# Purpose of the experiment:

The purpose of the experiment is to evaluate the weakness in the existing client-server model and overcoming that with the help of a peer – to – peer model protocol named IPFS. By using this protocol, the nodes in a peer – to – peer network does not provide the actual data and the destination to where the data is getting transferred. This helps in maintaining the anonymity during the and prevents the recon phase of the Advanced Persistent Threats.

The files which were chosen for this experiment are 5 excel files for easy recognition and understanding of data during analysis in Wireshark. The data in the files consist of columns and rows which represent data belonging to a particular entity in each row.

10mb.xls

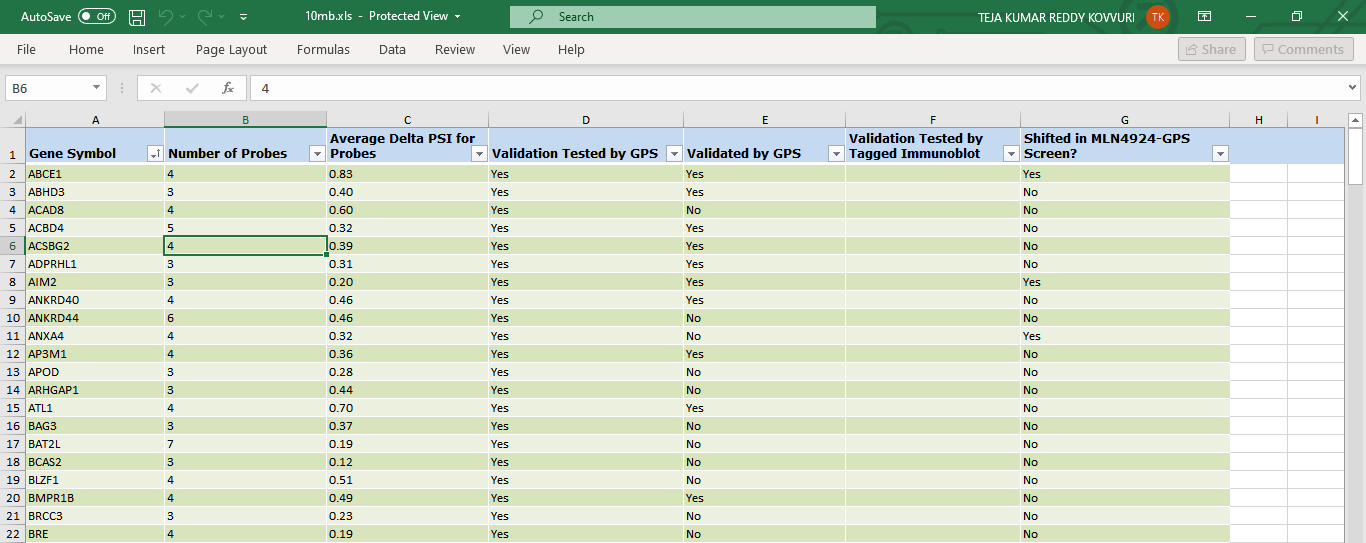


Figure 1-10mb.xls

The Appendix Section has the remaining figures for 20mb.xls, 70mb.csv, 110mb.csv, and 1.3gb.csv files.

# Python – Client Server model

**Server Machine – Ubuntu ipfs Server (192.168.56.103)**

The server machine hosts the files which will be transferred to the clients when requested.

**Client Machine – Ubuntu ipfs Client – 1 (192.168.56.104)**

The Client Machine-1 requests the server for the file and the server sends the requested file.

**Client Machine – Ubuntu ipfs Client – 2 (192.168.56.105)**

The Client Machine -2 also performs the similar action like the Client Machine-1 to request the file from Server Machine.

