Efficient Regular Expression Signature Generation for Network Traffic Classification

Vinoth George C[1], Vinodh Ewards[2]

[1]Department of Computer Science and Engineering, Karunya University
Coimbatore, India
vinothgeorge@karunya.edu.in

[2]Department of Computer Science and Engineering, Karunya University,
Coimbatore, India
ewards@karunya.edu

Abstract: ***Regular expression signatures are most widely used in network traffic classification for trusted network management. These signatures are generated by the sequence alignment of the traffic payload. The most commonly used sequence alignment algorithm is Longest Common Subsequence (LCS) algorithm which computes the global similarity between two strings but it fails in consecutive character matches. This paper presents a new divide and conquer alignment algorithm for generating regular expression signature by rewarding contiguous character matches. The results indicate that the sequence alignment algorithm that used is the space efficient way and the algorithm outperforms LCS in terms of efficiency and accuracy.***

Keywords: Traffic classification; Payload based method; Signature, sequence alignment.

1. Introduction

Traffic classification in network system is to identify and monitor the application traffic and to ensure the Quality of Services (QoS) to the user. Through traffic classification network administrator can discover what are applications are run by the end user. Internet applications have been initially identified by port based method which uses well known port numbers in TCP or UDP headers. Generating signatures for applications is not an easy task, in general signatures are generated by manual analysis of protocol information and packet traces. However this approach has become inaccurate and deriving the signatures manually is time consuming. Moreover due to rapid increase in evolution of network application, signatures are subject to change at certain interval of time. To keep the signature up-to-date the high cost manual signature will be repeated time to time. Payload based traffic classification is most widely used in industries due to it automatically generates regular expression signatures based on the payload information without any prior knowledge of port numbers [14][15]. Regular expression signature provides expressive power and flexibility and also supported by various IDSes and traffic classification system like l7-filter [10].

Regular expression signatures are generated by sequence alignment of traffic payload. The process would take labeled training data set as input and generates regular expression signature for matching the application classes presented in the data. The most commonly used sequence alignment algorithm is Longest Common Subsequence [1] which finds the global similarity between two strings and fails to perform consecutive matches in the sequence.

This paper tackles the problem of sequence alignment of payload that rewards consecutive character matches. It is a gap-minimizing alignment model that uses the scoring scheme. This algorithm uses the divide and conquer version for finding sequence alignment.

2. Related Work

Traffic classification is most important for trusted network management. Traditionally port based method is used for application identification but it becomes inaccurate when more and more new applications are developed. Later applications are identified by protocol specification but it is not suitable for new emerging applications. A signature based traffic classification is found in which signature is generated by payload information of the application. This type of classification is widely adopted for intrusion detection system to detect worms in the early stage.

Sequence alignment methods are inspired by methods used in bioinformatics, and gained popularity in network applications [8].

Needleman-Wunsch algorithm is a dynamic programming algorithm used for the extracting optimal alignment from the set of sequence [11]. It is a scoring scheme in which alignment are specified by a similarity matrix. This algorithm maximizes similarity scores to give maximum match. It is a global alignment algorithm that takes quadratic time for gap penalty. After computing the score for every cell a traceback function is called to determine the actual set of operations. This algorithm formulated its problem in terms of maximizing similarity. The problem of this dynamic algorithm is that it take more time to fill the matrix of two sequence.

Smith Waterman algorithm is the local sequence alignment algorithm. It compares the segments of all possible lengths instead of looking at the total sequence and optimizes the similarity measures [4]. It is similar to that of Needleman Wunsch algorithm the main differences is that negative scoring matrix cells are set to zero, which allows the local alignment visible. For every cell in the matrix the algorithm calculates all possible paths leading to it. These paths can be of any length and may contain insertions, deletions and indels. It takes on gap penalty for better sequence alignment.

