Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Automated Layer Identification and Classification Of Network Packets

Shancy P. Varghese
Department of Computer Science and Engineering, NCERC, Calicut University

Abstract*— This paper presents a traffic classification scheme to improve classification performance and also the performance of the system. In the proposed scheme we are grouping the correlated flows and classifying it by using a classifier combination framework and aggregating naive Bayes (NB) predictions of the correlated flows. To improve the performance of the system we are detecting the denial-of-service (DOS) packets. We are also classifying the packets according to their size and identifying the applications running on the computers connected to the network to improve the performance of classification. The experimental result shows that packets are classified and layers are identified. We require very less amount of training data for the classification.

***Keywords*— Traffic classification, network security, naive Bayes.**

I. INTRODUCTION

Traffic classification is an automate process which categorizes computer network traffic according to various parameters for example, base on port number or protocol into a number of traffic classes. Traffic classification also plays an important role in modern network management, such as quality of service (QoS) control. Accurate network traffic classification is fundamental to numerous network activities, from security monitoring to accounting, and from Quality of Service to providing operators with useful forecasts for long-term provisioning [1]. Real-time traffic classification has the potential to solve difficult network management problems for Internet service providers and their equipment vendors. Network operators need to know what is flowing over their networks promptly so they can react quickly in support of their various business goals.

Traditional traffic classification techniques may rely on the port numbers specified by different applications or the signature strings in the payload of IP packets [2]. This cannot be supported due to dynamic port numbers and it may affect users' privacy policy. It is a big challenge for current network management is to handle a large number of emerging applications, where it is almost impossible to collect sufficient training samples in a limited time. In a complex network situation it is difficult to obtain a high performance using small set of training data. Also traffic labeling is time consuming for new and encrypted applications. The server will respond all the packets which are coming to it. The hacker or intruder who wants to decrease the efficiency of the server will send dummy packets, which contains no information. To avoid that we are setting a threshold value and if the value of the packet is greater than the threshold we are marking it has attack packet. The applications which are running in the systems which are connected to this network can be identified based on the port number [3]. In this paper we provide a traffic classification scheme to effectively improve the classification performance, to detect the DOS packets and to identify the applications running on the systems that are connected to the network.

The remainder of the paper is structured as follows. Section II reviews some related works. The traffic classification scheme is proposed in Section III. Section IV presents the experimental results. Finally, the paper is concluded in Section V.

II. RELATED WORK

Classifying traffic flows according to the applications that generate them is an important task for effective network planning and design, and monitoring the trends of the applications in operational networks. Support Vector Machines (SVM) represents one of the most promising Machine Learning (ML) tools that can be applied to the problem of traffic classification in IP networks. In the case of SVMs, there are still open questions that need to be addressed before they can be generally applied to traffic classifiers. Having being designed essentially as techniques for binary classification, their generalization to multi-class problems is still under research. Furthermore, their performance is highly susceptible to the correct optimization of their working parameters.

Applying one of the approaches to solving multi-class problems with SVMs to the task of statistical trace classification, and describe a simple optimization algorithm that allows the classifier to perform correctly with as little training as a few hundred samples. The accuracy of the proposed classifier is then evaluated over three sets of trace traces, coming from different topological points in the Internet. Although the results are relatively preliminary, conform that SVM based classifiers can be very elective at discriminating trace generated by different applications, even to reduced training set sizes [4].

The task of identifying the optimal set of flow attributes that minimizes the processing cost, while maximizing the classification accuracy. The dynamic classification and identification of network applications responsible for network traffic flows offers substantial benefits to a number of key areas in IP network engineering, management and surveillance. This proposes a novel method for traffic classification and application identification using an unsupervised machine learning technique. Flows are automatically classified based on statistical flow characteristics. Evaluate the efficiency of approach using data from several traffic traces collected at different locations of the Internet. Use feature selection to find an optimal feature set and determines the influence of different features [6]. The main disadvantage is that the accuracy is only 86%. Another one is quantify the performance in terms of processing time and memory consumption and to investigate the trade-off between the approach's accuracy and processing overhead.

