



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

FALL SEMESTER **2023-24**

LAB **ASSESSMENT -1**

NAME:- Namit Mehrotra

Registration Number:- 21BCE0763

Course Name:- Information Security Analysis and Audit Lab BCSE353E

Slot:- L57+L58

Date:- 10-06-2023

EXERCISE 1-A: Examination of Packages.xml from an Android phone

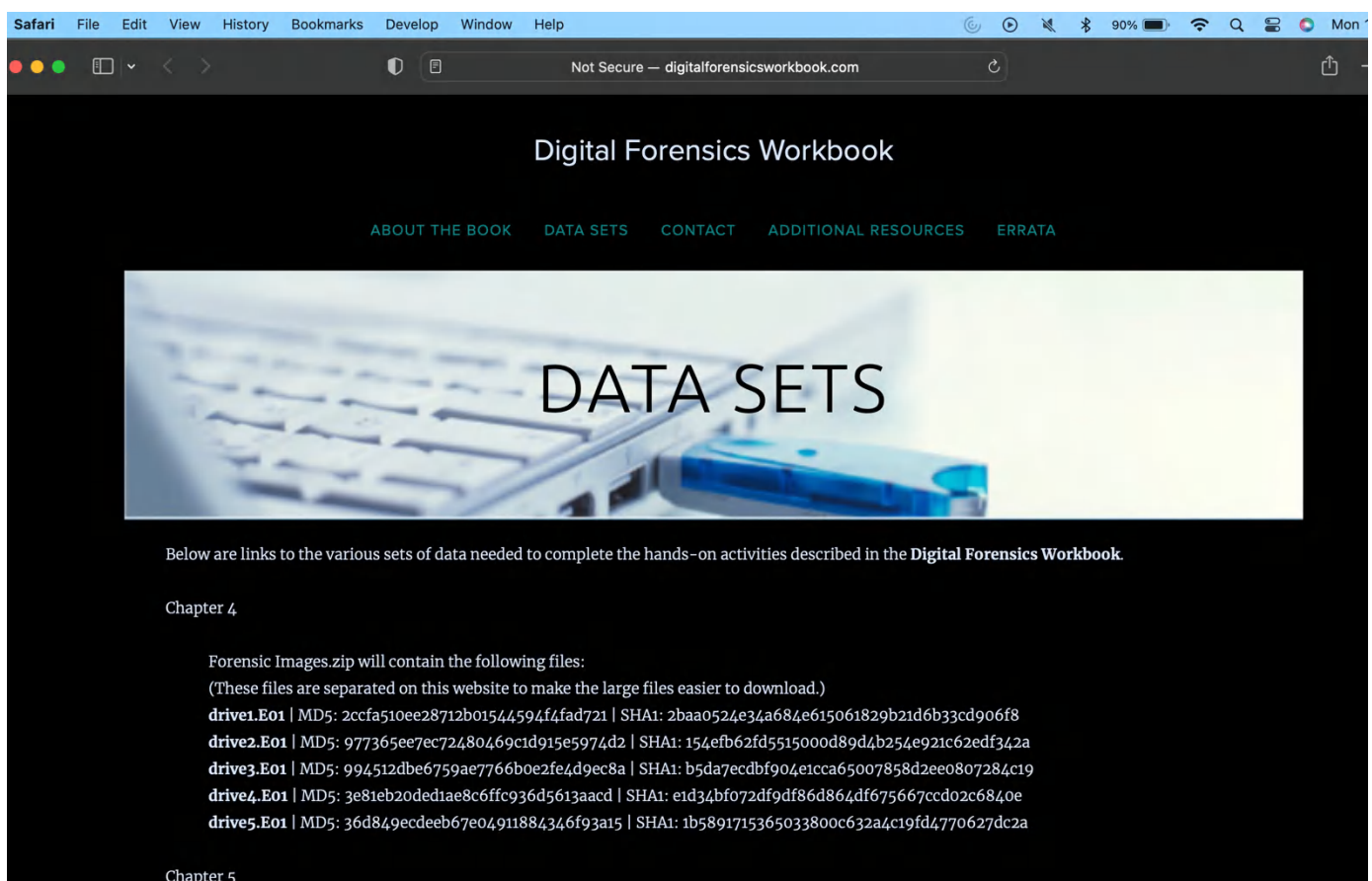
I. AIM: Analyse the packages.xml file from a phone to identify apps and their associated permissions (Download the file named “packages.xml.zip” from the: <https://www.digitalforensicsworkbook.com/data-sets>)

II. TOOLS REQUIRED:

1. Products: packages.XML from <https://www.digitalforensicsworkbook.com/data-sets>)
2. Internet browser: Google Chrome
3. Manufacturer: various

III. STEP BY STEP PROCEDURE:

1. Download the file named “packages.xml.zip” from the <https://www.digitalforensicsworkbook.com/data-sets>.
2. Extract the file packages.xml from the compressed file and place it on the desktop.



The screenshot shows a Safari browser window displaying the Digital Forensics Workbook website. The page has a dark header with the site name and navigation links: ABOUT THE BOOK, DATA SETS, CONTACT, ADDITIONAL RESOURCES, and ERRATA. Below the header is a large banner image of a computer keyboard with the text "DATA SETS" overlaid. Underneath the banner, a paragraph states: "Below are links to the various sets of data needed to complete the hands-on activities described in the Digital Forensics Workbook." This is followed by a section for "Chapter 4" which lists five forensic image files (drive1.E01 to drive5.E01) with their corresponding MD5 and SHA1 hashes. The page number "Chapter 5" is visible at the bottom left.

Digital Forensics Workbook

ABOUT THE BOOK DATA SETS CONTACT ADDITIONAL RESOURCES ERRATA

DATA SETS

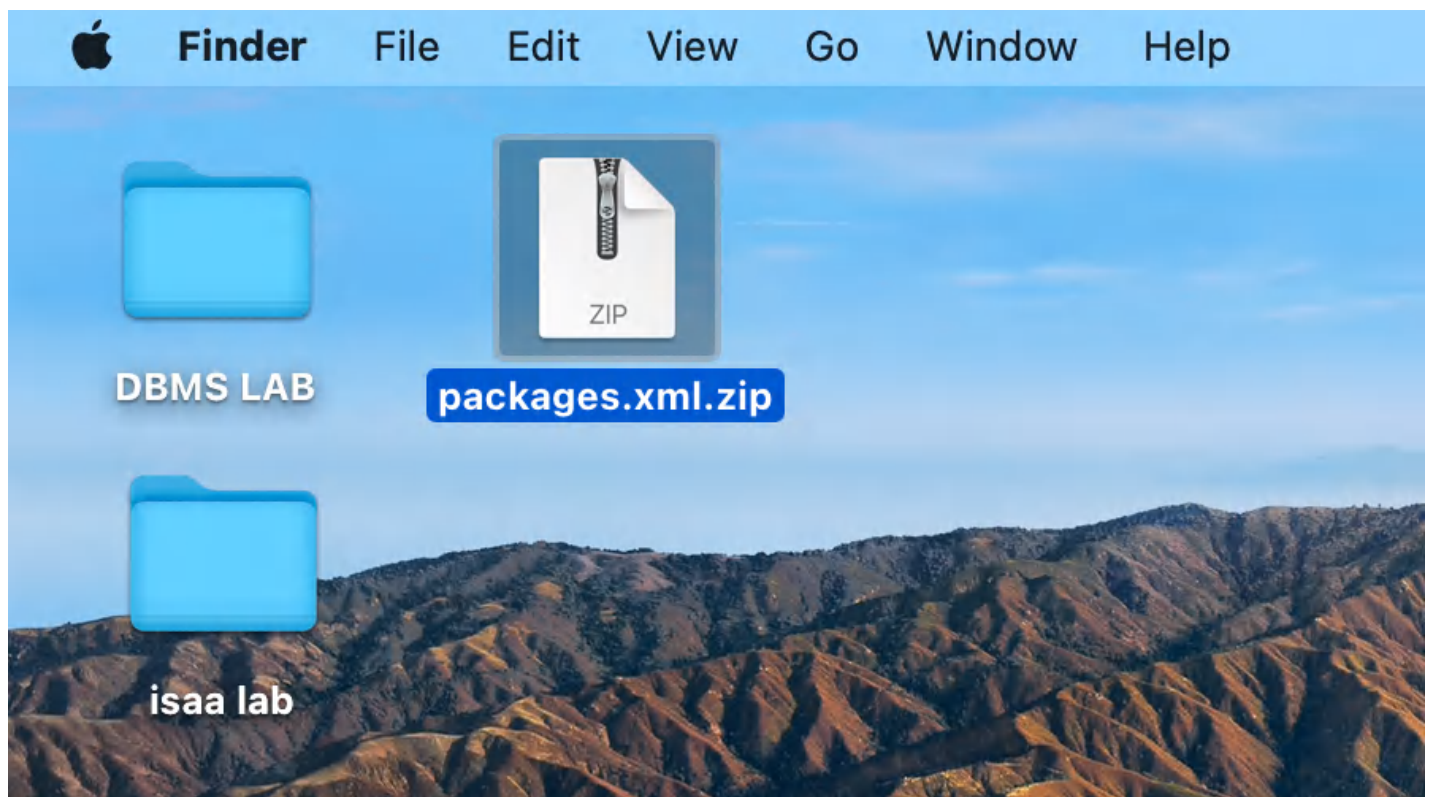
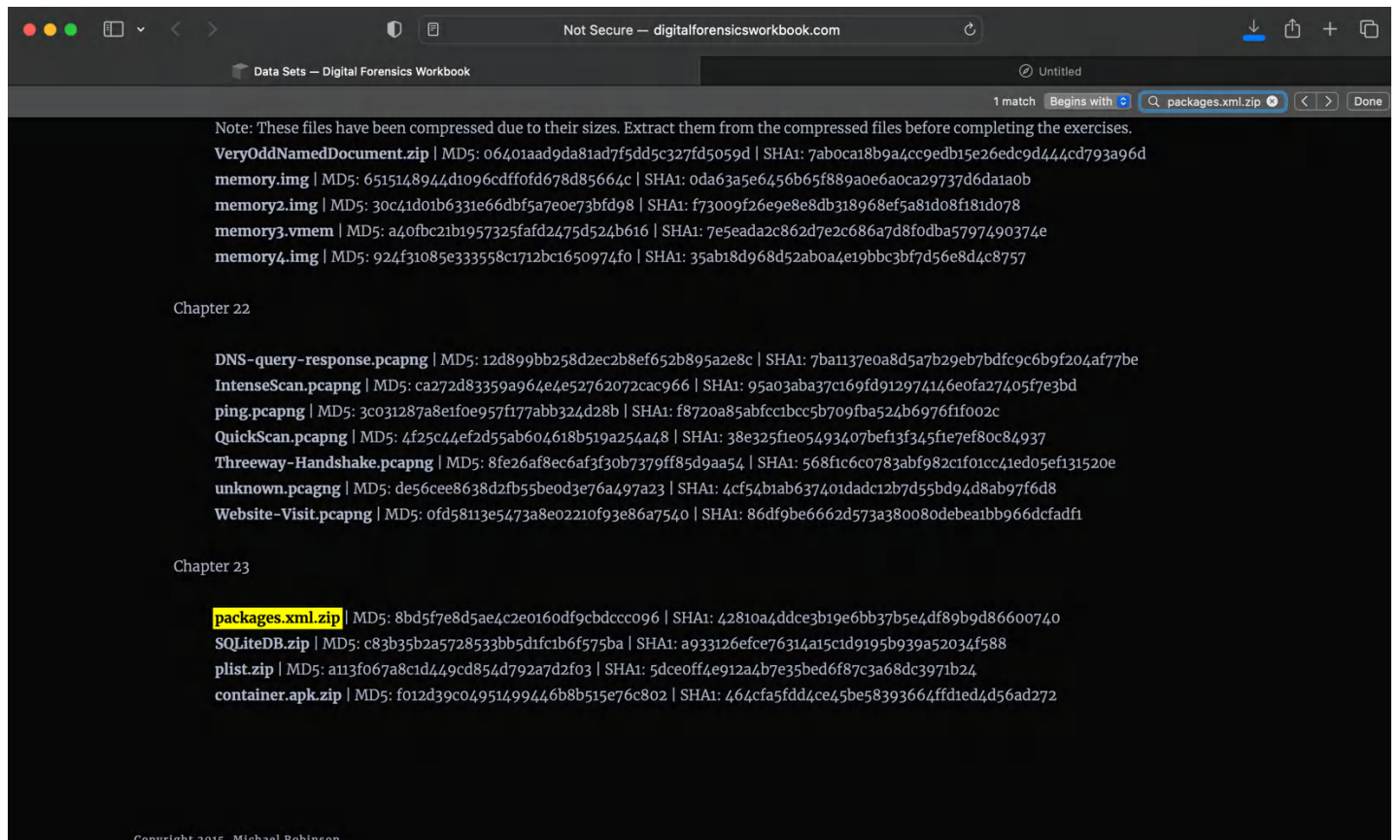
Below are links to the various sets of data needed to complete the hands-on activities described in the Digital Forensics Workbook.

Chapter 4

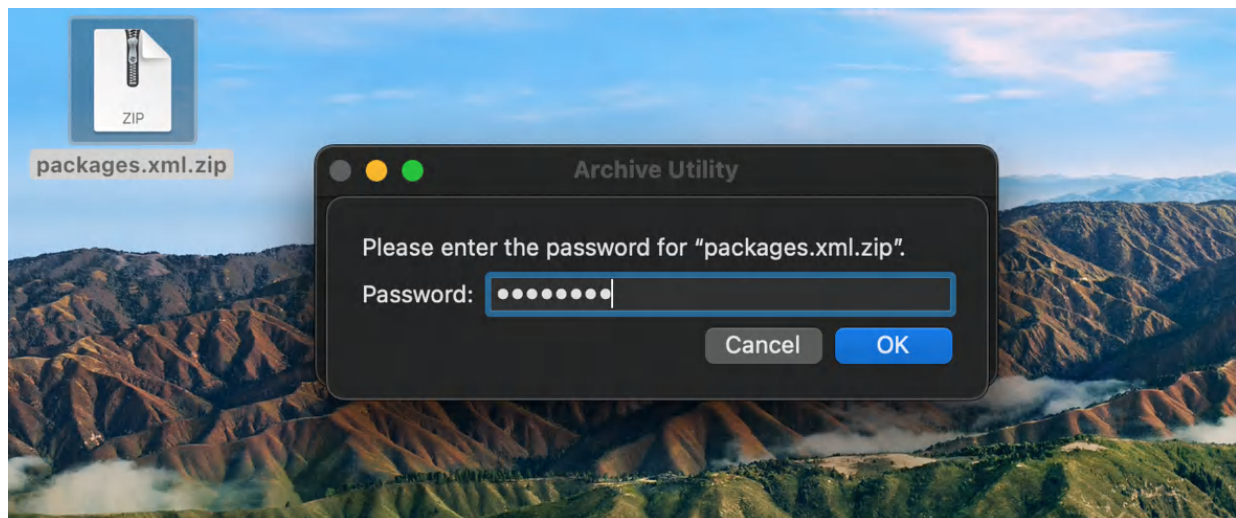
Forensic Images.zip will contain the following files:
(These files are separated on this website to make the large files easier to download.)

