SigManager: Automatic Payload Signature Management System for the Classification of Dynamically Changing Internet Applications

Kyu-Seok Shim[1], Young-Hoon Goo[1], Sungyun Kim[2], Mi-Jung Choi[2] and Myung-Sup Kim[1]
Dept. of Computer and Information Science, Korea University, Sejong, Korea[1]
Department of Computer Science, Kangwon National University, Chuncheon, Korea[2]
{kusuk007, gyh0808, tmskim }@korea.ac.kr[1], {kyun, mjchoi}@kangwon.ac.kr[2]

Abstract— Today's network environment is becoming very complicated. Accordingly, traffic classification for network management becomes difficult. For the study of traffic classification, the development of automatic payload signature generation system was carried out very actively. However, the existing automatic payload signature generation system has problems such as semi-automatic system, disposable signature generation, false-positive signature generation and not up-to-date signature. Therefore, we propose the SigManager. SigManager performs all process such as traffic collection, signature generation, signature management and signature verification. The traffic collection stage automatically collects ground-truth traffic through TMA and TMS. The signature management stage removes unnecessary signatures and the signature generation stage generates the new signatures. Finally, the signature verification stage removes the false-positive signatures. We solved the problem of existing automatic signature generation system through this system. As a result of applying this system to campus network, we could maintain high completeness and low false-positive rate for 4 applications.

Keywords—Traffic Classification, Payload Signature, Sequence Pattern Mining, Automatic Management, Automatic Generation, Network Traffic

I. INTRODUCTION

Today, the most applications utilize network resources. The utilization rate of network resources is increased and the amount of network management target traffic exponentially increases. Applications are increasing the type of service according to user needs. Network administrators can classify traffic for each application. However, it is not possible to classify traffic for each service in the application. Therefore, efficient network management becomes difficult[1,2].

In this paper, we focus on payload signatures in various levels. The payload signature is a unique and continuous substring in payload of the same application traffic. However, in order to generate the payload signature, a lot of time and money is needed. Most of the existing methods generate the payload signature in similar manner. First, a manager gathers traffic of an application to extract the signature. Second, the user finds the substring that commonly occurs while comparing the contents of the payload. After extracting the common substring, the string can uniquely be used to performs the verification of an application. Therefore, depending on the extraction operator, there can be a difference between the quality of the signature, which leads to disadvantages in signature objectivity. Finally, since the signature extraction operation consumes times and necessary frequent work, the signatures of all the applications are either hard or impossible to be updated.

In order to overcome these problems, the studies of automatic payload signature generation have been actively carried out [3-5]. Most studies on the automatic payload signature generation use a method for automatically extracting common substrings from packets payloads. However, there are limitations in these studies. First, the basic condition for generating signatures is that the user needs to collect traffics directly. Second, some of the extracted signature can be a disposable signature because the collected traffic is extracted for a short period. The signature must have a set of signatures that can detect all the functions of that application. The extracted signature in a short period cannot be assumed to be able to detect all the functions of the application. Third, some of the extracted signatures can be false-positive signatures, because they did not pass the signature verification step. Finally, the signatures cannot always be kept as the latest signature. Since it is impossible to recognize when the traffic pattern changes, it is also impossible to response instantly to the changes unless a human does it manually.

Therefore, we propose a fully automated signature update system in order to overcome these limitations. The proposed system automatically performs all processes such as traffic collection, signature management, signature generation and signature verification. It can overcome the disadvantages of the existing automatic signature generation system. Finally, as the proposed system operates continuously, even when the traffic patterns change, there will be an instant response to that change. In section 2, we review the previous work in the traffic classification and the automatic signature generation. In section 3, we propose the fully automatic signature update system. We

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00513) and by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Korea government(MOTIE) (No. NRF-2017R1A2B4010205)

evaluate the proposed system in section 4 and finally advert the conclusion and future work in section 5.

II. Related Work

The payload signature has the high accuracy and coverage. However, the extracting signature is very difficult and time consuming. Therefore, studies of automatic payload signature generation are in the limelight in the field of network management. The existing methods are LASER (LCS-based Application Signature ExtRaction), Autosig, and SigBox.

LASER automatically generates an application signature, in the form of a sequence of substrings, in the payload of packet by using a modified version of the LCS (longest common subsequence) algorithm. The inputs of this algorithm are two distinct byte streams of packet payloads that belong to two flows. In order to improve the system's performance in terms of execution time and accuracy, this method only considers the first N packets of a flow and groups these packets by their size, since large packets are not likely to carry the same kind of information as the small ones. Finally, the method compares two inputs to get the longest common subsequence between them, and then compare it with another subsequence iteratively to refine it.

