DELIMITER SEARCH FOR LOCATING RULE PATTERN IN INTRUSION DETECTION SYSTEMS

Kruatrachue B., Threepak T., Hongsuwan T.

Department of Computer Engineering, Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang.Thailand

ABSTRACT. The goal of this research is to improve the run time performance of Network Intrusion Detection Systems (NIDS). The system performs intrusion detection by matching specific intrusion patterns with the incoming network packet data. Since the numbers of patterns is large (up to 2000) and keep increasing leading to huge search time. In this paper, we proposed a search time reduction by using delimiter search and implementing intrusion patterns into a dictionary of patterns with burst tries structure. The search time is compared between the snort Boyer-Moore algorithm and proposed method. The result of the proposed method on average take only 4.96 percent of time of original snort.

1. INTRODUCTION

Network Intrusion Detection System (NIDS) is the system which detect the intruder patterns from packet data capturing from network. The early NIDS is from the UC Davis Network Security Monitor [1, 2] using exact string matching. More recently, several NIDS tools have been developed and use broadly such as SNORT [3] and BRO [4]. Both systems still use exact string matching technique. The difficulties in implementing NIDS systems is to perform intrusion pattern matching in near-real-time in order to catch up with incoming network traffic. From these difficulties, the attacker can elude the NIDS by using insertion, evasion and Denial of Service techniques [5].

To overcome DOS, we try to improve intrusion pattern matching speed by finding delimiters in rules and in incoming data packet and use the delimiters and distance between delimiters for intruder patterns matching. Since the size of rules and packet is decrease, the search time is also decrease

In order to verify the speed improvement of the proposed method we modified snort intruder patterns structure and matching methods and make comparisons on various type of network traffic.

2. SNORT INTRUDER PATTERN STRUCTURE AND SEARCHING ALGORITHM

Snort is a network intrusion detection system. Its function is to capture data from network and checking captured data with intrusion patterns. All the intrusion patterns are groups into a two dimensional list of rules as show in Figure 1. If content of packet data match any rule, it will alert to system administrator. This help system administrator know the intruder as fast as possible and secure the system.

The two dimensional list consist of two node type, Rule Tree Node (RTN) and Option Tree Node (OTN) as shown in Figure 1. The RTN contain the common properties of the rule, such as the source address, destination address, source port, destination port and protocol type (TCP, ICMP, UDP) of packet. The OTN contain the information for various options that can be added to each rule, such as TCP flags, ICMP codes and types, packet content.

If a content check is required, Snort uses a Boyer-Moore pattern matching algorithm to check the content string held in the OTN against the entire packet. If no string match exists, Snort will proceed to the next OTN in the list. This is a very time consuming process even though the Boyer-Moore perform some smart skipping without a shift and check a rule for the whole length of packet. Since the number of rule is large and keep increasing.

3. SNORT RULES

Snort rules are the set of sentence that describes about what snort do if it found some data in data packet matching with the content of rule. The grammar of rule is including snort response, rules of header and rules of content.

alert tcp any any -> any 80 (msg:"IIS-cmd?";flags:PA; content:".cmd?&"; nocase;)

From the above rule, the response of snort is alert if packet has tcp protocol and send from any IP and any port to any ip address at port 80. It has PSH and ACK flag set and the content have exact word ".cmd?&" in case non-sensitive. Other example of snort rules contents are shown in Figure 2.

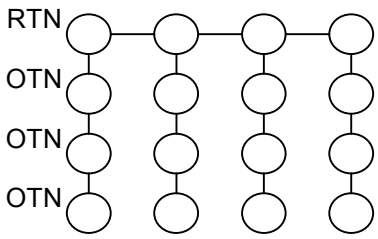

Fig. 1. Two-dimension link list of snort rule

cgi-win/wwwuploader.exe
_vti_bin/_vti_aut/author.exe
iissamples/sdk/asp/docs/codebrws.asp
scripts/iisadmin/ism.dll?http/serv
adsamples/config/site.csc

