Response to a Question About UEFI SecureBoot Mechanism Which Question Was Asked in the 2012 DMCA 1201 Exemption Proceedings

From: New Yorkers for Fair Use
http://www.nyfairuse.org

US Mail Address:
New Yorkers for Fair Use
622A President Street
Brooklyn, NY 11215

To: Mr. David O. Carson
General Counsel
U.S. Copyright Office
P.O. Box 70400
Washington, D.C. 20024

We here answer this question posed by the Copyright Office:

> At the June 5 hearing, questions were raised concerning Microsoft's new Windows 8 operating system and its supposed ability to prevent the unauthorized installation of third party independent operating systems. On this point, the Office referenced an article recently posted on Boingboing entitled "Lockdown: free/open OS maker pays Microsoft ransom for the right to boot on users' computers." See
>
> http://boingboing.net/2012/05/31/lockdown-freeopen-os-maker-p.html.
>
> In light of this article, please discuss whether and how Windows 8 would prevent another operating system from being installed on a particular computing device.

In an explanatory flyer called "What is DRM?" written and distributed by NYFU in 2003 we described the difference in the situation under the old system, that is, the system which allowed ownership of a computer by any person who bought one, and the system which the Englobulators intend to create and enforce. Here is a description of (part of) the old system, which today still obtains:

> 5. You may install an operating system different from the one the computer came with. This operating system might be one you downloaded off the net. You may use this operating system to connect to the Net, and you may freely send your work to others on the Net, as they can send their stuff to you. If the system is a GNU/Linux or free BSD system, you may look at all the source code of this operating system. If you can program, you may modify the operating system by rewriting parts of it, or adding to it, or removing parts of it. If you choose, you may share your work with others by placing copies of your code on a website, or by emailing copies to other people. In turn, other people may modify your work. Groups of programmers and users may freely work together to improve certain programs, or to learn about computers, or even just to make art.

Here is what could be the situation if we do not stop the Englobulators:

> 5. You may not install an operating system different from the one the computer came with. Installing a different operating system is a felony. No operating system is freely available on the Net. You may use only the operating system that came installed on the computer to connect to the Net. Use of any other operating system to connect to the Net is a felony. Possession of a GNU/Linux or free BSD system is a felony. All operating systems must be licensed by a joint government-cartel-monopoly licensing body. You may not look at the source code of any licensed operating system. You may not modify the operating system in any way. You are not allowed to distribute by any means any unlicensed program.

Cory Doctorow's blog post

> http://boingboing.net/2012/05/31/lockdown-freeopen-os-maker-p.html

explains that soon, unless we take action, the world will move far toward the second situation, in which the right to own a computer will be denied to most people.

Cory rightly calls the change to the new system proposed by Microsoft "a titanic shift".

We will describe first the present situation, with regard to installing an operating system on a standard home computer with an x86 cpu. Then we will describe the situation that will obtain if Microsoft has its way. In the second case we will discuss both x86 systems and ARM systems.

Present Situation:

I will again use the example I used at the 11 May 2012 hearing.

I bought a used T400 Thinkpad laptop, an x86 machine. I met the seller at a restaurant in New York City, and together we looked at the laptop. I was shown that it ran some Microsoft operating system, and we looked at some of the hardware, as shown to us by a Microsoft utility. I was satisfied and so I bought the machine. The seller left, and I then attempted to boot a different operating system, namely Kent Robotti's fine "Recovery Is Possible", for which see:

> http://en.wikipedia.org/wiki/Recovery_Is_Possible
> [page was last modified on 5 June 2012 at 20:19]

I had a bootable copy of RIP on a little "flash drive". RIP is designed to be run from either a flash drive or a CD.

Technical Matter About Situation Today:

When you turn on a Thinkpad, or most any home x86 machine, some hardware starts running. This hardware is usually the BIOS, or EFI, or UEFI, all of which manage starting up many different components inside the computer and also do some checking of components and some initialization of these components. Today many machines have UEFI as the let us call it, the pre-boot hardware and firmware. In many cases the UEFI emulates the old BIOS system, so that to the person turning the machine on, the system looks like an old system which uses the BIOS to boot.

