

List of Experiments

1. Perform an Experiment for port scanning with nmap

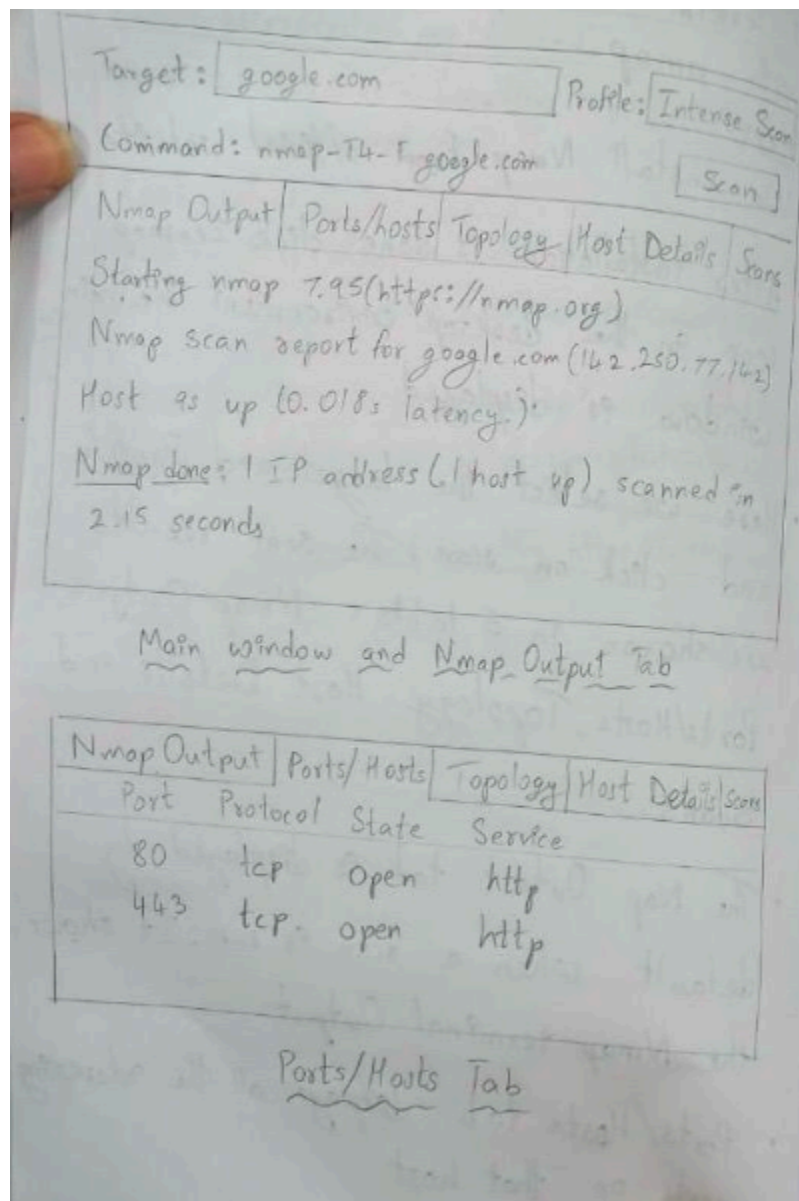
If you have permission to scan a network, here's a basic procedure for conducting a port scan using Nmap:

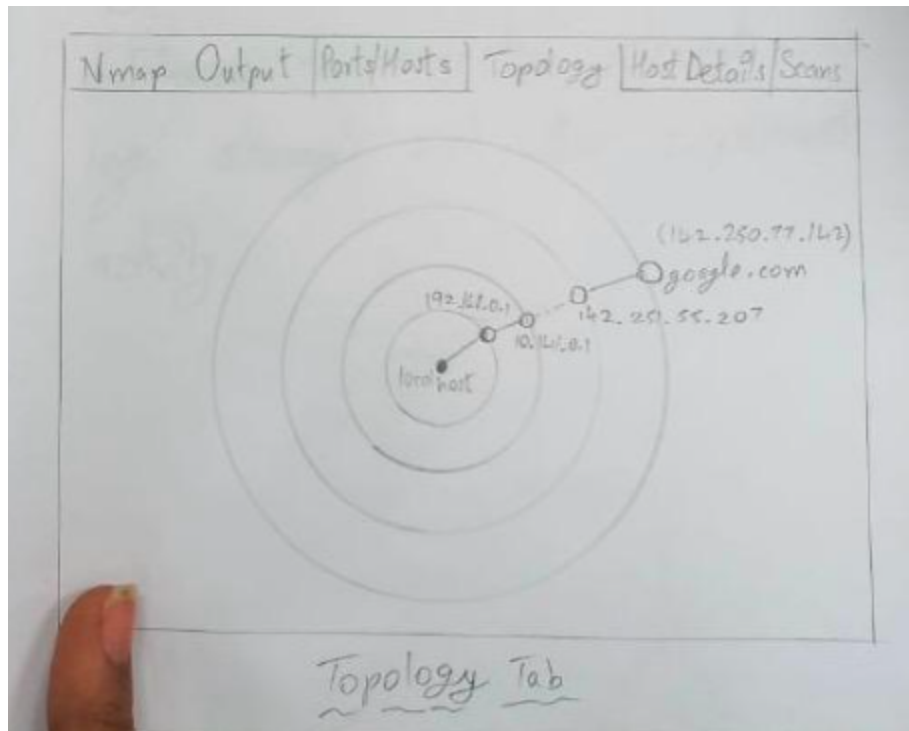
1. Install Nmap: Ensure that Nmap is installed on your system. You can download it from the official Nmap website or install it via package managers like apt (for Debian-based systems) or brew (for macOS).
2. Choose Target: Decide on the target IP address(ping) or hostname that you want to scan. Make sure you have permission to scan this target.
3. Run Nmap Scan: Open a terminal and run the Nmap command. Here's a simple example: `nmap [target]` Replace [target] with the IP address or hostname of the target you want to scan.
4. Review Results: After the scan completes, review the results to see which ports are open, what services are running on those ports, and any other information Nmap provides.
5. Interpret Results: Understand the implications of the open ports and services. Open ports might indicate services that could potentially be exploited if they are misconfigured or have known vulnerabilities.

Steps:

- * We Install Nmap from official website.
- * After installation is done, click Zenmap icon in the desktop environment. The main window is displayed.
- * Here we select the target and profile and click on scan. The scan results are shown in 5 tabs: Nmap Output, Ports/Hosts Topology, Host Details and Scans.
- * The Nmap Output tab is displayed by default when a scan is run. It shows the Nmap terminal Output
- * Ports/ Hosts tab displays all the interesting ports on that host.
- * Topology tab is an Interactive view of connections between hosts in a network Hosts are arranged in concentric rings.
- * Host Details Tab breaks all the information about a single host in a hierarchical Display.

* Scans tab shows all the scans that are aggregated to make up the network inventory.





2. Setup a honeypot(ids) and monitor the honeypot on the network

1. Choose a Honeypot Software: There are several honeypot solutions available, each with its own strengths and weaknesses. (kfsensor)
2. Set Up the Honeypot: Install the chosen honeypot software on a dedicated virtual machine or server. Follow the installation instructions provided by the honeypot project.
3. Configure the Honeypot: Depending on the honeypot software you've chosen, you may need to configure settings such as:
 - Listening ports (e.g., SSH, HTTP, FTP)
 - Emulated services and vulnerabilities
 - Logging options
4. Monitor the Honeypot: Monitor the honeypot for any activity. This can include:
 - Checking logs generated by the honeypot software for connection attempts, login attempts, and other suspicious activity.
 - Setting up alerts or notifications for specific events, such as successful logins or exploit attempts.

5. Analyze the Data: Regularly review the data collected from the honeypot to identify patterns and trends.

Start	Duration	Protocol	Sensor Port	Name
13-10-2024	295.82	TCP	443	115 HTTP
13-10-2024	30.92	TCP	80	115
13-10-2024	0.015	TCP	550	TCP Closed Port.

Main Window of
KFSensor

3. Install Jcrypt/Cryptool tool (or any other equivalent) and demonstrate Asymmetric, Symmetric crypto algorithm, Hash and Digital/PKI signatures

Demonstrate Cryptographic Functions:

Asymmetric Encryption (RSA):

- Generate a RSA private key.
- Extract the public key from the private key.
- Encrypt a message with the public key.
- Decrypt the message with the private key.

Symmetric Encryption (AES):

- Generate a random AES key.
- Encrypt a message with AES.
- Decrypt the message with AES.

Hash Function (SHA-256):

- Generate a hash of a file using SHA-256.

Digital Signature (RSA + SHA-256):

- Sign a file using RSA private key and SHA-256.

- Verify the signature using the corresponding public key.

Install Cryptool from the official website.

Assymmetric Encryption:

- * Click on Encrypt/Decrypt, then choose Asymmetric RSA Demonstration.
- * We can type prime no. p,q or generate the prime number then we click encrypt. This encrypts the text given using RSA algorithm .

Input text: hello

Encryption into ciphertext:

23366 07428 41521 41521 34310

Symmetric Encryption:

- * Select Encrypt Decrypt → Symmetric (classic) > Caesar.
- * The Caesar Key entry window opens where we enter the key 5. This will encrypt the file based on caesar cipher algorithm .
- * Input : hello

Caesar encryption of <Input>Key < F, OFFSET:0>mjqqat

Hash and Digital/PKI signatures:

- * We select Digital Signatures/PKI>Signature Generation.
- * Select the hash function, SHA -1. Then we generate a key using RSA algorithm. Hence we choose prime numbers p and q.
- * We provide certificate. This is done by entering the name, first name and pin to create a certificate. The certificate has been created successfully.

4.Generate minimum 10 passwords of length 12 characters using open SSL command

Step-1 OpenSSL download, under product select 2 nd URL in table(Version 3.2.1)

Step-2 Go to Program files, copy the OpenSSL path and edit the path in system variables

Step-3 In CMD:-Type 2 lines in null.docx

Code:

```
for /l %i in (1,1,10) do (openssl rand -base64 15 | openssl enc -base64 -A | powershell -Command "$input.Substring(0, 12)")
```

Step-4 We get 10 random keys as output

5.Client-Server Applications

i)Brute Force

ii)Caesar cipher

iii)cap to lower

iv)DES

v)MD5

6.Working with sniffers for monitoring network communication (Wireshark).

Wireshark is a powerful network protocol analyzer that allows you to capture and interactively browse the traffic running on a computer network. It supports a wide range of protocols and provides various features for deep inspection of network packets. Here's a basic overview of how to work with Wireshark:

1. **Installation:** First, you need to download and install Wireshark from the official website (<https://www.wireshark.org/>). It's available for Windows, macOS, and Linux.
2. **Capture Traffic:** Once installed, you can start Wireshark and select the network interface you want to capture traffic from. Then, click on the "Start" button to begin capturing packets.
3. **Filtering:** Wireshark captures a lot of traffic, so it's often useful to apply filters to focus on specific packets or protocols. You can use display filters to filter packets based on various criteria like IP addresses, ports, protocols, etc.
4. **Packet Inspection:** Wireshark allows you to inspect individual packets in detail. You can view packet headers, payloads, and other information. It also provides various analysis tools and statistics.
5. **Save and Export:** You can save captured packets to a file for later analysis or export them in various formats. Wireshark supports standard packet capture file formats like PCAP and PCAPNG.
6. **Advanced Features:** Wireshark offers many advanced features like packet coloring, packet marking, protocol dissectors, custom plugins, etc., to enhance your analysis capabilities.
7. **Security Considerations:** When using Wireshark to capture network traffic, be aware of the legal and ethical considerations. Make sure you have permission to

capture traffic on the network you are monitoring, and avoid capturing sensitive information like passwords or personal data.

Overall, Wireshark is an essential tool for network administrators, security professionals, and anyone interested in understanding and troubleshooting network protocols and traffic. It provides a wealth of information about network activity and helps in diagnosing network issues, analyzing security threats, and optimizing network performance

7.Using Snort, perform real time traffic analysis and packet logging.

1. **Open-Source IDS/IPS:** SNORT is a free and open-source intrusion detection and prevention system (IDS/IPS) offering real-time network traffic analysis and data packet logging.
2. **Rule-Based Detection:** It uses a rule-based language combining anomaly, protocol, and signature inspection to detect malicious activities like DoS/DDoS attacks, buffer overflows, and port scans.
3. **Alert System:** SNORT generates alerts when detecting malicious packets, providing real-time threat notifications to network administrators.
4. **Packet Logging:** Through its packet logger mode, SNORT logs network packets to disk in a hierarchical directory structure based on the IP address of the host network.
5. **Protocol Analysis:** It performs protocol analysis, enabling admins to capture and examine data in protocol layers, aiding in detailed threat analysis.
6. **OS Fingerprinting:** SNORT identifies the operating system of devices accessing the network using unique TCP/IP stack characteristics.
7. **Flexible Deployment:** It can function as a sniffer, IDS, or a full IPS solution to monitor, detect, and block attack vectors on IP networks.
8. **CGI Attack Detection:** SNORT is capable of detecting Common Gateway Interface (CGI) attacks, among other vulnerabilities.
9. **Real-Time Monitoring:** It monitors inbound and outbound network traffic in real-time to identify and respond to potential threats.
10. **User-Friendly and Free:** SNORT is a widely used, cost-free tool, suitable for individuals and organizations for enhancing network security.

Procedure:-

Create a path variable and point it at snort.exe variable name->path and variable value->c:\snort\bin.

Click OK button and then close all dialog boxes.

Open command prompt and type the following commands:

SNORT can be configured to run in three modes:

1. Sniffer mode 2. Packet Logger mode 3. Network Intrusion Detection System mode

Sniffer mode:-

snort -v

Prints out the TCP/IP packets header on the screen

Snort -vd

Shows the TCP/IP ICMP header with application data in transit.

Packet Logger mode:-

snort -dev -l C:\snort\log [create this directory in the C drive] and snort will automatically know to go into packet logger mode, it collects every packet it sees and places it in log directory.

snort -dev -l C:\snort\log -h ipaddress/24

This rule tells snort that you want to print out the data link and TCP/IP headers as well as application data into the log directory.

snort -l C:\snort\log -b

This is binary mode logs everything into a single file.

Conclusion: -

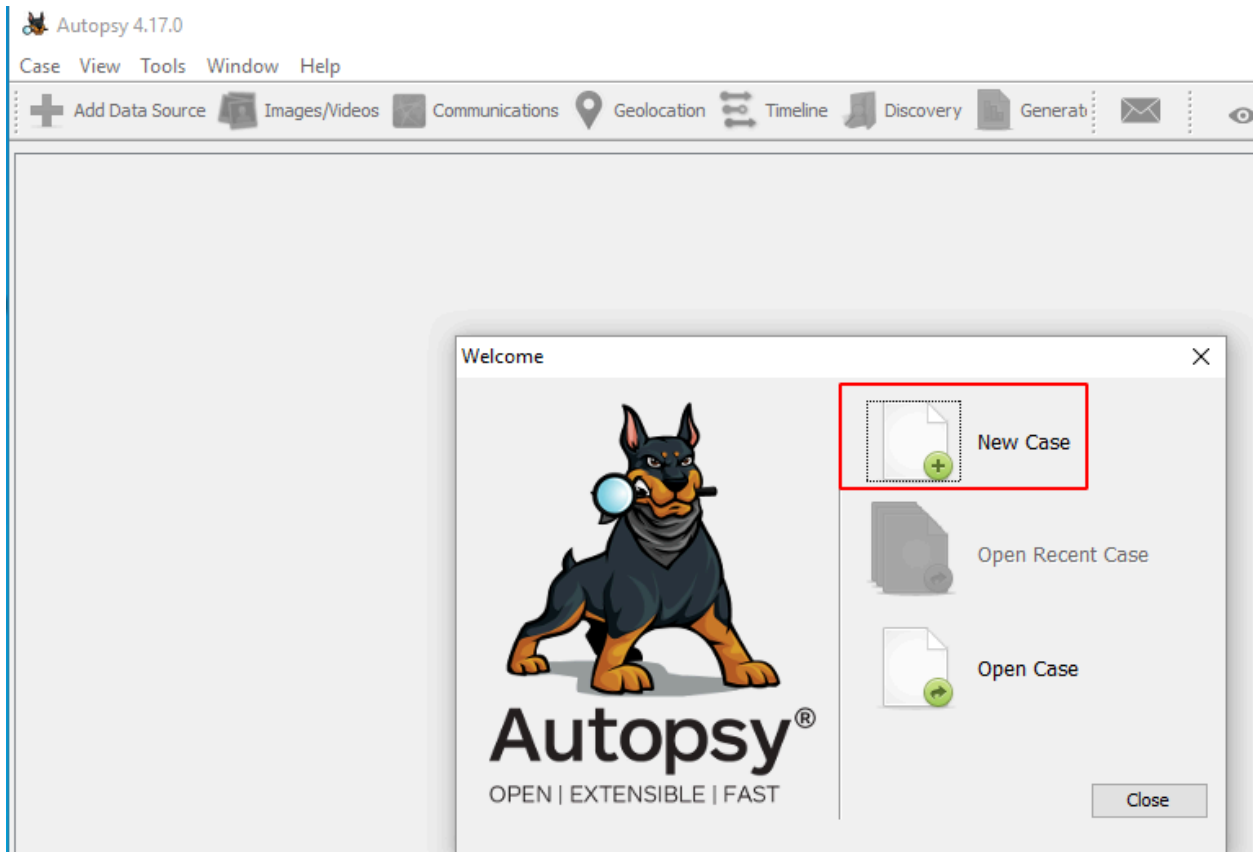
Hence we can create log file to record the packet data using snort.

