A system for detecting network intruders in real-time

[#1]Dhivya.J,[#2]Saritha.A.,
PG Student, Department Of Computer Science and Engineering[#1],
AP/CSE, Department Of Computer Science and Engineering[#2]
School of Engineering, Vels University, Chennai, India.

Abstract— *In this paper, we propose Securitas, a protocol identification system used for network trace, which exploits the semantic information in protocol message formats. LTE first cleans log messages and then clusters the cleaned log messages based on the DBSCAN algorithm. At last it infers message templates by LDA Gibbs sampling algorithm. Experimental results show that LTE approach infers and gets multiple log message formats at the same time with more than 90% accuracy and 100% recall.*

Keywords — *Latent Dirichlet Allocation, machine learning, network security, protocol identification*

Introduction

This paper concerns the protocol message format specifications from the network traces of application protocols. the contents of network packets to implement their functionalities, Where detailed knowledge of the protocol specifications is important.

.

I. Background and Motivation

To identify massive network and security logs that record network events is difficult for diagnosing in large-scale network sectors. Extracting log message formats is an important and necessary step to achieve the goal. However, it is time-consuming and costly to automatically and efficiently extract log message formats from massive network and security logs of many different types.

In this paper, we propose a semantics classification system, Securitas, takes traces of network as input and effectively identifies the traces of the target protocol from mixed Network traffic[1]. Our proposed approach performs statistical inference methods and machine learning techniques

II. Related Work

To identify application protocols from network traces, prior methods fall into three categories: port-based analysis, payload-based analysis , and behavior-based analysis,Payload-based approaches can be classified into two subcategories:

1) protocol parsing-based methods, and 2) protocol signature-based (i.e., application fingerprint) methods.

A. *Protocol Parsing-Based Methods*

Protocol parsing-based methods extract protocol features by analyzing the network traces and the binary code of protocols. Reverse engineering with manual efforts is a major instrument for protocol parsing, but it is very time-consuming and laborious Several works paid their attentions on automating protocol parsing. Lim *et al.* proposed an analysis tool to extract output data formats on stripped executables . To automated protocol reverse engineering by tainting data received from the network. In comparison to such methods, we do not require the executable code of application

B. Protocol Signature-Based Methods

Protocol signature-based methods analysis only the payloads of network traces. Such methods involve two ways to implement their functionalities, such as manual analysis and automatic analysis.

Generally, the methods of manual analysis extract application level protocol Note that manual analysis process is an unscalable task it takes advantage of statistical machine learning techniques to identify protocol signatures based on application-level content. The proposed approach automated the construction of protocol signatures based on IP traffic payload content.

III. LOG TEMPLATE EXTRACTION

LTE consists of three major modules, namely Information Filter Module, Message Clustering Module, and Template Extraction Module[2]

.

1) Information Filter

The input to LTE is a set of network and security log messages derived from different network and security devices and services. These log messages usually have hundreds of log message formats, and each of them has a specific set of template words.

2)Message Clustering

The input of this module is the log message .We use the algorithm DBSCAN (Density-Based Spatial Clustering of Applications with Noise)[2]clustering algorithm to as the metric for cluster validation because of the following two reasons First, DBSCAN does not require one to specify the number of clusters in the data. Second, DBSCAN has a notion of noise, and is robust .

3) Template Extraction

The template extraction module is used to identify the log message template words that appear in each cluster of log messages. The input of this module is each cluster of log messages and the output is the formats of log messages in each cluster. We now present our Latent Dirichlet Allocation (LDA) approach to extract template words. Each cluster is a corpus of log messages

IV EXPERIMENTAL RESULTS

To test the effectiveness of Securitas in protocol identification, In this section, we report our experimental validation of our proposed method using real log data . The input to LTE is the real-world network logs containing multi formats and its

output is the inferred corresponding log message.

A. Experimental setup

SSH process records the secure log in Linux for user’s login action. We collect 1746 SSH secure logs from a realworld product network, which constitute our experimental data set. To evaluate LTE approach , we first need to analyze these input logs and get their formats manually. After the manual[2]

analysis, we know that there are 13 different formats and 20 real template words in our experimental logs. For example,some texts that can be easily found not to be template words,

e.g. file path =usr=local=.