When the BIOS, let us say, is finished checking and initializing hardware, the BIOS transfers control of the machine to a certain place in memory, and starts whatever program is at that memory location. Before it transfers control, the BIOS looks at a list of places to find the first program to run. Often it will look first at a certain partition on the hard drive. The BIOS can check that what is at that position is bootable. If it is, it boots it. If what is at the first position is not bootable, the BIOS looks at the second on its list of possible places a kernel, that is, the basic program which underlies all other programs, might be. If there is something bootable at the second place, the BIOS boots that. If the second thing is not bootable, the BIOS looks at a third place, and so on until the BIOS either finds a bootable program (which is usually a full fledged kernel) and boots that, or the BIOS gets to the end of the list without finding anything bootable, in which case, usually a funny noise is made by the BIOS, letting the user know that nothing bootable was found.

The user may change the order of the list of places the BIOS looks by hitting some particular key at startup time, usually within a few seconds of pushing the on button. If the user does not hit the special key, the startup sequence is as above. If the user does hit the key the user is shown a "BIOS screen", by which is meant an interface to the BIOS itself. A series of menus is offered and one of the choices is "Change Boot Order". This is how I was able to boot RIP from the flash drive. I changed the boot order to have the BIOS try the "USB Hard Drive" first, before the "Boot Partition on the Main Hard Drive".

There was in my case one small complication. Today many BIOSes can have a password set. When I tried to change the boot order on the laptop I was told that "Only the Administrator Can Change Boot Order". I called the seller, and he told me the password, I entered the password, changed the boot order, and ran RIP.

I then spent thirty minutes running some tests of the laptop, using the tools provided in RIP. Satisfied, I carried the laptop home.

I then traveled to Washington DC, and gave a short, perhaps less than clear, demonstration of

1. booting RIP on the laptop

2. using the cfdisk utility, erasing the Microsoft OS from the hard drive

After the hearing, within a few days, I started up the laptop, and saw that indeed the Microsoft OS was gone. That is, when the BIOS checked for an OS at the position the Microsoft kernel had been, the BIOS found nothing bootable. I then installed my usual operating system, Debian:

http://en.wikipedia.org/wiki/Debian
[page was last modified on 7 July 2012 at 14:21]

End of laptop story: I used the laptop yesterday and it ran fine, as I expected. It will replace an older Thinkpad shortly.

We have recounted this story in such detail so that the Copyright Office may have a clear picture of the steps required to remove an installed OS and to install a new OS on a standard x86 computer today.

Political, Business, and Legal Material about Today's Situation:

When I bought the laptop the deal was made between me and the seller. When I installed Debian, I got the copy of Debian I installed by pulling an "iso image" off a Debian repository by means of http. That is, I went to a website and downloaded, in the usual way, a file which is a whole Debian OS, kernel, utilities, desktop software and all, and also an "installer" a specialized piece of software which helps to get Debian copied to a hard disk, and set up so that one has now an "installed OS" on the hard drive. This is what "installing an OS" means. The OS is placed on a convenient permanent memory device, the BIOS boot order is set so that the OS's kernel is booted, and things are arranged so that "it works". In my case, I did the install from a CD, and then, to finish the install, I installed, over the Net, further software which is for me standard, such as Emacs, SCM, ssh, and rsync:

http://en.wikipedia.org/wiki/Emacs
[page was last modified on 1 July 2012 at 11:28]

http://en.wikipedia.org/wiki/SCM_%28Scheme_implementation%29
[page was last modified on 2 March 2012 at 16:07]

http://en.wikipedia.org/wiki/Secure_Shell
[page was last modified on 26 June 2012 at 14:09]

http://en.wikipedia.org/wiki/Rsync
[page was last modified on 4 July 2012 at 02:32]

Already installed from the CD were Apache, the most used web server, and bash, a standard Unix shell made by Project GNU, which is also included in OS X. gcc, the GNU Compiler Collection, also called the GNU C Compiler,

was also automatically installed from the CD. Until recently gcc was also Apple's standard C compiler, but for the latest version of OS X another free compiler, LLVM, is standard. The kernel installed was the Linux kernel whose original author is Linus Torvalds, who remains head of the team today:

> http://en.wikipedia.org/wiki/Linus_Torvalds
> [page was last modified on 28 June 2012 at 10:13]

Main Point of Political, Business, and Legal Material about Today's Situation:

I pulled a whole system, one of the several major free systems, off the Net. The Net was largely built, and today, mainly runs on free software, that is software that is freely licensed. For decades now anybody with a home machine could pull a operating system off the Net, or before fast Net connections were common, buy a CD, or a stack of diskettes, or even take the machine to a place with a fast connection, and, without asking permission of anyone, install the free OS on their machine. (If they could figure out how to; it was harder in the old days; installers have gotten better.)

