

Re: Protecting the Operating System

Source: <http://www.derkeiler.com/Newsgroups/alt.computer.security/2006-10/msg00044.html>

- *From:* "Vanguard" <vanguard.news@xxxxxxxxxxxxx>
 - *Date:* Thu, 28 Sep 2006 22:44:55 -0500
-

"nemo_outis" <abc@xxxxxxx> wrote in message news:Xns984C53303B13Babcxyzcom@xxxxxxxxxxxxx

Yes, bootstrap programs can be moved to alternate locations and another bootstrap program ran FIRST which then chains to the wherever the original one got moved. The MBR is not protected if physical access is not protected.

Yes, this can be done, but:

1. it is not quite as straightforward as you make out
2. it is easily thwarted.

Regarding point 1, it takes a fair level of technical skill to write one's own MBR to splice into the chain. Moreover, unless the modified MBR can do what it wishes *as well as return control to the original encrypted boot process* all within one track, then it will have to put its malware elsewhere on the HD. And there our malefactor has a bit of a problem: there's no place to put it without risking trashing some encrypted file on the HD (Yes, the malefactor can just "roll the dice" that he will not trash a file, not trash an important file, or the mess will not be noticed, but that is hardly an – ahem! – elegant solution.

Regarding point 2, it is very easy to boot up from, say, a known good read-only CD and verify the MBR (or even just overwrite it). Then one would boot again, this time from the encrypted HD.

Much easier and faster than verifying the state of the entire HD. This two-stage boot is seldom done, I agree, but that's because the risk of MBR tampering is largely a theoretical one that is strongly discounted in most folks' threat model.

Regards,

The MBR is the first sector on the hard drive. In there you get 446 bytes for your bootstrap code. Rather than

Re: Protecting the Operating System

have it load a program that is stored within a partition, use the first track which isn't accessible to any partition. Several boot managers work this way by putting the rest of their program in the unused first track. $64 \text{ sectors/track} \times 512 \text{ bytes/sector} = 32\text{KB}$. Remember that the *normal* MBR bootstrap program reads the partition table (also in the MBR) to find which partition to boot from. That doesn't preclude having the custom MBR program boot from some other device which can still read the partition table in the normal MBR (or you copied the entire MBR with bootstrap area and partition table).

Boot managers have long been able to boot to partitions on other drives than the first physical one discovered by the BIOS. Since you are not restricting physical access to the computer, I can install my own hard drive inside. For a laptop (or desktop), you could clone (using a physical sector copy and not one that needs to read into the file system which is more a "logical" image) the owner's drive to your *bigger* drive so their partitions don't occupy all the drive's space and you'll have LOTS of space to do your nasty stuff. I doubt the BIOS does much more than behave as a loader to put the normal bootstrap program into memory and pass control to it, and you could do that with the huge nasty stuff you put on the unused area of the larger hard drive.

The BIOS loads your custom MBR bootstrap program and passes control to it. Your custom bootstrap program then loads your bigger program(s) from your part of the larger hard drive. Your part of the hard drive wouldn't even be in the partition table to be discovered there. The partition table is read by the standard bootstrap program to find the starting sector for the active partition but your custom bootstrap program will already have that info. Your custom bootstrap program then loads your nastyware. With 32KB for the possible size of the MBR bootstrap area and 1st unused track, it could even scan for your nastyware in the unused portion of the hard drive. Your nastyware then loads the original bootstrap program that you copied out of the MBR but under the envelope of its control.

Presumably your security product that is encrypting the drive and using its own bootstrap program to provide the other half of the key should provide a means of saving a backup of the MBR or allow you to regenerate it (using information only known to you or by having a cert saved to other media). So the above scenario would be thwarted by simply restoring the "good" MBR bootstrap area but that would require the user to know something was awry to know they needed to restore. However, I just brought up one scenario and am not going to spend months figuring out others. If DriveCrypt was worthwhile as a business solution, perhaps it stores a hash or image of its bootstrap program within the encrypted volume so it can verify it the bootstrap program that is currently in the MBR is its own rather than some nastyware substitute. Simply relying upon the inability to decrypt because the MBR bootstrap area got written with another program is not sufficient security. You need something like a 2-way view: the MBR bootstrap, if theirs, will only decrypt those volumes for which it was keyed for and the loader (that should get started BEFORE loading the OS) should look back at the *physical* MBR to determine it was its bootstrap program and that it wasn't executed from somewhere else. The good MBR bootstrap program would be needed (somewhere) to access the encrypted volume while the boot program that gets loaded from that encrypted volume does a system integrity check to ensure its boot loader is in the physical MBR so it was probably the one used to access the encrypted volume. Of course, the nastyware could load its copy of the original bootstrap program (for the security product) into memory, copy the original bootstrap program back into the physical MBR, and then pass control to the memory copy of the original bootstrap program. This makes it the nastyware less hardy as it has only one shot at capturing the login credentials but once is probably enough.

If you're not going to physically restrict access to the guts of the computer, you can't protect it from being modified. The MBR is a weak spot for the whole-drive encryption products. So far, all I've seen are users saying that it is not likely. What I haven't heard is details stipulating how these products protect their MBR bootstrap programs. Such vulnerabilities are not something the security vendors want to discuss in public.