Volume 6, Issue 9, September 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Efficient PCAP Extraction Tool

Sheena*, Krishan Kumar, Gulshan Kumar
SBSSTC, Ferozepur, Punjab,
India

Abstract— Due to exponential growth of the Internet services like online shopping, Internet banking, social communication, etc. a large amount of network data is being exchanged. In order to analyze the network data, it is captured in some format and stored on disk. The most common and widely used method to capture the network data is PCAP i.e. packet capture. There are a number of tools are being developed for extracting information from captured files, but the available tools have some common drawbacks, For this reason we have developed a system i.e. Network Traffic Migration System (NTMS) that removes some of the drawbacks. So, this paper represents the NTMS that can extract the selective information based on different filters, export the data in some format like CSV or text file and also have the ability to merge different headers and their fields in a single file.

Keywords— Network; Extraction; PCAP; IP;TCP; ICMP; Connection Records; Netflow

I. INTRODUCTION

The PCAP file contains protocols like HTTP, Ethernet, TCP, IPv4, IPv6, ICMP, etc. But, PCAP file is not directly used by data mining or machine learning techniques for further analysis. Therefore, there is a requirement to extract information from the PCAP file so that the extracted information can be directly used for the data mining and machine learning techniques. There are a number of tools available that can extract the information from the captured file. Most commonly used tools to extract the PCAP format file directly is Wireshark, TCP extract, TCP dump, Pick-Packet, Network-miner, Choasreader, etc. The most of the tools having some common limitations. Firstly, the available tools extracts all the information based on different header protocols collectively, therefore it requires more storage space and more processing time. Secondly, these tools cannot extract information based on individual header protocols from the PCAP. Thirdly, most of the tools are not able to export only the selected data output in CSV or text file. Lastly, one of the main limitation is that, these tools are not able to merge the different header protocols and their fields in the single file.

In this work, we propose the **Network Traffic Migration System(NTMS)** to address the specified limitations. The proposed work provides the information based on different types of protocol headers. The information in NTMS can be filtered in two different ways: basic filter and traffic filter. The first type of filter is a basic level filter which is based on network parameters like source address, destination address, source port, destination port, etc. The different types of basic filters are IPv4, IPv6, TCP, Ethernet, ICMP. And the second type of filter is the traffic level filter which is based on different traffic parameters like counting the number of packets based on the same source to same destination port, protocol type, service flags, etc. In addition to the filtering of network traffic, NTMS provides the information based on network statistics like the connection information, flow information, count packets based on different protocols.so that the user can extract the individual information based on specified protocols depending upon their requirements. The NTMS provides the facility to export the selected information in the form of CSV or text file. One of the major contributions of NTMS is to merge the different headers protocols and their different header fields in a single file. The result of existing work compared with the NTMS shows that, the information extracted from this system is more usable and reliable for further analysis by machine learning and data mining techniques.

The rest of the paper is organized in the following way: *Section 2.* Describes the related work of other packet capture extraction tools. *Section 3* , represents the research gaps founded and which gap is being resolved by the proposed tool. *Section 4,* Describes the architecture and design of proposed tool. *Section 5,* includes process of different filters, *Section 6,* results and conclusion *Section 7,* future scope and *Section 8,* include the References.

II. RELATED WORK

Data capturing is an important phase for analyzing the data. The captured data are being used further for preprocessing or storage purposes. The most common and widely used capture format is a packet capture (PCAP). To extract the required information from the packet capture requires packet capture extraction tools. There are many packet capture extraction tools are available. Some of the tools studied, described below:

TCP dump is a most powerful and widely used command-line packets sniffer or package analyzer tool which is used to capture or filter TCP/IP packets [2]. It provides insight into the traffic activity on a network. It can create csv file, but with the help of Wireshark only.

Wireshark is a Packet analyzer, tries to display the packet data as detailed as possible [4]. A favorite tool of every network security hacker is Wireshark. It used the libpcap library behind the scenes to capture packets off the network.

The libpcap is the most basic library and most widely used for packet capturing. Almost every network security tool which requires packet capturing is based on libpacp. It can display data based on different header fields. It can import pcap format file directly, but cannot create csv file based on different header fields separately.They took pick-packet as an example of network monitoring tool and tried to implement text string searching on HTTP compressed data on fly [3]. But the basic condition for filtering the data on fly is that packets must receive in order.

They compared the various file extraction tools like tcp flow, wireshark, choas reader, tshark extractor, network minor[11]. They worked on the host based forensic that can extract the protocols like HTTP, FTP< TFTP, SMP. They worked on separate extraction of file.

