# Inference of Network Unknown Protocol Structure using CSP(Contiguous Sequence Pattern) Algorithm based on Tree Structure

Kyu-Seok Shim, Young-Hoon Goo, Min-Seob Lee, Huru Hasanova and Myung-Sup Kim
Dept. of Computer and Information Science
Korea University
Sejong, Korea
{kusuk007, gyh0808, chenlima2, hhuru, tmskim}@korea.ac.kr

***Abstract*—As Internet traffic generation grows and new applications and malicious acts continue to emerge, traffic to be analyzed is growing rapidly. Most network security threat traffic is communicated using unknown protocol. Thus, protocol reverse engineering is very important to address network security issues. While various protocol reverse engineering methods have been studied, there is no single standardized method to extract protocol specification completely yet, and each of methods has some limitations. This paper proposes to extract the static fields of the protocol. The method uses CSP algorithm based on Apriori to extract the common strings. However, we propose the method of extraction of a protocol static field using the CSP algorithm based on the tree structure because it is not possible to extract all static fields with only CSP algorithm. This method allows extraction of all static fields that are infrequent but possible, not just frequently occurring. This method has been validated by experiments with HTTP protocol.**

***Keywords—protocol reverse engineering; Field Format; Message Format; Flow Format; State Machine; CSP Algorithm;***

## I. Introduction

As Internet traffic generation grows and new applications and malicious acts continue to emerge, traffic to be analyzed is growing rapidly. Many of the various protocols that arise under this environment are unknown or at least documented in private. Therefore, techniques for efficiently processing and analyzing a wide variety of different data are required, and the technologies for closed protocol architecture analysis must be preceded for effective network management and security [1].

Traffic analysis is the first fundamental task to be undertaken to establish a fair wired and wireless integrated Internet ecosystem and to establish policies for the smooth communication of networks. In addition, architectural analysis techniques for non-closure protocols are essential to enhancing network management performance. Also, as daily life relies upon the cyber space based on information and communication technologies, cyber terrorism is emerging as a new global shared concern, and is an urgent period for the securing of infrastructure that is used to secure the technologies.

Protocol reverse engineering[2-7] can be used to benefit a wide range of network security. In the field of network monitoring, information can be obtained about the unknown traffic originating in the target network, so the traffic generated by the normal application is classified as flow. It can be utilized QoS (Quality of Service) policy settings such as to determine network usage, establish expansion plans, and bandwidth control from classification of traffic caused by unknown protocol in small flows.

The current traditional protocol reverse engineering approach is inefficient by an analyst who manually identifies network messages and extracts a protocol architecture. As a result, rapid response to today's high-speed, high-capacity network environments and a range of highly intelligent, malicious acts requires automatic protocol reverse engineering technology.

This paper proposes to extract the field format for protocol reverse engineering. The proposed method uses the CSP (Contiguous Sequence Pattern) algorithm to separate the Internet traffic on a message-by-message basis and find the static fields that are common in each message[8-9]. Additionally, a tree structure is used to extract fields with low probability. Figure 1 shows the concept of unknown protocol architecture analysis.

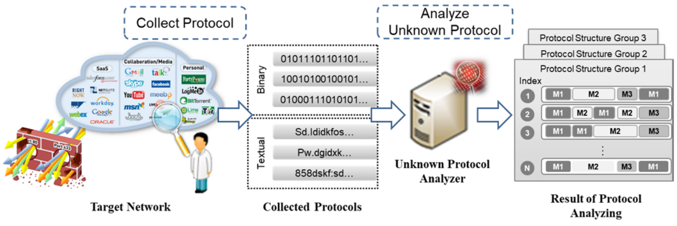

**Figure 1. Unknown protocol architecture analysis**

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2015R1D1A3A01018057) and by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00513)

978-1-5386-3416-5/18/$31.00 ©2018 IEEE

This paper refers to related work in section 2, following the introduction of this section. Section 3 describes the structure of a closed protocol analysis using the proposed method. Section 4 describes the static fields extraction system using the proposed system. After drawing an experiment result from section 5 and referring to conclusions and future studies in section 6, this paper concludes.

