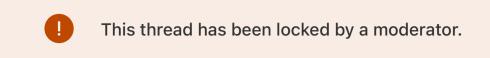# A Peek Behind the NECP Curtain

⚠️ This thread has been locked by a moderator.

👁 207

From time to time the subject of NECP grows up, both here on DevForums and when I'm responding DTS incidents. I've posted about this before but I wanted to collect those tidbits into single coherent post.

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

Share and Enjoy

—

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

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

A Peek Behind the NECP Curtain

NECP stands for Network Extension Control Protocol. It's a subsystem within the macOS networking stack that control which programs have access to which network interfaces. It's vitally important to the Network Extension subsystem, hence the name, but it's used in many different places. Indeed, a very familiar example of its use is the Settings > Mobile Data [1] user interface on iOS.

NECP has no explicit API, although there are APIs that are offer some insight into its state. Continuing the Settings > Mobile Data example above, there is a little-known API, `CTCellularData` in the Core Telephony framework, that returns whether your app has access to WWAN.

Despite having no API, NECP is still relevant to developers. The Settings > Mobile Data example is one place where it affects app developers but it's most important for Network Extension (NE) developers. A key use case for NECP is to prevent VPN loops. When starting an NE provider, the system configures the NECP policy for the NE provider's process to prevent it from using a VPN interface. This means that you can safely open a network connection inside your VPN provider without having to worry about its traffic being accidentally routed back to you. This is why, for example, an NE packet tunnel provider can use any networking API it wants, including BSD Sockets, to run its connection without fear of creating a VPN loop [1].

One place that NECP shows up regularly is the system log. Next time you see a system log entry like this:

```
type: debug
time: 15:02:54.817903+0000
process: Mail
subsystem: com.apple.network
category: connection
message: nw_protocol_socket_set_necp_attributes [C723.1.1:1] setsockopt 39 SO_NECP_ATTRIBUTES
…
```

you'll at least know what the `necp` means (-:

Finally, a lot of NECP infrastructure is in the Darwin open source. As with all things in Darwin, it's fine to poke around and see how your favourite feature works, but **do not incorporate any information you find into your product**. Stuff you uncover by looking in Darwin is not considered API.

[1] Settings > Cellular Data if you speak American (-:

[2] Network Extension providers can call the `createTCPConnection(to:enableTLS:tlsParameters:delegate:)` method to create an `NWTCPConnection` that doesn't run through the tunnel. You can use that if it's convenient but you don't *need* to use it.

`Network Extension`

Reply

Posted 2 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