Skype Traffic Identification Based SVM Using Optimized Feature Set

Hongli Zhang[1, 2], Zhimin Gu[1], Zhenqing Tian[2]
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[2]Media College, Inner Mongolia Normal University, Hohhot 010022, China
EMail:zhanghl@imnu.edu.cn,zmgu@x263.net,tianzq@imnu.edu.cn

***Abstract*—Skype traffic recognition is a challenging problem due to the encryption and dynamic port number. Accuracy and timely traffic classification is critical in network security monitoring and traffic engineering. In this paper, we propose an online recognition method based on SVM (support vector machine) machine learning method. As the feature set is optimized instead of redundant, our method is able to compute faster and more accuracy. Experimental results on Collage campus data sets show that our method performs better on both speed and efficiency. Moreover, the robustness of our method is demonstrated on the other non-Skype traffic such as MSN (Microsoft Service Network), PPLive (Peer to Peer LIVE) application.**

Keywords- SVM; Skype;speed;efficiency;

I. INTRODUCTION

Internet traffic monitoring and control have attracted an increasing amount of interest in the past few years. In security perspectives, fast identification of malicious traffic will help security control and isolation of attackers. From the QoS perspective, accurate classification of different traffics helps to identify the application utilizing network resources, and facilitate the instrumentation of QoS for different applications. Furthermore, network operators can trace the growth of different applications and provision network accordingly to accommodate the diverse needs of user population.

Given that Skype is one of most popular VoIP applications used on the Net, it is important to identify it both efficiently and rapidly for network engineering, management tasks and security protect. Moreover, being an encrypted application and tunneling itself through different ports, identifying Skype becomes even more challenging.

One classification approach to classifying applications is to inspect the payload of every packet. However, this is not realistic in practice. Firstly Skype application has encrypted and user data is privacy. Secondly, there is a high computational and storage overhead which effect on the voice quality of service. In the literature [1][2][3], employ matching patterns and packet payloads analyses.

Another approach to classifying traffic is using well-known TCP/UDP port numbers. This solves the issues regarding privacy concerns as well as the requirement for a high computational and storage overhead. However this approach has become increasingly inaccurate, mostly because Skype uses nonstandard ports to avoid detection, to by-pass firewalls or circumvent operating systems restrictions. Thus, other techniques are needed to increase the accuracy of network traffic identification.

Recently, to overcome the deficiencies of traditional traffic classification methods with port/protocol and IDS-type signature matching, several machine learning techniques were proposed to classify Skype traffic, each with reasonable successes.

Various classification approaches have been adopted to identify Skype traffic such as C4.5, RIPPER, Naïve Bayesian, Hidden Markov models [3-9].

In the literature [10] proposed a fundamentally different approach to classifying traffic flows which does not use port numbers, payload information. In contrast to previous methods, our approach is based on observing and identifying patterns of host behavior at the transport layer. Thus this technique is not able to identify distinct applications. More recently, In [11] investigate the extent to which common application protocols can be identified using only the features that remain intact after encryption—namely packet size, timing, and direction information. They employed a k-Nearest Neighbor (KNN) and Hidden Markov Model (HMM) machine learning method to compare the performance. Even though their approach can classify distinct encrypted applications, their performance on SSH classification is only 76% detection rate and 8% false positive rate. In [12] proposed a method to detect applications in SSL encrypted connections. Method uses only the size of the first few packets of an SSL connection to recognize the application, which enables an early classification. It runs in three steps: recognition of SSL connections, detection of the first packet containing application data, and recognition of the encrypted applications. Even though this representation can enable early identification of an application, port numbers were used for classification in their work. Thus, not only its performance will drop when port numbers are changed but also it is not robust since it requires new training each time.

Analyze of Skype traffic become very popular in the last few years. In [13] present an analysis of the Skype behavior. In [14] monitored Skype traffic using relay nodes. In [15] studied Skype traffic to find signatures. In [16] adopted two techniques to detect Skype traffic. These were Chi-Square test and Naive Bayesian classifier. In [17] proposed Skype identification algorithm based on observable part of Skype protocol. First candidate Skype host are detected using traditional IP and port-based identification together with a special signaling flow identification method. Then Skype calls are discovered exploiting the properties of speech flows, timing of voice packet and candidate hosts found in the first step.

978-1-4244-8106-4/$26.00

In contrast to the related work, our proposed system can potentially be applied to any encrypted application since it applies only flow based statistics without using the IP addresses, port numbers and payload data. Last but not the least, none of the previous work employing machine learning techniques to detect different application traffic investigated the robustness of such techniques to different encrypted applications and different network traces.

