International Journal of Security and Its Applications
Vol.8, No.2 (2014), pp.251-260
http://dx.doi.org/10.14257/ijsia.2014.8.2.26

Based on WDM of Disk Immunity Systems

Peng Hai-Yun and Zhou Wen-Gang

Department of Computer Science; Zhoukou Normal University; Zhoukou 466000
Hangfan_2007@163.com

Abstract

It is a tough problem to pretect computer data in the public environment. In order to prevent hard disk data from virus infection, malice destory and deleting by mistake, this paper illustrates the design and implementation of a disk immunity system based WDM filtered driver. This system can effectively protect hard disk data. Benchmark results show that this system has a few effects for the disk I/O performance.

Keywords: *Configuration Module; Drivers; Isolated area*

1. Introduction

Along with the computer virus and rogue malware become more and more rampant increasingly. It's an important problem how to protect the computer disk data from destruction effectively. Especially in some public places, such as the Internet, electronic reading rooms, computer rooms which are infected and destructed by computer virus and malicious, the data error probability is greater. The less frequent reloading system, the easier workload of maintenance is, so a variety of data backup protection scheme emerge as the times require. Present in the public rooms of the most widely used is the disk hardware protection card and disk immune system. The two principles is consistent, it can restore the part or all of the partitions to the previous content when the system starts up, prior to any disk protection partition modify both failure, so as to achieve the purpose to protect the disk data. The two common design ideas is to take over the system on the disk read and write operations, such as creation, delete, modification and any of the disk write operations are being redirected to disk isolation region, after a system reboot redirection table empty, all data is restored to its original state, any malicious tampering, disoperation, virus damage will failure. The disk protection card interrupt in the hardware layer BIOS to take over responsibility for the disk read and write operations of INT13, and the disk immune software system is in the level of driver over the disk read and write operations, thus realizes the data protection. This paper focuses on a Windows operating system disk immune system design and implementation.

2. Disk Immune System Implementation Principles

Windows driver model WDM (Windows Driver Model) is introduced in the Microsoft device driver model [1] [1], aims to provide a flexible way to simplify the development of driver, on the basis of the realization of new hardware support reduce and drop the number and complexity of the need to develop drivers. WDM realizes a modular, hierarchical structure type of driver, Windows I/O subsystem is also a packet driver system. In the system, each of the I/O operation can pass an IRP (I/O Request Package) description; the working process of driver is the IRP process. Usually the IRP is first to sent to the device stack top driver, and then gradually be passed to the lower level driver. After each layer of the driver finish IRP packets, they are forwarded down until IRP packet processing done with.

ISSN: 1738-9976 IJSIA

In Windows environment the realization of disk immune principle is though passing WDM level filtering driver based on disk level [3], catch the disk I/O read and write IRP request packet, determine whether it aims at the write operation of disk protection partition. if let the request starting sector position at the isolation region position, at the same time establish an index lookup table, as the result of the indexed lookup table stored in the memory, after the machine restart, the index table initialize to empty automatically, read-in the protected area data is lost, so as to protect the data results, its realization principle be shown in Figure 1.

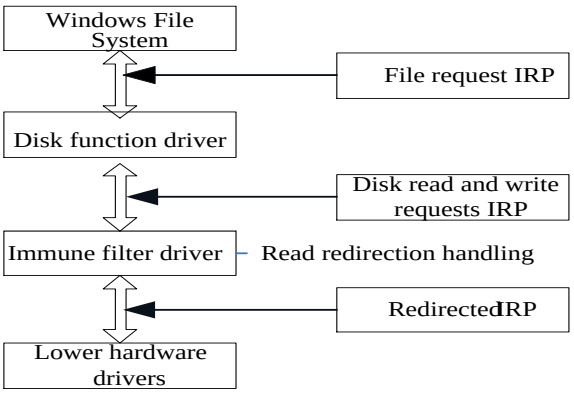

Figure 1. Windows Disk Immune System Implementation Principle Diagram

WDM driver development tool package needed:

(1)VC6.0

Used to write programs and compile user mode filter driver to communicate with the program. If no corresponding user programs and editing tools, then Source Insight, but do not want to set the VC-driven development environment, and a simple command-line compiler WDM driver, you can omit this development tools.

