Patch Different *OS

John McIntosh

CL CLEARSEC LABS

Objective by the Sea

version 7.0

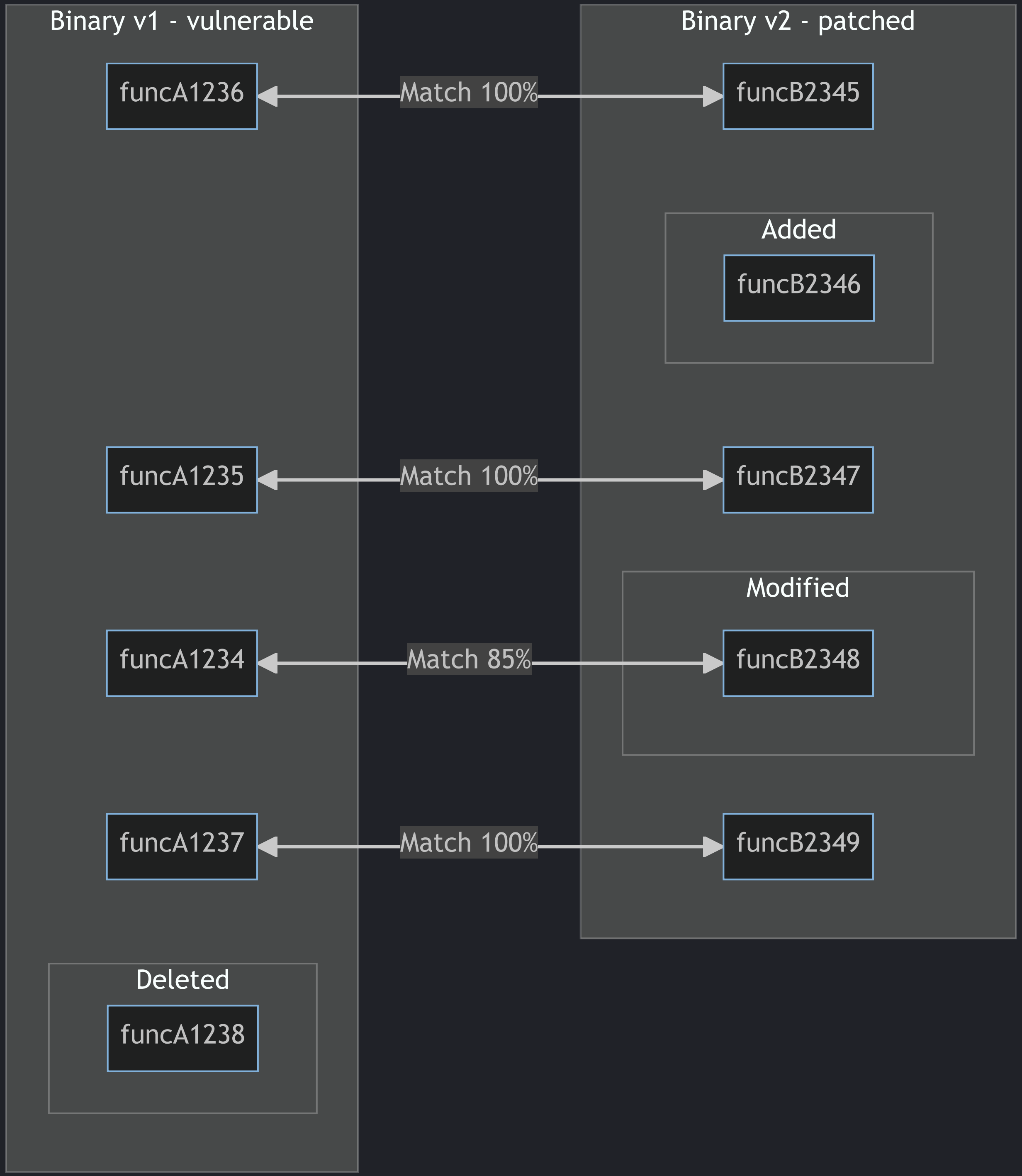
whoami

- @clearbluejar
- security researcher
- founder @clearseclabs
- open-source dev
- speaker / blogger
- CVE North Stars

Binary Diffing

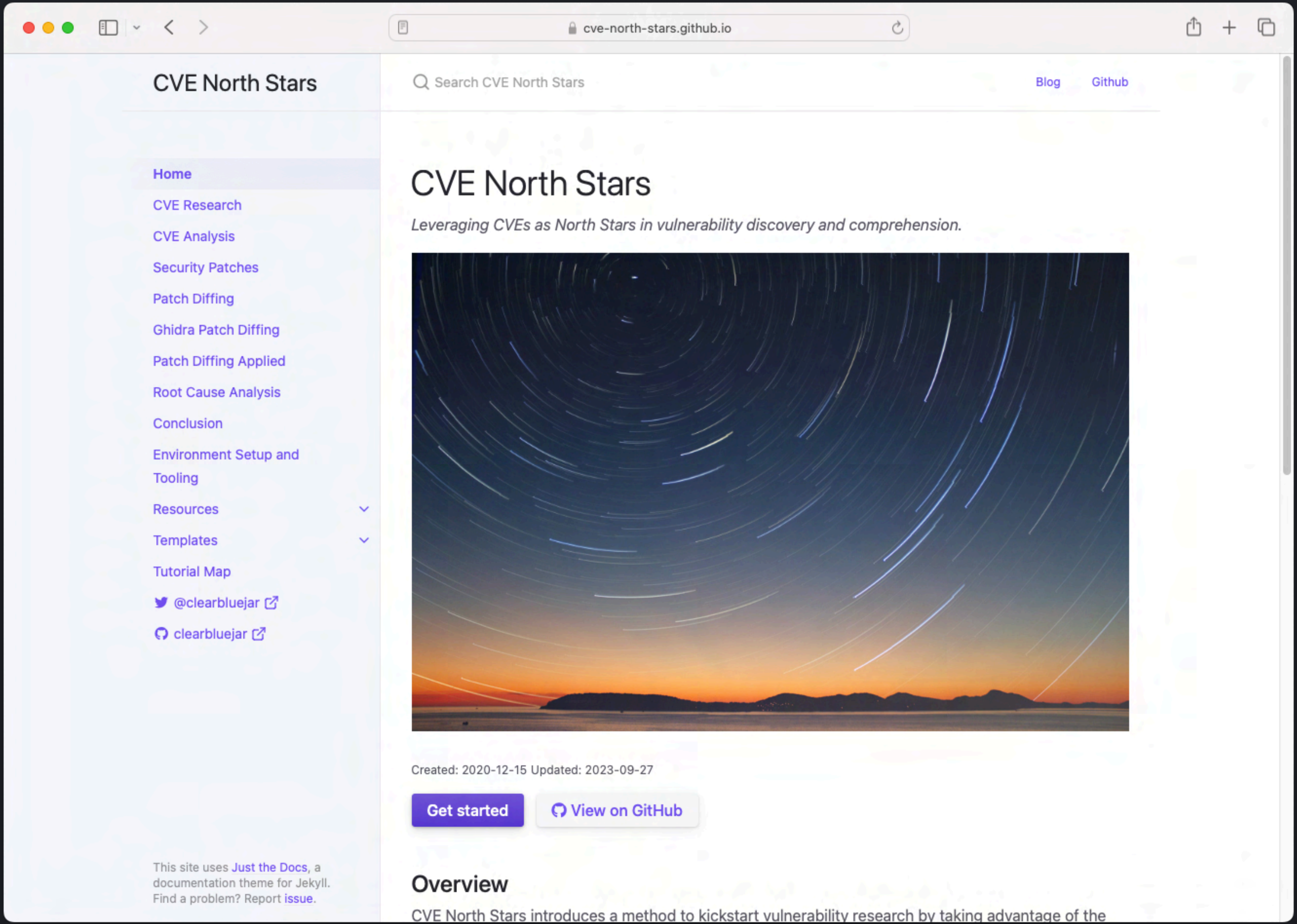
Purpose

Find the *added*, *deleted*, and *modified* functions and data between two binaries.



Why?

- **Reverse Engineering**:
  - Port previous reverse engineering work to new binary
- **Vulnerability Research**:
  - Determine whether or not a security update actually addressed the root issue, or was a shallow fix
- **Malware analysis**:
  - Find similar code in one malware set to another, or correlate new versions of malware to old


cve-north-stars.github.io

CVE North Stars

Search CVE North Stars

Blog Github

- Home
- CVE Research
- CVE Analysis
- Security Patches
- Patch Diffing
- Ghidra Patch Diffing
- Patch Diffing Applied
- Root Cause Analysis
- Conclusion
- Environment Setup and Tooling
- Resources
- Templates
- Tutorial Map
- @clearbluejar
- clearbluejar

This site uses Just the Docs, a documentation theme for Jekyll. Find a problem? Report issue.

CVE North Stars

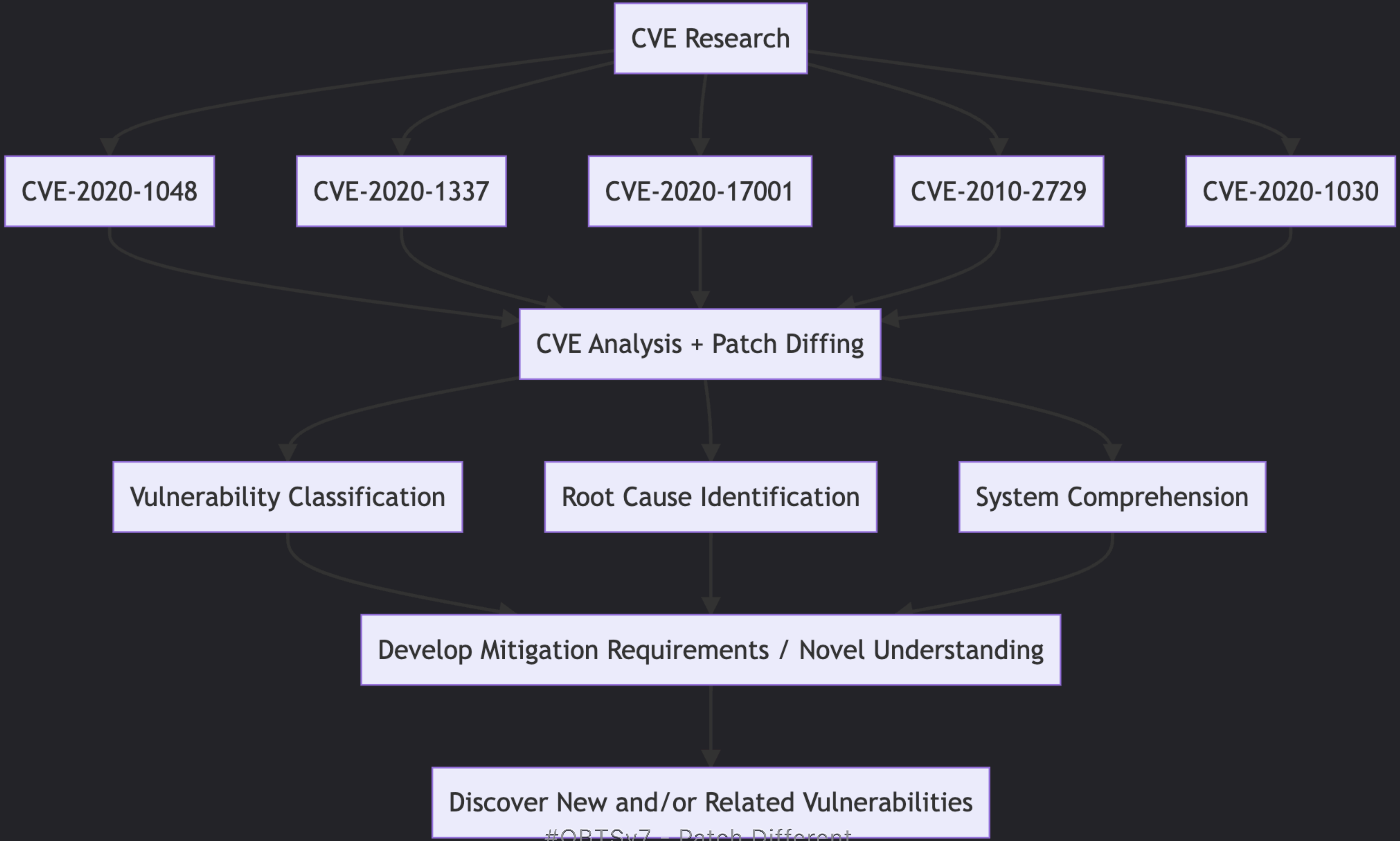
Leveraging CVEs as North Stars in vulnerability discovery and comprehension.

Created: 2020-12-15 Updated: 2023-09-27

Get started | View on GitHub

Overview

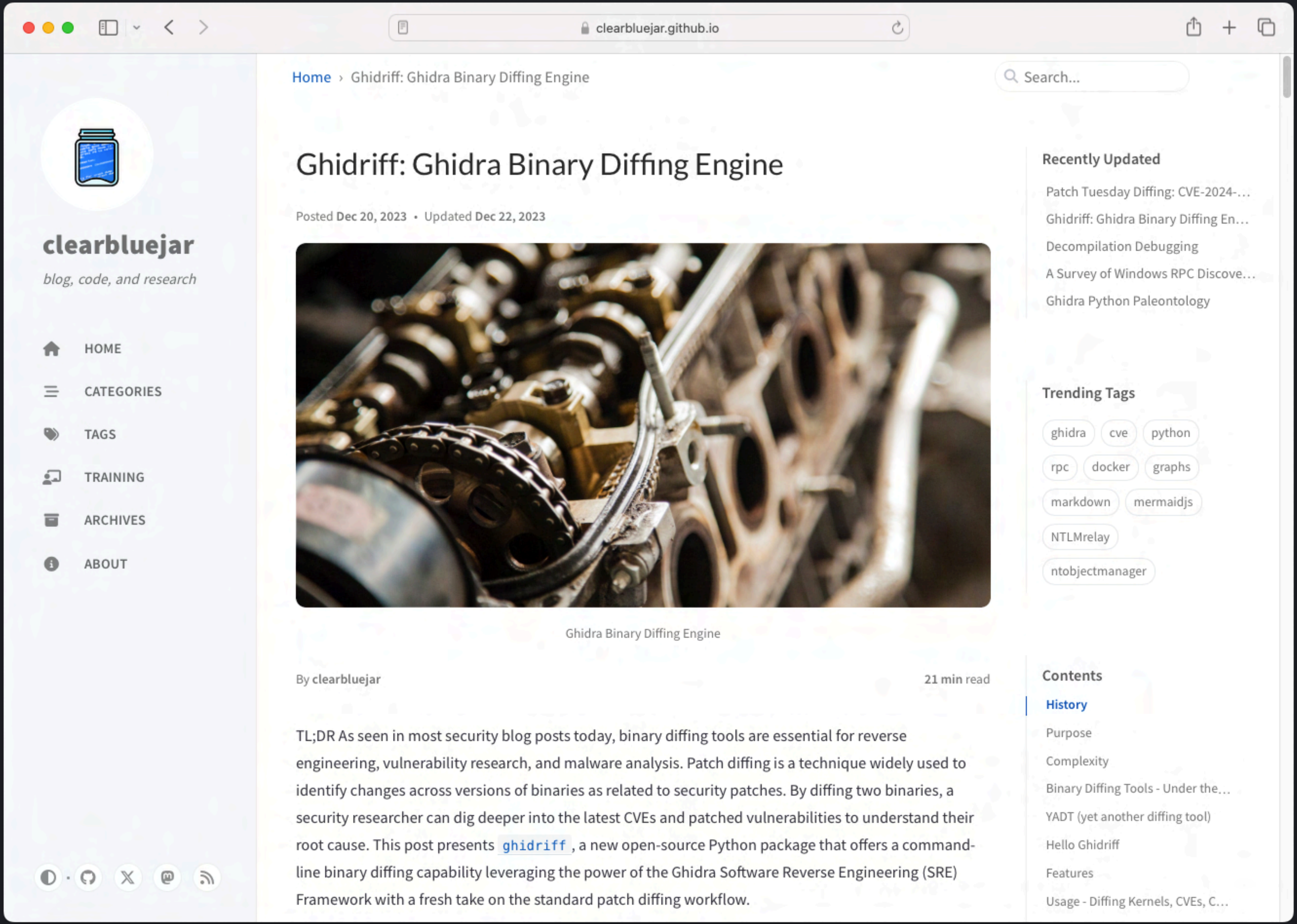
CVE North Stars introduces a method to kickstart vulnerability research by taking advantage of the


CVE Research
CVE-2020-1048
CVE-2020-1337
CVE-2020-17001
CVE-2010-2729
CVE-2020-1030
CVE Analysis + Patch Diffing
Vulnerability Classification
Root Cause Identification
System Comprehension
Develop Mitigation Requirements / Novel Understanding
Discover New and/or Related Vulnerabilities

Patch Diffing

“
- is a single source of truth when there is no other
- provides clarity in root cause analysis
- can improve your competence in reverse engineering and understanding of modern vulnerabilities
”

CVE North Stars



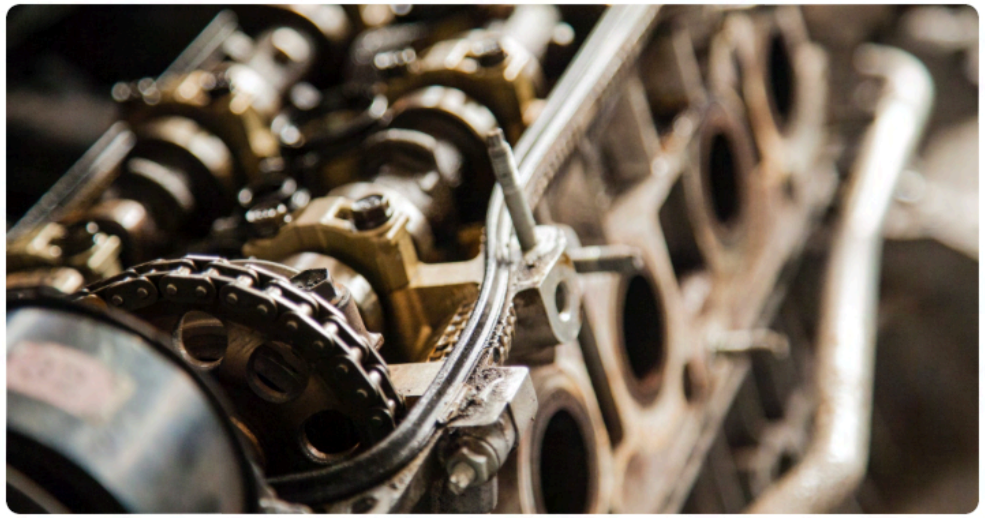
Home › Ghidriff: Ghidra Binary Diffing Engine

clearbluejar

blog, code, and research

Ghidriff: Ghidra Binary Diffing Engine

Posted **Dec 20, 2023** • Updated **Dec 22, 2023**



Ghidra Binary Diffing Engine

By **clearbluejar** **21 min** read

TL;DR As seen in most security blog posts today, binary diffing tools are essential for reverse engineering, vulnerability research, and malware analysis. Patch diffing is a technique widely used to identify changes across versions of binaries as related to security patches. By diffing two binaries, a security researcher can dig deeper into the latest CVEs and patched vulnerabilities to understand their root cause. This post presents `ghidriff`, a new open-source Python package that offers a command-line binary diffing capability leveraging the power of the Ghidra Software Reverse Engineering (SRE) Framework with a fresh take on the standard patch diffing workflow.

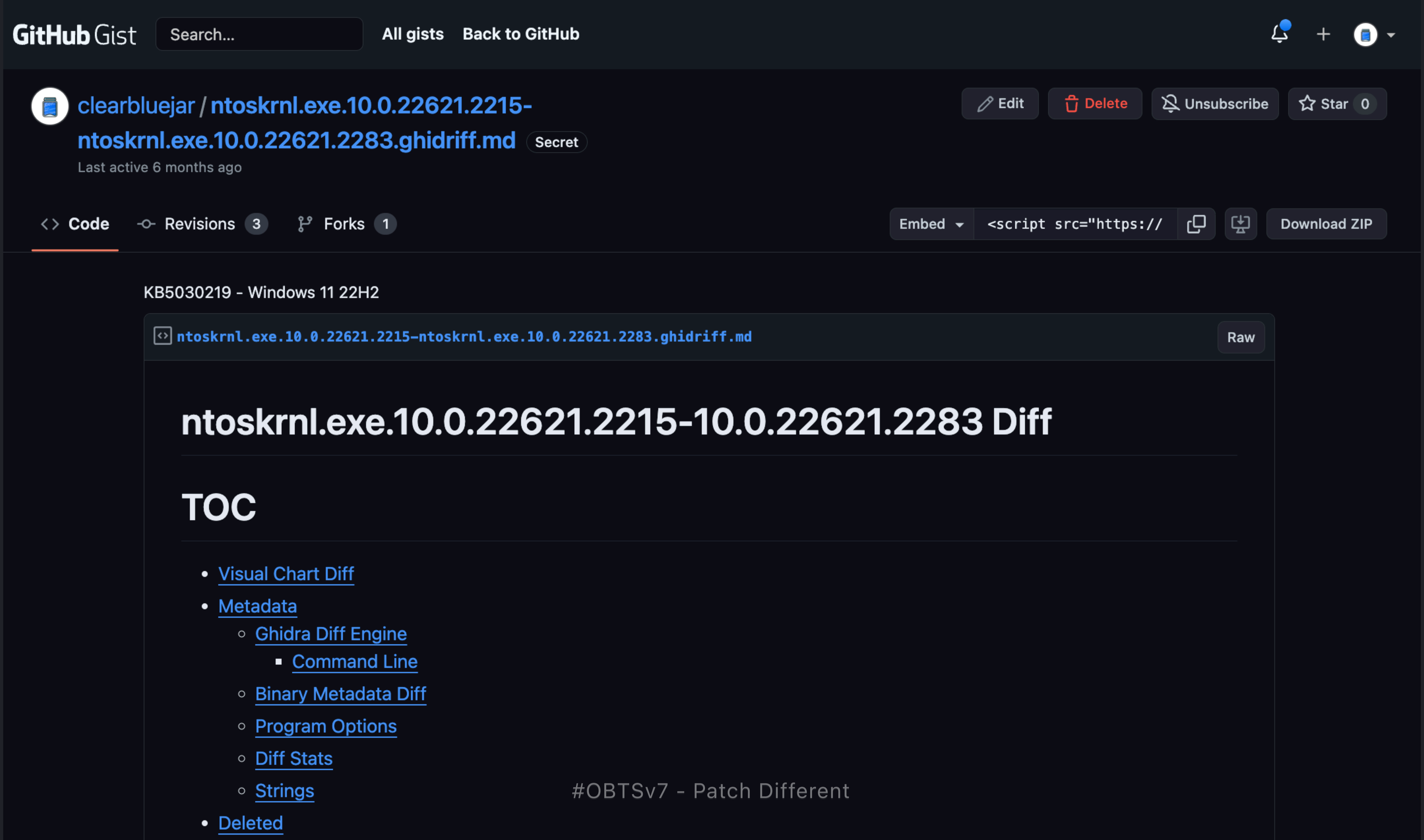
Recently Updated

- Patch Tuesday Diffing: CVE-2024-…
- Ghidriff: Ghidra Binary Diffing En…
- Decompilation Debugging
- A Survey of Windows RPC Discove…
- Ghidra Python Paleontology

Trending Tags

ghidra cve python rpc docker graphs markdown mermaidjs NTLMrelay ntobjectmanager

Contents



clearbluejar / ntoskrnl.exe.10.0.22621.2215-ntoskrnl.exe.10.0.22621.2283.ghidriff.md Secret

Last active 6 months ago

KB5030219 - Windows 11 22H2

ntoskrnl.exe.10.0.22621.2215-ntoskrnl.exe.10.0.22621.2283.ghidriff.md

ntoskrnl.exe.10.0.22621.2215-10.0.22621.2283 Diff

TOC

clearbluejar
@clearbluejar

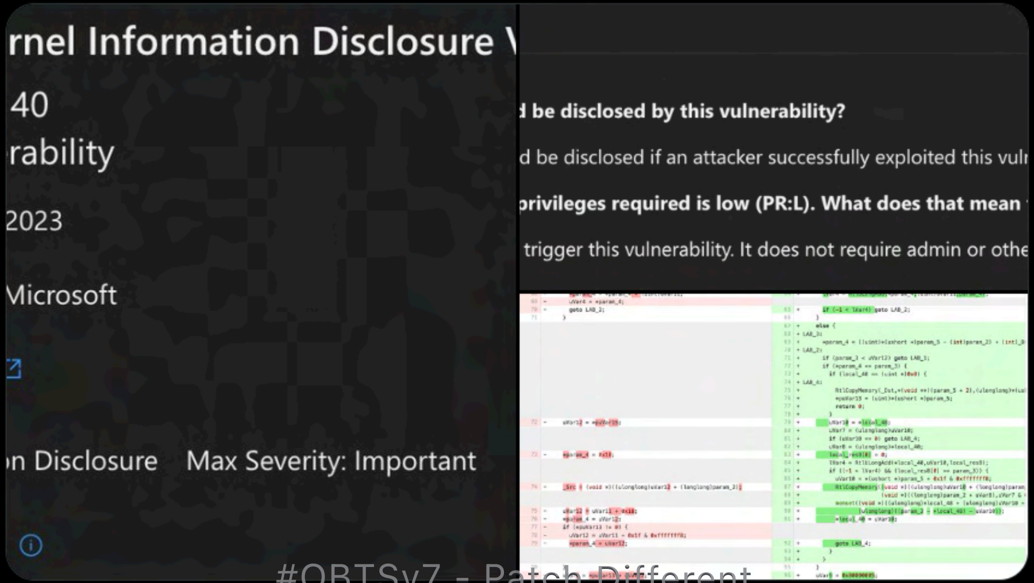
Patch diffing when you have no blog post, no Github poc, only binaries!
#patchdiffinginthedark

Let's try CVE-2023-38140 with #ghidriff: gist.github.com/clearbluejar/b
...

Windows kernel, info disclosure, uninitialized memory...
Maybe there were some new calls to memset? 🤔
gist.github.com/clearbluejar/b...

