TELKOMNIKA, Vol.11, No.5, May 2013, pp. 2387 ~ 2393
ISSN: 2302-4046

A New Multi-tree and Dual Index based Firewall Optimization Algorithm

Cuixia Ni*, Guang Jin, Xianliang Jiang
Faculty of Information Science and Engineering
Ningbo University, Ningbo, China
*Corresponding author, e-mail: nicuixianihao@163.com, jinguang@nbu.edu.cn, jiangxianliang@foxmail.com

Abstract

Using statistical analysis strategy, a large-scale firewall log files is analyzed and two main characteristics, the protocol field and the IP address field, is extracted in this paper. Based on the extracted features and the characteristics of multi-tree and dual-index strategy, we design a better firewall optimization algorithm. Compared with the Stochastic Distribution Multibit-trie (SDMTrie) algorithm, our proposed algorithm can greatly decrease the preprocessing time and improve the searching and filtering process.

Keywords*: firewall, rule optimization, rule filtering*

1. Introduction

The packets filtering of a firewall is to classify the packets received into different classes according to the preconfigured rules. These classes are defined by multiple fields, such as the protocol, source IP address (sIP), destination IP address (dIP), source port number and destination port number, and so on. If the fields of a packet header match to a certain entry in the rule table of the firewall, then the packet will be handled according to the action field of the matched rule. However, statistical results show that 17% of the filters specify only one field, 23% specify three fields, and 60% specify four fields [1].

In previous work, researchers have put forward many firewall optimization algorithms and techniques [2] [3] [4]. These algorithms have enhanced the filtering rate and improved the ability to withstand attacks. However, it is difficult for an algorithm that is superior both on time and space.

In this paper, we deeply analyze the characteristics of packets passing through the firewall. And we find that the packets in general have following characteristics: (1) the protocol field only contains 6 finite values. (2) The IP address field indicates the aggregation characteristic. And the packets characteristics directly reflect the firewall rules' characteristics. Based on the above discovery, this paper uses the multi-tree and dual index to store the firewall rules and proposes a new search algorithm for firewall rules, namely Multi-Tree and Dual Index Search (MTADIS) algorithm. Compared with SDMTrie, our scheme can greatly decrease the preprocessing time, improve the filtering rate and reduce the comparison times in the searching process

2. Research Method

Researchers have proposed many firewall optimization algorithms in different aspects. How to improve the time and space performance of firewall has become the focus. Next, we will describe some well-known firewall optimization algorithms, such as Trie tree-based algorithm, Decision Tree-based algorithm, TCAM-based algorithm

2.1. Trie Tree-Based Algorithm

Trie tree has been widely used for packet classifications. Feng jun etc. [5] proposed the non-collision hash and jumping table Trie-tree (NHJTTT) algorithm. The NHJTTT turns the

Received January 11, 2013; Revised March 5, 2013; Accepted March 15, 2013

multi-dimensional packet classification into a two-dimensional packet classification. In practical application, the combination of source/destination port and protocol field is very limit. Thus only 3 fields could be used to construct a non-collision hash function. Only one time is needed to find a packet. But NHJTTT is likely to generate the space explosion. In order to further improve the NHJTTT, Feng jun etc. [6] proposed the SDMTrie algorithm. The SDMTrie contains four parts: constructing the non-collision hash function based on the destination/source port and protocol type field, splitting the 64bit IP address into four slices, converting the four slices into 16-bit binary value, and using the result to construct trie tree and place the classification rules index at the leaf nodes. The improvement can reduce the time and space complexity. Wang cheng etc. [7] proposed a 2D multibit tries (2DMTs), a dynamic programming algorithm which uses a bucketing scheme. Experiments show that the 2DMTs algorithm is superior to existing 2D packet classification schemes in terms of both memory requirements and the times of the memory accessing. In [8], the authors use range representation of prefixes and propose an efficient priority trie structure. Performance evaluation shows that the proposed priority trie is good in performance metrics such as lookup speed, memory size, update and scalability.

