The Program Design of Network Firewall Based on Windows

Zhang Yu
College of Computer and Information Engineering
Harbin University of Commerce
Harbin, China
Zh y@hrbcu.edu.cn

***Abstract*—This thesis focuses on the design of the personal firewall and its realization under the operation system of Windows. Firstly, the paper presents the background of introducing of the personal firewall, the development of Domestic and foreign personal firewall, the security problem of network and the technology of computer firewall. Then, analyses thoroughly the technology of the capturing of network data under the operation system of Windows, and introduces overall frame about the operation system of Windows as well as structural drawing about the network system of Windows. Finally, the paper explains the design of the project and the implement of all the modules in detail, illuminates the testing and the analysis of performance of the program, and summarizes the entire paper. During the development, we adopt the software designing idea of structurization and modularization, which improves the transplantation and agility of the system. Three modules communicate each other, adopting the share memory technology, the message of Windows and the I/O controlling code.**

Keywords-network securit; personal firewall; software process improvement; layered service provider

I. INTRODUCTION

Firewall, network security products and the use of the largest security products increasingly attracted users and R & D institutions. From the application point of view, the firewall is basically can be divided into two types: enterprise-class firewall and personal firewall. As the Windows operating system is the most widely used PC operating system, developed under the Windows operating system, personal firewall are numerous. More well-known abroad are AtGuard, BlackICE PC Protection, ZoneAlarm, Tiny Personal Firewall, Norton Personal Firewall and Sygate Personal Firewall and so on, while domestic Rising Personal Firewall, Kingsoft Personal Firewall and other products.

All of these personal firewall based on Windows operating system, the difference is that Windows, network data packet interception technology. Used by the interception technology, personal firewall points: Based on the SPI (Service Provider Interface) personal firewall, based on TDI (Transport Drivers Interface) personal firewall, based on NDIS hook-driven personal firewall, based on NDIS Intermediate Driver Personal Firewall . Most personal firewalls combined use of 22 technologies, such as the famous Fairbanks Personal Firewall is based on SPI and the combination of NDIS hook driver technology.

SPI-based personal firewall: Using the benefits of this technology is to get more information call the process of Winsock, which can be used to achieve QoS (Quality of Service), data stream encryption or other purposes. However, this technique to intercept data packets under the most fatal shortcoming is that the only conduct at the Winsock level, if the application directly through the TDI calls TCP/IP to send data packets, this method can do nothing.

II. WINDOWS NETWORK PACKET INTERCEPTION TECHNOLOGY

As the personal firewall is running on the protected host's software, which is related to the host operating system and network protocols. This research project is based on the Windows operating system, personal firewall software; you must have a clear understanding of the structure for the Windows operating system and network protocols. Windows-based operating system, personal firewall, and the core technology lies on Windows operating system, network packet interception technology.

A. The Overall Structure of Windows Operating System

The overall structure of Windows operating system is divided into two levels, above the application layer, the following as the core layer, its structure as shown in Figure 1.

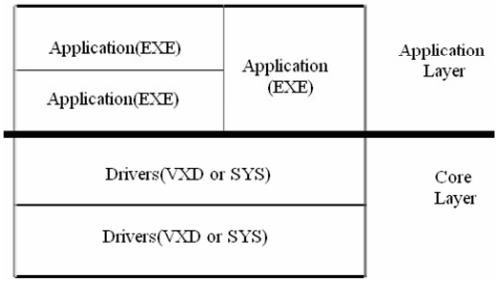

Figure 1. The structure of Windows operating system

The application (. Exe) work at the application layer, dynamic link libraries (. DLL) also belong to the scope of the application layer, the dynamic link library is called when an application to become a part of the application, so they are not essentially different from [9]. EXE and DLL are two ways of working in different applications, EXE is a stand-alone modules can be directly executed by the process of protection mechanisms WINDOWS protection, other programs no right to directly use this program modules and data. DLL is a shared library, which provides a standard interface for other programs call the network, it can not run.

