COMP6845 Report 1

Antheo Raviel Santosa

Term 2, 2024

Contents

1 Introduction

1.1 Background

1.1.1 Jim's Forensics (JF) have been engaged by Penelope Legal (PL) in relation to proceedings with Caelus Engineering (CE).

1.1.2 On the 14^{th} of September, 2021, PL instructed JF by letter containing a series of instructions on which we were to proceed.

1.2 My instructions

1.2.1 As a result of the above engagement, on the 14^{th} of September, 2021, PL advised me to prepare a report addressing the following issues:

(a) **Question 1** - Does the data on the laptop provide any indications regarding the reason for its destruction and disposal? If so, what?

(b) **Question 2** - Is there any indication of an attempt to "cover tracks", such as deleting or obscuring data?

(c) **Question 3** - Is there any indication of hidden data, *i.e.*, encryption? If so, are you able to make the data usable?

(d) **Question 4** - Any other matters that are relevant to the use or misuse of the laptop?

(e) **Question 5** - Has the additional information changed opinions on the events which occurred?

(f) **Question 6** - Your recommendation regarding any further enquiries and examinations that needed to be conducted.

1.3 Provided materials

1.3.1 We are provided the following materials from which we are to base our investigations on:

(a) a raw image of the hard disk of Alyx Hamilton's laptop,

(b) a photo of the collection site, and

(c) profiles of the staff at Caelus Engineering.

(d) a copy of the logs of entry and exit of personnel from the gates at Caelus Engineering,

(e) a screenshot of the CCTV overlooking the entry gates at Caelus Engineering,

(f) a dump of the memory of the laptop used by Alyx Hamilton,

(g) a copy of the network traffic capture data from Alyx Hamilton's laptop, as well as a log of the TLS keys necessary to decrypt the data,

(h) and copies of the log of the email correspondence of Alyx Hamilton and Michael Harris.

1.4 Qualifications

1.4.1 I am a second year computer science student at the University of New South Wales, currently enrolled in the COMP6845 course, otherwise known as *Extended Digital Forensics*.

1.5 Information relied upon

1.5.1 I have relied upon the following information in the preparation of my report:

(a) a letter containing instructions addressed to JF dated 14^{th} of September, 2021,

(b) a second letter containing further instructions addressed to JF,

(c) an image of the storage of the laptop used by Alyx Hamilton,

(d) a dump of the memory of the laptop used by Alyx Hamilton,

(e) copies of the logs of the email correspondence of Alyx Hamilton and Michael Harris,

(f) a copy of the network traffic capture data from Alyx Hamilton's laptop, as well as a log of the TLS keys necessary to decrypt the data,

(g) a screenshot of the CCTV overlooking the entry gates at Caelus Engineering,

(h) a copy of the logs of the entry and exit of personnel from the gates at Caelus Engineering,

(i) a photo of the collection site, and

(j) profiles of the staff at Caelus Engineering.

1.6 Disclaimer

1.6.1 I have made all inquiries which I believe are desirable and appropriate for the purposes of this report. There are no matters of significance which I regard as relevant to my opinions which have been withheld.

1.6.2 This report has been prepared solely for the use of PL. In accordance with the usual practice of JF, I expressly disclaim all responsibility to any other person or entity (other than the Court) for any reliance on the content of this report. This report should not be copied or distributed to any other person or entity, other than in connection with the above matter.

1.7 Assumptions

1.7.1 In preparing this report, I have made the following assumptions:

(a) the information described in the background section above is accurate, and

(b) the computers identified by CE, and subsequently imaged, were used by Alyx Hamilton.

1.8 Structure of this report

1.8.1 The remaining sections of this report addresses the following:

(a) In Section 2, I set out information with respect to Question 1 above;

(b) In Section 3, I set out information with respect to Question 2 above;

(c) In Section 4, I set out information with respect to Question 3 above;

(d) In Section 5, I set out information with respect to Question 4 above;

(e) In Section 6, I set out information with respect to Question 5 above;

(f) In Section 7, I set out information with respect to Question 6 above;

1.8.2 For convenience, I note that I have organised my findings in the same order as the questions outlined in my letter of instruction. I have also included questions, in some instances paraphrased, from the letter of instruction before presenting my corresponding findings. Additionally, I have referred to the appropriate section number as contained in the letter of instruction in my responses.

1.9 Code of conduct

1.9.1 I understand that my report is required for the purpose of proceedings in the Supreme Court of New South Wales. Accordingly, I confirm that I have read and agree to be bound by the Expert Witness Code of Conduct (Schedule 7) of the Uniform Civil Procedure Rules 2005 (NSW).

2 Question 1: Reason for destruction

2.1 Procurement and extraction of evidence

2.2 Extraction of evidence

2.2.1 This section describes the method with which the evidence laid out in the rest of this report is procured.

2.2.2 A disk image is a digital replica of the contents of a physical drive, including any unallocated space.

2.2.3 I note that there are no additional cryptographic hashes provided with the disk image, nor is the disk image formatted in a forensically admissible format (such as E01).

2.2.4 I extracted the information presented from this point onwards from the provided disk image using the Autopsy forensics toolkit.

Table 1: Google Drive URLs accessed

Date accessed	Title	URL
2020-09-04 14:31:21 AEST	PROJECTS 19:20 - Google Drive	https://drive.google.com/drive/folders/1S7ETsR
2020-09-04 14:27:57 AEST	Shared Drive - Google Drive	https://drive.google.com/drive/folders/1sinCWR
2020-09-04 14:27:55 AEST	Shared with me - Google Drive	https://drive.google.com/drive/shared-with-me

Table 2: List of attached external USB drives

No.	Device make	Device model	Device ID	Date/Time
1	SanDisk Corp.	Cruzer Blade	4C530300831216101192	2020-09-04 10:11:55 AEST
2	SanDisk Corp.	Product: 55A5	4C530000050910115114	2020-09-04 15:50:09 AEST

2.3 Email

2.3.1 I note the existence of an email from `alyx.hamilton@caelusengineering.com.au` sent to the address `johndavis5891@gmail.com` at 2020-09-04 16:27:18 AEST with the subject: *"It's done"*. Some notable details include:

(a) The body of the email contains only the following line: *"The files are copying over as we speak. What should I do now? Where do I meet you?"*.

