Autopsy Lab

Your Name Here

Date

Mr. Ramsey – Computer Forensics & Investigations

East Tennessee State University

## Purpose

The purpose of this lab exercise is to use Autopsy/Sleuth Kit within the SIFT workstation to examine a disk image and extract evidence from it.

## Materials

* Installation of the SIFT workstation on VMWare Player ver. 5.0.0
* Downloaded copy of a USB disk image (*usbkey.img*)
* Lab instructions (Windows Forensics Challenge)

## Procedure

#### Initial Steps

Initially, the SIFT Workstation machine was run in VMWare Player. The file usbkey.img was than copied to the */cases/usbkey/* directory. In a terminal window, the command file *usbkey.img* was then run. This provided information about the image, including the fact that the file system is MSDOS5.0.

Autopsy was then run in the SIFT Workstation. A new case was created and called “usbkey.” Then, **Add Host** was selected to add a host (ensuring that the time zone was set to EST5EDT). The image file was then added to the case (**Type** was set to *Partition*). After making note of the file system type (FAT16), the ADD button was clicked to finish adding the image to the case.

#### Global Analysis

The overall analysis of the image was displayed by clicking Analyze -> Image Details. From this display it was possible to confirm that the file system in use was indeed FAT16. Each cluster consists of one sector, which simplified calculating file size to disk space consumption.

The File Name Layer was then examined. A first impression was that one file *\_IMMY~1.DOC* had been deleted. Investigator also noted that one of the columns in the report was entitled “Metadata.” Further examination of Autopsy help files revealed that if there is a value in this column, it represents a pointer from the file name structure to the file’s metadata. The number in the column represents the address of the metadata structure. Clicking this link will display the file’s metadata in the bottom pane. If there is not valid metadata, nothing will be displayed there. If the address of the metadata structure has been overwritten, the entry will read “*realloc*.”

#### File Analysis

Suspect files were then analyzed. The first file, *\_IMMY.DOC*, was determined to be a MS (or Open Office) document file. Using the Export button in Autopsy, the file was exported and saved as evidence in the case file. Then, Open Office was successfully used to open the file and examine its contents (see Fig. 1).

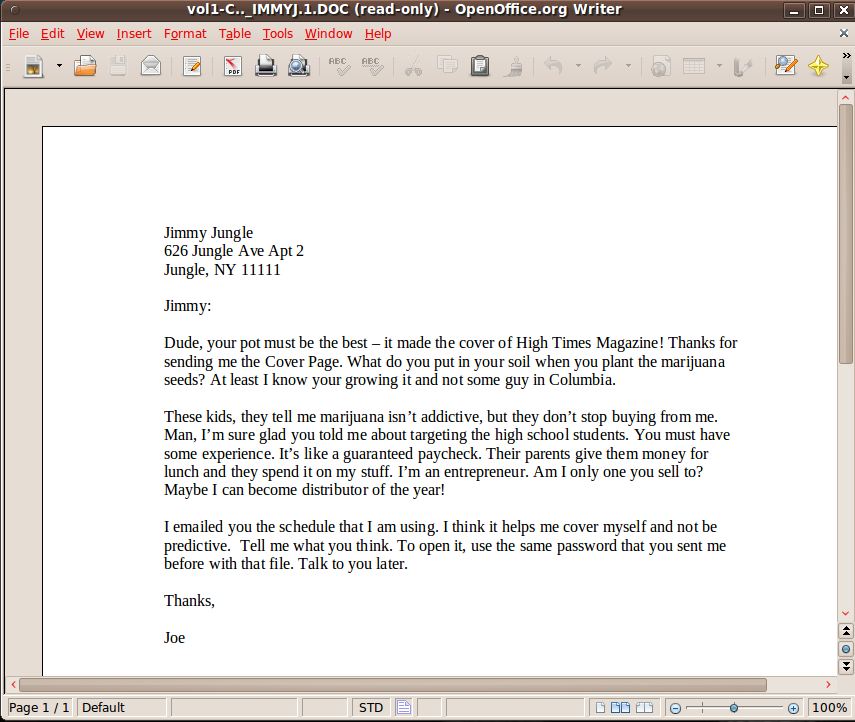


Figure : Recovered word processor document

The file makes mention of an image file allegedly sent from one suspect to another, a schedule utilized by an alleged marijuana dealer for his sales and delivery route to various local high schools, and the existence of a password. Then, an attempt was made to examine the file *cover page.jpg*. On the initial attempt, it was discovered that the file was in some way corrupted. Further analysis was then deferred.

A third file of interest, *Scheduled Visits.exe*, was then examined. After clicking on the file name link, Autopsy indicates that its file type is “empty (ZIP archive data…” -- see Fig. 2).

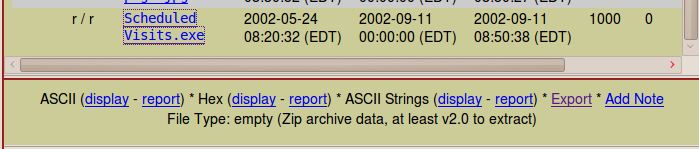


Figure : A .exe file identified as a .zip archive

It is somewhat unusual for a ZIP archive to have an *.exe* file name extension. Displaying the file in HEX format revealed that the first few bytes of the file contain the characters “PK,” suggesting that the file is a *pkzip* archive. Extraction of the file was then attempted by copying it to the case file folder and changing the extension to *.zip*. An error message was displayed advising that the correct end of directory signature was not found and that the file was not a proper zip file. In order to determine what the correct end of file signature was, two test ZIP files (of identical size) were created and examined. Examining both test files in the Bless Hex Editor revealed that each had a signature of *0x50 0x4B 0x05 0x0*6 (see Fig. 3).

