Poster: Network Message Field Type Recognition

Stephan Kleber
stephan.kleber@uni-ulm.de
Institute of Distributed Systems, Ulm University
Ulm, Germany

Frank Kargl
frank.kargl@uni-ulm.de
Institute of Distributed Systems, Ulm University
Ulm, Germany

ABSTRACT

Existing approaches to reverse engineer network protocols based on traffic traces lack comprehensive methods to determine the data type, e. g. float, timestamp, or addresses, of segments in messages of binary protocols. We propose a novel method for the analysis of unknown protocol messages to reveal the data types contained in these messages. Therefore, we split messages into segments of bytes and interpret these as vectors of byte values. Based on the vector interpretation, we can determine similarities and characteristics of specific data types. These can be used to classify segments into clusters of the same type and to identify their data type for previously trained data types. We performed first evaluations of different applications of our method that show promising results up the a data-type-recognition precision of 100 %.

CCS CONCEPTS

• **Security and privacy** → **Network security**; • **Networks** → **Protocol testing and verification**; • **Computing methodologies** → *Cluster analysis.*

ACM Reference Format:
Stephan Kleber and Frank Kargl. 2019. Poster: Network Message Field Type Recognition. In *2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19), November 11–15, 2019, London, United Kingdom.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3319535.3363261

1 INTRODUCTION

Protocol reverse engineering based on traffic traces infers the behavior of unknown network protocols by analyzing observable network messages. It is often applied to understand malware communication [2], or to validate the correct and secure implementation of network services [12]. Existing approaches either concentrate on textual protocols or apply byte-wise multiple sequence alignment. Textual protocol analysis is well understood since natural language processing (NLP, e.g., Latent Dirichlet Allocation [11] or Apriori [8]) is applicable to it. For binary protocols, a typical approach is to byte-wise align messages using sequence alignment (e. g., Needleman-Wunsch [1, 3]). While sequence alignment infers probable field candidates, it does not characterize the content of the field candidates.

Characterizing the contents of message fields is valuable to determine the data type and to reveal the meaning of binary data segments of network messages. For example, recognizing a segment of a message as being a counter, timestamp, or checksum field adds semantic interpretation to the message that hitherto was an unintelligible sequence of bytes. Neither existing approaches based on NLP nor on sequence alignment provide a method to accomplish a semantic interpretation on this level. Thus, the type of inferred fields cannot be distinguished beyond binary and textual segments of messages by supporting manual introspection [6].

Our approach recognizes fine-grained field data types of binary protocols from the analysis of segments of network messages. We quantify the similarity of message segments across all messages. This similarity is the input to classify groups of similar segments into field data types. Finally, we use this classification to label similar segments from unknown protocols with the according data type.

2 APPROACH

The messages we use to train and test our approach are collected from traces of real network protocols. The traces we analyze are publicly available[1]. We pre-process each raw trace by filtering for only the desired protocol, removing duplicates of the payload, and truncating them to 1 000 messages each.

For each individual message from these traces, we first obtain subsequences of messages by tshark and NEMESYS as detailed below. The obtained subsequences are used as atomic segments that approximate protocol fields. For the analysis, we interpret each of these segments as a vector of byte values. To calculate a similarity value for each pair of segments, we defined a dissimilarity measure from the Canberra distance [7]. The pairwise dissimilarities of segments are used as affinity values for clustering by DBSCAN. DBSCAN [4] determines clusters of similar segments by identifying cores of high density within noisy data. The density is high in a neighborhood of segments that have a low dissimilarity to each other. Each cluster finally constitutes a group of similar segments and thus, with high probability, a field data type. We utilize this approach in three modes of application:

Mode 1. First we generate segments that perfectly represent true fields from the knowledge of true field boundaries obtained from tshark[2] dissectors. We label the segments with their true data type, which we also determine from the tshark dissectors. We cluster the segments that were generated in the described manner. In this first mode of application, we can use the ground truth known from tshark to validate our approach and to prepare the second mode of application. This mode of application can only be performed with known protocols for which a tshark dissector exists.

