International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 06, June 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

AN IMPROVED AND OPTIMIZED PATTERN MATCHING ALGORITHM BASED ON INTRUSION DETECTION SYSTEM

Renuka S. Matkar[1], Yogeshwar. M. Kurwade[2], Dr. Vilas M. Thakare[3]

SGBAU, Amravati, Maharashtra (India)

ABSTRACT

With the rapid development of computer technology, nowadays Computer security has already become a heat topic. Intrusion detection system, as the second line of defense after the firewall in computer security system, can well improve the Computer security performance. Intrusion Detection System plays a reasonable supplementary role for the firewall in the network security. It can help protect computers from network attacks and improve the security and reliability of the computer. The pattern matching algorithm is one of the core algorithms used in the intrusion detection system. The increasing complexity of network attacks has lead the community to employ more expressive representations, which require the full power of an improved and optimized algorithm to solve the problem. In this paper, an optimized algorithm is proposed through analyzing the advantages and disadvantages of the main pattern matching algorithm of current Intrusion Detection System. And it also proves that the optimized algorithm has better matching efficiency than the original algorithm through simulation experiments. This paper analyzes the pattern matching algorithm which is used widely in intrusion detection technology, and proposes the improved pattern matching algorithm which increases the skip distance of the text string pointer, and reduces the number of comparisons.

Index Terms—AC algorithm, BM algorithm, Intrusion detection; pattern matching, String matching.

I. INTRODUCTION

With the rapid development of computer technology, Nowadays Computer security has already become a heat topic. Intrusion detection system, as the second line of defense after the firewall in computer security system, can well improve the Computer security performance. Intrusion Detection System plays a reasonable supplementary role for the firewall in the network security. It can help protect computers from network attacks and improve the security and reliability of the computer. At present intrusion detection system analysis module uses the pattern matching technology [1]. The pattern matching algorithm is one of the core algorithms used in the intrusion detection system. The increasing complexity of

network attacks has lead the community to employ more expressive representations, which require the full power of an improved and optimized algorithm to solve the problem[2]. Network security applications, e.g. network-based intrusion detection systems (NIDS) and firewalls, perform deep packet inspection to identify malicious traffic. In deep packet inspection, the contents of network traffic are matched against patterns of

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

malicious traffic to identify attack-carrying packets. In the past, patterns were represented by keywords that could be efficiently matched using string matching algorithms, e. g. KMP, Boyer-Moore, Wu-Manber, and Aho-Corasick. The increasing complexity of network attacks has lead the community to employ more expressive representations, which require the full power of highly efficient and optimized pattern matching algorithm [3].

Intrusion Detection and Prevention Systems (IDPSs) are used to detect malicious activities of intruders and also prevent from the same. These systems use signatures of known attacks to detect them. Signatures are identified through pattern matching algorithm which is the heart of IDPSs. Due to technological advancements, network speed is increasing day by day, so pattern matching algorithm to be used in IDPS should be fast enough so as to match the network speed. Therefore choice of pattern matching algorithm is the critical to the performance of IDS and IPS. Several pattern matching algorithms exist in literature, but which pattern matching algorithm will give best performance for IDPS is not known at hand [4]. Pattern matching plays an important role in many computer related fields, such as information retrieval, intrusion detection, data compression, content filtering, gene sequence comparison and computer virus signature matching. It is a basic problem in computer science. Pattern matching is one of the major issues in the area of computational biology. Pattern matching will continue to grow and need changes from time to time. The pattern matching algorithm is the core algorithm of the entire anti-virus software. Combining the advantages of fast calculation of hash function and parallel pattern matching of automata, it has significant performance advantages in the circumstance of virus signature matching [5].

In this paper, an optimized algorithm is proposed through analyzing the advantages and disadvantages of the main pattern matching algorithm of current Intrusion Detection System. And it also proves that the optimized algorithm has better matching efficiency than the original algorithm. So the performance of the system can be improved if this algorithm is applied to the intrusion detection system. In order to further ensure the security of computers, intrusion detection system is required to work in high efficiency. This paper analyzes the pattern matching algorithm which is used widely in intrusion detection technology and proposes the improved pattern matching algorithm which increases the skip distance of the text string pointer, and reduces the number of comparisons.