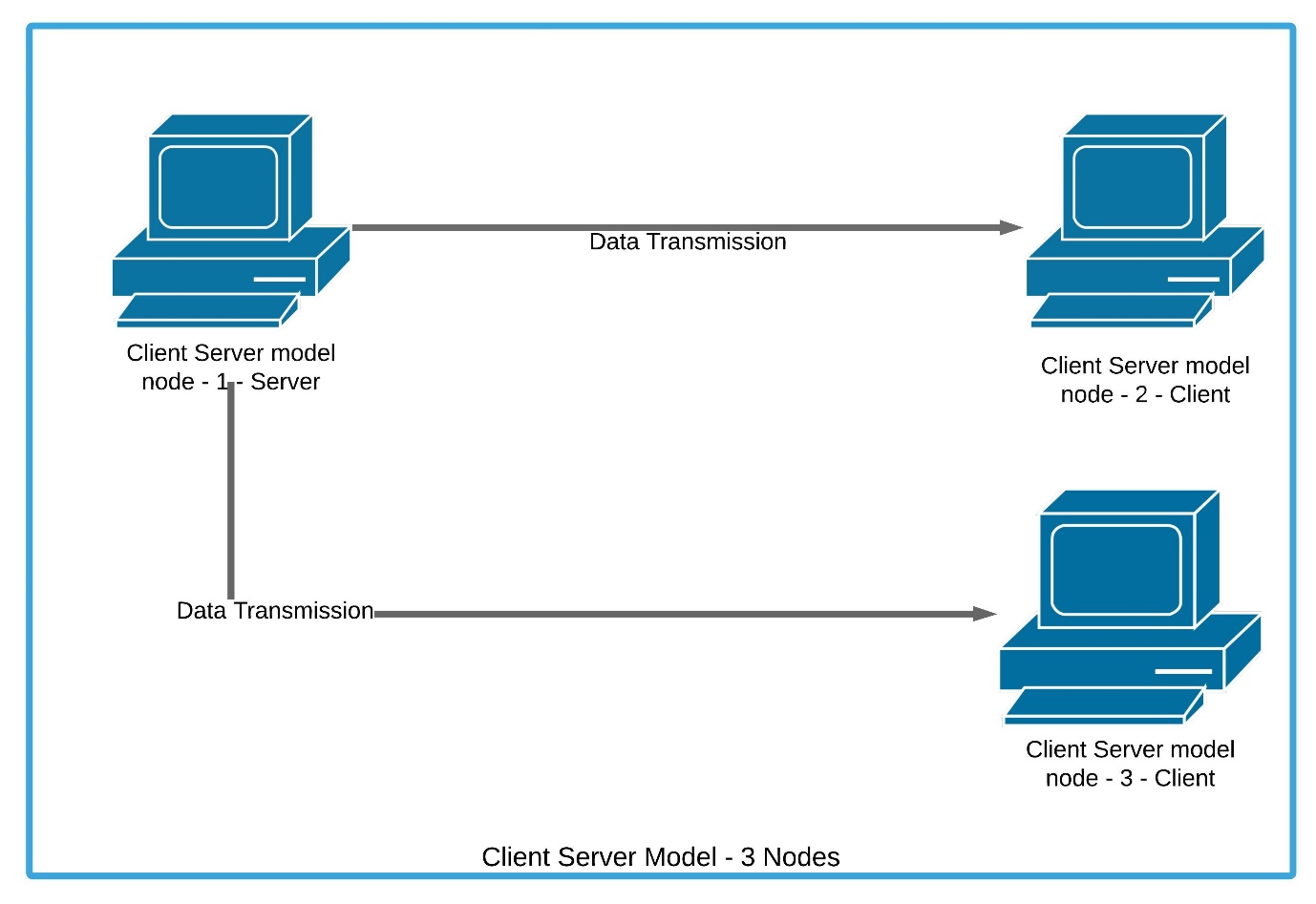


Figure 2-Client-Server Model

## Roadmap:

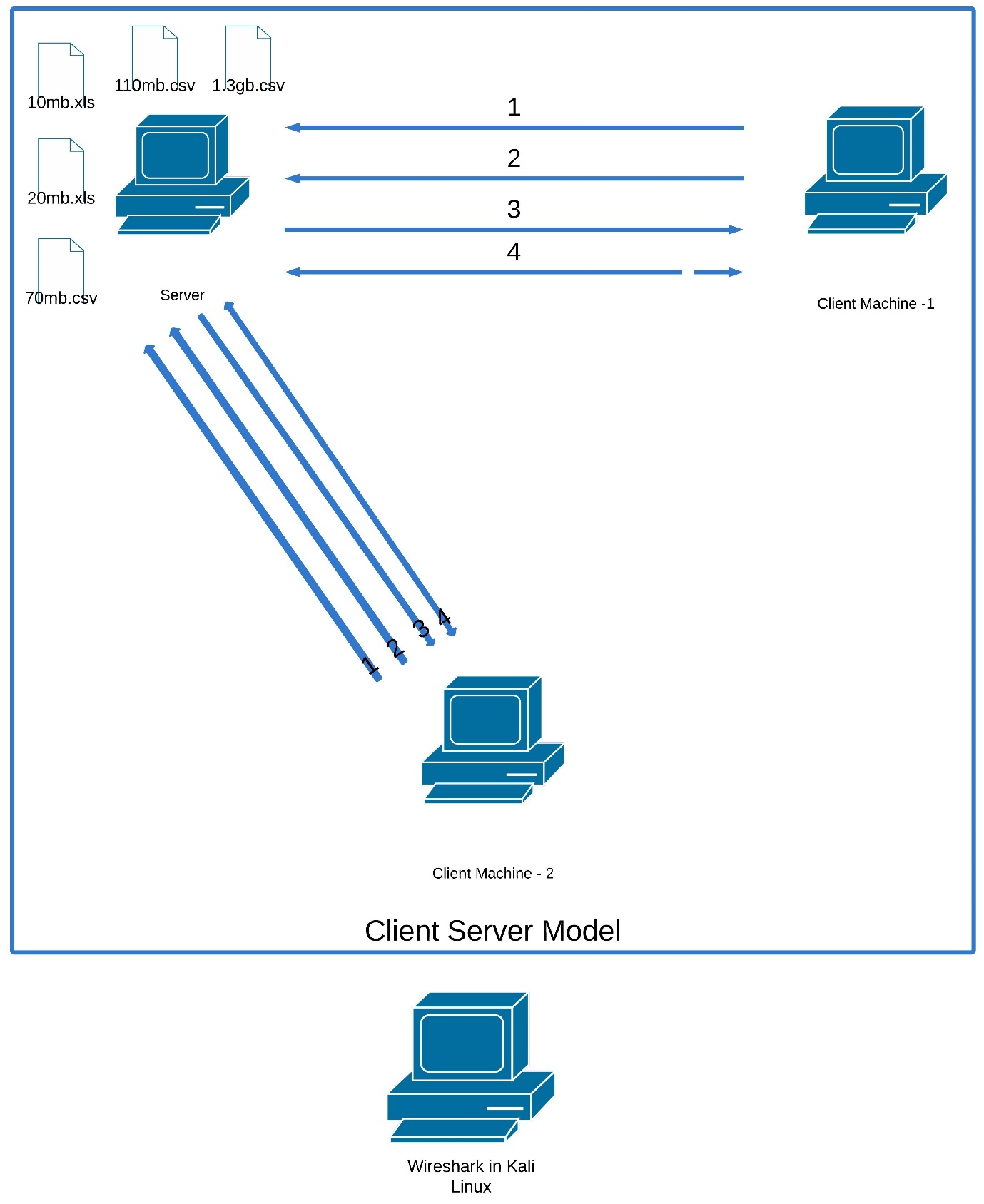


Figure 3-Roadmap of Client Server Model

Steps:

1. Client Machine 1 or 2 connects with Server Machine.
2. Client Machine 1 or 2 requests the file from server.
3. Server Transfers the file to the requested Client Machine 1 or 2.
4. The source and Destination Addresses are visible in Wireshark along with the actual data which is not encrypted.

A client server model has been established with the help of Python programming language. The server starts and listens for connections from clients. Once the clients get connected, the file transfer between the nodes take place as per the request of the client. The server accepts connection from each client at a time and finishes the transmission of data before accepting connection from another client.

With the help of Wireshark, the results will be analysed, and further experiments are evaluated. The figure-6 representing Wireshark in Kali Linux can be seen in Appendix section.

## Experiment – 1 – 10mb.xls:

1. Server starts and listens for client to establish a connection and send the requested file.



Figure 4-Python Client Server model - Server machine

1. Client starts and establishes connection with server and request for a file.

Client Output:

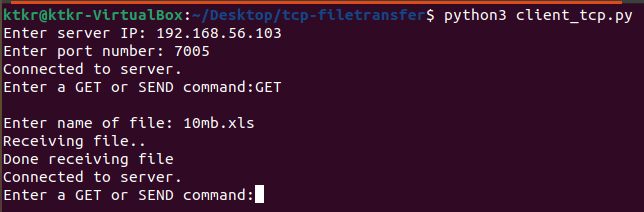
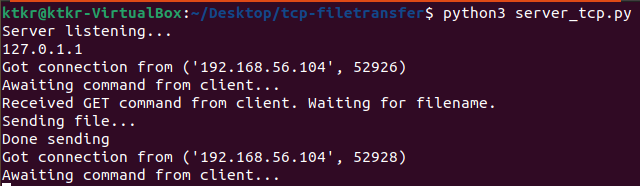


Figure 5-Python Client Server model - Client Machine-1

The file has been successfully received by the client.

Server Output:

Figure 6-Server Machine sending data

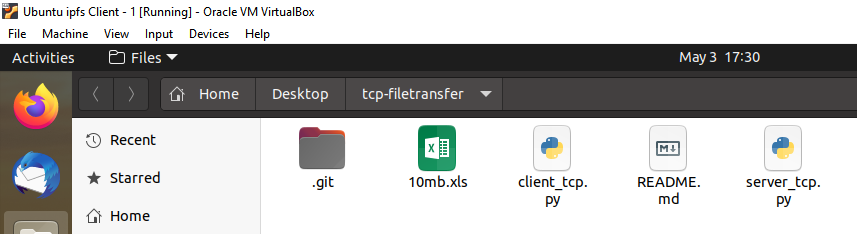


Figure 7-Transferred file visible in Client Machine-1

The traffic between server and client has been filtered in Wireshark to analyze the data and it has been identified that an adversary can see the actual data.

