An improved QS algorithm for pattern matching of bit stream

Tao Zhao[1,a], Yang Jianbo[2], Liu Peng[3,b]

[1]Information Countermeasure Department, Aviation University of Air Force, Changchun 130022, China

[2]Information Countermeasure Department, Aviation University of Air Force, Changchun 130022, China

[3]Information Countermeasure Department, Aviation University of Air Force, Changchun 130022, China

[a]email:1272247952@qq.com, [b]email:13874534@qq.com,

Keyword: intrusion Detection; Pattern Matching; Bit Stream; QS Algorithm; Fast-skip

Abstract. On the condition of bit stream, for the reason of simple character sets of target strings and Pattern strings, the probability of appearance of absolutely bad character is very low. The bad character shift is not suited for pattern matching, because of it's low efficiency. This paper gives the definition of bad string on the base of bad character shift, as well as comes up with bad sting shift. In the improved QS algorithm, the bad character shift is regarded as the trigger condition of bad string shift, and the bad string shift can expand the average skip length of bad character shift. In this way it can make the matching widows skip longer.

1 Introduction

The core of most detection engine of IDS based on regulation is pattern match algorithms. Pattern match algorithms can be divided into single pattern matching algorithms and multiple pattern match algorithms[3,4]. And among these algorithms single pattern matching algorithms is widely used in IDS. So it's important to lucubrate the single pattern matching algorithms[5-7]. Applying on actually engineering project, the most common of these algorithms are KMP(Knuth-Morris-Pratt)[8], BM(Boyer–Moore)[9], QS(Quick Search)[10] and so on. And QS is characteristic of simple, fast and easy to implement. It's the most commonly used algorithm.

The core of QS is bad character shift[11].But the matching efficiency is very low for pattern matching of bit stream,because of the simple character set {0 1}. The existed methods[12-14] are still used 1 bit character as the unite to generate the shifting length. They don't solve the problem of simple character set.

This paper present a new shifting regulation. Which generate shit set based on strings instead of 1 bit character for the pattern matching of bit stream.

2 BF and QS algorithm

This paper define the target string is $T=[T_1,T_2...T_{l_T}]$,and the pattern string is $P=[P_1,P_2...P_{l_P}]$,where l_T is the length of target string,and l_P is the length of pattern string.Both T_i and P_i choose from {0 1}.

Pattern matching is to find the the positions of substrings of T that is match with P,which is given by

$$Po=\{i\,|\,T_iT_{i+1}..T_{i+ml-1}=P,1\le i\le l_T-l_P+1\} \tag{1}$$

BF is a algorithm that when meeting mismatch the matching widow shift 1 bit length.But QS will use the next 1 bit character to compute shifting length, which is named as bad character shift:

$$shift(T_{i+l_p})=\begin{cases} ml+1,T_{i+l_p}\notin P \\ \min\{j\,|\,T_{i+l_p}=P_{l_p-j+1},1\le j\le l_p\} \end{cases} \tag{2}$$

It's easy to hnow that the average shifting length of BF is 1 bit. And through calculation,the average shifting length of QS is only 2 bit for pattern matching of bit stream.

3 The improved QS algorithm

3.1 Bad String and Bad String Shift

Figure 1 shows a process of BF algorithm. There is a continuous mismatch from matching widow 1 to matching widow 5. As we can see,matching widow skip string “0110”. If we regard “0110” as a bad character , refering to bad character shift, get the matching widow 6.

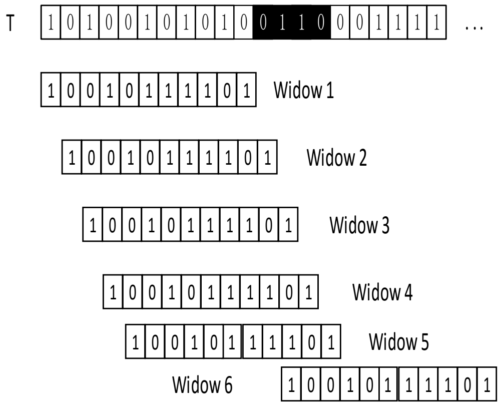
T
1 0 1 0 0 1 0 1 0 1 0 0 1 1 0 0 0 1 1 1 1 …
1 0 0 1 0 1 1 1 1 0 1 Widow 1
1 0 0 1 0 1 1 1 1 0 1 Widow 2
1 0 0 1 0 1 1 1 1 0 1 Widow 3
1 0 0 1 0 1 1 1 1 0 1 Widow 4
1 0 0 1 0 1 1 1 1 0 1 Widow 5
Widow 6 1 0 0 1 0 1 1 1 1 0 1

