An Investigation on the Identification of VoIP traffic: Case Study on Gtalk and Skype

Riyad Alshammari and A. Nur Zincir-Heywood
Dalhousie University, Faculty of Computer Science
Halifax NS B3H 1W5, Canada
(riyad,zincir)@cs.dal.ca

***Abstract*—The classification of encrypted traffic on the fly from network traces represents a particularly challenging application domain. Recent advances in machine learning provide the opportunity to decompose the original problem into a subset of classifiers with non-overlapping behaviors, in effect providing further insight into the problem domain. Thus, the objective of this work is to classify VoIP encrypted traffic, where Gtalk and Skype applications are taken as good representatives. To this end, three different machine learning based approaches, namely, C4.5, AdaBoost and Genetic Programming (GP), are evaluated under data sets common and independent from the training condition. In this case, flow based features are employed without using the IP addresses, source/destination ports and payload information. Results indicate that C4.5 based machine learning approach has the best performance.**

I. Introduction

The increasingly popular Peer-to-Peer (P2P) Voice over Internet Protocol (VoIP) applications have gain huge success in the last few years and are becoming a major communication service for enterprises and individuals since the cost of VoIP calls is much cheaper than the traditional public Switched Telephone Networks (PSTNs), the voice and video quality is getting better, the communication is free of charge if placed directly from VoIP end user to another one and the dynamic approach to circumvent restrictive network environments such as firewalls and Network Address Translation (NAT) boxes is possible. To date, there are many VoIP products that are able to provide high call quality such as Skype [1], Gtalk [2], Microsoft Messenger (MSN) [3], and Yahoo! Messenger (YMSG) [4]. Thus, an efficient classification of such VoIP traffic represents a fundamental issue for network management tasks such as managing bandwidth budget and ensuring quality of service objectives. Naturally, the process of traffic classification has several unique challenges including: non-standard utilization of ports, embedding of services within encrypted channels, dynamic port-to-application relationships, and the real-time nature of the domain.

Skype is a proprietary P2P VoIP application. On the other hand, Gtalk is an instance messenger developed by Google that allows its users to place voice calls, send text messages, check emails and transfer files. Gtalk provides very similar services as of MSN, YMSG and Skype since it has abilities for voice call, instant messaging and buddy lists. In practice, it has resemblance with Skype application since Gtalk encrypts its traffic; however the fundamental protocols and techniques employed are relatively distinctive. Thus, the goal of this work is to develop a model that distinguishes Gtalk/Skype traffic from non-Gtalk/non-Skype traffic without using IP addresses, port numbers or payload information using features based on flow information. In order to identify Gtalk/Skype traffic, three different machine learning algorithms, namely, AdaBoost, C4.5 and GP, are employed.

II. Related Work

To the best of our knowledge, the focus on the literature for detecting VoIP traffic is on Skype traffic. Skype is one of the most commonly used VoIP applications (Skype has 246 million users and around 10 million users are logged in online at any given time [5]). Skype analysis has become popular in the last few years, in part due to the combination of the encrypted operation and dynamic nature of the port assignment making traditional methods of traffic identification redundant. Baset et al. present an analysis of the Skype behavior such as login, NAT and firewall avoidance, and call setting up under three different network conditions [6]. Suh et al. concentrate on the classification of relayed traffic and monitored Skype traffic as an application using relay nodes [7]. Relay node is part of the decentralized Skype network that can ease the routing of Skype traffic to bypass NATs and firewalls. They used several metrics based on features such as inter-arrival time, bytes size ratio and maximum cross correlation between two relayed bursts of packets to detect Skype relay traffic. Their results (a 96% true positive and 4% false positive) show the technique is reliable in recognizing relayed Skype sessions but it might not be appropriate to classify all Skype VoIP traffic. Bonfiglio et al. introduced two approaches to classify Skype traffic [8]. The first approach is to classify Skype client traffic based on Pearson's Chi-Square test using information revealed from the message content randomness (e.g. the FIN and ID fields).Their second approach is to classify Skype VoIP traffic based on Naïve Classifier using packet arrival rate and packet length. They obtained the best results when the first and second approaches were combined. They achieved approximately 1% false positive rate and between 2% to 29% false negative rate depending on the data sets they employed. On the other hand, we focus on encrypted tunnel identification without using the IP addresses, port numbers and payload data. In our previous

work, we have compared five different classifiers using flow feature set to classify SSH/Skype traffic [9]. In that work, results show that the C4.5 based approach outperforms other algorithms on the data sets employed.

