TCP Packet Eavesdrop Using Wireshark

Priyavrat Chaudhary, Ajay Kumar Singh

Abstract: ***We use TCP and UDP for the communication purpose on the internet. If anyone wants to communicate over the internet, it would not be possible without TCP and UDP protocol. Here, our aim is to analyze TCP packets with the help of Wireshark. It is a free of cost open-source software that is used to analyze packets on the network. To study TCP packets the various protocol header fields have been used such as frame number, frame length, Source and destination IP, header length and window size value etc. In addition, we have also talked little bit about the HTTP packet. In this paper we have also used flow graph feature of Wireshark in order to analyze TCP in detail.***

Index Terms**: Wireshark, Packets, HTTP, Ethernet, TCP, UDP.**

I. INTRODUCTION

Wireshark is a tool that is used to analyze the packets [1]. It sniffs the packets going to and from the host over the network and displays it in human readable form. It is a tool that gives the clear picture of each and every packet. Wireshark reads and investigates data from http requests, cookies, forms etc. It receives the frames on the data link layer at the network Interface card [2]. It decodes the data inside the packet with the help of programs, giving us a clear picture of what is happening inside the cable [3] and thus makes it suitable for network administrator or an analyzer to read the data easily. It is tool filled with various features like flow graph, TCP stream and HTTP stream features, we can also measure the RTT of each packets and compare a group of packs with another group. Wireshark can also capture VLAN packets. VLAN is used today by many enterprises where different network device can connect together which are in the same domain independent of their location [4].

A. Need for Wireshark tool

Nowadays every organization has subnets within the building [5]. A device located at one part of the building would need a resource located at the other part of the same building and if any fault occurs in the process of resource sharing whole work slows down and thus it affects the turnover of the organization directly or indirectly. Hence we need a tool with the help of which we could exactly locate the glitches in the network and that's too in quick time. Hence we use a packet sniffer tool called Wireshark. Nowadays network administrator makes use of it to troubleshoot the faults in the network and also for the security purpose.

B. Importance of Sniffing a Network

Sniffing of a network is important as security point of view. A normal traffic has different patter than the abnormal traffic. Network administrator observes the patter of traffic flowing through his organization network and if he finds any suspicious behavior in the traffic, he can block that traffic for the mean while and then he will look preventive measures. So in this way it also helps in securing the hosts inside the network of an organization.

II. LITERATURE REVIEW

A. *Usage of Wireshark*

Wireshark is mostly used by network administrators, developers for the security purpose. It simplifies their work and also speed up the process of troubleshooting the network.

B. *Disadvantages of Wireshark*

Every good thing has its bad uses. Similarly, wireshark can be easily misused to illegally listening the private data of some other user in the same network.

C. *TCP Packet Frame Format*

The Transmission Control Protocol is the mostly used transport protocol that provides mechanisms to establish a reliable connection with some basic authentication, using connection states and sequence numbers [6]. Figure 1 shows TCP header format. TCP is known to send different types of data ranging from HTTP, telnet and FTP to HTTPS World Wide Web pages [7].

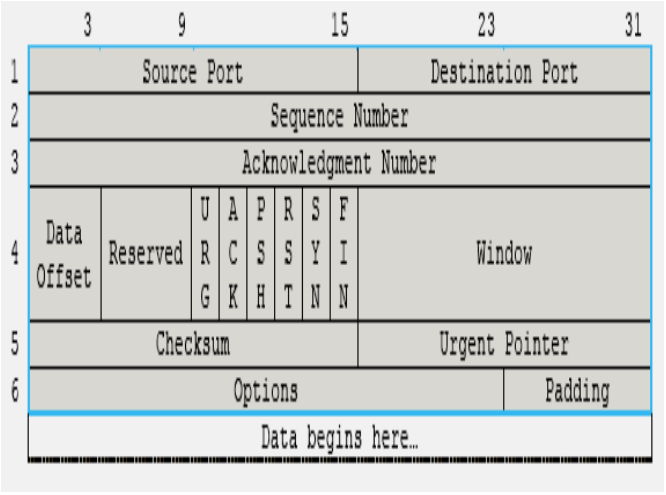

Fig. 1 TCP Header

Revised Manuscript Received on December 22, 2018.

