Tunneling Protocols Identification Using Light Packet Inspection

Keihan Kazemi
Department of Electrical and Computer Engineering
Isfahan University of Technology
Isfahan, Iran
keihan.kazemi@ec.iut.ac.ir

Ali Fanian
Department of Electrical and Computer Engineering
Isfahan University of Technology
Isfahan, Iran
a.fanian@cc.iut.ac.ir

***Abstract*—Network traffic identification is an essential component for effective network analysis and management. Deep Packet Inspection is one of the main methods for traffic identification. DPI methods have high processing cost and require sufficient memory and CPU resources, which lead to the low efficiency. Furthermore, these methods search payload of the packets which may raise the privacy concern of the network users. In this paper we propose an approach for network tunneling protocols identification which is called Light Packet Inspection that overcomes the weaknesses of traditional DPI methods. We introduce major tunneling protocols such as IPsec, PPTP and OpenVPN communication mechanism in detail, and give an analysis of their packets and traffic behaviors. The experiment results show that the proposed approach can identify tunnels in the early period of time with high accuracy and low processing cost.**

Keywords-traffic identification;Light packet inspection; tunneling protocol

I. INTRODUCTION

Nowadays, security and privacy are important issues in the computer networks. Tunneling protocols play a great role in maintaining anonymous and safe transaction while working with the internet. These protocols are used to protect the data being exchanged and the legitimate user can only interpret data without intruding any unauthorized third party. Therefore, tunneling protocols are a major way to securely transmit information between two ends. With the extensive use of VPN, a variety of tunneling protocols are emerging. In this paper, we discuss three popular tunneling protocols (PPTP, IPsec and OpenVPN) with each being appropriate for a different special tunneling target.

Tunneling protocols can be used in both legal and illegal activities. Legal tunnel is a tunnel that is authorized to set up by the network administrator always via VPN tunnels for a particular purpose. Against, illegal tunnels make possible the violations of the network security policies. For example, some malicious software such as spywares, viruses and worms may use from tunneling to hide their traffic from firewalls or IDSs. The use of these tunnels is a serious threat for network security. Furthermore identify tunneling traffic like other network traffic, is an essential component for network management due to issues such as lawful interception, network design and engineering, quality of service, dynamic access control and network security analyses. The tunneling traffic analysis can increase the ability of network administrators for effective network management. Several technologies have been proposed for traffic identification so far. Payload-based methods play a more important role in the traffic identification. These methods identify the applications by matching the payload for known application signature [1]. This technique is also called as Deep Packet Inspection (DPI). DPI leads to the traffic identification with high accuracy. While this approach can be accurate, its drawback is that it is computationally costly. In addition, access the contents of the packets is inconsistent with the legal issues related to privacy of the users. DPI is not practical for online analysis because payload packet traces are too large to be able to store and match in high-speed links. In this paper, we propose a high-speed and memory efficient payload-based method for tunneling traffic identification, which is suitable for online usage. In the proposed method to identify protocols only initial packets of the payload are required. This method is faster and consumes less memory and storage in comparison with other existing DPI methods. So, the proposed method is named “Light Packet Inspection” (LPI). Also, the accuracy of this method is comparable with other DPI approaches. As mentioned above, this method only requires initial packets of the payload that refer to negotiation phase of tunnel establishment. In LPI only a few bits of the initial packets are required. Therefore, due to the low memory and processing consumption, it can be used in high speed networks. We introduce LPI method that uses heuristics and a group of rules that identifies tunneling applications from a number of well-known internet applications.

The remainder of the paper is organized as follows. First, in section II we summarize traffic identification strategies. Section III presents tunneling protocols and analyzes their characteristic and introduces LPI method. The experimental results are presented in Section IV. We conclude the paper and mention the areas of future work in Section V.

II. TRAFFIC IDENTIFICATION TECHNIQUES

Traffic Identification techniques divided into three categories: port-based, payload-based and machine learning

978-1-4673-7609-9/15/$31.00 ©2015 IEEE

methods. A port-based technique is the simplest and the most rapid method in traffic identification. But some applications select the ports randomly or some users set-up their applications to use the well-known port numbers to prevent port-based filtering systems. Madhukar and Williamson have done detailed analysis of the port based classification and showed that 30-70% of the data did not classify properly [2].

