Protocol Identification System Based on Apriori Algorithm

Wang XiaoPeng*, Sun Yunxiao, Wang Bailing, He Hui and 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, the program automatically extracts four most common features of a protocol: characteristic string, session tag, packet length, and packet order. It is experimentally demonstrated that this progran can significantly improve the efficiency of feature extraction, and can be extended to other areas such as intrusion detection and extracting worm signature.

Keywords: *Protocol identification; Automatically extracted features of protocols; Apriori algorithm*

1. Introduction

With the rapid development of the Internet, there comes endless network applications, and so do the corresponding network protocols. In order to facilitate the allocation of network resources and security management, 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 and Sen proposed an automatic method to extract network traffic characteristics of application-layer protocols, and verified it in FTP, SMTP, POP3, HTTP protocols [2]. In this method, the protocol identification of the training Trace is based on a fixed port number. However, to strengthen the security, currently many communication protocols are using dynamic, disguised or encrypted ports to avoid network firewall filter and host limits [3,4].To solve this problem, researchers have proposed improved identification approach, based on deep packet inspection [5, 6]. By finding out the feature string in packet payload, this approach composed of the application-layer protocol feature library, and identify the network traffic by feature matching.

The premise of accurate identifition in DPI is to find out the feature of application-layer protocol,which have great impact on the accuracy of recognition rate, accuracy rate, and false recognition rate. There are two common methods of feature extraction:

(1) Reading the application-layer protocol RFC, and then obtaining features from RFC;

(2) Using package-catching tools, such as Wireshark, to capture packages in Protocol communication process, and then finding out application-layer features through manually comparing each packet with corresponding flow.

However, currently definition documents of many new application-layer protocol is not publicly informed, such as Thunder. Even if some protocols are opened, the frequent update visions bring greater challenge. Furthermore, 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 which only contain the same kind of protocols, and then process the data in training Trace; after that, using Apriori algorithm, extract frequent itemsets from the data, and extract feature of packet length; 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. Significances of the elements are shown in Table 1:

Table 1. Tuple4 represents the significances

Item	Description
File	PCAP file that contains bytes
Stream	Byte's TCP_STREAM In PCAP
Packet	The packet number of bytes in STREAM
Offset	Byte offset of the packet

In the progress of feature extraction, it is needed to compare bytes in different flows for many times. That is to say, to find every byte, the program needs to go through a PCAP file from the beginning, which is too often and low-efficient. In order to solve the above issues, this paper sets an index of 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.

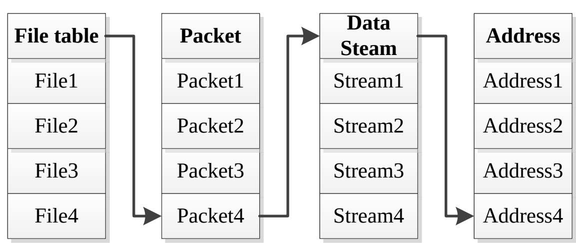


Figure 1. Index Map diagram

2.2. Four Characteristics of Protocols

After analysing feature extraction process, four common protocol features are summed up, which are respectively, character string, session tag, packet length, packet order [7-9]. 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.

Character string: it mainly includes the version number, control instructions, and so on. It can be obtained through horizontally contrasting multiple strings.

Session tag: in c/s structure-based applications, servers often need to maintain more than one session, and general client and server will generate a random number to mark the session. This number remains constant in one session. If respectively considering data received from the client and the server as each half-stream, the session tage can be found out by comparing those half-stream to extract the frequent itemsets.

Packet length: in most protocols packet length is represented by 2 bytes, since it has been able to express [0,65535] value [10], and enough to cover a variety of maximum transmission unit MTU. So it is assumed that all packet lengths are represented by 2 bytes. Two adjacent bytes are composed as packet length and then verify whether it can represent the packet length.

Packet order: packet order can be mined using output of character string. Assumpting the protocol holds 4 bytes to represent the packet order, in the first 256 packet of the data exchange process (0xFF), the first three bytes of the packet order must be 0. So these 3 bytes of 0 can be excavated in Character string. For the byte following each character string, it is needed to verify whether it is incrementing sequence. If it is, the packet order can be ectracted.

2.3. Frequent Itemsets Mining Algorithm

In frequent itemsets mining algorithms , Apriori algorithm is one of the more popular, which applies a recursive method to generate frequent itemsets [11-12]. And the core algorithm is briefly described as follows:

```
L1 = {large , 1-itemsets};
for (k=2; Lk -1&sup1; k++) do begin
 C =apriori-gen( Lk -1);
 for all transactions t&Icirc;D do begin
       Ct =subset( Ck ,t);
 for  (all candidates c&Icirc; Ct )  do
       c.count++;
       end
 L ={c&Icirc; Ck |c.count&sup3;minsup}
end
```

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.

