GEOMETRIC EFFICIENT MATCHING ALGORITHM FOR FIREWALLS

Phaltane Anjali.D[1], Jondhale S.D[2]

[1,2] *Computer Engineer, Computer, P.R.E.C Loni B.K, Maharashtra, India*

Abstract

Concept of Firewall is the most important thing in network and the traffic which is passing through network perimeter needs to be filtering the traffic that is going to pass through it. Thus there is potential risk in this process. As each packet needs to be checked with each firewall rule to find the matching rules. 'Geometric Efficient Matching Algorithm ' is one of the computational geometry algorithm which gives practically better solution for the purpose of finding the rule which exactly matches. With the help of firewall rule statistics we have generated random model of perimeter rule which is not uniform. We also reduced the space requirement up to 2-3 MB for 5000 rules. Also it solves problem of firewall misconfiguration for firewall packet matching

Key Words: *Network level protection and security, Network Communication.*

----------***----------

1. INTRODUCTION

An intranet can have connections to outside internet but such connections are usually protected through a security mechanism called as firewall. A firewall may be hardware or it may be a software program running on a secure host computer. Firewall examines all traffic routed between two networks to see if it meets certain rules. If it does, it is routed between the networks otherwise it is stopped. Firewall filters both inbound and outbound traffics. But most of the network administrator provides less attention over outbound traffic. There are different types of firewalls. Whenever we are dealing with packet matching firewall, it filters the traffic based on source and destination IP address, protocol number and ports. They are also important in deciding which rule should be applied to a particular packet. Routers are different from firewall; it matches the packet based on IP address.

As packet matching firewall needs to consider different issues like firewall rule recertification, firewall misconfiguration etc. That's why there should be special algorithm for firewall. Firewall filters both inbound and outbound traffics. But most of the network administrator provides less attention over outbound traffic.

There are different types of firewalls. Whenever we are dealing with packet matching firewall, it filters the traffic based on source and destination IP address, protocol number and ports. They are also important in deciding which rule should be applied to a particular packet. Routers are different from firewall; it matches the packet based on IP address. As packet matching firewall needs to consider different issues like firewall rule recertification, firewall misconfiguration etc. That's why there should be special algorithm for firewall.

2. EXISTING SYSTEM

Most of the firewall s which are used now a day is of type statefull. Statefull firewall matching is a type in which whenever the first packet of the flow is allowed to pass through a firewall then remaining traffic of the current flow will also be allowed to pass through that firewall.

And also all the outbound traffic related to that flow will be allowed to pass through the statefull firewall. That means statefull firewalls provides less security for the outbound traffic. The network administrator does not write outbound rules so strictly but the outbound traffic is not so secure.

Now suppose if the user in outbound in network is accessing web mail, news sites or Facebook may be having any type of intrusion such as Trojan horse in their desktop ,then this traffic passing through the firewall will not be block. So the network administrator should have to take care of the outbound rules because the outbound can also become inbound as well.

The implementation of state fullness of firewall is basically done by two different searching mechanisms 1) The slow algorithm 2) The Fast state lookup mechanism. Such a design of the statefull firewalls gives a high performance for the TCP connections which are explicitly long and in this case the mechanism fast state lookup let most of the packet travel through firewall. As it is useful for long TCP connections but such design will become a bottleneck for the connectionless UDP, short TCP connections and also for ICMP traffic.

We are going to show that 'Geometric Efficient Matching Algorithm' is best as its packet matching speed is same as of the slow algorithms and also there is no loss of packets on the host computer.

3. PROPOSED SYSTEM

In this paper we are using one of the computational geometry algorithm names as 'Geometric Efficient Matching Algorithm' for firewall. We are using this algorithm for the purpose of packet matching in firewall. The time required for this algorithm for packet matching is O (n log d) where d is number of rules in the rule-bases of firewall and n is the total number of fields to be match. This algorithm has a space complexity (worst-case) of O (d^n).For the protocols UDP and TCP n=4 and the search time required for this is O (log d) with a worst–case space complexity of O (d^4).

The data structure of this algorithm has a tremendous control on the order of fields that is to be matched. We have created rule-bases for a firewall in such a way that no attacker can attack and decrease the performance of firewall. The fields to be match are: the source and destination IP address, port numbers of source and destination which is more suitable for filtering of UDP, TCP as well as ICMP

We should note that only a structure of bad rule bases will effect on the space complexity and not because of the Packets which are encounter by the firewall. For evaluating the 'Geometric Efficient Matching Algorithm' we have studied random rule simulations which shows that 1 micro second per packet can pass through firewall using this algorithm.

