TDDC03 Projects, Spring 2006

Protecting Mac OS X

Restricting use on the x86 platform

Johan Bengtsson
johbe496@student.liu.se

Love Thyresson
lovth296@student.liu.se

Supervisor: Viiveke Fåk

Protecting Mac OS X

Restricting use on the x86 platform

Johan Bengtsson
johbe496@student.liu.se

Love Thyresson
lovth296@student.liu.se

University of Linköping, Sweden

Abstract

In late autumn 2005 Apple Computers announced that they would switch their entire computer product line to Intel processors. Along with the decision came an announcement that the Mac OS X operating system has been secretly developed for the Intel platform as well as for the Power PC. Since Apple only wants Mac OS X to be used on Apple hardware, measures have been taken to prevent their operating system from being used on bulk PC hardware. This article discusses how Apple protects their operating system, from both piracy and misuse.

1. Introduction

Following up their announcement from last year, Apple computers released their first version (10.4.3) of the Mac OS X operating system for the x86 platform. This has presented them with a number of problems in regards to preventing any x86-system, e.g. bulk PC hardware, to be able to run OS X. The differences between a new Macintosh computer and a PC can be narrowed down to two components; the security module TPM (Trusted Platform Module), and the BIOS replacement EFI (Extensible Firmware Interface). We will study how these two techniques are used in Mac OS X.

Furthermore we discuss possible alternative solutions to improve or maintain the level of security provided.

2. Mac OS X protection measures

In this chapter we will study the protections used in Mac OS X.

2.1 Background

The Mac OS X operating system is now available in two flavors, a PowerPC version and an x86-version. It is now possible to run the system on an ordinary PC [2]. However, this is not possible without modifications to the operating system itself.

So, why would anyone want to install and run OS X on an ordinary PC instead of a Macintosh? Imagine that you could build your own Macintosh clone using bulk PC hardware. Then you could get your own Macintosh a lot cheaper compared to buying the hardware from Apple. The OSX86 project forum [2] constantly update their list of OS X compatible hardware, with the main goal to build an OS X compatible computer as cheap as possible. Their current shopping list will get you a compatible computer for under $200. If you also want a license for OS X the price will sum up to around $330 [3]. Compare that price with the currently cheapest computer available from Apple, the Mac Mini, with a price tag of $599 [4] and you will surely understand why anyone would want to build their own Macintosh clone.

Since Apple Computers generate most of their income from hardware sales [5], their main concern would be to prevent anyone from using Mac OS X without using a Macintosh computer.

Apple is trying to eliminate this problem by making sure their operating system only runs on Apple specified hardware. This is accomplished by implementing a trusted hardware solution called TPM.

2.2 Extensible Firmware Interface

Extensible Firmware Interface, or EFI for short, is one of the necessary components to build a TPM protected system. EFI is a replacement for the PC BIOS and provides a different way to handle both system calls and the boot-up process.

2.2.1 History

EFI is currently at version 2.0 and was initially developed by Intel [6] as a replacement for the ageing PC BIOS. It provides POST (Power On Self Test) as well as an interface between the physical hardware and the operating system. At this time EFI is maintained and being developed by the Unified EFI Forum [7].

2.2.2 Implementation

One of the key differences between PC BIOS and EFI is that EFI allows vendors to create operating system independent device drivers. Since Apple has been using this aspect in Open Firmware on the PowerPC platform for quite a few years, this was the obvious choice. EFI provides an interface between layers, see figure 1.