Priyavrat Chaudhary, Meerut Institute of Engineering and Technology, Meerut, UP, India

Ajay Kumar Singh, Meerut Institute of Engineering and Technology, Meerut, UP, India.

1. Source Port – It is the port to which receiver sends data. Operation system can choose any number to identify the source port for the receiver TCP. Zero is used if there is no use of source port.
2. Destination Port – It is the port to which sender sends data. Server is assigned only well defined ports.
3. Sequence number – This is 32-bit number which provides a sequence to the TCP segments sent on the network. The packets must be organized in order towards receiver side to be able to run audio, video or text file.
4. Acknowledgement Number – It is a 32-bit number which signifies the value of the next byte that receiver is expected to send.
5.
6. Offset – It is 32-bit number which represents values in words. It signifies the total length of the header in the fragmented TCP segment.
7.
8. Reserved - 4 bits reserved for future use. This is unused and must contain binary zeroes.
9.
10. Flags – It is a 6-bit field in which each bit represents a Flag bit. These Flag bits are defined below:
    - URG – It signifies that the payload in the TCP segment contains very important data. If the value of this field is added to the sequence number we get the last byte of the urgent data.
    - ACK – If this bit is set on it signifies that TCP header contains acknowledgement value.
    - PSH – If this bit is set then the data at the receiver side is not buffered instead it is directly sent to the application layer.
    - RST - Reset. Tells the peer that the connection has been terminated.
    - SYN – If this bit is set on it represents that the client wants to establish a connection with the server.
    - FIN – It indicates that either the sender wants to end the connection.

11. Window – It is a 16-bit field. This field informs the receiver that the maximum amount of data it can send back to the sender within the TCP segment.

12. Checksum - Frame check sequence of TCP can be used to identify erroneous data [8].

10 Urgent Pointer - It signifies the importance of the payload in the TCP segment.

III. EXPLANATION OF A TCP PACKET USING WIRESHARK

In this research paper we have taken a single example of a HTTP request to a web server. The client makes a GET/HTTP/1.1 request for a host www.footytube.com and the web server send back the request HTTP/1.1 OK.

Before we start analyzing the following TCP conversation, we need to who is the send and what is the host. Here, sender's IP address is 10.11.10.13 and server is footytube whose IP address is 174.129.253.150. As we know in TCP the whole data is not sent at one go. Data is divided into segments and stored in buffers before being sent out. In this research paper we have tried to show the transmission of each packet that has been sent across the network between the client and the host with the help of Wireshark. There are in total 37 packets that have been sent across in both directions between the host and the client.

In frame 534, the client wants to make a connection. The frame length, IP header length, source and destination IP address of this frame is shown in figure 2.1.The TCP randomly select an initial sequential number. Here it is taken as zero for the sake of convenience. Note that A SYN segment cannot carry data, but it consumes one sequence number [11]**.** This frame wants to establish the connection with server since its SYN flag bit is on. This frame initially set the window size to 8192 bytes which means sender of this packet will drop any TCP segment sent back to it with the size more 8192 bytes. See figure 2 and figure 3.

