Trusted Path in OS X

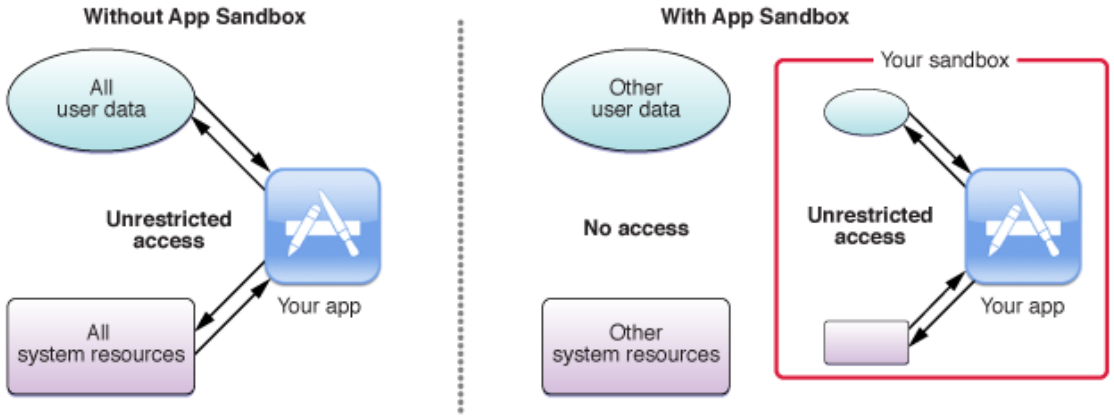
Daniel Moen Antonsen

IMT3501 - Software Security
Obligatory exercise 1
September 3, 2014

In Linux systems, Trusted Path Execution (TPE) mechanisms are used to prevent malicious code from being run in different parts of the system, at the execution of a file. Access to executables are only permitted if the user is considered trusted a user, or if the path from where the executable is being run, is considered a trusted path.

Although this concept is considered a good solution to a problem that is difficult to solve, OS X has taken a different approach to handling the situation. It does so by using a sandboxing technique, a security measure released with Snow Leopard (formerly known as seatbelt in Leopard).

Sandboxing works by forcing processes/applications to run inside a confined sandbox space with restricted access to anything outside of the sandbox itself. In a sandboxed space, all application data (except user saved files) are placed in safe, user containers paths and the application is restricted to access anything outside of these containers without the users concent. Application behaviors such as creating directories, spawning processes, configuring network settings, communicating with other apps, and much more becomes highly restricted in this sate, making it exceedingly difficult for a attacker to exploit a application security hole.



App with and without sandboxing [3]

Implementations

After the release of Lion, Apple made it a required that developers had to implement sandboxing in their applications in order to pass code review and distribute it via the App Store, which has not surprisingly caused lot of controversy as many applications are no longer able to interact with different parts of the system like it used to. In short, developers must now specify what kind of

access the application needs in order to do its job. These privileges are called entitlements and a service deamon process named *sandboxd* manages whether the application is using more or less entitlements then strictly needed.

As far as the OS goes, it does not force sandboxing by default so any application that is not sandboxed already, must use a wrapper to archive this result. It is worth noting however, that Apple tries to make sure that all applications, even those outside of App Store is trusted and safe to run. This is done with the implementation of Gatekeeper, an built in security measure that determines whether an application is authentic and trusted.

Sandbox wrapper

To wrap an application, the executable /usr/bin/sandbox-exec must be issued along with a (-f) sandbox profile plus the full path of application to be run. Profiles are essentially rules that determine the behavior of a particular application as to what it can and can not do, and OS X comes with a preset of profiles located in /usr/share/sandbox

In the example below, the TextEdit app is sandboxed with the preset profile *lockdown.sb*. [1]

```
sandbox-exec -f /usr/share/sandbox/lockdown.sb /Applications/TextEdit.app/Contects/MaxOS/TextEdit
```

References

[1] William Barker Charles Edge, *Enterprise mac security: Mac os x snow leopard*, 2 ed., Apress, 06 2010.

[2] Matthias Gransrigler, *Os x lion app sandbox and its implications on applications*, 03 2014.

[3] Apple Inc., *About app sandbox*, `http://tinyurl.com/q9o5jpu`.

[4] Jonathan Levin, *Mac os x and ios internals: To the apple's core*, 1 ed., Wrox; 1 edition, 11 2012.