ISSN:0975-9646

Sadiya Sheikh et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (3) , 2016, 1238-1241

Identification of Network Protocol using Collapsed Variational Bayesian Inference Algorithm

Sadiya Sheikh[#1], Lalit Dole[*2]

[#1, *2]*Computer Science and Engineering Department, Nagpur University*
CRPF Gate No. 3, Hingna Road, Digdoh Hills, Nagpur, Maharashtra 440016,India

***Abstract*—Nowadays a trend is being introduced in which the traffic over the Internet do not have predefined specifications such as port numbers. So they cannot be classified easily from a huge mixed traffic. The main reason behind not having proper specifications for application protocols is to penetrate firewalls or escape administrative control. Many systems are present that can identify and classify the mixed traffic over network but require prior protocol specifications. So these systems do not work well for the current emerging trend mentioned above. Such systems find their application in mainly deep packet inspection module of IDS systems which nowadays are increasingly introduced in large scale in various companies. In this paper, we propose a system that exploits the semantic information from the raw network traces and classify them into identified application protocol without requiring a need for prior protocol specifications. The feature extraction module in the system is being implemented using Collapsed Variational Bayesian Inference Algorithm as in previous existing system the same module was implemented using Gibbs Sampling algorithm which is an iterative algorithm that is very time consuming. Our aim would be to evaluate the quality of our proposed system by calculating three metrics namely recall, precision and F-Measure and then comparing the results with those of prior existing system that uses Gibbs Sampling algorithm to find whether our approach is an efficient one or not.**

***Keywords*— Network traffic classification, application protocol, deep packet inspection.**

I. INTRODUCTION

This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website. Today's scenario, all around the world, is that, might it be a small company or an MNC, each and every one wants their data to be safe. Not only data but their communication with their clients etc. to be super confidential and should be safely reaching their clients etc.. We know that where there is a matter of confidentiality, there is also an increased threat of intruders. The intruders, might be a person or might they be our competitor, system or human, can cause much harm to us. So to protect our important information there should be something or some system that can identify these intruder attacks and should take some precautionary measures at right time to pass through these attacks safely. Nowadays, there are systems that have increasing popularity in market for resolving the above mentioned issues. Such systems are namely Intrusion Detection/Prevention Systems.

These systems can be hardware or a software application program as per the requirement of the company. It studies the network or system activities on a regular basis for anything to be malicious. If it identifies some doubtful activity then it informs about it to the corresponding authority. As we have previously mentioned that it studies either network or system activities, according to which the systems goals are parted which results into two main sub-divisions of the system. Two systems are namely

- Netwrk basd Intusion Detction Sytems
- Host based Intusion Detction Sytems

We mainly concentrate on Network based systems which keep an eye on the inside network activities, identifies the attacks if any. It also keeps an alert system ready to be activated whenever any attack is identified. These systems will prepare a complete summary about which attack has arrived at which time and how many times it has arrived in history. By this information we mainly get idea of the attacker.

The main part in IDS systems is the deep packet inspection part. It works as a network packet filter which checks the packet for data and the genuineness of the protocol by which it is sent. There is a need of such system that can check the genuineness of the protocol occurring in the particular communication.

The system should be able to accurately identify the protocols being used throughout the communication. By identifying the protocol we can identify which application on sender side has sent the message or namely packet. This identification of protocol is important while prioritizing the Quality of Service and while examining the network security.

Nowadays, more than 30% of the internet traffic is unidentifiable by their port numbers as they do not have particular port numbers. There is a purpose of not assigning a particular port number to them which is not to get detected by any network security systems. There should be some system that without relying on the port numbers can accurately identify the protocol being used so that it can be tracked for it's activities.

Also according to a study, maximum internet users are from China. As per increase in Internet traffic, the IP network structure is also changing or becoming more complicated. Network protocol identification has become an important task as to a person, company or to a country. The main example is of parents wanting their kids to be away from porn sites or to restrict their time of playing

online games. Companies should keep control over competitors attacks. Countries need to be safe from malicious information. So these type of activities should be catched out efficiently. This is only possible when there is a system which without having idea of port number can accurately identify the protocol being used to send the particular packet.