## II. RELATED WORK

This section refers to the work related to the protocol static field extraction system using the CSP (Contiguous Sequence Pattern) algorithm based on tree structure proposed in this paper. First, we explain how to study the unknown protocol and the key study in the unknown protocol study. In addition, CSP algorithm is explained and then discusses the difference between this method and the CSP algorithm method.

Although many existing studies present and verify methodologies for analyzing the private protocol architecture, there are no currently standardized methodologies available, and they exist for each of the various, complicated, large networks today. This section describes the criteria for classification of unknown protocol structures and describes the detailed mechanism definitions and differences in the methodologies by the classification criteria described. Based on this, the existing academic studies explain the current status of technologies used in prior methods.

However, passive, private protocol architecture analysis can accurately recover all protocol elements, but it is error prone, time-consuming, and manual, and therefore would not be able to cope with exponential growth in application. The SAMBA project took 12 years to create the Microsoft Server Message Block (SMB) protocol, and for the WINE (Wine Is Not an Emulator) project, the first API to be stabilized was launched in 15 years.

The CSP algorithm is one of the modified sequential pattern algorithms that are suitable for the protocol syntax extraction. The original version of the sequential pattern algorithm looks for sequential buying patterns based on the market's purchase history data. However, the protocol syntax requires the extraction of a continuously connected partial sequence within a message rather than simply a series-series sequence occurring from the message. For example, to extract a value in static field, we need to extract a string of characters that are connected (byte stream) in succession, not the set of characters that appears in the message. Therefore, the purpose of the CSP algorithm is to extract the sequential pattern that is tangent to it consecutively.

The algorithm is based on the Apriori attribute, which says " sequence subsets that occur frequently, and sequence subsets that do not occur frequently ". The basic apriori-generation algorithm accesses each level with a Length-1 sequence Length-2 sequence, …, and Length-k sequence, and determines the number of candidate sequences (Ck) that occur during each Level and determines the candidate's sequence. The CSP algorithm integrates a revised version of these basic Apriori algorithms, AprioriAll, AprioriTID, AprioriHash, and improves performance through modifications.

This paper proposed the method of extraction of a protocol static field using the CSP algorithm based on the tree structure. This method is included in the Syntax Inference. This method has the benefit of being able to extract even Field that is not satisfied with support values if it belongs to the syntax of the protocols, compared to the existing CSP algorithms alone. In addition, when there are strings such as " Content - Length " and " Content - Type", the existing CSP is only extracted with "Content -". On the other hand, the proposed method has the benefit of being extracted as "Content -", "Length" and "Type".

## III. PROTOCOL REVERSE ENGNEERING SYSTEM STRUCTURE

The CSP algorithm needs to be classified for network traffic to inference the unknown protocol architecture. Traffic is transmitted on a per-packet basis, and packets with the same source/destination IP, source/destination port, and protocol can be grouped into flow. However, it is difficult to extract static fields from the protocols by packets or flow, this study defines messages.

The protocol reverse engineering system consists of three steps. The first is the message assemble step of reconfiguring the traffic trace composed of packets per message. The second is to extract a static field using extracted messages. In this step, the method of CSP (Contiguous Sequence Pattern) algorithm based on tree structure is used. The third is the step of composing the message using a static field. The following process can inference the structure of an unknown protocol. Figure 2 shows this process.

