

1 Introduction

1.01 Summary Background of Case

I have been instructed to give my expert opinion on, possible breach of network of University of Grand Fenwick's computer science department. During pandemic, the IT manager found a USB plugged in to a computer in one of the faculty's office. The faculty member has denied any knowledge of the USB and claimed that it was not there when he left the office. The university administration has assigned the task to investigate the matter, the contents of the USB, and find if there are any other system breached done.

2 My investigation of the facts

2.01 Assumed Facts

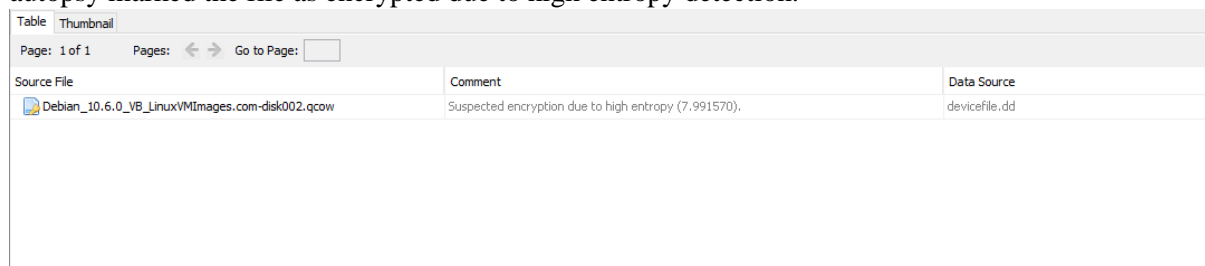
One assumes that on accepting my instructions the information provided to me by B. Kokintz, University of Grand Fenwick, Dept. of Computer Science is correct.

It is also assumed that the information provided by the IT manager of the department and the faculty staff member is accurate and sound at the time of my investigation.

2.02 Enquiries / investigation into the facts by the expert

For my investigation, I used a number of software's. The first software used was **Autopsy** version 4.13.0 to perform automated analysis of the provided image. After running the autopsy and getting results from the automated analysis I started searching for the evidence that may answer the above raised issues. Another software that I used was **DB Browser for SQLite** version 3.12.1 to view the ".sqlite" database files found during the analysis of the USB image. Another tool I used was a WinHex, which is used to view the hex data of file. WinHex was used to analyse the contents of suspicious files.

Once Autopsy was started, a new case was created for the analysis. The case details were entered in all the fields. After entering the case details, all the ingest modules were selected to run on the forensic image to get maximum results from the automated analysis. Once the automated analysis was done, large number of deleted files were recovered, including images, documents, windows dll binaries and other files. After getting the files, I started checking the files manually to look for files that may provide an evidence which can be used to link the USB drive to the case. After going through number of files, a file with name "**Debian_10.6.0_VB_LinuxVMImages.com-disk002.qcow**" was found and autopsy marked the file as encrypted due to high entropy detection.



Source File	Comment	Data Source
Debian_10.6.0_VB_LinuxVMImages.com-disk002.qcow	Suspected encryption due to high entropy (7,991570).	devicefile.dd

Figure 1: Shows the image of encrypted file found from the automated analysis.

After doing some research on the file extension and reading articles about it, I found that the qcow [2], file format or disk images are used by QEMU [1], a hosted virtual machine monitor. It is a format used

for taking images of virtual machines or disks associated with specific guest operating systems. The main purpose of using this file format is that the files grow in size as data is added to the image [3].

From this, I assumed that the attacker might have taken a disk image of the system running Debian Linux on it and most specifically the disk image of the system the USB was found attached. Another assumption I can make here is that the attacker had the disk image from his own virtual machine and might have tried to run that image from the staff computer to get a backdoor access to the system so that he can further escalate into the network. Another evidence that supported my assumption was that a number of jpeg files, which I exported and on seeing the contents of those files I found default wallpapers of Debian Based Linux operating system.