The approach mainly concentrates on the extraction process of application-level specifications for network application protocols. Such a task works by analyzing static network traffic traces. As there is a little information available at the network level. Such information about application protocol specification is very helpful in various security-related things. For instance, they are helpful in intrusion detection systems for performing deep packet inspection, and these are used in black-box fuzzing tools easy implementation. It is also used in the automated generation of protocol fuzzers [23] to execute black-box testing of server programs which accept input from network.

As Internet is famous as a business infrastructure, many attacks on it, especially denial-of-service attacks such as TCP SYN flooding [2], Teardrop and Land [3] grows. Because of less security in TCP/IP, responsibility is a must for protecting the sites against network attacks. Although firewalls are used to prevent network attacks, they cannot prevent some specific attacks such as TCP SYN flooding. The deep knowledge of such protocol specifications for finding a number of security problems is invaluable. Consequently, intrusion detection systems (IDS) are increasingly deployed . In IDS, there is a deep packet inspection process done that parse the stream of network into segments or parts with semantics of application-level, and detection rules are applied to only some particular parts of the traffic where application protocol detection plays a major role. Various present methods for protocol detection used for deep packet inspection are time consuming. So the proposed approach can be used to lessen the burden of execution.

A. Problem Definition

Nowadays a vast Internet traffic is not classifiable that means they do not have predefined port numbers which are standardly assigned by IANA. The popular example of it is Skype which does so to easily pass through security services. Attackers mainly divert their traffic over these non-identifiable ports so as to easily escape through them. So we need a system that without depending on the standard assigned port numbers, can accurately identify the associated protocol used for transferring the particular packet over the Internet.

B. Objectives

- A system is to be designed that takes network packets as input and without taking any other systems help should approximate the exact application by which all packets are received by identifying all the associated or used protocols.
- The system should identify the protocol without requiring the need of any previous information about that protocol nor any documentation required associating to that protocol.

Our proposed approach is an experiment of finding whether or not it is possible to apply the above mentioned algorithm for the betterment of the Securitas working or not. System generally involves three major phases

1. Modeling phase
   →Data collector
   →N-gram generation
   →Keyword identification
2. Training phase
   →Data collector
   →N-gram generation
   →Feature Extractor
   →Learning Module
3. Classification phase
   →N-gram generation
   →Feature Extractor
   →Classifier

II. LITERATURE SURVEY

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. The related works reflect three major ways for identifying the application protocol. These are:

- Port based analysis :
- Payload-based analysis
- Behavior-based analysis

The payload-based analyzing systems fall mainly in two categories namely protocol parsing methods and application fingerprint method. The first one is protocol parsing methods that concentrately examines the executable code and traces of target protocol that helps it to get the protocol message format by which it can get a better understanding of the flow of protocols over the network which are associated with each other. This needs a tool called reverse engineering that is a very hectic and a time consuming task. The other one is application fingerprint method that works only by considering the payload of the packets or traces. It only concentrates on the payload of the traces. Two ways of this method are being used by previous researchers. They are manual analysis and automatic analysis. The results of manual analysis are not up to the mark.

Binpac [2] and GAPA [3] are generic protocol analyzers which require protocol grammars as input. Because of having protocol information helps in identifying and understanding applications which can communicate on ports which are non-standard. A number of systems [10, 11, 17, 18] have been proposed that study the network traces which generates by recording the client-server communication. Special meaning of the protocol can be indicated when the network traces are examined for the occurrence of common structures or bytes.

In [2], a system named FlowSifter was proposed which extracts application protocol field using a systematic framework. It invents a new model for grammar which was named as Counting Regular Grammars (CRG).In [6], with the use of TCP or UDP header's well known default server network-port numbers, Internet applications have been identified. But this method is inaccurate.

III. METHODOLOGY

System generally involves three major phases

1. Modeling phase
   →Data collector
   →N-gram generation
   →Keyword identification
2. Training phase
   →Data collector
   →N-gram generation
   →Feature Extractor
   →Learning Module
3. Classification phase
   →N-gram generation
   →Feature Extractor
   →Classifier

i. Data Collector

The main purpose of this part is to collect two types of data. First is the traces corresponding to the target protocol that will be used by learning and classification module for training purpose. The other type of data of traces that do not particularly belong to any target protocol and is categorized as unwanted data.