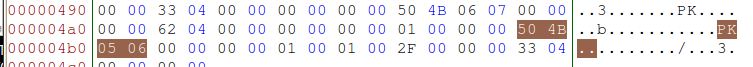


Figure : End of file signature in test .zip file

The first two bytes represent the ASCII characters “P” and “K,” respectively. Investigator then searched the file for occurrences of “PK.” Three hits were discovered in Sector 487 (the beginning sector of the suspect file) and Sector 491. Sector 491 was examined by clicking on the “Hex” link and it was discovered that the EOF signature was present. Investigator then extracted Sectors 487-491 and saved them as file name “sect487-491.raw.” When an attempt was made to unzip the file, Investigator was prompted for a password. This indicated that the file had been successfully “carved” out of the image file. The need for a password also indicated that further investigation would be required before and if the file could successfully be unzipped.

Investigator then examined the file “cover page.jpg” again. Selecting the “Metadata” tab and entering “8”in the “Dir Entry Number:” field displayed the metadata for the file (alternatively, clicking on the “8” link in the Metadata column of the “cover page.jpg” row in the “File Analysis” tab could have been done to accomplish identical results). It was then noted that, though the file size was reported to be 15585 bytes, the file was shown to occupy a single sector, sector 834.



Figure : Metadata for file *cover page.jpg*

Selecting the sector, by clicking on its link, revealed that Sector 834 was actually unallocated space. The number of sectors the file should have occupied was calculated to be 31, which matches up to an entry in the “FAT Contents” section of the image details (see Fig. 5).

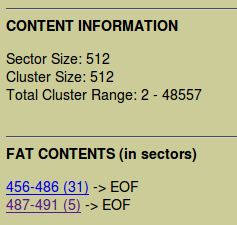


Figure : FAT Contents of the image file

Though the “31” appearing in the FAT Contents may have been coincidental, Investigator elected to carve out the 31 sectors starting with Sector 456 and discovered that they indeed constituted the missing .jpg image file. The file was then exported to the case folder and displayed (see Fig 6).



Figure : Recovered .jpg image file

Checking to see if any further information was concealed within the file, Investigator searched for ASCII strings contained within by clicking the display link next to “ASCII Strings” at the top of the display (see Fig. 7). At the bottom of the list, the string “pw=goodtimes” was found. Using “goodtimes” as for the key to unzip the “sect487-491.raw” file worked and produced the file, “Scheduled Visits.xls.” Opening that file in Open Office revealed another incriminating piece of evidence against the suspect (see Fig. 8).



Figure : Displaying ASCII strings contained within .jpg file

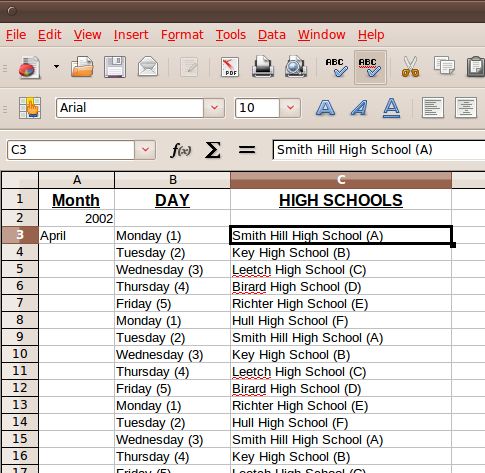


Figure : Spreadsheet detailing suspect's visits to area schools.   
Narcs would love to have this.

## Observations

This lab demonstrated low-level examination of digital data recovered from a hypothetical suspect. The files that were examined had been deleted and/or manipulated by the suspect in order to hide their content from an investigator and thus hide the subject’s involvement in criminal activity (ultimately found to be sale of marijuana at local schools). In addition to having to perform byte-level examination of files, it the investigator also had to piece together information obtained and how it could be used to recover more evidence. Oftentimes in a criminal investigation, the value of recovered evidence can only be determined by its relationship with other evidence and, of course, to the alleged crime.

In this case, the first recovered file, the word processor document file, provided potentially incriminating evidence in the form of an admission to criminal activity. It also made mention of another file and the existence of a password. From there, the investigator, using the Autopsy forensics tool, teased information out of the image file, again sometimes operating at the byte level to recover obfuscated data and ultimately recovered the information the suspect had sought to hide.

It was convenient that, while some data manipulation had taken place to hide incriminating information, the three files needed to “crack the case” were the only ones present on the recovered drive. When one takes into consideration that USB sticks are typically two to eight GB in size, and often used to store far more than one’s dope-peddling records, then the knowledge, skills, and persistence required by real-world scenarios becomes evident. A similar examination of such a drive, even with knowledge of the suspected offense, would have been far more involved. However, as digital technology continues to become more pervasive in modern society, its misuse will also continue to become more common.