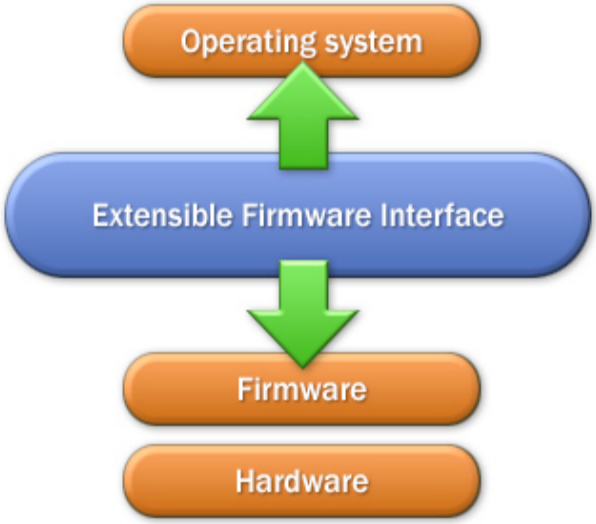

Figure 1: EFI hardware layers [8]

A problem with the PC BIOS is that modern operating systems cannot use the real-mode functions provided by the BIOS. Instead they have to use its own protected mode drivers for each piece of hardware it wants to use. Using EFI, manufacturers will be free to write their own operating system independent hardware drivers, which can be included within the device itself and then be accessed directly by the operating system. A typical application could be to retrieve updates to an operating system before the actual installation takes place.

Extensions to EFI can be loaded from virtually any non-volatile storage device attached to the computer. This provides the possibility (for e.g. OEM manufacturers) to create a small hidden partition to extend the standard EFI BIOS functions. EFI provides the ability to boot into an EFI shell prior to loading the actual operating system. This shell can be used to run diagnostics, play DVDs and CDs even before the operating system is loaded.

2.3 Trusted Platform Module

2.3.1 History

Trusted Computing Group, or TCG for short is the latest buzz in the computer industry [9]. When Microsoft announced they would start cooperating with TCG, or their implementation of it, Next Generation Secure Computing Base, the computer industry went wild. Next Generation Secure Computing Base is widely known under its development name, Palladium. Their official motive for this action was to be able to prevent malicious software, virus, worms, trojans, etc. to take control of the computer. The majority of PC users however, also realized that it would most likely be used for DRM (Digital Rights Management) purposes, e.g. to prevent the spreading and usage of copyrighted material. However, there are areas that could greatly benefit of the advantages a TPM provides, e.g. improved security in e-commerce, encryption of e-mails, etc.

Now, what does all of this have to do with Apple? Since the switch to x86 began all of their computers has included some sort of trusted computing solution, namely Trusted Platform Module, or TPM for short [1]. In Apple's case, TPM is used to prevent Mac OS X to run on bulk PC hardware, not to perform any control over the user. This might change in future versions, but for now it's simply a method of prevention.

2.3.2 Architecture

The idea of TPM is built upon a chain of trust [11], where the combination of hardware, software and firmware provides the root of this chain. The trust is then extended from the root to every link in the chain, i.e. the other parts of the platform. Much of the trust that you get from the TPM comes from the so-called Endorsement key. The Endorsement key is a 2048 bits key-pair. One key of the pair is private and the other one is public. The private key is unique for every TPM and is protected within the TPM at all time.

Within the TPM there are three kinds of certificates. First we have the Endorsement Certificate, which contains the public part of the Endorsement key. The Endorsement certificate guarantees that the TPM is genuine. There's also a Platform Certificate, which is provided by the platform vendor, which guarantees that the security components are genuine. The last type of certificates is the Conformance certificate, which guarantees the security properties of the platform.

We have not found much information about how these certificates are used. The Endorsement Certificate contains the public part of the Endorsement key and must therefore be used for accessing it. The use of the other certificates are quite vague and they seem to be more like digital certification stickers.

The TPM has a set of built-in cryptographic capabilities such as an RSA accelerator, a SHA-1 hash engine and a random number generator. These built-in

cryptographic functions can only be executed within the TPM hardware itself hence no software or hardware outside the TPM has access to these cryptographic functions.

For more in-depth information regarding TPM, please see Mario Strasser's semester thesis "A Software-based TPM Emulator for Linux" [13].

