Study of Computer Forensics and different tools used for forensic investigation

Aim of the Experiment:

Study of Computer Forensics and different tools used for forensic investigation

What are Digital Forensics?

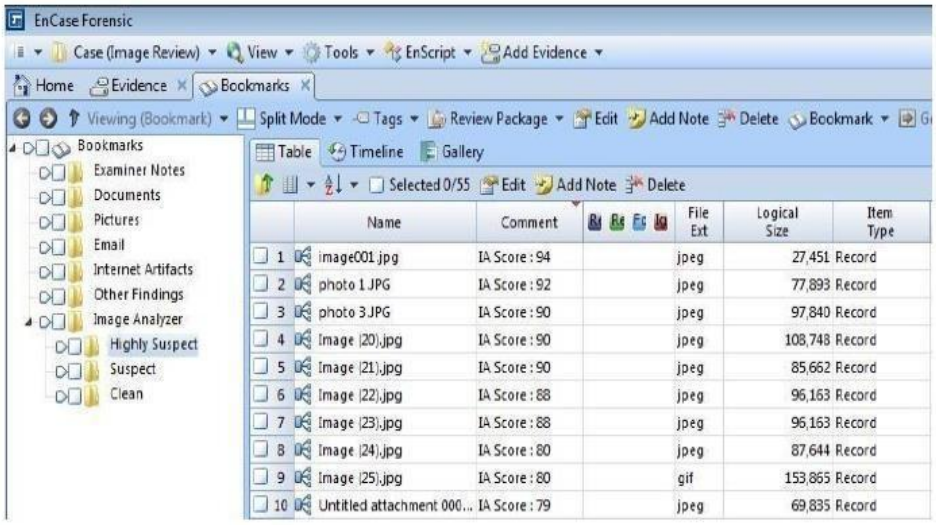
Digital forensics is the field of determining who was responsible for a digital intrusion or other computer crime. It uses a wide range of techniques to gain attribution to the perpetrator. It relies upon the fundamental concept that whenever a digital intrusion or crime is committed, the perpetrator inadvertently leaves a bit of themselves behind for the investigator to find. These "bits" could be entries in log files, changes to the registry, hacking software, malware, remnants of deleted files, etc. All of these can provide clues and evidence to determine their identity and lead to the capture and arrest of the hacker.

As a hacker, the more you know and understand about digital forensics, the better you can evade the standard forensic techniques and even implement anti-forensic measures to throw off the investigator.

The Digital Forensic Tools

Just like in hacking, there are a number of software tools for doing digital forensics. For the hacker, becoming familiar with these tools and how they work is crucial to evading them. Most digital forensic investigators rely upon three major commercial digital forensic suites.

1. Guidance Software's EnCase Forensic
2. Access Data's Forensic Tool Kit (FTK)
3. Prodiscover



These three suites are comprised of multiple tools and reporting features and can be fairly expensive. While these suites are widely used by law enforcement, they use the same or similar techniques as the free open-source suites without the fancy interfaces.

By using the open-source and free suites, we can come to understand how such tools as EnCase work without the expense. EnCase is the most widely used tool by law enforcement, but not necessarily the most effective and sophisticated. These tools are designed for user-friendliness, efficiency, certification, good training, and reporting.

There are a number of the free, open-source forensic suites, including the following three.

1. The Sleuthkit Kit (TSK)
2. Helix
3. Knoppix

The Forensic Tools Available in BackTrack

In addition, there are a large number of individual tools that are available for digital forensics, some of which are available in our BackTrack and Kali distributions.

Some of the better tools in BackTrack include the following, among many others.

- sleuthkit
- truecrypt
- hexedit
- autopsy
- iphoneanalyzer
- rifiuti2
- ptk
- exiftool
- evtparse.pl
- fatback
- scalpel
- dc3dd
- driftnet
- timestomp

What Can Digital Forensics Do?

Digital forensics can do many things, all of which the aspiring hacker should be aware of. Below is a list of just some of the things.

- Recovering deleted files, including emails
- Determine what computer, device, and/or software created the malicious file, software, and/or attack

- Trail the source IP and/or MAC address of the attack
- Track the source of malware by its signature and components
- Determine the time, place, and device that took a picture
- Track the location of a cell phone enabled device (with or without GPS enabled)
- Determine the time a file was modified, accessed or created (MAC)
- Crack passwords on encrypted hard drives, files, or communication
- Determine which websites the perpetrator visited and what files he downloaded
- Determine what commands and software the suspect has utilized
- Extract critical information from volatile memory
- Determine who hacked the wireless network and who the unauthorized users are And that' just some of the things you can do with digital forensics!

What Is Anti-Forensics?

Anti-forensics are techniques that can be used to obfuscate information and evade the tools and techniques of the forensic investigator. Some of these techniques include the following.

- **Hiding Data:** Hiding data can include such things as encryption and steganography.
- **Artefact wiping:** Every attack leaves a signature or artefact behind. Sometimes it's wise to attempt to wipe these artefacts from the victim machine so as to leave no tell-tale trail for the investigator.
- **Trail Obfuscation:** A decent forensic investigator can trail nearly any remote attack to an IP address and/or MAC address. Trail obfuscation is a technique that leads them to another source of the attack, rather than the actual attack.
- **Change the timestamp:** Change the file timestamp (modify, access, and change) to evade detection by forensic tools.

List of Forensic tool

Forensics Field Tools Forensics Field

Tools **FTKImager** Forensic disk imager

and file recovery. **Log Parser Lizard**

GUI

Flexible and powerful log file parser. It also does much much more.

Noxcivis Field Toolkit

The Noxcivis Field Toolkit (NFT) is a free and open interface that allows forensic examiners and collection teams to collect information from a computer.

Active@ Partition Recovery

Autopsy

Forensics tool. Autopsy is a digital forensics platform and graphical interface to The Sleuth Kit® and other digital forensics tools. It can be used by law enforcement, military, and corporate examiners to investigate what happened on a computer. You can even use it to recover photos from your camera's memory card.And that' just some of the things you can do with digital forensics!

CAINE (Computer Aided Investigative Environment)
CAINE (Computer Aided Investigative Environment) is an Italian GNU/Linux live distribution created as a project of Digital Forensics. CAINE represents fully the spirit of the Open Source philosophy because the project is completely open, everyone could take the legacy of the previous developer or project manager. The distro is open source, the Windows side (Wintaylor) is open source and, the last but not the least, the distro is installable, so giving the opportunity to rebuild it in a new brand version, so giving a long life to this project.

Capture-BAT Download Page | The Honeynet Project

Capture-BAT Download Page Capture BAT is a behavioural analysis tool of applications for the Win32 operating system family. Capture BAT is able to monitor the state of a system during the execution of applications and processing of documents, which provides an analyst with insights on how the software operates even if no source code is available. Capture BAT monitors state changes on a low kernel level and can easily be used across various Win32 operating system versions and configurations.

cFAIR Technologies Tools

cFAIR Technologies Tools for forensics and eDiscovery

Digital Forensics Framework (DFF)

Open Source Digital investigation software DFF (Digital Forensics Framework) is a free and Open Source computer forensics software built on top of a dedicated Application Programming Interface (API). It can be used both by professional and non-expert people in order to quickly and easily collect, preserve and reveal digital evidence without compromising systems and data. **EnCase Forensic Imager**
FREE software to capture a forensically sound copy of data.
Explorer Suite Suite of executable file forensics utilities.
File and Partition Recovery Software
Free download Partition Recovery Software, Deleted Partition Recovery, Active Partition Recovery Software. Realize partition data recovery with Free Partition Recovery Software, Free Active Partition Recovery Software, Free Disk Partition Recovery Tool, Free NTFS Partition Recovery Tool, Recovery Partition, Hard Disk Recovery, Drive Partition Recovery, Deleted Partition Recovery and Hard Drive Partition Recovery Tool.

Commands for Data and Network Security

AIM:

To implement commands for data and network security.

COMMANDS:

1. ipconfig:

Command : ipconfig
Function : Provide information regarding the IP address configuration of the device
Usage : ipconfig
Options :

/all	Display full configuration information.
/release	Release the IPv4 address for the specified adapter.
/renew	Renew the IPv4 address for the specified adapter.
/flushdns	Purges the DNS Resolver cache.
/registerdns	Refreshes all DHCP leases and re-registers DNS names
/displaydns	Display the contents of the DNS Resolver Cache.
/showclassid	Displays all the dhcp class IDs allowed for adapter.
/setclassid	Modifies the dhcp class id.

Output:

```
C:\Users>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f520:55cc:9492:b969%12
   IPv4 Address. . . . . . . . . . . : 192.168.0.103
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1

Ethernet adapter vEthernet (WSL):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2991:d721:2aa2:e61e%40
   IPv4 Address. . . . . . . . . . . : 172.23.96.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :
```

2. ping:

Command: ping
Function : It allows the user to check the connectivity of our system to another host
Usage : ping www.google.com
Options :

-t Ping the specified host until stopped.
-a Resolve addresses to hostnames.
-n count Number of echo requests to send.
-l size Send buffer size.
-f Set Don't Fragment flag in packet (IPv4-only).
-r count Record route for count hops (IPv4-only).
-s count Timestamp for count hops (IPv4-only).
-j host-list Loose source route along host-list (IPv4-only).
-k host-list Strict source route along host-list (IPv4-only).
-w timeout Timeout in milliseconds to wait for each reply.
-R Use routing header to test reverse route also (IPv6-only).
-p Ping a Hyper-V Network Virtualization provider address.

Output:

```
C:\Users>ping www.google.com

Pinging www.google.com [142.250.183.228] with 32 bytes of data:
Reply from 142.250.183.228: bytes=32 time=4ms TTL=118
Reply from 142.250.183.228: bytes=32 time=4ms TTL=118
Reply from 142.250.183.228: bytes=32 time=4ms TTL=118
Reply from 142.250.183.228: bytes=32 time=4ms TTL=118

Ping statistics for 142.250.183.228:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 4ms, Average = 4ms
```

3. netstat:

Command: netstat
Function : Displays protocol statistics and current TCP/IP network connections.
Usage : netstat
Options:

-a Displays all connections and listening ports.
-f Displays Fully Qualified Domain Names (FQDN) for foreign addresses.
-o Displays the owning process ID associated with each connection.
-p Shows connections for the protocol specified by proto
-r Displays the routing table.
-s Displays per-protocol statistics.
-t Displays the current connection offload state.

Output:

```
C:\Users>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:9012         DESKTOP-816HLDS:62949  ESTABLISHED
  TCP    127.0.0.1:49674        DESKTOP-816HLDS:49675  ESTABLISHED
  TCP    127.0.0.1:49675        DESKTOP-816HLDS:49674  ESTABLISHED
  TCP    127.0.0.1:49676        DESKTOP-816HLDS:49677  ESTABLISHED
  TCP    127.0.0.1:49677        DESKTOP-816HLDS:49676  ESTABLISHED
  TCP    127.0.0.1:62341        DESKTOP-816HLDS:62342  ESTABLISHED
  TCP    127.0.0.1:62342        DESKTOP-816HLDS:62341  ESTABLISHED
  TCP    127.0.0.1:62343        DESKTOP-816HLDS:62344  ESTABLISHED
  TCP    127.0.0.1:62344        DESKTOP-816HLDS:62343  ESTABLISHED
```

4. Hostname:

Command: hostname
Function : Displays the hostname of the system
Usage : hostname
Options :
-a Displays the alias name of the host
-d Displays the domain name
-s Display the short version of the hostname
-v Expand all output to verbose

Output:

```
C:\Users>hostname
DESKTOP-816HLDS
```

5. getmac:

Command: getmac
Function : Displays MAC address for each network adapter in the computer
Usage : getmac
Options :
/S Specifies the remote system to connect to.
/U Specifies the user context under which the command should execute.
/P Specifies the password for the given user context.
/V Specifies that verbose output is displayed

Output:

```
C:\Users>getmac

Physical Address    Transport Name
=================== ==========================================================
30-D0-42-09-BE-2A   Media disconnected
64-6C-80-D5-90-61   \Device\Tcpip_{8B2D29E0-DFD3-42FE-B4D7-D718EC86B03C}
```

6. tracert

Command: tracert
Function : Trace the route during the transmission of the data packet over the destination host.
Usage : tracert www.google.com
Options :
-d Do not resolve addresses to hostnames.
-h Maximum number of hops to search for target.
-j Loose source route along host-list (IPv4-only).
-w Wait timeout milliseconds for each reply.
-R Trace round-trip path (IPv6-only).
-S Source address to use (IPv6-only).

Output:

```
C:\Users>tracert www.google.com

Tracing route to www.google.com [142.250.183.228]
over a maximum of 30 hops:

  1     2 ms     2 ms     1 ms  192.168.0.1
  2     4 ms     2 ms     2 ms  10.213.0.1
  3     *        *        *     Request timed out.
  4    18 ms   167 ms   127 ms  10.200.150.30
  5     4 ms     5 ms     3 ms  72.14.212.80
  6     5 ms     4 ms    10 ms  142.251.227.217
  7     4 ms     3 ms     6 ms  209.85.253.85
  8     7 ms     4 ms     4 ms  maa05s23-in-f4.1e100.net [142.250.183.228]

Trace complete.
```

7. nbtstat:

Command: nbtstat

Function : Displays protocol statistics and current TCP/IP connections using NBT

Usage : nbtstat -a 172.16.212.133

Options :

- -a (adapter status) Lists the remote machine's name table given its name
- -c (cache) Lists NBT's cache of remote [machine] names and their IP addresses
- -n (names) Lists local NetBIOS names.
- -r (resolved) Lists names resolved by broadcast and via WINS

Output:

```
C:\Users>nbtstat

Displays protocol statistics and current TCP/IP connections using NBT
(NetBIOS over TCP/IP).

NBTSTAT [ [-a RemoteName] [-A IP address] [-c] [-n]
        [-r] [-R] [-RR] [-s] [-S] [interval] ]

  -a   (adapter status) Lists the remote machine's name table given its name
  -A   (Adapter status) Lists the remote machine's name table given its
                        IP address.
  -c   (cache)          Lists NBT's cache of remote [machine] names and their IP addresses
  -n   (names)          Lists local NetBIOS names.
  -r   (resolved)       Lists names resolved by broadcast and via WINS
  -R   (Reload)         Purges and reloads the remote cache name table
  -S   (Sessions)       Lists sessions table with the destination IP addresses
  -s   (sessions)       Lists sessions table converting destination IP
                        addresses to computer NETBIOS names.
  -RR  (ReleaseRefresh) Sends Name Release packets to WINS and then, starts Refresh

  RemoteName   Remote host machine name.
  IP address   Dotted decimal representation of the IP address.
  interval     Redisplays selected statistics, pausing interval seconds
               between each display. Press Ctrl+C to stop redisplaying
               statistics.
```

8. telnet:

Command: telnet

Function : It enables one computer to connect to local computer

Usage : #telnet 38.76.11.19

Options:

Close	Closes the current telnet connection
Mode type	Sets the transmission type
Quit	Terminates the Telnet client including all active connections
Set	Modifies the connection parameters

Status Shows whether the connection is active

Output:

```
→  telnet www.google.com 80
Trying 142.250.77.100...
Connected to www.google.com.
Escape character is '^]'.
^]
```

9. Tasklist:

Command: tasklist

Function : Displays a list of currently running processes on the local computer

Usage : tasklist

Options :

/S Specifies the remote system to connect to.

/U Specifies the user context under which the command should execute.

/P Specifies the password for the given user context.

/M Lists all tasks currently using the given exe/dll name.

/V Displays verbose task information.

Output:

```
C:\Users>tasklist

Image Name                     PID Services        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0      3,740 K
Secure System                   76 Services                   0     23,320 K
Registry                       112 Services                   0     55,080 K
smss.exe                       544 Services                   0        784 K
csrss.exe                     1016 Services                   0      4,568 K
wininit.exe                    728 Services                   0      4,188 K
services.exe                  1052 Services                   0     15,244 K
```

10. ARP:

Command: arp

Function : Displays and modifies the IP-to-Physical address translation tables

Usage : ARP -a 192.168.43.255

Options :

-a Displays current ARP entries by interrogating the current protocol data.

-v Displays current ARP entries in verbose mode.

-d Deletes the host specified by the hostname variable

-s Adds the host and associates the Internet address with the Physical address

Output:

```
C:\Users>arp -a 192.168.43.255
No ARP Entries Found.
```

11. Pathping:

Command: pathping

Function : It provides the combination of the best aspects of tracert and ping

Usage : pathping 172.16.9.254

Options :

-g Loose source route along host-list.

-h Maximum number of hops to search for target.

-I Use the specified source address.

-n Do not resolve addresses to hostnames.

-p Wait period milliseconds between pings.

-q Number of queries per hop.

-w Wait timeout milliseconds for each reply.

Output:

```
C:\Users>pathping 172.16.9.254

Tracing route to 172.16.9.254 over a maximum of 30 hops

  0  DESKTOP-816HLDS [192.168.0.103]
  1  192.168.0.1
  2  10.213.0.1
  3     *        *        *
Computing statistics for 50 seconds...
            Source to Here   This Node/Link
Hop  RTT    Lost/Sent = Pct  Lost/Sent = Pct  Address
  0                                           DESKTOP-816HLDS [192.168.0.103]
                                 0/ 100 =  0%   |
  1    3ms     0/ 100 =  0%     0/ 100 =  0%  192.168.0.1
                                 0/ 100 =  0%   |
  2   14ms     0/ 100 =  0%     0/ 100 =  0%  10.213.0.1

Trace complete.
```

12. Route:

Command: route

Function : Displays and modifies the entries in the local IP routing table

Usage : route

Options :

-f Clears the routing tables of all gateway entries.

-p When used with the ADD command, makes a route persistent across boots of the system.

-n Displays routing table in full numeric form

-Cn Kernel's routing cache information

Output:

```
C:\Users>route print
===========================================================================
Interface List
 10...30 d0 42 09 be 2a ......Realtek PCIe GbE Family Controller
  3...66 6c 80 d5 90 61 ......Microsoft Wi-Fi Direct Virtual Adapter
 16...76 6c 80 d5 90 61 ......Microsoft Wi-Fi Direct Virtual Adapter #2
 12...64 6c 80 d5 90 61 ......Qualcomm QCA9377 802.11ac Wireless Adapter
  1...........................Software Loopback Interface 1
 40...00 15 5d 8c 9a 59 ......Hyper-V Virtual Ethernet Adapter
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.103     35
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      172.23.96.0    255.255.240.0         On-link       172.23.96.1   5256
      172.23.96.1  255.255.255.255         On-link       172.23.96.1   5256
   172.23.111.255  255.255.255.255         On-link       172.23.96.1   5256
      192.168.0.0    255.255.255.0         On-link     192.168.0.103    291
    192.168.0.103  255.255.255.255         On-link     192.168.0.103    291
    192.168.0.255  255.255.255.255         On-link     192.168.0.103    291
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link       172.23.96.1   5256
        224.0.0.0        240.0.0.0         On-link     192.168.0.103    291
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link       172.23.96.1   5256
  255.255.255.255  255.255.255.255         On-link     192.168.0.103    291
===========================================================================
Persistent Routes:
  None
```

```
IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
 40   5256 fe80::/64                On-link
 12    291 fe80::/64                On-link
 40   5256 fe80::2991:d721:2aa2:e61e/128
                                    On-link
 12    291 fe80::f520:55cc:9492:b969/128
                                    On-link
  1    331 ff00::/8                 On-link
 40   5256 ff00::/8                 On-link
 12    291 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```

13. FC:

Command: FC file1.txt file2.txt

Function: Comparing two files

```
C:\Users\Public\record\CF>FC hi.txt hello.txt
Comparing files hi.txt and HELLO.TXT
***** hi.txt
GOOD
***** HELLO.TXT
***
*****
```

14. cipher:

Command: cipher file_name

Function: New files added to directly will not be encrypted.

```
C:\Users\Public\record\CF>cipher hello.txt

 Listing C:\Users\Public\record\CF\
 New files added to this directory will not be encrypted.