(b) The email is found on the `\\Root - Mailbox\IPM_SUBTREE\[Gmail]\Bin`.

(c) This email is the only correspondence found between Alyx Hamilton and `johndavis5891@gmail.com`.

(d) This email is the last email found on the disk image of Alyx Hamilton's laptop

2.4 Accessing Google Drive

2.4.1 Some of the most web history on the device displays the rapid access of various pages in the Google Suite. These are briefly listed in Table 1

2.5 Attached USB drives

2.5.1 Table 2 details the make, model, and ID of two external USB drives that are recorded to have been connected to this device, as well as the time they were attached.

2.6 Discussion

The email found on 2.2.1 appears to be suspicious from multiple aspects. The most noticeable among which is its cryptic language. Furthermore, the recipient is not an address within Caelus Engineering, nor is it one which otherwise appears in other emails in Alyx Hamilton's mailbox. Lastly, the email is found in the `Bin` directory, which indicates attempts at hiding it.

The web activity discussed in 2.3.1 were also found to be within temporal proximity of the email discussed above. They directly precede the web history entry belonging to said email. The site accessed, belonging to "Google Drive" and "Google Docs", are benign, but may be used as a method of exfiltrating data.

Likewise, the external USB devices that are noted to have been connected to the device in 2.4.1 are occurences that are otherwise benign, but may be used as a method of exfiltrating data.

3 Question 2: Covering tracks

3.1 Deleted email

3.1.1 The email discussed previously in 2.2.1 was found on the folder `\\Root - Mailbox\IPM_SUBTREE\[Gmail]\Bin`, which indicates an attempt at deleting the email.

3.2 Deleted project archives

3.2.1 ZIP archives containing what appears to be project files appear to have been deleted and are found in the 'Recycle Bin', as can be seen in Figure 4 in the appendix.

4 Question 3: Encryption

4.1 Introduction

4.1.1 The NIST[1] defines "encrypt" as to "cryptographically transform data to produce cipher text".

(a) The NIST[2] further defines "cryptography" as "the discipline that embodies the principles, means, and methods for the transformation of data in order to hide their semantic content, prevent their unauthorized use, or prevent their undetected modification".

(b) The NIST[3] further defines "cipher text" as "data in its encrypted form".

4.1.2 Encryption can therefore be defined as the transformation of data in order to ensure its secrecy and integrity.

4.1.3 Encryption is a virtually ubiquitous feature of modern computing technology, as its nature of providing secrecy has a wealth of uses, whether benign (such as protecting one's financial information when performing a transaction online) or malicious (such as obscuring evidence of illegal activities).

4.2 Procurement and extraction of evidence

4.2.1 This section describes the methodology with which the evidence laid out in the rest of this report is procured.

4.2.2 A pcap, or *packet capture* file, is a file which stores network packet data. Such a file is typically procured by capturing the packets which passes through a device (note that this does not necessarily mean that every packet captured involves said device). Such a file logs all network communication that said device "hears".

(a) A "packet" is a small segment of a larger message, which are sent through computer networks, and are recombined by the recipient to obtain the whole message.[4]

4.2.3 JF is provided the pcap file procured from Alyx Hamilton's laptop by CE.

4.2.4 A cryptographic hash (using the MD5 algorithm) is provided alongside the pcap file. The calculated hash of the pcap file appears to match the provided hash, as can be seen in Figure 5. Assuming that the integrity of the file containing the provided hash is maintained, this means that the pcap file has not been tampered with throughout the span of time between the calculation of the provided hash and the time of verification.

4.2.5 Secure Sockets Layer (SSL), as well as its successor, Transport Layer Security (TLS), are protocols for encrypting, securing, and authenticating communications that take place on the internet.[5] It is used to ensure the secrecy and authenticity of communication over the internet.

4.2.6 Because the pcap file contains packets encrypted using SSL, a log of the SSL keys necessary to decrypt them is also provided to JF by CE. A cryptographic hash of the log is also provided, and the calculated hash of the log file appears to match that of the provided hash, as can be seen in Figure 6. Similar to the pcap file, this means that the pcap file has not been tampered with throughout the span of time between the calculation of the provided hash and the time of verification, assuming that the integrity of the file containing the provided hash is maintained.

[1] NIST, "encrypt," in Glossary | CSRC. [Online]. Available: https://csrc.nist.gov/glossary/term/encrypt. [Accessed: 22-Jul-2024].

[2] NIST, "cryptography," in Glossary | CSRC. [Online]. Available: https://csrc.nist.gov/glossary/term/cryptography. [Accessed: 22-Jul-2024]

[3] NIST, "cipher text," in Glossary | CSRC. [Online]. Available: https://csrc.nist.gov/glossary/term/cipher_text. [Accessed: 22-Jul-2024]

[4] Cloudflare, "What is a Packet?" Cloudflare. Available: https://web.archive.org/web/20240622023826/https://www.cloudflare.com/learning/netw layer/what-is-a-packet/. [Accessed: 22-Jul-2024].

[5] Cloudflare, "How does SSL work?" Cloudflare. [Online]. Available: https://www.cloudflare.com/learning/ssl/how-does-ssl-work/. [Accessed: 22-Jul-2024].

4.2.7 Wireshark is a network traffic analyser.[6] It is a tool that can be used to capture packets from the host device's network interface, creating a pcap file, and analysing pcap files.

(a) The pcap file can be accessed in Wireshark by clicking `File > Open` on the top menu bar, and then selecting the file from the resulting file explorer dialog.

(b) After opening the pcap file, the SSL key log can be imported by selecting `Edit > Preferences` on the top menu bar, selecting `Protocols > TLS` on the preferences menu, and then clicking the the `Browse` button underneath `(Pre)-Master-Secret log filename` and selecting the SSL key log file, as can be seen in Figure 7.