III. Classifier Methodologies

In this work, we are interested in the application of supervised machine learning (ML) based techniques to network traffic classification, specifically classification of VoIP traffic. The reason we took a ML based approach is the need for automating the process of identifying such traffic but in terms of automatically creating the signatures (rules) that are necessary to classify VoIP as well as automating the process of selecting the most appropriate attributes for those signatures. A further explanation of ML and traffic classification can be found in [10]. In this research, we employed three machine learning algorithms, namely, C4.5, AdaBoost and SBB-GP. A more detailed explanation of C4.5 and AdaBoost algorithms can be found in [11] whereas a more detailed explanation of GP can be found in [12].

IV. Evaluation Methodology

In order to train our ML based classifiers, we needed a controlled environment, where the ground truth is known. Thus, we generated VoIP traffic using different applications on a testbed that we set up. This testbed involved several PCs connected through the Internet and several network scenarios were emulated using Gtalk and other (e.g. Primus, Yahoo messenger) popular VoIP applications. To this end, we observed how Gtalk/Skype reacts to different network restrictions. Moreover, the effects (if any) of different types of access technologies (i.e. WiFi and Ethernet) were also investigated, as well as their combination. Overall, we have conducted over 100 experiments equivalent to more than 25 hours of VoIP traffic. In these experiments, we generated and captured more than 6 GB of traffic at both ends, where approximately 34 million packets were transmitted.

For this work, a Gtalk client was installed on each of the three windows XP machines. The first machine was a Pentium 4 2.4 GHz Core 2 Duo with 2 GB RAM, the second machine was a Pentium 4 2 MHz Core 2 Duo with 2 GB RAM, and the third machine was a MacBook 2 GHz Intel Core 2 Duo with 2 GB RAM. Two machines had a 10/100 Mv/s Ethernet and the third machine had a wireless 10/100 Mv/s card. Furthermore, one was connected to 1 GB/s network while the others were connected to a 10/100 Mb/s network. All three machines had Windows XP Service Packet 2 and all experiments were done using the Gtalk client version 1.0.0.104. In all experiments, we have observed the Gtalk behavior from both ends.

These scenarios include: i) Firewall restrictions on one user end and no restriction at the other end; ii) Firewall restrictions at both users ends; iii) No restrictions at both users ends; iv) Use of wireless and wire-line connections; v) Blocking of all UDP connections, and vi) Blocking of all TCP connections. It should be noted here that during these experiments all the Internet communications went through our networks firewall. The firewall was configured to permit access to the aforementioned restrictions such as do not permit anything, or permit limited well known port numbers such as port 22, 53, 80 and 443.Wireshark [13] and NetPeeker [14] were used to monitor and control network traffic. NetPeeker was used to block ports and to allow either both TCP and UDP traffic, or only UDP or TCP traffic in order to analyze the behavior of the Gtalk client. On the other hand, Wireshark was used to capture traffic from both users ends.

The general call set up between the caller and callee for voice calls is as follows: caller transmits a standard audio file to callee. We used an English spoken text (male and female audio files) without noise and a sample rate of 8 kHz, which was encoded with 16 bits per sample and can be downloaded at [15]. The wav-file was played and then the output of Windows media player was used as input for Gtalk, Primus (soft Talk Broadband (softTBB)) and Yahoo messenger (Encrypted with zfone) clients using a microphone. Wireshark was used to capture the traffic from both users' ends. We have made this testbed traffic publicly available to the research community, too [16].

