International Journal of Security and Its Applications
Vol. 10, No. 7 (2016) pp.185-194
http://dx.doi.org/10.14257/ijsia.2016.10.7.16

Research on Network Intrusion Detection Method based on Regular Expression Matching

Yi Wang

Shandong Women's University, ji'nan 250300, China
123114821@qq.com

Abstract

Along with the rapid development of network technology and the rich variety of network applications, all kinds of network attacks and abnormal behavior is more and more rampant, which seriously interferes with the normal operation of the Internet. Based on regular expression, this paper studies the method of network intrusion detection, and proposes an improved grouping algorithm (IGA) to improve Yu algorithm based on the concept of expansion coefficient (EC). And we use the modified regular expression grouping algorithm to effectively group the regular expression set. Based on the open source software, we implement the improved regular expression matching engine, and carry out the grouping algorithm experiments and performance testing experiment. The experimental results we obtained is that the overall state number of IGA algorithm number is reduced by an average 27% compared with the overall state number of Yu algorithm under the condition of the same number of clusters, and the compilation time of MRegex 1.5 increases with the increase of the number of rules compared with PCRE. We can draw the following conclusions: (1) the total number of DFA state of the IGA algorithm is reduced in the same number of cases, and it also can reduce the memory space of the memory DFA, and the space complexity of the matching algorithm. 2) It can improve the efficiency of intrusion detection system. 3) Compared with PCRE (Perl Compatible Regular Expressions), the compilation time of MRegex 1.5 increases with the increase of the number of rules, and its matching speed can meet the requirements of the detection of gigabit network traffic.

Keywords*: Regular expression, Intrusion detection, modified regular expression grouping algorithm, road properties, Yu algorithm*

1. Introduction

In order to effectively curb the growing complexity and flooding of various network attacks, network management personnel must find the network traffic anomaly from the massive network monitoring data and timely to take effective measures to prevent traffic anomalies [1] . Network traffic anomaly detection technology research is the study of the abnormal network traffic appearing in what time and what place, and exception analysis is based on the network traffic anomaly detection technology to cause further analysis and diagnosis of network traffic anomalies.

As an important technology in network operation and maintenance and network defense, network anomaly detection and analysis technology can detect network anomaly, detect and locate the network faults and performance problems in time, and take corresponding measures to solve the problem, and they can effectively reduce the network failure time and reduce the loss of business interests [2, 3]. In the network information warfare, network traffic anomaly detection can detect the attack behavior of the other party in time, and then take the counter measures to ensure the normal operation of the network system of the important departments of our country. Therefore, the real-time detection of abnormal network traffic has important significance for improving the

ISSN: 1738-9976 IJSIA

reliability of the network and ensuring the quality of network service, and the research and application of real time detection and diagnosis technology are very urgent [4].

In order to solve the problem of the expansion of the state index when the regular expression is merged in the matching engine based on DFA, a modified regular expression grouping algorithm is proposed in this paper. The application of regular expression engine based on DFA can greatly improve the speed of intrusion detection, which can meet the performance requirements of gigabit network.

2. Research Contents and methods

In this chapter, we first introduce the knowledge of regular expressions, NFA and DFA. Then, we propose an improved regular expression grouping algorithm based on the expansion coefficient. Finally, we implement a regular expression matching engine based on DFA matching, and carry out related experiments

2.1. The Principle of Regular Expression, NFA and DFA

Table 1. Common Meta Characters

Symbol	Explain	Example	Explain of the example
^	If it appears at the beginning of the pattern, it indicates that the string is matched from the starting position of the string.	^abc	The match with the string "zabcxy" is a failure, and the string "abcxyz" match is successful.
$	If it appears at the end of the pattern, it indicates that the string is matched from the end position of the string.	abc $	The match with the string "zabcxy" is a failure, and the string "xyzabc" match is successful.
\	It is the transfer marker, which can make the next character be labeled as an escape character	\x	The 2 digital characters after the specified are sixteen ASCII code of a character
.	Match any single character except newline outside	xy.mn	The match with the string "xyzmn" and "xy.mn" are all successful.
?	Match 0 or 1 arbitrary single characters	?abc	The match with the string "abc" and "mabc" are all successful.
+	Match 1 or more of any single character	abc+	It matches the string "abcd", but it does not match the string "abc"
*	Match 0 or more of any single character	abc*	It matches the string "abc" and "abcd".