Network managers are inevitably called upon to associate network traffic with particular applications. Indeed, this operation is critical for a wide range of management functions ranging from debugging and security to analytics and policy support. Traditionally, managers have relied on application adherence to a well established global port mapping: Web traffic on port 80, mail traffic on port 25 and so on. However, a range of factors — including firewall port blocking, tunneling, dynamic port allocation, and a bloom of new distributed applications — has weakened the value of this approach.

Analyze three alternative mechanisms using statistical and structural content models for automatically identifying traffic that uses the same application-layer protocol, relying solely on flow content. In this manner, known applications may be identified regardless of port number, while traffic from one unknown application will be identified as distinct from [5].

A flow classification mechanism based on three simple properties of the captured IP packets: their size, inter-arrival time and arrival order, Traffic classification mechanisms belongs to a wide set of tools That helps the allocation, control and management of resources in TCP/IP networks, and improve the reliability of Network Intrusion Detection Systems. An effective mechanism for the classification of the traffic flows according to the application layer protocols that generated them can suggest suitable measures to prevent or ease network congestion, to deploy QoS–aware mechanisms successfully, or to counter network attacks. This approach belongs to yet another class of techniques, those which try to classify network traffic relying exclusively on the statistical properties of the flows [7].

Distributed Denial-of-Service (DDoS) attacks are a critical threat to the Internet. The key idea is to prioritize a packet based on a score which estimates its legitimacy given the attribute values it carries. Once the score of a packet is computed, this scheme performs score-based selective packet discarding where the dropping threshold is dynamically adjusted based on the score distribution of recent incoming packets and the current level of system overload [3].

III. CLASSIFICATION SCHEME

This section presents a novel NB-based classification scheme to deal with the correlated flows in an effective way, which can significantly improve the classification performance even with a small set of supervised training data. Also detecting the Dos packets and identifying the application running in the systems in the network.

3.1Classification process

The classification process focuses on flow-level traffic classification. In the preprocessing, the system captures IP packets crossing a target network and constructs traffic flows by checking the headers of IP packets. A flow consists of successive IP packets with the same 5-tuple: source IP, source port, destination IP, destination port, and transport layer protocol. Apply a heuristic way to determine the correlated flows. If the flows observed in a certain period of time share the same destination IP, destination port, and transport layer protocol, they are determined as correlated flows. For the classification purpose, a set of flow statistical features are extracted and discretized to represent traffic flows. A novel approach is proposed for traffic classification, namely aggregation of correlated NB predictions. A Naive Bayes (NB) classifier is a simple probabilistic classifier based on applying Bayes therom with strong independence assumptions. A more descriptive term for the underlying probability model would be independent feature model. Naive Bayes belongs to a group of statistical techniques that are called supervised classification as opposed to unsupervised classification. In simple terms, a naive Bayes classifier assumes that the presence or absence of a particular feature of a class is unrelated to the presence or absence of any other feature, given the class variable. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting.

The process of classification consists of two steps. In the first step, the single NB predictor produces the posteriori class-conditional probabilities for each flow. In the second step, the aggregated predictor aggregates the flow predictions posteriori probabilities to determine the final class.

3.2DoS Detection

One of the major threats to cyber security is Denial-of-Service (DoS) attacks in which victim networks are bombarded with a high volume of attack packets originating from a large number of machines. The aim of such attacks is to overload the victim with a flood of packets and render it incapable of performing normal services for legitimate users. DoS attacks can be launched by unsophisticated casual attackers using widely available DoS attack tools. Our main aim is to identify the packets that are normal or attack/worm packets. To find weather the packet is attack or normal, we are calculating the threshold value. If the threshold value is higher than the measured value then we are considering the packet as attack or it is a normal packet.

3.3Packet and application identification

We will find out the Packet size of the incoming packet and classify according to size. We will also find out the Protocol Distribution of the packets and classify them. We will show packet size and protocol in the form of pie graphs.To

identify the applications which are running in the systems which are connected to this network. This is based on the port number. According to the port number we are identifying which application is running.

