ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

BDD for Implementation of Packet Filter Firewall and Detecting Phishing Websites

Naresh Shende
Vidyalankar Institute of Technology

Prof. S. K. Shinde
Lokmanya Tilak College of Engineering

Abstract*—Packet filtering using BDD is the one of the major contemporary firewall design techniques. An important design goal is to arrive at the decision at the packet only. Implementation of such packet filter using Binary Decision Diagram (BDD) gives more advantages in terms of memory usage and look up time as compare to other packet filter firewall technique. List-based packet filter has to prove the best with rules promotion method, but considerably lacks its efficiency when compared to BDD-based approach. Our BDD-based design for packet filter firewall will take less storage space and less look-up time for accepting or rejecting the incoming packets. In this paper we also implement how to detect the phishing sites. CANTINA examines the content of a web page to determine whether it is legitimate or not, in contrast to other approaches that look at surface characteristics of a web page, for example the URL and its domain name. CANTINA makes use of the well-known TF-IDF (term frequency/inverse document frequency) algorithm used in information retrieval, and more specifically, the Robust Hyperlinks algorithm. They also show that we can use CANTINA in conjunction with heuristics used by other tools to reduce false positives (incorrectly labeling legitimate web sites as phishing), while lowering phish detection rates only slightly.

***Index Terms*—Domain name, Firewall, Packet, Phishing, URL.**

I. INTRODUCTION

Packet filtering is the one of the major contemporary firewall design techniques. An important design goal is to arrive at the decision at the packet only. Implementation of such packet filter using Binary Decision Diagram (BDD) gives more advantages in terms of memory usage and look up time [9].

In the case of the list-based packet filter firewall where rules are checked one by one for each incoming packet, the time taken to decide on a packet is proportional to the number of rules [3].

A firewall is a combination of hardware and software used to implement a security policy governing the flow of network traffic between two or more networks. In its simplest form a firewall acts as a security barrier to control traffic and manage connections between internal and external network hosts. The actual means by which this is accomplished varies widely, and ranges from packet filtering and proxy service to state full inspection methods. A more sophisticated firewall may hide the topology of the network it is employed to protect, as well as other information, including names and addresses of hosts within the network. The ability of a firewall to centrally administer network security can also be extended to log incoming and outgoing traffic to allow accountability of user actions and to trigger alerts when unauthorized activities occur.

Firewalls have proven to be useful in dealing with a large number of threats that originate from outside a network. They are becoming ubiquitous and indispensable to the operation of the network. The continuous growth of the Internet, coupled with the increasing sophistication of attacks, however, is placing further demands and complexity on firewalls design and management. Increased firewall complexity, undoubtedly, brings with it increased vulnerability and reduced availability of individual network services and applications.

II. MOTIVATION AND OBJECTIVES

The aim of this project is to find an internal representation for access lists capable of providing fast lookup with reasonable memory requirements. Since access lists are consulted frequently (that is, for each arriving packet) and modified infrequently in comparison, the ability to perform fast lookups is the most important factor. This justifies using a potentially large amount of effort initially to create an efficient internal representation. The reasoning behind why binary decision diagrams (BDDs) may be a good choice for the internal representation follows.

Each rule of an access list describes the condition, based on the values of fields within the packet header that a packet must meet in order to have the corresponding action affected. Thus the condition of a rule is simply a logical, or Boolean, expression involving certain fields of the packet header to be accepted, a packet must satisfy this expression if the rule's action is accept, and not satisfy the expression if the rule's action is reject.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

Binary decision diagrams (BDDs) are a well-known data structure for storing and manipulating Boolean expressions compactly and efficiently, are therefore a natural choice for representing the access lists of a packet filter.

Other research has focused on the development of better user interfaces for anti-phishing tools. It helping users determine if they are interacting with a trusted site. Our work with CANTINA focuses on developing and evaluating a new heuristic based on TF/IDF (Term frequency/inverse document frequency), a popular information retrieval algorithm [10].