8.Perform email analysis using Autopsy tool.

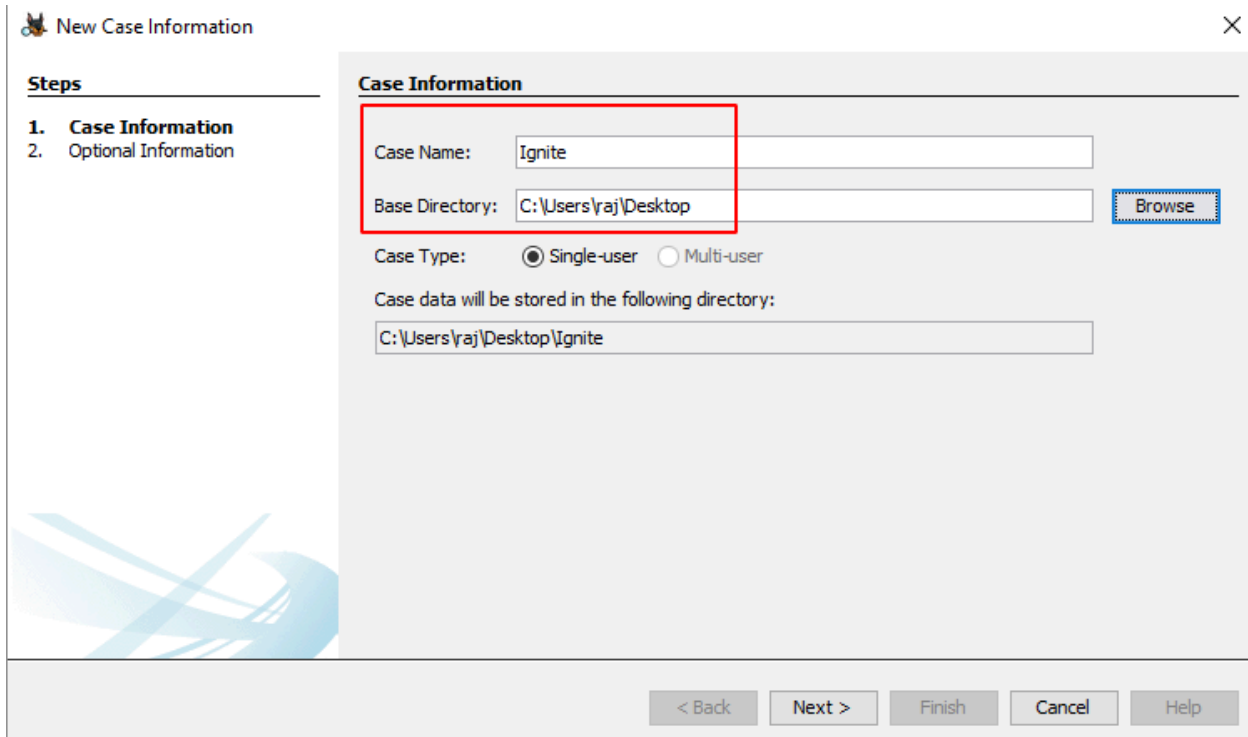
Autopsy is an open-source tool that is used to perform forensic operations on the disk image of the evidence. The forensic investigation that is carried out on the disk image is displayed here. The results obtained here are of help to investigate and locate relevant information. This tool is used by law enforcement agencies, local police and can also be used in the corporates to investigate the evidence found in a computer crime. It can likewise be utilized to recuperate information that has been erased.

Creating a new Case

Run the Autopsy tool on your Windows Operating System and click on “New Case” to create a new case.



Then fill in all the necessary case information like the case name and choose a base directory to save all the case data in one place.



New Case Information

Steps

- 1. Case Information**
- Optional Information

Case Information

Case Name:

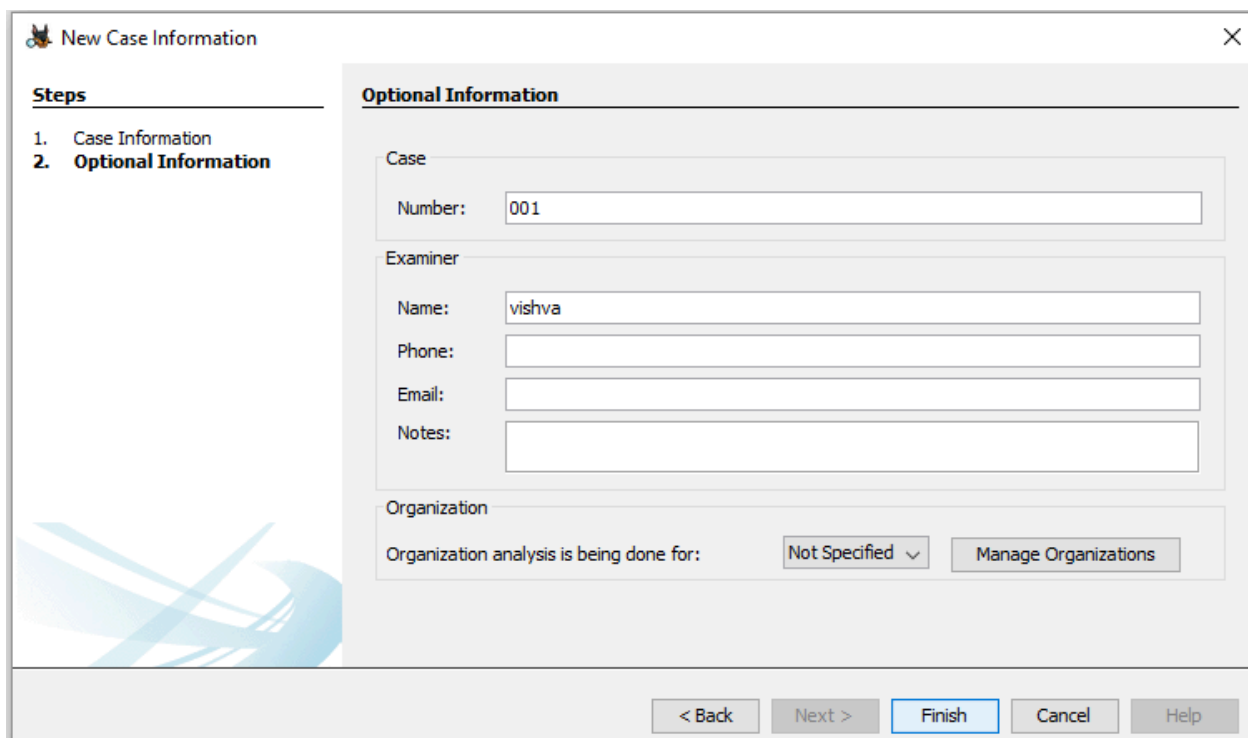
Base Directory:

Case Type: ☒ Single-user ☐ Multi-user

Case data will be stored in the following directory:

< Back Next > Finish Cancel Help

You can also add additional optional information about the case if required.



New Case Information

Steps

- Case Information
- 2. Optional Information**

Optional Information

Case

Number:

Examiner

Name:

Phone:

Email:

Notes:

Organization

Organization analysis is being done for:

< Back Next > **Finish** Cancel Help

Now let us add the type of data source. There are various types to choose from.

Disk Image or VM file: This includes the image file which can be an exact copy of a hard drive, media card, or even a virtual machine.

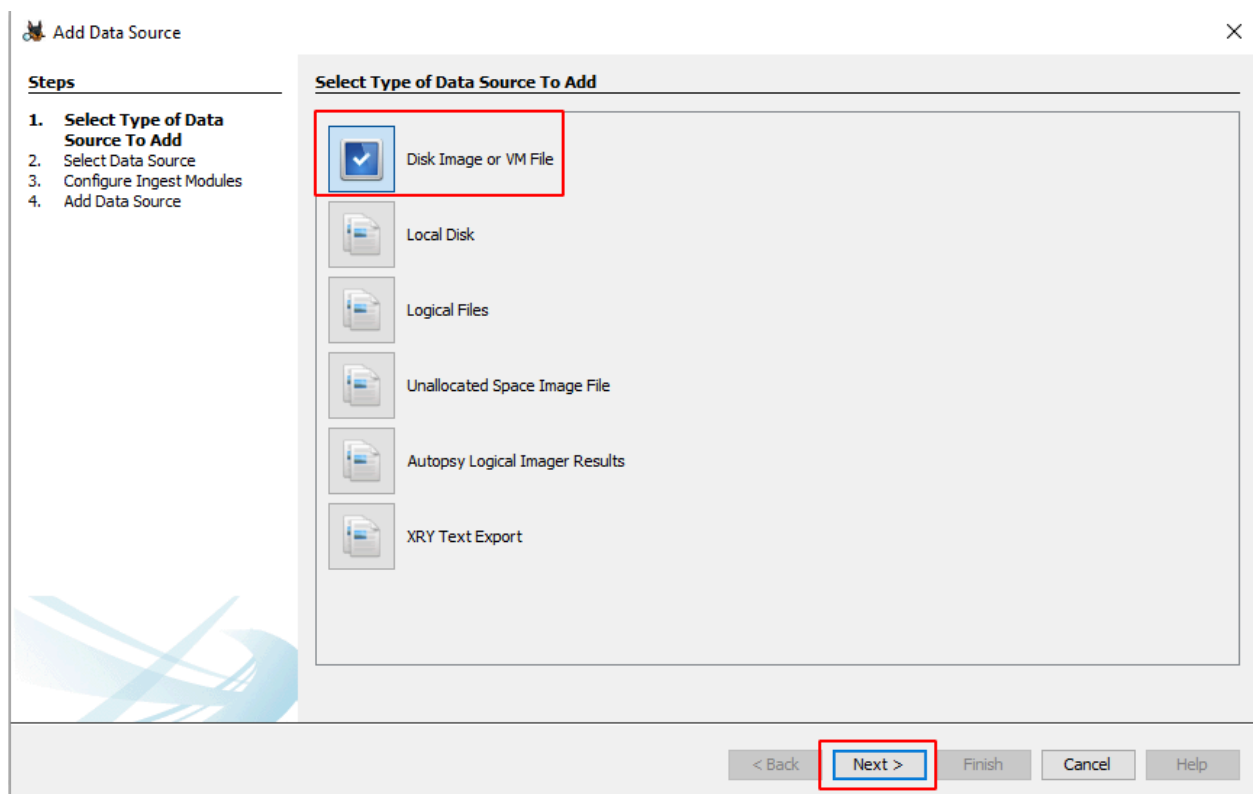
Local Disk: This option includes devices like Hard disk, Pen drives, memory cards, etc.

Logical Files: It includes the image of any local folders or files.


Unallocated Space Image File: They include files that do not contain any file system and run with the help of the ingest module.

Autopsy Logical Imager Results: They include the data source from running the logical imager.

XRY Text Export: This includes the data source from exporting text files from XRY,



Now let us add the data source. Here we have a previously created image file, so we will add the location of that file.

 Add Data Source X

Steps

1. Select Type of Data Source To Add
- 2. Select Data Source**
3. Configure Ingest Modules
4. Add Data Source

Select Data Source

Path: Browse

☐ Ignore orphan files in FAT file systems

Time zone:

Sector size:

Hash Values (optional):

MD5:


SHA-1:

SHA-256:

NOTE: These values will not be validated when the data source is added.

< Back Next > Finish Cancel Help

Next, you will be prompted to **Configure the Ingest Module**.

 Add Data Source X

Steps

1. Select Type of Data Source To Add
2. Select Data Source
- 3. Configure Ingest Modules**
4. Add Data Source

Configure Ingest Modules

Run ingest modules on:

☒ Recent Activity

☒ Hash Lookup

☒ File Type Identification

☒ Extension Mismatch Detector

☒ Embedded File Extractor

☒ Picture Analyzer

☒ Keyword Search

☒ Email Parser

☒ Encryption Detection

☒ Interesting Files Identifier

☒ Central Repository

☒ PhotoRec Carver

☒ Virtual Machine Extractor

☒ Data Source Integrity

Select All Deselect All History

The selected module has no per-run settings.

Extracts recent user activity, such as Web browsing, recently us...

Global Settings

< Back Next > Finish Cancel Help

The contents of the Ingest module are listed below:

INGEST MODULE	
Recent Activity	It is used to discover the recent operations that were performed on the disk, like the files that were viewed recently.
Extension Mismatch Detector	It is used to identify files whose extensions were tampered with or had been changed to hide the evidence.
Hash Lookup	It is used to identify a particular file using its hash value.
File Type Identification	This is used to identify files based on their internal file signatures than just the file extensions.
Embedded File Extractor	It is used to extract embedded files like .zip, .rar, etc. and use those files for analysis.
Keyword Search	This is used to search for any particular keyword or a pattern in the image file.
Email Parser	This is used to extract information from email files if the disk holds any email database information.
Encryption Detection	This helps to detect and identifies encrypted password-protected files.
Interesting File Identifier	Using this feature the examiner is notified when results pertaining to the set of rules that are defined to identify a particular type of file.
PhotoRec Carver	This helps the examiner to recover files, photos, etc. from the unallocated space on the image disk.
Virtual Machine Extractor	It helps to extract and analyze if any Virtual machine is found on the disk image.
Data Source Integrity	It helps to calculate the hash value and store them in the database.

Data Source information displays basic metadata. Its detailed analysis is displayed at the bottom. It can be extracted one after the other.

Views

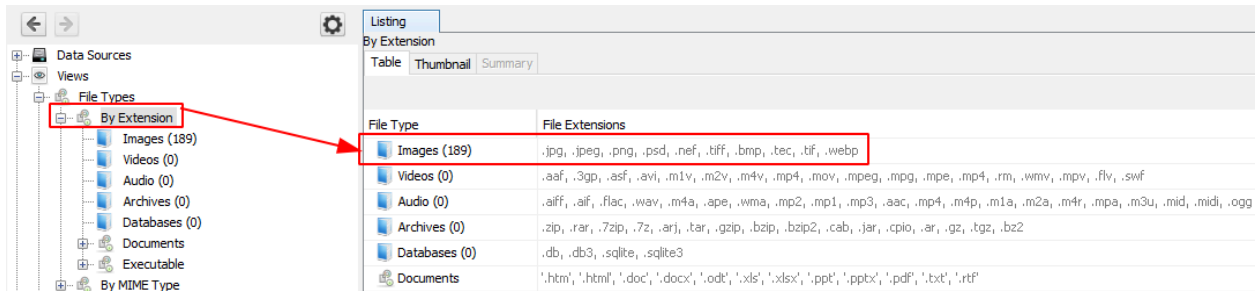
File Type: It can be classified in the form of File extension or MIME type.

It provides information on file extensions that are commonly used by the OS whereas MIME types are used by the browser to decide what data to represent. It also displays deleted files.

Note: These file types can be categorized depending on Extension, Documents, Executables.

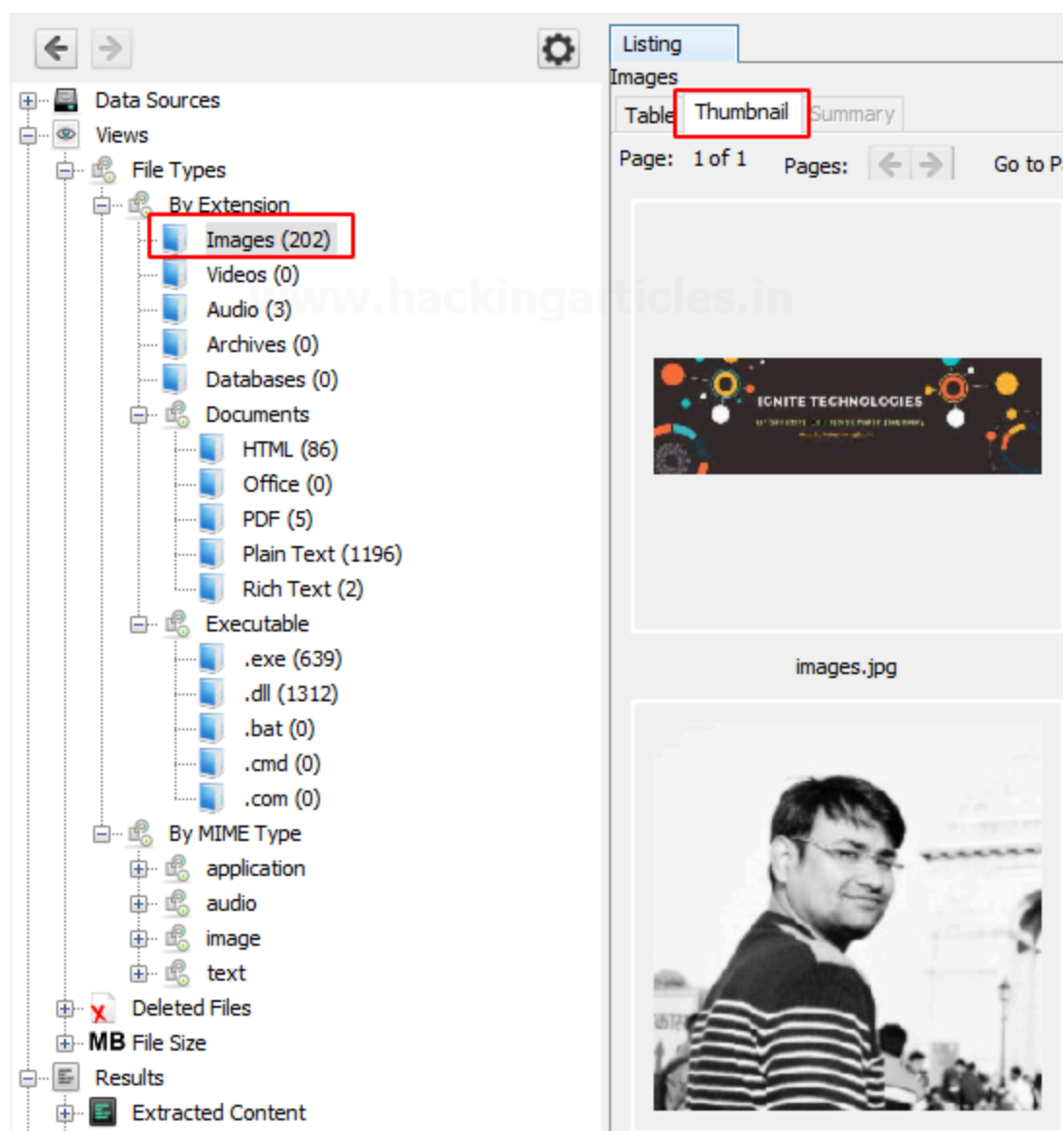
By Extension

In the category Filetypes by extension and you can see that this has been sub-divided into file types like images, video, audio, archives, databases, etc.