4.2.8 Volatile memory is a type of computer memory which loses its content when power is turned off or lost.[7]

4.2.9 While not the only type of volatile memory, Random Access Memory (RAM), which is also commonly referred to as *memory*, is the most widely recognised type of volatile memory. Other types of volatile memory, such as CPU registers and caches exist, but are typically very small in size (with sizes measured in bits[8] and kilobytes[9]).

4.2.10 Volatility is a framework for extracting digital artifacts from samples of volatile memory, or more precisely, RAM.[10] Specifically, I have used version 3 of Volatility.

4.2.11 Volatility can be obtained via the `pip` package manager for Python. For the sake of reproducibility and convenience, a copy of the Dockerfile I have used to set up my working environment for Volatility 3, alongside instructions on how to use it, is provided in Section 5.2

4.2.12 After Volatility 3 is procured, the provided memory image can then be accessed using Volatility 3 by passing it as a command-line argument, like so: `vol.py -f Windows-7-x64-Pro-Snapshot7.7z [plugin name]`, where `Windows-7-x64-Pro-Snapshot7.vmem` is the path to the memory dump. This path assumes that the memory dump file resides in the current working directory; it may need to be adjusted otherwise.

(a) Volatility 3 comes with a set of plugins, which instructs the tool on how to analyse the memory dump and present the resulting information. An introduction on the various plugins available to both Volatility 2 and 3, and how to use them, can be found on HackTricks.

4.2.13 Docker is a tool which creates and manages *containers*, which are isolated environments containing all the code and dependencies of an application.[11] This tool facilitates the creation of standardised, uniform environments across different machines.

4.2.14 A *Dockerfile* is a file containing instructions pertaining the creation of a container image.[12] A Docker image is a standalone executable used to create a container;[13] it is effectively a *template* that can be used to instantiate containers.

4.3 Network encryption

4.3.1 Quick UDP Internet Connections (QUIC) is an encrypted-by-default transport protocol originally developed by Google[14] that is supported by the Google Chrome web browser, beginning small-scale deployments of the original implementation, gQUIC in 2013, and eventually default-enabling its successor, IETF QUIC in 2021 with Chrome 93.[15]

[6] Wireshark Foundation, "Wireshark," GitLab. [Online]. Available: https://gitlab.com/wireshark/wireshark. [Accessed: 22-Jul-2024].

[7] NIST, "Volatile Memory," in Glossary | CSRC. [Online]. Available: https://csrc.nist.gov/glossary/term/volatile_memory. [Accessed: 22-Jul-2024].

[8] E. Edwards, "Memory," Imperial College London. [Online]. Available: https://www.doc.ic.ac.uk/ eedwards/compsys/memory/index.html. [Accessed: 22-Jul-2024].

[9] K. Huck, "Cache Lines and Cache Size," National Institute of Computer Science. [Online]. Available: https://www.nic.uoregon.edu/ khuck/ts/acumem-report/manual_html/ch03s02.html. [Accessed: 22-Jul-2024].

[10] Volatility Foundation, "Volatility 3," GitHub. [Online]. Available: https://github.com/volatilityfoundation/volatility3. [Accessed: 22-Jul-2024].

[11] Oracle, "What is Docker?" Oracle. [Online]. Available: https://www.oracle.com/au/cloud/cloud-native/container-registry/what-is-docker/. [Accessed: 22-Jul-2024].

[12] Docker, "Dockerfile reference," Docker Docs. [Online]. Available: https://docs.docker.com/reference/dockerfile/. [Accessed: 22-Jul-2024].

[13] Amazon Web Services, "What's the Difference Between Docker Images and Containers?" AWS. [Online]. Available: https://aws.amazon.com/compare/the-difference-between-docker-images-and-containers/. [Accessed: 22-Jul-2024].

[14] A. Ghedini, "The Road to QUIC," Cloudflare Blog. [Online]. Available: https://blog.cloudflare.com/the-road-to-quic. [Accessed: 22-Jul-2024].

[15] Chromium Project, "QUIC, a multiplexed transport over UDP," Chromium. [Online]. Available: https://www.chromium.org/quic/. [Accessed: 22-Jul-2024].

4.3.2 I note that the version of Google Chrome running on Alyx Hamilton's laptop is 84. This can be found by extracting the list of DLLs from Google Chrome processes found on the memory dump using Volatility, as can be seen in Figure 8. The use of memory analysis here further ascertains that this is the version of Google Chrome that is being actively used on Alyx Hamilton's laptop, rather than one that is only installed. Two points of observation can be drawn from this:

(a) It does not appear to be possible to decrypt the QUIC packets sent to and from Google Chrome in the provided pcap file, because the export of QUIC secrets is only available on version 89 onwards of Google Chrome.[16]

(b) It appears that the use of QUIC is manually enabled on the Google Chrome browser running on Alyx Hamilton's laptop, as the protocol is only enabled by default from version 93 onwards of Google Chrome.

4.4 Analysis and discussion

4.4.1 There are two methods of network encryption discovered from the provided artefacts: SSL encryption, and QUIC encryption.

(a) SSL encryption is almost ubiquitously enabled by default on most modern programs. It is unlikely that this is intentionally enabled by some party.

(b) QUIC encryption is likely enabled with intention by some party, as it is not enabled by default on version 84 of Google Chrome, which is in use on Alyx Hamilton's laptop. Furthermore, the toggle to enable the use of QUIC can only be accessed by entering `chrome://flags/` in the browser's URL bar and setting `Experimental QUIC protocol` to `Enabled`[17], rendering the possibility that this feature is enabled by accident (without intent) unlikely.

4.5 Conclusion

Indications of encryption were discovered, though not always accompanied by indications of *intent* to encrypt.

Network encryption Using the provided SSL key log, SSL-encrypted packets were able to be decrypted and "made usable". However, QUIC-encrypted packets, which specifically contains communications to and from the Google Chrome browser running on Alyx Hamilton's laptop, were not able to be decrypted due to the technical limitations present in the version of Google Chrome in use.

[16] Wireshark Foundation, "The TLS/QUIC sessions can't be decrypted," GitLab. [Online]. Available: https://gitlab.com/wireshark/wireshark/-/issues/17111. [Accessed: 22-Jul-2024].

