Network User Action Detection based on PSD Signature through Encrypted Traffic Analysis

Jee-Tae Park
Computer Information and Science
Korea University
Sejong, Korea
pjj5846@korea.ac.kr

Ui-Jun Baek
Computer Information and Science
Korea University
Sejong, Korea
pb1069@korea.ac.kr

Chang-Yui Shin
Defense Agency for Technology and Quality
Daejeon, Korea
superego99@gmail.com

Min-Seong Lee
Computer Information and Science
Korea University
Sejong, Korea
min0764@korea.ac.kr

Jeong-Woo Choi
Computer Information and Science
Korea University
Sejong, Korea
choigoya97@korea.ac.kr

Myung-Sup Kim
Computer Information and Science
Korea University
Sejong, Korea
tmskim@korea.ac.kr

***Abstract*— User action detection through network traffic analysis plays an important role in information security and network management. In previous studies, we proposed a rule-based user behavior detection system, and it showed high detection performance. However, due to the problem of high dependence on SNI information, detection performance was lowered in certain applications. This can significantly degrade detection performance when the SNI signature cannot be defined. In this paper, we additionally use the PSD signatures to solve the problem of high SNI dependence in previous studies. To verify the proposed method, a detection performance comparison experiment is performed for each signature, and the highest detection performance is shown when PSD is applied together.**

***Keywords*— Network Management, Network Security, User Action Detection, Rule-based Traffic Analysis**

I. INTRODUCTION

In recent years, the rapid development of network technology and the exponential growth of users have led to the proliferation of various applications[1-3]. Research has been conducted on network traffic analysis to ensure smooth network services through effective network management and robust network security. Network traffic analysis encompasses various detailed studies, and among them, user action detection plays a crucial role in both network security and management[2-4].

In terms of network management, user action detection contributes to efficient network operation and resource allocation. It involves analyzing user activity and traffic patterns to optimize network performance and resource utilization. By monitoring user activities, such as resource consumption and application usage patterns, network administrators can make informed decisions regarding resource allocation and enhance the overall user experience. Moreover, user action detection assists in the timely detection of potential faults or anomalies within the network, enabling proactive measures to ensure network availability and minimize service disruptions.

In terms of network security, user action detection plays a crucial role in safeguarding valuable information and assets. It involves identifying and preventing malicious activities that can compromise the confidentiality, integrity, and availability of network systems and data. By detecting and responding to malicious activities such as malware propagation, DDoS attacks, and port scanning, user action detection helps fortify the network against various security threats. Additionally, it aids in intrusion detection by identifying unauthorized intrusion attempts, enabling swift defensive measures to mitigate potential risks. Furthermore, user action detection systems are effective in detecting anomalies or deviations from normal behavioral patterns, allowing for the early detection of internal threats, such as insider attacks or infrastructure anomalies.

In our previous study [5], we introduced a rule-based method for detecting user actions. The method exhibited exceptional detection performance, particularly for Microsoft Office 365, and showcased promising results when applied to other applications as well. In our previous study [6], we proposed automatic rule generation to solve the limitations of consuming a lot of time and effort in the manual rule generation process in [5].

However, both of the proposed methods exhibit a significant reliance on SNI (i.e., Server Name Indication). These methods use header and SNI information, which are typically associated with the target action, to generate rules. Nevertheless, header information is seldom used due to the prevalence of dynamic IPs, ports, and protocols, as well as the widespread use of encrypted traffic over TCP port 443. Consequently, the dependency on SNI becomes crucial for action classification. For example, in the SNI signature used to identify login activity in Adobe Creative Cloud [6], occurrences of "cchome.adobe.io" can lead to duplicate detections since it may appear multiple times besides login. Moreover, if the SNI signature cannot be defined, certain actions cannot be detected.

To address this issue of high SNI dependency, this paper proposes a user action detection method employing PSD (Packet Size Distribution). PSD represents the distribution of packet sizes within a network flow, with packet directionality determining the sign. By utilizing PSD, we aim to mitigate the challenges posed by the heavy reliance on SNI and enhance the effectiveness of behavior detection.