drive1.E01 | MD5: 2ccfa510ee28712b01544594f4fad721 | SHA1: 2baa0524e34a684e615061829b21d6b33cd906f8
drive2.E01 | MD5: 977365ee7ec72480469c1d915e5974d2 | SHA1: 154efb62fd5515000d89d4b254e921c62edf342a
drive3.E01 | MD5: 994512dbe6759ae7766b0e2fe4d9ec8a | SHA1: b5da7ecdbf904e1cca65007858d2ee0807284c19
drive4.E01 | MD5: 3e81eb20ded1ae8c6ffc936d5613aacd | SHA1: e1d34bf072df9df86d864df675667ccd02c6840e
drive5.E01 | MD5: 36d849ecddeb67e04911884346f93a15 | SHA1: 1b5891715365033800c632a4c19fd4770627dc2a

Chapter 5

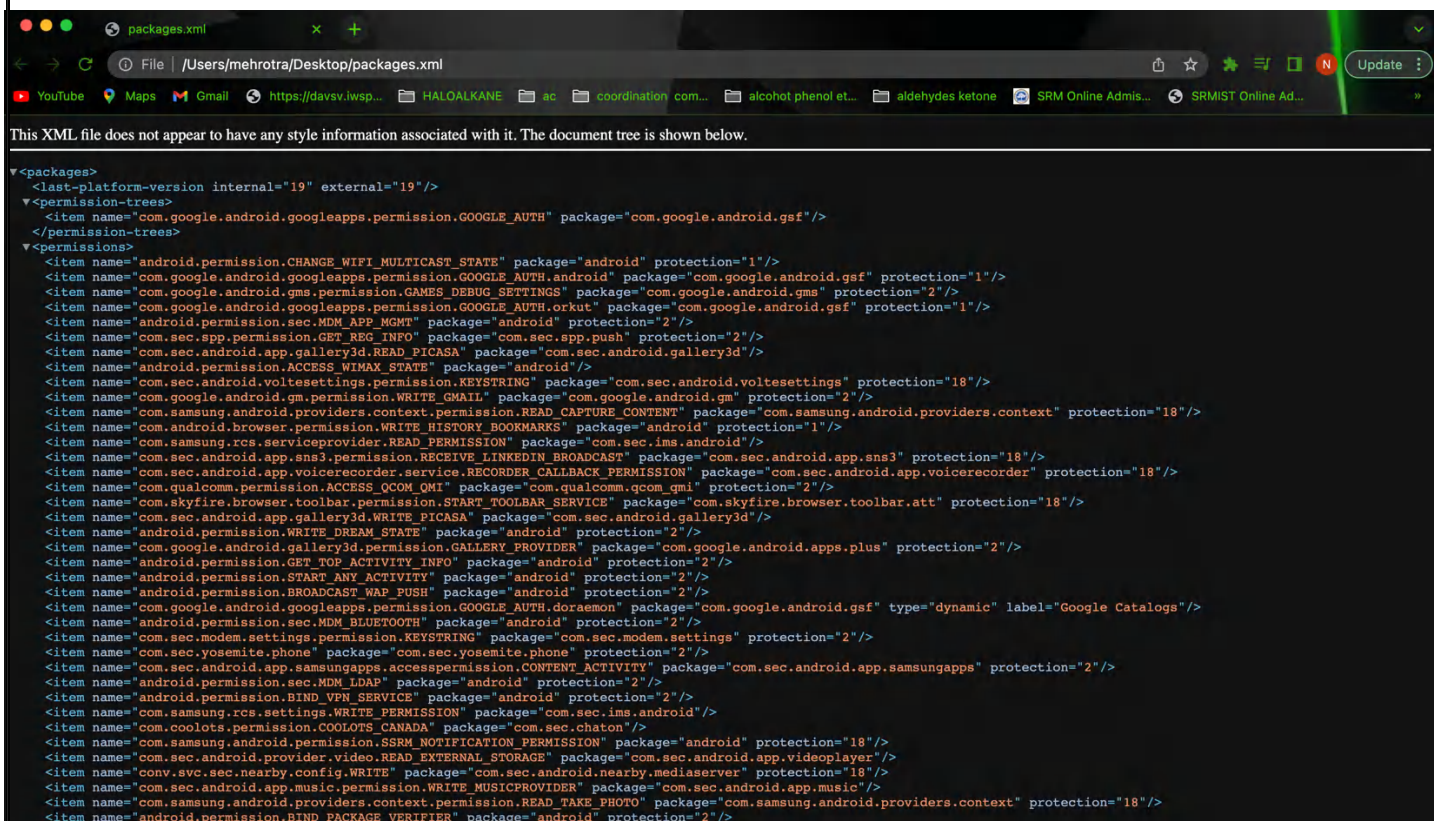


2. Enter the password hands-on to unzip the file

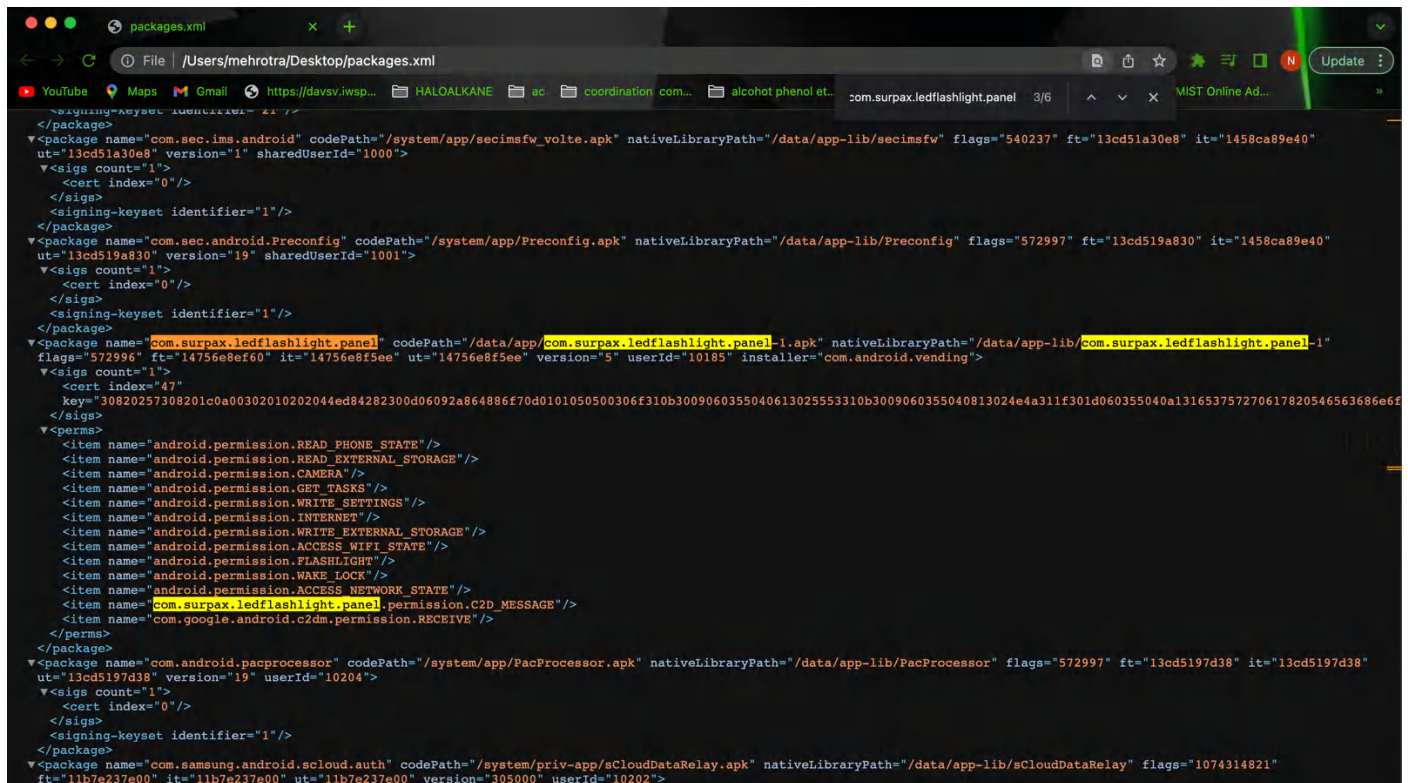


π

3. Open the file in a browser capable of displaying XML. Screenshot below shows the top portion of the XML file.



4.Scroll down through the packages.xml file to the section containing the permissions for com.surpax.ledflashlight.panel.



```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package name="com.sec.ims.android" codePath="/system/app/secimsfw_volte.apk" nativeLibraryPath="/data/app-lib/secimsfw" flags="540237" ft="13cd51a30e8" it="1458ca89e40"
    ut="13cd51a30e8" version="1" sharedUserId="1000">
    <sigs count="1">
      <cert index="0"/>
    </sigs>
    <signing-keyset identifier="1"/>
  </package>
  <package name="com.sec.android.Preconfig" codePath="/system/app/Preconfig.apk" nativeLibraryPath="/data/app-lib/Preconfig" flags="572997" ft="13cd519a830" it="1458ca89e40"
    ut="13cd519a830" version="19" sharedUserId="1001">
    <sigs count="1">
      <cert index="0"/>
    </sigs>
    <signing-keyset identifier="1"/>
  </package>
  <package name="com.surpax.ledflashlight.panel" codePath="/data/app/com.surpax.ledflashlight.panel-1.apk" nativeLibraryPath="/data/app-lib/com.surpax.ledflashlight.panel-1"
    flags="572996" ft="14756e8ef60" it="14756e8f5ee" ut="14756e8f5ee" version="5" userId="10185" installer="com.android.vending">
    <sigs count="1">
      <cert index="47"
        key="30820257308201c0a00302010202044ed84282300d06092a864886f70d0101050500306f310b3009060355040613025553310b3009060355040813024e4a311f301d060355040a131653757270617820546563686e6f"
      >
      </cert>
    </sigs>
    <perms>
      <item name="android.permission.READ_PHONE_STATE"/>
      <item name="android.permission.READ_EXTERNAL_STORAGE"/>
      <item name="android.permission.CAMERA"/>
      <item name="android.permission.GET_TASKS"/>
      <item name="android.permission.WRITE_SETTINGS"/>
      <item name="android.permission.INVERT_SCREEN"/>
      <item name="android.permission.WRITE_EXTERNAL_STORAGE"/>
      <item name="android.permission.ACCESS_WIFI_STATE"/>
      <item name="android.permission.FLASHLIGHT"/>
      <item name="android.permission.WAKE_LOCK"/>
      <item name="android.permission.ACCESS_NETWORK_STATE"/>
      <item name="com.surpax.ledflashlight.panel.permission.C2D_MESSAGE"/>
      <item name="com.google.android.c2dm.permission.RECEIVE"/>
    </perms>
  </package>
  <package name="com.android.pacprocessor" codePath="/system/app/PacProcessor.apk" nativeLibraryPath="/data/app-lib/PacProcessor" flags="572997" ft="13cd5197d38" it="13cd5197d38"
    ut="13cd5197d38" version="19" userId="10204">
    <sigs count="1">
      <cert index="0"/>
    </sigs>
    <signing-keyset identifier="1"/>
  </package>
  <package name="com.samsung.android.scloud.auth" codePath="/system/priv-app/sCloudDataRelay.apk" nativeLibraryPath="/data/app-lib/sCloudDataRelay" flags="1074314821"
    ft="11b7e237e00" it="11b7e237e00" ut="11b7e237e00" version="305000" userId="10202">
  </package>
</packages>
```

IV. OBSERVATIONS:

We can see that there are many permissions provided for com.surpax.ledflashlight.panel as listed below:

- I. To read the state of phone
- II. To read external Storage
- III. To access camera
- IV. To receive tasks from the Phone Operation Manager
- V. To write/change/update settings
- VI. To access the internet
- VII. To write external storage
- VIII. To access state of WiFi
- IX. To access flashlight
- X. To access Network State
- XI. To intimate lock status While it is normal for a flashlight program to have access to the camera for the purpose of accessing the flash.

PERMISSIONS:

```
▼<perms>
  <item name="android.permission.READ_PHONE_STATE"/>
  <item name="android.permission.READ_EXTERNAL_STORAGE"/>
  <item name="android.permission.CAMERA"/>
  <item name="android.permission.GET_TASKS"/>
  <item name="android.permission.WRITE_SETTINGS"/>
  <item name="android.permission.INTERNET"/>
  <item name="android.permission.WRITE_EXTERNAL_STORAGE"/>
  <item name="android.permission.ACCESS_WIFI_STATE"/>
  <item name="android.permission.FLASHLIGHT"/>
  <item name="android.permission.WAKE_LOCK"/>
  <item name="android.permission.ACCESS_NETWORK_STATE"/>
  <item name="com.surpax.ledflashlight.panel.permission.C2D_MESSAGE"/>
  <item name="com.google.android.c2dm.permission.RECEIVE"/>
</perms>
```

V. INFERENCES:

While it is normal for a flashlight program to have access to the camera:

(1) for the purpose of accessing the flash, one might question whether it is normal for a flashlight program to have access to the Internet and be able to (2) read (3) and write to (4) external storage media.

While it is common for flashlight apps to have access to the camera, the presence of additional permissions like internet access, read, and write access to external storage might raise some concerns about the app's behavior and its potential handling of user data. It is advisable to carefully review the app's permissions, privacy policy, and user reviews to make an informed decision before installing and using such apps.

EXERCISE 1-B: Examination of a SQLite database from a Mobile App

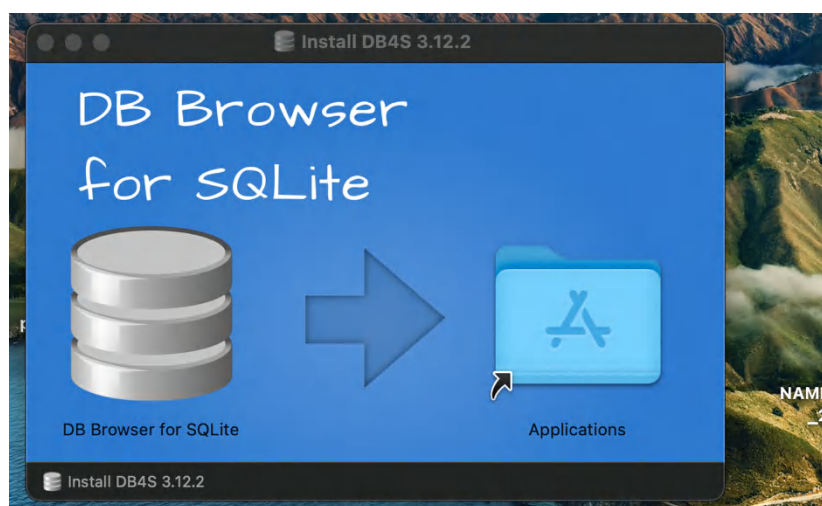
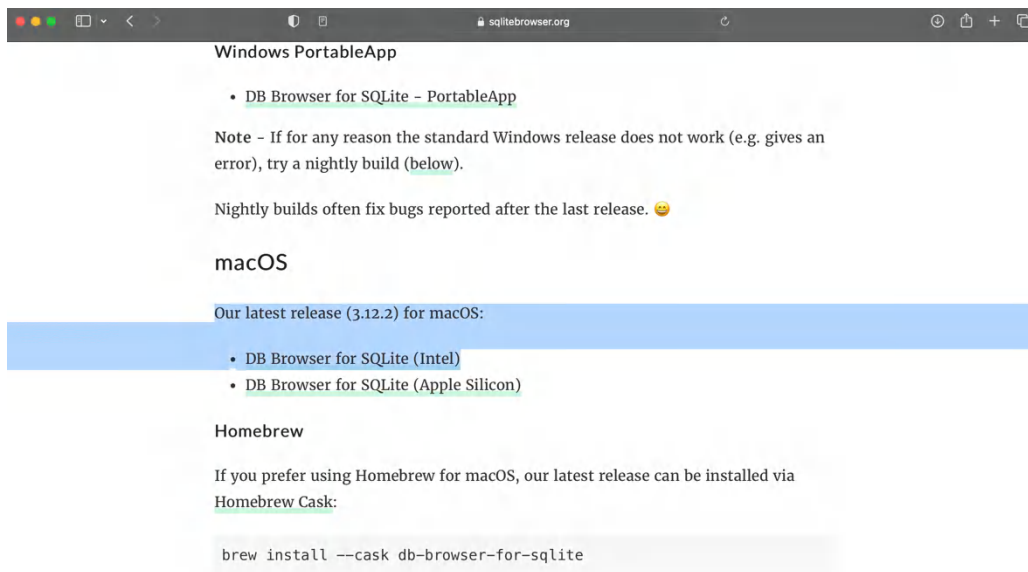
I. AIM: Use DB Browser for SQLite to analyse a SQLite database retrieved from BBM (BlackBerry Messenger) on an Android phone (Download the compressed file named “SQLiteDB.zip” from the <https://www.digitalforensicsworkbook.com/data-sets>)