Machine learning techniques use statistical features of the network traffic such as maximum or minimum packet lengths in each direction, flow durations or inter-packet arrival times [3]. In the first step, machine learning algorithm is trained with a set of features and the model is constructed, then the model is applied to identify unknown application using previously learned rules. Machine learning methods have some limitations in traffic detection. For example the statistical features of traffics are not steady in every network and in all the time. Therefore, when test data set is collected at the same point of the train data set, the accuracy evaluated on it might not be comparable from when the test data set and train data set are collected in different time and place [4]. Machine learning based method is unable to recognize different applications that have similar statistical features (e.g., different application based on P2P). Machine learning methods can be vulnerable to mistreatment of data exchange. For example, if the packets are sent at fixed timing intervals with uniform size, this mistreatment will reduce the accuracy of the traffic detection system. Moreover, behavior of some applications is complex, for example tunneling application can tunnel different protocols and the traffic behavior of tunneling application is affected by the protocol inside the tunnel. So, statistical features can't classify them accurately. These methods usually have lower accuracy for special traffic classification and are difficult to model all traffics.

With experimental nature of the statistical techniques and limitations, the mostly used method for tunneling traffic identification is still DPI. Although several payload and statistical based systems including snort, tie and tstat exist, but these methods cannot identify tunneling traffic. These traffic detection systems inspect the content of IP packets independently of each other. However, these schemes can defrag the fragmented packet for inspection, but they cannot reconstruct the segment of UDP or TCP protocol for checking signatures of an application which split a segment to different IP packets. Since, to establish a tunneling protocol different IP packets must be transmitted in negotiation phase, these schemes cannot detect them. In the proposed method, data packets according to the state diagram of tunneling protocols are placed together effectively, so that if the specification of a tunneling protocol is found, our proposed method can detect it.

III. LIGHT PACKET INSPECTION FOR TUNNELING PROTOCOLS IDENTIFICATION

In theory, payload-based analysis is not useful in encrypted traffic identification. But in practice tunnel applications need an initial handshake and negotiation phase. Since, the exchanged messages in this phase are not encrypted, using payload-based techniques in the presence of encryption are possible [5]. As mentioned before, our proposed method, called LPI, uses the messages exchange in this phase to identify tunneling protocols. LPI method is constructed from the standard protocol specifications and manual observation and analysis. In this section, at first, the details of communication mechanism of three tunneling protocols (IPsec, PPTP and OpenVPN) are presented, and then LPI method for tunneling protocol identification is presented. Fig. 1 shows the LPI method for tunneling protocol identification. As shown at this figure, the proposed method tries to find rules for identifying each of messages in the initial phase of tunneling protocols.