Jacobson-Vo is a flexible gap minimizing alignment model for sequence alignment [9]. The runtime performance of this algorithm depends on the distribution of characters in the input strings. It is the pairwise alignment method that includes precise locational information per substring. To extract the common subsequence an arbitrary element in the last subsequence is selected after that remaining sequence of strings are selected for alignment process. It strictly follows the goals of substring maximization and gap minimization and collects the alignment via back pointer traversal. Finally this algorithm outperforms Smith-Waterman approach in terms of efficiency in generating alignment sequence..

3. Sequence Alignment through Longest Common Subsequence Algorithm

The Longest Common Subsequence algorithm is used for finding common subsequences from the set of sequences [3]. LCS is normally used in bioinformatics for pattern matching in DNA sequence. In case of using this algorithm in sequence alignment of traffic payload there must be some alteration in the algorithm. This algorithm has various constraints like minimum substring length, number of packets per flow and packet size comparison to find the common sequences from the packet trace. Sequence alignment is important in computer science, while constructing an alignment two strategies are followed. Global alignment focuses on the alignment with full sequences and the local alignment which emphasizes the subsequence of the input sequence.

Consider the two sequences of the same length, and then shorten each sequence by removing the last element to find the common sequence and finally append the removed element. If the two sequences of not same length then the common sequence will be the longer of two sequences. LCS is a pairwise alignment model that includes maximizing a similarity scoring model. LCS process the scoring matrix row by row to yield the better alignment. Setting scoring parameters is for the efficient sequence alignment of strings. The major issue with the LCS algorithm is that it finds similarity in terms of number of matched characters rather than consecutive matches. Consider the following sequences S and T shown in Fig.1. Sequence alignment is performed based on the LCS algorithm.

```
S: IN----------TERESTINGLY
T: INFORMATIC--SBI----LY
      LCS: INTSILY
```

Figure 1: Alignment of Strings using LCS algorithm

LCS fails to match the contiguous characters therefore it maximizes the gap while aligning the sequences. Thus it produces signature of false positive [2]. The quadratic time taken for LCS algorithm for sequence alignment is O(nm) and the linear space bound will be 2xO[min(nm)]. The algorithm with accuracy spends too much time and input limit. The strategy of building guide tree would be based on some specific inputs. Thus makes the LCS algorithm as an inefficient way to generating signatures for traffic classification.

4. Efficient Sequence Alignment by Divide and Conquer Alignment Algorithm

By studying the various sequence alignment techniques and to reward contiguous character matches an extended version of Hirschberg's algorithm [5] is used for sequence alignment of payload traffic to generate regular expression signature. It is generally applicable algorithm for optimal sequence alignment which is observed from the computational biology. It solves the string matching problem in linear space and determines the edit distance for two strings and alignments. The algorithm divides the string S into S1and S2 and for the string T it finds the prefix T1 and T2. The alignments of S and T will be formed by the concatenation of alignments for T1 and S1 and then for rev(S2) and rev(T2). String S is bisected, but the procedure of dividing T relies on the following statement. For any *i* dividing S into two parts.

$$\min\{d(S(1..f),T(1..f))+d(rev(S(i+1..|S|)),rev(T(j+1..|T|))\} \quad (1)$$

The edit distance for the entire string S and T will be equals the minimum string among all combined edit distance for the first two parts of these strings and their second parts. The alignment can be divided into two sub alignments one of the strings S(1..f) and T(1..f) and one for string rev(S(i+1..|S|))and rev(T(j+1..|T|)). The ultimate task is to find the prefix T(i..f) of T that can be matched with the first half of S and thereby the suffix T(j+1..|T|) of T that can be matched with the second half of S.

The entire recursive algorithm for the sequences S and T produces the tree as shown in Fig.2. The leave of the tree contain optimal alignment.