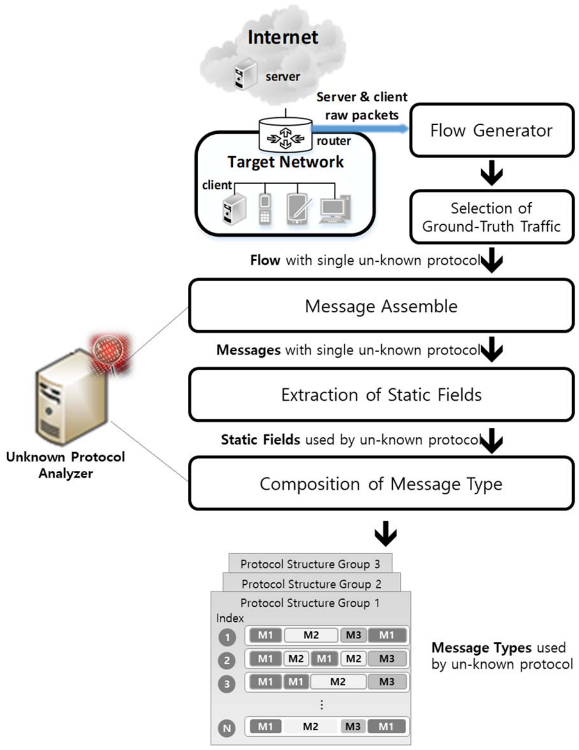
Internet
server
Server & client raw packets
router
Target Network
client
Flow Generator
Selection of Ground-Truth Traffic
Flow with single un-known protocol
Message Assemble
Messages with single un-known protocol
Extraction of Static Fields
Static Fields used by un-known protocol
Composition of Message Type
Unknown Protocol Analyzer
Protocol Structure Group 3
Protocol Structure Group 2
Protocol Structure Group 1
Index
1
2
3
N
M1
M2
M3
Message Types used by un-known protocol

**Figure 2. Analyze of unknown protocol Overview**

First, the step to define messages are defined as Message Assemble. When communicating with the two hosts, the data are transmitted and received per packet and must be reassembled into message units, which are application tier level data units (ADU). This study defined the methodology for splitting the packets of each flow into message units. For protocols using UDP as the transport layer protocol, the message unit is defined as one packet, and for protocols using TCP as the transport layer protocol, the message unit is a set of packets that are defined sequentially in the same direction. Figure 3 shows the message code representation of Message Assemble.

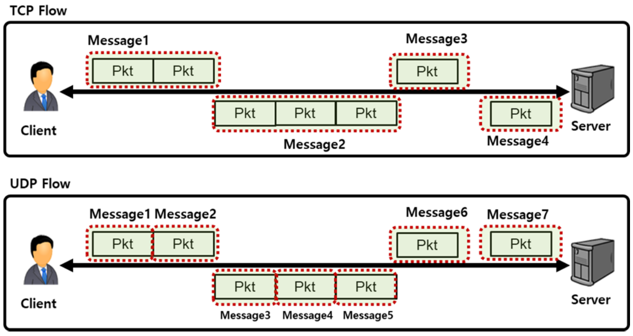

**Figure 3. Message Assemble**

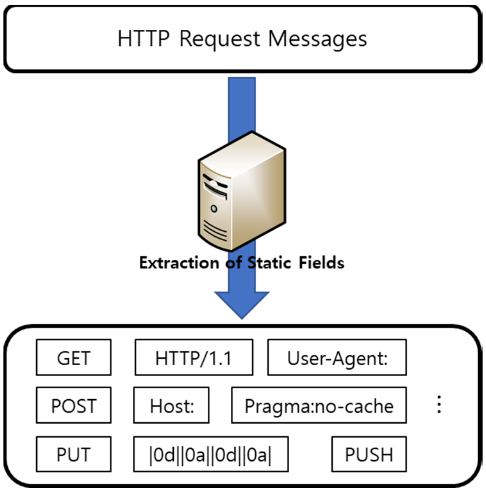

**Figure 4. Example of static fields in HTTP request messages**

Second, the step of extraction of static fields extracts the static fields that constitute the messages. The purpose of this step is to extract all static fields for the detailed message type configuration. Extract not only the fields that are the most active of those that occur, but also the less active of those that occur. For example, the "GET" that constitutes the first part of the HTTP traffic request message is the most frequent field to occur. However, there are various fields for the request message, such as "POST", "PUSH", "PUT", and etc. Using the only CSP algorithm can cause the less frequently occurring fields to be missed. To extract these fields too, this paper uses the CSP (Contiguous Sequence Pattern) algorithm based on tree structure. Figure 4 shows the result of extraction of static fields in HTTP request messages.