Let us click on images and explore the images that have been recovered.


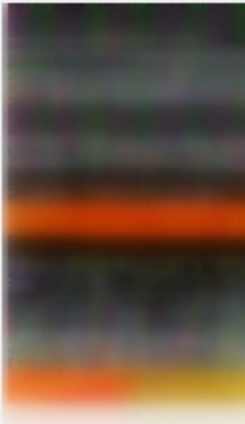
We can also view the thumbnail of the images.



On viewing the thumbnail, you can view the file metadata and details about the image.

TableThumbnailSummary

Page: 1 of 1
Pages:
Go to Page:
Image

images.jpg

/img_Ignite.E01/images.jpg

HexTextApplicationFile MetadataContextResultsAnnotationsOther

From The Sleuth Kit istat Tool:

MFT Entry Header Values:

Entry: 49 Sequence: 1

\$LogFile Sequence Number: 16885331

Allocated File

Links: 1

\$STANDARD_INFORMATION Attribute Values:

Flags: Archive

Owner ID: 0

Security ID: 271 (S-1-5-21-1276730070-1850728493-30201

Created: 2020-11-26 08:20:24.482672700 (PST)

File Modified: 2020-11-26 08:20:24.667704200 (PST)

MFT Modified: 2020-11-26 09:00:35.829441300 (PST)

Accessed: 2020-11-26 08:59:53.860554000 (PST)

\$FILE_NAME Attribute Values:

Flags: Archive

Name: \$R3RSEBH.jpg

Parent MFT Entry: 40 Sequence: 1

Allocated Size: 8192 Actual Size: 7641

Created: 2020-11-26 08:20:24.482672700 (PST)

File Modified: 2020-11-26 08:20:24.667704200 (PST)

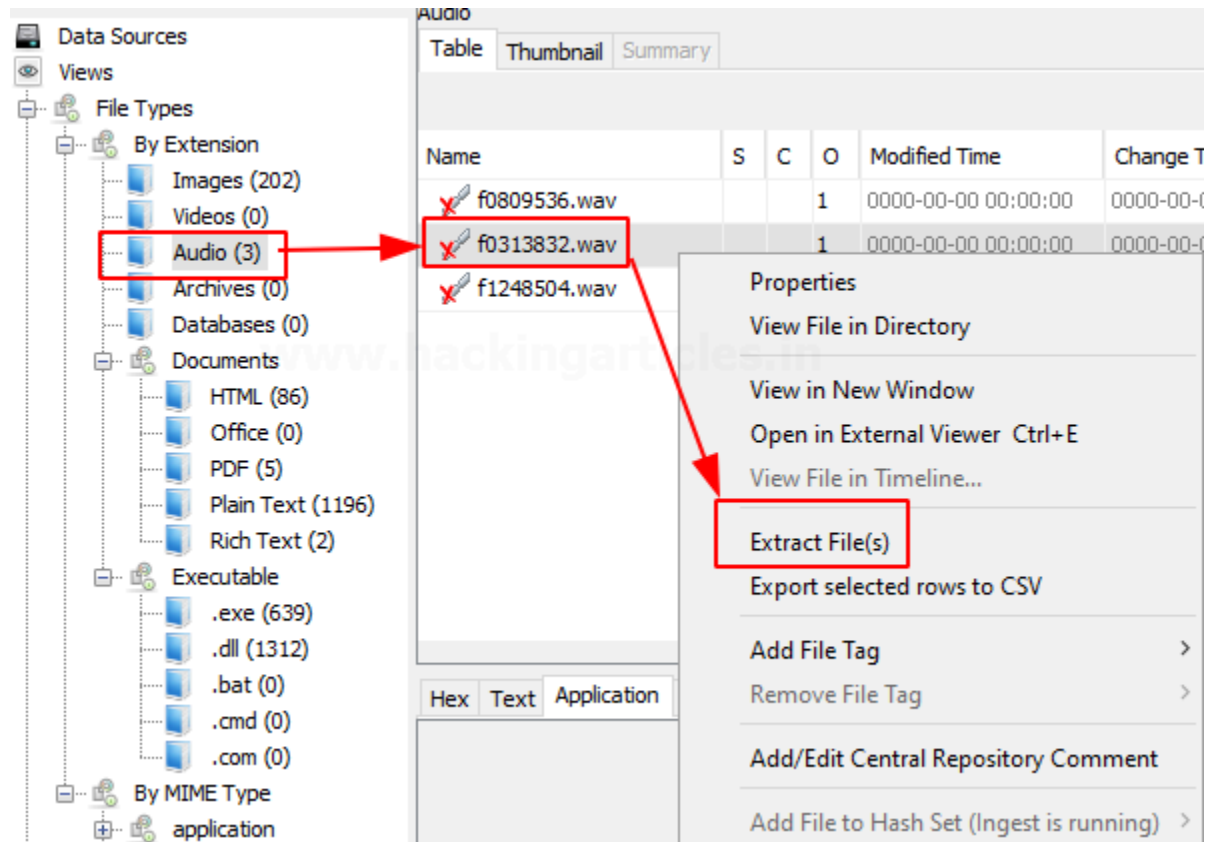
MFT Modified: 2020-11-26 08:59:01.714957400 (PST)

Accessed: 2020-11-26 08:59:01.704974100 (PST)

\$OBJECT_ID Attribute Values:

Object Id: 3fd39b21-2f45-11eb-ala0-001b10002aec

Here we can also view a few audio files that have been recovered. We can extract these files from the system and hear to them using various software.



Documents

The documents are categorized into 5 types: HTML, office, PDF, Plain Text, Rich Text.

On exploring the documents option, you can see all the HTML documents present, you can click on the important ones to view them.

The screenshot shows a file explorer on the left and a web browser interface on the right. The file explorer displays a tree view with categories like 'By Extension' and 'By MIME Type'. The 'HTML (86)' folder is highlighted. The web browser interface shows a list of HTML files, with 'Forensic Investigation Autopsy Forensic Browser in Linux.html' selected. Below the list, the 'Indexed Text' tab is active, displaying the content of the selected file.

Name	S	C	O
Forensic Investigation Autopsy Forensic Browser in Linux.html			1
a.html			1
a_002.html			1
fastbutton.html			1
like.html			1

The 'Indexed Text' tab displays the following content:

```

Hacking Articles

Raj Chandel's Blog

* CTF Challenges
* Penetration Testing
* Web Penetration Testing
* Red Teaming
* Donate us
* Courses We Offer
  o Bug Bounty
  o Computer Forensics
  o Ethical Hacking
  o Red Teaming

Forensic Investigation: Autopsy Forensic Browser in Linux

posted inCyber Forensics on August 13, 2020 by Raj Chandel
SHARE
Save






```

On exploring the PDF option, you can also find the important PDF in the disk image.

Listing

PDF

Table Thumbnail Summary

Name	S	C	O	Modified Time	Chan
 \$IO2Y1Z5.pdf			1	2020-11-26 09:04:18 PST	2020-
 \$RO2Y1Z5.pdf			1	2020-02-29 11:02:57 PST	2020-
 Android Pentesting.pdf			1	2020-10-23 01:42:19 PDT	2020-
 Bug Bounty Course Details.pdf				2020-11-26 09:04:18 PST	2020-
 f0184904.pdf			1	0000-00-00 00:00:00	0000-

Hex Text Application File Metadata Context Results Annotations Other Occurrences

1 of 5 29%

POWERED BY IGNITE TECHNOLOGIES

ANDROID PENTESTING

Similarly, the various Plain text files can also be viewed. You can also recover deleted plain text files.

Version

Images (202)

Media (0)

Audio (3)

Archives (0)

Databases (0)

Documents

HTML (86)

Office (0)

PDF (5)

Plain Text (1196)

Rich Text (2)

Executable

.exe (639)

.dll (1312)

.bat (0)

.cmd (0)

.com (0)

IME Type

Application

Audio

Image

Text

Files

Content

data (6)

de Bin (4)

Downloads (3)

Hits

Hits

Messages

g Items

Name	S	C	O	Modified Time
\$IK1MRRO.txt			1	2020-11-26 08:56:
\$RK1MRRO.txt			1	2020-11-26 08:55:
USB.txt			1	2020-09-09 07:15:
Ignite.E01.txt				2020-11-26 08:56:
f0484218.txt			1	0000-00-00 00:00:

Hex Text Application File Metadata Context Results And

Strings Indexed Text Translation

Page: 1 of 1 Page < > Matches on page: - of - Mat

ONOTICE: The imaging operation was cancelled!

Created By AccessData® FTK® Imager 4.3.1.1

Case Information:

Acquired using: ADI4.3.1.1

Case Number: 001

Evidence Number: AU001

Unique description: Hacking Articles

Examiner: Vishva

Notes:

Information for E:\Ignite:

Physical Evidentiary Item (Source) Information

[Device Info]

Source Type: Logical

[Drive Geometry]

Bytes per Sector: 512

Sector Count: 125,821,080

[Physical Drive Information]

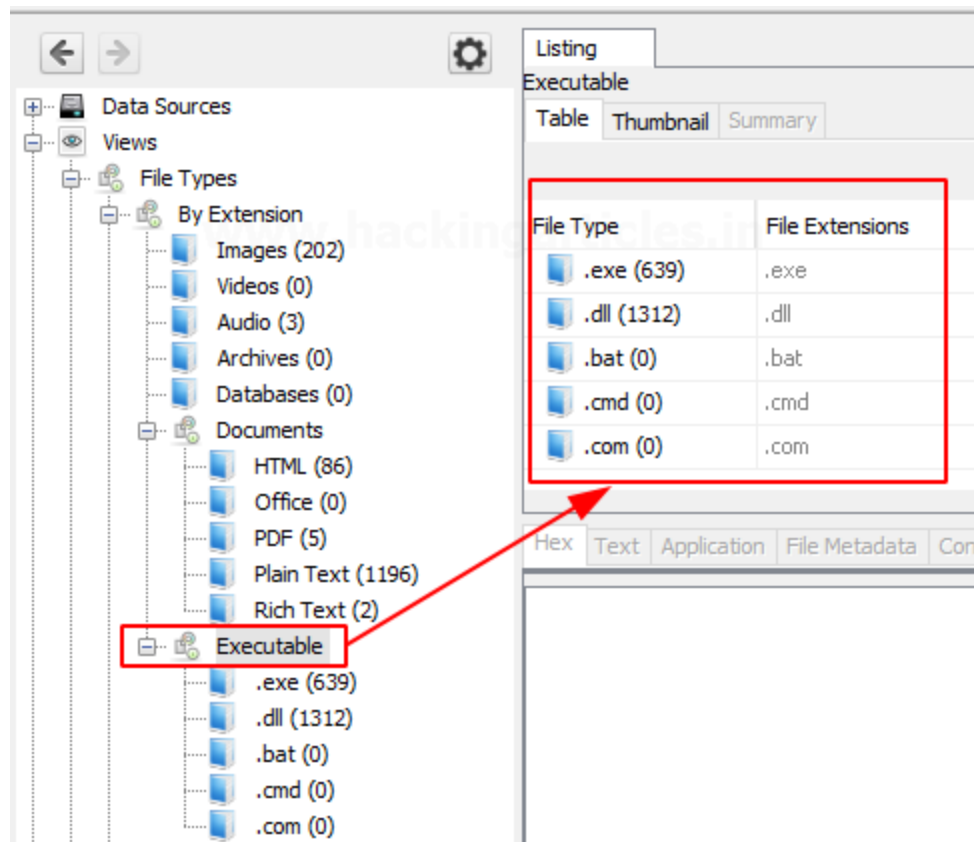
Removable drive: False

Source data size: 61436 MB

Sector count: 125821080

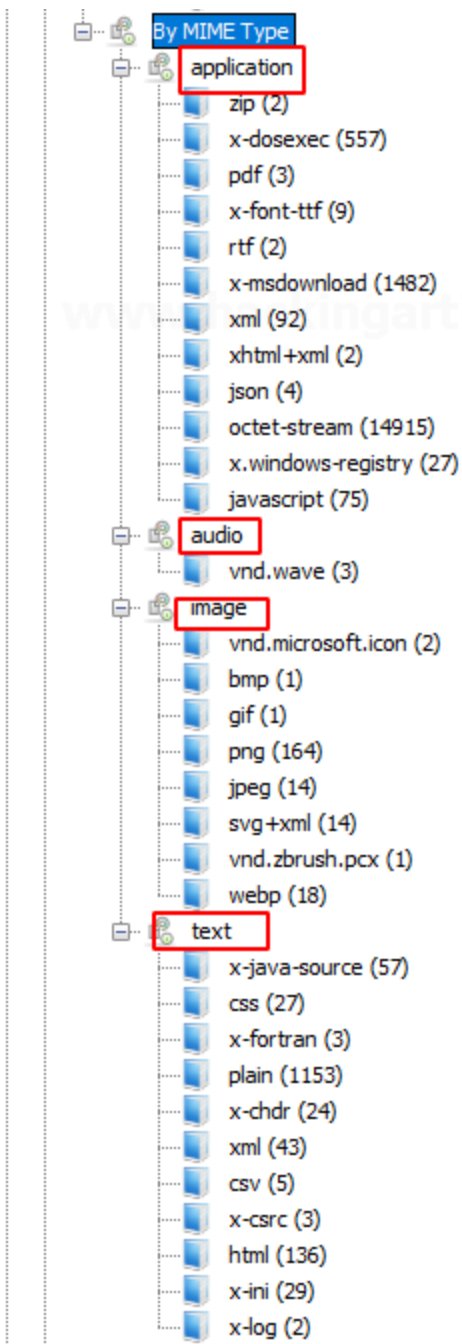
Executables

These file types are then sub-divided into .exe, .dll, .bat, .cmd and .com.



By MIME Type

In this type of category, there are four sub-categories like application, audio, image, and text. They are divided further into more sections and file types.



Deleted Files: It displays information about the deleted file which can be then recovered.

File System

Name	S	C	O	Modified Time
20201014.mem			0	2020-10-13 13:39:50 PDT
adencrypt.dll			0	2020-05-11 21:03:46 PDT
adencrypt_gui.exe			0	2020-05-11 21:03:46 PDT
adfbs_globals.dll			0	2020-05-11 21:03:46 PDT
adfs_globals.dll			0	2020-05-11 21:03:46 PDT
ADG_EULA.rtf		1		2020-02-05 15:48:36 PST
ADIso.exe			0	2020-05-11 21:03:46 PDT
ADIsoDLL.dll			0	2020-05-11 21:03:48 PDT
adshattrdefs.dll			0	2020-05-11 21:03:48 PDT
adtz_globals.dll			0	2020-05-11 21:03:48 PDT
ad_globals.dll			0	2020-05-11 21:03:46 PDT
ad_log.dll			0	2020-05-11 21:03:46 PDT
boost_chrono-vc140-mt-1_59.dll			0	2020-05-11 21:03:48 PDT
boost_date_time-vc140-mt-1_59.dll			0	2020-05-11 21:03:46 PDT
boost_filesystem-vc140-mt-1_59.dll			0	2020-05-11 21:03:50 PDT
boost_regex-vc140-mt-1_59.dll			0	2020-05-11 21:03:50 PDT
boost_system-vc140-mt-1_59.dll			0	2020-05-11 21:03:50 PDT
boost_thread-vc140-mt-1_59.dll			0	2020-05-11 21:03:50 PDT
FTK Imager.exe			0	2020-05-11 21:04:10 PDT

MB Size Files: In this, the files are categorized based on their size starting from 50MB. This allows the examiner to look for large files.

MB File Size

Size Range	MB 50 - 200MB (1)	MB 200MB - 1GB (2)	MB 1GB+ (3)
MB 50 - 200MB (1)			
MB 200MB - 1GB (2)			
MB 1GB+ (3)			

Results

Perform email analysis using Autopsy Tool.