The principle of regular expression

The regular expression is also called a pattern, which is a single string that is specifically written and matches all the strings that match the rules in the string set [5, 7]. Regular expressions are text type, which is usually composed of common characters and special characters. Common characters include characters, numbers, underscores, and punctuation marks that are not used as a character. If we want to match the punctuation mark used as a meta character, we must use the escape character 'a' to transfer, for example, to match the character "*", "*" should be used to match. Ordinary characters in regular expressions match the characters in a text string. Special characters can be divided into Meta characters (Table 1), transfer characters (Table 2) and expression symbols (Table 3).

Table 2. Common Escape Characters

Symbol	Explain	Example	Explain of the example
\b	Match the word boundaries.	\bM.*\bXY\b	It matches the string "M.XY", and does not match the string "M_XY" , because "_" is not the word boundary.
\d	Match any single digits in 0~9	400030\d\d	It matches any two numbers in the string "400030"
\D	In contrast to the \d, and it matches any single non 0~9 number	\b\D*	Match all non-digit beginning strings
\w	Matches any single letter, number, or underscore character	\b\w\w\w\b	It matches the string consisting of 3 letters, numbers, and underscores.
\W	In contrast to the \w, and it matches any non-single letter, number or underscore character		
\s	It matches any single spaces, tabs, or other form feed blank characters		
\S	In contrast to the \s, and it matches any non-single spaces, tabs, or other form feed blank characters		
\r	Match carriage return character		
\n	Match newline character		
\t	Match table character		

Table 3. Expressions Type Symbols

Symbol	Explain	Example	Explain of the example
{m}	Match m individual characters	\d{2}	It is equivalent to “\d\d”
{m,}	Match at least m individual characters		
{m,n}	Match m to an individual characters	AB{1,2}	Match the string which appears 1 or 2 "B" after "A", such as “AB” and “ABB”
()	It's a packet sub match, and it can be expressed as sub expression	(ab)*c	Match“c”、“abc”、“abcbc”and so on
|	x|y	(ab)|c	Match ”c” and ”ab”
[]	Specify a range of characters	\(?0\d{2}[]-)?\d{8}	It matches the string "010-12345678" or “(010)12345678”, but it does not match the string "(010)-12345678". Because of the ')' and '-' in the model is a series; we only take one of them.

The principle of NFA and DFA

NFA (nondeterministic finite-state automata) is translated into a non-deterministic finite state automaton. In a sense, it is a machine model with simple memory ability [8]. A non-deterministic finite automaton NFA M is a five tuple, and its mathematical definition is $M=(K,\sum ,f,S,Z)$. These five elements are explained as follows:

K is a finite set of states, and its every element called a state;

$\sum$ is a finite set of input symbols, each of which is called an input symbol;

f is a conversion function, and it is also an image of the subset from $K\times\sum^{*}$ to K;

$S\subset K$, It is a non-empty initial state set;

$Z\subset K$, it is a final state set, and the final state can be accepted as the state or the end state.

DFA (deterministic finite-state automata) is translated as a deterministic finite state automaton. The relationship between DFA and NFA is close, and the mathematical definition of NFA is basically applicable to DFA, but it is obviously different. A deterministic finite automaton DEA M is a five tuple, and its mathematical definition is $M=(K,\sum ,f,S,Z)$. These five elements are explained as follows:

K is a finite set of states, and its every element called a state;

$\sum$ is a finite set of input symbols, each of which is called an input symbol;

f is a conversion function, and it is also an image from $K\times\sum$ to K, such as $f(k_i,a)=k_j(k_j\in K,k_i\in K)$, which represents the current state of k_i., when the input character is the a, it will be converted to the next state k_j, and we call k_j a successor state of k_i;

$S\subset K$, it is the only one initial state;

$Z\subset K$, it is a final state set, and the final state can be accepted as the state or the end state.