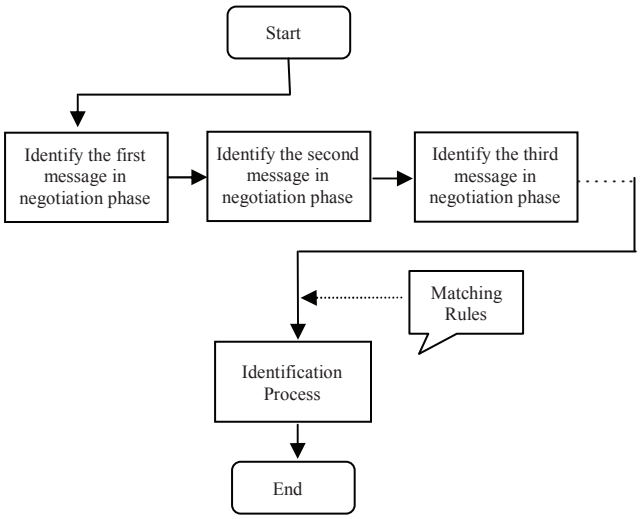

Figure 1. LPI method for tunneling protocols identification

A. IPsec

IPsec (Internet Protocol Security) is a protocol that provides security at the IP layer. IPsec is a universal protocol that is used to construct VPN [6]. IKE protocol is used in the first stage of IPsec connection. This protocol is a security association management protocol used in IPsec to establish a shared secret key and authentication between communicating peers. IKE consists of two phases [7]. Phase 1: establish an authenticated secure tunnel. Phase 2: using a secure channel established in phase 1 for provision of security services such as IPsec. If phase 1 is destroyed, the tunnel will not be established, so for detection of IPsec tunnel, we need to recognize phase 1. The IKE protocol itself is not totally secure. As the first four messages in the main mode of IKE phase 1 are not encrypted. We can find when the IKE phase 1 exchange is integrated, the secure tunnel is established. Consequently, in this paper, detection method is considered to phase 1 of the protocol. There are two common modes in IKE phase 1: aggressive mode and main mode. Main mode contains 6 messages and aggressive mode has 4 messages. In aggressive mode, fewer exchanges are made but some weaknesses can be seen such as it does not provide the identity protect and the peer id's are not kept secret, furthermore both sides have exchanged information before there is a secure channel. Therefore, it is possible to sniff the wire and discover who formed the new security association. Therefore it is insecure and it is vulnerable to man in the middle and DoS attacks. For additional functionality of

main mode, the main mode is considered. IKE consists of ISAKMP, Oakley and SKEME. The structure of the ISAKMP header that is transmitted in IKE is shown in Fig. 2.

<table>
<tr><td colspan="5">Initiator Cookie</td></tr>
<tr><td colspan="5">Responder Cookie</td></tr>
<tr><td>Next Payload</td><td>Major Version</td><td>Minor Version</td><td>Exchange Type</td><td>Flag</td></tr>
<tr><td colspan="3">Message Id</td><td colspan="2">Message Length</td></tr>
</table>

Figure 2. ISAKMP Header

The initiator cookie and responder cookie are specified by the ISAKMP peers to provide SA establishment, SA notification or SA elimination. Next payload field indicates the ISAKMP payload type of the first payload in the message including SA payload, identification payload, transform and key exchange. Exchange type Indicates the message orderings in the ISAKMP exchanges and includes (main mode, quick mode and other modes). Main mode uses six messages. We examine these six messages in order to find whether the IKE exchanges are legitimate or not. Table I shows details of these six messages.

TABLE I. IKE PHASE 1 EXCHANGE

Messages	Initiator Cookie	Responder Cookie	Next Payload	Exchange Type
First Message	Nonce 1	0	1	2
Second Message	Nonce 1	Nonce 2	1	2
Third Message	Nonce 1	Nonce 2	4	2
Forth Message	Nonce 1	Nonce 2	4	2
Fifth Message	Nonce 1	Nonce 2	5	2
Sixth Message	Nonce 1	Nonce 2	5	2

With LPI method, it can possible to identify the protocol even if an attacker wants to tempt identification protocol by injecting irrelevant data. To identify IPsec protocol, if the responder cookie of the message is 0 assumed that it is the negotiation request message. In the following steps the rules for IPsec identification are presented.

1. If Responder Cookie is 0 & Next Payload is 1 & Exchange Type is 2 then record Initiator Cookie as Nonce1 and Source IP as the ClientIP.
2. If Initiator Cookie is Nonce1 & Next Payload is 1 & Exchange Type is 2 then record Responder Cookie as Nonce2 and Source IP as the ServerIP.
3. If Initial Cookie is Nonce1 & Responder Cookie is Nonce2 & Next Payload is 4 & Exchange Type is 2 & Source IP is ClientIP.
4. If Initial Cookie is Nonce1 & Responder Cookie is Nonce2 & Next Payload is 4 & Exchange Type is 2 & Source IP is ServerIP.
5. If Initial Cookie is Nonce1 & Responder Cookie is Nonce2 & Next Payload is 5 & Exchange Type is 2 & Source IP is ClientIP.
6. If Initial Cookie is Nonce1 & Responder Cookie is Nonce2 & Next Payload is 5 & Exchange Type is 2 & Source IP is ServerIP.

B. PPTP

PPTP (Point-to-Point Tunneling Protocol) is one of the VPN protocols that is developed by Microsoft. PPTP operates at Layer 2 of the OSI model [8]. This protocol creates a TCP control connection between the VPN client and VPN server to establish a tunnel and encapsulate the packets into the PPP packets. PPTP uses GRE to encapsulate packets between client and server. This protocol also supports data encryption and compression. PPTP supports two types of message: control and data messages. Control messages are not encrypted and used for establishment, management, maintenance and teardown the tunnel and session. A control connection is created for each PAC[1], PNS[2] pair and operates over TCP. After the tunnel is created, data messages carry encapsulated PPP packets across the tunnel. At the setup time of PPTP connection, many control packets are exchanged between server and client to establish tunnel and session for each direction. One peer requests the other peer to assign a specific tunnel and session through these control packets. Then, using this tunnel and session, data packets are exchanged with the compressed PPP frames as payload. There are different control messages to maintain VPN connections, but here only the control messages used for PPTP detection are discussed. These messages are presented in the Fig. 3.