U hello.txt
```

15. recimg:

Command: recimg

Function: Create custom recovery images

```
C:\Windows\system32>recimg /showcurrent

\\?\GLOBALROOT\device\harddisk1\partition1\CustomRecoveryImage\02-19-2015
RecImg: Operation completed successfully
```

16. Net View:

Command: net view

```
C:\Windows\system32>net view
Server Name            Remark

-----------------------------------
\\MASQWW-1-PC
The command completed successfully.
```

Result:

The usage of basic networking commands has been understood and executed successfully.

Creating Image using Forensic Tools

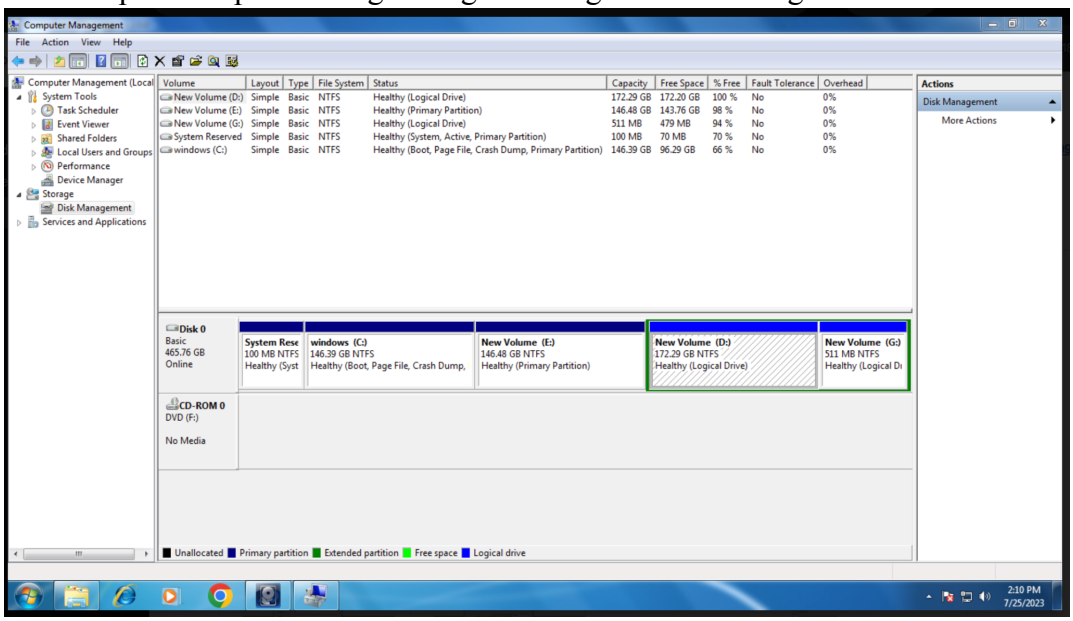
AIM:

To create an image of a hard drive using access data FTK Imager.

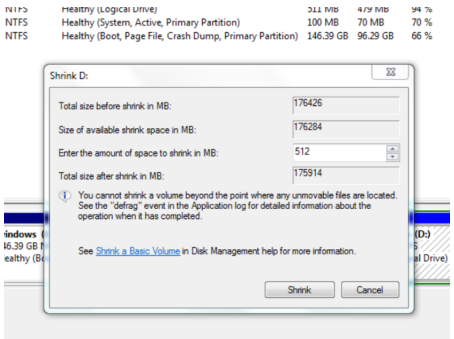
PROCEDURE:

DISK PARTITIONING:

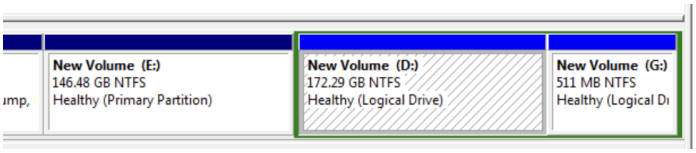
1. Open disk partitioning settings > storage > Disk management. Choose a drive.



2. Give a size of about 512 MB and shrink the disk.

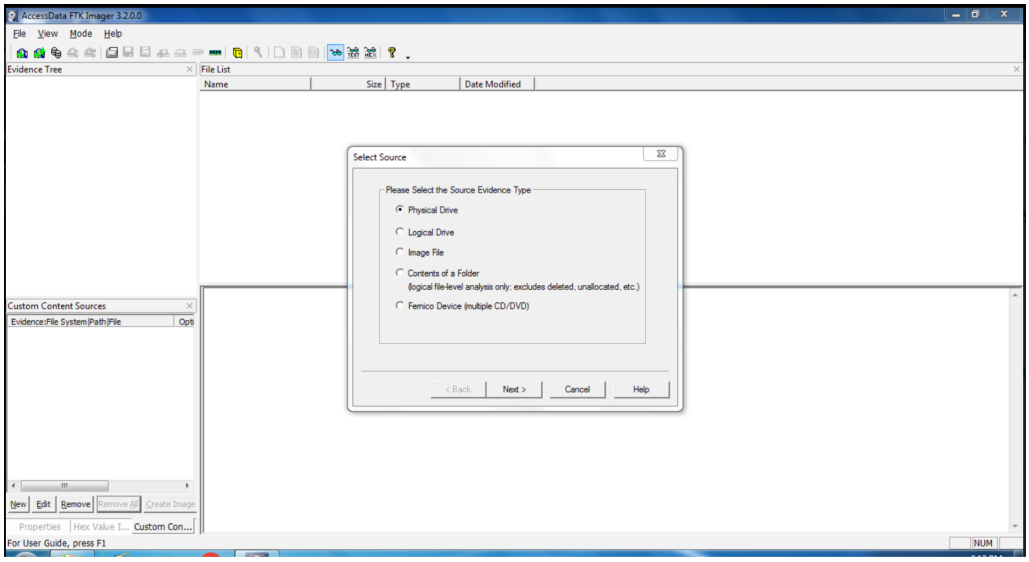


3. New partition is created with 512MB of size.

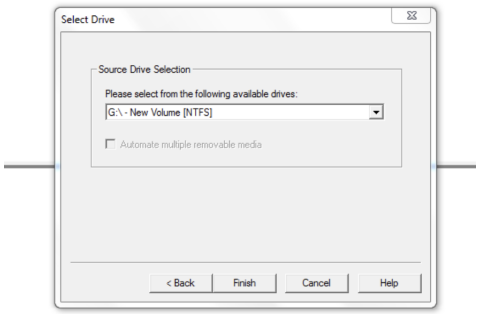


ACCESS DATA FTK IMAGER:

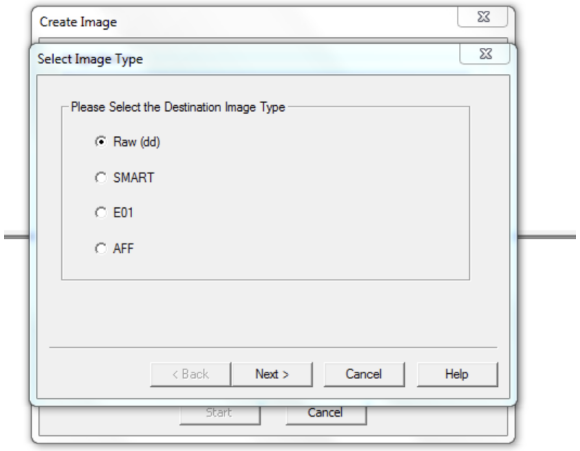
1. Click file > create Disk Image
2. Choose Logical Drive.



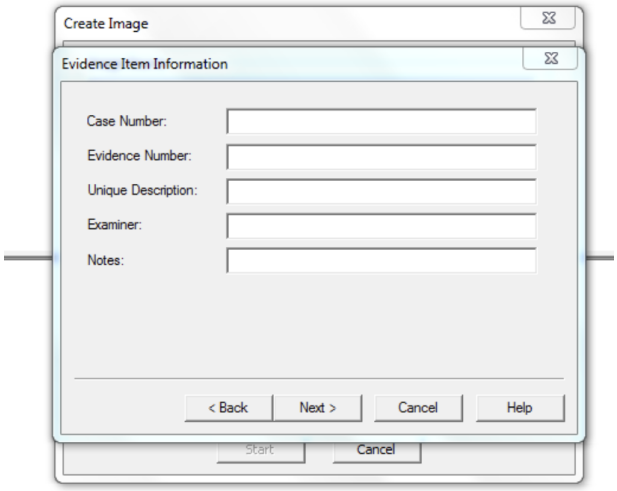
3. Choose the newly created disk.



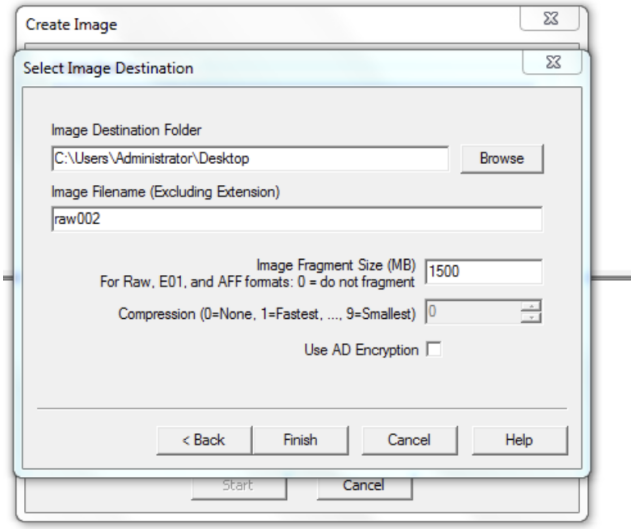
4. Add the Destination for the Image and choose “raw(dd”)



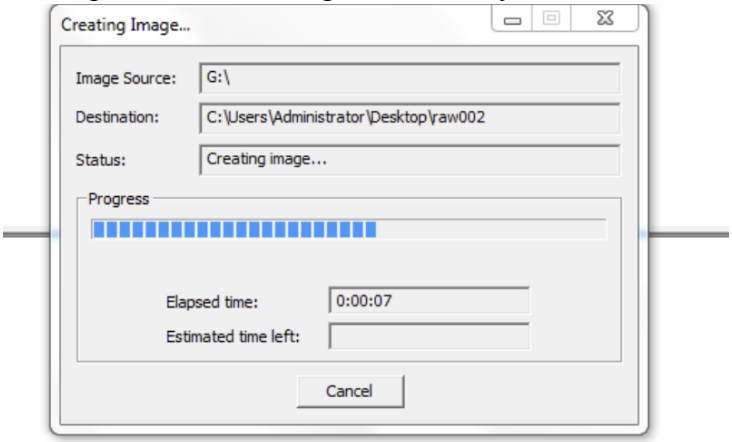
5. Enter the evidence details:

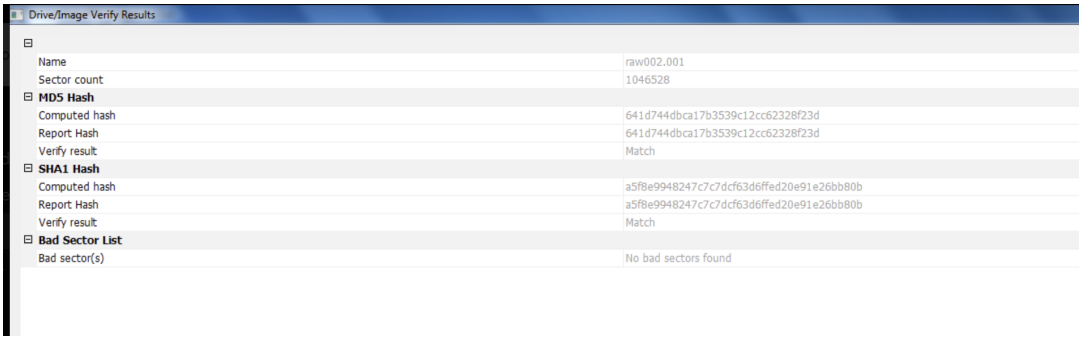


6. Choose the destination folder of Image:



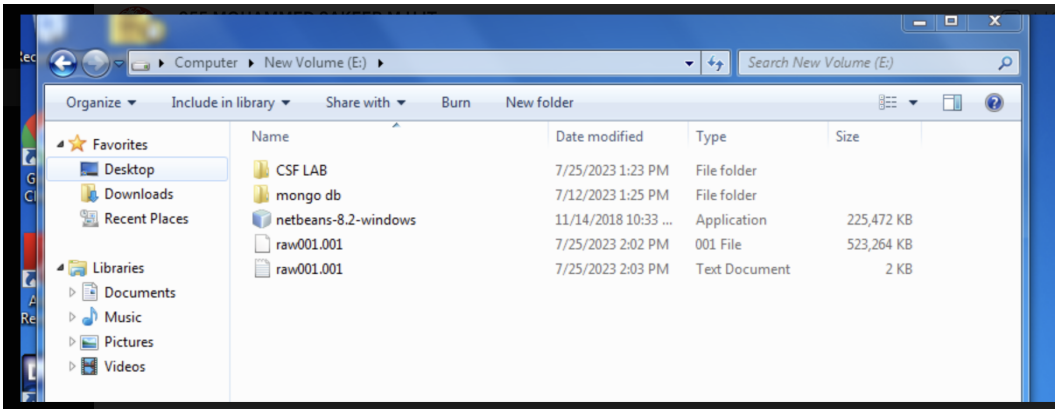
7. Finish and start the process. New Image successfully created in the destination folder.





Drive/Image Verify Results

Name	raw002.001
Sector count	1046528
MD5 Hash	
Computed hash	641d744dbca17b3539c12cc62328f23d
Report Hash	641d744dbca17b3539c12cc62328f23d
Verify result	Match
SHA1 Hash	
Computed hash	a5f8e9948247c7c7dcf63d6ffed20e91e26bb80b
Report Hash	a5f8e9948247c7c7dcf63d6ffed20e91e26bb80b
Verify result	Match
Bad Sector List	
Bad sector(s)	No bad sectors found



Computer ▸ New Volume (E:) ▸ Search New Volume (E:)

Organize ▾ Include in library ▾ Share with ▾ Burn New folder

Favorites
Desktop
Downloads
Recent Places

Libraries
Documents
Music
Pictures
Videos

Name	Date modified	Type	Size
CSF LAB	7/25/2023 1:23 PM	File folder	
mongo db	7/12/2023 1:25 PM	File folder	
netbeans-8.2-windows	11/14/2018 10:33 ...	Application	225,472 KB
raw001.001	7/25/2023 2:02 PM	001 File	523,264 KB
raw001.001	7/25/2023 2:03 PM	Text Document	2 KB

OUTPUT:

raw001.001 - Notepad

File Edit Format View Help

```
Created By AccessData® FTK® Imager 3.2.0.0

Case Information:
Acquired using: ADI3.2.0.0
Case Number: case001
Evidence Number: evidence001
Unique description: drive
Examiner: KE
Notes: drive image

---------------------------------------------------------------

Information for E:\raw001:

Physical Evidentiary Item (Source) Information:
[Device Info]
 Source Type: Logical
[Drive Geometry]
 Bytes per Sector: 512
 Sector Count: 1,046,528
[Physical Drive Information]
 Removable drive: False
 Source data size: 511 MB
 Sector count:    1046528
[Computed Hashes]
 MD5 checksum:    641d744dbca17b3539c12cc62328f23d
 SHA1 checksum:   a5f8e9948247c7c7dcf63d6ffed20e91e26bb80b

Image Information:
 Acquisition started:   Tue Jul 25 14:02:41 2023
 Acquisition finished:  Tue Jul 25 14:02:53 2023
 Segment list:
  E:\raw001.001