The regular expression is equivalent to NFA, that is, each regular expression has a NFA (which matches the same string), and vice versa. From the regular expression to the NFA conversion method has many kinds, but the most famous method is the construction algorithm that proposed by Thompson in 1968.

The NFA corresponding to the regular expression is constructed by the partial NFA of each sub expression, which uses different construction methods for different operators, which uses different construction methods for different operators [9, 10]. Part of the NFA does not match the state; instead, they have one or more of the arrow pointing to the null. Complete the construction process by connecting these arrows to the matching state.

Thompson construction algorithm: the regular expression is converted to NFA.

Input: the regular expressions “r” on the alphabet $\sum$

Output: NFA N which can be able to accept the L(r)

Method: First of all, decomposing the elements of the r, for each element of the r in accordance with the following rule 1 and rule 2 to process in order to generate NFA. If the symbol “a” in the r appears many times, then for the each time of “a” appears there is a need to generate a separate NFA. After that, according to the regular expression r, the NFA will be generated in accordance with the following rule 3.

Rule 1: for an empty marker 1, as shown in Figure 1 (a), it generates a NFA.

Rule 2: for the elements "a" in the alphabet $\sum$, as shown in Figure 1 (b), to generate the following NFA.

Rule 3: to make the NFA of the regular expressions r_1 and r_2 respectively is M_1 and M_2:

For regular expression $r_1|r_2$, according to the Figure 1(c) way to generate the NFA N($r_1|r_2$), and for the regular expression r_1r_2 , according to the Figure 1(d) way to generate the NFA N $(r_1|r_2)$, and for the regular expression $(r_1)^*$, according to the Figure 1(e) way to generate the NFA N(r_1^*).

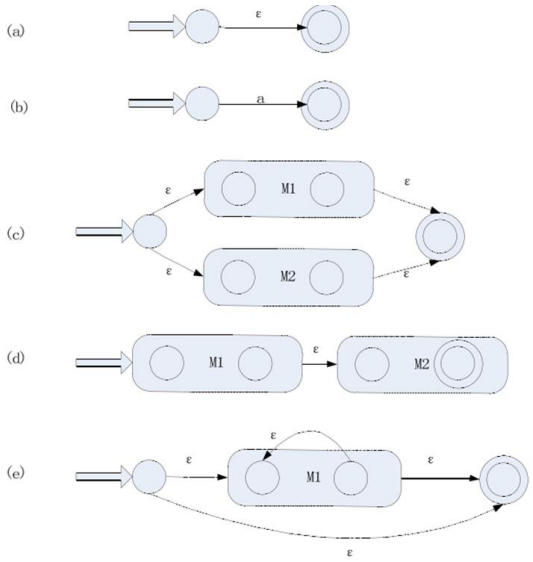

Figure 1. Thompson NFA Construction Algorithms

With the rapid development of network technology and network applications, the application of deep packet inspection is more and more widely used in the fields of firewall, intrusion detection system, traffic classification, network audit, anti-virus software and so on. Regular expressions can be used flexibly and efficiently to describe the string, and gradually replace the exact string pattern matching, which is an important method to detect deep packet. In the open source intrusion detection system Snort, Bro and many network security products have been widely used.

The regular expression method is based on the compound type condition, and it can be used to express a set of rules, and it also supports the association of rules, which can be used to match the characteristics of the data packets more accurately. Using regular expressions to describe the various protocols and attacks in the application layer are more accurate and more effective than the traditional method of extracting accurate matching strings. However, compared with the multi pattern matching algorithm based on keywords, the regular expression matching algorithm has many defects, such as large space cost, high occupancy load and slow processing speed, so many research works are devoted to improving the performance of regular expression matching algorithm.