These operating systems, and attendant software, starting with the Berkeley Software Distribution of Unix in about 1980, were built by academics, engineers, and unpaid enthusiasts, because they could. The bars to installing and developing software were all internal to the engineering (with perhaps some bars due to price of machines, in the early days), if the software was freely licensed.

Today, in my paid work I use Debian, SCM, Apache, ssh, and R, for which see

> http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html?_r=1

> http://bits.blogs.nytimes.com/2009/01/08/r-you-ready-for-r/

As mentioned at the 5 June 2012 hearing, Apple when failing, due partly to lack of a decent OS, turned again to Steve Jobs, who offered a system, the old NeXT system, which ran on a free kernel, which looks from above like FreeBSD:

> http://en.wikipedia.org/wiki/OS_X
> [page was last modified on 8 July 2012 at 00:07]

To repeat: Today I can buy low cost powerful computer hardware. I can install a free operating system of my choice, without asking permission of anyone. I control the computer hardware in my house. Because of this one right, the right to own a computer, and because of copyright law and cunningly crafted free licenses, and the work of thousands of paid authors and unpaid authors of free software, in practice I can run the best software in the world at home. (I admit that for some purposes, some source secret software is better, unfortunately; but we of the Movement work to repair such faults!) And if I want and I am capable, I can write and publish software which might, next year, be used by millions of people.

In brief: today for every standard x86 home computer, if I have a copy of an OS, and I am sitting before the computer, I can install that OS.

These considerations, which Aaron Williamson of the SFLC also laid out, answer again the claim made by Jesse Feder of the BSA, which appears on page 235 of the transcript of the 5 June 2012 hearing:

> The language in the legislative history is substantial adverse effect. So we're talking about something more than an adverse effect that affects a few people, you know, a few diehard people who want to tinker with the guts of their computer. We're talking about something that is significant in the context of the marketplace that we're talking about. And we're talking about a marketplace of millions and millions of phones, millions and millions of other devices.

The "few diehard people who want to tinker with the guts of their computer" built gcc, Emacs, Apache, the Linux kernel and a hundred other pieces of software which are the underlying software of the "business model" of Apple's iPhone and iPad, and of Google's Android systems.

Technical, Political, Business, and Legal Material about Next Year's Situation:

Let us consider two cases:

1. I want to buy for my own personal use a new Thinkpad laptop.

2. I want to buy for my own personal use a new ARM device which, as delivered, has Microsoft Windows 8 installed.

Case 1: I order the new Thinkpad, and it is delivered. I open the box, I go into the "BIOS screens", actually the UEFI interface, and I set the boot order so that the "USB Hard Drive" is first in the boot order. I plug in a flash drive, on which is the latest RIP OS. I shut down the Thinkpad and then turn it on again. The part of the UEFI called "SecureBoot" then informs me that UEFI will not boot my RIP OS, because the OS is not signed with the "Microsoft Signing Key".

Now if Microsoft and the hardware vendor strictly adhere to the formal promise made to all buyers of such hardware, then I should be able, perhaps with old fashioned annoying poking around in the UEFI disable "SecureBoot". At that point I should be able to boot my RIP OS. I then use RIP to poke around looking at the hardware, and, as I did at the 11 May 2012 demonstration, I remove the Microsoft OS. All is now in order for me to install Debian.

Again, I go into the, let us call it by its proper name, the UEFI, and I set the boot order so the CD is first. I place the Debian CD in the CD tray, push it in, and start the machine. Because I have disabled "SecureBoot", the UEFI should boot the CD, perhaps after warning me that what I am doing is dangerous, and I install as before.