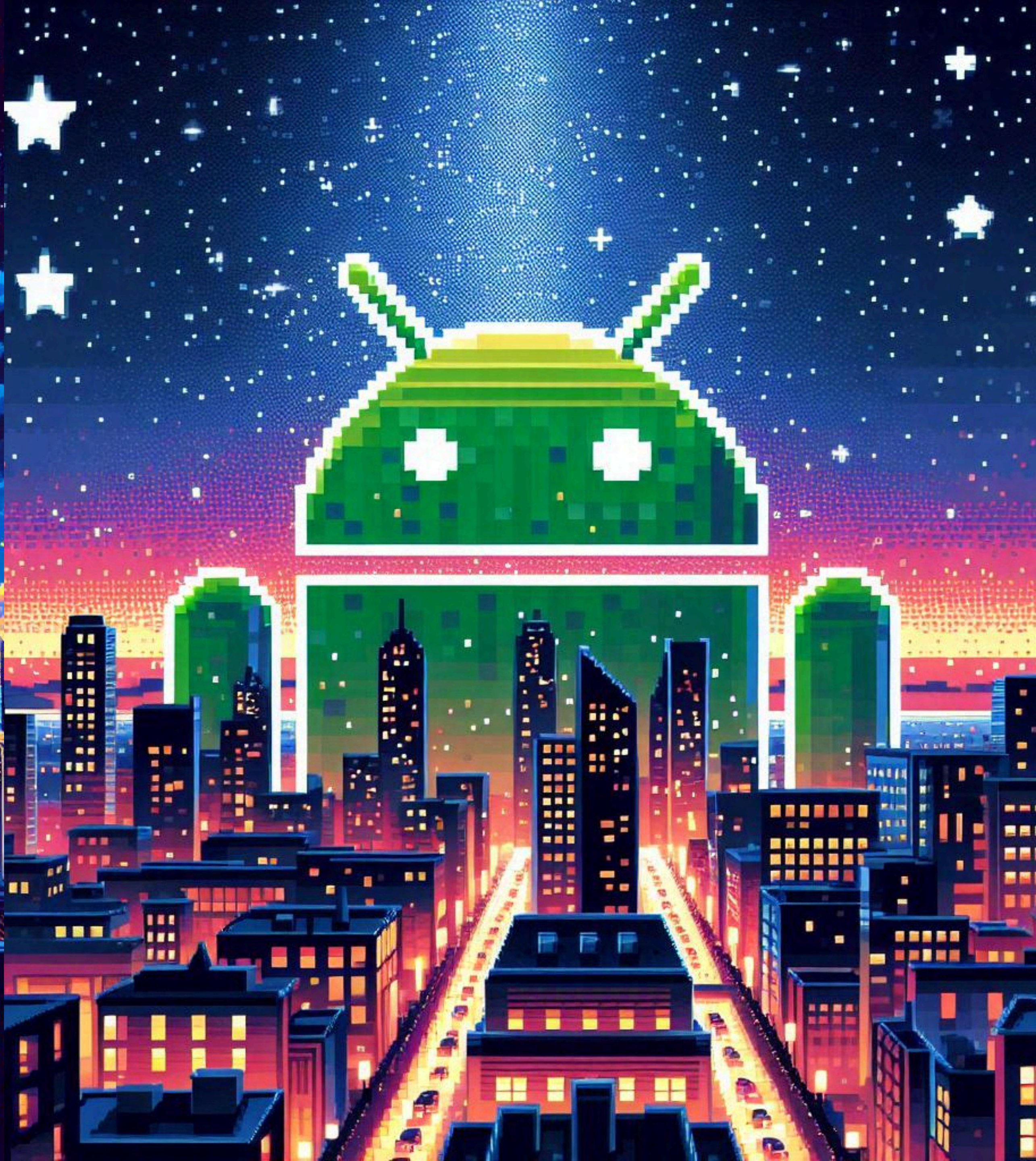
Yep. With patch diffing, we can step into the light. 🧐
gist.github.com/clearbluejar/b...

If you prefer a side by side view:
diffpreview.github.io/?b3bd4cd18600f...


rnel Information Disclosure \
40
rability
2023
Microsoft
n Disclosure Max Severity: Important

d be disclosed by this vulnerability?
d be disclosed if an attacker successfully exploited this vul
privileges required is low (PR:L). What does that mean
trigger this vulnerability. It does not require admin or othe

Last edited 1:23 AM · Oct 10, 2023 · **15.3K** Views



What's Different on *OS?

Different on *OS

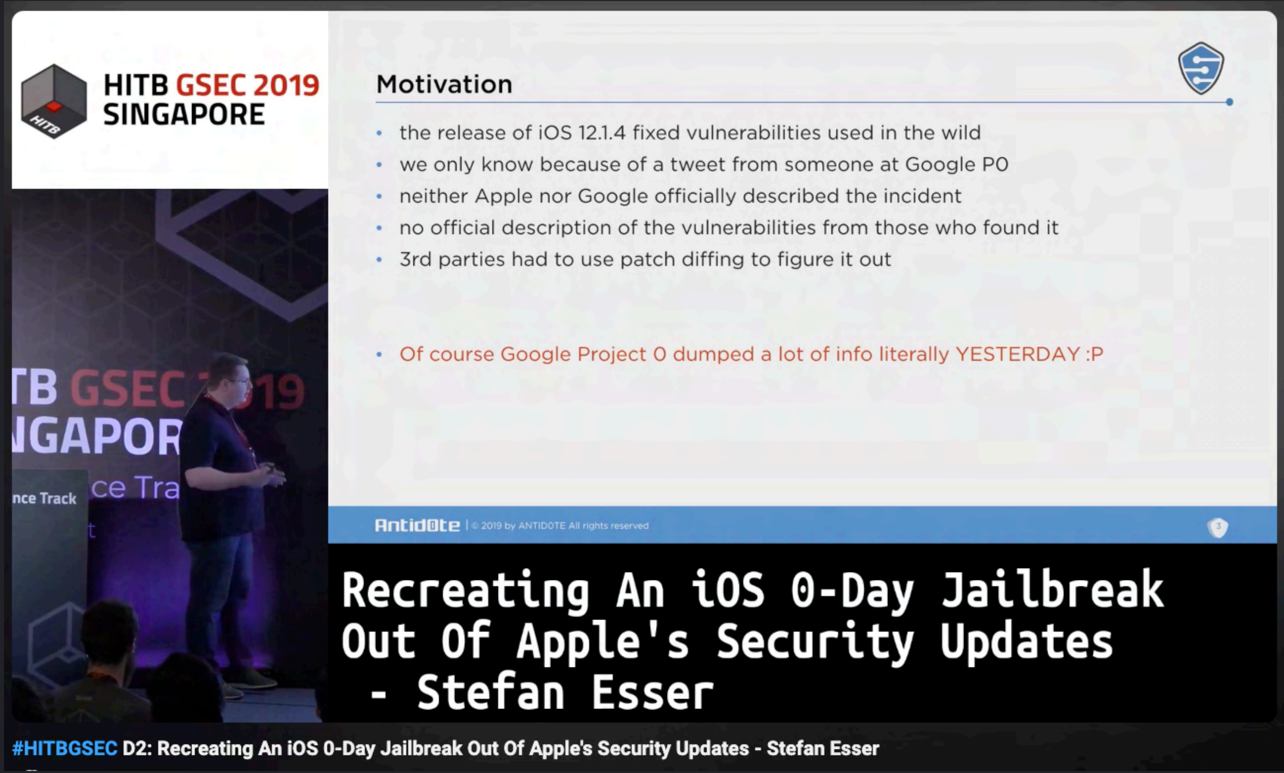
- Binary Acquisition
- Apple Ecosystem
- File System Layout

Basic Patch Diffing the Same

- Find vulnerable binary
- Find patched binary
- SRE tooling / Analysis
- Diff
- Root Cause

Previous Apple Diffing Research (~2019):

- Recreatingan OS 0day Jailbreak Out of Apple Security Updates
- A Story of an Apple Patch



Motivation

- the release of iOS 12.1.4 fixed vulnerabilities used in the wild
- we only know because of a tweet from someone at Google P0
- neither Apple nor Google officially described the incident
- no official description of the vulnerabilities from those who found it
- 3rd parties had to use patch diffing to figure it out
- Of course Google Project 0 dumped a lot of info literally YESTERDAY :P

Antid0te | © 2019 by ANTIDOTE All rights reserved

Recreating An iOS 0-Day Jailbreak Out Of Apple's Security Updates - Stefan Esser

#HITBGSEC D2: Recreating An iOS 0-Day Jailbreak Out Of Apple's Security Updates - Stefan Esser

What's *Different* For This Talk?

```diff
- expert Apple researchers
+ researcher
- IDA
- Diaphora
+ Ghidra
+ ghidriff
+ ipsw
+ CVEs++
```

#OBTSv7 - Patch Different

#OBTSv7 - Patch Different

Talk Overview

- High Level Diffing on *OS
- Apple Security Updates
- Diffing IPSW
- Diffing *OS Binaries
- Root Causing CVEs

High Level Diffing on *OS

*OS Diffing Strategy

- Compare binares across OS versions (IPSWs)
- Ignore most changes
- Focus on a single CVE

High Level Patch Diffing on *OS

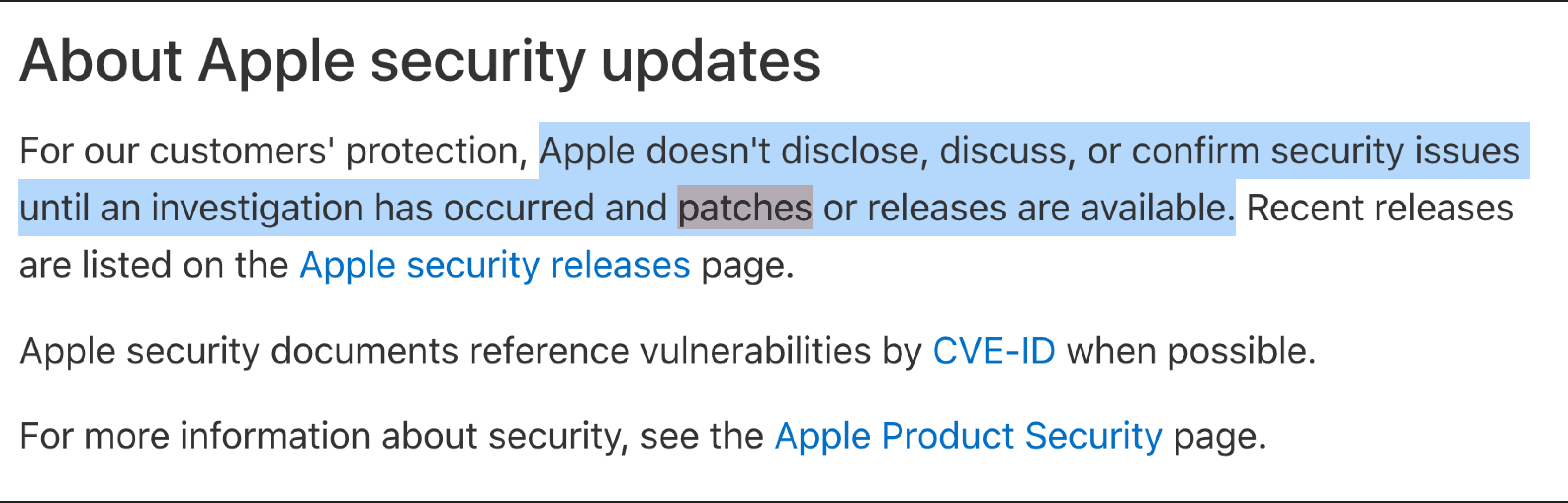
1. Identify a CVE of interest
2. Download corresponding IPSW update and N-1
3. Determine changes for update
4. Map binaries to CVE
5. Extract the related file(s)
6. Diff the binaries
7. Root cause the vulnerability

Where Do We Find CVES?

Apple Security Releases

Apple seems to run a "Patch Monday" instead of Tuesday

Double Edged Sword



About Apple security updates

For our customers' protection, Apple doesn't disclose, discuss, or confirm security issues until an investigation has occurred and patches or releases are available. Recent releases are listed on the Apple security releases page.

Apple security documents reference vulnerabilities by CVE-ID when possible.

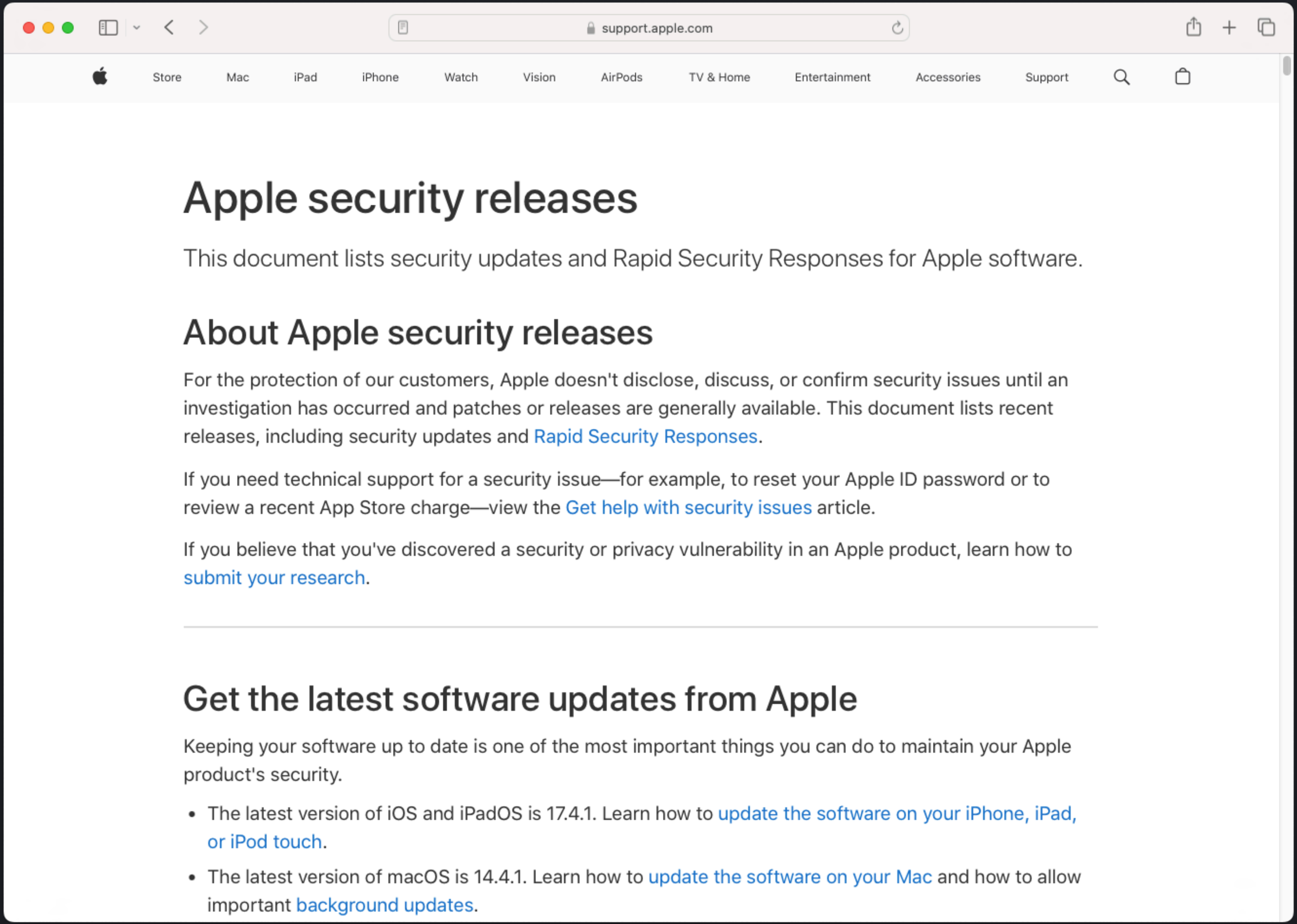
For more information about security, see the Apple Product Security page.

“

… a security patch is a form of vulnerability disclosure that is always public.

”

CVE North Stars



Apple security releases

This document lists security updates and Rapid Security Responses for Apple software.

About Apple security releases

For the protection of our customers, Apple doesn't disclose, discuss, or confirm security issues until an investigation has occurred and patches or releases are generally available. This document lists recent releases, including security updates and Rapid Security Responses.

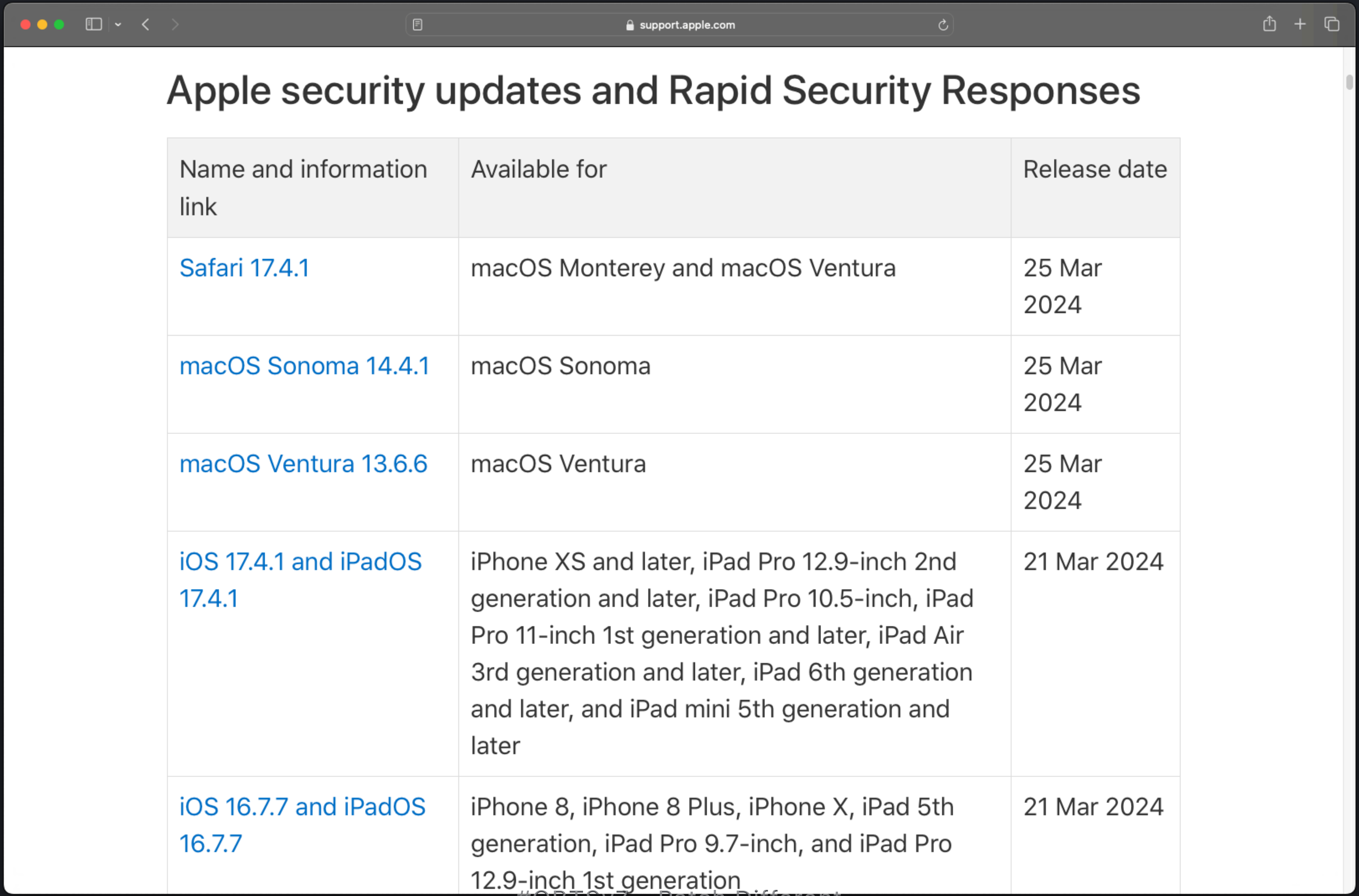
If you need technical support for a security issue—for example, to reset your Apple ID password or to review a recent App Store charge—view the Get help with security issues article.

If you believe that you've discovered a security or privacy vulnerability in an Apple product, learn how to submit your research.

Get the latest software updates from Apple

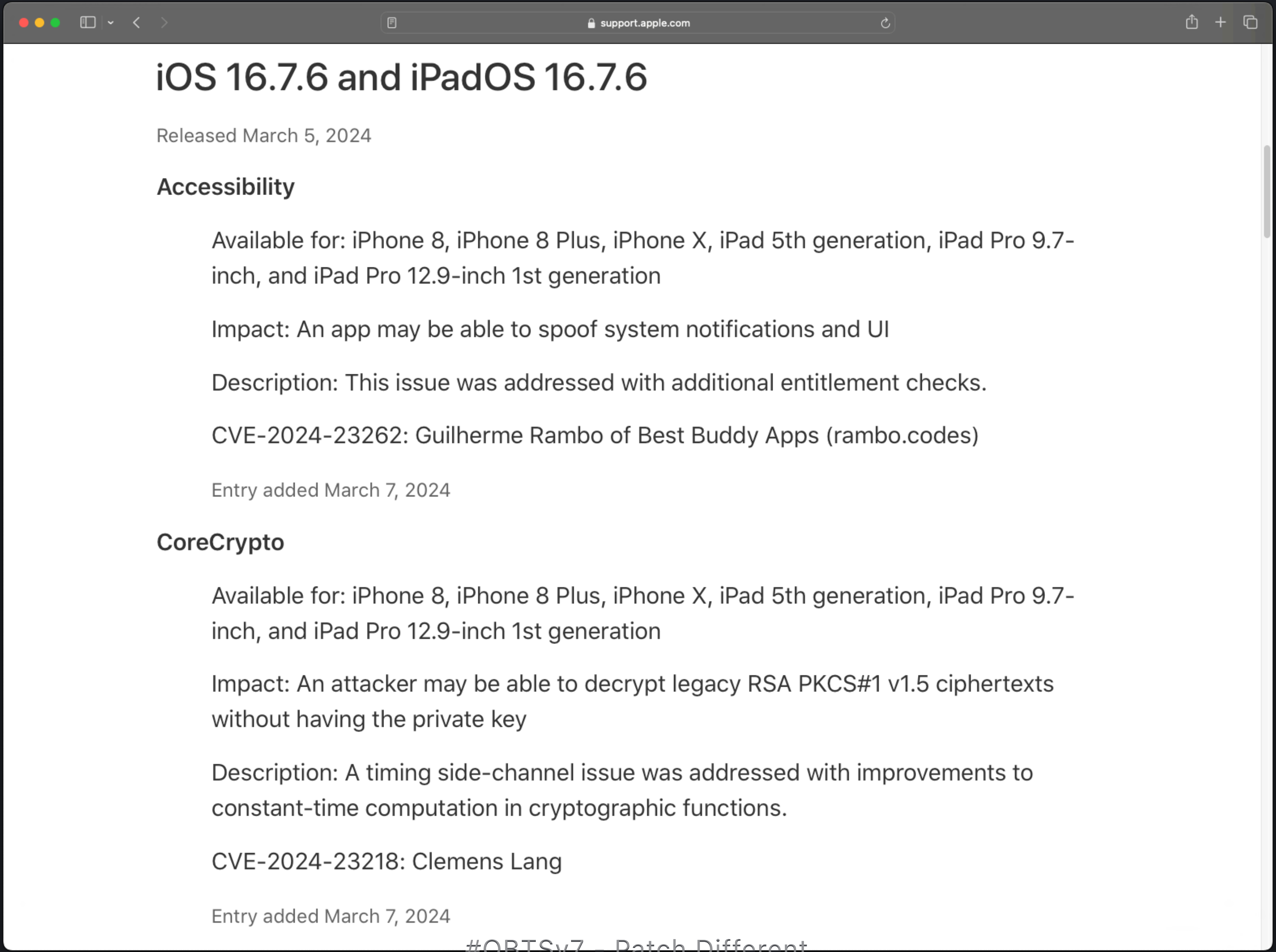
Keeping your software up to date is one of the most important things you can do to maintain your Apple product's security.

- The latest version of iOS and iPadOS is 17.4.1. Learn how to update the software on your iPhone, iPad, or iPod touch.
- The latest version of macOS is 14.4.1. Learn how to update the software on your Mac and how to allow important background updates.


support.apple.com

Apple security updates and Rapid Security Responses

Name and information link	Available for	Release date
Safari 17.4.1	macOS Monterey and macOS Ventura	25 Mar 2024
macOS Sonoma 14.4.1	macOS Sonoma	25 Mar 2024
macOS Ventura 13.6.6	macOS Ventura	25 Mar 2024
iOS 17.4.1 and iPadOS 17.4.1	iPhone XS and later, iPad Pro 12.9-inch 2nd generation and later, iPad Pro 10.5-inch, iPad Pro 11-inch 1st generation and later, iPad Air 3rd generation and later, iPad 6th generation and later, and iPad mini 5th generation and later	21 Mar 2024
iOS 16.7.7 and iPadOS 16.7.7	iPhone 8, iPhone 8 Plus, iPhone X, iPad 5th generation, iPad Pro 9.7-inch, and iPad Pro 12.9-inch 1st generation	21 Mar 2024



support.apple.com

iOS 16.7.6 and iPadOS 16.7.6

Released March 5, 2024

Accessibility

Available for: iPhone 8, iPhone 8 Plus, iPhone X, iPad 5th generation, iPad Pro 9.7-inch, and iPad Pro 12.9-inch 1st generation

Impact: An app may be able to spoof system notifications and UI

Description: This issue was addressed with additional entitlement checks.

CVE-2024-23262: Guilherme Rambo of Best Buddy Apps (rambo.codes)

Entry added March 7, 2024

CoreCrypto

Available for: iPhone 8, iPhone 8 Plus, iPhone X, iPad 5th generation, iPad Pro 9.7-inch, and iPad Pro 12.9-inch 1st generation

Impact: An attacker may be able to decrypt legacy RSA PKCS#1 v1.5 ciphertexts without having the private key

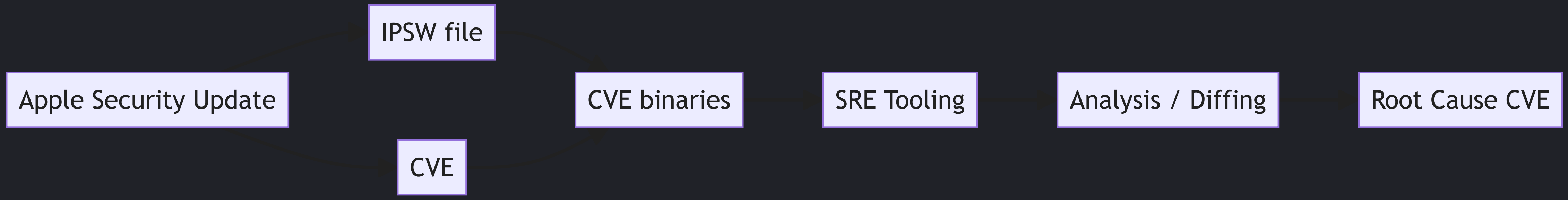
Description: A timing side-channel issue was addressed with improvements to constant-time computation in cryptographic functions.

CVE-2024-23218: Clemens Lang

Entry added March 7, 2024

Apple Security Updates

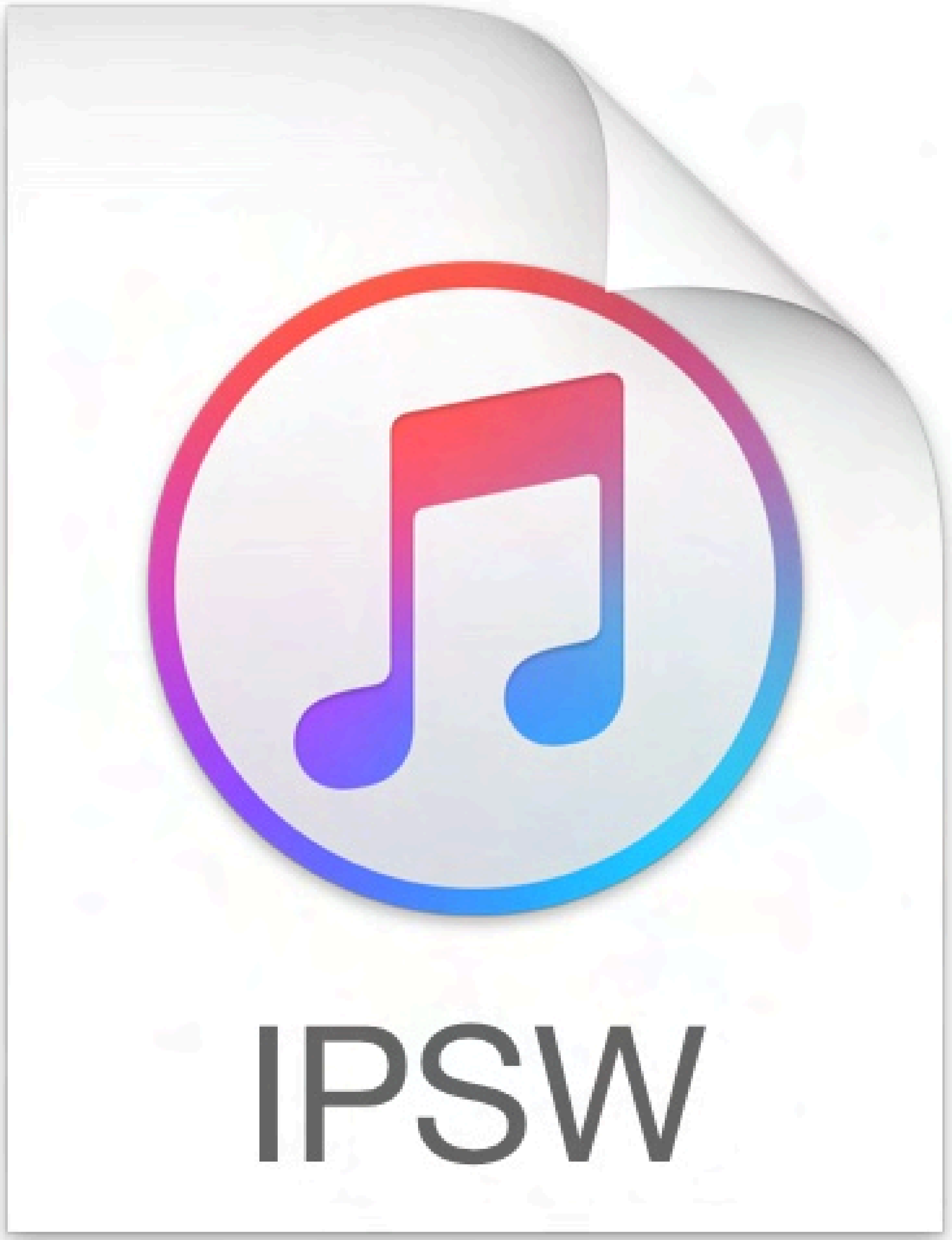
- a list of OS provided updates and fixes to CVEs.
- use as a starting point for looking for CVEs to diff
- Each IPSW update correlates to specific OS build



High Level Patch Diffing on *OS

1. Identify a CVE of interest
2. Download corresponding IPSW update and N-1
3. Determine changes for update
4. Map binaries to CVE
5. Extract the related file(s)
6. Diff the binaries
7. Root cause the vulnerability

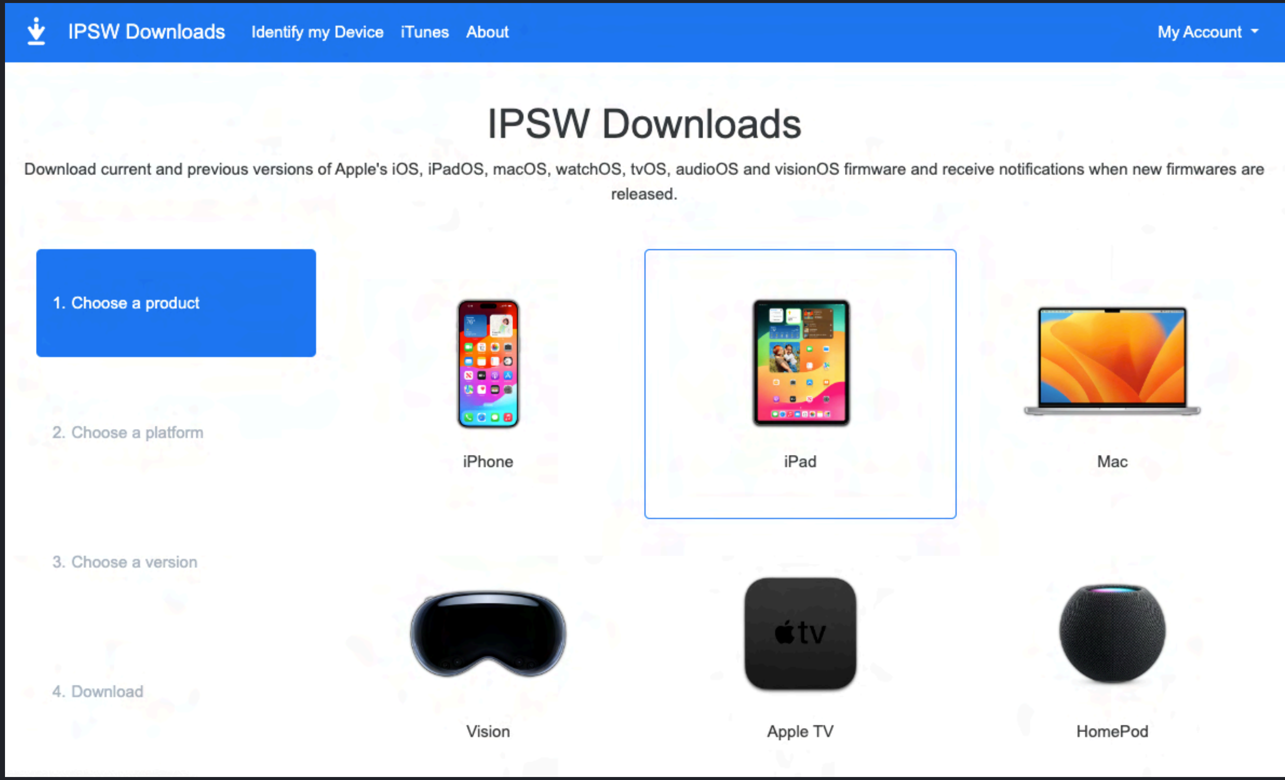
IPSW



An **IPSW** (iPhone Software) file is Apple's file format for delivering OS firmware updates.

You can use a 3rd party website to get the links…

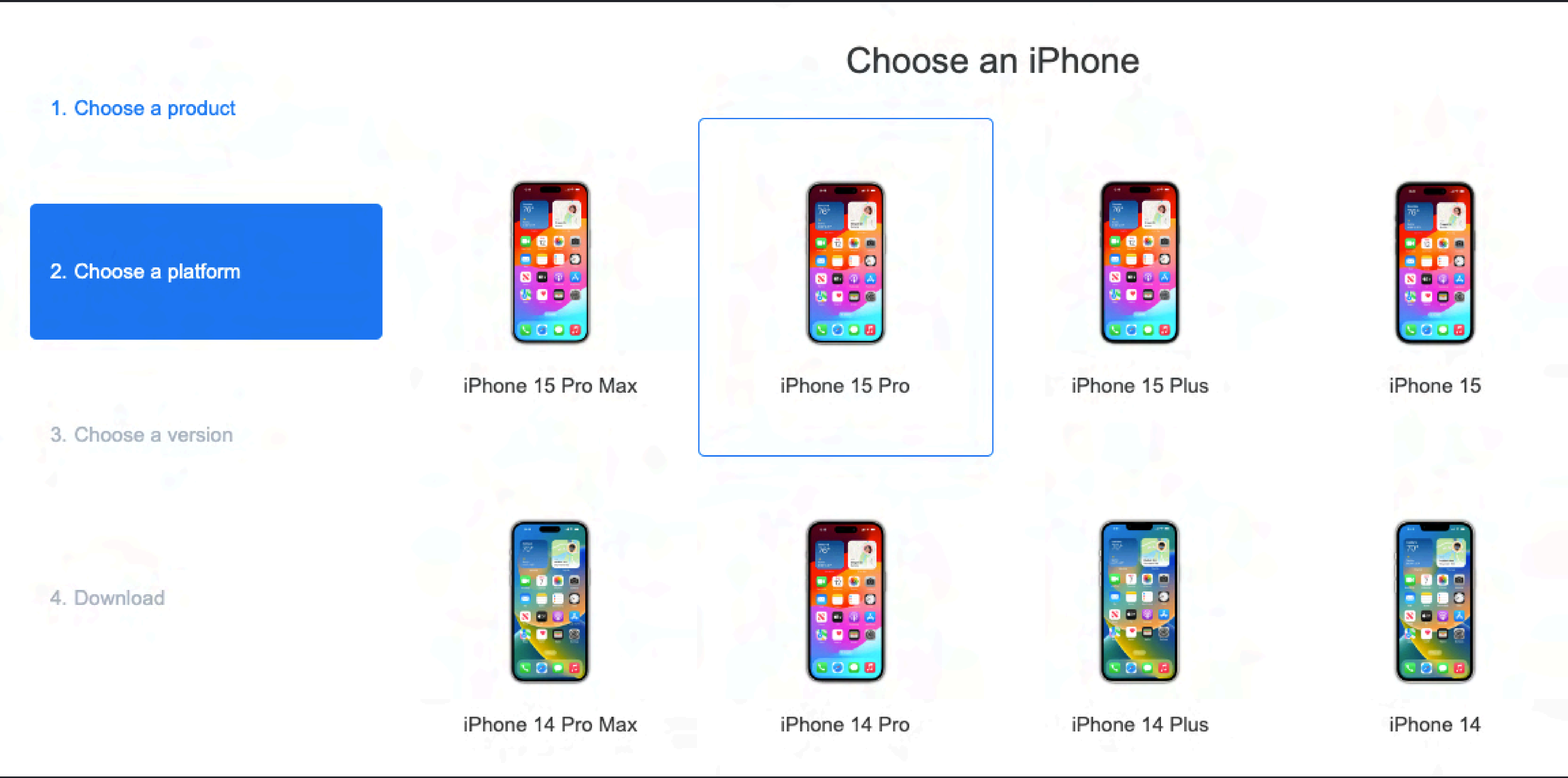
ipsw.me



IPSW Downloads

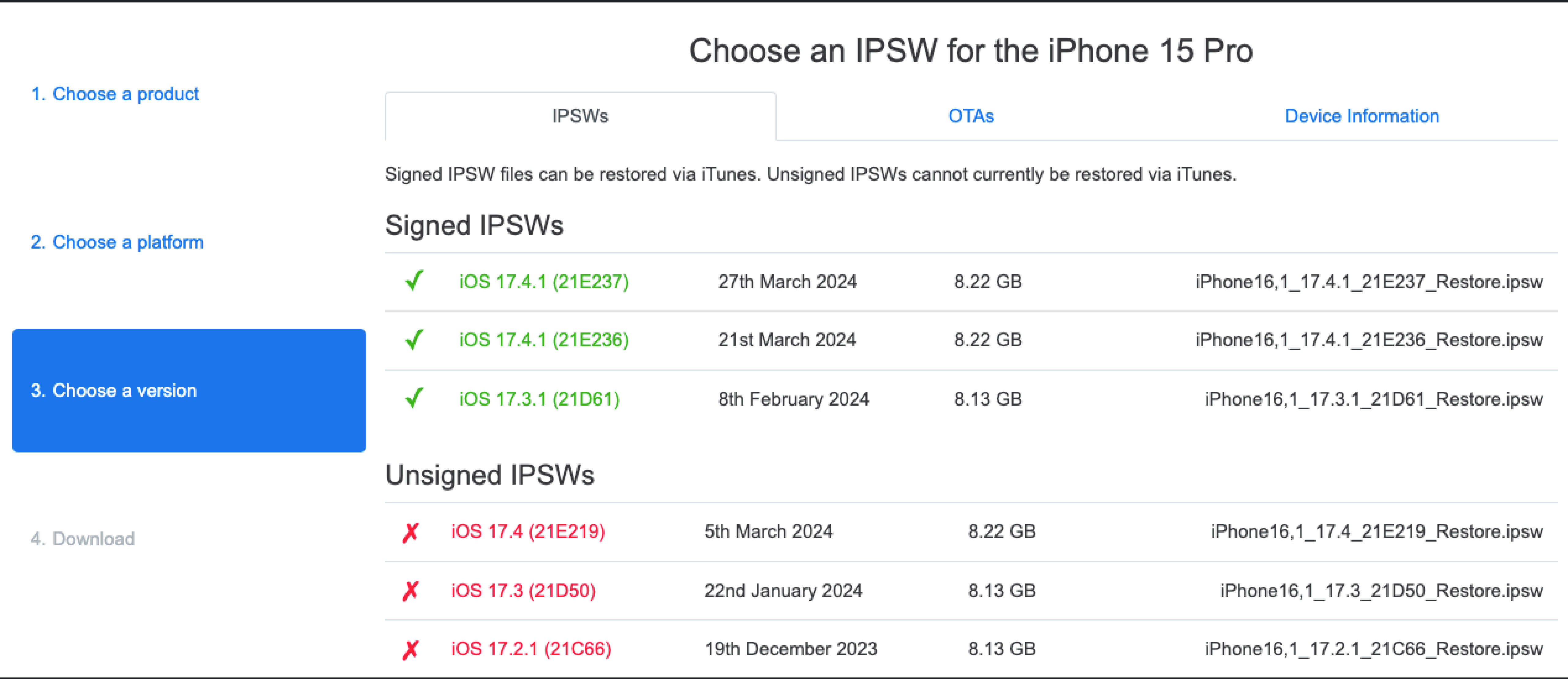
Download current and previous versions of Apple's iOS, iPadOS, macOS, watchOS, tvOS, audioOS and visionOS firmware and receive notifications when new firmwares are released.

1. Choose a product
2. Choose a platform
3. Choose a version
4. Download



Choose an iPhone

1. Choose a product
2. Choose a platform
3. Choose a version
4. Download



Choose an IPSW for the iPhone 15 Pro

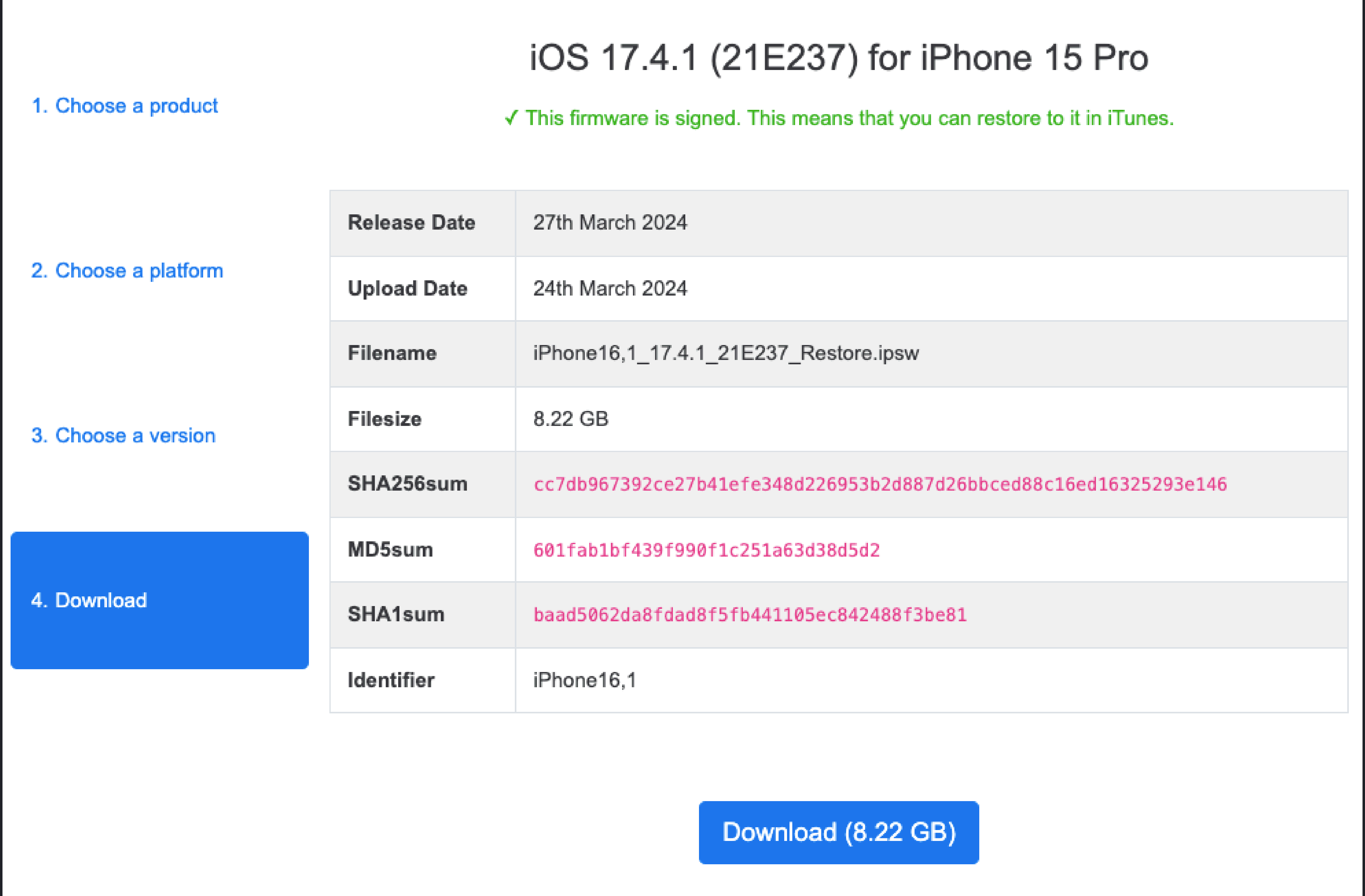
Signed IPSW files can be restored via iTunes. Unsigned IPSWs cannot currently be restored via iTunes.

Signed IPSWs

✓	iOS 17.4.1 (21E237)	27th March 2024	8.22 GB	iPhone16,1_17.4.1_21E237_Restore.ipsw
✓	iOS 17.4.1 (21E236)	21st March 2024	8.22 GB	iPhone16,1_17.4.1_21E236_Restore.ipsw
✓	iOS 17.3.1 (21D61)	8th February 2024	8.13 GB	iPhone16,1_17.3.1_21D61_Restore.ipsw

Unsigned IPSWs

✗	iOS 17.4 (21E219)	5th March 2024	8.22 GB	iPhone16,1_17.4_21E219_Restore.ipsw
✗	iOS 17.3 (21D50)	22nd January 2024	8.13 GB	iPhone16,1_17.3_21D50_Restore.ipsw
✗	iOS 17.2.1 (21C66)	19th December 2023	8.13 GB	iPhone16,1_17.2.1_21C66_Restore.ipsw



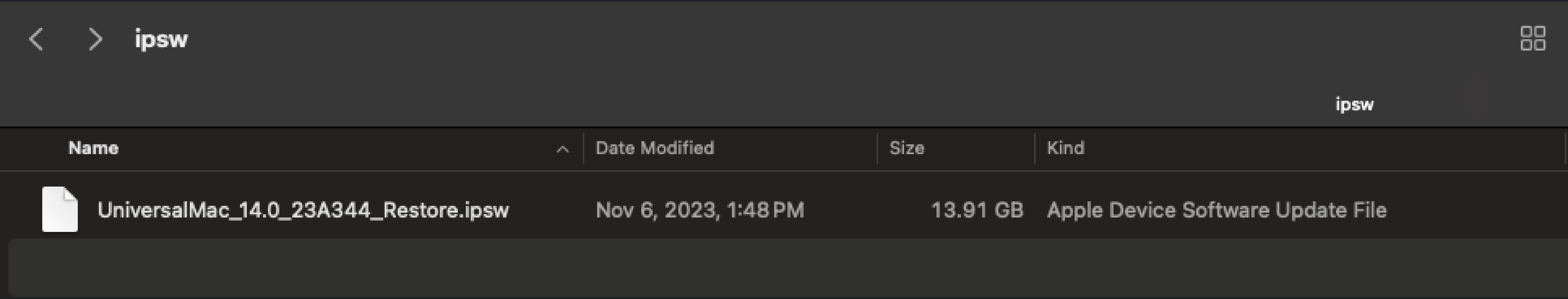
iOS 17.4.1 (21E237) for iPhone 15 Pro

✓ This firmware is signed. This means that you can restore to it in iTunes.

Release Date	27th March 2024
Upload Date	24th March 2024
Filename	iPhone16,1_17.4.1_21E237_Restore.ipsw
Filesize	8.22 GB
SHA256sum	`cc7db967392ce27b41efe348d226953b2d887d26bbced88c16ed16325293e146`
MD5sum	`601fab1bf439f990f1c251a63d38d5d2`
SHA1sum	`baad5062da8fdad8f5fb441105ec842488f3be81`
Identifier	iPhone16,1

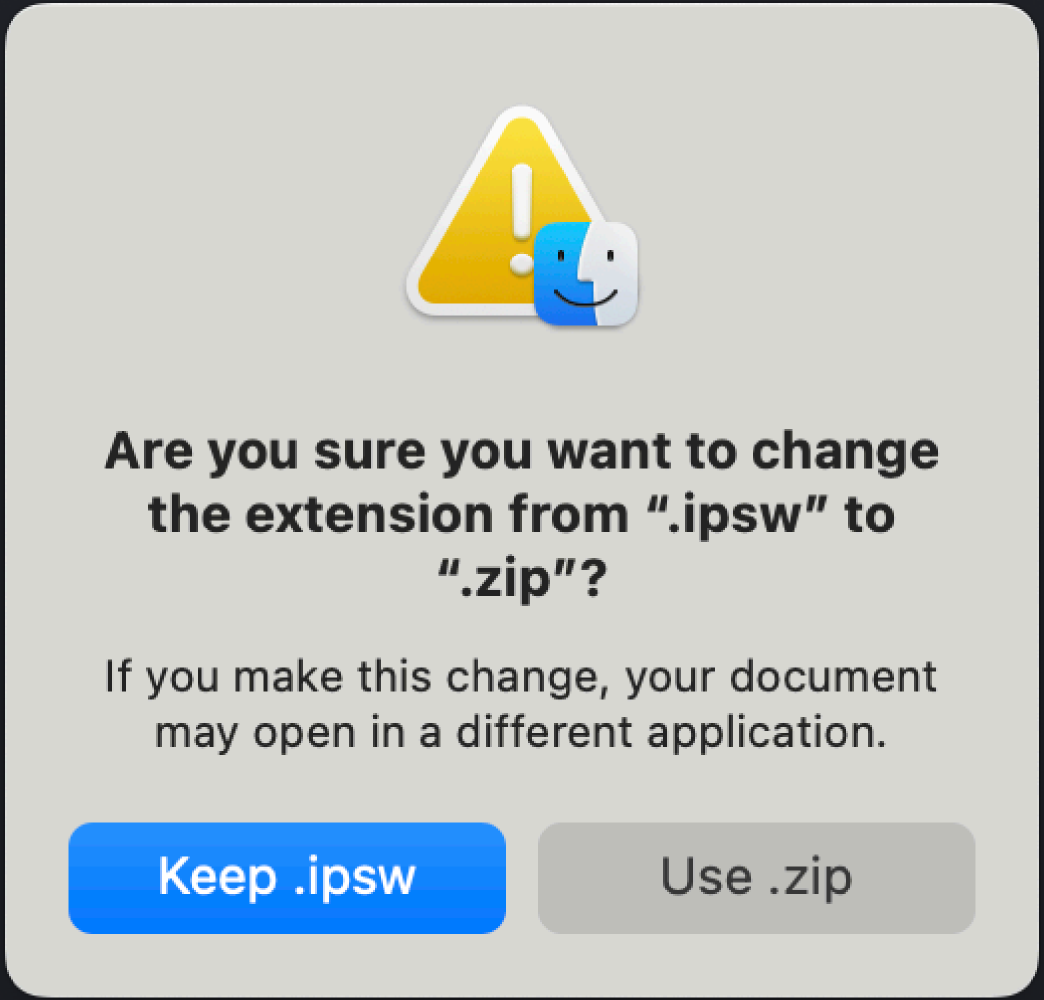
https://updates.cdn-apple.com/2024WinterFCS/fullrestores/052-80037/75771938-2991-445D-96FE-F6ACD03EF744/iPhone15,4_17.4.1_21E237_Restore.ipsw

```
Device: iPhone15,4
Version: 14.4
Build: 21E237
```


ipsw

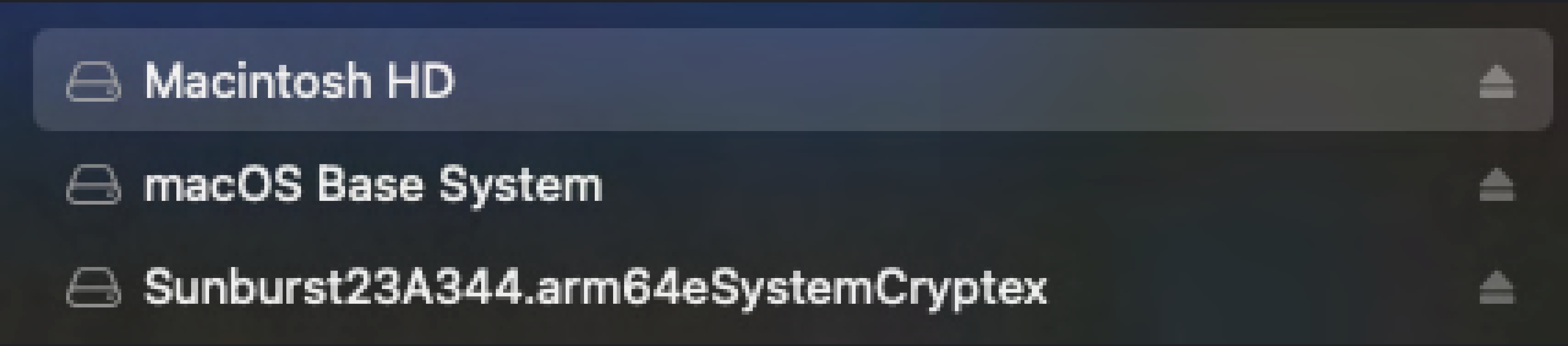
ipsw

Name	Date Modified	Size	Kind
UniversalMac_14.0_23A344_Restore.ipsw	Nov 6, 2023, 1:48 PM	13.91 GB	Apple Device Software Update File


Are you sure you want to change the extension from ".ipsw" to ".zip"?

