

Collect PCAPs to Troubleshoot Webex/Jabber Issues on iOS Devices

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background information](#)

[Configure](#)

[Verify](#)

Introduction

This document describes how to collect PCAPs on iOS devices to troubleshoot issues with Jabber and Webex App.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

macOS
Wireshark
Jabber
Webex App

Components Used

The information in this document is based on these software versions:

Wireshark 4.2.2
MacBook Pro with macOS Sonoma 14.5
Xcode 15.4

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background information

Packet captures are a fundamental tool in network analysis and troubleshooting. They allow network administrators and engineers to monitor and analyze the traffic passing through a network, helping to identify issues, optimize performance, and ensure security. Collecting packet captures from an iPhone can be particularly valuable for diagnosing problems with mobile applications like Jabber and Webex App,

iPhones do not natively support packet capturing so this is done using a Mac Terminal and Remote Virtual Interface (RVI) configuration.

Configure

Step 1. Install Xcode and Wireshark:

Ensure you have Xcode installed on your Mac. You can download it from the Mac App Store.

Verify Wireshark is installed on your Mac.

Step 2. Enable Remote Virtual Interface (RVI):

Connect your iPhone to your Mac using a USB cable.

Open **Terminal** on your Mac.

Find the device identifier by running:

```
xcrun xctrace list devices
```

```
~ % xcrun xctrace list devices
[== Devices ==
-M
iPhone (17.6.1) - Connecting (0000803
```

Enable RVI by running:

```
rvictl -s <device-identifier>
```

Replace <device-identifier> with the identifier found in the previous step. This creates a virtual network interface that you can capture traffic from.

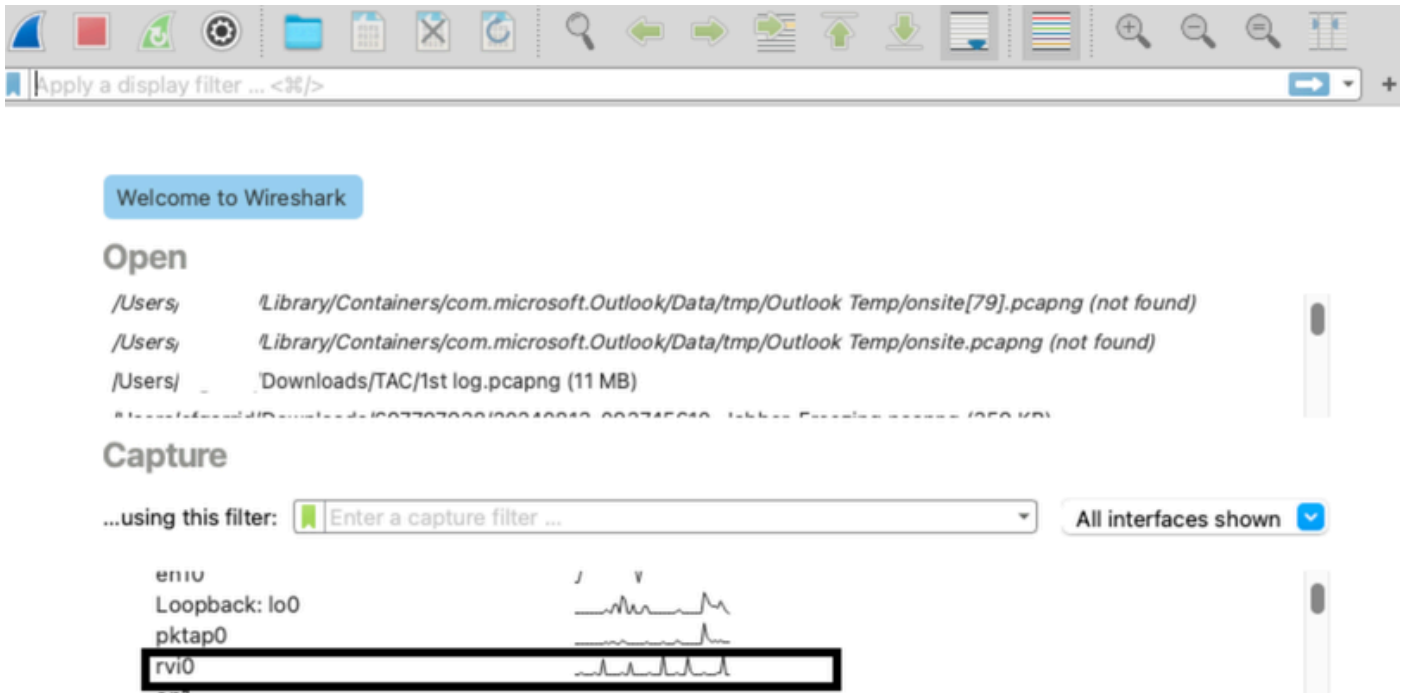
```
~ % rvictl -s 0000803
Starting device 0000803 [SUCCEEDED] with interface rvi0
```

Step 3. Start Capturing with Wireshark:

Open Wireshark.

Look for an interface that starts with rvi0. This is the virtual interface created for your iPhone.

Start a packet capture on the rvi0 interface.



Step 4. Generate Traffic on the iPhone:

Perform the actions on your iPhone to capture traffic (for example: browse, app usage).

Stop Capturing:

Stop the capture in Wireshark after you have collected the necessary data.

Disable RVI:

In Terminal, run:

```
rvictl -x <device-identifier>
```

Replace <device-identifier> with the identifier used previously.

```

-M ~ % rvictl -x 0000803

Stopping device 0000803 [SUCCEEDED]

```

Verify

Once you collect the Wireshark packet capture, you can troubleshoot the issue.