Fine-grained Identification for SSL/TLS Packets

Lingjing Kong, Ying Zhou, Guowei Huang, and Huijing Wang
(Corresponding author: Lingjing Kong)

School of Computer Science, Shenzhen Institute of Information Technology
No. 2188, Longxiang Rd, Longgang District, Shenzhen, Guangdong, China
(Email: lingjk11@gmail.com)

(Received June 9, 2019; Revised and Accepted Dec. 3, 2019; First Online Feb. 1, 2020)

Abstract

SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocols have been widely used in data transmission to protect the security and integrity of the data. However, due to the encryption of SSL/TLS, the application data over transmitted packets are invisible and difficult to be distinguished by traditional port-based and DPI(Deep Packet Inspection) ways. Though the method based on statistical features can overcome shortages of the above two ways, it is still hard to achieve fine-grained identification. In this paper, we proposed a solution to extract fingerprint information and then identify the types of flows during handshake phase to avoid inspecting the encrypted data and privacy violation. Besides, two hash tables are built to help fast identify the packets with the same APP ID in the same or among different conversations. Finally, 300 flows are captured and the experiment results show the method is accurate and efficient.

Keywords: Fine-Grained; Identification; SSL/TLS

1 Introduction

For the security and privacy of data transmission, SSL [7]/TLS [6] (Secure Sockets Layer/Transport Layer Security) has been widely applied in the encryption transmission in many aspects(such as e-banking, email, VPN), especially in web security. Because of the encryption by using SSL/TLS, the data transmitted over the Internet are invisible, thus is difficult for network traffic classification and identification. Traffic identification is the significant part for network management, quality of service and network security. The identification of traffic can distinguish the application types of the packets so as to better manage the network, allow or deny bad packets. However, existed methods such as port-based method only identify SSL/TLS packets, but is difficult to identify the concrete types of SSL/TLS packets(For example, Google email type or some companies' VPN types). Even though most enterprises utilize traditional DPI(Deep Packet Inspection) method to achieve identification of SSL/TLS packets, it is not easy to identify non-transparency payload. Besides, brute decryption needs more cost and may also violate the privacy protection.

In view of the above facters, in this paper, we proposed a fine-grained method to identify the SSL application types accurately and fast without touching encrypting information. This approach will be an essential basis for network audit, network management and network security, even for packets label. The main contributions are as follows:

1) To distinguish SSL/TLS packets accurately, we proposed a fine-grained method through fingerprint extraction and matching during the handshake phase.

2) Two hash tables are built so as to fast distinguish packets followed the identified packets in the same flow or among different flows.

The rest of the paper is organized as follows. Section 2 reviews SSL/TLS principle and the related work. Section 3 shows four modules in this methodology and introduces the fine-grained identification algorithm. Section 4 shows the experimental results and discusses the performance of this methodology. Section 5 concludes the whole paper.

2 Related Work

Port-based method is a traditional way to identify the application of packets through recognizing port numbers. These port numbers are usually well-known port numbers registered in IANA (Internet Assigned Numbers Authority) [5] and can be identified by comparing with the records stored in IANA. For example, normally HTTP packets are transmitted by port number 80 and ftp uses port number 21 to transmit data. Port-based method is simple and easy to realize, but not reliable. More flows may not use well-known port numbers to perform data transmission, and some kind of flows may use dynamic port numbers to establish conversations such as P2P flows [1, 3, 11]. Even, to avoid the firewalls certain flows hide their port numbers. All these above make port-based method unreliable.

DPI(Deep Packet Inspection) is a method by inspecting the content of packets payloads and find out the fingerprint information to determine the application types of flows. It is widely applied in companies and has been proved accurate and reliable [4, 12]. However, traditional DPI methods only perform coarse identification for TLS/SSL [3]. Because the payloads of the packets are not transparent and clear after encryption. If using brute force techniques, it will be costly and may sometimes violate the private laws. So traditional DPI is difficult to realize fine-grained identification for TLS/SSL.

