ISSN:0975-9646

Region Based Algorithm Selection for Network Security

V.Asaithambi[#1], Dr.N.Rama[*2]

[1#]*Department of Computer Science, Govt. Arts College*
Nandanam,Tamilnadu,India-600 035.
[2*]*Department of Computer Science, Presidency College*
Chennai,Tamilnadu,India-600005.

***Abstract*-Generally to secure a network from denial-of-service to Smurf attacks, hackers that perpetrate exploits, it is necessary to perform the tasks like Searching for multiple strings in packet payloads, approximate string matching, IP traceback via probabilistic marking, IP traceback via logging, detecting worms, etc. To execute the tasks, many algorithms are used. Breadth First Search (BFS), Depth First Search (DFS) are two standard search algorithms used to traverse any network. The search performance of these algorithms may differ node to node. i.e. The search performance of BFS may be better for some nodes and the search performance of DFS may be better for some other nodes. To search a particular node, only one algorithm is enough when it is better than the other. This paper gives an idea to select an algorithm for a node before start the search, such as the algorithm is better than the other algorithm in the aspect of search performance.**

***Keywords* — BFS, DFS, Network Security, Tree, NIDS.**

I. INTRODUCTION

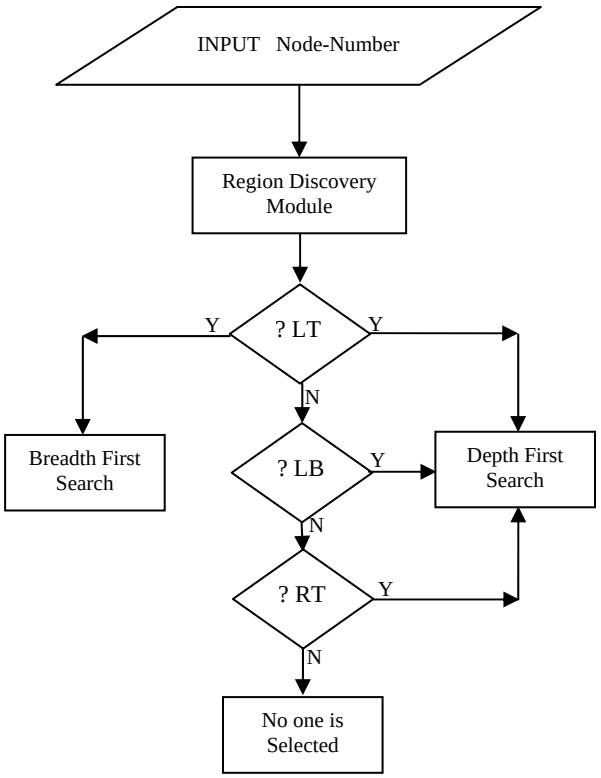
Now a days internet is required everywhere and in everything for each and every activity. The internet is a fully connected network by which all networked equipment can be connected. The network is defined as connection of computers or other components with a communication media like wire or wireless. To identify a component like computer in a network, many search algorithms are used. They are called traversing algorithms and also used in intrusion detection of network security.

II. LITERATURE SURVEY

All the network activities are monitored by a network based intrusion detection system (NIDS). The process of monitoring the events occurring in a network is called intrusion detection[2]. The various attacks in a network are monitored and the traffic in the network is analyzed to make the network performance good. To analyze the network traffic, the traversal algorithms are used. Also the algorithms are used in host based intrusion detection. The network elements like switches, routers and gateways are considered as nodes of the network[3]. Paxson's Bro monitors packet traffic entering and leaving the target network[4]. All string matching algorithms compare a set of strings in the rule-set to the data seen in the packets that flow across the network [5]. The brute force algorithm simply attempts matching the pattern in the target at successive positions from left to right[6]. It is also called native searching algorithm and it requires only a fixed amount of extra memory space. Boyer Moore is the most widely used algorithm for string matching in intrusion detection system, the algorithm compares the string of the input starting from the rightmost character of the string[6]. A depth-first search (DFS) is one of a blind search techniques. It extends its path all the way to a leaf before backtracking and finds another path[1]. A breadth-first search (BFS) explores nodes nearest the root before exploring nodes further away[1].

III. SYSTEM ARCHITECTURE

The following figure Fig.1. shows that the flow of the system. The Region Discovery Module contains two techniques. First one divides a tree horizontally and second one divides the tree vertically.



IV. PROPOSED WORK

A tree can be divided in to four regions. Left Top(LT), Left Bottom(LB), Right Top(RT) and Right Bottom(RB). All the nodes in a region has similar properties. For example the performance of the depth first search on the nodes in LB region is better than the performance of the breadth first search on the nodes in the same region. So the region of a node tells that the search algorithm well suited for the same node. To discover the region of a node the region discovery module is used. It has two techniques

1. Divide by two technique
2. Vertical chopping technique

A tree in Fig.2 can be represented using parent array[1] as shown in Fig.3. The tree in Fig.2 has 11 nodes and 10 links. The parent array is a one dimensional array in which only the parents of a tree are stored and the children are used as indices[1].

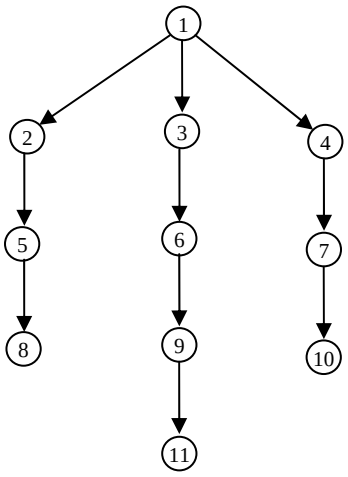


Fig 2. A General Tree

1	2	3	4	5	6	7	8	9	10	11
0	1	1	1	2	3	4	5	6	7	9

Fig.3. Parent array for the tree in Fig.2

Divide by two technique: The tree can be divided horizontally by using this technique. This technique discovers a node at which the tree is divided into top region and bottom region. All nodes in a level in which the identified node is existing, forms the horizontal boundary for the tree. This can be discovered as follows.

Boundary level = ceil(Total number of levels/2)
Boundary node = last node(boundary level)

For example in the given tree in Fig.2

Boundary level = ceil(5 / 2)
= 3.
Boundary node = last node(level 3)
= 7

All the nodes which has node number less than and equal to boundary node number are considered as top region nodes and remaining all nodes are considered as bottom region nodes.

Vertical chopping technique: The tree can be divided vertically by using this technique. This technique gives vertical boundary for a tree. In this boundary the first node is the root node. The middle child of the root node is taken a second node of the vertical boundary. Then the right most child of the second node is considered as third node. Similarly all the right most child from this node are considered recursively until reach the leaf node. In the example tree, the nodes 1,3,6,9,11 are in vertical boundary. Then the leftmost nodes in each level are taken and considered as left boundary. In the example tree the nodes 1,2,5,8,11 are in left boundary. The nodes between vertical boundary and left boundary in each level are considered as left region nodes. All the remaining nodes in the tree are considered as right region nodes. The following table shows the left region and right region nodes in the given tree.

Table-I. Vertical boundary table

Level	Left Boundary	Vertical Boundary	Left Region Nodes	Right Region Nodes
1	1	1	1	-
2	2	3	2,3	4
3	5	6	5,6	7
4	8	9	8,9	10
5	11	11	11	-

V. RESULT AND DISCUSSIONS

A tree generator was used to generate five random trees with number of nodes 7000, 10000, 12000, 15000 and 17000. Also a tree partition tool was developed using C language to divide the tree in both vertically as well as horizontally to get the four regions using the above discussed techniques. All the nodes in all the trees were tested through a region locater software and all the data were collected. The following table shows the node distribution in each tree.

Number of nodes	Left Top	Left Bottom	Right Top	Right Bottom
7000	1872	2756	985	1387
10000	2701	3032	1927	2340
12000	3377	4725	1823	2075
15000	3007	5013	2902	3483
17000	4291	4892	3234	4583

Table-II. Region-wise node distribution

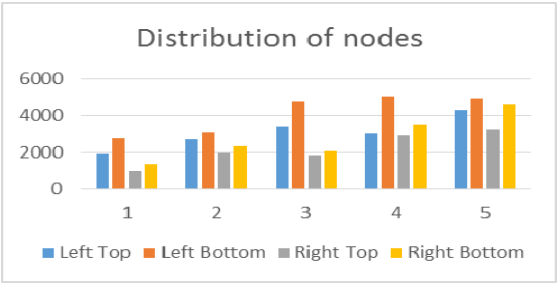


Fig 4. Random Distribution of nodes

Number of nodes	Algorithm	Left Top	Left Bottom	Right Top	Right Bottom
7000	DFS	1465	2749	4	692
	BFS	307	7	981	695
10000	DFS	2247	3027	5	1174
	BFS	454	5	1922	1166
12000	DFS	2986	4711	4	1041
	BFS	391	14	1819	1034
15000	DFS	3152	4999	6	1740
	BFS	913	14	2896	1743
17000	DFS	4006	4889	8	2289
	BFS	285	3	3226	2294

Table-III. Region-wise algorithm selection

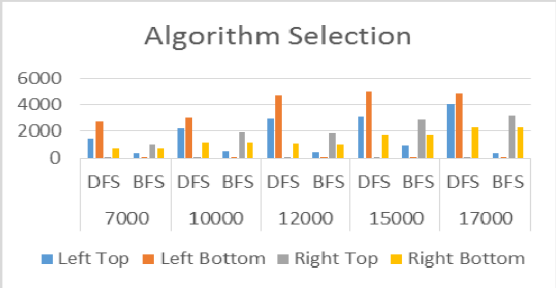


Fig 5. Selection of Algorithm

VI. Conclusion

The traversal algorithms are chosen based on the region in which a node is existing. The region of a node can be identified using simple calculations explained in this article. This technique is used to select right algorithm for searching right node instead of using blind search technique. The search with the prior knowledge of the node increases the speed of searching. This can be used in network security for easy and fastest traversing.

References

[1] Asaithambi V, Zackariah N, Dr.Nirmala K, Decision Equations for Efficient Search Algorithms for Network Security, IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012) March 30, 31, 2012, ISBN: 978-81-909042-2-3 ©2012 IEEE,p:285-289

[2] G.C.Tjhai, M.Papadaki, S.M.Furnell, N.L.Clarke, Investigating the problem of IDS false alarms An experimental study using Snort, internet, 253-267, 2008.

[3] Anand Patwardhan, Jim Parker, Anupam Joshi, Michaela Iorga, Tom Karygiannis, Secure Routing and Intrusion Detection in Ad Hoc Networks, Proceedings of the 3rd International Conference on Pervasive Computing and Communications(PerCom 2005), Kauai Island, Hawaii.

[4] Vipin Das, Vijaya Pathak, Sattvik Sharma, Sreevathsan, MVVNS.Srikanth, Gireesh Kumar T, Network Intrusion Detection System Based on Machine Learning, International Journal of Computer Science & Information Technology (IJCSIT), Vol 2, No 6, December 2010.

[5] Tuck N., Sherwood T., Calder B., Varghese G., "Deterministic Memory-Efficient String Matching Algorithms for Intrusion Detection", Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, VOL.4, PP.2628 –2639, 7-11 March 2004.

[6] Qutaiba Ibrahim, Sahar Lazim, Applying an Efficient Searching Algorithm for Intrusion Detection on Ubicom Network Processor, *International Arab Journal of e-Technology, Vol. 2, No. 2, June 2011*

[7] C. J. Alpert and A. B. Kahng UCLA Computer Science Department, Los Angeles, CA 90024-1596 : *A General Framework for Vertex Orderings, With Applications to Netlist Clustering*

[8] Danny Z. Chen, Ovidiu Daescu, Xiaobo (Sharon) Hu and Jinhui Xu Journal of Algorithms Volume 49, Issue 1 , October 2003, Pages 13-41 *Finding an optimal path without growing the tree*

[9] Matti Nykänen and Esko Ukkonen, Department of Computer Science,University of Helsinki, P.O. Box 26, (Teollisuuskatu 23), 00014, Helsinki Finland Journal of Algorithms Volume 42, Issue 1 , January 2002, Pages 41-53 *The Exact Path Length Problem*.

[10] Daniel K. Blandford Guy E. Blelloch Ian A. Kash, Computer Science Department Carnegie Mellon University, Pittsburgh, PA 15213 fblandford,blelloch,iakg@cs.cmu.edu, *An Experimental Analysis of a Compact Graph Representation*.

[11] www.cs.uni.edu/~schafer/courses/161/sessions/s09.ppt.

[12] http://www.ugrad.cs.ubc.ca/~cs320/Lectures/cs320lec5.pdf.

[13] http://www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html.

[14] http://wwwin.cisco.com/cpress/cc/td/cpress/internl/dns/index.htm

[15] Sabih H. Gerez.(1999) John Wiley & Sons, *Algorithms for VLSI Design Automation*.

[16] N.A. Sherwani. (1999) Tata Mc Hill, *Algorithm for VLSI Physical Design Automation*.

[17] Ellis Horowitz and Sartaj Sahni, (1999) Tata Mc Hill, *Fundamentals of Computer Algorithms*.

[18] Matt Curtin (1997) Kent Information Services, *Introduction to Network Security*.

[19] S.Anuradha, G.Raghu Ram, K.E.Sreenivasa murthy, V.Raghunath Reddy and D.R.Srinivas (2009) International Journal of Recent Trends in Engineering, *DB Routing algorithm*.