The remainder of this paper is organized as follows: Section 2 describes Skype protocol. Section 3 introduces the classification methodology using the SVM-based method with an RBF kernel function, the performance evaluation method using cross-validation and discriminator selection algorithms. Section 4 presents the experimental results and their analysis. Section 5 concludes the paper and discusses potential future works.

II. SKYPE OVERVIEW

Skype is a peer-to-peer VoIP client developed by KaZaa [18] that allows its users to place voice calls and send text messages to other users of Skype clients. In essence, it is very similar to the MSN and Yahoo IM applications, as it has capabilities for voice calls, instant messaging, audio conferencing, and buddy lists. However, the underlying protocols and techniques it employs are quite different.

A Skype client (SC) opens a TCP and a UDP listening port at the port number configured in its connection dialog box. SC randomly chooses the port number upon installation. In addition, SC also opens TCP listening ports at port number 80 (HTTP port), and port number 443 (HTTPS port). Unlike many Internet protocols, like SIP and HTTP, there is no default TCP or UDP listening port. Figure 15 shows a snapshot of the Skype connection dialog box. This figure shows the ports on which a SC listens for incoming connections.

The Skype website explains: "Skype uses AES (Advanced Encryption Standard) – also known as Rijndel – which is also used by U.S. Government organizations to protect sensitive information. Skype uses 256-bit encryption, which has a total of 1.1 x 1077 possible keys, in order to actively encrypt the data in each Skype call or instant message. Skype uses 1536 to 2048 bit RSA to negotiate symmetric AES keys. User public keys are certified by Skype server at login."

We conjecture that SC uses a variation of the STUN and TURN protocols to determine the type of NAT and firewall it is behind. We also conjecture that SC refreshes this information periodically. This information is also stored in the Windows registry. Unlike its file sharing counter part KaZaa, a SC cannot prevent itself from becoming a super node.

III. CLASSIFICATION METHOD

A. SVM

SVM algorithm is a new general learning algorithm based on statistical theory [19]. It tries to find the optimal result with limited information provided by small set of samples. Suppose the set of training samples is T= $\{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)\}$, where the i_{th} sample $x_i \in R_p$, $y_i \in \{-1, 1\}$, $i = 1,\ldots,N$. The two-class classification problem is as follows.

$$\min_{w,b} \frac{1}{2}\|w\|^2 + c\sum_{i-1} \xi_i \qquad (1)$$

Where, w is a normal vector, it is perpendicular to the hyper plane. The constant C is only as an additional constraint on the Lagrange multipliers. ξ_i is a non-zero penalizing function. We get the decision function, that is, the classifier that can decide x which belongs to class.

$$f(x) = \begin{cases} 1, g(x) > 0 \\ -1, others \end{cases} \qquad (2)$$

Where g(x) is a Kernel function .Kernel function is the important concept in the SVM theory. Selecting different kernel is an important aspect in the SVM-based classification. Commonly used kernel functions include linear kernel function, polynomial kernel function, RBF (radial basis function) and S-type (sigmoid) kernel function. Through the mapping effect of the kernel function, SVM avoids the dimension disaster of computer in the high-dimensional. In the literature [19], author tested different kernel functions. Clearly, RBF kernel function gives the best classification accuracy. We suggest RBF, which is kernel function is

$$k(x, y) = e^{-\gamma\|x-y\|^2} \qquad (3)$$

B. Cross Validation

If the training samples and the testing samples are the same, the test accuracy will be higher than the actual situation, so we use the cross-validation [20] in our experiments. In n-fold cross-validation, we first divide the training set into n subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining n-1 subsets. Thus, each instance of the whole training set is predicted once. So cross-validation accuracy is the percentage of data which are correctly classified. The prediction accuracy by cross-validation can more precisely reflect the performance on classifying unknown data.

In general, the value of n doesn't influence the cross-validation accuracy much if it is small enough compared to the number of samples. In experiment we set n = 10.