Image Verification Results:
 Verification started:  Tue Jul 25 14:02:56 2023
 Verification finished: Tue Jul 25 14:02:58 2023
 MD5 checksum:    641d744dbca17b3539c12cc62328f23d : verified
 SHA1 checksum:   a5f8e9948247c7c7dcf63d6ffed20e91e26bb80b : verified
```

RESULT:

Creation of Image Using forensic tools done successfully.

Hiding and Extracting Image using the Hex Editor

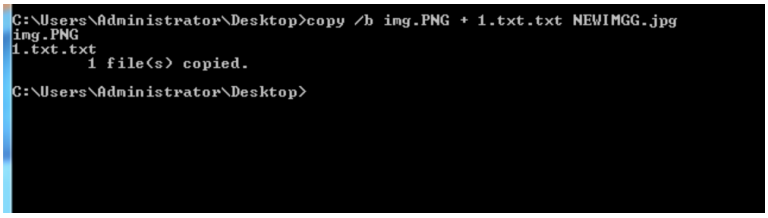
AIM:

To hide and extract text files using an image using Hex editor.

PROCEDURE:

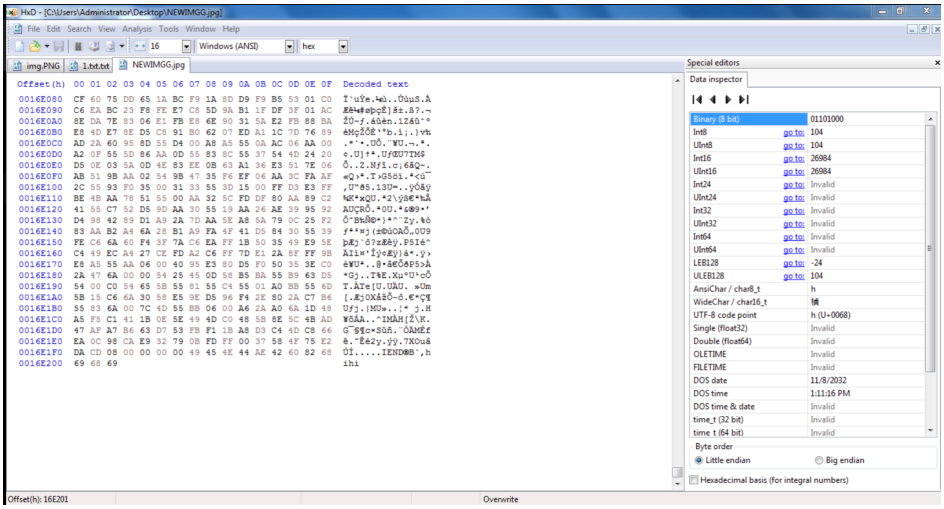
1. Create a text file and image file and save them in the same folder.
2. Open terminal, navigate to the directory where the image and text files are stored.
3. Combine the text and image file using this command:

   copy /b image_path + text_file_path new_file_name



4. The file “NEWIMGG.jpg” is created which combines both the text and image files created earlier.
5. Open hex Editor
6. Open the “NEWIMGG.JPG” in hex editor. The contents of text and image files are appended in hex code form at the end of the file.

OUTPUT:



RESULT:

Implementation of hiding and extraction of text file using an image with the hex editor done successfully.

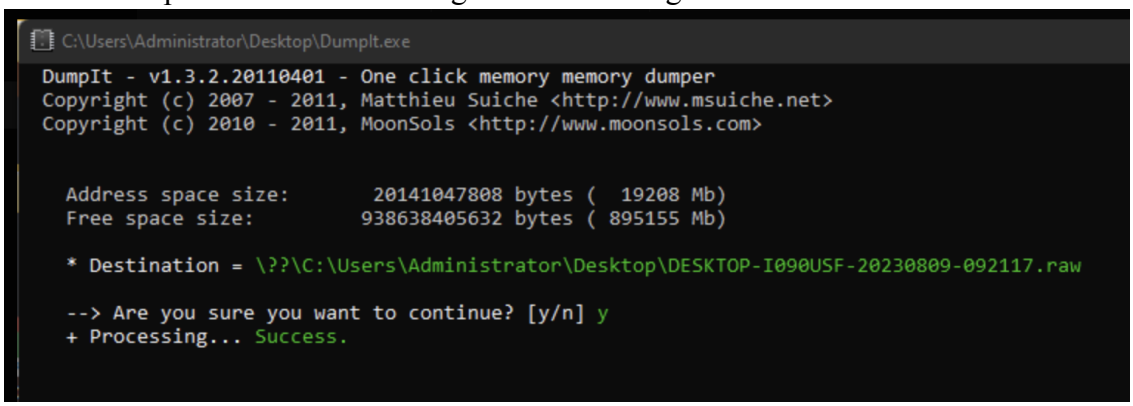
Create Raw Image Using DumpIt and Collect Email Evidences using Bulk Extractor

AIM:

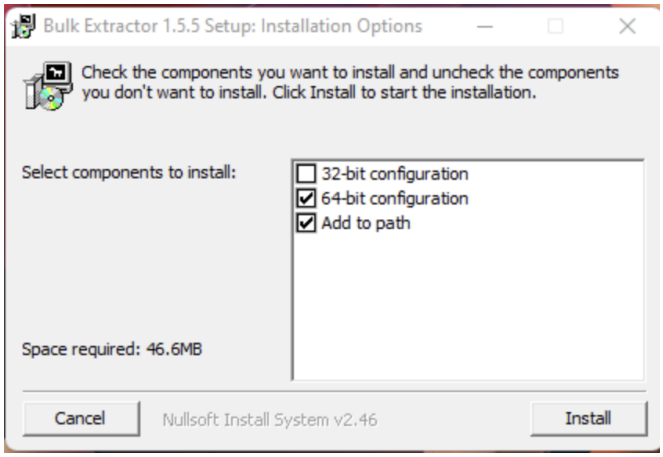
To use a fast memory acquisition tool to retrieve the files from a target location.

PROCEDURE:

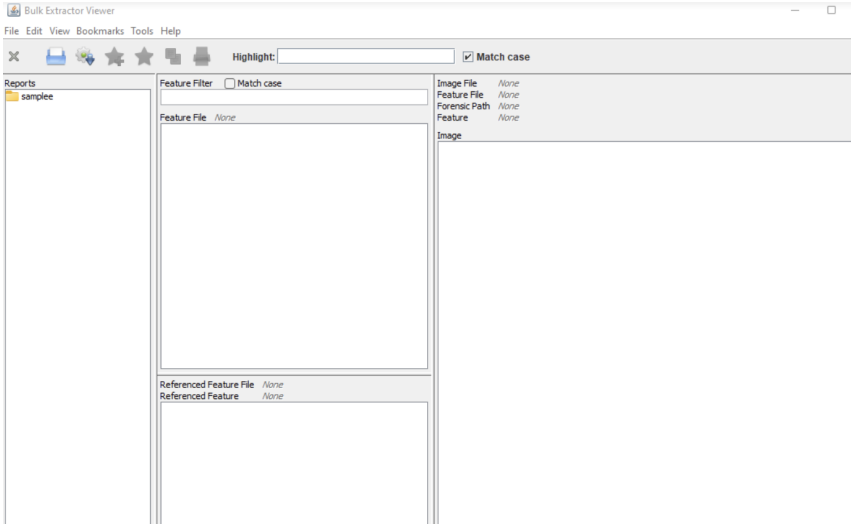
1. Run DumpIt to create a raw image file for the target location.



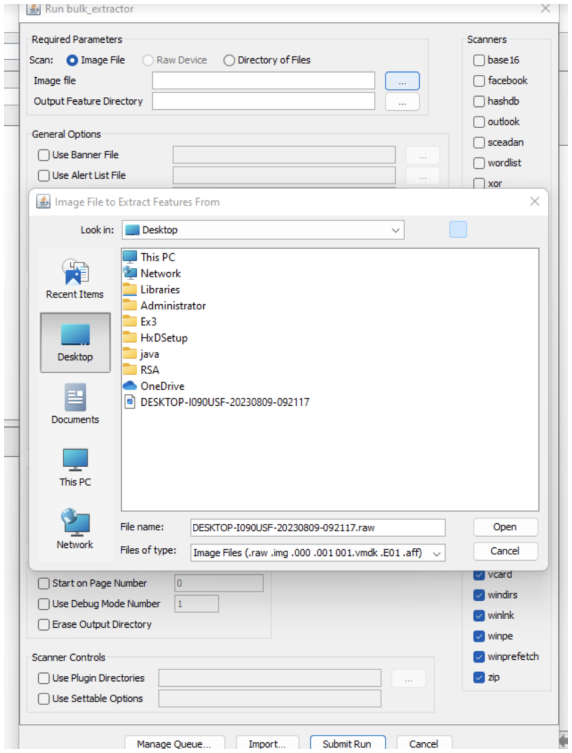
2. Install bulk_extractor 1.5.5



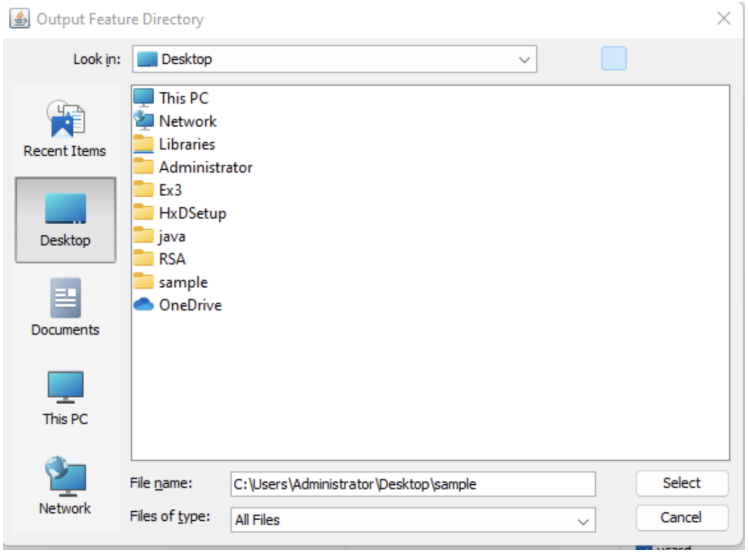
3. The raw file created using DumpIt is used inside bulk extractor to extract all the details



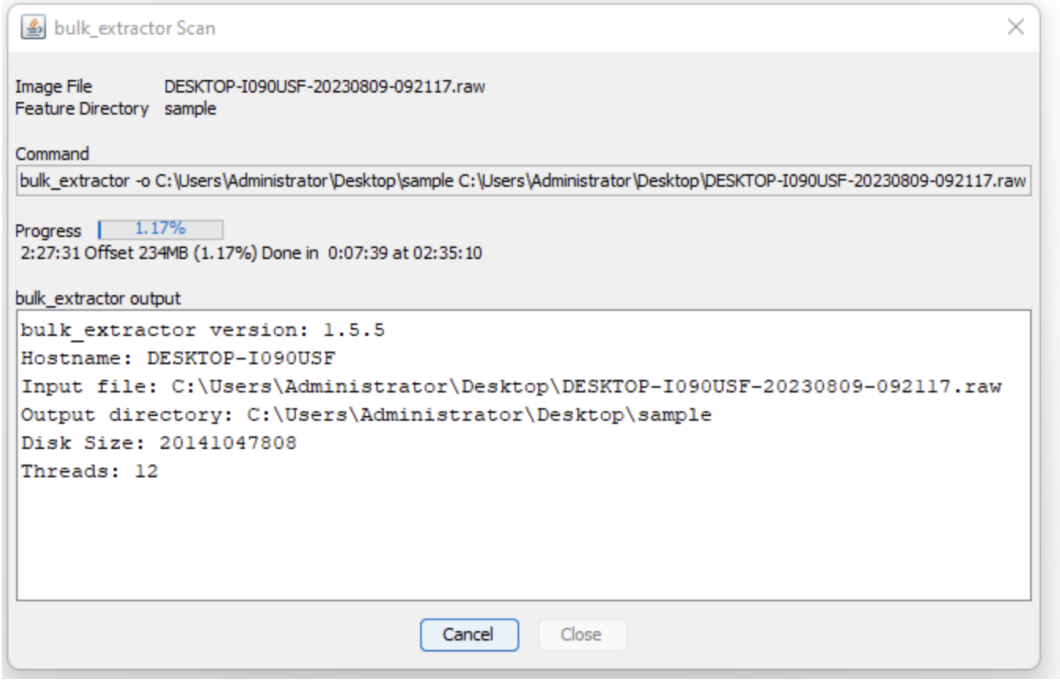
4. Tools > Run bulk_extractor. Select the raw file created



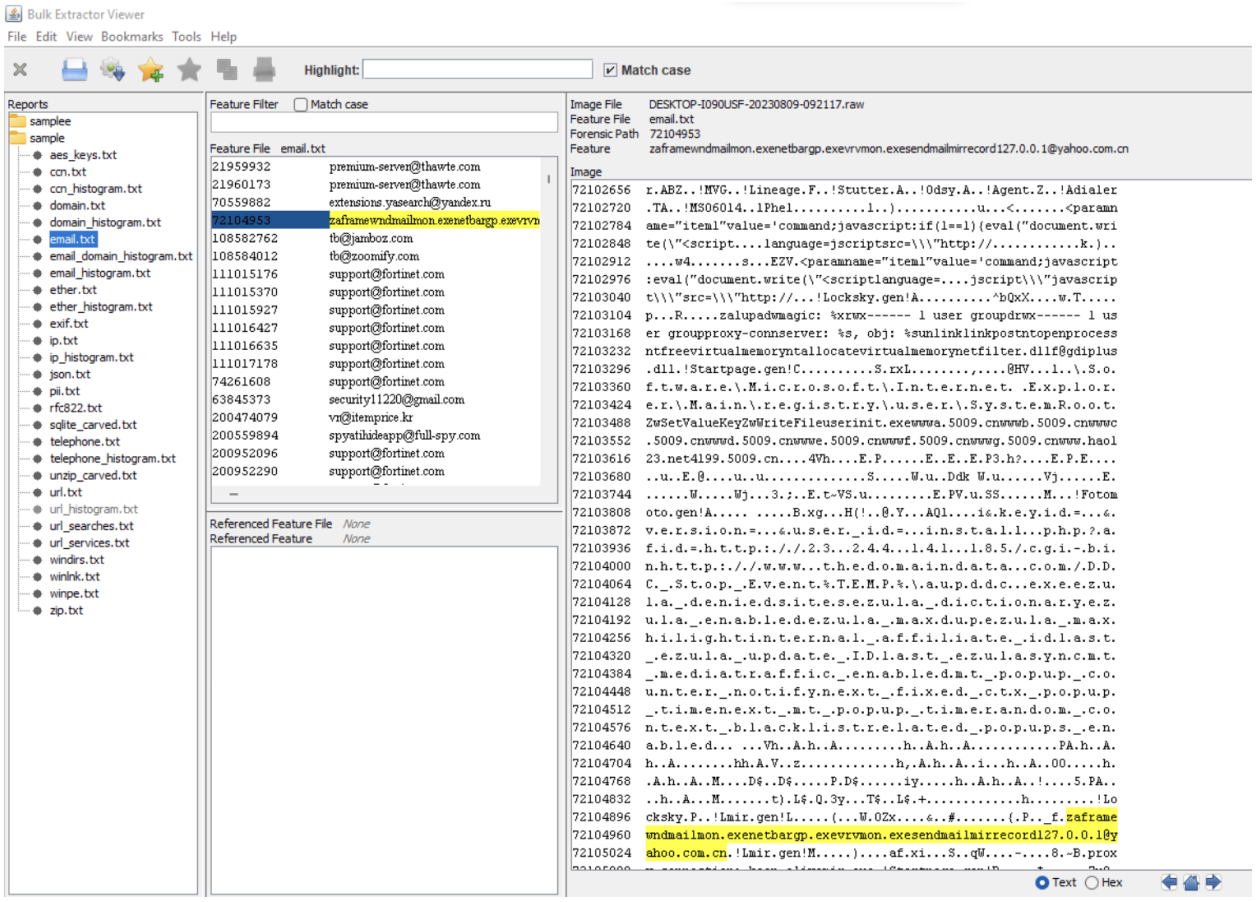
5. Choose the output directory to store the extracted files.



6. Submit and run. Open the destination file to view the output.



OUTPUT:



RESULT:

Extraction of files from Memory using fast memory acquisition tools done successfully.

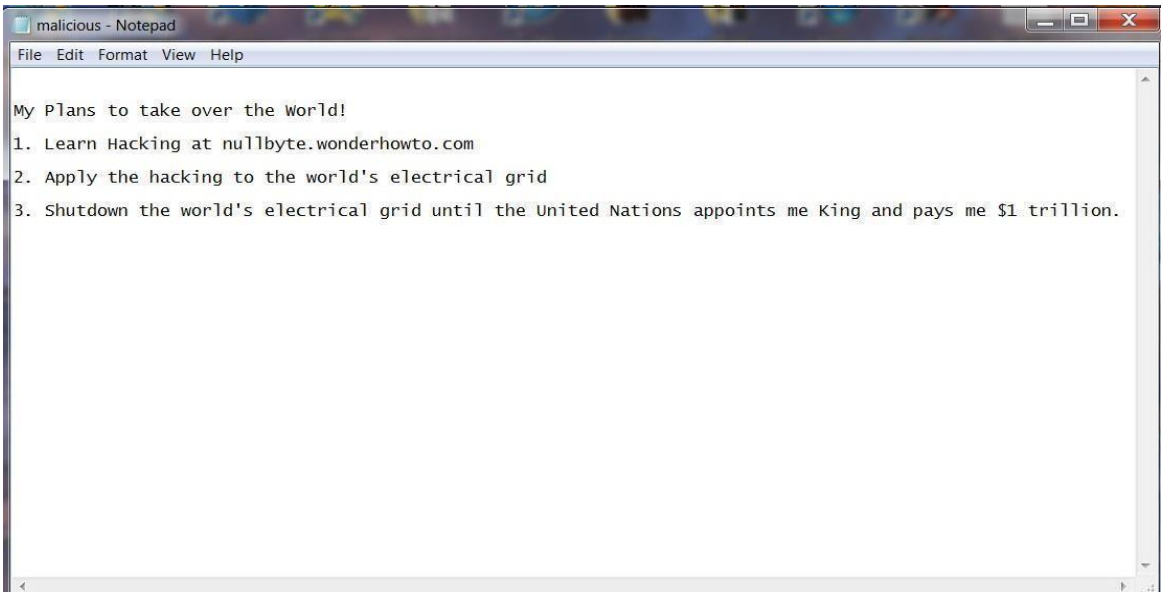
Recover the Deleted Files Using Forensic Tools

AIM:

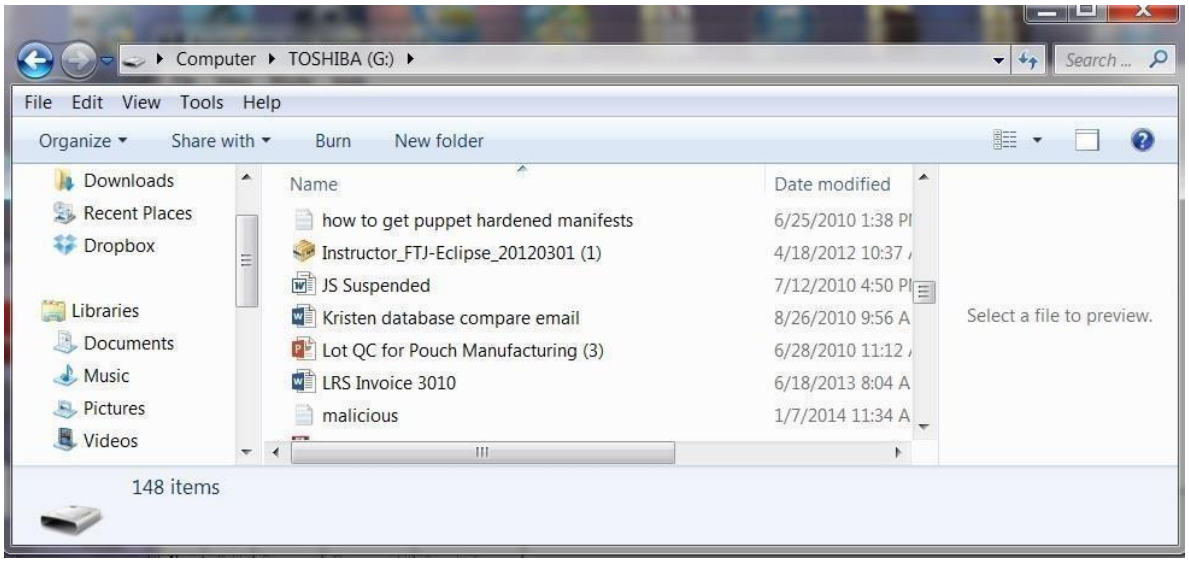
To Recover Deleted Files using Forensic Tools.

PROCEDURE:

1. Create a File: create a malicious document. We will call this document "Malicious" and create it with Notepad in Windows.

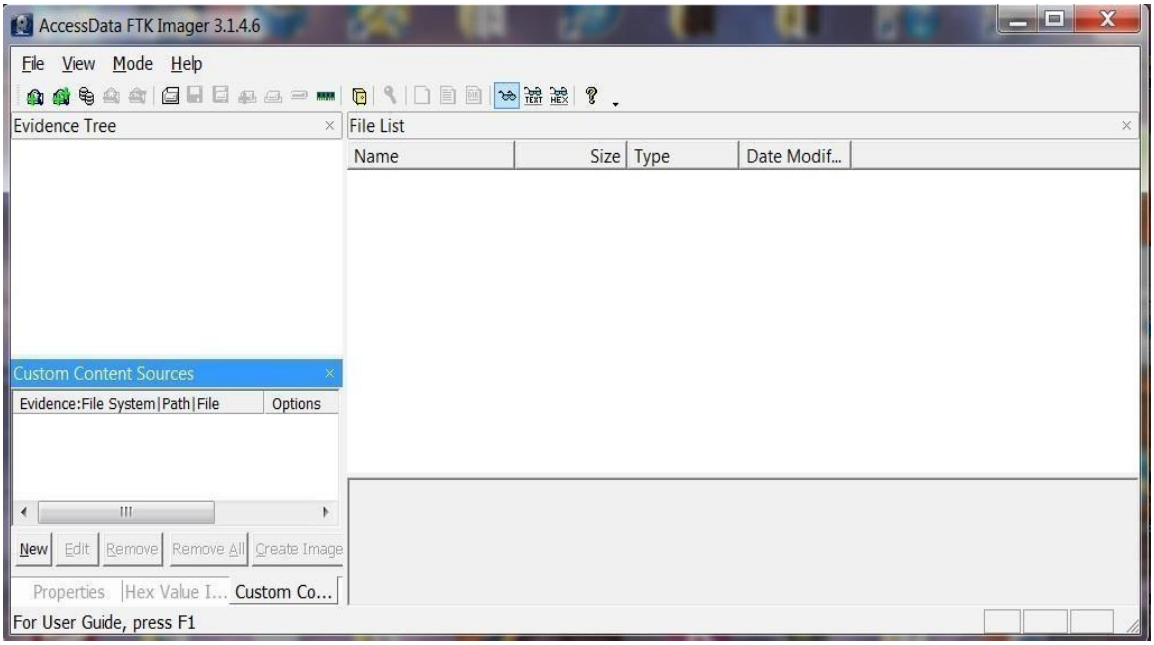


2. Delete the File



Right-click on the malicious file and select delete.Delete in the recycle bin too.

3. Create an Image

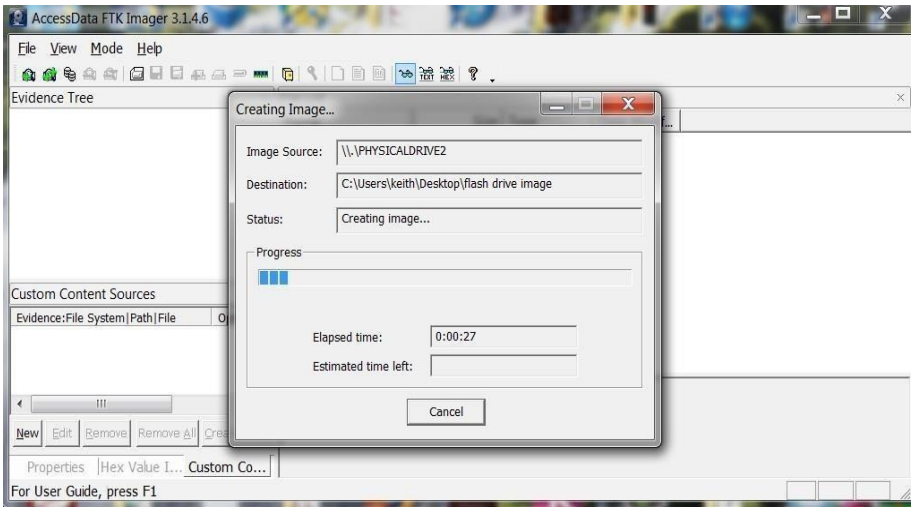


Go to the menu at the top of the application and select:

- **File -> Create Image**

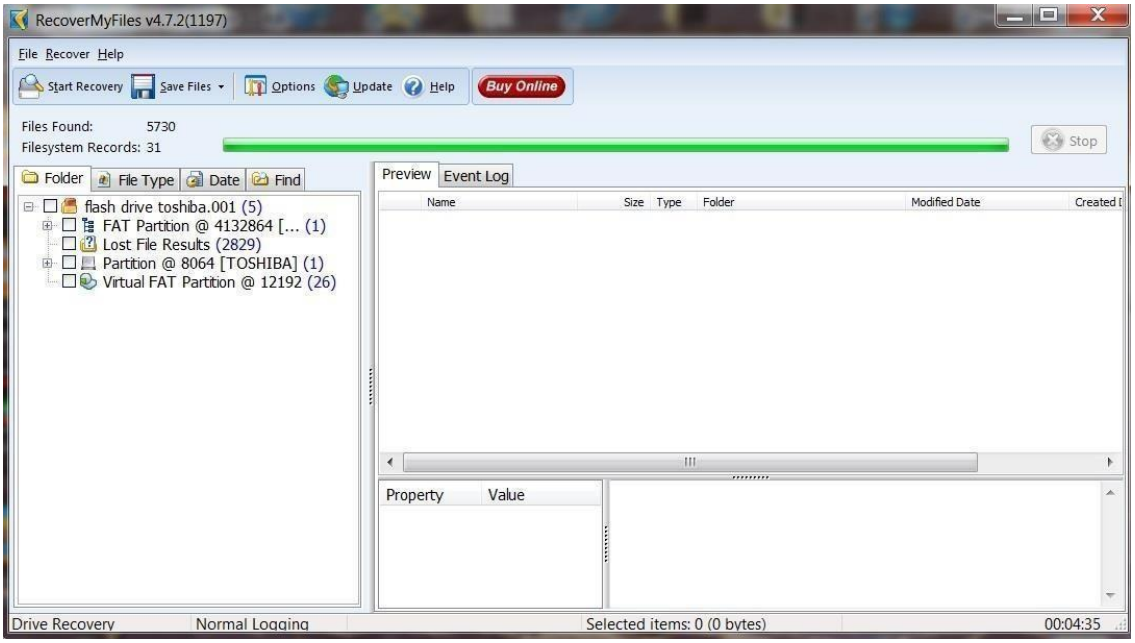
It will open a wizard that will walk you through the process of opening a case and ask you for a case number, evidence number, examiner name, etc.

Finally, it will ask for a location of the physical drive you want to image, a destination directory and a name for the image file. When you are done with all these administrative tasks, FTK Imager will begin the process of creating a forensically sound bit-by-bit image of your drive.



4.Recover Deleted Files

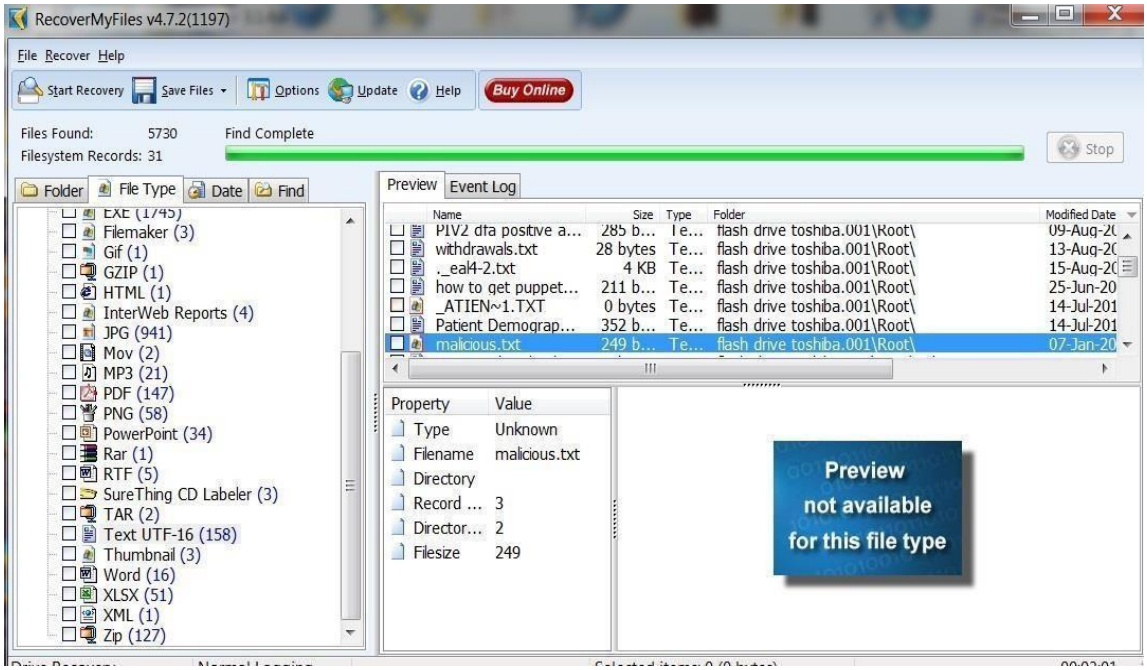
Once you have installed RecoverMyFiles, select the Start Recovery icon in the upper left corner. It will ask you to select either Recover Files or Recover Drive. Select Recover a Drive. It will then search and display all your drives like that in the screenshot below. Since we are using a forensic image, select Add Image button to the right. You will need to provide a path to your image file created with FTK.



Once you select an image file, start the automatic file recovery. When the recovery is completed, you will see a screen similar to the one below.

Then select the File Type tab above the Explorer window to categorize the files by type.

As you can see, there are numerous file types recovered from this flash drive. Since our malicious document was a .txt, we have selected the TXT UTF-16 file type. It then puts all 158 .txt files on display in the upper right window. As you can see, it has recovered our malicious.txt file and everything on it. Busted!



RESULT:

Thus the deleted files are recovered using Forensics Tools.

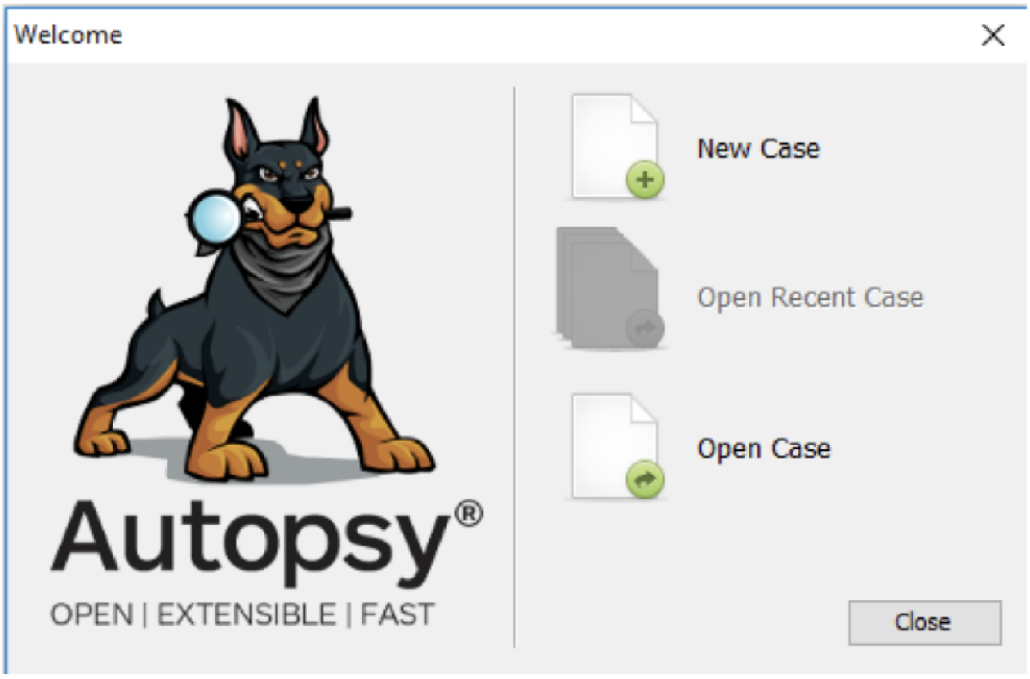
Live Forensics Case Investigation using Autopsy Tool

AIM:

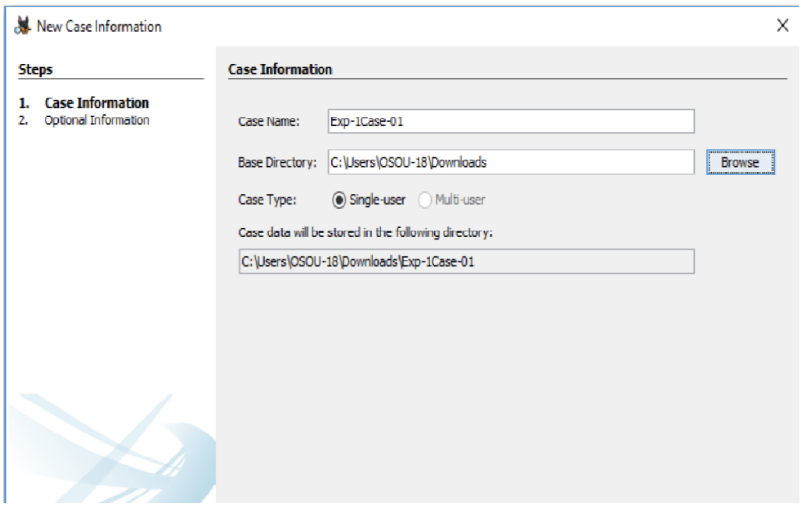
To perform a live forensics case investigation using Autopsy.

PROCEDURE:

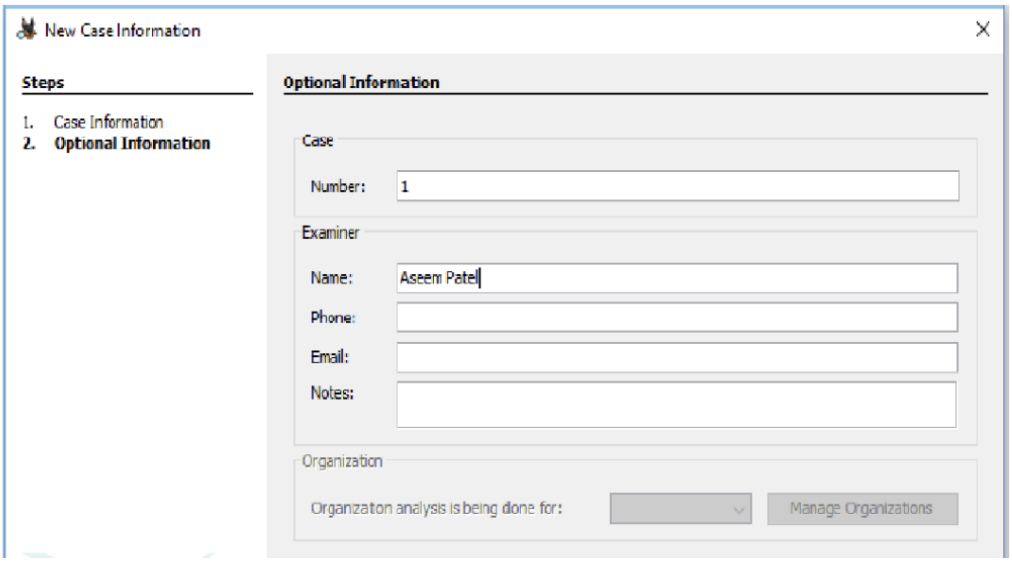
1. Download Autopsy and install it on your PC. Click the new case option.



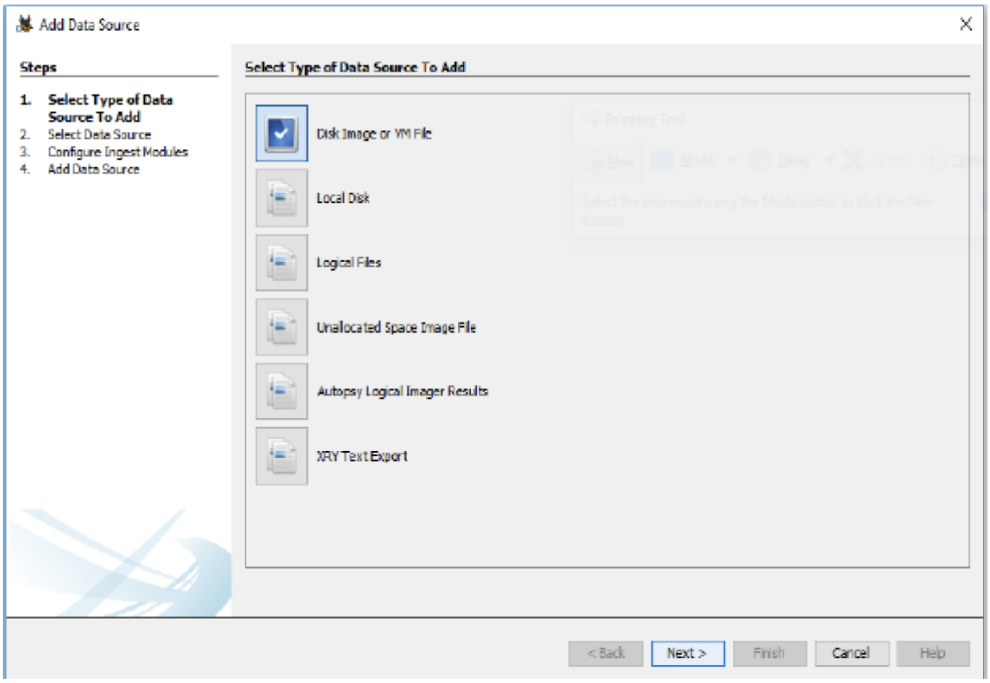
2. A new page will open. Enter the details in 'Case Name' and 'Base Directory' and choose the location to save the report, example, Autoreport. Then, click on next to proceed to the next step.

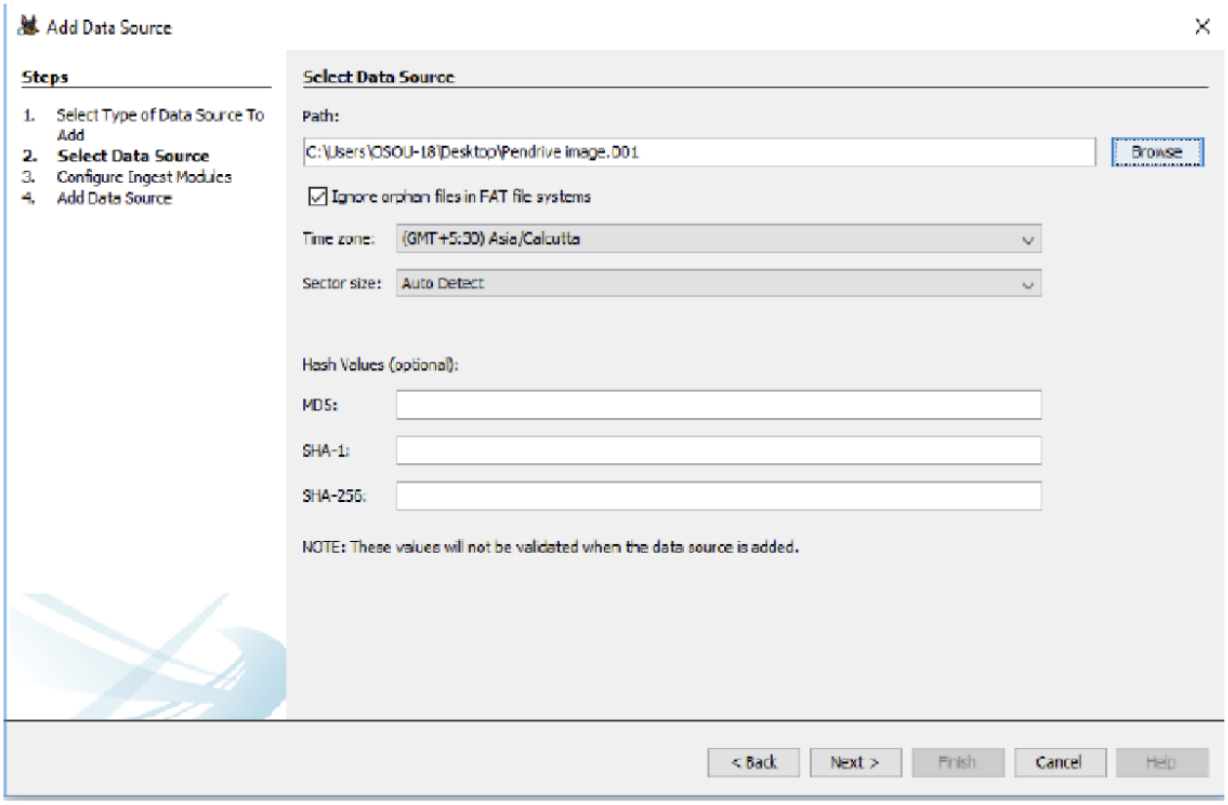


3. Here in the next step, you have to enter the case number and Examiner details and click on finish to proceed to the next step

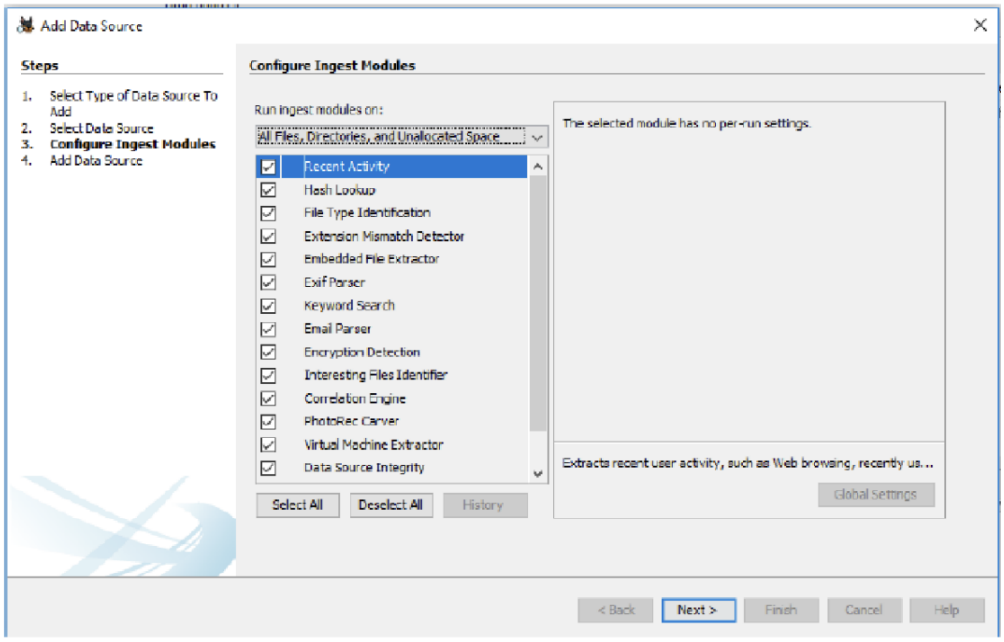


4. A new window will open. It will ask for a data source in Step 1. Select source type to add & browse the file Path and click on NEXT option to proceed further.

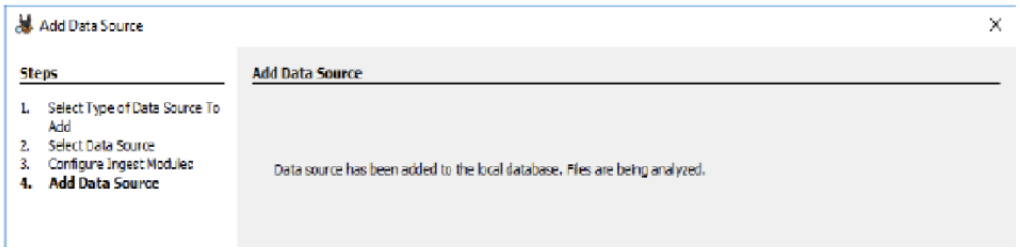




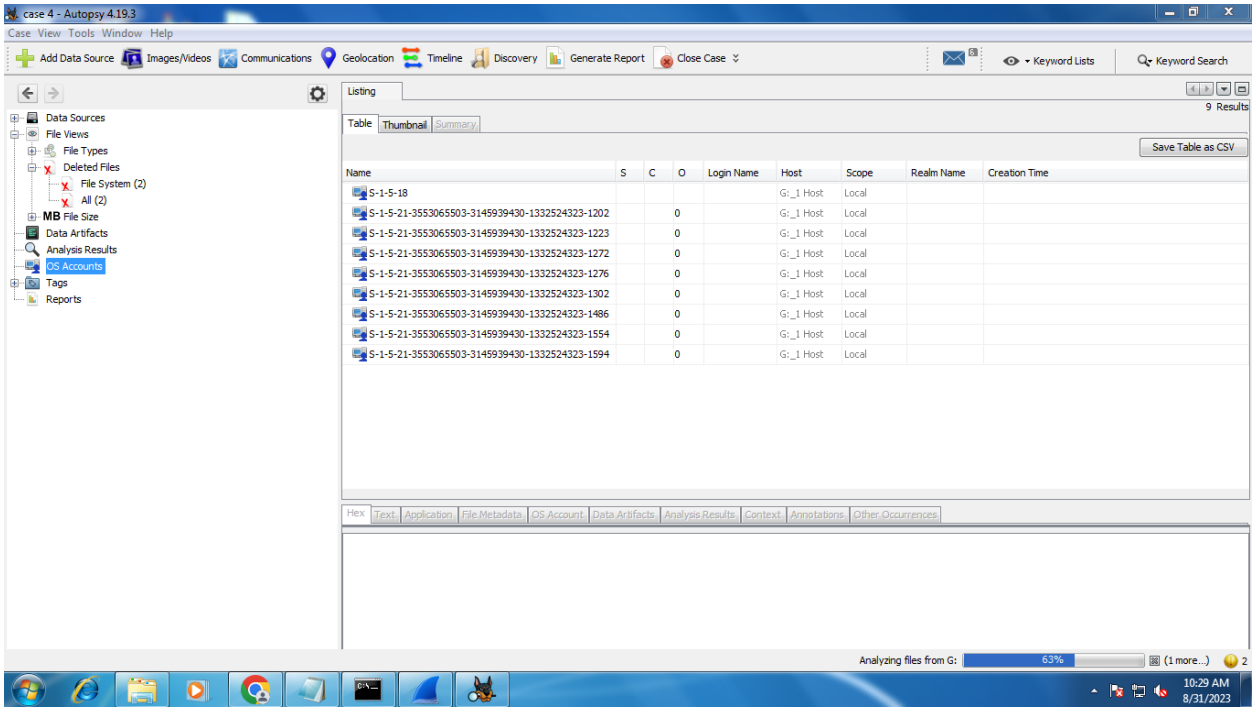
5. Configure ingest Modules. I have chosen all the modules as I am looking for complete information on evidence device or disk or system etc. and click next to proceed further.



6. In Add Data Source, just click on Finish to generate the report of the device and you can perform a complete investigation on the victim device or system or any other disk. It will process the data source and add it to the local database.



7. After Process completion, it will show the Forensic Investigation Report. Now click on Devices Attached option, it will show the list of the attached device with the system. Now click on EXIF Metadata (Exchangeable image file format for images, sound used by Digital Camera, Smartphone and scanner), click on Installed Programs to see the entire installed programs in the system, Click Operating System Information. It will show the entire operating system list, Now Select Operating System User Account Option. It will Display the name of all the user Accounts, Now click on Recent Documents Option, it will display the latest created or opened documents, Click Web Bookmarks Option to see all the bookmarks by system users in different browsers, To see web cookies, select web cookies option, To See Web Downloads, Click on Web Downloads option, To check internet History, click on Web History Option, To see the history of internet search, click on Web Search Option, To see the list of all email ids in the system, click on email address



RESULT:

A live forensic case investigation was successfully performed using the Autopsy tool.

Analyze Network Traffic Packets using Wireshark Packet Tracing

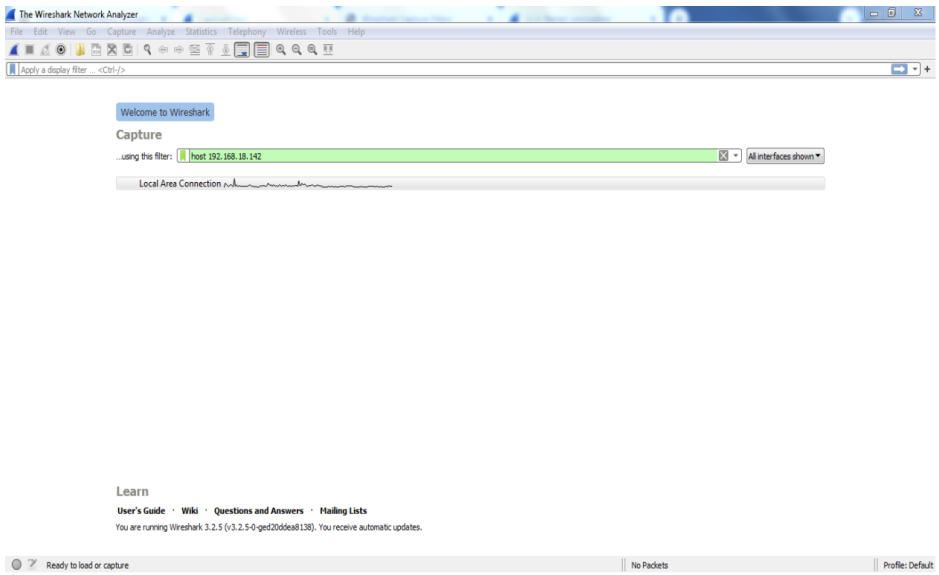
AIM:

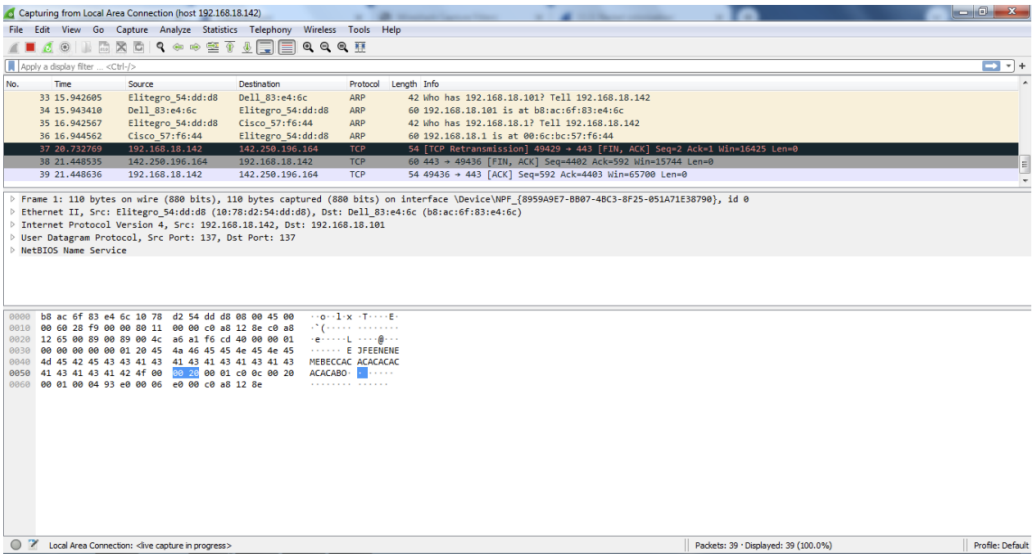
To analyze network traffic packets using wireshark packet tracing

PROCEDURE:

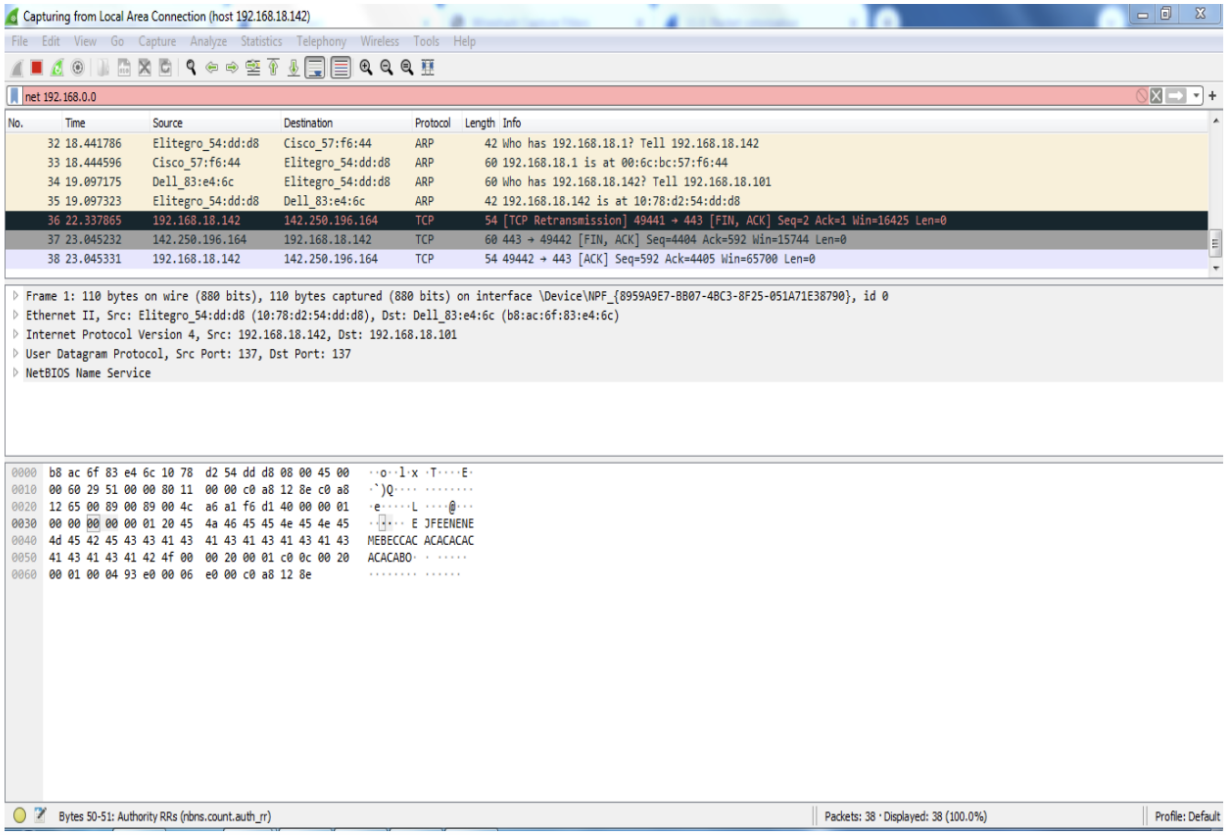
1. Open Wireshark, you see a screen showing you a list of all the network connections you can monitor.
2. It will have a capture filter field to only capture the network traffic you want to see.
3. Some of the best features of Wireshark are the capture filters, display filters and coloring rules.
4. Capture filters limit the captured packets by the chosen filter
5. Examples of Capture filter:

i. host IP-address: This filter limits the captured traffic to and from the IP address

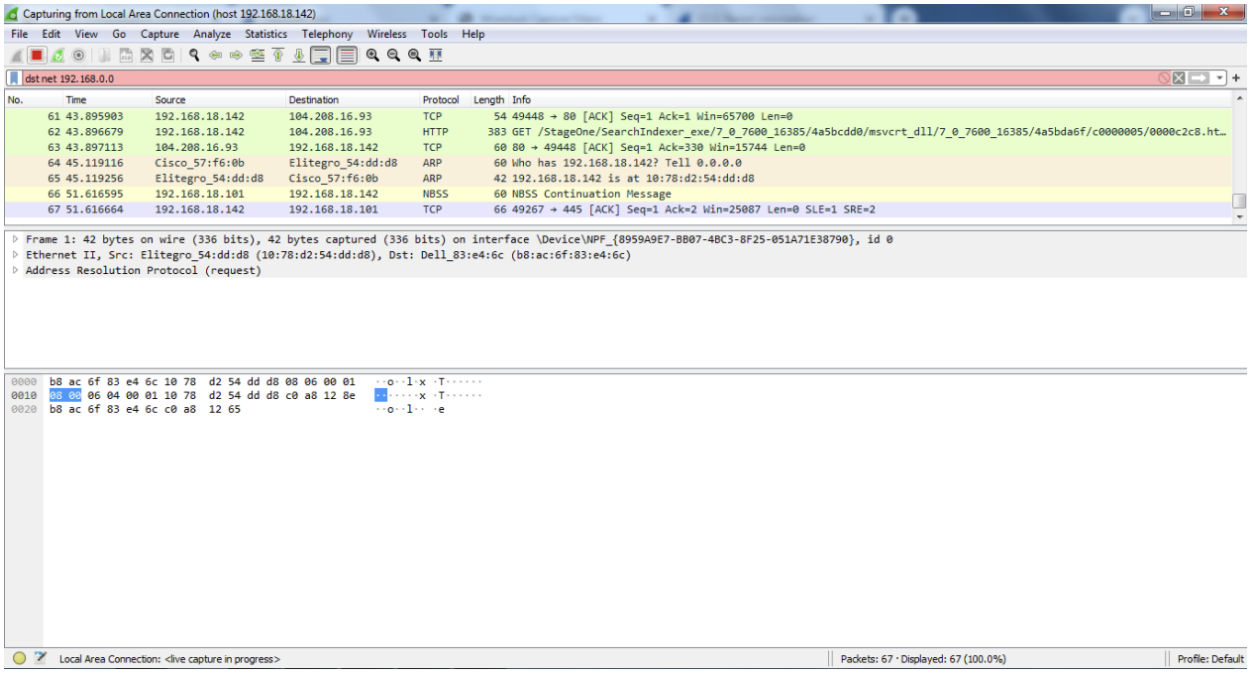




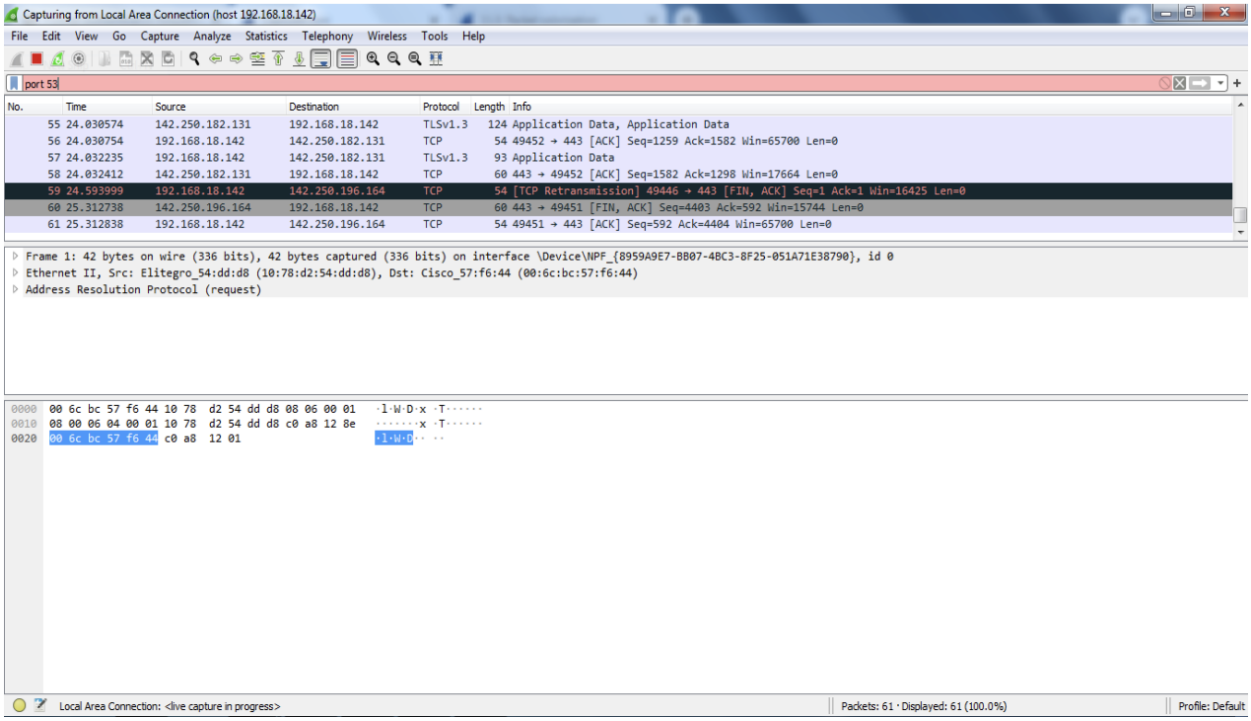
ii. net 192.168.0.0/24: This filter captures all traffic on the subnet



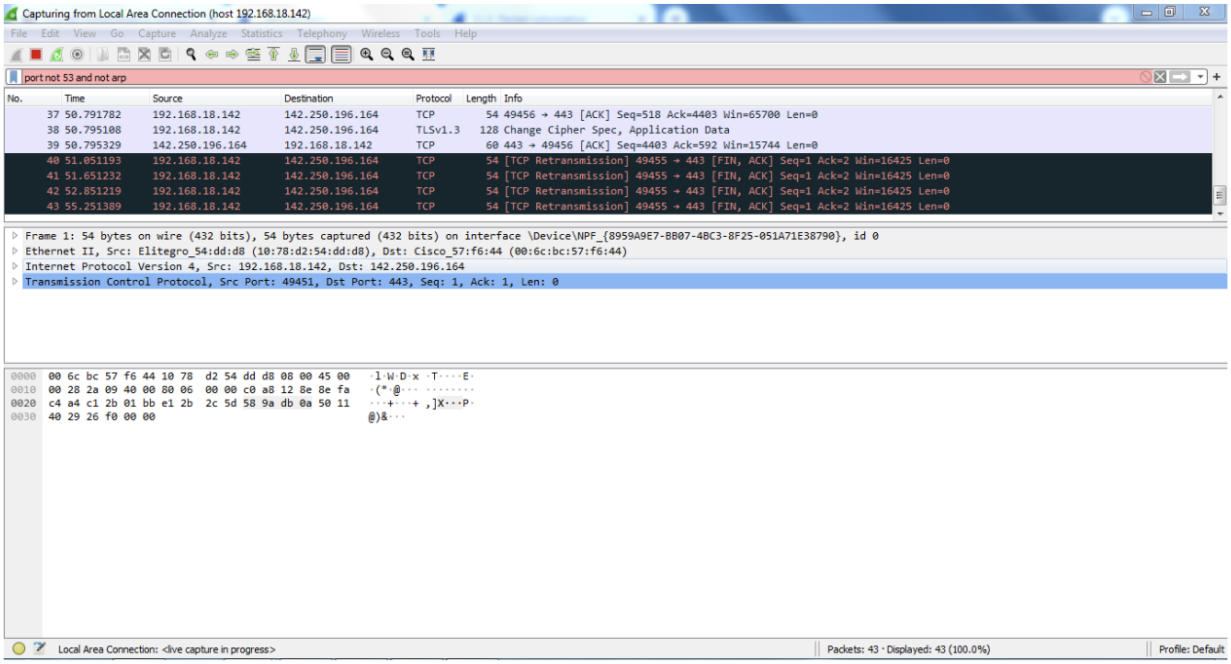
iii. dst host IP-address: Capture packets sent to the specified host



iv. port 53: Capture traffic on port 53 only

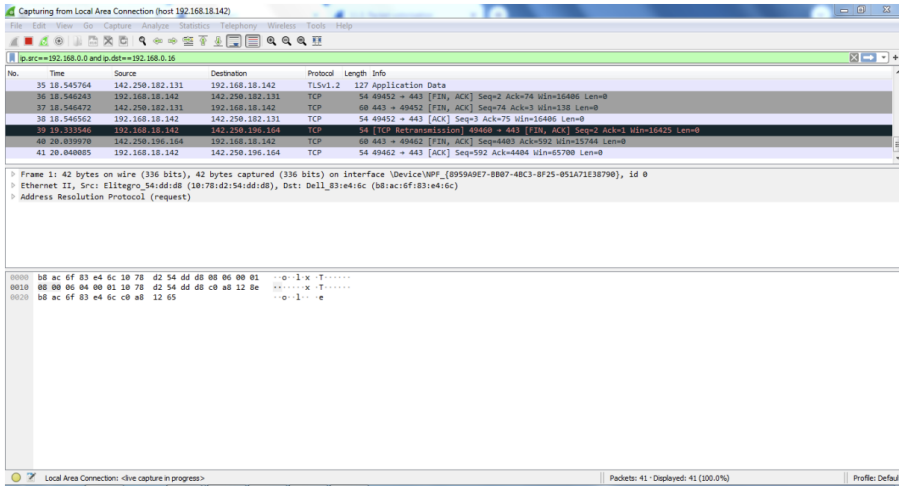


