

*Term Project*

# PCAP-UDP Parser

*By Group*

Martin Sárkány - 410016

Balaji Kommuru - 459208

Mohammad Akhtar - 459199

# PCAP-UDP Parser

## Aim

To parse the PCAP file and parse the UDP datagrams in the file

## Scope of the work

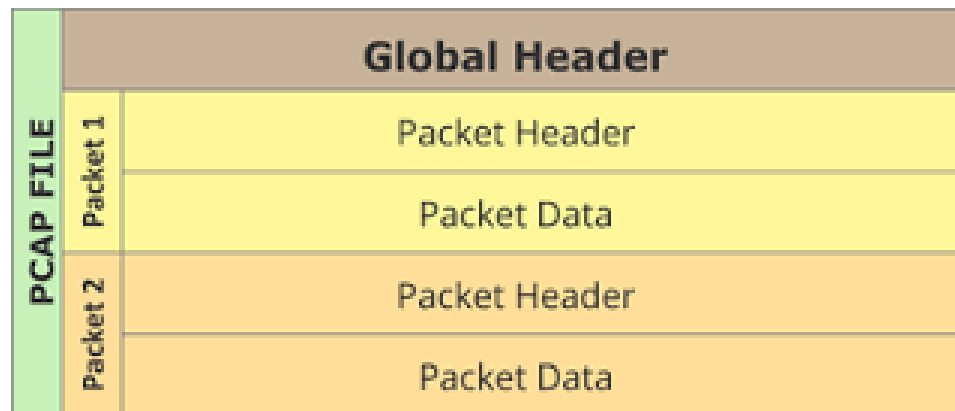
- Parsing the PCAP file
- Processing the UDP datagrams
- Producing the statistics
  - Total number of UDP datagram
  - Minimum/Maximum data size of UDP
  - No of corrupt data UDP

## Limitations

- IPv4 packets only

# PCAP

- PCAP (**P**acket **C**apture) is a basic format for capturing the network traffic
- With extension .pcap
- Structured pcap file:



# PCAP Global header format

- Size of 24 bytes.
- The first 4 bytes **d4 c3 b2 a1** , the magic number
- GMT time zone offset , accuracy of the timestamps in the capture.
- Maximum length of the captured packets (data) in bytes.
- The last 4 bytes in the global header specify the Link-Layer Header Type. (i.e. Ethernet, PPPoE, Frame Relay etc,.

# Header format

- 16 bytes
- The timestamp in Seconds. This is the number of seconds since the start of 1970.
- The microseconds part of the time at which the packet was captured.
- The size of the frame in bytes.

# Parameters captured

- Capture time
- Accuracy in microseconds
- Source IP
- Destination IP
- Source port
- Destination port
- Data size

# Design and Implementation

- Coding is done in C
- Structures used:
  - parser\_t – lists of frames, packets and datagrams
  - frame\_t – structure containing frame data
  - packet\_t – structure containing packet data
  - datagram\_t – structure containing datagram data

# Design and Implementation (cont.)

- Processing file is done in 3 steps
- Main functions:
  - int parse(parser\_t\* parser, char\* filename)
    - reads file and loads ethernet frames into list
  - int process\_frames(frame\_t\* frame\_list, packet\_t\*\* packet\_list\_p)
    - extracts IPv4 packets from frame\_list and stores them in packet\_list\_p



# Design and Implementation (cont.)

- `int process_packets(packet_t* packet_list, datagram_t** datagram_list_p)`
  - extracts UDP datagrams from `packet_list` and stores them in `datagram_list_p`

# Testing

- Pcap files with
  - Various types of packets
  - Small to big sizes of files taken from public places
  - No UDP packets
- Fuzzing:
  - strings of different formats including file name
  - Bogus file with .pcap extension

# Contribution of members

- Martin Sárkány -
  - Core structure design
  - Coding
- Balaji Kommuru-
  - UDP checksum code (under debug)
  - Coordination , testing and documentation
- Mohammad Akhtar
  - Coding for statistics, testing and debugging
  - Documentation

# References

- Wikipedia
- Github
- <https://www.elvidence.com.au/understanding-time-stamps-in-packet-capture-data-pcap-files/>