* Once the new case is created and the data source as selected in the Autopsy tool

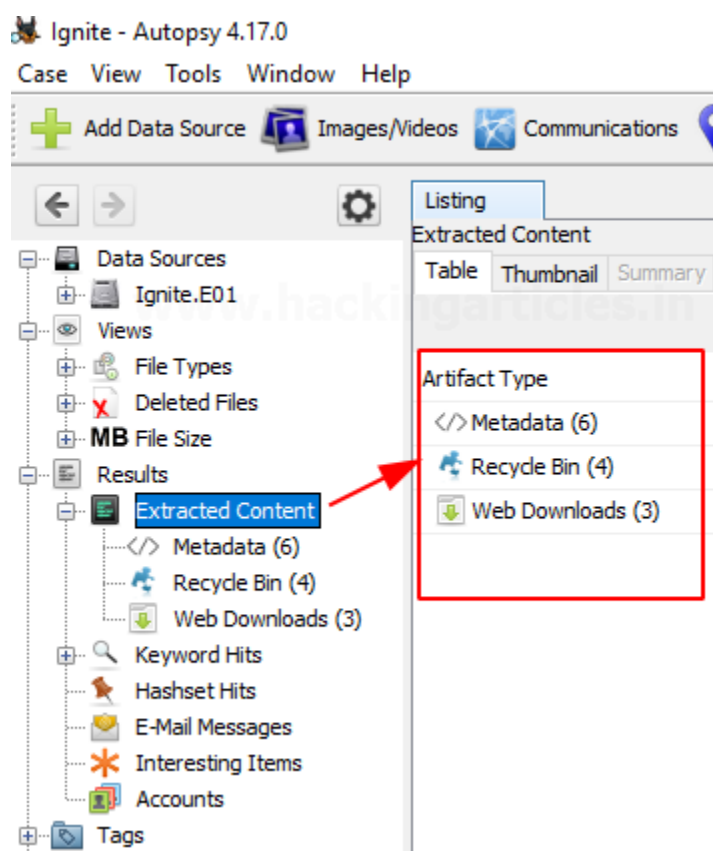
* Under Analysis Results we select keyword hits. Then we select Email Addresses from the given options.

Now we can view the available email addresses. Then we can choose to export into a CSV format for email analysis.

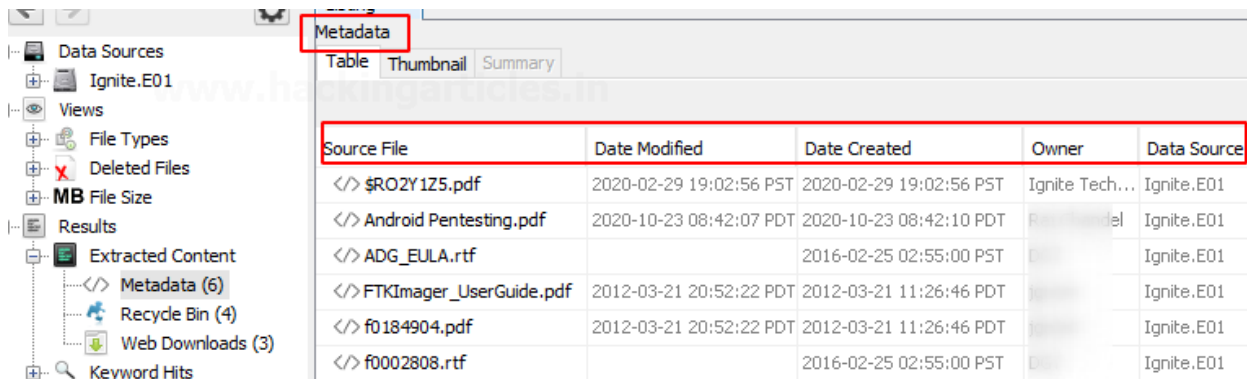
* For the email analysis, we are searching for hamilton;hamilton @acme.com. On selecting we get the deleted items.dbx and Inbox.dbx for our email analysis..

In this section, we get information about the content that was extracted.

Extracted Content: All the content that was extracted, is segregated further in detail. Here we have found metadata, Recycle Bin, and web downloads. Let us further view each one of them.



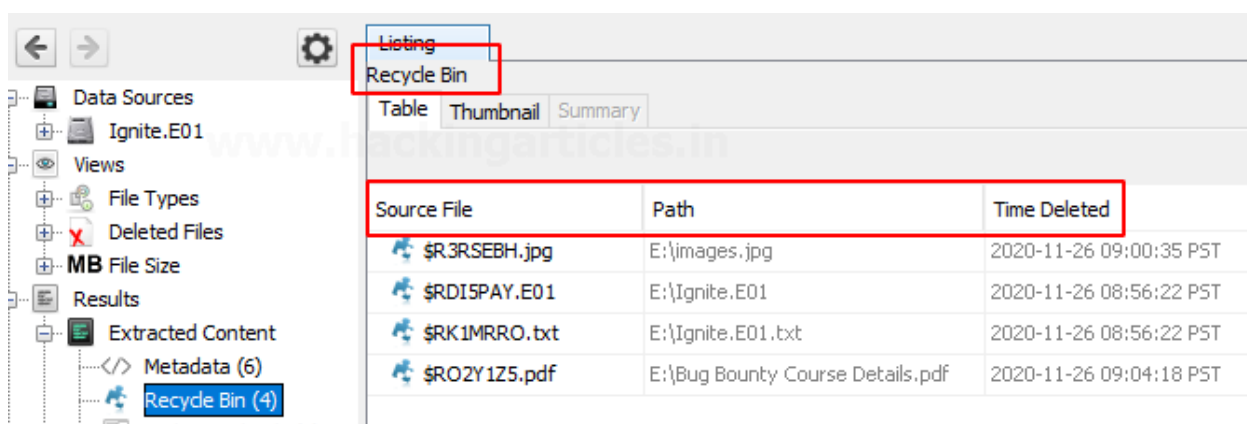
Metadata: Here we can view all the information about the files like the date it was created, to was modified, file's owner, etc.



The screenshot shows the 'Metadata' view of the Ignite.E01 file. The left sidebar lists various categories like Data Sources, Views, File Types, Deleted Files, MB File Size, Results, Extracted Content, Recycle Bin, Web Downloads, and Keyword Hits. The 'Metadata' category is selected, and a table of extracted files is displayed. The table has columns for Source File, Date Modified, Date Created, Owner, and Data Source. The files listed include \$ROZY1Z5.pdf, Android Pentesting.pdf, ADG_EULA.rtf, FTKImager_UserGuide.pdf, f0184904.pdf, and f0002808.rtf.

Source File	Date Modified	Date Created	Owner	Data Source
</> \$ROZY1Z5.pdf	2020-02-29 19:02:56 PST	2020-02-29 19:02:56 PST	Ignite Tech...	Ignite.E01
</> Android Pentesting.pdf	2020-10-23 08:42:07 PDT	2020-10-23 08:42:10 PDT	...	Ignite.E01
</> ADG_EULA.rtf		2016-02-25 02:55:00 PST	...	Ignite.E01
</> FTKImager_UserGuide.pdf	2012-03-21 20:52:22 PDT	2012-03-21 11:26:46 PDT	...	Ignite.E01
</> f0184904.pdf	2012-03-21 20:52:22 PDT	2012-03-21 11:26:46 PDT	...	Ignite.E01
</> f0002808.rtf		2016-02-25 02:55:00 PST	...	Ignite.E01

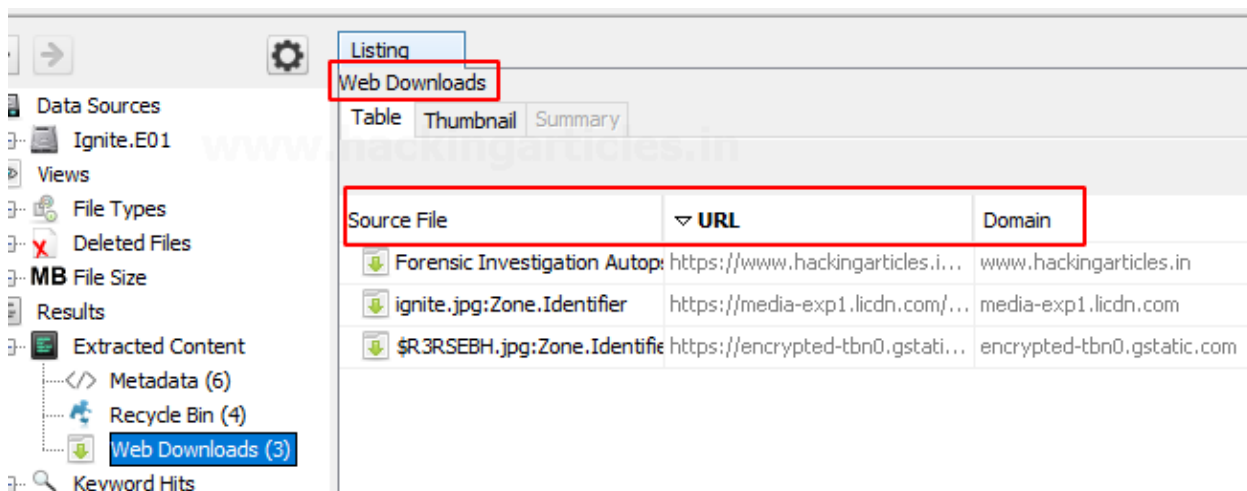
Recycle Bin: The files that were put in the recycle bin are found in this category.



The screenshot shows the 'Recycle Bin' view of the Ignite.E01 file. The left sidebar is the same as the previous screenshot. The 'Recycle Bin' category is selected, and a table of deleted files is displayed. The table has columns for Source File, Path, and Time Deleted. The files listed include \$R3RSEBH.jpg, \$RDISPAY.E01, \$RK1MRRO.txt, and \$ROZY1Z5.pdf.

Source File	Path	Time Deleted
\$R3RSEBH.jpg	E:\images.jpg	2020-11-26 09:00:35 PST
\$RDISPAY.E01	E:\Ignite.E01	2020-11-26 08:56:22 PST
\$RK1MRRO.txt	E:\Ignite.E01.txt	2020-11-26 08:56:22 PST
\$ROZY1Z5.pdf	E:\Bug Bounty Course Details.pdf	2020-11-26 09:04:18 PST

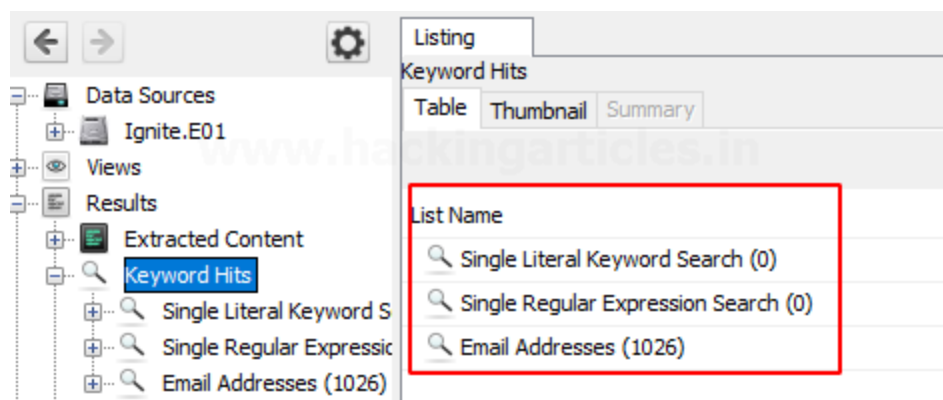
Web Downloads: Here you can see the files that were downloaded from the internet.



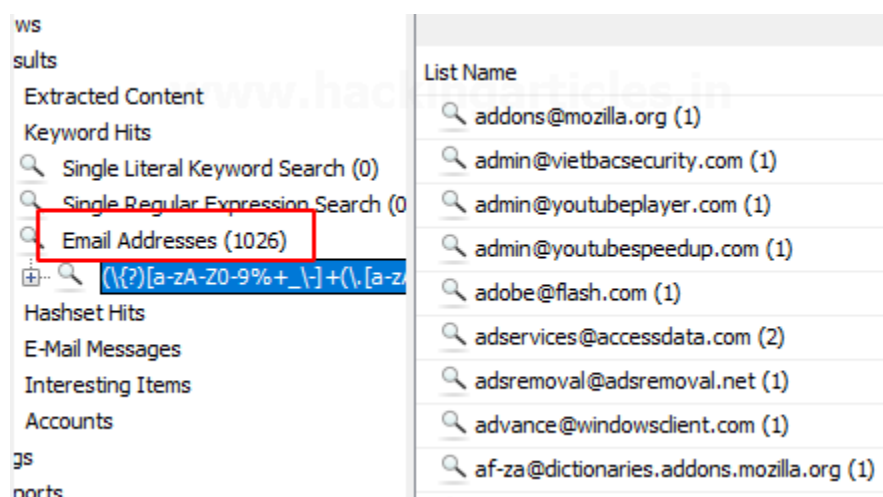
The screenshot shows the 'Web Downloads' view of the Ignite.E01 file. The left sidebar is the same as the previous screenshots. The 'Web Downloads' category is selected, and a table of downloaded files is displayed. The table has columns for Source File, URL, and Domain. The files listed include Forensic Investigation Autop, ignite.jpg:Zone.Identifier, and \$R3RSEBH.jpg:Zone.Identifier.

Source File	URL	Domain
Forensic Investigation Autop	https://www.hackingarticles.i...	www.hackingarticles.in
ignite.jpg:Zone.Identifier	https://media-exp1.licdn.com/...	media-exp1.licdn.com
\$R3RSEBH.jpg:Zone.Identifier	https://encrypted-tbn0.gstatic...	encrypted-tbn0.gstatic.com

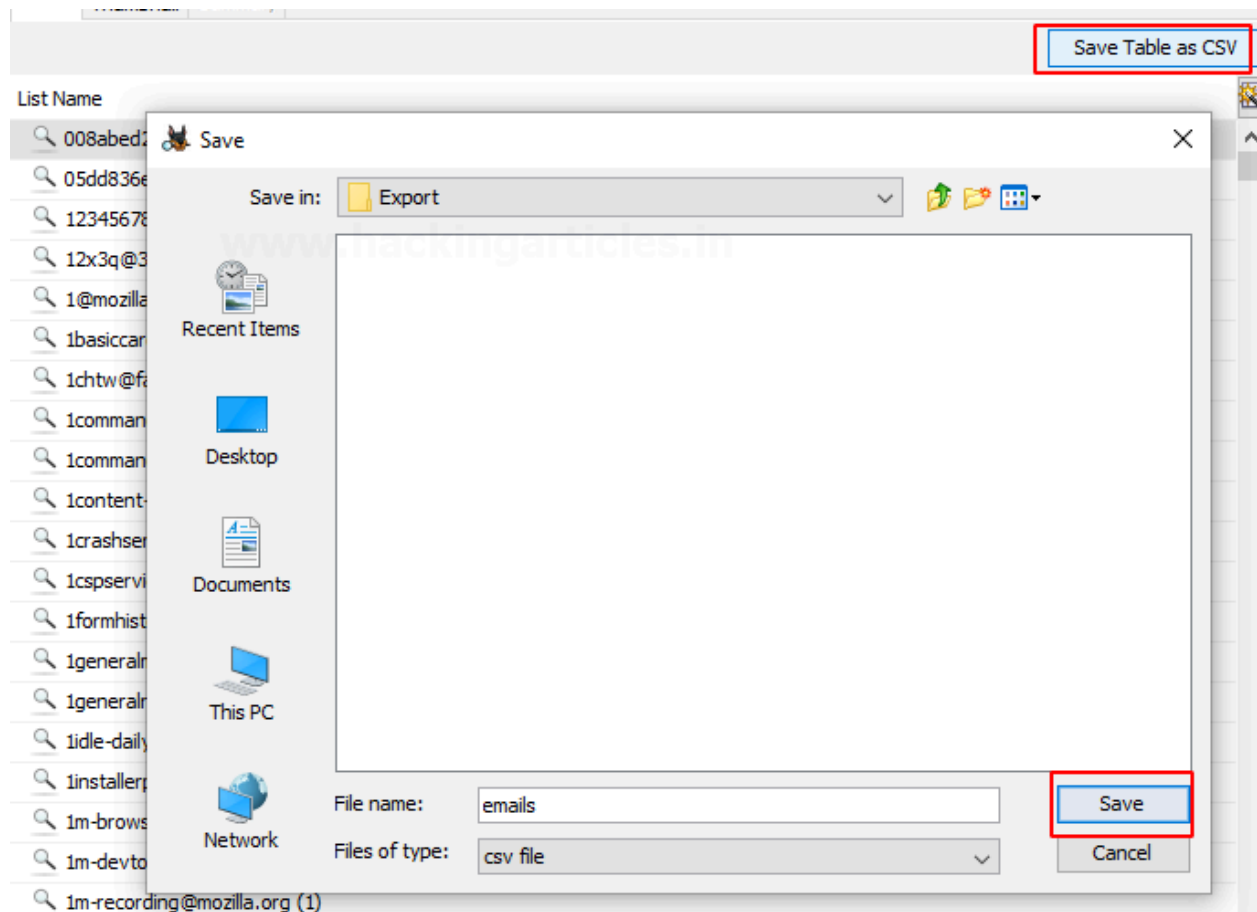
Keyword Hits: In this, any specific keywords can be looked up for in the disk image. The search can be conducted concerning the Exact match, Substring matches, Emails, Literal words, Regular expressions, etc.



You can view the available email addresses.

