Firewall Designing Using FCD

Tanveer Ahmed
University School Of Information Technology
GGSIPU, DWARKA, NEW DELHI

Abstract:
Firewall today stand as a barrier between the internals of an organizations and the external network. The network is overwhelmed with a lot of malicious content threatening to compromise the secure information of an organization. In the paper we try to present a solution to design a firewall based on a graphical approach and making use of experiences learned from traversing the path of such a graph which is called FCD. FCD in itself equipped with several parameters which can help the network administrator to monitor the extent up to which a packet is able to travel in our layered firewall. Finally, we propose a way how past experiences gained in the filtering of packets can be utilized and prove useful in determining while selecting a particular module of such a program, thereby allowing more efficient use of the resources available

Keywords: Firewall, Network Security, Graph.

1. Introduction
A firewall is a device or set of devices designed to permit or deny network transmissions based upon a set of rules and is frequently used to protect networks from unauthorized access while permitting legitimate communications to pass. A firewall is one of the most crucial part of the organization whether private or government. It is placed at the entry point and filters out the data based on packet contents. A firewall can operate in at most 3 layers of the TCP/IP model.

- Circuit-Level Gateway
  Circuit level gateways work at the session layer of the OSI model and monitor TCP handshaking between packets to determine whether a requested session is legitimate.

- Network layer and packet filters
  These firewalls operate at a relatively low level of the TCP/IP protocol stack, not allowing packets to pass through the firewall unless they match the established rule. These firewalls are of the following two types:
  1. ***State full:*** These firewalls maintain context about active sessions, and use that "state information" to speed packet processing. E.g. For a particular type of connection maintenance of a particular TCP session.
  2. ***Stateless:*** Stateless firewalls require less memory, and can be faster for simple filters that require less time to filter than to look up a session

- Application-layer filter

  Application-layer firewalls work on the application level of the TCP/IP stack and intercept all packets traveling to or from an application.

The firewall at the top most layer i.e. application layer deals with a lot of traffic. Such traffic contain huge amount of information whether authentic or unauthentic, which sometimes can become real tricky to detect, so this part of the system is the one where the real risk lies[3]. The content of such illegal traffic can therefore pass through a loophole and can cause serious damage. To prevent the intrusion more and more organizations are going for the option of network-neutrality[10], though the technique only limits the ability of a user to receive a particular type of data, it does not allow the attacker to attack the system in context . So what we need is, to be absolutely structured in designing and the way the security requirements are implemented in simple words, how illegal threats are filtered out as firewalls designed without any structure have been found to have crashed when tested beyond the limits and suffers from configuration errors[2].

2. Related Work
The methodology used in the paper is an extension to FDD[1]. The algorithm and techniques are already been discussed in [5].The language for the implementation uses HLFL[6] otherwise vendor specific languages can cause slight variations in implementation. In the paper we have tried to keep the classification rules very simple, though the classification of packets based on rules are discussed in detail in [11].

3. Firewall Construction
The firewall architecture can be either a single layer or multilayer [3].The steps required to construct in construction of any firewall as discussed in the paper can be summarized as follows:

- Security requirements definition.
- Defining the proper course of action to be taken
- Defining criteria and assigning weights to each requirement as per policy.
- Construction using the Firewall Construction Diagram in bottom-up manner.
-

The goal here is to avoid the use of unnecessary or redundant rules. The HLFL[6] like FLIP[7] can be deployed for real implementation.

4. Firewall Construction Diagram

The security of any company is based on certain rules which are programmatically implemented using various techniques (Iptables[8], Swarm techniques[9]). Any internal or external access to the system within the corporation is filtered out using a sequence of queries as discussed by [4].

The following code snippet shows how policies and security requirements can be implemented using queries

```
 Struct access {
Src_addr inetaddr
Dest_addr inetaddr;
Struct host_group *src;
Movement boolean;
}

Movement

0-IN
1-OUT
```

The format of a query to such rule list is

```
Struct{
Struct host_group *src;
Ation_taken boolean;
Struct authority *service;
}

ACTION
0-DISCARD
1-ACCEPT

Struct host_group{
Struct list *list_of_ipaddress;
}
```

Authority means what are the capabilities assigned to a particular group.
Host_group defines the group to which a particular user belongs. This inturn can be used as one of the criteria to have confidence in the user and allow him to by pass the layers of the firewall.

Example of the above could be- "which group of computer are allowed to access the database xyz stored at the location abc in the city 123".