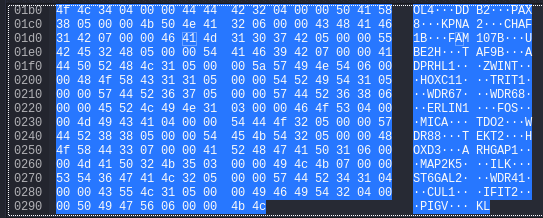


Figure 8-Actual data visibility in Wireshark

The same data “WDR67” is also identified in the actual file named 10mb.xls.

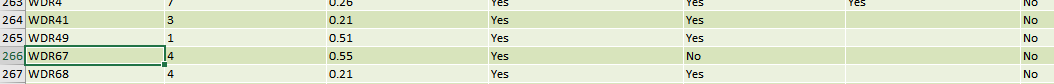
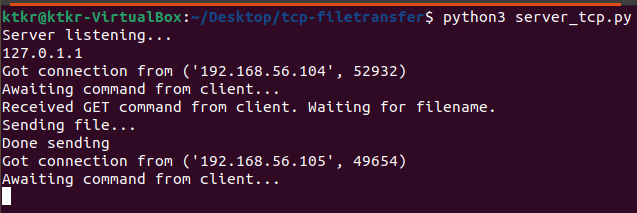


Figure 9-Data found in Actual File (10mb.xls)

### Table for Experiment – 1 – 10mb.xls:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 1 | 192.168.56.104 | Yes | WDR67 | Yes – WDR67 |

After serving Client -1, the server is ready to serve Client – 2.

Figure 10-Server ready to serve to next client(Client Machine-2)

## Experiment – 2 - 20mb.xls:

In the 2nd Experiment, the identification of actual data is performed on the 20mb.xls file in Wireshark. The identified data is “149.29”. The figure-6 is available in Appendix Section. Here the 149.29 number is identified in the actual excel file. The actual data in Excel file can be seen in figure-7 in the Appendix Section.

### TableforExperiment – 2 - 20mb.xls**:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 2 | 192.168.56.105 | Yes | 149.29 | Yes – 149.29 |

## Experiment – 3 – 1.3gb.csv:

Next, the 1.3gb.csv file has been transferred from server to client.

The actual data is again identified in Wireshark. The figure-8 in Appendix section shows the data. The email “evie.hamby@gmail.com” has been identified in the actual excel file in Figure-9 of Appendix Section.

### Table - Experiment – 3 – 1.3gb.csv:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 1 | 192.168.56.104 | Yes | evie.hamby@gmail.com | Yes – evie.hamby@gmail.com |

## Experiment – 4 – 110mb.csv:

In this experiment, the “110mb.csv” file is transferred to the node - client – 2. The data identified in the Wireshark is “44005”. The Figure-10 in Appendix Section represents the data in Wireshark. The actual data in excel sheet can be seen in Figure-11 in Appendix Section.

### Table - Experiment – 4 – 110mb.csv:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 2 | 192.168.56.105 | Yes | 44005.63 | Yes – 44005.63 |

## Experiment – 5 – 70mb.csv:

In this experiment, the file “70mb.csv” is transferred to the node - client – 1. The actual data identified in the Wireshark is “Global Rank, Tld Rank”. The Figure-13 in Appendix Section represents the data. The actual data can be seen in the excel file of figure 14 in Appendix Section.

### Table - Experiment – 5 – 70mb.csv:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 1 | 192.168.56.104 | Yes | Global Rank, Tld Rank | Yes – Global Rank, Tld Rank |

This shows that there is no security, and anyone can see the plain text using tools like Wireshark. An adversary will also know to which node/address the data has been passed to as well. This leads to more several attacks by adversary on the communication.

# IPFS – Peer – to - Peer model

**Server machine – Ubuntu ipfs Server (192.168.56.103)**

The IPFS Server machine connects with all the nodes using IPFS Protocol. Then it sends the file/data to all the nodes and the file/data will only be accepted by the legit Client Machine.

**Client machine – Ubuntu ipfs Client – 1 (192.168.56.104)**

By using the IPFS Protocol, the Client Machine-1 connects with all the other nodes including the Server Machine. It requests the file/data from the Server Machine which will be served by the Server.

**Client machine – Ubuntu ipfs Client – 2 (192.168.56.105)**

The Client Machine-2 also does the similar job like the Client Machine-1 and requests file/data from the Server Machine.

