Using of Time Characteristics in Data Flow for Traffic Classification

Pavel Piskac and Jiri Novotny

Institute of Computer Science, Masaryk University,
Botanicka 68a, 62100 Brno, Czech Republic
piskac@mail.muni.cz,novotny@ics.muni.cz

Abstract. This paper describes a protocol detection using statistic information about a flow extended by packet sizes and time characteristics, which consist of packet inter-arrival times. The most common way of network traffic classification is a deep packet inspection (DPI). Our approach deals with the DPI disadvantage in power consumption using aggregated IPFIX data instead of looking into packet content. According to our previous experiments, we have found that applications have their own behavioral pattern, which can be used for the applications detection. With a respect to current state of development, we mainly present the idea, the results which we have achieved so far and of our future work.

Keywords: protocol detection, time characteristics, flow, IPFIX, pattern

1 Introduction

Information about protocols on a network is very important in many areas. Planning of networks and their topology need to know how the networks will be used in order to their proper configuration. The security point of view needs to recognize protocols in order to find botnets, viruses, spam or intrusions. For example security teams get and analyze data in real-time, protocol detection represents interesting extension of their services and as such will be welcomed.

This work deals with an idea of protocol detection using extended information about a flow. In the first step, we used only inter-packet gaps which seem to be different for each protocol. Then we added information about packet sizes. The most difficult and unsolved problem yet is to find a method which will reduce variability caused by a network mainly jitter.

1.1 Related Work

The first attempts to detect protocols were done using well-known port numbers assigned by IANA [6]. Since the applications are able to change their default port numbers, this method is nowadays weak.

Looking for pattern inside packets is called deep packet inspection. Since each packet has to be read and its content compared with pattern database, this

method is not suitable for high-speed networks. Another weak point is encrypted communication, because all the specific byte strings are hidden. In spite of these disadvantages, deep packet inspection so far brings the best detection accuracy.

Protocol detection on high-speed networks is based on aggregation which separates the important information and reduces computational power. Such methods use behavioral analysis or flow statistics. Data classification by host behavior looks for connection patterns of end points and deduces what the host is doing [7]. This method is not suitable for precise protocol detection, but it splits connections into groups (WWW, email, etc.) instead.

Extended flow statistics can carry various information about packets, e.g. inter-arrival packet times, packet direction, packet sizes, etc. [3]. Protocol detection based on these techniques is faster than deep packet inspection and it is able to classify the data even if it is encrypted [5]. Another approach detects protocols using only the first four packets of a TCP session [1].

2 Protocol Detection

Our first attempts were done in order to show the possibility of taking advantage of the extended flow statistics analysis. This approach was new at the time and we had to verify whether the time characteristics can be used for protocol detection or not. Since it was, as mentioned, a test, we did not try to achieve the best results, we wanted to verify the idea of time characteristics.

2.1 Early Phase of Research

FlowMon probes deployed at Masaryk University generate raw NetFlow data which is not capable to measure inter-packet gaps. Hence we decided to use Flow Time Statistic (FTS) tool which was developed as a testing tool for Liberouter project [8]. FTS generates IPFIX or text data where an additional information about time characteristics is included. Since FTS is a testing tool, it is not suitable for backbone networks monitoring due to insufficient speed of data processing. FTS was installed on our server where it listened to a network device and stored data in plain text format. This kind of data collecting was sufficient at this experimental phase of research. Data visualization was done as a plugin for NfSen [10].

The classification itself was done by vector matching algorithms. FTS computes statistics about flows and these values create a vector. There were obtained pattern vectors from captured data which were then compared with unknown connection vectors. Protocol detection is based on similarity of unknown and pattern vectors. Using this method, we are able to detect dictionary attacks on SSH protocol. There are better tools with higher accuracy but our goal was to show whether the time characteristics can be used for this purpose. As a first step, we used the following properties: number of packets and bytes, identification number of network and transport layer, minimal, maximal, average and

standard deviation inter-arrival time of packets in a flow. These values were chosen according to results from previous experiments.

