Abstract- In network security, Intrusion Detection System plays a reasonable supplementary role for the firewall. It improves the security and reliability of the computer and helps protect computers from network attacks. At present intrusion detection system analysis module uses the pattern matching technology. In this article, through analyzing the advantages and disadvantages of the main pattern matching algorithm of current Intrusion Detection System an optimized algorithm was proposed. And through simulation experiments, it also proved that the optimized algorithm has better matching efficiency than the original algorithm. So if this algorithm is applied to the intrusion detection the performance of the system can be improved.

Keywords— Intrusion Detection System (IDS), Pattern Matching Algorithm.

I. INTRODUCTION

Now days the network has expanded to very large scales and the data exchange between the networks and the amount of the network traffic has increased tremendously. On the other hand, very large numbers of computers around the world stay connected to Internet all day long. These computers store very important data such as emails, documents, pictures and videos. Some of these computer may install uncertified programs like Trojan horse, that send sensitive information belong to users such as user name, password and credit card number. The computers have become a favorite target for complex of attacks that requiring more complex analysis to detect it, this is because of the growth of the Internet users and the user data. The Intrusion Detection System (IDS) becomes an important part of any modern network for guaranteeing the security issue of information systems [1, 2].

To detect intruders: anomaly detection, misuse detection (signature detection) there are two basic techniques used. To uncover abnormal patterns of behavior Anomaly Detection is designed, the IDS establish a baseline of normal usage patterns, and anything that widely deviates from it is flagged as a possible intrusion [3]. Misuse Detection, are also called signature detection, uses specifically known patterns of unauthorized behavior to detect and predict subsequent similar attempts. These specific patterns are called signatures. Therefore in case of Misuse Detection at the heart of IDS is the attack signature. Through approaches like Network Grapping / Pattern Matching, Protocol Decode/Analysis, Heuristic and Honey pot the signatures can be generated. Misuse detectors often work in Current intrusion detection systems, where the packets in the monitored network are compared against a repository of signatures that define characteristics of an intrusion. Successful matching causes alerts to be fired. In a given network packet the signature often consists of one or more specific binary patterns. The signature can be described as a Boolean relation called rule [4].

To recognize an attack an intrusion detection system must knows a signature for this attack, and thus require continuous updates of their signature database. Also continuous research to analyze new attacks. The rest of this paper is organized as follows. In section II influence on ids are reviewed and in section III some single pattern algorithm are discuss and in section IV multi pattern algorithm are discuss. In other two section related works are reviewed and concludes this paper.

II. IDS INFLUENCE FACTOR

Many factors can affect on the IDS performance, High speed of modern network is important factor of IDS, because the detection engine is searching and comparing the packets against signatures rule policies. But due to the large amount of incoming packets the IDS will lose control of all these packets and consequently, will drop a number of packets. As result, these drooped packets may contain some malicious pattern that can pass successfully [7]. And another important factor string matching algorithm. This is because most of the string matching algorithms are not designed directly to work with IDS; they are designed to work with other applications such searching on dictionaries [6]. One another factor, the rule policies are growing rapidly. Snort rule policies are increasing and reaching more than 8000 rules [13]. All factors are affected on IDS performance and increased the processing time.

III. SINGLE PATTERN MATCHING

A. The most basic method of approaching the problem of pattern matching is the Brute Force (BF) algorithm. This technique is very simple and easy to follow. Let’s assume we have text (input) T with length n and a pattern (keyword) P with size m. The algorithm begins by comparing the pattern to the text, scanning left to right, and one character at a time until there are no more matching characters. If a mismatch occurs, the algorithms shift the pattern one character to the right. The brute force algorithm requires no preprocessing of the pattern .also it does not require any extra space in addition to the memory required to store the pattern and the text.
B. BOORE-MOORE ALGORITHM (BM)

Boyer-Moore algorithm is the most efficient string matching algorithm in usual application [23]. BM algorithm is the single pattern based algorithm. The algorithm scans the characters of the pattern from right to left beginning with the rightmost one. In case of a mismatch (for a complete match of the whole pattern) the algorithm uses two tables or functions, which is used to move the sliding window to the right. These two shift functions are called the good suffix shift (also called matching shift) and the bad character shift (also called the occurrence shift). On small pattern size the algorithm is extremely fast but it is slower when it is working with large pattern size.

C. BOORE-MOORE HORSPOOL (BMH)

The modified version of BM algorithm with a little different change is BMH. Horspool algorithm is considered to be one of string Matching algorithm. Unlike, BM algorithm the BMH algorithm uses only bad character shift whereas, the Boyer-Moore algorithm uses two tables: bad character shift and good suffix shift. Boyer-Moore Horspool Algorithm searches for the pattern from left to right and the shift value in it is depending on the pattern size determined in the bad character shift table. In most of the situations BMH is faster in its performance because of the longer shift value [21]. BMH Algorithm has best average case performance. It is simple to implement and has a low space complexity. Therefore, if the pattern size is small or large it can be implemented in any cases that need the exact string matching algorithm [9, 10]. It still needed little improvement and enhancements in its characters comparison speed. We choose it through all others algorithms to enhance its performance because of its advantages.

IV. MULTI PATTERN MATCHING ALGORITHM

Single pattern matching algorithm used in intrusion detection system is efficient when there are not so many patterns to process. However, if there are a large number of patterns to process, it will be less efficient [17].

