Automatic Rule Generation Method for User Action Detection from Traffic Data

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

Chang-Yui Shin
Defense Agency for Technology and Quality
Daejeon, Korea
superego99@dtaq.re.kr

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

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

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

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

***Abstract*—User action detection is an important role in network management and security. To identify and detect user action accurately, much research has been conducted. In this paper, we propose an automatic rule generation method that improves the existing rule generation process. In our previous research, we proposed the rule-based user action detection method that shows high detection accuracy. However, there are several problems in that the rule generation process is too complicated and it takes a lot of time and effort. We conducted several comparative experiments to verify the proposed method with the previous rule generation method. Rule generation time is greatly reduced without performance degradation compared to the previous rule generation method.**

Keywords— Network Management, User Action Detection, Rule based Traffic Analysis, Automatic Rule Generation

I. INTRODUCTION

With the development of network technology and the increase in users, various applications are occurring. Applications are designed to be used by as many users as possible by providing various convenient functions and various contents [1-3]. Various research has been conducted for efficient network management and safe network security, and one of them is user action detection.

User action detection plays an important role in network security and management. In terms of network security, hackers are used to disguising as a normal user in order not to be detected by a firewall or Intrusion Detection System (IDS) [6] [7]. Attackers analyze the vulnerability through target web application analysis in advance, and through this, the attack is performed as a normal user to prevent it from being properly detected in a firewall or IDS.

In terms of network management, administrators need to know user activity for network optimization and resource management. Monitoring application users based on user activity is useful for user privacy enhancement and network service provision.

Recently, many cloud-based application services such as SaaS (Software as a Service) are being widely used. SaaS services are provided in the type of subscription, unlike other applications. For subscription-type services, the cost to be spent varies according to the period of use, license information, and the number of users. Therefore, administrators of enterprise using the SaaS services should detect the user action to prevent excessive spending.

User action detection has been conducted for various applications in many researches [8-12]. In the previous research, we proposed a rule-based user action detection method for the SaaS application [12]. We defined detection rules for each application in advance and proposed the user action detection method based on the defined rules. However, the existing rule generation method in which a person manually defines the rule after pre-analysis of the target application needs a lot of time and effort. To solve the problem, we propose a method of automatically generating detection rules. The proposed method simplifies the rule generation procedure through sequence analysis, which is commonly performed during the rule generation process, and enables the automatic generation of rules.

The specific contributions of this paper can be summarized as follows:

- We explain the need for user action detection research and introduce various researches on user action detection including our previous proposed method (i.e. rule-based user action detection).
- We present the problem of our previous proposed method (i.e. it takes a lot of time and effort to generate a rule). Therefore, we propose an automatic rule generation method to solve this problem.
- To verify the validity of the proposed method, we conducted several comparative experiments by using the SaaS application (i.e. Adobe Creative Cloud). We were able to detect with high performance for each user action of the application. In addition, compared with our previous research, spending time can be greatly reduced without performance degradation.

The remainder of this paper is organized as follows. In Section 2, we will discuss the related work and, describe the proposed method in Section 3. The experiments that have been conducted to validate the proposed method using the SaaS traffic (Adobe Creative Cloud) are presented in Section 4. Finally, we conclude the paper and remark the future research in Section 5.

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) and "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-004).

II. Related Work

In this section, we provide an overview of related research on traffic classification and user action detection. We categorize the traffic analysis into application classification, and malware detection and describe its related work After that, we focus on user action detection and describe its related research.

A. Traffic Classification

In the early stage of research that does not use encrypted traffic, the traffic analysis was based on port-based method [2] and deep packet inspection (DPI) method [3]. However, the use of encrypted traffic makes these traditional methods no longer applicable. Encrypted traffic analysis is largely categorized into application classification [4] [5] and malware detection [6] [7].

In [4], Wang et al, proposed an encrypted traffic classification method based on a convolutional neural network. The authors used a one-dimensional convolutional neural network (1D-CNN) to achieve the encrypted traffic classification. In [5], A. S. Iliyasu and H. Deng proposed a semi-supervised learning approach using the Deep Convolutional Generative Adversarial Network (DCGAN) for the encrypted traffic classification. They achieved 89% accuracy for their experiments. The author in [6] proposed an intrusion detection system based on deep learning. They applied several public datasets (e.g. KDDCup 99, NSL-KDD, UNSW-NB15, WSN-DS, Kyoto, and CICIDS 2017) to verify their proposed method. In [7], McLaughlin et al. proposed a novel android malware detection system based on deep neural networks. They achieved good performance for android malware detection compared with other methods.