Algorithm description:

```
for all lpackage in half_stream
 for all c in lpackage
       if con.a == c
             a.count++
       else
             new con;
             con.a = c;
             con.count = 1;
 end
end
```

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.

Algorithm chart in Figure 2.

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. The providing shell commandsares are as follows:

```
pacp filename            pcap//file directory
index -tcp -udp missionname  //Indexed
rindex filename          // File indexing
sindex                   // Displays the current
analy -s 1 <=3 >=4 (-ip 172.12.12.12 -r -s)// Select stream
-l N1 N2 N3 <N >N // Horizontal contrast each stream of the N-th
package
-d N//Vertical contrast in a stream of top n
-n num             // Minimum occurrences
-l num       //Analysis of packet length field
result -s -f -a// Output protocol analysis results
exit
```

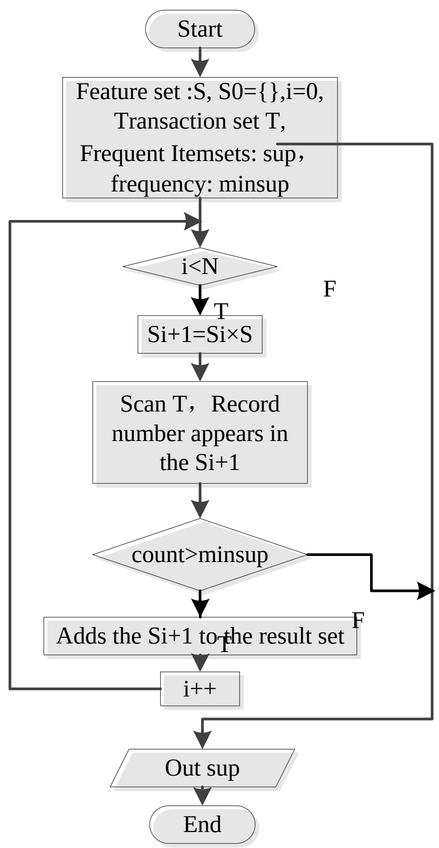


Figure 2. Algorithm Chart

2.6. Output Signature Files

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

3. Authentication protocol characteristics

System need to verify the characteristics after extracting the features of protocol , In order to describe the simulation clearly, we will give some definitions first :

Table 2. Signature File

0	1	2	3	4	5	6	7	8	9
0	00	2A	38	**	**	**	**	**	**
1	**	**	**	**	**	**	**	**	**
2	**	**	**	**	**	**	**	**	**
3	**	00	00	00	01	4D	E2	**	**
4	00	00	00	00					

Definition 1, Protocol feature set: $T = \{t_1, t_2, \cdots, t_n\}$, ti is a vector about agreement feature。

Definition 2, Feature vector of the agreement: $t = < P_1, P_2, P_3 >$

Definition 3, $P = \{L, C\}$, L is the packet length characteristics; C is the load characteristics

This paper include: the package vector compare and agreements vector of compare .

(1) Vector equal rules

Specific descriptions are as follows:

P_m and P_n denote the vector of Packet character. Then , we can get the Derivation formula

$$(L_m = L_n \&\& C_m = C_n) => P_m = P_n \tag{1}$$

t_m and t_n denote the vector of protocol character. then ,we can get the derivation formula:

$$(P_{m1} = P_{n1} \& P_{m2} = P_{n2} \& P_{m3} = P_{n3}) => t_m = t_n \tag{2}$$

(2)Vector containing the rules

Specific descriptions are as follows:

P_m and P_n denote the vector of Packet character. Then , we can get the Derivation formula:

$$(L_m = L_n \& C_n = NULL \& C_n = NULL) \| (L_m = NULL \& C_m = C_n) => (P_m > P_n) \tag{3}$$

t_m and t_n denote the vector of protocol character. then ,we can get the derivation formula:

$$(P_{m1} = P_{n1} \& P_{m2} = P_{n2} \& P_{m3} > P_{n3}) => (t_m > t_n) \quad (4)$$

Based on the rules and definitions, the process of agreement conflict detection in figure