Different from DPI, the method based on statistical characteristics of Internet traffic do not inspect the content of packets and mainly foucus on extracting the statistical features of packets or flows [2, 8, 10, 13, 14]. Commonly, these features describe the characteristics of packets behaviors or flow behaviors. They can be packets size, packets intervals or durations of flows, *et al.* It works when facing TLS/SSL packets because it can utilize the statistics to roughly distinguish SSL/TLS packets, but fine-grained identification is also a tough task.

In this paper, we propose a fine-grained identification methodology, which can identify the types of SSL packets accurately and avoid the privacy problem. And through the establishment of two hash tables, the speed of identification is also improved.

3 SSL/TLS Background

3.1 SSL/TLS Overview

SSL is a protocol above TCP layer that utilizes encryption technology to guarantee the security of transmitted data and avoid hijack by the third party. TLS is the subsequent protocol following SSL 3.0 (the latest version of SSL). They are all used to protect the security and integrity of the data. SSL/TLS layer is based on TCP/IP structure, which can be clearly seen in Figure 1.

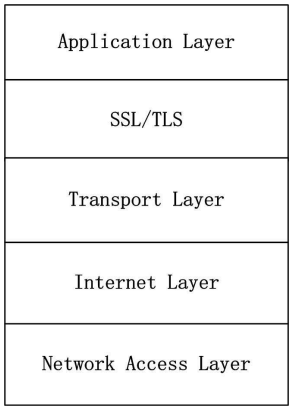

Figure 1: SSL/TLS layer

From Figure 1 we can learn that, SSL/TLS works over the transport layer, and the protocol information can be analyzed and extracted above this layer. SSL/TLS includes two phases: Handshake phase and application data transmission phase. Handshake phase starts before application data transmission phase to validate the identity of two communication endpoints, negotiate encryption algorithms and exchange keys. The data transmitted in handshake phase is transparent and easily identified and extracted.

In this paper, the identification module is performed in this phase and the process of handshake phase will be showed in the next part.

3.2 Handshake

The process of handshake phase can be seen in Figure 2.

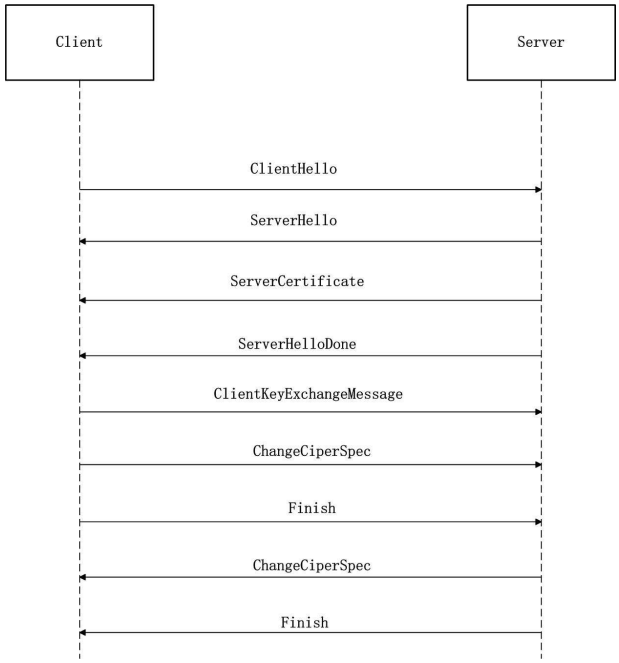

Figure 2: Process of handshake phase

In the handshake phase, there are commonly the following steps to finish handshake session which can be seen in Figure 2:

Step 1. The client firstly sends ClientHello message to start handshake session.

Step 2. The server sends to ServerHello to respond the client. Then the server will send X.509 certificate [9] to the client. And a ServerKeyExchange message and CertificateRequest message may be sent in some cases. Then the client will send ServerHelloDone message to finish the hello phase.

Step 3. The client sends the ClientKeyExchange message to the server and immediately follows the ChangeCiperSpec. Then the client sends finish messeage to complete the handshake session.

Step 4. The server sends ChangeCiperSpec message to the client.At this time,the handshake session is finished. After that the application data begin to be transmitted.