II. TOOLS REQUIRED:

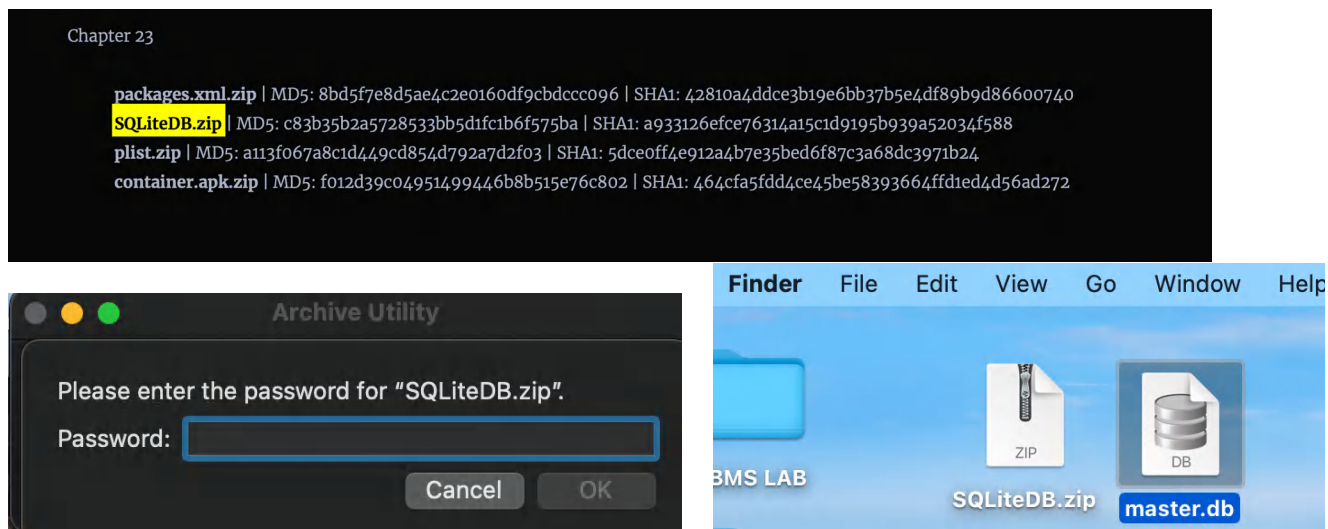
1. Product: DB Browser for SQLite
2. Manufacturer: Mauricio Piacentini, René Peinthor and Martin Kleusberg
3. Website: <http://sqlitebrowser.org>

III. STEP BY STEP PROCEDURES

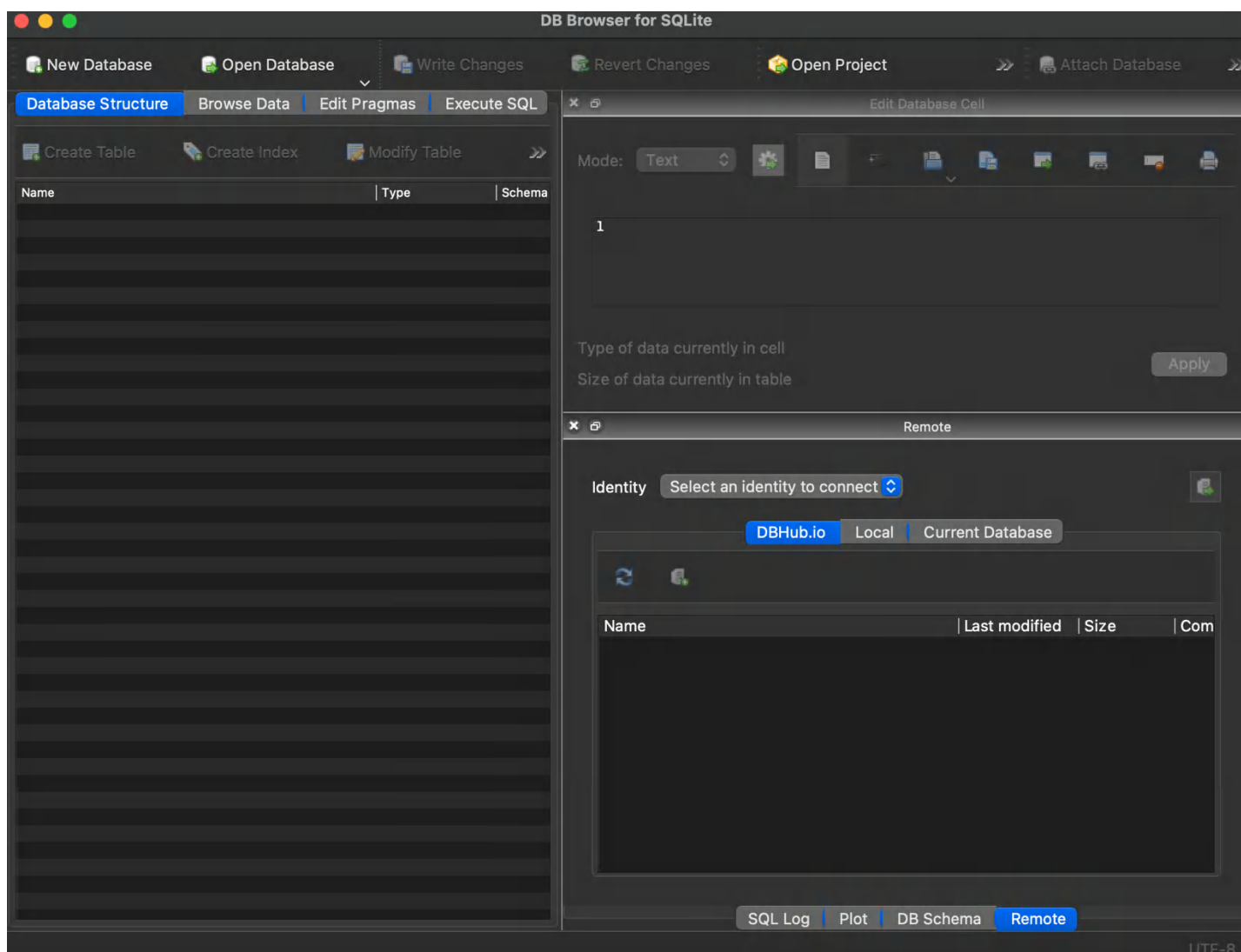
1. Download and install DB Browser for SQLite.



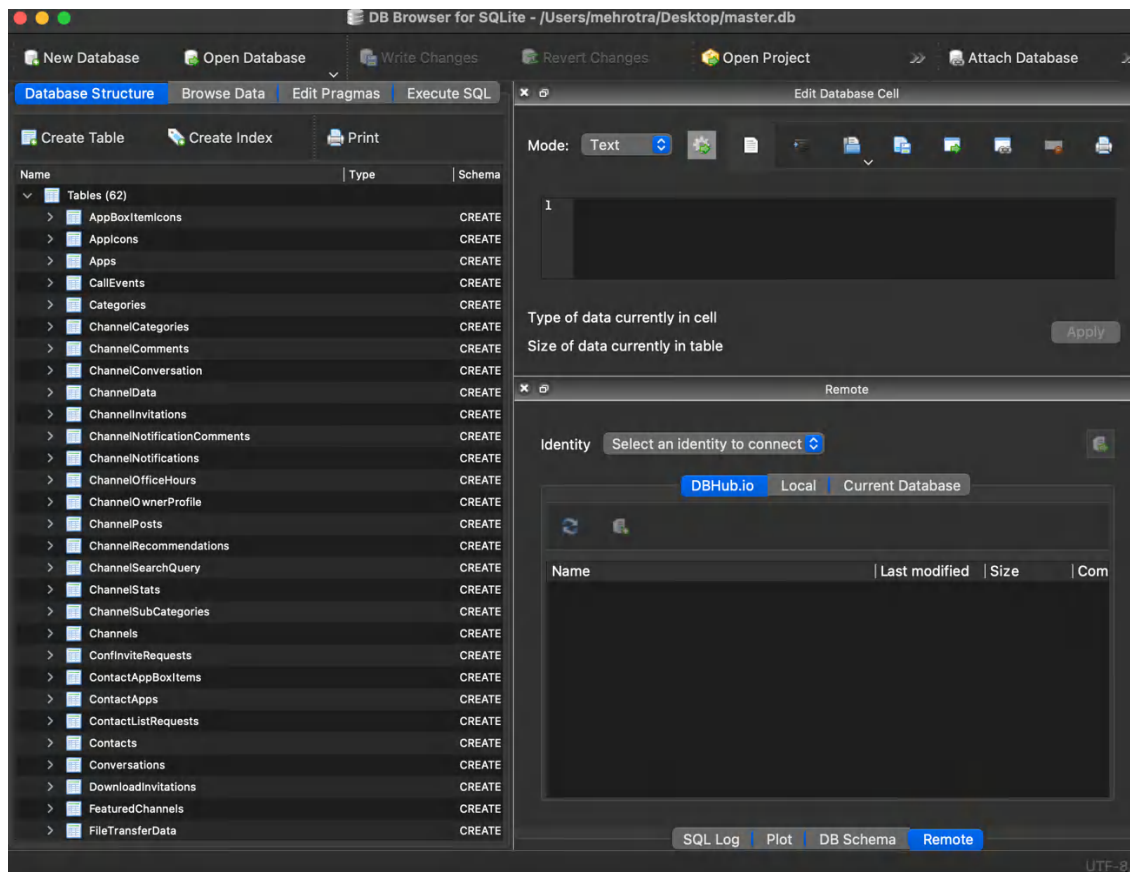
2. Download the compressed file named “SQLiteDB.zip” from the <https://www.digitalforensicsworkbook.com/data-sets>. Extract the database from the compressed file and place it on your desktop.



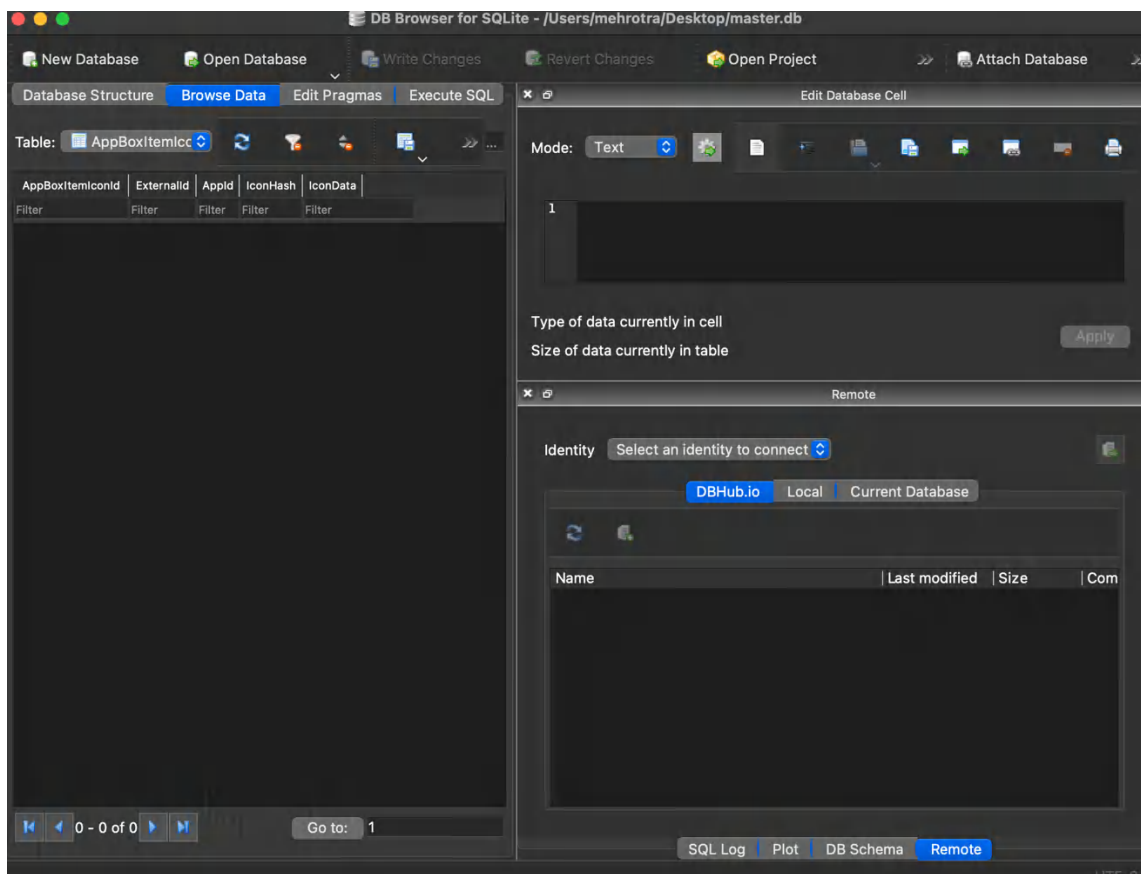
3. Launch DB Browser for SQLite.