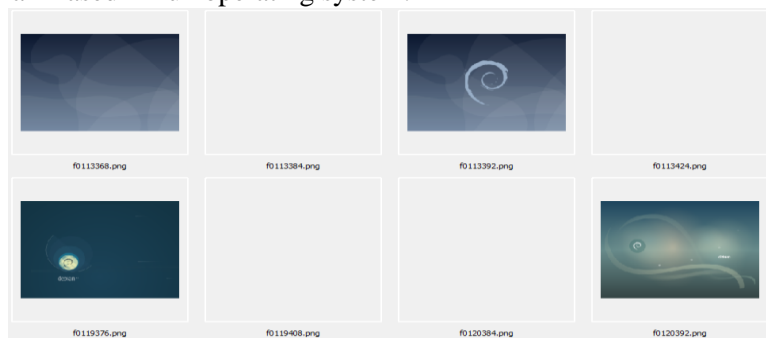


Figure 2: Shows the wallpapers of Debian Linux found during search on Images.

A **vmdk (Virtual Machine Disk Format)** [10] file was also found which is the disk image of the virtual machine created using Oracle VirtualBox. From this I assumed that a virtual machine of Debian Linux may have been used by the suspect.

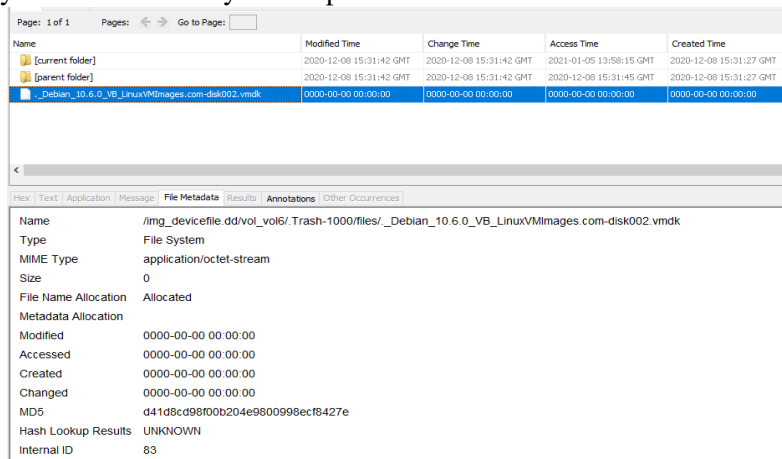


Figure 3: Shows the vmdk file of Debian Linux VM found during analysis.

Further going through the artifacts found by autopsy, a file with pdf extension was found. I exported the file to the local directory in my forensic workstation and by viewing the contents of file using Adobe Reader I found that the file contained the course guideline of “Cloud and Mobile Forensic”, a subject that staff member whose computer was found with the attached USB might be teaching. From this an assumption of possible data theft can be made.

Name	Modified Time	Change Time	Access Time	Created Time	Size
6104COMP Cloud and Mobile Forensics - Coursework 1.pdf	2021-02-11 10:36:03 GMT	2021-02-11 10:36:04 GMT	2021-02-11 10:42:18 GMT	2021-02-11 10:42:18 GMT	167156

Figure 4: Show the PDF file found during analysis and indicating data leak

The attacker who left the USB might have tried to steal the data from the computer by copying it on the USB and then later remove it. Another file with gif extension was also found. After exporting the file to local directory and opening it in image viewer, I found the Grand Fenwick University's flag in the gif. That evidence supported my assumption of data extraction from the computer.

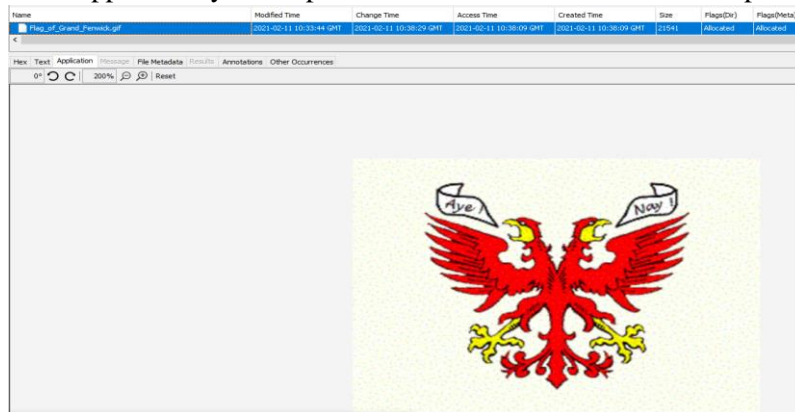


Figure5: Show the logo of University found in the .gif file during analysis.