IV. EXPERIMENTAL RESULTS

The programs were written in java. The profiler program that generated the nominal profile from the packet trace consumed about 1.5 MB of memory space. For each 10-second window comprising about 50,000 packets, the program executed for approximately 0.5 seconds on a 1.5 GHz Intel Pentium PC. The program read the nominal profile and packet traces, generated the attack packets, created scorebooks, and selected the packets to drop. It consumed about 40MB of memory space, and executed for approximately 0.5 seconds for each 1-second window comprising 5,000 legitimate and 150,000 attack packets. This amount of traffic was roughly equivalent to 1-2 Gbps speed. It is believed that the execution time and memory requirements can be greatly improved by optimization and hardware support.

The Fig 1shows the choose devise and options window. This window will appear first when we run the program. We are selecting the type of connection through this window. Also we can specify whether we need full packet or only the header or the size of data we want to retrieve. We have to mark the promiscuous mode because normally a network interface will only receive packets directly addressed to the interface. Promiscuous mode allows the interface to receive all packets that it sees weather they are addressed to the interface or not.

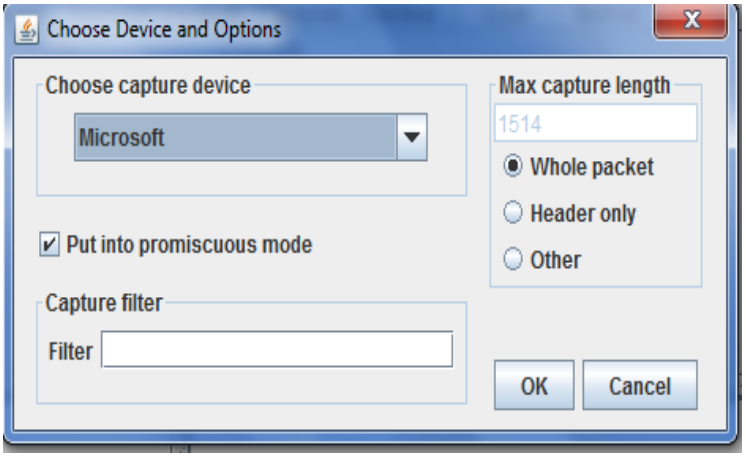

Fig1 Choose Devise and Options

The Fig2 shows the classification window. In the figure the protocol distribution describes the classification of received packets. In this the packets are classified as tcp packet, udp packets and others. The result of classification is also shown in the form of a pie graph. The packet distribution shows that the classification of packets according to their size. That is we are grouping the packets to different groups according to their sizes. result of classification is also shown in the form of a pie graph. In the calculation window it showing the value obtained after calculating. To detect weather the incoming packet is a normal packet or an attack packet. The result window shows the type of incoming packet. That is weather the incoming packet is attack or not.

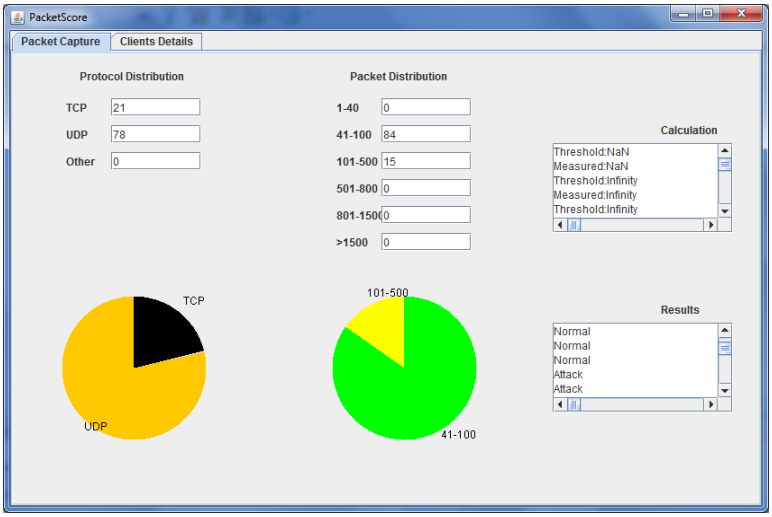

Fig.3 Classification window

We can get the classification results in the form of graphs which is easy to understand. From the graphical representation we can identify the amount of tcp flows, udp flows etc. Also the applications as well as their port number can be listed to identify the applications running in the systems which are connected to the network.