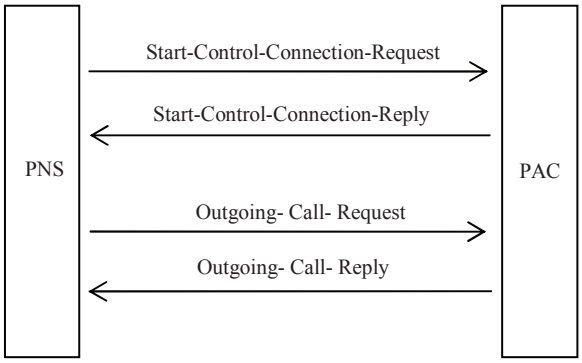

Figure 3. PPTP Connection Establishment

The Start-Control-Connection-Request is a PPTP control message that is used to establish the control connection between a PNS and a PAC. Each PNS-PAC pair requires a dedicated control connection to be established. A control connection must be established before any other PPTP messages can be issued. The Start-Control-Connection-Reply is a PPTP control message that is sent in reply to a received Start-Control-Connection-Request message. The Outgoing-Call-Request is sent by the PNS to the PAC to indicate that an outbound call from the PAC is to be established. This request provides the PAC with information that is required to make the call. The Outgoing-Call-Reply is a PPTP control message that is sent by the PAC to the PNS in response to a received Outgoing-Call-Request message.

Fig. 4 shows the first 12 bytes of control messages header that is useful for our LPI method.

[1]PPTP Access Concentrator
[2]PPTP Network Server

Length	PPTP Message Type
Magic Cookie (4 bytes)	
Control Message Type	Reserved

Figure 4. Control Messages Header of PPTP Protocol

Length field is total length in octets of this PPTP message including the entire PPTP header. Possible values for PPTP message type are 1 for control message and 2 for management message. Management messages are not defined currently. The magic cookie is always sent as the constant 0x1A2B3C4D to allow the receiver to ensure that it is properly synchronized with the TCP data stream. The control message type defines the specific type of control message. The currently defined control messages types are: 1 for Start-Control-Connection-Request, 2 for Start-Control-Connection-Reply, 7 for Outgoing-Call-Request and 8 for Outgoing-Call-Reply. We examine these four control messages in order to find whether the PPTP tunnel is established and it is legitimate or not. Details of these four messages are shown in the following table.

TABLE II. PPTP PHASE 1 EXCHANGE

Messages	PPTP Message Type	Magic Cookie	Control Message Type
Start-Control-Connection-Request	1	1A2B3C4D	1
Start-Control-Connection-Reply	1	1A2B3C4D	2
Outgoing-Call-Request	1	1A2B3C4D	7
Outgoing-Call-Reply	1	1A2B3C4D	8

However, we can build the identification method with more and more complex rules, but the goal of the proposed method is creation a method with high accuracy and minimum rules. In following, the steps of PPTP protocol identification are presented.

1. If PPTP Message Type is 1 & Magic Cookie is 0x1A2B3C4D & Control Message Type is 1 then record Source IP as PNS IP.
2. If PPTP Message Type is 1 & Magic Cookie is 0x1A2B3C4D & Control Message Type is 2 then record Source IP as PAC IP.
3. If PPTP Message Type is 1 & Magic Cookie is 0x1A2B3C4D & Control Message Type is 7 & Source IP is PNS IP.
4. If PPTP Message Type is 1 & Magic Cookie is 0x1A2B3C4D & Control Message Type is 8 & Source IP is PAC IP.

C. *OpenVPN*

OpenVPN is a VPN implementation based on the SSL/TLS protocol [9]. It uses encryption, authentication and certification features of the OpenSSL library to protect network traffic [10]. OpenVPN uses TCP or UDP as its transport protocol. To bypass restrictive firewalls, OpenVPN can be configured to use well known TCP port such as 80 and 443 or UDP port 53. OpenVPN protocol has the control and data channels. The control channel is used in the initial phase of OpenVPN which includes OpenVPN connection initialization and SSL/TLS handshaking. In this phase, some activity including session starting, key exchange, cipher suite agreement, identity authentication and so on is done. After SSL/TLS handshake, encrypted data packets pass from data channel. The P_DATA message type represents encrypted, encapsulated tunnel packets. The sequence of OpenVPN signaling message is depicted in Fig. 5.