[17] M. Geniar, "Enable QUIC protocol in Google Chrome," Mattias Geniar's Blog. [Online]. Available: https://ma.ttias.be/enable-quic-protocol-google-chrome/. [Accessed: 22-Jul-2024].

5 Appendix

5.1 Figures

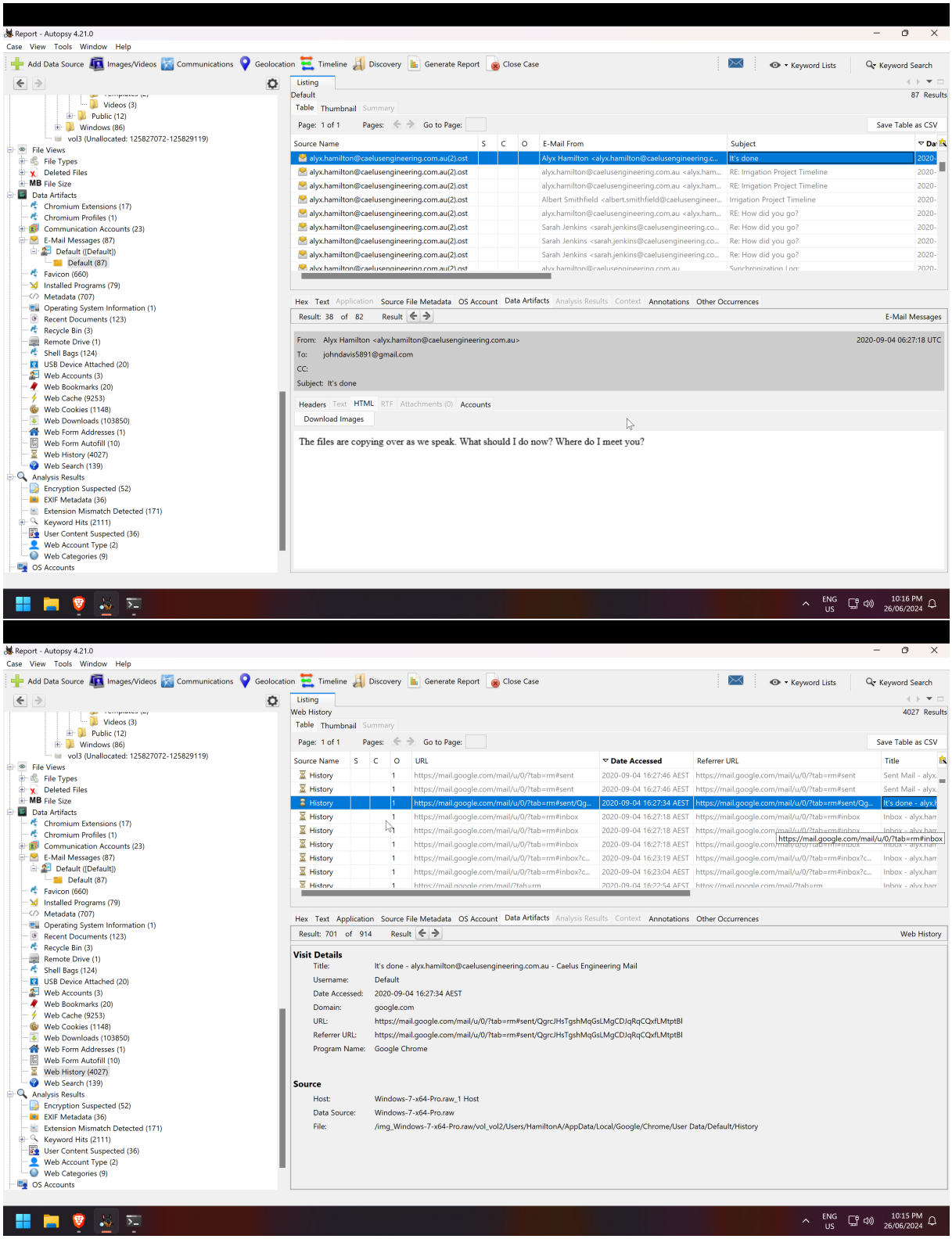


Figure 1: Cryptic email

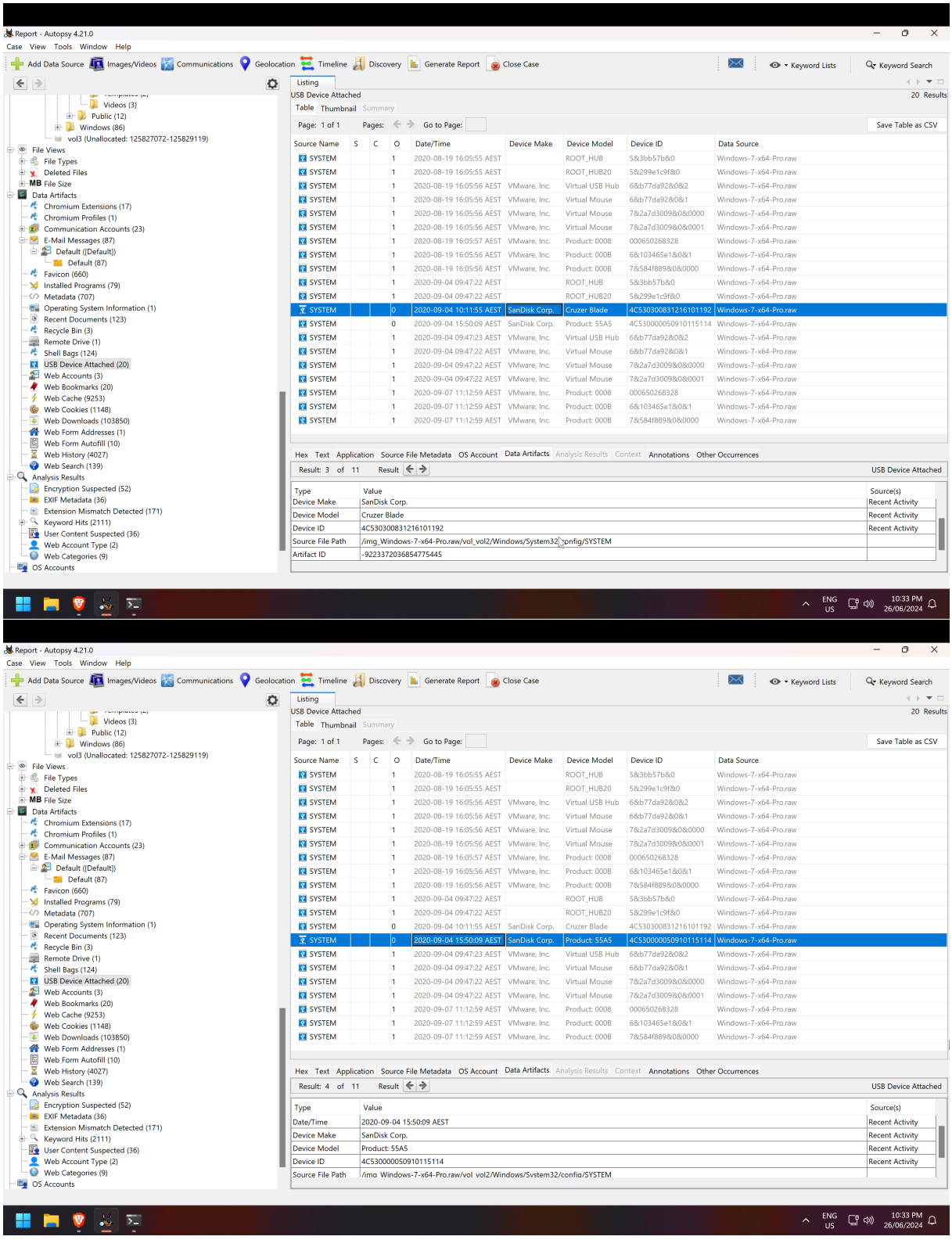
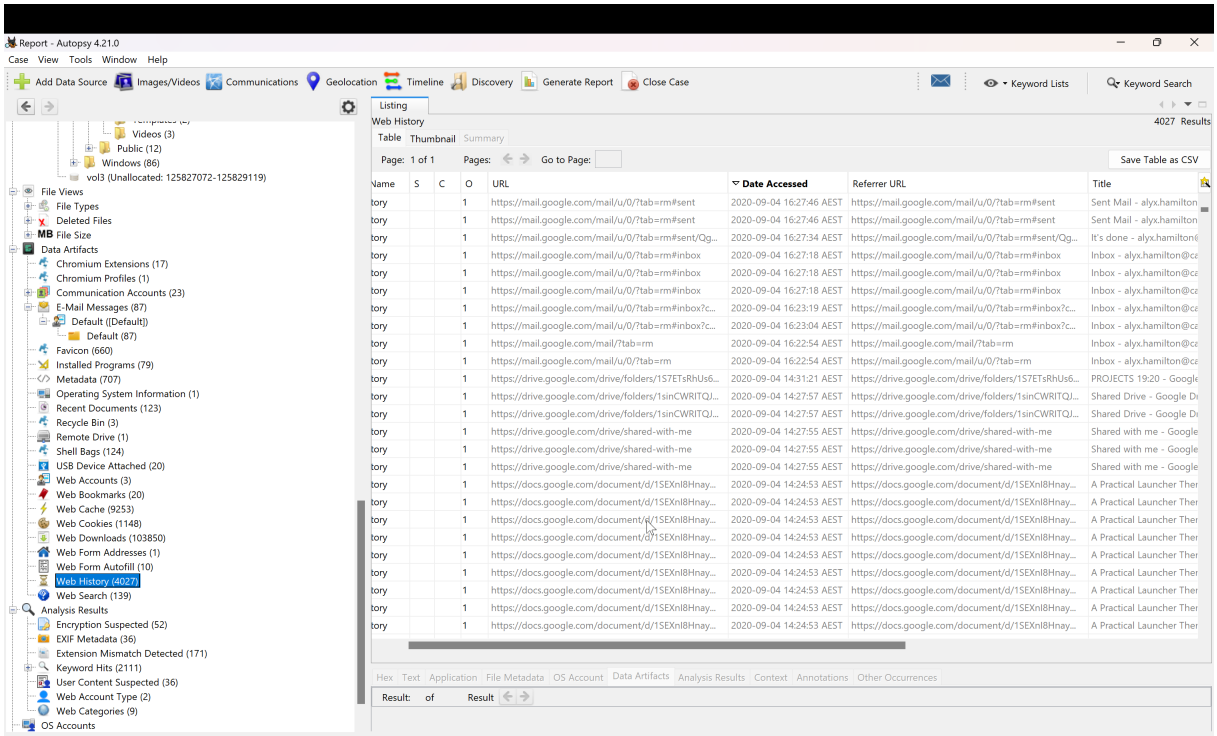
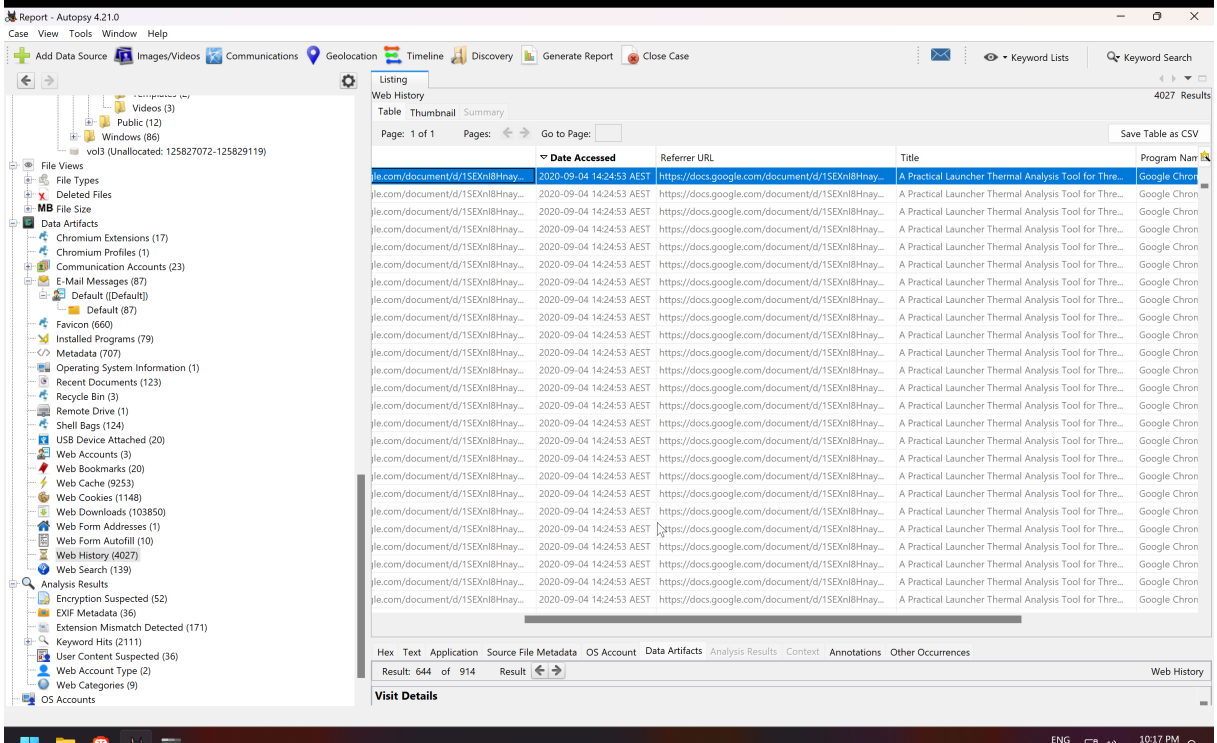
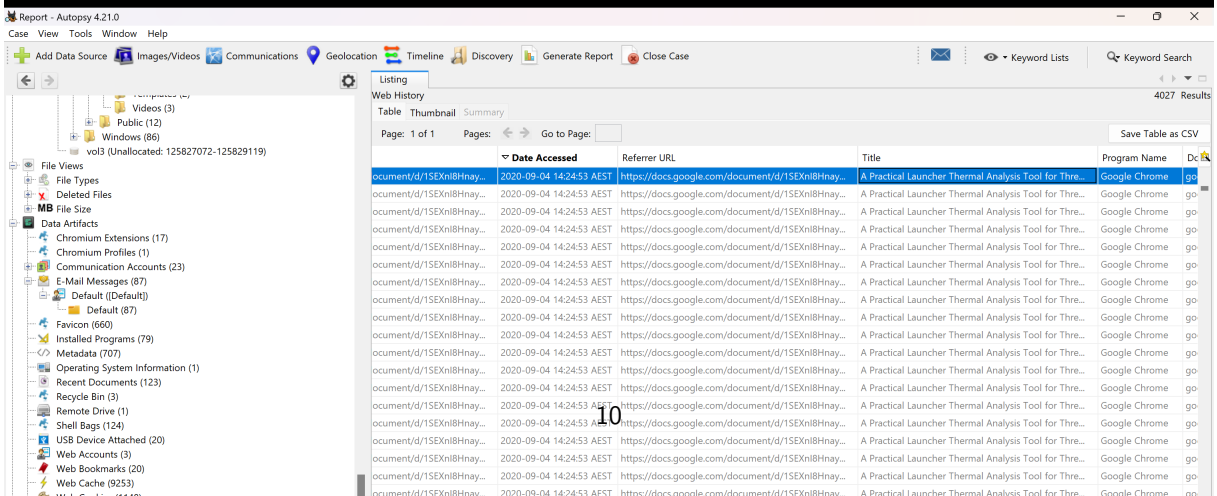