Autosig also generates an application signature automatically, which extracts multiple common substring sequences from input flows as application signature. First, it divides the payload of a set of flows into short substrings called shingles. After extracting all of the relevant, common shingles, Autosig merges them if they are neighbors or overlap. Next, a substring tree is constructed to create all possible combinations of substrings. These combinations are considered as signatures.

SigBox uses the Apriori algorithm[6] to solve the above disadvantage. The above methods are necessary preprocessing and postprocessing in order to compare two strings. The preprocessing is setting the order of traffic and grouping the traffic. The postprocessing integrates the generated substring into one rule. However, SigBox extracts substring likely to become signatures by increasing the length-1 all the substrings candidates. Therefore, this method does not take much time to the extraction process and does not required preprocessing and postprocessing. In this paper, we extract signature using Sigbox.

III. The Proposed System: SigManager

The automatic payload signature generation system for each service is configured should be as shown in figure 1. The proposed system consists of GT (Ground Truth) traffic generator, signature manager, automatic signature generator and signature verifier. First, GT traffic generator collects ground truth traffic for the application using the TMA (Traffic Measurement Agent) and TMS (Traffic Measurement Server) [7]. Second, the signature manager outputs the ground-truth traffic that is not analyzed by using the total ground-truth traffic and the existing signatures. Also, this step can have signature management. The signature management refers to deletion of unused signatures in existing signatures. If the signature is deleted, it is possible to reduce the overload of the system. Third, signature generation extracts the signature for each service using SigBox the classified traffic as input. Finally, signature verifier verifies the extracted signatures based on accuracy.

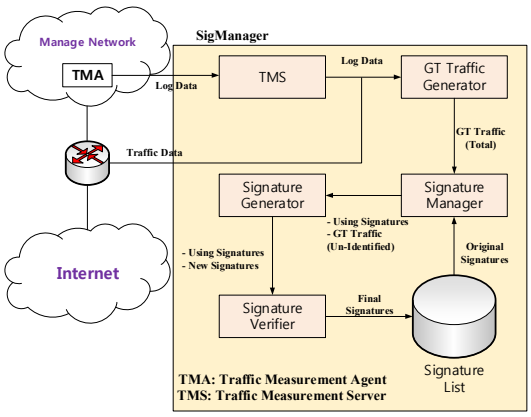

Figure 1. SigManager

A. GT Traffic Generator

One major limitation on current studies is manual collection of traffic. Although GT Traffic Generator automatically collects the ground-truth traffic from each application, in most cases, the collected traffic is incorrect traffic in this process. The proposed system uses the TMA for generating the ground-truth traffic. TMA is installed on each host leaving the log data. Table 1 shows information that includes the log data from TMA.

B. Signature Manager

We generate the signature using the collected ground-truth traffic. However, this method consumes high system overload and a lot of time as it consistently generates the same signatures of a same application. Also, it is necesscary that the management method is applied to unused signatures among the existing signatures. The signatures management stage manages the signatures by classifying unidentified traffic and deleting unused signatures.

The signature consists of header information, contents, and score. Header information consists of source/destination IP address, source/destination port number and L4 protocol. The content is the unique pattern in an applicaion. Score is used for removing disposable signatures. Score represents the number of times used in analysis. Score has an initial value of 1 with the signature generation. If the signature was not used in the analysis then the score is decreased by 1 and if the signature was used in the analysis then the score is increased by 1. When the score is 0, the signature is removed. The reason for using the score in this system is to maintain the normal signatures and to remove the disposable signatures.

In this process, the removed signatures are classified into two types. The first type includes the disposable signature that generates traffic at only specific time. Typical disposable signatures contain the date keyword. This disposable signature is removed since is not used on the date in this process. The second type is the signature that does not incidentally occur in the inputted traffic data. We use the score to prevent such

errors. As a result, this system outputs the signature set that removes disposable signatures and unanalyzed traffic.

C. Automatic Signature Generator

Signature Generator is used to modify the sequential pattern algorithm (AprioriAll) for signature extraction. In the process of automatic signature generation, the system generates sequences for the payload traffic extraction. From the extracted payload traffic, length-1 content signatures are an alphabet. From the extracted length-1 content signatures length-2 content signatures are created with deletion of unwanted content parts. The process continues to length-k until no more content signatures to generate higher lengths in the extraction of common strings [8].

The first type is the content signature that is continuous and common string. The second type is the packet signature combination that is discovered in the same content signatures. The third type is the packet signature combination that is discovered in the same flow. In content signature extraction stage, a minimum support is given to a set of sequences and the sequences that have minimum support are extracted. In this stage, we use the AprioriAll algorithm for extraction of content signature. The content set which is the algorithm's output, is a continuous string of sequence strings.