There are at the application layer below the layer, called the core layer. In Windows95/98, the core layer of the program extension VXD, on Windows NT/2000/XP, the core layer process extension SYS, these programs are

978-0-7695-4009-2/10 $26.00 © 2010 IEEE
DOI 10.1109/MVHI.2010.116

called drivers, drivers, applications for the upper underlying support.

Such results can be achieved layered code-sharing. To protocol driver, for example, a system where there are many applications that use the same network protocol, the driver calls out a separate code sharing can be achieved. Like a DLL that can be all of the EXE calls, all procedures can be called with a protocol driver. In this way, the operating system can make agreements on the application, transparent and all applications do not care about protocol implementations, as long as the interface functions provided for in accordance with the appropriate action can be.

This hierarchical structure has the advantage of greater security can be achieved. Because, as protocol driver sort of procedures, effective implementation of the procedures and code of rigor, robustness requirements are very high. Once the program has problems, the system may be paralyzed, so the operating system divided into two levels, you can give them each a different operating privileges.

B. Windows Network System Architecture

To intercept network data under Windows operating system , we must have a clear understanding of Windows network systems, the next figure 2 shows the user mode and kernel-mode network system diagram under the Windows operating system[10].

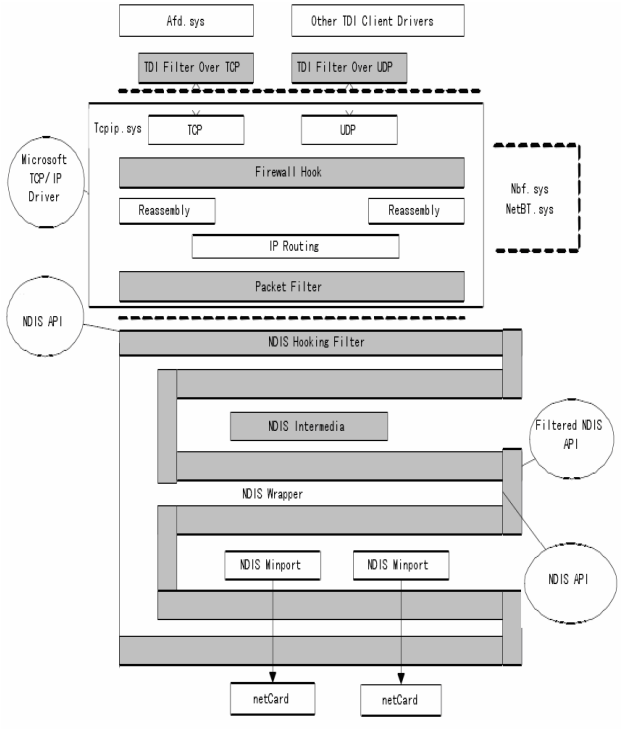

Figure 2. Network System Architecture of Windows kernel-mode

In general, you want to block the network data packets under Windows, you can carry out at two levels: user mode (USER-MODE), and kernel mode (KERNEL-MODE). In user mode, data packet filtering is simple and convenient; in kernel mode, data packet filtering is powerful but complex.

III. THE DETAILED DESIGN OF SYSTEM

A. Module Division

The whole process is divided into three parts: the main program, application-layer filtering module, the core layer of filter modules. Figure 3 shows the main features of the system structure chart.

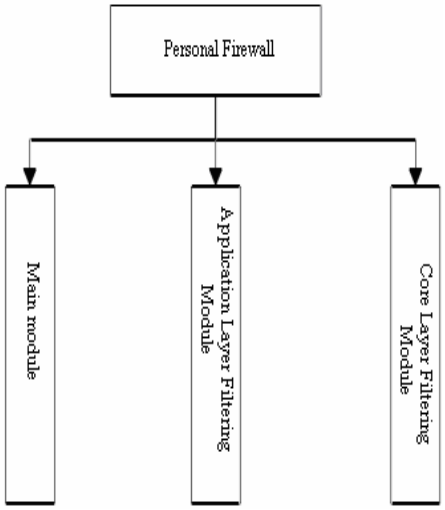

Figure 3. The main function structure chart

Windows message interface is a special interface, form, which is based on the Windows operating system message mechanism. This interface between the application form is suitable. If two modules need to communicate, then the message interface is an ideal choice.