v. port not 53 and not arp: Capture all traffic except DNS and ARP traffic

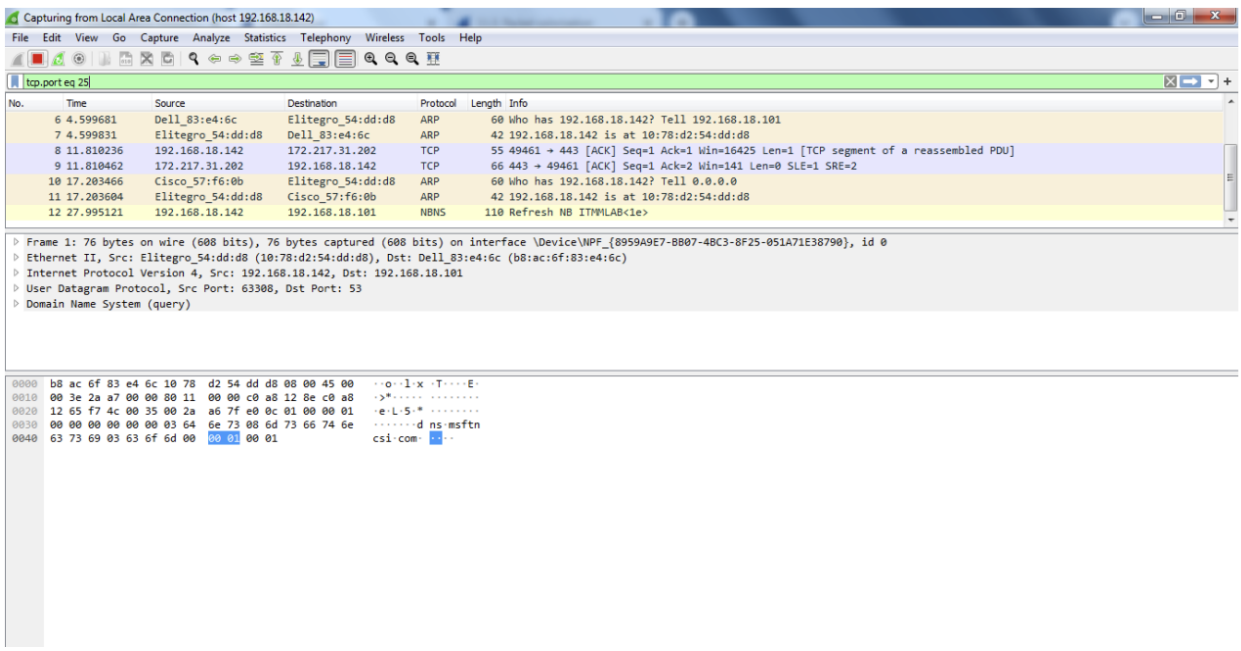


6. Wireshark display filters change the view of the capture during analysis
7. Examples of Display filters:

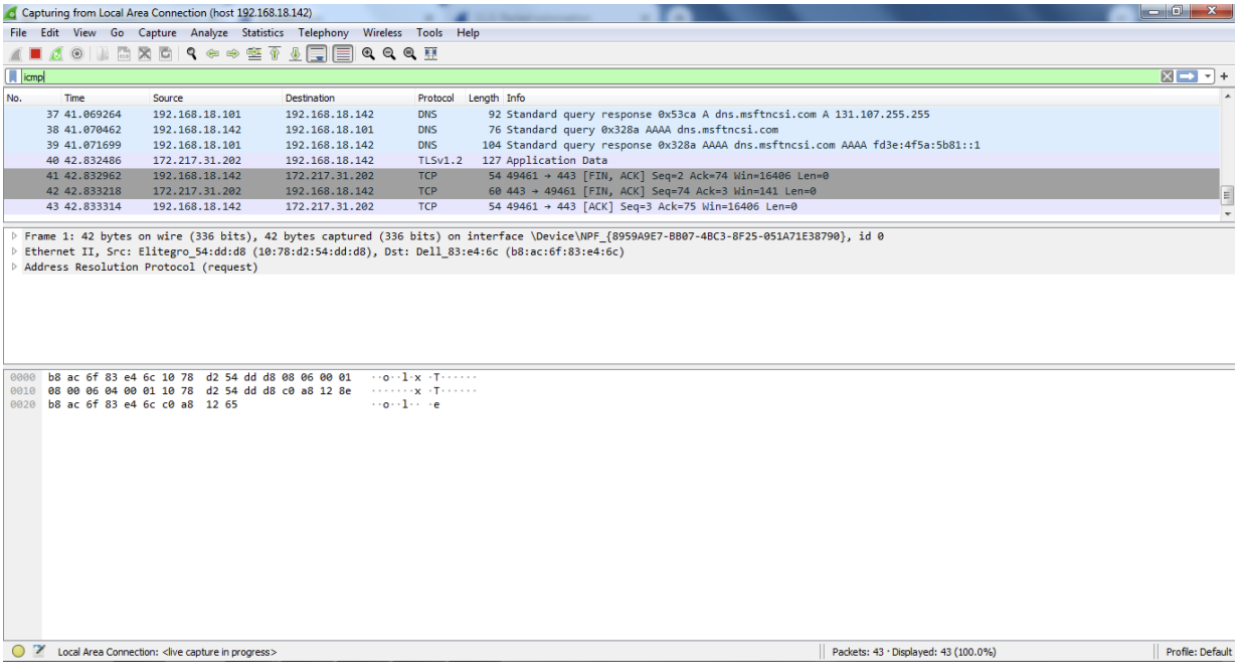
i. ip.src==IP-address and ip.dst==IP-address: This filter shows packets sent from one computer (ip.src) to another (ip.dst). You can also use ip.addr to show packets to and from that IP.



ii. tcp.port eq 25: This filter will show you all traffic on port 25, which is usually SMTP traffic

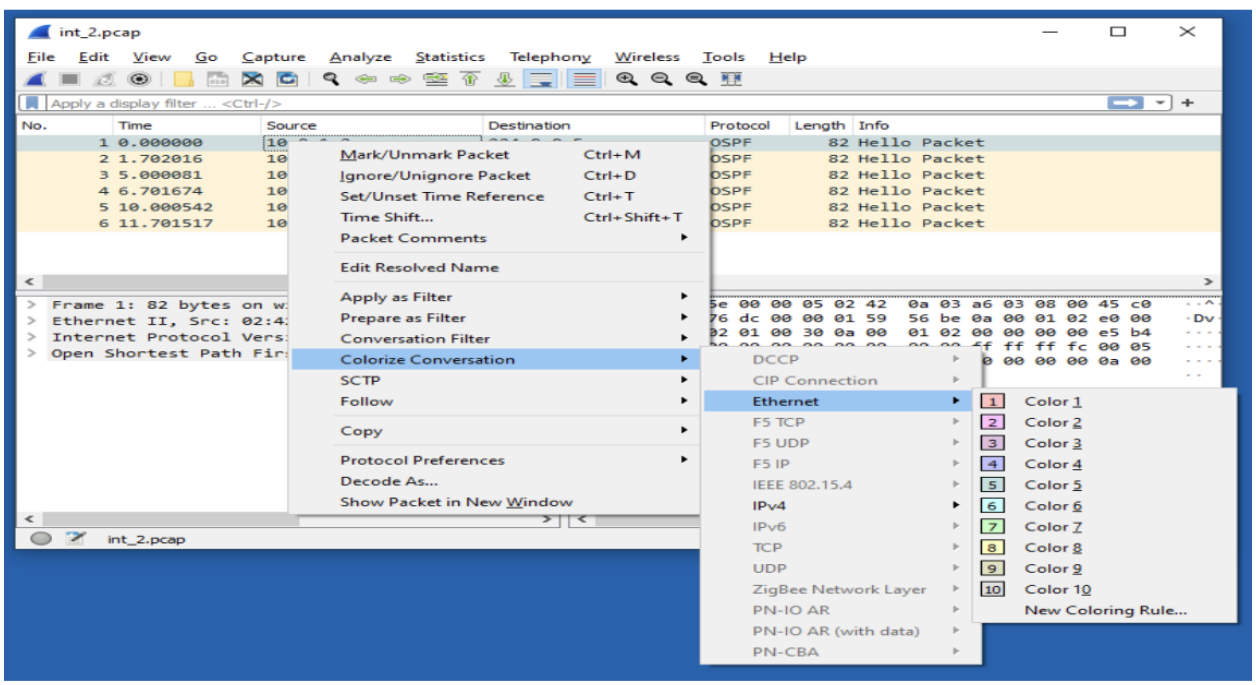


iii. icmp: This filter will show you only ICMP traffic in the capture, most likely they are pings

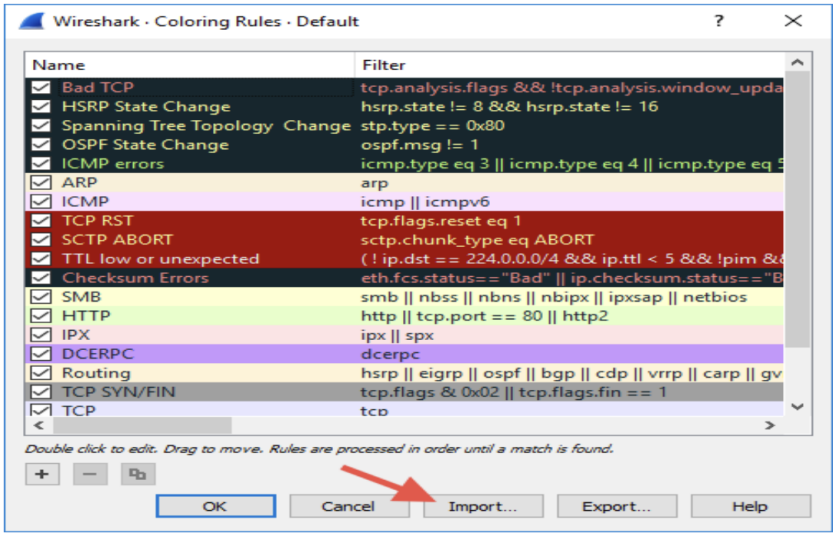


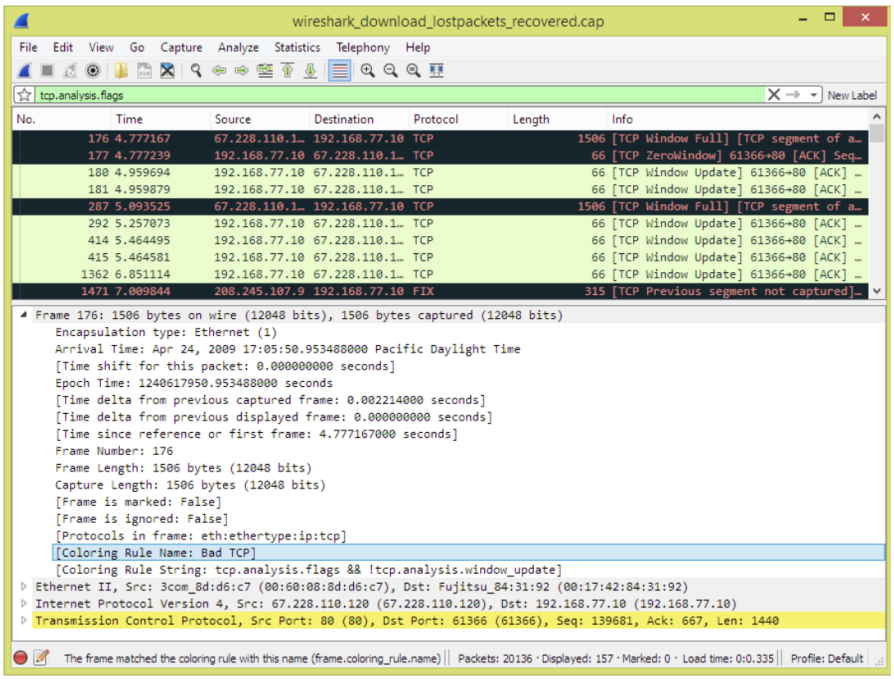
8. ColoringRules: A collection of coloring rules examples

There are two types of coloring rules in Wireshark: Temporary filters can be created by picking a packet in a capture file, right-clicking it, and hovering over Colorize Conversation. The slide-out menu will reveal options that allow you to define the type of conversation (IPv4, IPv6, TCP, etc)



Permanent filters can be created by using one of the coloring rules files, download it to your local machine, select View → Coloring Rules in Wireshark, and click the Import button.





RESULT:

Thus to analyze network traffic packets using wireshark packet tracing was executed successfully

IMPLEMENTATION OF ENCRYPTION AND DECRYPTION USING KLEOPATRA

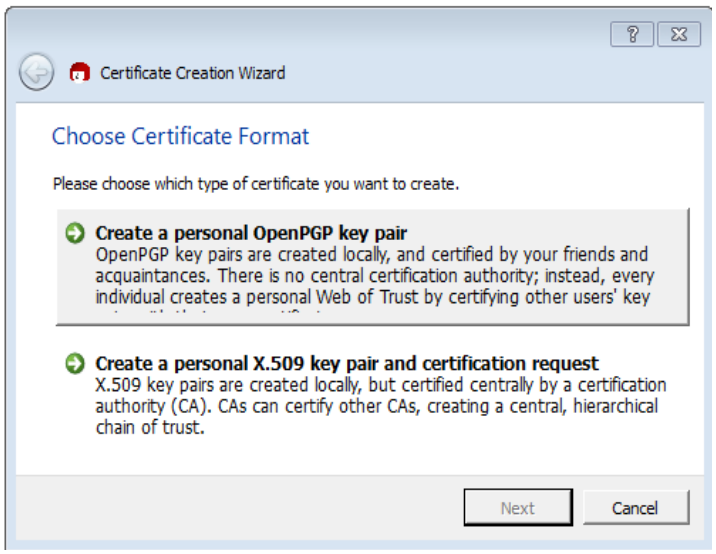
AIM:

To implement encryption and decryption using Kleopatra tool

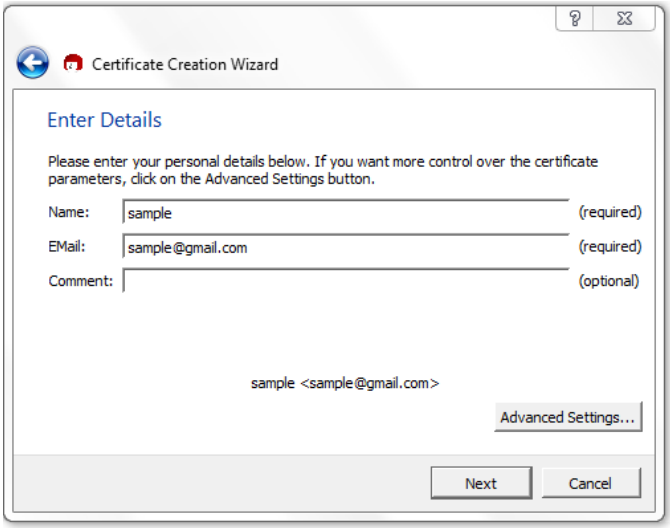
STEPS:

1. Create a digital certificate by providing the details:

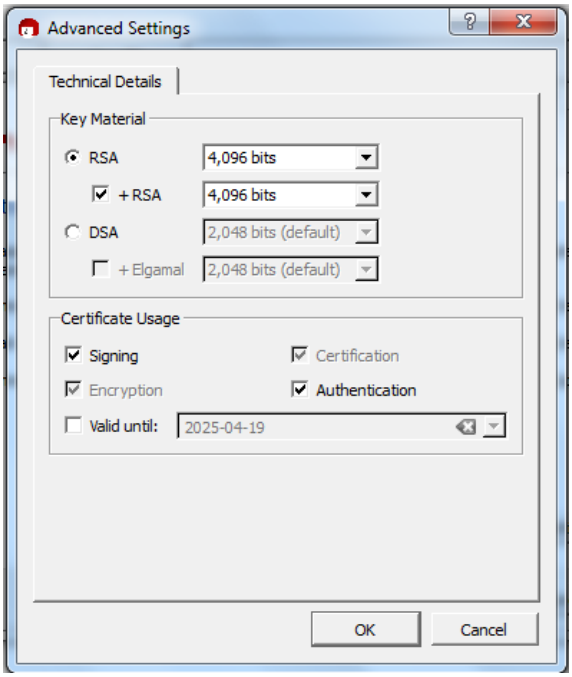
   File > New Certificate > Set the format to OpenPGP.



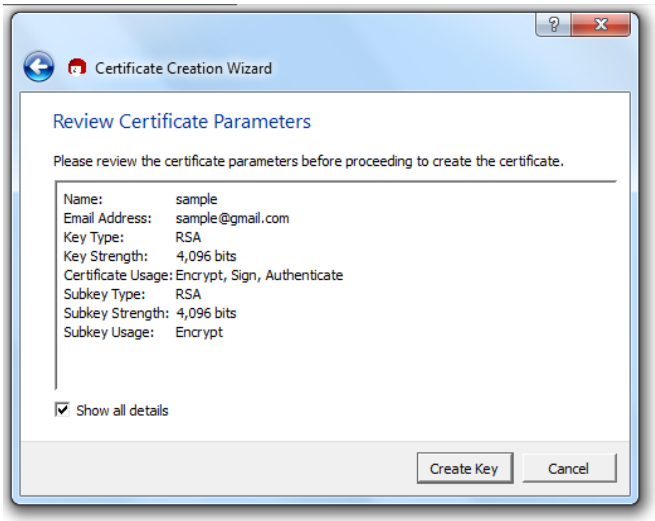
2. Enter name, email and comment (optional).



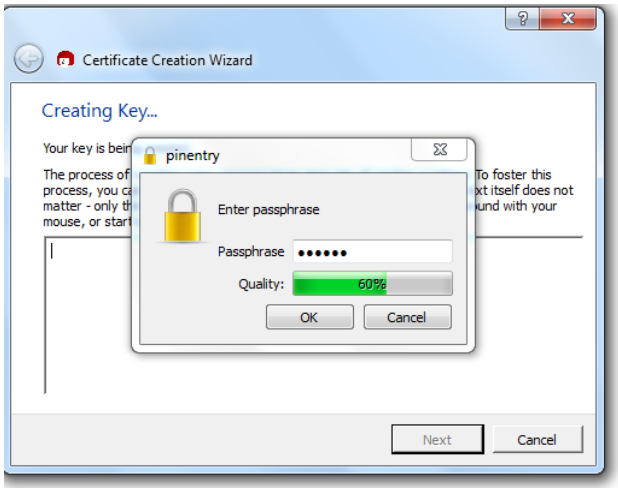
3. Choose the RSA key length (4096 bits).



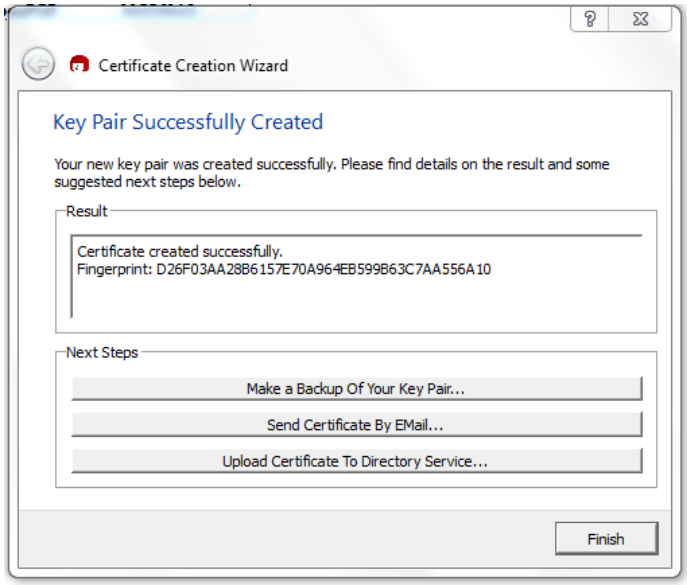
4. Choose signing, certification and authentication as mode. All details will be viewed. Proceed to key creation.

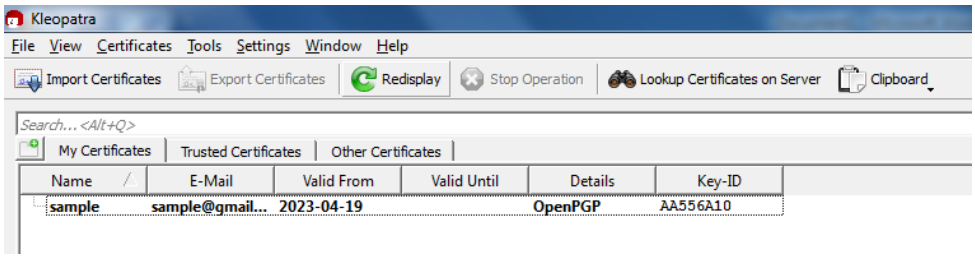


5. Set a passphrase which is the private key.

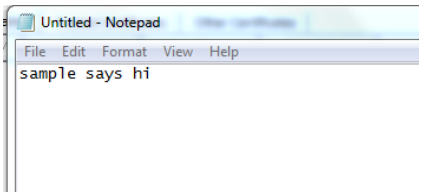


6. A public key is generated.

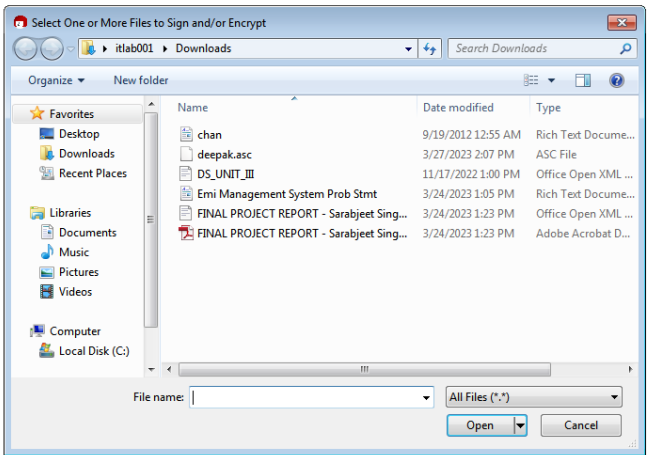


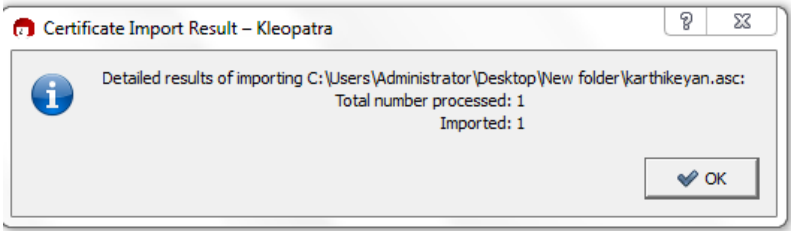


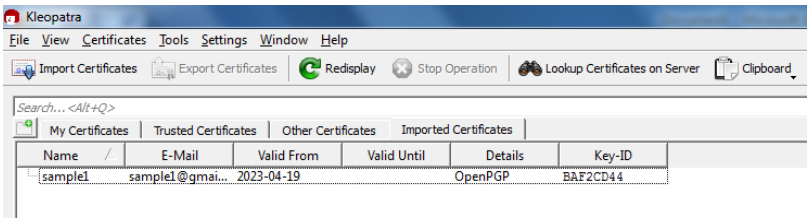
7. Create a message to encrypt



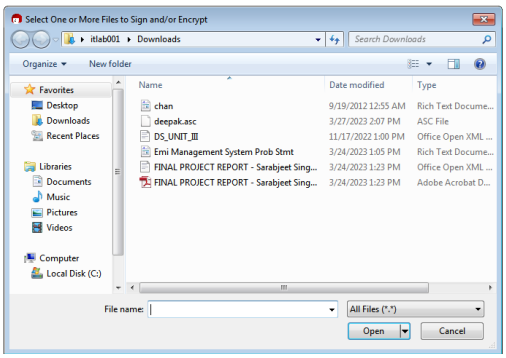
8. Import recipient's public key.
   IMPORT CERTIFICATE > CHOOSE THE FILE

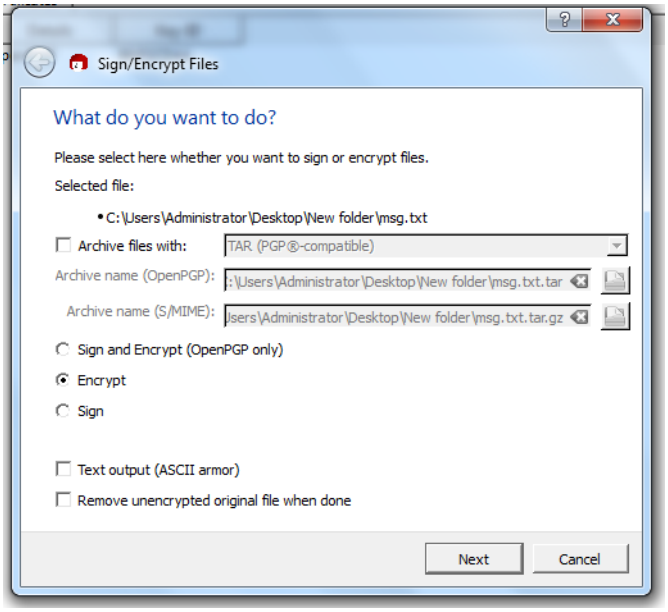




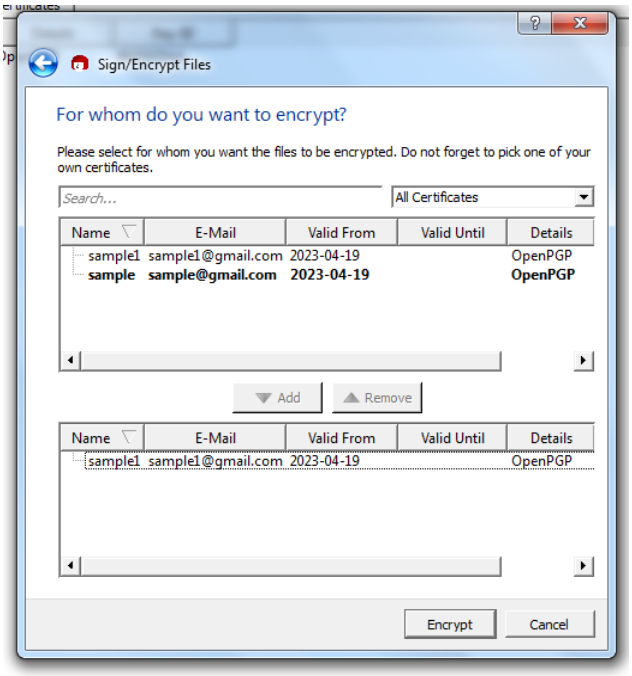


9. To Encrypt choose the file and set it to 'Encrypt'.
   FILE > SIGN / ENCRYPT > CHOOSE THE FILE.

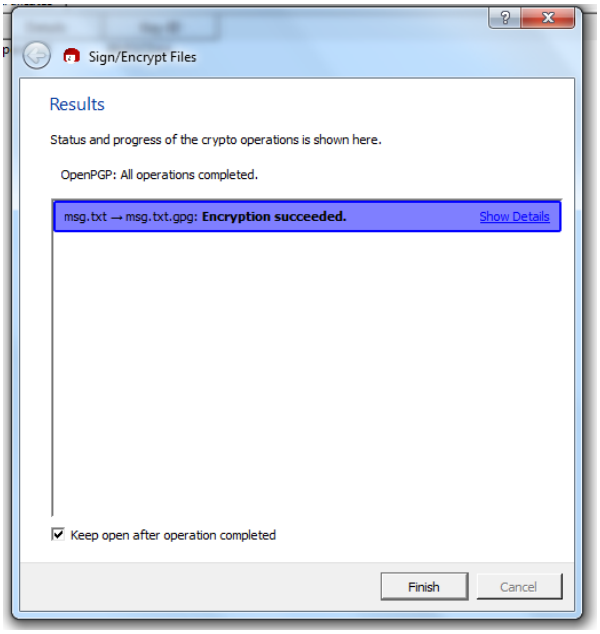




10. Add the recipient's public key to encrypt and proceed to encryption.

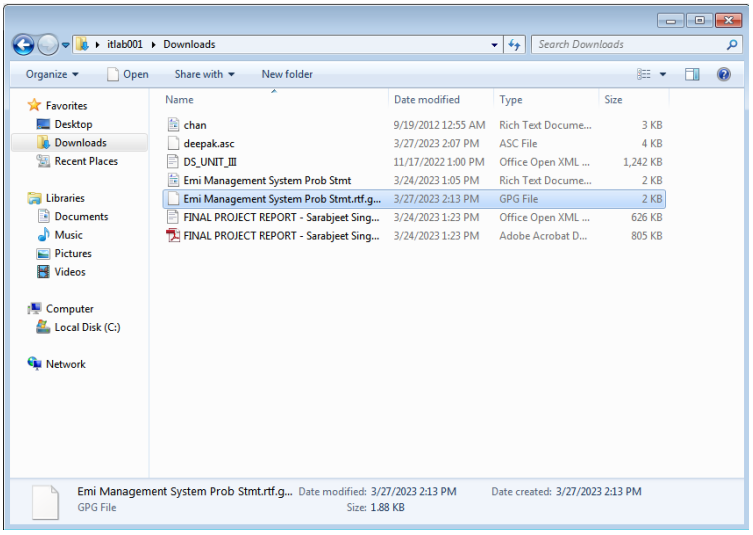


11. Encrypted files will be saved in GFG format.

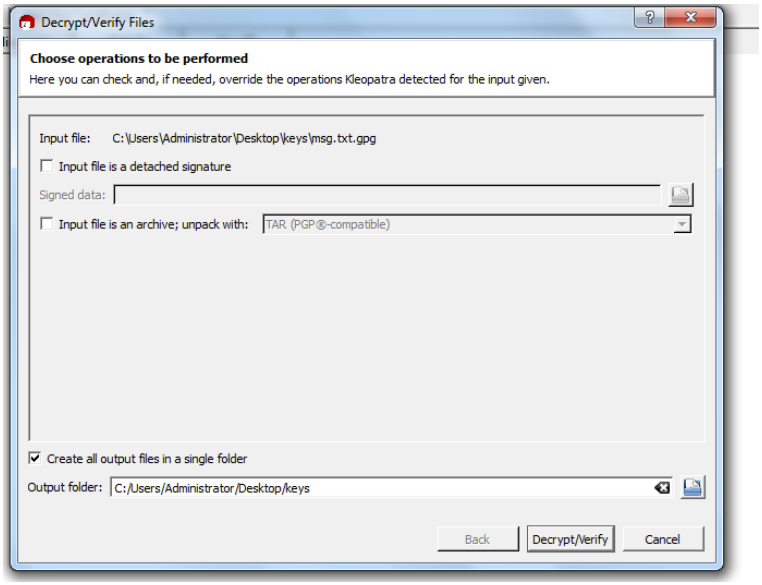


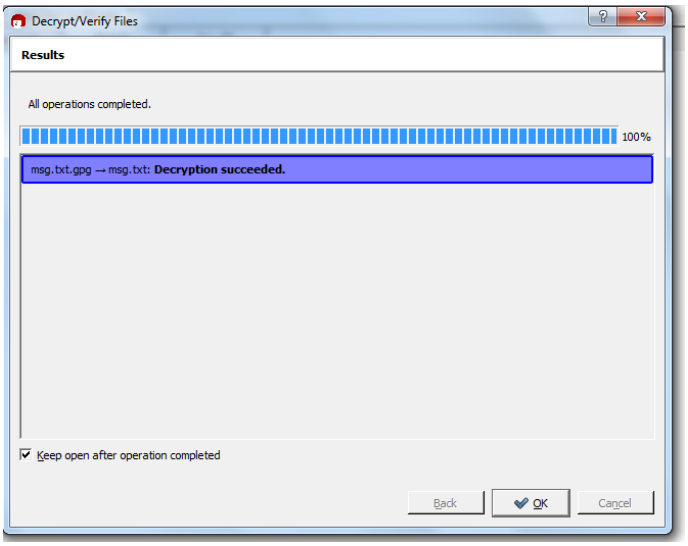
12. Send the file to the recipient.

13. Recipient downloads and select the file for decryption.
    FILE > DECRYPT / VERIFY FILES > CHOOSE THE FILE.



14. The file will be decrypted by clicking 'Decrypt/Verify' and the decrypted message will be saved in the output folder specified.





RESULT:

Implementation of encryption and decryption done successfully using the Kleopatra tool.

IMPLEMENTATION OF DIGITAL SIGNATURE USING KLEOPATRA

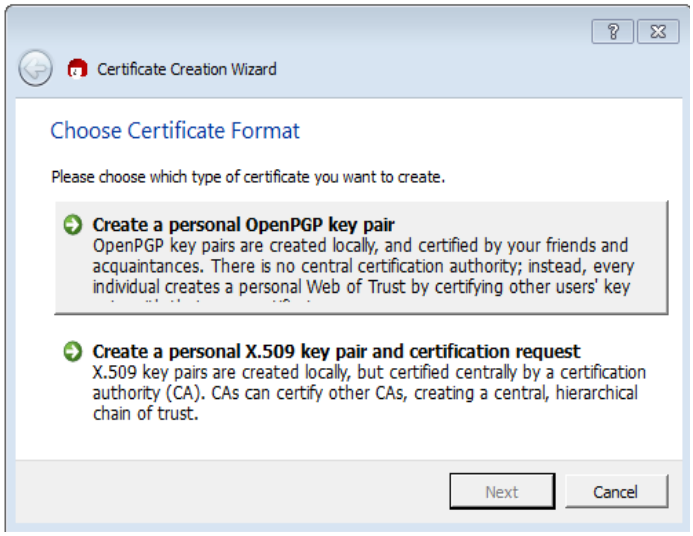
AIM:

To implement digital signature using Kleopatra tool

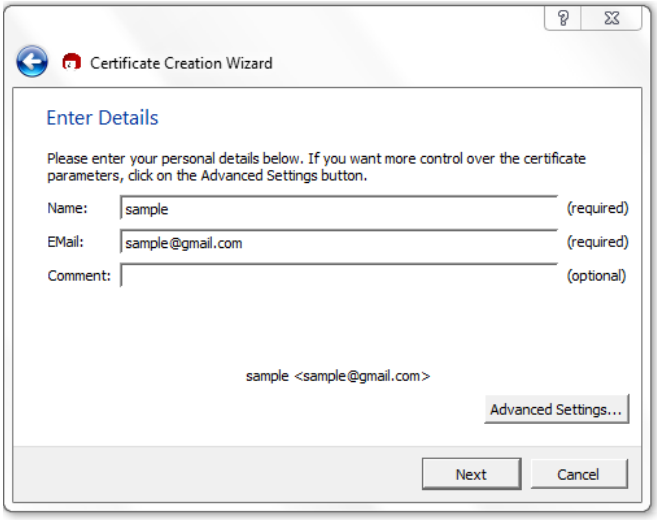
STEPS:

1. Create a digital certificate by providing the details:

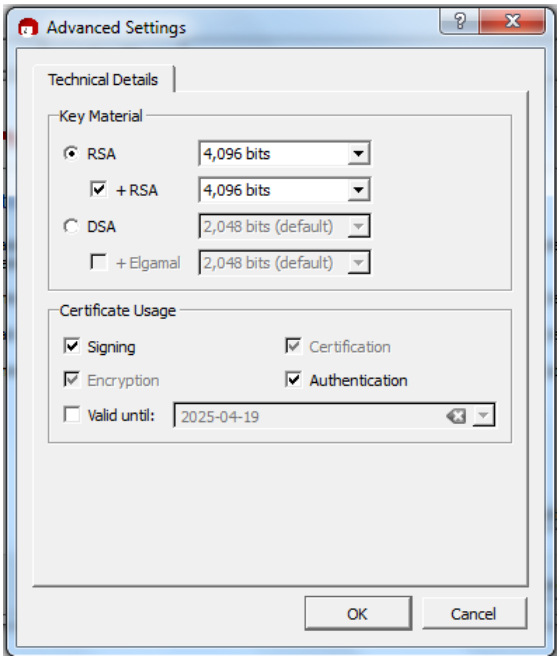
   File > New Certificate > Set the format to OpenPGP.



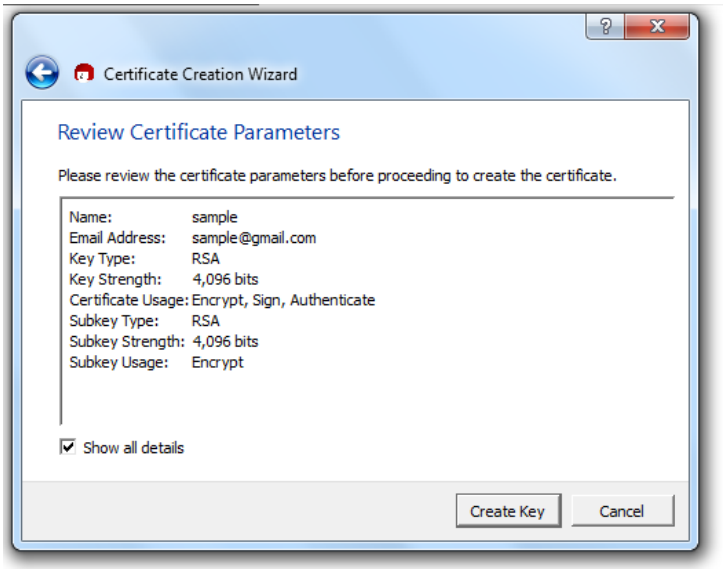
2. Enter name, email and comment (optional).



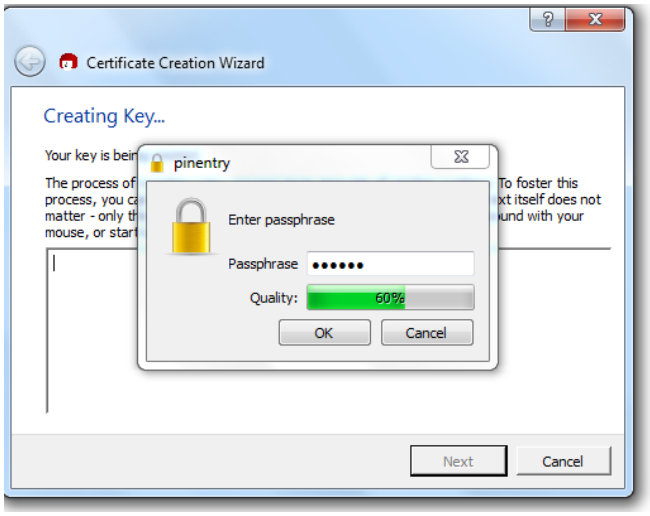
3. Choose the RSA key length (4096 bits).



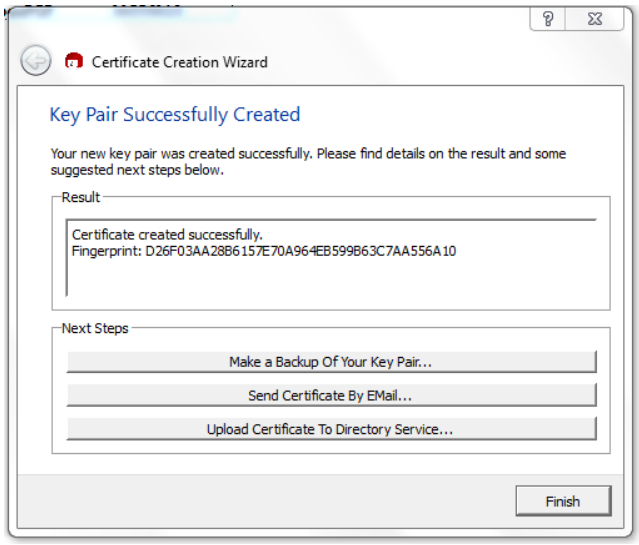
4. Choose signing, certification and authentication as mode. All details will be viewed. Proceed to key creation.

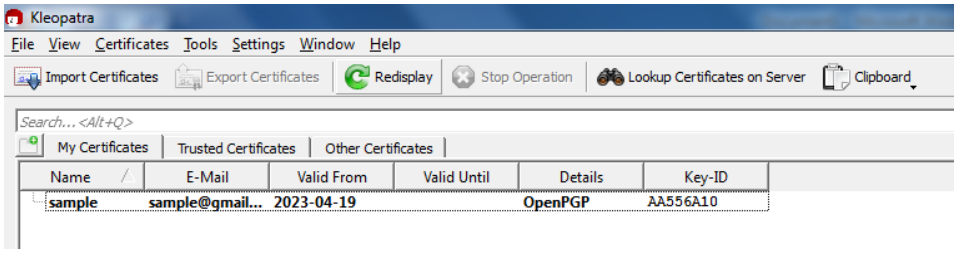


5. Set a passphrase which is the private key.

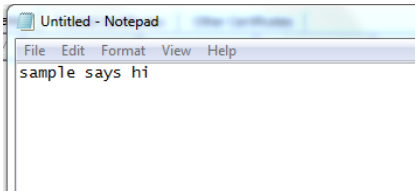


6. A public key is generated.

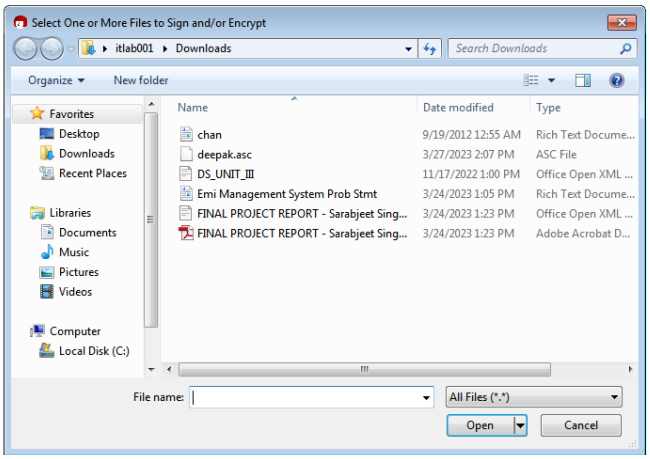


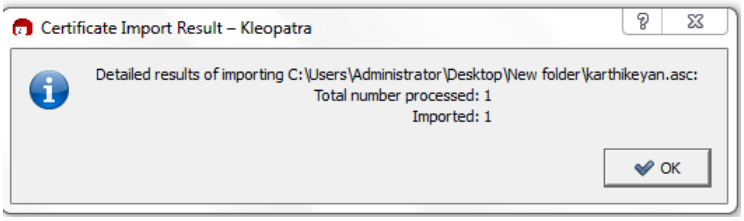


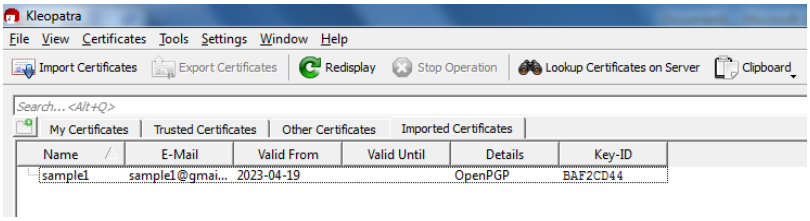
7. Create a message to encrypt



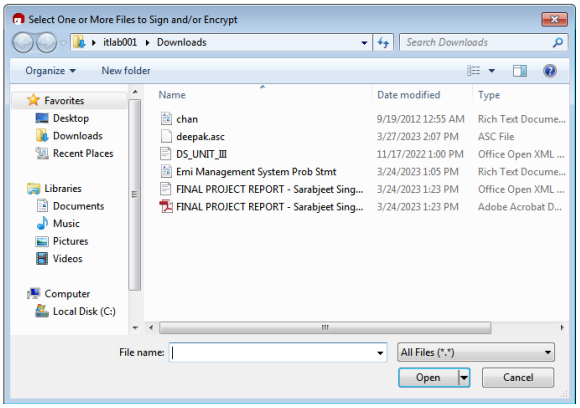
8. Import recipient's public key.
   IMPORT CERTIFICATE > CHOOSE THE FILE

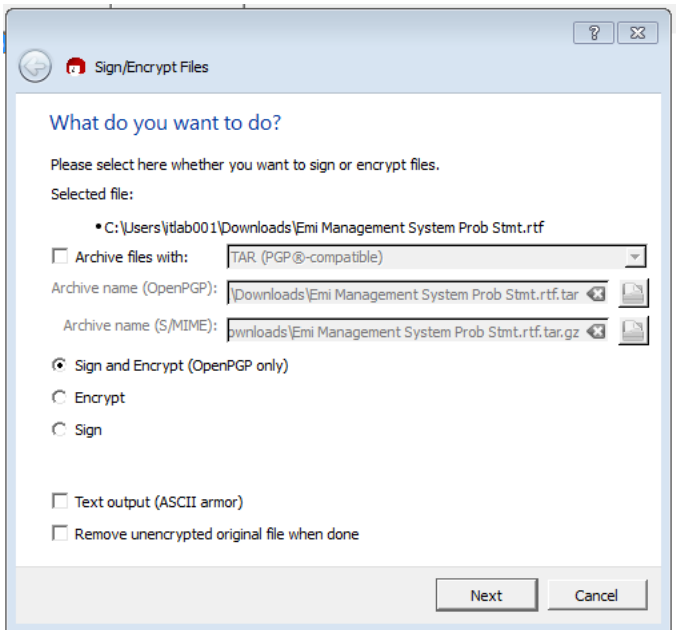


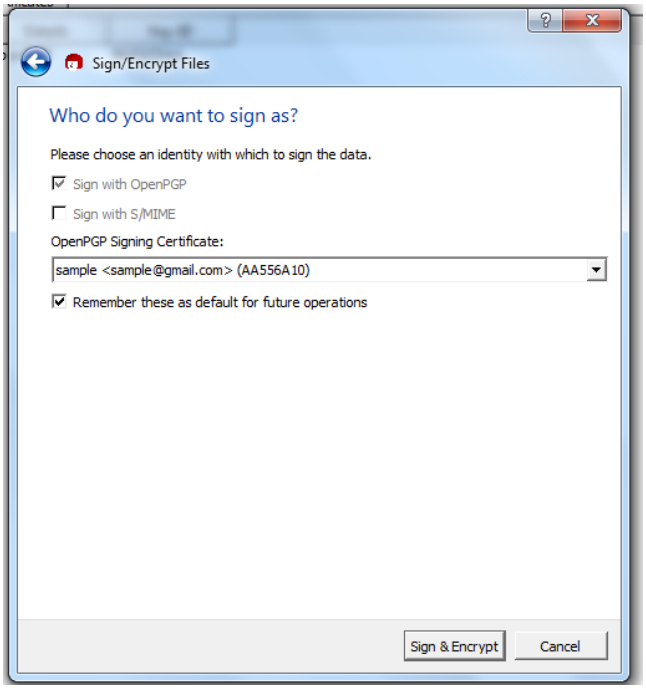




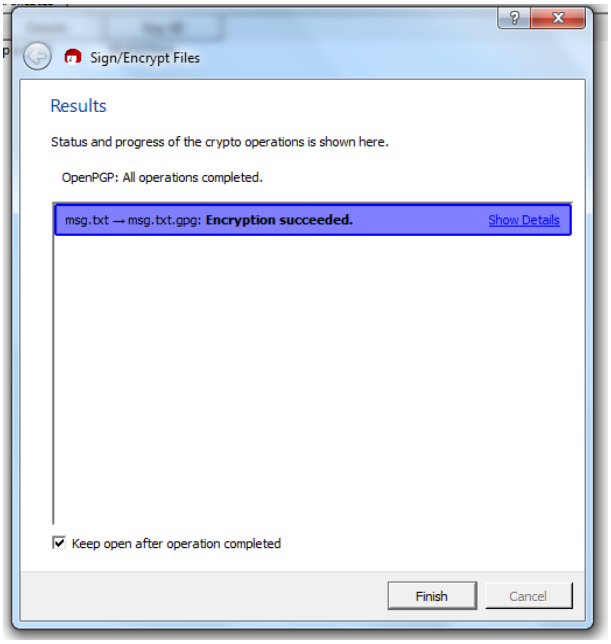
9. To Encrypt choose the file and set it to 'Sign and Encrypt'.
   FILE > SIGN / ENCRYPT > CHOOSE THE FILE.





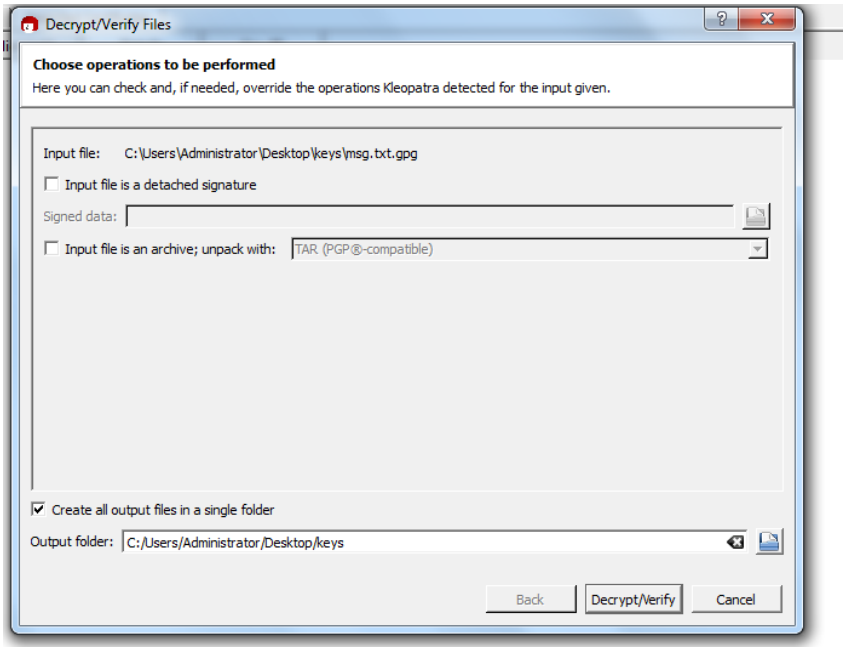


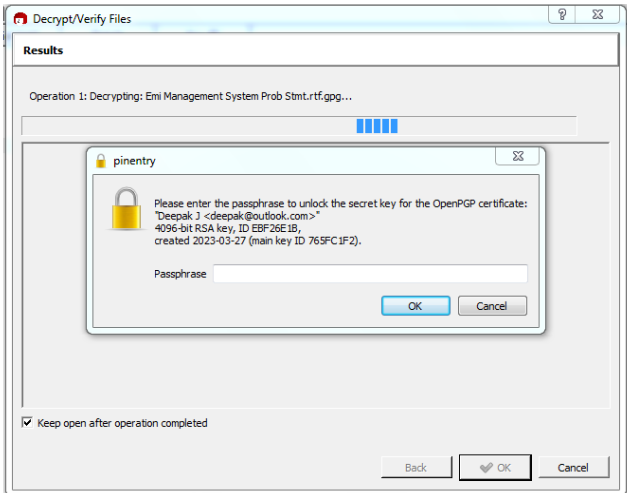
10. Encrypted files will be saved in GFG format.

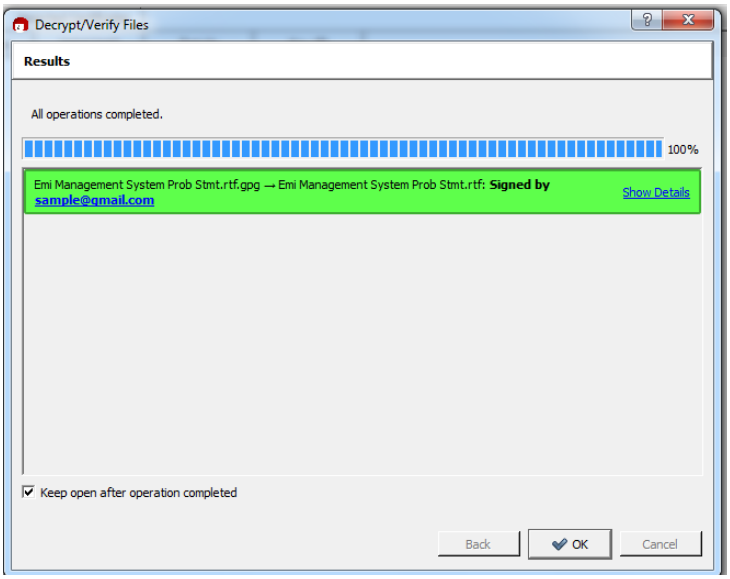


11. Send the file to the recipient.

12. Export sender's public key. The recipient must sign the encrypted file with the public key of the receiver and decrypt it using their passphrase.







RESULT:

Implementation of digital signatures done successfully using the Kleopatra tool.

Intrusion detection System using Snort Tool

AIM:

To use the Snort Intrusion Detection System tool to write a set of rules for access control.

STEPS:

Step-1:

1. Login as administrator and open common folder.
2. Paste the snort file to C drive.
3. Paste the winPcap-4-1-3 to the desktop and install
4. Open command prompt and type **ipconfig /all.**
5. Note down the IPv4 address that is **192.168.18.118.**

Step-2:

Open Snort > etc > snort.

In Step-1:

1. Add **ipvar HOME_NET 192.168.18.143**.
2. Add i**pvar EXTERNAL_NET any.**
3. Add **ipvar DNS_SERVERS $HOME_NET.** Add **$HOME_NET** to all other **SMTP_SERVERS.**
4. Add the following commands:
   var RULE_PATH C:\snort\rules
   var PREPROC_RULE_PATH C:\Snort\preproc_rules
   var WHITE_LIST_PATH C:\Snort \rules
   var BLACK_LIST_PATH C:\Snort\rules

In Step-2:

1. Add **config logdir:C:\Snort\log**

In Step-3:

config pcre_math_limit: 3500
config pcre_match_limit_recursion: 1500

In Step-4:

1. Add **dynamicpreprocessor** directory **C:\Snort\lib\snort_dynamicpreprocessor**
2. **dynamicengine C:\Snort\lib\snort_dynamicengine\Sf_engine.dll**
3. **#dynamicdetection directory/user/local/lib/snort_dynamicrules.** Add comment to this line.

In Step-6:

1. Include **C:\Snort\etc\classification.config.**
2. Include **C:\Snort\etc\reference.config.**

In Step-7:

1. Add **include $RULE_PATH/local.rules**.

In Step-9:

1. Add **include threshold.conf.**

Step-3:

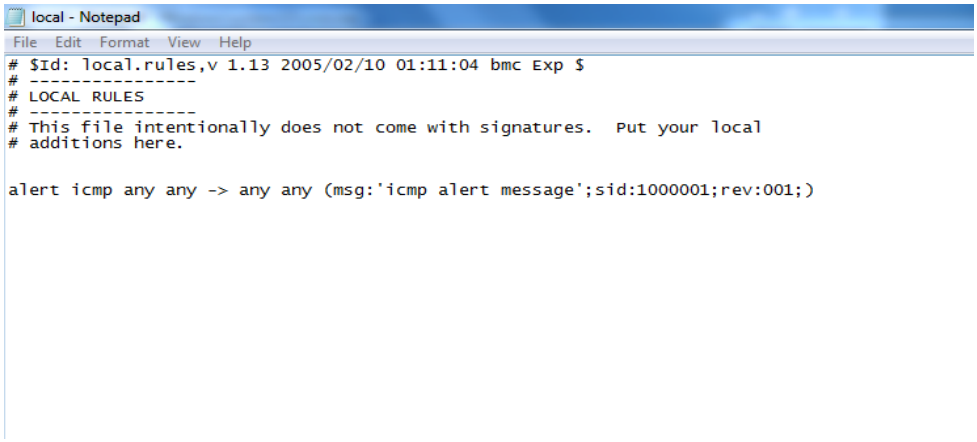
1. Go to Snort > rules > local in C drive.
2. Add the rules according to the given question. In this case add:
   alert icmp any any -> any any(msg: 'icmp alert message';sid:100001;rev:001;)
3. Comment all the other alert rules.

Step-4:

1. Open command prompt.
2. Change the directory from **C:\Administrator to C:\Snort.**
3. Type cd **C:\Snort\bin.**
4. Type **snort -c C:\snort\etc\snort.conf -A console -i 1.**

OUTPUT:

local.rules:



C:\Snort\bin> snort -c C:\snort\etc\snort.conf -A console -i 1

Commencing packet processing (pid=1256)

04/19-10:04:48.232881 [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:6c8e:ebb4:8d43:34f3 -> ff02:0000:0000:0000:000
0:0001:fffa:e9ad
04/19-10:04:48.233411 [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:15f4:6113:2bfa:e9ad -> ff02:0000:0000:0000:000
0:0001:ff43:34f3
04/19-10:05:10.410631 [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:2c5c:6180:957c:8d5e -> ff02:0000:0000:0000:000
0:0000:0000:0016
04/19-10:05:10.474756 [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:2c5c:6180:957c:8d5e -> ff02:0000:0000:0000:000
0:0000:0000:0016
04/19-10:05:46.028084 [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:2c0f:7065:0854:8ddd -> ff02:0000:0000:0000:000
0:0001:ff7e:c7fa
04/19-10:05:46.028798 [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:0456:8957:1b7e:c7fa -> ff02:0000:0000:0000:000

