Is there trusted path implementations in OSX

Per Christian Kofstad
IMT3501, Software Security
Oblig 1 - s.nr: 120916

***Abstract*—An argument of wherever Apple OS X does have trusted path technology built in the operating system or not.**

I. What is trusted path

Trusted path is a mechanism of ensuring that the user is communicating whit the right components, not hacked or modified ones. As described in "NIST - Guide for Developing Security Plans for Federal Information Systems" [1], it is stated (definition below) that trusted path has to do with the confidence that the system actually does what you think it does, and not anything else.

> *A mechanism by which a user (through an input device) can communicate directly with the security functions of the information system with the necessary confidence to support the system security policy. This mechanism can only be activated by the user or the security functions of the information system and cannot be imitated by untrusted software.*

A much used example is the Ctrl+Alt+Del command inn the windows systems, applied in the windows NT and later versions. This enables the user to jump to the loginscreen despite anything that might be running on the computer, and presenting something on the screen at the moment. There is no other program that can or is allowed to use this combination. We can see in this example that it fits the definition from NIST [1]. The question is, is there implementations in the Apple OS X operating system that can be argued to be trusted path implementations?

II. OS X Sandbox

The Sandbox app protection technology [2] in OS X is an example of a trusted path. The developer and the user that uses a sandboxed application must give permission to every resource the app needs or wants to use “outside” its own code. This means that if a user wants to open files from for instance his own personal documents with a sandboxed application he must drag the files into the application, to specifically allow the application to access the files in another way.

If the application tries to access recourses that it is not allowed to access it will simply be blocked. Sandbox is a trusted path in a way that it will only gain access to resources that you have allowed it access to. This means that you know what your app can do and can not do, and more importantly, you can restrict the application only to reach the resources that you want it to reach.

III. XProtect

Xprotect [3] is not strictly speaking a trusted path technology, but it helps other technologies with obtaining more secure operations. Xprotect is more like a traditional anti-virus and malware scanner. But since there could be malware that uses a zero-day exploit, the XProtect program will not be able to recognize it, and therefore cannot be considered a trusted path, since there is a way to get around it.

IV. Gatekeeper

Gatekeeper [4] is a protection technology that in a way is an example of trusted path. The technology controls applications that is installed on your system. It can be restricted to only allow downloads from “Mac App Store” [5] or ”Mac App Store and identified developers.” This allows the user to be ensured that the programs the user install is authentic to what the developers created it to be, and that apple has approved of the programs. This technology uses a cryptographic hash against a signature that ensures the programs to be authentic. If the code is changed the hash will also change, then it will not match against the signature and gatekeeper will block the program.

Gatekeeper ensures that whatever program you download it is checked against “Mac App Store” [5] or other approved developers. This can help ensure the user that the user gets what the user is asking for, and not malware or other types of bad code.

V. File Quarantine

File Quarantine [3] is a protection technology in OS X that is an example of a trusted path. The technology is implemented in Safari, Messages, iChat and Mail. The technology checks every file that is received or downloaded within these applications and puts them in quarantine. Only files that is verified and signed through Gatekeeper will be downloaded and can be opened without showing a popup warning. If you download a file that is not a signed, the file is automatically quarantined by the system. When doing the quarantining some meta data is stored connected to the file. When a user tries to open the file, a warning popup occurs. Before this popup occurs, the file is automatically checked (by XProtect [3]), informing you if the file has a known malware signature or not. If it does not have a known signature, you are showed when it was downloaded, by whom, and from which site or sender.

The user still have to know about this feature and use it right, like everything else. But, if this is used the right way, it has the potential to help users identify wherever a file is what the user think it is, or not. Therefore you can argue that this is a technology that is an example of a trusted path.

VI. Key combinations

In the Apple OS X operating system you also have some keyboard shortcuts [6] similar to what you find i windows with ctrl+alt+del [7]. For more detailed examples, see the full list in the article "OS X: Keyboard shortcuts" by the Apple support team [6]. In this article you find reference to three shortcuts that could be argued to be trusted path.

> ***Control + Shift + Eject*** *is the lock-screen shortcut.*
> ***Control + Shift + Esc*** *is the force quit shortcut.*
> ***F11*** *Show Desktop*

However I could not find any key combination that helps users lock-up and create a trusted path scenario for the system login screen. Since this is the much used and well known example of trusted path scenarios, it may not have been a focus area from the developer teams of OS X.

VII. Conclusion

OS X has a few good security features built in. However you must use them right in order for them to work, and they must be turned on. If you turn them off, in order to avoid annoying warnings and popups, you are basically leaved unprotected. This is like every other system that tries to protect the users. Apple can seem strict with the Gatekeeper [4], Sandbox [**?**] and the possibility only to allow programs from the Mac App StoreMacAppStore. This is a good combination of protection features and technology, however if all of them truly provide a secure path scenario according to our definition [1], is arguable. If things is arguable, it may not be as clear as it should be when it comes to security definitions. As the last and final statement, the prime example of a trusted path Ctrl+Alt+Del combination in Windows, when you want to log on, does not have an equal in OS X.

References

[1] P. B. Marianne Swanson, Joan Hash, “Guide for developing security plans for federal information systems,” *NIST Special Publication*, no. 800-18, February 2006. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-18-Rev1/sp800-18-Rev1-final.pdf

[2] A. D. Team, “App sandbox design guide,” *developer.apple.com*, 2014. [Online]. Available: https://developer.apple.com/library/mac/documentation/security/conceptual/AppSandboxDesignGuide/AppSandboxQuickStart/AppSandboxQuickStart.html

[3] A. S. Team, “Os x: About the "are you sure you want to open it?" alert (file quarantine / known malware detection),” *support.apple.com*, 2014. [Online]. Available: http://support.apple.com/kb/HT3662

[4] ——, “Os x: About gatekeeper,” *support.apple.com*, 2014. [Online]. Available: http://support.apple.com/kb/HT5290

[5] Apple, “Info about mac app store,” *Apple.com*, 2014. [Online]. Available: http://www.apple.com/no/osx/apps/app-store.html

[6] A. S. Team, “Os x: Keyboard shortcuts,” *support.apple.com*, August 2014. [Online]. Available: http://support.apple.com/kb/ht1343

[7] M. S. Team, “Keyboard shortcuts for windows,” *support.microsoft.com*, September 2014. [Online]. Available: http://support.microsoft.com/kb/126449