A Method for HTTP-Tunnel Detection Based on Statistical Features of Traffic

Yao-jun DING
Department of Computer
Northwestern Polytechnical University
Xi'an , China
yaojunding@gmail.com

Wan-dong CAI
Department of Computer
Northwestern Polytechnical University
Xi'an , China
caiwd@nwpu.edu.cn

***Abstract*—HTTP-tunnel is always used by Trojans and backdoors to avoid the detection of firewalls, and it is a threat of network security. HTTP-tunnel traffic is encrypted now, and the only way to detect the HTTP-tunnel traffic is based on statistical features of transport layer. There are a few methods in detection of HTTP-tunnel, and the statistical fingerprinting is an effective method. The method of statistical fingerprinting is instability because the features which the method using is the packet size and the inter-arrival time, and its accuracy is determined by the volume of training set. We suggested a method based on C4.5 algorithm which using the features of packet and flow. Comparing to the algorithm of fingerprint, the C4.5 algorithm had some advantages in stability, accuracy and efficiency in our experiment.**

Keywords-HTTP-Tunnel; Statistical Fingerprinting; C4.5 algorithm; Network Security

I. INTRODUCTION

Nowadays, many Trojans and spywares were used to steal secret information of internet users. Although many users have installed the security software to protect their computers, Trojans can pass the inspection of firewalls by using HTTP-tunnel which can disguise other protocols as HTTP. It is import for network security to detect HTTP-tunnel traffic. Because the HTTP-tunnel traffic transport on port of 80, and the traffic has been encrypted, it is difficult to detect. There are some methods to identify application protocols based on statistical features of traffic. These methods can identify traffic which have been encrypted, and use Machine Learning algorithm to construct the model. HTTP-tunnel can be treated as a special protocol, and we used statistical features of traffic to discriminate HTTP traffic and HTTP-tunnel traffic.

II. RLATED WORK

There are several methods for HTTP-tunnel detection, but it is difficult for accurate detection because most of the traffic have been encrypted. Kevin Borders[1] proposed a method called web tap which based on the features of traffic, such as request regularity, bandwidth usage, inter request delay time, and transaction size, and the filter they design can detect Trojan and spyware accurately.

Manuel Crotti[2] proposed a method based on statistical fingerprint which used three statistical features: inter–arrival time, size and order of the packets. The method set up a fingerprint of HTTP traffic first, then compares a traffic flow to the fingerprint, and calculated the anomaly score. If the anomaly score higher than a threshold, the traffic flow was determined as HTTP-tunnel. The method need a large training set which consist of HTTP flows without HTTP-tunnel flows, and it is a little difficult to get such training set. If the training set was not large enough, the detection precision will be influenced. In this paper, we proposed a new method which using more traffic features to detect HTTP-tunnel traffic, and the precision was still high when the training set was not large enough.

Compare to the literature discussed above, the main contributions of this paper are:

- Applying C4.5 algorithm in HTTP-tunnel detection, and proved the feasibility through experiment.
- Extraction more traffic features of transfer layer, and realized the amalgamation of packet features and flow features.
- The comparing C4.5 algorithm with statistical fingerprint algorithm in precision and time they consumed through the experiment on actual data.

III. HTTP-TUNNEL DETECTION ALGORITHM

A. Basic Concepts

1) HTTP-Tunnel

HTTP-Tunnel is a technique by which communications performed using various network protocols are encapsulated using the HTTP protocol, the network protocols in question usually belonging to the TCP/IP family of protocols. The HTTP protocol therefore acts as a wrapper for a covert channel that the network protocol being tunneled uses to communicate.

2) Traffic flow

Traffic flow is a set of packets which have the same five tuple: source IP, destination IP, source port, destination port, transport protocol. In this paper, we discussed flows on TCP, and a full TCP flow begin with the three handshakes. Traffic flow include bidirection flow and unidirection flow, because our target was HTTP-tunnel, so we only discussed the unidirection flow from client to server.