Milcom 2015 Track 3 - Cyber Security and Trusted Computing

1545[3]

B. Evaluation Results

The message clustering and template extraction

process of

LTE approach uses the following parameters:

1) parameters " and minPts in DBSCAN algorithm

2) the maximum number of iterations L in Gibbs

algorithm

3) hyper-parameter

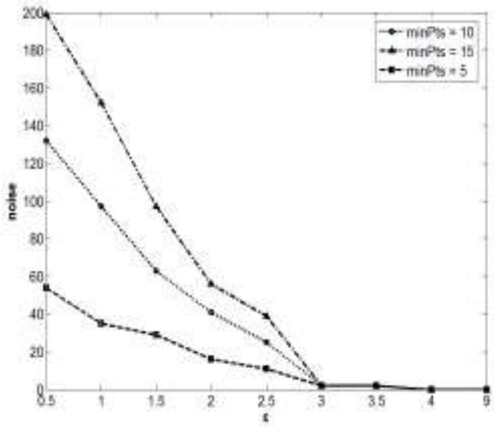

Fig. 1. Noise for different in DBSCAN

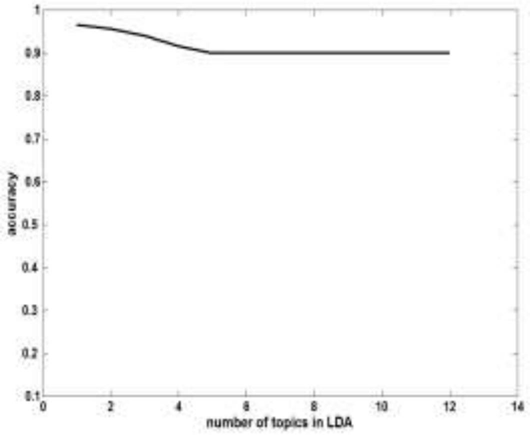

Fig.2 Change of accuracy

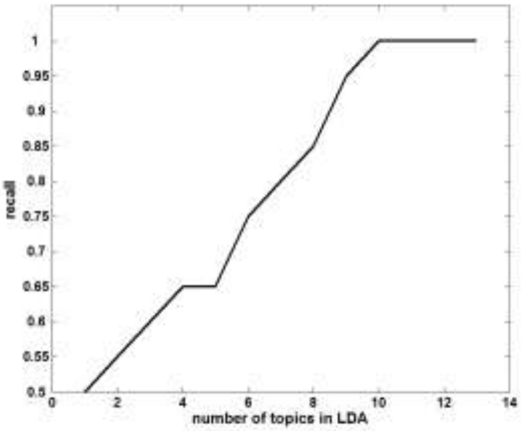

Fig.3 Change of recall

At last, we evaluate the accuracy and the effectiveness of the template words generated by LTE approach using different number of topics generated by LDA, as shown in Fig. 2 and Fig. 3. From Fig. 2 we can find that the accuracy decreases with the increase of the number of topics, but is always no less than 90%. Because with the increase of the number of topics in LDA increase, LTE approach may obtain some words that are not real template words. From Fig. 3, we can find that the recall increases with the increase of the number of topics, and tends to be a steadily numerical value at last. Because the total number of real template words in given network and security logs is a definite value