2.3.3 Protection Modes

The TPM can be used in four different modes; binding, sealing, signing and sealed signing. They all provide different levels of security and are applicable in different scenarios.

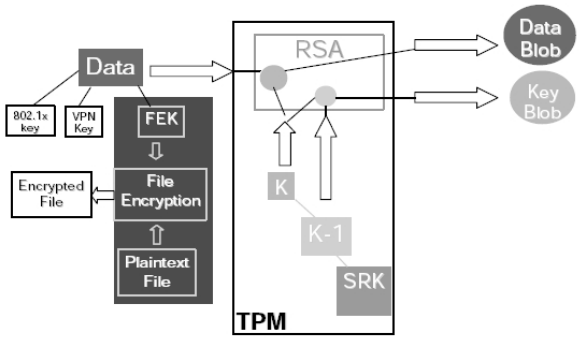

Figure 2: Encryption using the TPM [11]

Binding

Using this mode, the data will be encrypted using a public key. Decrypting the data will be done using the corresponding private key. This key is stored, and protected, within the TPM. It can be stored either as a migratable or non-migratable key. Using the migratable mode, the key is transferable between multiple TPM devices, while the non-migratable mode binds the key to the specific TPM.

Sealing

Sealing is a special case of binding where the data is protected by using a non-migratable key. The data is also bound to the specific platform configuration. This is done by concatenating the data with platform specific information before encryption. After decrypting the same data blob the information is compared to the current platform's configuration and if it matches the data will be released.

While this provides a high level of security, all data will be lost in case of e.g. a hardware failure.

Signing

The signing mode uses a signing only key. This key is used for signing data and cannot be used for encryption. It is used to verify the origin of a specific data blob.

Sealed Signing

Sealed signing is used to verify that the platform that signed the message meets specific configuration requirements.

3. Alternative solutions

In this chapter we will present and discuss three different protection techniques.

3.1 Background

Since the x86-version of OS X was revealed to the public, it has been both hacked and tweaked numerous times to be more compatible with bulk PC hardware [12]. Apple seems to have taken measures to protect itself when they implemented TPM, but is has been widely observed that this is not enough. Surely it will prevent the masses from installing their operating system, and that may very well be their intention. However, to be more on the safe side there are more approaches Apple could take to prevent misuse of their operating system.

3.2 Hardware dongle

One approach which has been successfully deployed in other software areas is to use a hardware dongle, that is a piece of hardware containing cryptographic checksums, which can be plugged into, e.g. a USB port. This type of protection has been more or less successfully used in many professional software systems. However, since this is essentially what TPM is, it might not prove as good as expected. If TPM can be circumvented, chances are a hardware dongle will be even easier to bypass.

3.3 Product activation

Product activation is probably one of the more efficient ways to block out unwanted users. If a specific serial can be paired with a random code generated from the hardware during the installation, and then be encrypted and stored locally, these values could be compared at boot time.

The problem with this kind of protection is quite obvious. As soon as any hardware in the system changes, the operating system will stop working. This could be solved in a number of ways, the easiest being to contact the vendor to unlock the specific copy.

3.4 Serial number

The commonly used serial number protection is included in OS X. At installation time you have to enter a valid serial number to be able to continue with the installation. Apple has chosen to not have the installation program validate the entered serial number with a central server. Instead they just check locally if the serial number is valid. By having this design Apple are not able to block keys at installation that are known to have been spread. One advantage with the chosen design is that there is no need for any Internet connection or phone cards to the support, which in turn makes the installation process easier for the user.

One aspect Apple has yet to consider is to check the serial when performing system updates and disallow updates for bad serials. This might in turn leave many systems vulnerable for attacks. Badly protected systems could lead to more successful intrusion-attempts, which would seriously damage Apple's reputation of having a secure operating system.

4. Conclusions

This chapter presents our conclusions.