Vector comparison is done by methods which compute average distance between vectors, root-mean-square (RMS) distance, Euclidean distance and angle between vectors [2] in order to test which one is the most suitable for our purpose and which increases the detection accuracy. Experiments were made and we obtained results of around 90 % successful detection and around 7 % false positives. Average distance between vectors, RMS and Euclidean distance were the most accurate. We captured testing data on our network and it presented a real-life communication.

2.2 Current Phase of Research

Previous part showed that protocol detection based on time characteristics is possible. The results can be considered as acceptable because we did not use any specialized vector comparison method and the pattern vectors were also chosen manually which was sufficient at this stage. The current phase of development is devoted to making the method capable of detecting the whole protocols instead of one specific situation. The newest version of FTS was extended by computing statistics of packet sizes.

One of the most difficult parts of the detection is minimizing the influences in time characteristics caused by a network. Packet inter-arrival times always change as they go through the network and they are never the same. Thus we have to find techniques of how to compute and reduce the inadvisable variability. Another difficult part represents the number of situations which can occur during the communication, e.g. established connection, dictionary attacks, file transfer or tunneling other protocols, etc. These two points cause the biggest problems and we have to deal with them.

In order to detect more than one protocol, we make experiments with Quality Threshold (QT) clustering algorithm [4]. QT clustering provides automatic division into groups where the number of groups is not predefined and it gives always equal results for equal input data. The disadvantage is the speed of computation, which is slower than for example K-Means clustering algorithm [9]. No results from application of QT clustering algorithm are presented here, because they are currently being researched.

3 Conclusion

In this paper, we have shown that statistics about inter-packet gaps in flows can be used for detecting one specific situation of one protocol. Protocol detection is based on vector comparison where predefined pattern vectors are compared to unknown connections vectors. The disadvantage of this approach is that there has to be one pattern vector for each situation, e.g. successful or unsuccessful authentication.

This first part of our work was done in order to verify whether time characteristics are suitable for protocol detection. The result is that time characteristics have to be extended by some other statistics about flow which will increase accuracy and ability to detect various protocols. This work was the starting phase of the future work.

The biggest challenge is to reduce variability caused by network and find out which flow statistics are characteristic for each protocol. As soon as we will figure out how to do that, we can concern on practical implementation.

3.1 Future Work

Future work is an iterative process. At first, we have to reduce the variability in time characteristics and find the proper set of attributes which will the vectors consist of. Automatized division into groups and practical experiments will take place afterwards. We expect that these phases will repeat multiple times.

As soon as we will be satisfied with the detection algorithm, we can customize network probes in order to gain this information from the flows and export it in IPFIX format to our data store where the data will be processed and displayed to users. All the steps are leading towards more accurate and fast protocol detection and thus improved security on computer networks.

Acknowledgements. This material is based upon work supported by the Czech Ministry of Defence under Contract No. SMO02008PR980-OVMASUN200801.

References

1. Bernaille, L., Teixeira, R., and Salamatian, K. (2006). Early Application Identification. In ADETTI/ISCTE CoNEXT Conference, Lisboa, Portugal.
2. Deza, M. M., Deza, E. (2009), Encyclopedia of Distances, Springer Berlin Heidelberg.
3. Erman, J., Arlitt, M., and Mahanti, A. (2006). Traffic classification using clustering algorithms. SIGCOMM.
4. Heyer, L. J., Kruglyak, S., and Yooseph, S. (1999). Exploring expression data: Identification and analysis of coexpressed genes, in Genome Research, 9(11):1106–1115.
5. Hjelmvik, E., John, W. (2010). Breaking and Improving Protocol Obfuscation, Technical Report No: 2010-05, Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden.
6. Internet Assigned Numbers Authority (IANA), `http://www.iana.org/assignments/port-numbers`
7. Karagiannis, T., Papagiannaki, K., and Faloutsos, M. (2005). Blinc multilevel traffic classification in the dark. SIGCOMM.
8. Liberouter, `http://www.liberouter.org/`.
9. MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, Berkeley, USA, pp. 281–297.
10. Netflow Sensor (NfSen), `http://nfsen.sourceforge.net/`.