International Journal of Electrical, Electronics and Computer Engineering **1**(1): 88-90(2012)

ISSN No. (Online) : 2277-2626

A Review of Intrusion Detection Method Based On Automatic Pattern Matching

Garima Shrivastava and Anurag Jain

Department of Computer Science and Engineering,

Radharaman Inst of Technology and Science, Bhopal (M.P.)

(Recieved 20 March 2012 Accepted 10 April 2012)

ABSTRACT : 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. In this article an optimized algorithm was proposed through analyzing the advantages and disadvantages of the main pattern matching algorithm of current Intrusion Detection System. And it also proved that the optimized algorithm has better matching efficiency than the original algorithm through simulation experiments. So the performance of the system can be Improved if this algorithm is applied to the intrusion detection.

Keywords: Intrusion Detection, based on pattern matching.

I. INTRODUCTION

A great number of factors contribute to grant computer system access to aggressors, eventually resulting on further private information stealing or other risks. Some type of security disruption has already occurred in the majority of computers, causing external aggressors (or still legitimate users) to obtain some sort of unauthorized access. Even a supposed safety system may be vulnerable to legitimate users abusing their privileges, or being compromised by improper practices. Inside this scenario, once the attack is considered inevitable, must be used some mechanisms to detect and protect the system against aggressors trying to penetrate or disrupt it, or even legitimate users making bad use of their privileges. Nowadays, one of the most used security mechanisms is the firewall system, responsible for protecting threats inside from outside network. But a firewall is not a complete tool, due to it is not able to work on the internal side of the network. Neither it is able to stop an intrusion attempt or privileged misuse, for which it has not been previously programmed .So, a suitable approach is to use a system able to protect the internal network with an adaptive behavior, which can learn new patterns of actions to identify an attacker. It is also desirable that this tool, once with the capacity to detect malicious actions, takes some prevention or counter–measures decisions about the situation, based on the knowledge given to it. The rest of this paper is organized as follows. In section II and III some related works are reviewed. Section IV concludes this paper

II. PATTERN MATCHING ALGORITHM

Pattern matching, that is character string matching, to find a certain character string in the target character string . If pattern string appears in target string once or several times, such case is called successful match, or called unsuccessful match. There are Single pattern matching and multipattern matching [8].

A. BM algorithm

The principles to realize the BM algorithm: 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. 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, [1]. If the matching fails, 2 Offset functions Badchar and goodsuffix in preprocessing will work out the distance which pattern strings P moves toward right, and align T and P again to match.

B. BMHS algorithm

The BMHS algorithm is: 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. 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. If $T[k]$ does not exist in the pattern strings, but $T[k+1]$ exists in the pattern strings. [7].

C. Improved single pattern matching algorithm

This essay put out the improved algorithm on the premise of BMHS algorithm. This improved algorithm uses the method that match and move somewhere the characters appear least, [2]. This improved algorithm is called BMHSL algorithm in the following. The main idea of this algorithm is: the process of matching includes preprocess stage and matching stage. Preprocess stage is to analyze the features of pattern strings and T text before starting to match, which

make primary preparation for matching [4]. On the basis of BMHS algorithm, matching stage is to combine the result of preprocess with matching Around alternately, then to match.

(1) Preprocess stage: first, scan the pattern string P, work out character set Σ_1 in *P*; then scan *T* text, and work out the character set Σ_2, and the frequency t of every character belonged to ?1 and existing in Σ_2 (that is , the frequency t in P of every character in $(\Sigma_1 \cap \Sigma_2)$, and then find out character K which appears least and record the locate [*i*] of *K* appearing in text *T* and the distance *d*[*i*] between two *K*. if Σ_1 doesn't belong to the subset of Σ_2, that is some characters in Σ_1 don't exist in Σ_2, pattern strings P will not find the matched characters strings in *T* text, which means the matching fails.