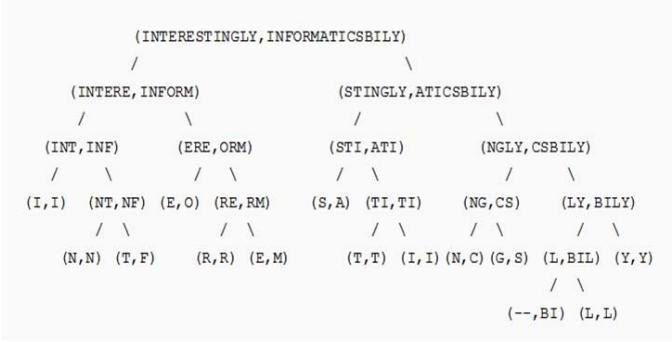

Figure 2: Construction of alignment tree

This is the extended version of Hirschberg's algorithm it minimizes the gap by matching consecutive characters from the set of strings. The alignment of the set of strings is shown in the Fig.3.

```
S: INTERES--TING---LY
T: INFORMATICSBILY
      ALIGN: INRTILY
```

Figure 3: Alignment of Strings using Divide and Conquer alignment algorithm

It is the flexible gap minimizing alignment model suitable for network traffic. The algorithm recursively divides the strings to extract the sequence alignment. The algorithm still takes the O(nm) time, but needs only O(min{n,m}) space. Thus it is also a space efficient way to perform sequence alignment between two sets of data.

5. Signature Generation Methodology

The architecture of signature generation using divide and conquer alignment algorithm is shown in Fig.4. The various components in signature generation process are (1) Preprocessing module that extracts packet payload. (2) Substring Extraction module extract the string sequence from the payload data. (3) Sequence alignment is the module used for extracting common subsequence from the strings. (4) Signature construction through the common sequence. The resulted signature is in regular expression format.

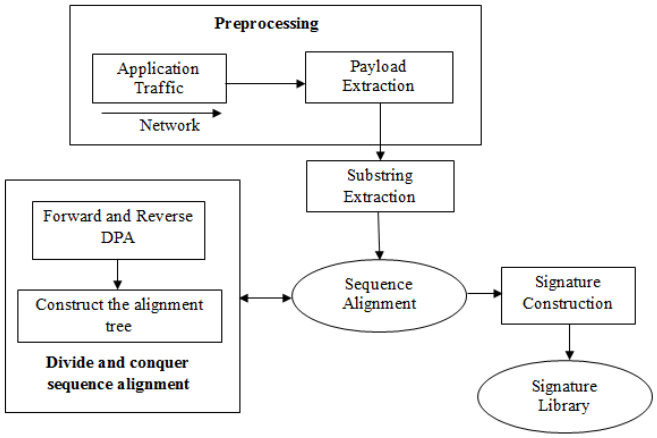

Figure 4: System Architecture

Preprocessing

This module observes raw packet from the network traffic based on the 5-tuples. The raw packets that is referred here are belongs to the target application. To collect the packet trace for every running process in the OS the tools like Libpcap or Wireshark is used. The extracted packet contains all the header information and the data payload of particular application as shown in Fig.5. It divides the each packets based on the payload size, number of packets per flow. The tools that used extract the network information from the default interface from which the application runs.

Substring Extraction

This module extracts the string sequence from the packet information. The string contains the payload data which is in hexadecimal format or as strings. Standard library is used to extract the strings of payload and these strings were used to find the applications. Substring extraction can be done by suffix tree algorithm. Most of the automatic signature generation techniques use the substring extraction by processing the suffix and prefix of the payload string sequence.

```
**********************ICMP Packet************************

Ethernet Header
   |-Destination Address : 00-0B-86-13-03-80
   |-Source Address      : 0C-EE-E6-87-91-CC
   |-Protocol            : 8

IP Header
   |-IP Version        : 4
   |-IP Header Length  : 5 DWORDS or 20 Bytes
   |-Type Of Service   : 0
   |-IP Total Length   : 84  Bytes(Size of Packet)
   |-Identification    : 0
   |-TTL      : 64
   |-Protocol : 1
   |-Checksum : 62830
   |-Source IP        : 10.10.3.226
   |-Destination IP   : 74.125.236.209

ICMP Header
   |-Type : 8   |-Code : 0
   |-Checksum : 60256

IP Header
    00 0B 86 13 03 80 0C EE E6 87 91 CC 08 00 45 00          .....▯▯........E.
    00 54 00 00                                              .T..
UDP Header
    40 00 40 01                                              @.@.
Data Payload
    15 AA 0B CB A3 32 53 50 FB A3 0E 00 08 09 0A 0B          .....2SP........
    0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B          ................
    1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B          .... !"#$%&'()*+
    2C 2D 2E 2F 30 31 32 33 34 35 36 37                      ,-./01234567

###########################################################
```

Figure 5: Packet information extracted using Libpcap

Sequence Alignment

This module used to extract the common subsequence from the set of strings to get an alignment of strings to construct the signatures for every application. Divide and conquer alignment algorithm uses the scoring matrixes in which scores are identified based on the forward and reverse DPA which is the pairwise alignment of the strings [6]. The alignment can be verified by backtracking corresponding scoring matrix of both the formats. The entire recursion algorithm produces the alignment tree based on the progressive alignment and the leaves of the alignment tree contain the optimal alignment. The sequence alignment is performed for all the payload strings of particular application and a single sequentially aligned string is extracted [7].

Signature Construction

This is the final module which transforms the sequentially aligned strings into regular expression signatures. This can be done by replacing the strings with regular expression symbols. The generated signatures will be stored in the signature library which will be used for identifying the applications.

6. Signature Evaluation

In this study we consider various applications like FTP, HTTP, skype and yahoo. Signatures are generated based on the two sequence alignment strategies like LCS and divide and conquer alignment. These signatures were used in the traffic classification system like l7-filter. Both the alignment process takes the same time for extracting common subsequence but the difference is linear space. Divide and conquer alignment takes minimum space for generating payload alignment strings. Comparison of signatures generated by both the alignment process is shown in Table 1.

Table 1: Comparison of Regular expression signatures

Application	*Signature generated by LCS*	*Signature generated by Divide and conquer alignment*
HTTP	^HTTP/1.1|Host:.u|Connection:keep-alive|User-Agent:Mozilla/5.0|Accept:|Accept-Encoding:ga|Accept-Language:en-US.*en;q=0.8|Accept-Charset:ISO-8859-1	^HTP/1|1|Host:|e|Conection:kep-alive|User-Agent:Mozila/5|0|Acept:|Acept-Encoding:ga|Acept-Language:en-US,en;q=0|8|Acept-Charset:ISO-859-1
FTP	^200ettr|.{rw}-.*/x409ar72005pt--a|.{rw}-.*/x06u2200pt--ati-|.{rw}-.*/x40a2200pttab|-.*/x409t1204uo	^20etr|{rw}-.*/x409ar7205pt-a|{rw}-.*/x06u220pt-ati-.rwr-.*/x40a220ptab.rwr-.*/x409t1204uo
Skype	^.F..-...../x02 ./x02 .....@.....	^.F.-./x02 ./x02 .@.
SMTP	^220[\x09-\x0d -~]* (e?smtp|simple mail)	^22[\x09-\x0d]*(e?smtp)
POP3	^(\+ok .*pop)	^(\+ .*pop)
Counter Strike	^.*.y.,?.Ei.IP an.de_s.cstrike.Counter-Strikeoure	^.*.y.,?.E.i.I lae.de_s.cstrike.Counter-Strike.dw.
Yahoo	^CA-30.120310000000Z|140314120000Z0^1.0.U|US1.0.U|CA1.0.U|Sunnyvale1.0.U|Yahoo! Inc.1.0.U|login.yahoo.com0|	^CA-30|120310Z|140314120Z0^1|0|U|US1|0|U|CA1|0|U|Sunyvale1|0|U|Yaho! Inc|1|0|U|login|yaho|com0|
Bittorrent	^(\x13bittorrent protocol|azver\x01$|get /scrape\?info_hash=)	^(\x13bittorrent protocol|get \?info_hash)

The signatures shown in Table 1 are used in l7-filter for traffic classification and the new alignment model outperforms the previously used LCS algorithm in terms of space and efficiency. The performance of signature generation can be improved by optimization, and the algorithm can be parallelized to allow the signature generation time to be reduced significantly. The algorithm takes O(nm) space for sequence alignment, but it needs only O(min{nm}) space. This alignment process produces less false positive and false negative signatures. It is a gap minimizing alignment model that generates efficient regular expression signatures for network traffic classification. These signatures have been used in the intrusion detection system for identifying worms [12] [13] and in the traffic classification system like l7-filter for identifying applications..