Note that in this scenario, I will, whenever I boot the installed Debian, be running without whatever protection against "boot kits" "SecureBoot" provides. If I wish to avail myself of this protection, some more fooling with the UEFI is required. By the formal promise of Microsoft I should be able to throw away the (public part) of the Microsoft Signing Key, and create and place in the "SecureBoot" memory my own (public part) Signing Key, let us call it Jay Signing Key. I must keep carefully secret the other half, the private part, of Jay Signing Key. That private part I can use to sign the Debian kernel I choose, and then my (now slightly more hardened) laptop will boot that Jay signed kernel, and indeed any kernel I sign.

To repeat: Here is how "SecureBoot" works:

Factory Case: Microsoft has a Private Microsoft Signing Key. This key has a mate: the Public Microsoft Signing Key. All machines at the factory are loaded with the Public Microsoft Signing Key. Microsoft uses its Private Microsoft Signing Key to sign a kernel. "SecureBoot" is turned on. The Microsoft kernel is installed to the hard drive, and the rest of the Microsoft OS. Now, when the laptop is turned on, "SecureBoot", using its Public Microsoft Signing Key, checks that the kernel has been signed with the Private Microsoft Signing Key. Any kernel that is not signed UEFI refuses to boot.

Home Case where I wish to use "SecureBoot": I must create my own Private and Public signing keys. Then I play the part of Microsoft in the Factory Case.

Note the difference in the two cases: In the Factory Case, Microsoft has a power to decide what can be installed conveniently on the laptop. Further, if a Fedora OS were installed by the means set forth in Matt Garrett's article:

http://mjg59.dreamwidth.org/12368.html

using the Microsoft controlled Fedora Signing Key, which key is junior to the Microsoft Signing Key, Microsoft could, in the most common cases, stop the Fedora kernel from booting. Microsoft could do this by sending a signal to the home machine, as long as the machine has electrical power and is connected to the Net, even if no standard OS is running. UEFI has the capability to send information and receive commands over the Net. If SecureBoot is on, then the UEFI will obey any command which is signed by the senior Signing Key, which, as planned today, will always be Microsoft's Signing Key. After disablement of the Fedora kernel, by revocation of the Fedora Signing key, the home user could still turn off SecureBoot and run the Fedora kernel. Or, if the user wishes, they could remove the senior Signing Key and place their own senior Signing Key in the UEFI.

Note that these last two capabilities will only be provided if Microsoft and the hardware vendor keep strictly to the terms of Microsoft's formal promise. And even if Microsoft and the vendor scrupulously attempt to keep their promise, it is likely that, at least for the first few years of the New Regime, there will be bugs in the implementation of UEFI, and the bugs are much more likely to disable the capabilities

1. to turn off SecureBoot

2. permit installation of User's Own Signing Key

than to disable booting of the factory installed Microsoft OS. Why? Because not even Microsoft can deny a refund if the factory installed Microsoft OS does not boot at all. But certainly Microsoft and the hardware vendor can deny a refund if the user fails to install another OS. With such a complex new system as UEFI Microsoft and the hardware vendor will be able to claim that it is all the fault of the user.

Thus we see that under the proposed New Regime for x86 home computers, Microsoft will have a new, direct and much more effective means of suppressing installation of other OSes.

Short version: Under the proposed New Regime for x86 home computers, installing another OS will be more difficult than it was under the Old Regime. If everything works as Microsoft says it will, the hindrance to installing another OS, though serious, would not amount to impossibility for persons expert in x86 computers, with a good knowledge of UEFI.

Connection of Case 1 with Proposed Exemption 4:

1. I want to buy for my own personal use a new Thinkpad laptop.

that is, the x86 case, with Proposed Exemption 4: It is possible that some experts would rather disable SecureBoot, by non-standard means, than jump through absurd hoops just to easily and securely install their own choice of OS. In this case the expert should not be at risk of violating the anti-circumvention clause of the DMCA. Because removing an OS, and installing another, is a means of running lawful software, and has nothing to do with violation of copyright.