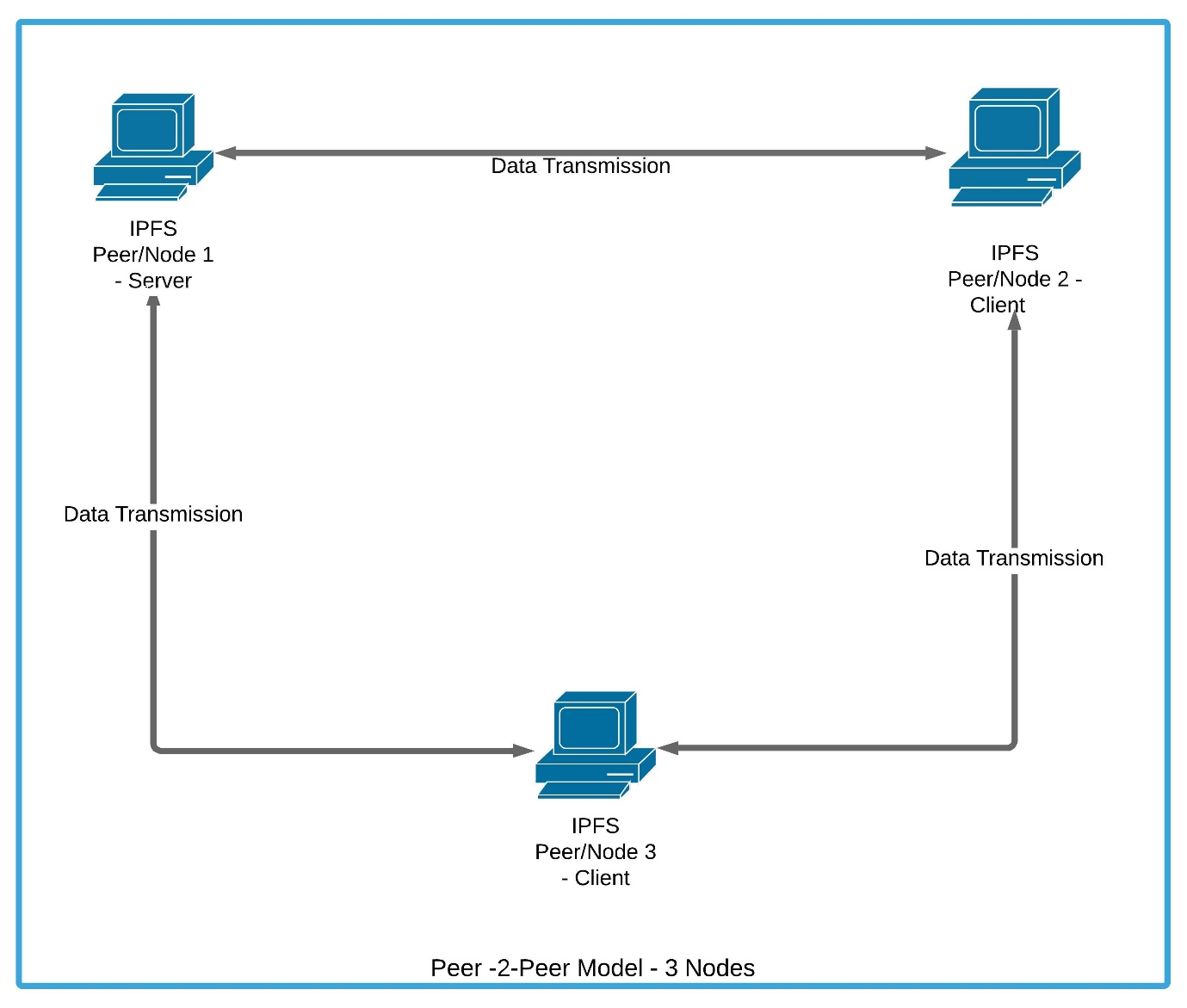


Figure 11-Peer-to-Peer Model using IPFS Protocol

## Roadmap:

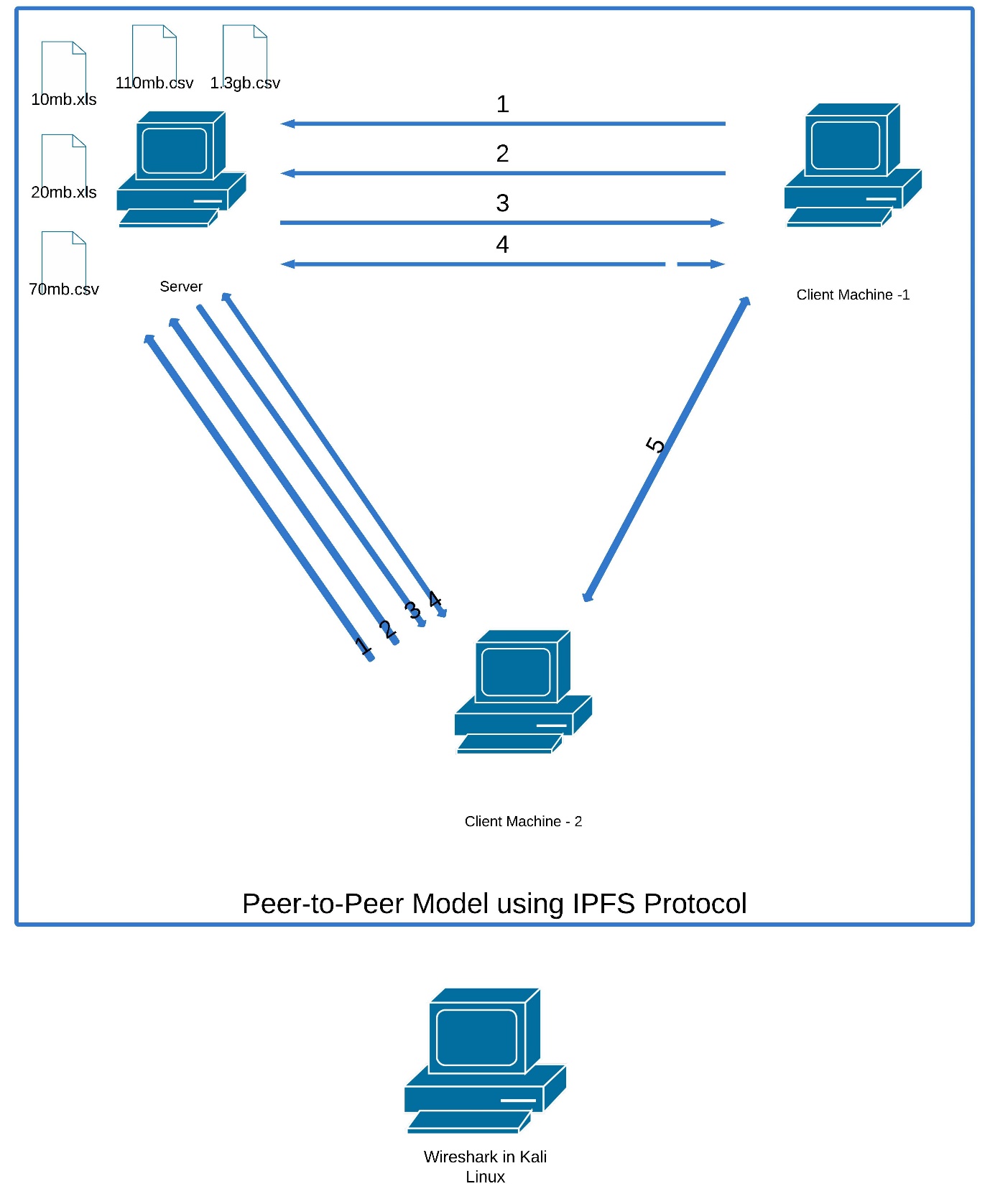


Figure 12-Peer-to-Peer Model using IPFS Protocol Roadmap

Steps:

1. Client Machine connects with Server Machine.
2. Client Machine requests the file from server.
3. Server Sends the Requested file (encrypted). The Server Broadcasts the data to all Nodes in the network.
4. Source and Destination addresses are not identifiable in Wireshark as all nodes participate in data transfer.
5. Clients Communicate with each other.

The same experiments have been conducted on the same nodes using IPFS protocol. In this experiment, the traffic will be analyzed to see any actual data or to identify the destination to where the data is transferred.

## Experiment – 1 – 10mb.xls:

**Server:**

IPFS stores the files in its database and generates a unique hash which will be used by the client/other node to request the file.

**ipfs add 10mb.xls**

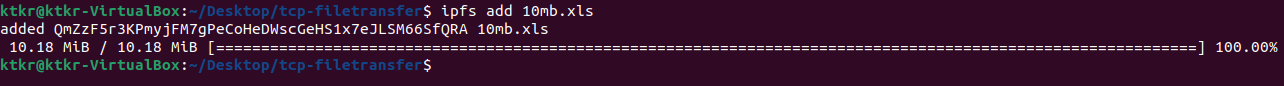


Figure 13-Peer-to-Peer Model Server Machine using IPFS Protocol

**Client - 1:**

In Client – 1, a request has been sent to the ipfs server/node for the file as follows:

**Ipfs get QmZzF5r3KPmyjFM7gPeCoHeDWscGeHS1x7eJLSM66SfQRA**

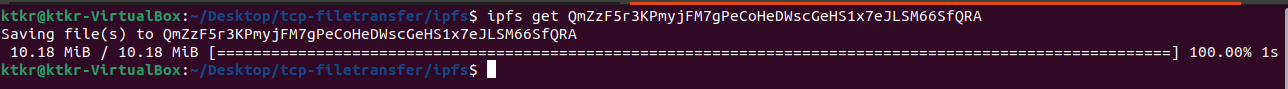


Figure 14-Peer-to-Peer Model Client Machine-1 using IPFS Protocol

With the help of Wireshark, the entire data transmission has been captured and analyzed.