III. RESEARCH GAPS

- The available tools extracts all the information based on different header protocols collectively, therefore it requires more storage space and more processing time.
- These tools cannot extract information based on individual header protocols from the PCAP.
- Most of the tools are not able to export only the selected data output in CSV or text file.
- One of the main limitation is that, these tools are not able to merge the different header protocols and their fields in the single file.

The summary of existing tools is given in Table I.

Table I Summary of existing tools

Tool Name:	**Author:**	**Year:**	**Type:**	**Benefits:**	**Limitation:**
Wireshark	Gerald Combs	1998	Packet sniffer and packet analyzer	It interprets an extremely large number of protocols and easily extracts files	It can not export the selective data.
Pick-Packet monitoring tool	Barjesh Pande	2002	Packet monitoring	Configuration file generator for network layer and application layer, suppports the real time searching form text string in application and packet content	Basic condition of filtering the data is that the packets must receive in order
Chaos Reader	Brendan Gregg	2004	Packet analyzer	Effective at generating connection information	Extraction of incomplete files
TCP Extract	Padres & Harbour	2005		Extracts files with their original names	Extract packet capture but not in a reliable and predictable fashion, only support HTTP protocol
TCP Flow	Soderberg	2010	Packet sniffer and packet analyzer	Able to parse a network packet capture, handling protocol headers, fragments, and out of order packet delivery	Capable of processing only TCP packets
Network Miner	Davidoff & Ham	2012	Packet sniffer and packet analyzer	Supports file extraction from several protocols, provides large amounts of data for further analysis and has a graphical interface.	Takes longest time to run

In this paper, gaps like extraction of data based on selective information, exporting the selected output in csv, text file, fetching connection records, flow information directly from pcap has been removed.

IV. NETWORK TRAFFIC MIGRATION TOOL

A. About NTMT:

Like other tools, this system is also used for the extraction of captured data. It helps in the extraction of selective data based on different filters like TCP, IP, Ethernet, etc. in more detail and also exports selective data in CSV or text file format. It is very easy to use. Whereas other packet capture extraction tools, extract the whole packet capture information at once. The architecture of the proposed system consists of three components, is shown in figure 1.

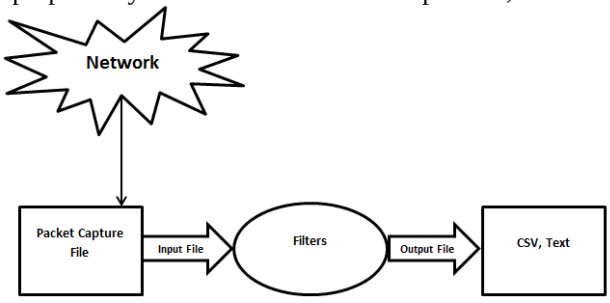

Figure 1: Architecture of NTMT.

- Input file: The input file format is PCAP i.e. packet capture format.
- Filters: Different types of filters are used to fetch the information from packet captures namely Ethernet, IPv4, IPv6, ICMP, TCP, connection records, count packets, flow information.
- Output file: It supports the CSV, Text file as the output file format.

B. Design of NTMS:

The NTMS design consists of methodologies used, filter levels, filter types and its process. These components are described in detail in further subsections. The detailed design view of the proposed tool is shown in figure 2.

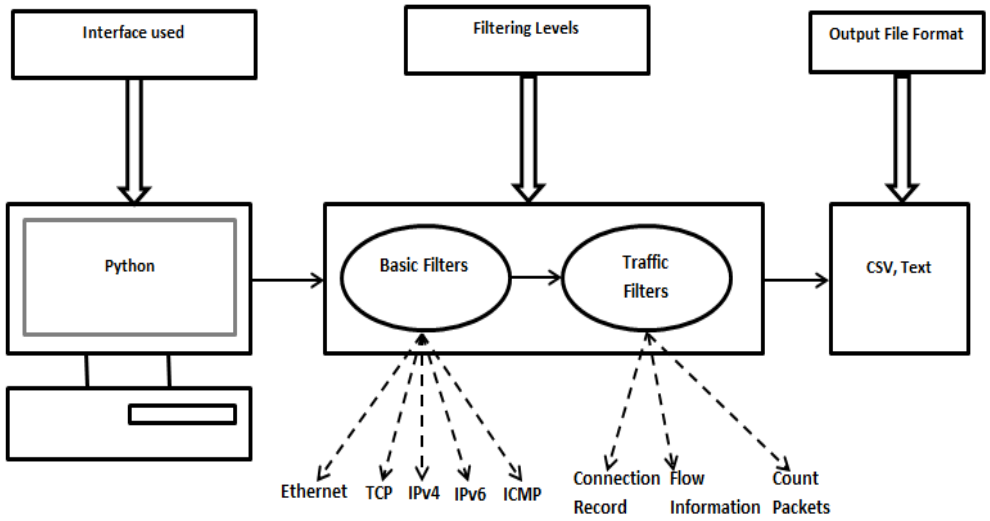

Figure 2: Design of NTMT.