C. Feature Selection

Network traffic is represented using flow-based features. In this case, each network flow is described by a set of statistical features. Here, the feature is a descriptive statistic

that can be calculated from one or more packets. To this end, NetMate [21] is employed to process data sets, generate flows and compute feature values. Flows are bidirectional and the first packet seen by the tool determines the forward direction. Moreover, flows are of limited duration. UDP flows are terminated by a flow timeout. TCP flows are terminated upon proper connection teardown or by a flow timeout, whichever occurs first. The TCP flow time out value employed in this work is 600 seconds [21]. The flows as defined by the features, we extract a set of features in table Ⅰ, from which the machine learning model provides a label {Skype, non-Skype} for each flow. As discussed earlier, features such as IP addresses, source/destination port numbers and payload are excluded from the feature set to ensure that the results are not dependent on such biased features.

These features are simple and well understood within the networking community. They represent reasonable benchmark feature sets to which more complex features might be added in the future.

TABLE I. FEATURESETT

ID	Feature Description
1	Transport protocol type(tcp or udp)
2	Total number of packets in the flow
3	Average number of packets in the flow
4	Duration of flow
5	average of packets in the forward direction per second
6	Number of packets in the forward direction per second
7	average of packets in the backward direction per second
8	Number of packets in the backward direction per second
9	The average window size9
10	Protocol ID
11	Minimum Packet length observed (not counting ACK, RST, SYN packets)
12	Maximum packet length observed
13	The number ratio of send and receive packets
14	The byte ratio of send and receive packets
15	The source port of a flow
16	The destination port of a flow
17	total number of relays used
18	The variance of send packets'size
19	The variance of receive packets'size
20	The number of SYN packets
21	The number of RST packets
22	The number of FIN packets

D. Discriminators Selection-Feature Optimization

We select many dimensions of features to train the SVM models. Among them some features have great positive effect on the classification, while others maybe have little, even negative influence. Therefore, we must get the best combination of features for classification. We select Sequential forward selection. This method can search a good result with much less calculation. We begin with 0 feature chosen; sequentially append 1 feature which can arrive at the best classification result with the chosen features. We sequentially do this work and get the combination with the minimum time for maximum accuracy.

IV. EXPERIMENTS AND CLASSIFICATION RESULT

A. Data Collection

In our experiments, the performance of the SVM classification approaches is established on data sources: campus traces. The campus data sets consist of network traffic from the campus network between the university and the commercial Internet during a continuous eight-hour period in January 2009. Given the privacy related issues which university may face, data is filtered to scramble the IP addresses and further truncate each packet to the end of the IP header, so that all payloads are excluded.

We use NetMate to process packet traces, classify packets to flows a compute feature values. After all the traces had been collected, the flows of each application were divided into a training dataset and an evaluation dataset. Flow division was done randomly. Approximately 200 flows were generated per target application and 100 flows per target application were assigned to the training dataset; the rest were put to the evaluation dataset.

For each bidirectional flows, 22 parameters are computed from the packet headers to be the discriminators for the classification algorithms. These parameters are obtainable in real time from the packet header without storing the packet. We collect both UDP and TCP traffic in dataset.

B. Optimization Feature :The Rate of accuracy and time

In general, different features will have different effects on the classification time and accuracy. Some features may have greater positive effects on the classification, while others maybe have smaller effects. Moreover, some features may even have negative impacts. Therefore, we must carefully choose the best combination of features for SVM discriminators to optimize classification. The main discriminator selection methods are mentioned.

In order to find the best discriminator set from the available 22 parameters for SVM classification, we use the sequential forward selection method. The result is shown in Table Ⅱ. For convenience, we only use the feature ID to represent the features in Table Ⅰ. The best discriminator set obtained is {16, 18, 9, 15, 13, 14, 2, 1, 3}, and it yielded a classification accuracy of 93.47% and time of 4.82second.the rate (accuracy/time) is 0.1939 which is the maximum. We can get Optimization feature with the minimum time for maximum accuracy. The rate value from different feature set is shown in figure 1.

C. Evaluation method

In traffic classification, two metrics are typically used in order to quantify the performance of the classifier: Detection Rate (DR) is equation (4) and False Positive Rate (FPR), is equation (5). In this case, DR will reflect the number of Skype flows correctly classified whereas FPR will reflect the number of Non-Skype flows incorrectly classified as Skype. Naturally, a high DR rate and a low FPR would be the desired outcomes. They are calculated as follows:

TABLE II. COMPARISON OF TIME AND ACCURACY

Selected feature set	time	accuracy
16	10.2	80.12%
16,18	8.31	89.15%
16,18,9	9.14	90.68%
16,18,9,15	7.26	91.635%
16,18,9,15,13	5.23	92.15%
16,18,9,15,13,14	5.17	93.08%
16,18,9,15,13,14,2	5.08	93.12%
16,18,9,15,13,14,2,1	4.81	93.23%
16,18,9,15,13,14,2,1,3	4.82	93.47%
16,18,9,15,13,14,2,1,3,17	4.93	94.67%
16,18,9,15,13,14,2,1,3,17,19	5.01	94.13%
16,18,9,15,13,14,2,1,3,17,19,6	5.05	94.08%
16,18,9,15,13,14,2,1,3,17,19,6,20	5.43	93.86%
16,18,9,15,13,14,2,1,3,17,19,6,20,22	5.71	93.17%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21	6.32	93.01%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21,4	7.98	93.05%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21,4,5	8.21	92.83%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21,4,5,8	8.95	92.67%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21,4,5, 8,11	9.35	92.15%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21,4,5, 8,11,7	9.21	91.68%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21,4,5, 8,11,7,6	10.6	91.33%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21,4,5, 8,11,7,6,10	11.9	91.56%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21,4,5, 8,11,7,6,10,11	12.5	91.68%
16,18,9,15,13,14,2,1,3,17,19,6,20,22,21,4,5, 8,11,7,6,10,11,12	16.3	90.17%
Best selecte sequence is 16,18,9,15,13,14,2,1,3	4.82	93.47%

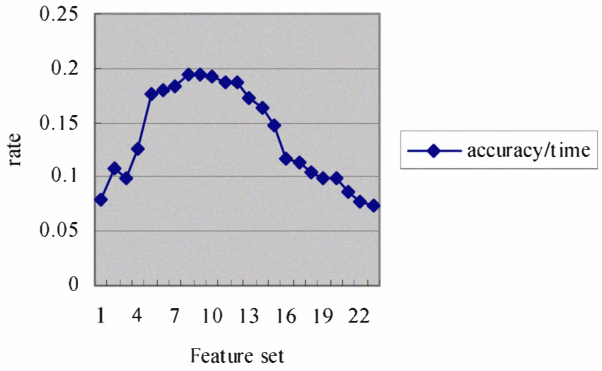

Figure 1. different feature set has different rate value

$$DR = 1 - \frac{\#\,FNClassifications}{TotalNumberSkypeClassifications} \quad (4)$$

$$FP = \frac{\#\,FPClassifications}{TotalNumberNon-SkypeClassifications} \quad (5)$$

Where FN (False Negative) means Skype traffic classified as Non-Skype traffic. Once the aforementioned attribute vectors are prepared for the data sets, Then SVM classifiers are trained on the data. To this end, WEKA, which is an open source tool for data mining tasks, is used. It is employed with its default parameters to run SVM on the data sets. WEKA provides an easy to use interface for machine learning algorithms employed.

D. Classification Result and analysis

We choose three feature set to classify Skype traffic. set1={16,18,9,15,13,14,2,1,3};set2={16,18,9,15,13,14,2,1};set3={16,18,9,15,13,14,2,1,3,17};set 1 has 9 features,set2 has 8 features,set3 has 10 features. Methods mentioned above employed to identification Skype traffic. The evaluation value DR and FP compute as table Ⅲ. These results indicate that the optimized features selected to represent the traffic seem to be sufficient.set2 is the optimized feature which can achieve maximum 99% detection rate and 3% misclassify rate by SVM algorithm.

TABLE III. RESULT OF THE CLASSIFIER

SET	SVM	
	DR	*FP*
set1={16,18,9,15,13,14,2,1,3}		
Skype	0.94	0.03
Non-Skype	0.94	0.05
set2={16,18,9,15,13,14,2,1}		
Skype	0.99	0.03
Non-Skype	0.94	0.04
set3={16,18,9,15,13,14,2,1,3,17}		
Skype	0.97	0.03
Non-Skype	0.94	0.07

V. CONCLUSION AND FUTURE WORK

Security and QoS intervention become necessary in the early stage of the traffic flow. Identifying the traffic after the network flow could be too late. Much of this existing research focuses on the achievable accuracy (classification accuracy) of different machine learning algorithms. Timely identification did not considerate in research.

In our work, we have employed SVM machine learning algorithms for realizing fast and accuracy Skype traffic classification using feature parameters obtainable early in the traffic flow.

Experiments were carried out using traffic samples collected from a real campus backbone. The results show that time factor is very important for real voice application such as Skype. This result gets Optimization feature with the minimum time for maximum accuracy.

For the data sets tested, the optimized feature set only contains nine discriminators including time and accuracy factors.