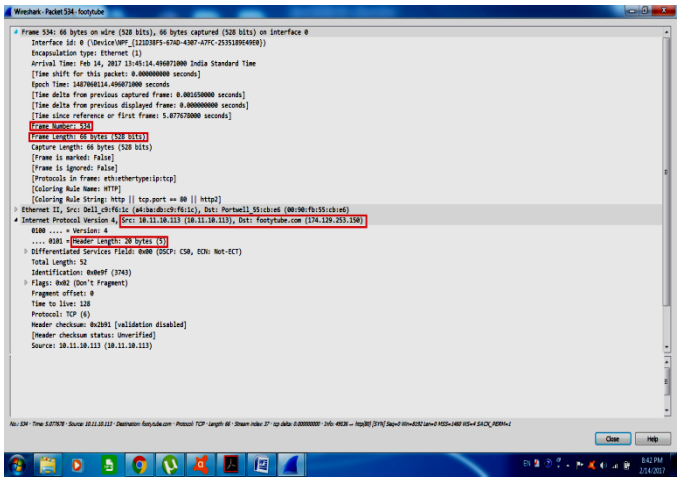

Fig. 2 Setting the window size

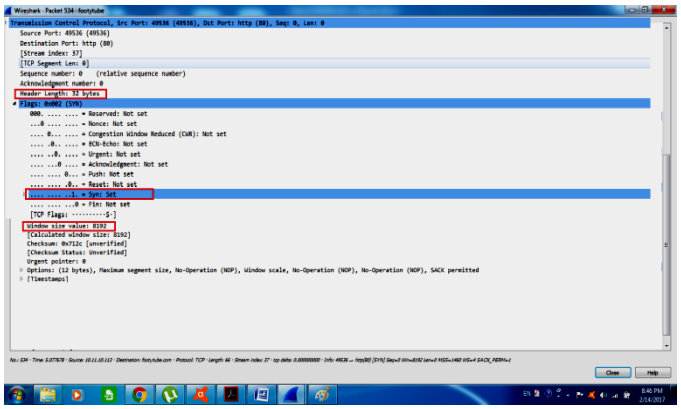

Fig. 3 Size is more it will drop the packet

Next frame is 536, it is sent from the server side. Its SYN flag bit is on which shows that server is also ready to establish a connection with the client. In addition its ACK flag bit is also on. It is an acknowledgement of the first frame that is sent by the client. The ACK number is 1 which means the server expects 1st byte from the client. See figure 4 and 5.

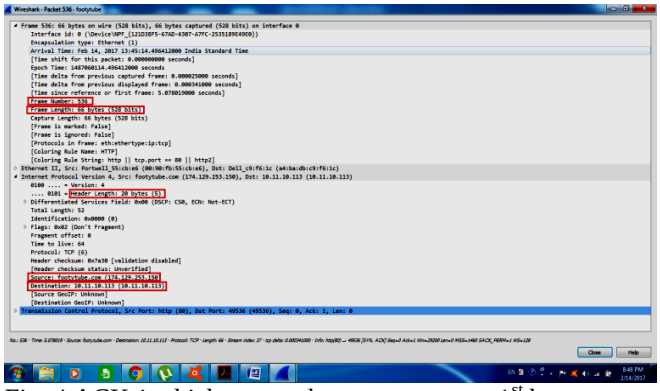

Fig. 4 ACK 1 which means the server expects 1st byte

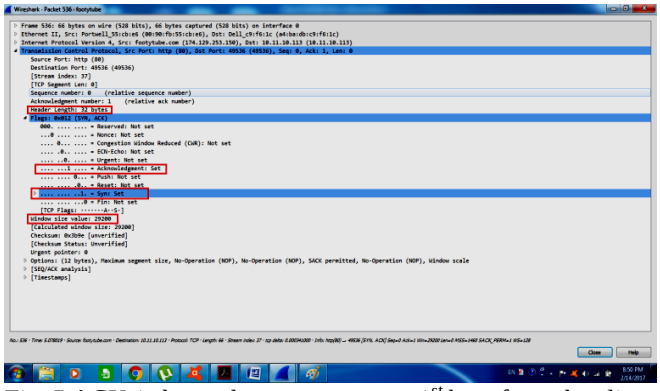

Fig. 5 ACK 1 shows the server expects 1st byte from the client

Packet 537 has been sent by the server which we can check out by seeing the source and destination IP address in the IP header field. It is an acknowledgement of the previous TCPSYN packet. See the figure 6 and 7.