2.2. Regular Expression Grouping Algorithm

The work principle of regular expression matching engine is to transform the regular expression into finite state machine, and then use the finite-state machine to scan the content of the test. Regular expression matching has two ways, one is a deterministic

finite state automaton (DFA), and the other is a non-deterministic finite state automaton (NFA). NFA is characterized by high efficiency of storage space, the state transition path is not unique, the matching of the time efficiency is low; the characteristic of DFA is that the state transition path is unique, and the matching time efficiency is high, but when the number of rules is increased, the number of DFA States is exponential, and the demand of storage space is increased, which leads to that the DFA is difficult to be practical.

For the problem of state space explosion in the presence of regular expression matching, a variety of DFA based storage efficient regular expression matching algorithm is proposed to reduce the storage space requirement of DFA. D2FA, XFA and other methods are through the elimination of redundant DFA state transfer, which can achieve good compression effect, but cannot reduce the DFA state number. Yu *et. al.,* first proposed DFA packet algorithm (hereinafter referred to as the Yu packet algorithm), but there can be improvement in search strategy. Beechi *et. al.,* put forward a hybrid automaton engine with the advantages of both NFA and DFA. Li Lipeng *et. al.,* proposed a regular expression matching algorithm based on Bloom filter, which can save storage space and improve matching speed, but it has a certain error matching rate. There are researchers who design and improve the FPGA architecture based on the regular expression matching technology, but the FPGA clock frequency is lower than the general-purpose processor, cannot be scalable.

Yu algorithm

Yu *et. al.,* first proposed the concept of the interaction between regular expressions, using this concept to construct the relation graph of regular expression sets, using heuristic search strategy to group.

The approximate process of Yu algorithm is as follows:

1) To calculate the interaction between any two elements in the set of regular expressions, and to construct the function relation graph G of the regular expression set

2) Select a regular expression that is minimal with other regular expressions, and join it in the group NG.

3) Select a regular expression R which is the least number of connections with the group NG;

4) The NG and R merged into DFA, if the DFA size exceeds the threshold value, jump to step 5, otherwise R will be added to the NG;

5) Determine if all of the regular expressions in the G are checked, and if so, then jump to the step 3, otherwise the NG group is finished;

6) Determine whether there is a regular expression without grouping in the G, if there is a jump to step 2, otherwise the algorithm is completed.

Improved grouping algorithm

In this paper, we propose an improved grouping algorithm (IGA) to improve Yu algorithm based on the concept of expansion coefficient (EC), which can reduce the memory space occupied by DFA more effectively. In order to make use of the interaction between regular expressions to construct a regular expression set, we give the definition of expansion coefficients.

Definition 1: for the regular expression R_1, R_2, the expansion coefficient calculation formula is as follows:

$$EC(R_1R_2) = \frac{\#DFA(R_1R_2)}{\#DFA(R_1)+\#DFA(R_2)} \quad (1)$$

When the expansion coefficient EC>1, it represents the interaction between R_1 and R_2. The greater the value of the EC is, the more powerful the two regular expressions are compared to the previous state. So we need to try to isolate R_1 and R_2, and try not to assign them to the same DFA group, thereby reducing the total number of states. When the expansion coefficient EC< 1, it indicates that R_1 and R_2 do not interact, you can put R_1 and R_2 in the same group. When the expansion coefficient EC=1, it indicates that the R_1 and R_2 after the merger of the state number is equal to the number of states before the merger, but the possibility of such a situation is very small.

On the basis of the concept of the expansion coefficient, the improved algorithm is designed:

Input: a regular expression collection R the size of which is n.
Output: the grouping results for R.

For more intuitive description of the algorithm, assuming that there is a collection of 5 regular expressions to calculate the expansion coefficient between regular expressions, and construct the interaction diagram, as shown in the following diagram:

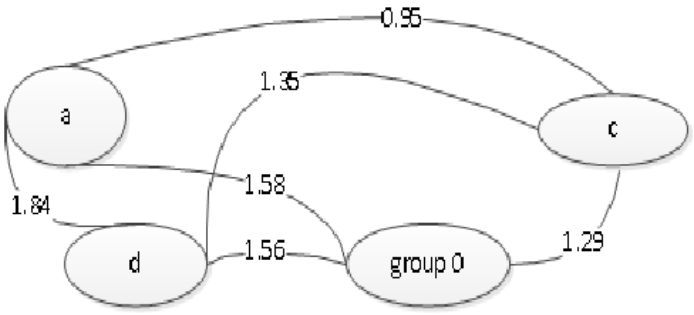

Figure 2. Obtain Children Nodes of a Group

The "Size-limit" is the DFA state number threshold for each packet. Select the vertex C which is associated with the group 0 vertices and the expansion coefficient of which is minimum. Then delete the vertex c from figure G, and add it to the Group 0, continue to repeat this process until the Group 0 adds any one of the smallest expansion coefficient will exceed "Size-limit". If the state number of Group 0 is more than "Size-limit", a new Group 1 is established, and then the vertices are added to the Group 1 until all vertices are grouped.

2.3. Regular Expression Matching Engine

First step: we need to extract the regular expression marked as PCRE from the open source intrusion detection software Snort, and eliminate the regular expressions that repeat the number of times over 1000, and save it to the new regular expression library file.

Second step: using the improved regular expression grouping algorithm proposed in the last section to compile all the regular expressions in the regular expression library file, the packet DFA is obtained.

The third step is to capture the network packets in real time. The regular expression matching engine is matched to the payload part. If the matching is successful, it will output the alarm information according to rule number.

2.4. Experimental Method

Grouping algorithm experiment

The hardware configuration of the experiment is Intel Dual-Core 2.5 GHz CPU, and the memory is 4.OOGB, and the operating system is Windows 7. The regular expression is converted to DFA using the open source software Regex 1.4 which is developed by Becchi, and the Yu algorithm and IGA algorithm are implemented by C++ language, named as MRegex 1.5. The regular expression sets are respectively e gathered 100 rules from L7-filter, the 100 rules concentrated extract of Snort rules and 100 rules of the Bro rules. And the grouping results of IGA algorithm and Yu algorithm are compared on different rule sets.

Performance testing experiment

Step 1: select 100, 1000, and 10000 regular expressions from the Snort rule base, and use Tcpdump to crawl the network traffic of the 1GB in the lab network and save it as a file.

Step 2: use the PCRE library used in intrusion detection software Snort to parse the regular expression, and load the regular expression, and then match the files saved in step 1; record the experimental results, and repeat the average of the 10 time.

Step 3: use the regular expression engine MRegex 1.5 to parse the regular expression, and match the files saved in step 1 after compiling the regular expression, and record the results of the experiment, repeat the average of 10 times.

3. Experimental Results and Analysis

The experimental results of the three rule sets are embodied in the following three tables.

Table 4. Comparison of Grouping Results using L7_filter

Grouped threshold	Yu algorithm		IGA algorithm		Reduction ratio (%)
	Grouping number	State number	Grouping number	State number	
7000	6	26156	6	21894	18.13
9000	5	34390	5	27458	21.94
11000	4	39562	4	29564	27.14

Table 5. Comparison of Grouping Results using Snortr

Grouped threshold	Yu algorithm		IGA algorithm		Reduction ratio (%)
	Grouping number	State number	Grouping number	State number	
7000	6	27953	6	22406	21.77
9000	5	32345	5	26265	20.36
11000	5	38084	5	28633	26.42

Table 6. Comparison of Grouping Results using Bro

Grouped threshold	Yu algorithm		IGA algorithm		Reduction ratio (%)
	Grouping number	State number	Grouping number	State number	
7000	4	14453	4	9443	36.37
9000	4	17624	4	11776	34.38
11000	3	21532	3	15786	27.09

From the comparison result of Table 4, Table 5 and Table 6 we can see that under the condition of the same number of clusters, the overall state number of IGA algorithm

number is reduced by an average 27% compared with the overall state number of Yu algorithm. The overall state number of DFA is reduced, which can reduce the memory space of DFA and reduce the space complexity of the matching algorithm. Compared with the Yu algorithm, the main improvements of the IGA algorithm are the following 3 points:

(1) IGA algorithm uses the expansion coefficient to calculate the relationship between the regular expression, and the Yu algorithm is simply based on the number of the regular expression before and after the number of States to indicate the relationship between the regular expressions.