Figure 2: Attached USB drives


Report - Autopsy 4.21.0
Case View Tools Window Help
Add Data Source Images/Videos Communications Geolocation Timeline Discovery Generate Report Close Case
Keyword Lists Keyword Search
Listing
Web History
4027 Results
Table Thumbnail Summary
Page: 1 of 1 Pages: Go to Page:
Save Table as CSV
Name S C O URL Date Accessed Referrer URL Title
Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences
Result: of Result
ENG US 10:17 PM 26/06/2024


Report - Autopsy 4.21.0
Case View Tools Window Help
Add Data Source Images/Videos Communications Geolocation Timeline Discovery Generate Report Close Case
Keyword Lists Keyword Search
Listing
Web History
4027 Results
Table Thumbnail Summary
Page: 1 of 1 Pages: Go to Page:
Save Table as CSV
Date Accessed Referrer URL Title Program Name
Hex Text Application Source File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences
Result: 644 of 914 Result
Web History
Visit Details
ENG US 10:17 PM 26/06/2024


Report - Autopsy 4.21.0
Case View Tools Window Help
Add Data Source Images/Videos Communications Geolocation Timeline Discovery Generate Report Close Case
Keyword Lists Keyword Search
Listing
Web History
4027 Results
Table Thumbnail Summary
Page: 1 of 1 Pages: Go to Page:
Save Table as CSV
Date Accessed Referrer URL Title Program Name

