



# Internet Traffic Classification Tool

By:

Fatima Hasan 17L-4020

Samra Fakhar 17L-4031

Nuzha Khalid 17L-4162

Shanzay Gauhar 17L-4236

# INTRODUCTION





# Introduction

- Identification of different types of packets
- Identification of application layer protocols such as HTTP/S, DNS, SMTP, FTP, VoIP and more
- Important for ISPs and different applications such as firewalls and intrusion detection systems
- Different methods exist
- One of the methods is identification based on fixed port numbers
- Live capture using Wireshark
- Wireshark for parsing pcap file



# Some Well Known Port Numbers

Port Number	Protocol
20	File Transfer Protocol FTP-Control
21	FTP-Data
25	Simple Mail Transfer Protocol SMTP
53	Domain Name System DNS
80	Hypertext Transfer Protocol HTTP
443	HTTPS

BACKGROUND





# Background

- Many methods of classifying network traffic including port number based, payload based, flow statistics based and behaviour based
- Payload-based classification uses Deep Packet Inspection (DPI) to identify the type of traffic
- Stores these patterns inside a database and then accesses the DB to detect the type of packet
- Can be used to detect network anomalies and is used in intrusion detection systems
- Does not work well with the applications that do not have well known patterns such as P2P applications
- Additional cost of maintaining the database
- Invades the privacy policies of the users



## Background (contd.)

- Flow based statistics
- It uses the information that is available in packet headers for example length of the packet, TCP window size, packet interarrival time etc
- This information is then used by different machine learning techniques to predict the type of traffic
- The machine learning methods include naive bayes, K-nearest neighbours, artificial neural networks, clustering, regression and support vector machines
- Use of MLP quite popular



## Background (contd.)

- Behavioural classification method
- Looks at the entire traffic received by an endpoint in the network
- The traffic patterns are then examined to identify the application running on the target host
- Different applications have different patterns
- P2P applications contact a number of peers using the same port for each host,
- a Web server is contacted by clients with multiple parallel connections.





## Background (contd.)

- Port based method
- According to some articles, the port based method might not provide very reliable results in the near future since the services of the internet are growing and the new applications may not have a well known port number assigned by IANA.
- However, for the applications that do have reserved port numbers, this method would provide 100% accurate results
- In some cases it may be important to only classify some applications, and not all. In such circumstances, this method would provide very good results.

# DESIGN AND METHODOLOGY





# Design and Methodology

- Language used is Python
- Tool for capturing live packets is Windump
- Command line packet sniffer which captures TCP/IP packets sent or received over a network
- We use this to write the captured data to a pcap file



# Design and Methodology (Contd.)

## Python Libraries

### i) PyShark

- Parsing pcap file
- Uses wireshark dissectors
- Easily get the source and destination port of packets