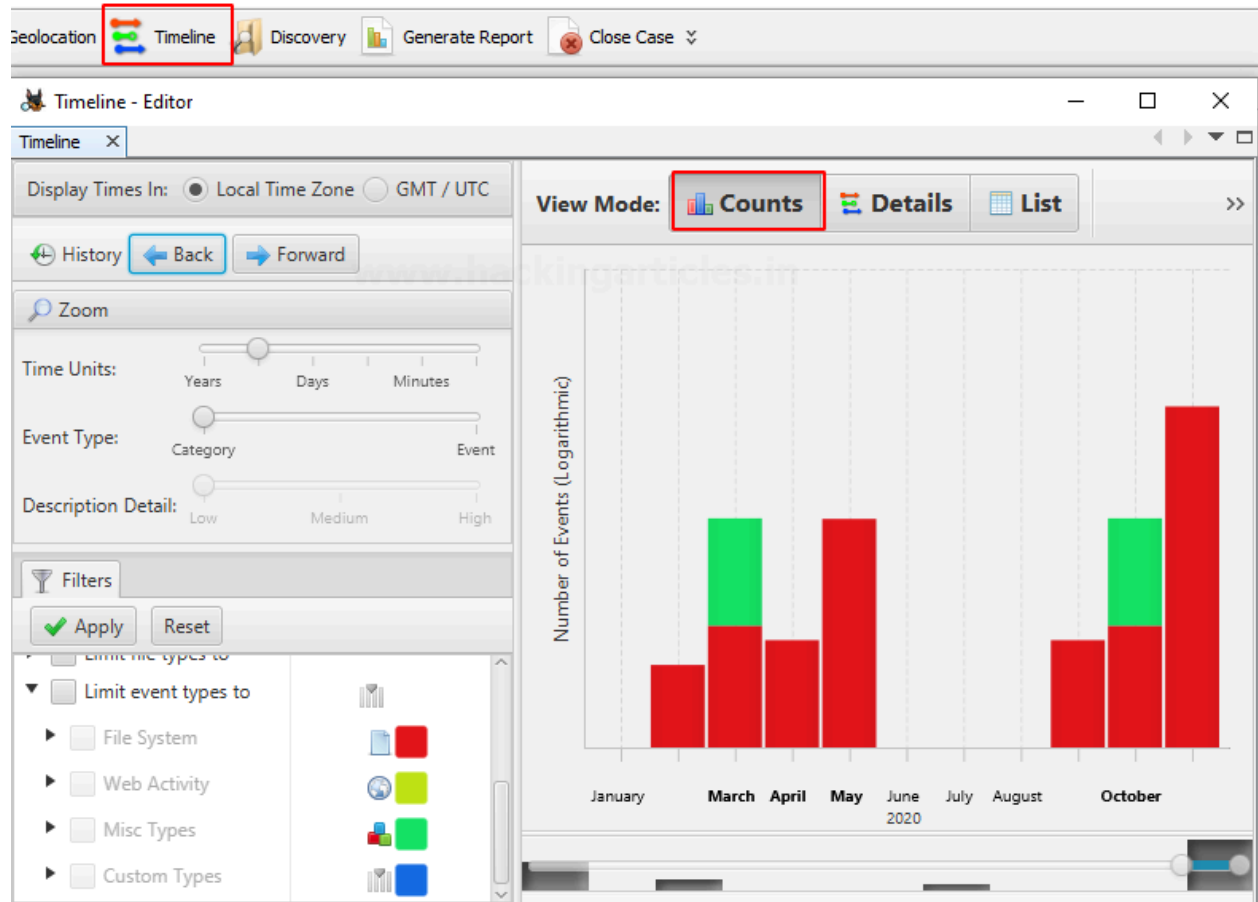


You can choose to export into a CSV format.



Timeline

By using this feature you can get information on the usage of the system in a statistical, detailed, or list form.



Display Times In: ☒ Local Time Zone ☐ GMT / UTC

History ← Back → Forward

Zoom

Time Units: Years Days Minutes

Event Type: Category Event

Description Detail: Low Medium High

Filters

☒ Apply ☐ Reset

Limit event types to

- ☐ File System
- ☐ Web Activity
- ☐ Misc Types
- ☐ Custom Types

Hidden Descriptions

View Mode: Counts **Details** List >>

All Events (Filtered)

.../S-1-5-21-1276730070-1850728493-30201559-1001/\$RO21

.../S-1-5-21-1276730070-1850728493-30201559-1001/\$I

Document Last Saved (2) 46

Document Created (2)

/Bug Bounty Course Details.pdf (4)

... 3

/USB.txt (4)

2

2

/20201014.mem (4)

January April June August November 2020

Start: Jan 27, 2020 11:33:00 PM 🕒 🔍 🔍 >>

Timeline - Editor

Timeline X

Display Times In: ☒ Local Time Zone ☐ GMT / UTC

History ← Back → Forward

Filters

☒ Apply

☐ Must include text:

☐ Must be tagged

☐ Must have hash hit

▶ ☐ Limit data sources to

▶ ☐ Limit file types to

▼ ☐ Limit event types to

▶ ☐ File System

▶ ☐ Web Activity

▶ ☐ Misc Types

▶ ☐ Custom Types

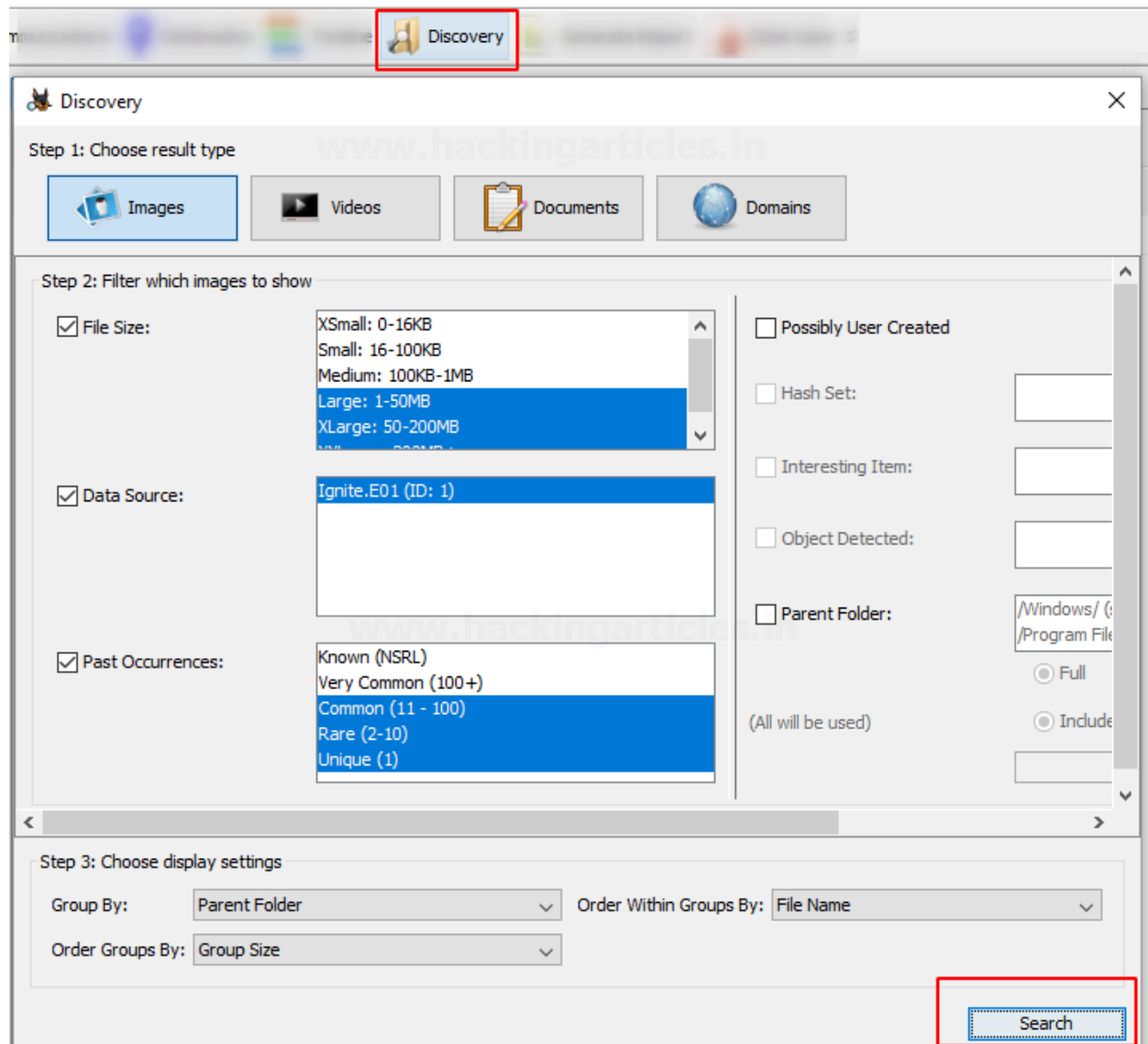
View Mode: Counts Details **List**

Date/Time	Event Type	Description	Tags
2020-02-06 05:18:36	M__	/\$Orpha ... LA.rtf	
2020-03-01 00:32:57	M__	/\$RECY ... Z5.pdf	
2020-03-01 08:32:56	Document L...	Documen ... d : :	
2020-03-01 08:32:56	Document ...	Documen ... d : :	
2020-03-10 09:42:02	M__	/\$Orpha ... gpl.txt	
2020-03-10 09:48:50	M__	/\$Orpha ... gpl.txt	
2020-04-10 21:12:08	__B_	/\$RECY ... Z5.pdf	
2020-04-10 21:12:08	__B_	/Bug Bo ... ils.pdf	
2020-05-12 01:06:40	M__	/\$Orpha ... ter.dll	
2020-05-12 09:33:46	M__	/\$Orpha ... ui.exe	
2020-05-12 09:33:46	M__	/\$Orpha ... _59.dll	
2020-05-12 09:33:46	M__	/\$Orpha ... als.dll	
2020-05-12 09:33:46	M__	/\$Orpha ... log.dll	

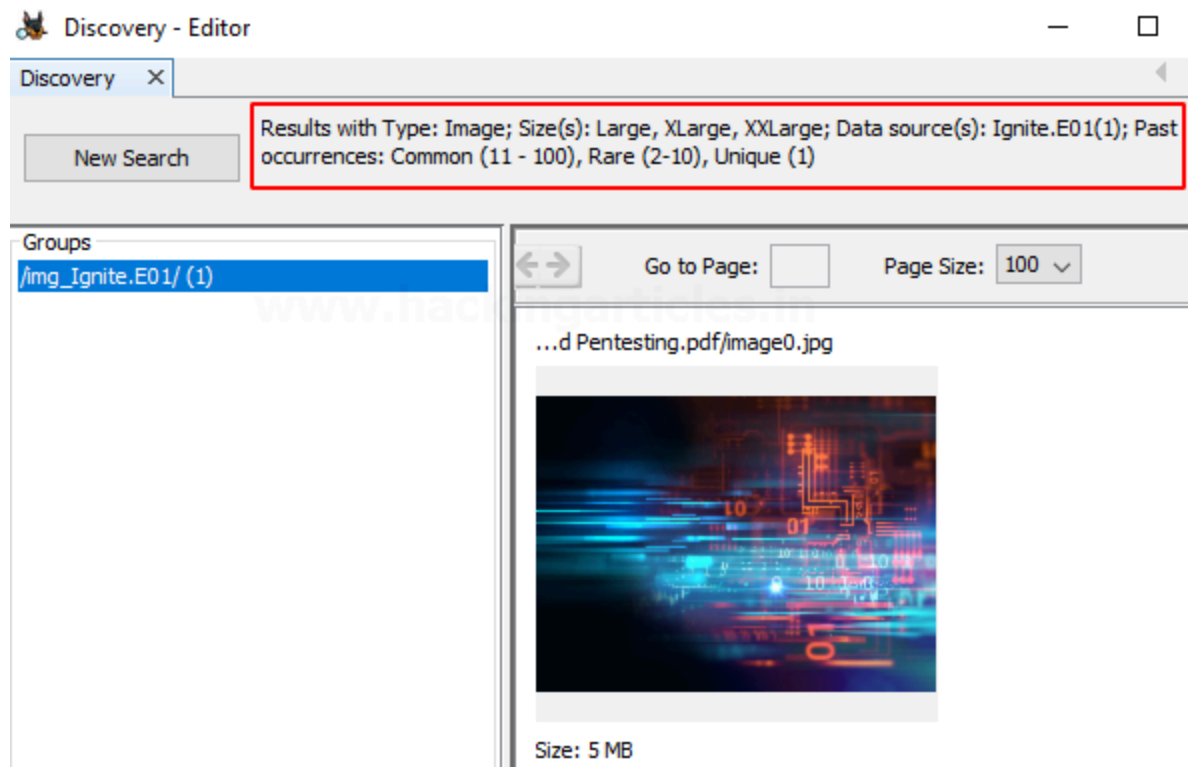
Start: Jan 27, 2020 11:33:00 PM

Discovery

This option allows finding media using different filters that are present on the disk image.

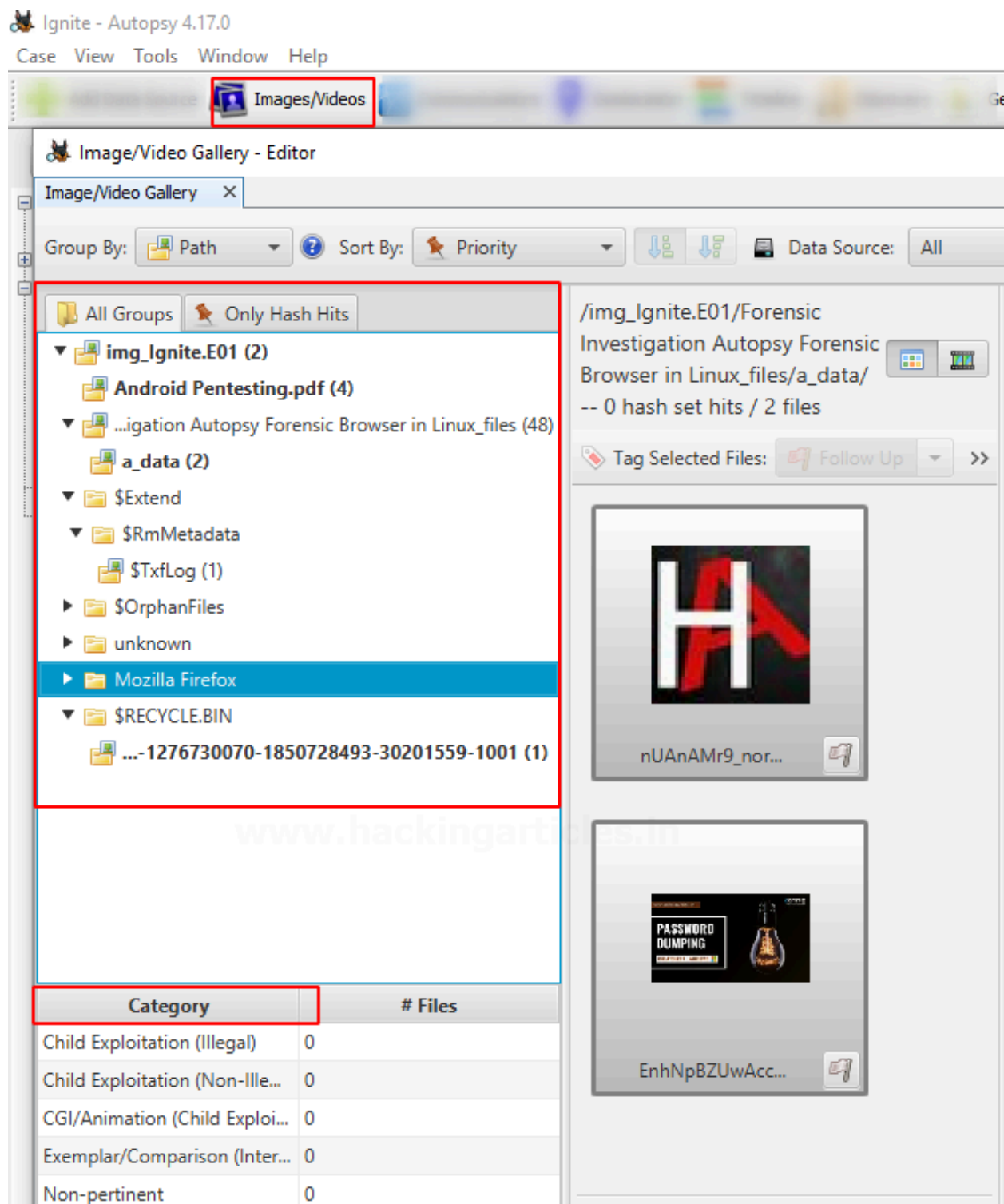


According to the selected options, you can get the desired results.



