUSB.
7. In the next section, click on the *"Browse"* button. Navigate to and open the file named *OSFClone.bin*.
8. Finally, click the *"Write to UFD"* button to install **OSFClone** to your USB Flash Drive.

OSFClone allows us to perform the same utilities as ImageUSB without the need of an OS, for this software is self-booting and runs independently of the host's OS. Therefore, in order to use this software you must change the BIOS settings of the host PC to run OSFClone once it is downloaded onto a thumb drive where you will see the following screen:

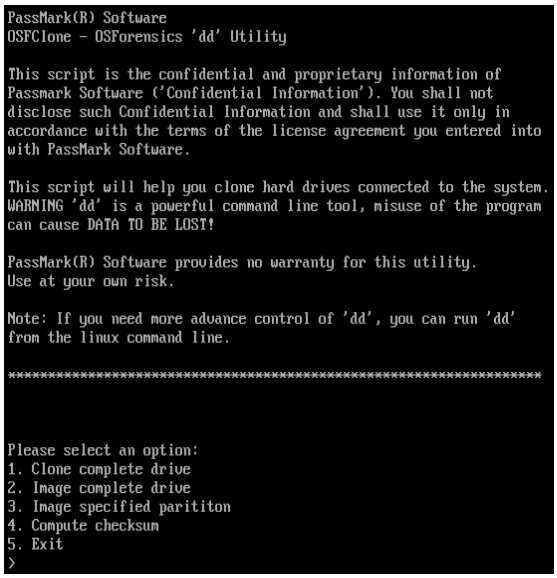

Figure 3: Actions provided by OSFClone upon reboot.

Here we can access all of the USB drives currently mounted to the host PC, perform data acquisitions, and verify the MD5 checksum values.

Now that the image file has been created I placed it in a google drive directory for my partner to further analyze and mount using the rest of the OSFCollection.

Mounting Target drive with the suspect's image file to begin file analysis:

For my portion of the project, I wanted to demonstrate the 'write image to USB drive' feature. To do so I first had to zero out my own target drive using the 'zero USB drive'

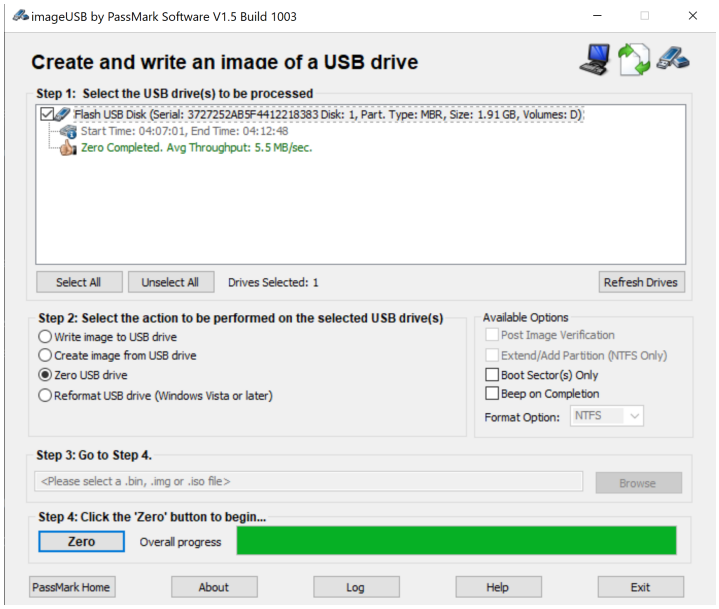

Figure 4: Zero out target USB.

While my target drive was being zeroed out. I went ahead and downloaded the image that Matthew had created earlier through ImageUSB from Google Drive. Because this drive was made by Matthew, he included notes about the hidden data so that I knew what to look for on my end.