B. The Limitation of Statistical Fingerprint

Statistical fingerprint is an image which denoted by a matrix whose rows and columns represent the statistical features of internet traffic. The features they used in fingerprint include packet size which denoted by s and inter-arrival time which denoted by Δt, and a traffic flow can be denoted by a vector $(s_i, \Delta t_i)$.

978-1-61284-486-2/11/$26.00 ©2011 IEEE

Statistical fingerprint algorithm can be expressed as below:

Training set which consist of HTTP flows should be collected first, and compute vector $(s_i, \Delta t_i)$ for every packet.

For each flow, using their vector $(s_i, \Delta t_i)$ to construct the fingerprint in a matrix, and the vector is a point in the matrix.

Each flow of testing set which consist of HTTP flows and HTTP-tunnel flows was compared to the fingerprint and calculated anomaly score using formula 1.

$$S(\vec{x} \mid \omega_t) = \left| \frac{\log_{10} \prod_{i=1}^{r} p(x_i \mid \omega_t)}{r} \right| \quad (1)$$

$$p(x_i \mid \omega_t) = M_i(s_i, \ \Delta t_i) \quad (2)$$

In formula 1, the variable $\vec{x}$ denoted the vector which was composed of packets of the flow, and the variable ω_t denoted the type of the flow was HTTP。

Finally, we can decide a traffic flow was HTTP-tunnel or not based on formula 3.

$$\omega(\vec{x}) = \begin{cases} \omega_t & S < T_{acc} \\ \omega_r & \texttt{else} \end{cases} \quad (3)$$

In formula 3, the T_{acc} was a threshold which was very important for the decision. If the variable S was less than T_{acc}, the flow was HTTP flow. Otherwise, the flow was HTTP-tunnel flow.

Through the analysis above, we can find some limitation of statistical fingerprint algorithm.

- The precision of the algorithm depend on a large volume of HTTP flows. If there are not enough HTTP flows, the precision will be instable.
- The fingerprint was a two-dimensional matrix, and only two statistical features can be used.
- The algorithm constructed the HTTP fingerprint only, and the detection precision completely depends on the HTTP traffic flows which used to construct fingerprint. threshold influence

C. Detection Methods Based on Algorithm of C4.5

Algorithm of C4.5 is one of the decision tree classification algorithms, and the algorithm includes training phase and testing phase. In the training phase, we used HTTP traffic flow and HTTP-tunnel traffic flow to train the classification rules. The process of training the classification rules contains following steps.

The first step is to collect HTTP traffic and HTTP-tunnel traffic, and we need to filter the data and construct the traffic flows.

Second, we extract features of packets and flows and construct feature vector, and one feature vector denote a traffic flow. These vectors make up of the training set which contain HTTP traffic and HTTP-tunnel traffic.

We define training set as D, and we need to compute every feature's Gain Ratio to determine which feature can be defined as the node of decision tree. In formula 4, we compute the entropy of training set D which defined as $Info(D)$ and P_i defined as probability of a traffic flow belong to class of C_i which denoted HTTP traffic or HTTP-tunnel traffic.

$$Info\ (D) = -\sum_{i=1}^{m} P_i \log_2 (P_i) \quad (4)$$

We supposed that feature A has v kind of values in training set D, and defined as $\{a_1, \ a_2, \ \ldots, \ a_v\}$. If feature A was a node of decision tree, training set D can be divided into v subsets, defined as $\{D_1, \ D_2, \ \ldots, \ D_v\}$. We want every subset D_j was pure, which means that all the flows in D_j belong to one class, whether HTTP or HTTP-tunnel, but it is almost impossible. So we need to compute the entropy D which divided by feature A and defined as $Info_A(D)$, and calculate the Information Gain of feature A, which defined as $Gain(A)$ in formula 5 and 6.