2.2. Decision Tree-Based Algorithm

Decision tree has also been widely used in packet classification. HyperCuts [9] is the typical decision tree algorithm. But HyperCuts ignores the characteristics of rules, and thus the temporal performance is not very good. In order to improve HyperCuts, Zhen qiang etc. [10] proposed a Multiple Decision Tree (MDT) algorithm. They analysed the characteristics of rules completely and created a multiple decision tree. Compared with HyperCuts, the MDT is superior in the preprocessing time, memory consumption and searching time. Separated HyperCuts could take up a lot of memory. Therefore Bo etc. [11] proposed a shared HyperCuts consumeing less memory.

2.3. TCAM-Based Algorithm

TCAM has been widely used in packets classification for its fast lookup speed and simple operation. However, TCAM is not well-suited for representing rules containing range fields. To solve this problem, Bremler Barr [12] proposed a SRGE (Short-Range Gray Encoding) algorithm. SRGE encodes range endpoints as binary-reflected Gray codes and then represents the resulting range by a minimal set of ternary strings. Experiments show that SRGE can significantly reduce the expansion of short ranges. However, the existing range encoding schemes, e.g., DRIPE [13] (Database Independent Range PreEncoding) or SRGE, are usually single-field schemes. In [14], the authors proposed an efficient multi-field range encoding scheme to solve the problem of storing ranges in TCAM. Experiments show that it uses less TCAM memory than existing single-field schemes. In packets classification, the existing range encoding schemes usually disregard the semantics of classifiers and loss significant opportunities for space compression. In [15], the authors developed a new approach to range encoding by taking classifier semantics into consideration. They viewed encoding as a topological transformation process from a colored hyper-rectangle to another where the color is the decision associated with a given packet. The proposed techniques can reduce the space utilization at a large scale.

3. MTADIS Scheme

3.1. The Assumption of Rules Characteristics

In practical applications, the firewall rules mainly contain seven fields as shown in Table 1.

Table 1. Firewall Rules Example

No.	Filtering Filed					Action
	Protocol	sIP	Source Port	dIP	Destination Port	
1	TCP	10.10.10.1	80	192.168.0.10	0~1023	Accept
2	TCP	10.10.11.2	808	192.168.0.11	23	Accept
3	UDP	10.10.98.2	90	192.168.45.9	8000	Accept
4	*	*	*	*	*	Deny

Due to the non-uniform representation of the port field, only the protocol field and IP address fields will be analyzed. The existing schemes mainly use the prefix relationship to design algorithms. We consider that the firewall rules in a particular field may have certain characteristics. In order to prove the assumption, we analyze the firewall log file datasets.

3.2. Firewall Dataset Analysis

In order to prove the assumption in section 3.1, we analyze one week (March 18th to March 24th) firewall log file of the Networks Center of Ningbo University. First, we extract data records with the same protocol in the dataset. The statistical results show that the protocols mainly contain six categories: TCP1, TCP2, UDP1, UDP2, ICMP1, ICMP2. Here TCP1 and TCP2 represent the data transmission between different regions respectively, and the same with the other two protocols. The quantity distribution is shown in Table 2.

Table 2. The number of each protocol

Day	TCP1	TCP2	UDP1	UDP2	ICMP1	ICMP2
1	10800384	154887	2090075	1031549	89015	46486
2	1394619	146520	3585140	1427945	114514	76566
3	3153299	180406	4656820	2368113	138570	84973
4	5139493	193741	1750506	2125359	134727	74224
5	4686715	188337	3677777	2143213	123380	87021
6	8182519	181033	3099410	3487625	134570	79788
7	10883442	184043	2996754	1377958	131095	74474
Total	44240471	1228967	21856428	13961761	865871	523532