Fig.1 Bad String

This paper define such string as General Bad String(Bad String for short): if there is a continue mismatch from matching widow i to $i+L$,the L-length substring located in $i+l_p$ of target string is General Bad String. It’s easy to prove that Bad String has the following characteristics:

a. The length of bad string depends on the time of continuous mismatch.And the range of values is from 0 to l_p-1.

b. If the bad string match with a substring of pattern string, the substring mustn’t be located in the end of pattern string.

Refering to the bad character shift, this paper present a “bad string shift”:

$$J=\{j\,|\,P_jP_{j+1}..P_{j+LBS-1}=b_1b_2...b_{LBS}, j\in[1,ml-LBS]\} \tag{3}$$

$$shift(BS)=\begin{cases} ml-\max(J)-L_{BS}+1, J\notin\phi \\ \quad ml-L_{BS}+1, J\in\phi \end{cases} \tag{4}$$

As shown in figure 1, the length of shifting is 8 bit, while the BF algorithm is 1 bit, what’s more, the QS algorithm is 2 bit. So we can make the conclusion that bad string is a efficient shifting rule.

3.2 A new BF algorithm based on bad string shift

As can be see from the definition and characteristics of bad string, because of the varied length of bad strings,so there should be different shift sets corresponding to different length. That will cause a great wast of storage space. In order to solve the problem, this paper present the definition of Special bad string:

Assuming that the length of special bad string is l_{BS}. When the time of mismatch is greater than l_{BS}, the special bad string is the l_{BS}-length long string after the first mismatching widow.

The steps of the BF algorithm based on bad string shift is as follows:

Step 1:

Generate shift set according to bad string shift and l_{BS}.

Step 2:

The initial position of matching widow is the first character of the target string(p=1), and there is a variable *R* to record the mismatch time.When $p<l_T-l_P$, if the strings in matching widow match with each other,records the position of matching widow, and makes *R* equal to 0. Then the matching widow shifts 1 bit forward. But if not match, makes *R* is equal to *R*+1.If $R>l_{BS}$, shifting the matching widow according to the bad string and bad string shift. And then makes *R* equal to 0.If $R\le l_{BS}$, the matching widow shifts 1 bit forward.

Step 3:

When $p \geq l_T - l_P$, stop the string-match process and output the data we need.

3.3 A new QS algorithm based on bad string shift

It's easy to know that improving the growth rate of *R* is a way to increase the frequency of bad string. What's more, it also make the algorithm efficient because of reduce the frequency of shift.

If string matching algorithm replace BF algorithm with QS algorithm during the process of appearance of bad string(the growth of R), the frequency of shift can be improved. So this paper present the new QS algorithm based on bad string shift. The process of the algorithm is shown on figure 2：