B. User Action Detection

User action detection has been variously performed as user action identification, inference, detection, and classification in other research. Park and Kim [8] proposed traffic on KakaoTalk, which is used as a messenger. The authors defined 11 actions such as send a message or join the chat room, and classified user actions with an accuracy of 99.7% through the Random Forest and clustering method. Hou Chengshang, et al. [9] analyzed the MMTLS encryption protocol used in WeChat. The authors defined 7 actions of the WeChat application based on their MMTLS protocol analysis such as browsing moments or opening a mini program. They classified these actions by using 5 learning algorithms such as Naïve Bayes, Random Forest, Decision Tree, Logistic Regression, and SVM. Among these algorithms, Random Forest achieved the best performance with a 92.5% F1-score. In [10], Conti et al. proposed a framework to analyze encrypted network traffic and infer the user actions in mobile application. The authors used 7 applications (e.g. Gmail, Facebook, Twitter, Tumblr, Dropbox, Google+, and Evernote) and defined the user actions for each application. They used ML techniques, hierarchical clustering, and Random Forest for user action classification. H. Wu, Q. et al. [11] proposed a method of user action identification on Instagram. The authors categorized the 9 actions and classified 99.8% accuracy by using the SVM machine learning algorithm.

In [12], we proposed a rule-based user action detection method for SaaS applications. We defined user actions to detect and analyzed the target SaaS application in advance. Then, we defined the rules by using the traffic analysis results and detected the user actions based on the defined rules. Although the proposed method presented high accuracy, it is hard to define the rule and spend a lot of time and effort in the process of rule generation. Therefore, in this paper, we propose a method of automatic rule generation to solve the problem.

III. Proposed Method

In this section, we will briefly describe our previous research, and then explain the proposed automatic rule generation method

A. Rule based User Action Detection

User action can be defined in various forms depending on the defining person and target application. In [9], user action is defined according to detailed functions of WeChat messenger (e.g. browsing moments, open mini program, pay to service, pay to a friend, pay to a group, browsing subscription, and advertisement click), and in [10], detailed action (e.g. send a message, post user status, open chats, status button) is defined. to define user action. In our previous study [12], we defined the user action as information for the administrator's network monitoring and expenditure management of subscription services for SaaS applications. We analyzed Office 365 and defined 6 actions: application start, login, feature (e.g. PowerPoint, Word, Excel) start, feature end, logout, and application end. The format of the user action and the defined rule was used in the same way as in our previous research [12].

B. Automatic Rule Generation

We define a set of packets with the same 5-tuples information as a flow. In most cases, a single action generates a set of distinct flows and generated flow set is different for each action. The rule-based method defines the common packet and flow lists occurring in a single action as a user action detection rule.

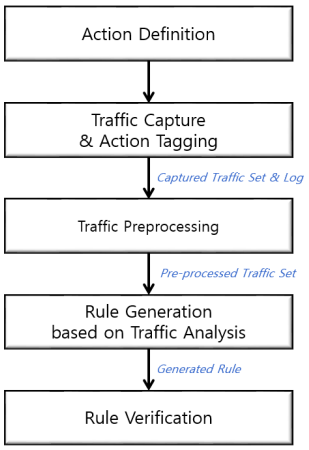

Figure 1. Entire Process of Automatic Rule Generation

To derive a common flow set in a particular action, we prepare the collected traffic sets{S} for a single action of the target application and the action execution time range{TR} of each traffic set. Action execution time range is obtained for the time corresponding to +5 and -5 seconds from the

action execution time{T}. For example, if action X is detected at 12:00:30, action execution time is 12:00:30 and the action execution time range is 12:00:25 to 12:00:35.

The entire process of the automatic rule generation method is shown in Figure 1. First, we define the action of the target application from the user's view. Among the various actions, 10 Traces of traffic set for rule generation and 5 Traces of traffic set for rule verification are collected for each of the actions. Traffic traces are collected by using Wireshark for each defined behavior. Then, the user behavior execution time and traffic set are tagged with each other, and it is saved as a log file.

Traffic preprocessing is shown in Figure 2. In the traffic preprocessing process, a packet-based traffic set is converted into a flow-based traffic set. After traffic preprocessing, we get the action execution time from the log file and obtain the action execution time range for each traffic set. By using the obtained action execution time range and the tagged traffic set, the traffic in the corresponding time range is cut as shown in Figure 2.

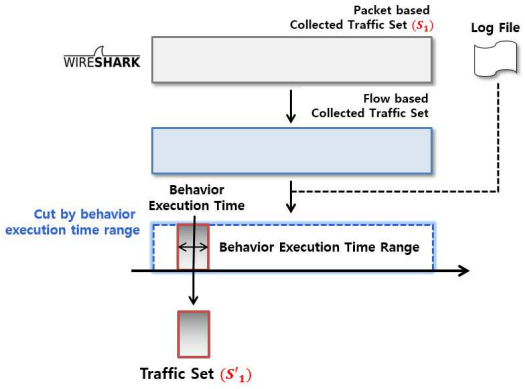

Figure 2. A Process of Traffic Preprocessing

Rule generation is shown in Figure 3. In the rule generation process, the target traffic set is analyzed by inputting the flow-based preprocessed traffic. As a result of the analysis, the flow and packet lists are derived.

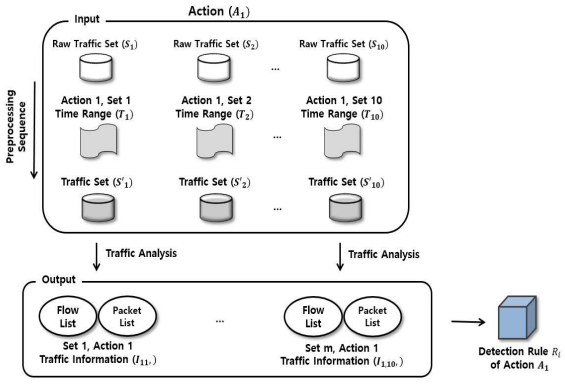

Figure 3. A Process of Rule Generation

The same process is performed on 10 traffic set traces, and flow and packet lists for each trace are derived. Then common flow and packet lists are derived from 10 traces, and the rule is generated using the common flow and packet lists.

In the rule verification process, validation experiments are conducted by using the generated rule. The detection result is derived in the order of user host, target application, usage action, and detection time. We compared the detection result with tagged its action by using 5 verification traffic traces. If the detection time is within the execution time range using the log file tagged in advance, it is defined as a true detection, and if it is not within the execution time range, it is defined as a false detection. In addition, when the detection time and host, target application, and action are inconsistent, it is judged as a false detection. If even one false detection occurs in the 5 verification traffic traces, the generated rule is discarded. The criteria for true detection and false detection are shown in Figure 4.

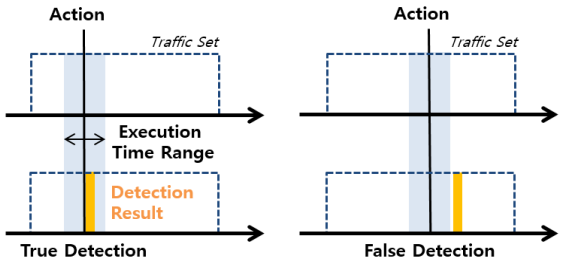

Figure 4. A Criteria for True Detection and False Detection

IV. Experiment

We conducted several user behavior detection experiments using the generated rule. To verify the validity of the proposed method, comparison experiments are performed between the previous rule generation method and the proposed automatic rule generation method. In the two rule-based action detection methods, we compare the rule generation time and detection performance of the generated rules.

A. Experiment Environment

For objective comparison, rules were generated in two ways using Adobe Creative Cloud, which was not used in previous research [12]. Traffic was collected for 5 traces for the comparison experiment of the two methods. The traffic information used in the experiment is shown in Table 1. Traffic information represents the number of flows and packets of each trace, application information represents an application name and user action sequence represents an action sequence performed in each trace. The user action of the target application is defined as Application Start (AS), Login (LI), Logout (LO), and Application End (AE). In the 5 traces, the sequence of user actions was the same in the order of AS-LI-LO-AE, and the traffic collection time and action time interval were set differently. We recorded the application name, user action, user action execution time, and host IP for each trace to compare user action detection results with actual action. As for the evaluation measurement, we used recall, precision, and f-measure for positive detection and false detection.

TABLE I. DESCRIPTION OF THE TRAFFIC DATA

	Traffic Information		Application Information	User Action Sequence
	Flow	Packet		
Trace 1	928	8,911	**[SaaS]** Adobe Creative Cloud	AS–LI–LO–AE
Trace 2	512	3,301	**[SaaS]** Adobe Creative Cloud	AS–LI–LO–AE
Trace 3	612	3,402	**[SaaS]** Adobe Creative Cloud	AS–LI–LO–AE
Trace 4	587	6.850	**[SaaS]** Adobe Creative Cloud	AS–LI–LO–AE
Trace 5	362	7.523	**[SaaS]** Adobe Creative Cloud	AS–LI–LO–AE

B. Experiment Result

The experimental results are shown in Table 2 and 3. Table 2 shows the detection results of the previous rule generation and automatic rule generation method. As shown in Table 2, both rule generation methods show an average of 96~100% recall and precision. By comparing the average recall, precision, and f-measure, It can be seen that the performance is similar.