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \quad (5)$$

$$Gain(A) = Info(D) - Info_A(D) \quad (6)$$

We use Split Information to control the number of branch of decision tree. The Split Information of feature A which defined as $SplitInfo_A(D)$ can be calculated in formula 7.

$$SplitInfo\ _A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2 \frac{|D_j|}{|D|} \quad (7)$$

Finally, we calculated the Gain Ratio of feature A in formula 8.

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)} \quad (8)$$

We repeated above steps to compute every feature's Gain Ratio, and the feature who's Gain Ratio larger than others was a node of decision tree. Then, we can find other nodes of decision tree in the same way.

Compare to the algorithm of statistical fingerprint, algorithm of C4.5 has some advantages:

Using HTTP traffic and HTTP-tunnel traffic to train the classification model, and the classification rules were built by the model. The volume of training set which the model need was smaller than fingerprint.

The features we used in our algorithm contain packet features and flow features, and these features have been proved to be effective for traffic classification. In the algorithm of fingerprint, there were only three features of packet used to classify.

The classification rules in algorithm of C4.5 contain many comparisons between feature's value and node of decision, and didn't depend on a threshold.

IV. EXPERIMENT RESULTS

A. Experiment Dataset

Data used in the experiment were captured on the edge gateway of our campus network, and we captured internet traffic for a week. We sampling data from the traffic, and the sample dataset contains 8435 flows. We can see the detail description of the experiment dataset in table Ⅰ.

TABLE Ⅰ EXPREMNET DATASET

	flows	Training flows	Testing flows
HTTP	2060	1360	700
SMTP over HTTP	2296	1515	781
P2P over HTTP	2018	1332	686
Huigezi	2064	1362	702

B. Features of Traffic Flow

Features used in our algorithm not only contain some statistical features of flow, but also contain some features of first four packets. We can see the details in Table 2.

TABLE Ⅱ FEATURES USEND IN ALGORITHM

No	Abbreviation	Description
1	Pkt_len_min	Minimum of Packet length
2	Pkt_len_max	Maximum of Packet length
3	Pkt_len_mean	Mean of Packet length
4	Pkt_len_var	Variance of Packet length
5	Int_arr_min	Minimum of inter-arrival time
6	Int_arr_max	Maximum of inter-arrival time
7	Int_arr_mean	Mean of inter-arrival time
8	Int_arr_var	Variance of inter-arrival time
9	Pkt1_len	Th first packet's length of flow
10	Pkt2_len	The second packet's length of flow
11	Pkt3_len	The third packet's length of flow
12	Pkt4_len	Th fourth packet's length fo flow
13	Pkt1_inter	The first packet's inter-arrival time
14	Pkt2_inter	The second packet's inter-arrival time
15	Pkt3_inter	The third packet's inter-arrival time
16	Pkt4_inter	The fourth packet's inter-arrival time
17	Flow_byte	Flow size(bytes)
18	Flow_dur	Flow duration
19	Flow_pkt	Total number of packets of Flow
20	Payload_mean	mean payload length excluding headers
21	Class	Type of flow

C. Experiment Tools

In our experiment, we used the Weka-3.6.1[10] to realize the algorithm of C4.5, and run on a personal computer, which was composed of a CPU of Pentium-4 2.80G Hz and a RAM of 1G Bytes.

D. Experiment Result Analysis

We use two metrics known as Recall and Precision which often used in many Machine Learning literatures to compare the effectiveness of the two algorithms. These metrics are defined as follows:

Recall: Percentage of members of class X correctly classified as belonging to class X.

Precision: Percentage of those instances that truly have class X, among all those classified as class X.

We also use another two metrics known as Training Time and Testing Time to compare the efficiency of two algorithms, and defined as follows:

Training Time: The time used to train classification model.

Testing Time: The time used to test classification model.

We used the training set and testing set which described in Table Ⅰ in our experiment. Firstly, We sampled data from training set and the proportion from 20% to 100%, and we can see the classify accuracy of two algorithm in figure 1.

