Resolving Hardened Runtime Incompatibilities

This thread has been locked by a moderator.

242

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 Hardened Runtime Incompatibilities

The hardened runtime enables a number of security checks within a process. Some coding techniques are incompatible with the hardened runtime. Best practice is to enable the hardened runtime on all code so that you uncover these problems early. However, some folks only notice these problems when they go to distribute their product. Specifically, Developer ID distribution requires notarisation and the notary service requires the hardened runtime.

The typical symptoms of this failure is that the program launches but then fails almost immediately afterwards. Sometimes this failure results in the program just not working, sometimes it triggers a crash, and sometimes the program terminates itself by calling `exit`. In some cases this failure is accompanied by diagnostic printed to `stdout`, so it’s worth running the program from Terminal if you can. For advice on how to do that, see Resolving Trusted Execution Problems.

If you suspect that this problem is caused by a hardened runtime incompatibility, the diagnostic test is easy: Temporarily disable the hardened runtime and see if the problem goes away.

Once you’ve confirmed the problem is an incompatibility with the hardened runtime, you have two choices:

- Fix your code to avoid the problem.
- Apply a hardened runtime exception entitlement to disable one specific security feature of the hardened runtime.

In general, the first option is best because it leaves your product with the best security.

IMPORTANT When confronted by a hardened runtime incompatibility, some folks apply *all* of the hardened runtime exception entitlements. This is a mistake for three reasons:

- Some entitlements are subsets of other entitlements. For example, there’s no point applying `com.apple.security.cs.allow-unsigned-executable-memory` if you’re already applying `com.apple.security.cs.disable-executable-page-protection`.
- Disabling library validation with the `com.apple.security.cs.disable-library-validation` entitlement makes it harder to pass Gatekeeper. For more on this, see Resolving Gatekeeper Problems Caused by Dangling Load Command Paths.
- The hardened runtime exists for a reason: To enhance your product’s security.

Don’t apply a hardened runtime exception entitlement without first understanding what the actual problem is.

Debug hardened runtime incompatibilities like you would any other problem: Step through the code, add logging, and so on. The goal is to isolate which part of your code works with the hardened runtime disabled but fails with it enabled.

Once you’ve found that code, the fix is usually pretty obvious. For example, if your program needs to generate executable code on the fly, you must:

- Allocate that memory using `mmap` with the `MAP_JIT` flag.
- Write to that memory using `pthread_jit_write_protect_np` or one of its related APIs. See the `pthread_jit_write_protect_np` man page for details.
- Sign the problem with the `com.apple.security.cs.allow-jit` entitlement.

If you need help with this, feel free to ask here on DevForums.

If you don’t control the code that has the hardened runtime incompatibility — this happens most often when using a third-party language runtime — ask the vendor how to proceed. They might have an updated version of their code, or specific advice on what hardened runtime exception entitlements to apply.

If their advice is “Apply all the hardened runtime exception entitlements!”, think carefully about your vendor choices (-:

Gatekeeper Code Signing Notarization

Reply

Posted 4 months ago by eskimo

Add a Comment

Developer › Forums

Platforms
iOS
iPadOS
macOS
tvOS
watchOS

Tools
Swift
SwiftUI
SF Symbols
Swift Playgrounds
TestFlight
Xcode
Xcode Cloud

Topics & Technologies
Accessibility
Accessories
App Extensions
App Store
Audio & Video
Augmented Reality
Business
Design
Distribution
Education
Fonts
Games
Health & Fitness
In-App Purchase
Localization
Maps & Location
Machine Learning
Security
Safari & Web

Resources
Documentation
Curriculum
Downloads
Forums
Videos

Support
Support Articles
Contact Us
Bug Reporting
System Status

Account
Apple Developer
App Store Connect
Certificates, IDs, & Profiles
Feedback Assistant

Programs
Apple Developer Program
Apple Developer Enterprise Program
App Store Small Business Program
MFi Program
News Partner Program
Video Partner Program
Security Bounty Program
Security Research Device Program

Events
App Accelerators
App Store Awards
Apple Design Awards
Apple Developer Academies
Entrepreneur Camp
Tech Talks
WWDC

To view the latest developer news, visit News and Updates.