Third, the step of composition of message type. The purpose of this step is to use extracted static fields to form the message types, which eventually inferences the structure of the unknown protocol. This step matches all static fields extracted to all messages for configure the message type. Then, we can extract messages with only static fields, and these messages can be duplicated. Duplicate messages are deleted and only one message remains, with all processed messages becoming the message type for the protocol. Extracted message types have all cases of entered protocol. Figure 5 shows an example of message types and the message type extraction process.

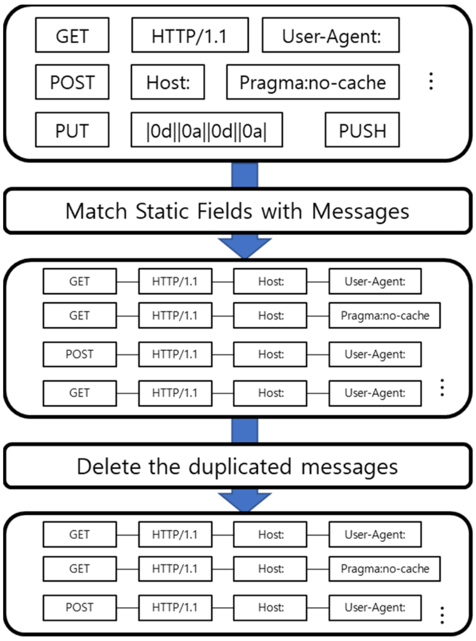

**Figure 5. Example of Message Types in HTTP request messages**

## IV. Extraction of Static Field using CSP Algorithm Based on Tree Structure

CSP is an algorithm based on sequence pattern algorithm. A sequence pattern algorithm was first introduced in [11]. This algorithm, a type of data mining technique, detects time-series patterns in a database containing sequences of values or events. This technique very closely resembles association rule mining [12], in that both are similar processes for discovering frequent patterns in large datasets; however, the objective of association rule mining is to extract concurrent patterns from the same transaction, while that of a sequence pattern algorithm is to extract patterns with a certain order from different transactions [13].

The support threshold is very important for this algorithm. The measure is the ratio of sequences having the target subsequence to the total sequences. A minimum support predefined by the operator is used, because the number of possible subsequences is very large, and operators have different interests and purposes. Thus, we have to prune out uninterested patterns using the minimum support during the early stage.

In this paper, the structure of unknown protocol is inferred using CSP algorithm based on tree structure. This process applies to the extraction of static fields step in the unknown protocol inference architecture. Extract the highest support value with common string using the CSP algorithm in separated messages via step of message assemble. A tree is created dividing the front and back based on the extracted common string. The divided messages repeat the above process until the common string is no longer extracted. If a static field with a support value of less than 100% has been extracted, then there are some remnant messages that are not used for extraction. Extract the common string again with only the messages that were not used for extraction. Repeating this process will allow extraction of not only frequently seen static fields, but also of the various static fields occurring at the same position. Figure 6 shows the process of extraction of static fields using CSP algorithm based on tree structure.

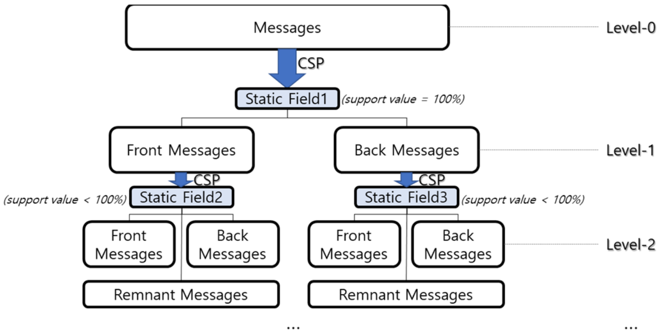

**Figure 6. Extraction of static fields using CSP algorithm based on tree structure**

There are several considerations at this step. If the above process is carried out with no conditions, the noise static fields can be extracted, and can be time consuming to extract. First, extraction of static field must be determined to how level. The level is the stage of the message to be extracted, as shown in figure 6. Second, post-processing is needed to remove duplicate static fields, after extracting all static fields. Problems with duplicate processing may occur in this work.