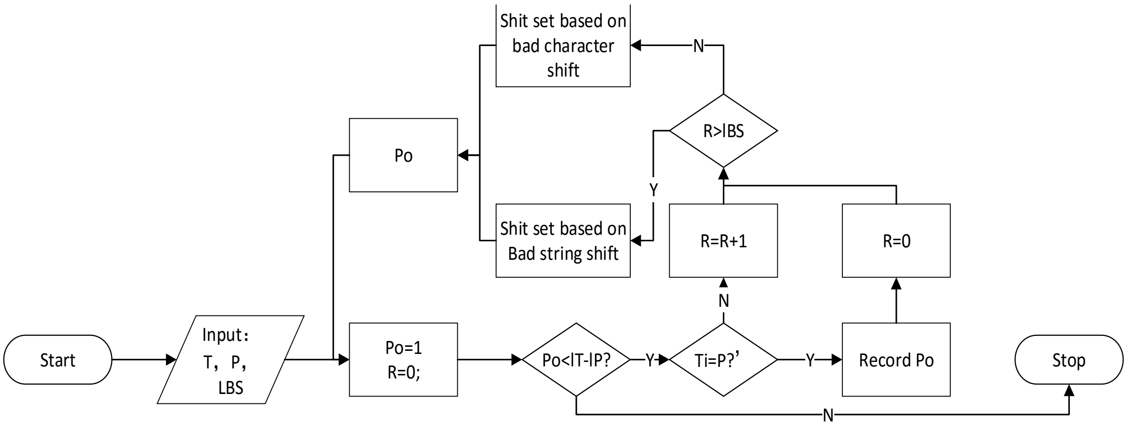

Fig.2 The flow chartof BQS

It can be seen from figure 2 that the process is similar to the BF algorithm based on bad string shift, except the growing algorithm of *R*, which is more efficient that take advantage of bad character shift of QS algorithm. And there is a example of BQS used in string match compared with figure 1:

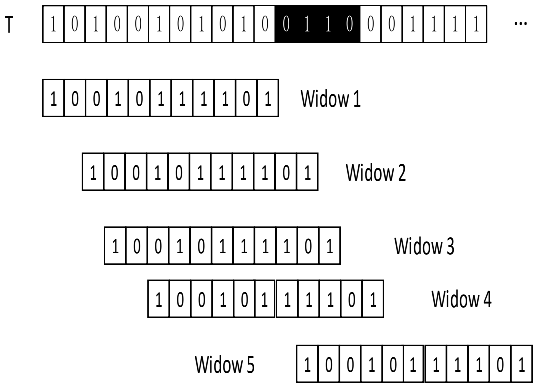

Fig.3 The process of BQS

Both of figure 1 and figure 3, the l_{BS} is set as 4 bit.Before the appearance of bad string the time of mismatch of “BF” and “QS” respectively are 5 and 4. What's more, as we can see in the figure 3,the length from matching widow 1 to 4 is 5 bit is longer than l_{BS}, which is 4 bit. But the length from matching widow 1 to 3 is 3 bit is shorter than l_{BS}. So the actual matching widow is only 3. The matching widow 4 is not necessary to judge is matching or not. Define such widow as virtual match widow.

4 Algorithm performance

4.1 The relationship between algorithm performance and l_{BS}

The length of target string is set as 2000 bit. And the range value of pattern strings' length is from 8 bit to 12 bit.Recording the frequency of l_{BS} which makes the lowest shifting times. By 1000 times of Monte Carlo experiments, the conclusion is shown on figure 4.

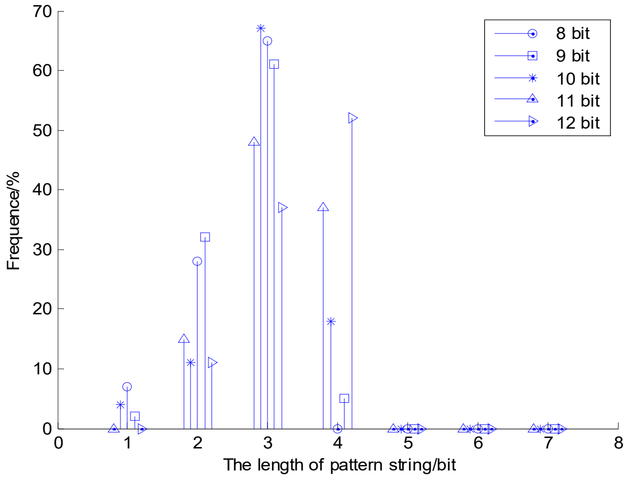

Fig.4 The relationship between algorithm performance and l_{BS}

It's easy to see that different length has different best value of l_{BS}. But we can get it through the Monte Carlo experiment.

4.2 Algorithm contrast

The length of target string is set as 10000 bit and the the length of pattern string is set as 4~40 bit. Using BF、QS、BBF and BQS eparately, and record the shifting times of matching widow. By 1000 times of Monte Carlo experiments, the conclusion is shown on figure 5, and the BBF is the shorthand of BF algorithm based on bad string shift, the BQS is the shorthand of QS algorithm based on bad string shift.

