Browser Identification Based on Encrypted Traffic

Changjiang Liu, Jiesi Han* and Qiang Wei
National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu Sichuan, China
*Corresponding author

***Abstract*—Network traffic encryption brings security to network communication, however it also brings challenges to network monitoring. As more and more major websites use encryption protocol to protect imformation of visitors, it is a burning issue to identify client when session is encrypted. In this paper, we present browser identification of HTTPS client based on packet length. The sequence of request packet length is different among browsers and our experiment shows that it is possible to recognize what kind of browser the traffic comes from according to length sequence. We theoretically analyze the possibility of using the request packet length to identify browsers and show the method of using length sequence to establish dictionary. The dictionaries are used to distinguish unknown traffic flow. Our experiment results show that we can get accurate results of browser identification in HTTPS communication through packet length analysis.**

Keywords-HTTPS; accecss traffic; packet length; browser identificaion

I. INTRODUCTION

Being aware of browsers used by client can help administrators to protect network security [1]. Nowadays, we are able to identify browsers through the features in plain-text network traffic, such as User-Agent and Cookies[2], but with the rising popularity of encryption network protocol, it becomes impossible to get these features.

HTTPS protocol which means HTTP runs over SSL/TLS encryption protocol[1] , is the most commonly used encryption network protocol. In HTTPS communication, the HTTP packet data will be encrypted, a SSL/TLS head will be added to the beginning of HTTP data and MAC data will be added to the end of HTTP data. The workflow of SSL/TLS data packaging can be described as Figure I. Different versions of SSL/TLS protocol may use different algorithms to compress data, calculate MAC data and encrypt data.

Before data communication, host and server would exchange some encryption massages, this period is known as SSL/TLS Handshake[1]. After handshake, the communication data will be encrypted and we cannot observe HTTP information anymore.

II. RELATED WORK

The long-term SSL/TLS traffic identify in the Internet was presented by Levillain and Ebalard in 2010[2]. They detected certificate chains which did not comply with the standard through SSL/TLS handshake fingerprint. Another study of SSL/TLS traffic was raised by Holz and partners in 2011[3], who also focus on certificate properties. Roni and Langberg classified encrypted network flows by their application type[4].Velan and Milan found that the initiation of an encrypted connection and the protocol structure give away much information about traffic clissificaion[5].The SSL/TLS protocol and its applications were analyzed by Qualys SSL Lab[6] , they proposed the idea of HTTP client fingerprinting using the information of SSL/TLS handshake. Martin Husák and colleagues gave a way to estimate User-Agent of a client in HTTPS communication through the fingerprint of initial SSL/TLS handshake in 2015[7].However, due to the fuzziness of the fingerprint, the identification of browsers was not accurate.Salusky and Thomas disclosed for fingerprinting and identifying client applications based on the analysis of client requests in an HTTP-based communication[8].For the algorithm of traffic identification, Alshammari and his colleagues assessed the robustness of machine learning for classifying encrypted traffic[9]. They foound that the C4.5 based approach performs much better than adaboost, support vector machine, Naive Bayesian and RIPPER.

This paper propose a method to identify browsers through packet length of request traffic on a particular Web page. Our research questions are focus on:

- Which packets of all the traffic we captured can be used to identify browser?
- Which features of HTTPS session can be used to identify browser?
- How much information do we need to make decision?
- How is the identification accuracy change over time
- What degree of identification accuracy can we achieve?

This paper is divided into four sections. The experiment design and experiment environment are described in Section 2. In this section, we carry out theoretical analysis and give the answers of question 1. As browsers can work in cache banned mode or cache allowed mode, we elaborate the browser identification of these two mode and give the answer of question 2 in Section 3. The experiment results, which give the answer of remaining problems, are present in Section 4. Finally, we give the conclusion of our experiments in Section 5.

III. ANALYSIS

HTTPS means HTTP over SSL/TLS[10]. The process of SSL/TLS data packaging can be described as Figure I.

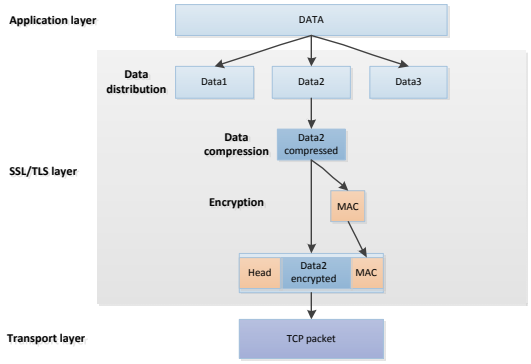

FIGURE I. SSL/TLS ENCRYPTION PROCESS

HTTPS protocol uses the SSL/TLS protocol to cut and encrypt the data of the application layer. We assume that the length of application layer data is x, SSL/TLS encryption algorithm is f, length of TCP head, IP head and Ethernet head are l_{tcp_head}, l_{ip_head} and $l_{ethernet\_head}$. The length of HTTPS packet can be expressed as:

$$l_{packet} = f(x) + l_{tcp\_head} + l_{ip\_head} + l_{ethernet\_head} \quad (1)$$

The length of TCP, IP and Ethernet head are common in general through RFC[11] document, so the length of HTTPS packet is mostly determined by f and x.