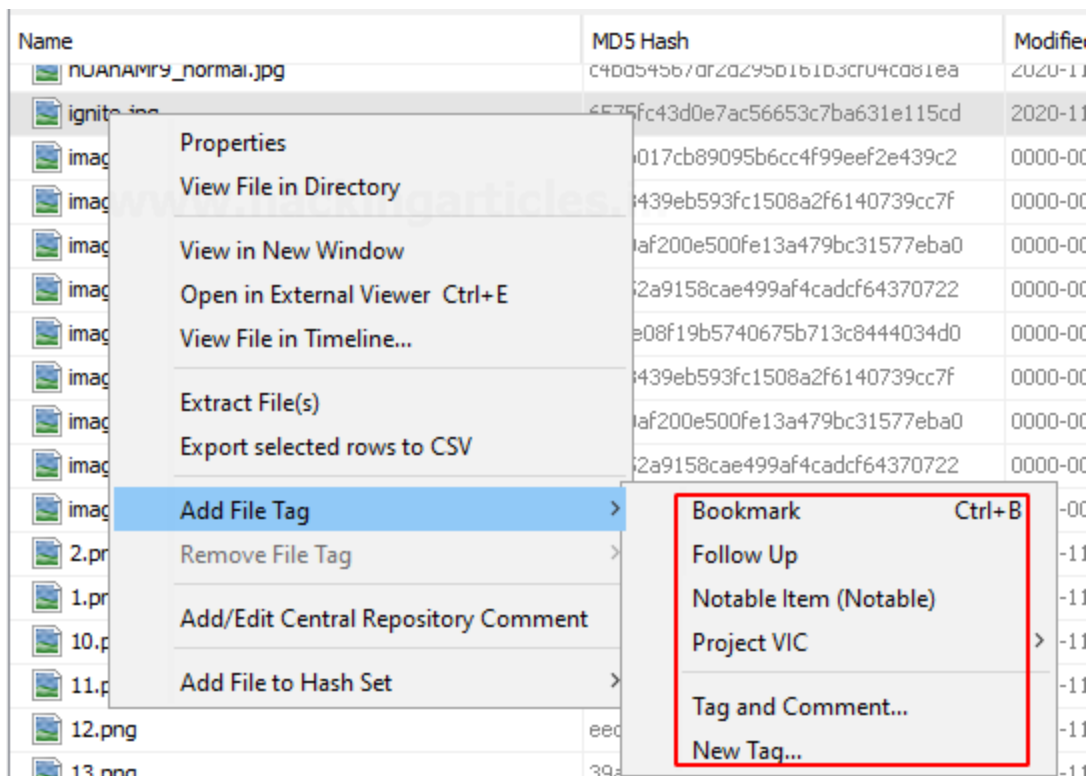
Images/Videos

This option is to find images and videos through various options and multiple categories

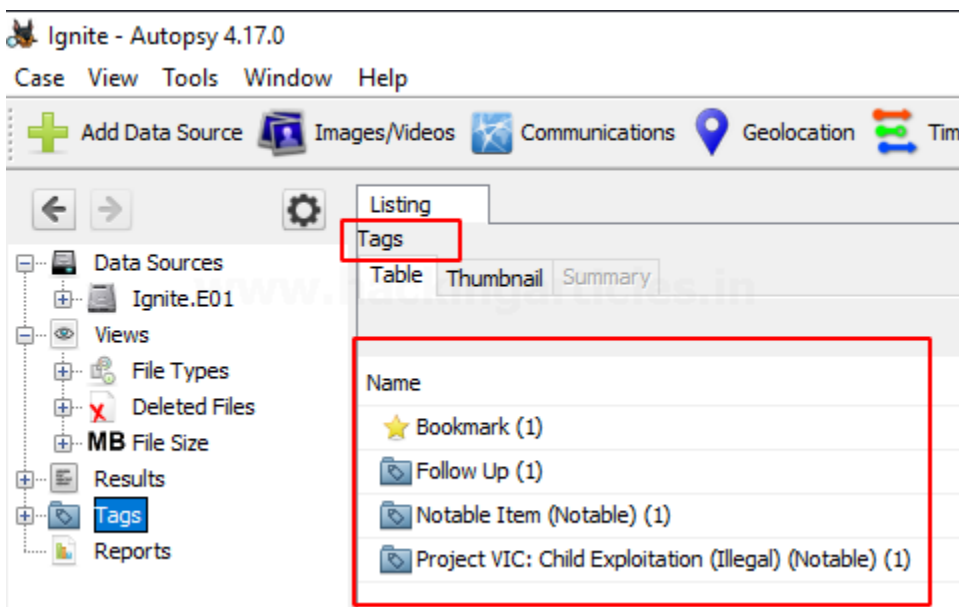


Add File Tag

Tagging can be used to create bookmarks, follow-up, mark as any notable item, etc.

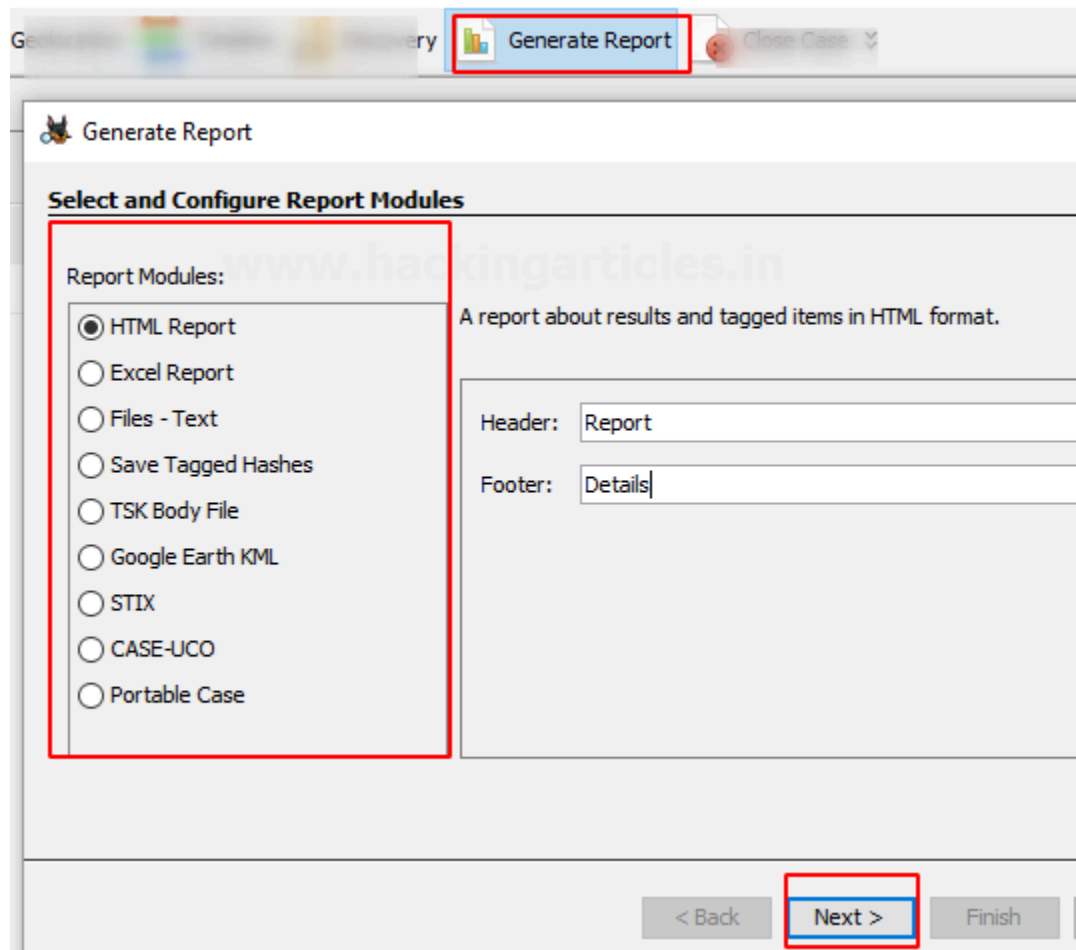


Now when you see the tags options, you will see that files were tagged according to various categories.



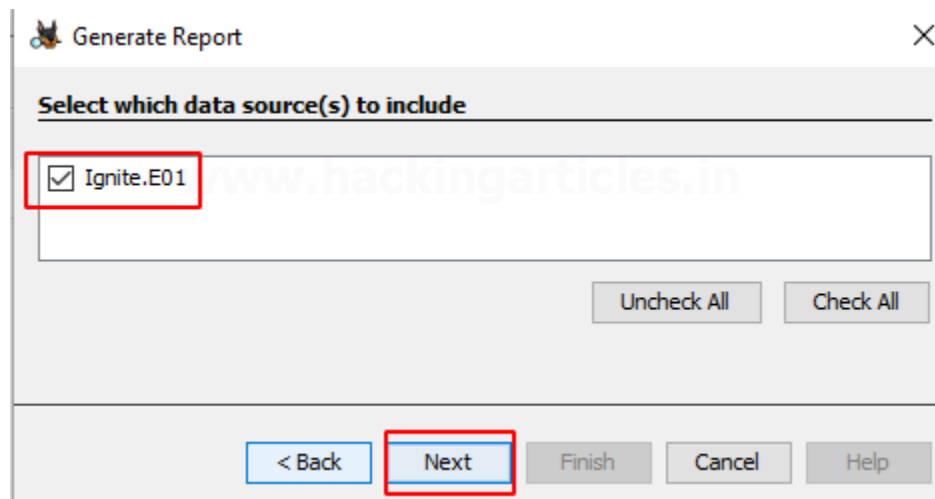
Generate Report

Once the investigation is done, the examiner can generate the report in various formats according to his preference.




The screenshot shows a software window titled "Generate Report". At the top, there is a toolbar with a "Generate Report" button (highlighted with a red box) and a "Close Case" button. The main area is titled "Select and Configure Report Modules". On the left, under "Report Modules:", there is a list of radio buttons: "HTML Report" (selected), "Excel Report", "Files - Text", "Save Tagged Hashes", "TSK Body File", "Google Earth KML", "STIX", "CASE-UCO", and "Portable Case". On the right, there is a description: "A report about results and tagged items in HTML format." Below this, there are input fields for "Header:" (containing "Report") and "Footer:" (containing "Details"). At the bottom right, there are three buttons: "< Back", "Next >" (highlighted with a red box), and "Finish".


Check the data source whose report needs to be generated.



The screenshot shows the same "Generate Report" window, but at a different step. The title bar now includes a close button (X). The main area is titled "Select which data source(s) to include". Below this, there is a list box containing "Ignite.E01" (highlighted with a red box). At the bottom right, there are two buttons: "Uncheck All" and "Check All". At the bottom of the window, there are five buttons: "< Back", "Next" (highlighted with a red box), "Finish", "Cancel", and "Help".

Here we chose to create the report in HTML format.

Source Module Name	Report Name	Created Time	Report File Path
 HTML Report		2020-11-28 15:42:58 IST	C:\Users\raj\Desktop\Ignite\Reports\Ignite HTML Rep

 Report Generation Progress...









Complete

HTML Report : <C:\Users\raj\Desktop\Ignite\Reports\Ignite HTML Report 11-28-2020-15-42-58\report.html>

Complete

Kudos! Your Autopsy Forensic Report is ready!

Report Navigation

-  Case Summary
-  Keyword Hits (1026)
-  Metadata (6)
-  Recycle Bin (4)
-  Tagged Files (4)
-  Tagged Images (4)
-  Tagged Results (0)
-  Web Downloads (3)

Autopsy Forensic Report

HTML Report Generated on 2020/11/28 15:42:58

Case:	Ignite
Case Number:	001
Number of data sources in case:	1
Examiner:	vishva

Image Information:

Ignite.E01

Timezone:	America/Los_Angeles
Path:	C:\Users\raj\Desktop\Ignite.E01

Software Information:

Autopsy Version:	4.17.0
Android Analyzer Module:	4.17.0
Central Repository Module:	4.17.0
Data Source Integrity Module:	4.17.0
Drone Analyzer Module:	4.17.0

9.Perform Registry analysis and get boot time logging using process monitor tool.

Process Monitor is an advanced monitoring tool for Windows that shows real-time file system, Registry and process/thread activity. It combines the features of two legacy Sysinternals utilities, Filemon and Regmon, and adds an extensive list of enhancements including rich and non-destructive filtering, comprehensive event properties such as session IDs and user names, reliable process information, full thread stacks with integrated symbol support for each operation, simultaneous logging to a file, and much more. Its uniquely powerful features will make Process Monitor a core utility in your system troubleshooting and malware hunting toolkit.

Process Monitor Capabilities Overview

1. **Comprehensive Data Capture:** Captures extensive data, including operation input/output parameters and thread stacks for root cause analysis.
2. **Non-Destructive Filtering:** Enables flexible filtering without losing any captured data.
3. **Detailed Process Insights:** Records reliable details such as image path, command line, user, and session ID for processes.
4. **Configurable Columns:** Allows customization and reorganization of event property columns.
5. **Advanced Filtering:** Supports setting filters for any data field, even those not displayed as columns.
6. **Scalable Logging:** Handles tens of millions of events and gigabytes of log data with advanced architecture.
7. **Process Tree Visualization:** Displays hierarchical relationships between processes in a trace.
8. **Native Log Format:** Preserves all captured data for reloading in other Process Monitor instances.
9. **Tooltips for Quick Access:** Includes process and detail tooltips for easy viewing of image information and formatted data.
10. **Boot-Time Logging:** Supports logging all operations from system boot for comprehensive traceability.

Procedure:-

Open the ProcessMonitor folder, you will see five files:

Eula.txt – The license agreement you'll have to accept before running procmon.

procmon.chm – The help file which contains all of the provided documentation.

Procmon.exe – The main EXE that will launch the correct procmon instance (x86 or x64).

Procmon64.exe – The x64 procmon binary.

Procmon64a.exe – The alpha 64 procmon binary.

Now run procmon by invoking the ~\ProcessMonitor\procmon.exe file.

You can see below a typical procmon capture in progress.

The moment you run procmon, it begins capturing many different kinds of Windows events.

Under the Operation column, there are various icons each representing different classes of Windows events.

Procmon captures events from five different classes:-

1. Registry
2. Filesystem
3. Network
4. Processes
5. Profiling events

Each event in all classes is represented in a single list pane of seven columns:-

Time of day – The time the event occurred.

Process name – The name of the process that triggered the event.

PID – The process identifier.

Operation – The type of event like if the process opened a file, changed a registry key value, etc.

Path – The path to the object the event interacted with like a file path, registry path, etc.

Result – This column will contain numerous values to indicate the result of the event. This value can be as simple as SUCCESS or specific to the event like REPARSE, BUFFER OVERFLOW, NAME NOT FOUND, etc.

Detail – This column contains all of the nitty-gritty detail once you pinpoint an event you'd like to see. If you'd rather not see a certain column or would like to see what other columns you have available, right-click on any column header and choose Select columns. You'll be presented with a dialog box where you can customize the viewable columns.

In the event window, double-click on an event. You can find many more details about the process and the event itself by viewing the event, process and stack tabs.

Enabling and Disabling Captures

You have complete control over the capture process.

You can either disable the entire capture process or disable capturing by event class. On the top menu bar, you'll see a magnifying glass icon (below). If the

magnifying glass is a red X over it, that means the capture is disabled. Otherwise, the capture is enabled. If you'd rather be more selective, you can also control the capture of each event class. In the menu bar, you'll see five of the same icons being displayed in the operation column. By clicking on these buttons, you can enable and disable entire event classes.

Conclusion:

Hence we can successfully perform Registry analysis and get boot time using process monitor tool.

10. Perform File type detection using Autopsy tool

Autopsy is an open-source tool that is used to perform forensic operations on the disk image of the evidence. The forensic investigation that is carried out on the disk image is displayed here. The results obtained here are of help to investigate and locate relevant information. This tool is used by law enforcement agencies, local police and can also be used in the corporates to investigate the evidence found in a computer crime. It can likewise be utilized to recuperate information that has been erased.

Creating a new Case

Run the Autopsy tool on your Windows Operating System and click on "New Case" to create a new case.

Then fill in all the necessary case information like the case name and choose a base directory to save all the case data in one place.

You can also add additional optional information about the case if required.

Now let us add the type of data source. There are various types to choose from.

Disk Image or VM file: This includes the image file which can be an exact copy of a hard drive, media card, or even a virtual machine.

Local Disk: This option includes devices like Hard disk, Pen drives, memory cards, etc.

Logical Files: It includes the image of any local folders or files.

Unallocated Space Image File: They include files that do not contain any file system and run with the help of the ingest module.

Autopsy Logical Imager Results: They include the data source from running the logical imager.

XRY Text Export: This includes the data source from exporting text files from XRY, Now let us add the data source. Here we have a previously created image file, so we will add the location of that file.

Next, you will be prompted to **Configure the Ingest Module**.The contents of the Ingest module are listed below:

The contents of the Ingest module are listed below:

INGEST MODULE	
Recent Activity	It is used to discover the recent operations that were performed on the disk, like the files that were viewed recently.
Extension Mismatch Detector	It is used to identify files whose extensions were tampered with or had been changed to hide the evidence.
Hash Lookup	It is used to identify a particular file using its hash value.
File Type Identification	This is used to identify files based on their internal file signatures than just the file extensions.
Embedded File Extractor	It is used to extract embedded files like .zip, .rar, etc. and use those files for analysis.
Keyword Search	This is used to search for any particular keyword or a pattern in the image file.
Email Parser	This is used to extract information from email files if the disk holds any email database information.
Encryption Detection	This helps to detect and identifies encrypted password-protected files.
Interesting File Identifier	Using this feature the examiner is notified when results pertaining to the set of rules that are defined to identify a particular type of file.
PhotoRec Carver	This helps the examiner to recover files, photos, etc. from the unallocated space on the image disk.
Virtual Machine Extractor	It helps to extract and analyze if any Virtual machine is found on the disk image.
Data Source Integrity	It helps to calculate the hash value and store them in the database.

Data Source information displays basic metadata. Its detailed analysis is displayed at the bottom. It can be extracted one after the other.

Views

File Type: It can be classified in the form of File extension or MIME type.

It provides information on file extensions that are commonly used by the OS whereas MIME types are used by the browser to decide what data to represent. It also displays deleted files.

Note: These file types can be categorized depending on Extension, Documents, Executables.