Furthermore, we have also generated Yahoo messenger traffic (encrypted with Zfone) and Primus VoIP traffic as well as online banking traffic in order to distinguish Gtalk and Skype traffic from these similar applications. Furthermore, Zfone traffic is another encrypted VoIP traffic we generated. Zfone [17] is a software that secures VoIP calls over the Internet. Zfone works by intercepting all the unencrypted VoIP channel and securely protect the VoIP channel by encrypting all the VoIP packets. We used Zfone to secure all Yahoo Messenger audio calls. Zfone detects Yahoo packets and encrypts them as they are sent by the caller machine and detects the encrypted packets received by the callee machine and decrypts them.

On the other hand, we also generated non-encrypted VoIP traffic using Primus Session Initiation Protocol (SIP) client [18]. Primus Enterprise VoIP network deploys the SIP [19] to set up, validate and complete calls over the Internet. We used the Primus softTBB to make calls to Public Switched Telephone Network (PSTN) for voice services (hard line phone) and Mobil Cell phone. The softTBB client runs on a PC or a laptop and connects to the Primus SIP Network over the Internet. Depending on what we call, i.e. a mobile phone or a PSTN phone, Primus SIP network routes the calls to the final destination differently.

Last but not the least, we have also employed network traces captured on the campus network of our university. To this end, university traces employed in this work contain DNS, FTP, SSH, MAIL, HTTP, HTTPS and MSN traffic. Thus, we have traffic traces of 11 applications that have similar behavior to Gtalk. In short, we believe that the traffic traces we employed in this work are representative of traces that can be encountered in real life. It should be noted here that University traffic traces were captured on the Dalhousie University Campus network by the University Computing and Information Services Centre (UCIS) in January 2007. The University traces are labeled using a commercial classification

tool (PacketShaper), which is a deep packet analyzer [20], by the university's network team, UCIS (i.e. not by us). Finally, establishing the ground truth for the traces (Gtalk, Primus etc.) that we generated on our testbed was not a problem, since we knew exactly which applications were running in every experiment. In this work, for Gtalk/Skype traffic identification, we have used a sampled subset of Gtalk traces and mix them with University traces as the training data set. Naturally, the rest of the University traces, Zfone traces, Primus traces, online banking traces and the rest of the Gtalk traces are used as the testing data set. In total, Test traces consist of 44,588,269 flows.

Finally, all the traffic is represented to the classifiers using a flow based feature/attribute set. The flow based feature set is a descriptive statistic that can be calculated from one or more packets for each flow. To this end, NetMate [21] is employed to generate flows and compute 22 feature values, Table I. Flows are bidirectional and the first packet seen by the tool determines the forward direction. In this work, we consider only UDP and TCP flows.

V. Empirical Evaluation

In traffic classification, two metrics are typically used in order to quantify the performance of the classifier: Detection Rate (DR) and False Positive Rate (FP). In this case, DR will reflect the number of Gtalk/Skype flows correctly classified and is calculated using $DR = \frac{TP}{TP+FN}$; whereas FP rate will reflect the number of non-Gtalk/non-Skype flows incorrectly classified as Gtalk/Skype and is calculated using $FPR = \frac{FP}{FP+TN}$. Naturally, a high DR rate and a low FP rate are the most desirable outcomes. Moreover, False Negative, FN, implies that Gtalk/Skype traffic is classified as non-Gtalk/Skype traffic, and False Positive, FP, implies that non-Gtalk/non-Skype traffic is classified as Gtalk/Skype traffic.

All three candidate classifiers are trained on the training data using fifty runs to generate 50 different models for each run so that the results are statistically valid. Weka [22] is employed with default parameters to run C4.5 and AdaBoost. Fifty runs of the C4.5 algorithm are performed using different confidence factors to generate different models for C4.5 and fifty runs of the AdaBoost algorithm are performed using different weight thresholds to generate different models for AdaBoost. We used the same SBB-GP classifier's default parameters as in [12]. Fifty runs of the SBB-GP algorithm are performed using different population initializations to generate different models.

A. Results

Results given in Figures 1 and 2 illustrate that C4.5 based classification approach is much better than other algorithms employed in identifying the Skype flow traffic based on the training data set. We use these trained models, on all of the complete traces employed.