The rule-bases which we have collected are selected from Algosec Firewall Analyzer in random fashion i.e. selection of destination port is not a range rather it is taken from set of 200 values which are the common values.

We have generated a random rule that is a perimeter rule model. With the help of this rule bases we analyze that the order of fields is generally responsible for size of data structure. The order of evaluation is: the source and destination port number, source and destination IP address. With the help of this evaluation order we will reduce the overall space complexity of this algorithm.

If we are using 2 part heuristic splitting approach then for rule bases of 5000 rules we require a size of data structure about 15 MB .But if are using 3-part heuristic splitting approach than 2 MB of data structure size is required for about 10000 rules.

4. THE ALGORITHM

4.1 Definitions

The packet matching in firewall finds the first rule which will going to match more than one fields from its packet header. Each rule is having a set of ranges [Sr, Pr] for r=1, 2…..n in this each range will correspond to nth field of the packet header. Fields are having values 0<=Sr, Pr<=Qr where Qr =2^32 -1 for the IP address and the value of Qr=65535 for port number, and Qr =255. The header field numbering is as shown in the following table 1.

Some of the firewall do matching of packets based on some additional fields of header .E.g. TCP flag, IP option as well as packet payload. So all rule bases do not consider these things. All the firewall rules uses generally only 5 fields that is source and destination IP address , source and destination port numbers and the protocol field. The 'Geometric Efficient Matching Algorithm' is used generally for the rules which use IP addresses of contiguous ranges which can also be used for the enterprise firewalls. As an enterprise firewall uses contiguous range of IP addresses. '*' in the field n indicates any value in [0, Qr]

Table-1: Header field Numbering

Numbering	Description	Storage Required
0	The IP address of source	32 bit
1	The IP address of destination	32 bit
2	Port number of source	16 bit
3	Port number of destination	16 bit
4	Protocol field	**A** bit

4.2 The sub-division of Space

Consider sub-division of one dimensions, in this one range is defined by each and every rule and that splits the overall space into 3 parts. If there are x rules that are overlapping then the one dimensional space is divided into a simple range of (2x-1). A number of winner rules are being assigned to each of these simple ranges. And in the case of n-dimensions all the rules are projected on one of the axis which we have to choose for the projection which will actually reduce the space of set of rules to (n-1) than using one dimension sub-division. These rules are called 'active rules'. Continuing in this fashion, we recursively sub-divide (n-1) dimensional space. For one dimensional space algorithm we uses 1 level of sub-division and for 4-dimensional we are using 4-level of sub-division. In this way n-dimensional space is converted to simply hyper-rectangles, each hyper-rectangle representing a winning rule and it is then translated into a search algorithm.

4.3 Protocol Field

We have to consider the protocol header field before considering the search data structure. This protocol field differs from rest of 4 fields: some of the 256 possible values are generally used to define a numerical range of values of the protocol. All this is collected and validated from real firewall rules. The values seen in the protocol field of the firewall rules are the specific protocols and wildcards '*' and not a non-trivial range is there.

The Geometric Efficient Matching Algorithm has to deal only with the single value present in the protocol field and there is a special treatment for rules having'*' as a protocol. All the firewall rules are preprocessing by the protocol and we build search data structure from this for each protocol. The GEM search algorithm only operates on 4 fields from header field.

A packet not only belongs to one protocol but also it is affected by the protocol='*' rules. For that reason each and every packet has to be searched twice:1) In its own protocols data structure and 2) In '*' structure. Each search will result into a winner rule. In the rest of this paper, TCP protocol is focused and that protocol has n=4 dimensions, also same is applied for ICMP as well as UDP.

4.4 GEM Search Data structure

The search data structure for GEM algorithm has three parts. The first part contains array of pointers for each protocol number with cell which contains '*' protocol. The second and third part is built separately for each protocol. The second part of the GEM search data structure contains protocol database header which generally consists of information which is about order of data structure levels. The fields of packet header are checked in an order and in same order it is being encoded as 4 tuple of field numbers with the help of numbering shown in Table1. The protocol database header has pointer to the first level as well as pointer to a number of simple ranges in that level.

The levels of data structure are represented in the third part of the GEM search data structure. Each level is nothing but the set of nodes and each node is an array. Also each array cell defines a simple range, and also specifies a pointer to next node on next level. The last level contains the simple range information which consists of the number of winner rule.

4.5 Search Algorithm

The packet header field consists of 4 fields: source IP address, destination IP address, protocol number, port number. The protocol number field is first checked and then to select a protocol database header field we have to go to search data structure. Binary search is applied on each and every level to find the simple matching range level by level. The final level will give us the desired result that is the number of the matching rule.