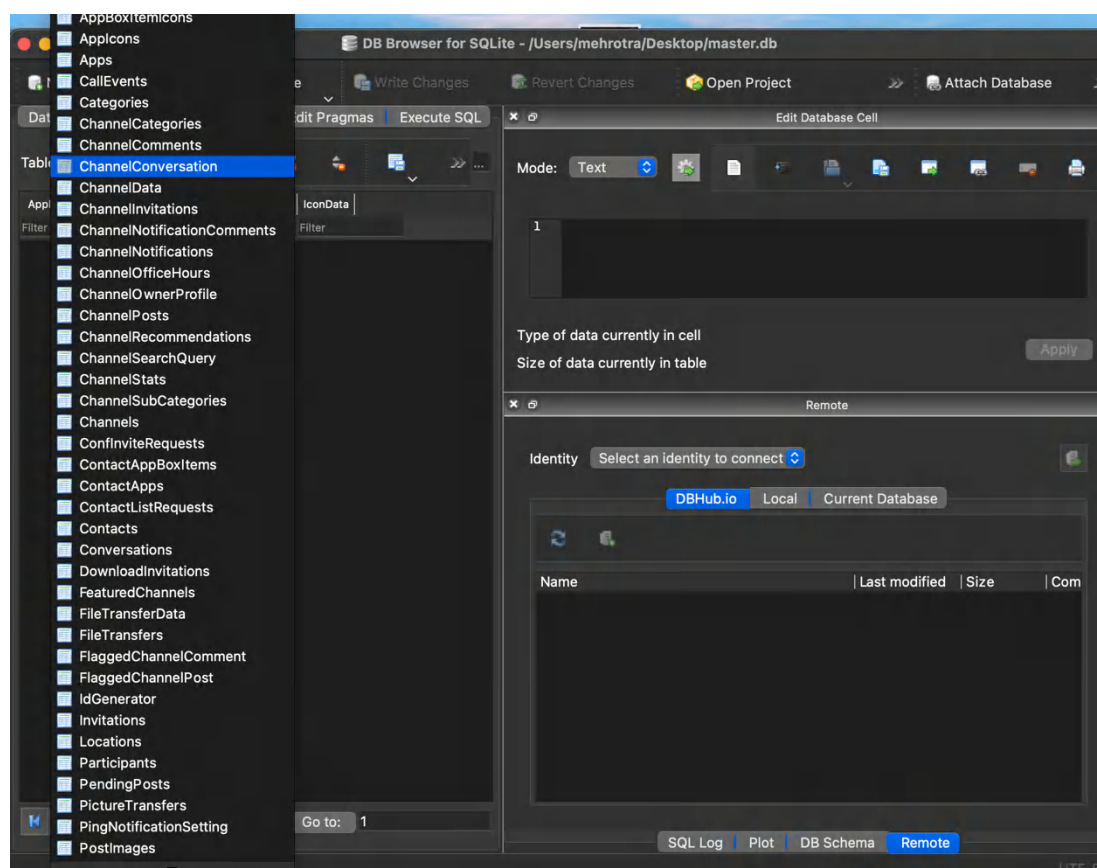
4. From the main menu of DB Browser for SQLite, select “File” and then select “Open Database.” Navigate to master.db. The file will be displayed as in the screenshot attached below.



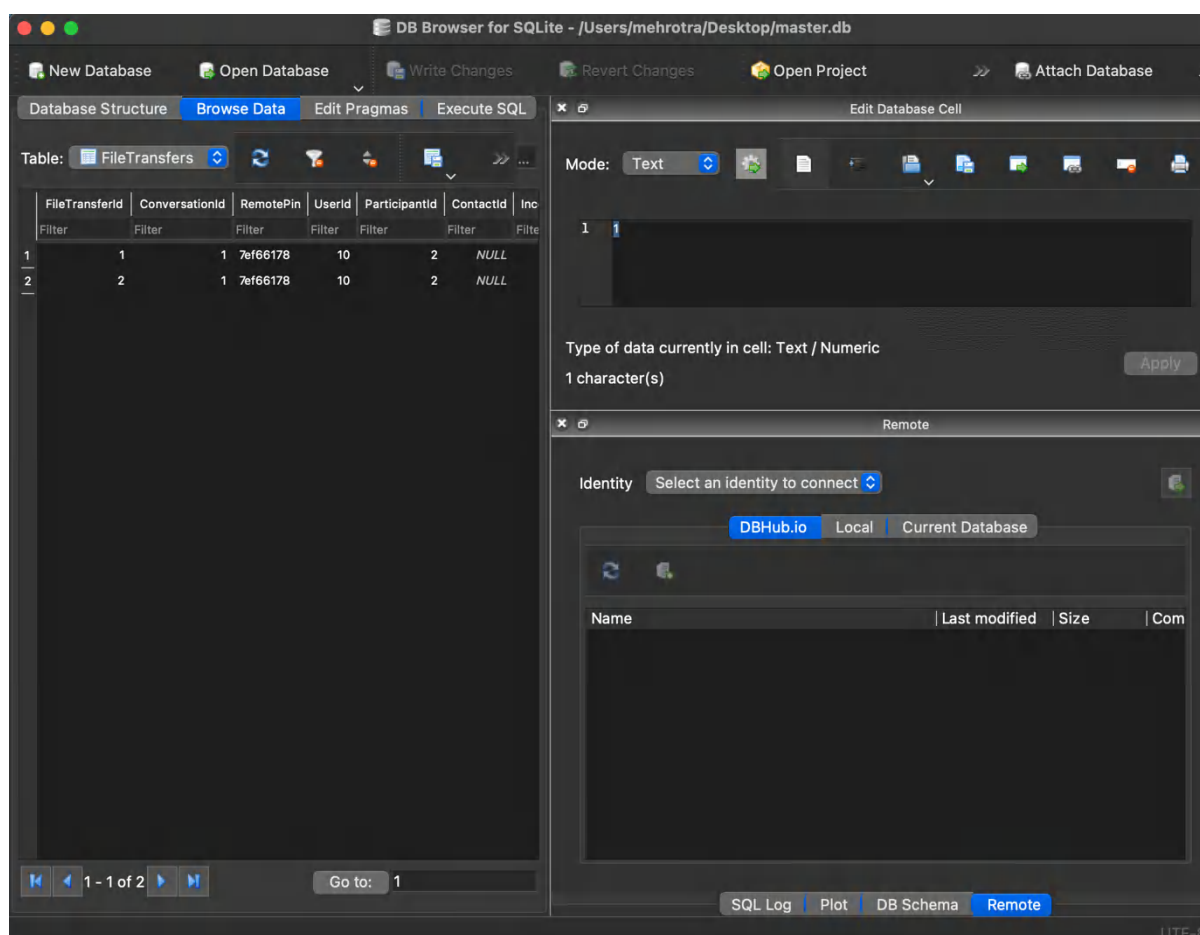
5. Upon opening the file, the tables and their schema will be listed i. Click the tab named “Browse Data” to view the data contained within the tables.



6. Click on the Table pull-down menu to see the list of tables in the SQLite database.



7. Go down to the table named “File Transfers.” And check its data i.e. how many files were transferred, what were there names to whom were they sent.



IV. OBSERVATIONS:

| | FileTransferId | ConversationId | RemotePin | UserId | ParticipantId | ContactId | Inc |
|---|----------------|----------------|-----------|--------|---------------|-----------|--------|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 | 1 | 7ef66178 | 10 | 2 | NULL | |
| 2 | 2 | 1 | 7ef66178 | 10 | 2 | NULL | |

In the “File Transfer” table, there are records of two file transfers shown in screenshot attached below. The first file i.e. 1406832981515.jpg, which was a picture in phone’s camera store, was sent to User ID 10. The second file, named 7ef66178.jpg, was received from the person with user ID 10.

V. INFERENCE:

DB Browser for SQLite was used to analyze a SQLite database retrieved from BBM (BlackBerry Messenger) on an Android phone. The file transfers made were observed using the “File transfer” table of DB Browser for SQLite.

EXERCISE 1-C:

I. AIM: Use pList Editor Pro to examine the contents of a .plist retrieved from an iOS device during a logical acquisition

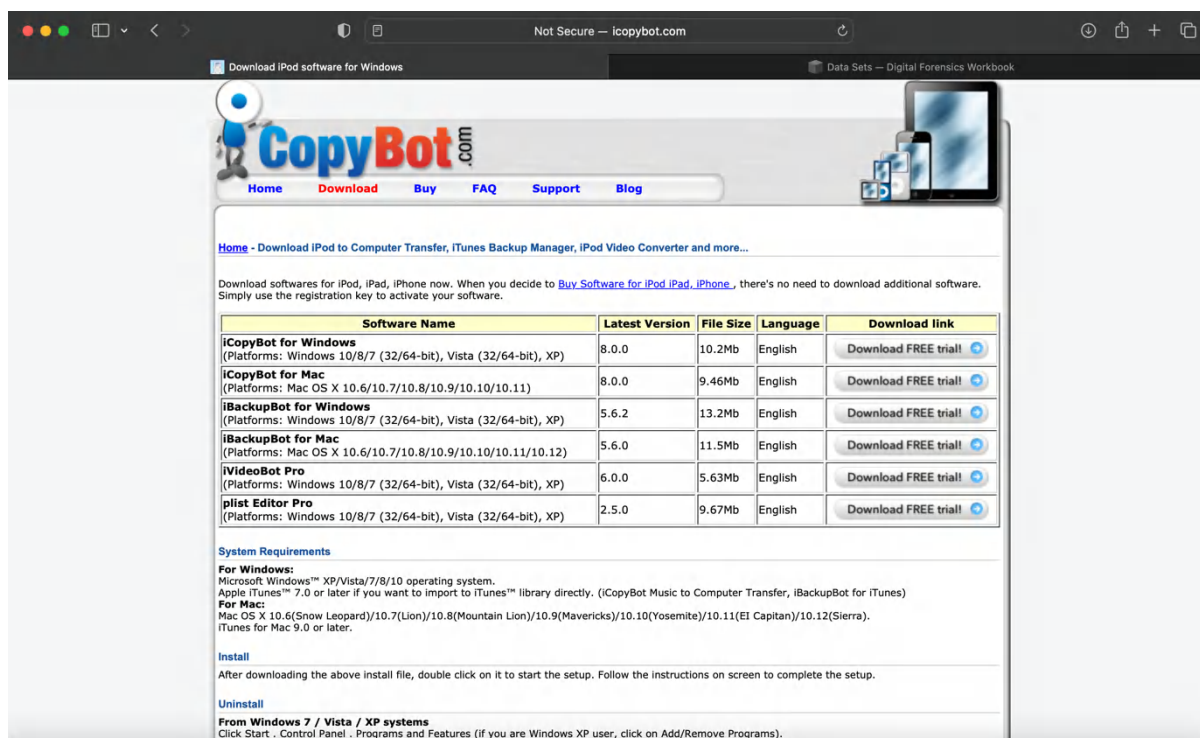
(Download the file named “plist.zip” from the <https://www.digitalforensicsworkbook.com/datasets>)

II. TOOLS USED:

1. Product: pList Editor Pro
2. Manufacturer: VOWSoft, Ltd.
3. Web site: <http://www.icopybot.com/plist-editor.htm>

III. STEP BY STEP PROCEDURE:

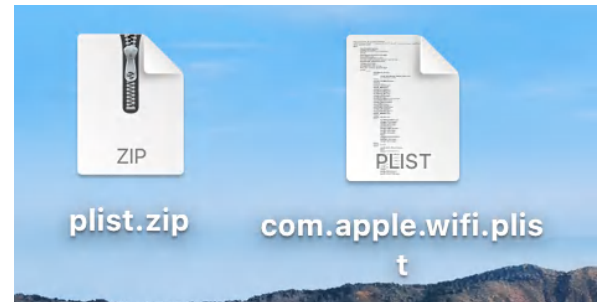
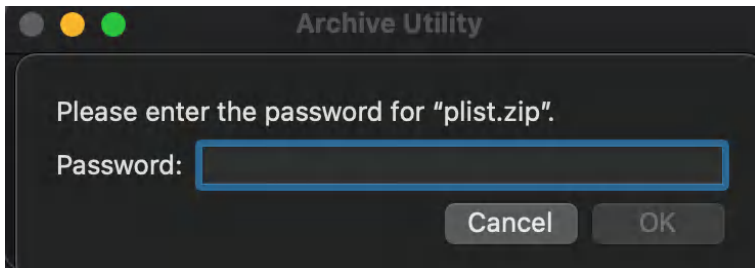
1. Download and install pList Editor Pro.



2. Download the file named “plist.zip” from the Digital Forensics Workbook web site and extract the contents to your desktop.

Chapter 23

packages.xml.zip | MD5: 8bd5f7e8d5ae4c2e0160df9cbdcc096 | SHA1: 42810a4ddce3b19e6bb37b5e4df89b9d86600740
SQLiteDatabase.zip | MD5: c83b35b2a5728533bb5d1fc1b6f575ba | SHA1: a933126efce76314a15c1d9195b939a52034f588
plist.zip | MD5: a113f067a8c1d449cd854d792a7d2f03 | SHA1: 5dce0ff4e912a4b7e35bed6f87c3a68dc3971b24
container.apk.zip | MD5: f012d39c04951499446b8b515e76c802 | SHA1: 464cfa5fdd4ce45be58393664ffd1ed4d56ad272

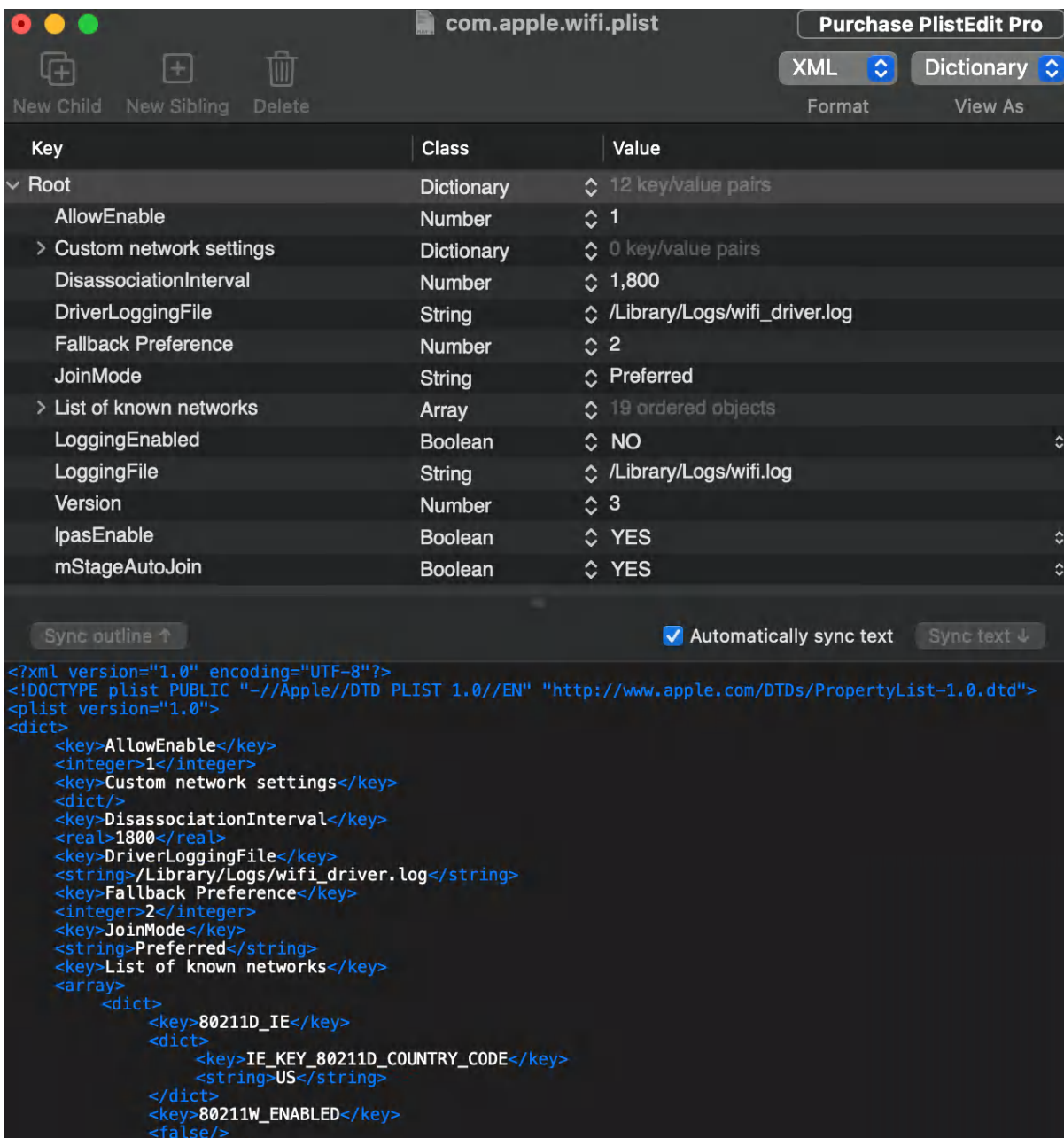


3. Launch pList Editor Pro.

4. From the main menu select "File" and then select "Open."

Browse to the com.apple.wifi.plist file on your desktop.