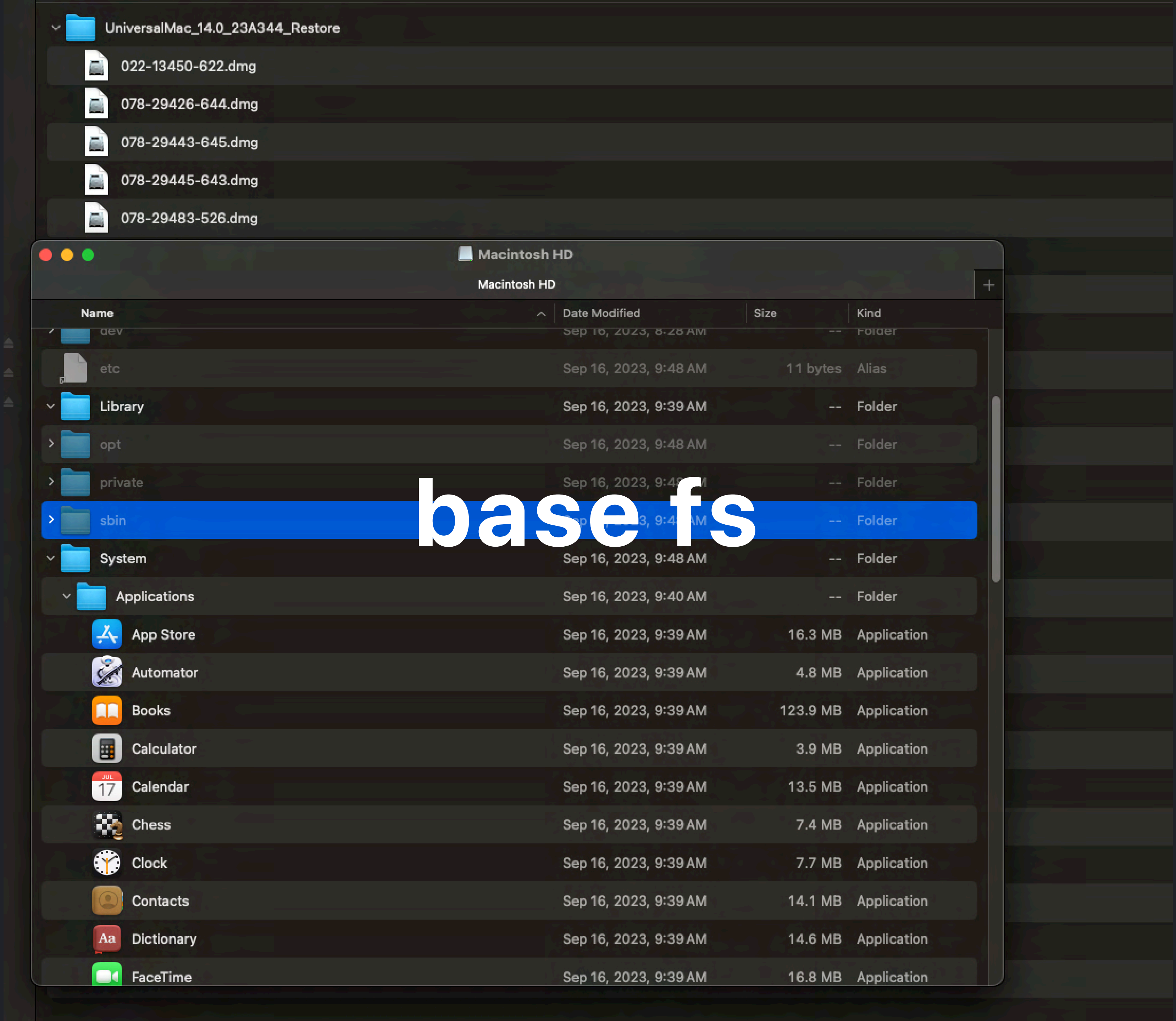
If you make this change, your document may open in a different application.

```
% unzip -l UniversalMac_14.1_23B74_Restore.ipsw
Archive: UniversalMac_14.1_23B74_Restore.ipsw
  Length      Date    Time    Name
---------  ---------- -----   ----
2019362579  01-09-2007 09:41   022-15754-105.dmg
 23068672  01-09-2007 09:41   097-26322-108.dmg
163577883  01-09-2007 09:41   097-26333-108.dmg
8066777054  01-09-2007 09:41   097-26346-096.dmg
163577883  01-09-2007 09:41   097-26379-107.dmg
4995416064  01-09-2007 09:41   097-26443-097.dmg
  2871603  01-09-2007 09:41   AppleDiagnostics.dmg
        0  01-09-2007 09:41   BootabilityBundle/
        0  01-09-2007 09:41   BootabilityBundle/Restore/
        0  01-09-2007 09:41   BootabilityBundle/Restore/Bootability/
        0  01-09-2007 09:41   BootabilityBundle/Restore/Bootability/BootabilityBrain.framework/
       33  01-09-2007 09:41   BootabilityBundle/Restore/Bootability/BootabilityBrain.framework/BootabilityBrain
       26  01-09-2007 09:41   BootabilityBundle/Restore/Bootability/BootabilityBrain.framework/Resources
```


Macintosh HD

macOS Base System

Sunburst23A344.arm64eSystemCryptex



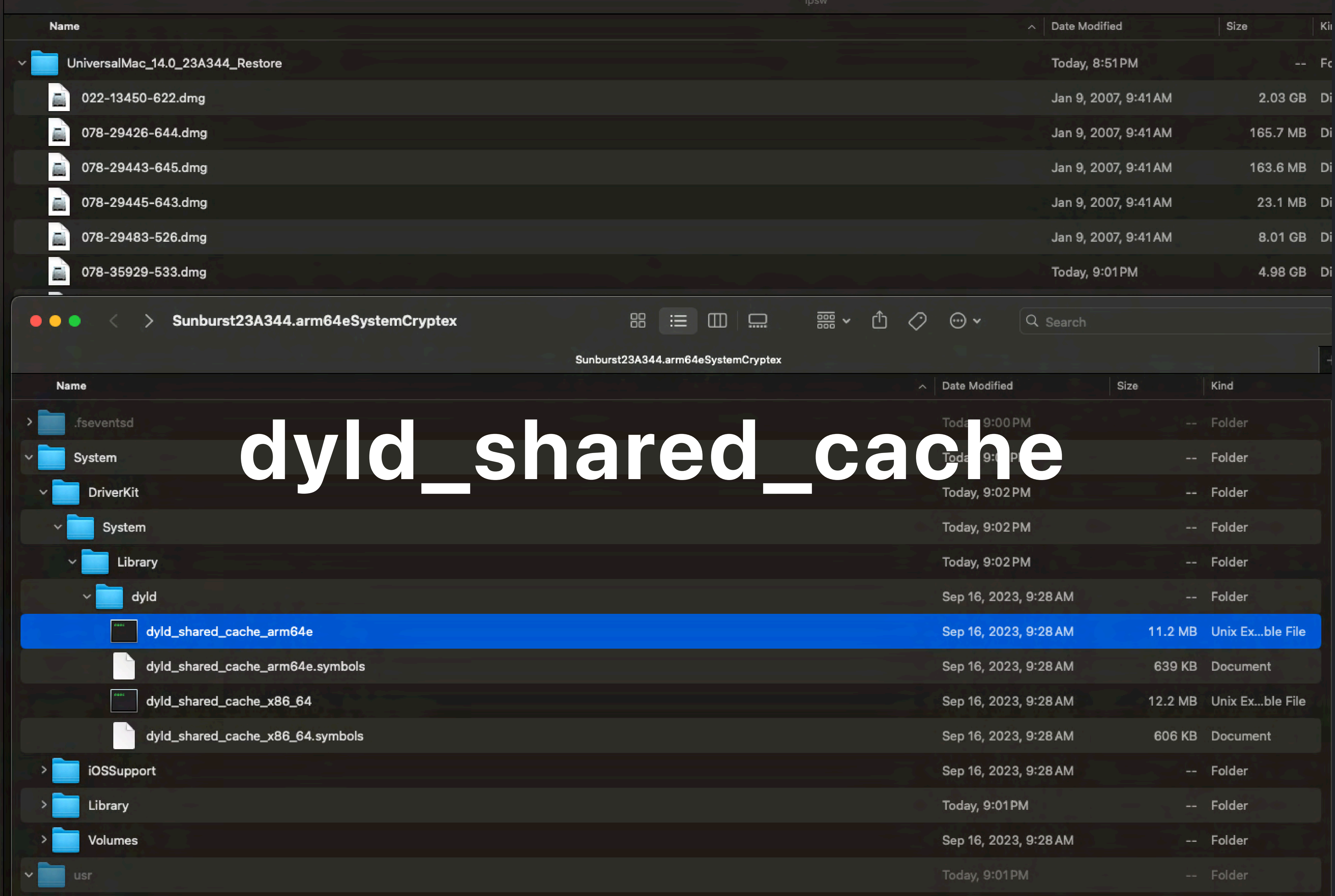
UniversalMac_14.0_23A344_Restore
- 022-13450-622.dmg
- 078-29426-644.dmg
- 078-29443-645.dmg
- 078-29445-643.dmg
- 078-29483-526.dmg

Macintosh HD

Name	Date Modified	Size	Kind
dev	Sep 16, 2023, 8:28 AM	--	Folder
etc	Sep 16, 2023, 9:48 AM	11 bytes	Alias
Library	Sep 16, 2023, 9:39 AM	--	Folder
opt	Sep 16, 2023, 9:48 AM	--	Folder
private	Sep 16, 2023, 9:4 M	--	Folder
sbin	3, 9:4 AM	--	Folder
System	Sep 16, 2023, 9:48 AM	--	Folder
Applications	Sep 16, 2023, 9:40 AM	--	Folder
App Store	Sep 16, 2023, 9:39 AM	16.3 MB	Application
Automator	Sep 16, 2023, 9:39 AM	4.8 MB	Application
Books	Sep 16, 2023, 9:39 AM	123.9 MB	Application
Calculator	Sep 16, 2023, 9:39 AM	3.9 MB	Application
Calendar	Sep 16, 2023, 9:39 AM	13.5 MB	Application
Chess	Sep 16, 2023, 9:39 AM	7.4 MB	Application
Clock	Sep 16, 2023, 9:39 AM	7.7 MB	Application
Contacts	Sep 16, 2023, 9:39 AM	14.1 MB	Application
Dictionary	Sep 16, 2023, 9:39 AM	14.6 MB	Application
FaceTime	Sep 16, 2023, 9:39 AM	16.8 MB	Application

base fs

dyld_shared_cache

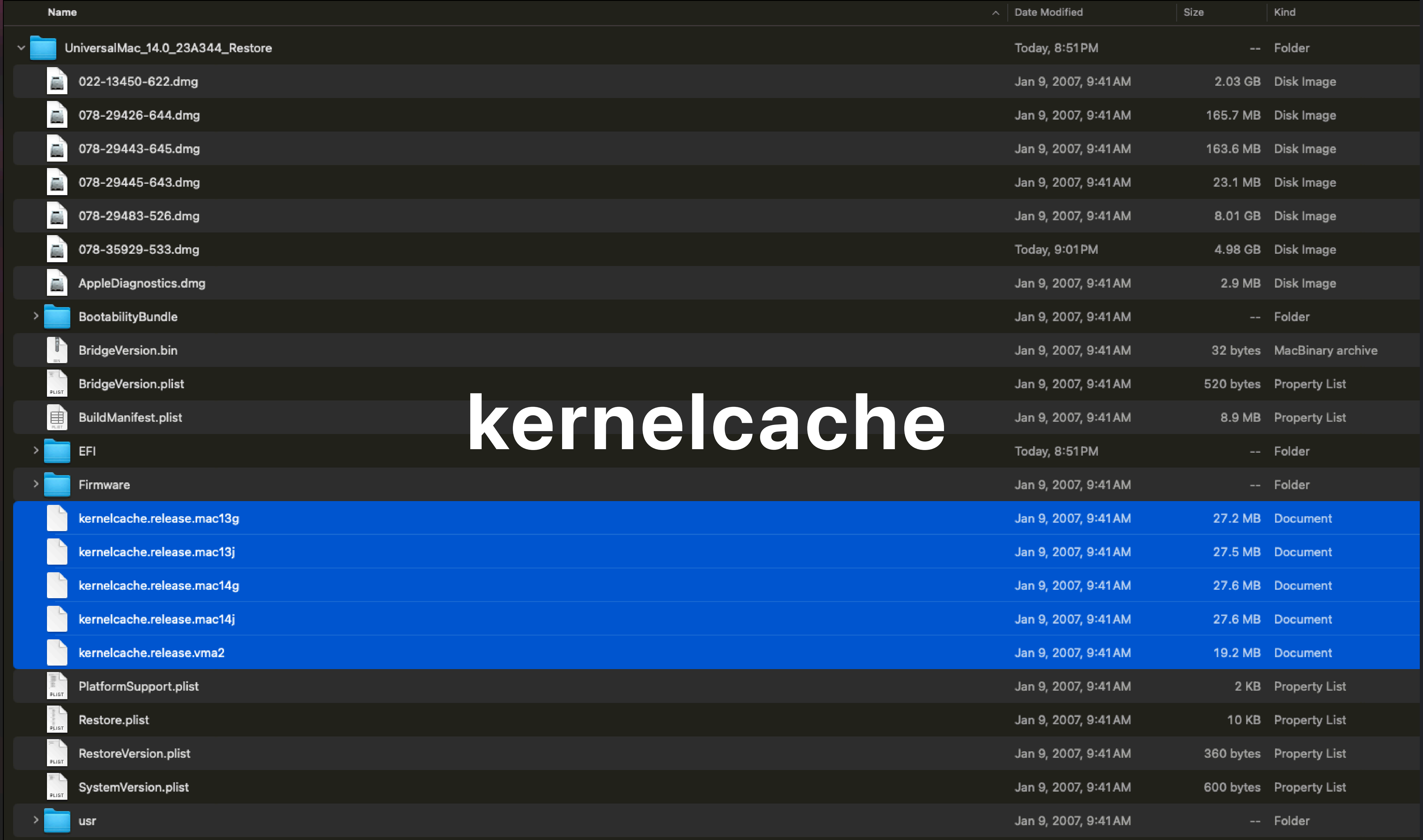


Name	Date Modified	Size	Ki
UniversalMac_14.0_23A344_Restore	Today, 8:51 PM	--	Fo
022-13450-622.dmg	Jan 9, 2007, 9:41 AM	2.03 GB	Di
078-29426-644.dmg	Jan 9, 2007, 9:41 AM	165.7 MB	Di
078-29443-645.dmg	Jan 9, 2007, 9:41 AM	163.6 MB	Di
078-29445-643.dmg	Jan 9, 2007, 9:41 AM	23.1 MB	Di
078-29483-526.dmg	Jan 9, 2007, 9:41 AM	8.01 GB	Di
078-35929-533.dmg	Today, 9:01 PM	4.98 GB	Di

Sunburst23A344.arm64eSystemCryptex

Name	Date Modified	Size	Kind
.fseventsd	Today, 9:00 PM	--	Folder
System	Today, 9:0 PM	--	Folder
DriverKit	Today, 9:02 PM	--	Folder
System	Today, 9:02 PM	--	Folder
Library	Today, 9:02 PM	--	Folder
dyld	Sep 16, 2023, 9:28 AM	--	Folder
dyld_shared_cache_arm64e	Sep 16, 2023, 9:28 AM	11.2 MB	Unix Ex...ble File
dyld_shared_cache_arm64e.symbols	Sep 16, 2023, 9:28 AM	639 KB	Document
dyld_shared_cache_x86_64	Sep 16, 2023, 9:28 AM	12.2 MB	Unix Ex...ble File
dyld_shared_cache_x86_64.symbols	Sep 16, 2023, 9:28 AM	606 KB	Document
iOSSupport	Sep 16, 2023, 9:28 AM	--	Folder
Library	Today, 9:01 PM	--	Folder
Volumes	Sep 16, 2023, 9:28 AM	--	Folder
usr	Today, 9:01 PM	--	Folder

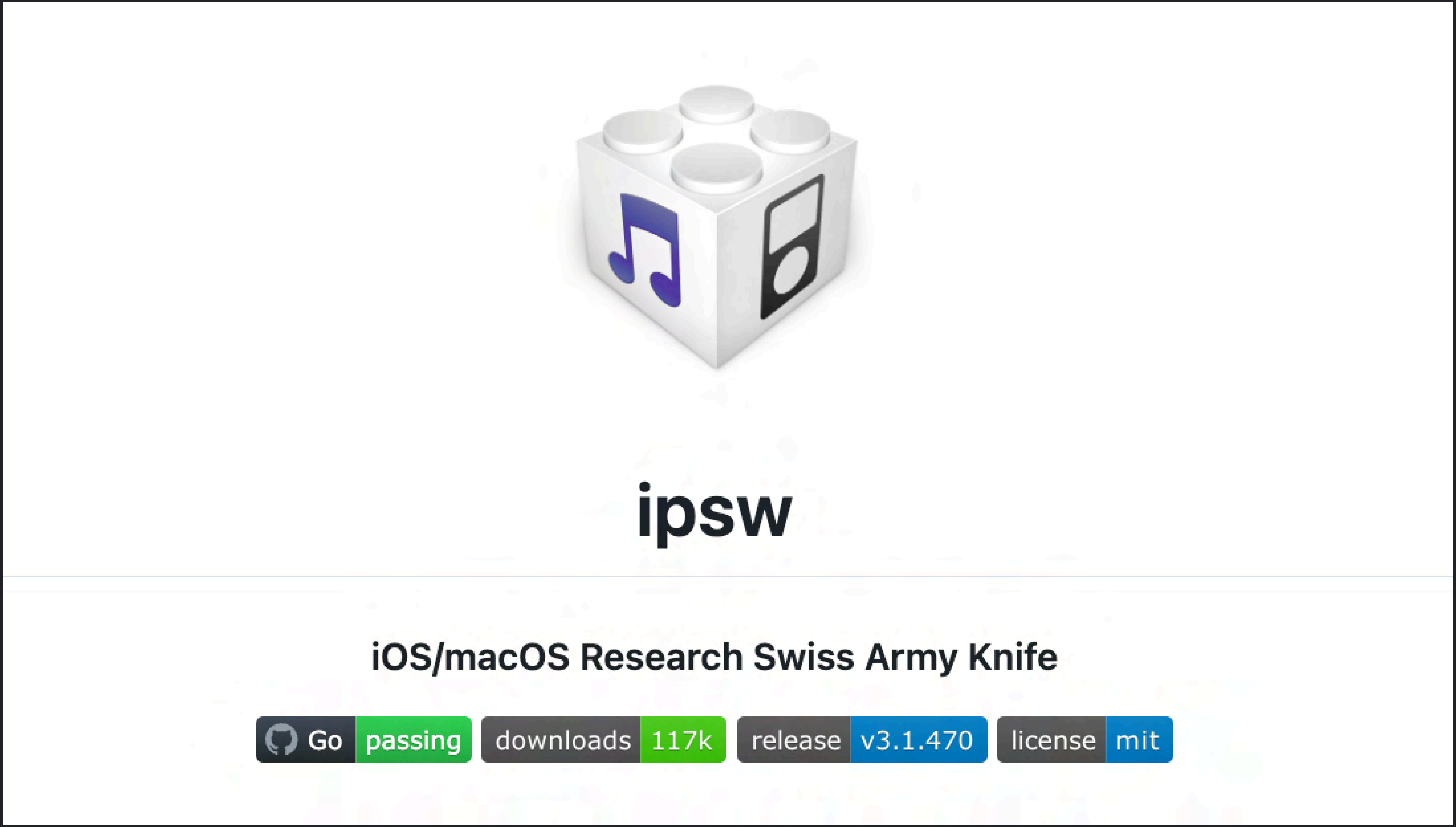
kernelcache



Name	Date Modified	Size	Kind
UniversalMac_14.0_23A344_Restore	Today, 8:51 PM	--	Folder
022-13450-622.dmg	Jan 9, 2007, 9:41 AM	2.03 GB	Disk Image
078-29426-644.dmg	Jan 9, 2007, 9:41 AM	165.7 MB	Disk Image
078-29443-645.dmg	Jan 9, 2007, 9:41 AM	163.6 MB	Disk Image
078-29445-643.dmg	Jan 9, 2007, 9:41 AM	23.1 MB	Disk Image
078-29483-526.dmg	Jan 9, 2007, 9:41 AM	8.01 GB	Disk Image
078-35929-533.dmg	Today, 9:01 PM	4.98 GB	Disk Image
AppleDiagnostics.dmg	Jan 9, 2007, 9:41 AM	2.9 MB	Disk Image
BootabilityBundle	Jan 9, 2007, 9:41 AM	--	Folder
BridgeVersion.bin	Jan 9, 2007, 9:41 AM	32 bytes	MacBinary archive
BridgeVersion.plist	Jan 9, 2007, 9:41 AM	520 bytes	Property List
BuildManifest.plist	Jan 9, 2007, 9:41 AM	8.9 MB	Property List
EFI	Today, 8:51 PM	--	Folder
Firmware	Jan 9, 2007, 9:41 AM	--	Folder
kernelcache.release.mac13g	Jan 9, 2007, 9:41 AM	27.2 MB	Document
kernelcache.release.mac13j	Jan 9, 2007, 9:41 AM	27.5 MB	Document
kernelcache.release.mac14g	Jan 9, 2007, 9:41 AM	27.6 MB	Document
kernelcache.release.mac14j	Jan 9, 2007, 9:41 AM	27.6 MB	Document
kernelcache.release.vma2	Jan 9, 2007, 9:41 AM	19.2 MB	Document
PlatformSupport.plist	Jan 9, 2007, 9:41 AM	2 KB	Property List
Restore.plist	Jan 9, 2007, 9:41 AM	10 KB	Property List
RestoreVersion.plist	Jan 9, 2007, 9:41 AM	360 bytes	Property List
SystemVersion.plist	Jan 9, 2007, 9:41 AM	600 bytes	Property List
usr	Jan 9, 2007, 9:41 AM	--	Folder

But how to make sense of all the files?

Enter `ipsw`



ipsw

iOS/macOS Research Swiss Army Knife

`ipsw` speaks fluent IPSW so we don't have to

“ Download and Parse IPSWs (and SO much more)
https://github.com/blacktop/ipsw ”

What is `ipsw` ? 🤔

- IPSW downloader/exploder
- OTA downloader/exploder
- macho parser
- ObjC class-dump
- Swift class-dump 🚧
- dyld_shared_cache parser
- kernelcache parser
- img4 parser/decrypter
- device-tree parser
- ARM v9-a disassember
- research tool

Diffing IPSW

IPSW Questions

- How can we download the correct IPSWs?
- How can we compare two different IPSWs?
- Whats inside the DSC? How can we look?
- How do we know which files have been updated?
- How can we extract files?