(2)Windows DDK

WDM driver developers must use development kit, which comes with the development of the document before writing any WDM driver, a good read the DDK documentation for developing helpful. And there comes a lot of driver instance, can be developed on the basis of these instances, this will accelerate the speed of development. In addition DDK, IFS DDK also suitable for the development of WDM filter driver.

(3)Source Insight

Read the code of the commonly used tools, and work with the editing program is also very convenient. Its associated functions do very well, but does not generate independent file programming (VC will generate a lot of files attached to write the program).

(4)Compuware DriverStudio

This tool is equivalent to the function encapsulates the DDK, WDM driver developers to make faster, suitable for developers already familiar with DDK people use, if it is beginner, it is recommended not to use this tool, first learn to use the DDK development, then then consider the speed of development problems. And this tool is quite difficult to set up the development environment.

(5)VMware.Workstation

This is a virtual machine software, use this tool to test drive it is appropriate, to avoid a lot of unnecessary trouble, because sometimes the driver will engage in system crashes, if

debugging on the actual machine, the machine could not get started. The use of virtual machines, even collapse, and only need to restore the files.

3. Modular Decomposition

Disk immune system comprises configuration module, index table module and data redirection module (including data read and write redirect) as well as data merge module consists of four modules, such as shown in Figure 2:

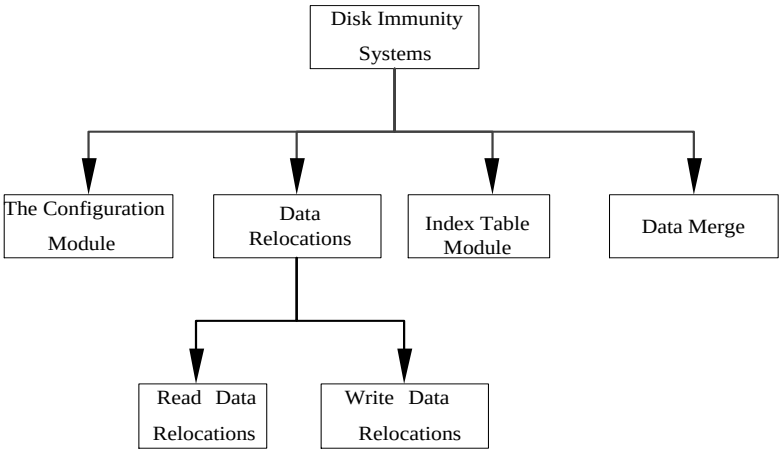

Figure 2. Disk Immune System Module Structure Diagrams

Wherein the flow of the data module: Configuration module receives configuration user mode programs, so the input of the module for the user mode application, SCSI routines (in order to facilitate a statement on the deal, said filter driver to read and write disk write I / O request packet distribution routines (Dispatch routines) to SCSI routines.) To determine whether a request packet queue, so the output of the module is configured SCSI routines based on the configuration information to configure the module; Data relocation module called by the security thread, and the index of the index table module provides data relocated to a temporary file, so it is thread-safe inputs and index table module, The data relocation module will eventually modify the index table to hold relocation information, and if it is a read request, then he will return to the safety line program to read out the data from the temporary file, Therefore, it can be considered the input module and the data relocation index table module, and the output data of simply, However, due to security thread has been called data redirection module, so have to get to the data in the temporary file, Input data can be considered a merge module is thread-safe. For more information about data flows refer to Figure 3 (note that this figure is not a data flow diagram, data flows only at runtime).