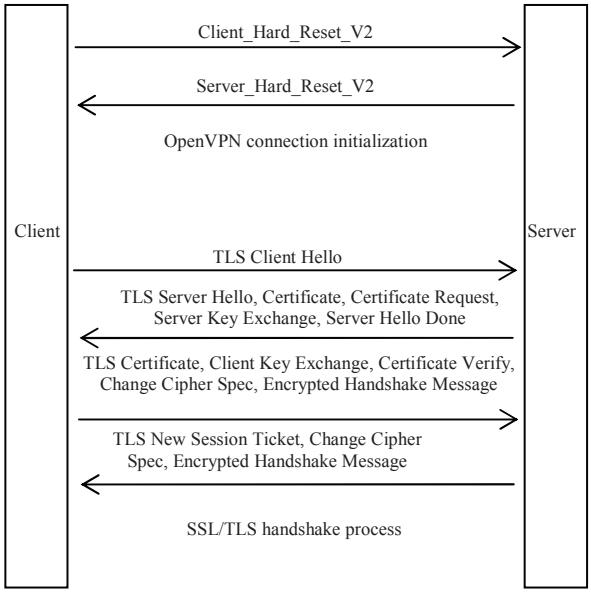

Figure 5. OpenVPN Establishment Process

In addition to the messages shown in Fig. 5, there are two message types in the OpenVPN control channel including P_CONTROL and P_ACK message types. P_CONTROL is a control channel packet and P_ACK is an acknowledgement indicating P_CONTROL packets has been received. For better identification of OpenVPN protocol, OpenVPN packet format is shown in Fig. 6.

Packet Length	Opcode	Key Id	Session Id	HMAC
Packet Id	Packet Id Array Length	Packet Id Array	Remote Session Id	Message Packet Id

Figure 6. OpenVPN Header

As [11] referred, the packet length field is only in TCP protocol. The opcode field identifies the type of control packets. The opcode for the Client_Hard_Reset, Server_Hard_Reset, P_Control and P_Ack messages are respectively, 7, 8, 4 and 5. The session id is a 64 bits random value to identify TLS session. The HMAC is a standard SHA-1(20 bytes) or MD5 (16 bytes). If TLS-authentication is specified, HMAC and packet id field is in the header. Packet id includes sequence number and optional timestamp. The packet-id array length field is a single byte that indicates how many acknowledgments are in the buffer. If this byte is zero, the packet id array and remote session id field is not present in the message. The remote session id is 8 bytes. The value of this field is the session id of packets that are sent from the other side. Message packet id field is the sequence number in each

direction. Each TLS record is sent via the OpenVPN control messages, can be fragmented and the message packet id field identifies the current fragment. In OpenVPN protocol, the number and location of header fields are different according to the following issues:

- Protocol is configured on the TCP or the UDP?
- Is TLS-authentication specified?
- If TLS authentications is specified, SHA-1 algorithms is used or MD5 algorithm?
- Is packet id field contains the optional time stamp?
- Value of packet id array length is zero or one?

The first control message that is exchanged in OpenVPN protocol is Client_Hard_Reset message. At first, we examine the transport layer header to find out it uses TCP or UDP protocol. Our analysis indicates that the value of packet id array length field in Client_Hard_Reset message is always zero, so the packet id array and remote session id field are not present in Client_Hard_Reset message. If TLS-authentication is specified, all messages have HMAC signature. According to the first packet size can be realized whether OpenVPN used TLS-authentication and optional timestamp or not. This is shown in table III.

