The Implementation of Hard-Disk Protection Method Based on Disk Filter Driver

Yangwu Liu [1], Xisan Wei[1]

[1] School of Mathematics and Computer Science, Hubei University of Arts and Science
Xinagyan 441053, Hubei Province, China

Abstract

In this paper, it analyzes the working mechanism for the hard disk protection based on the current disk filter driver. The core ideology of the mechanism is to redirect the I/O request packet to the disk operation and to restore data automatically after restart the system in the case of transparent operating system to its upper storey. Also, it specifically analyzes the safety trouble by new virus, like the emerging virus, which can penetrate through the filter driver and results the failure of disk protection, and realizes a method to prevent this penetration. With monitoring the IRP in the kernel layer of operating system, it can identify the requests which have the virus read and write features and prevent it from sending requests to real disk. These methods enhance the protection function of operating system and improve the security of the hard disk protection system.

Keywords: *Filter driver, Hard-disk, Restore, Pass-through, IRP.*

1. Introduction

At present, the hard disk protection (restore) technique is normally making use of Microsoft's system architecture to detect and filter the operations of reading and writing disk in the system so as to achieve the disk data protection.

However, the emerging viruses, such as robot dog which can penetrate the filter drive and lead the failure of disk protection. Thus, it is necessary that the supervision for the IRP in the kernel layer of operating system is enforced and the method is found out using the request with feature reading and writing virus to prevent the request to real disk so as to prevent this kind of penetration and enhance the security of the restoring functions.

2. Principle of Disk Filter Drive

Microsoft's Windows management for the equipment in the kernel is hierarchical, at the same time, the access method to any device for users will be sent to the driver of target device object as final IRP I/O request packet.
The user writes and installs the drive which can be load by the system with certain specifications for the system to load in specific so that the operation request of user's layer for some equipment go through these drivers layer by layer according to certain sequence, each layer of the driver can be selected to modify, download and discard these requests.

In general, a disk filter driver is established making use of a disk filter equipment and attach to disk volume equipment, i.e. Device\Hard-disk\Volumes, so that the system can monitor the reading, writing and redirection for data on the disk.

The advantage of hard drive restore technology lie in its device-independent because filter driver is lie in disk drive and the user layer, it allows us neither consider the disk capacity in current computer system nor consider whether the disk is IDE or SCSI because these work will be completed by lower-level disk drive. The system will change reading and writing operation of the files from the user-level into IRP and deliver to the kernel disk object. And then, the kernel sent IRP one by one according to driver order hanged on the equipment. As it is completed by the driver, we can do slightly complex calculations in the driver to achieve the following performance: the storage area for user is smaller than the protected areas, the storage area can be discontinuous and the validity of the storage area is specified, etc.

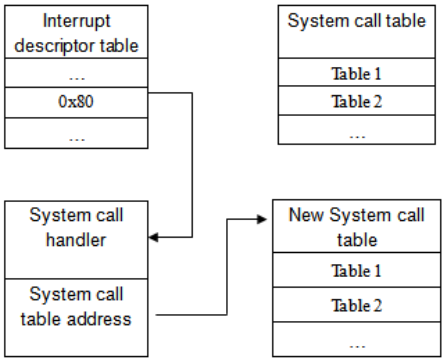

Fig. 1 Redirect the system call table.

3. Design of Hard-Disk Restore

After loading the file system in accordance with the normal operation, the disk drive is loaded in next step, here disk drive is hanged in the event of protect partition. Between loading the file system and the disk driver, the system interrupts the operation and joins its filter driver, and then, disk driver is hanged and the system is start up.
There are two kinds of methods to realize: Backup restore and Mapping restore.

3.1 Backup restore

Backup restore is the most common methods of restore. As shown in Figure 2, assuming that the users want to protect the data of 10 sectors, the data will be backed up to another area, we assume that the region is sector 100, so when the user carries on the operation again, the sector 10 is actually operated. If the sector 10 is destroyed, the data in sector 100 will be to re-copy to a buffer sector 10 in the recovery process.

The general steps of the backup restore can be listed as follows.

Step 1: Judgment whether it needs to be protected.

Step 2: Record the section information which is copied. For example: Establishes an index form record the relations of sector 10 and sector 100.

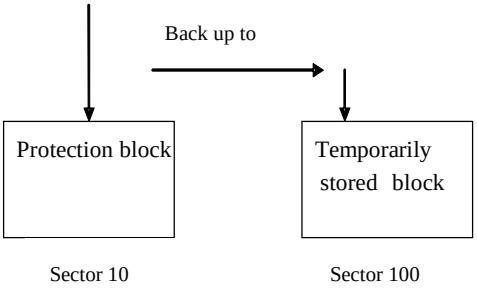

Fig. 2: Backup restore

Advantages of backup restore: simple and safe.
Shortcoming of backup restore: It takes too much space. Because a sector can store 512 bytes, but it should be used 8 bytes when it records the backup relationship of index table, so a lot of disk space will be occupied.
Solution of backup restore: Create a valid data bit map, which can reduce the index table.

3.2 Mapping Restore

Similarly, if the number of sectors which need to be protected is 10, these sectors will be protected. When the user implement the writing operation, Superficially, the user still operates the sector 10, in fact, the data has been written to the other sectors, such as the sector 100, then the user operates sector 10 again, it will be transferred to the sector 100. If the user wants to restore data, it restores directly the data in sector 10.

- A reading operation for user is shown in Figure 4. When the data is loaded, first of all, the user reads the protection bitmap and checks the module. If the module is protected, he will look into the map to search the module of data storage.

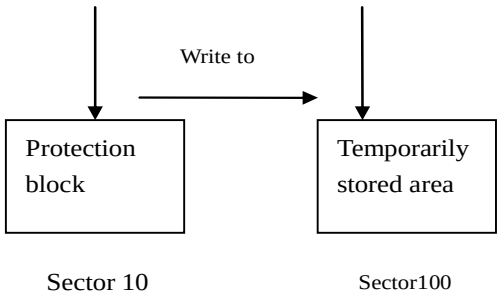

Fig. 3: Mappings restore.

- Meanwhile, in order to prevent the application procedure from accessing directly the physical disk devices, that is, \Device\Hard-disks\DRX, bypassing the filter driver of disk volume, a disk filter drive is commonly re-created and attached to the physical disk device as to filter the reading and writing on real physical disk device.

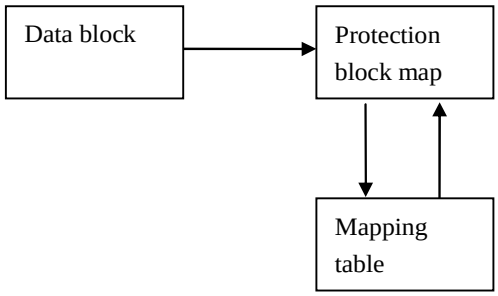

Fig. 4: Read operation.

4. Realization

Microsoft's own restore system is based mainly on the filtering of the file system. By installing a file system filter driver sr. sys with Srrstr.dll, Srsvc.dll and Srclient.dll, filtering of system will monitor the systems and some change of application file. And then, it will use the copy on write technology to automatically create restore points

which is easy to be identified. The System Restore will lapse if it skips the file system of operation protection of reading and writing.

Based on the protective system of the disk filter drive just implement filtration treatment for the reading and writing of subarea and disk, it will not truly revise the real data on the physical disk. After restarting the system, all data in buffer is discarded, the system returns the original state.

4.1 Realization process of filter drive

We know that the stack of windows management drive equipment uses the Attached-Device domain in Device-Object. For example, after an IRP of disk request is send to disk equipment of Disk.sys, it will be then forwarded to the port equipment of bus Atapi.sys. There are actual relations: Attached Device domain of Atapi.sys equipment (such as kDevice\Ide\IdeDevicePOTOLO-X) = Disk.sys equipment (such as \Device\ harddisk0\DR0).

However, Windows system does not rely on this domain to decide which equipment to send for the IRP. In most cases, the drive saves the object which is attached to the next level drive equipment in the device extension of device object which is created by itself. For example, classpnp.sys is the driver of processing routines of Device\Harddisk0\DR0, it saves the DeviceObject of kDevice\Ide\IdeDevicePOTOLO-X of lower structure in the equipment Device extension (DeviceObject-> DeviceExtension), and then, the Device-Object is used when which needs to transmit IRP down. Because the DeviceObject is stored in the custom structure, the user won't know the IRP will be sent to which lower equipment without understanding the structure of the detection tool. Only if the pointer of DeviceObject in the structure is replaced, we can easily hijack the IRP to our driver and make filter work in it. This change isn't known by the lower and upper driver.

The I/O manager is an extensive structure; the user can expand the function of I/O subsystem by developing filter driver program. The I/O manager support layered driver model, the process of each I/O request packet (IRP) will go through each layer driver till one of the layered driver responses and completes the request. The I/O manager will create the driver object and transfer to the device driver as a parameter at the entrance of the driver when Windows I/O manager is loaded device driver. The driver will use this pointer which defines the routine and register to the I/O manager, so that the I/O manager can call these routines in proper time. The driver object is stored in a DRIVER_OBJECT structure.

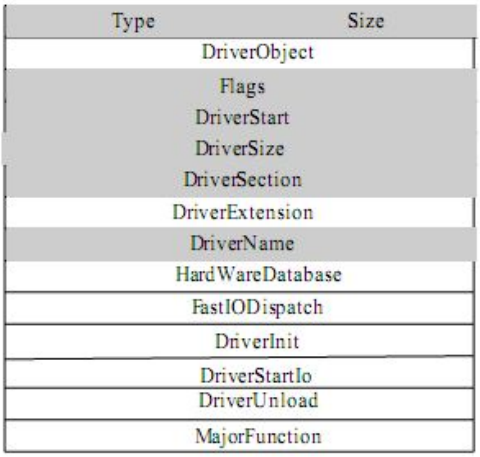

Fig. 5: DeviceObject of the structure.

The driver always needs to create device objects to serve for Win32 applications or other drivers. The device object describes physical or logical equipment. And then, the same driver object can create many device objects.

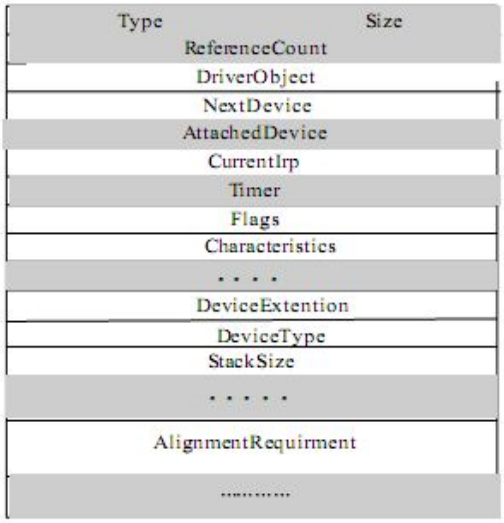

Fig. 6: DriverObject of the structure.

We create false DriverObject and DeviceObject (we don't use the system standard function IoCreateDevice, but allocate memory by ExAllocatePool), and then determines the relationship between the upper and lower equipments of disk drive through the AttachedDevice domain. The user fills some important domains to false DeviceObject and maintains consistent with the original lower DeviceObject, such as StackSize. the DriverObject domain

will be filled with false DriverObject, at the same time, the MajorFunction of DriverObject filled with the processing function will search the device extension of the upper equipment, find the field position saved the lower device and fill it with false DeviceObject, as a result, Hijack is completed finally. Next, all IRP sent to the bus driver by disk equipment driver will go through our processing function, so long as we care about the content through the MajorFunction filter, for example, implement write operation to protected disk or modify the real number of sectors etc.. And then, we can achieve the disk protection (reduction) operation by calling the processing function of the original lower equipment.

4.2 Maintaining the multithreading and multi-CPU

Firstly, we should consider the multithreading and multi-CPU, there sometimes may be some reading and writing IRP for the disk which arrive to drive interior at the same time. In view of the exclusive events for transferring the memory block, we might perform queuing process to IRP. The system creates driver object, meanwhile to assign the DriverStartIo routine. In addition, IRP_MJ_WRITE and IRP_MJ_READ should be lined up because the similarity for IRP, thus, we can assign both routines as the same one:

```
……
DriverObject->DriverStartIo = DFIoStart;
    DriverObject->MajorFunction[IRP_MJ_READ]
= DFReadWrite;
    DriverObject->MajorFunction[IRP_MJ_WRITE]
= DFReadWrite;
……
```

The system delivers the received IRP to the StartIO formation in the IRP_MJ_WRITE and IRP_MJ_READ routine：

```
NTSTATUS DFReadWrite(IN PDEVICE_OBJECT
DeviceObject, IN PIRP Irp)
  {
IoMarkIrpPending(Irp);
IoStartPacket(DeviceObject, Irp, 0, NULL);
return STATUS_PENDING;
  }
```

In the StartIo routine, the system process IO and apply for the next one with IoStartNextPacket function：

```
void DFIoStart(IN PDEVICE_OBJECT
DeviceObject, IN PIRP Irp)
  {
    ……
IoStartNextPacket(DeviceObject, TRUE);
    ……
  }.
```

5. File System Filter Pass-Through Technology

At present, the hard disk restore card (Recovery Card) normally uses the disk filter driver to process data recovery and restore. Because it works in the core layer, it is hard to be broke its stability. With more and more virus such as robot dog which can penetrate the protection card, it isn't suitable once but for all the protections. In the future, the lower level such as the monitor of the directly port I/O should be improved.

5.1 The principle of penetrate filter drive

The key realizing the restoring function of restoring software based on filter driver or interrupt-redirection is to intercept I/O operation somewhere during the process of disk I/O operation.

The application program of user mode will call API when it needs to do all kinds of disk operations, and then, API will enter kernel mode and be executed by I/O manager which will create an IRP and deliver it to various kernel mode drivers to process in turn. A device level interruption will be taken place in Kernel mode driver during the process of IRP. The interrupt handler will operate hardware to finish the operation of IRP by calling the related functions in hardware abstraction layer and send the result back to application program of user mode reversely after the operation is finished.

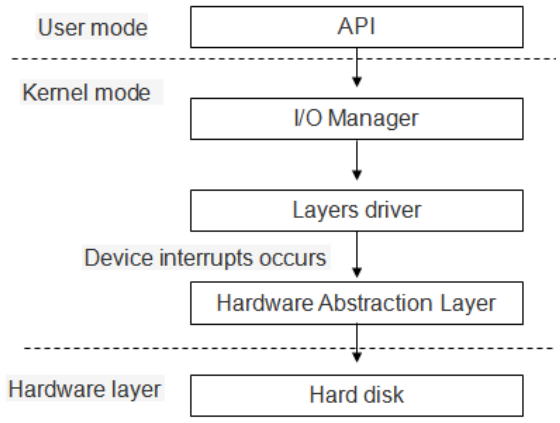

Fig. 7: Disk I/O operations of the basic processes

Due to restoring software adds the software driver in each layer in Figure 8, it must be processed so as to intercept the disk I/O operations before which is executed, As shown in Figure 7, the restoring software based on the interrupt

redirection at the position of device level interruption will modify the address of interrupt handler and intercept the disk I/O operation.

In order to penetrate the restoring software, the key of penetrating design is to ensure that the disk operation which user expected won't be intercepted by the two kinds of restoring software before which is delivered to the hardware process. An idea way is that the I/O operation is directly delivered to hardware to process. For example, the robot dog virus is just designed on this way. The I/O operation process loaded virus is shown in Figure 9.

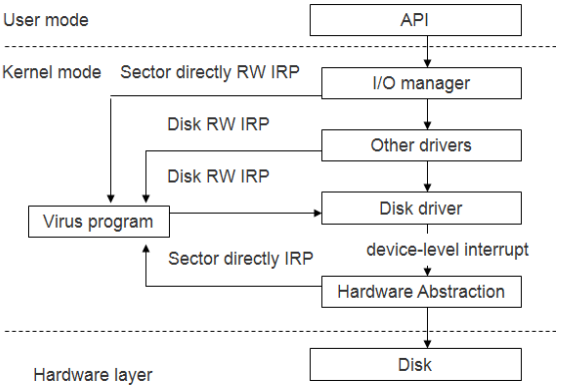

Fig. 8: The disk I/O operations processes in the virus program is loaded

As shown in Figure 8, the read-write IRP in sector is namely I/O operation not to be restored by user. This kind of IRP created by I/O manager will be processed by penetrate software directly, since it bypasses the original driver of all layers in the operating system, it can penetrate the restoring software based on filter driver. At the same time, the penetrate software calls the hardware abstraction layer function during the process of conducting this kind of IRP and interruption in the device level will not happen, so it can penetrate the restoring software based on interrupt-based redirection.

5.2 Implementation of Pass-through

Step 1: Breakthrough the limitation of driver

There are many ways to breakthrough the restriction of read-write driver. Here, we realize it by setting the IOPM mode. Changing IOPM in kernel mode driver needs two unpublished calling functions:Ke386 IoSetAccessProcess and Ke386SetIoAccessMa. The function of Ke386IoSetAccessProcess is to set of IoAccessMap for some process, the function of Ke386SetIoAccessMap is to set a number IOPM for some IoAccessMap.

Step 2: Sector directly RW

The penetrate software can call the functions of hardware abstraction layer to operate the I/O port and realize the read-write of sector directly after breakthrough the restriction for the read-write of Driver in operating system. For read-write operation of disk, you can adopt the custom's I/O control code mode except common read-write operations. In order to realize the read-write in sector directly, three I/O control codes must be defined in the penetrate software. They are IOCTL_DETECT, IOCTL_READ_SECTOR and IOCTL_ WRITE_SECTOR, their functions are getting disk information, reading sector directly and writing sector directly respectively.

Step 3: Intercept the read-write operation of disk

The core intercepting the read-write disk operation is to save and modify the dispatch function's address of disk driver. After modifying the dispatch function's address of the disk driver, the IRP sent to the disk driver originally is still sent to the disk driver, during the accomplishment of the IRP, it will call the modified dispatch function instead of the original dispatch function of the disk driver. The dispatch function's address of main function code related to the read-write disk operation includes IRP_MJ_SHUTDOWN,IRP_MJ_FLUSH_BUFFERS, IRP_MJ_CREATE,IRP_MJ_CLOSE,IRP_MJ_READ, IRP_MJ_WRITE, IRP_MJ_DEVICE_CONTROL and IRP_MJ_INTERNAL_DEVICE_ CONTROL.

Step 4: Distribute dispatch function

The dispatch function of IRP_MJ_SHUTDOWN, IRP_MJ_FLUSH_BUFFERS and IRP_MJ_INTERNAL_DEVICE_CONTROL is ShutFlush; the dispatch function of IRP_MJ_ CREATE and IRP_MJ_CLOSE is CreateClose; the dispatch function of IRP_MJ_READ and IRP_MJ_ WRITE is ReadWrite; the dispatch function of IRP_MJ_DEVICE_CONTROL is DeviceControl.

Initialize the disk reading and writing operation queue Function prototype of initialized queue as follows:

```
VOID ExInitializeNPagedLookasideList(
    IN PNPAGED_LOOKASIDE_LIST Lookaside,
    IN PALLOCATE_FUNCTION Allocate OPTIONAL,
    IN PFREE_FUNCTION Free OPTIONAL,
    IN ULONG Flags,
    IN SIZE_T Size,
    IN ULONG Tag,
    IN USHORT Depth
```

```
);
```

Among them, Look aside is defined a queue, allocate is a distributed node function, Free is a released node function, Flags and Depth are retention values which are always zero, Size is the size of node, tag is a sign of the queue.

5.3 The implementation of pass-through

1) To be sure that save the changed objects

- Get the disk quantity

We can realize the function through calling the function of IoGetConFigureurationInformation.

- Get the Device object of current disk partition one.

"\ Device, HarddiskX \ Partition0" represents the symbolic links of the first partition of the current disk, the X value ranges from zero to disk quantity. We can get the file object and the device object which linked to this symbol by calling the function of IoGetDeviceObjectPointer.

- Get the current disk partition Numbers

Calling IoBuildDeviceIoControlRequest function and creating an IRP which has a main function code IRP_MJ_DEVICE_ CONTROL and a vice function code DISK_GET_DRIVE_LAYOUT. Then, IoCallDriver function is called to finish this IRP. This IRP will return the current disk partition numbers after it executed successfully. The prototype of IoBuildDeviceIoControlRequest function is as follows:

```
PIRP  IoBuildDeviceIoControlRequest(
    IN ULONG  IoControlCode,
    IN PDEVICE_OBJECT  DeviceObject,
    IN PVOID  InputBuffer  OPTIONAL,
    IN ULONG  InputBufferLength,
    OUT PVOID  OutputBuffer  OPTIONAL,
    IN ULONG  OutputBufferLength,
    IN BOOLEAN  InternalDeviceIoControl,
    IN PKEVENT  Event,
    OUT PIO_STATUS_BLOCK  IoStatusBlock
    );
```

- Verify the effectiveness of the device object

Calling NtOpenFile function and opening a device object next partition of current disk. If it successes, there will prove that the device object is effective, and then close the device object and realize the saving and changing operations. Loop operation will complete the saving and changing operations for all disk partitions.

2) Realize the saving and changing operations

In order to realize these functions, we need to define two data structures for saving the information of disk drive object and disk device object. The data structure is defined as below:

```
typedef struct _major_table
{
 PDRIVER_OBJECT  DrvObj;
 PDRIVER_DISPATCH  ul[28];
struct _major_table *next;
}MajorTable,*PMajorTable;
```

This structure above is used to save driver object and all dispatch functions' addresses. The DrvObj is to save driver object and ul[28] is to save all dispatch functions' addresses which are 28.

```
typedef struct _dev_table
{
struct _major_table * Pmajtab;
 PDEVICE_OBJECT  devObj;
 ULONG  num;
struct _dev_table *next;
}DevTable,*PDevTable;
```

This structure is to save device objects. Pmajtab points to the address saved by the driver object of this device object, devObj is to save device object, num is to indicate the disk number of the device object and partition numbers, the high 16 bits of the num is to save the disk number and the low 16 bits of the num is to save the partition number. All the disk partition information will be linked by the field such as 'next'.

Through above-mentioned data structures, i.e. major table and dev_table, it will save all disk partition information of diver object and device object in the form of linked list. After saving each partition, it will change the dispatch function address of the partition driver object into the function address of the initial distributed dispatch function.

5.4 The queue management of disk R/W operation

Allocate memory in queue. Realize it by calling ExAllocateFromPagedLookasideList function.

Save the information related to IRP. The data structure defined in the penetrate software represents the related information of IRP, the data structure is as follows:

```
typedef struct _context
{
PIO_COMPLETION_ROUTINE oldCplRot;
 PVOID  oldContext;
 ULONG  control;
ULONG  majFun;
} Context,*PContext;
```

OldCplRot is to save the routine finished by the IRP, OldContext is to save the Context field of the IRP, Control

is to save the field of Control and majFun is to save the main function code of the IRP.

Set the complete routine for IRP. It can be defined by itself or it can be the dispatch function of the saved disk driver. We can Call IoSetCompletionRoutine function to realize, the prototype of the function is defined as below:

```
VOID    IoSetCompletionRoutine(
    IN PIRP   Irp,
IN                  PIO_COMPLETION_ROUTINE CompletionRoutine,
    IN PVOID   Context,
    IN BOOLEAN   InvokeOnSuccess,
    IN BOOLEAN   InvokeOnError,
    IN BOOLEAN   InvokeOnCancel
    );
```

IRP represents the I/O request which need to be processed, CompletionRoutine is the set completion routine, and Context is the related information saved by the last step. When InvokeOnSuccess, InvokeOnError and InvokeOnCancel are true, they represent respectively the success, failure and cancel returned by IRP calling the completion routines.

Release allocated memory in queue. It can be realized by calling ExFreeToPagedLookasideList function. The queue management of read-write disk operation defined four dispatch functions in initialization.

5.5 Sector to R/W

The penetrate software realize read-write operation in sector directly by designing I/O control code and using the mode of driver PIO RW. The penetrate software needs to define an additional I/O control code to enquiry if the current disk driver can be uninstalled except to define the I/O control code of the sector directly RW, IOCTL_DETECT, IOCTL_WRITE_SECTOR and IOCTL_READ_SECTOR. The defined additional I/O control code is IOCTL_ UNLOADQUERY.

The main function of the I/O control code is to enquiry uninstall. It confirms that all disk operation is finished and avoids the loss of disk RW operations by judging the status and times of RW.

5.6 Prevention

After the virus drive loaded successfully, it needs to penetrate reduction and sends the IRP of sector RW to the disk device object directly bypassing the reduction disk filter drive. If the reduction system can modify the information of the related disk device object in the windows kernel before robot virus drive starts, it can hide the true disk device objects' pointers and make all other programs and device objects' pointers point to the pointer of disk filter drive device of reduction system, The IRP sent by robot virus will all go through the reduction disk filter drive.

Modify all information of the bottom disk device objects in system, make the robot dog virus can't get the true information of the bottom device object no matter what layers or what levels. The pointers of the disk device object which the robot dog virus got all point to the pointer of the reduction system disk filter driver's device object, and achieve the purpose to defense the robot virus to penetrate it.

Here, we use the method of Inline Hook IofCallDriver to intercept the IRP sent by robot virus and know what information of the bottom disk device object it wants to access. We modify the information in advance and the robot virus can't get the true information of the bottom disk device objects.

6. TEST EFFECT

The program stays in the ROM chip booted ahead of the system after it was realized, make reduction system can establish a disk volume equipment smoothly and filter files through the disk equipment filter drivers. It is loaded during computer starts and protects documents not to be amended from the kernel layer.

Table 1: Test environment

Item	Version
OS	Microsoft Windows XP (Service Pack 3)
CPU	QuadCore Intel Core 2 Quad Q8200, 2333 MHz
BIOS	AMI 2TKT00AUS 11/19/07
Motherboard ID	64-0100-009999-00101111-050509-Eaglelake$5HKT18AG_08.00.15
Motherboard chipset	Intel Eaglelake G41
IDA Pro	V 4.7
Cbrom	V 2.20

Installing device drivers on Windows XP is extremely simple. The entire process consists of only two steps:

Step 1: Copy the necessary files to the system：
These files include the driver executable image (.sys) file, as well as any other files that are required by the driver.

Step 2: Create the necessary Registry entries:

These entries indicate when the driver is to be started, and also store any driver specific or device specific information that the driver may need during its initialization process. Each driver in Windows XP must have its own key in the Registry, named with the driver's name, under the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet \Services key.

Testing steps:

Step 1: Load hard disk protection drive on a normal Windows system and execute samples program of robot dog virus. Restart the computer then you can see system has been written Trojan virus by the virus.

Step 2: Write the realized program to the hard disk protection card on another uninfected computer by the virus, start it in advance before the computer starts and load it.

Step 3: Execute the sample program of robot dog virus. Restart the computer.

You will see the system is not infected by the virus when you restart the computer after having executed the sample of robot dog virus. It proves that adding prevention penetrate function to computer is effective. After many times tests, it shows that the method can avoid attacks by robot dog virus and some other type variant virus.

7. SUMMARY AND THE FUTURE WORK

The protection technology of hard disk based on filter driver realizes the restore or backup function by adding restoring software to kernel module of OS in the form of driver. The driver makes all access to disks through the "filter" of restoring software firstly, and then submits them to disk driver of operating system to be processed. This can filter the writing operations to hard disk and achieve the purpose of data reduction. However, the new emerging virus such as "robot dog" can penetrate disk filter drivers and execute read-write operation of hard disk directly. In this paper, we analyzed the methods of pass-through protection and figured out some methods on how to prevent this kind of virus in hard disk protection system. The experimental results proved that our method can prevent the varieties of robot virus and other viruses effectively.

The protection (restore) software needs to conduct data filtering protection in bus level so as to prevent some penetration techniques which aims at hard disk protection. The best way is to set the I/O access ports breakpoints by means of debug register and to intercept reading and writing operations to the ports of hard disk drive controller to fight piercing attack. Monitoring I/O operation of user mode directly is also an available method to prevent the restoring penetration under the ring3. In addition, user mode applications allowed to access I/O is a very dangerous operation; the protection software for hard disk should pay attention to intercept the privilege escalation of progress.

Acknowledgments

This work was supported by the scientific research program No.2010YA024 from Xiangyan University.

References

[1] Peisert S, Bishop M, Karin S, et al. "Analysis of computer intrusions using sequences of function calls" , IEEE Transactions on Dependable and Secure Computing, 2007, Vol. 4, No. 2, pp. 137-150.
[2] Dalton C, Gephardt C and Brown R, "Preventing hypervisor-based rootkits with trusted execution tech-enology", Network Security, No. 11, 2008, pp. 7-12.
[3] Levine John, Gizzard Julian, Bowen, Henry . "Detecting and categorizing kernel-level rootkits to aid future detection" , IEEE Security and Privacy, Vol.4, 2006, pp. 27-32.
[4] Chunxia Zhang, Peng Ji, "Application of WDM device driver in the power-supply controlling system of HIRFL-CSR", Computer Engineering, Vol. 20， No. 2, 2005, pp. 189-191.
[5] Ling Li, "Driver-developing model of virtual position sensor mapped by mouse", Journal of Zhejiang University (Engineering Science), Vol. 40，No. 8，2006，pp. 1344-1347.
[6] Tan X B, Xi H S, "Hidden semi Markov model for anomaly detection" , Applied Mathematics and Computation, Vol. 205, No. 2, pp. 562-567, 2008.
[7] Jinqian Liang, Xiaohong Guan, "A virtual disk environment for providing file system recovery" , Computers & Security, Vol.25, No. 8, pp. 589~599, 2006

Yangwu Liu was born in 1972, is a Chinese living in Xiangyang City, Hubei Province. He graduated from Wuhan University of Technology in 2009 with education in computer science and technology. He is now the director of the Basic Science Lab under the School of Math and Computer of Hubei University of Arts and Science. He has been involved in academic development, scientific research and talents cultivation. He is good at building and maintenance of LAN, maintenance of computer hardware and lab management.

Xisan Wei was born in Xiang yang of Hubei province on February, ninth 1976. He received his engineering degree in computer science and his master degree in computer technology professional from Wuhan University in 2010 . Since 2004 he is Technician at the School of Mathematics and Computer Information Systems and Software Lab, Hubei University of Arts and Science. Currently, he has the qualifications of the project management division.