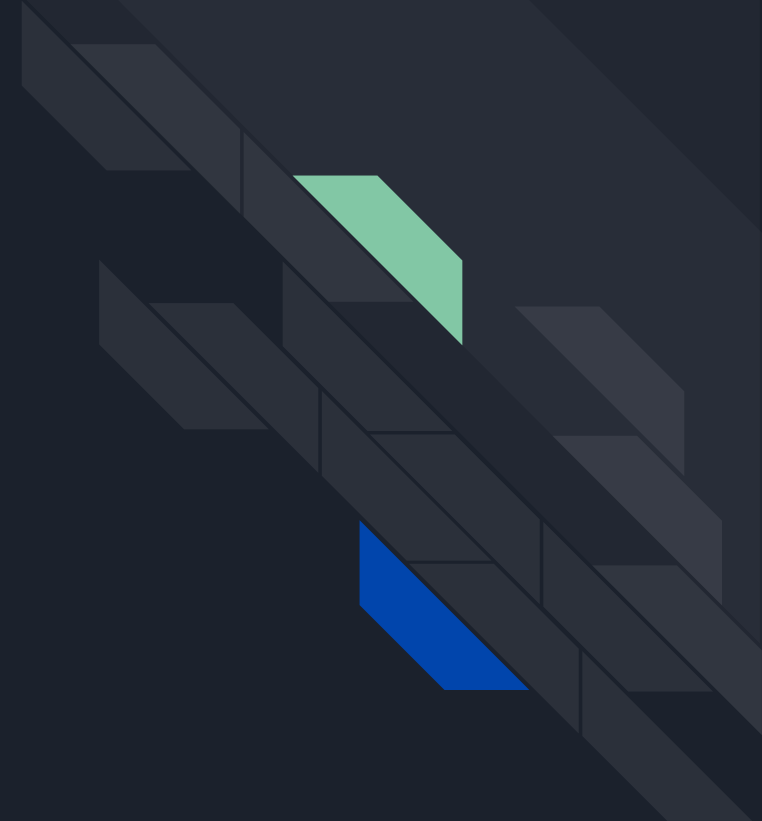
### ii) PyQt5

- UI Development

### iii) Matplotlib and Numpy

- Matplotlib for data visualization including tables and graphs
- Numpy for the mathematical calculations

# TESTING AND EVALUATION



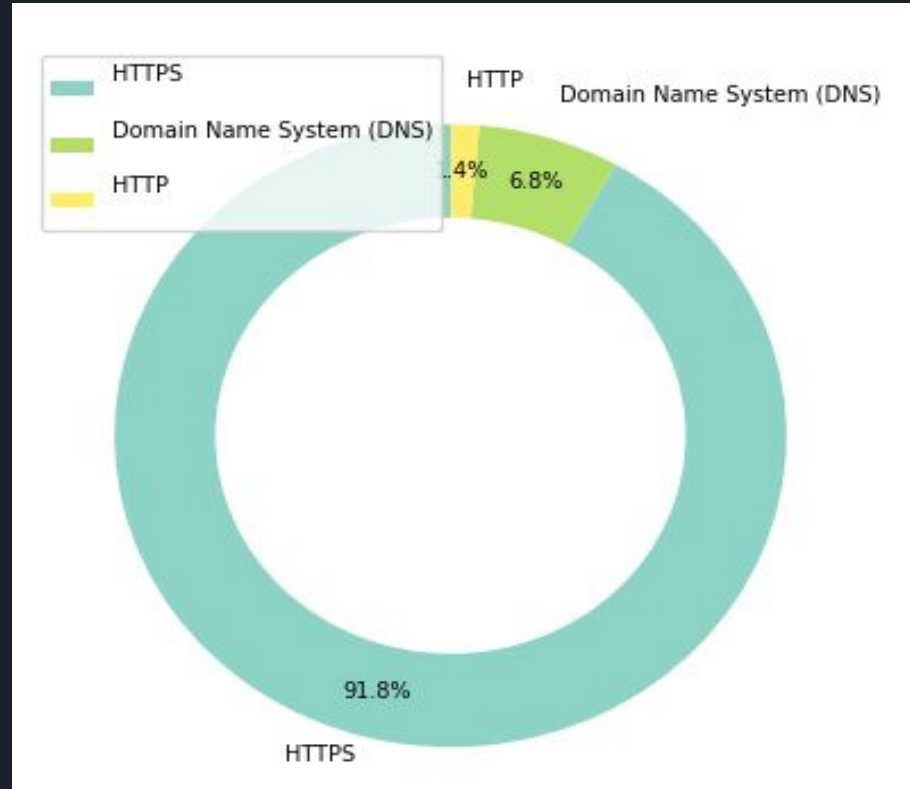


# Testing and Evaluation

- Displayed results in 4 different formats
  - a table
  - a pie chart
  - a bar graph
  - a percentage bar graph.
- Captured data for around 10-15 seconds results displayed ahead

