

Re: Is my idea can be implement?

Re: Is my idea can be implement?

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2008-08/msg00040.1>

- *From:* "Robin" <digitalhuman@xxxxxxxxxxx>
 - *Date:* Mon, 4 Aug 2008 14:36:59 +0800
-

Hi David,

Thanks for your response.

First of all I'll make clear of my idea:

- (1) Not every sectors on the disk will be encrypted with my "WDE", such as the first cylinder of the disk which contain the MBR;
- (2) The boot code of MBR will replaced by my code, and the other sectors of the first cylinder will contain my code which is PCI BIOS functions and INT13 filter and system loader, It's can be called "my boot loader";
- (3) Before load system code, such as ntldr, the "encryption enviroment" will be ready. From then on all of the disk reading and writting options are filtered by my code(my int13 filter, they do the encryption and decryption and 'raw' disk read and write);

Seagate's FDE is a good idea to implement whole disk encryption, as you said, they should be supported by special BIOS, but I don't want do that.

OK, come back to my idea:

- (1) Pre-OS: The INT13 filter is implement reading and writting sectors in secure mode, and PCI's functions is to implement the driving of PCI card to do the encrypt and decrypt operations, these function will be called by my INT13 filter.
- (2) After OS start up: The PCI driver will drive my PCI crypto card, and disk filter driver do the reading and writting disk sectors in a secure mode, they will "use" PCI device to do the encrypt and decrypt operations through PCI driver.

But, here is the focus of my problem, when the system starting, assume that the PCI driver is loaded first, my INT13 filter is still working in the orignal mode, which will access PCI crypto card in the pre-os manner, is there have any conflict? Such and i386 protection mode problem, or the address of the PCI configuration space, status and otherwise.

Thanks Devid

Robin 04/08

"David Craig" <drivers@xxxxxxxxxxx> wrote in message

Re: Is my idea can be implement?

Re: Is my idea can be implement?

news:%23e2tSFe9IHA.3612@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

There are several phases in booting from a hard drive:

1. First the BIOS will use INT 13h, usually the 0x4n variants with current BIOS to load the MBR into 0000:7C00h. It is then executed.
2. The MBR code will load the partition boot record (PBR) from the active partition and it will be executed at the same address.
3. The PBR code will load the initial OS code and what this is named varies depending upon the OS being loaded.
4. Windows will load all boot start drivers into memory. Hint: this is IMPORTANT!!!
5. Each boot start driver will have its DriverEntry() invoked. Maybe the add device routine will be invoked followed by a start IRP.
6. After all those boot start drivers are up and running the OS will stop using the INT 13h BIOS software interrupts.

In step 5 you need to query the BIOS code (your extension if you have one or code loaded by the MBR if you use that method). The information passed into your boot start driver will be needed to do the encryption from then on. You might consider looking at Seagate's FDE drives that have encryption in the drive's firmware. That does require the BIOS be written to support those drives since no hard drive is accessible until password credentials, used to obtain access to the encryption key, are provided, validated, and presented to the drive.

When doing this type of code regardless of whether it is on-drive, or a PCI device you will find VMWare useful. Also an ICE will be needed or a lot of patience and knowledge about how storage is used in PCs.

"Robin" <digitalhuman@xxxxxxxxxxxx> wrote in message

[news:O\\$N2Nid9IHA.5668@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:O$N2Nid9IHA.5668@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

Hi all,

I want encrypt whole disk by PCI crypto-card, it's not a partition encryption but WHOLE DISK, so the system partition of OS will be encrypted.

To implement this idea, I will:

- (1) Drive my PCI crypto-card before the system booting (so called pre-os driver);
- (2) Hook my int13 filter to BIOS, which will do the decryption use my PCI crypto-card;
- (3) Implement a PCI driver for NT, which will drive my PCI crypto-card after the system starting up;
- (4) Implement a disk filter driver for NT, which will implement the read and write of disk sectors in secure mode (encryption and decryption);

My problem is: When the system starting up, the driver of disk and PCI crypto-card will be loaded,

- (1) If the PCI crypto-card driver will be loaded first, can INT13 filter call the PCI pre-os 'driver' after it is loaded? If not, the gaps is system can't read encrypted sectors any more!
- (2) If the disk filter driver will be loaded first, the INT13 filter will be pasted, any read and write of the disk sectors will be processed by the driver, but at this time, the PCI driver not loaded, can I call the "pre-os driver funtion" to do crypto?

Is there any suggestion ?

Thanks and best regards

Re: Is my idea can be implement?

Re: Is my idea can be implement?

Robin 04/08