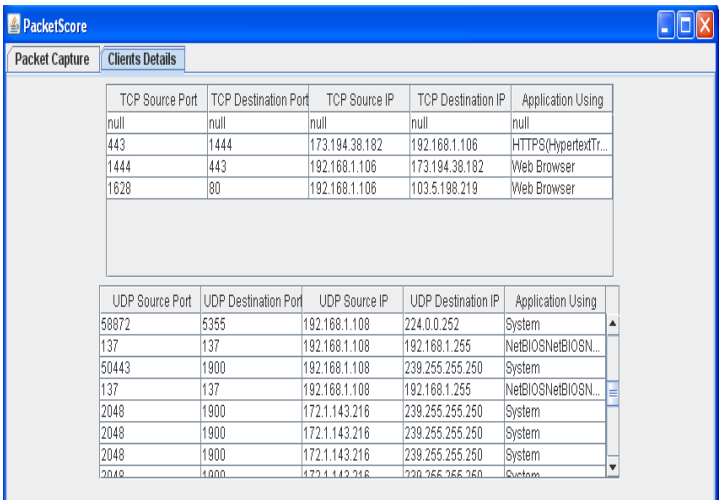

Fig 4:Application identification window

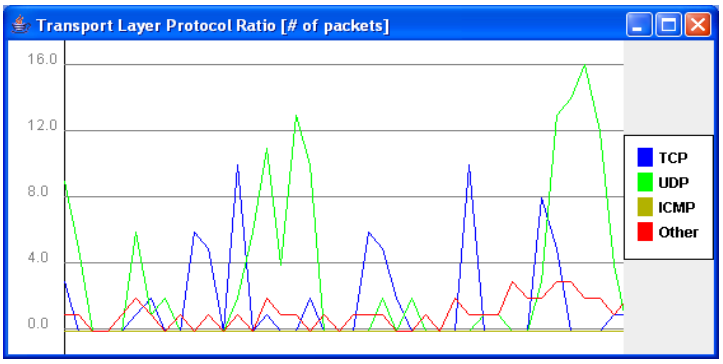

Fig 5 Transport layer protocol ratio

Fig.4 represents the applications running on the systems connected. It contains source port, destination port, source ip destination ip and the application using. Fig 5 shows the graphical representation of the classification result of transport layer protocols received. It includes the tcp, udp, icmp and others.
Paragraph comes content here. Paragraph comes content here.

V. CONCLUSIONS

This will outline the used to defend against DoS attacks. We will find out the statistics of the packets. We can detect any attack in the packets. Also we are identifying the applications which are running in the systems connected to the network. This scheme is able to incorporate flow correlation information into the classification process. This effectively improves the performance of traffic classification without time consuming

.

REFERENCES

[1] Jun Zhang, Chao Chen, Yang Xiang, Wanlei Zhou, and Yong Xiang, "Internet traffic classification by aggregating naïve bayes prediction," in Proc. SIGCOMM Comput. Commun. Rev., Jan. 2013, vol. 8, pp. 5–18.

[2] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multi- level traffic classification in the dark," in Proc. SIGCOMM Comput. Commun. Rev., Aug. 2005, vol. 35, pp. 229–240.

[3] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in SIGMETRICS Perform. Eval. Rev., Jun. 2005, vol. 33, pp. 50–60.

[4] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classifi- cation and application identification using machine learning," in Proc. Ann. IEEE Conf. Local Computer Networks, Los Alamitos, CA, 2005, pp. 250–257.

[5] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salama- tian, "Traffic classification on the fly," in Proc. SIGCOMM Comput. Commun. Rev., Apr. 2006, vol. 36, pp. 23–26.

[6] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," in Proc. SIGCOMM Comput. Commun. Rev., Jan. 2007, vol. 37, pp. 5–16.

[7] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," Commun. Surveys Tuts., vol. 10, no. 4, pp. 56–76, 4th Quarter 2008.

[8] Y.-S. Lim, H.-C. Kim, J. Jeong, C.-K. Kim, T. T. Kwon, and Y. Choi, "Internet traffic classification demystified: On the sources of the dis- criminative power," in Proc. 6th Int. Conf., Ser. Co-NEXT'10, New York, 2010, pp. 9:1–9:12, ACM