After doing some more analysis, I came across a text file which had an email and most probably a password in it. This can be the attacker email and password, I assumed this by looking at the domain of the email address. The Proton mail is a very secure email client used mostly by people with malicious intents because the encryption used by proton is very high grade and they keep their users anonymous. Number of web pages were also found during the analysis, which indicate that our suspect was into using proton mail. Also, from the host name in the email, no link was found to a person, thus making the assumption stronger that email and password found in the text file belong to the attacker, because no connection of the email was found to a person, thus making the attacker anonymous as he used random hostname and a very secure email client.

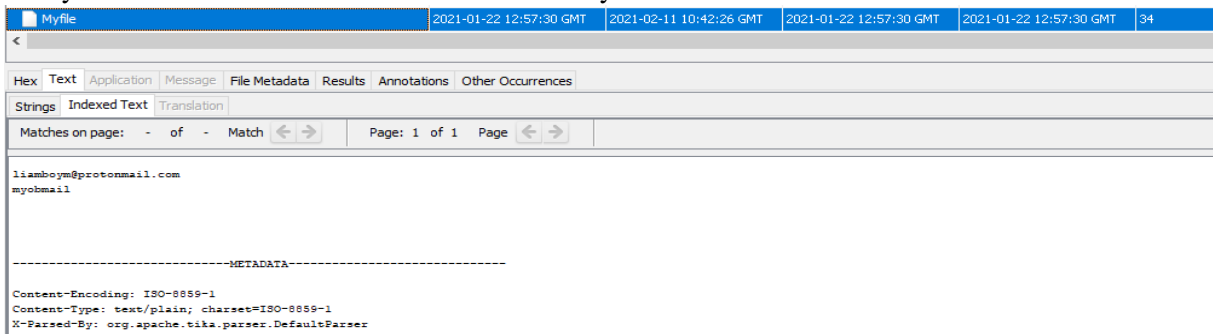


Figure 6: Shows the email and possibly password found in the text file.

Number of database files were also found during the analysis. I exported the files into local directory and used SQL Browser [8] to load the contents of the files. On opening the files in **SQL Browser** [8], I found them corrupted, which means that the attacker might have corrupted the data base files making them unreadable. A web page was also found showing the release history of SQLite [9] which also support the assumption that the suspect was into using this program and had manipulated with the database files we found during the investigation of the forensic image of the USB.

Name	Modified Time	Change Time	Access Time	Created Time	Size
f1088016.sqlite	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	303104
f1088640.sqlite	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	45056
f1088840.sqlite	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	24576
f1119096.sqlite	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	57344
f1186200.sqlite	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	32768
f1195992.sqlite	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	249856

Figure 7: Shows the database files found during examination

On analysing the deleted executable files, an executable file was found which was hardcoded with a number of IP Address ranges, which indicate that this might be some sort of network scanner or the network exploiter. Network scanner takes IP address or a complete range of IP addresses and then starts probing the IP addresses to look for open ports and service running on those ports. Once the open ports and service are identified, the attacker may try to exploit any vulnerable service running on found open ports which may give attacker the access of the system. A very solid proof that indicates that the security of university network was breached, or the suspect may have tried to breach the security.