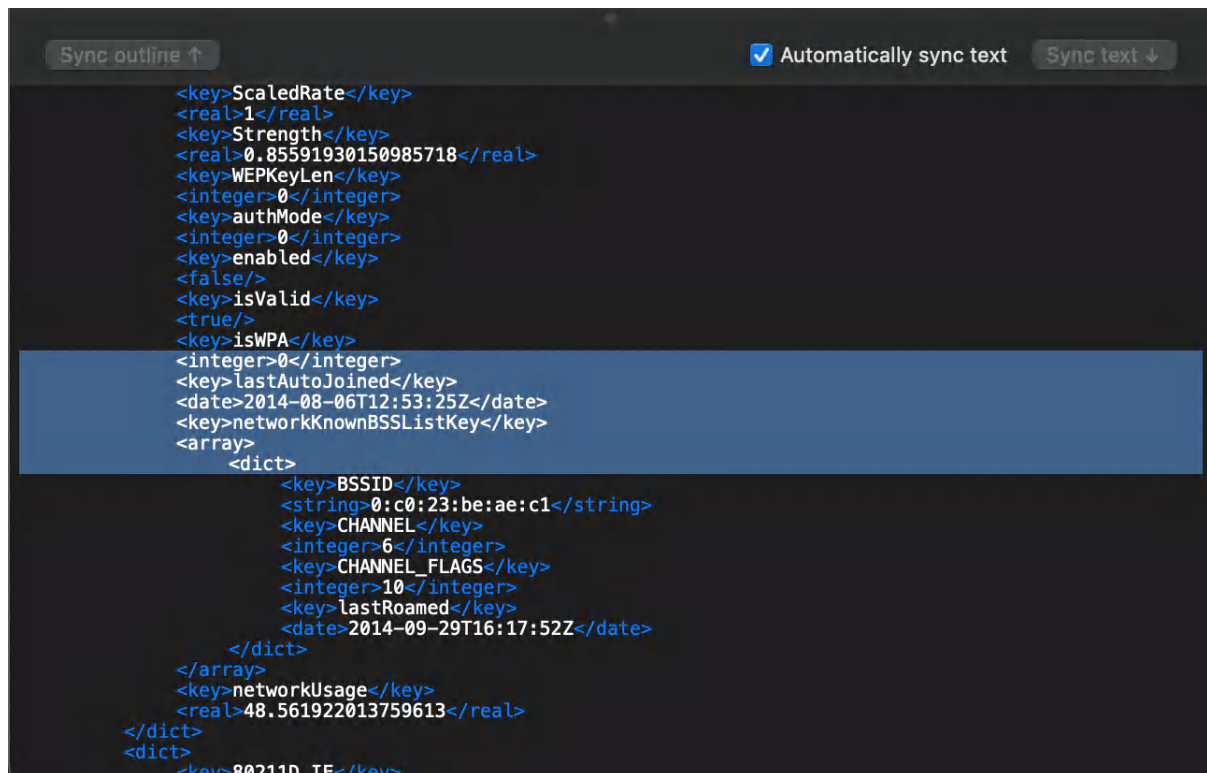
5. Upon opening the file, the results will appear as shown in the screenshot attached below.



6. Based on a review of the file's contents, what is contained within this plist file?

This file contains a list of all of the wireless access points to which the iOS device connected.

7. Search through the plist file and identify the last time the iOS device connected to the wireless access point associated with Starbucks or Barnes and Noble, *i.e.*, attwifi

A screenshot of a code editor displaying a plist file. The editor has a dark theme. At the top, there are two buttons: "Sync outline ↑" and "Automatically sync text" (checked), followed by a "Sync text ↓" button. The code is XML-based, representing a list of network connections. A specific entry is highlighted in blue, showing details for a connection to "attwifi". The highlighted entry includes keys for "lastAutoJoined" (2014-08-06T12:53:25Z), "networkKnownBSSListKey" (an array of dictionaries), and "networkUsage" (48.561922013759613). The dictionary in the array shows "BSSID" (0:c0:23:be:ae:c1), "CHANNEL" (6), "CHANNEL_FLAGS" (10), and "lastRoamed" (2014-09-29T16:17:52Z).

```
<key>ScaledRate</key>
<real>1</real>
<key>Strength</key>
<real>0.85591930150985718</real>
<key>WEPKeyLen</key>
<integer>0</integer>
<key>authMode</key>
<integer>0</integer>
<key>enabled</key>
<false/>
<key>isValid</key>
<true/>
<key>isWPA</key>
<integer>0</integer>
<key>lastAutoJoined</key>
<date>2014-08-06T12:53:25Z</date>
<key>networkKnownBSSListKey</key>
<array>
  <dict>
    <key>BSSID</key>
    <string>0:c0:23:be:ae:c1</string>
    <key>CHANNEL</key>
    <integer>6</integer>
    <key>CHANNEL_FLAGS</key>
    <integer>10</integer>
    <key>lastRoamed</key>
    <date>2014-09-29T16:17:52Z</date>
  </dict>
</array>
<key>networkUsage</key>
<real>48.561922013759613</real>
</dict>
<dict>
  <key>80211D_IF</key>
```

IV. OBSERVATIONS:

The iOS device last connected to attwifi on August 6, 2014 at 12:53:25 Zulu.

V. INFERENCE:

The following inferences can be drawn about the last time the iOS device connected to the wireless access point associated with Starbucks or Barnes and Noble (attwifi):

i. SSID: The SSID (Service Set Identifier) of the network is provided as "attwifi". This is the identifier for the wireless network associated with Starbucks or Barnes and Noble.

ii. Last Auto Joined: The "lastAutoJoined" key indicates the date and time when the device last automatically joined the attwifi network. In this case, the date is specified as "2014-08-06" and the time as "12:53:25" in UTC (Zulu) time zone.

iii. Network Known BSS List: The "networkKnownBSSListKey" key contains an array with information about the known Base Station Subsystem (BSS) of the attwifi network. It includes the following details:

a.BSSID: The BSSID (Basic Service Set Identifier) of the access point associated with Starbucks or Barnes and Noble is given as "0:c0:23:be:ae:c1".

b.CHANNEL: The channel number of the access point is specified as 6.

c.CHANNEL_FLAGS: The channel flags associated with the access point are represented by the integer value 10.

d.Last Roamed: The "lastRoamed" key indicates the date and time when the device last roamed (switched) to this particular access point. In this case, the date is specified as "2014-09-29" and the time as "16:17:52" in UTC (Zulu) time zone.

Based on this information, the inference is that the iOS device last connected to the wireless access point associated with Starbucks or Barnes and Noble (attwifi) on August 6, 2014, at 12:53:25 (UTC time). Additionally, it provides details about the specific access point (BSSID) and the channel information.

EXERCISE 1-D : Network Traffic Identification

I. AIM:

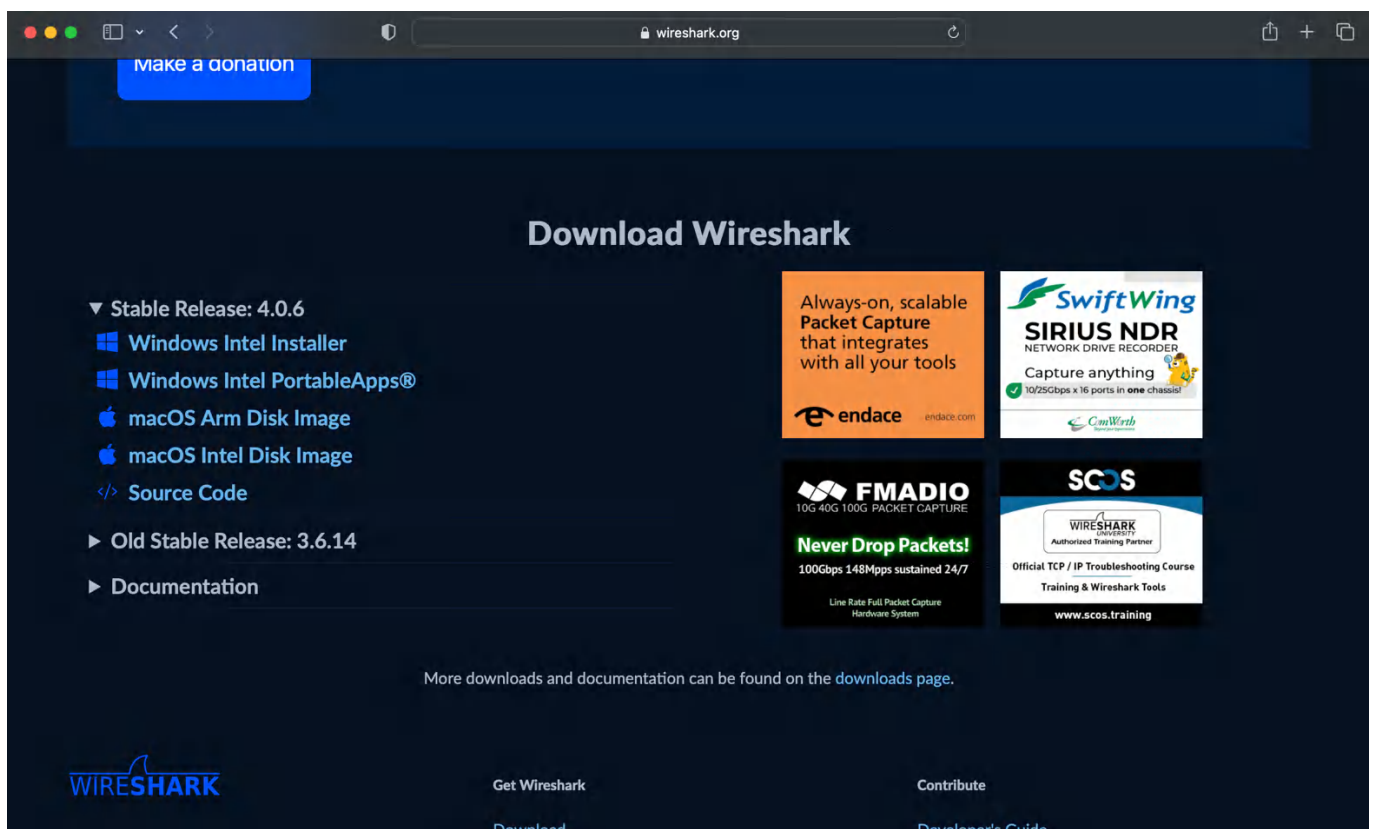
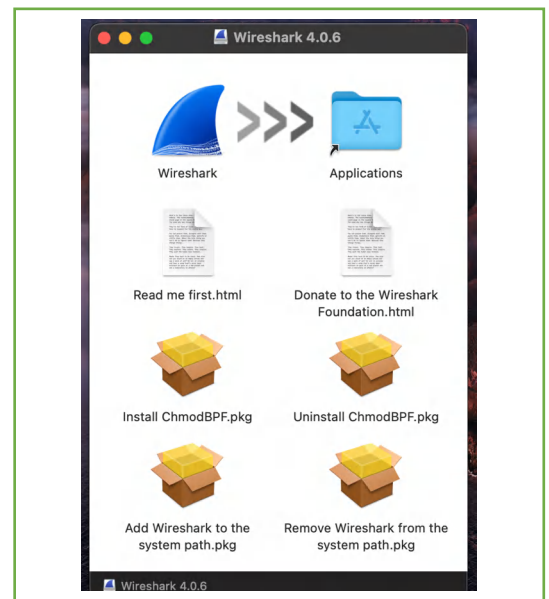
Use Wireshark to review the results of a previously recorded Ping, filter the results, and export results to a new file. (Download the following packet capture from the [https:// www.digitalforensicsworkbook.com/data-sets: ping.pcapng](https://www.digitalforensicsworkbook.com/data-sets/ping.pcapng))

II. TOOLS USED:

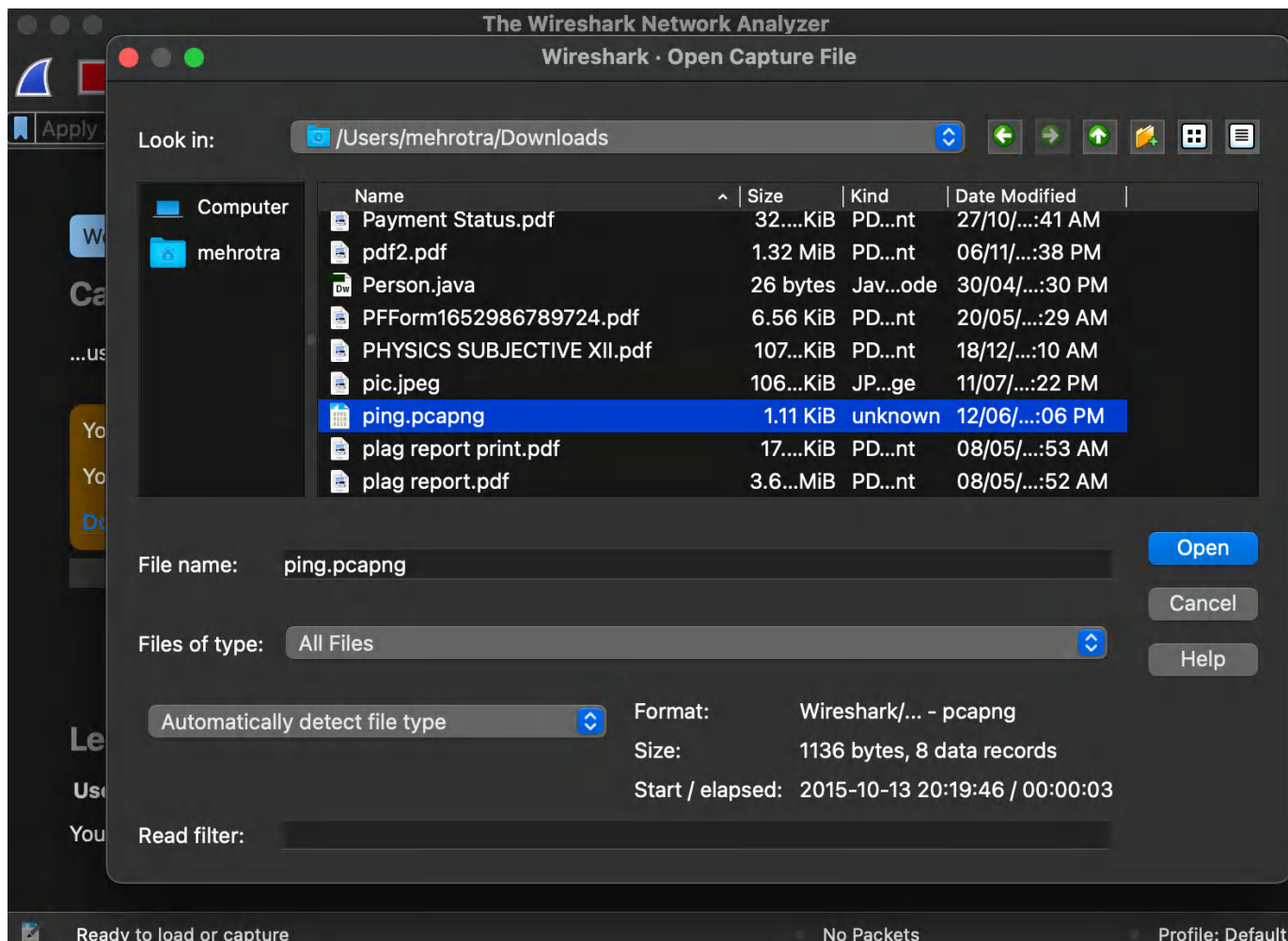
- 1.Product: Wireshark
2. Manufacturer: Wireshark Foundation
3. Web site: <https://www.wireshark.org>