From Figure 2, we noticed that the second SSL/TLS packet from server to client, the server sends the ServerCertificate message to the client. This message includes X.509 certificate where the fingerprint information can be extracted, transformed as features and used to identify the types of TLS/SSL packets. The fingerprint information can be the organization of the application publisher, the department of the application publisher and the purpose of application and so on.

4 The Proposed Method

4.1 The Model of The Proposed Method

Different from most other network flows, SSL flows begin to transmit encrypted data after communication tunnel established in handshake phase. For SSL flows, the packet payloads are transparent in this stage which belongs to the early stage in the whole communication. In this paper, the identification is performed in this early stage and utilizing transparent information in ServerCertificate message and cached hash tables to achieve fine-grained and fast identification.

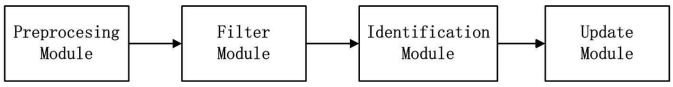

Figure 3: Four modules of fine-graind identification method

The proposed model is mainly composed of four modules: Preprocessing module, filter module, identification module and update module as seen in Figure 3.

Preprocessing module: Two cached hash tables are built to store hash values of APP ID for flows having been identified. Thus it can be used to fast identify packets with the same APP ID through matching with the items stored in these two tables.

Filter module: Create predefined rules to identify if a flow is TLS/SSL flow or not.

Identification module: Extract fingerprint information from the SeverCertificate message and transformed identification features. A newly flows can be identified through computing the similarity with the features.

Update module: Compute the APP ID of newly identified flows and update to the two hash tables.

4.2 Preprocessing Module

In this module, two hash tables will be built. The first hash table H_1 stores the mapping relationships between application types and corresponding application ID (Identification) numbers , that is $<APPID_1, Type>$. *Type* is application types and $APPID_1$ is the corresponding ID numbers within the same conversation(network flow) which can be computed by five-tuple elements(source IP address, destination Ip address, source port, destination port and transport protocol). All items recorded in H_1 infact indicate the packets in the same flow with the same application types.

H_2 indicates the application of packets from different flows and stores mapping relationships $<APPID_2, Type>$. $APPID_2$ can be computed by four-tuple elements (source IP address, destination Ip address, destination ports and transport protocol) because different flows with the same application types have the same four tuples. The items recorded in H_2 indicate the packets in different flows with the same application types.

H_1 and H_2 are utilized to process oncoming packets. When a packet comes, get $APPID_1$ of the packet and query H_1 to check if there is a matching item in H_1. If the matching result is true, the application type can be directly obtained; If the result is false, query H_2. If there is a matching item in H_2, then the application type can be obtained; if not, perform futher identification in the filter module and identification module.

This preprocessing module can fast the packets distinguishing which have been identified and avoid unnecessary work for them.

4.3 Filter Module

Filter module aims at identifying whether a packet is a SSL packet based on the predefined rule. The predefined rule describes the features of the packets and can be expressed as follows:

$$\begin{aligned} filterRule \quad = \quad & \{dir = 0, count = 1, dstport = 443\} \\ & \&\&\{dir = 0, count = 3, offset = 0, \\ & feature = 0 \times 16\}. \end{aligned}$$

In the formula *dir* represents the transmission direction of the packet. If $dir = 0$, it indicates the data is transmitted from the client to the server. If $dir = 1$, it indicates the data is transmitted from the server to the client. *count* indicates the number of the packet located in the whole flow compared with all the other packets in the same flow. *dstport* is the destination port number and offset is the offset in the payload of this packet, while $feature$ is the fingerprint information of the offset.

So $dir = 0, count = 1, dstport = 443$ refers to the port number of the first packet from the client to the server is 443. $dir = 0, count = 3, offset = 0, feature =$ 0x16 refers to the payload offset of the third packet from the client to the server is 0, and starting from the first byte in the third packet, the fingerprint information is 0x16.

4.4 Identification Module

In this module, the fingerprint information will be extracted from the payload of predetermined packets (PrePks), and compare with the records in the feature library. According to the comparsion result, the specific types of PrePks can be get, and the mapping relationships $<APPID_1,\ Type>$ and $<APPID_2,\ Type>$ will be updated in H_1 and H_2. Then the following packets after PrePks will be directly identified through filter module.

