Extract Protocol Characteristics by Apriori Algorithm

Wang XiaoPeng[1,*] ,Sun Yunxiao, Wang Bailing, He Hui, Liu Yang

Department of Computer Science & Technology
Harbin Institute of Technology at Weihai, Shandong, China
*winxp_007@hotmail.com

Abstract: This paper presents a set of programme to extract the application-layer protocol features based on frequent itemsets mining. It can be extended to other areas such as intrusion detection and extracting worm signature.

Keywords: Protocol identification; features of protocols; Apriori algorithm

1 Introduction

With the rapid development of the Internet, identifying classifications of protocols are required. Traditional approaches to application-layer protocol identification are mainly achieved on the base of protocol ports defined by the IANA[1]. Haffner proposed an automatic method to extract network traffic characteristics of application-layer protocols[2]. However, currently many communication protocols are using dynamic ports [3].To solve this problem, researchers have proposed improved identification approach, based on deep packet inspection [6]. However, the efficiency and reliability of manual analysis are relative low. Therefore, we presents a solution of automatic protocol feature extraction.

2 The Method of Protocol Feature Extraction

Based on an improved Apriori algorithm, this paper proposes an automatic extraction method. It first captures training Trace, after that, extract frequent itemsets from the data; finally generate files of protocol features .

2.1 Processing Training Trace

The object of feature extraction is bytes; therefore, a four-tuple, tuple4 (file,stream,packet,offset), is needed to identify a single byte.

In the progress of feature extraction, it is needed to compare bytes in different flows for many times. In order to solve the above issues, this paper sets an index of

[1] Supported by the National Science Nature Foundation of China under Grant No 61170262

training Trace. With a four-level index, all of the PCAP files are only gone through once, which could not only reduce frequency of file operations, but also significantly improve the efficiency. Figure 1 maps diagram for an index.

2.2 Four Characteristics of Protocols

After analysing feature extraction process, four common protocol features are summed up. Character string is directly applied for protocol identification; session tag, package length and packet order cannot improve recognition rates, but can greatly improve the accuracy.

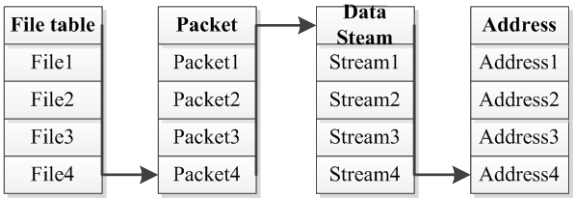

Fig 1 index map diagram

2.3 Frequent Itemsets Mining Algorithm

In frequent itemsets mining algorithms , Apriori algorithm is one of the more popular [7], which applies a recursive method to generate frequent itemsets .

Frequent 1- itemsets is first generated as L1; then frequent 2-itemsets is generated as L2; the algorithm will not stop until there is an r making Lr empty. Here in the k-th iteration, a candidate itemset is generated as Ck; each itemset of Ck are generated from two (k-2)-connecting frequency sets which both belong to Lk-1 with only one different item. The items set of Ck is candidate set for frequent set, and the last frequency set Lk must be a subset of Ck. each item of Ck would be verified in transaction database to determine whether to join Lk or not. It is a bottleneck to verify performance of the algorithm.

2.4 Improvement of Apriori Algorithm

In order to identify correlation from the frequent items, the Apriori algorithm applies a number of Cartesian products, which reduce the efficency. Physical meanings of frequent items have been constrained by packet load index. Therefore, in feature extraction, we can only extract frequent items, but not calculate the correlation between them. Then in this program, we simplify the process of Apriori algorithm. The simplified scanning process is as follows:

Step 1: Scanning the first character of the first package, recording relative offsets and character, setting the frequency to 1, and recording serial number;

Step 2: Continuing to scan the next character, and executing S1 in subsequent operations until the end of the packet;

Step 3: Sequentially scaning the next packet; comparing whether there are characters of same relatively offset; if there are, increasing the count, recording serial numbers and jumping to S5; if there is not, executing S4;

Step 4: Recording this character, setting the occurrence as 1, and recording serial numbers.

Step 5: Skipping back to S2 until the whole packet is scanned.

After packet scanning, a frequent itemsets-A is obtained which accords to appearance degree. A is the character string that we need. The first few bytes of the character string , the session tag and packet order can all be achieved after scanning, but there are no distinguish among the three. A new frequent itemsets-B is needed to be obtained from frequent itemsets mining the other half-stream in the same stream. Contrasting A with B,if the byte's offset is different but value is the same, this byte is session tag.

Plus 1 to the offset of undistinguished string, and determing whether the bytes are into arithmetic sequence in a half-strame. If it is, the bytes is packet order; if it is not, the bytes is character string.

2.5 Mining Process Based on Shell-control

When confronting a new protocol, users often have no idea about how to quickly identify the Protocol features, so attempting experiments are needed. In order to ensure the reliability and practicability, we refined the feature mining process with corresponding each step to a shell. Therefore users can guide feature extraction process with feedbacks.

2.6 Output Signature Files

Valid information in the process of protocol extracion is detailly recorded in signature files. The format is as follows:

Table 1 Signature file

0	1	2	3	4	5	6	7	8	9
0	0	2	3	*	*	*	*	*	*
1	*	*	*	*	*	*	*	*	*
2	*	*	*	*	*	*	*	*	*
3	*	0	0	0	0	4	E	*	*
4	0	0	0	0					

3 Test Results

System platform: Linux
Application-layer protocols: OpenVPN, N2N
Test metric: Recognition rate, Accuracy rate, Feature redundant
Test Results：

Table 2 Test results of OpenVPN and N2N

Test metric	OpenVPN	N2N
Number of tests	1000	1000
Recognition rate	100%	100%
Accuracy rate	97.1%	98.5%
Feature redundant	0.27	0.41

4 Conclusions

This paper analyzed the current feature extraction approaches and presented a set of simple and practical automatic feature extraction methods. Based on Apriori algorithm, this method could automaticly extract of frequent itemsets from training Trace, amend the results, and finally generat signature files of protocols. Besides, the speed of feature extraction has been greatly improved, which can better meet the practical needs.

References

1. Liu Xingbin, Yang Jianhua, Automated mining of packet signatures for traffic identification at application layer with apriori algorithm. Chinese Journal of Computers, 29(12): pp51-59 (2008).
2. Thomas K, Ander B, Nevil B. File-sharing in the Internet: a Characterization of P2P Traffic in the Backbone. UC, Riverside(2003).
3. Ran Hongmin, Chai Sheng, Feng Tie, Research of peer-to-peer botnets[J]. Application Research of Computers, vol.27. pp3628-3633 (2010).
4. Liu Pin, Zhang Senqiang. Research on the Detection before Scan[J]. Computer Engineering and Applications, vol.7.pp145-148 (2005).
5. Cheng Keqin, Deng Lin, Wang Jibo. Design and implementation of Windows personal firewall based on application layer. Journal of Hefei University of Technology,vol.05. pp695-700 (2011).