This work was supported by the Technology Innovation Program grant funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea) and the Korea Evaluation Institute of Industrial Technology (KEIT) (No. 20008902, Development of SaaS SW Management Platform based on 5Channel Discovery technology for IT Cost Saving)

The contributions of this paper can be summarized as follows:

- We explain the necessity of user behavior detection research in terms of network management and security. In addition, we also present the problem of high dependence on SNI in our previous research [5, 6].
- We additionally employ PSD information to address the limitations identified in previous studies. Our method involves the utilization of three key components: header, SNI, and PSD, to establish rules for detecting user behavior accurately. To validate the effectiveness of our proposed approach, we conduct experiments to compare the performance of individual signatures based on header, SNI, and PSD, as well as the combined method that incorporates all three signatures. We focus on evaluating the detection performance using Microsoft Office 365, as discussed in [5]. In addition, since the proposed method is applied to encrypted traffic, it can also be applied to other studies using encrypted traffic.

The rest of this paper is organized as follows. In Section 2, we will discuss the related work and, describe the proposed system and analysis method in Section 3. In Section 4, we conduct an experiment comparing detection performance by a signature using Microsoft Office 365. Finally, we conclude the paper and outline future research directions in Section 5

II. Related Work

This section presents an overview of related works on user action detection. User action detection has been extensively studied, with most research focusing on learning-based approaches. These studies typically revolve around identifying specific applications and defining their detailed actions, albeit with variations in the chosen applications and behavior definition methods.

Hou et al. [7] defined seven actions for WeChat and conducted experiments using various algorithms to classify these defined actions. The Random Forest algorithm demonstrated the best performance in their study. Coull and Dyer [8] proposed a method for encrypted traffic analysis targeting Apple's instant messaging service. They utilized packet sizes and defined five user actions, including "start typing," "stop typing," "send text," "send attachment," and "read receipt."

Several studies have conducted user action detection across multiple applications. Grolman et al. [9] applied transfer learning to identify user actions, achieving an F1-measure of 0.8 in their experiments conducted on Twitter and Facebook. Conti and Mauro [10] focused on Android encrypted traffic and investigated seven applications, including Gmail, Facebook, Twitter, Tumblr, Dropbox, Google+, and Evernote. They defined various actions for each application and validated their proposed method through traffic collection experiments.

In our previous work [5], we introduced a rule-based method for detecting user action and presented high detection performance for Microsoft Office 365. However, this method exhibited two limitations: a heavy reliance on SNI and the need for manual rule generation, which proved time-consuming and labor-intensive. To address the challenge of manual rule generation, we proposed an automatic rule generation method [6]. We conducted an experiment with Adobe Creative Cloud, and the automatic rule generation method can significantly reduce the rule generation time without degrading the detection performance.

In this paper, we propose a method for detecting user action by utilizing PSD signatures. This method aims to address the problem of high dependency on SNI, which was identified as a limitation in our previous work [5]. To verify the proposed method, we compare the detection performance for each signature.

III. Proposed Method

The entire system of rule-based user action system and rule format is consistent with previous research. In this paper, we focus on the generation of PSD signatures and their application in detecting user action.

PSD is the distribution of packet size values within a flow, represented as an integer vector. It is classified as either + or -, depending on the direction of the packet. We calculate the packet sizes within a flow, excluding the TCP 3-handshake, and starting from the TLS handshake. For instance, in flow *A*, if the 1st packet in the forward direction has a length of 100, the 2nd packet in the backward direction has a length of -100, and the 3rd packet in the forward direction has a length of 30, the PSD of flow *A* would be represented as [100, -100, 30]. PSD can be defined differently according to the number of packets used, and we used five packets in this paper.

Unlike headers and SNI, PSD values are variable. Even for the same flow occurring in two different datasets for the same user action, the packet sizes can vary slightly. Therefore, unlike the approach of defining string-based signatures (i.e., headers, SNI), PSD signatures are defined by setting representative values and threshold values separately.