The same process of increasing length-k under the same minimum support while removing unsatisfying content signature strings, iterates until there are no more content signature strings to be extracted. In the last stage of the extraction, the inclusion relation is checked and the sub-content signature strings are deleted. Finally, the set of generated content sequences are passed on to the packet signature extraction stage. The packet signature extraction stage which is the next stage, is very similar to the content signature extraction stage. In the content signatures extraction process when content sequences are composed, instead of using payload traffic string the extracted content signature will be used if they were composed by the sequence strings of the payload traffic instead of using payload traffic string. Figure 2, shows the packet sequences extraction stage.

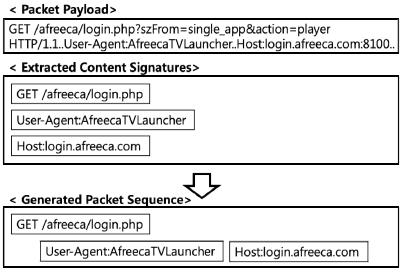

Figure 2. Packet Sequence Extraction

Similar to figure 3, content signatures occurring in same packets are extracted and the process iterates until no more possible extraction can be done. When the packet signatures are extracted the inclusion relation is checked, in which the included content signature subsets are deleted. Finally, the extracted packet signature set is passed on to the flow level signature extraction stage.

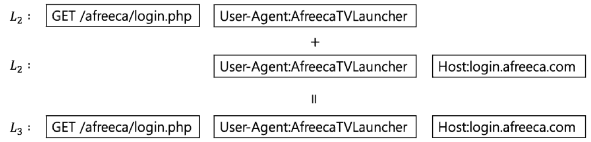

Figure 3. Packet Signature Extraction

As indicated below, not only the continuous strings that are commonly discovered in the content signatures but also packet signatures and flow signatures generation are needed for minimizing false positive rates. Simply, the content signature has a high coverage rate, but the false-positive rate is also high. In comparison, the packet signature is more accurate than content signature and the flow signature is more accurate than the packet signature. Through this process false positives are highly reduced in signature generation.

D. Signature Verifier

The proposed system includes the verification step for extracted signatures in the process. This system generates flow signatures with low false-positive. This step is an essential step because a flow signature might have possible false-positives. The signature loses its meaning at the moment of clssification another application, not a specific application. The signature verification is targeted to the signature analyzing other applications that is, False-Positive signature. Signatures without the false-positive are included in the final signature. In the verification process, it will use the false-positive value.

F-measure is a formula that measures the value of a given weight of the precision and recall. In this paper, we use the F-measure value computed from the precision and recall. If the weight is added to the precision, value gradually decreases to 1. If the weight is added to the recall, B value gradually increases to 1. If we give the same weight it is to secure the 1 value. In this paper, we use a fixed B value for the F-measure and use a weight of 0.1 for the precision. This value was set to place more weight on precision. The maximum F-measure value is 1 and the minumum F-measure value is 0. We only use the signature of F-measure with a minimum of 0.95 to a final signature. Through the proposed method the signature is updated continuously and an unused signature is removed and a new signature is extracted. A new signature maintains high accuracy during the verification process [9].

IV. EVALUATION

This chapter evaluates the performance of the SigManager. We chose the 4 most frequently used applications for experimatation and evalution of the application from automatically collected traffic. We collected groud-truth traffic after installing the TMA on 8 hosts using traffic from 8 hosts and TMA log data. The four applications include AfreecaTV for video and broadcast services, Facebook for social network services, Kakaotalk for messaging services, and uTorrent for file sharing and transfer services.