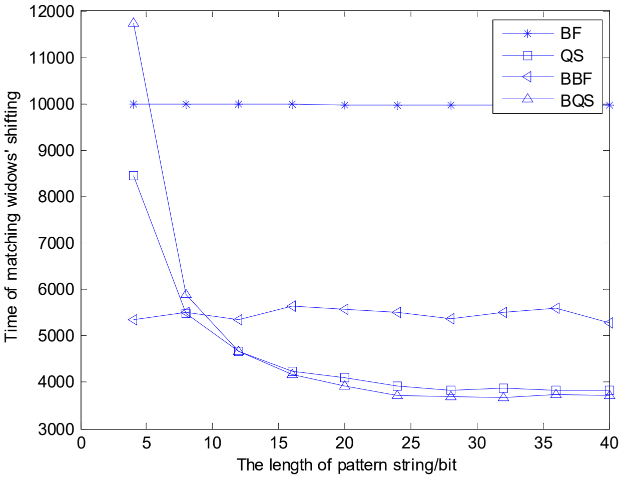

Fig.5 Comparison of Algorithm performance

As we can see, the shifting time will decrease the length of pattern string increase for BBF and BQS. But for BF and QS the trend is relatively stable. What,s more, when the length of pattern string is longer than 7 bit, the matching time of BBF and lower than BF and QS.And BQS is of the lowest matching time. So it is the most efficient algorithm when the length of pattern string is longer than 7 bit.And throng calculation, the shifting time can be reduced by 3.67% compared with QS.

5 Conclusion

For the QS algorithm is not suitable for the pattern matching of bit stream. This paper present the definition of bad string and the bad string shift. The QS algorithm based on bad string shift is efficient for pattern matching of bit stream. And it also proved that even for BF algorithm, when used the bad string shift,it can be more efficient than QS algorithm. This paper presents a new way to improve the efficient of QS algorithm for the pattern matching of bit stream.

References

[1] PAN C, YANG L H, GONG Wei-Hua, GU Hui, CHEN Min-Zhi .Schema Matching Research Progress: A Brief Survey[J]. Computer Systems & Applications, 2010,19(11):265-275

[2] DU X K,LI G H, WANG J Q, TIE J, LI Y H. Schema Matching Method Based on Information Unit[J]. Journal of Software, 2015,26(10): 2596-2613

[3] LIU W G, HU Y G. DHSWM: An improved multi-pattern matching algorithm based on WM algorithm[J].Journal of Central South University (Science and Technology),2011,42(12):3766-3767

[4] SONG T, LI D N, WANG D S, XUE Y B. Memory Efficient Algorithm and Architecture for Multi-Pattern Matching[J]. Journal of Software, 2013,24(7):1651-1665

[5] Zhong Qiuxi, Wan Hui, Xie Peidai, et al. An efficient packet pre-filtering algorithm for NIDS[J]. Lecture Notes in Electrical Engineering, 2012, 126: 113-120.

[6] K. Prabha*and Dr. S. Sukumaran. Single-Keyword Pattern Matching Algorithms for Network Intrusion Detection System. International Journal of Computer and Internet Security2013,5(1): 11-18

[7] JIANG Q M, WU N, LIU W H.A Fast Pattern Matching Algorithm in Intrusion Detection System[J]. Journal of Xi’an Jiaotong University,2009,43(2):59-60

[8] Knuth D E, Morris H, Pratt V R. Fast pattern matching in strings[J]. SIAM Journal on Computing, 1977, 6(2): 323−350.

[9] Boyer R S, Moore J S. A fast string searching algorithm[J].Communications of the ACM, 1977, 20(10): 762−772.

[10] Sunday D M. A very fast substring search algorithm[J].Communications of the ACM, 1990, 33(8): 132-142.

[11] MA Z F,YAN G S, GUO G F. A fast improved pattern matching algorithm based on BM[J].Journal of Control and Decision, 2013,28(12):1857-1858

[12] JIN L.Study on Bit Stream Oriented Unknown Frame Head Identification[D]. Shanghai Jiao Tong University,2011

[13] WU X H.Performance analysis of improved quick search algorithms for pattern matching. Computer Engineering and Applications, 2014, 50（2）：44-48.

[14] Zeng C H. Duan Z H.An Enhanced Quick Search Algorithm for String Matching[J].Computer & Digital Engineering,2010，38（7）：48-49.