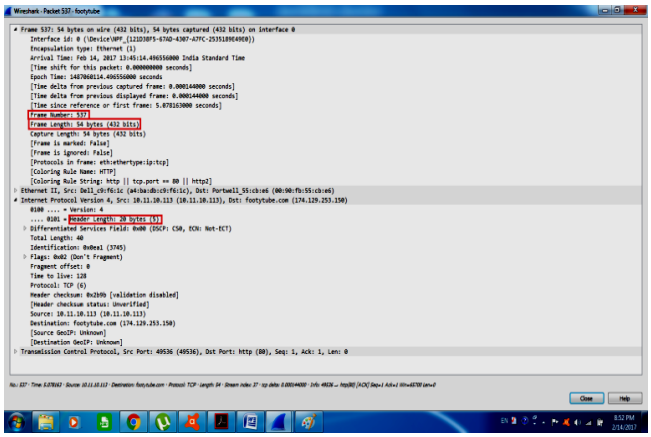

Fig. 6 Source and destination IP address

In order to understand the TCP connection we have used "Flow Graph" feature of wireshark. To open it First go to statistics menu then in the drop-down list select flow graph which will open the flow graph window.

In flow graph window change the "All packet" to "Displayed packets" in show option while in Flow type option change the "All Flows" to "TCP Flows". These two changes are marked in the red rectangular box in the figure 8. The TCP flow graph is continued in figure 9 as well in order to cover all the packets of TCP flow.

In Frame 555, the client sends HTTP request for the host footytube.com to the web server on port 80.

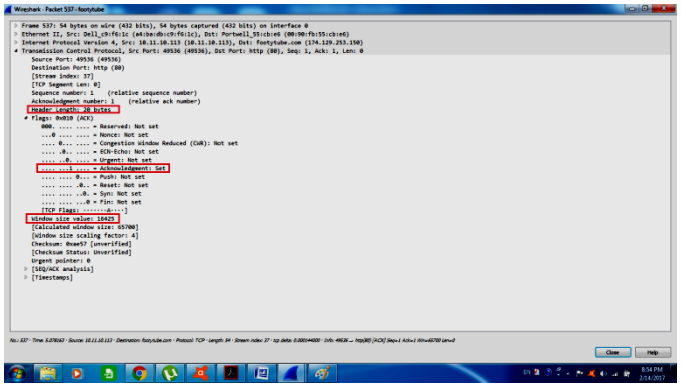

Fig. 7 Acknowledgement of the previous TCPSYN packet

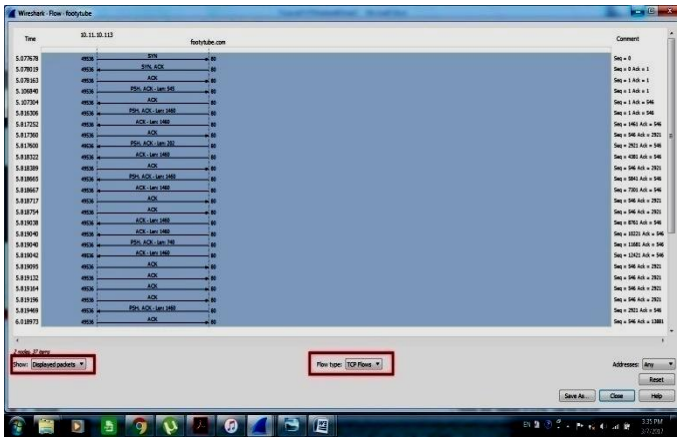

Fig.8 Change the "All packet" to "Displayed packets"

The relative sequence number is 1 and the length of the TCP packet is 545 bytes. In this TCP segment the PSH and ACK flag bit is on.