This searching procedure is repeated for '*' protocol to find another matching winner rule. From these two we select one with having lower rule number.

Binary search is applied on an array of 2x entries, where x is maximum number of active rules. Two searches are carried out: one for packets protocol and one for search in'*' data structure. Search time required for d levels is O (d log x). The '*' search data structure have only 2 levels for the IP address , so the search time is generally dominated by the search time for levels of TCP search data structure.

4.6 Build Algorithm

For each protocol the build algorithm is executed once. The rule-base and the field order to be used are given as an input to the build algorithm. The order field is the most important in the contents of each level of the data structure as well as the header fields tested order is also important. In order to check 4 fields we choose 4! =24 possible orders. The geometric sweep-line algorithm is used to build the data structure.

In the same manner all the levels of the search data structure are built. Starting with active rule set from the previous level, all the rules with certain protocol are active for the first level. Then for this level we built the set of the critical points are nothing but the end points of ranges that are projection of active rules on axis corresponding to the recently checked fields.

The lists of the critical points are sorted in ascending order and run sweep-line algorithm. Two implicit critical points are present there: max value for each level and 0. Each and every critical point is corresponding to starting of simple range which relates to the active rules subset.

The set of active rules are calculated with the help of each simple range for that purpose we choose all rules which is overlapping with the simple range in the current field. From this new active rule set, we continue to the next level for each simple range. We have to list the number of "winner rule" in the bottom most level.

The space complexity in the worst case for GEM algorithm is O (n^d) and the space complexity for TCP or UDP. The build time for Gem algorithm is O ((n log n)^d) for d levels. The space complexity of GEM algorithm is better than that of naïve linear search algorithm.

5. CONCLUSIONS AND FUTURE WORK

We have seen that for firewall packet matching the Geometric Efficient Matching Algorithm (GEM) is very efficient and practical approach. Its packet matching speed is analyzed on live traffic with the real firewall rule bases. The packet matching speed of the GEM algorithm is better than that of naïve linear search approach which is used in the existing system. On realistic statistics we have generated the real firewall rule bases, the space complexity of GEM algorithm is also better. Thus for firewall packet matching GEM algorithm will be a better solution.

We have to note that some algorithms of Qiu et al [25], Gupta, and McKeon [16] may be well solution for software implementation in kernel. Thus implementation and testing of this entire algorithm will also be interesting by using rule base, same hardware and the traffic load.

Also we will use more than 4 fields and explore the GEM algorithms behavior on it. How to encode the non-range fields? How to achieve the best space complexity? What will be the order of header fields? Also how GEM algorithm will perform for IPV6 with IP addresses of 128 bits?

REFERENCES

[1]. G. Varghese, Baboescu and Singh S, "Packet classification for core routers: Is there an alternative to cams", in Proc. IEEE INFOCOM, 2003.
[2].D.P.Dobkin, R.J.Lipton,"Multidimensional searching problems", SIAM J. Compute. , vol.5, no. 2, pp. 181-186, 1976
[3]. V.Srinivasan,"A packet classification and filter management system", in Proc. IEEE INFOCOM, 2001, pp. 1464-1473.
[4]. M. Wald Vogel, G. Varghese, J. Turner, and B. Plattner, "Scalable high speed IP routing lookups," in Proc. ACM SIGCOMM, September 1997,pp. 25–36.
[5]. V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel, "Fast and scalable layer four switching," in Proc. ACM SIGCOMM, 1998, pp.191–202.
[6] P. R. Warkhede, S. Suri, and G. Varghese, "Fast packet classification for two-dimensional conflict-free filters," in Proc. IEEE INFOCOM, 2001, pp. 1434–1443.
[7] Dmitry Rovniagin and Avishai Wool, Senior Member, IEEE,"Geometric Efficient Matching Algorithm for the firewall"
[8] D. Decasper, Z. Dittia, G. Parulkar, and B. Plattner, "A software architecture for next generation routers," in Proc. of ACM Sigcomm '98, sept 1998.
[9] A. Wool, "Architecting the Lumeta firewall analyzer," in Proceedings of the 10th USENIX Security Symposium, Washington, D.C., August 2001, pp. 85–97.

BIOGRAPHIES

Phaltane Anjali D was born in Kolhar B.K, Maharashtra, India .Completed Diploma in Computer Technology from Government Polytechnic Ahmednagar and pursuing B.E Computer Engineering from Pravara Rural Engineering College, Loni.

Prof .Jondhale S.D completed B.E Computer Engineering from Pravara Rural Engineering College, Loni.