Mode 2. From the clusters of segments from application mode 1, we determine templates of common field data types using ground

CCS '19, November 11–15, 2019, London, United Kingdom

[1] NTP, NBNS, SMB, and DHCP filtered from http://download.netresec.com/pcap/smia-2011/; DNS filtered from https://ictf.cs.ucsb.edu/archive/2010/dumps/ictf2010pcap.tar.gz; all URLs accessed 19 Sept 2019
[2] Terminal interface of Wireshark, see https://www.wireshark.org

truth obtained from multiple protocols. Such a data-type template represents typical features of a field data type. The templates need to be generated just once and can then be applied in all future protocol analyses. The templates are used to recognize sufficiently similar segments in hitherto unseen messages. A data type template is defined by the mean of all feature vector components and their covariance matrix from the cluster of a data type. To recognize segments in traces of unknown network protocols, we iterate a sliding window of the template's byte-length over all message bytes from the unknown trace. Comparing the respective subsequence of message bytes to the template via the Mahalanobis distance [9] enables us to quantify the confidence of whether the subsequence is similar to a learned template. A subsequence that is sufficiently similar to a template is interpreted as new segment and its type is recognized as the one represented by the matching template.

Mode 3. The third mode of application is independent of the first two and also works in case that no known templates match in messages of an unknown network protocol. Naturally, for unknown protocols, no dissector is available to generate segments from messages. Therefore, we resort to a heuristic to generate segments: NEMESYS [5] obtains message segments from unknown protocols. We then cluster these segments in the same way as in mode 1 for the tshark output. Despite that the original kind of data type cannot be recognized in this mode of application, our approach groups segments into clusters that represent the same kind of data. Further analysis can reveal the data type, e. g., by correlation [10].

3 EVALUATION

We show the validity of our approach for each of the described application modes by cluster visualization and recognition statistics. We visualize the dissimilarity of segments compared to the clustering results in Figure 1 and Figure 2 to rationalize the clustering and recognition approaches. For this purpose, the figures show the topology of the high-dimensional feature vectors as the dissimilarities between the segments. To plot the relative dissimilarities of the high-dimensional feature vectors, they must be represented in a two-dimensional plane. We accomplish this by Multi-dimensional Scaling (MDS)[3]. MDS arranges the segments of the analysis in these *topology plots* to represent the dissimilarities *relative to each other*[4].

3.1 Clustering Ground Truth Segments

As described in application mode 1, we classify segments into clusters of their field type. As an example showing the analysis of NTP, Figure 1 is the MDS-plot of the dissimilarity of segments from this application mode. In this figure, the true field types of each segment are color-coded, so it can visually be verified that segments of the same data type are close in terms of their dissimilarity. Thus, the plot illustrates that areas of low dissimilarity fit to data types and that these areas are dense in the of sense of DBSCAN's definition, making this clustering approach applicable. The resulting clusters are labeled as *glows* around the segment *dots*. Therefore, it can visually be distinguished that the clusters inferred by our method match the perception of dense areas. We verify this visual indication of

[3]using the Python module sklearn.manifold.MDS, see https://scikit-learn.org/

[4]The *absolute* values of the positions in the plot are insignificant, which is the reason why we omitted any ax labels in these figures.

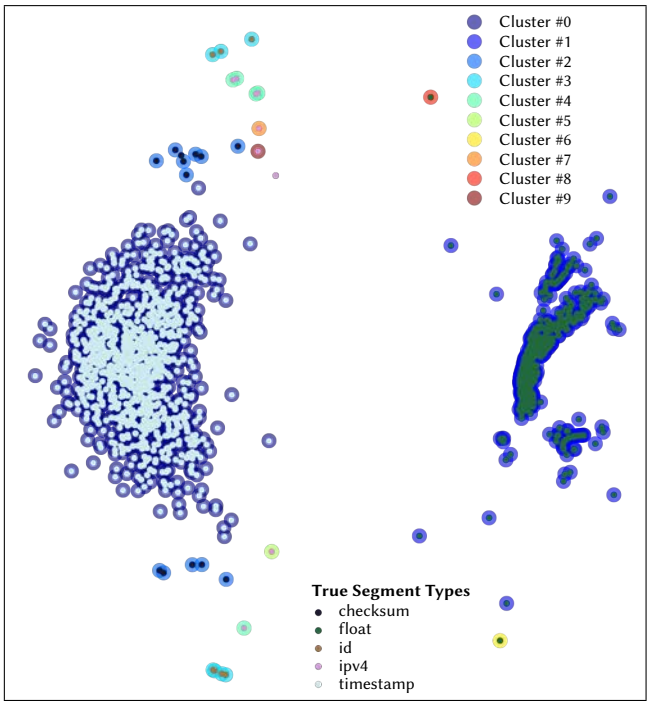

Figure 1: Topology of dissimilarities (by MDS) between segments generated by tshark from NTP messages. The segments are colored according to cluster membership (*glow*) and true field type (*dot*).