Predetermined packets refers to the first five packet($dir = 1, count = 5$) from the server to the client for exchanging X.509 certificates in the handshake phase. Fingerprint information is the features that can be used to identify the specific application types of packets including $countryName$, $stateOrProvinceName$, $localityName$, $organizationName$, $organizationalUnitName$ and $commonName$.

The fingerprint information and the related location in the packets' payload are stored in the linked list as binary numbers. Then the similarity between linked list and the items recorded in feature library is computed:

$$Sim(F, F_k) = \sum_{j=1}^{n} |f_j - f_{kj}|.$$

In the above formula, F is the linked list(infact a vector including n elements), F_k is the kth record in the feature library(also a vector including n elements), f_j the jth bit of F, f_{kj} is the jth bit of the kth record.

If $Sim(F, F_k) = 0$, the application type of predetermined packets is the corresponding type of the kth feature record.

4.5 Update Module

Based on the identification module, new application types are identified and the corresponding $APPID_1$ and $APPID_2$ are computed, and then new mapping relationships will be come up. Finally, H_1 and H_2 are updated by new mapping relationships. The subsequent packets with the same $APPID_1$ and $APPID_2$ can be fast identified by the latest H_1 or H_2.

4.6 Fine-grained Identification Algorithm

TLS/SSL packets get fast and accurate identification through preprocessing module, filter module and identification module. Here, the concrete identification algorithm will be given in Algorithme 1.

The identification procedure can also be described in Figure 4.

Algorithm 1 fine-grained identification algorithm

```
Require: The packet Pk
Begin
Compute APPID_1 of Pk, get (APPID_1)_pk
for i = 1 to |H_1|
if (APPID_1)_pk == (APPID_1)_i
The application type of Pk is (Type)_i
else Compute APPID_2 of Pk, get (APPID_2)_pk
for j = 1 to |H_2|
if (APPID_2)_pk == (APPID_2)_j
The application type of Pk is (Type)_j
else Perform filterRule
Extract the fingerprint information
Compute the Similarity
Determine the application type
Update the mapping relations to H_1and H_2
End
```

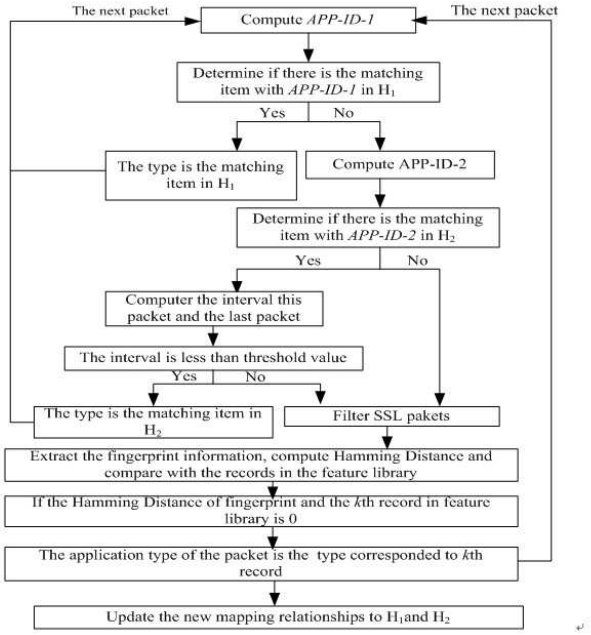

Figure 4: fine-grained identification algorithm

5 Experiment and Analysis

5.1 Dataset

In this paper, to validate this methodology, we captured 300 flows including 10 types TLS/SSL flows including e-mail, e-banking, e-commence, VPN, *et al*, which can be seen in Table 1.

For each type, 30 flows are captured and totally 300 flows are collected. In the whole dataset, 60% of flows are used for model training, while 40% are used for model testing.