Name	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known
f1052200.exe	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	5556224	Unallocated	Unallocated	Unknown

Hex	Text	Application	Message	File Metadata	Results	Annotations	Other Occurrences
Strings Indexed Text Translation							
Matches on page: - of - Match < > Page: 1 of 14 Page < >							
<pre> 213.169.061.000-213.169.061.255 213.169.106.000-213.169.106.255 213.169.107.000-213.169.107.255 213.169.216.216-213.169.216.255 213.169.239.150-213.169.239.159 213.169.239.032-213.169.239.039 213.169.280.000-213.169.281.255 213.170.032.000-213.170.062.255 213.170.129.160-213.170.129.167 213.170.129.184-213.170.129.191 213.170.130.120-213.170.130.127 213.170.130.128-213.170.130.135 213.170.130.176-213.170.130.179 213.170.130.190-213.170.130.193 213.170.130.200-213.170.130.203 213.170.131.112-213.170.131.119 213.170.131.160-213.170.131.167 213.170.131.192-213.170.131.207 213.170.136.232-213.170.136.239 213.170.137.000-213.170.137.007 213.170.137.032-213.170.137.039 213.170.137.188-213.170.137.191 213.170.137.200-213.170.137.203 213.170.138.208-213.170.138.215 213.170.138.224-213.170.138.227 213.170.139.040-213.170.139.049 213.170.139.120-213.170.139.133 213.170.139.128-213.170.139.255 213.170.140.028-213.170.140.031 213.170.140.044-213.170.140.047 213.170.140.144-213.170.140.151 213.170.140.200-213.170.140.207 213.170.141.000-213.170.141.011 213.170.142.000-213.170.142.003 213.170.142.012-213.170.142.019 213.170.142.016-213.170.142.019 </pre>							

Figure 8: Show the IP address found hardcoded in the executable file found in the deleted files.

Another suspicious file that was found during the analysis was named “**sfk.doc**”. From the extension of the file one can assume that this is a document file, but after exporting it to local drive and analysing the contents of file in **WinHex**, I found that it was an executable file.

```

sfk.doc
Offset  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  ANSI ASCII
00000000  5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00 MZP  yy
00000010  B8 00 00 00 00 00 00 40 00 1A 00 00 00 00 00  .  @
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .
00000030  00 00 00 00 00 00 00 00 00 00 00 00 01 00 00  °  ' i!, Li!
00000040  BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90  This program mus
00000050  54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73  t be run under W
00000060  74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57  in32 $?
00000070  69 6E 33 32 0D 0A 24 37 00 00 00 00 00 00 00 00
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000100  50 45 00 00 4C 01 08 00 88 1F 05 57 00 00 00 00 PE L  ^ W
00000110  00 00 00 00 E0 00 8F 81 0B 01 02 19 00 04 01 00  à
00000120  00 70 02 00 00 00 00 00 DC 17 01 00 00 10 00 00  p  Ü
00000130  00 20 01 00 00 00 40 00 00 10 00 00 00 02 00 00  @
00000140  05 00 00 00 06 00 00 00 05 00 00 00 00 00 00 00  "œ" @
00000150  00 10 04 00 00 04 00 00 22 F8 80 01 02 00 40 81  @
00000160  00 00 10 00 00 40 00 00 00 00 10 00 00 10 00 00  @
00000170  00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00
00000180  00 90 01 00 04 0E 00 00 00 C0 01 00 14 4F 02 00  à  O
00000190  00 00 00 00 00 00 00 00 28 8A 80 01 78 16 00 00  (ŠE x
000001A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001C0  00 B0 01 00 18 00 00 00 00 00 00 00 00 00 00 00  °
000001D0  00 00 00 00 00 00 00 00 04 93 01 00 14 02 00 00  "
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001F0  00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00  .text
00000200  44 F2 00 00 00 10 00 00 00 F4 00 00 00 04 00 00  ó
00000210  00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60  .
00000220  2E 69 74 65 78 74 00 00 64 0F 00 00 00 10 01 00  .itext d
00000230  00 10 00 00 00 F8 00 00 00 00 00 00 00 00 00 00  @
00000240  00 00 00 00 20 00 00 60 2E 64 61 74 61 00 00 00  .data
00000250  88 0C 00 00 00 20 01 00 00 0E 00 00 00 08 01 00  ^
00000260  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 C0  @  à
00000270  2E 62 73 73 00 00 00 00 BC 56 00 00 00 30 01 00  .bss 4W 0

```

Figure 9: Show the hex contents of sfk.doc file found to be a keylogger impersonating as document file.