PM_SESSION_NOTIFY message
Definition: # define
PM_SESSION_NOTIFYWM_USER +200
Parameter LPARAM: Session Structure Address
Parameters RPARAM: Operation Types
define CODE_CHANGE_SESSION 0 // session attribute changes in
define CODE_DELETE_SESSION 1 // Remove session
define CODE_APP_EXIT 2 // application exit

Role: In order to track the user-level network activity status, whenever a new session creation, session attributes to change the conversation destruction, DLL module will notify the main module by PM_SESSION_NOTIFY, using shared memory to pass an array of members of the relevant events took place the session.

PM_QUERY_ACL_NOTIFY message
Definition: # define
PM_QUERY_ACL_NOTIFYWM_USER +201
Parameter LPARAM: application-layer structure addresses the rules
Parameters RPARAM:

Role:When there is no current application access rules in the rules file, DLL module will notify the main module and asked whether the main module to add access rules by PM_QUERY_ACL_NOTIFY.

When there are multiple processes simultaneously call the same DLL, when, Win32 approach is not a common DLL's code and data segment, but rather a call for each application, a copy of the establishment of a DLL and this copy of the application for the private, such a mechanism

so that each independent of each other between applications are conducive to data protection.

There is a certain advantage of such a mechanism, but sometimes also compiled bring some trouble, when you need different processes to share data, share data variable method can be used. Keyword # pragma can be used to define the data variables.

The system of shared data is divided into two categories: one is the process of loading DLL when initialized, the data defined in .. initdata segment; a type of data have not been initialized, they are defined in. Uninitdata section.

# pragma data_seg (". initdata")
HWND g_hNetFilterWnd = NULL; // main window handle
UCHAR g_ucWorkMode = PF_PASS_ALL; // work modes
# pragma data_seg ()
# pragma bss_seg (". uinitdata")
RULE_ITEM g_Rule [MAX_RULE_COUNT]; // application-layer rules
ULONG g_RuleCount;
QUERY_SESSION g_QuerySession [MAX_QUERY_SESSION]; // to send session inquiry to the main program
SESSION g_SessionBuffer [MAX_SESSION_BUFFER]; // to send session information to the main program
TCHAR g_szPhoenixFW [MAX_PATH]; // Record the main program path
# pragma bss_seg ()
declare the properties of these two paragraphs.
SECTIONS
initdata READ WRITE SHARED
.uinitdata READ WRITE SHARED

In order to allow the main module to access a shared DLL data, the main module shared memory settings using exported functions, in which IO control mode is as follows:

1. IO_CONTROL_SET_WORK_MODE
Definition: # define IO_CONTROL_SET_WORK_MODE 0
Role: set work mode of user-level.
2. IO_CONTROL_GET_WORK_MODE
Definition: # define IO_CONTROL_GET_WORK_MODE 1
Role: obtain work mode of user-level.
3. IO_CONTROL_SET_PHOENIX_INSTANCE
Definition: # define IO_CONTROL_SET_PHOENIX_INSTANCE 2
Role: set the main module information.
4. IO_CONTROL_GET_SESSION
Definition: # define IO_CONTROL_GET_SESSION 3
Role: get a conversation.
5. IO_CONTROL_SET_RULE_FILE
Definition: # define IO_CONTROL_SET_RULE_FILE 6
Role: Set the application layer rules.