The level threshold definition is associated with a reduction in noise static field and performance time. If threshold is not defined, noise static fields will be increased. It is because, although all valid static fields have been extracted, it will continue to try to extract the static fields. Static fields extracted after all valid static fields are extracted are highly likely noise static field.

To solve this problem, we focus on the length and number of messages used for static field extraction. If the number and length of messages are not satisfied with the predefined thresholds, the following steps are not performed. The method may have different criteria for each divided message. Statistics can be applied by multiple experiments, but the nature of the methodology makes it impossible. Because message that is constantly divided into front, back, and remnant can cause side effects when applied under the same conditions. As a result of an experiment with level 8, we have confirmed that the first front message extracts all static fields at level 3, and noise static fields were extracted during the remaining 5 runs.

## V. Conclustion and Future Work

We proposed the structure of unknown protocol analysis system and how to extract static field. The unknown protocol analysis system consists of message assemble, extraction of static field, composition of message type. By extracting the static fields, the method of extraction of static field using CSP algorithm based on tree structure was able to extract static fields that had not previously been extracted using only the traditional CSP, and was able to effectively extract all static fields. In addition, definition and solution of possible problems with this method are presented. We also experimented with HTTP protocols to prove the validity of this method. It was proven by comparing the answer to extract static field from HTTP protocol. We will apply it to various protocols as well as HTTP protocols in the future. In addition, it is also planning to conduct a comparative experiment with exist method.

## References

[1] John Narayan, Sandeep K. Shukla, T. Charles Clancy, A Survey of Automatic Protocol Reverse Engineering Tools, Journal ACM Computing Survey, Vol. 48, Issue. 3, No. 40, 2016.

[2] J. Caballero, H. Yin, Z. Liang, and D. Song. Polyglot: Automatic Extraction of Protocol Message Format using Dynamic Binary Analysis. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, 2007.

[3] W. Cui, M. Peinado, K. Chen, H. J. Wang, and L. Irun-Briz. Tupni: Automatic reverse engineering of input formats. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*. 2008.

[4] P. M. Comparetti, G. Wondracek, C. Kruegel, and E. Kirda. Prospex: Protocol specification extraction. In *Proceedings of the 30th IEEE Symposium on Security and Privacy (S&P)*, 2009.

[5] J. Caballero and D. Song. Automatic protocol reverse-engineering: Message format extraction and field semantics inference. International Journal of Computer and Telecommunications Networking 57, 2. Elsevier, 451–474, 2013.

[6] Weidong Cui, Jayanthkumar Kannan, and Helen J. Wang. Discoverer: Automatic protocol description generation from network traces. In USENIX Security Symposium. 2007

[7] Jian-Zhen Luo, and Shun-Zheng Yu. 2013. Position-based automatic reverse engineering of network protocols. Journal of Network and Computer Applications 36, 3 (2013), 1070–1077, 2013.

[8] R. Agrawal and R. Srikant, "Mining sequential patterns," in Proceedings of the 11th IEEE International Conference on Data Engineering, 1995, pp. 3-14.

[9] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proceedings of the 20th International Conference on Very Large Data Bases, 1994, pp. 487-499.

[10] Nikita Borisov, David J. Brumley, Helen J. Wang, and Chuanxiong Guo. 2007. Generic application-level protocol analyzer and its language. In Network and Distributed System Security Symposium.

[11] Sung-Ho Yoon, Jun-Sang Park, Ji-Yeok Choi, Youngjoon Won, and Myung-Sup Kim, "HTTP Traffic Classification based on Hierarchical Signature Structure ," IEICE Transactions on Information and Systems, Vol.E98-D, No.11, , Nov. 2015, pp. 1994-1997

[12] R. Agrawal and R. Srikant, "Mining sequential patterns," in Proceedings of the 11th IEEE International Conference on Data Engineering, 1995, pp. 3-14.

[13] R. Agrawal, T. Imieli, and A. Swami, "Mining association rules between sets of items in large databases," in Proceedings of ACM SIGMOD International Conference on Management of Data, 1993, pp. 207-216.