the validity of our approach by result precision statistics in the following evaluation of application mode 2.

3.2 Recognize Field Types from Templates

As described in application mode 2, we recognize types of newly seen segments from templates of typical field data types. Thus, we can recognize, e. g., counting numbers, IDs, IP and MAC addresses, and timestamps. By further heuristics, chars [5] and flags can also be identified.

Using templates derived from these clusters, we recognize similar segments in a test set of messages that were not used in creating the templates. For the presented evaluation, we used a combination of 100 messages of each DHCP, DNS, NBNS, NTP, and SMB as training set to derive templates. We tested the approach with different traces of the same binary protocols truncated to 1000 messages each. We compared the data type of segments as recognized by our approach in mode 2 to the true data type derived from tshark for the analyzed protocol. We present the recognition quality using the well-known quality measure "precision". For each data type, the precision (P) is the ratio of correctly recognized segments (TP) among all true (TP) and false recognitions (FP) of one data type: $P = \frac{TP}{TP+FP}$.

As Table 1 shows for each of our test traces, we are able to recognize most data types with high precision. The precision stays below of its optimum value of 1.0 where the segments' data differs from the learned template further than a predefined threshold. We measure this difference in terms of the Mahalanobis distance. A

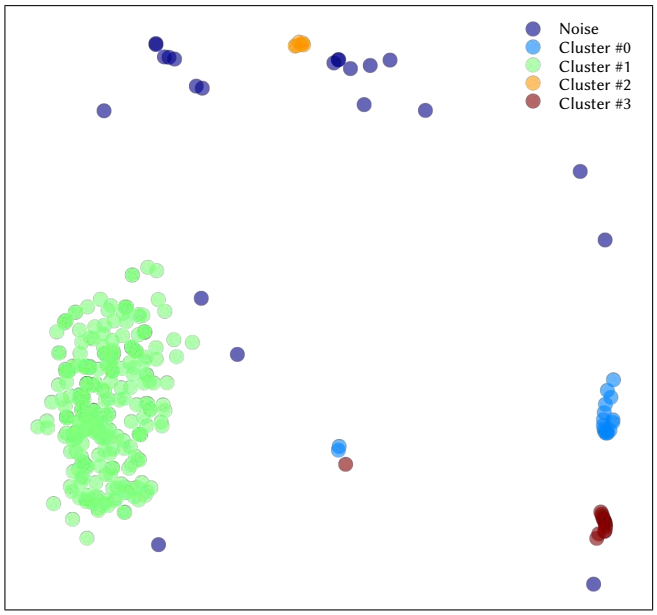

Figure 2: Topology of dissimilarities (by MDS) between NEMESYS-generated segments from DNS. The segments are colored according to cluster membership.

near-zero precision results from completely unrelated training and testing data.

3.3 NEMESYS clustering

We also did a preliminary evaluation of application mode 3 by classifying similar segments without the utilization of ground truth. As an example, Figure 2 illustrates the results for DNS: The topology of dissimilarities and clusters shows distinct dense areas of segments that constitute similar data values. To emulate the lack of ground truth, we performed the segmentation by NEMESYS [5]. However, the inevitable heuristic nature of this approach introduces distortions of the true field boundaries and as a consequence also of the data types. For example, the density cores of NTP are too blurred to clearly separate all contained data types. To solve this issue is subject of our future work.

Table 1: Recognition precision for 1000 messages each.
At positions denoted with "n/a", the protocol does not contain the respective field type. ts is timestamp, mac is MAC address.