4. the main module and SYS module IOCTL

DeviceIoControl interface provides basic mechanisms for Win32 applications and SYS module, when adding filtering rules we have to add the appropriate IOCTL control code, the following is IOCTL control code involved in this system:

1. IOCTL_PTUSERIO_ENUMERATE
Definition: # define IOCTL_PTUSERIO_ENUMERATE \ CTL_CODE (FSCTL_PTUSERIO_BASE, \ 0x201, METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
Role: enumerate bound adapter.
2. IOCTL_PTUSERIO_OPEN_ADAPTER
Definition: # define IOCTL_PTUSERIO_OPEN_ADAPTER \ CTL_CODE (FSCTL_PTUSERIO_BASE, \ 0x202, METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
Role: open an adapter.
3. IOCTL_PTUSERIO_SET_OID
Definition: # define IOCTL_PTUSERIO_SET_OID \ CTL_CODE (FSCTL_PTUSERIO_BASE, \ 0x203, METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
Role: Set NDIS object identification information.
4. IOCTL_PTUSERIO_QUERY_OID
Definition: # define IOCTL_PTUSERIO_QUERY_OID \ CTL_CODE (FSCTL_PTUSERIO_BASE, \ 0x204, METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
Role: query NDIS object identification information.
5. IOCTL_PTUSERIO_QUERY_STATISTICS
Definition: # define IOCTL_PTUSERIO_QUERY_STATISTICS \ CTL_CODE (FSCTL_PTUSERIO_BASE, \ 0x205, METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
Role: access to network activity states
6. IOCTL_PTUSERIO_RESET_STATISTICS
Definition: # define IOCTL_PTUSERIO_RESET_STATISTICS \ CTL_CODE (FSCTL_PTUSERIO_BASE, \ 0x206, METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
Role: Reset the network active.
7. IOCTL_PTUSERIO_ADD_FILTER
Definition: # define IOCTL_PTUSERIO_ADD_FILTER \ CTL_CODE (FSCTL_PTUSERIO_BASE, \ 0x207, METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
Role: Add a filter rule.
8. IOCTL_PTUSERIO_CLEAR_FILTER
Definition: # define IOCTL_PTUSERIO_CLEAR_FILTER \ CTL_CODE (FSCTL_PTUSERIO_BASE, \ 0x208, METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
Role: remove filtering rules.

IV. FUNCTIONAL TESTING AND PERFORMANCE ANALYSIS

At this point a personal firewall system design implementation work has been completed, we will do testing and performance evaluation for NetDefender, the testing includes functional testing and performance testing and so on.

A. *Test Method*

In test environment, we build a local area network, the connection between computers use 100M switch, the test method is to set up an FTP server within a local area network, testing, installation NetDefender personal firewall, transmission 1G data files, calculate the open NetDefender Personal Firewall before and after the time required, while the firewall is turned on before and after contrast to other data transmission speed.

B. *Test Results*

By NetDefender personal firewall tests, in the following function has reached the design requirements. Figure 4 is a screenshot NetDefender running a personal firewall.

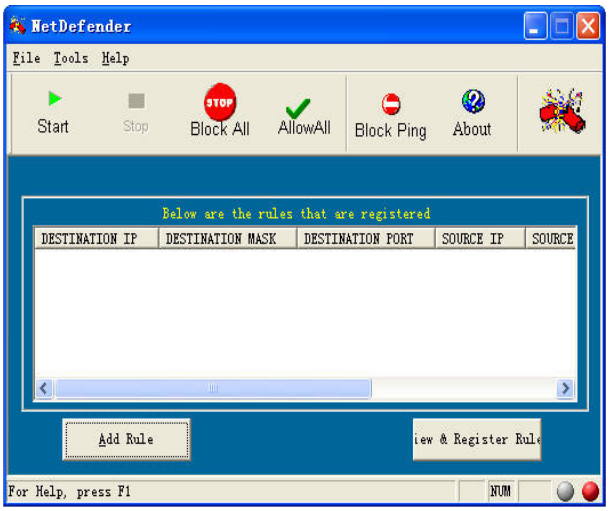

Figure 4. Screenshot of main program

In NetDefender personal firewall performance tests, in order to more clearly understand their performance, we have done a comparison in data transfer speeds before and after loading the firewall the network, but also, and has done a comparison with the more popular personal firewall on the network. The following test results are completed in the LAN, in order to reduce the impact of the test results of the incidental factor, and we tested five times, and then calculate the average. TABLE I details the impact data transfer before and after opening the personal firewall.