(2) Matching process: move the whole pattern string *P* directly to make the characters in pattern strings *P* align with character *K* in text *T*. matching starts from pm, the end of pattern strings *p*, if pm can match with the corresponding characters in text *t*, pm moves to *p*1, compare that whether *p*1 matches with the corresponding characters in text *t*. pm–1, *p*2, pm–2, *p*3 do as the same way. No matter where the matching fails, it is necessary to move pattern strings *p* towards right. [5] After right movement, use the above way to match until find out the character strings matching with pattern strings. If at the end of the text *t* there is no character strings matching with *p*, matching finally fail.

III. SOME MORE PATTERN MATCHING ALGORITHM

A. AC algorithm

Aho A.V and Corasick M.J proposed many pattern matching algorithm with high efficiency, which is called for 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. The matching process starts from the root of the tree. AC algorithm is more efficient than Single pattern matching when matching many pattern strings. [3]. However, when matching with text strings, AC algorithm scans one by one, not leap. Therefore, in the case of less pattern strings, its functions show little.

B. AC–BM algorithm

The principle of this algorithm is that : first, at the preprocess stage, according to the idea of AC algorithm. In the matching 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 of 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 [10]. 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 Badcha and Goodsuffix decide the distance of movement.

C. Improved AC – BM algorithm

AC–BM algorithm combing BMHSL algorithm, this paper proposes many pattern matching algorithm are proposed which is called AC–BM algorithm for short [6]. And the specific idea: this matching process includes preprocess stage and matching stage. (1) preprocess 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.

(2) Matching process move the pattern tree from the right of text pattern to left, align the root node of pattern tree with the location of *w*1 in the text string directly at the beginning. The same prefix of pattern strings need not match, but the other parts. First, compare those characters in the text which corresponds with node marked 1 in the pattern tree [9]. After matching finishes nodes marked 1, possible pattern string can be got. Then compare nodes marked 0, in such a way, text string which matches the possible pattern strings can be found.

IV. CONCLUSIONS

The current intrusion attack means 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 Single pattern the improved algorithm and Many pattern the improved algorithm. These two algorithms are superior to original algorithms by excrement. These two algorithms must improve the System testing performance when applied into intrusion detection system.

REFERENCES

[1] Qin Hai–sheng, WEI Hai–lan, LI Xin–hua, LI Jun–hui Research and optimization of Pattern Matching Algorithm Based on Instrusion Detection System, 2011.

[2] Urjita Thakar , Nirmal Dagdee Intrusion Attack Pattern Analysis and Signature Extraction for web Services Using Honeypots. 2008.

[3] Ibrahiem M M,Mohammed SM.A novel algorithm for solving the string matching problem[J]. *International journal of computational Intelligence and applications*, 2006, **6**(4):499–510.

[4] S. Antonatos1, M. Polychronakis1, P. Akritidis1, K.G. Anagnostakis2 and E.P.markatos1piranha: fast and memory–efficient pattern matching for intrusion detection.

[5] Mike Fiskyx and George Varghesey xLos Alamos National Laboratory yUniversity of California San Diego Applying Fast String Matching to Intrusion Detection.

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

[7] Gao Chao–Gin, Chen Yuan–Yan, li mei . A facing the intrusion detection of rapid Multi–pattern matching algorithm [J]. *Journal of computer application*, 2008,**28** (1) : 82–85.In Chinese

[8] Cheng Yu–qing, Mei Deng–Hua.The improved of Intrusion Detection System BM pattern matching algorithm [J]. *Computer Technology and Development*, 2009 **19** (3) :172–174. (In Chinese).

[9] Li Shu–zheng. The reserch of fast pattern matching algorithm Based on Snort system [D]. Master Thesis, Jilin University, 2009.4. In Chinese

[10] Wan Guo–Gen, Zhi Guang. Improved AC–BM string matching algorithm [J]. University of Electronic Science and Technology, 2006.8. In Chinese.