A. AC ALGORITHM

In order to solve that problem, Aho A.V and Corasick M.J proposed Many pattern matching algorithm with high efficiency to solve this problem, which is called for short AC algorithm [18]. Experimental results show that the algorithm is faster and the detection performance has been improved obviously than some previous algorithms when searching a large number of patterns [24]. AC is a multi-pattern string matching algorithm to organize multiple pattern strings in order to get searching of multiple patterns done in one pass that adopts a finite state automaton. From the root of the tree the matching process starts. It turns to another situation standard for the longest prefix, if the scanning show the character is not the next character of pattern strings, and this situation is the suffix of current situation [16]. It shows that there exists the character String in text t matching with pattern strings when the matching process meets the leaf node of the pattern tree. Regardless of the number of patterns its time complexity is O (n), [19]. However, it does not excavate any heuristics to avoid unnecessary character comparisons; AC suffers an intrinsic deficiency for high speed.

B. AHO-CORASICK_BOORE-MOORE ALGORITHM (AC_BM)

The Aho-Corasick_Boyer-Moore (AC_BM) [8] algorithm is an efficient string pattern matching algorithm that is derived from the Boyer-Moore algorithm to improve efficiency when there are multiple string patterns to search against. Multiple string pattern matching is one of the core problems in the rule-based Intrusion Detection System (IDS) such as Snort [11]. The AC_BM algorithm can be used to improve the efficiency of the detection engine of the IDS. The basic idea of the algorithm is that the string patterns are first built into a pattern tree, and then the algorithm slides the pattern tree using bad character and good prefix shifts to find the matches.

C. AC-BM-GA ALGORITHM

On the basis of AC-BM algorithm, and used the leap idea of Genetic algorithm we proposed AC-BM-GA algorithm. The principle of this algorithm is that: first, at the preprocess stage, according to the idea of AC algorithm. In the matching Process, we follow the simple steps of genetic algorithm. [25] Starting by a random generation of initial population, then evaluate and evolve through selection, recombination, and mutation. Finally, the best individual (pattern) is picked out as the final result once the optimization meets its target. The figure shows the result of comparison of AC-BM and AC-BM-GA algorithm in figure x-axis shows the detection rate and y-axis shows amount of string taken by user. Proposed algorithm has the high detection rate as compare to exiting algorithm.

V. RELATED WORK

In 2002 M. Fisk and G. Varghese proposed the Setwise Boyer-Moore-Horspool (Setwise BMH) algorithm [14], which constructs a trie that stores all patterns based on the
suffixes, and creates a leap table based on the bad character heuristic. To improve the matching efficiency the compact tries will get rid of many unnecessary comparisons. A leap will be calculated by selecting the minimum leap among those generated by all patterns using the bad character criterion when mismatch occurs. In 2002 C.J.Coit, S. Staniford, and J. McAlerney proposed the AC_BM algorithm [15], which is very similar to the Set wise BM algorithm. BMH constructs a tree of all patterns based on the prefixes instead of suffixes makes it differ from Set wise BMH. Also, AC_BM takes both bad character and good suffix heuristics to calculate the leap when there is a mismatch.

The focus of this paper is on multi-pattern string matching algorithms for NIMS. Excluding the single-pattern string matching algorithm BM, the other four algorithms can be classified into three categories. Automaton based AC, trie based Set wise BMH and AC_BM, and table based WM. table structure is to be more efficient in space, since they do not need to store every pattern in the data structure generally.[12] However, Because the three categories all benefit from their own advantages it is hard to judge which category would be better in searching time: At each node automaton based algorithm do not need to do naïve comparisons as they store the matching result; trie based algorithms narrow down the comparison scope by leveraging their Compact data structure; while through direct table lookups table based algorithms provide very fast search speed even though they have to perform naïve comparisons when there is a potential match [22].

VI. CONCLUSION

In order to further ensure the security of computers, intrusions detection system is required to work in high efficiency. The current intrusion attack means is so diversified and complicated. This paper analyzes the Single pattern and many patterns matching algorithm which is used widely in intrusion detection technology, and proposes to improved algorithm. In this paper we enhanced the existing multipattern algorithm to avoiding unnecessary comparisons with the help of Genetic algorithm. We use kddcup99 dataset for train and test the performance. This reduces the detection time and improves the System testing performance when applied into intrusion detection system [20].

REFERENCES


[18] S. Antonatos1, M. Polychronakis1, P. Akritidis1, K.G. Anagnostakis2 and E.P.markatos ipiranga: fast and memory-efficient pattern matching for intrusion detection.

[20] Ramakrishnan Kandhan Nikhil Teletia Jignesh M. Patel
Computer Sciences Department, University of Wisconsin–
MadisonSigMatch: Fast and Scalable MultiPattern
Matching.

[21] Gao Chao-Gin, Chen Yuan-Yan, li Detection of rapid
Multi-pattern matching A facing the intrusion algorithm
[J].journal of computer application, 2008,: 82-85.In
Chinese.

[22] Cheng Yu-qing, Mei Deng-Hua. The improved of Intrusion
Detection System BM pattern matching algorithm [J].
Computer Technology and Development, 2009 19 (3):172-

[23] Li Shu-zheng. The research of fast pattern matching
algorithm Based on Snort system [D]. Master Thesis, Jilin
University, 2009,4 In Chinese.

matching Algorithm [J]. University of Electronic Science

[25] Haidong Fu, Fan Hu.Intrusion Detection Algorithm of
Artificial Immune Based on Decision Tree and Genetic
Algorithm.