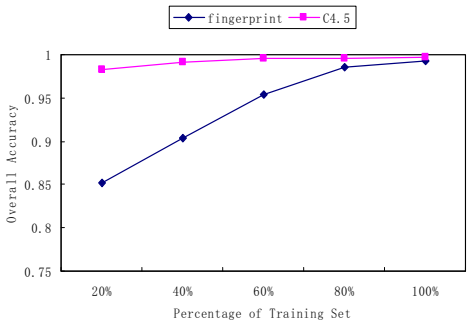

Figure 1 Overall Accuracy of two algorithms on different percentage of training set

From figure 1, we can find that accuracy of fingerprint depend on the volume of training set. The C4.5 was better than fingerprint when the training set was not enough.

Then we used the whole training set and testing set in experiment to compare two algorithms' Precision and Recall, and we can see the result in Table Ⅲ.

TABLE Ⅲ FLOW ACCURACY of TWO ALGORITHMS

	Fingerprint		C4.5	
	Precision	Recall	Precision	Recall
HTTP	0.992	0.985	0.996	0.992
SMTP over HTTP	0.936	0.933	0.953	0.945
P2P over HTTP	0.953	0.972	0.968	0.991
Huigezi	1	1	1	1

Finally, we tested the efficiency of two algorithm, and the result we can see in Table Ⅳ.

TABLE Ⅳ EFFICIENCY of TWO ALGORITHM

	Fingerprint	C4.5
Training time (s)	10.62	6.03
Testing time (s)	8.53	1.56

V. CONCLUSION

Many Trojans and spywares use HTTP-Tunnel technology to pass the inspection of firewall and steal user's secret information. In this paper, we proposed a method to

detect the HTTP-Tunnel flows. The algorithm of C4.5 is an efficiency method for classification, and we used this algorithm in our method. It is also efficiency for HTTP-Tunnel detection which has been proved in our experiment. We used the statistical features of first four packets and some effective features of flows in our methods, and compared the accuracy and efficiency of our method to fingerprint, and the results show that our method behaved well when the training set was not enough.

VI. ACKNOWLEDGMENTS

This work was supported by the National High Technology Research and Development Program 863 (2009AA01Z424).

REFERENCES

[1]. Kevin Borders, Atul Prakash, Web Tap: Detecting Covert Web Traffic, Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS), Oct. 2004.

[2]. Manuel Crotti, Maurizio Dusi, Traffic Classification through Simple Statistical Fingerprinting, ACM SIGCOMM Computer Communication Review, Vol. 37, No. 1, pp. 5-16, Jan. 2007.

[3]. Manuel Crotti, Maurizio Dusi, Tunnel Hunter: Detecting Application-Layer Tunnels with Statistical Fingerprinting, Elsevier Computer Networks (COMNET), Vol.53, No.1, pp.81-97, Jan. 2009.

[4]. Andrew W. Moore, Denis Zuev, Internet traffic classification using Bayesian analysis techniques, in: Proceedings of the 2005 ACM SIGMETRICS, 2005, pp. 50–60.

[5]. Thomas Karagiannis, Konstantina Papagiannaki, Michalis Faloutsos,Blinc: multilevel traffic classification in the dark, in: Proceedings of the ACM SIGCOMM 2005, 2005, pp. 229–240.

[6]. N. Williams, S. Zander, G. Armitage, A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification, SIGCOMM Computer Communication Review, October 2006.

[7]. A.W. Moore, D. Zuev, M Crogan, Discriminators for use in flow-based classification, Technical Report RR-05-13, Department of Computer Science, Queen Mary, University of London, September 2005.

[8]. Wei Li, Andrew W. Moore, Marco Canini, Classifying HTTP traffic in the new age, in: ACM SIGCOMM 2008, Poster, August 2008.

[9]. Lei Yu, Huan Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of the 20th International Conference on Machine Learning (ICML'03), 2003.

[10]. I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufmann Publishers, 2005.