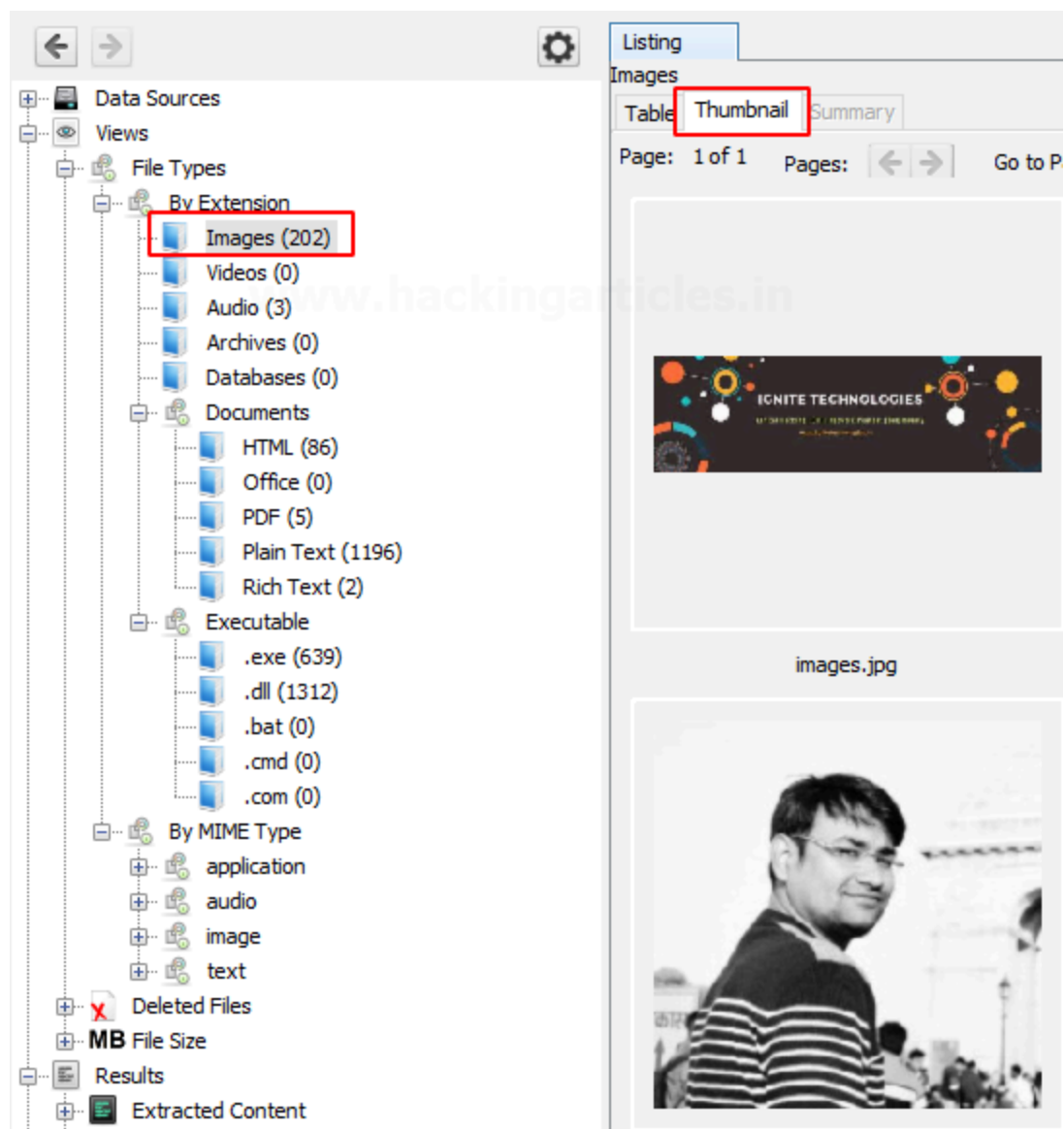
By Extension

In the category Filetypes by extension and you can see that this has been sub-divided into file types like images, video, audio, archives, databases, etc.



Let us click on images and explore the images that have been recovered.


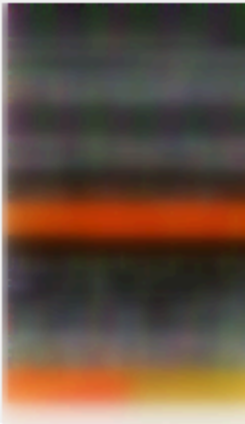
We can also view the thumbnail of the images.



On viewing the thumbnail, you can view the file metadata and details about the image.

TableThumbnailSummary

Page: 1 of 1
Pages:
Go to Page:
Image

images.jpg

/img_Ignite.E01/images.jpg

HexTextApplicationFile MetadataContextResultsAnnotationsOther

From The Sleuth Kit istat Tool:

MFT Entry Header Values:

Entry: 49 Sequence: 1

\$LogFile Sequence Number: 16885331

Allocated File

Links: 1

\$STANDARD_INFORMATION Attribute Values:

Flags: Archive

Owner ID: 0

Security ID: 271 (S-1-5-21-1276730070-1850728493-30201

Created: 2020-11-26 08:20:24.482672700 (PST)

File Modified: 2020-11-26 08:20:24.667704200 (PST)

MFT Modified: 2020-11-26 09:00:35.829441300 (PST)

Accessed: 2020-11-26 08:59:53.860554000 (PST)

\$FILE_NAME Attribute Values:

Flags: Archive

Name: \$R3RSEBH.jpg

Parent MFT Entry: 40 Sequence: 1

Allocated Size: 8192 Actual Size: 7641

Created: 2020-11-26 08:20:24.482672700 (PST)

File Modified: 2020-11-26 08:20:24.667704200 (PST)

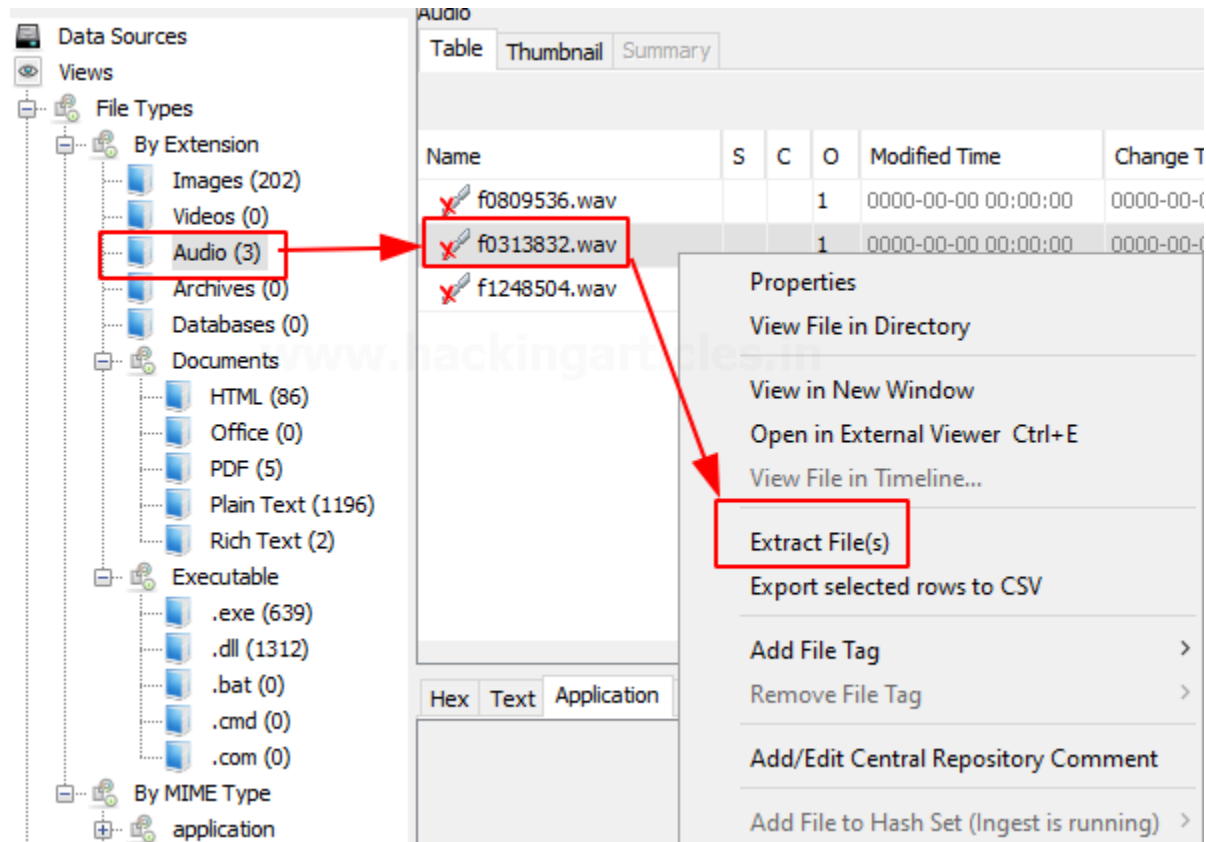
MFT Modified: 2020-11-26 08:59:01.714957400 (PST)

Accessed: 2020-11-26 08:59:01.704974100 (PST)

\$OBJECT_ID Attribute Values:

Object Id: 3fd39b21-2f45-11eb-ala0-001b10002aec

Here we can also view a few audio files that have been recovered. We can extract these files from the system and hear to them using various software.



Documents

The documents are categorized into 5 types: HTML, office, PDF, Plain Text, Rich Text.

On exploring the documents option, you can see all the HTML documents present, you can click on the important ones to view them.

The screenshot shows a file explorer on the left and a web browser interface on the right. The file explorer displays a tree view with categories like 'By Extension' and 'By MIME Type'. The 'HTML (86)' folder is highlighted. The web browser interface shows a list of HTML files, with 'Forensic Investigation Autopsy Forensic Browser in Linux.html' selected. Below the list, the 'Indexed Text' tab is active, displaying the content of the selected file.

File Explorer Tree View:

- By Extension
 - Images (202)
 - Videos (0)
 - Audio (3)
 - Archives (0)
 - Databases (0)
 - Documents
 - HTML (86)**
 - Office (0)
 - PDF (5)
 - Plain Text (1196)
 - Rich Text (2)
- Executable
 - .exe (639)
 - .dll (1312)
 - .bat (0)
 - .cmd (0)
 - .com (0)
- By MIME Type
 - application
 - audio
 - image
 - text
- Deleted Files
- File Size
- Results
- Extracted Content
- Metadata (6)
- Recycle Bin (4)
- Web Downloads (3)
- Keyword Hits
- Hashset Hits
- E-Mail Messages

Web Browser Interface:

Listing

HTML

Table Thumbnail Summary

Name	S	C	O
Forensic Investigation Autopsy Forensic Browser in Linux.html			1
a.html			1
a_002.html			1
fastbutton.html			1
like.html			1

Hex Text Application File Metadata Context Results Annotations Other Occ

Strings Indexed Text Translation

Page: 1 of 3 Page Matches on page: - of - Match

Hacking Articles

Raj Chandel's Blog

- * CTF Challenges
- * Penetration Testing
- * Web Penetration Testing
- * Red Teaming
- * Donate us
- * Courses We Offer
 - o Bug Bounty
 - o Computer Forensics
 - o Ethical Hacking
 - o Red Teaming

Forensic Investigation: Autopsy Forensic Browser in Linux

posted inCyber Forensics on August 13, 2020 by Raj Chandel

SHARE






Save

On exploring the PDF option, you can also find the important PDF in the disk image.

Listing

PDF

Table Thumbnail Summary

Name	S	C	O	Modified Time	Chan
 \$IO2Y1Z5.pdf			1	2020-11-26 09:04:18 PST	2020-
 \$RO2Y1Z5.pdf			1	2020-02-29 11:02:57 PST	2020-
 Android Pentesting.pdf			1	2020-10-23 01:42:19 PDT	2020-
 Bug Bounty Course Details.pdf				2020-11-26 09:04:18 PST	2020-
 f0184904.pdf			1	0000-00-00 00:00:00	0000-

Hex Text Application File Metadata Context Results Annotations Other Occurrences

1 of 5 29%

POWERED BY IGNITE TECHNOLOGIES

ANDROID PENTESTING

Similarly, the various Plain text files can also be viewed. You can also recover deleted plain text files.

Version

Images (202)

Media (0)

Audio (3)

Archives (0)

Databases (0)

Documents

HTML (86)

Office (0)

PDF (5)

Plain Text (1196)

Rich Text (2)

Executable

.exe (639)

.dll (1312)

.bat (0)

.cmd (0)

.com (0)

IME Type

Application

Audio

Image

Text

Files

Content

data (6)

de Bin (4)

Downloads (3)

Hits

Hits

Messages

g Items

Name	S	C	O	Modified Time
\$IK1MRRO.txt			1	2020-11-26 08:56:
\$RK1MRRO.txt			1	2020-11-26 08:55:
USB.txt			1	2020-09-09 07:15:
Ignite.E01.txt				2020-11-26 08:56:
f0484218.txt			1	0000-00-00 00:00:

Hex

Text

Application

File Metadata

Context

Results

Analysis

Strings

Indexed Text

Translation

Page: 1 of 1 Page

Matches on page: - of - Mat

```

NOTICE: The imaging operation was cancelled!

Created By AccessData® FTK® Imager 4.3.1.1

Case Information:
Acquired using: ADI4.3.1.1
Case Number: 001
Evidence Number: AU001
Unique description: Hacking Articles
Examiner: Vishva
Notes:

-----

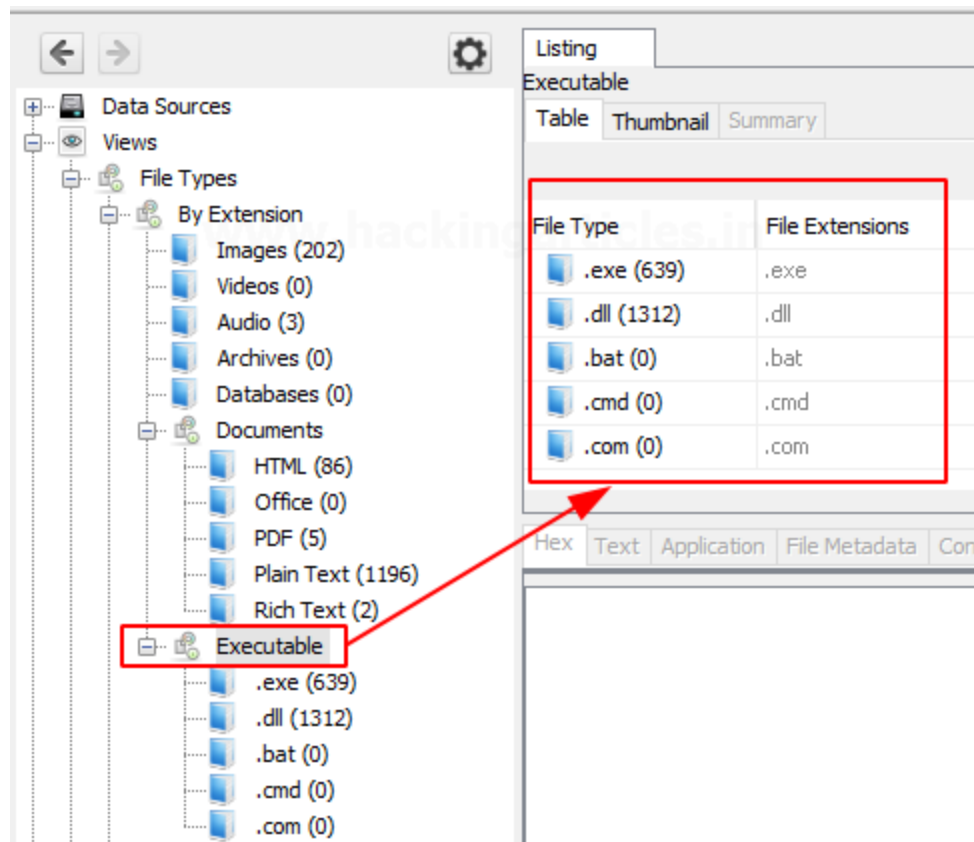
Information for E:\Ignite:

Physical Evidentiary Item (Source) Information
[Device Info]
Source Type: Logical
[Drive Geometry]
Bytes per Sector: 512
Sector Count: 125,821,080
[Physical Drive Information]
Removable drive: False
Source data size: 61436 MB
Sector count: 125821080

```

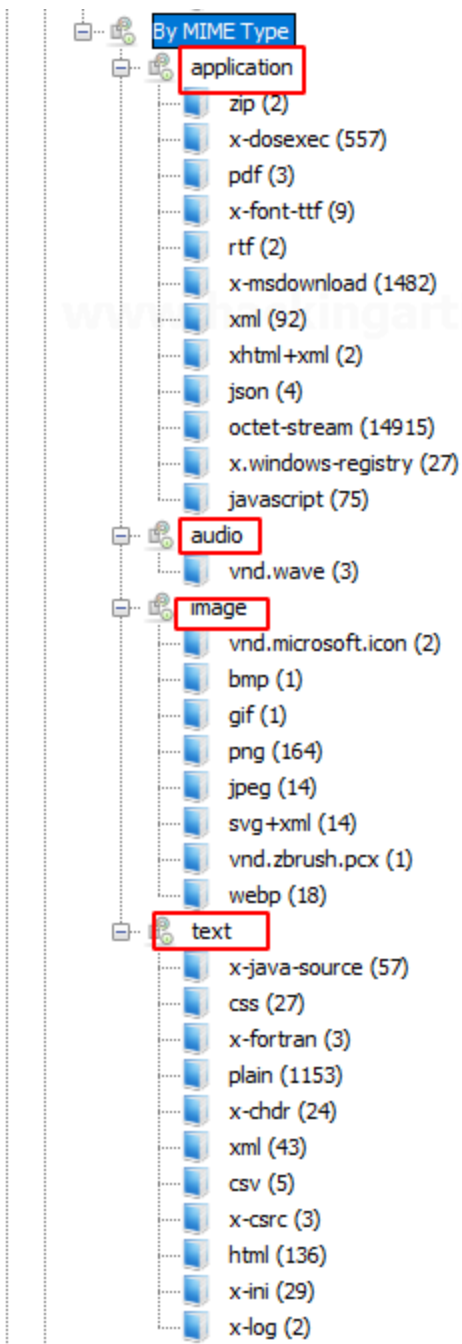
Executables

These file types are then sub-divided into .exe, .dll, .bat, .cmd and .com.