II. BACKGROUND

Along with the rapid development of computer technology, people's lives are increasingly dependent on computers. The rapid development of the Internet increases the freedom of application. But at the same time, because of its inherent openness, universality and freedom, it requires a higher demand of information security. Pattern matching plays an important role in many computer related fields, such as information retrieval, intrusion detection, data compression, content filtering, gene sequence comparison and computer virus signature matching [1].

Nowadays Computer security has already become a heat topic. Intrusion detection system, as the second line of defense after the firewall in computer security system, can well improve the Computer security performance. Intrusion Detection System plays a reasonable supplementary role for the firewall in the network security. It can help protect computers from network attacks and improve the security and reliability of the computer. At present intrusion detection system analysis module uses the pattern matching technology [2].

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

The pattern matching algorithm is one of the core algorithms used in the intrusion detection system. The increasing complexity of network attacks has lead the community to employ more expressive representations, which require the full power of an improved and optimized algorithm to solve the problem [3].

Single-pattern matching appeared first, and there are some classic algorithm such as the Knuth-Morris-Pratt (KMP) algorithm and the Boyer-Moore (BM) algorithm, which offer some lessons and inspirations for the development of later multi-pattern matching algorithm. Aho-Corasick algorithm, a variant of the Knuth-Morris-Pratt algorithm, was the first algorithm to solve the multiple string pattern matching problems in linear time based on automata approach. Boyer-Moore algorithm and a suffix automaton to search for the occurrence of multiple patterns in an input string. Wu-Manber algorithm is a simple variant of the Boyer-Moore algorithm that uses the bad character shift for multiple pattern matching. The working of existing algorithms is discussed in [4].

Many scientists have put forward the improvement of BM algorithm. The advancements in BM algorithm with new approach are presented in [5]. Intrusion Detection System plays a reasonable supplementary role for the firewall in the network security. It can help protect computers from network attacks and improve the security and reliability of the computer. In order to further ensure the security of computers, intrusion detection system is required to work in high efficiency. So, to improve the performance of system, there is a need to develop more optimized algorithms.

This paper is organized as follows, Section I gives the brief introduction about the pattern matching algorithms and it's applications along with proposed algorithm. Section II gives basics about the working and hierarchy of various pattern matching algorithms. So the performance of the system can be improved if optimized algorithm is applied to the intrusion detection system. Section III states the previous work done. Section IV gives an introduction to existing algorithms. Section V explains the proposed methodology. Then the results are shown on the basis of proposed methodology. Finally, the conclusion is drawn.

III. PREVIOUS WORK DONE

Reverse Colussi *et. al.* (2009) [1] proposed RC algorithm which states that comparisons of different pattern matching algorithms are done in specific order given by the preprocessed phase. The time complexity of preprocessing phase is O(m2) and searching phase is O(n). Crochemore *et. al*. (2008) [2] and Rytter *et. al*. (2008) [1] proposed a Two Way algorithm (TW). The Two Way algorithm uses an idea related to the short maximal suffix of the pattern to calculate the shifting lengths of pattern in text string. The Two Way algorithm's time complexity with the short maximal suffix is O(n).

S.Nirmala Devi et al. (2012) [2] and Dr.S.P Rajagopalan et al. (2012) [4] proposed the Index based Pattern Matching using Multithreading method performs pre-processing to get the index of the first character of the pattern in the given text. By using this index as the starting point of matching, it compares the Text contents from the defined point with the pattern contents. Raju Bhukya et al. (2012) [2] and DVLN Somayajulu et al. (2011) [2] proposed IBSPC. In IBSPC indexes has been used for the DNA sequence. After creating the index the algorithm will search for the pattern in the string using the index of least occurring character in the string.

Corasick M.J et.al (2008) [3] proposed many pattern matching algorithm with high efficiency to solve this problem, which is called for short AC algorithm. In the preprocess stage, AC algorithm form several pattern

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

strings waiting for matching, according to their features into Tree finite state automata, and decide the next situation according to matching characters.

Brute force et.al (1994) [4] proposed brute force or Naive algorithm which is the logical place to begin the review of exact string matching algorithms. It compares a given pattern with all substrings of the given text in any case of a complete match or a mismatch. It has no preprocessing phase and did not require extra space. The time complexity of the searching phase of brute force algorithm is o(mn).

Boyer-Moore et.al. (1977) [5] proposed Boyer-Moore (BM) algorithm published in 1977. It performed character comparisons in reverse order from right to the left of the pattern and did not require the whole pattern to be searched in case of a mismatch. The time and space complexity of preprocessing phase is O(m+|∑|) and the worst case running time of searching phase is O(nm + |∑|). The best case of Boyer-Moore algorithm is O(n/m).