Running time dynamic analysis:

According to the data flow between modules Figure 3 to analyze the situation disk immunization program is running. When the filter driver is loaded, each module will be initialized, which saved a good configuration module reads the configuration information (in the configuration information is the user mode application once configured), then initialize the configuration information is protected in accordance with the sector range. Data relocation module will attempt to open the temporary file, the index table module is built with an empty index table and initializes its own memory allocator (in order to accelerate the speed of memory allocation, the index table module implements its own memory allocator). If in the process of loading, problems of any one module, then the disk immunization program will not run properly, so it can not protect the disk (of course,

the above module initialization error probability is very low, negligible) *etc*. These modules initialization is complete, the filter driver is also loaded a success. Then began to enter the next step of action.

When a user attempts to access the file read operation which caused the disk, reference numeral 1 generates data flow, that is, the disk write I/O request packet from the upper layer of the filter driver to the filter driver Driver.

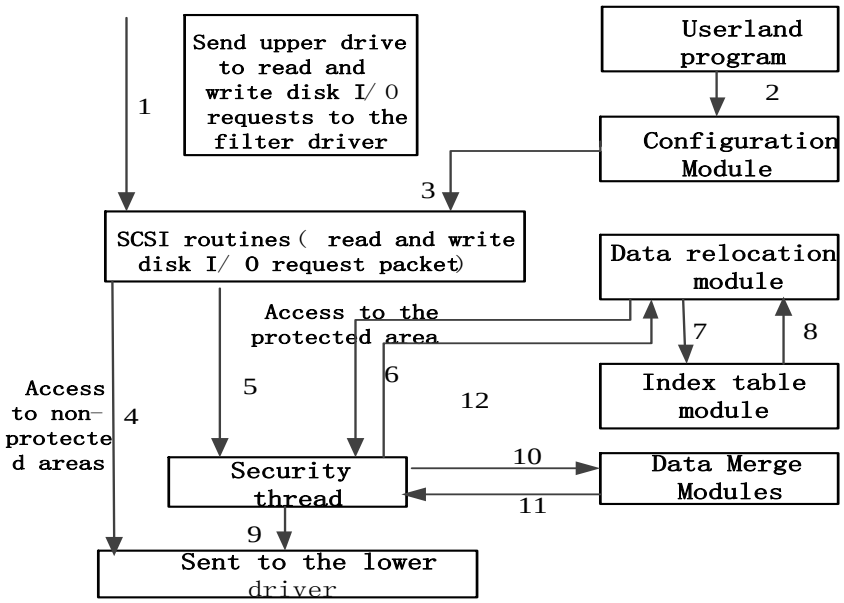

Figure 3. Data Flow Diagram Module

Reference numeral 2 represents the flow of data configuration of the disk of the user and immunization programs, the setting range of the protection zone.

Data flow numeral 3 represents a SCSI configuration information based on the configuration module provides routines to determine the read and write disk I/O requests are queued packet needs to be handed over to the security thread, in fact, is to determine the sector requested range is within the scope of the protected area.

Data flow numeral 4 represents a SCSI routines based on the configuration information to determine the I/O request packet queue does not need to be thread safe because it is located outside the area of literacy sector is protected, so you can directly the next layer drive sent.

Data flow is represented by reference numeral 5 SCSI routines based on the configuration information to determine the I/O request packet queue needs to be thread safe because it is read sector is located inside the protected area.

Reference numeral 6 denotes the flow of data is removed from the queue thread-safe I/O request packet, and extracts the start sector number of sectors and the corresponding read data of the read and write the data to the re-location module treatment.

Data flow numeral 7 represents a relocatable module passed the appropriate data to the index table module, so that the corresponding index table module for processing, when a write request, the index table module updates the index and returns relocations, re-positioning module Based on this data can be written to a temporary relocation information file, if a read request, the index table module returns relocation information, so you can re-positioning module to retrieve data based on this information and eventually pass security thread.

Numeral 8 represents the flow of data is returned to the index table module data relocation module, the module returns the results of the index table index table may be updated or return relocation information.