III. STEP BY STEP PROCEDURE:

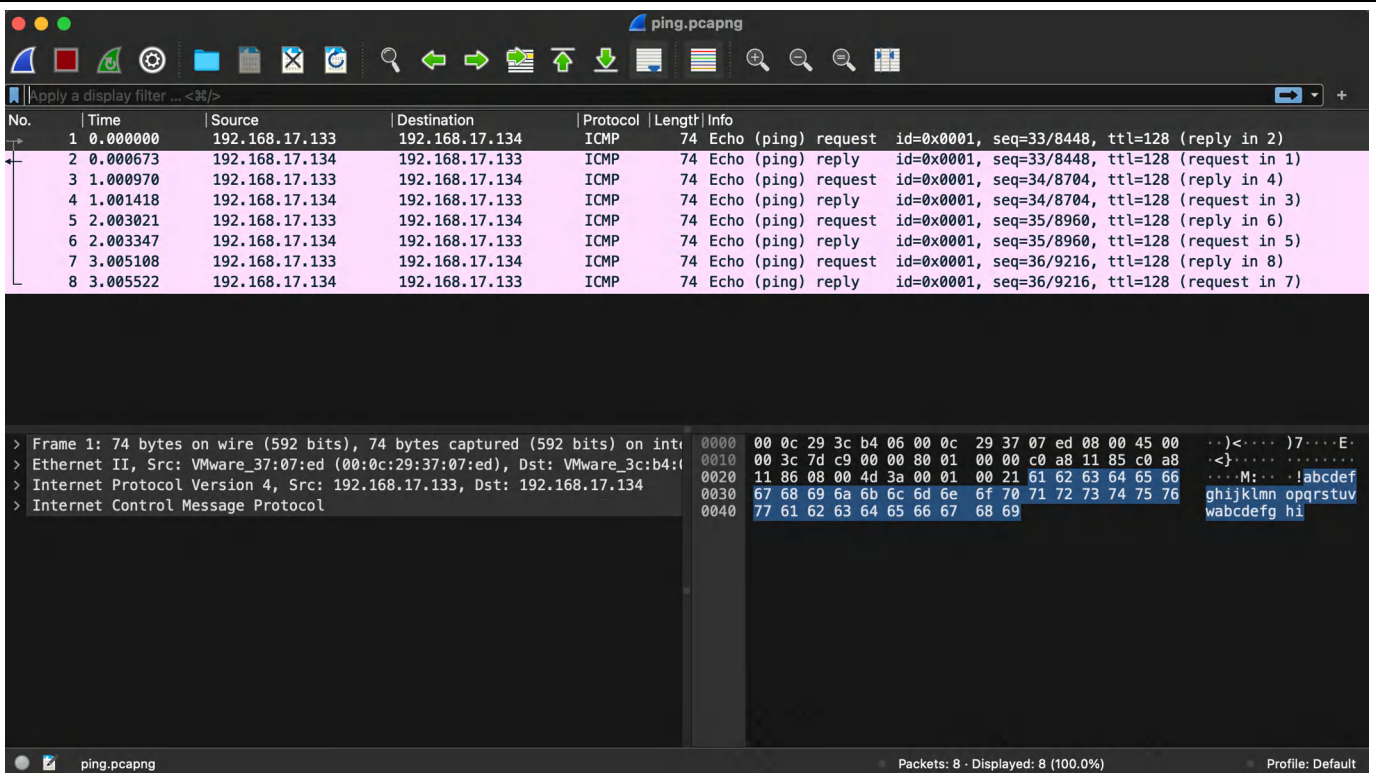
1. Download and install Wireshark and its dependencies.
2. Download the following packet capture from the [https:// www.digitalforensicsworkbook.com / data-sets: ping.pcapng](https://www.digitalforensicsworkbook.com/data-sets/ping.pcapng).
3. Launch Wireshark. You will see a screen identical to the one shown in screenshot attached below.



DNS-query-response.pcapng | MD5: 12d899bb258d2ec2b8ef652b895a2e8c | SHA1: 7ba1137e0a8d5a7b29eb7bdfc9c6b9f204af77be
 IntenseScan.pcapng | MD5: ca272d83359a964e4e52762072cac966 | SHA1: 95a03aba37c169fd912974146e0fa27405f7e3bd
 ping.pcapng | MD5: 3c031287a8e1foe957f177abb324d28b | SHA1: f8720a85abfcc1bcc5b709fba524b6976f1f002c
 QuickScan.pcapng | MD5: 4f25c44ef2d55ab604618b519a254a48 | SHA1: 38e325f1e05493407bef13f345f1ef80c84937
 Threeway-Handshake.pcapng | MD5: 8fe26af8ec6af3f30b7379ff85d9aa54 | SHA1: 568f1c6c0783abf982c1f01cc41ed05ef131520e
 unknown.pcapng | MD5: de56cee8638d2fb55be0d3e76a497a23 | SHA1: 4cf54b1ab637401dad12b7d55bd94d8ab97f6d8
 Website-Visit.pcapng | MD5: ofd58113e5473a8e02210f93e86a7540 | SHA1: 86df9be6662d573a380080debea1bb966dcfadf1



4. From the main menu select “File,” select “Open,” and then open the file named ping.pcapng. The packet will be opened and displayed as shown in Figure 22-2. Clicking on a packet in the Packet List Pane will display the details for that packet in the Packet Details Pane and Packet Bytes Pane.



- Based on the Packet List Pane, how many packets were sent back and fourth?

Eight packets were sent back and fourth!. (Four were requests and four were replies. By default, Ping on Windows will send four requests. Ping on Mac OS and Linux will send requests continuously until stopped.)

- What was the source of the Pings and what was the target of them?

The source of the Pings was the host with the IP address of "192.168.17.133" and the target was 192.168.17.134#. (In order to obtain this information, a packet with an "Echo (ping) request" must be reviewed.)

- What protocol is used with Ping?

ICMP (Internet Control Message Protocol) is used with Ping.

- What are the sequence numbers associated with the Pings? There are four sequence numbers associated with each pair (request/reply).

They are:

33, 34, 35, and 36%. (Wireshark presents the numbers in both Little Endian (LE) and Big Endian (BE) formats.)

- How many hops can the Ping pass before it is dropped by a router?

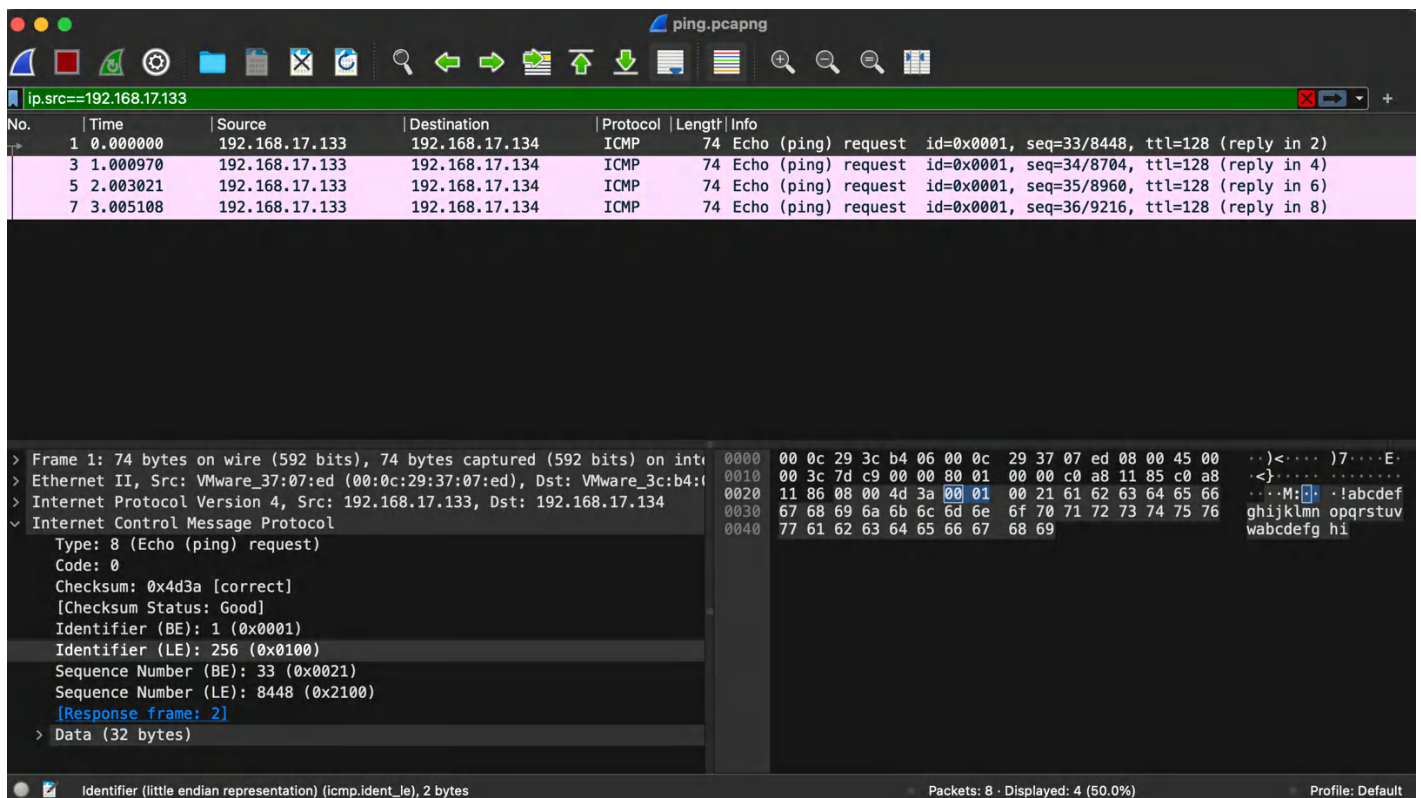
Pings can travel through 128 hops/routers before they are dropped.

10. How long did it take for the first reply/request to traverse the network for the first PING? It took 0.000673 seconds for the first reply/request to reach the target and return'.

11. In the textbox next to the word “Filter” enter the following text to filter the results of a particular source IP address:

ip.src==192.168.17.133

12. After entering the text, press the “Apply” button. The results will be identical to those shown in screenshot attached below.

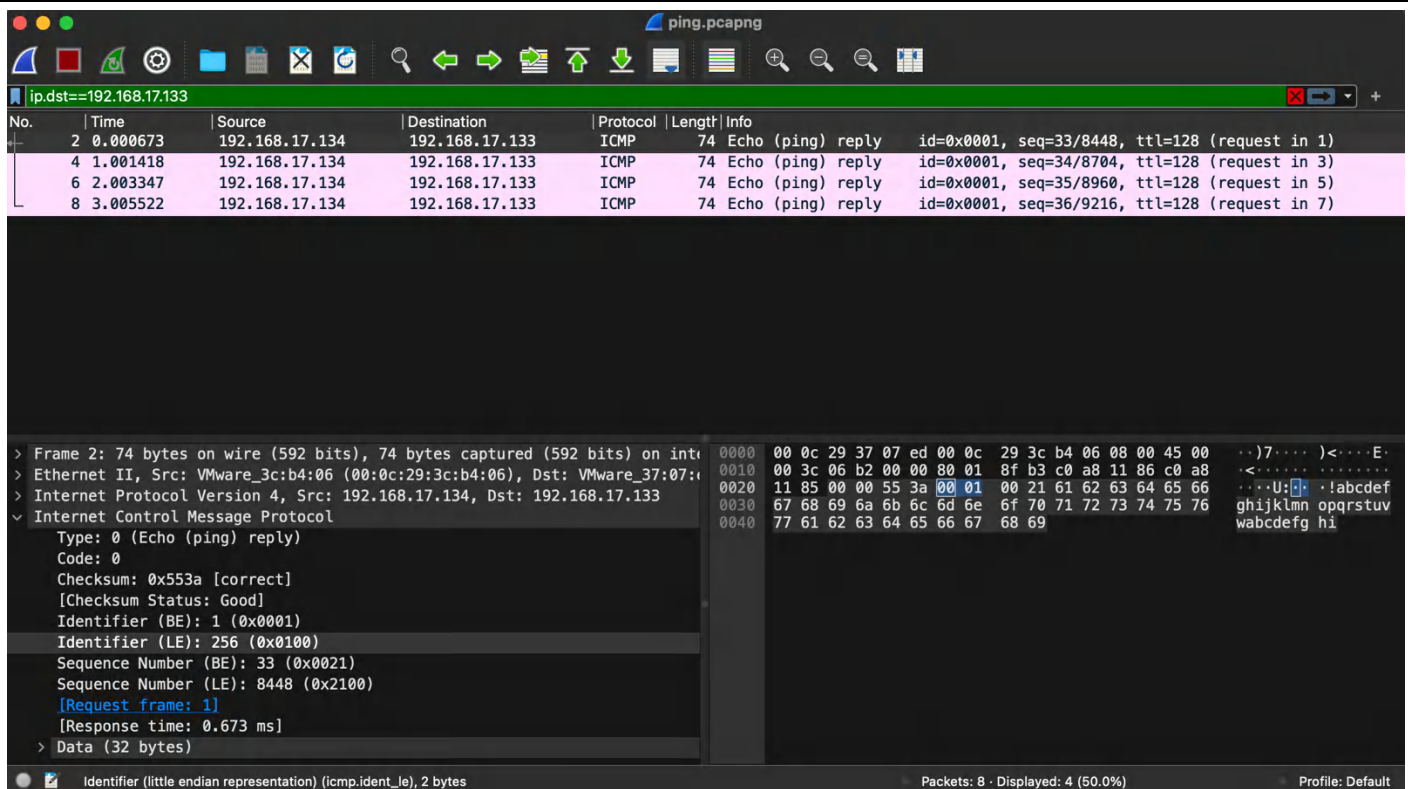


13. How many packets are displayed and what are their relative numbers? Four packets are displayed. They have the relative numbers of 1, 3, 5, and 7.

14. Based on the filter, do the displayed packets show requests or replies? The displayed packets show only the requests and not the replies.