Boyer-Moore Horspool *et.al* (1990) [5] proposed BMH which did not use the shifting heuristics as Boyer-Moore algorithm used. It used only the occurrence heuristic to maximize the length of the shifts for text characters corresponding to right most character of the pattern. It's preprocessing time complexity is O(m+|∑|) and searching time complexity is O(mn).

IV. EXISTING METHODOLOGY

A. BM algorithm:

These are some principles to realize the BM algorithm: 1) at the beginning of matching, align pattern strings P and text T from left to right, but the matching operation starts from right to left.

2) If the character and position in P matches with the character in text T, T and P will move a position toward left at the same time and then make comparison.

3) If the matching fails, two offset functions Badchar and Goodsuffix in preprocessing will work out the distance in which pattern string P moves toward right, and align T and P again to match.

Here are the specific definitions of functions Badchar and Goodsuffix [1].

a) Use Badchar to move: Work out the deviant of every character in the T character collection. If character C in T appears many times in P, the last position can be used to work out the deviant, the mathematical formula is as follows:

$$\text{Badchar(c)} = \begin{cases} m, any\ c\ not\ in\ P_m\ and\ c \neq P_j (1 \leq j \leq m-1) \\ m - j, here\ j = MAX\{j | c = pj, 1 \leq j \leq m-1\}, \\ c \in P \end{cases}$$

b) Use Goodsuffix to move: When the matching fails, substring U in the pattern strings has already been matched with one substring in T, if there is still a character string U in P, pattern strings will be moved some distance to make next substring U match, or leap over the whole distance [1].

B. BMHS algorithm

In 1990, Sunday put forward BMHS algorithm on the basis of BMH algorithm. The main Specific ideas of BMHS algorithm is:

1. First align pattern strings and text T and then compare them from right to left. But when the matching fails, Use last character pm in the pattern strings and $t(k+1)$ which is next to the character $t(k)$ in T text which is corresponding to the pm to work out the left moved distance.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

2. If the character $t(k+1)$ does not exist in the pattern strings, move toward $m+1$, the right moving distance is more than BMH algorithm.

3. If $t(k)$ does not exist in the pattern strings, but $t(k+1)$ exists in the pattern strings, BMH algorithm right shift distance is more than BMHS algorithm distance. In many cases, BMHS algorithm has higher matching efficiency than BMH algorithm [2].

C. AC Algorithm:

When many pattern strings need to match, Single pattern matching is not efficient. Aho A.V and Corasick and M.J proposed multiple pattern matching algorithms with high efficiency to solve this problem, which is called as short AC algorithm. In the preprocess stage, AC algorithm form several pattern strings waiting for matching, according to their features into Tree finite state automata, and decide the next situation according to matching characters. Tree finite state automata states that prefix of every pattern strings can be signed by only one situation, even though several pattern strings have the same prefix. The matching process starts from the root of the tree. If the scanning shows that the character is not the next character of pattern strings, it turns to another situation standard for the longest prefix, and this situation is the suffix of current situation. When the matching process get the leaf node of the pattern tree it shows that there exists the character string in text t matching with pattern string. AC algorithm is more efficient than single pattern matching when matching multiple pattern strings. However, when matching with text strings, AC algorithm scans one by one, not leap. Therefore, in the case of less pattern strings, it has little performance [3].

The good suffix rule is used as a formula in AC algorithm.

1. $shift(j) = min\ (s|(p(j+1 \dots m) = p(j-s+1 \dots m-s)))$

Where, assuming shift j is the distance which P skips to right, m is the length of pattern string P, j is the current matched character position, s is the distance between and t or the distance between X and p".

D. AC-BM algorithm

In 1993, on the basis of AC algorithm, Jang Jong used the leap idea of BM algorithm and proposed AC-BM algorithm. The principle of this algorithm is that it has two stages: 1) preprocess phase, 2) searching/matching phase. In the matching process, align the right of pattern strings with shortest length of character of pattern tree with the right of target string, and then match from right to left of pattern tree. The matching in the process, align the right of pattern strings with shortest length of character of pattern tree with the right of target string, then match from right to left in pattern tree. The matching in the pattern tree starts from the root node of the tree to leaf node. There exists matching character strings at leaf node. This matching character strings go through from root node of the tree to leaf node. In the matching process, when corresponding character doesn't match, the matching fails. Pattern tree needs to move left, and the function Bad char and Good Suffix decide the distance of movement [4]. The mathematical model for calculating shift distance used in the algorithm is as shown below:

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

$skip(p(i)) = pLen - i + 1$

E. AC-BML algorithm

There are many multi pattern matching algorithms in existence; the most widely used is AC-BM algorithm. The existing AC-BM algorithm is invented for short. There is an improved AC-BM algorithm for long pattern and string. This algorithm includes preprocess stage and matching stage. First, form several pattern strings waiting for matching, according to their features into Tree finite state automata in the structuring, if degree of one node is greater than 1, this node is marked "1", and its child node is marked "1", too. Other nodes are marked "0". Then apply BMHSL algorithm to the same prefix of pattern strings (from root node of pattern tree to the part whose degree is not "1"), find out all the locations *wi* where character string in the text can match with prefix and record them [5]. It uses the Bad Char shift rule as:

Badchar(c)

$$= \begin{cases} minlength(p), (if\ c\ is\ not\ in\ any\ pattern\ strings, \\ \quad the\ distance\ of\ movement\ is\ the\ length\ of \\ \quad\quad shortest\ pattern\ strings) \\ m, means\ the\ matched\ charcter\ in\ other\ pattern \\ \quad\quad strings\ and\ compared\ character \end{cases}$$

Where, minlength(p) is the minimum distance in the pattern.

V. ANALYSIS AND DISCUSSION

BM algorithm takes the max value between skip(X) and skift j as the skip distance. The time complexity of BM algorithm pretreatment is O(ms) , space complexity is O(s), s is the length of the limited character set related to P and T. The time complexity of search phase is O(m) . In the best case time complexity is O(m), in the worst-case time complexity is O(mn) [1].

BMHS algorithm is an improved and simplified algorithm from BM. It only considered "bad character" strategy when moving pattern string. It will first compare the character which text string pointer pointed and the last character of pattern, if they are equal then compare the remaining *m*-1 characters. It has space complexity of O(s/2) [2].

No matter which character caused the text to mismatch, there will be a character which alignment with the last position of the pattern string in the text, to determine the pattern string's shift to the right. BMHSL algorithm has time complexity of O(mn/2) [3].

In AC algorithm, it shows that when the amount of pattern string is less, Single model algorithm is the first choice. But as the amount of pattern strings increase, the time which many pattern needs add slowly while time of Single model algorithm changes quickly [4].

AC-BML algorithm is better than its original algorithm. The object of experiment is the same; pattern string is the same, under such a situation, check out that how much time CPU takes in different algorithm. By comparison, AC - BML algorithm is improved on the basis of AC - BM algorithm. It is because an AC-BML takes less number of comparisons [5].

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Methodology/ Algorithm	Advantages	Disadvantages
1. BM algorithm	1. Most searches takes O(m+n) So it is faster. 2. Quick searching	1. Runs in time O(mn) in the worst case. 2. Suitable for smaller strings only
2. BMHS algorithm	) 1. Efficient than the existing algorithms. 2. Significantly faster than BM algorithm.	1. Need to improve shift decisions of an algorithm.
3. AC algorithm	1. Provides faster pattern matching than 2. Use of large windows makes it more useful.	1. Slower when alphabet size is small. 2. Takes more number of comparisons
4. AC-BM algorithm	1. Due to character jump technique, pattern matching is improved.	1. Time complexity is not so improved. 2. More memory consumption
5. AC-BML algorithm	1. As three cases are used, time and space complexity is drastically improved	1. More number of comparisons. 2. Use of large dataset makes it complex.

VI. PROPOSED METHODOLOGY

An optimized algorithm is proposed through analyzing the advantages and disadvantages of the main pattern matching algorithm of current Intrusion Detection System. And it also proves that the optimized algorithm has better matching efficiency than the original algorithm through simulation experiments. This paper analyzes the pattern matching algorithm which is used widely in intrusion detection technology, and proposes the improved pattern matching algorithm which increases the skip distance of the text string pointer, and reduces the number of comparisons. The proposed methodology focuses on an improved and optimized algorithm for pattern matching.