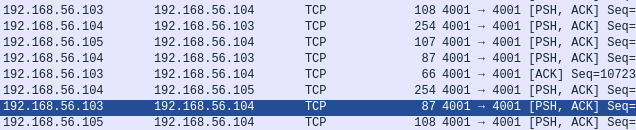


Figure 15-Analysis of Traffic using Wireshark in Kali Linux

In IPFS as a peer – to - peer model the actual data will be transferred to all the nodes and the legitimate node will receive the data and others will reject the data. This helps in maintaining the anonymity and any adversary may not know which node is the source and which node is the destination as every node act as both source and destination as shown in the above picture.

The actual data is not visible, and encryption has been applied on the data before passing the data to the other nodes.

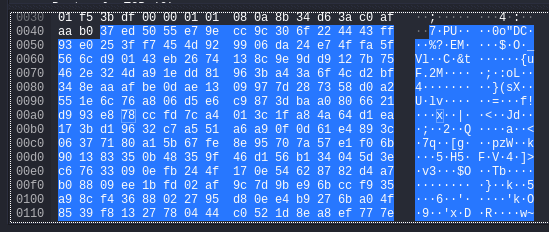


Figure 16-Data Analysis in Wireshark - Data is Encrypted

### Table - Experiment – 1 – 10mb.xls:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 1 | 192.168.56.104 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |

## Experiment -2 – 20mb.xls:

The 20mb.xls file has been transferred from source to destination.

**Server:**

**ipfs add 20mb.xls**

Figure-20 in Appendix Section represents this function.

**Client - 2:**

**ipfs get QmZ6LnMDVoRabqXZ6ZNH7r651GCTdXWbjyDshi8wVL4au3**

Figure-21 in Appendix Section represents this function.

Here as well Wireshark is unable to identify the actual data. Figures-22, and 23 represent the functions in Appendix Section. Figure-22 shows the traffic for 20mb.xls file. Figure-23 shows the encrypted data in Wireshark.

### Table - Experiment -2 – 20mb.xls:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 2 | 192.168.56.105 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |

## Experiment – 3 – 1.3gb.csv:

**Server:**

**ipfs add 1.3gb.csv**

Figure-24 shows the server machine using IPFS Protocol for 1.3gb.csv in Appendix Section.

**Client - 1:**

**ipfs get QmPwHhvmAAx8P3pjJVKRq3eyjUv7q7fqWvvpecrPo2ZsAP**

Figure-25 shows the Client Machine-1 using IPFS Protocol for 1.3gb.csv file in Appendix Section.

Figure-26 in Appendix Section shows the traffic in Wireshark.

Figure-27 in Appendix Section shows the encrypted data in Wireshark.

### Table - Experiment – 3 – 1.3gb.csv:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 1 | 192.168.56.104 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |

In all the above experiments conducted using IPFS, the actual source and destination are unknown, and the actual data is also not visible and encrypted by IPFS protocol.

## Experiment – 4 – 110mb.csv:

**Server:**

**ipfs add 110mb.csv**

Figure-28 in Appendix Section shows the Peer-to-Peer model Server Machine using IPFS Protocol for 110mb.csv file.

**Client - 2:**

**ipfs get QmTBCQKJJSibbc18hujtES3PtT7QSgyYfGjqaxoyjCd3Zb**

Figure-29 in Appendix Section shows the Peer-to-Peer Model Client Machine-2 using IPFS Protocol for 110mb.csv file.

Wireshark was unable to identify the actual data.

Figure-30 in Appendix Section shows the traffic in Wireshark.

Figure-31 in Appendix Section shows the encrypted data in Wireshark.

### Table - Experiment – 4 – 110mb.csv:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 2 | 192.168.56.105 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |

## Experiment – 5 – 70mb.csv:

**Server:**

**ipfs add 70mb.csv**

Figure-32 in Appendix Section shows the Peer-to-Peer Model Server Machineusing IPFS Protocol for 70mb.csv file.

**Client - 1:**

**Ipfs get QmP1fnLScFNmmDXJbcwYyDBJks2surdVZUszHWMXQXVb2U**

Figure-33 in Appendix Section shows the Peer-to-Peer Model Client Machine-1 using IPFS Protocol for 70mb.csv file.

Wireshark did not find the actual plain data again.

Figure-34 in Appendix Section shows the traffic in Wireshark for 70mb.csv file.

Figure-35 in Appendix Section shows the encrypted data in Wireshark for 70mb.csv file.

### Table - Experiment – 5 – 70mb.csv:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 1 | 192.168.56.104 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |

# Summary Table for comparison of Client-Server model and Peer-to-Peer model:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Experiments results** | | | **Client – Server Model** | | | | **Peer-to-Peer Model** | | | |
| **10mb.xls** | **Server** | **Client** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** | **Destination address – actual client/node address** | **Original Data visibility** | **Sample of original data** | **Original data identification in actual file** |
| Ubuntu ipfs Server | Ubuntu ipfs Client – 1 | 192.168.56.104 | Yes | WDR67 | Yes – WDR67 | 192.168.56.104 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |
| **20mb.xls** | Ubuntu ipfs Server | Ubuntu ipfs Client – 2 | 192.168.56.105 | Yes | 149.29 | Yes – 149.29 | 192.168.56.105 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |
| **1.3gb.csv** | Ubuntu ipfs Server | Ubuntu ipfs Client – 1 | 192.168.56.104 | Yes | evie.hamby@gmail.com | Yes – evie.hamby@gmail.com | 192.168.56.104 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |
| **110mb.csv** | Ubuntu ipfs Server | Ubuntu ipfs Client – 2 | 192.168.56.105 | Yes | 44005.63 | Yes – 44005.63 | 192.168.56.105 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |
| **70mb.csv** | Ubuntu ipfs Server | Ubuntu ipfs Client – 1 | 192.168.56.104 | Yes | Global Rank, Tld Rank | Yes – Global Rank, Tld Rank | 192.168.56.104 – unknown in Wireshark as all nodes acts as source and destination | NO | Data encrypted | Yes – cannot identify in Wireshark as data was encrypted |

When the server machine using IPFS Protocol in Peer-to-Peer Model sends a file to all nodes, all the nodes receive the same data in encrypted form. The data also varies from node to node. The server sends data which is encrypted differently for each node.