The investigation into BDD representations of access lists includes an analysis of:

1. ***Lookup algorithm complexity*:** Associated with the BDD representation of access lists is the lookup algorithm which uses the BDD to make filtering decisions about incoming packets. Best, worst and average case performance is considered, as well as the relationships between them.
2. ***Memory usage*:** The memory requirements of BDD representations of access lists are analyzed. BDDs cannot guarantee good space complexity for arbitrary Boolean expressions in the worst case BDD sizes can grow exponentially with the number of Boolean variables in the expression.
3. ***Improve the operational cost***: We describe a traffic-aware optimization framework to improve the operational cost of firewalls. Based on this framework, we design a set of tools that inspect and analyze both multidimensional firewall rules and traffic logs and construct the optimal equivalent firewall rules based on the observed traffic characteristics.

III. BINARY DECISION DIAGRAM

A Binary Decision Diagram (BDD) is a data structure that is used to represent a Boolean function. Operations on BDDs are performed on the compressed representation of these functions. Figure 1 BDD is a rooted directed acyclic graph and consists of one or many decision nodes and two terminal nodes and edges which connect these nodes [10].

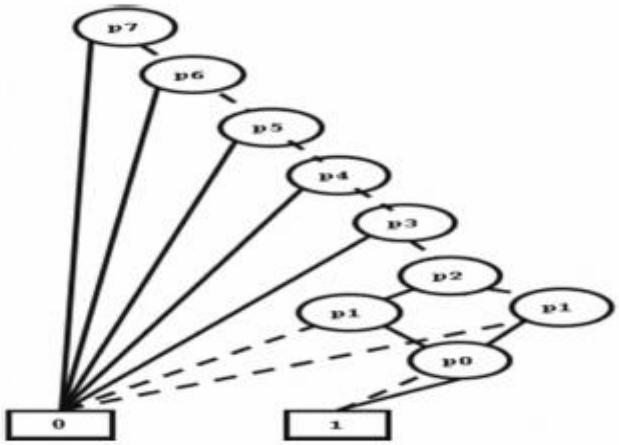

Fig 1 Example of BDD

A. USE OF BDD IN PACKET FILTER FIREWALL

Each packet contains header information like protocol, source address, source port, and destination address and destination port. All these are represented using numbers and can be converted in to the equivalent binary format. Suppose, the protocol is TCP and its number is 6, which can be converted to binary as 00000110. For example purpose we give the sample access list in the following table.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

id	sourceip	sourceport	destip	destport	protocol	action
1	01100101000000001010100011000000	0100010100000000	10001010111010110001001001111100	0000000001111110	00010001	Reject
2	01101001000000001010100011000000	0000000000110100	00001111001001001100001010101101	0100010100000000	00000110	Reject
3	11111111000000001010100011000000	0100010100000000	01110010000000001010100011000000	0000000001100000	00010001	Reject
4	11111111000000001010100011000000	0100010100000000	01110010000000001010100011000000	0000000001011010	00010001	Reject
5	11111111000000001010100011000000	0100010100000000	01110010000000001010100011000000	0000000001011010	00010001	Accept
6	11111010111111111111111111101111	0100010100000000	01100111000000001010100011000000	0000000101000000	00010001	Reject
7	11111111000011111010100011000000	0100010100000000	00010011000011111010100011000000	0000000011101110	00010001	Reject

Table 1 Sample access list containing rules

B. DESIGN AND IMPLEMENTATION OF ALGORITHM

In this section we are going to discuss about the implementation details and what are the modules implemented for the successful completion of the packet filter firewall project.

1. Development of a Module to generate binary File

We have implemented a module which automatically takes the access list as input and gives the corresponding binary file as output.

Input: Rule List
Output: containing binary form of the rule list.
1. Take each rule from the rule list.
2. Extract the protocol number, source address and port, and destination address and port.
3. Convert each part in to its binary format in required number of bits.
4. Store the each part of the converted binary form in to the file.

2. Development of a Packet Filter

In this module packet filter firewall takes the binary file as the input file and decides the action of the incoming and outgoing packets.