Table 1: TLS/SSL traffic types

Application	organizationName	commonName
QQ	Shenzhen Tecent System company	antibot.qq.com
WeChat	Shenzhen Tecent Computer Systems Compan	*.wx.qq.com
VPN	\346\267\261\345\234\263\344\277\241\346	*.sziit.edu.cn
QQ Email	Shenzhen Tecent System company	*.mail.qq.com
Google Browser	Google LLC	*.googleapis.com
Taobao	Alibaba(China) Technology Co., Ltd.	*.taobao.com
Industrial and Commercial Bank of China	Software Development Center	mybank.icbc.com.cn
163 Email	NetEase(Hangzhou)Network Co., Ltd	*.163.com
Xiami Music	Alibaba(China) Technology Co.,Ltd.	*.xiami.net
Youku	Alibaba(China) Technology Co., Ltd.	*.youku.com

5.2 Fingerprint Extraction and Feature Representation

The fingerprint information is extracted from X.509 certificate, and includes: $countryName$, $stateOrProvinceName$, $localityName$, $organizationName$, $organizationalUnitName$ and $commonName$.

All these fingerprint information are transformed to Hexadecimal numbers as features to identify the application types. For example, if the application is 163 mail, the fingerprint information are:

$countryName = CN$
$stateOrProvinceName = Zhejiang$
$localityName = Hangzhou$
$organizationName = NetEase(Hangzhou)$
$NetworkCo., Ltd$
$organizationalUnitName = MAILDept.$
$commonName = *.mail.163.com.$

The corresponding hexadecimal numbers of fingerprint information can be seen in Figure 5.

```
31 0b 30 09 06 03 55 04  06 13 02 43 4e 31 11 30
0f 06 03 55 04 08 13 08  5a 68 65 6a 69 61 6e 67
31 11 30 0f 06 03 55 04  07 13 08 48 61 6e 67 7a
68 6f 75 31 2c 30 2a 06  03 55 04 0a 13 23 4e 65
74 45 61 73 65 20 28 48  61 6e 67 7a 68 6f 75 29
20 4e 65 74 77 6f 72 6b  20 43 6f 2e 2c 20 4c 74
64 31 13 30 11 06 03 55  04 0b 13 0a 4d 41 49 4c
20 44 65 70 74 2e 31 17  30 15 06 03 55 04 03 0c
0e 2a 2e 6d 61 69 6c 2e  31 36 33 2e 63 6f 6d
```

Figure 5: Hexadecimal numbers

Then through matching to the feature library to judge if the application packets belongs to 163 mail application.

5.3 Evaluation

For evaluating the method, four items are involved: True Positive (TP),True Negative(TN),False Positive(FP) and False Negative(FN). They are usually used to evaluate the results of traffic identification.

- True Positive: The type of the flow is X, and the prediction result is also X.
- True Negative: The type of the flow is not X, and the prediction result is not X.
- False Positive: The type of the flow is not X, and the prediction result is X.
- False Negative: The type of the flow is X, and the prediction result is not X.

The accuracy is the most useful indicator values, which can evaluate the good or bad of the identification method. It can be calculated as follows:

$$Accuracy = (TP + TN)/(TP + FP + TN + FN).$$

Through fine-grained identification algorithm, we got 98.4% result which show good performance. It has been proved this method is very effective to identify the specific types of SSL/TLS packets.

After the identification module, the APP ID will be recorded in the two hash tables. hash tables are recorded as $< index, value >$. If the application is 163 mail, $h(x)$ is hash function, $binary(x)$ is the binary transform function, in hashtable h_1,

$x = \{192.168.21.160, 163.177.151, 110, 58473, 443, 6\}$
$index = binary(x)$
$value = 5$
In hashtable h_2,
$x = 192.168.21.160, 163.177.151, 110, 443, 6$
$index = binary(x)$
$value = 5$

"5" represents the value of 163 mail type. Other packets with the same APP ID can be fast identified through hash computing.

5.4 Compasion with Traditional Methods

Compared with traditional methods, our method realized a more concrete and fine-grained distinguishment for TLS/SSL packets, which help better control and manage network. Besides, because of the fast running of hash

function and search, it only takes tens of milliseconds for hash computation. The building of two hash tables reduces redundant work and speed up the identification work. All these help improve the accuracy and increase the efficiency.