7. Conclusion

This paper deals with the sequence alignment problem in generating application signature for network traffic classification. Previously used LCS algorithm failed to match the consecutive characters thus increases the gap. To tackle this problem a divide and conquer alignment model which is derived from Hirschberg's algorithm is used to minimize the gap by rewarding contiguous character matches. This is the space efficient way to perform sequence alignment between two sets of data. The signature generated will be close to the real life handcrafted signatures because it contains regular expression features that are necessary for identifying applications. Thus this sequence alignment algorithm improves the efficiency and accuracy of the signature.

References

[1] Amir A, Hartman T, Kapah O, Shalom R, Tsur D. Generalised LCS. In: Proceedings of Theoretical Computer science archive, ACM vol409, p. 438-449, 2008.

[2] Byung-Chul P, Won YJ, Myung-Sup K, Hong JW. Towards automated application signature generation for traffic identification. In: Proceedings of the Network Operations and Management Symposium, 2008. NOMS 2008. IEEE; 2008, p. 160–7.

[3] Chen J. B. A survey of the longest common subsequence problem and its related problems. Submittted at Department of CSE, National Sun Yat-sen University, 2005.

[4] Costas S, Rahman S. A New efficient algorithm for computing the longest common subsequence. In: Proceedings of TOC Systems- Symposium on Parallelism in Algorithms and Architectures. ACM vol 45;2009.

[5] Drozdek A. Hirschberg's algorithm for approximate matching. In: Proceedings with Computer Science, Duquesne University, Pitssburg, p.91-100,2002.

[6] Haque W, Aravind A, Pairwise sequence alignment algorithms. In: Proceedings in ISTA '09, Pages 96-103, ACM 978-1-60558-478-2.

[7] Hirschberg D. S. A linear space algorithm for computing maximal common subsequence.Princeton University. Published in Communications of the ACM CACM, Vol 18, p-341-343, 1975.

[8] Johnstone J. A survey on sequence matching and alignment algorithms.

[9] Kreibich C, Crowcraft J. Efficient sequence alignment of network traffic. In: Proceedings IMC '06.

[10] Kim H-A, Karp B. Autograph: toward automated, distributed worm signature detection. In: Proceedings of the 13th Conference on USENIX Security Symposium, vol. 13, San Diego, CA, USENIX Association; 2004.

[11] Sara A, Arabi K, Hany M. Fast dynamic algorithm for sequence alignment based on bioinformatics. International Journal of computer application vol 37, 2012.

[12] Tang. Y., Xiao. B. and Lu. X. (2009) "Using a bioinformatics approach to generate accurate exploit- based signatures for polymorphic worms." Computer Security; 28(8):827–42.

[13] Wang. K, Cretu G , and S. J. Stolfo. Anomalous payloadbased worm detection and signature generation. In Proc. Of Recent Advances in Intrusion Detection (RAID), 2005.

[14] Wang Y, Xiang Y, Yu. SZ. An automatic application signature construction system for unknown traffic. Concurrency and Computation–Practice and Experience 2010;22(13):1927–44.

[15] Wang. Y., Xiang. Y., Zhou. W. and Yu. S. (2012) "Generating regular expression signatures for network traffic classification in trusted network management." In: Proceedings with Journal of Network and Computer Applications Volume 35, Issue 3, Pages 992–1000.

Author Profile

Vinoth George C is a Post Graduate Scholar in Karunya University.

Vinodh Ewards is an Assistant Professor (SG) in Karunya University.