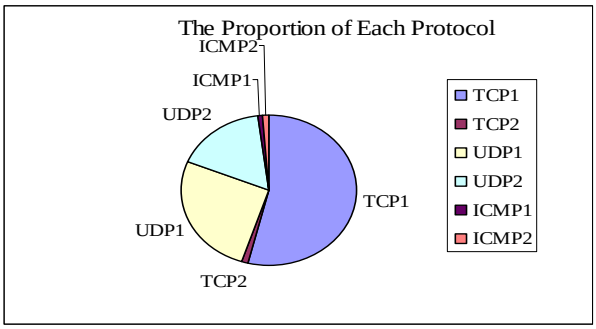

Figure 1. The Proportion of each protocol

Then, we calculate the proportion of each protocol in the total number of packets. As we can see in Figure 1, the proportion of TCP1, TCP2, UDP1, UDP2, ICMP1 and ICMP2 are 54%, 1%, 26%, 17%, 1% and 1% respectively.

Next, we analyze the aggregation relationship of IP address. First, we select out 20000 sIP and 20000 dIP respectively in each protocol, and then remove repeated IP addresses. The number of the rest of the sIP and dIP are show in Table 3.

Table 3. The number of sIP and dIP

	Protocol	Number		Protocol	Number
Source IP	TCP1	158	Destination IP	TCP1	689
	TCP2	230		TCP2	264
	UDP1	103		UDP1	256
	UDP2	158		UDP2	320
	ICMP1	22		ICMP1	287
	ICMP2	182		ICMP2	311
The max proportion of the total number		1.5%	The max proportion of the total number		3.445%

3.3. Algorithm Structure and Implementation

Based on the above analysis, we establish a dual index for Firewall rules. It can greatly accelerate the search efficiency. The specific scheme is shown in Figure 2. The MTADIS includes the establishment of the dual index and the searching process. It use a multi-tree data structure to store data and establish the first level index according to the protocol field. Next, for each protocol, it create the second level index according to the sIP.