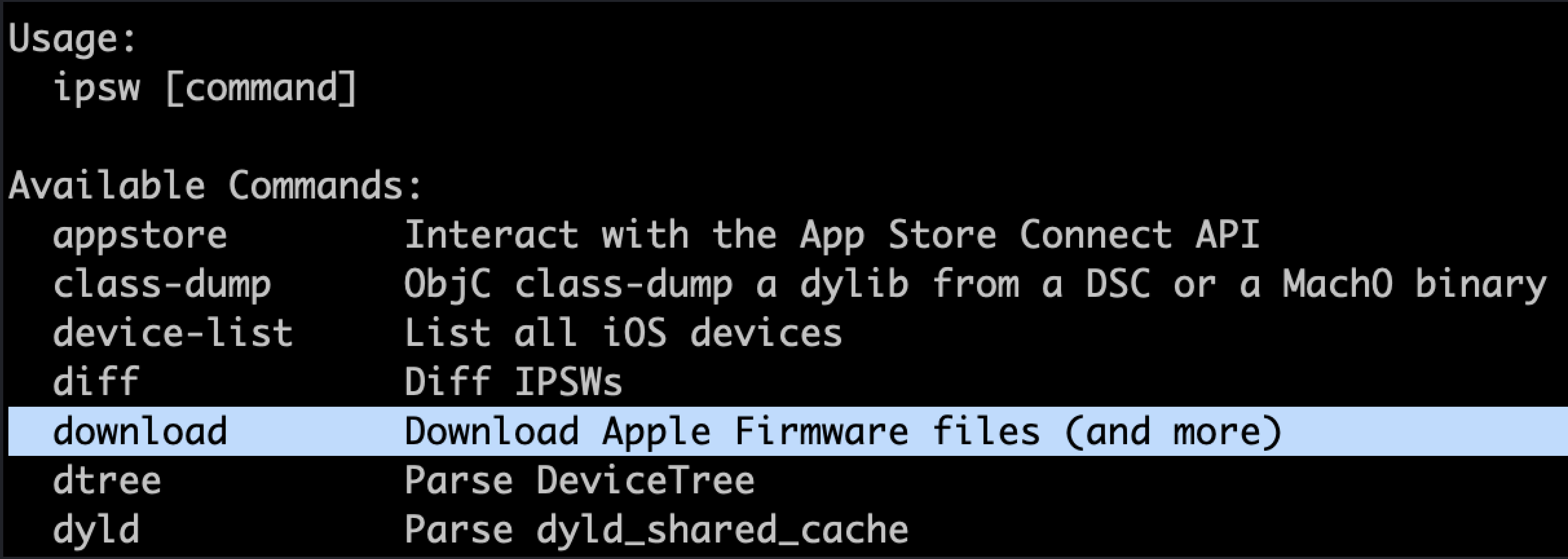
`ipsw` to the rescue

IPSW Questions

- How can we download the correct IPSWs?
  - `ipsw download --device <device>`
- How can we compare two different IPSWs?
  - `ipsw diff <IPSW old> <IPSW new>`
- Whats inside the DSC? How can we look?
  - `ipsw extract -d <IPSW>`
- How can we extract files?
  - `ipsw extract -f -p <file>`

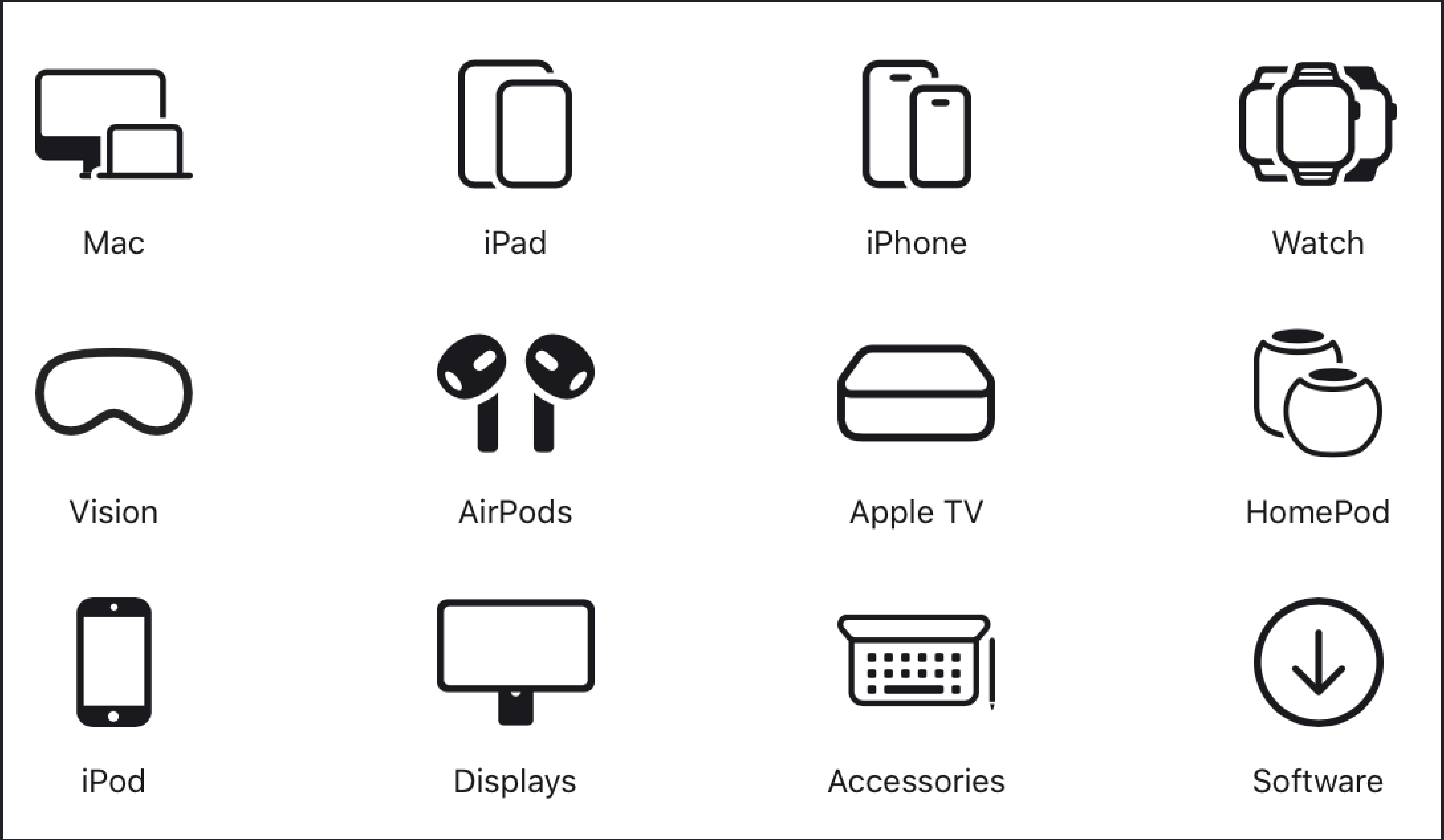
Downloading IPSWs

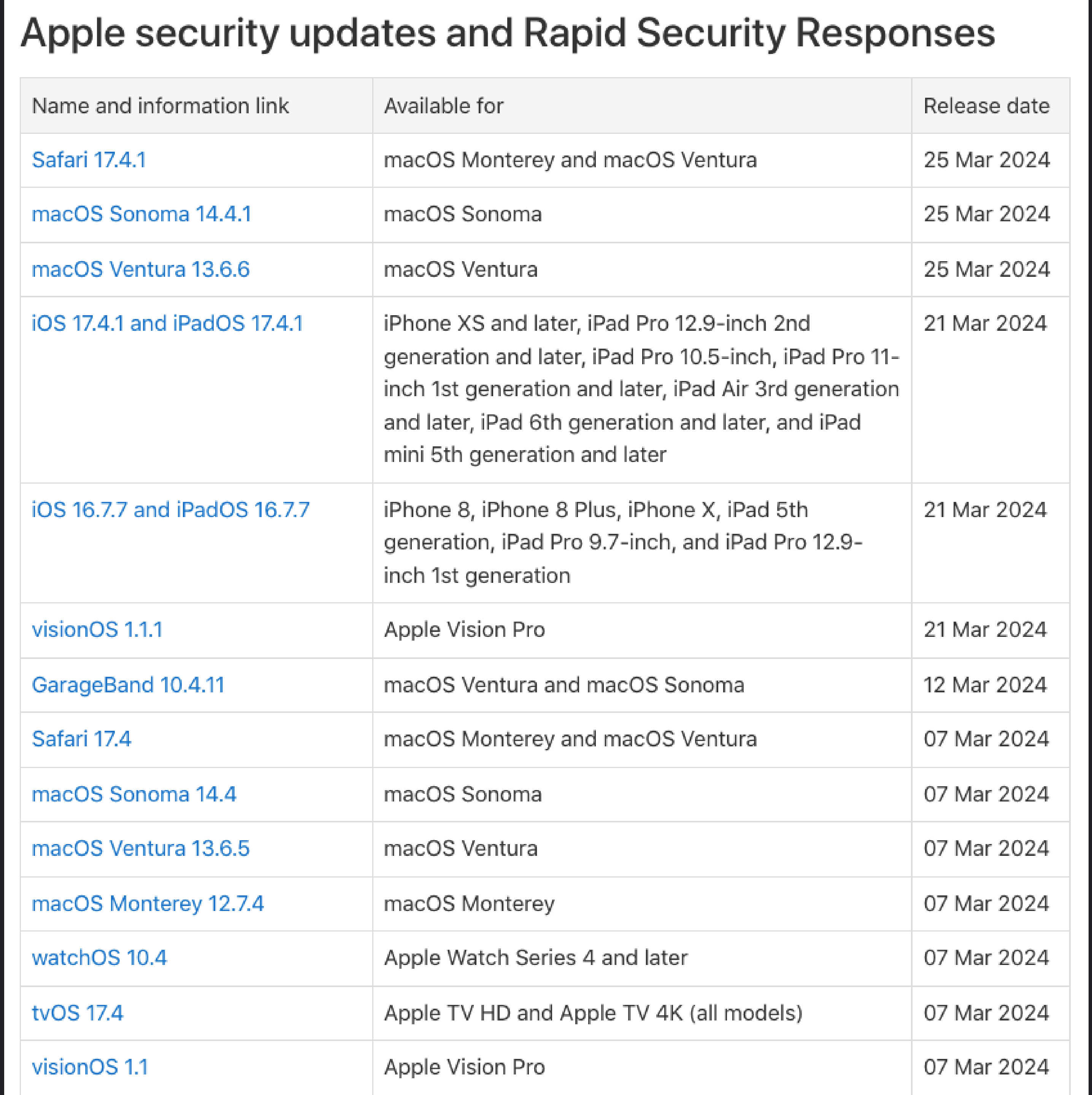
Download all or specific IPSWs with a single command...



```
Usage:
  ipsw [command]

Available Commands:
  appstore        Interact with the App Store Connect API
  class-dump      ObjC class-dump a dylib from a DSC or a MachO binary
  device-list     List all iOS devices
  diff            Diff IPSWs
  download        Download Apple Firmware files (and more)
  dtree           Parse DeviceTree
  dyld            Parse dyld_shared_cache
```

Select a device

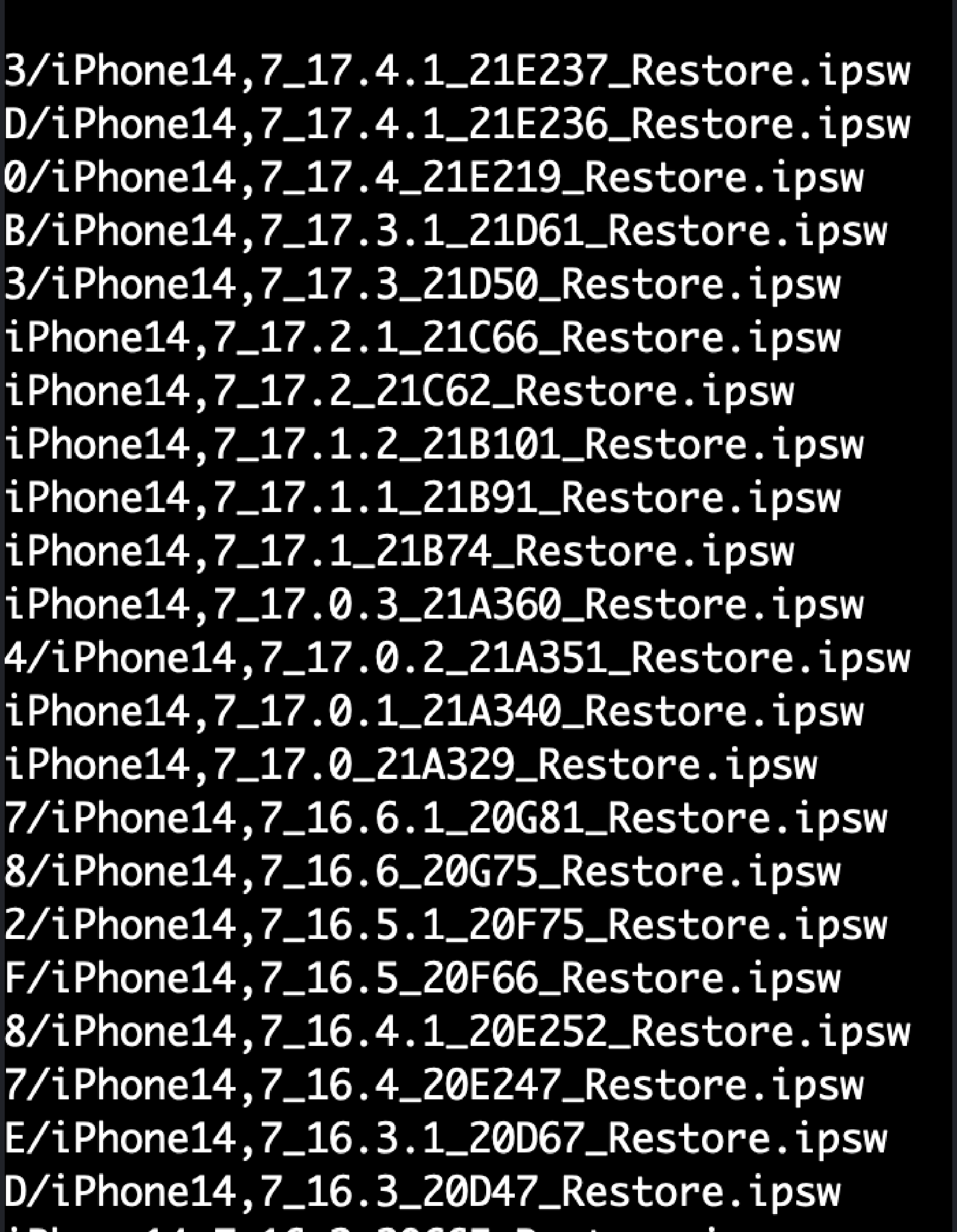




Apple security updates and Rapid Security Responses

Name and information link	Available for	Release date
Safari 17.4.1	macOS Monterey and macOS Ventura	25 Mar 2024
macOS Sonoma 14.4.1	macOS Sonoma	25 Mar 2024
macOS Ventura 13.6.6	macOS Ventura	25 Mar 2024
iOS 17.4.1 and iPadOS 17.4.1	iPhone XS and later, iPad Pro 12.9-inch 2nd generation and later, iPad Pro 10.5-inch, iPad Pro 11-inch 1st generation and later, iPad Air 3rd generation and later, iPad 6th generation and later, and iPad mini 5th generation and later	21 Mar 2024
iOS 16.7.7 and iPadOS 16.7.7	iPhone 8, iPhone 8 Plus, iPhone X, iPad 5th generation, iPad Pro 9.7-inch, and iPad Pro 12.9-inch 1st generation	21 Mar 2024
visionOS 1.1.1	Apple Vision Pro	21 Mar 2024
GarageBand 10.4.11	macOS Ventura and macOS Sonoma	12 Mar 2024
Safari 17.4	macOS Monterey and macOS Ventura	07 Mar 2024
macOS Sonoma 14.4	macOS Sonoma	07 Mar 2024
macOS Ventura 13.6.5	macOS Ventura	07 Mar 2024
macOS Monterey 12.7.4	macOS Monterey	07 Mar 2024
watchOS 10.4	Apple Watch Series 4 and later	07 Mar 2024
tvOS 17.4	Apple TV HD and Apple TV 4K (all models)	07 Mar 2024
visionOS 1.1	Apple Vision Pro	07 Mar 2024

1. Download patched IPSW
2. Download N-1 IPSW
3. Run `ipsw diff`

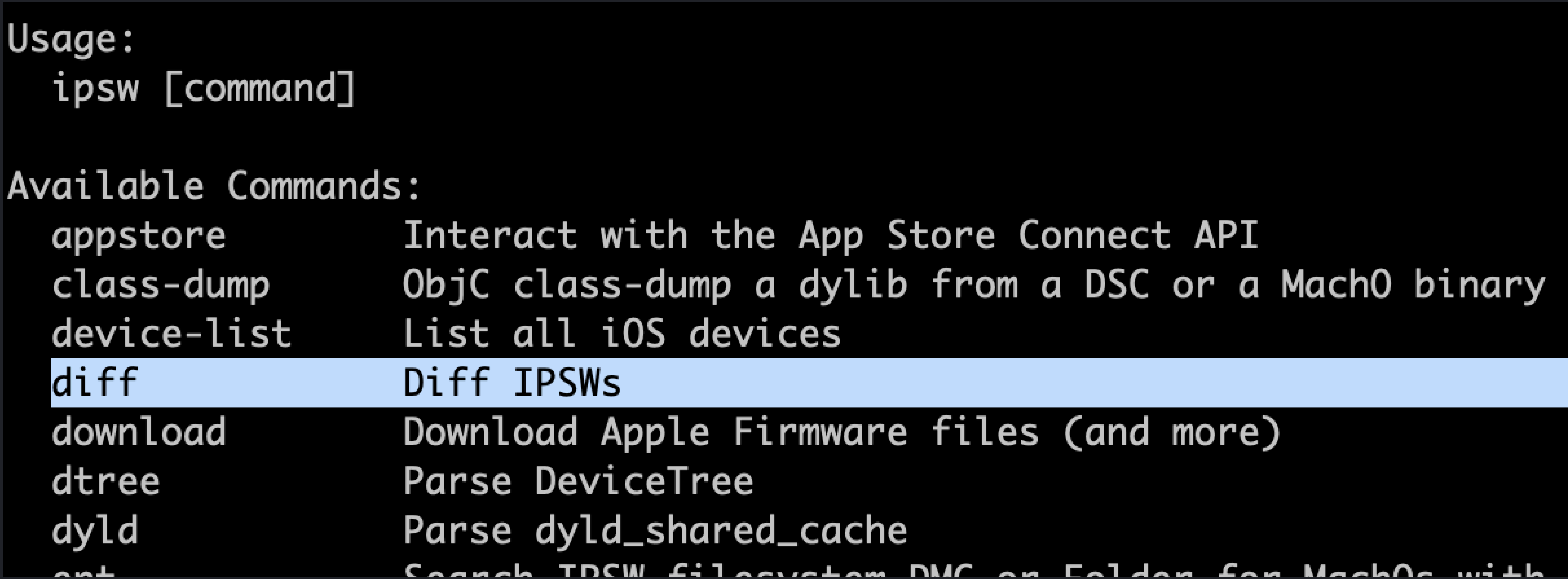


```
3/iPhone14,7_17.4.1_21E237_Restore.ipsw
D/iPhone14,7_17.4.1_21E236_Restore.ipsw
0/iPhone14,7_17.4_21E219_Restore.ipsw
B/iPhone14,7_17.3.1_21D61_Restore.ipsw
3/iPhone14,7_17.3_21D50_Restore.ipsw
iPhone14,7_17.2.1_21C66_Restore.ipsw
iPhone14,7_17.2_21C62_Restore.ipsw
iPhone14,7_17.1.2_21B101_Restore.ipsw
iPhone14,7_17.1.1_21B91_Restore.ipsw
iPhone14,7_17.1_21B74_Restore.ipsw
iPhone14,7_17.0.3_21A360_Restore.ipsw
4/iPhone14,7_17.0.2_21A351_Restore.ipsw
iPhone14,7_17.0.1_21A340_Restore.ipsw
iPhone14,7_17.0_21A329_Restore.ipsw
7/iPhone14,7_16.6.1_20G81_Restore.ipsw
8/iPhone14,7_16.6_20G75_Restore.ipsw
2/iPhone14,7_16.5.1_20F75_Restore.ipsw
F/iPhone14,7_16.5_20F66_Restore.ipsw
8/iPhone14,7_16.4.1_20E252_Restore.ipsw
7/iPhone14,7_16.4_20E247_Restore.ipsw
E/iPhone14,7_16.3.1_20D67_Restore.ipsw
D/iPhone14,7_16.3_20D47_Restore.ipsw
```

```
ipsw download ipsw --device Macmini9,1 -V -b 23A344
```

```
ipsw download ipsw --device Macmini9,1 -V -b 23B74
```

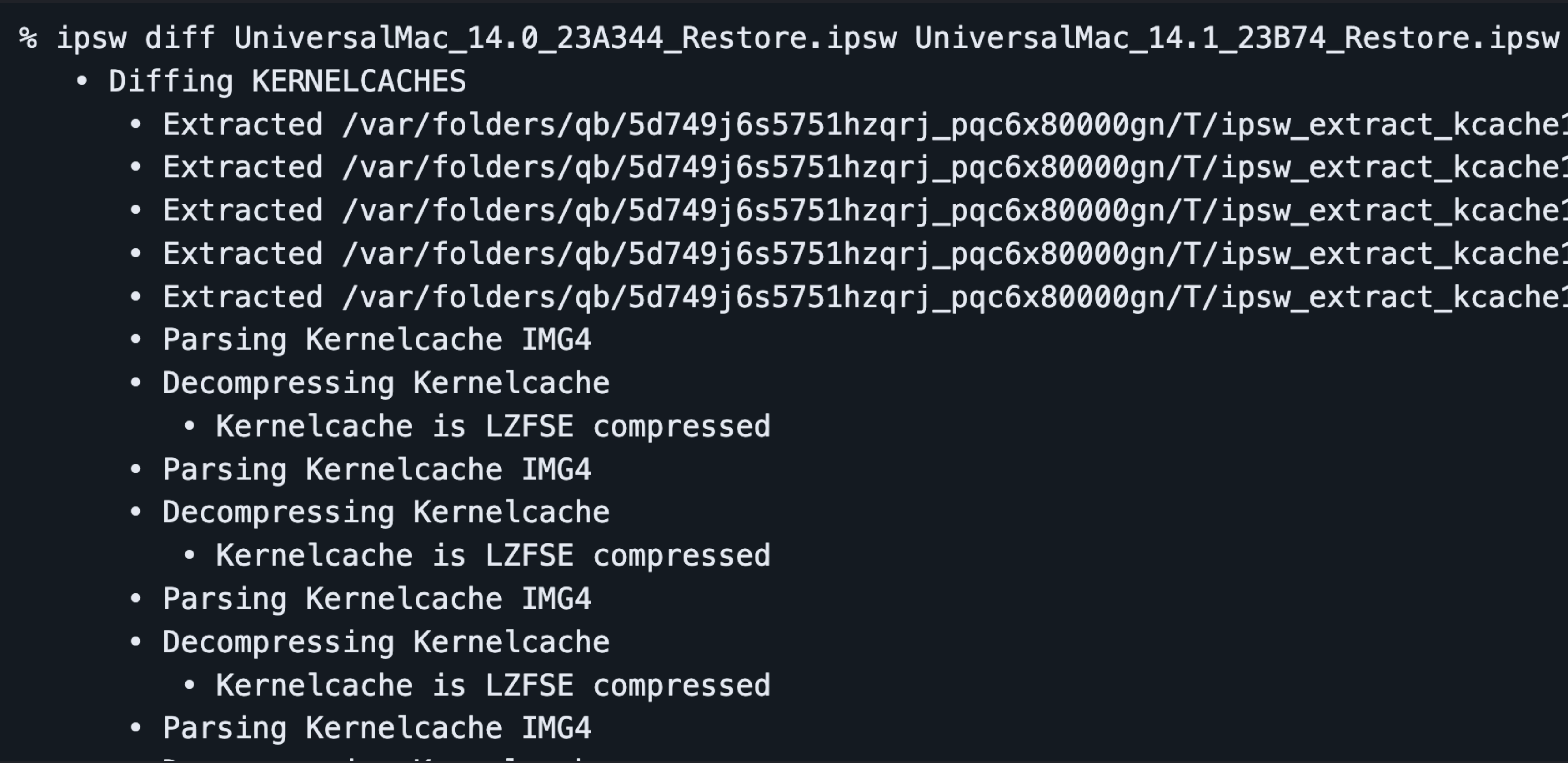
Compare Two Firmwares



```
Usage:
  ipsw [command]

Available Commands:
  appstore        Interact with the App Store Connect API
  class-dump      ObjC class-dump a dylib from a DSC or a MachO binary
  device-list     List all iOS devices
  diff            Diff IPSWs
  download        Download Apple Firmware files (and more)
  dtree           Parse DeviceTree
  dyld            Parse dyld_shared_cache
  ent             Search IPSW filesystem DMG or Folder for MachOs with
```

ipsw diff

```
ipsw diff <old_ipsw> <new_ipsw>
```

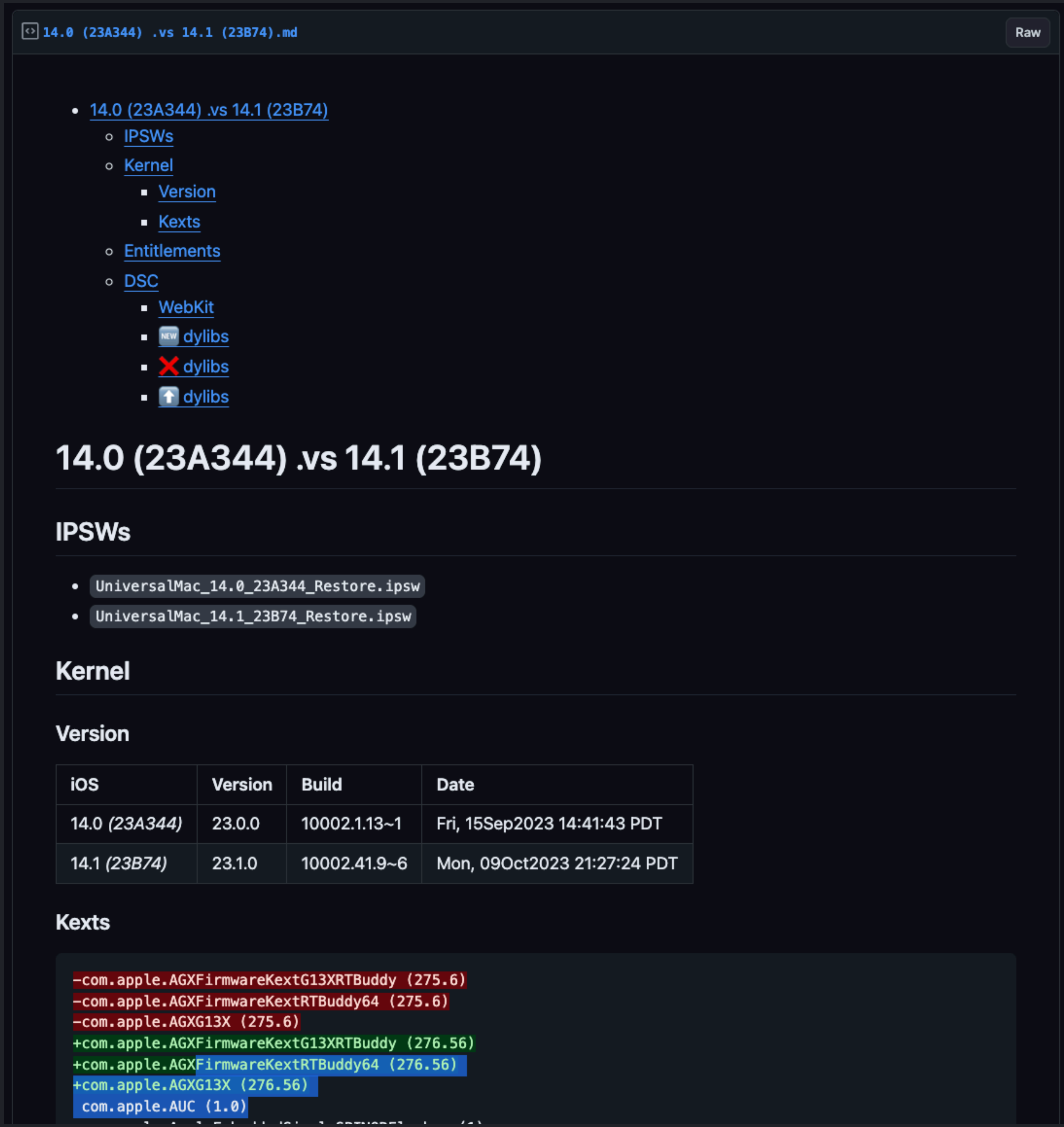


```
% ipsw diff UniversalMac_14.0_23A344_Restore.ipsw UniversalMac_14.1_23B74_Restore.ipsw
   • Diffing KERNELCACHES
      • Extracted /var/folders/qb/5d749j6s5751hzqrj_pqc6x80000gn/T/ipsw_extract_kcache
      • Extracted /var/folders/qb/5d749j6s5751hzqrj_pqc6x80000gn/T/ipsw_extract_kcache
      • Extracted /var/folders/qb/5d749j6s5751hzqrj_pqc6x80000gn/T/ipsw_extract_kcache
      • Extracted /var/folders/qb/5d749j6s5751hzqrj_pqc6x80000gn/T/ipsw_extract_kcache
      • Extracted /var/folders/qb/5d749j6s5751hzqrj_pqc6x80000gn/T/ipsw_extract_kcache
      • Parsing Kernelcache IMG4
      • Decompressing Kernelcache
         • Kernelcache is LZFSE compressed
      • Parsing Kernelcache IMG4
      • Decompressing Kernelcache
         • Kernelcache is LZFSE compressed
      • Parsing Kernelcache IMG4
      • Decompressing Kernelcache
         • Kernelcache is LZFSE compressed
      • Parsing Kernelcache IMG4
```

`ipsw diff`

MacOS 14.0 (23A344) .vs 14.1 (23B74)

(ipsw version: 3.1.428, BuildTime: 2023-12-11T20:45:51Z)



14.0 (23A344) .vs 14.1 (23B74).md

Raw

14.0 (23A344) .vs 14.1 (23B74)

IPSWs

- `UniversalMac_14.0_23A344_Restore.ipsw`
- `UniversalMac_14.1_23B74_Restore.ipsw`

Kernel

Version

iOS	Version	Build	Date
14.0 *(23A344)*	23.0.0	10002.1.13~1	Fri, 15Sep2023 14:41:43 PDT
14.1 *(23B74)*	23.1.0	10002.41.9~6	Mon, 09Oct2023 21:27:24 PDT

Kexts

```
-com.apple.AGXFirmwareKextG13XRTBuddy (275.6)
-com.apple.AGXFirmwareKextRTBuddy64 (275.6)
-com.apple.AGXG13X (275.6)
+com.apple.AGXFirmwareKextG13XRTBuddy (276.56)
+com.apple.AGXFirmwareKextRTBuddy64 (276.56)
+com.apple.AGXG13X (276.56)
 com.apple.AUC (1.0)
```

IPSW diff TOC (late 2023)

- IPSW metadata
- Kernel
  - version
  - KEXTs
- Entitlements
- DSC
  - dylibs / frameworks

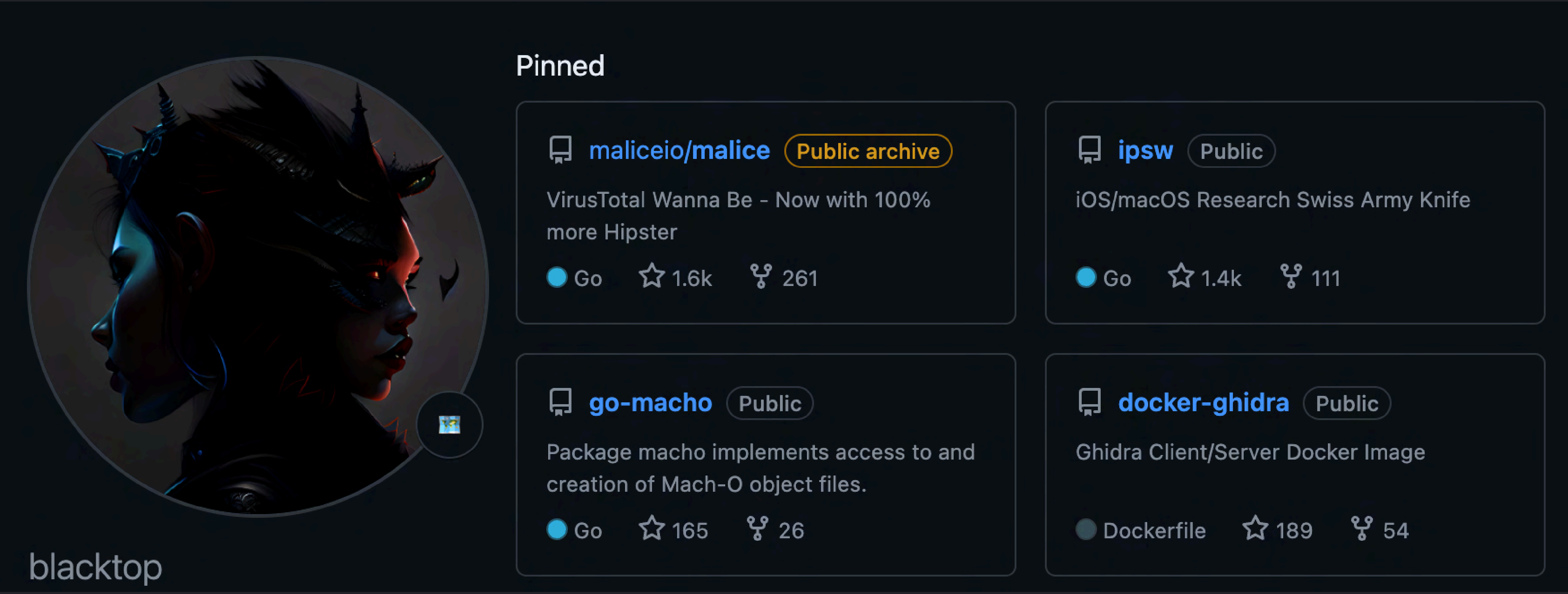
First Diffing Attempt...

