Trusted

by John Gruber

It started with a forum thread and the following front-page blurb at OSx86, a site produced by people attempting to crack the Intel version of Mac OS X to get it to run on non-Apple computers:

> We've discovered that Rosetta uses TCPA/TPM DRM. Some parts of the GUI like ATSServer are still not native to x86—meaning that Rosetta is required by the GUI, which in turn requires TPM. See the forum topic here.

"TPM" stands for Trusted Platform Module, an industry standard hardware chip that provides for "trusted computing". There's an industry consortium, the Trusted Computing Group, whose primary promoters are AMD, HP, IBM, Intel, Microsoft, Sony, and Sun.

The whole "trusted computing" thing is highly controversial. The Wikipedia, as usual, has a fine overview, covering both what trusted computing is, and why it's controversial:

> **Trusted computing** is a term coined by the TCPA referring to the goal of their family of open specifications to make computers more secure through the use of dedicated hardware. Critics, including academics, security experts, and users of free and open source software, contend, however, that the overall effect (and perhaps intent) of trusted computing is to impose unreasonable restrictions on how people can use their computers.

The Wikipedia also has a short entry on TPM.

The gist of the situation, reportedly, is that at least some portions of the Intel Developer Transition Kit version of Mac OS X are strung up in such a way that depends upon the TPM module on the Transition Kit motherboard. I say "reportedly" because I can't say for certain there is in fact such a module on the motherboard, because I don't have a Developer Transition Kit, and if even if I did, I'd be under an NDA that would prevent me from discussing it. But I think it's safe to assume these reports are true.

The effect of this, surprise surprise, is that the version of Mac OS X that ships with the Developer Kit hardware only runs on Apple's Developer Kit Hardware. Which state of affairs is exactly what Apple stated, clearly and publicly, when they announced the transition at WWDC in June. What Apple hasn't stated publicly is *how*, technically, they were going to keep Mac OS X compiled for x86 from running on non-Apple x86 computers. The TPM chip is, apparently, at least part of the answer.

It's hard to see how this news is surprising to anyone. Given (a) that Apple has stated, adamantly, that Mac OS X will only run on Apple hardware; and (b) that despite a lot of effort and a lot of interest, no one has been able to get the Developer Kit Mac OS X release to run on non-Apple hardware; it seems rather obvious that the Developer Kit boxes contain some sort of hardware that the software is tied to. This affects no one other than those who hope to install bootleg copies of Mac OS X on their x86 PCs.[1]

And so ends the reality-based portion of the saga.

[1] Which is not to say the Developer Kit release of Mac OS X is uncrackable. No software is uncrackable. What we know, though, is that it isn't easily crackable.

Tin Foil Hat Country

On Monday, the story was picked up by Slashdot, with the following submission blurb:

> Several people have discovered that the new Intel kernel Apple has included with the Developer Kit DVD uses TCPA/TPM DRM. More specifically, it includes "a TCPA/Palladium implementation that uses a Infineon 1.1 chip which will prevent certain parts of the OS from working unless authorized.

From here, we go to Cory Doctorow, who, after reading the "news" on Slashdot, penned a scathing diatribe on Boing Boing, titled, "Apple to add Trusted Computing to the new kernel?". It starts:

> People working with early versions of the forthcoming Intel-based MacOS X operating system have discovered that Apple's new kernel makes use of Intel's Trusted Computing hardware. If this "feature" appears in a commercial, shipping version of Apple's OS, they'll lose me as a customer—I've used Apple computers since 1979 and have a Mac tattooed on my right bicep, but this is a deal-breaker.

First, it seems a bit euphemistic to describe people trying to crack Mac OS X to run on non-Apple x86 hardware as "people working with early versions of the forthcoming Intel-based MacOS [*sic*] X operating system". "People working with" creates the impression that these are legitimate developers, expressing legitimate concerns.

Doctorow continues:

> I travel in the kinds of circles where many people use GNU/Linux on their computers—and not only use it, but actually call it GNU/Linux instead of just "Linux," in the fashion called for by Richard Stallman. Some of these people give me grief over the fact that I use Mac OS X instead of GNU/Linux on my Powerbook, because the MacOS is proprietary.

There is a word for these people. That word is *asshole*. No, wait, *zealot*. OK, there are two words for these people.

> When my free software companions give me grief over this, I tell them that I'm using an OS built on a free flavor of Unix, and that most of the apps I use are likewise free—such as Firefox, my terminal app, etc.
>
> Here's the important part though: when I use apps that aren't free, like Apple's Mail.app, BBEdit, NetNewsWire, etc, I do so comfortable in the fact that they save their data-files in free *formats*, open file-formats that can be read by free or proprietary applications. That means that I always retain the power to switch apps when I need to.

Here at least, Doctorow makes complete sense. Indeed, open, interchangeable file formats are much more important than free software.[2] Best are apps that directly read and write to open formats; good enough are apps that import from and export to open formats.[3]

But then he enters tin foil hat country:

[2]Although it's arguable whether Apple Mail's new .emlx file format counts as "open". It is a plain text format—more or less an RFC822 mail message followed by plist-format XML data—but the flags in the plist data are undocumented.

[3]This is one of several reasons why few people are shedding tears over the continuing demise of StuffIt—the .sit and .sitx file formats are completely proprietary; no software other than StuffIt can unpack a .sit archive.

> The point of Trusted Computing is to make it hard—impossible, if you believe the snake-oil salesmen from the Trusted Computing world—to open a document in a player other than the one that wrote it in the first place, unless the application vendor authorizes it. It's like a blender that will only chop the food that Cuisinart says you're allowed to chop. It's like a car that will only take the brand of gas that Ford will let you fill it with. It's like a web-site that you can only load in the browser that the author intended it to be seen in.

Certainly such a scenario is *a* potential use of Trusted Computing DRM mechanisms—and such a scenario would indeed be dreadful—but it's a far stretch to call it *the* "point of Trusted Computing". In the actual case here, Apple's Developer Transition Kits—which, I'll remind you, may bear *zero* resemblance, internally or externally, to the actual Intel-powered computers Apple will eventually ship to real customers—are (reportedly) using TPM for one and only one purpose: to prevent the OS from being run on non-Apple hardware.

There is no indication, none, zero, not even a whiff, that Apple intends to enable, let alone encourage, developers to create software with the TPM file-access authorization-locking described by Doctorow above. None.

This is not about third-party software developers limiting access to *your* data. This is about Apple limiting access to *their* operating system.

Even if you ignore the fact that there isn't any evidence, just *think* about it. What motivation would Apple have for allowing or encouraging this sort of lock-in? Ever since the transition to Mac OS X, Apple has been moving in the direction of openness, releasing the entire low-level OS core as open source, bending over backwards for compatibility with Windows wherever possible—and but now they're going to allow for this? It boggles the mind.

Or consider iTunes, the most prominent software from Apple that makes use of DRM. The DRM in iTunes doesn't add any restrictions to music you already own. Apple went to bat *for* users' rights with the ITMS, getting better rights and lower pricing (for hit singles at least) than what the record labels wanted to offer. And if you still don't like the ITMS DRM policies—which is a perfectly reasonable stance—you can simply not use the ITMS and still be a happy iTunes / iPod user.

Back to Doctorow:

> What this means is that "open formats" is no longer meaningful. An application can write documents in "open formats" but use Trusted Computing to prevent competing applications from reading them.

This is completely wrong. The existence of a TPM chip on the motherboard does not mean that application developers will be able to create apps which can produce files which can only be opened in the app that created the file. You'd need operating system support for that, including a way to prevent files from being copied to other computers, and adding a feature like this to the OS would be suicidal. In what way could such a capability possibly be construed as beneficial to users? And, I repeat, there is no evidence whatsoever that Apple plans to do this.

If a file is written to disk in an open format, no TPM chip can prevent you from opening it in another program. And if your data is written to disk in an encrypted format, with the decryption keys tied TPM hardware outside your control—then your software is clearly not writing to an "open format".

> Apple may never implement this in their own apps (though I'll be shocked silly if it isn't used in iTunes and the DVD player), but Trusted Computing in the kernel is like a rifle on the mantelpiece: if it's present in act one, it'll go off by act three.

No one has ever claimed that copy-protected DVDs are an "open format", nor has Apple ever claimed that DRM-protected music from ITMS is an "open format". The whole point of the DRM is that the formats *aren't* open. These restrictions aren't from Apple, they're from the entertainment industry. Even if they wanted to, Apple couldn't sell non-DRM protected major-record-label music from the ITMS, nor could they ship a version of DVD Player which allowed you to, say, export video from a commercial DVD to an unprotected QuickTime movie.

If you're not happy with these restrictions in iTunes and DVD Player, blame the entertainment industry. Again, I see no reason to believe that Apple is interested in making these restrictions even more onerous than they already are.

> So that means that if Apple carries on down this path, I'm going to exercise my market power and switch away, and, for the first time since 1979, I won't use an Apple product as my main computer. I may even have my tattoo removed.

Down *what* path, though? Down the path where individual Mac apps start using TPM-enabled DRM to prevent you from opening your files with other applications? Well, duh, if Apple does that, nearly *every* Mac user will be in line for a new brand of computer.

Or is the mere presence of a bogeyman TPM chip on the motherboard, which chip is used as nothing more than a dongle that ties Mac OS X to Apple-sanctioned hardware, enough to count as "carr[ying] on down this path"? If that's the case, Doctorow might as well start packing his bag.

It'll be interesting to see what platform Doctorow switches to. Windows is seemingly out of the question, considering that Windows Vista is slated to contain numerous TPM-based security features.

But he can always switch to GNU/Linux, right? Actually, wrong—because the Linux kernel has driver support for TPM, too. Go ahead and look at the source code of `drivers/char/tpm/tpm_infineon.c`, which starts:

```
/*
 * Description:
 * Device Driver for the Infineon Technologies
 * SLD 9630 TT Trusted Platform Module
 * Specifications at www.trustedcomputinggroup.org
```

I suppose he could compile his own version of the Linux kernel minus any support for TPM, but the wiser course would be to realize that TPM support, in and of itself, is no more or less evil than support for any other chips on a motherboard.

Decency

When I told a few friends this week that I planned to write about Doctorow's outburst, several indicated that I shouldn't bother, more or less on the grounds that Doctorow deserves to be cut a bit of slack because his heart is in the right place. The idea being, *OK, sure, he's out of line in complaining about this, but at least he's just looking out for users' rights.*

But the fact that his intentions are good doesn't make it right to let this pass. Doctorow is clearly ascribing deviousness onto Apple, without a single shred of evidence to back it up. But anyone who knows him as both an Apple aficionado and a champion of users's rights, and who trusts his opinions on such matters, is going to take away from this the idea that Apple is doing something malicious, and that bad things are likely to start happening with users' data on Intel-powered Mac systems.

The idea that Doctorow's critique, despite being unfounded, was good for the community simply because of its anti-DRM slant is like saying McCarthyism was good for the country simply because of its anti-Communist slant.

Yet here we are, with Tim O'Reilly calling Doctorow's post "a great rant":

> In his usual lucid way, Cory goes on to explain why open data is more important than free software, and how the proposed DRM cuts to the heart of the essential freedom to switch to another program, or another computer. Well worth a read.

Except that the only "proposed DRM" that would get in the way of any "freedom to switch" exists only in completely fabricated scenarios from Doctorow himself.