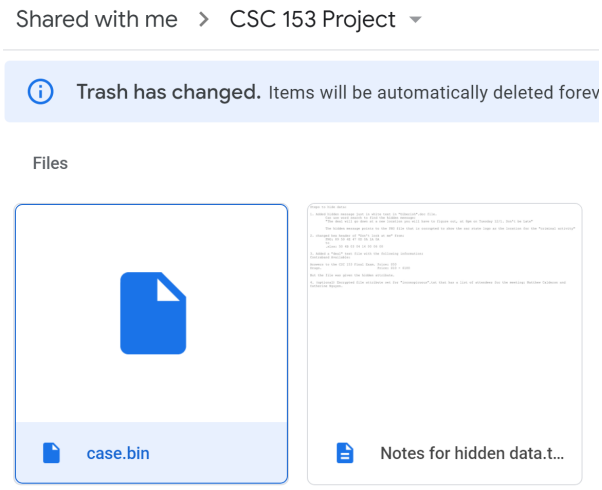

Figure 5: Downloading image file case.bin from Google Drive.

Once the USB drive's zero out and the case.bin download was complete, I performed the 'write the image to USB drive' feature which resulted in a failure because my target drive was smaller in byte size compared to case.bin.

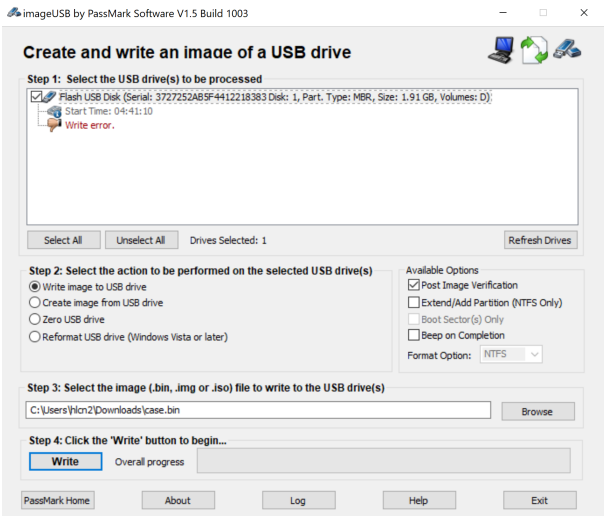

Figure 6: Failed to write the image to target USB.

Despite the failed attempt the contents on the target drive ‘SUSPECT (D:)’ seems to retain all the original files, but to avoid any errors. I proceeded with using case.bin for OSFMount.

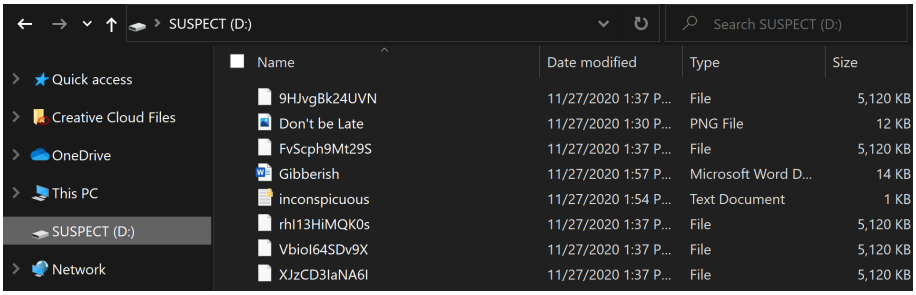

Figure 7: Files within the Target drive ’SUSPECT (D:)’ after writing ‘case.bin’ into it.

For the next portion of the project, we have to install OSFMount and these are the following requirements for installation:

- OS - 64-bit Windows 7 SP1, 8, 10, & Server 2008 & 2012 (32-bit Windows not supported).
- User must have administrative privileges.
- RAM - 128 MB the more the better.
- Disk Space - 10 MB for installation files

To install

- Use this https://www.osforensics.com/tools/mount-disk-images.html and press on the download link.
- Then following the .exe prompts to launch OSFMount.

Once OSFMount is launched, we can begin to mount ‘case.bin’ with the following steps:

- Press ‘mount new...’.
- Step 1: Select ‘Disk image file’.
    - Find the designated file, ‘case.bin’.
- Step 2: Select partitions
    - Mount the entire image.
- Step 3: Additional options for the mounted virtual disk

- Check read-only drive
- Logical Drive Emulation vs. Physical Drive Emulation
- Drive type
- Drive letter, ‘S:’ for SUSPECT.

After completing the steps above our image file ‘case.bin’ should be mounted to our device.