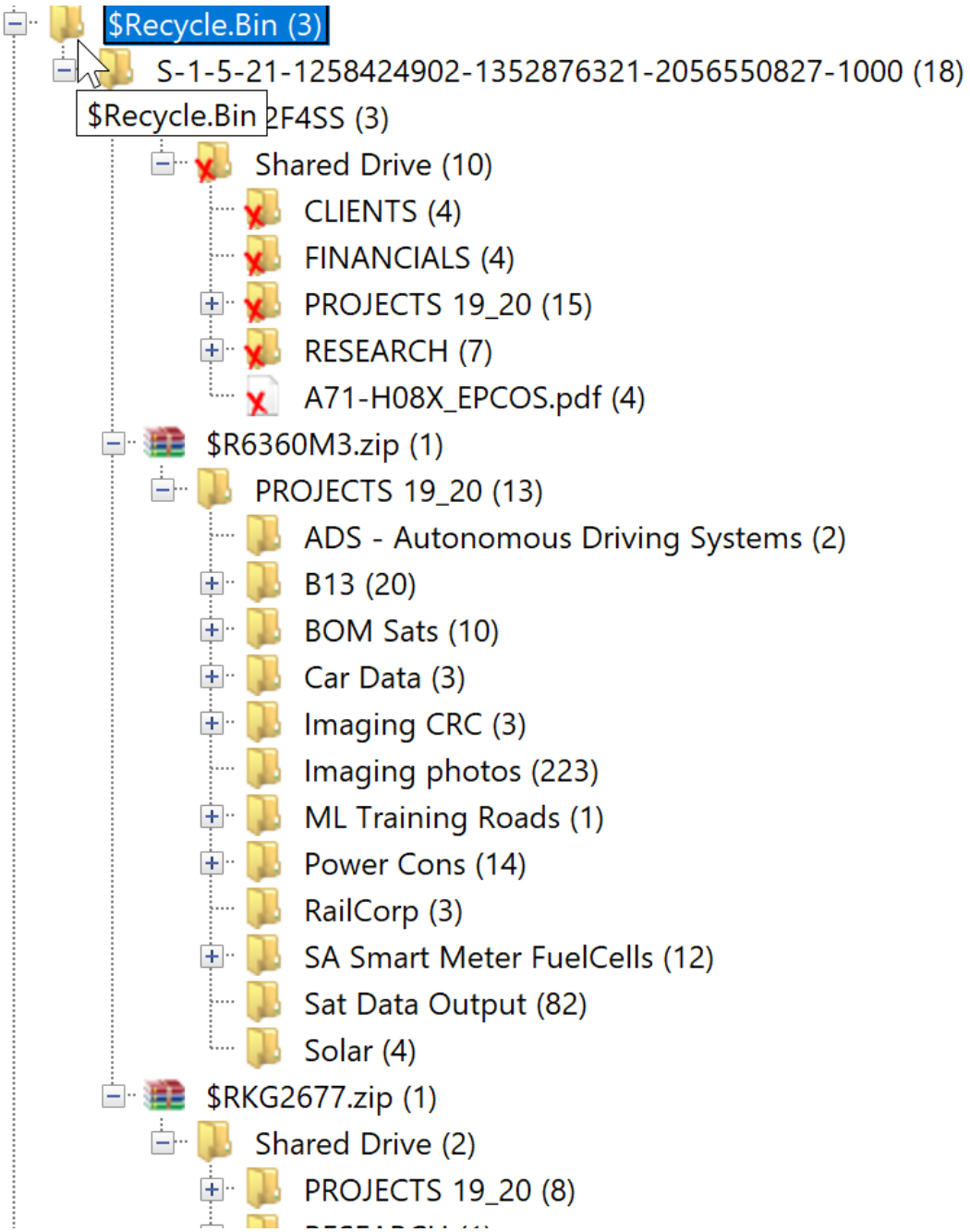


Figure 4: Recycling bin

```
bash-5.2$ cat files.hash | rg hamiltona_network_log.pcapng
45330b81a6a3c0b7cc5dfd4428e1e737  hamiltona_network_log.pcapng
bash-5.2$ md5sum hamiltona_network_log.pcapng
45330b81a6a3c0b7cc5dfd4428e1e737  hamiltona_network_log.pcapng
bash-5.2$ if diff <(md5sum hamiltona_network_log.pcapng) <(rg 'hamiltona_network_log.pcapng' files.hash); then echo "hash matches"; else echo "hash does not match"; fi
hash matches
```

Figure 5: Cryptographic hash verification of the packet capture file

```
bash-5.2$ cat files.hash | rg ssl-keys.log
6ab3c33164a2eeffbb3c1b850e80213d  ssl-keys.log
bash-5.2$ md5sum ssl-keys.log
6ab3c33164a2eeffbb3c1b850e80213d  ssl-keys.log
bash-5.2$ if diff <(md5sum ssl-keys.log) <(rg 'ssl-keys.log' files.hash); then echo "hash matches"; else echo "hash does not match"; fi
hash matches
```

