Network Access Control using Software Based Firewall System

Pavan Poudel
Computer and Electronics Engineering
Kantipur Engineering College
TU, Nepal
poudelpavan@kec.edu.np

***Abstract*—When an internal network is connected to an external network such as Internet, it may suffer from external unauthorized access due to the openness of networks. It is possible for hackers to get access to an internal network, this pose great danger to the network resources and information security. In this paper, software based firewall system was proposed to ensure the integrity and confidentiality of information on the network. The proposed software based firewall system has the ability to determine which network traffic should be allowed in or out of the network. For this, the windows filtering platform (WFP) has been used with which any filtering rule can be added for the access control. The filtering rules can be applied based on the IP address, port number or application layer protocol types. The software based firewall system goes a long way in protecting an internal network from external unauthorized traffic penetration. Moreover, the implementation of the software based firewall system is easy.**

Keywords— Internet Security; Software Based Firewall; Computer Network Security; Windows Filtering Platform.

I. INTRODUCTION

The Internet is a network of computer networks [1]. It has evolved from the interconnection of networks around the globe. Interconnection is a good thing; it allows the free exchange of information via the Web, e-mail and file transfer. But it also carries a risk that your Internet connection may be used by "hackers" or "crackers" to gain unauthorized access to your local network. Availability of computing facilities can also be targeted by Denial of Service (DoS) attacks.

With the increase reliance on computer network, focus should also be given to monitor the traffic in and out of the system network. With the movement of data or information in and out of networks that has given birth to network security threat. The worst situation may arise when the internal computer network is connected to the Internet. Because of the Internet's openness, every corporate network connected to it is vulnerable to attack. Hackers on the internet could break into the network and do harm in a number of ways; they can steal or damage important data, damage individuals computer or their entire network, and use the internal network computer resources [2]. Due to some of these security threats, there was the need to build a defensive mechanism that ensures that hackers and their likes are not allowed into the network. Sets of rules are applied to control the type of networking traffic flowing in and out of the system. Firewalls are designed to stop unwanted or suspected traffics from flowing into the internal network.

Firewalls are crucial elements in network security, and have been widely deployed in most businesses and institutions for securing private networks. The function of a firewall is to examine each incoming and outgoing packet and decide whether to accept or to discard the packet based on its policy [3]. However, managing firewall rules, especially for enterprise networks, has become complex and error-prone. Firewall filtering rules have to be carefully written and organized in order to correctly implement the security policy. The firewall policy orders how the firewall should handle network traffic for specific IP addresses and address ranges, protocols, applications and content types based on the organisation's information security policies [5]. In addition, inserting or modifying a filtering rule requires thorough analysis of the relationship between this rule and other rules in order to determine the proper order of this rule and commit the updates [6].

II. LITERATURE REVIEW

The packet filtering systems route packets between internal and external hosts, but they do it relationally. They allow or block certain types of packets in a way that reflects a site's own security. Every packet has a set of headers containing certain information. This information is highly essential to the router and it includes; IP source address, IP destination address, Protocol (whether the packet is a TCP, UDP, or ICMP packet, TCP or UDP source port, TCP or UDP destination port, and ICMP message type and etc) [2].

The attempts of malicious access and attacks from the Internet to the internal computers of organizations never stop today and corresponding countermeasure for each technique is required. Most organizations introduce some firewall facilities as one of the solutions to protect their internal computers as well networks from those attacks [4]. Most of the existing firewall systems are implemented on hardware, that is, they are hardware based. Because of the hardware platform, the firewalls have the following shortcomings: they are very expensive; being hardware based, most of the firewall requires extensive configuration procedure; Network administrators are specially trained to handle the firewall system; each vendor has specific configuration procedures for their firewall systems. The implication of this is that the knowledge in one firewall system may not be applicable in another system; most of the hardware based firewall system cannot be upgraded. The limitations of the hardware based firewall are reasons for our adoption and the implementation of software based approach to firewall development.

By providing a simpler development platform, WFP is designed to replace previous packet filtering technologies such as Transport Driver Interface (TDI) filters, Network Driver Interface Specification (NDIS) filters, and Winsock Layered Service Providers (LSP) [7]. Starting in Windows Server 2008 and Windows Vista, the firewall hook and the filter hook drivers are not available; applications that were using these drivers should use WFP instead.

III. OVERVIEW

A. Windows Filtering Platform

Windows Filtering Platform (WFP) is a new architecture in starting from Windows Vista and Windows Server 2008 that enables independent us to filter and modify TCP/IP packets, monitor or authorize connections, filter Internet Protocol security (IPsec)-protected traffic, and filter remote procedure calls (RPCs) [7]. Filtering and modifying TCP/IP packets provides unprecedented access to the TCP/IP packet processing path where one can examine or modify outgoing and incoming packets before additional processing occurs. One can more easily create firewalls, antivirus software, diagnostic software, and other types of applications and services by accessing the TCP/IP processing path at different layers.

WFP provides APIs (Application Programming Interfaces) so that one can participate in the filtering decisions that occur at several layers in the TCP/IP protocol stack. WFP also integrates and provides support for next-generation firewall features such as authenticated communication and dynamic firewall configuration that is based on an application's use of the Windows Sockets API. This capability is also known as an application-based policy.

WFP provides higher performance, less programming complexity, and built-in diagnostic support. Additionally, one can use the built-in filtering engine for both Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) traffic. As WFP provides a strong security framework, correctly configured filters cannot be bypassed.

1. Purpose

Windows Filtering Platform (WFP) is a set of API and system services that provide a platform for creating network filtering applications. With the help of WFP API, developers can write code to interact with the packet processing that takes place at several layers in the networking stack of the operating system so that network data can be filtered and also modified before it reaches its destination [8]. WFP API provides developers to implement firewalls, intrusion detection systems, antivirus programs, network monitoring tools, and parental controls. WFP also provides infrastructure for IPsec policy management, change notifications, network diagnostics, and stateful filtering.

Windows Filtering Platform is a development platform and not a firewall itself. Applications developed with the WFP API use the common filtering arbitration logic that is built into WFP.

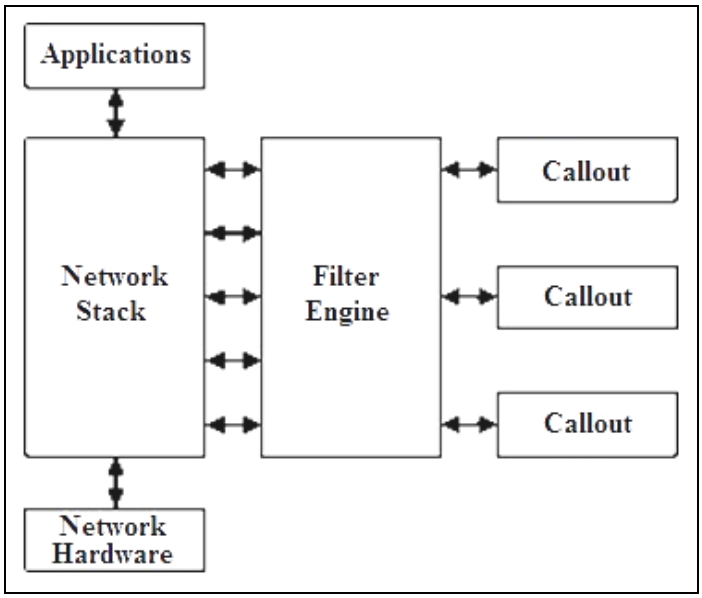

Fig. 1. WFP Architecture

The WFP API consists of a user-mode API and a kernel-mode API. We are using only the user-mode portion of the WFP API.

2. WFP Architecture

The WFP architecture as shown in fig.1 consists of the following components:

a) Filter Engine:

The filter engine contains a user-mode component and a kernel-mode component, which together perform all of the filtering operations on network data. The filter engine contains multiple filtering layers that map loosely to the operating system's networking stack layers. The user-mode component performs RPC and IPsec filtering. The kernel-mode component performs filtering at the network and transport layers of the TC/IP stack.

Base Filtering Engine Service (BFE) is a service that controls the operation of the Windows Filtering Platform [9]. The Base Filtering Engine (BFE) is a user-mode service (bfe.dll running in a svchost.exe process) that coordinates the WFP components. The principal tasks performed by BFE are adding and removing filters from the system, storing filter configuration, and enforcing WFP configuration security. Applications communicate with BFE through the WFP management functions (defined in WFP API reference).

b) Callout Drivers:

The callout drivers are used when initial filtering of the packets is not enough to determine whether they should be dropped, permitted or modified [10]. Callout drivers provide additional filtering functionality by adding custom callout functions to the filter engine at one or more of the kernel-mode filtering layers. Callouts support deep inspection and packet as well as stream modification.

After a callout driver has added its callout functions to the filter engine, filters that specify a given driver's callout function can be added to the filtering process. Such filters can be added by either a user-mode management application or by the callout driver itself. The kernel-mode interface, delivered in the Windows Development Kit, should only be used where needed and not as a substitute for the user-mode API.

c) Network Stack:

It is generally of OSI model. The newly created filter will be applied to different layers which we can choose to apply in the filter argument. Generally, we choose the Transport layer to implement filter.

3. WFP Operation

Windows Filtering Platform (WFP) performs its tasks by integrating the following basic entities: Layers, Filters, Shims, and Callouts. We Implemented Layers and Filters to implement our Firewall System.

a) Layers:

A layer is a container managed by the filter engine whose function is to organize filters into sets. A layer is not a module in the network stack. Each layer has a schema that defines the type of filters that can be added to it. Layers may contain sub-layers to manage conflicting filter requirements such as "Block TCP ports above 1024" and "Open port 1080". The rules for managing filtering conflicts are determined by Filter Arbitration.

Within a sub-layer, filter arbitration is performed as follows:

- Compute the list of matching filters ordered by weight from highest to lowest.
- Evaluate matching filters in order until a "Permit" or a "Block" is returned (filters can also return "Continue") or until the list is exhausted.
- Skip the remaining filters and return the action from the last evaluated filter.

WFP contains a set of built-in sub-layers. Every layer inherits all the built-in sub-layers. Users can also add their own sub-layers.

b) Filters:

A filter is a rule that is matched against incoming or outgoing packets. The rule tells the filtering engine what to do with the packet, including to call a callout module for deep packet or stream inspection. For example, a filter may specify "Block traffic with a TCP port greater than 1024" or "Call out to IDS for all traffic that is not secured."

A boot-time filter is a filter that is enforced at boot-time as soon as the TCP/IP stack driver (tcpip.sys) starts. A boot-time filter is disabled when BFE starts. A filter is marked as boot-time by setting the FWPM_FILTER_FLAG_BOOTTIME flag when FwpmFilterAdd0 is invoked.

A run-time filter is a filter that is enforced after BFE starts. A run-time filter can be static, dynamic, or persistent depending on the way it was created.

c) Classification:

Classification is the process of applying filters to network traffic (packet, stream, or event) in order to determine a result of "Permit" or "Block" for that traffic. For one packet, stream, or event there is one classification call per layer. During classification, the properties (for example, source address) of the packet, stream, or event are compared with filter conditions set on filters at the layer where the classification is invoked. When matches are found, the Filter Arbitration algorithm is used to determine the result of the classification process.

A classification request is triggered by a shim.Classification actions using filter could be either:

- Permit
- Block

At boot-time, as soon as the TCP/IP stack driver (tcpip.sys) starts, the kernel-mode filter engine enforces the security policy of the system through boot-time filters.

4. WFP APIs

These are some of the WFP APIs that we will be using to write our firewall:

1) FwpmEngineOpen0 - This API is used to create a session with the Windows packet filtering engine.
2) FwpmSubLayerAdd0 - This API adds a new sub-layer to the packet filtering engine.
3) FwpmFilterAdd0 - This API adds filters (rules) to a sub-layer.
4) FwpmFilterDeleteById0 - This API removes existing filters from a sub-layer.
5) FwpmSubLayerDeleteByKey0 - This API deletes the sub-layer which was added by FwpmSubLayerAdd0.
6) FwpmEngineClose0 - This API closes the session opened by FwpmEngineOpen0.

Here are the steps to write a firewall using the above mentioned APIs:

1) Create a session using FwpmEngineOpen0.
2) Add a sub-layer using FwpmSubLayerAdd0.
3) Now, add filters using FwpmFilterAdd0. If you have "n" filters, then this API needs to be called "n" times.

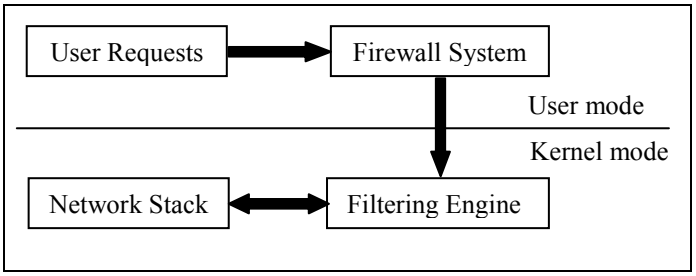

Fig. 2. Software Based Firewall System Architecture

IV. System Development

Fig. 2 shows the overall system architecture of software based firewall system. At first, user logins as an administrator. The Username and Passwords are stored in files. After the login, user requests to the firewall system for the services. The services include the IP address access control or the port number access control. As per the requirements, user can create new firewall rules or can edit or delete the existing firewall rules. The firewall rules created at the user mode will invoke the filtering engine at kernel mode and finally implemented through the network stack.

Firewall system works with windows filtering engine. The filtering engine is embedded with filter. Once the Base Filtering Engine (BFE) starts in user mode, persistent filters are added to the platform, boot-time filters are disabled. After BFE starts, run-time filters can be added by firewall agents, or by custom firewall solutions. BFE processes these filters and sends them to the appropriate filter engine layer for enforcement. BFE also accepts authentication settings and sends these settings to the IPsec keying modules (IKE/AuthIP).

At any time, filters and authentication settings can be added, removed or changed in the system through the RPC interface exposed by the BFE. Sub-layers and callout modules can likewise be added or removed.

Data flow:

Inbound data flow follows following pattern.

- A packet comes into the network stack.
- The network stack finds and calls a shim.
- The shim invokes the classification process at a particular layer.
- During classification, filters are matched and the resultant action is taken.
- If any callout filters are matched during the classification process, the corresponding callouts are invoked.
- The shim acts on the final filtering decision (for example, drop the packet).

Outbound data flow follows a similar pattern.

```
        Enter Details of Source

Source IP Address : 50.87.153.144
Subnet Mask       : 255.255.255.255

        Enter Details of Destination

Destination IP Address  : 192.168.1.1
Subnet Mask             : 255.255.255.255
```

Fig. 3. Adding inbound IP rule to the software based firewall system

```
How do you want to add Rule?

    1.By Port Number
    2.By Protocol Type

Choose Your Option: 1

Enter the Port Number   : 80
Enter the Local IP Address : 192.168.1.1
Enter the Subnet Mask   : 255.255.255.255
```

Fig. 4. Adding inbound Port rule to the software based firewall system

V. Result and Discussion

A. *Inbound ip rule*

This is an IP block rule applied for the remote system accessing inside IP address. For Inbound IP rule, following are required as shown in fig. 3:

- Source IP Address (Outside IP Address) and its Subnet Mask
- Destination IP Address (Inside IP Address) and its Subnet Mask

B. *Outbound ip rule*

This is an IP block rule applied for the local System to access Remote system. For the Outbound IP Rule, following are required:

- Source IP Address (Inside IP Address) and its Subnet Mask
- Destination IP Address (Outside IP Address) and its Subnet Mask

C. *Inbound port rule*

This rule is a Port block rule for the Remote IP Address to prevent it for using a local port. For the Inbound Port Rule, following are required as shown in fig. 4:

- Remote IP Address with Subnet Mask
- Local Port Number (OR Protocol Name)

```
How do you want to add Rule?

        1.By Port Number
        2.By Protocol Type

Choose Your Option: 2

Enter the Protocol Name: HTTPS
Enter the Remote IP Address: 202.166.193.8
Enter the Subnet Mask   : 255.255.255.255
```

Fig. 5. Adding outbound Port rule to the software based firewall system

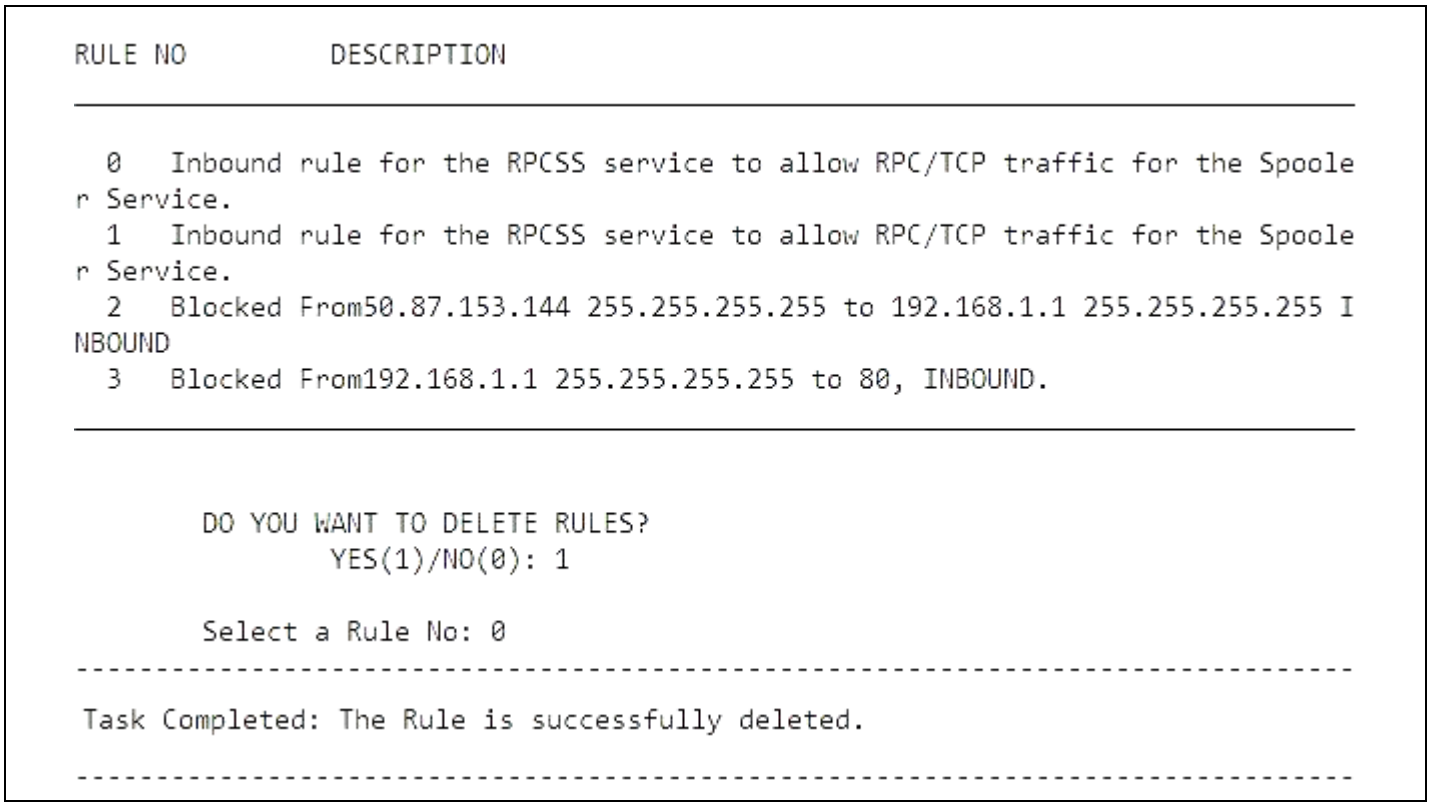

Fig. 6. View and edit rules to the software based firewall system

D. Outbound port rule

This rule is a Port block rule for the Local IP Address to prevent it from using a local port. For the Outbound Port Rule, following are required as shown in fig. 5:

- Local IP Address with Subnet Mask
- Local port Number (OR Protocol Name)

E. User access control

This rule is a part of our firewall system which includes that a particular user with a username is prevented to access a particular IP address and Port number.

This rule is an Outbound Rule such that the user cannot use a port of itself or it cannot access a remote IP Address.

For the User Access Control Rule, following are required:

- Username
- Remote IP Address (for blocking an IP Address for the User).
- Local Port Number (for blocking a Port for the User).

F. Apply default rules

Default rule is the set of rules such that all the IP block rules and Port block rules are cleared from the system and brings it at default condition. It does not require any parameters.

G. View and edit rules

This is another feature of the firewall system such that all the firewall rules applied to the system can be viewed and they can be edited or deleted individually as shown in fig. 6.

The proposed system can be used in any computer network for the security. The implementation is easy as the filtering rules are easy to create based on either IP address, or port number. Those IP addresses which are known to be infected and dangerous to the computer system can be added to the IP block list in the application. Similarly, the different ports can also be blocked and allowed as per user requirement to give access to perform the different data and information sharing through the internet. For example, port numbers 20 and 21 for File Transfer Protocol (FTP) can be added to the port block list to protect the file transfer operation. Moreover, the hierarchical security can be implemented using the proposed system. The proposed system has application in schools, colleges, cyber cafes, and other computer training centers where the network security is needed. Apart from its easy implementation, the proposed system is cost effective as well.

VI. CONCLUSION

Information security has become an important concept in any organizations due to the fact that an unprotected information system can be exposed to danger in a network as a result of penetration tools at the disposal of hackers and crackers. Therefore, there is need to ensure adequate protection of internal network from hackers. To achieve this, software based firewall system is one solution which is technically

feasible with the capabilities of controlling the access through IP addresses and Port addresses. This work focuses on the firewall system that filters what goes in and comes out of the network. It will have the ability to block an unauthorized traffic and allow authorized traffic using the filters defiend in the system using windows filtering platform.

ACKNOWLEDGMENT

The authors acknowledge the handful discussion with Mr. Ajay Kumar Shrestha.

REFERENCES

[1] Tannenbaum, Andrew S. Computer Networks. New Dehli: Dorling Kindersley Pvt Ltd, 2003.

[2] Preetham, V. V. Internet Security And Firewalls. Cincinnati, Ohio: Premier Press, 2002.

[3] Liu, A.X.; Gouda, M.G., "Firewall Policy Queries," in Parallel and Distributed Systems, IEEE Transactions on , vol.20, no.6, pp.766-777, June 2009

[4] Otsuka, Tomokazu; Gada; Yamai, Nariyoshi; Okayama, Kiyohiko; Jin, Yong, "Design and Implementation of Client IP Notification Feature on DNS for Proactive Firewall System," in Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual , vol.3, no., pp.127-132, 1-5 July 2015

[5] C. Poovinayaga Sastha and Dr. V. Palanisamy, 'A Simple Taxonomy Survey of Firewall Policies', International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 7, pp. 319-321, 2012.

[6] Al-Shaer, Ehab S., and Hazem H. Hamed. "Modeling and management of firewall policies." Network and Service Management, IEEE Transactions on 1.1 (2004): 2-10.

[7] Kresten, Proteus Valre. "Windows Filtering Platform." (2012).

[8] Msdn.microsoft.com, 'Windows Filtering Platform (Windows)', 2015. [Online]. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/aa366510(v=vs.85).aspx. [Accessed: 30- Sep- 2015].

[9] H. Gelfenbeyn, 'Base Filtering Engine (BFE) service is missing: why did it happen and how to restore it', Hageltech.com, 2015. [Online]. Available: http://www.hageltech.com/blog/2012/02/07/base-filtering-engine-problems.html. [Accessed: 30- Sep- 2015].

[10] Komodia.com, 'WFP - Windows filtering platform high level overview', 2015. [Online]. Available: http://www.komodia.com/wfp_hl. [Accessed: 30- Sep- 2015].