Mathematically, a firewall is set of rules which are defined as:

<Action>→<Decision>

where a rule is defined as

$\mathbf{R(i)=a(1) \cup a(2) \cup a(3) \cup a(4) \cup a(5)\ldots\ldots\ldots\ldots a(n)}$

$R(i) = i^{th}$ rule

$A(i) = i^{th}$ action

Also, decision $\in$ {accept, discard}

An FCD is a acyclic and directed graph shown below has the following characteristics

1. Each edge in the graph is assigned weight according to the cost of invoking a particular action.
2. Each node in the except the root node is given an additional child node representing the action discard. The main idea to use the decision (discard) with each and every node is to identify the problem as soon as possible and not allow the firewall engine to move further down the tree, thereby reducing the running complexity of the program.
3. Here each edge represented by the set ***a()*** represents the non empty subset of the set ***Filter()***.

$\text{Filter()} = \{\, a(1) \cup a(2) \cup a(3) \cup a(4)\ldots\ldots\ldots\ldots a(n)\ \}$

In general terms
a() represent the action taken in the context of a particular security policy of the organization.
Filter() is the set of actions taken to ensure the security policy of the organization.

4. The nodes here have some computational capabilities which work under supervision of root to provide the filtering of packets.

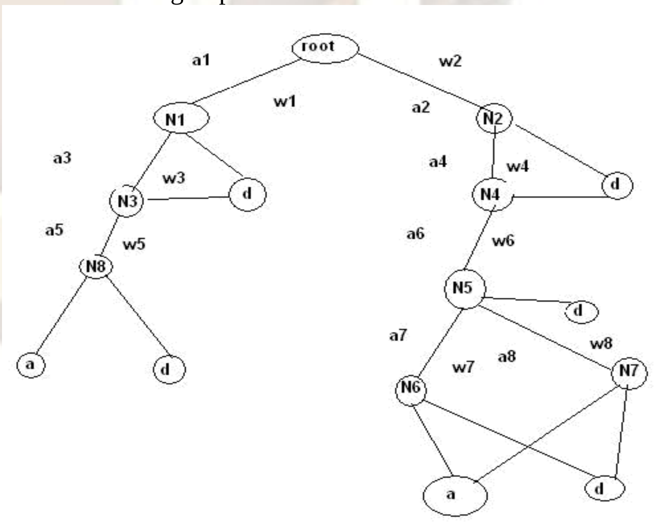


Fig 1.0 FCD

A decision path starts from the root and end at the terminal node representing the decision.

The rule which form the basis for filtering is represented by a particular decision path.
R(1)= root,a1,N1,d
R(2)=root,a1,N1,a3,N3,d
R(i)=root,a1,N1,……..a(n),N(n),a/d

The root here represents the topmost computational node as we move further down the tree the complexity of the filtering mechanism increases.

Example

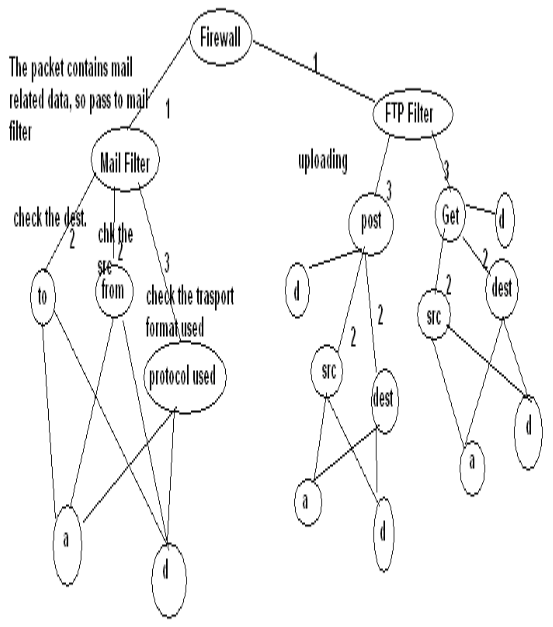


Fig 1.1 FCD demonstrating Filter isolation

The design of such firewalls allows multiple layers to be accommodated for the same filtering module thus allowing more control over determination of breach at any level.

P() is the set of all the packets entered into the system via the application gateway.

A packet p[d] travels a complete decision path to find the whether it is accepted or not.

For a packet P(i), the Severity of Penetration is defined as the sum of all the weights of the edges of the decision path.

$$\boldsymbol{SoP} = \sum_{i=1}^{k} \boldsymbol{w(i)}$$

Where w(i) represents weight of the ith edge and k is the total no. of such edges.