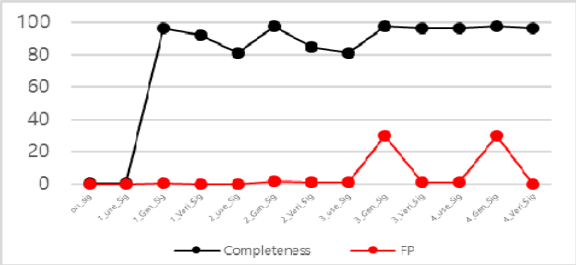

Figure 4. The completeness and FP of AfreecaTV

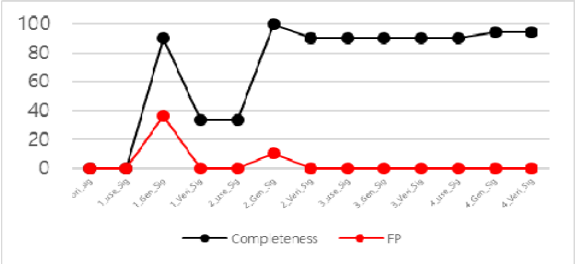

Figure 5. The completeness and FP of Facebook

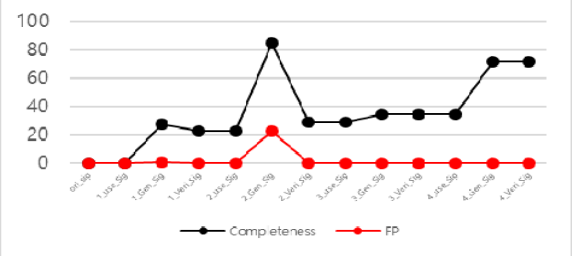

Figure 6. The completeness and FP of Kakaotalk

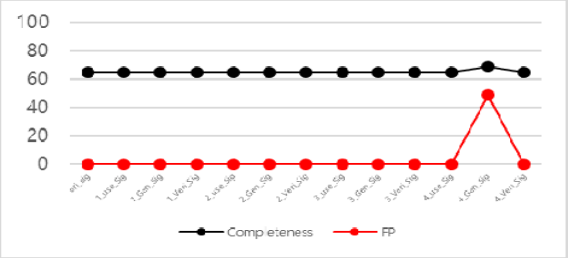

Figure 7. The completeness and FP of uTorrent

Figure 4,5,6 and 7 shows the completeness and false-positive rate of the respective application in SigManager process. When using the signatures from original signatures, the number of signatures are decreased but the completeness does not show a big difference. For example, there are 140 original signatures of AfreecaTV and 2 using signatures of AfreecaTV, but the completeness is the same at 0.9%. This result was confirmed from the disposable signatures extracted in the signature generation stage. We confirmed the increase of the number of signatures, completeness and false-positive when extracting the new signatures. Through this result, this system is not only extracting the normal signatures but also extracting the false-positive signatures. This proves that the verification process is essential for extracting the final signatures. Then we confirmed the false-positive had a major decrease. For example, false-positive of new signatures of Facebook is 31.2%. After removing false-positive signatures, false-positive of final signatures of Facebook is 0.16%. This figure decreased by 31.04%. Therefore, the signatures verification stage deletes the signatures that are possibly false-positives.

V. Conclusion

In this paper, we proposed the SigManager which performs ground-truth traffic collection, signature management, signature generation and signature verification. This system solves the problems of the existing systems that manually or semi-automatically collect traffic. In addition, the limitation of extracting disposable signature was solved by selecting only the signature used in the signature management. The problem of extracting false-positive signature was solved in signature verification stage. In this paper, the proposed system was cumulative only to normal signatures and deleted the disposable signatures and false-positive signatures. Finally, this system increased the completeness and decreased the false-positive rates as the number of times of execution increases.

In future work, we will adopt other methods of collecting ground-truth traffic different from TMA and TMS. TMA method is possible to classify ground-truth traffic that is necessary for TMA to install to each host. We also plan to apply more optimized algorithms to improve the speed of the current system.

References

[1] M.-S. Kim, Y. J. Won, and J. W.-K. Hong, "Application-level traffic monitoring and an analysis on IP networks," ETRI journal, vol. 27, pp. 22-42, 2005.

[2] B. Park, Y. Won, J. Chung, M. S. Kim, and J. W. K. Hong, "Fine-grained traffic classification based on functional separation," International Journal of Network Management, vol. 23, pp. 350-381, Sep 2013.

[3] B.-C. Park, Y. J. Won, M.-S. Kim, and J. W. Hong, "Towards automated application signature generation for traffic identification," in Network Operations and Management Symposium, 2008. NOMS 2008. IEEE, 2008, pp. 160-167.

[4] X. Feng, X. Huang, X. Tian, and Y. Ma, "Automatic traffic signature extraction based on Smith-waterman algorithm for traffic classification," in Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on, 2010, pp. 154-158.

[5] H.-A. Kim and B. Karp, "Autograph: Toward Automated, Distributed Worm Signature Detection," in USENIX security symposium, 2004.

[6] R. Agrawal and R. Srikant, "Mining sequential patterns," in Data Engineering, 1995. Proceedings of the Eleventh International Conference on, 1995, pp. 3-14.

[7] S.H.Yoon, H.G.No, M.S.Kim, "Internet Application Traffic Classification using the TMA(Traffic Measurement Agent)", 29th KIPS, Daegu, Kyungll University, May. 17, 2008, Vol.15, No.1, pp.946-949.

[8] K.S.Shim, S.H.Yoon, S.K.Lee, S.M.Kim, W.S.Jung, M.S.Kim, " Automatic Generation of Snort Content Rule for Network Traffic Analysis" , KICS, Vol.40, No.04, April, 2015, pp666-677.

[9] D.M.POWERS, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.", Journal of Machine Learning Technologies, Dec, 2011