15. In the textbox next to the word “Filter” enter the following text to filter the results of a particular source IP address: *ip.dst==192.168.17.133*

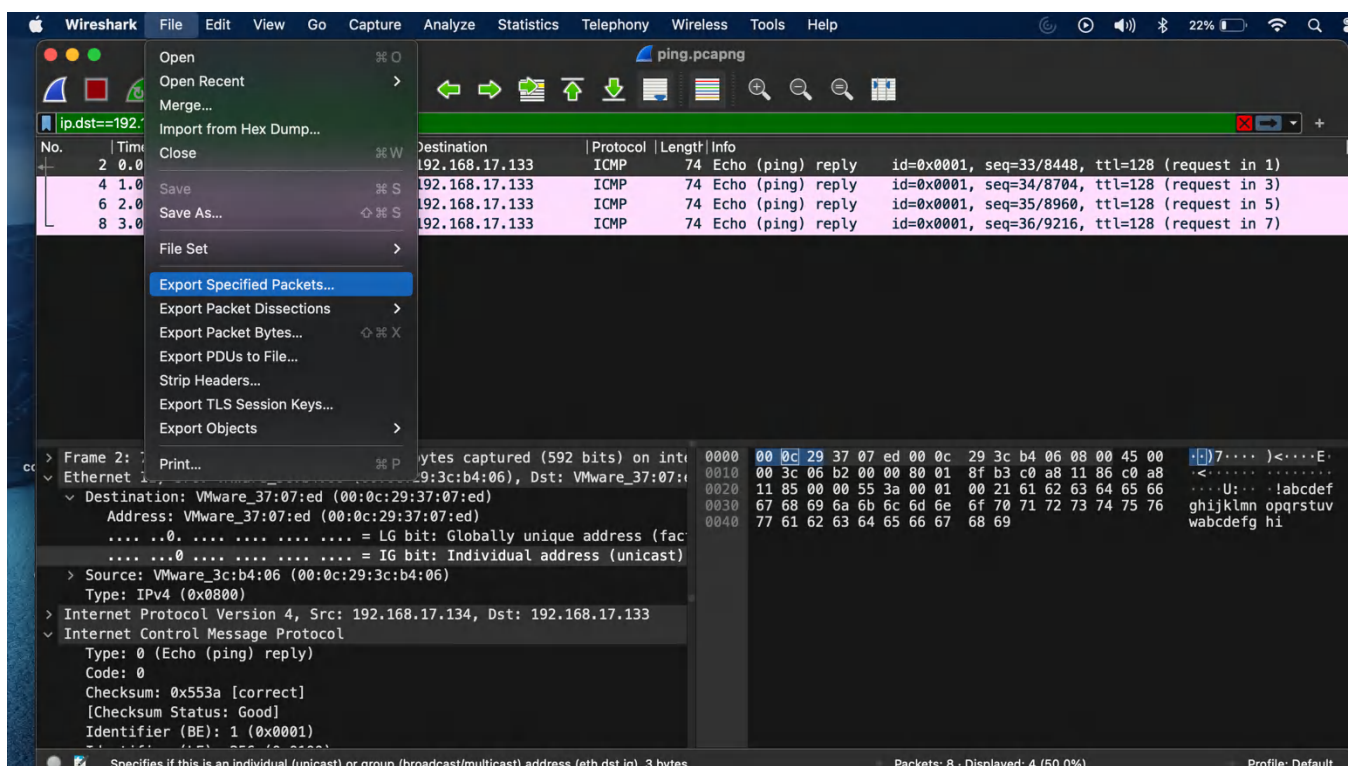
16. After entering the text, press the “Apply” button. The results will be identical to those shown in screenshot attached below.



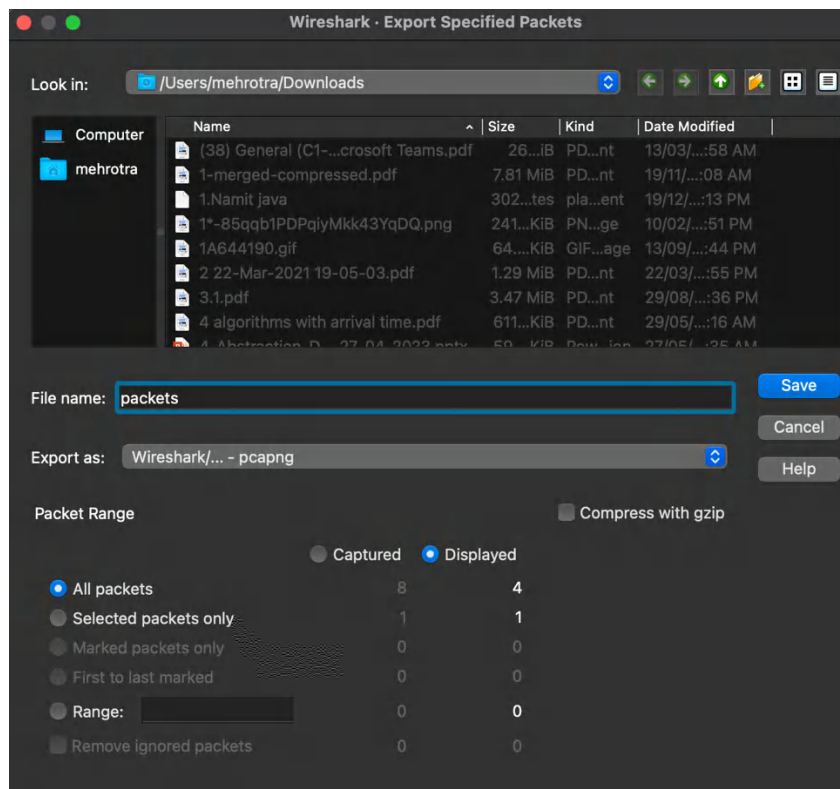
17. How many packets are displayed and what are their relative numbers?
Four packets are displayed. They have the relative numbers of 2, 4, 6, and 8.

18. Based on the filter, do the displayed packets show requests or replies?
The displayed packets show only the replies and not the requests.

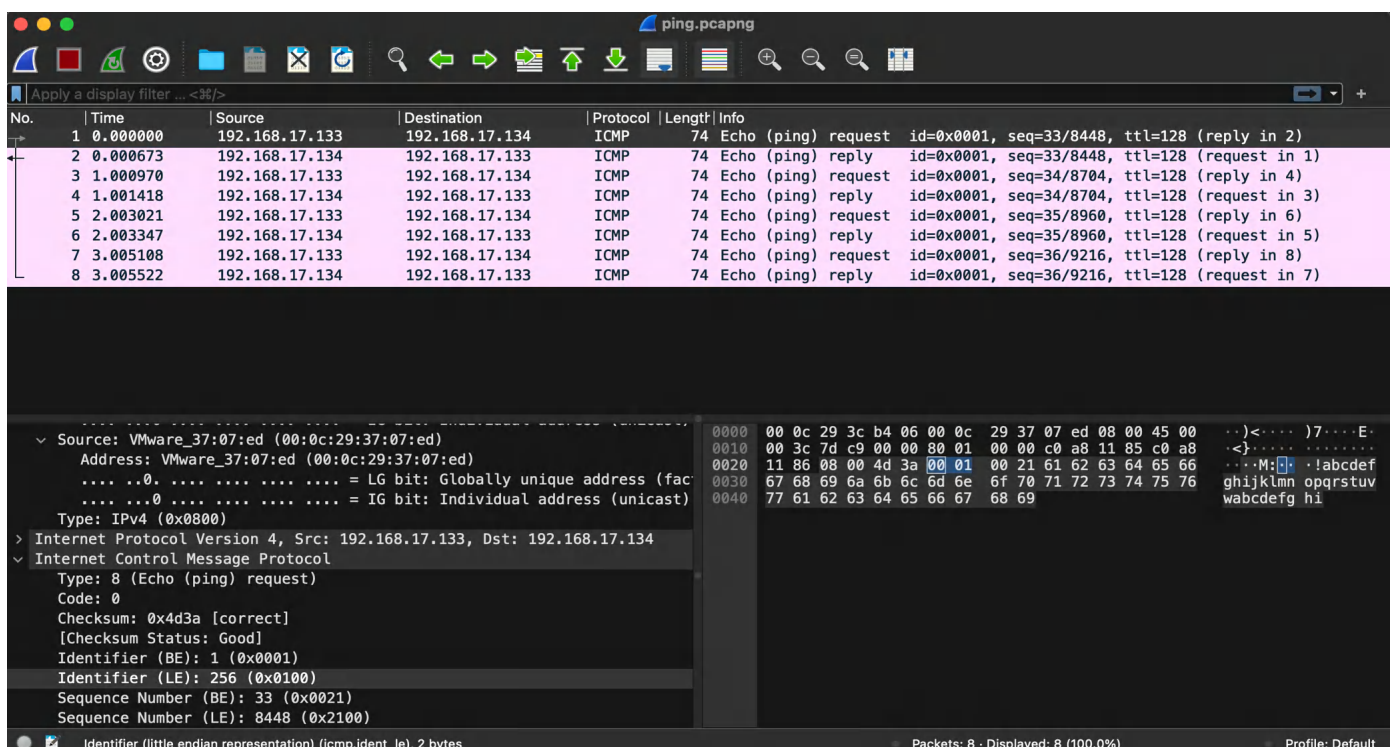
19. To save these packets in their own packet capture file, select the “File” menu and then select “Export Specified Packets” from the pull-down menu. The “Wireshark Export Specified Packets” dialog box will appear as shown in screenshot below.



20. Enter a name for the file. Ensure the radio button next to “Displayed” is checked. Click the “Save” button. (Note: If you wanted to specify the packets to be saved by range of packets, the “Range” option can be used where packets can be specified by an individual number, *e.g.*, 3, as a consecutive range with a dash, *e.g.*, 4–7, or as a split group using commas as a separator, *e.g.*, 1, 3, 7.



21. After saving the packet capture, click the “Clear” button next to the filter to remove the filter.



IV. OBSERVATIONS:

For *ip.src==192.168.17.133*, four packets are displayed. They have the relative numbers of 1, 3, 5, and 7. The displayed packets show only the requests and not the replies.

For *ip.dst==192.168.17.133*, four packets are displayed. They have the relative numbers of 2, 4, 6, and 8. The displayed packets show only the replies and not the requests.

V. INFERENCE:

The following inferences can be drawn:

1. Requests from source IP 192.168.17.133:

a. Four packets with relative numbers 1, 3, 5, and 7 are displayed.

b. These packets represent ping requests sent from the source IP address 192.168.17.133.

c. Since only the requests are displayed, it indicates that the corresponding replies are not shown.

2. Replies to destination IP 192.168.17.133:

a. Four packets with relative numbers 2, 4, 6, and 8 are displayed.

b. These packets represent ping replies received by the destination IP address 192.168.17.133.

c. Since only the replies are displayed, it indicates that the corresponding requests are not shown.

In summary, the ping data reveals that there were four ping requests sent from the IP address 192.168.17.133, and four ping replies received by the IP address 192.168.17.133. The relative numbers assigned to the packets help identify the correspondence between the requests and replies, even though the actual content of the requests and replies is not provided in the given information.

EXERCISE 1-E : Network Traffic Identification:

DNS Query

I. AIM: Use Wireshark to review the results of a previously

recorded DNS query (Download the following packet capture from the <https://www.digitalforensicsworkbook.com/data-sets: DNS-queryresponse.pcapng>)

II. TOOLS USED:

- 1.Product: Wireshark
2. Manufacturer: Wireshark Foundation
3. Web site: <https://www.wireshark.org>

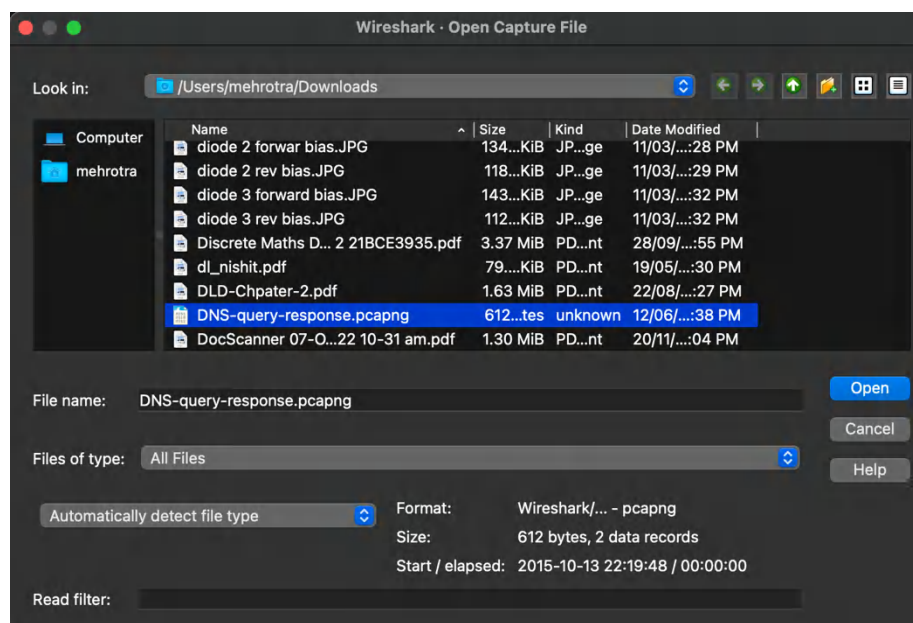
III. STEP BY STEP PROCEDURE:

1. Download the following packet capture from the <https://www.digitalforensicsworkbook.com/data-sets: DNS-query-response.pcapng>.



2. Launch Wireshark.

3. Open the network packet capture named “DNS-query- response.pcapng.” The packet capture will appear in Wireshark as shown in screenshot.



4. How many packets are involved in a DNS query?

Two packets are transmitted: one packet for the query and the other for the response

The image shows a Wireshark packet capture of a DNS query and response. The packet list at the top shows two packets: a standard query (No. 1) and a standard query response (No. 2). The packet details pane on the left shows the structure of the first packet (No. 1), which is a standard query. The packet bytes pane on the right shows the raw data of the first packet, including the Ethernet II header, Internet Protocol Version 4 header, and User Datagram Protocol header.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|--------------|--------------|----------|--------|---|
| 1 | 0.000000 | 192.168.0.35 | 209.18.47.61 | DNS | 74 | Standard query 0xc80f A centralops.net |
| 2 | 0.184470 | 209.18.47.61 | 192.168.0.35 | DNS | 90 | Standard query response 0xc80f A centralops.net A 208.101.16.74 |

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface
Ethernet II, Src: VMware_37:07:ed (00:0c:29:37:07:ed), Dst: ARRISGro_0f:ca:07 (d4:05:98:0f:ca:07)
Source: VMware_37:07:ed (00:0c:29:37:07:ed)
Address: VMware_37:07:ed (00:0c:29:37:07:ed)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.0.35, Dst: 209.18.47.61
User Datagram Protocol, Src Port: 64596, Dst Port: 53
Source Port: 64596
Destination Port: 53
Length: 40
Checksum: 0xc154 [unverified]
[Checksum Status: Unverified]
[Stream index: 0]

The image shows a Wireshark packet capture of a DNS query and response. The packet list at the top shows two packets: a standard query (No. 1) and a standard query response (No. 2). The packet details pane on the left shows the structure of the second packet (No. 2), which is a standard query response. The packet bytes pane on the right shows the raw data of the second packet, including the Ethernet II header, Internet Protocol Version 4 header, and User Datagram Protocol header.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|--------------|--------------|----------|--------|---|
| 1 | 0.000000 | 192.168.0.35 | 209.18.47.61 | DNS | 74 | Standard query 0xc80f A centralops.net |
| 2 | 0.184470 | 209.18.47.61 | 192.168.0.35 | DNS | 90 | Standard query response 0xc80f A centralops.net A 208.101.16.74 |

Address: VMware_37:07:ed (00:0c:29:37:07:ed)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.0.35, Dst: 209.18.47.61
0100 = Version: 4
... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 60
Identification: 0x16f5 (5877)
000. = Flags: 0x0
... 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: UDP (17)
Header checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.0.35
Destination Address: 209.18.47.61
User Datagram Protocol, Src Port: 64596, Dst Port: 53
Domain Name System (query)
Transaction ID: 0xc80f
Flags: 0x0100 Standard query
0... .. = Response: Message is a query
... 0... .. = Opcode: Standard query (0)
... 0... .. = Truncated: Message is not truncated
... 1... .. = Recursion desired: Do query recursively
... 0... .. = Z: reserved (0)
... .. = Non-authenticated data: Unacceptable
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
[Response in: 2]

IV. OBSERVATIONS:

- How long did it take to get a response for the DNS query? It took 0.1844 seconds for a response to be returned. DNS has very little overhead and no error checking.