- CVE in a system application (Mach-O executable)
- `ipsw diff` ing a MacOS IPSW and the diff failed
- No Macho binary changes listed

`ipsw diff` Improvements

- MacOS IPSW broken
  - https://github.com/blacktop/ipsw/issues/369
- ipsw diff - add machos file diffs
  - https://github.com/blacktop/ipsw/issues/371
- ipsw diff - Extract changed files / command gen
  - https://github.com/blacktop/ipsw/issues/372

blacktop



blacktop

Pinned

maliceio/malice Public archive

VirusTotal Wanna Be - Now with 100% more Hipster

● Go ☆ 1.6k 261

ipsw Public

iOS/macOS Research Swiss Army Knife

● Go ☆ 1.4k 111

go-macho Public

Package macho implements access to and creation of Mach-O object files.

● Go ☆ 165 26

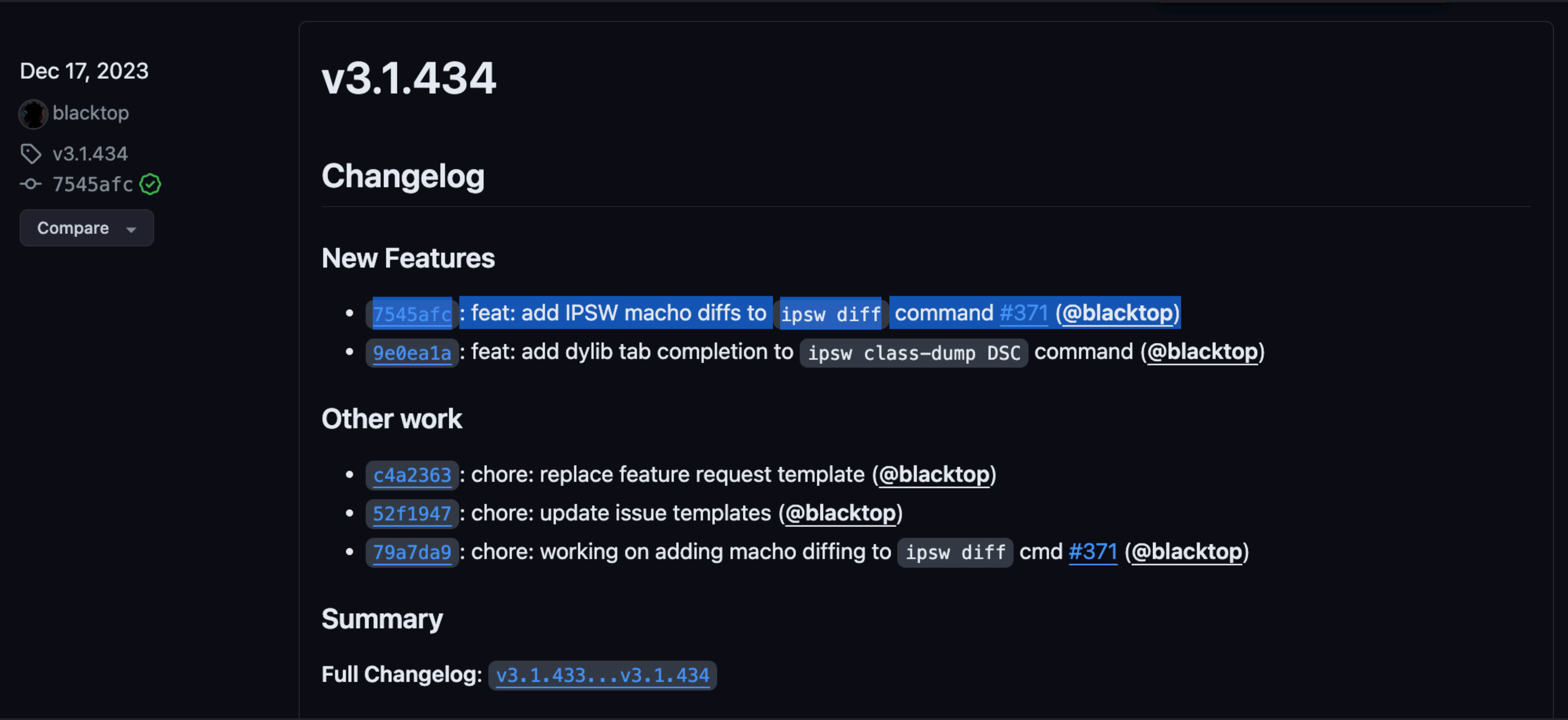
docker-ghidra Public

Ghidra Client/Server Docker Image

● Dockerfile ☆ 189 54

blacktop

- Responded and had these things added in ~3 days.. 👏
- Senior security researcher at Trenchant 🤓
- Expert in Apple security, all things IPSW 🍎 📱
- Writes awesome software 💻 💪
- Answered countless questions and provided many insights in this talk
- Introduced me to hack different


Dec 17, 2023

blacktop

v3.1.434

7545afc

Compare

v3.1.434

Changelog

New Features

- `7545afc`: feat: add IPSW macho diffs to `ipsw diff` command #371 (**@blacktop**)
- `9e0ea1a`: feat: add dylib tab completion to `ipsw class-dump DSC` command (**@blacktop**)

Other work

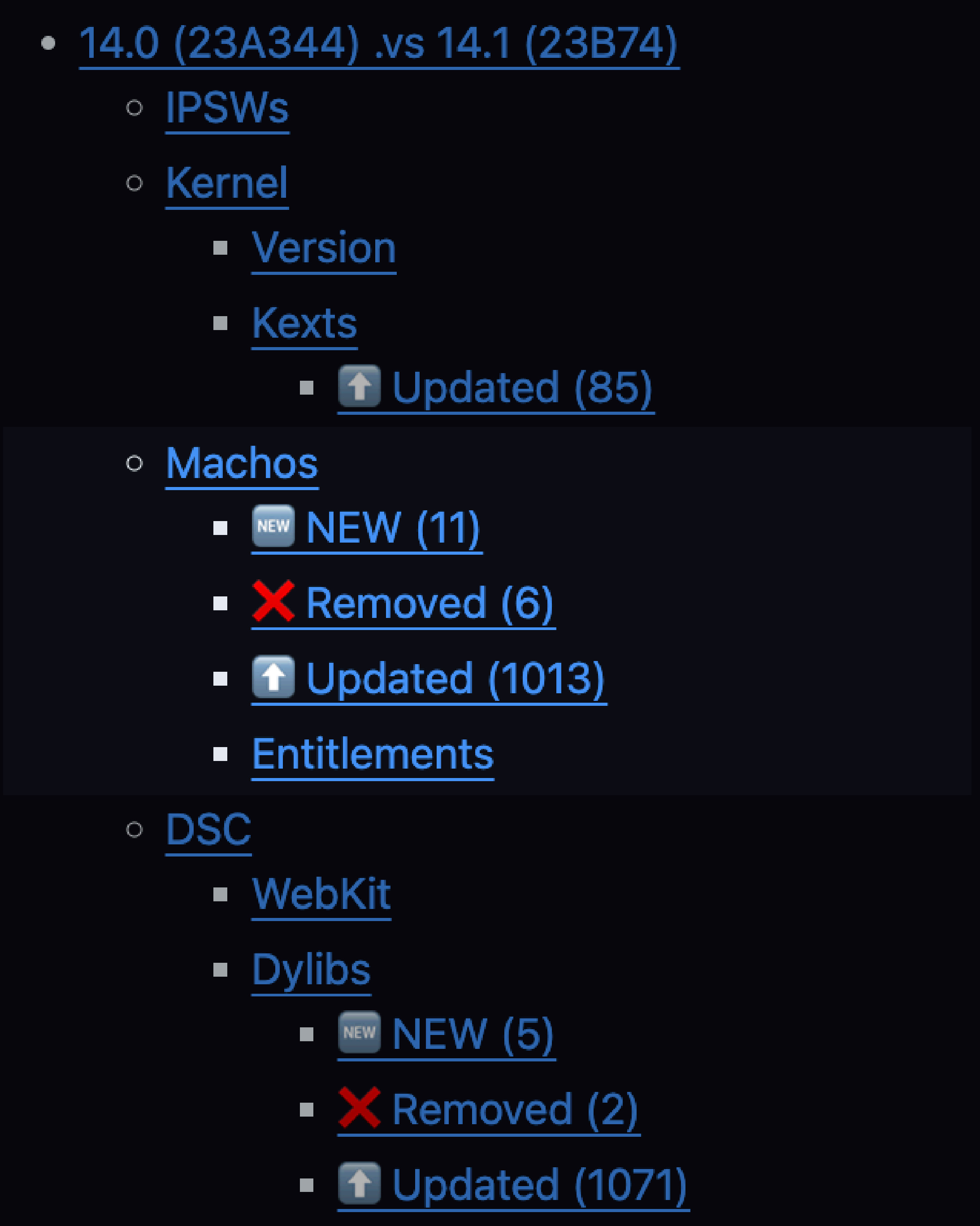
- `c4a2363`: chore: replace feature request template (**@blacktop**)
- `52f1947`: chore: update issue templates (**@blacktop**)
- `79a7da9`: chore: working on adding macho diffing to `ipsw diff` cmd #371 (**@blacktop**)

Summary

Full Changelog: `v3.1.433...v3.1.434` `ipsw diff`

MacOS 14.0 (23A344) .vs 14.1 (23B74)

(ipsw version: 3.1.434+)



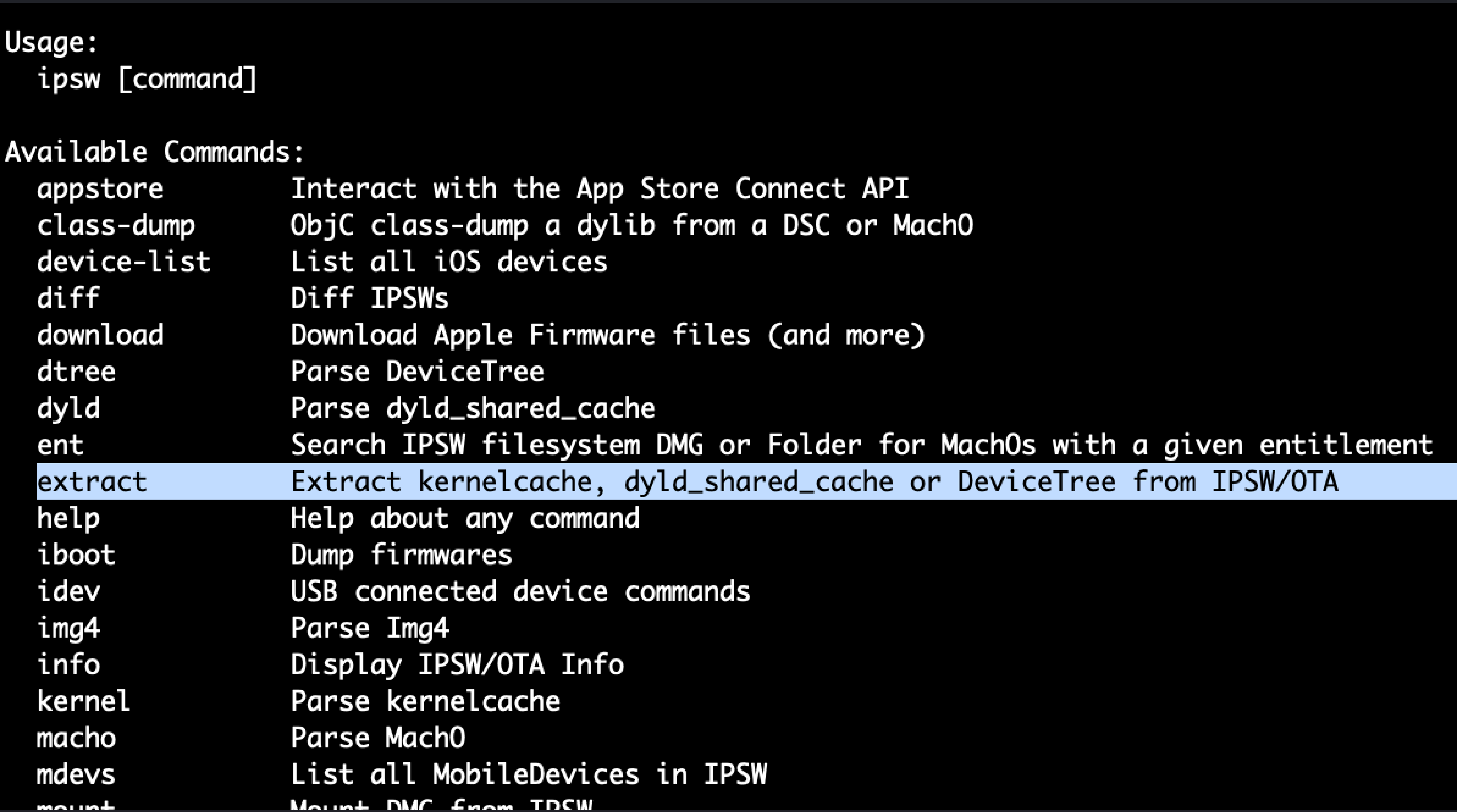
- 14.0 (23A344) .vs 14.1 (23B74)
  - IPSWs
  - Kernel
    - Version
    - Kexts
      - ⬆️ Updated (85)
  - Machos
    - 🆕 NEW (11)
    - ❌ Removed (6)
    - ⬆️ Updated (1013)
    - Entitlements
  - DSC
    - WebKit
    - Dylibs
      - 🆕 NEW (5)
      - ❌ Removed (2)
      - ⬆️ Updated (1071)

High Level Patch Diffing on *OS

1. Identify a CVE of interest
2. Download corresponding IPSW update and N-1
3. Determine changes for update
4. Map binaries to CVE
5. Extract the related file(s)
6. Diff the binaries
7. Root cause the vulnerability

Diffing *OS Binaries

Extracting Binaries from IPSW



```
Usage:
  ipsw [command]

Available Commands:
  appstore        Interact with the App Store Connect API
  class-dump      ObjC class-dump a dylib from a DSC or MachO
  device-list     List all iOS devices
  diff            Diff IPSWs
  download        Download Apple Firmware files (and more)
  dtree           Parse DeviceTree
  dyld            Parse dyld_shared_cache
  ent             Search IPSW filesystem DMG or Folder for MachOs with a given entitlement
  extract         Extract kernelcache, dyld_shared_cache or DeviceTree from IPSW/OTA
  help            Help about any command
  iboot           Dump firmwares
  idev            USB connected device commands
  img4            Parse Img4
  info            Display IPSW/OTA Info
  kernel          Parse kernelcache
  macho           Parse MachO
  mdevs           List all MobileDevices in IPSW
  mount           Mount DMG from IPSW
```

Searching For Binaries

- Root fs
  - mostly app / macho executables
- kernelcache
  - kernel, KEXT
- dyld_shared_cache (DSC)
  - dylibs, frameworks

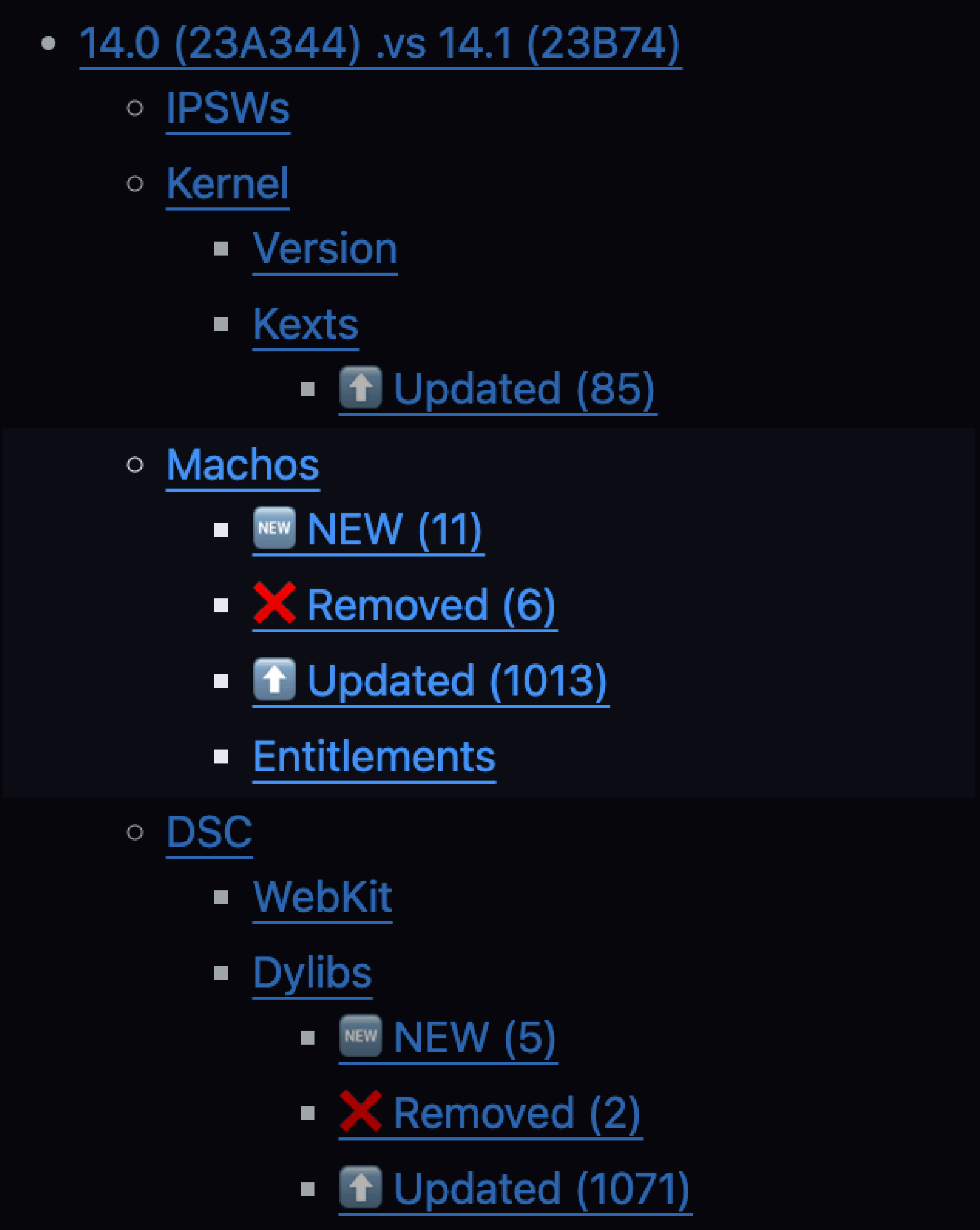
Tip: Binary Missing? - Check DSC

If you search on iOS for most libraries, such as libSystem, you'll be wasting your time.
... the actual file is not present on the file system.

To save time on library loading, iOS's dyld employs a shared, pre-linked cache, and Apple has moved all the base libraries into it...

Jonathan Levin - Mac OS X and iOS Internals 2013

The `ipsw diff` result helps you know where to extract from.



- 14.0 (23A344) .vs 14.1 (23B74)
  - IPSWs
  - Kernel
    - Version
    - Kexts
      - ⬆️ Updated (85)
  - Machos
    - 🆕 NEW (11)
    - ❌ Removed (6)
    - ⬆️ Updated (1013)
    - Entitlements
  - DSC
    - WebKit
    - Dylibs
      - 🆕 NEW (5)
      - ❌ Removed (2)
      - ⬆️ Updated (1071)

High Level Patch Diffing on *OS

1. Identify a CVE of interest
2. Download corresponding IPSW update and N-1
3. Determine changes for update
4. Map binaries to CVE
5. Extract the related file(s)
6. Diff the binaries
7. Root cause the vulnerability

Diffing *OS Binaries

```
egrep -i "ios|macho|apple|macos|Mach-O"
changelog | wc -l
```

88

Ghidra 11.0.3 Change History (April 2024)

Ghidra 11.0.2 Change History (March 2024)

- Importer:Mach-O. The dyld_shared_cache loader no longer throws an exception when importing newer versions that use - dyld_cache_slide_info5. (GP-4457)

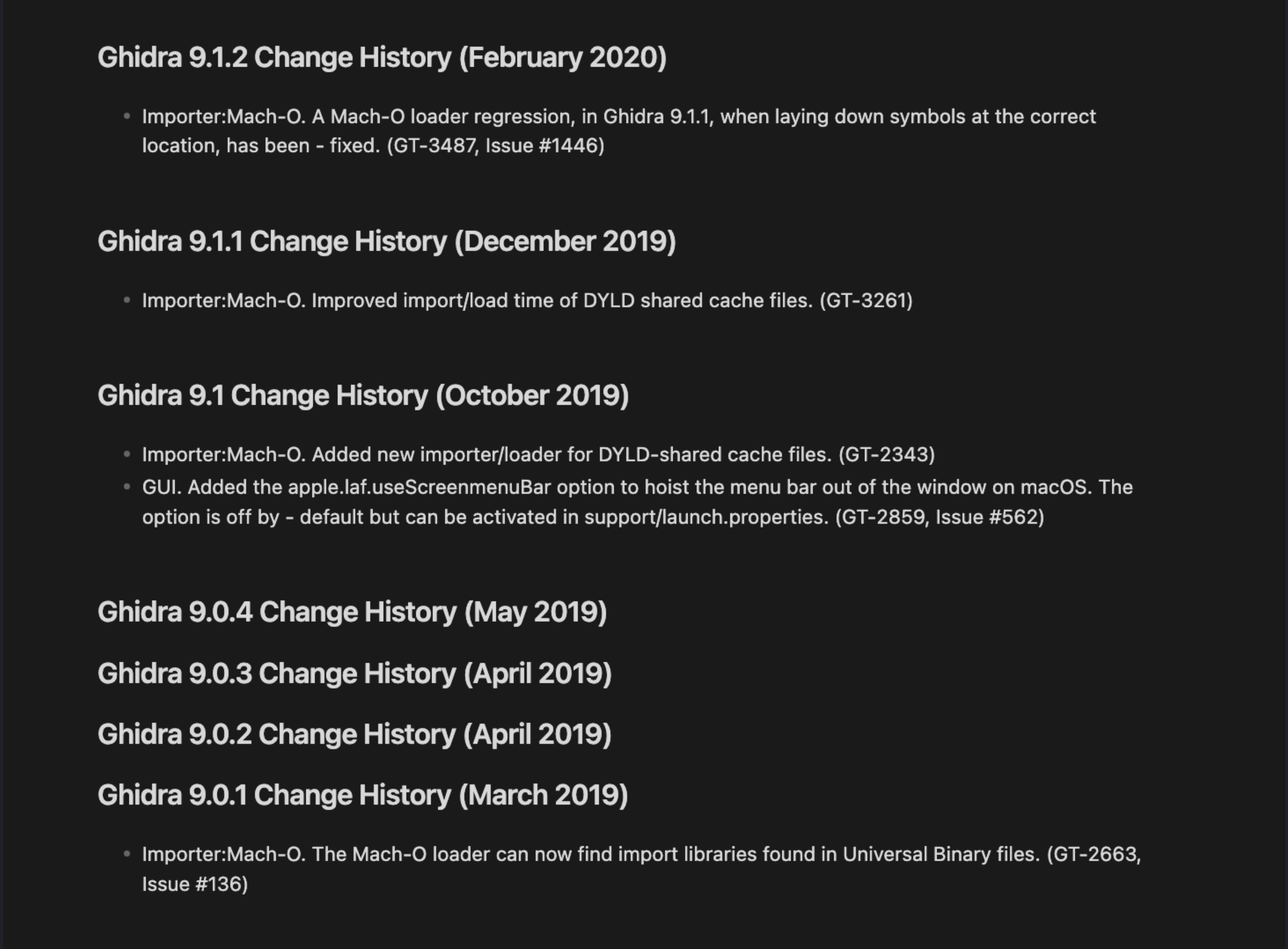
Ghidra 11.0.1 Change History (January 2024)