To further check the file, I uploaded it to VirusTotal [4] to see if it is detected by antivirus or not. From the VirusTotal [4] results, the file was marked as malicious and detected as keylogger [5]. The keyloggers [5] are used to capture the keystroke user enter into the computer, and then send them to the attacker through email. This supports our previous assumptions and from the email address we found, we can conclude that the attacker injected the system with the keylogger to capture key strokes, and the email password was used as recipient to send the capture key logs to the attacker by means of email.

DETECTION	DETAILS	RELATIONS	BEHAVIOR	COMMUNITY
Allbaba		ⓘ RiskWare:Win32/Spyrix.f96b7963		Comodo ⓘ Fis.noname@O
DrWeb		ⓘ Program.Spyrix.6		ESET-NOD32 ⓘ A Variant Of Win32/KeyLogger.Spyrix.N
Fortinet		ⓘ Riskware/Spyrix		Gridinsoft ⓘ Spy.Keylogger.ddlc
Kaspersky		ⓘ UDS:DangerousObject.Multi.Generic		Malwarebytes ⓘ RiskWare.SpyrixKeylogger
McAfee		ⓘ Artemis:F2D8FF70505B		McAfee-GW-Edition ⓘ Artemis
Qihoo-360		ⓘ Win32/Trojan.9ad		Acronis ⓘ Undetected
Ad-Aware		✔ Undetected		AhnLab-V3 ✔ Undetected

Figure 10: Shows the result of sfk.doc after scanning it on VirusTotal.

Further looking for more evidence I ran keyword search on the name I found on the email to see if it reveals any other files that may contain the same name, but no interesting results were found from the keyword search.

By analysing delete files, I found two windows registry files [6],[7]. The registry files contained registry keys and other data. By analysing the registry key, I found that there was some manipulation with registry keys, specifically registry keys of windows hypervisor and the boot.

2 Future Plan

In future if a case like this happens and the investigator had to follow the guidelines, then this section can provide a complete plan for future investigation in cases like this and others also. The investigator can follow the same procedure as discussed below and conduct a complete forensic investigation.

3.0.1 Data Collection

An important part of any forensics investigation is identifying the computer to be analysed. Securing the computer is important to prevent tampering, but the area around the computer is also important as well. The evidence collecting phase of the investigation may be very complex with many different types of physical evidence requiring collection. Once the physical evidence is collected, check the computer for power source, if power source is connected and the computer is turned on, then it is very important to collect all volatile evidence from the computer before it gets remove by power failure or someone may try to tamper the evidence remotely. After all evidence has been collected and stored for keeping, the evidence that cannot be collected should be processed.

In case of computers and other digital devices, the digital evidence cannot be seized then this type of evidence must be image on site.

Ensure all media that have been used to receive evidence have been properly sanitized and prepared for use. An appropriate tool must be used to take forensic image of the computer system based on the operating system. If the computer is in running state, then it is important to dump the memory of the system as well as other information like process info, running services, established network connections, etc.

3.0.2 Examination

It is imperative that all work media and hard drives used in the examination process must be sanitized and certified or verified as clean. This eliminates the possibility of data corruption due to residual information from previous investigations be processed. Analysis, research or any investigative work must never be performed on the actual digital evidence or forensic image. When forensic images of digital evidence are made make sure to make a working of the copy at the same time and verify both copies by matching their hash values. The image can be mounted under Linux operating system with /dev/loopback device and be processed as a normal file system. A predefined work structure should be created on the working case drive. This structure provides directories for each category of evidence file and process state. Autopsy automatically categorizes files by Case and Host.