- What is the IP address of the host, which submitted the DNS query? The host with the IP address 192.168.0.35 submitted the request.
What is the IP address of the DNS server?

The IP address of the DNS server is 209.18.47.61.

- Based on the information contained in the Packet Details

Pane, what protocol is used for DNS queries and what is the destination port?

DNS queries use the User Datagram Protocol and send traffic to UDP Port 53.

- What domain name was sought to be resolved? The DNS query was for centralops.net.

- To what IP address did the domain name resolve?

The domain named centralops.net resolved to the IP address 208.101.16.74

V. INFERENCES:

From the aforementioned observations, the following inferences can be drawn:

1. Response Time for DNS Query: It took approximately 0.1844 seconds for a response to be returned for the DNS query. This indicates the time it took for the DNS server to process the query and provide a response.

2. IP Address of the Host: The host with the IP address 192.168.0.35 submitted the DNS query. This IP address identifies the device that initiated the request.

3. IP Address of the DNS Server: The IP address of the DNS server is 209.18.47.61. This is the address of the server that received and processed the DNS query.

4. Protocol and Destination Port for DNS Queries: DNS queries use the User Datagram Protocol (UDP) as the transport protocol. The destination port for DNS queries is Port 53. This information is based on the Packet Details Pane.

5. Domain Name Being Resolved: The DNS query sought to resolve the domain name "centralops.net". This indicates that the user or the application was attempting to find the IP address associated with the given domain name.

6. Resolved IP Address: The domain name "centralops.net" resolved to the IP address 208.101.16.74. This is the IP address that was associated with the domain name in the DNS response.

These inferences provide details about the DNS query, including response time, IP addresses of the host and DNS server, the protocol and destination port used for DNS queries, and the domain name being resolved along with its corresponding IP address.

EXERCISE 1-F : Network Traffic Identification:

TCP Three-way Handshake

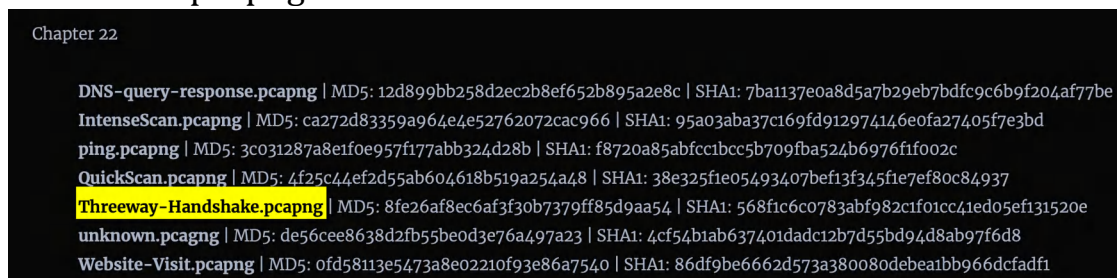
I. AIM: Use Wireshark to review the results of a previously recorded TCP Threeway Handshake. (Download the following packet capture from the [https:// www.digitalforensicsworkbook.com/data-sets](https://www.digitalforensicsworkbook.com/data-sets) :“ThreewayHandshake-Connection.pcapng”).

II. TOOLS USED:

- 1.Product: Wireshark
2. Manufacturer: Wireshark Foundation
3. Web site: <https://www.wireshark.org>

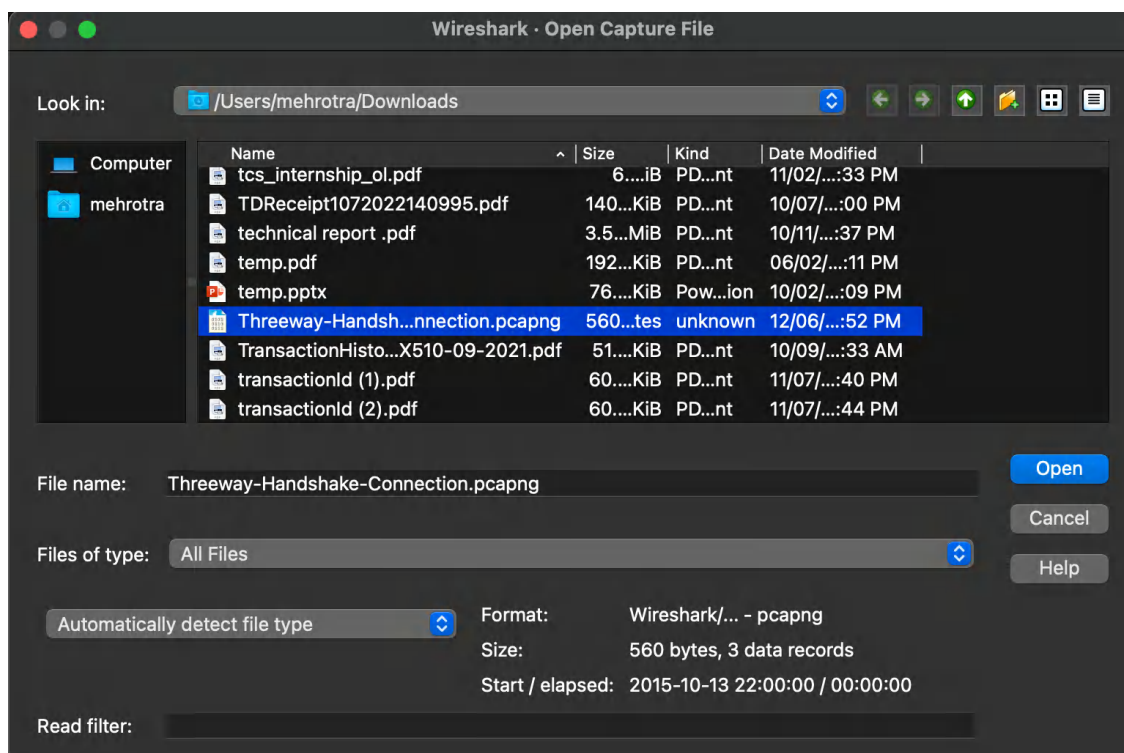
III. STEP BY STEP PROCEDURE:

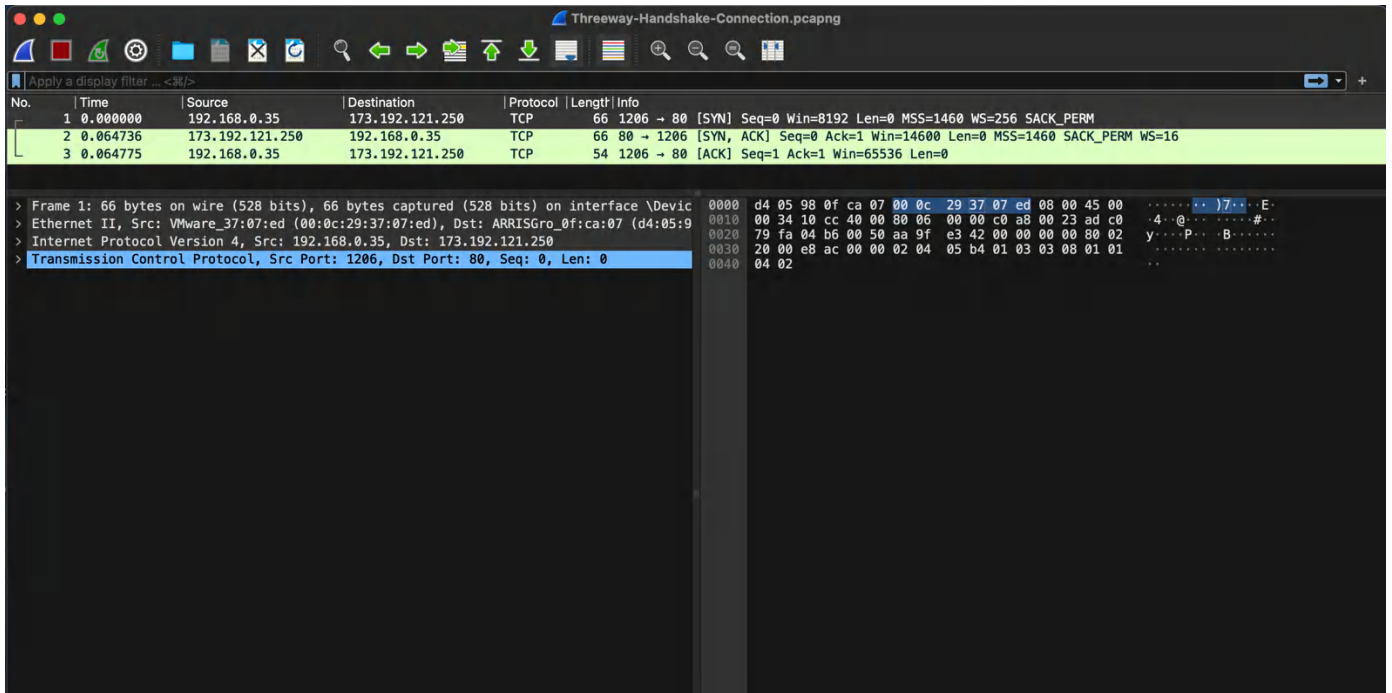
1. Download the following packet capture from the [https:// www.digitalforensicsworkbook.com/data-sets](https://www.digitalforensicsworkbook.com/data-sets): “Threeway- Handshake- Connection.pcapng.”



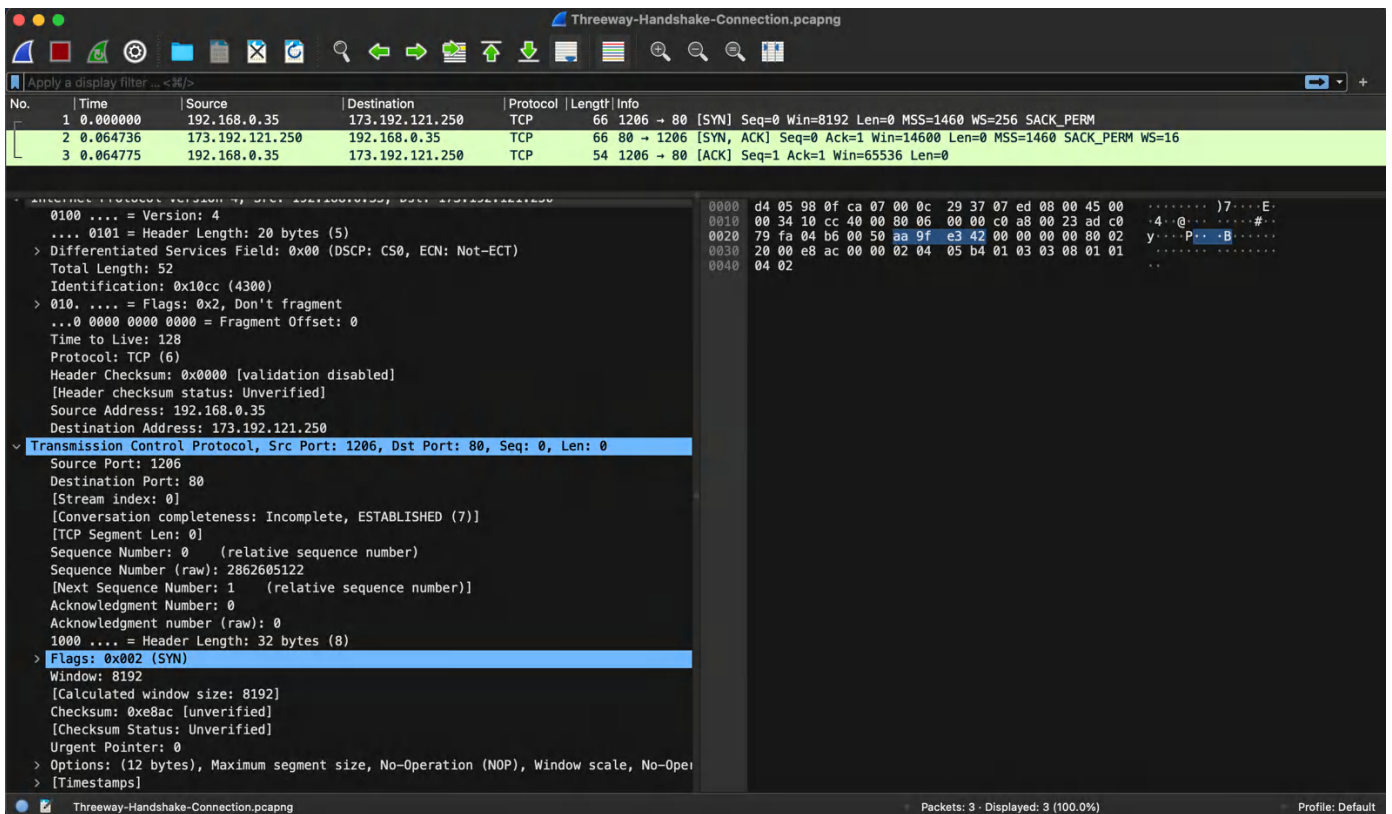
2. Launch Wireshark.

3. Open the file named “Threeway-Handshake- Connection.pcapng.” The results of the packet capture will appear as shown in screenshot.





IV. OBSERVATIONS:



- How long did it take for the three packets in the TCP Three- way Handshake to go back and forth between the source and destination? It took 0.06477 seconds for the three seconds to be processed!.
- What is the IP address of the host that initiated the TCP Three-way Handshake? What is the IP address of the host, which is responding to the Handshake? The IP address of the host, which initiated the Handshake, was 192.168.0.35 ". The IP address of the host responding to the request was 173.192.121.250.
- What was the likely type of server to which the connection was being established? Based on the destination port appearing in the Packet Details Pane, TCP port 80, the server is likely a webserver (HTTP).
- Based on the information in the Packet Details Pane, what flag is set in the first packet of the TTCP Three-way Handshake? The SYN flag is set to 1.
- Based on the information in the Packet List Pane, what are the flags for the second and third packets in the Handshake? The flags for the second packet is SYN and ACK. The flag in the third packet is ACK.

V. INFERENCES:

From the aforementioned observations, the following inferences can be drawn:

1. Duration of TCP Three-way Handshake: The three packets in the TCP Three-way Handshake took approximately 0.06477 seconds to go back and forth between the source and destination. This represents the time taken for the handshake process to establish a TCP connection.
2. IP Addresses of the Hosts: The host that initiated the TCP Three-way Handshake had the IP address 192.168.0.35. The host

responding to the handshake had the IP address 173.192.121.250. These IP addresses identify the source and destination hosts involved in the handshake.

3. Likely Type of Server: Based on the destination port mentioned in the Packet Details Pane, which is TCP port 80, it is likely that the connection was being established with a web server. TCP port 80 is commonly associated with HTTP (Hypertext Transfer Protocol), indicating that the server is likely a web server.

4. Flag in the First Packet of the Handshake: The first packet of the TCP Three-way Handshake has the SYN (Synchronize) flag set to 1. This flag is used to initiate the connection establishment process.

5. Flags in the Second and Third Packets of the Handshake: In the second packet of the handshake, the SYN (Synchronize) and ACK (Acknowledgment) flags are set. This indicates that the receiving host acknowledges the initial synchronization request. The third packet only has the ACK flag set, indicating the acknowledgement of the previous packet in the handshake process.

These inferences provide details about the duration of the TCP Three-way Handshake, the IP addresses of the initiating and responding hosts, the likely type of server being connected to, and the flags set in the packets of the handshake process.