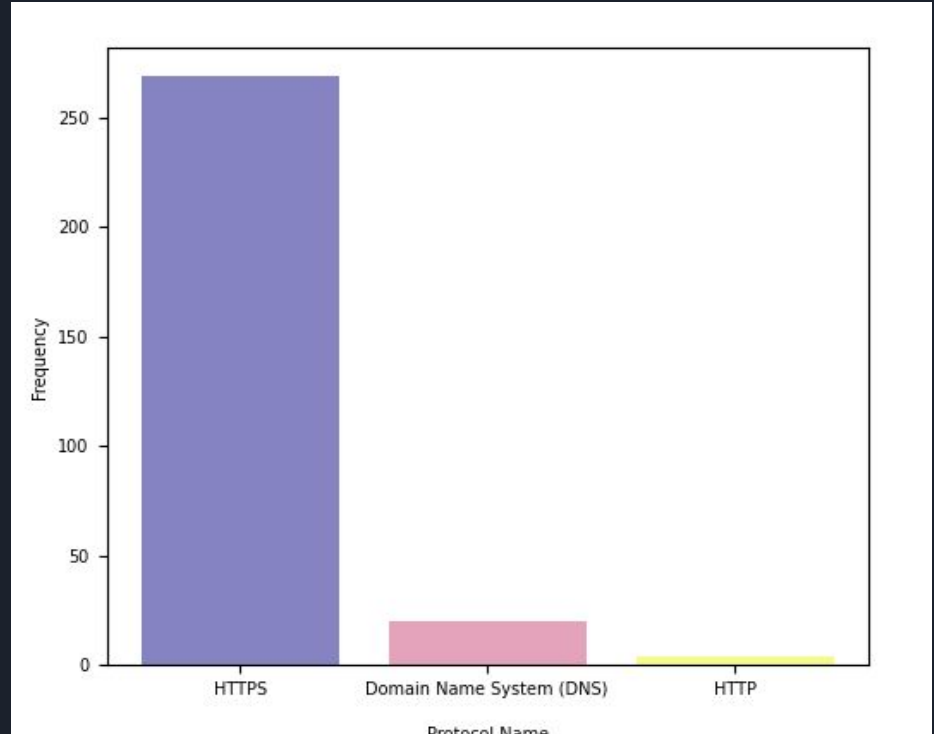
# Testing and Evaluation (Contd.)

## Pie Chart



# Testing and Evaluation (Contd.)

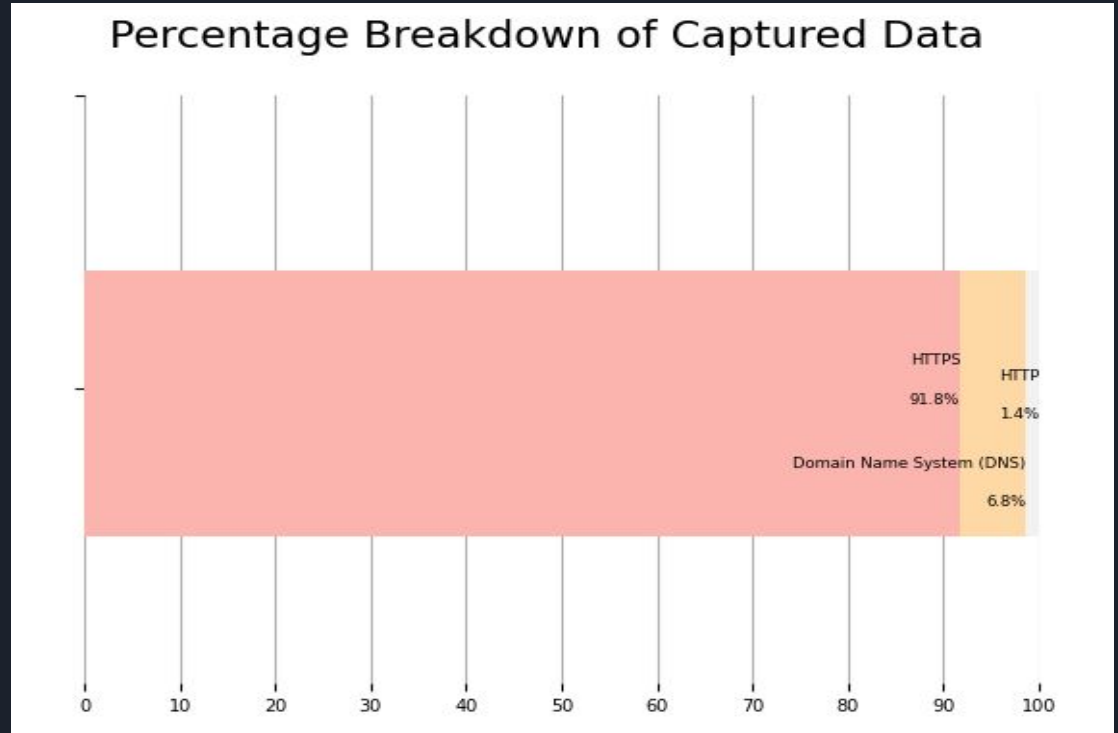
## Bar Graph





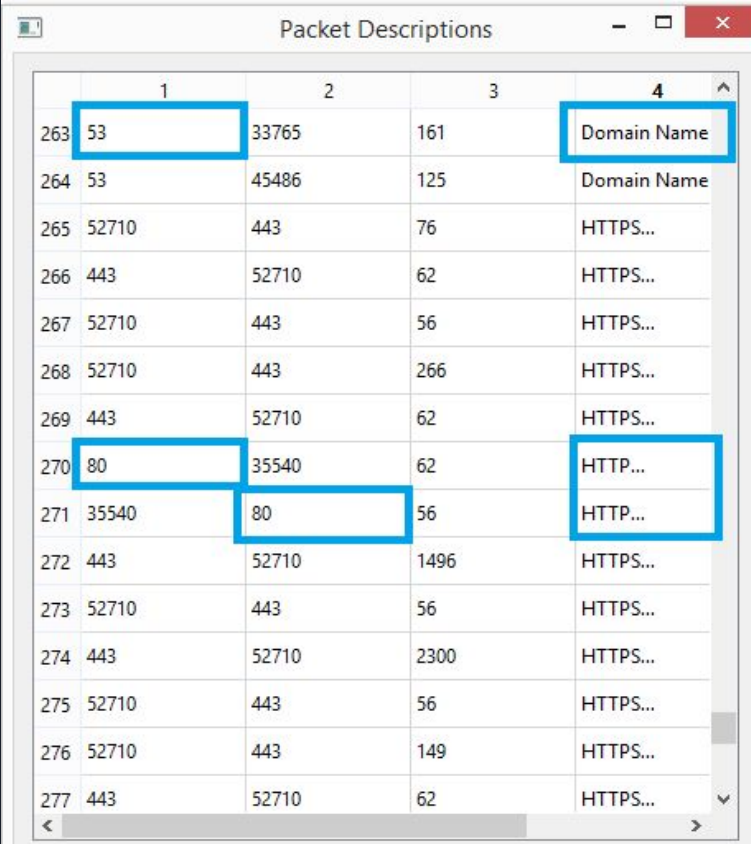
# Testing and Evaluation (Contd.)

## Percentage Breakdown



# Testing and Evaluation (Contd.)

- To check that our results were correct, we analyzed the table which was created by our tool, and checked the protocol against the port number. All the packets were correctly identified.



	1	2	3	4
263	53	33765	161	Domain Name
264	53	45486	125	Domain Name
265	52710	443	76	HTTPS...
266	443	52710	62	HTTPS...
267	52710	443	56	HTTPS...
268	52710	443	266	HTTPS...
269	443	52710	62	HTTPS...
270	80	35540	62	HTTP...
271	35540	80	56	HTTP...
272	443	52710	1496	HTTPS...
273	52710	443	56	HTTPS...
274	443	52710	2300	HTTPS...
275	52710	443	56	HTTPS...
276	52710	443	149	HTTPS...
277	443	52710	62	HTTPS...



## Testing and Evaluation (Contd.)

- We tested our tool on 4 different PCs, with different time intervals for capturing packets each time and evaluated the results. The protocols were identified correctly each time.
- Due to the limited time and resources, we were not able to test it as extensively
- Accuracy
  - This approach can correctly classify the protocols which use fixed port numbers
  - However, the ones that do not, are not identified by this method

# CONCLUSION





# Conclusion

Presenting the Network Traffic Classification based on Port Numbers identification

Different techniques and advanced classification models have been introduced in the field of networks to classify the real-time, interactive, and bulk traffic

Simple and efficient way to classify traffic is port number based

Gives 100% accuracy for those applications which used fixed port numbers

THANK YOU!