- ✓ The interface used: We used Python language for our work. Python is an open source scripting language developed by Guido van Rossum in the early 1990s. Python is very simple, easy to install and use. It is easy to install packages just by typing pip install 'package name'. It can import any package by just typing Import 'package name'. Very well written API. Python programs take much less time to develop.
- ✓ Filtering levels: There are two types of filtering levels, i.e. basic level and traffic level filtering. The **Basic level filter** is the filter that is based on different network parameters like source address, destination address, source port, destination port and much more. The different basic filters are IPv4 filter, IPv6 filter, Ethernet filter, ICMP filter, TCP filter. The **Traffic level filter** is the filter that is based on different traffic parameters like counting the number of packets based on the same source to the destination port, protocol type, service, flag, flow information, etc.
- ✓ Export format: Tool supports two types of output format, i.e. CSV and Text file format. **CSV file format** is a comma separated values file, which allows data to be saved in a table structured format. **Text file format** is a kind of computer file that is structured as a sequence of lines of electronic text.

V. FILTERING PROCESS

The purposed tool is helpful in the extraction of packets based on different filters. The various filters with their extracted bits information detail is described below:

a) *IPv4 Filter:* Internet protocol, which is used to get the information about IP packet, which consists of various fields. The extracted field description for this filter is described below:

Version: This signifies the current version of IP protocol being used whether version 4 or 6. It occupies 4 bits or 6 bits respectively.

IP Header Length: It is of 4 bits. It specify the length of the header.

Type of Service (TOS): It represents the bits to know that which type of service is being used. It occupies 8 bits. The TOS bit description is described in Table II.

Table III TOS' BIT DESCRIPTION

Bit:	Description:
0x00	Default
0x10	low delay
0x08	high throughput
0x04	high reliability
0x02	low monetary cost
0x02	ECN-capable transport
0x01	congestion experienced

Total Length: It signifies the total length in bytes. It is about 16 bits.
Identification: Unique ID is being used for identification. It is about 16 bits.
Flags: It comprised of 3 bits. The Flag field bit description, is shown in Table III.

Table IIIII Ip Flag' Bit Description

Flag bit:	Description:
1st Bit	Reserved
2nd Bit	Don't fragment
3rd Bit	More fragments

If DF = 1, it is set and IP datagram is never fragmented
If MF = 1, it is set and represents a fragmented IP datagram that has more fragments after it.
Fragments Offset: It is about 13 bits. It contains the offset from the beginning of IP datagram.
Time to Live (TTL): It occupies 8 bits. It counts the number of hops that IP datagram go through. The value of the TTL field is shown in Table III.

Table IVV Ttl Bit Description

TTL seconds:	Description:
64	Default TTL
255	Max TTL

Protocol: This field is used to indicate that which protocol to hand over the data to. It is of 8 bits. There a number of protocols, bits representation of Protocol field is given in Table V.

Table V Protocol Description

Protocol Number:	Description:
0	Dummy
1	ICMP
2	IGMP
3	gateway-gateway protocol
4	IP in IP
5	ST datagram mode
6	TCP
17	UDP
41	IPv6
46	Reservation Protocol
58	ICMP for IPv6
255	Raw IP packets

Header Checksum: It is of 16 bits. This field is used to calculate the value of header covering all the fields when sending data from source to destination and stores the result in the header. The value is again calculated when it reaches the destination to check whether value is still same or not. If value is same then data is not corrupted, else corrupted.
Source and Destination IP address: It uses 32 bit to store the address of source and destination.

b) *TCP Filter:* TCP stands for Transmission Control Protocol. It is sent as internet datagram. It consists of several information fields. The extracted fields information is described below:

Source Port: It is 16 bit port number used to signify the client port number who sent the request to the server.
Destination Port: It is also 16 bit port number, well known for client request.
Sequence number: This field is used to keep the track of data it sent to recover the damage data.
Acknowledgement number: This field is used to indicate the sender that data has been received successfully.
Flags: There are number of flags used to indicate which bit is set high. This field can be combination of more than one flag. The different flags bits description is shown in Table VI.

Table VI Tcp Flag'bit Description

Flag bit:	Description:
0x01	end of data
0x02	synchronize sequence

	numbers
0x04	reset connection
0x08	Push
0x10	acknowledgment number set
0x20	urgent pointer set
0x40	ECN echo, RFC 3168
0x80	congestion window reduced

Window: It is an option to increase the size of receiving window allowed in TCP above its max. value i.e. 65535 bytes.
Urgent pointer: Used to indicate the priority data transfer. If set to 1 then there's priority is invoked else 0.

c) *ETHERNET Filter:* Ethernet packet transports an Ethernet frame as its payload. The extracted fields of this filter is described below:

Source and Destination MAC address: Media Access Control also known as physical address. It is 48 bit address, unique identifier used for the communication.
Ethernet Type: This field is used to check which protocol is in the payload of Ethernet frame. The some most useful bits of this field and their description described in following Table VII.

Table VII Ethernet Type Description

Eth Bit:	**Extracted Bits as:**	**Eth Type:**
0x0800	2048	IPv4
0x0806	2054	ARP
0x86DD	34525	IPv6

d) *ICMP Filter:* Internet Control Message Protocol is one of the major protocol of IP suite. ICMP is used to relay query messages and to send error message. The extracted fields of ICMP and their description is as follow:

ICMP Type: This field is used to identify the type of message. The list of type bits and its corresponding names is given in Table VIII.

Table VIII Icmp Type Description

Type:	Name:
0	Echo Reply
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench
5	Redirect
6	Alternate Host Address
7	Unassigned
8	Echo
9	Router Advertisement
10	Router Selection
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request [RFC950]
18	Address Mask Reply
19	Reserved (for Security)
20-29	Reserved (for

	Robustness Experiment)
30	Traceroute
31	Datagram Conversion Error
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37	Domain Name Request
38	Domain Name Reply
39	SKIP
40	Photuris
41-255	Reserved

ICMP Code: This field is used to provide further information about the associated type field. The different code bits and its description are mentioned in Table IX.

Table IX Ethernet Code Description

Code:	**Name:**
0	No code/Net Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation Needed and Don't Fragment was Set
5	Source Route Failed
6	Destination Network Unknown
7	Destination Host Unknown
8	Source Host Isolated
9	Communication with Destination Network is Administratively Prohibited
10	Communication with Destination Host is Administratively Prohibited
11	Destination Network Unreachable for Type of Service
12	Destination Host Unreachable for Type of Service
13	Communication Administratively Prohibited
14	Host Precedence

	Violation
15	Precedence cutoff in effect

ICMP Checksum: This field provides the method for determining the message integrity.
Identifier: Unique ID is being used for identification of messages.
Sequence number: This field is used to keep the track of data it sent to recover the damage data.

e) *IPv6 Filter:* It is also known as next generation protocol. The address of IPv6 is 128-bit. It has built in authentication and privacy support. The important technologies of this version improve the IP protocol. The various header fields of this protocol are described below:

Version: Represents version of IP.
Traffic Class: It replaces the TOS field of IPv4. It specifies the quality of service.
Flow Label: It is used to maintain the sequential flow of packets. The label helps the router to identify that particular packet belongs to a particular flow. The hosts are required to set the flow to zero when originating a packet.
Payload Length: This field is used to tell the router how much the information does a particular packet contain.
Next Header: This field is used to indicate the type of extension header or upper layer PDU (protocol data unit). The list of extension header is given in Table X.

Table X Extension Header Description

Order	**Header Type**	**Next Header Code**
1	Basic IPv6 Header	-
2	Hop-by-Hop Options	0
3	Destination Options (with Routing Options)	60
4	Routing Header	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulation Security Payload Header	50
8	Destination Options	60
9	Mobility Header	135
	No next header	59
Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58

Hop Limit: It is used to stop the packet. The value of this field is decremented by 1 as it passes by a hop. Packet is discarded when the value reached at 0(zero).
Source and Destination address: It is 128 bit each. His field indicates the address of the originator and recipient of the packet respectively.

f) *NETFLOWS:* It is a network protocol used to collect IP traffic information and monitoring the traffic. By analyzing the flow of data volume, picture of network traffic can be built.

The special fields in the netflow are timestamp and count packets.

Timestamp: It is a sequence of encoded information to identify when a certain event occurred , usually by giving time or date and sometimes small fraction of seconds.