Fig. 2 . Examples of rule Content

4. DELIMITER TRANSFORM & BURST TRIES DICTIONARY

From the snort intrusion patterns identification method describe in previous section, this research proposed two improvements. Firstly, a dictionary of intrusion patterns implementing with burst tries structure as shown in Figure 3. The main idea is to check whether input data packet has any word (string) contain in the intrusion dictionary instead of checking each intrusion pattern in the data packet. This is a major speed up since only one pass to the whole length of packet all the intrusion matching is done. On the other hand, snort pass through the whole packet rule by rule so number of pass equal the number of rules. Although Boyer-Moore performs some smart skipping the matching is much slower.

Secondly, most incoming packet are not intruder, not contain the intruder patterns, so the search is optimize by searching of delimiter and distance instead of exact patern match. The original rule and incoming packet are transform into delemiter space as shown in Figure 4. The dictionary of delimiter rule are shown in Figure 5.

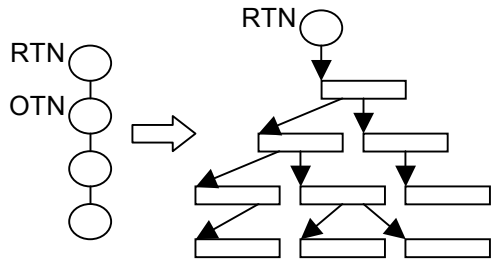

Fig. 3. from two-dimension link list to link of burst trie

adsamples/config/site.csc -> 9/6/4/3
admcgi/contents.htm -> 6/7.3
cgi-win/wwwuploader.exe -> 3-3/11.3

Fig. 4. transformation using delimiters : - / .

Since the pattern length in each rule is shorter as well as the incoming data packet. The search time should improve even though the whole packet need to have additional pass once to transform in to delimiter format before rule search. But this is done once for all the rule in the same RTN (all the rule in all RTN if we use the same delimiter for all RTN rules). Naturally, this assumption depend on the number of rule, rule length, packet length and the choice of delimiter. (more benefit toward high number of rule of long length packet and rules).

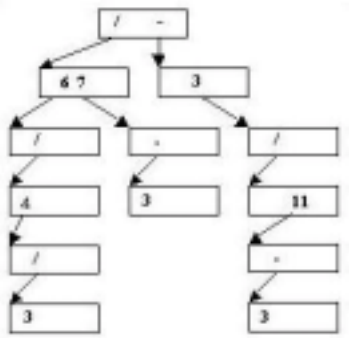

Fig. 5. Example of delimiter format rule in burst trie dictionary

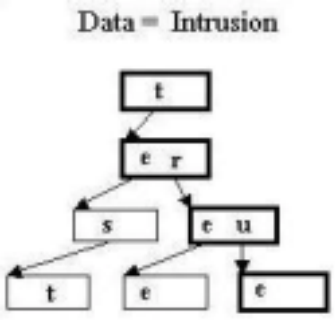

Figure 6. Counting Example

5. EXPERIMENTAL RESULT

In this experiment, the performance of three NIDS are compared. The first one is the original snort with Boyer-Moore search for each intrusion pattern. The second system is the snort with burst tries dictionary of intruder patterns. The third one is the second one with rule and packet in delimiter format.

The test data come from web transfer of various type text, binary, and picture. The comparisons are measure in number of comparison of all character in packet in each level of tries dictionary. These include the delimiter rule match, the exact rule match in case that the delimiter rule match occur and delimiter character match to transform original packet to delimiter format.

The detail of counting is shown in Figure 6 below. In this tries, there are only 3 rules test, tree and true. The incoming packet has strings “Intrusion” of 9 characters. First the “Intrusion” is checked by indexing using “I” in the top array. Since there is no rule begin with “I” the string is skipped one character to “ntrsuion” and checked for rule start with “n”. All characters in string “Intrusion” are checked by indexing once in the top array except character “t”. Since string “trusion” has the same first tree characters as rule “true”, there are 4 characters checked from the top array to the fourth level array (showed in bold box Figure 6). So the data in the packet are used in indexing once except for some characters that is the beginning part of the rule such as “tru”. The total count for comparison of “intrusion” is 8 (for “inrusion”) + 4 (“trus”) .