By MIME Type

In this type of category, there are four sub-categories like application, audio, image, and text. They are divided further into more sections and file types.



Deleted Files: It displays information about the deleted file which can be then recovered.

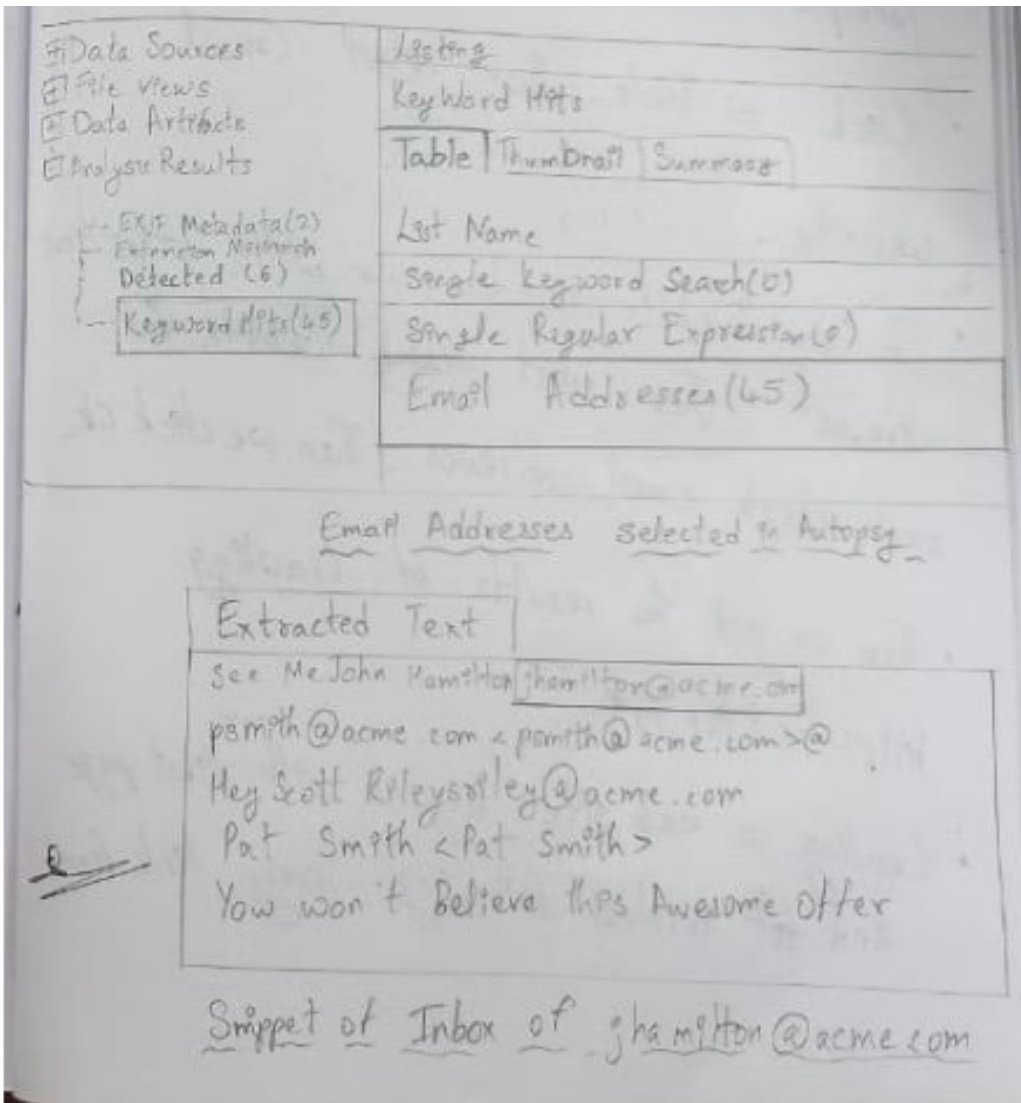
The screenshot shows the 'File System' view of a forensic tool. The left sidebar displays a tree structure with 'Data Sources', 'Views', 'File Types', and 'Results'. The 'File Types' section is expanded, showing 'By Extension' and 'By MIME Type'. The 'Deleted Files' folder is selected. The main pane shows a list of files with the following columns: Name, S, C, O, and Modified Time.

Name	S	C	O	Modified Time
20201014.mem			0	2020-10-13 13:39:50 PDT
adencrypt.dll			0	2020-05-11 21:03:46 PDT
adencrypt_gui.exe			0	2020-05-11 21:03:46 PDT
adfbfs_globals.dll			0	2020-05-11 21:03:46 PDT
adfs_globals.dll			0	2020-05-11 21:03:46 PDT
ADG_EULA.rtf			1	2020-02-05 15:48:36 PST
ADIso.exe			0	2020-05-11 21:03:46 PDT
ADIsoDLL.dll			0	2020-05-11 21:03:48 PDT
adshattrdefs.dll			0	2020-05-11 21:03:48 PDT
adtz_globals.dll			0	2020-05-11 21:03:48 PDT
ad_globals.dll			0	2020-05-11 21:03:46 PDT
ad_log.dll			0	2020-05-11 21:03:46 PDT
boost_chrono-vc140-mt-1_59.dll			0	2020-05-11 21:03:48 PDT
boost_date_time-vc140-mt-1_59.dll			0	2020-05-11 21:03:46 PDT
boost_filesystem-vc140-mt-1_59.dll			0	2020-05-11 21:03:50 PDT
boost_regex-vc140-mt-1_59.dll			0	2020-05-11 21:03:50 PDT
boost_system-vc140-mt-1_59.dll			0	2020-05-11 21:03:50 PDT
boost_thread-vc140-mt-1_59.dll			0	2020-05-11 21:03:50 PDT
FTK Imager.exe			0	2020-05-11 21:04:10 PDT

MB Size Files: In this, the files are categorized based on their size starting from 50MB. This allows the examiner to look for large files.

The screenshot shows the 'MB File Size' view of a forensic tool. The left sidebar displays a tree structure with 'Data Sources', 'Views', 'File Types', and 'Results'. The 'MB File Size' section is selected. The main pane shows a list of files categorized by size range.

Size Range
MB 50 - 200MB (1)
MB 200MB - 1GB (2)
MB 1GB+ (3)



11.LSB/MSB

12.PSN ratio

```
import numpy as np
```

```
import cv2
```

```
def calculate_psnr(original_image, compressed_image):
```

```
    # Ensure both images have the same dimensions
```

```
    if original_image is None or compressed_image is None:
```

```
        raise ValueError("One or both image files could not be loaded. Check the file paths.")
```

```
    if original_image.shape != compressed_image.shape:
```

```
        raise ValueError("Input images must have the same dimensions.")
```

```
    # Calculate Mean Squared Error (MSE)
```

```
    mse = np.mean((original_image - compressed_image) ** 2)
```

```
    if mse == 0: # means images are identical
```

```
        return float('inf')
```

```
    # Calculate PSNR using the formula
```

```
    max_pixel = 255.0 # max pixel value for 8-bit images
```

```
    psnr = 20 * np.log10(max_pixel / np.sqrt(mse))
```

```
    return psnr
```

```
# Example usage:
```

```
original = cv2.imread("C:/Users/user/Downloads/dog.png")
```

```
compressed = cv2.imread("C:/Users/user/Downloads/jpeg-optimizer_doggy.jpeg")
```

```
if original is None or compressed is None:
```

```
    print("Error: One or both images could not be loaded. Please check the file paths.")
```

```
else:
```

```
psnr_value = calculate_psnr(original, compressed)
print(f"PSNR: {psnr_value} dB")
```

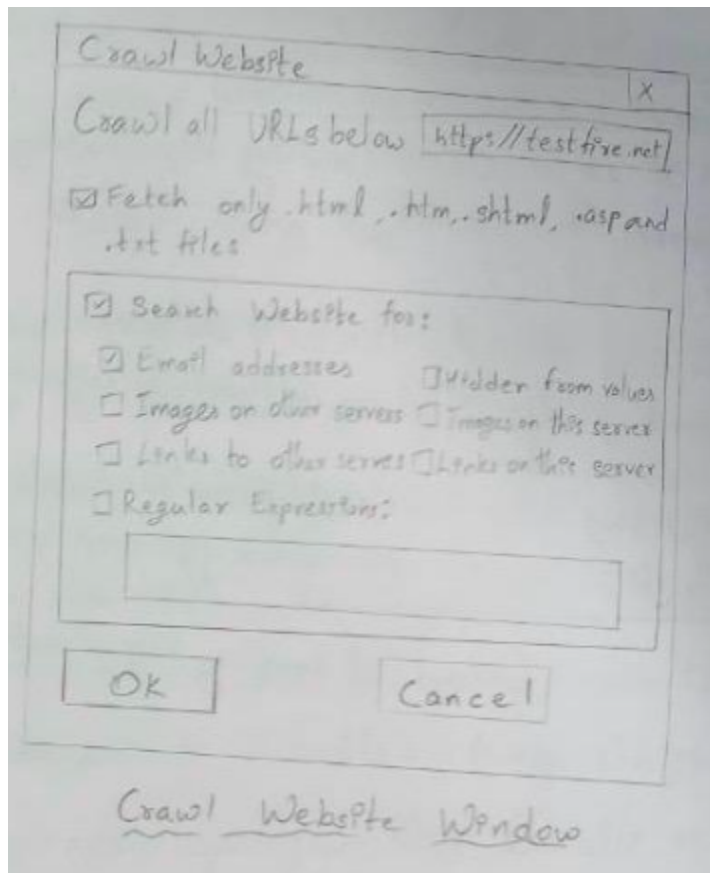
13. Perform Memory capture and analysis using FTK imager tool

14. Perform Network analysis using the Network Miner tool

15. Sam spade for crawling a website

Steps:

- * We download Sam Spade from the official website. Select the required as per the prompts to install.
- * Click on Tools, Then select Crawl website .
- * In Crawl Website window, we write <https://testfire.net>. Then select search website for and check email addresses. Then we click ok.
- * Then we get the results of crawling <http://testfire.net>.
- * Crawling a website begins with first page and involves following every link found.



16. Brute Force

```

import string
import itertools

def bruteforce_attack(password, max_attempts=2000000):
    # Define printable characters (characters you expect in the password)
    chars = string.printable.strip() # Using printable characters minus non-printable
ones
    attempts = 0

    # Iterate through all possible lengths from 1 to the length of the password
    for length in range(1, len(password) + 1):
        # Generate all possible combinations of characters of the current length
        for guess in itertools.product(chars, repeat=length):
            attempts += 1
            guess = ''.join(guess) # Convert tuple to string

            # If attempts exceed the limit, return None
            if attempts >= max_attempts:
                return (attempts, None)

            # Check if the generated guess matches the password
            if guess == password:
                return (attempts, guess)

    # If the password is not found after all combinations, return None
    return (attempts, None)

# Example usage:
password = "abc" # The password you're trying to crack
max_attempts = 2000000 # You can set this limit based on your preference

attempts, cracked_password = bruteforce_attack(password, max_attempts)

```

```
if cracked_password:
    print(f"Password cracked! It took {attempts} attempts: {cracked_password}")
else:
    print(f"Password could not be cracked within {attempts} attempts.")
```

17.Client Server Programs:

a.Capital to Small:

```
import socket
import threading

def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 9999
    server_socket.bind((host, port))
    server_socket.listen(5)
    print(f'Server started on {host}:{port}')

    while True:
        client_socket, addr = server_socket.accept()
        print(f'Got a connection from {addr}')
        data = client_socket.recv(1024).decode('utf-8')
        print(f'Received from client: {data.lower()}')
        response = 'Thank you for connecting'
        client_socket.send(response.encode('utf-8'))
        client_socket.close()

server_thread = threading.Thread(target=start_server, daemon=True)
server_thread.start()
```

Server started on 127.0.0.1:9999

```

import socket

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 9999
    client_socket.connect((host, port))
    message = 'Hello, server!'
    client_socket.send(message.encode('utf-8'))
    response = client_socket.recv(1024).decode('utf-8')
    print(f'Received from server: {response.lower()}')
    client_socket.close()

start_client()

```

```

Got a connection from ('127.0.0.1', 64529)
Received from client: hello, server!
Received from server: thank you for connecting
Got a connection from ('127.0.0.1', 64552)
Received from client: khoorqvhuyhu
Got a connection from ('127.0.0.1', 64556)
Received from client: khoorqvhuyhu

```

Client server caesar cipher

b.Caesar Cipher

```

import socket
import threading
def encrypt(text,s):
    result = ""

    # traverse text
    for i in range(len(text)):
        char = text[i]

        # Encrypt uppercase characters
        if (char.isupper()):
            result += chr((ord(char) + s-65) % 26 + 65)

        # Encrypt lowercase characters
        else:
            result += chr((ord(char) + s - 97) % 26 + 97)

    return result

def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 9999
    server_socket.bind((host, port))
    server_socket.listen(5)
    print(f'Server started on {host}:{port}')

    while True:
        client_socket, addr = server_socket.accept()
        print(f'Got a connection from {addr}')
        data = client_socket.recv(1024).decode('utf-8')
        if data:
            text,shift=data.split('|')
            shift=int(shift)
            print(f'Received from client: {text}')
            data1=encrypt(text,3)
            print(f'Received from client: {data1}')
            response = 'Thank you for connecting'
            client_socket.send(response.encode('utf-8'))
            client_socket.close()

server_thread = threading.Thread(target=start_server, daemon=True)
server_thread.start()

```

Server started on 127.0.0.1:9999

```

import socket

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 9999
    client_socket.connect((host, port))
    message = input("Enter text: ")
    shift = input("Enter shift value: ")
    client_socket.send(f'{message}|{shift}'.encode('utf-8'))
    response = client_socket.recv(1024).decode('utf-8')
    print(f'Received from server: {response}')
    client_socket.close()

start_client()

```

```

Got a connection from ('127.0.0.1', 64946)
Enter text: Hello
Enter shift value: 3
Received from client: Hello
Received from client: Khoor
Received from server: Thank you for connecting

```

c.MD5

```

import socket
import threading
import hashlib

def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 9999
    server_socket.bind((host, port))
    server_socket.listen(5)
    print(f'Server started on {host}:{port}')

    while True:
        client_socket, addr = server_socket.accept()
        print(f'Got a connection from {addr}')
        data = client_socket.recv(1024).decode('utf-8')
        print(f'Received from client: {data}')
        data1=hashlib.md5(b'data').hexdigest()
        print(f'Hashed value: {data1}')
        response = 'Thank you for connecting'
        client_socket.send(response.encode('utf-8'))
        client_socket.close()

server_thread = threading.Thread(target=start_server, daemon=True)
server_thread.start()

```

```
import socket

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 9999
    client_socket.connect((host, port))
    message = input("Enter text: ")
    client_socket.send(message.encode('utf-8'))
    response = client_socket.recv(1024).decode('utf-8')
    print(f'Received from server: {response}')
    client_socket.close()

start_client()
```

```
Got a connection from ('127.0.0.1', 65430)
Enter text: Hello
Received from client: Hello
Hashed value: 8d777f385d3dfec8815d20f7496026dc
Received from server: Thank you for connecting
```

d.DES

```
pip install Pycryptodome
```

```
import socket
import threading
from Crypto.Cipher import DES
from Crypto.Util.Padding import pad,unpad
import base64
def des(text,key):
    cipher=DES.new(key,DES.MODE_ECB)
    paddedText=pad(text.encode(),DES.block_size)
    cipherText=cipher.encrypt(paddedText)
    return base64.b64encode(cipherText).decode('utf-8')

def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 9999
    server_socket.bind((host, port))
    server_socket.listen(5)
    print(f'Server started on {host}:{port}')

    while True:
        client_socket, addr = server_socket.accept()
        print(f'Got a connection from {addr}')
        data = client_socket.recv(1024).decode('utf-8')
        if data:
            text,key=data.split('|')
            key=key.encode('utf-8')
            processText=des(text,key)

            print(f'Received from client: {data}')
            print(f'Processed : {processText}')
            response = 'Thank you for connecting'
            client_socket.send(response.encode('utf-8'))
            client_socket.close()

server_thread = threading.Thread(target=start_server, daemon=True)
server_thread.start()
```

Server started on 127.0.0.1:9999

```
import socket

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 9999
    client_socket.connect((host, port))
    message = input("Enter text: ")
    client_socket.send(message.encode('utf-8'))
    response = client_socket.recv(1024).decode('utf-8')
    print(f'Received from server: {response}')
    client_socket.close()

start_client()
```

```
Got a connection from ('127.0.0.1', 56098)
Enter text: hello server|qwertyui
Received from client: hello server|qwertyui
Processed : fUKTN+oLi4bzZBpd0pZ18Q==
Received from server: Thank you for connecting
```