To visualize which machine learning algorithms have the most diverse performance on the test data sets, Figures 3 and 4 show the DR and FPR for all 50 models on the test data sets. On average, C4.5 is much better than other machine learning algorithms on the test data sets in terms of high DR and low FPR. In the case of Skype, C4.5 based classification approach is much better than other machine learning algorithms employed in identifying the Skype traffic. C4.5 based system can correctly classify ≈99% of the instances with less than 1% FPR on combined test traces. For Gtalk, results show that again, the C4.5 classifier performs better than the other classifiers on all of the data sets. C4.5 classifier achieves ≈99% DR and 0.2% FPR on combined test traces. Moreover, the SBB-GP classifier is very competitive with C4.5 under test traces (particularly in terms of Gtalk False Positive rate and Gtalk Detection rate) whereas the AdaBoost based system performs the poorest of the three.

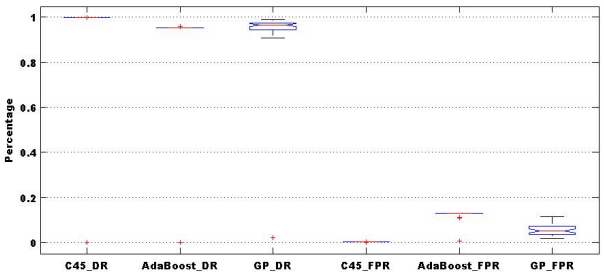

Fig. 1: Results on the Training Data set for Flow based Feature set for Skype detection.

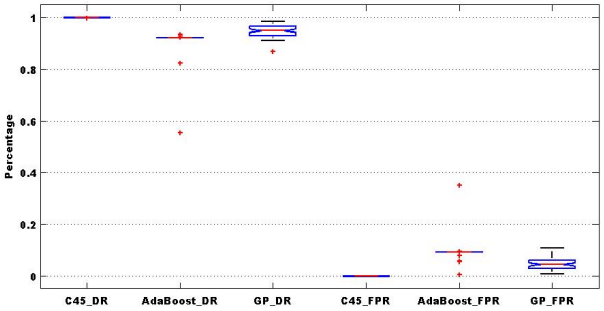

Fig. 2: Results on the Training Data set for Flow based Feature set for Gtalk detection.

VI. Conclusion and Future Work

In this work, we have evaluated three machine learning algorithms, namely AdaBoost, SBB-GP and C4.5, for classifying VoIP traffic in particular Gtalk and Skype traffic from a given traffic file. In this case, the classification based approach is employed with flow attributes.

In our experiments, the C4.5 based classifier can achieve a ≈99% DR and less than ≈1% FPR at its best test performance using the flow based feature set to detect Gtalk and Skype traffic. It should be noted again that in this work, automatically identifying VoIP traffic from a given network trace is

TABLE I: Flow based features employed (with Abbreviations)

Protocol (proto)	Duration of the flow (Duration)
# Packets in forward direction (fpackets)	# Bytes in forward direction (fbytes)
# Packets in backward direction (bpackts)	# Bytes in backward direction (bbytes)
Min forward inter-arrival time (minf_iat)	Min backward inter-arrival time (min_biat)
Std deviation of forward inter-arrival times (std_fiat)	Std deviation of backward inter-arrival times (std_biat)
Mean forward inter-arrival time (mean_fiat)	Mean backward inter-arrival time (mean_biat)
Max forward inter-arrival time (max_fiat)	Max backward inter-arrival time (max_biat)
Min forward packet length (min_fpkt)	Min backward packet length (min_bpkt)
Max forward packet length (max_fpkt)	Max backward packet length (max_bpkt)
Std deviation of forward packet length (std_fpkt)	Std deviation of backward packet length (std_bpkt)
Mean backward packet length (mean_fpkt)	Mean forward packet length (mean_bpkt)

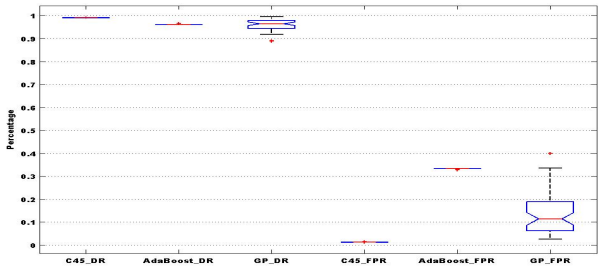

Fig. 3: Results on all Test Data sets for Flow based Feature set for Skype detection.