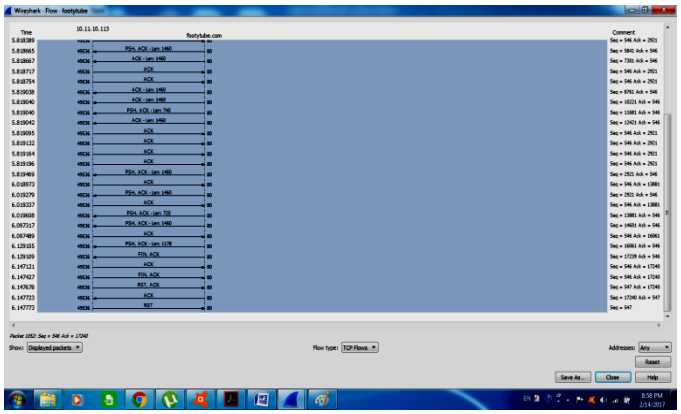

Fig. 9 Change the "All Flows" to "TCP Flows"

In Frame 556, the sever sends acknowledgement of the previous Frame i.e. frame 555. Acknowledgement field contains value 546 bytes, it means that the server expects next byte to be 546.

In frame 962, the server sends chunk of data for the first time. The total 1460 bytes has been sent by it. PSH and ACK flag bits are on. Server at port 80 sends TCP segment of 1460 length with relative sequence number 1.

ACK flag bit is on. Next expected byte in the acknowledgement field of TCP header is the same as in frame no 556. In this TCP segment PSG flag bit is set on which means the data should not be stored at the buffer of client TCP instead it should be immediately transfer to the application program at the client side.

In frame 963, Server sends another 1460 bytes of data to the client. In this frame only ACK flag bit is set on. The sequence field value is 1461.

In Frame 964, now client send an acknowledgement of the data it has received till now. The total 2960 bytes has been received by the client. In the Acknowledgement field it says 2961 this means it expects next byte to be this.

In frame 965, the total bytes on wire are 256 while the Total length field in the IP header shows value 1500. This means the total ipv4 length exceeds the packet length that is 242 bytes. This is an error in the packet. So TCP drops this packet and it does not reach the client port. See the figure 10.

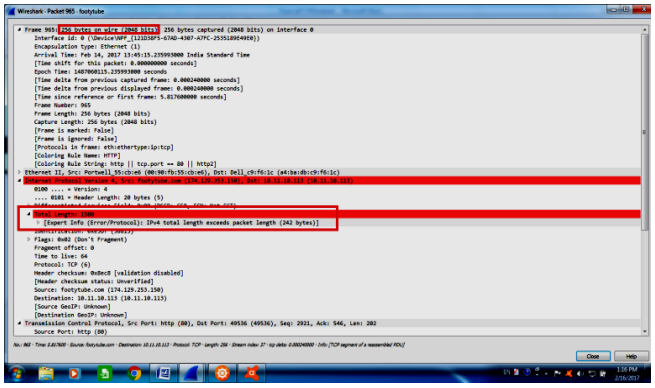

Fig. 10 Drop of packet

In Frame 966, Server sends another packet with 1460 bytes of TCP data. You can see in the info column of packet list pane, it says that the previous packet that is packet number 965 has not been received. See the marked portion in figure 11. Acknowledgement field value is same as in previous packet that is 546.

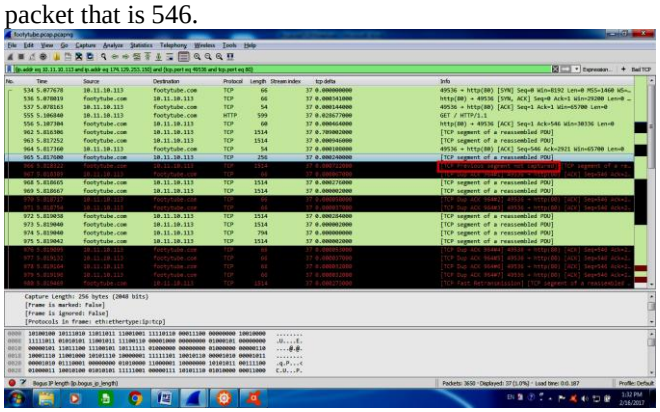

Fig. 11 Packet number 965 received

After receiving the out of order frame, client sends DUP ACK segments. As shown in figure 11, frame no. 967, 970, 971, 976, 977, 978 and 979 are TCP DUP ACKs.