# Appendix:

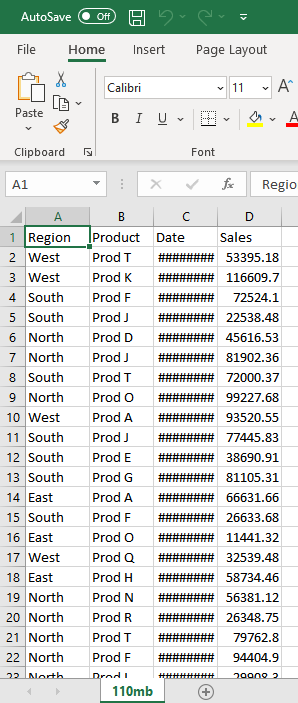


Figure 1-20mb.xls

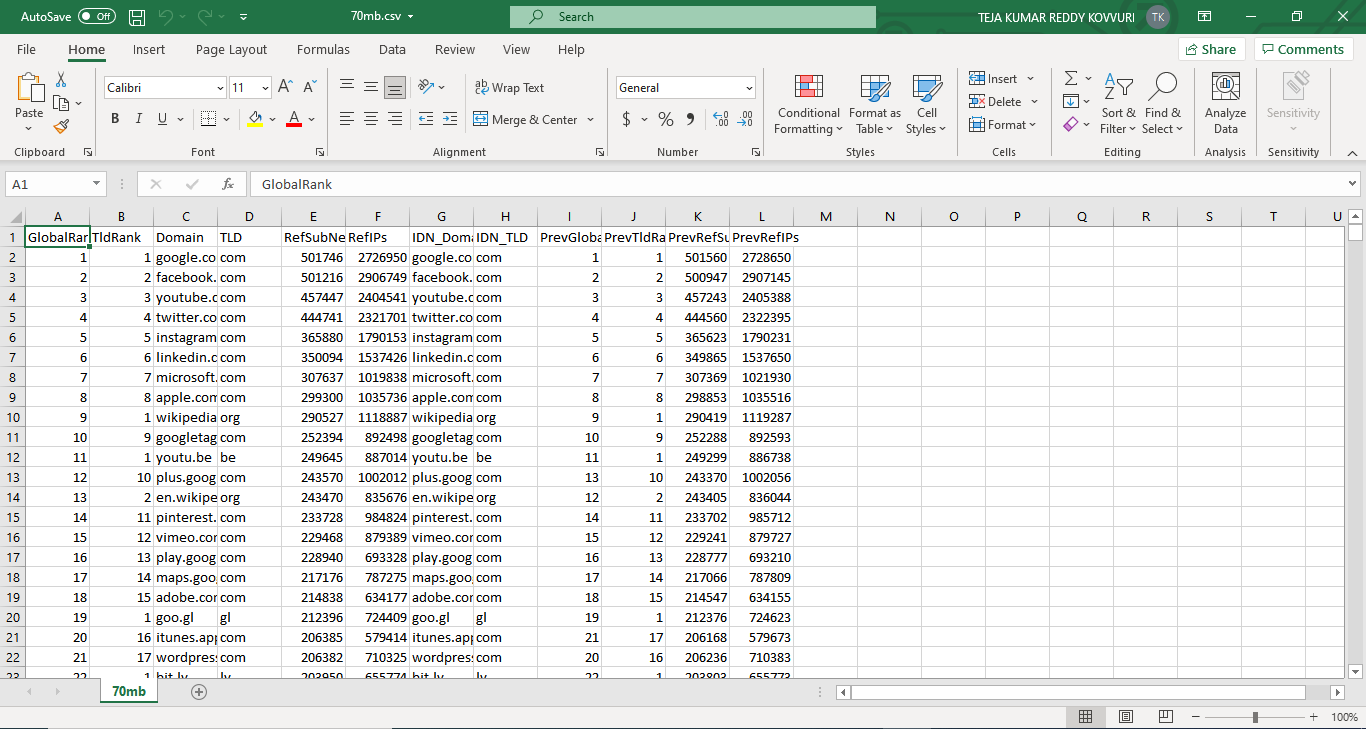


Figure 2-70mb.csv

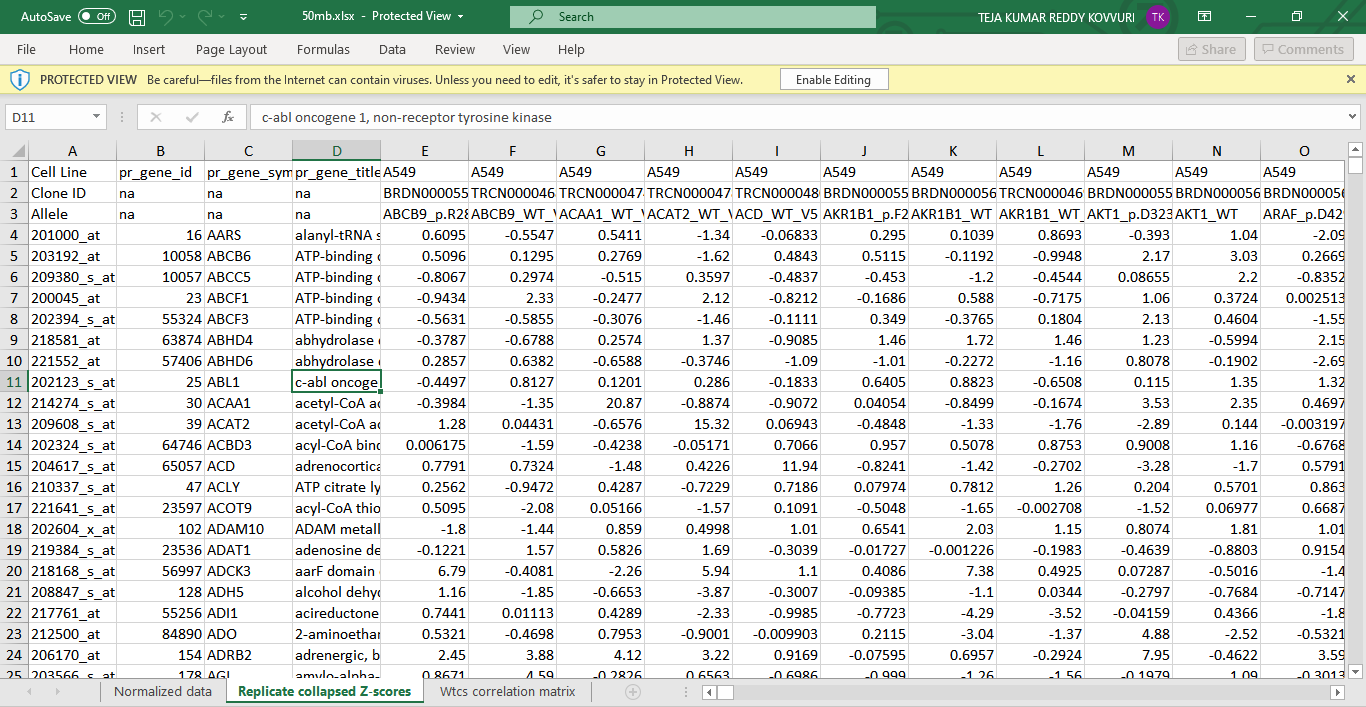


Figure 3-110mb.csv

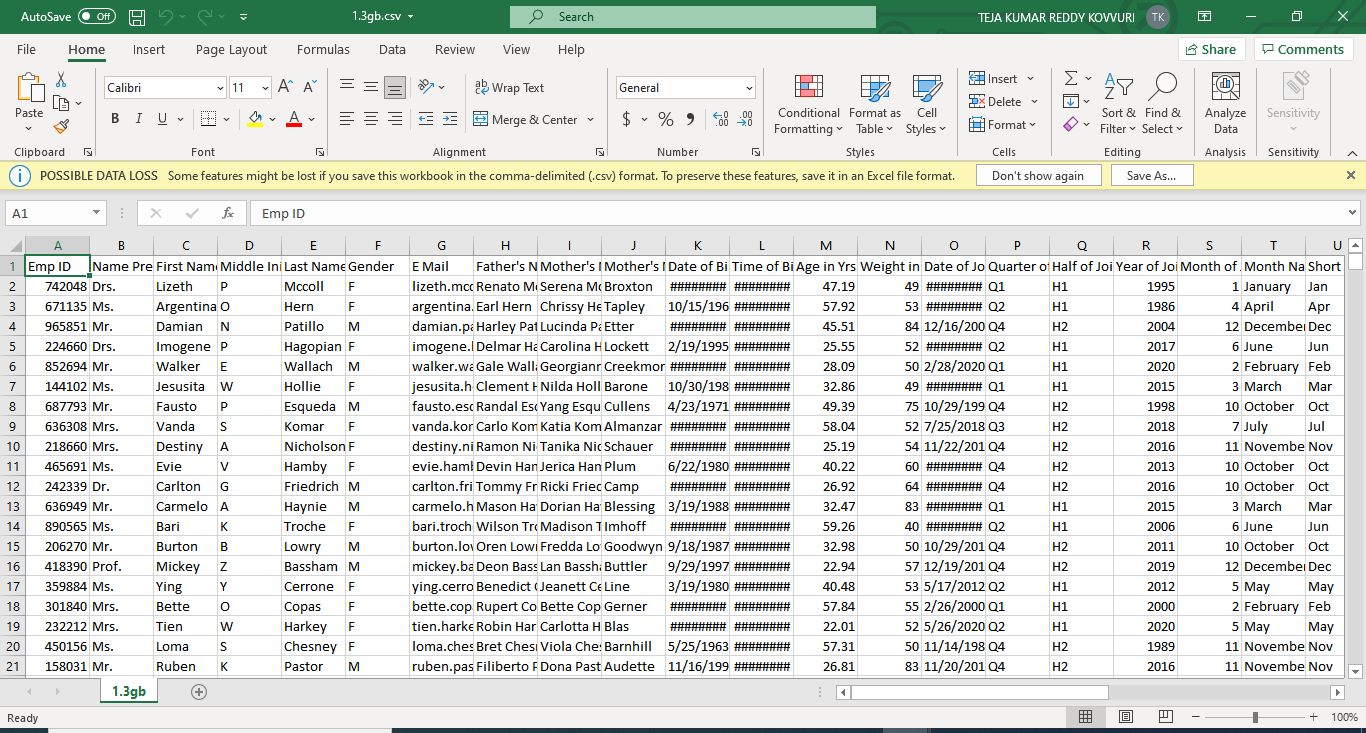


Figure 4-1.3gb.csv

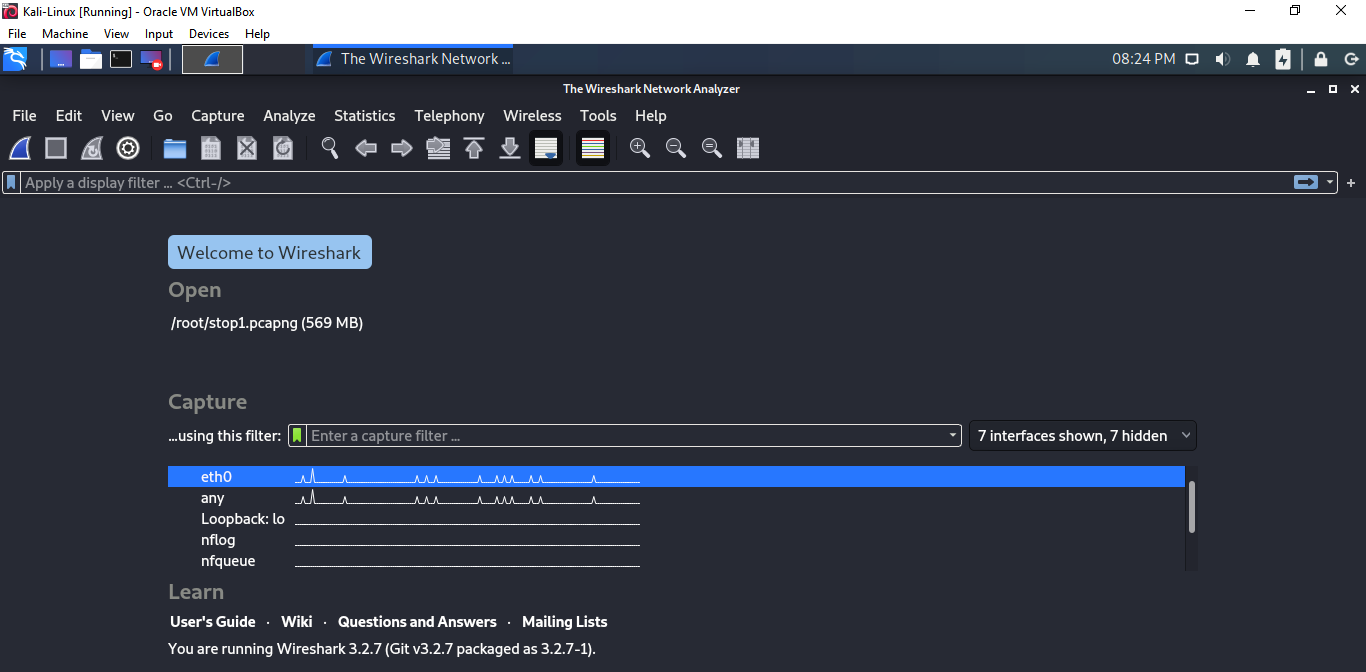