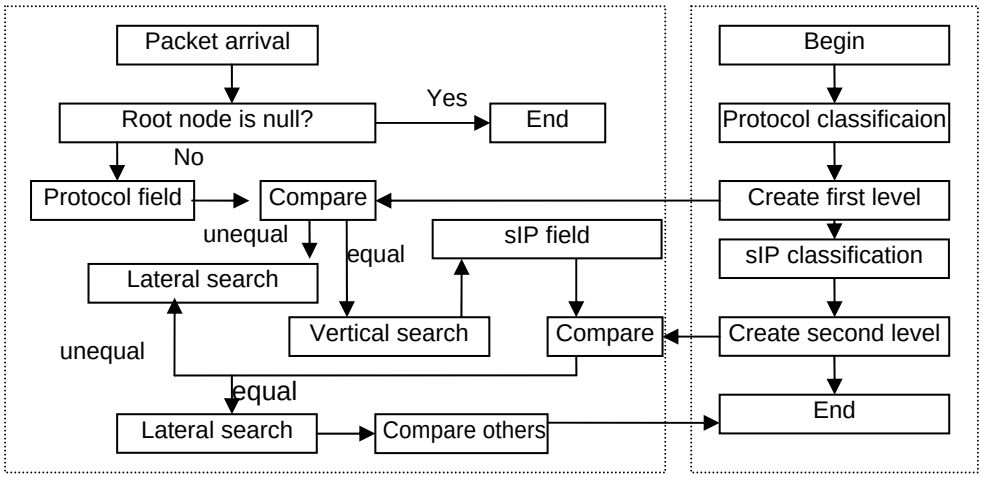

Figure 2. MTADIS scheme

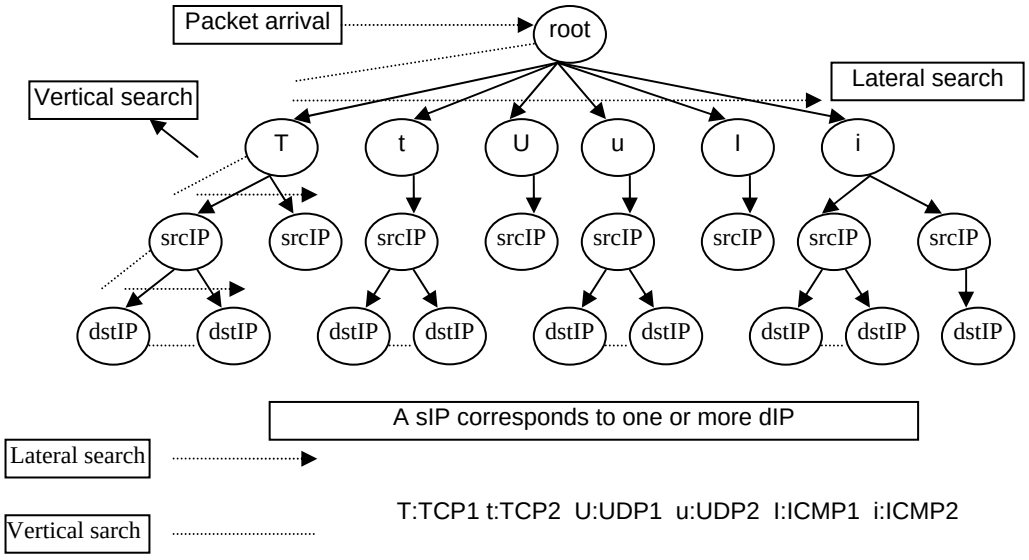

Figure 3. Data storage structure and search diagram

Figure 3 shows the data structure and search diagram. The path, from the root to a leaf node, is a complete rule. The search progress is as follows.

- When the packets arrive, judging whether the root node is empty. If it is true, then the search progress is end.
- Otherwise, comparing the protocol field with the first level index. If it is unequal, then executing lateral search.
- Otherwise, comparing the sIP with the second index. If it is unequal, executing lateral search.
- Otherwise, comparing with the other field and until to the end.

4. Simulate Experiments and Results

4.1. Firewall Rule Structure and Test Data

According to the characteristics of the Firewall datasets, we generate the datasets required in our experiments. The specific structure of the datasets is shown in Table 4.

Table 4. The Structure of The Datasets

Rule table scale				100		
	TCP1	TCP2	UDP1	UDP2	ICMP1	ICMP2
Proportion	54%	1%	26%	17%	1%	1%
Number	54	1	26	17	1	1
Number of sIP	1	1	1	1	1	1
Number of dIP for each sIP	54	1	26	17	1	1
Rule table scale				500		
Number	270	5	130	85	5	5
Number of sIP	5	1	5	5	1	1
Number of dIP for each sIP	54	5	26	17	5	5

When the rule table scale is 100, the quantity of TCP1, TCP2, UDP1, UDP2, ICMP1 and ICMP2 is 54, 1, 26, 17, 1 and 1 repectively. Table 4 shows the quantity of sIP and the quantity of dIP for each sIP. The same sIP is stored only once. When the scale of rule table is 500, we generate the rule set in the same way.

In order to guarantee each test data can be accurately matched, the basic test data we choose is the same with the firewall rule table. Then we get the certain scale test dataset by repeating the basic data 5 times, 10 times, 30 times….

4.2. Experimental Scene

Simulation experiments are consisting of two scenes. In the first scene, the scale of the firewall rule table is 100 and the test datasets are 100, 500, 1000 and 3000 respectively. In the second scene, the scale of the firewall rule table is 500 and the test datasets are 500, 1000, 3000 and 6000 respectively. In each scene, we compared the preprocessing time, filtering time and analyzed the comparison times with SDMTrie. In order to reduce deviation, we executed the algorithm 4 times or more for every test.

4.3. The Experimental Results and Performance Analysis

4.3.1. Preprocessing Time Analysis

In MTADIS, preprocessing time refers to the time required in building the multi-tree. For SDMTrie, preprocessing time consists of three parts: (1) Converting 64-bit IP address to a 16-bit binary number. (2) Creating a protocol table. (3) Creating a trie tree. For convenience, we calculate the total time of these parts. In order to reduce the experimental deviation, we use the average time to calculate the preprocessing time and the results are shown in Table 5. In which, T1~T4 represents the results of the four times execution and AVG represents the average value. As seen in Table 5, the average preprocessing time of MTADIS is relatively less than SDMTrie. The reason is that the structure of MTADIS is simpler than SDMTrie.