Data Type	chars	float	id	ipv4	mac	ts
DHCP	0.87	n/a	0.70	0.99	1.00	n/a
DNS	1.00	n/a	0.87	0.00	n/a	n/a
NBNS	1.00	n/a	n/a	1.00	n/a	n/a
NTP	n/a	0.96	0.00	0.85	n/a	1.00
SMB	0.82	n/a	0.01	n/a	n/a	0.33

4 CONCLUSION

The tshark dissectors provide reliable ground truth from known protocols to test our approach. To realistically obtain message segments for an unknown binary protocol, we propose to improve NEMESYS [5] in future work. During our ongoing work we optimize the segment inference to more accurately match true field boundaries in unknown protocols. Moreover, we envision to add intra- and inter-message semantics inference to be able to identify, e. g., length fields and message counter fields. Another future task will be to validate the approach on a broader basis of protocols for training and testing.

By our novel approach, we can identify similar segments and group these into clusters that represent the same data type regardless whether ground truth about the protocol is available (mode 1) or not (mode 3). For field types learned from a training set (mode 1), it is possible to label matching segments of unknown protocols according to their true data type (mode 2). Thus, the analysis of unknown messages from network traces can be improved by automatically determining the most relevant portions of data to investigate further.

REFERENCES

[1] Georges Bossert, Frédéric Guihéry, and Guillaume Hiet. 2014. Towards Automated Protocol Reverse Engineering Using Semantic Information. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security* (AsiaCCS). ACM.

[2] Chia Y. Cho, Domagoj Babić, Eui C. R. Shin, and Dawn Song. 2010. Inference and Analysis of Formal Models of Botnet Command and Control Protocols. In *Proceedings of the 17th ACM Conference on Computer and Communications Security* (CCS). ACM.

[3] Weidong Cui, Jayanthkumar Kannan, and Helen J. Wang. 2007. Discoverer: Automatic Protocol Reverse Engineering from Network Traces. In *Proceedings of 16th USENIX Security Symposium*. USENIX Association.

[4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*. AAAI Press.

[5] Stephan Kleber, Henning Kopp, and Frank Kargl. 2018. NEMESYS: Network Message Syntax Reverse Engineering by Analysis of the Intrinsic Structure of Individual Messages. In *12th USENIX Workshop on Offensive Technologies* (WOOT). USENIX Association.

[6] Stephan Kleber, Lisa Maile, and Frank Kargl. 2019. Survey of Protocol Reverse Engineering Algorithms: Decomposition of Tools for Static Traffic Analysis. *IEEE Communications Surveys and Tutorials*, 21, 1, (February 2019). Firstquarter.

[7] G. N. Lance and W. T. Williams. 1966. Computer Programs for Hierarchical Polythetic Classification ("Similarity Analyses"). *The Computer Journal*, 9, 1, (May 1966), 60–64.

[8] Jian-Zhen Luo and Shun-Zheng Yu. 2013. Position-Based Automatic Reverse Engineering of Network Protocols. *Journal of Network and Computer Applications*, 36, 3, (May 2013). Elsevier.

[9] Prasanta Chandra Mahalanobis. 1936. On the Generalized Distance in Statistics. *Proceedings of the National Institute of Science of India*, 2, 1, 49–55.

[10] David N. Reshef, Yakir A. Reshef, Hilary K. Finucane, Sharon R. Grossman, Gilean McVean, Peter J. Turnbaugh, Eric S. Lander, Michael Mitzenmacher, and Pardis C. Sabeti. 2011. Detecting Novel Associations in Large Data Sets. *Science*, 334, 6062, (December 2011), 1518–1524. American Association for the Advancement of Science.

[11] Yipeng Wang, Xiao-Chun Yun, Muhammad Zubair Shafiq, Liyan Wang, Alex X. Liu, Zhibin Zhang, Danfeng Yao, Yongzheng Zhang, and Li Guo. 2012. A Semantics Aware Approach to Automated Reverse Engineering Unknown Protocols. In *20th IEEE International Conference on Network Protocols* (ICNP). IEEE.

[12] Shameng Wen, Qingkun Meng, Chao Feng, and Chaojing Tang. 2017. Protocol Vulnerability Detection Based on Network Traffic Analysis and Binary Reverse Engineering. *PLOS ONE*, 12, 10, (October 2017). Public Library of Science.