6 Conclusions

In this paper, for TLS/SSL network flows identification, the fingerprint information in handshake phase is transparent, and easy extracted to distinguish different types of TLS/SSL flows. It also avoids violating privacy laws. Besides, two hash tables are built and network flows with the same APP ID can be fast identified by comparing with the items in hash tables. Finally, 300 flows for 10 TLS/SSL types are captured and the experiment shows the proposed method can achieve fine-grained and fast identification for TLS/SSL traffic.

Acknowledgments

This study was supported by Natural Science Foundation of Guangdong Province (2018A030310664, 2018A0303130055), Shenzhen Educational Science Planning Project (Grant No. ybfz18279), Fundamental Research Project in the Science and Technology Plan of Shenzhen (Grant No. JCYJ20160527101106061).

References

[1] N. Basher, A. Mahanti, and A. Mahanti,*et al.*, "A comparative analysis of web and peer-to-peer traffic," in *Proceedings of the 17th International Conference on World Wide Web*, pp. 287–296, Apr. 2008.

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," in *IEEE Communications Surveys & Tutorials*, pp. 1153–1176, 2016.

[3] Z. G. Cao, G. Xiong, and Y. Zhao,*et al.*, "A survey on encrypted traffic classification," in *International Conference on Applications and Techniques in Information Security*, pp. 73-81, 2014.

[4] T. Choi, C. Kim, and S. Yoon, *et al.*, "Content-aware internet application traffic measurement and analysis," in *Network Operations and Management Symposium*, pp. 511–524, Apr. 2004.

[5] M. Cotton, L. Eggert, and J. Touch,*et al.*, *Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry*, RFC 6335, 2011.

[6] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246, 2008.

[7] A. Freier, P. Karlton, and P. Kocher, *The Secure Sockets Layer Protocol Vers. 3.0*, RFC6101, 2011.

[8] M. A. Guvensan B. Yamansavascilar and A. G. Yavuz,*et al.*, "Application identification via network traffic classification," in *International Conference on Computing, Networking and Communications*, pp. 843–848, 2017.

[9] R. Housley, W. Polk, and W. Ford,*et al.*, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 3280, Apr. 2002.

[10] A. H. Laskari I. Sharafaldin and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *International Conference on Information Systems Security and Privacy*, pp. 108–116, 2018.

[11] A. Madhukar and C. L. Williamson, "A longitudinal study of P2P traffic classification," in *Modeling, Analysis, and Simulation on Computer and Telecommunication Systems*, pp. 179–188, Oct. 2006.

[12] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," *Passive and Active Network Measurement*, pp. 41–54, 2005.

[13] A. Proto, L. A. Alexandre, and M. L. Batista,*et al.*, "Statistical model applied to netflow for network intrusion detection," in *Transactions on Computational Science XI*, pp. 179–191, 2010.

[14] X. Yu M. Shafiq and A. A. Laghari,*et al.*, "Network traffic classification techniques and comparative analysis using machine learning algorithms," in *IEEE International Conference on Computer and Communications*, pp. 2451–2455, 2016.

Biography

Lingjing Kong received PhD degree from Southwest Jiaotong University, Sichuan, China, 2015. During 2013 to 2014, she also joined in University of Adelaide as a joint PhD student. She is currently a lecturer in Shenzhen Institute of Information Technology, Shenzhen, China. Her research area is network data analysis and machine learning.

Ying Zhou received PhD degree from Sun Yat-sen University, Guangzhou, China, 2014. In 2014, she joined Shenzhen Institute of Information Technology,Shenzhen, China. Her current research interests include local search algorithms and their applications, multiobjective optimization and other evolutionary computation techniques.

Guowei Huang received PhD degree from Nankai University, Tianjin, China, 2009. He is currently a associate professor in Shenzhen Institute of Information Technology, Shenzhen, China. His research area is distributed computer system and streaming media.

Huijing Wang received PhD degree from University of Science and Technology of China, An Hui, China, 2006. She is currently a associate professor in Shenzhen Institute of Information Technology, Shenzhen, China. Her research area is image fusion and enhancement.