TABLE III. CLIENT_HARD_RESET LENGTH

Messages	OpenVPN layer size (UDP,TCP)	Description
Client_Hard_Reset	14,16	No TLS-authentication
Client_Hard_Reset	34,36	HMAC (MD5) & No Timestamp
Client_Hard_Reset	38,40	HMAC (MD5) & Timestamp OR HMAC (SHA-1) & No timestamp
Client_Hard_Reset	42,44	HMAC (SHA-1) & Timestamp

So after seeing the first message, location of packet id array length field is determined and with checking its value, the location of message packet id field is specified. After seeing two messages of Client_Hard_Reset and Server_Hard_Reset, we have to look for SSL Handshake messages. Since, OpenVPN connection is established using SSL handshake protocol, SSL handshake messages can be found in the payload of P_CONTROL message. P_CONTROL messages have an extra field at the end of packet header as a message fragment with the maximum length 100 bytes. Between client and server, a lot of P_CONTROL messages are sent. The data of message fragment fields of P_CONTROL messages contain information related to SSL handshake messages. Except the first message in each direction, other SSL handshake messages can be anywhere in the message fragment field of P_CONTROL messages. For example Server Key Exchange message can be found in the middle of the P_CONTROL message. Therefore, it is not required looking for all the messages that are shown in the Fig. 5. Consequently, instead of looking for all messages, to identify OpenVPN flow, some heuristic can be declared. For example in one heuristic we are trying to find the first message of SSL handshake in each direction. After Client_Hard_Reset and Server_Hard_Reset, the messages with opcode 4 that are sent from the client are considered and first few bytes of the data in message fragment field are checked, if the value of these byte is one of the following value 20, 21, 22, 23 and the second byte is 3 and third byte is one of the following value 0, 1, 2, 3 and sixth byte is 1, in our analysis, we find that it is the format of SSL recording head in the TLS Client Hello message. For the opposite direction the same operation is done. To this end, when messages with opcode 4 are sent from the server to the client, the first few bytes of the data in message fragment field are considered, if these byte are one of the value of 20, 21, 22, 23 and the second byte is 3 and third byte is one of the value of 0, 1, 2, 3 and sixth byte is 2, it can be inferred that it is the format of SSL recording head and TLS Server Hello. Next, the same operation is done for TLS Certificate message and TLS New Session Ticket with the difference that sixth byte in TLS Certificate message is 11 and in TLS New Session Ticket message is 4. If this process is occurred 4 times, it can to be sure that SSL handshake is done. In the proposed method, these six messages are examined in order to find whether the OpenVPN tunnel is established and it is legitimate or not. Details of these six messages are shown in the following tables.

TABLE IV. OPENVPN PHASE 1 EXCHANGE

Messages	Opcode	Packet Id Array Length	Message Packet Id
Client_Hard_Reset_V2	7	0	0
Server_Hard_Reset_V2	8	1	0

TABLE V. CONTEX OF SSL/TLS HANDSHAKE MESSAGES

Messages	Opcode	First byte	Second byte	Third byte	Sixth byte
First exchange in SSL handshake	4	20, 21, 22, 23	3	0,1,2,3	1
Second exchange in SSL handshake	4	20, 21, 22, 23	3	0,1,2,3	2
Third exchange in SSL handshake	4	20, 21, 22, 23	3	0,1,2,3	11
Forth exchange in SSL handshake	4	20, 21, 22, 23	3	0,1,2,3	4

The steps for OpenVPN identification in LPI methods are as follows:

1. If Opcode is 7 & Packet Id Array Length is 0 & Message Packet-Id is 0 then record Source IP as the ClientIP & Session Id as Client Id.
2. If Opcode is 8 & Packet Id Array Length is 1 & Message Packet Id is 0 then record Source IP as the ServerIP & Session Id as Server Id.
3. If Opcode is 4 & Source IP is the ClientIP & Session Id is Client Id & first byte of Message Fragment is one of the value of 20, 21, 22, 23 and second byte is 3 and third byte is one of the value of 0, 1, 2, 3 and sixth byte is 1.
4. If Opcode is 4 & Source IP is the ServerIP & Session Id is Server Id & first byte of Message Fragment is one of the value of 20, 21, 22, 23 and second byte is 3 and third byte is one of the value of 0, 1, 2, 3 and sixth byte is 2.

5. If Opcode is 4 & Source IP is the ClientIP & Session Id is Client Id & first byte of Message Fragment is one of the value of 20, 21, 22, 23 and second byte is 3 and third byte is one of the value of 0, 1, 2, 3 and sixth byte is 11.
6. If Opcode is 4 & Source IP is the ServerIP & Session Id is Server Id & first byte of Message Fragment is one of the value of 20, 21, 22, 23 and second byte is 3 and third byte is one of the value of 0, 1, 2, 3 and sixth byte is 4.