Frame 980 is the fast-retransmitted frame which is marked in rectangle box in the figure 12. Finally, in frame 1042 server sends the HTTP response.

Packet 1039 is the last TCP packet sent by the client before host fulfills his HTTP request in the Frame 1042.

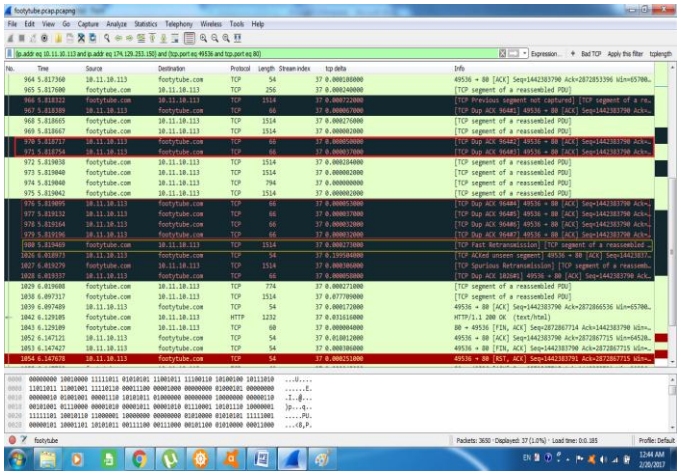

Fig. 12 TCP DUP ACKs

The sequence number in 1039 frame is the same as in previous TCP segments being sent from the client side on the other sequence number field of TCP segments sent from the server side is increasing in every packet unless it is an ACK packet with no payload.

Frame 1042 is a HTTP response to the HTTP request that was sent by the client in the frame number 555. Status 200 signifies the request has been successful. Now Host has sent all the data requested by the client. See figure 13.

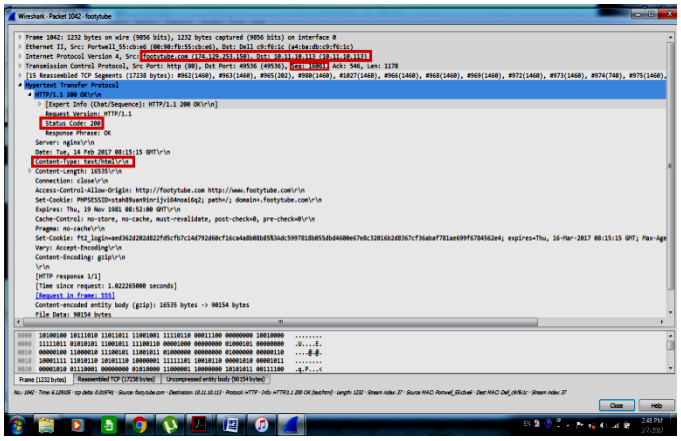

Fig. 13 HTTP Response Packet

A. *Tear-down*

Frame 1043 is sent by the host footytube.com with FIN bit set on. Now this means the host wants to terminate the connection since it has sent all the requested data. In Frame 1052 client acknowledges it.