Figure 5-Wireshark in Kali Linux

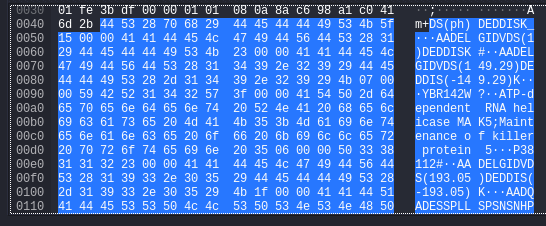


Figure 6-Data visibility in Wireshark

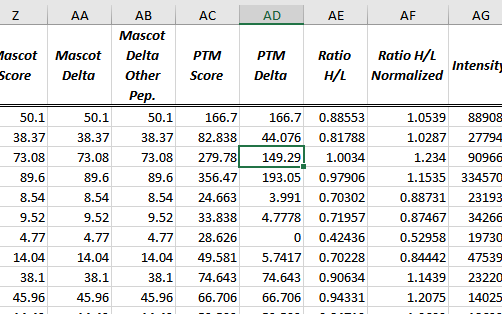


Figure 7-Actual data in Excel File

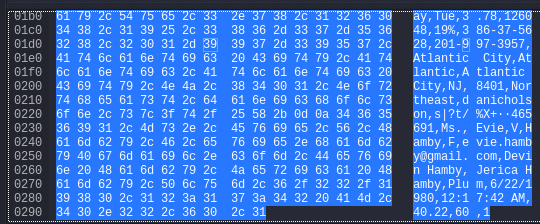


Figure 8-data identified - [evie.hamby@gmail.com](mailto:evie.hamby@gmail.com)