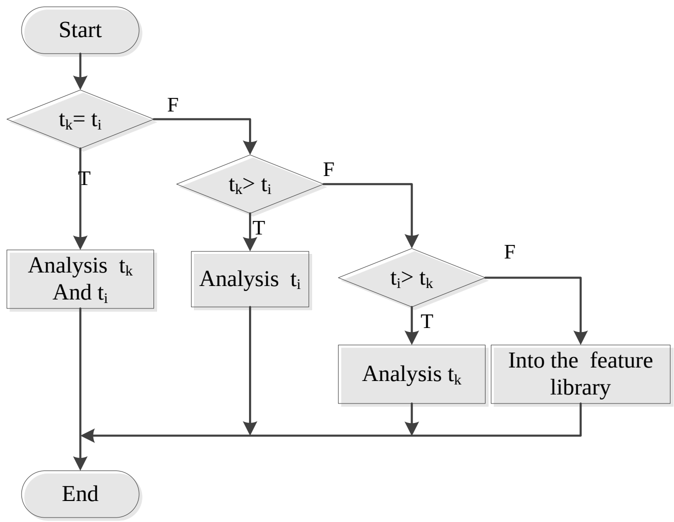


Figure 3. Protocol Verification Chart

4. Test Results

System platform: Linux

Application-layer protocols: OpenVPN, N2N

Test metric: Recognition rate, Accuracy rate, Feature redundant

Description：

1. OpenVPN is an SSL-based three-layer open source VPN protocols.

2. N2N is based on P2P VPN software, using UDP as the transport layer protocols.

3. Features redundant was adopted to for the measured output efficiency,and the lower the feature redundant system is the more excellent performance is. The number of total bytes in outputting characteristics of protocol is T; the number of bytes used for identifation is P. The feature redundant S= 1- P/T.

Test Results:

Table 3. 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

5. 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.

In future work, we will continue to provide support for Snort rules. That is, after the extraction, output the Snort detection rules to facilitate effectiveness of users. What is more, linking with the intrusion detection system could further ensure the real-time nature of intrusion detection system's rule base.

Acknowledgements

This research was supported by the National Science Nature Foundation of China under Grant No 61170262

References

[1] X. Liu and J. Yang, "Automated mining of packet signatures for traffic identification at application layer with apriori algorithm", Chinese Journal of Computers, vol. 29, no. 12, **(2008)**, pp. 51-59.

[2] K. Thomas, B. Ander and B. Nevil, "File-sharing in the Internet: a Characterization of P2P Traffic in the Backbone", UC, Riverside, **(2003)**.

[3] R. Hongmin, C. Sheng and F. Tie, "Research of peer-to-peer botnets", Application Research of Computers, vol. 27, no. 10, **(2010)**, pp. 3628-3633.

[4] M. Rajab, J. Zarfoss and F. Monrose, "A multi-faceted approach to understanding the botnet phenomenon", Proceedings of ACM SIGCOMM/USENIX Internet Measurement Conference (IMC'06), **(2006)**.

[5] Z. Leige, "Research of Identifying P2P Traffic Based on P2P Traffic Characters", School of Information Science and Engineering Central South University Changsha Hunan P.R.C., **(2009)**.

[6] L. Pin and Z. Senqiang, "Research on the Detection before Scan", Computer Engineering and Applications, vol. 07, **(2005)**, pp. 145-148.

[7] C. Keqin, D. Lin and W. Jibo, "Design and implementation of Windows personal firewall based on application layer", Journal of Hefei University of Technology, vol. 05, **(2011)**, pp. 695-700.

[8] Q. Guangchao and J. Ruiyu, "One Optimized Method of Apriori Algorithm", Computer Engineering, vol. 34, no. 23, pp. 196-199.

[9] L. Xingtao, S. Bing and X. Yingwen, "An improved Apriori algorithm for mining association rules", Journal of Shandong University, vol. 43, no. 11, **(2008)**, pp. 67-72.

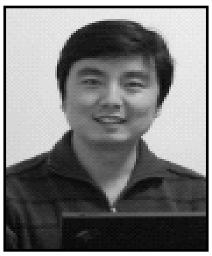
[10] Z. Xin and W. Xiao-dong, "Design and Implementation of Hybrid Broadcast Authentication Protocols in Wireless Sensor Networks", vol.2, **(2009)** January, pp6. 3-70.

[11] Y. E. Gelogo and R. D. Caytiles, "Threats and Security Analysis for Enhanced Secure Neighbor Discovery Protocol (SEND) of IPv6 NDP Security", vol. 4, no. 4, **(2011)** December, pp. 179-184.

[12] M. Venkatesulu and K. Kartheeban, "EAB-Euclidian Algorithm Based Key Computation Protocol for Secure Group Communication in Dynamic Grid Environment", vol. 3, no. 4, **(2010)** December, pp45-56.

Authors

Wang Xiaopeng, Engineer, Wang Xiaopeng ,He is working for Harbin Institute of Technology (abstract as HIT). His research is mainly on informa security, network security.



Wang Bailing is working for Harbin Institute of Technology (abstract as HIT) as an associate professor. He got the Ph.D. degree from HIT in 2006. His research is mainly on information security, network security, parallel computing.

Liu Yang, Associate Professor, Liu Yang, his research fields include Network information Security Technology, Internet of Things Security Technology, etc. He has participated in many projects of Ministry of Information Industry and National Science, and he has published over 20 academic papers in journals and conferences both home and abroad.