Processing the raw digital evidence is one of the tasks that can be mostly automated. The type of forensic image created of a physical drive will dictate additional processing. The following steps are presented to organize the processing to maximize the automated processing capabilities. All of the below data components, except allocated data, should be processed with Foremost and strings. Foremost will potentially identify any files and strings, which may provide any type of clues including email addresses, phone numbers, file names, passwords, URL addresses, IP addresses, etc. Foremost and strings are the Linux utilities used for forensics. The autopsy on the other hand automatically scans all the data from the image and automatically categorize it providing ease of use to the user.

Once the image of the drives has been taken and working copies made. The forensic analyst can run autopsy on the computer, open a new case and feed all the information related to that case. Once everything is set up, the evidence file is added into the autopsy and ingest modules are selected by default. The user can modify the ingest module selection based on its usage and then run the modules on them. The autopsy will run the selected modules on the evidence image, and once all the modules have run successfully, the artifacts collected by the autopsy will be shown in the left pane of the autopsy window, and the investigator can easily search through the artifacts for potential evidence.

All the allocated data from a partition or disk should be extracted into local folder on the forensic workstation for further processing. The contents of allocated data will generate all the files that comprise the files on that disk. These files should be manually analyzed using other tools like WinHex, etc.

During normal operations computer files are allocated and deleted. File systems are not usually very efficient at reusing previously deleted space. Normally, file systems are never completely utilized. The unused or unallocated file space can contain a wealth of information for the forensic investigator. Autopsy automatically goes through all of the unallocated space in the disk or image and recover deleted files. Those deleted files should also be exported externally and examined using other tools required for the analysis of the files.

In case of forensic investigation of a computer running Unix based operating system, then the swap is also very important for the forensic investigator. Swap space is an area allocated on the disk to which the system periodically copies memory contents. Swap space is used frequently when a user uses programs that use more memory than is available. The system copies some of the program to swap space in order to free up memory for the program currently needing more memory. Swap space can contain documents, pictures, passwords, and programs that a user previously used on the computer. Autopsy provides the functionality to recover swap space files from the disk or image. The swap space files once recovered should be exported locally to the case directory and then analyzed using the recommended tools.

In case of Windows based operating systems, Files are allocated in Windows by blocks, 1K or 4K or greater, depending upon the Operating Systems and size of the hard drive. The actual size and contents of almost every file are not an exact multiple of the file systems' block factor. That means that as files are deleted and overwritten, remnants of previous files using the same physical area on the disk may exist. This data can be extracted and analyzed, possibly providing leads or crucial evidence. The pagefile.sys is the Windows' swap file. This file can contain previous or current programs being executed and the data those programs are using, this could be complete graphics files or word documents. The data in this file is unstructured as compared to a file system. Anything that a program uses or has used may be contained in the pagefile.sys. This file should be processed with Foremost and strings for potential fragments of information. The hiberfile.sys file is used when a computer is put into hibernation mode. The content of memory and additional process information is written to this file. When the computer is restarted the contents of the file is read back into memory and program executions resume from the point at which they were stopped. As with the pagefile.sys this file can contain almost anything. This file also should be processed with Foremost and strings for potential fragments of information. System memory, if saved during collection on a live system, can contain passwords, user-ids, web site address, document, graphic images and program contents running on the computer at the time the image was taken. This is extremely valuable when working network intrusion and cases involving hacking. System memory, if saved during collection on a live system, can contain passwords, user-ids, web site address, document, graphic images and program contents running on the

computer at the time image was taken. This is extremely valuable when working network intrusion and cases involving hacking.

3.0.3 Refining Digital Evidence

This step involves processing the digital evidence collected during the investigation process to further refine the evidence and making it more presentable in the court of law if required.

Of then composite files are used to hide documents, passwords, pictures and other illegal content. A composite file is a file that contains other files, normally considering an archive file. Sometimes these archives are even password protected. All archive files on the system must be un-archived or decompressed so the contents can be inspected. The normal approach would be to create a directory with the name of the archive and un-archive the content into that directory for inspection.