Count Packets: It counts the total number of packets based on same source port to same destination port

g) *Connection Records:* Connection is basically the connectivity between the sender and the receiver. We can record the connection to configure fields given in Table XI.

Table XI Connection Record Field Description

Feature name	*Description*
duration	length (number of seconds) of the connection
protocol_type	type of the protocol, e.g. tcp, udp, etc.
service	network service on the destination, e.g., http, telnet, etc.
src_bytes	number of data bytes from source to destination
dst_bytes	number of data bytes from destination to source
flag	normal or error status of the connection
urgent	number of urgent packets

h) *Count Packet*: It counts the packets based on different protocols separately.

i) *Merge Different Protocols*: This feature helps in merging of different header protocols and their fields in a single file.

VI. RESULT AND DISCUSSIONS

This section describes the comparison of existing packet capture extraction tools with the proposed one. The comparison is shown in Table XII.

Table XII Comparison Of Other Tools With Proposed Tool

S. No.	**Features:**	**ChaosReader**	**TCP Flow**	**TCP Extract**	**Wireshark**	**Network Miner**	**Pick-Packet monitoring tool**	**Proposed Tool**
1	The Language used	Perl	C		C, C++	C#	Java	**Python**
2	Platforms support	Solaris 9, Redhat 9, Linux, and Windows 98	Linux		Linux, Windows	Windows, Linux		**Windows, Linux**
3	Capture Live packet	No	Yes	No	Yes	Yes		**No**
4	Import pcap file format directly	Yes	Yes	Yes	Yes	Yes	Yes	**Yes**
5	Filtering based on Different headers separately	No	No	No	Yes	No	Yes	**Yes**
6	Protocols supported	TCP/UDP, IP, ICMP	TCP	Only HTTP	IP, Eternet, ICMP, TCP	FTP, TFTP, HTTP, SMB and SMTP	TELNET, SMTP, HTTP, FTP etc.	**IPv4, IPv6, Ethernet, ICMP, TCP**
7	Connection Records	Yes	Yes	No	No	No	No	**Yes**
8	Flow Information	No	No	No	No	No	No	**Yes**
9	Export data only for selective filter	No	No		No	No		**Yes**
10	Easy to use	Yes	Yes	Yes	Yes	Yes	Yes	**Yes**

11	open source	Yes	Yes	Yes	Yes	Yes	No	**No**
12	Handles large amount of data	Yes	Yes	Yes	No	No	No	**Yes**
13	Count Packets based on different headers	No	No	No	No	No	No	**Yes**
14	Export file format	HTML	dump		CSV, XML, etc.		Pcap	**CSV, Text**
15	Merge different protocols	No	No	No	No	No	No	**Yes**

VII. CONCLUSIONS

In the proposed work, we created the individual filters like IPv4, IPv6, Ethernet, ICMP, TCP. The data can be exported in formats like CSV, Text file and also the NTMS can merge different header protocols and their fields in a single file. In order to extend the proposed work, more individual filters like HTTP, ARP, etc. will be inserted, and also more data export formats like arff, html.

REFERENCES

[1] Derrac, J., et al. "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework." (2015).

[2] Jacobson, Van, Craig Leres, and Steven McCanne. "Tcpdump/libpcap." (1987).

[3] Archita Dad, A. S. (2013). Performance improvement of a packet lter by ltering compressed packet. International Journal of Advanced Research in Computer and Communication Engineering, 2.

[4] Banerjee, U., Vashishtha, A., and Saxena, M. (2010). Evaluation of the capabilities of wireshark as a tool for intrusion detection. International Journal of Computer Applications,6(7).

[5] Bouckaert, R. R., Frank, E., Hall, M. A., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2010). Weka|experiences with a java open-source project. The Journal of Machine Learning Research, 11:2533{2541.

[6] G. Holmes; A. Donkin; I.H. Witten (1994). "Weka: A machine learning workbench" (PDF). *Proc Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia*. Retrieved 2007-06-25.. (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[7] Derrac, J., GARCia, S., Sanchez, L., and Herrera, F. (2015). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework.

[8] Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., and Tan, P.-N. (2002). Data mining for network intrusion detection. In Proc. NSF Workshop on Next Generation Data Mining, pages 21{30.

[9] Jacobson, V., Leres, C., and McCanne, S. (1987). Tcpdump/libpcap.

[10] Lee, W., Stolfo, S. J., et al. (1998). Data mining approaches for intrusion detection. In Usenix security.

[11] Stephen Deck, H. K. (2015). Extracting les from network packet captures. GIAC (GCIA) Gold Certication, pages 1-25.