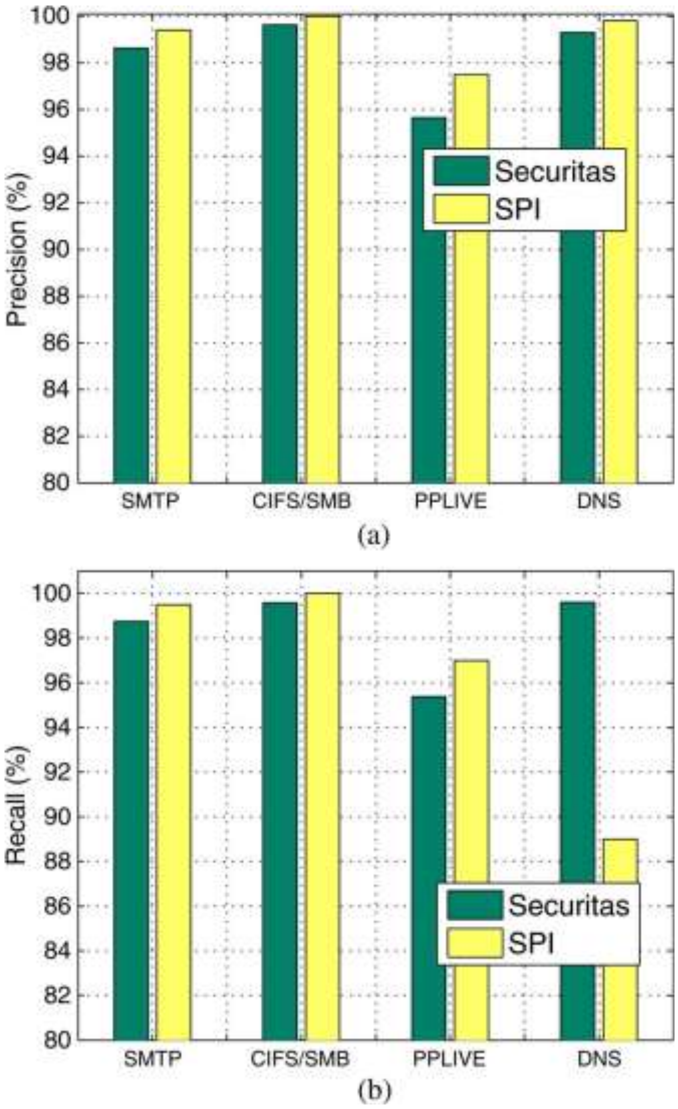

FIG. 4. EXPERIMENTAL RESULTS OF SPI AND SECURITAS. (A) PRECISION. (B) RECALL.

As it is shown in Fig. 4, for SMTP and CIFS/SMB protocols, the precision of both methods is above 99%, and the recall is above 98%. Note that the experimental results of SPI are slightly better than Securitas. We observe that there are 1.6 million SMTP flows involved in our testing data set, which hold about 3.54 GB, and SPI is only capable of parsing 2580 flows, which account for 67% of bytes carried by SMTP flows. Similarly, we have 5.6 million CIFS/SMB flows in our testing data set, which refer to 702MB. Since only 18 CIFS/SMB endpoints have at least 80 packets, SPI can account for 0.2% of bytes carried by CIFS/SMB flows.

Fig.4also shows the results by running Securitas and SPI for

PPLive and DNS protocols. We observe that Securitas outperforms that of SPI on recall of DNS protocol. In other cases, the results of the two methods are very close. However, there are about 1.4 million PPLive flows involved in our testing data set, which hold about 10.38 GB. SPI is able to parse 18 359 flows, which account for 90.5% of bytes carried by PPLive flows. Similarly,we observe about 38 million DNS flows in our testing data set, which refer to 11.26 GB. Due to only 6702 DNS flows having at least 80 packets, SPI is capable of parsing about 5% bytes carried by DNS flow.

V)Conclusion:

This paper proposed LTE approach to automatically extract the network format and security log. LTE leverages semantic information in log messages. It first cleans log messages and then clusters the cleaned log messages based on DBSCAN clustering algorithm .At last it infers message templates by LDA Gibbs sampling algorithm. Our approachhas the benefit that it does not require

any prior knowledge about log message formats.

Experimental results on real-world logs showed that LTE approach can accurately and efficiently infer multiple log message format specifications simultaneously

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions on this paper

REFERENCES

[1]A Semantics-Aware Approach to the Automated Network Protocol Identification
Xiaochun Yun, , Yipeng Wang, Yongzheng Zhang, , and Yu Zhou, 2015

[2] An Automatic Approach to Extract the Formats of Network and Security Log Messages Jing Ya1 Tingwen Liu, 2015

[3]W. Cui, J. Kannan, and H. J. Wang, “Discoverer: Automatic protocol reverse engineering from network traces,” in *Proc. 16th USENIX SS*, 2007

[4]T. Kimura, K. Ishibashi, T. Mori, H. Sawada, T. Toyono, K. Nishimatsu,
A. Watanabe, A. Shimoda, and K. Shiomoto, “Spatio-temporal Factorization of Log Data for Understanding Network Events,” 2014

[5] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, “Detecting Large-Scale System Problems by Mining Console Logs,”2009

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” 1996

[7] W. Cui, J. Kannan, and H. J. Wang, “Discoverer: Automatic protocol reverse engineering from network traces,” 2007

[8] Y. Wang *et al.*, “A semantics aware approach to automated reverse engineering unknown protocols,” 2012

[9] J. Zhang, C. Chen, Y. Xiang,W. Zhou, and A. Vasilakos, “An effective network traffic classification method with unknown flow detection,” 2013.

[11] J. Zhang, C. Chen, Y. Xiang,W. Zhou, and A. Vasilakos, “An effective network traffic classification method with unknown flow detection,” *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 2, pp. 133–147, Jun. 2013.