In the case of delimiter search, the procedure is the same except rules and packet are comparing with delimiter once to transform in to delimiter format. In this format, the tries structure has less number of levels and packet string length are much shorter.

From the calculation of string “Intrusion”, we can see that once each strings (Intrusion, ntrsion,…..,ion, on,n) is checked by indexing in the tries structure all the rules in the tries are checked. In comparing with the Boyer-Moore, it tries to locate each rules (test, tree, true) in string “Intrusion”. Hence the burst tries structure has much less number of comparison for larger number of rules.

A shell script is used to generate traffic to the linux system and the number of comparison for each NIDS is recorded under the same generated traffic and hardware. The results are shown in figure 7. The burst tries are much better than the original snort with Boyer-Moore. The number of comparison is only 4.96 % of snort’s. .The delimiter version of burst tries is slightly better than the burst tries alone (about 9.29 % faster).

Length of data (bytes)	103	165	204	250	299	348	408	450	508	551
Number of comparing										
Snort’s structure	2998	4714	5994	7267	8972					
Burst Tries	151	233	293	353	441					
Delimiter Burst Tries	149	220	269	322	401					

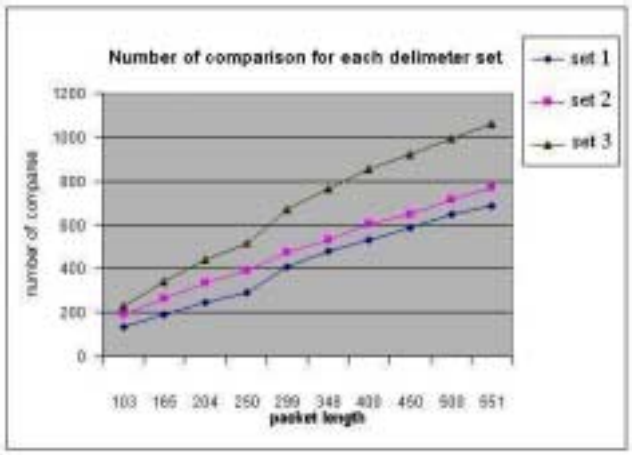

Fig. 8. Number of comparison for each delimiter set

The speed of delimiter burst tries depends on the delimiter. We compare three sets of delimiters shown in Figure 8. It is interesting to notice that some set of delimiters perform worst than the burst tries without delimiters.

6. CONCLUSION

This paper proposes the speed improvement of Network Intrusion Detection Systems analysis. The improvement is achieved by reducing the number of character comparison in each rule through burst tries structure and delimiter search. Transforming packet to delimiter and distance between delimiters reduce number of character comparisons. The result of the proposed method on average take only 4.96 percent of time of original snort

10413
498
437
400
11855
577
515
450
13200
660
584
500
14732
753
680
551
16118
844
786

Fig. 7. Number of comparisons for varied sized packets

REFERENCE

[1] L.T. Habergeon, G.V. Dias, K.N. Levitt, B. Mukherjee, (with J. Wood, D. Wolber), “A Network Security Monitor”. Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy. Oakland, CA, 7-9 May 1990, pp.296-304.

[2] L.T. Heberlein, “Network Security Monitor (NSM) – Final Report”. Lawrence Livermore National Laboratory project deliverable, http://seclab.cs.ucdavis.edu/papers/NSM-final.pdf

[3] Martin Roesch, “Snort – Lightweight Intrusion Detection for Networks” USENIX LISA Conference November 1999.

[4] Paxson, V., Bro: A System for Detecting Network Intruders in Real-Time, Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, January 1998

[5] T. Ptacek and T. Newsham, “Insertion, Evasion, and Denial of Service : Eluding the Network Intrusion Detection,” Secure Networks, Inc., Http://www.aciri.org/vern/Ptacek-NewshamEvasion-98.ps , Jan. 1998.