The proposed methodology gives an efficient improved and optimized algorithm for pattern matching which enhances the efficiency and reduces the number of comparisons. It also proves that the optimized algorithm has better matching efficiency than the original algorithm. So the performance of the system can be improved if this algorithm is applied to the intrusion detection system. In order to further ensure the security of computers, intrusion detection system is required to work in high efficiency. This paper analyzes the pattern matching algorithm which is used widely in intrusion detection technology and proposes the improved pattern matching algorithm which increases the skip distance of the text string pointer, and reduces the number of comparisons.

As BMH algorithm abandon good suffix completely, there always exists the case that a part of pattern string match with the main string while the prefix does not match in a real intrusion detection environment, so matching efficiency of bad character is lower than that of good suffix. According to the study of BM algorithm

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

and BMH algorithm, propose an improved algorithm. This algorithm first introduces an array *F* to label the frequency that every character appears in pattern string, then matching with the order of frequency, to increase mismatching probability, decrease match time and match numbers effectively.

Table 2: An improved and optimized pattern matching algorithm

IMPROVED and OPTIMIZED PATTERN MATCHING ALGORITHM
Step 1: Initialize **Step 2: Enter an input** **Step 3: Frequency calculation when character appears in pattern string** a) Statistics the frequency when every character appears in pattern string. b) Order each character in accordance with the occurrence frequency of small to large. The smaller the frequency, the more priority to match. c) The ending character takes precedence to match. **Step 4: Improve the text string pointer's skip distance** a) Compare *t*[*m*] with *P*[*m*] from right to left, and execute step 3 according to comparison. b) If the match is successful, comparing other correspond character continually accord with the frequency priority, and check if the frequency of that character is 1. c) If it is 1, save the pointer that first matched with a variable flag until mismatched. Step 5: If the matching is failure a) combine the character which is align with the last character of pattern string and its next character in text string to character M b) Check if it appears in pattern P. c) If it does, then move the text string pointer, to make the last right character M in pattern p align with character M in pattern T. Step 6: Checking Probability of character in a pattern a) If it does not appear, look that whether the second character of that combination appears in pattern *P*. b) If it does, the text string pointer's skip distance is M, else is m+1. Step 7: Finish

VIII. POSSIBLE OUTCOMES AND RESULT

An improved and optimized algorithm has better performance than the BM algorithm and BMH algorithm. On the times of character comparisons E-BM algorithm is less than the BM algorithm and BMH algorithm, reduced by far greater percentage; E-BM algorithm is far improved than existing algorithms than the BM algorithm and BMH algorithm in time performance. The analysis shows that, proposed algorithm can significantly reduce the character matching times and pattern strings' moving times; it can also improve the matching speed without

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

increasing the space complexity. E-BM algorithm's performance is higher than the BM algorithm and BMH algorithm, therefore, with great practical value.

IX. CONCLUSION

This paper presents new improved and optimized algorithm for exact pattern matching algorithms. The current intrusion attack is so diversified and complicated. In order to further ensure the security of computers, intrusions detection system is required to work in high efficiency. This paper analyzes the pattern matching algorithm which is used widely in intrusion detection technology, and proposes the improved and optimized multi pattern matching algorithm. This algorithm is superior to original algorithms by excrement. This will improve the System testing performance when applied into intrusion detection system.

X. FUTURE SCOPE

There is still need to improve the matching process for far more efficiency and to handle large amount of data. The future work is to incorporate and integrate various exact pattern matching techniques for better accuracy by further reducing the number of comparisons hence improving the time complexity.

REFERENCES

[1] Vibha Gupta, Maninder Singh, Vinod K. Bhalla, "Pattern Matching Algorithms for Intrusion Detection and Prevention System: A Comparative Analysis", 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 50-54, May 2014

[2] S. Patil, P. Rane, Dr. B. B. Meshram, "IDS vs IPS", International Journal of Computer Networks and Wireless Communications, Vol. 2, No. 1, pp. 232-235, September 2010

[3] J. Aldwairi, M. Monther, and D. Alansari. "Exscind: Fast pattern matching for intrusion detection using exclusion and inclusion filters." Next Generation Web Services Practices (NWeSP), 7th International Conference on Computer science technology, IEEE, pp. 45-48, May2009

[4] Zhongming Yang , Shen Lin , "A Single-pattern Matching Algorithm Based on Twice Jumps in the Intrusion Detection Systems", 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 2593-2596, May 2015

[5] LIU Pei-qian,FENG Jing-jing. "Improved BM Pattern Matching Algorithm", IEEE Transaction on security issues in computer science and Engineering, Vol. 20, issue no. 1, pp. 48-51