Provided the executable code of a particular application protocol, in the project it will only be run in a controlled environment to collect protocol data sets. The data set will be used offline as input to the system.

ii. N-gram Generation

This module will structurise the unstructured input data in form of n-sized tokens. First it breaks up each packet contents into a set of n sized-grams. We will be processing them same as we process natural language problems. For instance, the following contents of the packet are divided into grams of size five, an example packet payload "ACK_PACK_SENT_FROM_SYSADMIN\r\n" can be represented with the following grams:

•5-grams:

ACK_P,CK_PA,K_PAC,_PACK,PACK_,ACK_S,CK_SE, K_SEN,_SENT,SENT_,ENT_F,NT_FR,T_FRO,_FROM,F ROM_,ROM_S,OM_SY,M_SYS,_SYSA,SYSAD,YSAD M,SADMI,ADMIN,DMIN\r,MIN \r\n;

The n-gram sequences are the input to Keyword Identification and Feature Extractor.

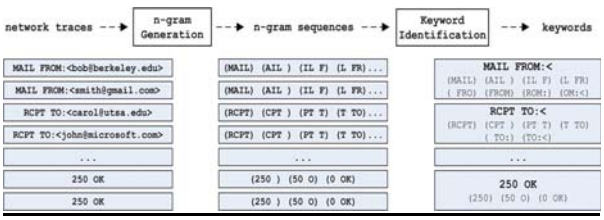


Fig. 2. Example of transforming n-grams into keywords

iii. Keyword Identification

This module perform analysis of target application protocol in offline mode. The input of this module is a sequence of n-grams generated by n-gram Generation, and the output to Keyword Identification is a protocol keyword. In the proposed system, we use the corresponding protocol keyword model to extract features in Feature Extractor.

iv. Feature Extractor

According model output from Keyword Identification, Feature Extractor extracts classification features. For example, if keyword model consists of keywords like MAIL TO,MAIL FROM,RECEIVER etc., then the feature extractor will extract the features that some meassage is being sent from someone to someone. By this conclusion we are very near to a conclusion that the protocol used might be SMTP. The output to Feature Extractor is the keyword distribution of packet .

Here in this module instead of applying Gibbs sampling for feature extraction process, we will implement a collapsed Variational Bayesian inference algorithm for LDA [11].

- **v.** Learning Module And Classifier Module

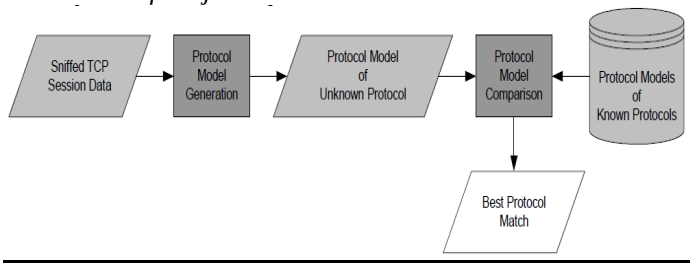
A. Learning Module

The Learning Module's objective is to identify automatically the traces of target protocol from the entire Internet traffic. Here we use supervised machine learning techniques such as SVM,C4.5 or BAYE'S NETWORK for training purpose. The input to this module is the features being classified in a feature extractor module, and the output to this module is the target protocol being detected.

B. Classifier Module

It's the last module of the system, and it makes decisions on each raw packet according to the detected protocol which is output of Learning Module. The unknown packets are classified into two groups, first group of target protocols and the second one of other than target protocol which will be classified as meaningless traffic.

Basic Flow Of Project:



The input is the TCP session packets which is given to the Protocol Model generation which is the keyword identification model. It gives us the unknown Protocol idea. Comparison of these unknown protocols identified are done with our target protocols which are pre-stored in our database. By this comparison we get perfect match for target protocols which will be given as output of our system.