We now come to the connection of Case 2 with Proposed Exemption 4:

2. I want to buy for my own personal use a new ARM device which, as delivered, has Microsoft Windows 8 installed.

This case is different because, under the proposed New Regime, if I buy an ARM computer, that is, a computer whose central processing unit is an ARM cpu, then Microsoft has demanded that all such computers that ship with Windows 8 installed, must be forever locked to Microsoft. Microsoft and the hardware vendor intend to make it impossible to install any OS except a Microsoft OS on such computers. Here the argument for Exemption 4 is absolute: I have bought the computer and I wish to run software which has nothing whatever to do with copyright violation. No modification of any software under restrictive copyright license is involved. The owner of the hardware just wishes to replace the OS with a different OS. The shipped OS will simply be erased, and a new OS laid down. Microsoft cannot claim that there is any copyright violation involved, nor indeed any circumvention. No technical means of controlling access to a copyrighted work is circumvented. Rather an OS is removed.

We point out that at the 5 June 2012 hearing, the other side argued that most "jailbreaking" was, or could lead to, copyright violation. Perhaps this is in part true, but there is a conflation of different things in the use here of the term "jailbreaking". Some things called "jailbreaking" occur at a higher level than the kernel, or in some rare cases, involve modifying the kernel. And we admit some people might do such things in order to commit copyright violations. But installing a new kernel and a whole new collection of utilities and applications above the kernel is nothing like such above the kernel twiddling. In the case of installing a new OS, we are at the farthest extreme from the sort of thing contemplated at the time the DMCA was passed. If removing a factory installed OS, and installing an OS of our choice is considered circumvention of a technical means designed to control access to copyrighted works, well we are very close to the claim that buying a computer with no OS, and installing the same chosen OS on the computer is also a circumvention. And as many people have pointed out, the logic of the DMCA anti-circumvention clause leads, if we are not careful, to just that outcome, that is, that only Englobulator approved OSes are allowed, all others being considered means of circumvention.

We here address an argument made by Jesse Feder at the 5 June 2012 hearing. In the transcript this argument appears on page 201, and again on page 238 and following. Jesse Feder's argument is that a miasma of implicit dreams of Apple and bizarre contracts of adhesion at point of sale of Apple hardware, coupled with the DMCA anti-circumvention clause, ends the relevance of the concept "ownership of a computer". Here is a clean reductio ad absurdum of this claim: As Brett Wynkoop demonstrated at the 11 May 2012 hearing the cell phone company which sold him the hardware and a contract for services certainly thinks it owns the hardware. Why? Because the cell phone company can put software which it then runs on the cell phone Brett carries. Brett cannot, without seizing root away from the company, remove this software which Brett does not want running on the phone he carries. For the case of Apple, recently some users of Apple's iPhone devices found out that Apple was not only carefully keeping track of where they were when the iPhone was on, but also storing this information in a file on the iPhone. The outrage was large, and Apple promised not to commit that particular offensive act again. The entity which can place and remove any file, and run and stop running any software, on a computer is the owner of that computer, or to use the hacker/cracker term, the 0wner of the computer.

At the 11 May 2012 tech demo hearing, I alluded to the use of words in such a way as to systematically mislead. Throughout we have used the word "SecureBoot" to refer to the subsystem of UEFI which checks whether a kernel is signed with a private key corresponding to the public key stored in the UEFI. We should be careful to distinguish two cases:

1. The owner of the hardware has set up UEFI so that only kernels signed by the owner boot.

2. Microsoft has set up UEFI so that only Microsoft signed kernels, or kernels signed with a key junior to Microsoft's key, boot.

In Case 2, use of the phrase "Secure Boot" is incorrect. Having a computer that only boots a Microsoft OS is not a case of "Secure Boot". Having a computer which you have installed Fedora on, and which Microsoft can remotely disable, is not a case of "Secure Boot".

On page 262 of the 5 June 2012 hearing transcript Jesse Feder commits this error in claiming that in Case 2, we have a "Secure Boot":

> MR. FEDER: Again, I think this comes back to really a security issue. That's the main reason why you put these sorts of locks in place, to prevent tampering with the operating system in a way that's harmful to the user.

In the above sense of preventing harms to the user, computer security has nothing to do with the anti-circumvention clause of the DMCA.