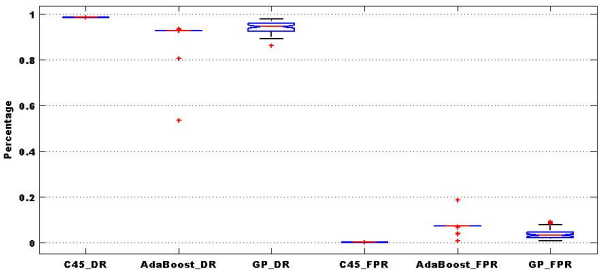

Fig. 4: Results on all Test Data sets for Flow based Feature set for Gtalk detection.

performed without using any payload, IP addresses or port numbers. Future work will follow similar lines to perform more tests on different and larger data sets in order to continue to evaluate the robustness of the proposed approach. Moreover, as a next step, we aim to train and evaluate classifiers for other encrypted applications.

ACKNOWLEDGMENT

This work was supported by MITACS grant. Special thanks to Dana Echtner for helping us in setting up the Google Talk experiments. All research was conducted at the Dalhousie University, Faculty of Computer Science NIMS Laboratory, http://www.cs.dal.ca/projectx. The full version of the paper contains more details and additional references[1].

[1]The full version of this paper is available at http://www.cs.dal.ca/research/techreports/cs-2010-?

REFERENCES

[1] Skype, http://www.skype.com/useskype/.
[2] "Google talk (gtalk)," last accessed October, 2009, http://www.google.com/talk/.
[3] "Msn messenger," last accessed October, 2009, http://webmessenger.msn.com/.
[4] "Yahoo messenger," last accessed October, 2009, http://messenger.yahoo.com/.
[5] "Skype reaches 10 million concurrent users," last accessed May, 2010, http://seekingalpha.com/article/50328-ebay-watch-59-earnings-growth-skype-reaches-10-million-concurrent-users.
[6] S. A. Baset and H. G. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–11.
[7] D. 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.
[8] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic: when randomness plays with you," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, 2007.
[9] R. Alshammari and A. N. Zincir-Heywood, "Machine learning based encrypted traffic classification: Identifying ssh and skype," in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, July 2009, pp. 1–8.
[10] T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *Communications Surveys Tutorials, IEEE*, vol. 10, no. 4, pp. 56 –76, fourth 2008.
[11] E. Alpaydin, *Introduction to Machine Learning*. MIT Press, 2004.
[12] P. Lichodzijewski and M. I. Heywood, "Managing team-based problem solving with Symbiotic Bid-based Genetic Programming," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2008, pp. 363–370.
[13] Wireshark, Last accessed Sep, 2008, http://www.wireshark.org/.
[14] "Net peeker," last accessed October, 2009, http://www.net-peeker.com.
[15] "Signalogic, speech codec wav samples," last accessed October, 2009, http://www.signalogic.com/index.pl?page=codec_samples.
[16] "Gtalk data set," last accessed March, 2010, http://www.cs.dal.ca/˜riyad/.
[17] "The zfone project," last accessed October, 2009, http://zfoneproject.com/getstarted.htm.
[18] "Primus softphone client," last accessed October, 2009, http://www.primus.ca/en/residential/talkbroadband/talkBroadband-softphone.htm.
[19] J. Rosenberg and H. Schulzrinne, "Sip: Session initiation protocol," June 2002, http://www.ietf.org/rfc/rfc3263.txt.
[20] PacketShaper, last accessed March, 2008, http://www.packeteer.com/products/packetshaper/.
[21] NetMate, http://www.ip-measurement.org/tools/netmate/.
[22] "WEKA software," http://www.cs.waikato.ac.nz/ml/weka/.