Ghidra 11.0 Change History (December 2023)

- Importer:Mach-O. dyld_shared_cache components extracted from Ghidra's DyldCacheFileSystem can now be added together - on-demand with the Add To Program feature. Broken references can be automatically resolved by right-clicking on them and - clicking References -> Add To Program. (GP-3753, Issue #5023)
- Importer:Mach-O. When loading System Libraries From Disk on macOS, the dyld_shared_cache will be searched for in more - default locations. (GP-3909)
- Importer:Mach-O. The MachoLoader now uses binding information (if present) to associate libraries with imported symbol name - without the need for those libraries to be already present/loaded in the project. (GP-3912)
- Importer:Mach-O. The MachoLoader can now load binaries with obfuscated segment and section names. (GP-3926, Issue #3876)
- Importer. Importing libraries that are referenced by absolute path (such as with Mach-O) now get saved to the project with - their folder structure intact. This fixes a potential DuplicateKeyException that could occur when using a Recursive Library - Load Depth greater than 1, and removes any ambiguity that could occur when linking a program to its libraries. (GP-3922)
- Importer:Mach-O. The MachoLoader now creates thunks on stubs. (GP-3248, Issue #3146)

Ghidra 10.4 Change History (September 2023)

- Importer:Mach-O. Libraries can now be loaded from both local directories and GFileSystems. This enables loading, for - example, Mach-O libraries directly from within the dyld_shared_cache file(s). (GP-2277, Issue #4162)
- Importer:Mach-O. Improved markup for Mach-O load command dat



Ghidra 9.1.2 Change History (February 2020)

- Importer:Mach-O. A Mach-O loader regression, in Ghidra 9.1.1, when laying down symbols at the correct location, has been - fixed. (GT-3487, Issue #1446)

Ghidra 9.1.1 Change History (December 2019)

- Importer:Mach-O. Improved import/load time of DYLD shared cache files. (GT-3261)

Ghidra 9.1 Change History (October 2019)

- Importer:Mach-O. Added new importer/loader for DYLD-shared cache files. (GT-2343)
- GUI. Added the apple.laf.useScreenmenuBar option to hoist the menu bar out of the window on macOS. The option is off by - default but can be activated in support/launch.properties. (GT-2859, Issue #562)

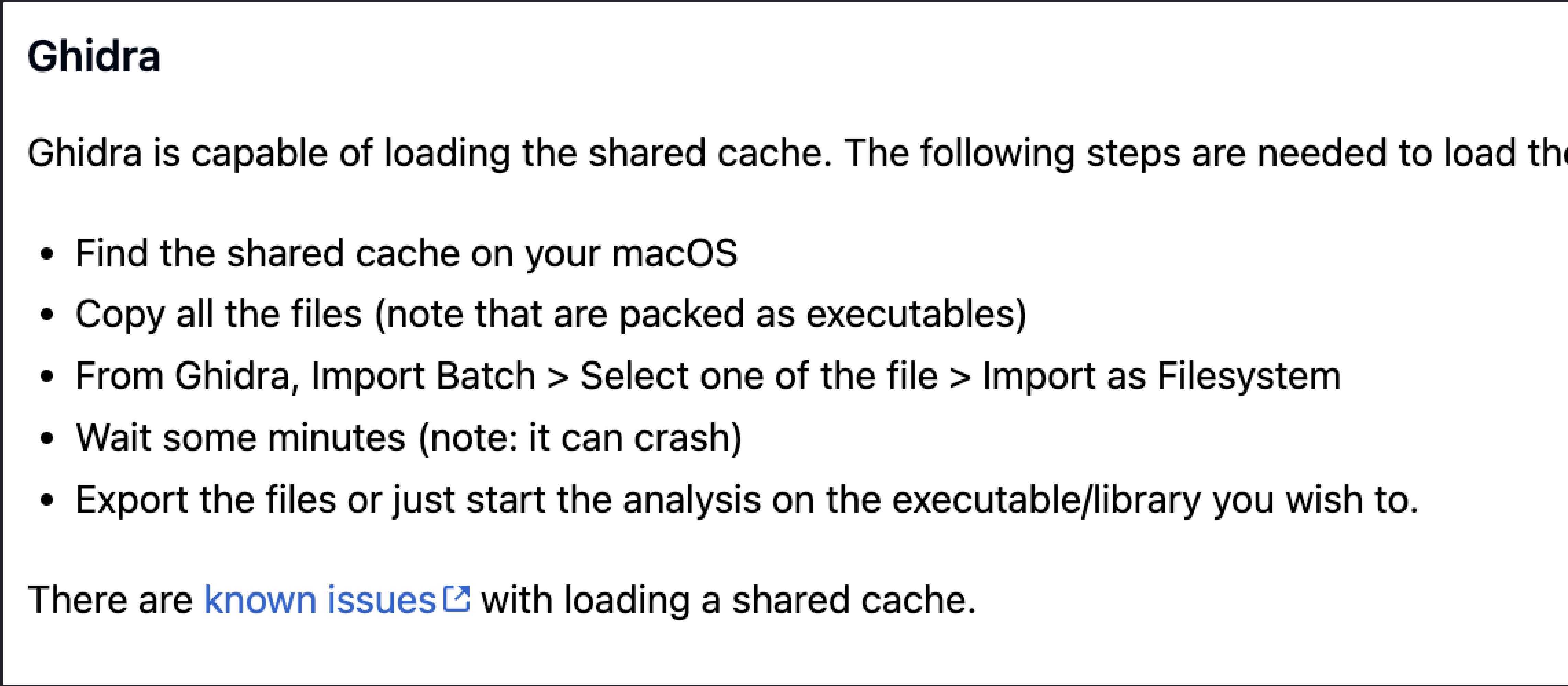
Ghidra 9.0.4 Change History (May 2019)

Ghidra 9.0.3 Change History (April 2019)

Ghidra 9.0.2 Change History (April 2019)

Ghidra 9.0.1 Change History (March 2019)

- Importer:Mach-O. The Mach-O loader can now find import libraries found in Universal Binary files. (GT-2663, Issue #136)



Ghidra

Ghidra is capable of loading the shared cache. The following steps are needed to load th

- Find the shared cache on your macOS
- Copy all the files (note that are packed as executables)
- From Ghidra, Import Batch > Select one of the file > Import as Filesystem
- Wait some minutes (note: it can crash)
- Export the files or just start the analysis on the executable/library you wish to.

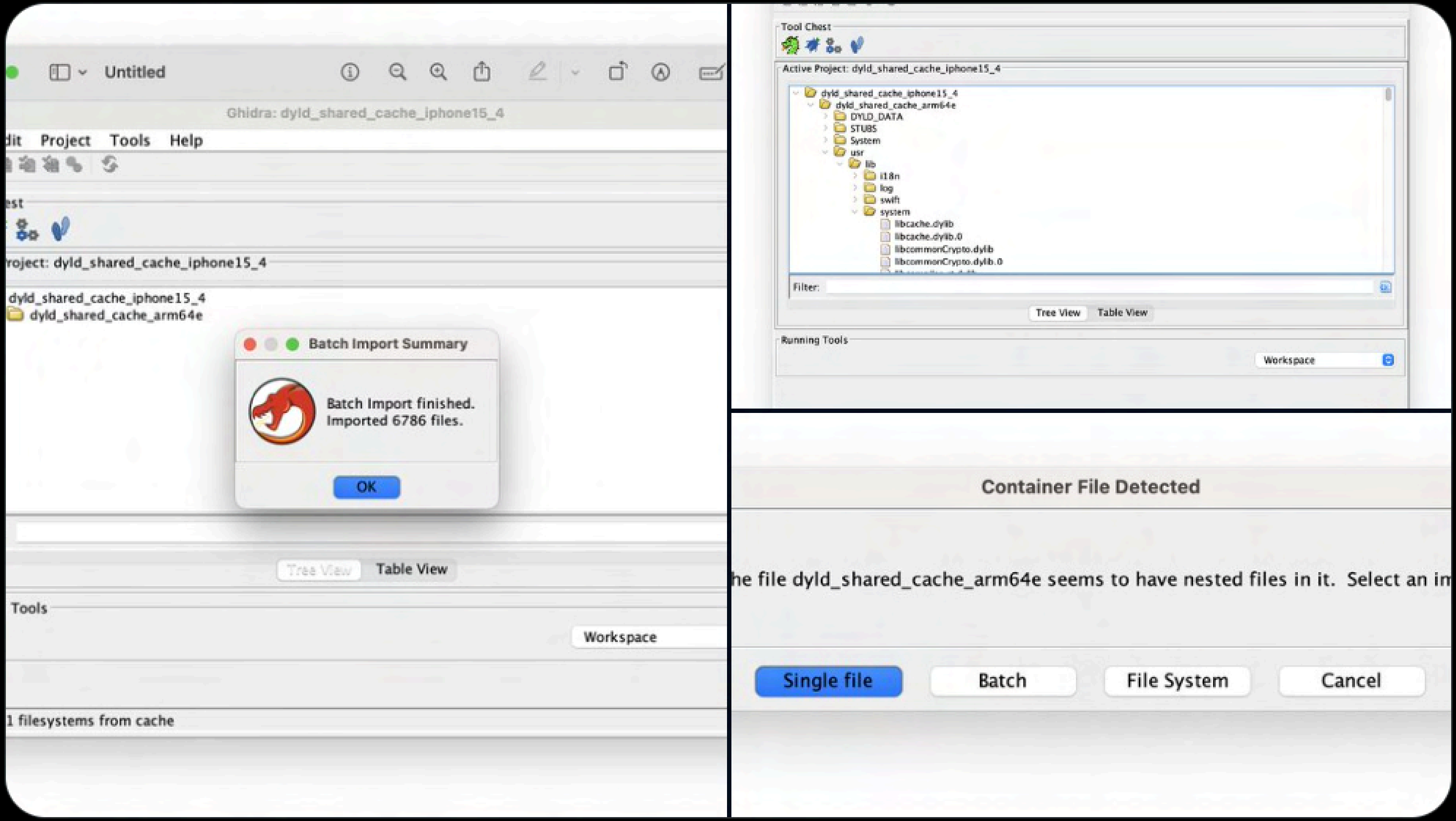
There are known issues with loading a shared cache.

https://theapplewiki.com/wiki/Dev:Dyld_shared_cache

clearbluejar
@clearbluejar

Can #ghidra parse and import the multi-GB iOS dyld_shared_cache?

Yes. Yes, it can. Although it may have taken all night, the import succeeded! 👀 🤯



Batch Import finished.
Imported 6786 files.

1 filesystems from cache

he file dyld_shared_cache_arm64e seems to have nested files in it. Select an im

10:24 AM · Feb 2, 2024 · **1,144** Views

Ghidra Analysis Quick Tip #1

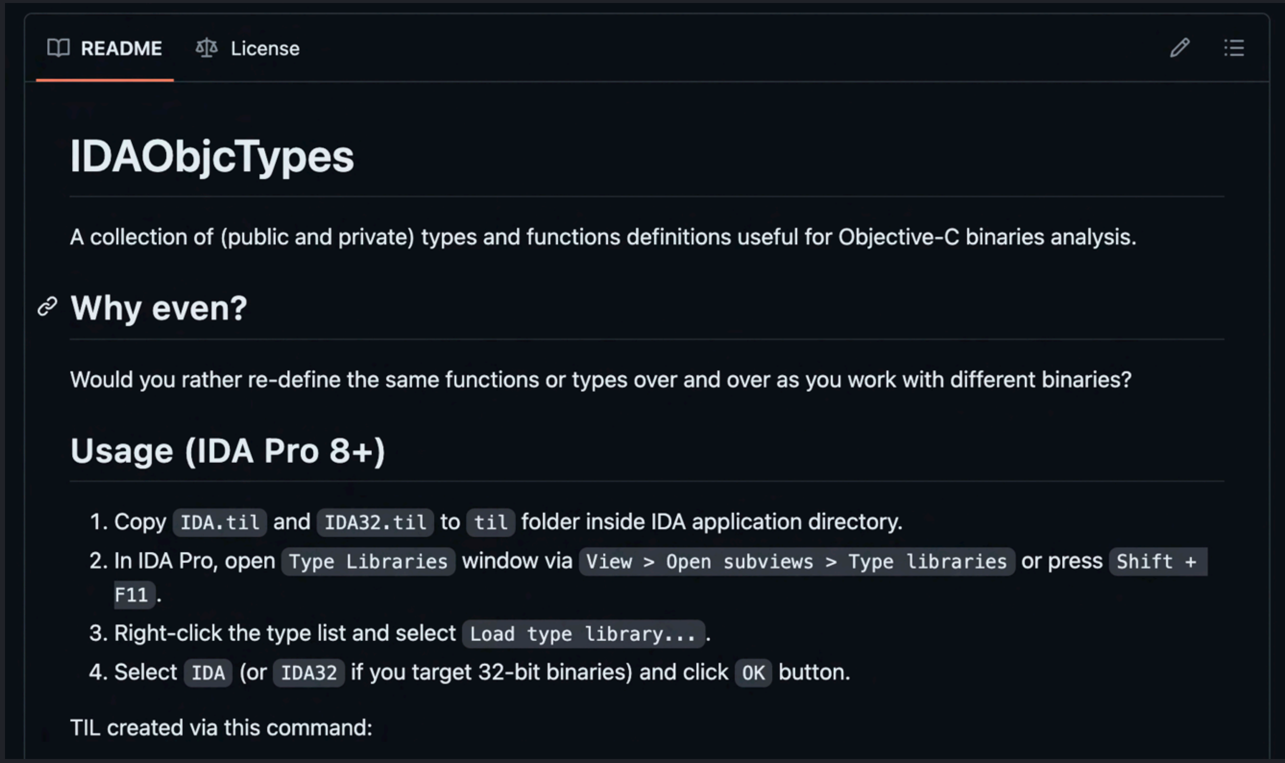
Decompiler Parameter ID - set to true!

https://gist.github.com/clearbluejar/1790c6fb2b01ec14c848d046a8a0199e/revisions

Ghidra Analysis Quick Tip #2

Leverage readily available type information

https://github.com/PoomSmart/IDAObjcTypes



README License

IDAObjcTypes

A collection of (public and private) types and functions definitions useful for Objective-C binaries analysis.

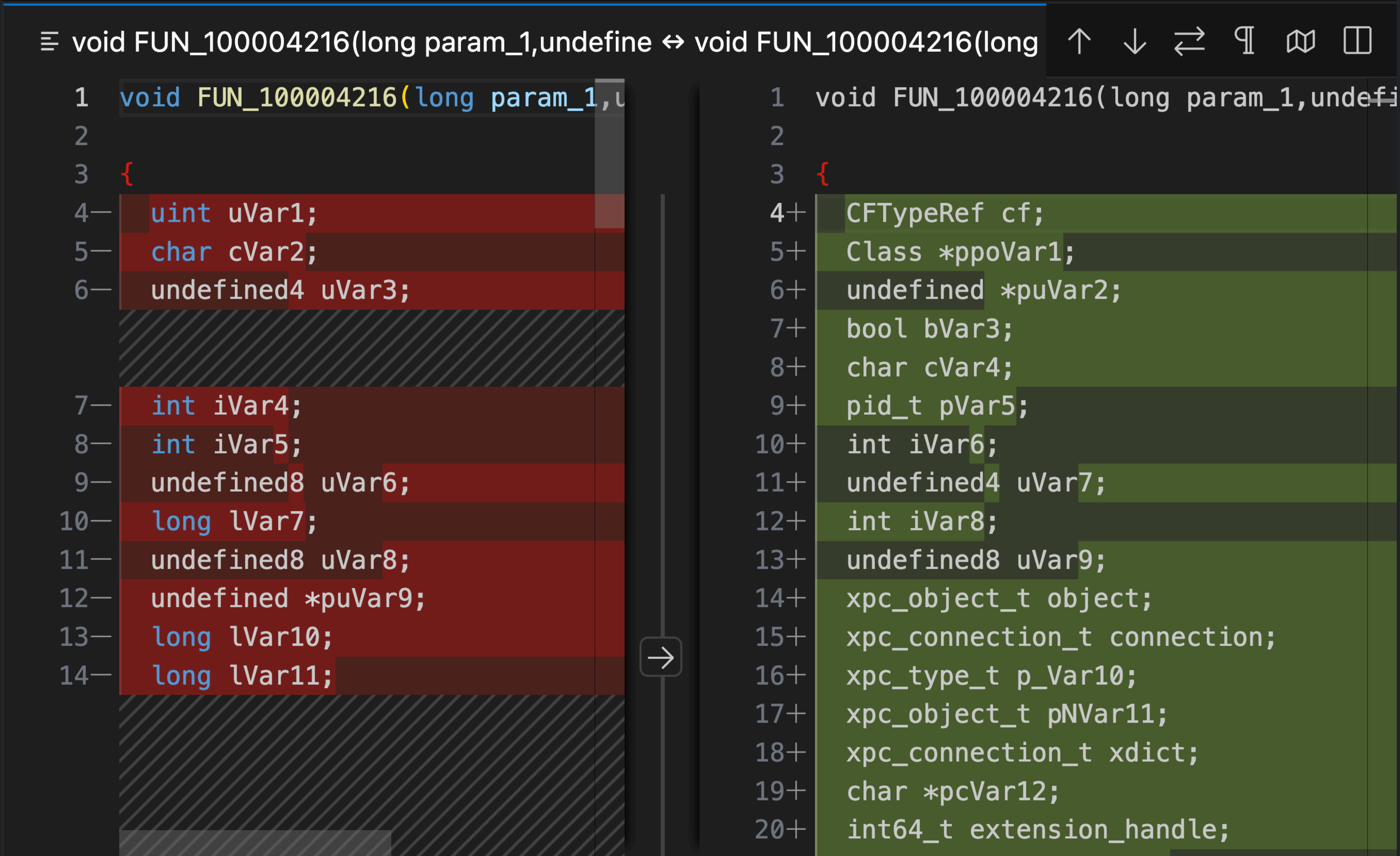
Why even?

Would you rather re-define the same functions or types over and over as you work with different binaries?

Usage (IDA Pro 8+)

1. Copy `IDA.til` and `IDA32.til` to `til` folder inside IDA application directory.
2. In IDA Pro, open `Type Libraries` window via `View > Open subviews > Type libraries` or press `Shift + F11`.
3. Right-click the type list and select `Load type library...`.
4. Select `IDA` (or `IDA32` if you target 32-bit binaries) and click `OK` button.

TIL created via this command:

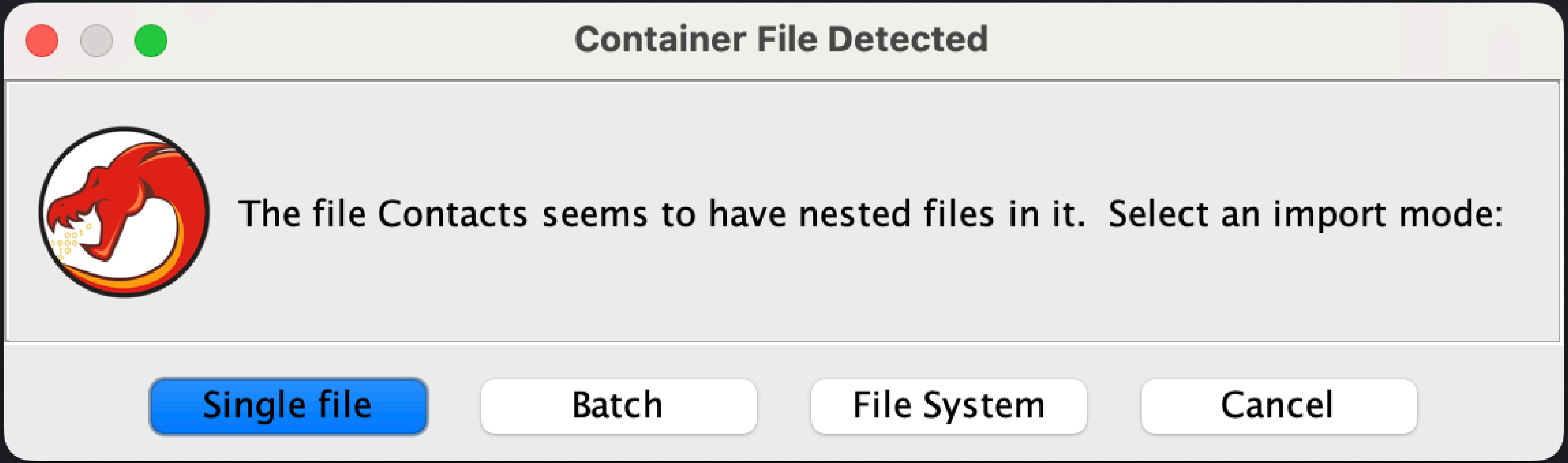


void FUN_100004216(long param_1,undefine ↔ void FUN_100004216(long

```
void FUN_100004216(long param_1,u

{
  uint uVar1;
  char cVar2;
  undefined4 uVar3;
  int iVar4;
  int iVar5;
  undefined8 uVar6;
  long lVar7;
  undefined8 uVar8;
  undefined *puVar9;
  long lVar10;
  long lVar11;
```

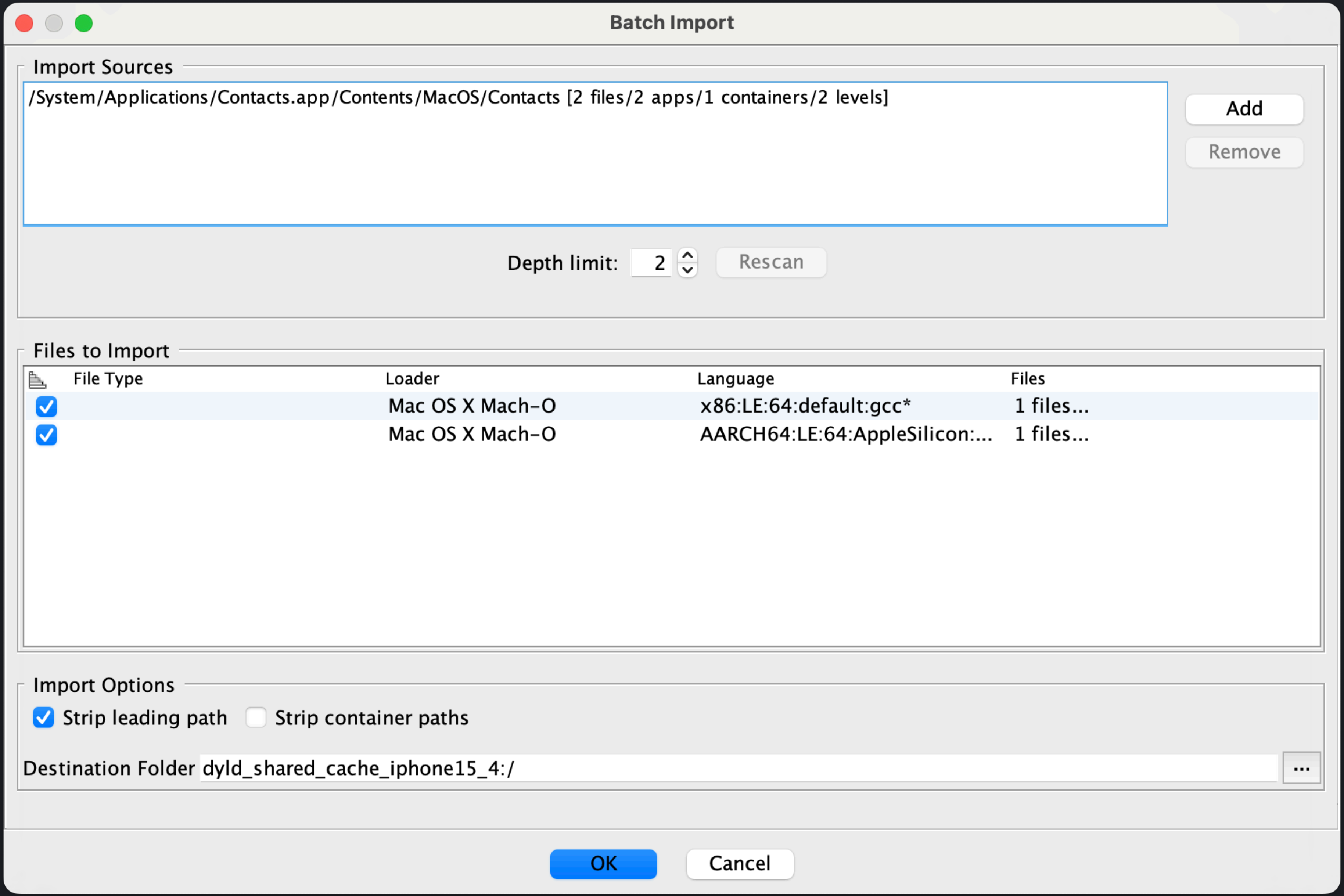
```
void FUN_100004216(long param_1,undefi

{
  CFTypeRef cf;
  Class *ppoVar1;
  undefined *puVar2;
  bool bVar3;
  char cVar4;
  pid_t pVar5;
  int iVar6;
  undefined4 uVar7;
  int iVar8;
  undefined8 uVar9;
  xpc_object_t object;
  xpc_connection_t connection;
  xpc_type_t p_Var10;
  xpc_object_t pNVar11;
  xpc_connection_t xdict;
  char *pcVar12;
  int64_t extension_handle;
```

Importing *OS Binaries



Container File Detected

The file Contacts seems to have nested files in it. Select an import mode:


Batch Import

Import Sources

/System/Applications/Contacts.app/Contents/MacOS/Contacts [2 files/2 apps/1 containers/2 levels]

Add
Remove

Depth limit: 2 Rescan

Files to Import

	File Type	Loader	Language	Files
✓		Mac OS X Mach-O	x86:LE:64:default:gcc*	1 files...
✓		Mac OS X Mach-O	AARCH64:LE:64:AppleSilicon:...	1 files...