(2) The selection criteria of IGA algorithm joining the regular expression to the new group is the least of the expansion coefficient, while the selection criteria of Yu algorithm is the minimum number of connections.

(3) IGA algorithm updates the expansion coefficient between the new group and other regular expressions, while the Yu algorithm does not update the number of connections between regular expressions.

The results of the performance test can be expressed in the following Table 7.

Table 7. Performance Comparison between PCRE and MRegex1.5

Rule number	PCRE		MRegex1.5	
	Compile	Matching	Compile	Matching
2000	0.0002	9.21	0.03	2.76
4000	0.0002	9.40	0.11	2.79
7000	0.0002	9.20	0.18	3.00
10000	0.0002	9.35	0.57	4.16
12000	0.0002	9.40	0.79	3.78

As can be seen from Table 7, the compilation time of MRegex 1.5 increases with the increase of the number of rules compared with PCRE, but the matching time is much less than that of PCRE. From a theoretical perspective, the 1GB file transfer takes less than 4 seconds, and the matching speed of MRegex 1.5 can meet the requirements of the detection of gigabit network traffic.

4. Conclusion

In this paper, we study the method of network intrusion detection based on regular expression matching, and we can find that in the case of the same number of groups, the total number of DFA state of the IGA algorithm is reduced by more than 27%, which can reduce the memory space of the memory DFA, and reduce the space complexity of the matching algorithm. At the same time, the IGA algorithm can increase the storage space and compile time, but the matching performance is greatly improved, and the efficiency of intrusion detection system is improved. And the compilation time of MRegex 1.5 increases with the increase of the number of rules compared with PCRE, but the matching time is much less than that of PCRE, and the matching speed of MRegex 1.5 can meet the requirements of the detection of gigabit network traffic.

References

[1] Yan B., Wang S. and Bao Y., "An Improved Grouping Spectrum Allocation Algorithm in Cognitive Radio [M]", // The Proceedings of the Third International Conference on Communications, Signal Processing, and Systems. Springer International Publishing, **(2015)**, pp. 89-96.

[2] He G., Wang Y. and Wu X., "A regular expression grouping algorithm based on partitioning method[C]",// Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on. IEEE, **(2012)**, pp. 271-274.

[3] Wei Q., L. Yun zhao and C. Yan jie, "Regular Expression Grouping Algorithm Based on Graph Partitioning [J]", Computer Engineering, vol. 38, no. 18, **(2012)**, pp. 137-139.
[4] Dequevy J. J., "Computer network intrusion detection: US, US8631496 [P]", **(2014)**.
[5] S. Veetil and Q. Gao, "Chapter 18–Real-time Network Intrusion Detection Using Hadoop-Based Bayesian Classifier [J]", Emerging Trends in ICT Security, **(2014)**, pp. 281-299.
[6] Guillén E. P., Parra J. R. and Mendez R. V. P., "Improving Network Intrusion Detection with Extended KDD Features [M]", // IAENG Transactions on Engineering Technologies. Springer Netherlands, **(2014)**, pp. 431-445.
[7] Jose A. E. and Gireeshkumar T., "Gigabit Network Intrusion Detection System Using Extended Bloom Filter in Reconfigurable Hardware [M]",// Proceedings of the Second International Conference on Computer and Communication Technologies. Springer India, **(2015)**, pp. 11-19.
[8] Podkolzin A., Ivanovic L. and Bolotov A., "Impulse regular expression matching:, US8650146[P]", **(2014)**.
[9] Hisashi T., Takayoshi S. and Junichi T., "Network Traffic Screening Using Frequent Sequential Patterns [M]", Japan: Springer, **(2012)**, pp. 363-375.
[10] Avalle M., Risso F. and Sisto R., "Efficient multistriding of large non-deterministic finite state automata for deep packet inspection[C]",// Communications (ICC), 2012 IEEE International Conference on. IEEE, **(2012)**, pp. 1079-1084.