Results so far show that approaches perform with a very high detection rate and a low false positive rate when the feature set is employed. In this case, these results also indicate that the optimized features selected to represent the traffic seem to be sufficient as well.

The proposed method is applicable to Skype encrypted network traffic, since it does not rely on the application payload for classification. Furthermore, as all the feature parameters are computable without the storage of multiple packets, the method lends itself well for real-time traffic identification.

One of the disadvantages of SVM-based might be examined inner workings of Skype Super Node relay calls. While Skype direct VoIP communications can be successfully identified by our system, it does not mean that all Skype can be identified. Though the feature of total number of relays (feature 17) used in our work, it can not choose in optimized feature.

ACKNOWLEDGMENT

This paper is partially supported by Beijing Key Discipline Program.

REFERENCES

[1] Zhang Y., Paxson V., "Detecting back doors", Proceedings of the 9th USENIX Security Symposium, pp. 157-170, 2000.

[2] Dreger H., Feldmann A., Mai M., Paxson V., Sommer R., "Dynamic application layer protocol analysis for network intrusion detection", Proceedings of the 15th USENIX Security Symposium, pp. 257-272,2006.

[3] Moore A., Papagiannaki K., "Toward the Accurate Identification of Network Applications", Proceedings of the Passive & Active Measurement Workshop, 2005.

[4] A. W. Moore, D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques", in Proceedings of ACM SIGMETRICS, Banff, Canada, June 2005.

[5] A. McGregor, M. Hall, P. Lorier, J. Brunskill, "Flow Clustering Using Machine Learning Techniques", Passive & Active Measurement Workshop, France, April 2004. SIGMETRICS, Banff, Canada, June 2005.

[6] S. Zander, T.T.T. Nguyen, G. Armitage, "Automated Traffic Classification and Application Identification using Machine Learning", in Proceedings of IEEE LCN, Australia, November 2005.

[7] Wright C., Monrose F., Masson G. M., "HMM Profiles for Network Traffic Classification", Proceedings of the ACM DMSEC, pp 9-15,2004.

[8] Haffner P., Sen S., Spatscheck O., Wang D., "ACAS: Automated Construction of Application Signatures", Proceedings of the ACM SIGCOMM, pp.197-202, 2005.

[9] Williams N., Zander S., Armitage G., "A Prelimenary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Comparison", ACM SIGCOMM Computer Communication Review, Vol. 36, No. 5, pp. 5-16 , 2006.

[10] Karagiannis, T., Papagiannaki, K., and Faloutsos, M, "BLINC: Multilevel Traffic Classification in the Dark",Proceedings of Applications,Technologies, Architectures, and Protocols For Computer Communications pp 229-240, 2005.

[11] Wright C. V., Monrose F., Masson G. M., "On Inferring Application Protocol Behaviors in Encrypted Network Traffic", Journal of MachineLearning Research, (7), pp. 2745-2769, 2006.

[12] Bernaille L., Teixeira R., "Early Recognition of Encrypted Applications",Passive and Active Measurement Conference (PAM), Louvainla- neuve, Belgium, April, 2007.

[13] S. Baset, H. Schulzrine, "An analysis of the skype peer-to-peer internet telephony protocol," in INFOCOM06: Proceedings of the 25th IEEE International Conference on Computer Communications, 2006.

[14] K Suh, D. R. Figueiredo, J. Kurose, and D. Towsley, "Characterizing and detecting relayed traffic: A case study using skype," in INFOCOM 06: Proceedings of the 25th IEEE International Conference on Computer Communications, Apr 2006.

[15] S. Ehlert, S. Petgang, T. Magedanz, and D. Sisalem, "Analysis and signature of skype VoIP session traffic," in CIIT 2006: 4th IASTED International Conference on Communications, Internet, and Information Technology, Nov/Dec 2006, pp. 8389.

[16] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, "Detailed analysis of skype traffic", IEEE Transactions on Multimedia, Vol. 11, No.1, Jan 2009.

[17] M. Perenyi, A. Gefferth, T. D. Drang, S. Molnar, "Skype traffic identification", IEEE, 2007.

[18] SIP, http://www.cs.columbia.edu/sip/, last accessedOctober 2006.

[19] Christianini N, Shawe-Taylor J. An introduction to support vector machines. Cambridge University Press 2000.

[20] Kohavi, R. A Study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp.1137–1143,1995.

[21] NetMate, http://sourceforge.net/projects/netmate-meter/(viewed August 2006).