Resolving App Sandbox Inheritance Problems

This thread has been locked by a moderator.

This post is part of a cluster of posts related to the trusted execution system. If you found your way here directly, I recommend that you start at the top.

Share and Enjoy
—
Quinn “The Eskimo!” @ Developer Technical Support @ Apple
`let myEmail = "eskimo" + "1" + "@" + "apple.com"`

Resolving App Sandbox Inheritance Problems

If you’re creating a product with the App Sandbox enabled and it crashes with a trap within `_libsecinit_appsandbox`, it’s likely that you’re tripping over one of the following problems:

- Nonheritable entitlements
- Changing sandbox
- Nothing to inherit

Nonheritable Entitlements

The most common cause of this problem is also the most obscure. If you have a sandboxed app with an embedded program that you run as a child process, a crash in `_libsecinit_appsandbox` is most likely caused by the embedded program being signed with entitlements that can’t be inherited.

Imagine an, SandboxInit, with an embedded helper tool, `NotHeritable`. When the app runs the helper tool as a child process, the helper tool crashes with a crash report like this:

```
Exception Type:        EXC_BAD_INSTRUCTION (SIGILL)
…

Thread 0 Crashed::  Dispatch queue: com.apple.main-thread
0  libsystem_secinit.dylib … _libsecinit_appsandbox.cold.7 + 49
1  libsystem_secinit.dylib … _libsecinit_appsandbox + 2096
2  libsystem_trace.dylib   … _os_activity_initiate_impl + 51
3  libsystem_secinit.dylib … _libsecinit_initializer + 67
4  libSystem.B.dylib       … libSystem_initializer + 286
5  dyld                    … invocation function for block in dyld4::Loade…
6  dyld                    … invocation function for block in dyld3::MachO…
7  dyld                    … invocation function for block in dyld3::MachO…
8  dyld                    … dyld3::MachOFile::forEachLoadCommand(Diagnost…
9  dyld                    … dyld3::MachOFile::forEachSection(void (dyld3:…
10 dyld                    … dyld3::MachOAnalyzer::forEachInitializer(Diag…
11 dyld                    … dyld4::Loader::findAndRunAllInitializers(dyld…
12 dyld                    … dyld4::APIs::runAllInitializersForMain() + 38
13 dyld                    … dyld4::prepare(dyld4::APIs&, dyld3::MachOAnal…
14 dyld                    … start + 388
```

The helper tool has trapped within `_libsecinit_appsandbox`. Look at the entitlements of the helper tool:

```
% codesign -d --entitlements - SandboxInit.app/Contents/MacOS/NotHeritable
…
[Dict]
    [Key] com.apple.security.app-sandbox
    [Value]
        [Bool] true
    [Key] com.apple.security.inherit
    [Value]
        [Bool] true
    [Key] com.apple.security.get-task-allow
    [Value]
        [Bool] true
```

The `com.apple.security.app-sandbox` and `com.apple.security.inherit` entitlements are fine: They configure the program to inherit its sandbox from its parent. The problem is the `com.apple.security.get-task-allow` entitlement, which is not compatible with this sandbox inheritance.

The `com.apple.security.get-task-allow` entitlement is often automatically injected by Xcode. For information on how to prevent this, see Embedding a Command-Line Tool in a Sandboxed App.

Some entitlements *are* compatible with sandbox inheritance. Unfortunately that list of entitlements is not documented (r. 93582428). The most commonly use ones are the hardened runtime exception entitlements documented in Hardened Runtime. The other entitlements on the list are pretty obscure.

Changing Sandbox

Another cause of a trap within `_libsecinit_appsandbox` is the child process trying to set up its own sandbox. If a sandboxed process runs another program as a child process, that child process always inherits its sandbox from the parent. If the program’s executable is signed with `com.apple.security.app-sandbox` but not `com.apple.security.inherit` — that is, it tries to set up a new sandbox — it will crash in `_libsecinit_appsandbox`. A process is not allowed to change its sandbox.

To check for this problem, look for the following in the crash report:

```
Application Specific Signatures:
SYSCALL_SET_PROFILE
```

This indicates that the process tried to setup its sandbox profile but that failed, in this case because it already has a sandbox profile.

To fix this problem, either:

- In the child, add the `com.apple.security.inherit` entitlement so that it inherits its sandbox from the parent.
- In the parent, run the program so that it’s not a child process. For example, you could launch it using `NSWorkspace`.

Nothing to Inherit

Another cause of a trap within `_libsecinit_appsandbox` is when a nonsandboxed process runs another program as a child process and that other program’s executable has the `com.apple.security.app-sandbox` and `com.apple.security.inherit` entitlements. That is, the child process wants to inherit its sandbox from its parent but there’s nothing to inherit.

To check for this problem, look for the following in the crash report:

```
Application Specific Information:
Process is not in an inherited sandbox.
```

There are three ways you might fix this problem:

- Sandbox the parent.
- Unsandbox the child.
- Run the child in its own sandbox by removing the `com.apple.security.inherit` entitlement.

Gatekeeper Code Signing Notarization

Reply

Posted 4 months ago by eskimo

Add a Comment