```
0:0001:ff54:8ddd
04/19-10:06:00.404842  [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:f486:1758:9afd:4fa2 -> ff02:0000:0000:0000:000
0:0000:0000:0016
04/19-10:06:00.564700  [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:f486:1758:9afd:4fa2 -> ff02:0000:0000:0000:000
0:0000:0000:0016
04/19-10:06:13.440205  [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:2c5c:6180:957c:8d5e -> ff02:0000:0000:0000:000
0:0000:0000:0016
04/19-10:06:13.469182  [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:2c5c:6180:957c:8d5e -> ff02:0000:0000:0000:000
0:0000:0000:0016
04/19-10:06:22.783891  [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:f486:1758:9afd:4fa2 -> ff02:0000:0000:0000:000
0:0000:0000:0016
04/19-10:06:23.074896  [**] [1:1000001:1] 'icmp alert message' [**] [Priority: 0
] {IPV6-ICMP} fe80:0000:0000:0000:f486:1758:9afd:4fa2 -> ff02:0000:0000:0000:000
0:0000:0000:0016
===============================================================================
Run time for packet processing was 206.180000 seconds
Snort processed 14448 packets.
Snort ran for 0 days 0 hours 3 minutes 26 seconds
   Pkts/min:        4816
   Pkts/sec:          70
===============================================================================
Packet I/O Totals:
   Received:       14509
   Analyzed:       14448 ( 99.580%)
    Dropped:           0 (  0.000%)
   Filtered:           0 (  0.000%)
Outstanding:          61 (  0.420%)
   Injected:           0
===============================================================================
SMTP Preprocessor Statistics
  Total sessions                          : 0
  Max concurrent sessions                       : 0
===============================================================================
dcerpc2 Preprocessor Statistics
  Total sessions: 0
===============================================================================
SSL Preprocessor:
   SSL packets decoded: 82
         Client Hello: 0
         Server Hello: 11
         Certificate: 1
```