Figure 6: Cryptographic hash verification of the SSL key log file

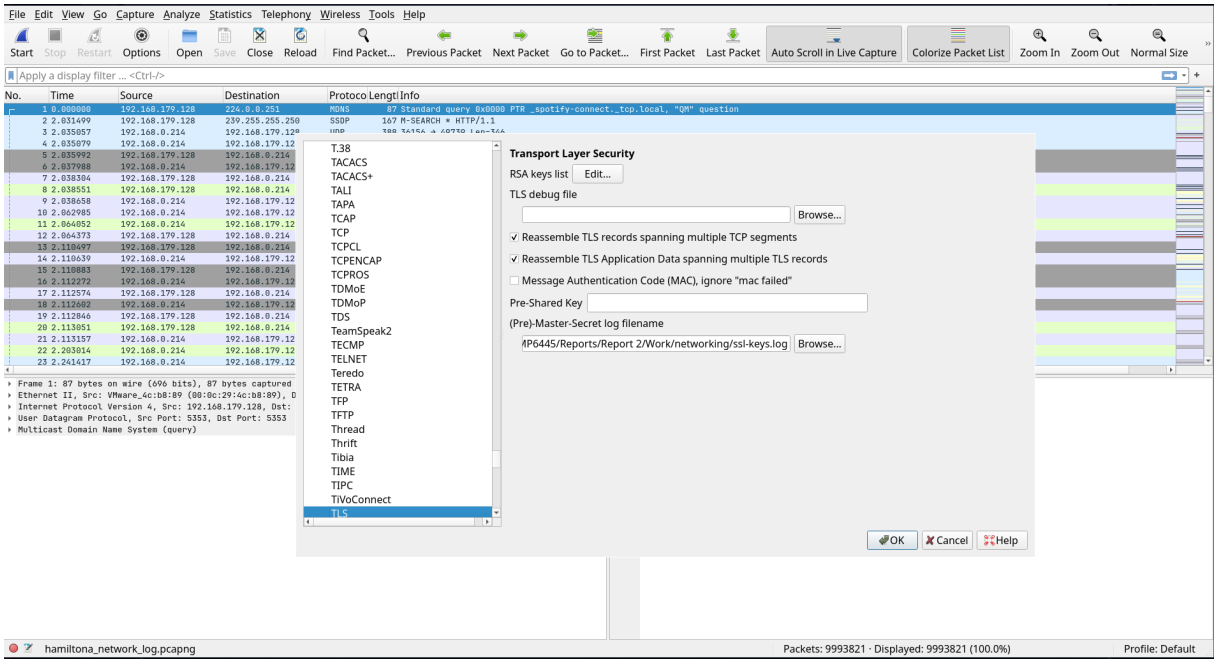


Figure 7: Importing the SSL key log to Wireshark

```
Work/drop2/memory
● [podman] ❱ vol.py -f Windows-7-x64-Pro-Snapshot7.vmem windows.pslist 2> /dev/null | rg chrome | awk '{print$1}' > pids

Work/drop2/memory
● [podman] ❱ while IFS= read -r pid; do
    vol.py -f Windows-7-x64-Pro-Snapshot7.vmem windows.dlllist.DllList --pid $pid 2> /dev/null >> dlllist
done < "pids"

Work/drop2/memory took 4s
● [podman] ❱ rg -i 'chrome.dll' dlllist
28:1084 chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:25.000000     Disabled
170:1428        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:26.000000     Disabled
243:3684        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:26.000000     Disabled
307:964 chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:30.000000     Disabled
355:2764        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:37.000000     Disabled
403:1508        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 06:22:54.000000     Disabled
451:3056        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 06:22:59.000000     Disabled
510:1084        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:25.000000     Disabled
652:1428        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:26.000000     Disabled
725:3684        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:26.000000     Disabled
789:964 chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:30.000000     Disabled
837:2764        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 04:17:37.000000     Disabled
885:1508        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 06:22:54.000000     Disabled
933:3056        chrome.exe      0x7fee4d20000   0x821c000       chrome.dll      C:\Program Files (x86)\Google\Chrome\Application\84.0.4147.135\chrome.dll       2020-09-04 06:22:59.000000     Disabled
```

Figure 8: Finding the version of Google Chrome used from the memory dump

5.2 Dockerfile for Volatility 3 Environment

5.2.1 Dockerfile

```
FROM alpine:edge

# Update and install necessary packages
RUN apk update && apk upgrade && apk add --no-cache \
    python3 \
    7zip \
    shadow \
    curl \
    git \
    tmux \
    neovim \
    starship \
    clang \
    zsh \
    bat \
```

```
    eza \
    fzf \
    ripgrep \
    fd \
    bind-tools \
    py3-virtualenv \
    termshark \
    traceroute \
    neomutt

# Add a new user
RUN adduser -D COMP6845
RUN usermod -aG wireshark COMP6845
RUN chsh -s /bin/zsh COMP6845

USER COMP6845
WORKDIR /home/COMP6845

# Clone necessary repositories
RUN git clone https://github.com/tmux-plugins/tpm /home/COMP6845/.local/share/tmux/plugins/tpm
RUN git clone https://github.com/zsh-users/zsh-syntax-highlighting.git /home/COMP6845/.local/share/zs
RUN git clone https://github.com/zsh-users/zsh-autosuggestions /home/COMP6845/.local/share/zsh/zsh-au
RUN git clone https://github.com/Aloxaf/fzf-tab /home/COMP6845/.local/share/zsh/fzf-tab

# Set up volatility
RUN mkdir -p /home/COMP6845/.local/bin
WORKDIR /home/COMP6845/.local/bin
RUN python3 -m venv volatility
RUN ./volatility/bin/pip install --upgrade pip
RUN ./volatility/bin/pip install volatility3

WORKDIR /home/COMP6845
RUN echo "alias 'vol.py'='/home/COMP6845/.local/bin/volatility/bin/vol'" >> .zshrc

# Create Mail directory and copy mbox files
RUN mkdir -p /home/COMP6845/Mail
COPY --chown=COMP6845:COMP6845 Emails/Harris/all.mbox /home/COMP6845/Mail/harris.mbox
COPY --chown=COMP6845:COMP6845 Emails/Hamilton/all.mbox /home/COMP6845/Mail/hamilton.mbox

# Configure NeoMutt
RUN echo 'set mbox_type=mbox' >> /home/COMP6845/.neomuttrc
RUN echo 'set folder=~/Mail' >> /home/COMP6845/.neomuttrc
RUN echo 'mailboxes ~/Mail/hamilton.mbox ~/Mail/harris.mbox' >> /home/COMP6845/.neomuttrc
RUN echo 'set spoolfile=~/Mail/hamilton.mbox' >> /home/COMP6845/.neomuttrc

CMD ["zsh"]
```

5.2.2 zshrc

```
# Lines configured by zsh-newuser-install
HISTFILE=~/.histfile
HISTSIZE=1000
SAVEHIST=1000
bindkey -e
# End of lines configured by zsh-newuser-install
# The following lines were added by compinstall
zstyle :compinstall filename '/home/COMP3141/.zshrc'

autoload -Uz compinit
compinit
# End of lines added by compinstall
```

```
alias vim=nvim
alias v=nvim

alias cat=bat

alias ls='eza --icons'
alias ll='eza --icons -l'
alias la='eza --icons -la'

# Plugins
source ~/.local/share/zsh/zsh-syntax-highlighting/zsh-syntax-highlighting.zsh
source ~/.local/share/zsh/zsh-autosuggestions/zsh-autosuggestions.zsh
source ~/.local/share/zsh/fzf-tab/fzf-tab.plugin.zsh

# Start starship
# ~/.zshrc
eval "$(starship init zsh)"
```

5.2.3 Instructions for use

Install and set up Docker or Podman. Then:

1. Create a new, empty directory and change into said directory.
2. Copy the contents of Section 5.2.1 into a file named `Dockerfile` in the current working directory.
3. Copy the contents of Section 5.2.2 intto a file named `.zshrc` in the current working directory.
4. Execute the following command to build the container image: `docker build -t comp6845-report2 .`
5. Execute the following command to initialise a new container from the image that was previously built and enter it: `docker run -it -name=COMP6845-Report2 comp6845-report2 zsh`