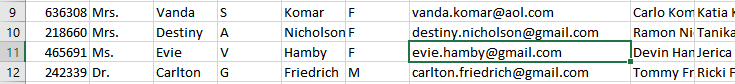


Figure 17-Actual Data in Excel File - evie.hamby@gmail.com

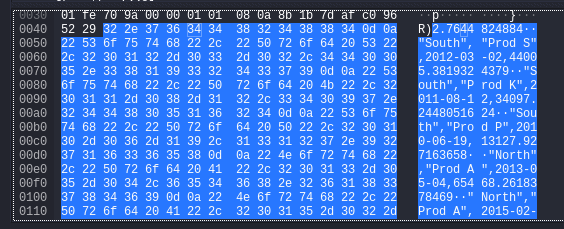


Figure 18-Data identified in Wireshark-44005

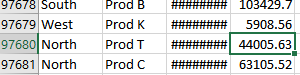


Figure 19-Actual Data in Excel FIle-44005

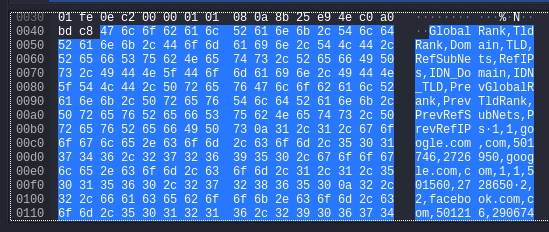


Figure 20-Data identified in Wireshark

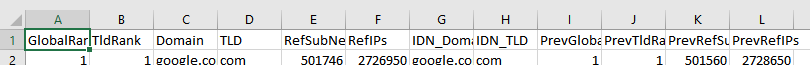


Figure 21-Actual data in Excel Data

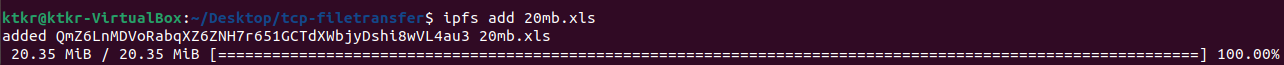


Figure 22-Peer-to-Peer Model Server Machine for 20mb.xls file



Figure 23-Peer-to-Peer Model CLient Machine -2 using IPFS Protocol

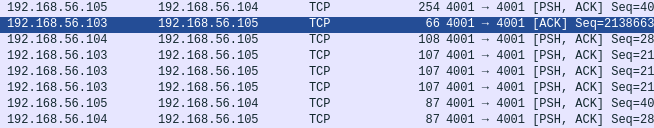


Figure 24-Traffic Analysis in Wireshark for 20mb.xls file

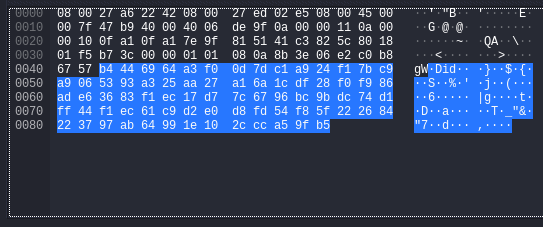


Figure 25-Encrypted data for 20mb.xls file in Wireshark



Figure 26-Peer-to-Peer Model Server Machine using IPFS Protocol for 1.3gb.csv file



Figure 27-Peer-to-Peer Model Client Machine-1 using IPFS Protocol for 1.3gb.csv file

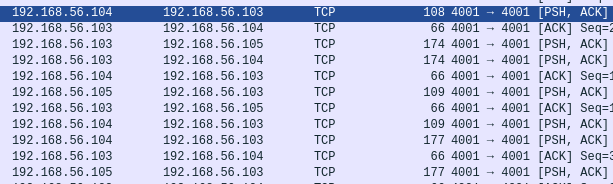


Figure 28-Traffic in Wireshark

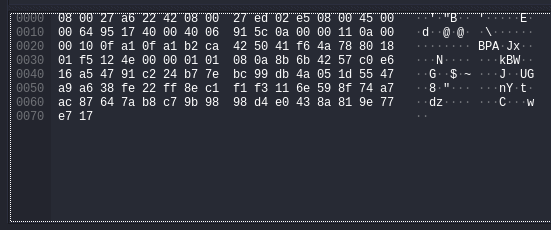


Figure 29-Encrypted Data in Wireshark



Figure 30-Peer-to-Peer Model Server Machine using IPFS Protocol for 110mb.csv file



Figure 31-Peer-to-Peer Model Client Machine-2 using IPFS Protocol for 110mb.csv file

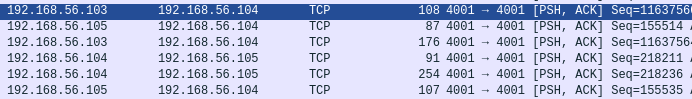


Figure 32-Traffic in Wireshark

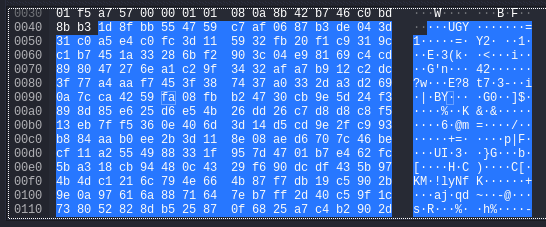


Figure 33-Encrypted Data in Wireshark



Figure 34-Peer-to-Peer Model Server Machine using IPFS Protocol for 70mb.csv file

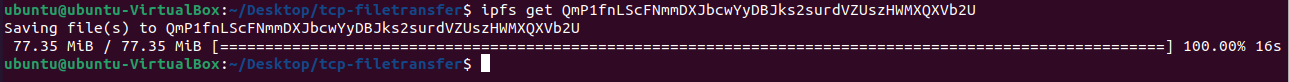


Figure 35-Peer-to-Peer Model Client Machine-1 using IPFS Protocol for 70mb.csv file

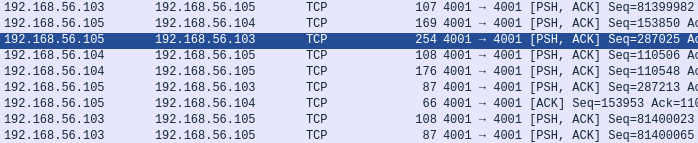


Figure 36-Traffic in Wireshark for 70mb.csv file

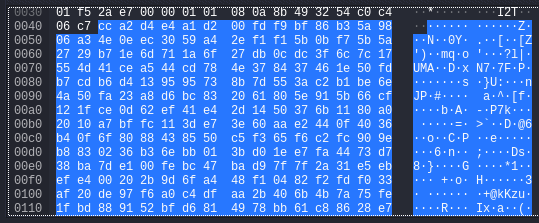


Figure 37-Encrypted data in Wireshark