2. 1 Algorithm: BDD Lookup algorithm

Input: binary form file.
Output: ACCEPT or REJECT the packet.

```
if (solid) then
        final_op = 1
else
        final_op = 0
lookup_ptr=first node of the BDD
while(lookup_ptr == 1 OR lookup_ptr == 0)
if (header_bits[lookup_ptr] == 1)
        lookup_ptr = high (lookup_ptr)
else
        lookup_ptr = low (lookup_ptr)
if (lookup_ptr ==1) then
        ACCEPT the packet
else
        REJECT the packet
```

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

This algorithm searches the BDD generated according to the rule set for which the BDD is given. There after it takes the each bit of the header and compares with the BDD. Once all the bits in the header are checked by traversing the BDD from root to leaf node the decision of packet's acceptability is considered. Normally the packet filter information is stored in a 2-dimensional vector, which contains the information all BDD nodes and their high and low edge values.

3. Algorithm of list-based packet filter with promotion.

```
Extract the incoming packet header information.
While access list is not empty final result = 1
do
        if action = PERMIT then
                ACCEPT the packet
                rule hit count[rule no]++
        else if action = DENY then
                REJECT the packet
                rule hit count[rule no]++
        else
                rule action is not defined
        end if
end while
if access list is empty final result = 0 then
        REJECT the packet
end if
//In the next iteration alter access list according to the rule hit count.
```

Algorithm 3 finds which rule is hit in the current iteration and it increments the number of counts of the corresponding rule. Likewise, we need to observe the large chunk of traffic data coming daily or weekly and find the correct order of the access list based on the promotion made on the rules. After all the packets are passed through the access list the algorithm sorts the list based on the descending order of the hit count of each rule. Thus list promotion helps the firewall to decide early on packets accept/reject decision as these are mostly hit by the most active rules present on top of the new list.

IV. A CONTENT BASED APPROACH FOR DETECTING PHISHING WEBSITES

CANTINA makes use of TF-IDF for detecting phishing sites. TF/IDF is a well-known information retrieval algorithm that can be used for comparing and classifying documents, as well as retrieving documents from a large corpus. In this algorithm, we describe how adapted Robust Hyperlinks for detecting phishing web sites. CANTINA works as follows:

- Given a web page, calculate the TF-IDF scores of each term on that web page.
- Generate a lexical signature by taking the five terms with highest TF-IDF weights.
- Feed this lexical signature to a search engine, which in our case is Google.
- If the domain name of the current web page matches the domain name of the N top search results, consider it to be a legitimate web site. Otherwise, we consider it a phishing site.

We weighted these heuristics.

Our heuristics included in Table 2:

Heuristic	**Suspected Phishing?**
Age of Domain	<= 12 months
Known Images	Page contains any known logos and not on
a domain	owned by logo owner
Suspicious URL	URL contains @ or -

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

Suspicious Links	Link on page contains @ or -
IP Address	URL contains IP address
Dots in URL	>= 5 dots in URL

Table 2 Heuristics used to reduce false positives

A. ALGORITHM BASIC TF- IDF+DOMAIN+ZMP

It is combination of the TF-IDF is calculate lexical signature based on the top 5 terms, submit that to Google, and check if the domain name of the page in question matches any of the top 30 results. ZMP is zero search results means that the page in question is labeled as a phishing site (ZMP is "zero means phishing") variants above. This combination turned out to have the best results, and is also called **Final-TF-IDF.**

We also presented an evaluation of CANTINA, showing that the pure TF-IDF approach can catch about 97% phishing sites with about 6% false positives, and after combining some simple heuristics we are able to catch about 90% of phishing sites with only 1% false positives.