Table 5. The average preprocessing time

The first scene (ms)

MTADIS					SDMTrie				
T1	T2	T3	T4	AVG	T1	T2	T3	T4	AVG
3	5	4	3	3.75	15	11	11	10	11.75

The second scene (ms)

MTADIS					SDMTrie				
T1	T2	T3	T4	AVG	T1	T2	T3	T4	AVG
11	12	9	9	10.25	21	21	17	23	20.5

4.3.2. Filtering Time Comparison

In MTADIS, Filtering time is the time required in searching Multi-tree and determining its behavior. For SDMTrie, Filtering time is the time needed in searching Multibit trie and

determining its behavior. In order to eliminate the deviation and ensure the accuracy of the experimental results, the experiments conduct several measurements and use the average time to determine the final filtering time. In the first scene, the four times test results are show in Table 6 and the final filtering time is shown in Figure 4. Table 7 is the result of the four times test and Figure 5 is the final filtering time comparison in the second scene.

Table 6. The filtering time of different tests in the first scene

numbers	MTADIS				SDMTrie			
	T1	T2	T3	T4	T1	T2	T3	T4
100	70	64	62	68	75	64	77	75
500	172	170	175	159	266	249	247	243
1000	210	212	218	217	319	302	302	335
3000	496	484	488	488	740	695	626	621

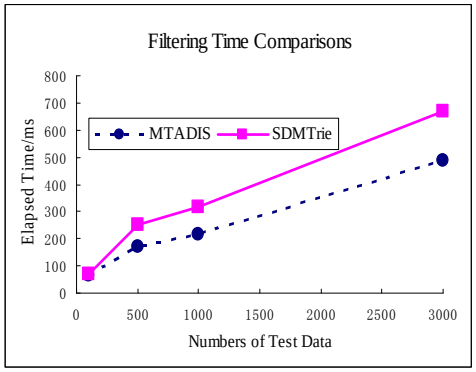

Figure 4. Filtering time comparisons

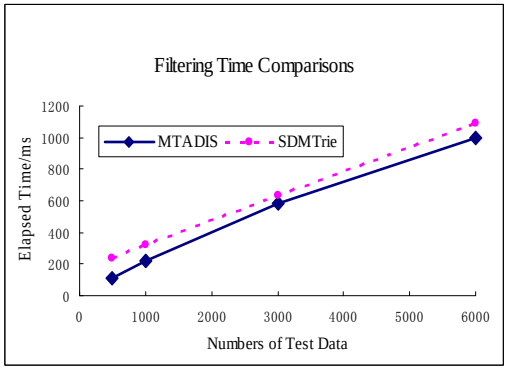

Figure 5. Filtering time comparisons

Table 7. The filtering time of different tests in the second scene

numbers	MTADIS				SDMTrie			
	T1	T2	T3	T4	T1	T2	T3	T4
500	117	107	108	112	245	223	242	245
1000	225	222	219	214	325	316	333	319
3000	604	588	572	570	589	676	631	635
6000	1006	987	1007	993	1033	1101	1114	1118

4.3.3. Comparison Times Analysis

To analyze the comparison times in the experiments, the first thing is to analyze the data structures of the two algorithms. MTADIS uses one data structure, Multi-tree, and each node represents an exact value of a certain field. For the specific three fields, the depth of the tree is 3. For each arriving packet, the compare times are between 3 and 8. SDMTrie uses table structure and trie structure. Trie is constructed by a 16-bit binary value. So the depth of the tree is 16. Each packet passing through the firewall needs to be compared 17 times at least and 22 times at most. From the above analysis, the comparison times of MTADIS are much less than SDMTrie.

5. Conclusion

This paper has deeply analyzed the firewall log files. From the analysis, we show that firewall rules have two important characteristics. According to the two characteristics, we propose the MTADIS algorithm. In order to verify the effectiveness of the algorithm, we conduct a set of experiments. The results show that the proposed algorithm can greatly decrease the preprocessing time, improve the filtering rate and reduce the comparison times in searching process.

