The Random Forest based Detection of Shadowsock's Traffic

Ziye Deng[1], Zihan Liu[1], Zhouguo Chen[2], Yubin Guo[2]
1. Center for Cyber Security
2. Science and Technology on Communication Security Laboratory
Email: mujinziye@163.com, hunandhan@sina.cn, czgexcel@163.com, ggyybb@hotmail.com

Abstract—**With the development of anonymous communication technology, it has led to the fact that the network monitoring is becoming more and more difficult. If the anonymous traffic can be effectively identified, the abuse of such technology can be prevented. Since the study of machine learning is rapidly developing these years, this paper applies the Random Forest Algorithm --- a semi-supervised learning method --- into the traffic detection of Shadowsocks. We can get over 85% detection accuracy rate in our experiments after applying Random Forest Algorithm by collecting train set, gathering features, training models and predicting results. With the scale of train set and test set increase, the detection accuracy rate gradually increases until it becomes constant. We will make several adjustments on train set, test set and feature set to reduce the false alarm rate and false rate when detecting.**

Keywords- detection; shadowsocks; random forest algorithm; machine learning

I. Introduction

With the number of requirements of oversea news is increasing in recent years, the news even contain sensitive information in politics, economics, democratic, financial, technology and so forth. In order to get around the firewall in this country and to archive more news, more and more people has learnt how to use the proxy software to obtain related materials. High-speed-developed proxy software, however, has become the illegal tool by exempting firewalls to broadcast a large amount of unflattering information. Thus, the frequency of occurrence of things like covert transaction is high. To effectively prevent these things happened and to immediately detect and arrest these criminals, it is necessary for us not only to detect and classify the traffic coming from proxy software, but also to respectively mark different suspicious labels to the encrypted traffics with certain features. After that, it can provide detailed information and data for supervisors to do the following decrypted work and content analysis.

In this paper, we briefly describe the running principles about Shadowsocks [1]. The next step is to comprehensively analyze the traffics from this proxy software and get the feature information from this software. Pointing to distinctive feature data and information, we use Libpcap [2] to resolve protocols, use machine learning to train the certain parts of feature data, and recognize after building models.

The structure of this paper is following: In section 2, we will introduce our preparation work and contributions for this paper. We will simply describe the running principles of this software in section 3. Section 4 will comprehensively demonstrate the detection principles of each traffic coming from this proxy software. The related laboratory results will be claimed in Section 5. At last, Section 6 will be the summary of this paper.

II. Related Work

For understanding the features about traffic deriving from Shadowsocks more accurate, we consult many papers and materials to understand its running mechanism to determine the methods of detecting Shadowsocks' traffic and to extract required feature information. We know, the running of Shadowsocks is great stable --- almost the most stable and fluent proxy software in our country, including the efficiency and speed of over the wall. In aspect of detecting Shadowsocks' traffic, however, there is no remarkable outcome. Because of the stability and efficiency of Shadowsocks, we can use Shadowsocks to connect anonymous network TOR [3], VPN [4] and other proxy, which become the severe threat to firewall and bring even more severe problem for security. In this paper, we proposal a detection method of Shadowsocks based on Random Forest Algorithm [5]. We handle the Shadowsocks' traffic by using machine learning [6], afterwards, we can detect the Shadowsocks' traffic at over 85% high accuracy rate by using semi-supervised learning [7], which can defend and handle the potential dangers at source.

At present, there is not much research in applying machine learning algorithms into traffic detection. The methods of traffic detection are mostly depending on artificial identification like block the ports, IPs etc. If we can use semi-supervised or unsupervised machine learning algorithms, the work of traffic detection can be simply done by machines, which can reduce the people's workload and improve the detection efficiency. Combining the machine learning algorithms and traffic detection is also one of the contributions in this paper.