4.1 Trusted Platform Module

We believe that the use of a TPM could be a good choice for the future. At the moment it gives a certain level of protection when it comes to blocking the use of Apple's operating system on a Macintosh clone. We are not sure if Apple is using the full potential of the TPM, but the usage of the TPM's features could always be extended in the future. For the future we do not see any real disadvantages about using a TPM as long as the user is a "nice user" that for example is not using pirated software. The TPM could be used for DRM purposes and hence, be used to prevent the use of pirated software. But in a world where nobody is using pirated software or using the computer for other illegal purposes, there would be no big use for a TPM. At least from many software companies' point of view.

4.2 Product activation

One could wonder why Apple has chosen to not have any activation of their OS, but one reason might be that Apple have not had any major problems with software piracy in the past because of their rather low market share (3% as of January 2006). When Apple was using the PowerPC processor, the majority of people bought their Macintosh from them. A license for the OS was included when buying your computer and that way they all had a license.

Today Apple is using the Intel processor. This change of platform makes it easier to build your own Macintosh clone cheaper, which could lead to a more noticeable problem with software piracy like for example Microsoft are experiencing [10]. The serial number might not be the strongest protection for preventing installation of pirated software, but with an activation process, like the one Microsoft are using, it might be a cheap and simple way to reduce piracy.

We believe Apple is making a mistake by not having any activation process. The fact that they have been blessed with a low rate of piracy in the past doesn't mean that they will be as fortune in the future.

4.3 Summary

Are the currently implemented protections in Mac OS X enough to prevent misuse? While the trusted computing solution prevents the average Joe from installing their operating system, it does not make it impossible. There is no such thing as an unbreakable system; it's all about the amount of effort needed to break it. And there will always be people who will try; they may not even be interested in using the system, but rather to prove it possible. Still, we believe that Apple has reached their intended goal using this level of protection.

The main advantage right now may be to prevent misuse, but in the future it seems quite likely that it will be used for DRM purposes connected with Apple's iTunes Music Store. This could limit the spreading of purchased copyrighted material.

Since Apple naturally do not want it to be known how their protection of the x86 version of Mac OS X is implemented, all web sites have been quickly closed down due to threats of law suits regarding violation of the DMCA [14]. Therefore we have not been able to find any detailed information on how the protection was circumvented, as well as to what extent the TPM is actually used.

Remember, even though it may sound like a TPM solution is only here to limit how you can use your computer, the same solution can still be used to protect your sensitive information.

References

[1] Trusted Platform Module specification https://www.trustedcomputinggroup.org/groups/pc_client/, 2003.

[2] The OSX86 Project forum http://forum.osx86project.org, 2006.

[3] Apple Computers Mac OSX information http://www.apple.com/macosx, 2006.

[4] Apple Store USA
http://www.apple.com/store, 2006.
[5] Apple Public Relations
http://www.apple.com/pr, 2006.
[6] Extensible Firmware Interface
http://www.intel.com/technology/efi/, 2006.
[7] Unified EFI Forum
http://www.uefi.org/, 2006.
[8] Wikipedia
http://en.wikipedia.org/wiki/Extensible_Firmware_Interface, 2006.
[9] Trusted Computing Group
http://www.trustedcomputinggroup.org, 2006.
[10] Business Software Alliance
http://www.bsa.org, 2006.
[11] S. Bajikar, Trusted Platform Module (TPM) based Security Notebook PCs - White Paper
http://developer.intel.com/design/mobile/platform/downloads/Trusted_Platform_Module_White_Paper.pdf, 2002.
[12] Maxxuss web page
http://www.maxxuss.org, 2006
[13] M. Strasser, A Software-based TPM Emulator for Linux
http://www.infsec.ethz.ch/people/psevinc/TPMEmulatorReport.pdf, 2004.
[14] Digital Millenium Copyright Act
http://www.copyright.gov/legislation/dmca.pdf, 1998.