IV. TOOLS USED

Tools are nothing but an environment that provides programmer a field to develop a program or a software in a particular language. There are various different tools that support different languages.the identification of the network protocols system is to be designed by us in java platform. For a java language there are several tools available that can provide an environment for java software development. Namely there are many like gradle, eclipse, intellij, yourkit, clover, mockito, jetty, hibernate, visualvm, etc.

The tool used for our project development is **eclipse 4.5.2**. As we are manually performing our tasks over the packets that are pre stored in a text file, we do not require any other special tool for implementation of our project. We are treating the packets as a language between two communicators so for dealing with the text data we implement data mining topics like feature extraction, keyword identification, learning as well as classification modules.

V. CONCLUSIONS

Performed the calculation of recall, precision and F-Measure after implementation of Keyword Identification and Feature Extraction modules. Here we implemented a new algorithm namely Collapsed Variational Bayesian Inference Algorithm. We have calculated the recall, precision and F-measure values for a supervised learning algorithm called Support Vector Machine. These values were calculated for four different values of 'W' i.e. W={500,1000,1500,2000}. For each pre considered protocol like FTP, SMTP etc. these values are calculated and the values determine that our algorithm is more efficient than the Gibb's Sampling algorithm.

In case of CVB, there is only one copy maintained of latent variable for corresponding each pair of document per word. Thus overall computational cost is O(MK) where M is the total number of unique pairs of document per word and the memory requirement is also the same as computational cost as each time one copy is maintained. Whereas Gibbs Sampling algorithm keeps record of current sample of every word. So memory requirement is O(N) whereas computational cost is O(NK) where N is the total number of words. By seeing this we can conclude that our algorithm is more efficient than Gibbs Sampling Algorithm as its computational cost is more than of CVB.

Future scope for this project can be computation of recall, precision and F-measure for both TCP as well as UDP protocols.

REFERENCES

[1] Xiaochun Yun, Yipeng Wang, Yongzheng Zhang, and Yu Zhou, "A Semantics-Aware Approach to the Automated Network Protocol Identification," in IEEE/ACM Transactions on NETWORKING,2015

[2] C. Meiners, E. Norige, A. X. Liu, and E. Torng, "FlowSifter: A counting automata approach to layer 7 field extraction for deep flow inspection," in *Proc. IEEE INFOCOM*, 2012, pp. 1746 – 1754.

[3] Z. Li *et al.*, "NetShield: Massive semantics-based vulnerability signature matching for high-speed networks," in *Proc. ACM SIGCOMM*, 2010, pp. 279 – 290.

[4] N. Borisov, D. J. Brumley, and H. J. Wang, "A generic application-level protocol analyzer and its language," in *Proc. NDSS*, 2007.

[5] R. Pang, V. Paxson, R. Sommer, and L. Peterson, "Binpac: A yacc for writing application protocol parsers," in *Proc. 6th ACM SIGCOMM Conf. Internet Meas.*, 2006, pp. 289 – 300.

[6] J. St Sauver, "A look at the unidentified half of Netflow (with an additional tutorial on how to use the Internet2 Netflow data archives)," 2008 [Online]. Available: http://www.internet2.edu/presentations/ jt2008jan/20080122-stsauver.pdf

[7] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: Automated construction of application signatures," in *Proc. ACM SIGCOMM MineNet*, 2005, pp. 197 – 202.

[8] J. Kannan, J. Jung, V. Paxson, and C. E. Koksal, "Semi-automated discovery of application session structure," in *Proc. ACM SIGCOMM IMC*, 2006, pp. 119 – 132.

[9] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G. M. Voelker, "Unexpected means of protocol inference," in *Proc. ACMSIGCOMM IMC*, 2006, pp. 313 – 326.

[10] Y. W. Teh, D. Newman, and M. Welling, "A Collapsed Variational Bayesian inference algorithm for Latent Dirichlet allocation," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran & Associates, 2007.

[11] Takuya Konishi, Takatomi Kubo, Kazuho Watanabe, and Kazushi Ikeda, "Variational Bayesian Inference Algorithms for Infinite Relational Model of Network Data," in IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS

[12] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for tcp traffic classification," *Comput. Netw.*, vol. 53, no. 14, pp. 2476 – 2490, 2009.