III. Background

A. Why it is difficult to detect Shadowsocks

The main reason why Shadowsocks is hard to be detected is that the running mechanism of Shadowsocks is pretty simple. Like most principles of proxy software, Shadowsocks firstly establish a SSH [8] based encrypted channel with servers which are outside the firewall;

978-1-5386-3022-8/17 $31.00 © 2017 IEEE
DOI 10.1109/IHMSC.2017.132

Secondly, Shadowsocks make proxy by using established channel, which means that requesting the real servers by SSH server. Lastly, Servers through SSH server send back the responding data by using established channel. Since SSH itself is based on RSA [9] encryption technology, firewalls cannot analyze the keywords of encrypted data during the transmission, which prevent the problem of re-connection. SSH exists the problem of targeted interference, so Shadowsocks split the socks5 [10] protocol into two parts: server-end and client-end. The detailed process is as following: firstly, client send request to communicate with local Shadowsocks based on sock5. Since local Shadowsocks would commonly be local host or routers and any other machines, they do not go through firewalls, thus, they will not be interfered by firewalls. Between the Shadowsocks client and the server, they can communicate through a variety of encryption methods, thus, the data packet going through the firewalls will be shown as common TCP packet. These data packets do not have obvious features and firewalls cannot decrypt these data, which result in that firewalls cannot detect and interfere these data. Lastly, Shadowsocks servers decrypt received encrypted data and send real requests to real servers, and send back the responding data toe Shadowsocks client. The detailed process is shown in Fig.1. We can see that each running step of Shadowsocks bypass the firewalls' detection, and will not be interfered by firewalls. That is the reason for why it is hard to artificially detect Shadowsocks.

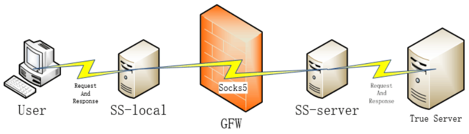

Figure 1. Communication Principle of Shadowsocks.

B. Random Forest Algorithm

In statistics, Random Forest Algorithm is a classifier algorithm. The classifier is an algorithm that determining which class of given sample data should belong to. The Random Forest contains many decision trees and trains the samples and make prediction. The predicted results are decided by most of decision trees in random forest. In our method, we will use Random Forest Algorithm to classify the traffic into two classifications, one the "Shadowsocks traffic", one is "none Shadowsocks traffic".

C. Definition

1) Data Packet: Packet is unit in network which is used to transmit and exchange data. The length of transmitted data the packet contains is not consistent. Packet will be packaged as frame during the transmission. The way of that is to add several information with certain format, like packet header, types of packets, message length, packet version and so forth.

2) Data Flow: Commonly under IP network, the network traffic can be defined as five-tuples: source IP address, destination IP address, source IP port, destination IP port and protocol number. Thus, the network packets with the same five-tuples can be viewed as the one same stream.

3) Biflow: Binary Flow, the data packet sets of same source IP address, destination IP address, source IP port, destination IP port and protocol number.

4) hostProfile: A set of all the packets that have been filtered after a period of time in one host (files are saved as .pcap format).

IV. Our Approach

A. Using Random Forest Algorithm to detect Shadowsocks' traffic.

Random Forest consists of many CARTs(Classification And Regression Tree) [11]. The train set each decision tree used is taken out from the total train set, the taken out train set will be put back to the total train set. When training nodes of each tree, the used train set is taken out from the total train set with certain random proportion, and these taken out train set will not put back to the total train set. The total number of train set is assumed and set as C, the proportion can be C, sqrt(C), 1/2sqrt(C), log2(C). In our experiment, we use default value as sqrt(C).

There are several steps to detect Shadowsocks' traffic when using Random Forest as following:

1) Determining all the data set and parameters the train process needed, including train set P, test set T, feature dimension F, the number of CART t, the depth of each CART d, the number of features the node used f, termination condition including the least samples in node s, the least information gain in node m.

2) Taking train set from the total train set P. The taken train set will put back to the total train set, and the number of the taken train set is equal to that of train set P(i), i represents the index of number. Setting P(i) as root, and starting train from the root.