Numeral 9 represents the flow of data is sent to the underlying security thread driven I/O request packet, this situation only occurs when a read request, because the security thread if you can not remove all the contents to be read from the temporary file, then it need for further sending the read I/O requests to the lower driver to get data, then the data obtained were combined with the data from the temporary file to obtain the final data.

Reference numeral 10 represents a flow of data is read into the data flow of the security thread after all the required data is generated, only the data read request will cause the current case, the security thread has completed access to the data and send the request to read from the temporary file lower the drive and the data has been successfully completed and received. Then the two parts of the data sent to mobile data merge module 10 of this data.

Data flow numeral 11 represents a data merge module after the merger of two parts data back to the security thread. At this point the security thread can complete the I/O request packet and the next I get back from the queue I/O request packet and a similar deal.

Data flow numeral 12 represents a relocatable module returns to the thread-safe data when read request packet to get from the temporary file data is returned when the write request, only to return to write successful.

3.1. The Configuration Module

The configuration module belongs to the user management program, users can adopt the module set isolated partition, as well as immune protect which the disk partitions especially, according to the users set specific partition can be calculated by a conserved region of the initial termination sector entries, the sector within the area of the disk is immune protection, the other disks region are not protected, can read and write normally. After the Configuration module receives users settings, make the configuration information read-in the registry, immune filter driver read the registry when it load to, access to the configuration information, and then according to the configuration information identify the current intercepted IRP packet whether are protected areas, we must do special redirection processing for the protection regional IRP, no protection IRP is issued to directly.

3.2. Index Table Module

Data written to disk is needed to protect the area relocated to a disk other places, but need to allow the system to read write these data, so we need to build an index table, this feature is that the index table, when the system is protected read when the data area, the index table to determine whether the data read is relocated, and if yes, according to the relocation information area of the disk to read and write access to the other data, if not, then a direct read access to the data area to be protected. Construction of the index is the key to achieving the immune disk read speed according to the search index is directly related to the protection of the speed driven, and therefore need to have some algorithm to be implemented, if a simple table with the order to build the index, each time a read request came, all by traversing the entire table to determine whether the order is relocated, then the speed will be a big problem, here I am using the algorithm HASH table [5] to achieve indexing functions. Keyword HASH table is the sector address of the disk, so that a reader is given sector, will soon be able to find a new address correspondence to be repositioned in the HASH table.

Construction of the index should be noted that another problem is that the filter driver is able to support multi-threaded routines for secure access, which need to

operate in a safe manner index table to avoid indexing table node lost. This is because the system may have several threads need to read the disk, and the disk read and write regions are located in a protected area, so there are several threads may need to modify the index table to identify relocation information, thus involving multi-line application security problem accessing the index table, it is easy to think of a way to safely operate the way by locking the index table. Another method is the index table operation will have put a thread. The former method is better understood, reliable and simple to implement, while the latter method was more suitable environment such as disk immunity.

The method I used is a second, why a simple and intuitive way to first do it? Reason is not only the need to access the index table multi-threaded safe, read and write temporary files also need to consider the multi-thread safe, imagine, if a line into an area is ready to read the file, when it is good to set the file pointer position, then re-thread scheduling, and another thread and set the file pointer position, so there have been access problems. Before a thread data is not read correctly. Of course, this is a case of multiple threads public file pointer, if the file pointer is not the same with each thread may not have this problem, but will lead to new problems, that is, every time a disk is protected area, should when you reopen the file, which is not ideal, even if it can do so, write it may still be a problem, because there may be multiple threads write files in one place, this will lead to inconsistent data, had to write the operating system courses , has seen similar examples, such as the readers who write synchronization problems. Since so many places to consider multi-thread-safe access, then are used to lock the way will greatly increase the cost, thus affecting system performance. So using a single thread to achieve is an ideal choice, compared with a locking mechanism may be more suitable. To facilitate the statement, later known as the thread-safe thread. Principles used to achieve single-threaded multi-threaded access is secure, when I read the protected area of the disk I/O request packet came, first put the request packet to the queue. The work done by the security thread is continuously removed from the queue I/O request packet, and then access the index table, and then access the temporary files as needed. So after all this is done, will return to the I / O request packet control rights to the original thread (which is a protected area and read and write disk I/O request packet into the thread queue). This brings up the question, is this request packet list must support multi-thread-safe access, the problem difficult to solve, because the DDK have related functions to achieve this functionality.