However, as the new standard appearing [16] [17], the scheme also has drawbacks. The scheme can't change the composition of the firewall rules periodically. In the future work, we will improve and further optimize the proposed scheme.

Acknowledgements

This research was supported in part by Major Projects of National Science and Technology (2011ZX03002-004-02), Zhejiang Provincial Technology Innovation Team (2010R50009), Natural Science Foundation of Zhejiang Province (LY12F02013), Ningbo Natural Science Foundation (2012A610014), Ningbo Municipal Technology Innovation Team (2011B81002)

References

[1] Gupta P, McKcown N. Packet classification on multiple fields. *ACM Computer Communication Revier.* 1999; 29(4): 146-160.
[2] Alex XL, Eric T, Chad R. *Firewall Compressor: An Algorithm for Minimizing Firewall Policies*. Proceedings of IEEE INFOCOM. USA; 2008.
[3] Viktor P, Jan K. *Fast and Scalable Packet Classification Using Perfect Hash Functions.* Proceeding of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays. 2009; 229-236.
[4] Yeim-Kuan C. Efficient Multidimensional Packet Classification with Fast Updates. *IEEE Transactions on Computers.* 2009; 58(4): 463-479.
[5] Fengjun S, Yingjun P, Xuezeng P. Study on an Absolute Aon-collision Hash IP Classification Algorithms. *Journal of Communications (in Chinese).* 2005; 26(2): 87-99.
[6] Fengjun S, Yingjun P, Xuezeng P, Bin B. Research on a Stochastic Distribution Multibit Trie Tree IP Classification Algorithm. *Journal of Communications (in Chinese).* 2008; 29(7): 109-117.
[7] Wencheng L, Sartaj S. Efficient 2D Multibit Tries for Packet Classification. *IEEE Transaction on Computers.* 2009; 58(12): 1695-1709.
[8] Hyesook L, Changhoon Y. Priority Tries for IP Address Lookup. *Transaction on Computers.* 2010; 59(6): 785-794.
[9] Singh S, Baboescu F, Varghese G. Packet classification using multidimensional cutting. ACM SIGCOMM 2003, Karlsruhe, Germany. 2003: 213-224.
[10] Zhenqiang L, Shengliang Z, Yan M, Xiaoyu Z. Multiple Decision Tree Algorithm for Packet Classification. *Journal of Electronics & Information Technology.* 2008; 30(4): 975-978.
[11] Bo Z, Eugene S. *On Constructing Efficient Shared Decision Trees for Multiple Packet Filters*. Proceedings of IEEE INFOCOM. San Diego. 2010: 1-9.
[12] Bremler-Barr A, Hendler D. *Space-Efficient TCAM-based Classification Using Gray Coding.* Proceedings of IEEE INFOCOM. 2007; 1388-1396.
[13] Lakshminarayanan K, Venkatachary S, Rangarajan A. *Algorithms for Advanced Packet Classification With Ternary CAMs.* Proceedings of ACM SIGCOMM. New York. 2005: 193-204.
[14] Yeim-Kuan C, Chun-I L, Cheng-Chien S. *Multi-Field Range Encoding for Packet Classification in TCAM*. Proceedings of IEEE INFOCOM. 2011.
[15] Chad RM, Alex XL, Eric T. Topological Transformation Approaches to TCAM-Based Packet Classification. *IEEE/ACM Transactions on Networking*. 2009; 19(1): 1-14.
[16] Rezazadeh J, Moradi M, Samad Ismail A. Fundamental Metrics for Wireless Sensor Networks localization. *International Journal of Electrical and Computer Engineering (IJECE)*. 2012; 2(4): 452-455.
[17] Kamaruzzaman S, Fitri Maya P, Bachok MT, Zurina S. An Improved Optimization Model of Internet Charging Scheme in Multi Service Networks. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(3): 592-598.