TABLE II. DETECTION RESULT OF THE EXPERIMENT

		Detection Result		
		Recall (%)	Precision (%)	F-measure
Previous Rule Generation Method [12]	**Trace 1**	99.52	100	99.75
	Trace 2	98.12	100	99.05
	Trace 3	100	95.23	97.55
	Trace 4	95.25	94.10	94.67
	Trace 5	100	96.41	98.17
	Average	98.58	97.148	97.83
Automatic Rule Generation Method	**Trace 1**	95.46	94.11	94.78
	Trace 2	96.44	100	98.18
	Trace 3	97.28	96.01	96.64
	Trace 4	100	98.25	99.11
	Trace 5	98.14	96.40	97.26
	Average	97.46	96.95	97.19

Table 3 shows the time taken for rule generation in the 2 methods. Rule aggregation is the process of merging the rules generated for each action into a rule for one application. In the case of the previous rule generation method, it takes 180 to 200 minutes for each action, and in the case of the automatic rule generation method, it takes 80 to 100 minutes for each action. Comparing the rule generation time of the two methods, the automatic rule generation method can significantly reduce the rule generation time

TABLE III. TIME SPENDING FOR THE RULE GENERATION

	Rule Generation Time (min)	
	Previous Rule Generation Method	Automatic Rule Generation Method
(Action 1) Application Start	188	80
(Action 2) Login	200	95
(Action 3) Logout	191	84
(Action 4) Application End	202	108
Rule Aggregation	30	30
Total	811	397
Average	162.2	79.4

V. CONCLUSION

We performed a user action detection experiment of the target application by using the automatic rule generation method. To verify the proposed method, we conducted comparative experiments between the previous rule generation method and the proposed method. We compared the detection performance and rule generation time of the two methods. The detection performance of both methods yielded an average of 96~100% recall and precision, and the rule generation time was significantly reduced compared to the previous rule generation method. The proposed method was able to significantly reduce the rule creation time without performance degradation. In future research, we will generate and apply the rules for multiple applications as other researches have applied. In addition, the rule generation method and analysis process will be supplemented by analyzing the causes of false positives that occur during comparative experiments.

REFERENCES

[1] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," IEEE Communications Magazine, vol. 57, no. 5, 2019, pp. 76-81.

[2] A. Madhukar and C. Williamson, “A longitudinal study of P2P traffic classification” in Proc. 14th IEEE Int. Symp. Modeling Anal., Simulation, Sep. 2006, pp. 179–188.

[3] M. Finsterbusch, C. Richter, E. Rocha, J. A. Muller, and K. Hanssgen, ‘ ‘A survey of payload-based traffic classification approaches,’’ IEEE Commun. Surveys Tuts., vol. 16, no. 2, 2nd Quart., 2014, pp. 1135–1156.

[4] W. Wang , M. Zhu, J. Wang, X. Zeng and Z. Yang, “End-to-end encrypted traffic classification with one-dimensional convolution neural networks,” 2017 IEEE International Conference on Intelligence and Security Informatics (ISI) IEEE, 2017, pp. 43-48.

[5] A. S. Iliyasu and H. Deng, "Semi-Supervised Encrypted Traffic Classification With Deep Convolutional Generative Adversarial Networks," IEEE Access, vol. 8, 2020, pp. 118-126.

[6] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep learning approach for intelligent intrusion detection system,” IEEE Access, vol. 7, 2019, pp. 41525–41550.

[7] N. McLaughlin, et al. G. Joon Ahn, “Deep android malware detection”, Proceedings of the seventh ACM on conference on data and application security and privacy, 2017, pp. 301–308.

[8] K.-W Park, and H. –S Kim, “Encryption Is Not Enough: Inferring user activities on KakaoTalk with traffic analysis”, in Proc International Workshop on Information Security Applications (WISA), 2015, pp.254-265.

[9] 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), 2018, pp. 1-8 .

[10] Conti, M., Mancini, L. V., Spolaor, R., and Verde, N. V. “Analyzing android encrypted network traffic to identify user actions”. IEEE Transactions on Information Forensics and Security, 11(1), 2015, pp. 114-125.

[11] H. Wu, Q. Wu, G. Cheng, and S. Guo, “Instagram user action identification based on multidimensional features,” in Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS), 2020, pp. 1111–1116.

[12] 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) 2022., in press.