Viewing Sandbox Violation Reports

This thread has been locked by a moderator.

1.5k

IMPORTANT This post has been replaced by an official document, Discovering and diagnosing App Sandbox violations. Yay! I'm leaving it here for historical context only.

The best way to view sandbox violation reports has changed over the years, so I thought I'd post some up-to-date info. I tested the following with Xcode 13.3 on macOS 12.2.

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

Viewing Sandbox Violation Reports

After enabling the App Sandbox, you may find that your app fails in some non-obvious way. That is, something within your app doesn't work, but your app doesn't display a permissions error and so you have no idea where to start. If you find yourself in that situation, look for a sandbox violation report:

1. Run the Console app.
2. For each line below, copy the line and paste it in to the search box at the top right.

   ```
   type:error
   subsystem:com.apple.sandbox.reporting
   category:violation
   ```

   This searches for sandbox violation reports.
   Note The exact query terms have changed over time. The above is accurate from macOS 12.2.
3. Click the Save button in the bar below the search box and enter a name for your saved search. I typically use "Sandbox" for this. In future, click on this saved search to skip the previous step.
4. Click "Start streaming".
5. Run your app and reproduce the problem.
6. Look for a sandbox violation report in the log. If you see one, follow the steps below to investigate.

Inside a Sandbox Violation report

A sandbox violation report log entry looks like this:

```
type: error
time: 11:58:26.009175+0000
process: sandboxd
subsystem: com.apple.sandbox.reporting
category: violation
message: Sandbox: AppSandboxViolat(5807) deny(1) file-read-data /Users/quinn/.ssh/id_rsa
```

The message includes a complete sandbox violation report. That's too big to include here, so I've added it as a text attachment.

Viewing Sandbox Violation Reports.txt

Look at the `Violation` field first:

```
Violation:       deny(1) file-read-data /Users/quinn/.ssh/id_rsa
```

This means that the App Sandbox blocked an attempt to read the data of the file at the path `/Users/quinn/.ssh/id_rsa`.

Next look at the thread backtraces. It's normally pretty easy to identify the thread responsible for the violation: It's the one blocked in a system call that could reasonably trigger this violation. For example:

```
Thread 0 (id: 291952):
0   libsystem_kernel.dylib   … __open + 10
1   Foundation               … _NSReadBytesFromFileWithExtendedAttributes + 167
2   Foundation               … -[NSData(NSData) initWithContentsOfFile:options:maxLength:error:] + 119
3   libswiftFoundation.dylib … NSData.__allocating_init(contentsOf:options:) + 77
4   libswiftFoundation.dylib … Data.init(contentsOf:options:) + 76
5   AppSandboxViolator       … ViewController.violateAction(_:) + 1507 (ViewController.swift:25)
6   AppSandboxViolator       … @objc ViewController.violateAction(_:) + 65 (<compiler-generated>:0)
7   AppKit                   … -[NSApplication(NSResponder) sendAction:to:from:] + 288
…
```

Here you see that AppKit has called the `ViewController.violateAction(_:)` method (frame 6) which has tried to created a `Data` value from the contents of a file (frame 5) which has eventually called `open` (frame 0), which is what triggered the violation.

Using this information, investigate and fix your sandbox incompatibility.

No Sandbox Violation Report

It's possible that you might not see a sandbox violation report even though the problem is caused by the App Sandbox. For example, imagine you have code like this:

```
let url: URL = … some file URL …
let data: Data
if access(url.path, R_OK) == 0 {
    data = try Data(contentsOf: url)
} else {
    data = Data()
}
```

The `access` system call never triggers a sandbox violation report. If the App Sandbox blocks access to `url`, this code will not fail, not generate a sandbox violation report, and set `data` to empty.

IMPORTANT Preflighting file system calls is racy and, in the worse case, can result in TOCTTOU vulnerabilities. Avoid writing code like this.

Debugging problems like this can be tricky.

App Sandbox

Reply

Posted 1 year 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.