Another problem that is faced during forensic examinations is identifying and processing encrypted and password protected files. This is probably one of the hardest and most-time consuming tasks in digital evidence examination, especially when the suspect does not want to cooperate. In cases of this nature, the investigator must rely upon some non-traditional means of gaining access to encrypted files. Some of these methods involves the use of password cracking programs and brute force techniques; seldom are these efforts successful. The investigator must attempt to locate or determine the password or encryption key by other means. Suspects and most computer users often hide or transcribe their passwords or phrases in obscure places. The investigator's job is to discover these locations if possible or build the case without this data.

Electronic Mail files often contain direct and indirect leads to evidence. E-mail files can contain actual evidence of criminal or inappropriate behavior. The investigator must thoroughly investigate e-mail containers on local and remote computers when possible. Extreme care should be taken when dealing with remote e-mail accounts. These accounts may be personal in nature and fall outside the control of corporate ownership. Although business provided e-mail services are considered business resources and can be searched by corporate investigators, tort issues may arise. If the business has not established policies concerning the use of business e-mail and the business owner rights, legal issues could arise. Corporate investigators should consult corporate legal counsel before accessing remote e-mail accounts. The investigator must be familiar with the many different methods of e-mail use: web-based email (Hotmail, Yahoo Mail, or Google Mail), remote e-mail services (pop3 accounts) and local e-mail mailboxes (Outlook and Outlook Express, Eudora, MS Exchange, Linux/Unix mail).

The process of generating file lists and hash values and gathering other information about the files can be greatly simplified with automated scripts. Computer systems can contain enormous quantities of files. For an investigator to properly analyze a file, the file's characteristics must be determined. What are the contents of the file? Is it an executable program (exe, com, sys, dll, msi, etc.), application data file (jpg, gif, pdf, doc, vbs, etc.), archive file (zip, arc, cab, tar, tgz, etc.)? Are the file contents correct for the file extension? What is the file size? Who is the owner of the file? When was the file created? When was the file last modified? When was the file last accessed? A cryptographic signature or hash value of a file is a quick way to make a comparison of two files. The open-source program sorter available in the SleuthKit can perform these functions and more. This generating of file lists (include dates and times) and hash values will allow the investigator to identify potential files for further analysis.

References

- [1] “QEMU Copy On Write disk image.” .
- [2] Wikipedia contributors, “qcow,” *Wikipedia, The Free Encyclopedia*, 14-Jan-2021. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Qcow&oldid=1000314654>. [Accessed: 26-Feb-2021].
- [3] *Whatext.com*. [Online]. Available: <https://whatext.com/qcow>. [Accessed: 26-Feb-2021].
- [4] “VirusTotal,” *Virustotal.com*. [Online]. Available: <https://www.virustotal.com/gui/>. [Accessed: 26-Feb-2021].
- [5] D. Swinhoe, “What is a keylogger? How attackers can monitor everything you type,” *Csoonline.com*, 11-Dec-2018. [Online]. Available: <https://www.csoonline.com/article/3326304/what-is-a-keylogger-how-attackers-can-monitor-everything-you-type.html>. [Accessed: 26-Feb-2021].
- [6] Wikipedia contributors, “Windows Registry,” *Wikipedia, The Free Encyclopedia*, 22-Feb-2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Windows_Registry&oldid=1008326974. [Accessed: 26-Feb-2021].
- [7] “Registry,” *Computerhope.com*. [Online]. Available: <https://www.computerhope.com/jargon/r/registry.htm>. [Accessed: 26-Feb-2021].
- [8] “DB Browser for SQLite,” *Sqlitebrowser.org*. [Online]. Available: <https://sqlitebrowser.org/>. [Accessed: 26-Feb-2021].
- [9] “SQLite Home Page,” *Sqlite.org*. [Online]. Available: <https://www.sqlite.org/index.html>. [Accessed: 26-Feb-2021].
- [10] “What is VMDK? What opens a VMDK? File format list from WhatIs.Com,” *Techtarget.com*. [Online]. Available: <https://whatis.techtarget.com/fileformat/VMDK-Virtual-Machine-Disk-file-for-VMware-virtual-machines>. [Accessed: 26-Feb-2021].