Saturday, January 4. 2014

Enemy inside

Of course, being able to install different "OS" on a SD card or on a hard disk isn't made by an exploit. It's the regular mode of operation (many enterprise computing companies are using the ability to update the firmware of hard disks very frequently, because like with any software the software on a hard disk isn't a miraculous exception from the rule "there no such thing as bug free software). From that point the commentator is right, i misused that word. There is no exploit in the normal sense because those devices are unprotected or let' say almost unprotected ("like sending APPO to the SD-card in order to reach the firmware loading sequence"). However the ability to run arbitrary code on devices you don't think about is an exploit in itself. Because of the implications of it.

When i read about it, i thought immediately about the HI virus. The HI has a "mode of operation" where it hides in certain cells of the immune system where it is not accessible for antiviral therapy. This capability to stay latent for a while in an unreachable reservoir is one of the points that make it so hard to kill HIV as far as i understand it.

In essence the ability to reprogram disks or SD cards looks to be pretty much much the same. You can hide the virus in the disks where it delivers the virus when it is triggered for example by accessing certain blocks. Once infected even updating would be pretty easy. Just wait for a file with a certain knock sequence. And you write a lot of files to disks you are not aware of ... just look into /Users/joergmoellenkamp/Library/Safari/LocalStorage for Mac OS X Safari or /Users/joergmoellenkamp/Library/Caches/Google/Chrome/Default/Cache of Google Chrome for Mac OS X.... or all the messages "this is spam". Just think about a GIF or JPEG which contains an encrypted version of a malware. With a compromised hard disk firmware the firmware has just to wait until you receive a file for example with the knock sequence and this signalling a new version of a malware is "attached" and starts to appending this version to your files when accessing them. You could scan for certain pattern that looks like a successful detection and removal of a malware (like deletions and subsequent TRIMs on locations containing malware) and stop to deliver modified binaries until you got a new version by such a side-band channel. And you are wondering why you can't get the malware from your disk.

When you think about it: It's hard to detect what you have such a nasty roommate on your disk. When you want to be sure that you haven't a virus on your hard disk you can carry the hard disk to a different system and scan it there to be sure that your virus scanner hasn't compromised. But how to do that when the computer bolted at the bottom at the hard disk called controller is compromised. All you could do is to try to take some scope probes to the flash memory of the disk containing the firmware to read it while the micro controller isn't powered as the micro controller is potentially lying about the contents of the flashes. At the end it's a freely programmable micro controller and you could program it to lie to you or someone else.

At the end there is just one reaction when your computer has been infected once by malware. Throwing it away. Destroy it. Never reuse it. Because you can't know if the malware hadn't infected on of the other computers inside your computer. And i speculated with a colleague roughly in 2005 about a router botnet in the coffee kitchen and wrote about it in 2009 in this blog after. Do you really know that you router is still running the software you think it does. Can we be sure that the television set is still running the software you are thinking it does run. I own a modern TV set. A Linux is running on it. However it looks like those devices have been rooted, there is someone who was able to run run a Debian on a Bravia. Essentially this server could to everything. How can i be sure that my TV set hasn't been compromised, how can i be sure that my TV isn't sending SPAM, offering Torrents or something like that or offering child porn, as someone was able to exploit it remotely. I could monitor tcp dumps but i can't do it all the time.

I have the following scenario in mind. As soon as you contract the first virus, it doesn't start to do harm but just starts to do some reconnaissance, finding our what system you run, download tools to exploit other systems and then remove itself from the system where i can be detected most easily. You think your router is yours, but it isn't. You think the satellite receiver is yours, but it isn't, you think the internet radio on your night stand is yours, but it isn't ... And then to start to wreak havoc on everything around. And you have no idea what's happening. And even when you know it: How to clean up this mess. How to get back to an home IT infrastructure that has a sole master?

At the end even infecting appliances with no remote exploits could be pretty easy. With an infected hard disk the hard disk has just to wait until you access a certain file known to be a firmware update and instead delivering the known good

version delivering a modified version of the firmware that has been already exploited.

I think the proof of concepts in the recent time to exploit storage components opened a door. Of course your wasn't able to trust storage before that it really delivers the data you have stored on it before. I'm saying this for years. But with this proof of concepts we get to a state where disks could lie intentionally and alter your data intentionally.

The longer you think about it, the more horrible it gets. And the worst problem: How to trust your systems again that you process private sensitive data on it like nudies of your significant other, like you tax declaration, like your banking account. The trust is the most important victim here. And it already starts to diminish when you just think about it.

And how to explain this to the people who print out the internet to read it aka politicians ... or to your parents at the next support cycle called public holiday visit. Ouch.

Perhaps i should give away my tin foil too freely ... i think i need it for a hat or two for myself ... because you share your home with a lof of enemies ... potentially. The only thing that stands between us and such a situation is the capability of hardware vendors to ensure that nobody can upload for example a non-vendor accredited firmware on a device. But given the track record of the past and all the devices that has been rooted in the past, i won't bet my money on it. Especially for devices that start to get dirt cheap.

```
Posted by Joerg Moellenkamp in English, Security at 21:56
```

This is what Measured Boot from a Trusted Platform Module is intended to mitigate.

Of course, this assumes that you get a "known clean" software install in the first place, and that the TPM is itself not compromised. Ken Thompson dissected this issue a very long time ago in his "On the Nature of Trust" paper, and it's still just as true today.

Ultimately, "you can't build something secure, on top of something that isn't". Sadly, too many people don't seem to appreciate this.

It doesn't do your sleep much good, though.
Anonymous on Jan 4 2014, 23:05

I basically agree with all your points, but don't draw the conclusion that we are completely defenseless. Yes, an adversary who knows your system completely and can test exploits on exact clones of all your computers will be able to subvert perfectly all your digital devices.

But in practice, this is not the case, and I don't see where the magic firmware that fits in the constrained resources available on a HDD controller, but is still able to know how to patch checksummed ZFS blocks, detect TV firmware upgrades and still have no observable performance/behaviour differences would come from.

Because obviously software has bugs, and so will any malware! And while we will not be able to spot the €1M handcrafted exploit that the NSA developed to specifically subvert your computer (for whatever reason) we might be able to spot the inadvertent change of serial-number that a less skilled attacker overlooked when patching your HDD firmware. Or the additional 1MB-block of RAM the system management mode on my computer suddenly consumes in addition, because the attacker didn't think I'd notice when he put the Spy-Mode in my BIOS that needs additional storage. Or, if I can read out the flash, the changed checksum in some of the computer UEFI-Firmware's blocks, without me remembering upgrading it (of course, the problem is getting details right, but I hope you get my idea). Having a public repository of "known-good" observables/checksums for widespread computers/accessories will help here.

So what I concluded from the revelations about the NSA computer-bug-catalogue in Der Spiegel, and the corresponding 30c3 talks viewable online is that we shouldn't only monitor the integrity of the "payload" data of our computer (kernel, modules, system files) but also try to verify many of the observable characteristics of all the low-level code running in various parts of the digital devices. Then we have a good chance to detect the subterfuge, or if not, at least make exploits much more complicated, therefore more expensive to produce, and in consequence being used less.

And, of course, handling components formerly considered "internal" like potentially malignant parts, checksumming in ZFS/BTRFS and only storing encrypted data is part of it, also raising the bar considerably.
Anonymous on Jan 5 2014, 10:53

It doesn't have to detect firmware upgrades. It just has to wait until you access a file that is a known firmware upgrade. And checksums? Well ... most people i know are still using VFAT and NTFS with no checksum protection of the data. And due to the concentration in almost all markets, the number of targets is astonishingly small, there are not many hard disks vendors and compromising just one would give you a large or a very large share.

Of course we are not defenseless ... however defense gets more and more difficult.
Anonymous on Jan 5 2014, 13:32