JUNE 17, 2014

Stop sneaky hackers from launching DMA attacks

Traveling to cyber spying hotbeds? Then beware of hackers compromising your system via DMA attacks

By **Roger A. Grimes** | **InfoWorld**

Direct Memory Access (DMA) is a controller feature that has been available at least since the original IBM PC. It can be used by hackers to compromise your otherwise very heavily protected computer. Fortunately, there are steps you can take to minimize DMA-based attacks.

Although DMA attacks have been possible for decades, they gained notoriety in 2009 when researchers discovered DMA could be used to compromise Microsoft's BitLocker Drive Encryption technology, according to a **Princeton University paper**. The white-hat researchers were able to recover the private keys in several popular encryption systems, including BitLocker, FileVault, dm-crypt, and TrueCrypt, using off-the-shelf components.

How DMA works

First, a little primer is in order. Computers with DMA controllers allow DMA-aware devices, operating systems, and programs to transfer data from the participating device directly into memory, bypassing the overworked CPU for the majority of the transfer. In a nutshell, DMA significantly increases the performance of your computer and its devices. DMA-aware devices include hard drive controller cards, video graphic cards, sound boards, network interface cards, and essentially any peripheral that has the need for speed.

In Windows, you can confirm the presence of a DMA controller by opening up Device Manager and looking for the Direct Memory Access Controller under System Devices. You may also see device drivers with names that include text along the lines of "1394 controllers (OHCI compliant)."

You can determine if DMA is turned on or off on your hard drive in Linux/Unix/BSD computers by running a command similar to `hdparm -d /dev/hda`, where `hda` is the name of your hard drive (or other device you are checking for DMA functionality). Even if you don't think you have a DMA-enabled device, if you have external DMA-enabling ports, such as FireWire, PCI, PCI Express,**Thunderbolt**, Expresscard, PCMCIA, or Cardbus, it's highly likely someone can connect a DMA-enabled device to your computer and read the contents of memory.

Plug and play

The problem with DMA is that it's essentially turned on on bootup, requires no authentication to read and write memory areas, and significantly impacts the performance of your computer if you disable it. But it's quite easy to abuse. Simply walk up to a computer with a DMA-enabled accessible port or device, insert your own cable (USB, FireWire, and so on) connected to another computer or laptop, and run widely available software to read and write the other computer's memory.

I was a bit dubious about the ease with which DMA attacks could be accomplished during the early days of Windows Vista and BitLocker. At the time, I even heard (now substantiated) rumors of executives who had their laptops' data stolen while they were visiting foreign countries and taking a shower in their hotel room. Then a white-hat hacker grabbed my BitLockerprotected PC and was able to read the contents of my memory in less than two minutes.

Certainly, the DMA attack must be a first-order method used by advanced attackers such as the NSA, and Microsoft

OSes aren't the only ones vulnerable. All sorts of systems are. Here's a **DMA attack against FileVault 2 on an Apple OSX Lion computer** using a Python script.

Microsoft's **response was limited at first**. Mitigations included protecting your computer from unauthorized physical access, using advanced (non-TPM) methods of protecting BitLocker's unlock keys, avoiding suspension modes that leave data in memory (use hibernation mode instead), preventing the ability to boot from anything but your primary hard drive, and disabling unneeded DMA devices or drivers. Such mitigations had to be limited; with so many devices and programs dependent on DMA, changing the specification promised to wreck performance and interoperability.

Weighing the risks

Since DMA attacks started showing up in the media again, security people have had to ask themselves: Is turning off DMA and suffering the severe performance penalty worth it, given the relatively small risk of DMA actually being abused? The reality is that most compromises happen purely in software and have little to do with hardware attacks (memory freeze attacks are another example of a hardware-based attack). Still, I was always a little nervous when leaving my laptop unattended in foreign hotels (not that I had any secrets worth stealing).

Fortunately, **Microsoft acted on this information**. Starting with Windows 8.1 and Windows Server 2012 R2, Windows refuses access to DMA access memory by external devices until after the Windows OS is in control to parse DMA requests.

I'm not sure if Linux/Unix/BSD or Apple has added similar protections, but you can turn DMA modes on and off by device. It's still a good idea to limit booting from any device except the primary hard drive, as well as to disable unneeded DMA ports. Some Unix users might want to create scripts that disable and enable DMA only when needed.

Either way, be aware of how easy it is to carry off DMA attacks (pre-Windows 8.1) and how you can minimize the risk. I'm not sure I'd disable DMA on every computer, but I would consider additional mitigations (if not on by default already) for portable computers used by anyone accessing very sensitive information within your sphere of management, especially if they travel to **countries known for sophisticated cyber spying**.

*This story, "**Stop sneaky hackers from launching DMA attacks**," was originally published at**InfoWorld.com**. Keep up on the latest developments in **network security** and read more of **Roger Grimes' Security Adviser blog** at*

*InfoWorld.com. For the latest business technology news, follow**InfoWorld.com on Twitter**.*

http://www.infoworld.com/d/security/stop-sneaky-hackers-launching-dma-attacks244409?source=IFWNLE_nlt_sec_2014-06-17