In TCP segment 1053 packet has not been received by server so client sends RST and ACK for segment 1054 and the sequence number is incremented by 1 [9]. So the sequence number becomes 547. Frame 1055 is an acknowledged TCP segment that has been sent by Host footytube.com to the client. Segment 1056 is used to terminate the connection with the server so WIN and LEN becomes zero. Figure 14 shows frames that have been used to tear down the connection, marked within the red rectangular box. Any service attacks [10] can be taken care by wireshark [11-12]

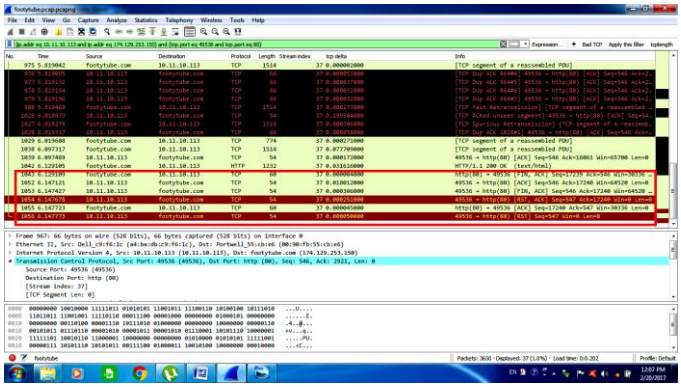

Fig. 14 Tear-down Connection

IV. CONCLUSION

In this research paper, we have deeply understood the flow of packets on transport layer using TCP protocol. Wireshark identifies each and every packet in details. It can be helpful in troubleshooting the network for example it can identify denial of service attacks. For broader view of how TCP flow is taking place we have successfully used Flow graph and TCP stream features. We have explained this by taking a simple example.

ACKNOWLEDGEMENT

We are thankful to our management and chairman sir for his kind support and motivation.

REFERENCES

1. C. Sanders, “Practical Analysis Using Wireshark to Solve Real-World Network Problems”, 2nd ed., W. Pollock, USA, 2007.
2. Anshul Kumar, “A Research Study on Packet Sniffing Tool TCPDUMP”, International Journal of Communication and Computer Technologies, vol. 01, Issue: 06, no.49, ISSN: 2278-9723, pp. 172-174, July, 2013.
3. Bartlett E. “Cable Communications Technology”, McGraw Hill, USA, 2005.
4. Minlan Yu, Jennifer Rexford, Xin Sun, Sanjay Rao and Nick Feamster, “A Survey of Virtual LAN Usage in Campus Networks”, IEEE Communications Magazine, July 2011.
5. D. L. A. Barber, S. Winkler and Ed. Washington, “The European computer network project”, Computer Communications: Impacts and Implications, D.C., pp. 192-200, 1972.
6. Huston G. and Telstra (), “The future of TCP, The internet protocol Journal”, Cisco Systems, vol. 3, no. 3, 2000.
7. Jin Qian and Tod Beardsley, “The TCP Split Handshake: Practical Effects on Modern Network Equipment”, Network Protocols and Algorithms, vol. 2, no. 1, ISSN 1943-3581, pp. 197-217, 2010.
8. K. S. Klower, “Improving the efficiency of the OSI checksum calculation”, Computer Communication Review, vol. 19, no. 5, pp. 44–55, Oct. 1989
9. Douglas E. Comer, “Internetworking with TCP/IP”, vol. 1, Prentice Hall Inc., 1991
10. Darshan Lal Meena and Dr. R.S. Jadon, “Distributed Denial of Service Attacks and Their Suggested Defense Remedial Approaches”, International Journal of Advance Research in Computer Science and Management Studies, vol. 2, ISSN: 2321-7782, pp. 183-197, 4 Apr. 2014.
11. Konika Abid and Ajay Kumar Singh, “ARP Spoofing Detection via Wireshark and Veracode”, International Journal of New Technology and Research (IJNTR), ISSN: 2454-4116, vol. 4, Issue 5, pp. 27 - 30, May 2018.
12. Konika Abid and Ajay Kumar Singh, “MAC Spoofing Detection Via Wireshark”, International Journal of New Technology and Research (IJNTR), ISSN: 2454-4116, vol. 4, Issue 3, pp. 67-71, March 2018.