TABLE I. TEST RESULTS

Used firewall	Test Project	Pre-loaded firewall	After Loading firewall
NetDefender	FTP（1G Bytes）	90s	101s
Skynet Personal Firewall	FTP（1G Bytes）	90s	98s
Rising Personal Firewall	FTP（1G Bytes）	90s	99.5s

V. CONCLUSIONS

This system has basically achieved the functional requirements of a personal firewall, but due to time and level, the system still exist many deficiencies. its shortcomings are mainly the following points: First of all, only achieved user-level packet filtering in the Windows operating system , has not been achieved in the core layer packet filter. The user layer is the higher level of a system, which makes some applications to bypass the blocked TDI direct calls directly through TCP/IP to send and receive data packets, that is the interception of data packets is not very thorough. Second, only achieved a personal firewall features, intrusion detection feature did not achieve. Intrusion detection is a very complicated technology, if coupled with intrusion detection features, its overall function is bound to greatly enhanced. Again, can not control the Network Neighborhood shared resources and the use of ICMP protocol packet filter. The main reason is the higher level of work.

Through the above study and summary, we can find the following need improvement: First, using the kernel packet interception technology packets on the core level, a more thorough filtering, application layer and the core layer to achieve the dual filter. Second, combine intrusion detection technology and other network security technologies to enhance security capabilities. Again, the use of low-level interceptor technology control online resource sharing and ICMP neighbor of the ping.

Contest between Network attack and network defense will never stop, as firewalls and intrusion detection technologies improve network security, hackers, and viruses means of attack are also constantly refurbished, but the current control firewall rule set etc., or more professional, It is difficult to understand for general Internet users. Therefore, the future design of the firewall should be more humane and popularization, increasing ease of use of a firewall.

REFERENCES

[1] Jian-Wei Hu, Jian-Long Tang, Shao-Quan Yang, et al. Against the principles of the network. Xi'an University of Electronic Science and Technology Press, 2004,102

[2] Chu Kuang. Network security and firewall technology (first edition). Beijing: People's Posts & Telecom Press, 2000.4

[3] Zhu Yan-hui. Windows Firewall and network packet interception technology. Electronics Industry Publishing House ,2002,2-10

[4] Wang Ping, Zhang Yue. Windows networking and communications programming. Posts & Telecom Press ,2006,283-285 ,333-338

[5] Li Zhijun. Personal Firewall research and design and implementation. University of Electronic Science and Technology Graduate Master's thesis .2003,21-22

[6] W. Richard Stevens.TCP / IP Xiangjie volumes. Fan Jianhua translation. Mechanical Industry Publishing House ,2000,6-7

[7] W. Richard Stevens.TCP / IP Xiangjie volumes. Fan Jianhua translation. Mechanical Industry Publishing House ,2000,6-7

[8] Douglas E.Comer.TCP / IP for Internet connectivity Volume I: principles, protocols and structures (fourth edition). Linyao will Hui, Du Wei Xuan translation. Electronic Industry Press, 2001,3

[9] AndrewS.Tanenbaum. Computer Networks (third edition). Xiong Gui Xi, Xiao-Hu Wang Xuan translation. Tsinghua University Press, 2001,23

[10] Gang Ka Wei SiM, GoncalvesMarcus. Firewalls technical guidelines. Song Min book translation. Mechanical Industry Press, 2000.11

[11] Peter Norton, Mike Stockman. Xiaoxiang studio translation. Network security guidelines. Beijing: People's Posts & Telecom Press, 2000.