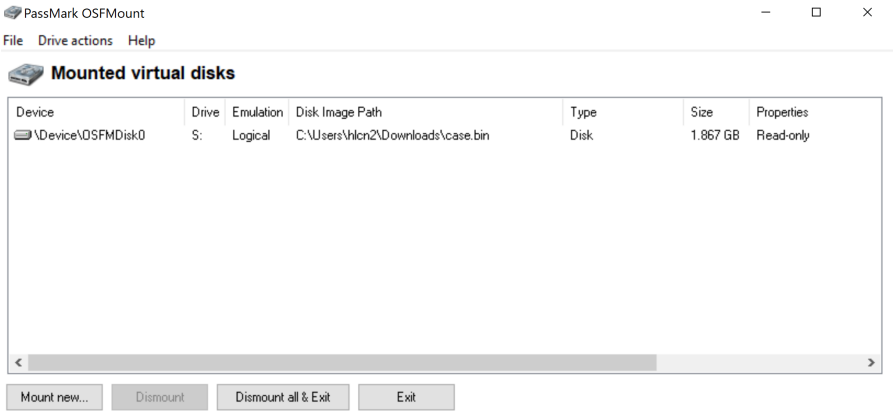

Figure 8: ‘case.bin’ successfully mounted to our Windows device as ‘SUSPECT (S:)’.

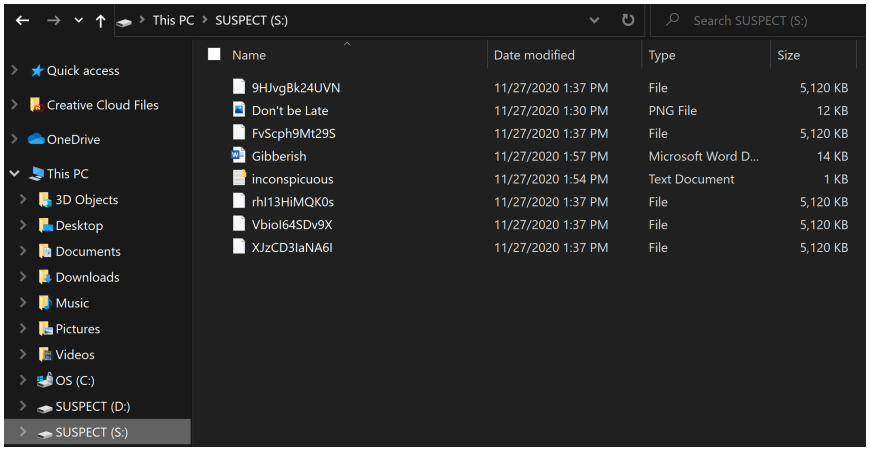

Figure 9: After mounting ‘case.bin’, these are the files.

OSForensics: Installation and creating a new case

In this part, I will go over some of the features available in OSForensics by installing it onto my Windows workstation and uncovering the hidden information in case.bin. To proceed I used this link to download OSForensics free trial, https://www.osforensics.com/download.html. Open the .exe file and launch OSForensics.

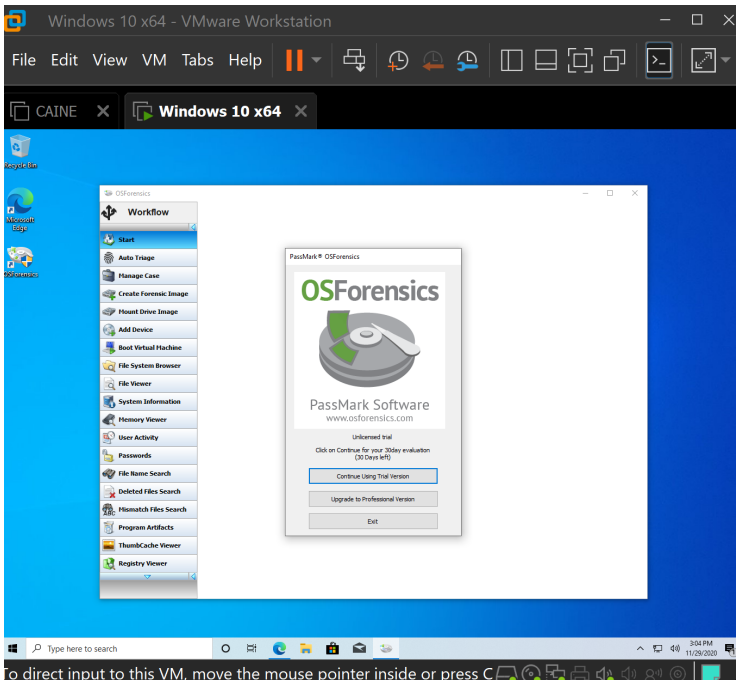

Figure 10: Launching OSForensics free trial on our workstation.

The next step is to mount ‘case.bin’ onto OSForensics. To do this click on ‘Mount Drive Image’, then the GUI for OSFmount will open up. Then press on ‘Mount new’ select the desired file ‘case.bin’ and mount the whole image. When that is complete you should be able to see that it has been mounted and ready for examination.

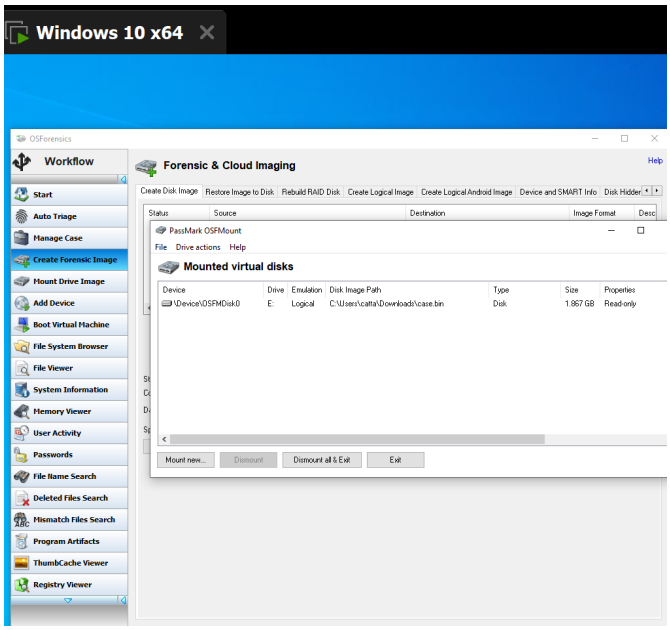

Figure 11: Mounting ‘case.bin’ onto OSForenics.

Prior to examining the files on ‘case.bin’ we first have to create a new case by clicking on ‘Manage Case’ -> ‘New Case’ and enter the information for the case. After that is complete we can then look at all the suspicious activity on ‘case.bin’.

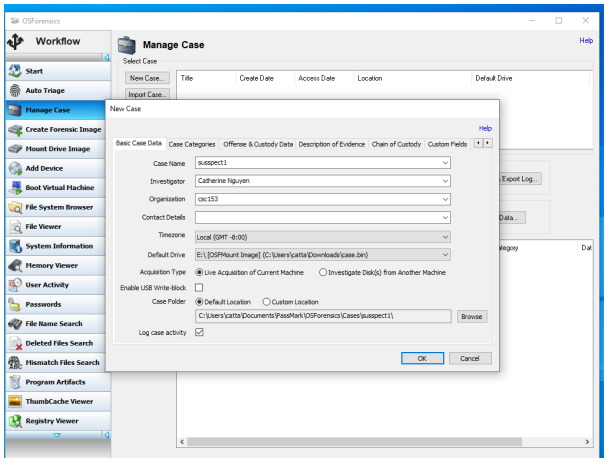

Figure 12: Creating a new case on OSForensics.

Result:

As a result of mounting ‘case.bin’, we can now see more information about the contents in the File System Browser. Upon first look, all files are present while DEAL.TXT is visible and inconspicuous.txt highlighted in yellow because it is encrypted. For more information about the files you can look at the Attributes column and you’ll see that DEAL.TXT has an ‘H’ denoting hidden and ‘E’ denoted encrypted.

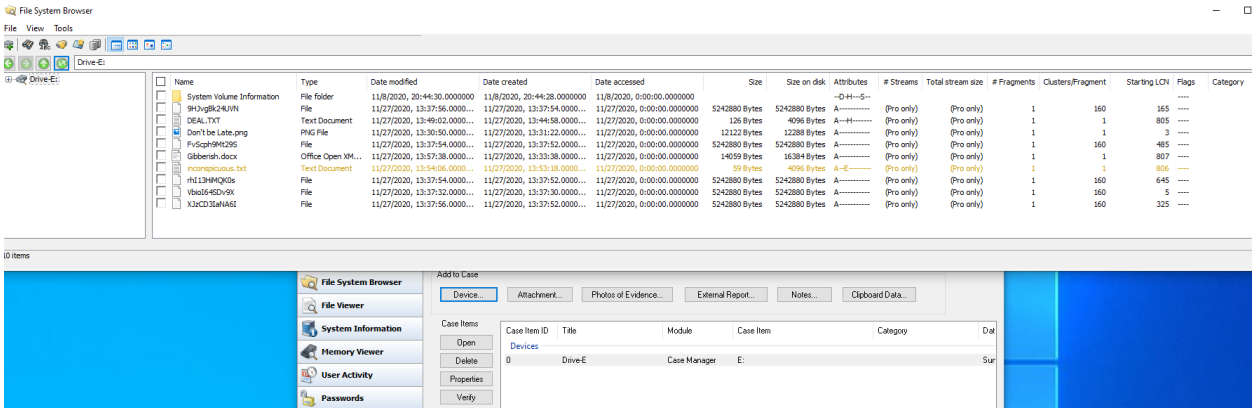

Figure 13: Opening case.bin in File System Browser.

Next is to open each file through File System Browser and you can see that when opening DEAL.TXT, which was hidden before, shows that the price of exam keys is $50 and drugs range from $10-$100.

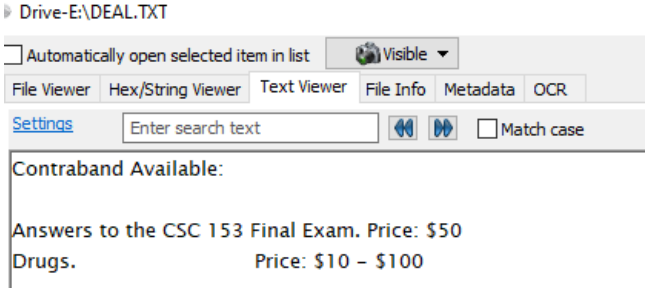

Figure 14: Contents of DEAL.TXT

Don’t be Late.png we were not able to open the file because the extension has been changed from .xlsx to a .png. So when you try to open the file it will say it is not supported. There are other tools out there to make it viewable, but for this demonstration, we will just show the ability to see hex headers in OSForensics.

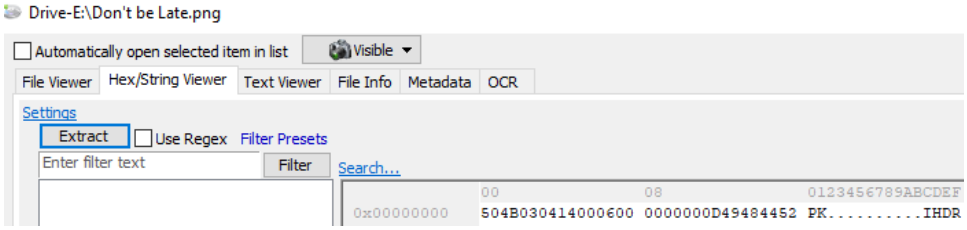

Figure 15: Hex header for Don’t be Late.png 50 4B 03 04 14 00 06 00 IS .xlsx file.

Next is Gibberish.docx. If we were to open this file in WordPad or Microsoft Word, you will notice that there is white space in between the two paragraphs of gibberish. However, when you open it through OSForensics we will be able to see the suspicious note in this file. I have highlighted the hidden information in this .docx file.

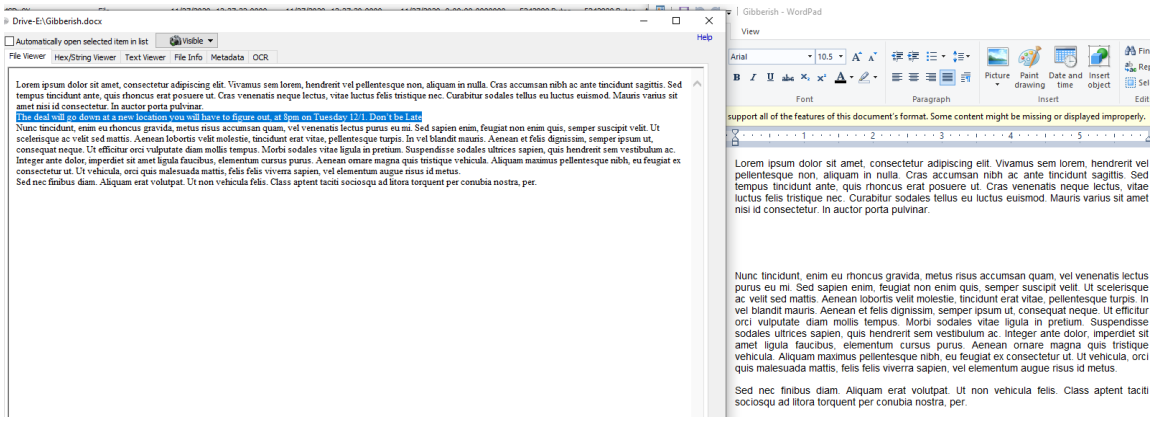

Figure 16: Gibberish.docx opened in OSForensics (left) and WordPad (right).

The last file is inconspicuous.txt, which was an optional feature we wanted to showcase, which is the encryption and decryption of a file. Ultimately, I was not able to decrypt the text file, but OSForensics does have the ability to do so.

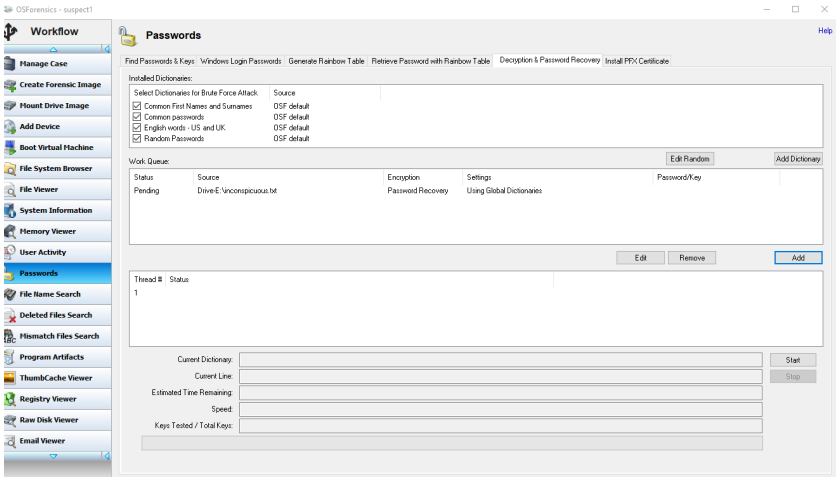

Figure 17: Passwords.

Conclusion:

In conclusion, this collection of tools proved to be extremely useful in simplifying and clarifying the criminal forensic investigation process. From intuitive GUI's to fast processes such as zeroing out a 2GB drive in four minutes, the OSFcollection is definitely recommended for data acquisitions, case management, and file analysis. While the subscription price for OSForensics is quite pricey, the rest of the tools and functionalities they offer is well worth the price in the long run.

Reference:

Documentation/Download Links:

OSForensics: https://www.osforensics.com/download.html

ImageUSB: https://www.osforensics.com/tools/write-usb-images.html

OSFClone: https://www.osforensics.com/tools/create-disk-images.html

OSFMount: https://www.osforensics.com/tools/mount-disk-images.html

Questions/Concerns:

Catherine Nguyen: catherinenguyen@gmail.com

Matthew Calderon: matthewcalderon@csus.edu

Work Distribution:

Name	Role
Catherine Nguyen	• **Use OSFmount to mount the suspect's image file.** • **Use OSForensics to find the hidden information for the investigation.** • **Contribute to reports, videos, and demonstrations.**
Matthew Calderon	• **Create a suspect drive and hide information with dummy/encrypted files.** • **Perform data Acquisition** • **Contribute to reports, videos, and demonstrations.**