3) If current node does not reach the termination condition, then randomly take f features from F dimension feature vectors, these taken f features will not put back to the F dimension feature vectors. Selecting the feature k which has the best classification effect and its threshold th from these f features. And utilizing these k features to make judge. The samples will be classified as left node if the value is less than the threshold, if the value is greater than the threshold, the samples will be classified as right node. If current node reaches the termination condition, set the current node as leaf node. The prediction output of this leaf node is the largest number of that classification c(j) in current node node sample set, the probability if the rate of c(j) in the current sample set.

4) Training all nodes until all the nodes are labeled as leaf nodes or are trained.

5) Training all CART until all CART are trained.

6) Predicting the train set T, the process of prediction is like the process of training. Determining from the current CART root node, if less than threshold of current node, the node will enter into left node, and if greater than threshold of current node, the node will enter into right node. The determination process will keep until leaf node outputs the predicted result.

7) Doing the determination and computation for all the predicted values outputted by CART, the largest sum of predicted probability in all trees is the predicted result, which means the total of each probability of c(j).

B. *Features of Shadowsocks' biflow*

From the foregoing process of Random Forest Algorithm, we can know that we need to determine the train set and feature dimensions. According to the network packets hostProfile and the properties of biflow, we proposal several features. Then, we capture a large amout of Shadowsocks' traffic, extract the certain feature values, and save them as train set. The detailed features are shown in Tab.1 partly. Besides these, we also have a 3000-dimension vector, which memorize whether the size of upstream and downstream data packets appear during the whole process of communication.

TABLE I. A PART OF FEATURES

Features	**Meaning**
totalPacketsNumber	The number of total packets
totalOutgoingPackets	The number of total outgoing packets
totalIncommingPackets	The number of total incoming packets
totalTransmissionTimes	The time of the whole transmission
incomingFraction	The fraction of incoming packets
outgoingFraction	The fraction of outgoing packets
maxiumBurstLenth	The maximal burst length
avgBurstLenth	The average burst length
burstTimes	The time of the whole burst

V. EXPERIMENTS AND RESULTS

A. *The process, computation and values of Random Forest*

The steps of experiments:

- Capturing pure Shadowsocks' traffic, dealing with these traffic, extract and save the certain features.
- Using Random Forest to model these value. In Random Forest Algorithm, we set the total value of CART as 100, set grade criterion as 'gini', set the number of extract features as sqrt(C), C is the total number of feature dimensions. The largest depth of tree is set as None until all the nodes are identified. The classified results labeled as two classifications, "Yes" and "No". The remaining parameters are set as the default parameters in Python's RandomForestClassifer function.
- Capturing detection traffic, including Shadowsocks' traffic and none Shadowsocks' traffic. Extracting the certain values of features and save them. Finally, using Random Forest Algorithm to build the models and to predict

B. *Data Collection*

Capturing 1G Shadowsocks' traffic in local host, using Libpcap lib in C to resolve these traffic. We do the computation with extracted features to get all the feature values, and save them into the database as train set. Capturing 1G none Shadowsocks' traffic in local host in 10 times, using libpcap as well to handle these data. Afterwards, capturing over 1G Shadowsocks' traffic and none Shadowsocks' traffic in 26 hosts randomly, handle these data in the same way and store into database as test set.

C. *How to compute.*

According to the judge results of Random Forest, we verify them artificially to evaluate the effectiveness of train set and accuracy rate of the Random Forest Algorithm. Next, we test detection accuracy rate of models built from different sizes of train set.

Shown from the Fig.2, the selection of feature set in our experiment has good effect. It can also prove that the apply of Random Forest Algorithm in Shadowsocks' traffic detection has remarkable outcome. We can also conclude from Fig.2 that using larger test set can have more accurate result.

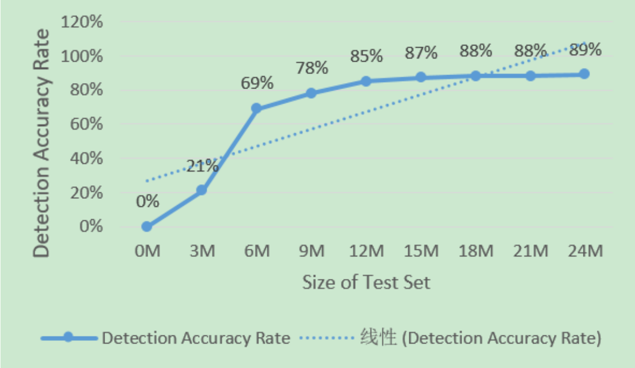

Figure 2. Detection Accuracy Rate

Fig.3 shows that the models built from larger train set have more precise detection rate.

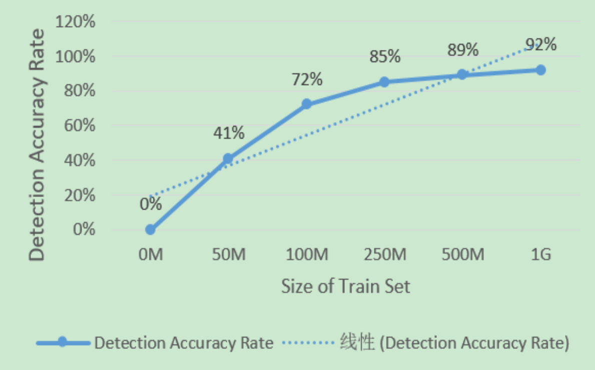

Figure 3. Detection Accuracy Rate of Train Sets' Size

VI. CONCLUSION

In our experiment, we verify that it is remarkable effective to apply machine learning into traffic detection. We also conclude that with the scale of train set increase which means the model is more complete, the accuracy rate of detection will also increase. Additionally, with the scale of test set increase, the accuracy rate of detection will also increase. Applying this semi-supervised machine learning algorithm into traffic detection can reduce false alarm rate, false rate and cost when comparing with the same way done artificially.

In our method, we adopt many features, it can improve detection accuracy rate to certain degree. However, it also increases the system burden, which makes the model relatively redundant. In future work, we will deeply study features to find the most effective feature properties, excluding several unnecessary features, optimize and simplify the model to improve the efficiency of the whole system.

ACKNOWLEDGMENT

The authors would like to thank Dr. Zhuo ZhongLiu and Senior Li Ruixing for their insightful comments and discussions about this work. The authors would like to thank the Center for Cyber Security for offering the enviroments to do experiments. The authors would like to thank Science and Technology on Communication Security Laboratory for providing opportunities to make deeper researhes.

REFERENCES

[1] Xu, Weiai Wayne, and Miao Feng. "Networked creativity on the censored web 2.0: Chinese users' Twitter-based activities on the issue of internet censorship." Journal of Contemporary Eastern Asia 14.1 (2015): 23-43.

[2] Jacobson, Van, and S. McCanne. "libpcap: Packet capture library." Lawrence Berkeley Laboratory, Berkeley, CA(2009).

[3] Fagoyinbo, Joseph Babatunde. The Armed Forces: Instrument of Peace, Strength, Development and Prosperity. AuthorHouse. 2013-05-24 [29 August 2014]. ISBN 9781477226476.

[4] Seid, Howard A., and Albert Lespagnol. "Virtual private network." U.S. Patent No. 5,768,271. 16 Jun. 1998.

[5] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest."R news 2.3 (2002): 18-22.

[6] Goldberg, David E., and John H. Holland. "Genetic algorithms and machine learning." Machine learning 3.2 (1988): 95-99.

[7] Zhu, Xiaojin. "Semi-supervised learning." Encyclopedia of Machine Learning. Springer US, 2011. 892-897.

[8] Ylonen, Tatu, and Chris Lonvick. "The secure shell (SSH) protocol architecture." (2006).

[9] Kaliski, Burt. "PKCS# 1: RSA encryption version 1.5." (1998).

[10] Krawetz, Neal. "Anti-honeypot technology." IEEE Security & Privacy 2.1 (2004): 76-79.

[11] Fonarow, Gregg C., et al. "Risk stratification for in-hospital mortality in acutely decompensated heart failure: classification and regression tree analysis." Jama 293.5 (2005): 572-580.