3.3. Data Merge

The system reads data of the protected disk area; the data may be modified, so the request contains a portion of the data from the redirection isolation region, while the other parts of the data from the disk protection area. The two part of the data need to be combined to get the true data. For example, suppose the file A disks are located zeroth to tenth sector (sector between the range is located in protected areas), now the system is modified by a part of the A file (assuming that the modification of the tenth sectors of data), and modifications of the tenth sector[5], the immune system filter driver will make it be redirected to the isolation region, the system read-in the A file not only need to read protected area zeroth to ninth sector data, but also need to read the isolation region in order to get to the tenth sector of the updated data. We can get updated data from merging the two part data, which is data merge module function.

3.4. Data Relocations

The filter driver that intercepts the IRP package of protection region, send to the immune protection I/O processing thread processing, according to the IRP package for read and write operations are processed separately, for read operations, first find it in the index table, if it is not found, the read request area is not rewritten, the original location

data is data of the user required, will make IRP packets be issued to bottom layer driver, if the lookup results show that all the data of read requests in the isolation region, which will be redirected to the isolation area and deliver it. If a read request at the same time involve protection region and isolation region two areas, and then send to the data merge modules, data consolidation, for write operations, after updating the index lookup table [6, 7], IRP packets will be issued to underlying hardware driver after the write address redirection.

4. Detailed Analysis of the Security Thread

Because of the security thread almost have interaction with all of the modules, so its complexity is higher, is the most difficult to deal with, so it is necessary to analysis to be listed separately.

From the above analysis, the thread already know security work mainly include: (1). With relocation module interaction, so as to finish writing data to a temporary file, or from a temporary file to read data. The difference between the two is based on the I/O request packet is read or written request for resolution.

(2) With the lower driver interaction, this situation occurs only in the I/O request packet is a read request packet, safety procedures need to merge the data was obtained from the temporary file, and access to the data from disk protected area, So need to get to the first two parts data, through interaction with the relocation module has been able to get the data in the temporary file, and the data in the disk for protected area need to lower driver to send the I/O request packet to complete the operation. It is important to note that here in the I/O request packet synchronization, only the lower level driver processes the security thread to send I/O request packet, security thread to continue. This synchronization operation is done by setting the routines and events waiting to finish [3].

(3) A merger with data interaction module, this kind of situation occurs only in the I/O request packet is a read request packet, security thread through data merge modules from the temporary file access to the data and by reading the disk protected area and access to merge the data to get the final data.

(4) The completion of the I/O request packet, when all of the above operation is complete, you can call the DDK to complete the I/O request packet routine to inform the I/O manager the I/O request packet has been completed. When you're done with an I/O request packet, security thread will enter the next I/O request packet processing. Is removed from the queue new I/O request packet, and return to (1) for processing.

(5) Safety shutdown thread exit, this operation occurs only when the shutdown, and filter drivers uninstall routine has been called. Uninstall routine setting a flag, and then shut down the event waiting to happen. When security thread shutdown when detected signs corresponding cleanup work, and finally set off event, exit security thread.

5. I/O Treatment Scheme

When immune filter driver intercepts disk read and write requests IRP package of the disk driving, immune filter driver module analyze IRP package according to configuration information which users program passed through the configuration module. If the disk area is accessed by I/O request shown by IRP Package is a protected area, immune filter driver has not any modifications, we will deliver the IRP package to the lower level driver module; when I/O request of IRP access the protected area, immune filter driver module send the current IRP request to immune protection I/O processing thread to treat especially, the specific process is the current IRP request insert into an I/O request queue by the thread maintained, Immune protection I/O processing thread process each IRP request of the request queue one by one, specific process mode is to call redirection module processing the current I/O IRP request packet; redirection module call the index

table module to search the current request region whether have been redirected, and according to the search results for the current IRP package to make appropriate redirection processing, after updating and maintenance the index lookup table[8], IRP package is sent to the lower drive module, thereby completing the redirection function. When it intercepts IRP packet read operations that requested data concerning the protection zone and the separation zone two region, it must be processed special read request merge application by data merge module, the data merge module relates the read operation of reserve and isolation region, after it must dispatch more than one IRP read requests and data merge, it can complete the current read operation.

6. System Performance Testing

The disk immune system through loading disk filter driver way to intercept disk I / O requests, redirection process which protected area data read and write requests, the system will introduce some software overhead into, in order to verify the impact of disk immune software system on the disk reading and writing performance, it tests the loading and unloading disk access performance of the host machine environment e of the immune system differently. Evaluation of disk access performance of the main index is I / O rate and the data transmission rate; here we use Iometer as the test program. Iometer is issued initially by Intel Company; currently it is open source test tools, and standard test program of testing server data transmission rate and I/O ratio. Experiments specific test environment: Pentium (R) 4 2.80GHZ CPU, 512MB DDR2, Seagate ST3200827AS 200GB hard disk memory.

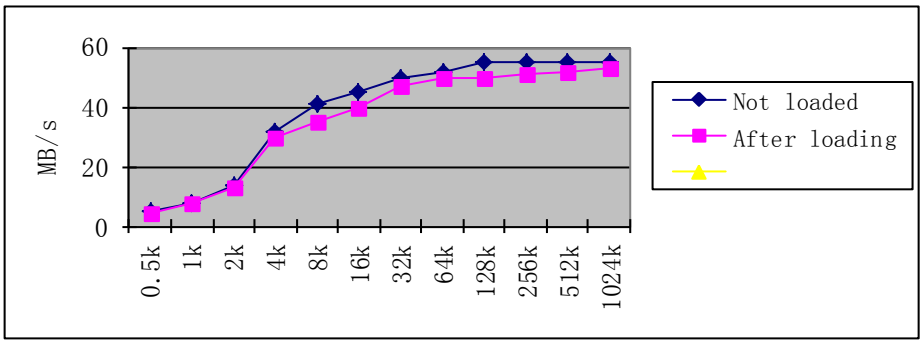

Figure 4. Loading Disk Immune System Sequence Written Performance Comparison

Figure 4 is a chart of loading disk immune protection routine sequence written performance comparison, it test the disk data transmission rate in detail according to the different block sizes. From the graph can be seen, after loading immune protection routine, it has an impact on the disk sequentially written performance finely. Immune protection system can effectively protect the data, and do not influence the disk performance significantly.

Figure 5 is a chart of disk loading immune protection routine sequential read performance comparison, the experiments test three sequential reading performance of different environments for, each is: 1, before not loading immune protection routine; 2, without any written operation after immune protection, a lookup table is vacant; 3, there is a great quantity of immune protection after loading a write operation, a lookup table is nonempty, the environment is the worst performance. From the graph can be seen, when the index lookup table is vacant, that is there is no a write request, a read request performance has only subtle differences, but after large quantities of write requests, an index lookup table is larger, search overhead becomes larger, at the moment, the search

overhead of read operation becomes much larger, another part of the read request must pass a data merge operation[9], a read operation may contain the read request of two discontinuous regions about protection zone and separation zone, the sequential read request turn into continuous multiple random access request, the read performance loss is large. When the data block becomes larger, performance loss is less, when the data block size is smaller than 32K, data transmission rate loss is relatively large, the value difference reaches to about 50% loss, but the aim of loading the immune protection system is in order to protect the partition, it will lose the read-in data after system restart, so users do not do too much write operation deliberately on the reserve, in that case it generally does not appear, so in such environment the performance loss of small data blocks request is acceptable. So if use WDM filter driver to realize disk immune function is feasible, because the additional performance loss is induced by the protective immunity that is in a reasonable range.

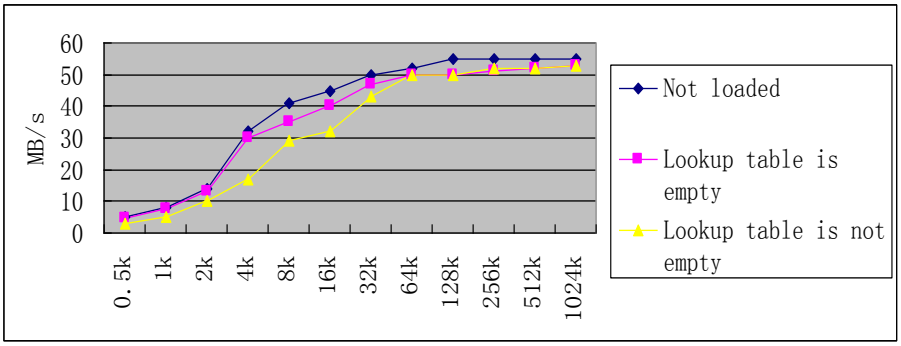

Figure 5. Loading Disk Immune System Sequence Read Performance Comparison

7. Conclusions

This paper introduces concrete design idea in detail which we use WDM filter driver to realize disk immune system, and analyzes specific modular division in detail, and detailed decompose the system I/O treatment flow sheet. Finally we test I/O properties of disk about load the disk protective immunity before and after, the test results show that the immune system has no much impact on disk I/O performance.

Acknowledgments

The work was supported by the National Natural Science Foundation of China under grant No.61103143.

References

[1] W. Oney, Microsoft Windows Driver Model: Microsoft Press, **(2003)**.
[2] Microsoft Corporation, Microsoft Corporation, Installable File system Kit Document, **(2003)**.
[3] C. Cant, "Writing Windows WDM Device Drivers, CMP, **(1999)**, pp. 321-370.
[4] Y. Weimin, "Data Structure (C Programing)", Tsinghua University Press, **(2009)**, pp. 143-233.
[5] M. Kallahalla, E. Riedel and R. Swaminathan, "PLUTUS: Scalable secure file sharing on untrusted storage", Conference on File and Storage Technology (FAST'03), San Francisco, CA, Published by USENIX, Berkeley, CA, **(2003)** March 31-April 2, pp. 29-42.
[6] A. Azagury, R. Cabetti, M. Factor, "A Two Layered Approach for Secuting an Object Store Network", SISW, **(2002)**.
[7] L. Zhicai, H. Kesen and S. Yanli, "Development of Windows WDM Device Driver Based on I/O Port Operation", Computer Engineering and Applications, vol. 11, **(2002)**, pp. 132-135.
[8] A. Baker and J. Lozano, "The Windows 2000 device driver book", Prentice Hall, vol. 11, **(2000)**, pp. 123-211.

[9] P.-H. Yun, Liya, "Design and Implementation of WDM-based Disk I/O Collector.Tv Engineering, vol. 11, **(2011)**, pp. 69-72.

Authors

Peng HaiYun, received the B.Eng degree in Computer science from Henan University and M.Eng degree in Computer science from Huazhong University of Science and Technology. She is currently researching on computer application technology.

Zhou Wen-Gang, received B.Eng Degree in computational mathematics from HeNan Normal University and the M.Eng Degree in computer application technology from North China University of technology in 1997 and 2007 respectively,He is currently researching on the analysis and design of Intelligent algorithm.