The datagram which have a very high value of SoP requires immediate attention of the network administrator. If a packet goes to the last node and is ultimately discarded then it shows how much the attacker/unauthorized user is familiar with our system, the depth up to which he can intrude and the likelihood of finding a loophole.

SoP $\propto$ Resources used/wasted

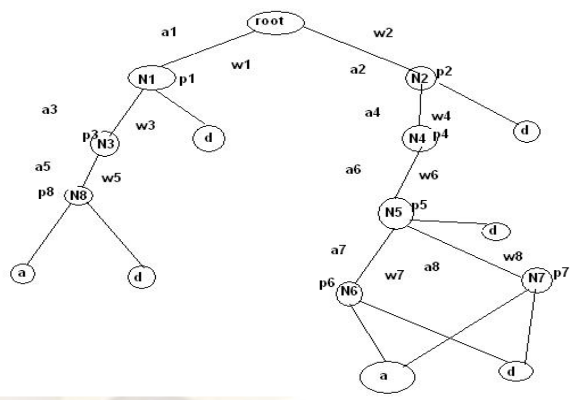
The designing of the FCD based firewalls can be extended with the fundamental concept of ACO[12], though in this case the positive feedback is obtained when a particular packet has been dropped. We can use additional information in FCD that will allow us to show confidence in selecting a particular path. That additional information is called Confidence of Traversal (CoT). CoT is usefull in deciding which filter to select while the packets are coming from a single host and within a single TCP session.

The following diagram shows and extension to the Fig 1.0 and an addition of CoT values to each node.

Where Confidence of Traversal is defined as

$$\boldsymbol{CoTn[i]} = \frac{\textbf{Packets passed through Node[i]}}{\textbf{Total no. of incoming Packets}} + \boldsymbol{CoTn\text{-}1[i]}$$

Here, CoT[i] denotes the Confidence of Traversal of node i and n denotes the no. of times a particular decision path has been traversed



Note:

1. CoT value must be stored for the particular TCP session in context; it will vary from session to session.
2. As soon as the root decides which filter to select, then only start with calculating the value of CoT.
3. The above equation shows the confidence in selection of a particular path based on heuristics learning rules i.e. it takes into account the previous confidence values for the packets.
4. Initially each non traversed node is assigned the CoT value of zero.

5. As soon as a particular TCP session is finished the default values must be restored.
6.p(i) stands for the CoT value for node i.

In the light of the above two definitions (SoP, CoT) we can deduce the following observation

- Two firewalls f and f ` are said to be equivalent iff both have the same no. of nodes and the corresponding values of SoP for a decision path and CoT for the nodes for the same given TCP session in a decision path is the same.

5. Conclusion

The techniques discussed in the paper can be directly applied to the implementation of any firewall, utilized to monitor any unauthorized access to the resources inside the organization, decide in optimum time period which path is the best of a packet and so on.
The FCD can also be extended and implemented as a neural net under supervised learning i.e. the experience gained under the supervision of the previous iteration.
The design approach can also recognize the presence redundant rules and remove them early in the designing module. The techniques used in the paper are not limited to the construction of firewall for stationary nodes, it can also be applied to the construction of the firewalls for mobile devices (PDAs).

6. Reference

1. Structured Firewall Design, Mohamed Gouda, Alex X. Xiu, pp 5-10, 2004
2. A quantitative study of firewall, Saurabh Jain, pp 9-11, 2004.
3. Firewall Design Principles, Dr. Krishnan, pp 3-5, 2004.
4. Algorithms for improving the dependability of firewall and filter rule lists, S. Hazelhurst, A. Attar, R. Sinnappan, 2000
5. A Firewall Analysis Engine, A. Mayer, A. Wool, E. Fang , 2000
6. High Level Firewall Language http://www.hlfl.org/
7. Specifications of A High-level Conflict-Free Firewall Policy
Language for Multi-domain Networks, Bin Zhang, Ehab Al-Shaer, Radha Jagadeesan, James Riely, Corin Pitcher, 2007.
8. IPtables http://www.netfilter.org/
9.*wikipedia.org/wiki/**Swarm**_intelligence*
10. A survey on network neutrality: a new form of discrimination based on network profiling, Khaled Deeb, Sean P. O'Brien Sr., Matthew E. Weiner, *2009*
11. Algorithm for packet classification, Pankaj Gupta, Nick McKeown , 20-32, 2001.
12 .http://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms.