

Recording Private Data in the System Log

This thread has been locked by a moderator.

366

I've discussed all of the following on DevForums before, but scattered across multiple threads. The issue of system log private data recently came up in another context and I decided to collect my thoughts in one place.

If you have any questions or comments, start a new thread and tag it with *OSLog* so that I see it.

Share and Enjoy

—

Quinn "The Eskimo!" @ Developer Technical Support @ Apple

let myEmail = "eskimo" + "1" + "@" + "apple.com"

Recording Private Data in the System Log

The unified system log on Apple platforms is a Thing of Wonder™. If you're not convinced, go read [Your Friend the System Log](#).

One great feature of the system log is private data: If you tag a value as private, it's omitted from the log by default. This keep personal information out the log, which is obviously a good thing.

In some circumstances it makes sense to record private data. Imagine, for example, that you're investigating a Gatekeeper problem on macOS and you see a log entry like this:

```
type: info
time: 2022-05-12 05:53:35.896830 -0700
process: XprotectService
subsystem: com.apple.xprotect
category: xprotect
message: SecAssessment results: <private> (null)
```

That looks interesting, but the assessment result, the thing that you really want to see, was omitted. Now look at the same log entry with the private data enabled:

```
type: info
time: 2022-05-12 05:55:21.673682 -0700
process: XprotectService
subsystem: com.apple.xprotect
category: xprotect
message: SecAssessment results: {
  "assessment:authority" = {
    LSDownloadRiskCategoryKey = LSRiskCategoryMayContainUnsafeExecutable;
    "assessment:authority:flags" = 0;
    "assessment:authority:source" = "_XProtect";
  };
  "assessment:remote" = 1;
  "assessment:verdict" = 1;
} (null)
```

That's *much* more helpful.

There are two ways to enable recording of private data in the system log:

- Add an `OSLogPreferences` property in your app's profile (all platforms).
- Create and install a custom configuration profile with the `SystemLogging` (`com.apple.system.logging`) payload (macOS only).

Log Preferences Property

Imagine an app with this code:

```
let log = Logger(subsystem: "com.example.apple-samplecode.PrivateValueLogger", category: "app")

let client: String = ...
log.debug("will answer the ultimate question, client: \(client)")
let answer: Int = ... lots of work ...
log.debug("did answer the ultimate question, answer: \(answer, privacy: .private)")
```

Note This uses the `Logger` API in Swift but the same logic applies to all system log APIs.

With this code both the `client` and `answer` values are considered private data, the first because that's the default for strings and the second because it's explicitly called out as private.

And if you run the app outside of Xcode you'll see that this private data is omitted:

```
type: debug
time: 16:55:23.994070+0100
process: PrivateValueLogger
subsystem: com.example.apple-samplecode.PrivateValueLogger
category: app
message: will answer the ultimate question, client: <private>

type: debug
time: 16:55:23.994108+0100
process: PrivateValueLogger
subsystem: com.example.apple-samplecode.PrivateValueLogger
category: app
message: did answer the ultimate question, answer: <private>
```

Note When you use Xcode to run the app, it sets an environment variable, `OS_ACTIVITY_DT_MODE`, that changes the default handling of log entries.

To record this private data, add the following to the app's `Info.plist` file:

```
<key>OSLogPreferences</key>
<dict>
  <key>com.example.apple-samplecode.PrivateValueLogger</key>
  <dict>
    <key>app</key>
    <dict>
      <key>Enable-Private-Data</key>
      <true/>
    </dict>
  </dict>
</dict>
```

Now the private data is recorded in the log:

```
...
message: will answer the ultimate question, client: The Mice

...
message: did answer the ultimate question, answer: 42
```

This might not seem like a big deal — if you're going to rebuild the app, you could just as easily change the code to log the values as public — but it has some important benefits:

- You only have to make this change in one place. If your app has lots of log points, this change affects them all.
- It works for code that you don't control. If you need to record private data for library code running in your process, including system frameworks, you can do just that.
- It works on both macOS and iOS.

Configuration Profile

macOS has an extra trick up its sleeve, namely, the `SystemLogging` configuration profile payload (`com.apple.system.logging`). Use this to enable private data without modifying your code. For example, here's an outline of a configuration profile that enables private data for the code shown earlier:

```
...
<dict>
  <key>PayloadContent</key>
  <array>
    <dict>
      ...
      <key>PayloadType</key>
      <string>com.apple.system.logging</string>
      ...
      <key>Subsystems</key>
      <dict>
        <key>com.example.apple-samplecode.PrivateValueLogger</key>
        <dict>
          <key>app</key>
          <dict>
            <key>Enable-Private-Data</key>
            <true/>
          </dict>
        </dict>
      </dict>
    </dict>
  </array>
</dict>
</plist>
```

Critically, this applies not just to your app but to all logging on the system. Continuing the Gatekeeper example from earlier, to see the assessment result, create a profile to enable private data for the `com.apple.xprotect` subsystem and the `xprotect` category.

For detailed information on constructing a configuration profile, see [Device Management](#). To get a head start with that, use [Apple Configurator](#) to create profile with a dummy payload and then use a text editor to modify that payload along the lines shown above.

OSLog

Reply

Posted 5 months ago by eskimo

Add a Comment

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the Apple Developer Forums Participation Agreement.

Developer > Forums			
Platforms <div> <div>iOS</div> <div>iPadOS</div> <div>macOS</div> <div>tvOS</div> <div>watchOS</div> </div>	Topics & Technologies <div> <div>Accessibility</div> <div>Accessories</div> <div>App Extensions</div> <div>App Store</div> <div>Audio & Video</div> <div>Augmented Reality</div> <div>Business</div> <div>Design</div> <div>Distribution</div> <div>Education</div> <div>Fonts</div> <div>Games</div> <div>Health & Fitness</div> <div>In-App Purchase</div> <div>Localization</div> <div>Maps & Location</div> <div>Machine Learning</div> <div>Security</div> <div>Safari & Web</div> </div>	Resources <div> <div>Documentation</div> <div>Curriculum</div> <div>Downloads</div> <div>Forums</div> <div>Videos</div> </div> Support <div> <div>Support Articles</div> <div>Contact Us</div> <div>Bug Reporting</div> <div>System Status</div> </div> Account <div> <div>Apple Developer</div> <div>App Store Connect</div> <div>Certificates, IDs, & Profiles</div> <div>Feedback Assistant</div> </div>	Programs <div> <div>Apple Developer Program</div> <div>Apple Developer Enterprise Program</div> <div>App Store Small Business Program</div> <div>MFi Program</div> <div>News Partner Program</div> <div>Video Partner Program</div> <div>Security Bounty Program</div> <div>Security Research Device Program</div> </div> Events <div> <div>App Accelerators</div> <div>App Store Awards</div> <div>Apple Design Awards</div> <div>Apple Developer Academies</div> <div>Entrepreneur Camp</div> <div>Tech Talks</div> <div>WWDC</div> </div>