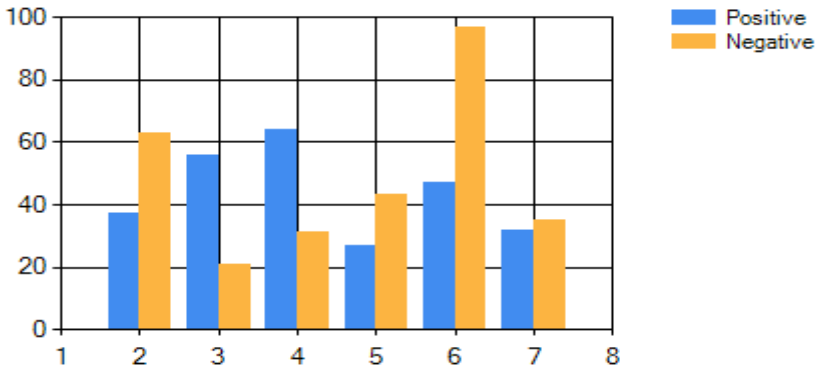

Fig 2. Comparison of Basic TF-IDF+domain+zmp algorithm with varying search results

V. CONCLUSION

In the last, conclusion of the project is to represent the access list in the form of BDD and implementation of such BDD-based packet filter firewall. List-based packet filter was proven to be the best with rules promotion method, but considerably lacks its efficiency when compared to BDD-based approach. Our BDD-based design for packet filter firewall takes less space for storage and less look-up time for accept or reject the incoming packets.

It is the first effort in using firewall traffic log information to design and optimize firewall rules sets. Both rule set based and traffic based optimizations are integrated in our firewall accelerating tool. The paper also introduces a novel adaptive anomaly detection/countermeasure mechanism to deal with short term and long term anomalies. We have started our efforts to validate the size and cost metrics and the optimization results. Furthermore, we are working on an efficient implementation for the algorithms to reduce the processing overheads of optimizations in the existing prototype.

We also presented the design and evaluation of CANTINA, a novel content-based approach for detecting phishing web sites. CANTINA takes Robust Hyperlinks, an idea for overcoming page not found problems using the well-known Term Frequency / Inverse Document Frequency (TF-IDF) algorithm, and applies it to anti-phishing. We described our implementation of CANTINA, and discussed some simple heuristics that can be applied to reduce false positives. We believe this paper is the first step in the design of a complete accelerating toolkit for firewall optimization.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

REFERENCES

[1] K. Ingham and S. Forrest, "A History and Survey of Network Firewalls", Technical Report 2002-37, University of New Mexico Computer Science Department, 2002.

[2] C. Shen, T. Chung, Y. Chang and Y. Chen, "PFC: A New High Performance Packet Filter Architecture", Journal of Internet Technology, Vo1.8, No.1, Page (s): 67-74, 2007.

[3] S. Acharya, J. Wang, Z. Ge, T. Znati and A. Greenberg, 'Traffic Aware Firewall Optimization Strategies", Proceedings of ICC, Page (s): 2225-30, 2006.

[4] M. Christiansen and E. Fleury, "An MTIDD Based Firewall", Telecommunication Systems 27:2-4, Page(s): 297-319, 2004.

[5] E. W. Fulp and S. J. Tarsa, "Trie-Based Policy Representations for Network Firewalls", Proceedings of ISCC, Page(s): 434-441, 2005.

[6] S. B. Akers, "Binary decision diagrams", Transaction on Computers, Vol. C-27(6), Page(s): 509-516, 1978.

[7] R. E. Bryant, "Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams", ACM Computing Surveys, Vol. 24, No.3, Page(s): 293-318, 1992.

[8] S. Hazelhurst, A. Fatti and A. Henwood, "Binary Decision Diagram Representations of Firewall and Router Access Lists", jtp:/(ftp.cs. wits.ac.zalpublresearchlreportsITR-Wi, 1998.

[9] G Paul, A Pothnal, C Mandal, B B Bhattacharya “Design and Implementation of Packet Filter Firewall using Binary Decision Diagram” Proceeding of the 2011 IEEE Students' Technology Symposium14-16 January, 2011, lIT Kharagpur

[10] Yeu Zhang, Jason Hong and Lorrie Cranor “CANTINA: A Content-Based Approach to Detecting Phishing Web Sites.