The Copyright Office should not allow the anti-circumvention clause of the DMCA to be used as part of a broad attack on the right to own a computer. The class of works defined in Proposed Exemption 4 is:

> Computer programs that enable the installation and execution of lawfully obtained software on a personal computing device, where circumvention is performed by or at the request of the device's owner.

The install disks of Debian, FreeBSD, Minix, Inferno, and many specialized OSes fall square in the middle of this definition. These install disks contain a bundle of programs. Each disk includes an installer program which does the work of installing. Such installer programs are nothing like the programs which were in use in the 1990s by some people who sometimes committed copyright infringement using these programs. The other programs on the disk have, for the great majority of install disks, nothing to do with circumvention of technical means of controlling access to copyrighted works. Removal of a factory installed OS and installation of any of the above OSes does not modify any works under restrictive copyright license, nor are any works under restrictive license used in such a way as to violate the license. Nor is copyright infringement more enabled by this procedure than if there were no removal of an installed OS. The end result of removing an installed OS and installing a new OS is that one has a computer running the new OS. For all the OSes named above there are many instances installed on computers which never had any other OS installed. The arguments made by opponents of Proposed Exemption 4 imply that every such installed free OS tends to cause copyright infringement.

Finally we quote from page 260 and following of the transcript of the 5 June 2012 hearing:

> MR. KASUNIC: Putting that back to copyright owners, is there any copyright interest in preventing someone from completely wiping the device?
> MR. METALITZ: It's not a copyright – I think Jesse already stated
> MR. FEDER: Yeah, it's not an infringement. I think –
> MR. KASUNIC: It may be a 1201 issue, but the question is, is 1201 somehow protecting a copyright interest or is it protecting a hardware interest?
> MR. METALITZ: Well, I know that the office has made this bifurcation of what interest is being affected. I think, just looking at the statute, question of installing new operating system is, I think, totally irrelevant to the proceeding.

There is further discussion of the point, whether, in the case where the installed OS is to be erased and another OS installed, systems such as UEFI "SecureBoot" control access to a copyrighted work, or whether such systems control access to the hardware. In the quote Steve Metalitz of Mitchell, Silberberg and Knupp, on behalf of seven national organizations of creators and copyright owners, forthrightly declares that "installing a new operating system is irrelevant to the proceeding". New Yorkers for Fair Use agrees, and has argued for this position for ten years and more, before the Copyright Office and in other places. Note that it is a few pages later that Jesse Feder makes the miasma argument, in order to claim applicability of the DMCA anti-circumvention clause to the act of removing an installed OS, and installing a new OS. We call attention to these pages because this testimony strongly supports New Yorkers for Fair Use suggestion of the first of two different things that the Copyright Office might do (the Office could do both also). The two suggestions appear below at the end of the Summary.

Summary:

We agree with the Software Freedom Law Center's account of how "SecureBoot" works. We add to that account these further facts:

1. There is a hierarchy of Signing Keys.

2. The most senior key, which will be Microsoft's on most x86 home computers sold next year, can be used to remotely disable junior keys, such as the planned Fedora and Red Hat keys.

3. In the case of most new x86 home computers sold next year, Microsoft has formally promised to allow two things:
   a. the turning off of "SecureBoot"
   b. the replacement of original most senior Signing Key with a new most senior signing key of the owner's making

4. Under Microsoft's Windows 8 for ARM plans, any ARM computer which is offered at point of sale with Windows 8 will come with Microsoft's most senior signing key, and no other junior keys. Microsoft has formally stated that
   a. "SecureBoot" cannot be turned off
   b. no replacement by the user of Microsoft's most senior key is allowed

We ask that the Copyright Office take action, so that we keep our right of private ownership of a computer. In particular, we ask that either,

1. The Copyright Office declare that replacement of a whole OS, by the owner of the hardware, does not constitute a circumvention of a technical means of controlling access to a copyrighted work.

or

2. The already granted broad exemption for cell phones be extended to other in-essentially different general purpose computers, that is, that Exemption 4 be granted.

We thank the Copyright Office for its work in these proceedings!

Jay Sulzberger
jays@panix.com

Brett Wynkoop
wynkoop@wynn.com

for
New Yorkers for Fair Use
http://www.nyfairuse.org