Fig. 1 presents the flowchart of the PSD signature generation. Initially, we generate the header and SNI signatures for the input traffic. Subsequently, we handle two scenarios for defining PSD signatures based on the presence or absence of an SNI signature. Our analysis of network

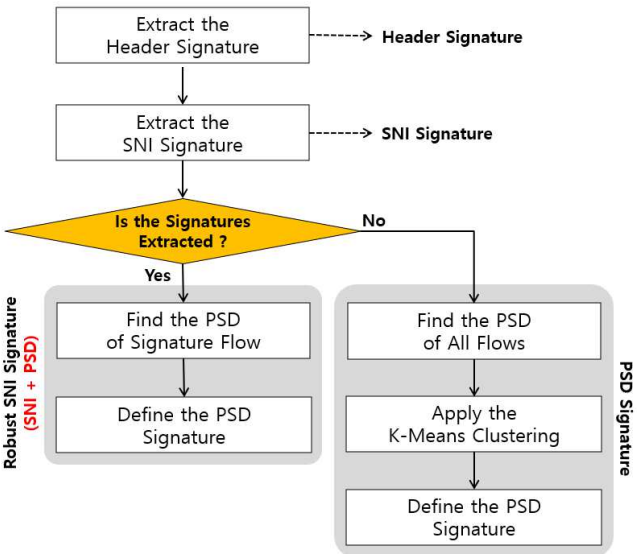

Fig 1. A Flowchart of the PSD Signature Generation

traffic revealed that flows with the same SNI signature across different traces exhibit similar patterns in terms of PSD. For example, when examining flows with the SNI "xxx.com" in two different traffic datasets collected for the same action, the PSD of the two flows may exhibit small differences, such as [10, 20, 30] and [11, 20, 31]. However, there can be cases where it is not possible to define an SNI signature due to the absence of common SNI information in the target action. In such cases, the target action cannot be detected in previous studies. To address this, we consider both of these scenarios and generate PSD signatures. The process of generating PSD signatures is categorized into two scenarios based on the availability of an SNI signature: when an SNI signature is available and when an SNI signature is unavailable. Fig. 2 shows the entire process of PSD signature generation.

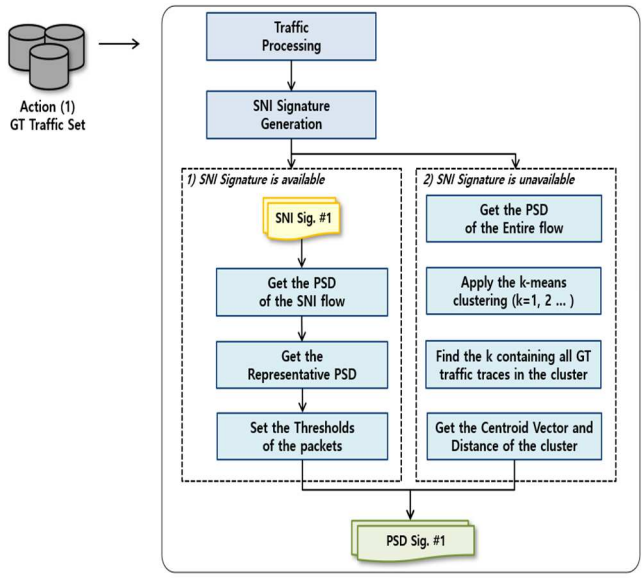

Fig. 2. An Entire Process of PSD Signature Generation

In the case where an SNI signature is available, we calculate the PSD for each flow by incorporating the corresponding signature. This process is applied across multiple collected traces, resulting in various derived PSDs. By computing the average PSD from these derived PSDs, as demonstrated in Equation (1) and (2), we obtain a representative PSD (i.e., Rep. PSD).

$$AvgPS_m = \frac{(\Sigma_{i=1}^{n} p_{i1})}{n} \quad (1)$$

p = packet size, n = the number of traces, m = target packet number

$$Rep\_PSD = \{AvgPS_1, AvgPS_2, \ldots AvgPS_5\} \quad (2)$$

Subsequently, we determine the threshold by analyzing the differences between the average PSD and each individual PSD, as outlined in Equation (3) and (4). Finally, the average PSD and threshold are defined as the PSD signature as shown in (5).

$$Thrd_m = \max\{|AvgPktSize_m - p_m|\} \quad (3)$$

$$Thrd\_Set = \{Thrd_1, Thrd_2, \ldots, Thrd_5\} \quad (4)$$

$$PSD\ Signature = \{\ Rep\_PSD, Thrd\_Set\ \} \quad (5)$$

In the case where an SNI signature is unavailable, we employ a clustering algorithm to derive the PSD signature. Initially, we calculate the PSD for all flows in the collected GT (Ground Truth) traffic. Subsequently, we apply the k-means clustering algorithm to the derived PSD values. By varying the value of k, we identify each cluster and determine the minimum k that encompasses all GT traffic traces within the cluster. Once we obtain the optimal value of k, we define the threshold as the maximum distance between the centroid vector and the cluster. Finally, we define the centroid vector and threshold for each cluster as the PSD signature.

IV. Experiment

A. Experiment Environment

To validate the effectiveness of our proposed method, we conducted an experiment to compare the detection performance of each signature. The rule generation process and experimental environment were kept consistent with the previous study, and Microsoft Office 365 was used as the target application. Table 1 provides information on the dataset utilized in the experiment. Traffic information indicates the number of flows and packets in each trace. For the rule generation phase, we collected 20 traces of GT traffic for each action, while performing all 4 actions in 5 traces for action detection. To evaluate the performance, we employed recall, precision, and f-measure as evaluation metrics.

We compared the performance of four signature methods: i) header, ii) SNI, iii) PSD, and iv) All. Among these methods, 'All' represents the approach that incorporates all three signatures in our proposed method. 'All' is applied in the order of header, SNI, and PSD signature in user action detection.

If a corresponding signature is not in the target behavior, it is considered as a non-detection (FN) since it cannot be detected. For instance, the header signature is only present during the application start of Microsoft Office 365 and is not defined for other actions. In such cases, the performance of the header signature in the login action is evaluated as FN.

TABLE I. A Description of the User Behavior

Rule Generation: Traffic Information			
Application	**Action**	**Flow**	**Packet**
Microsoft Office 365	App. Start	5,523	185,441
	Login	5,778	230,109
	Logout	6,120	100,279
	App. End	4,121	98,118
User Action Detection: Traffic Information			
Application	**Trace**	**Flow**	**Packet**
Microsoft Office 365	#1	1,523	124,449
	#2	1,622	100,290
	#3	3,588	119,622
	#4	2,619	228,166
	#5	2,605	215,259

B. Experiment Result

Table 2 shows the detection performance according to the applied signature. The results show the average recall, precision, and f-measure derived from 5 traces for each signature application method. Since Microsoft connects to a specific site when starting the application, information about the specific site is defined as a header signature. Therefore, the application start of Microsoft Office 365 can be detected with 100% accuracy based on this information.

The header signature is defined only for application startup and is detected with recall and precision of 100% targeting application start, but other actions are not detected. The SNI signature demonstrates recall and precision of 94-100% during the application start, login, and logout, comparable to the performance of the PSD signature. However, it exhibits lower recall and precision values of 77-80% at the application end. This decrease can be attributed to instances of non-detection when the SNI does not match the predefined signature due to variations in the traffic pattern in Microsoft Office 365. The PSD signature achieves recall and precision of 91-100% across 4 actions. In particular, the detection performance for the SNI signature is significantly lowered at the application end, but when the PSD signature is applied, it can be detected with an f-measure of about 95%. In the case of 'All' that utilizing all 3 signatures (header, SNI, and PSD), recall and precision of 97-100% are achieved across the 4 actions. However, it is important to note that the proposed method does not consider the behavior sequence. As a result, duplicate detections can occur when the same signature repeats, even if the target behavior has been detected. These instances of duplicate detection are considered FP.

As a result of the experiment, header, SNI, and PSD signatures show relatively high detection performance with an f-measure of about 90-100%, but there are limitations in individually applying them. The method using 3 signatures proposed in this paper showed overall higher detection performance than the case of applying the signatures individually.

TABLE II. . RESULT OF THE EXPERIMENTS

Actions	Signature Application Method	Detection Result		
		Recall (%)	Precision (%)	F-measure
Application Start	Header	100	100	100
	SNI	100	100	100
	PSD	95.6	90.18	92.81
	All	100	100	100
	Average	98.9	97.54	98.20
Login	Header	-	-	-
	SNI	94.25	96.36	95.29
	PSD	96.17	97.89	97.02
	All	100	98.25	99.12
	Average	96.80	97.5	97.14
Logout	Header	-	-	-
	SNI	96.88	100	98.42
	PSD	97.39	100	98.68
	All	99.28	100	99.63
	Average	97.85	100	98.91
Application End	Header	-	-	-
	SNI	77.29	80.12	78.68
	PSD	91.34	100	95.47
	All	97.68	100	98.83
	Average	88.77	93.37	90.99

V. CONCLUSION

In this paper, we discuss the significance of conducting research on network traffic analysis in the context of network management and security. Furthermore, we emphasize the importance of investigating user action detection as a specific area within this research domain. We provide an overview of related studies, including our previous studies, and refer to the limitations observed in previous research efforts.

In this paper, we additionally use the PSD signature, which is flow statistical information, to solve the problem of high dependence on SNI in previous studies. PSD represents the size distribution of packets in a flow and shows a similar pattern for each action. Based on this, we present two PSD signature generation methods. To verify the proposed method, an experiment is conducted with Microsoft Office 365. The detection performance of the case of using the 3 signatures individually and the case of using all of them were compared. The utilization of all signatures resulted in higher detection performance compared to using individual signatures alone. Furthermore, the high detection performance of the PSD signature compensates for the lower performance of the SNI signature, addressing the issue of high SNI dependency observed in existing methods.

As a future study, we plan to conduct performance comparison experiments for other applications. In addition, we plan to perform additional performance comparison experiments according to the number of packets used for PSD.

REFERENCES

[1] N. F. Huang, G. Y. Jai, H. C. Chao, Y. J. Tzang, and H. Y. Chang, "Application traffic classification at the early stage by characterizing application rounds," Information Sciences, Vol. 232, 2013, pp. 130-142

[2] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, et al., "A Survey on Internet Traffic Identification" IEEE Communications Surveys and Tutorials, Vol. 11, pp. 37-52, 2009

[3] A. Dainotti, A. Pescape and K. Claffy, "Issues and Future Directions in Traffic Classification,", IEEE Network, Vol. 26, no. 1, pp. 35-40, 2012.

[4] J. Zhao, X. Jing, Z. Yan, and W. Pedrycz, "Network traffic classification for data fusion: A survey," Information Fusion, vol. 72, pp. 22–47, Aug. 2021.

[5] J-T Park, M-S Lee, U-J Baek, C-Y Shin, and M-S Kim, "Rule-Based User Behavior Detection System for SaaS Application," in Proc. of the Asia-Pacific Network Operations and Management Symposium (APNOMS), pp.1-4, Sep. 28-30, 2022

[6] J-T Park, U-J Baek, C-Y Shin, M-S Lee, J-W Choi and M-S Kim, "Automatic Rule Generation Method for User Action Detection from Traffic Data", in Proc. of the 13th International Conference on ICT Convergence (ICTC 2022), pp.2037-2040, Oct. 19-21, 2022,

[7] C. Hou, J. Shi, C. Kang, Z. Cao, and X. Gang, "Classifying User Activities in the Encrypted WeChat Traffic," 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC), pp. 1-8, 2018

[8] S. E. Coull and K. P. Dyer, "Traffic analysis of encrypted messaging services: Apple iMessage and beyond," ACM SIGCOMM Computer Communication Review, vol. 44, no. 5, pp. 5–11, Oct. 2014

[9] E. Grolman *et al.,* "Transfer learning for user action identification in mobile apps via encrypted traffic analysis", IEEE Intelligent Systems, vol. 33, no. 2, pp. 40–53, 2018

[10] M. Conti *et al.,* "Analyzing Android encrypted network traffic to identify user actions." IEEE Transactions on Information Forensics and Security, vol. 11, no. 1, pp. 114-125, 2015