HTTP request packet consist of Request Line and HTTP Headers. The Request Line shows what content this packet request, the length of it is determined by the request content. HTTP Headers include Host, Connection, User-Agent, Cookie, and so on. User-Agent represent some information about operating system and browser, the length of it will change when browsers or operating systems different. Cookie is the cache data stored on local terminal. It is worth to note that browsers can work in cache allowed mode and cache banned mode, this is set by users. Then the length of application data of HTTP request packet is denoted as:

$$x = l_{request\,line} + l_{headers} \quad (2)$$

Different browsers may set different header, which makes their value different even they have the same request line.

Different browsers also have different encryption algorithm f to compress data, calculate MAC and encrypt data, this also leads to the diversity of packet length.

Then the length of request packet can be expressed as:

$$l_{packet} = F(l_{request\,line} + l_{headers}) \quad (3)$$

where F is a comprehensive function of f and remaining data.

So, when packet request for the same content, its packet length depends on the properties of browsers. This is why this article uses the length of request packet as browser feature.

Just like TCP handshake, SSL/TLS connection also process an initial handshake stage before data is transfered. The packets translated in this phase we call handshake packet. These packet deliver the infornation about encryption algorithm between browser and web server. Different browsers may have different encryption algorithms, so the length of handshake packet they send is also different.

Taking into account the situation that we need to use VPN to access some web page, we also analyze the traffic flow of VPN. Traffic encrypted by VPN is hard to distinguish between the handshake packet and the request packet. Fortunately, VPN encryption does not change the relative size of the original packet.

IV. BROWSER IDENTIFICATION BASED ON PACKET LENGTH

A. *Traffic Interception*

In this paper, we use the top 5 of most-used browsers according to Net Market Share[12] and Stat Counter[13] to conduct our experiment, they are IE, Chrome, Firefox, Safari and Opera. The total market share of these five browsers is more than 90% through the statistic result. We install these browsers on a windows 7 computer, using them to browser different web pages, and opening Wireshark to capture network traffic. We choose the top 10 of the most-visited encryption websites in Alexa to do experiments. Part of these websites use VPN to connect . The ten websites are showed in Table I.

Considering that user can set whether to allow the browser cache, we collect traffic in two modes, the first mode is the default setting and the second mode is cache banned setting. The cache setting of five browsers are shown in Table II.

TABLE I. TEST WEB SITES

Website	HTTPS or VPN	Visitors per million/million	Alexa ranking
Google.com	VPN	42.2	1
Youtube.com	VPN	37.3	2
Facebook.com	VPN	35.6	3
Baidu.com	HTTPS	12.4	4
Yahoo.com	HTTPS	10.8	5
Amazon.com	HTTPS	7.2	6
Wikipedia.org	HTTPS	9.7	7
Twitter.com	VPN	6.7	9
Live.com	HTTPS	6.5	11
Taobao.com	HTTPS	5.4	12

TABLE II. CACHE SETTING OF DIFFERENT BROWSERS

Browser	Cache setting
IE	Delete record when exit the browser
Chrome	Block third party cookie and site data
Firefox	Do not record history
Safari	Always block Cookie
Opera	Delete record when exit the browser Block third party cookie and site data

For each browser, the steps of traffic interception as follow:

(1) Close all programs that may produce request traffic ;

(2) Run python script to request websites recorded in a document and wait for 40 seconds when open each website;

(3) Open wireshark to catch traffic flow;

(4) Suspend python program for 30 seconds before request next website;

(5) Stop wireshark and save data before python open next website;

(6) Repeat (3)(4)(5) to capture enough traffic data;

(7) Change cache setting of browsers and repeat above steps to collect traffic of cache banning mode.

In fact, webpage may have some contents which change with time, for example news messages. In order to study the influence of web page content change with time, we repeat traffic interception in different days. For the first day, we have 10 data collections of each browser access different website. During the following week, we have 5 data collections of each browser access different website. In this paper, we collected 2250 data in total.

After traffic interception, we use source IP and protocol type to extract traffic from browsers to web servers. In practical application, the target computer may run more than one program which will produces request traffic. In this situation, we can add the destination IP address to filter traffic. Websites may have different servers to respond requests and the destination IP address of request packet may change over time. Fortunately, the number of server for each website is limited, we can count all IP address of websites by iterate visit website.

B. Feature Extraction

Considering that we need using VPN which runs on SSL protocol to visit some foreign websites in China, we use two kinds of way to extract packet length as feature.

For the HTTPS traffic, we extract the source IP, destination IP, source port, and destination port for each packet to form a four tuple. We use four tuple to reorder the packet, so that the packets with the same four tuple are adjacent.

$$Four\_Touple = [srcIP, dstIP, srcPort, dstPort] \qquad (4)$$

Then we we extract the length and type of each packet in the traffic flow to form the characteristic sequence. For example in the characteristic sequence $[(261,0),(223,1),(299,1)]$, 261, 223, 229 are length of three packets, 0 represents this packet is a handshake packet and 1 means this packet is a request packet.

Figure II describes the process of extracting feature from HTTPS traffic.

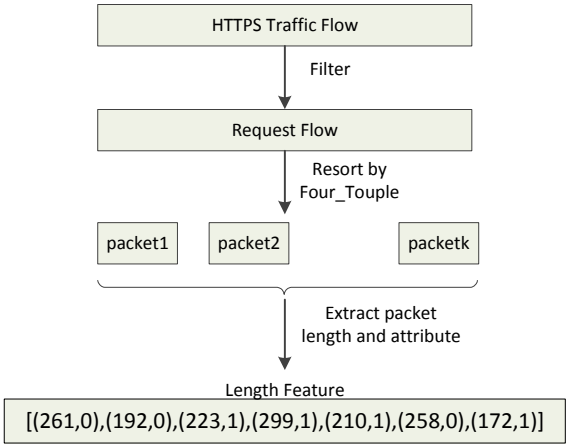

FIGURE II. FEATURE EXTRACTION OF HTTPS TRAFFIC

For traffic of VPN, we cannot separate the handshake packets from the traffic because VPN implement a second data encryption. We extract length of all packets to form the length sequence. But in this situation, we need do some preprocessing of the packet length sequence to improve the identification accuracy. The relative size of request packet length is mostly determined by Request line when type of encryption protocol version and browser are fixed, as we anayzed in chapter III. So we rank each length sequence to make the position of packet length which request for same contents more closer.

Figure III shows the process of extracting feature from VPN traffic.

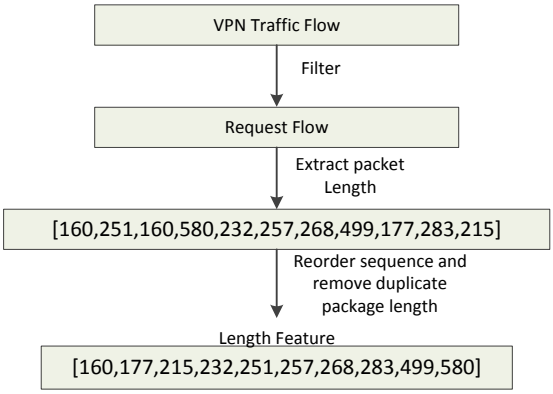

FIGURE III. FEATURE EXTRACTION OF VPN TRAFFIC

Due to the instability of the network environment, some of the request packets sent by the browser can not be transmitted to the web server, which leads to the packet retransmission. So for each length sequence, we also need to remove the duplicate packet length.

V. EXPERIMENT RESULTS

We use C4.5 to identify browsers. For the first day, we use 10 Fold Cross-Validation to test identify accuracy. The result is showed in Figure IV.

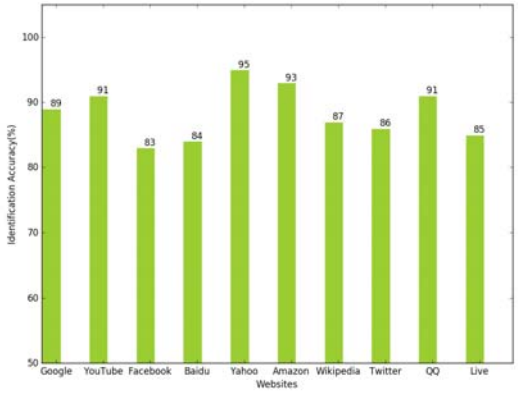

FIGURE IV. IDENTIFICATION ACCURACY OF 10 CROSS-VALIDATION USING DATA FROM THE FIRST DAY

The identification accuracy on these 10 websites are higher than 80%. Websites like YouTube, Yahoo, Amazon and QQ which have rich contents can get higher identification accuracy than 90%, because we can get more traffic data and thus obtain more information to determine the browser type.

In order to analyze the influence of time on the feature, we use data of first day to be training set and data of next five days to be test sets. The identification results are showed in Figure V:

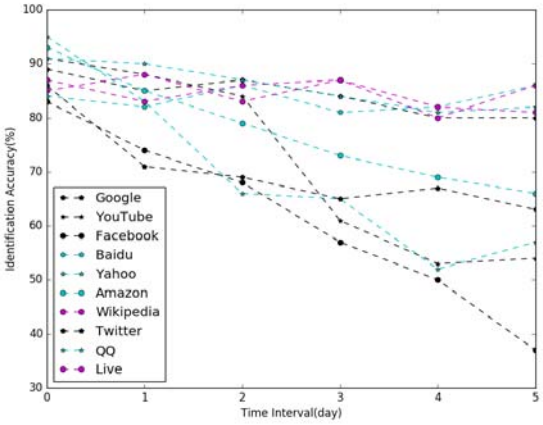

FIGURE V. IDENTIFICATION ACCURACY OF NEXT FIVE DAY WHILE USING DATA OF FIRST DAY TO BE TRAINING SET

From Figure V we can see that most of website can maintain a recognition accuracy of more than 70% when using data of next day to be testing set. However, with the increase of time interval, the recognition accuracy of each site has declined. Websites like Google, Baidu, Live and Wikipedia can maintain a more stable Identification accuracy, this is because this kind of websites have simple and stable contents. This means that the content have a low affection on packet length. Instead, contents of YouTube and Facebook changes a lot over time, this leads to a large change in the length of request packet.

VI. CONCLUSION

In this paper, we have shown that it is possible to identify browsers in HTTPS communication by using length sequence of request packets, even if the traffic is encrypted again by VPN. The influence factors of packet length are numerous, we take into account different cache setting for browser to improve the applicability of algorithm. We also test the stability of the feature over time.

Results show that C4.5 algorithm can achieve more than 80% identification accuracy base on packet length feature of different web pages. Our research also shows that for some web pages like Google, Baidu and Live, the length feature extracted in first day can maintain a recognition accuracy of more than 80% when using data of next five day to test. It means that the packet length feature in this paper has a better stability in content stable web sites.

Different types of browsers have different versions, our future work is to identify small version of browsers to get more accurate results.

REFERENCES

[1] A. Freier, P Karlton, and P Kocher. “The Secure Sockets Layer (SSL) Protocol Version 3.0,” The syncretic religion of Lin Chao-en /. Columbia University Press, 1980:374-380.

[2] O. Levillain, A. Ébalard, B. Morin B, et al. “One year of SSL internet measurement,” Proceedings of the 28th Annual Computer Security Applications Conference. ACM, 2012:11-20.

[3] R. Holz, L. Braun, N. Kammenhuber, et al. “The SSL landscape: a thorough analysis of the x.509 PKI using active and passive measurements,” ACM SIGCOMM Conference on Internet Measurement Conference. ACM, 2011:427-444.

[4] Roni Bar Yanai, M. Langberg, Peleg D, et al. Realtime Classification for Encrypted Traffic[C]// Experimental Algorithms, International Symposium, Sea 2010, Ischia Island, Naples, Italy, May 20-22, 2010. Proceedings. 2010:373-385.

[5] P. Velan, Milan. “A survey of methods for encrypted traffic classification and analysis,” International Journal of Network Management, 2015, 25(5):355–374.

[6] Qualys SSL Lab, “HTTP client fingerprinting using SSL handshake analysis,” https://www.helpnetsecurity.com/?id=16487.

[7] M. Husák, M. Čermá, T. Jirsí, et al. “HTTPS traffic analysis and client identification using passive SSL/TLS fingerprinting,”. Eurasip Journal on Information Security, 2016, 2016(1):1-14.

[8] W. Salusky, m.Thomas. “Client application fingerprinting based on analysis of client requests,” US, US8694608[P]. 2014.

[9] Alshammari R, Zincir-Heywood A N. Machine learning based encrypted traffic classification: Identifying SSH and Skype[C]// Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on. 2009:289-296.

[10] E Rescorla, HTTP Over TLS. IETF. Updated by RFCs 5785, 7230 (2000).http://www.ietf.org/rfc/rfc2818.txt

[11] https://en.wikipedia.org/wiki/RFC

[12] http://www.netmarketshare.com/

[13] http://gs.statcounter.com/