IV. EVALUATION

A. Data set

Unfortunately, none of the available data sets benefit from a variety of suitable protocols. For example Moore data set [12] and others, do not include tunneling traffic and therefore cannot be used to evaluate the accuracy of the proposed method in the detection of tunnel traffic. In this paper, we use the IUT.F.D data set that was produced at Isfahan University of Technology in the summer of 2014 from the real traffic. Table VI shows the traffic profile that is used as reference for evaluating LPI accuracy.

TABLE VI. PROTOCOL DISTRIBUTION OF OUR TRACE

Protocol Name	Size(Number of Flows)
HTTP	123548
P2P	36723
FTP	2834
SSL	1839
Mail	28397
Unknown	85631
Tunneling(PPTP, IPsec, OpenVPN)	12945

B. LPI PerformanceEvaluation

Since, available traffic identification schemes or DPI mechanisms unable to identify tunneling protocols; the performance of the proposed method is not comparable to other traffic identification schemes. Therefore, in this section the performance of the proposed scheme is discussed alone as following.

- High accuracy: LPI method has high accuracy and low error rates such that distinguishes each of the PPTP, IPsec and OpenVPN protocols with almost 100% accuracy from each other and from non-tunneling protocols.
- High speed: LPI method has a few rules, so introduces low overheads and high Performance. This method identifies tunneling protocols quickly such that for identifying IPsec, PPTP and OpenVPN, it is required to consider only at most six first packets, so it is feasible to use for online real-time traffic classification on high-speed links.
- Identify tunneling protocols in the early period of time: Whatever a method can detect connection earlier, the performance will be higher. To provide privacy of users, quality of service and resource saving, early detection is important. LPI method reads packets in negotiation phase as few as possible and requires first few bits of each packet, so reduce the privacy matter in DPI. Furthermore, it requires very low memory and CPU resource to store and match the payloads of packets, this is important specially when we use LPI in high-speed links where there are large numbers of connections at the same time.

V. CONCLUSION AND FUTURE WORK

Traffic identification is one of the main components in network monitoring and security. In this paper we focus on tunneling protocols that play a great role in security and safety of the transmitted data to the network. Although tunneling protocols benefit from encryption, but tunneling establishment consist of a negotiation phase for exchanging connection parameters at the beginning of the connection. Messages exchanged in this phase are clear and can be used for payload-based identification. In this paper, we propose a new traffic identification approach, called Light Packet Inspection, which overcomes the weaknesses of deep packet inspection methods. LPI requires initial packets of each flow. LPI is fast and requires significantly less CPU and memory storage than other DPI methods. The result shows that for having a good accuracy in payload-based methods, we do not need to capture the whole payload. Future work includes LPI development for other tunneling protocols.

REFERENCES

[1] F. Yang., Z. Zhang, "Research of application protocol identification system based DPI and DFI," Lecture Notes in Electrical Engineering, LNEE, vol. 127, pp. 305–310, 2012.

[2] A. Madhukar and C. Williamson, "A longitudinal study of P2P traffic classification," in 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and elecommunication Systems, September 2006.

[3] R. Alshammari and A. Nur Zincir-Heywood, "Unveiling skype encrypted tunnels using GP," in Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE, Barcelona, Spain, July 2010.

[4] T. Nguyen, and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," IEEE Communications Surveys and Tutorials, vol. 10, no. 4, pp. 56-76, 2008.

[5] A. Dainotti, A. Pescapé and K.C. Claffy, "Issues and future directions in traffic classification," IEEE Network journal, vol. 26, 2012.

[6] RFC 6071: IP Security (IPsec) and Internet Key Exchange (IKE).

[7] R. Perlman, C. Kaufman, "Key exchange in IPSec: analysis of IKE," Internet Computing, IEEE, vol. 4, Issue 6, pp. 50-56, 2000.

[8] RFC 2637: Point-to-Point Tunneling Protocol (PPTP)

[9] C. Hosner, "OpenVPN and the SSL VPN revolution," SANS Institute, (2004)

[10] The OpenSSL Project: The open source toolkit for ssl/tls, http://www.openssl.org

[11] http://openvpn.net/index.php/open-source/documentation.html

[12] http://cl.cam.ac.uk/research/srg/netos/nprobe/data/papers/sigmetrics.