```
        Server Done: 3
   Client Key Exchange: 1
   Server Key Exchange: 1
        Change Cipher: 10
            Finished: 0
    Client Application: 0
    Server Application: 31
              Alert: 1
  Unrecognized records: 45
  Completed handshakes: 0
       Bad handshakes: 0
     Sessions ignored: 23
    Detection disabled: 0
===========================================================================
SIP Preprocessor Statistics
  Total sessions: 0
===========================================================================
Reputation Preprocessor Statistics
  Total Memory Allocated: 0
===========================================================================
Snort exiting
```

RESULT:

Implementation of Intrusion Detection System using Snort tool done successfully.

Implementation of Password and User management System

AIM:

To implement user authentication and management systems using python.

ALGORITHM:

1. Import necessary libraries and modules:
    - import getpass for password input without echoing.
    - import hashlib for password hashing.
    - import csv for working with CSV files.
    - import re for regular expressions.
    - from pathlib import Path for file path management.
2. Create or open a CSV file (user.csv) to store user data, and handle cases where the file does not exist.
3. Define a function validate_password(password) to check if a password meets specific criteria (e.g., length, uppercase, lowercase, digits, special characters).
4. Define a function get_user_password() to interactively obtain a valid password from the user using getpass, calling validate_password until a valid password is provided.
5. Define a function sha256(password) to hash a given password using the SHA-256 algorithm.
6. Load user data from the CSV file into a list (e.g., user_data).
7. Check if the CSV file is empty; if so, add a header row.
8. Remove the header row (if present) from the user_data list.
9. Prompt the user to either:
    a. Log in (choose option 1), or
    b. Add a new user (choose option 2).
10. If the user chooses to log in (option 1):
    a. Prompt the user for their username.
    b. Use get_user_password() to get and hash the entered password.
    c. Iterate through user_data to check if the provided username and password hash match any existing records.
    d. Display "Login Successful" if a match is found, or "Login Failed" if not.
11. If the user chooses to add a new user (option 2):
    a. Prompt the user for their desired username.
    b. Use get_user_password() to get and hash the desired password.
    c. Iterate through user_data to check if the username already exists.
    d. If the username is found, display "User Already Exists" and registration fails.
    e. If the username is not found, append the new user to the CSV file and display "User added successfully."

CODE:

```
import getpass
import hashlib
import csv
import csv
import re
from pathlib import Path
filename = Path("user.csv")
filename.touch(exist_ok=True)
def validate_password(password):
    if len(password) < 8:
        return False
    if not re.search(r"[A-Z]", password):
        return False
    if not re.search(r"[a-z]", password):
        return False
    if not re.search(r"\d", password):
        return False
    if not re.search(r'[!@#$%^&*(),.?":{}|<>]', password):
        return False
    return True
def getPass():
    isValid = False
    pw = ""
    while not isValid:
        pw = getpass.getpass("Enter Password : ")
        isValid = validate_password(pw)
        if not isValid:
            print("Password not valid")
    return sha256(pw)
def sha256(pw):
    result = hashlib.sha256(pw.encode())
    hash1 = result.hexdigest()
    return hash1
csvFile = []
with open("user.csv", mode="r") as file:
    csvFile += csv.reader(file)
    file.close()
if len(csvFile) == 0:
    f = open("user.csv", "a")
    f.write("User,Password")
```

```
    f.close()
else:
    csvFile.pop(0)
signIn = int(input("Enter \n 1. Login \n 2. Add User : "))
if signIn == 1:
    user = input("User Name : ")
    pw = getPass()
    found = False
    for userList in csvFile:
        if userList[0] == user and userList[1] == pw:
            found = not found
            print("Login Successful")
            break
    if not found:
        print("Login Failed")
else:
    user = input("User Name : ")
    pw = getPass()
    found = False
    for userList in csvFile:
        if userList[0] == user:
            found = not found
            print("User Already Exist")
            break
    if found:
        print("Login Failed")
    else:
        f = open("user.csv", "a")
        f.write(f"\n{user},{pw}")
        f.close()
        print("User added successfully")
```

OUTPUT:

```
(userenv) kayee@KayEe-MacBook-Pro userauth % python3 authuser.py
Enter
 1. Login
 2. Add User : 2
User Name : karthi
Enter Password :
Password not valid
Enter Password :
User added successfully
(userenv) kayee@KayEe-MacBook-Pro userauth % python3 authuser.py
Enter
 1. Login
 2. Add User : 1
User Name : karthi
Enter Password :
Login Failed
(userenv) kayee@KayEe-MacBook-Pro userauth %
```

RESULT:

Implementation of user authentication and management system done successfully

Implementation of File Header Identification

AIM:

To write a program to find file header information.

ALGORITHM:

1.Start
2.Get the file name from the user
3. Split and take the extension of the file name.
4. Encode the extension.
5. Convert it to its equivalent hex value using hex()
6. Display the hex value.
7. Now convert the hex value to bytes.
8. Decode the hex value to its string value
9. Display the output.
10. Stop

CODE:

```
word = input("Enter File name: ")

word = word.split(".")[-1]

enc = word.encode()

print("Encoded Value: " + enc.hex())

actual = bytes.fromhex(enc.hex())

print("Decoded String: " + actual.decode())
```

OUTPUT:

```
C:\Users\Administrator\Desktop\pyy>python py2.py
Enter File name: file1.txt
Encoded Value: 747874
Decoded String: txt

C:\Users\Administrator\Desktop\pyy>python py2.py
Enter File name: file2.pdf
Encoded Value: 706466
Decoded String: pdf
```

RESULT:

Thus a program to fetch file header information has been implemented and the output was verified.

ANALYZING DISK STRUCTURE AND FILE SYSTEMS USING THE SLEUTH KIT TOOLS

AIM:

To analyze Disk Structure and File System using Sleuth Kit Tools

COMMANDS:

Volume Layer Tools:

- **mmls**: The **mmls** command parses and displays the media management structures on the image file or disk (i.e., the partition table). It will clearly show unallocated spaces before, after, or between volumes.

```
user@forensics:~$ mmls 10-ntfs-disk.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors
  Slot Start End Length Description
00: Meta 0000000000 0000000000 0000000001 Primary Table (#0)
01: ----- 0000000000 0000000062 0000000063 Unallocated
02: 00:00 0000000063 0000096389 0000096327 NTFS (0x07)
03: 00:01 0000096390 0000192779 0000096390 NTFS (0x07)
04: ----- 0000192780 0000192783 0000000004 Unallocated
```

File System Layer Tools:

- **fsstat**: The **fsstat** command displays file system information. Data of particular interest in the output of this command vary depending on the file system being examined but may include volume names, data unit sizes, and statistical information about the state of the file system.

```
user@forensics:~$ fsstat ubnist1.casper-rw.gen3.aff
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: Ext3
Volume Name:
Volume ID: 9935811771d9768b49417b0b3b881787
Last Written at: Tue Jan 6 10:59:33 2009
Last Checked at: Sun Dec 28 12:37:56 2008
Last Mounted at: Tue Jan 6 10:59:33 2009
Unmounted properly
Last mounted on:
Source OS: Linux
Dynamic Structure
Compat Features: Journal, Ext Attributes, Resize Inode, Dir Index
InCompat Features: Filetype, Needs Recovery,
Read Only Compat Features: Sparse Super, Has Large Files,
Journal ID: 00
Journal Inode: 8
```

Data Unit Layer Tools:

- **blkstat**: The **blkstat** command displays information about a specific data unit. Generally, this will simply be allocation status; however, on Ext file systems, the block group to which the block is allocated is also displayed.

```
user@forensics:~$ blkstat ubnist1.casper-rw.gen3.aff 521
Fragment: 521
Allocated
Group: 0
```

- **blkls**: The **blkls** command lists details about data units. Blkls can also be used to extract all unallocated space of the file system.

```
user@forensics:~$ blkls ubnist1.casper-rw.gen3.aff > ubnist1.
  casper-rw.gen3.unalloc
user@forensics:~$ ls -lath ubnist1.casper-rw.gen3.unalloc
-rw-r----- 1 cory eng 331M Sep 2 20:36 ubnist1.casper-rw.gen3.
  unalloc
```

- **blkcak**: The **blkcat** command will stream the content of a given data unit to STD- OUT. This is similar in effect to using dd to read and write a specific block

```
user@forensics:~$ blkcat ubnist1.casper-rw.gen3.aff 521 | xxd |
  head
0000000: 0200 0000 0c00 0102 2e00 0000 0200 0000 .........
0000010: 0c00 0202 2e2e 0000 0b00 0000 1400 0a02 .........
0000020: 6c6f 7374 2b66 6f75 6e64 0000 0c00 0000 lost+found.....
0000030: 1400 0c01 2e77 682e 2e77 682e 6175 6673 ......wh..wh.aufs
0000040: 011e 0000 1400 0c02 2e77 682e 2e77 682e .........wh..wh.
0000050: 706c 6e6b 015a 0000 1400 0c02 2e77 682e plnk.Z......wh.
0000060: 2e77 682e 2e74 6d70 021e 0000 0c00 0402 .wh..tmp.......
0000070: 726f 6673 025a 0000 0c00 0302 6574 6300 rofs.Z.....etc.
0000080: 045a 0000 1000 0502 6364 726f 6d00 0000 .Z...cdrom......
0000090: 031e 0000 0c00 0302 7661 7200 013c 0000 ......var..<..
```

Meta Data Layer Tools:

- **istat**: The **istat** command displays information about a specific metadata structure (ownership, time information, block allocations, etc.). As always, the exact information displayed is file system dependent.

```
user@forensics:~$ istat ubnist1.casper-rw.gen3.aff 20
inode: 20
Allocated
Group: 0
Generation Id: 96054594
uid / gid: 0 / 0
mode: rrw-r--r--
size: 123600
num of links: 0
Inode Times:
Accessed:        Tue Jan 6 10:59:33 2009
File Modified:   Wed Jan 7 07:59:47 2009
Inode Modified:  Wed Jan 7 07:59:47 2009
Deleted:         Wed Dec 31 16:00:17 1969
Direct Blocks:
28680 0 0 0 0 0 0 28681
0 0 0 0 0 0 0 28683
0 0 0 0 0 0 28684 0
0 0 0 0 0 0 28685
Indirect Blocks:
28682
```

- **ils**: The **ils** command lists the metadata structures, parsing and displaying the embedded dates, ownership information, and other relevant information.The ils output can be tuned to view as much (or as little) data as necessary.

```
user@forensics:~$ ils
Missing image name
usage: ils [-emOpvV] [-aAlLzZ] [-f fstype] [-i imgtype] [-b
  dev_sector_size] [-o imgoffset] [-s seconds] image [images]
  [inum[-end]]
    -e: Display all inodes
    -m: Display output in the mactime format
    -O: Display inodes that are unallocated, but were sill open
        (UFS/ExtX only)
    -p: Display orphan inodes (unallocated with no file name)
    -s seconds: Time skew of original machine (in seconds)
    -a: Allocated inodes
    -A: Unallocated inodes
    -l: Linked inodes
    -L: Unlinked inodes
    -z: Unused inodes (ctime is 0)
    -Z: Used inodes (ctime is not 0)
     -i imgtype: The format of the image file (use '-i list' for
        supported types)
    -b dev_sector_size: The size (in bytes) of the device
       sectors
    -f fstype: File system type (use '-f list' for supported
       types)
    -o imgoffset: The offset of the file system in the image
       (in sectors)
    -v: verbose output to stderr
    -V: Display version number
```

If we wanted to list all inodes that are allocated or that have been used at some point, we can do so with the -a and -Z flags:

```
user@forensics:~$ ils -aZ ubnist1.casper-rw.gen3.aff
...
st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_
  crtime|st_mode|st_nlink|st_size
1|a|0|0|1230496676|1230496676|1230496676|0|0|0|0
2|a|0|0|1231268373|1230496676|1231268373|0|755|15|4096
7|a|0|0|1230496676|1230496676|1230496676|0|600|1|4299210752
8|a|0|0|1230496679|0|1230496679|0|600|1|16777216
11|a|0|0|1230496676|1230496676|1230496676|0|700|2|16384
12|a|0|0|1230469846|1230469846|1231311252|0|444|19|0
13|a|0|0|1230615881|1225321841|1230615881|0|755|9|4096
...
```

- **ifind**: The **ifind** command finds the metadata structure referenced by the provided file name or the metadata structure that references the provided data unit address.

  For example, to find the inode that owns block 28680:

```
user@forensics:~$ ifind -d 28680 ubnist1.casper-rw.gen3.aff
20
```

File Name Layer Tools:

- **fls:** The fls command lists file names (deleted and allocated). By default it does not traverse the entire file system so you will only see the root directory of the volume being examined.

```
user@forensics:~$ fls ubnist1.casper-rw.gen3.aff
d/d 11:      lost+found
r/r 12:    .wh..wh.aufs
d/d 7681:    .wh..wh.plnk
d/d 23041:   .wh..wh..tmp
d/d 7682:    rofs
d/d 23042:   etc
d/d 23044:   cdrom
d/d 7683:    var
d/d 15361:   home
d/d 30721:   tmp
d/d 30722:   lib
d/d 15377:   usr
d/d 7712:    sbin
d/d 13:      root
r/r * 35(realloc):    .aufs.xino
d/d 38401:    $OrphanFiles
```

- **ffns:** The ffnd command finds file names that reference the provided metadata number

```
user@forensics:~$ ffind ubnist1.casper-rw.gen3.aff 20
File name not found for inode
```

```
user@forensics:~$ ffind ubnist1.casper-rw.gen3.aff 19
/root/.pulse-cookie
user@forensics:~$ ffind ubnist1.casper-rw.gen3.aff 21
/root/.synaptic/lock
```

- **img_stat**: The **img_stat** command will display information about the image format, including any hash information and other case-relevant metadata contained in the image. This tool is generally only useful when executed against forensic image container types.

```
user@forensics:~$ img_stat ubnist1.casper-rw.gen3.aff
IMAGE FILE INFORMATION
--------------------------------------------
Image Type: AFF
Size in bytes: 629145600
MD5: 717f6be298748ee7d6ce3e4b9ed63459
SHA1: 61bcd153fc91e680791aa39455688eab946c4b7
Creator: afconvert
Image GID: 25817565F05DFD8CAEC5CFC6B1FAB45
Acquisition Date: 2009-01-28 20:39:30
AFFLib Version: "3.3.5"
```

RESULT:

Therefore, the file system and Disk Structure is analyzed using sleuth kit.