Import Options

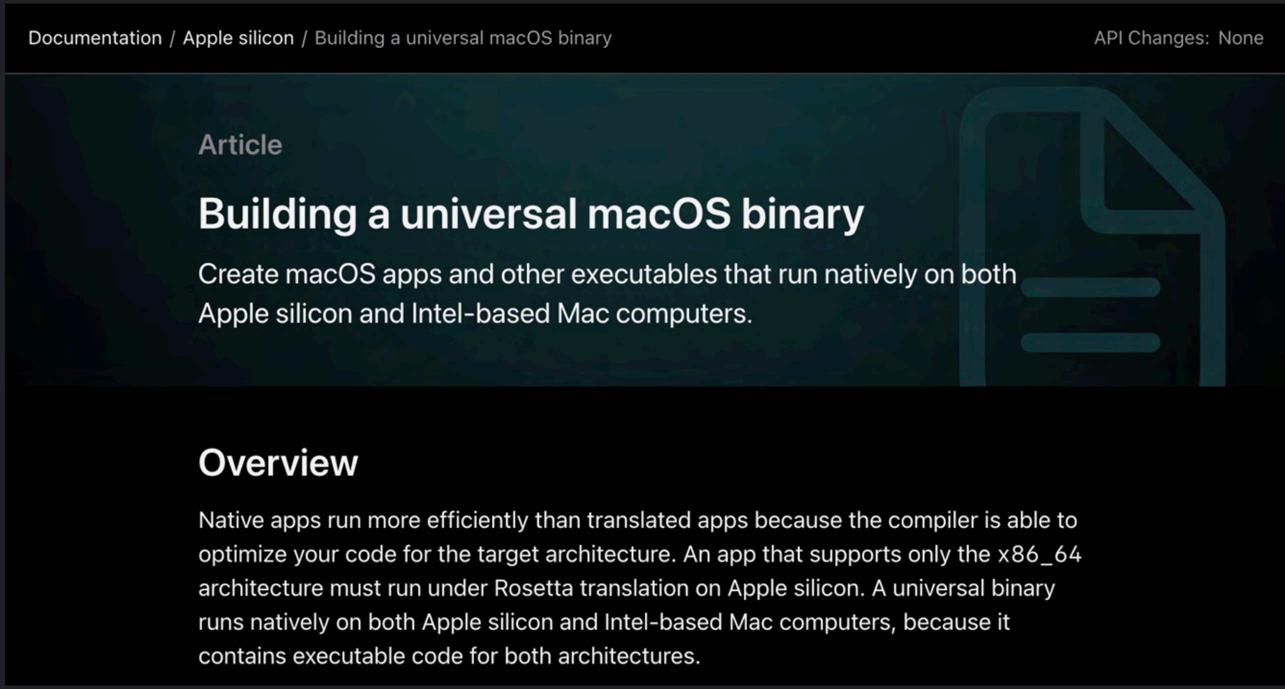
✓ Strip leading path ☐ Strip container paths

Destination Folder dyld_shared_cache_iphone15_4:/

OK Cancel

Oh right... Universal Binaries

https://developer.apple.com/documentation/apple-silicon/building-a-universal-macos-binary


Documentation / Apple silicon / Building a universal macOS binary

API Changes: None

Article

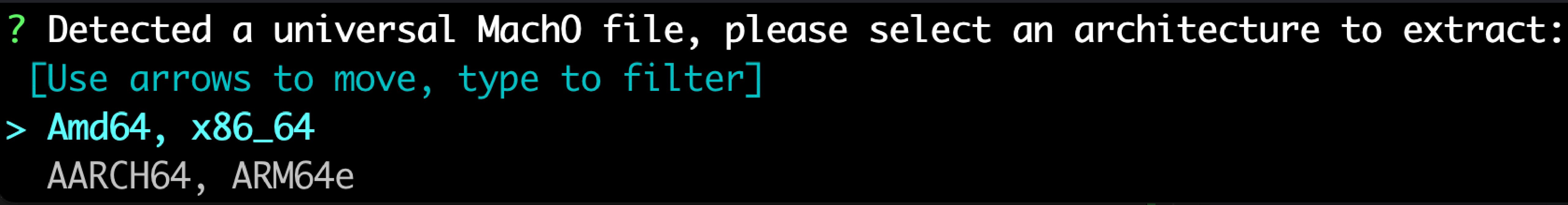
Building a universal macOS binary

Create macOS apps and other executables that run natively on both Apple silicon and Intel-based Mac computers.

Overview

Native apps run more efficiently than translated apps because the compiler is able to optimize your code for the target architecture. An app that supports only the x86_64 architecture must run under Rosetta translation on Apple silicon. A universal binary runs natively on both Apple silicon and Intel-based Mac computers, because it contains executable code for both architectures.

```
ipsw macho lipo Contacts
```



```
? Detected a universal MachO file, please select an architecture to extract:
 [Use arrows to move, type to filter]
> Amd64, x86_64
  AARCH64, ARM64e
```

Diffing the Binaries

0
0 0 1
1 0 0 0
1 1 0
0

`ghidriff`

https://github.com/clearbluejar/ghidriff

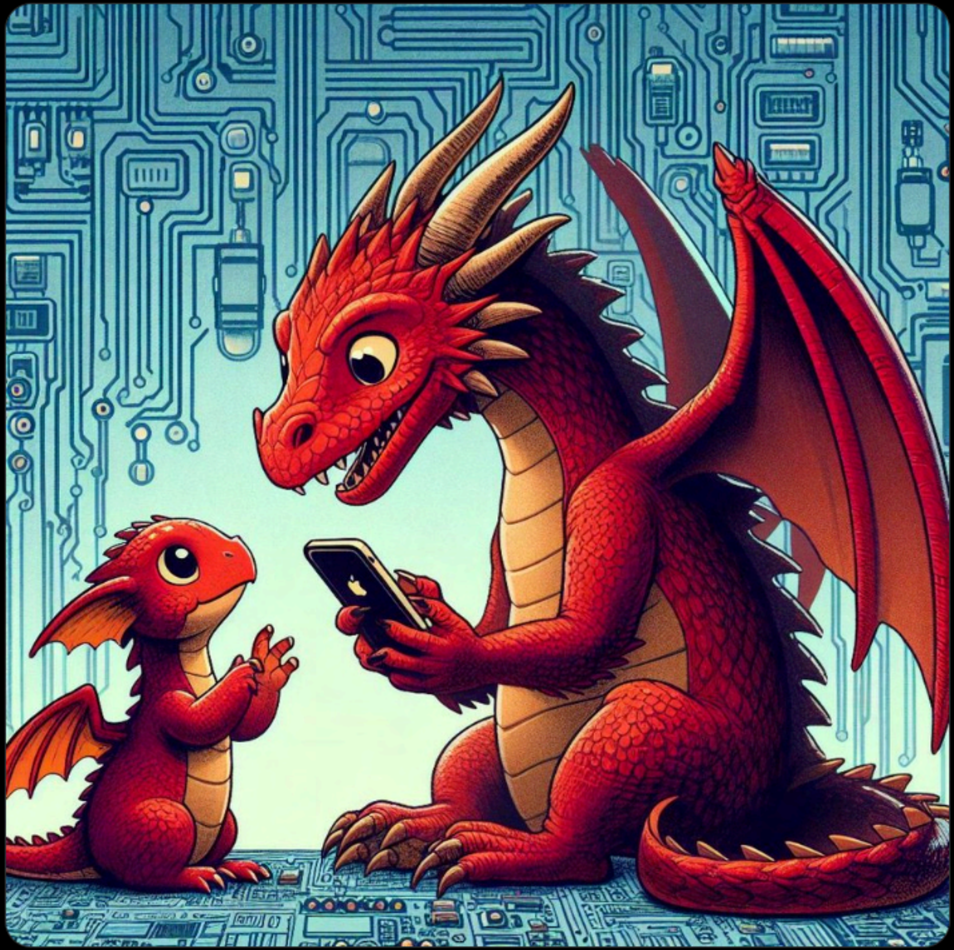
clearbluejar
@clearbluejar

starting to take a look at 🍎....

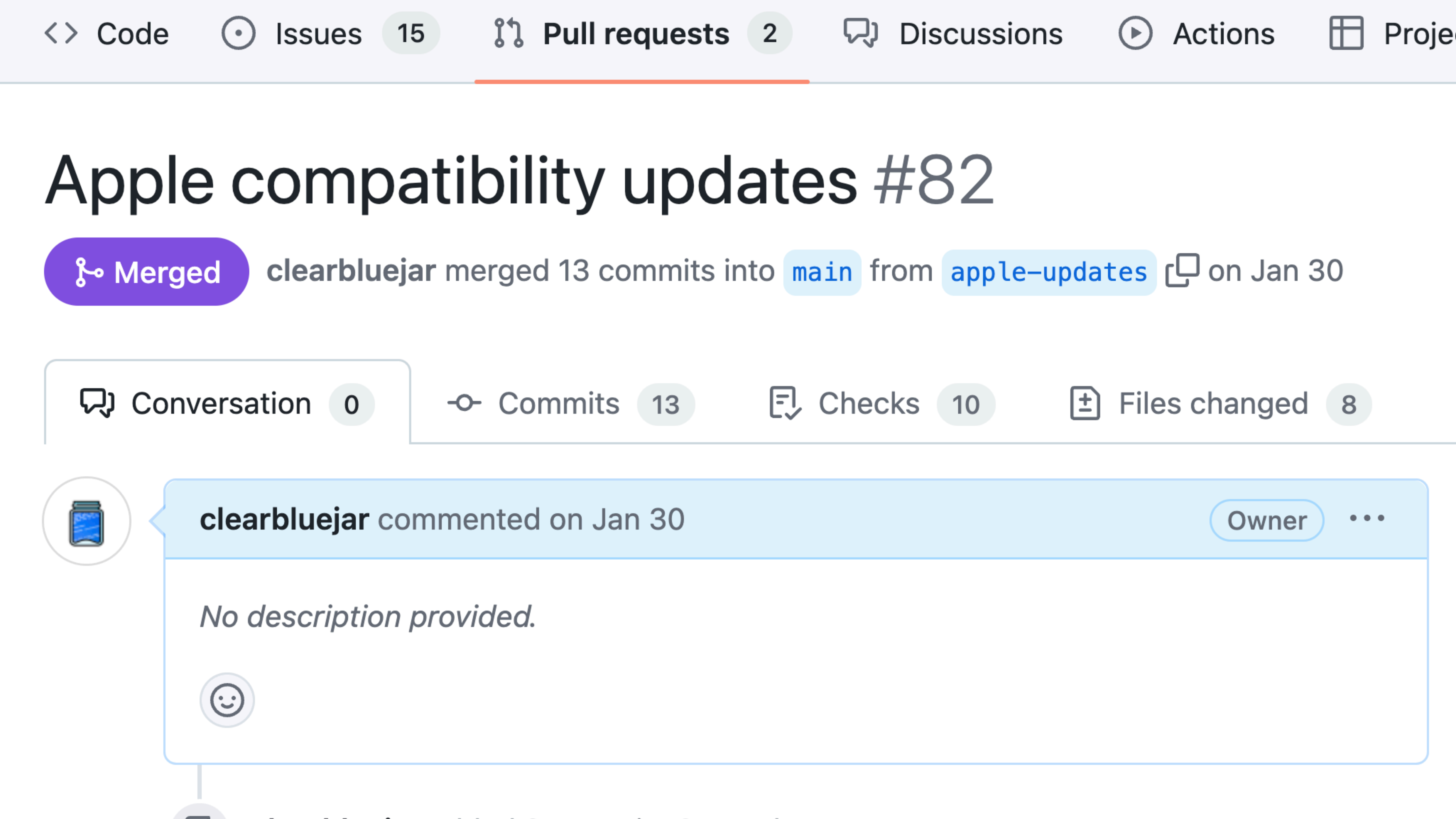
#ghidriff is slowly learning how to diff *OS 🤓

Some sample diffs incoming...

github.com/clearbluejar/g...



11:01 PM · Jan 31, 2024 · **722** Views



Apple compatibility updates #82

Merged clearbluejar merged 13 commits into `main` from `apple-updates` on Jan 30

clearbluejar commented on Jan 30 Owner

No description provided.

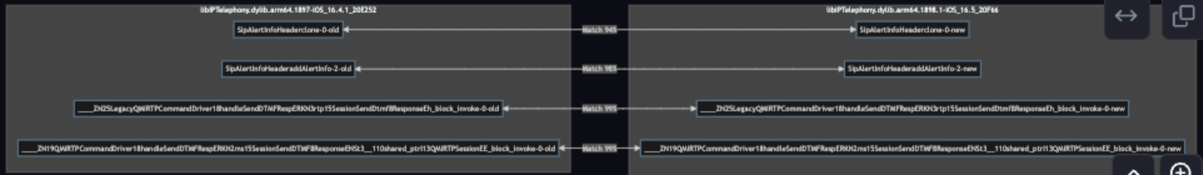
```
ghidriff <*OS old> <*OS new>
```

libIPTelephony.dylib.arm64.1897-iOS_16.4.1_20E252-libIPTelephony.dylib.arm64.1898.1-iOS_16.5_20F66.ghidriff.md Raw

libIPTelephony.dylib.arm64.1897-iOS_16.4.1_20E252-libIPTelephony.dylib.arm64.1898.1-iOS_16.5_20F66 Diff

TOC

Visual Chart Diff


libIPTelephony.dylib.arm64.1897-iOS_16.4.1_20E252
libIPTelephony.dylib.arm64.1898.1-iOS_16.5_20F66
SipAlertInfoHeaderclone-0-old
Match 94%
SipAlertInfoHeaderclone-0-new
SipAlertInfoHeaderaddAlertInfo-2-old
Match 98%
SipAlertInfoHeaderaddAlertInfo-2-new
____ZN25LegacyQMIRTPCommandDriver18handleSendDTMFRespERKN3rtp15SessionSendDtmf8ResponseEh_block_invoke-0-old
Match 99%
____ZN25LegacyQMIRTPCommandDriver18handleSendDTMFRespERKN3rtp15SessionSendDtmf8ResponseEh_block_invoke-0-new
____ZN19QMIRTPCommandDriver18handleSendDTMFRespERKN2ms15SessionSendDTMF8ResponseENSt3__110shared_ptrI13QMIRTPSessionEE_block_invoke-0-old
Match 99%
____ZN19QMIRTPCommandDriver18handleSendDTMFRespERKN2ms15SessionSendDTMF8ResponseENSt3__110shared_ptrI13QMIRTPSessionEE_block_invoke-0-new

High Level Patch Diffing on *OS

1. Identify a CVE of interest
2. Download corresponding IPSW update and N-1
3. Determine changes for update
4. Map binaries to CVE
5. Extract the related file(s)
6. Diff the binaries
7. Root cause the vulnerability

Root Causing *OS CVEs

How are Security Issues Fixed?

- Change integer types
- Add additional checks
- Change control flow
- Remove unsafe APIs
- OS level mitigations (ie SIP/Entitlements)
- Separation of privilege
- Declare not a secutiy boundary :)

Focus on the Changes

- code changes
- data changes
- additional entitlements
- etc

`ipsw` gives us high level visibility into which binaries have changed `ghidriff` gives us visibility into the code changes

CVE overview

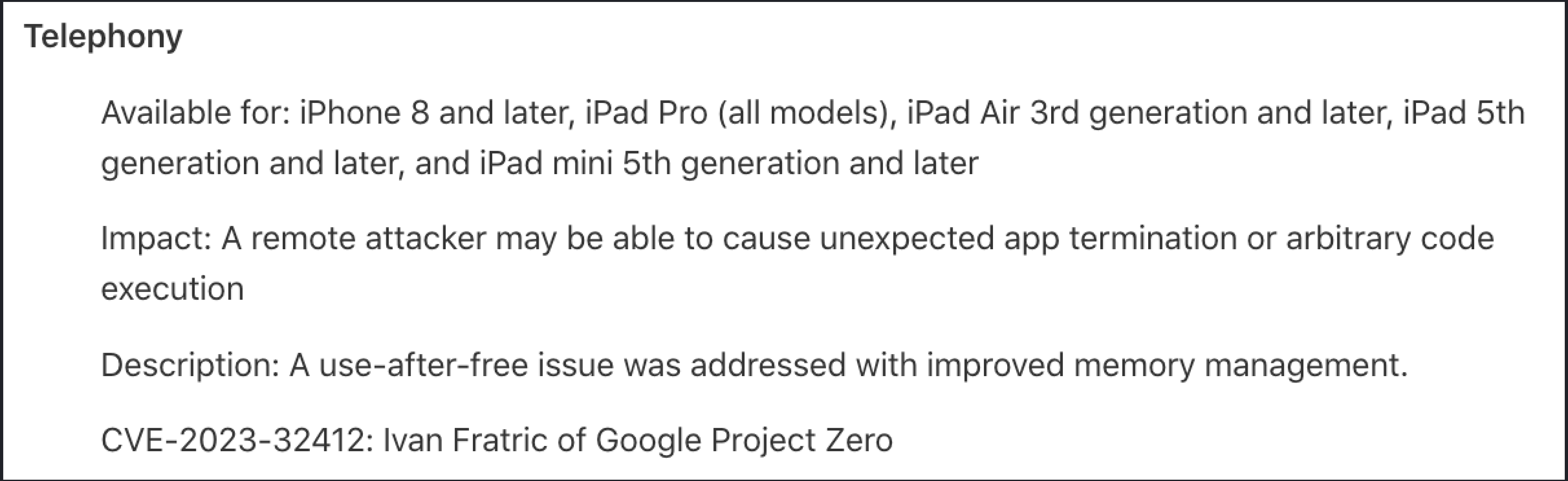
- CVE-2023-32412 - use after free
- CVE-2024-23218 - timing side-channel
- CVE-2023-42942 - toctou
- CVE-2022-46718 - sensitive data leak (??)
- CVE-2024-1580 - int overflow (??)

Identify CVE of interest

CVE-2023-32412

Apple Security Update - iOS / iPadOS 16.5

https://support.apple.com/en-us/HT213757



Telephony

Available for: iPhone 8 and later, iPad Pro (all models), iPad Air 3rd generation and later, iPad 5th generation and later, and iPad mini 5th generation and later

Impact: A remote attacker may be able to cause unexpected app termination or arbitrary code execution

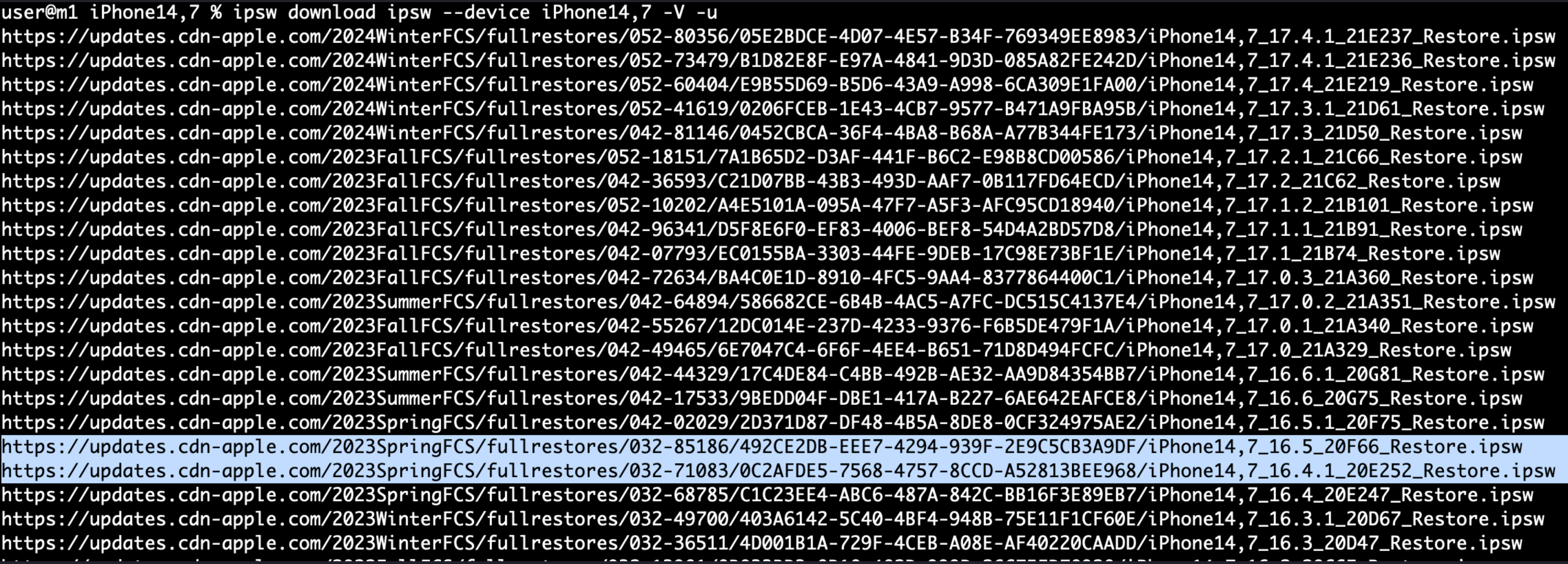
Description: A use-after-free issue was addressed with improved memory management.

CVE-2023-32412: Ivan Fratric of Google Project Zero

Download a corresponding IPSW and N-1

List all availables IPSW downloads

```
ipsw download ipsw --device iPhone14,7 -V -u
```



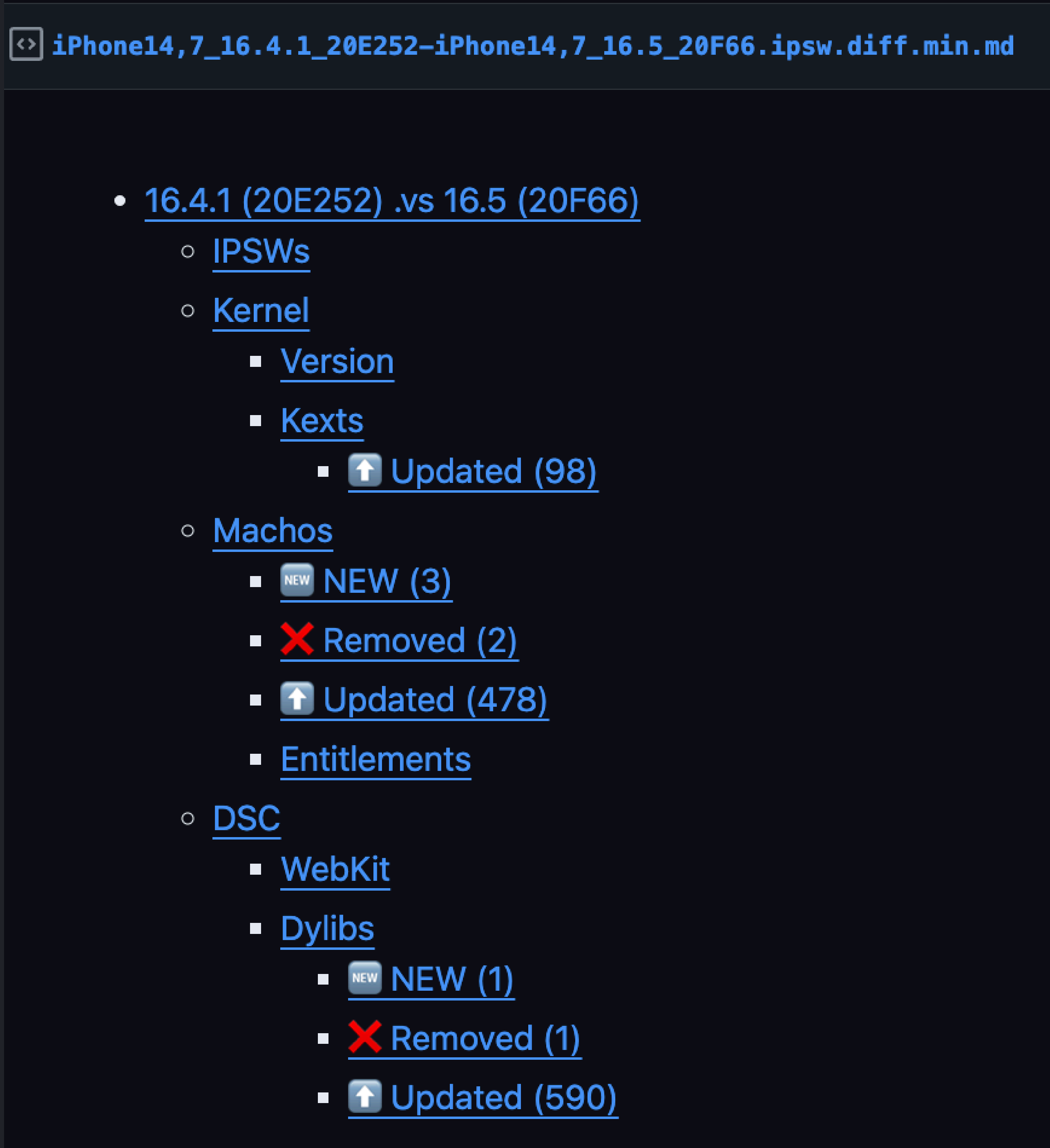
```
user@m1 iPhone14,7 % ipsw download ipsw --device iPhone14,7 -V -u
https://updates.cdn-apple.com/2024WinterFCS/fullrestores/052-80356/05E2BDCE-4D07-4E57-B34F-769349EE8983/iPhone14,7_17.4.1_21E237_Restore.ipsw
https://updates.cdn-apple.com/2024WinterFCS/fullrestores/052-73479/B1D82E8F-E97A-4841-9D3D-085A82FE242D/iPhone14,7_17.4.1_21E236_Restore.ipsw
https://updates.cdn-apple.com/2024WinterFCS/fullrestores/052-60404/E9B55D69-B5D6-43A9-A998-6CA309E1FA00/iPhone14,7_17.4_21E219_Restore.ipsw
https://updates.cdn-apple.com/2024WinterFCS/fullrestores/052-41619/0206FCEB-1E43-4CB7-9577-B471A9FBA95B/iPhone14,7_17.3.1_21D61_Restore.ipsw
https://updates.cdn-apple.com/2024WinterFCS/fullrestores/042-81146/0452CBCA-36F4-4BA8-B68A-A77B344FE173/iPhone14,7_17.3_21D50_Restore.ipsw
https://updates.cdn-apple.com/2023FallFCS/fullrestores/052-18151/7A1B65D2-D3AF-441F-B6C2-E98B8CD00586/iPhone14,7_17.2.1_21C66_Restore.ipsw
https://updates.cdn-apple.com/2023FallFCS/fullrestores/042-36593/C21D07BB-43B3-493D-AAF7-0B117FD64ECD/iPhone14,7_17.2_21C62_Restore.ipsw
https://updates.cdn-apple.com/2023FallFCS/fullrestores/052-10202/A4E5101A-095A-47F7-A5F3-AFC95CD18940/iPhone14,7_17.1.2_21B101_Restore.ipsw
https://updates.cdn-apple.com/2023FallFCS/fullrestores/042-96341/D5F8E6F0-EF83-4006-BEF8-54D4A2BD57D8/iPhone14,7_17.1.1_21B91_Restore.ipsw
https://updates.cdn-apple.com/2023FallFCS/fullrestores/042-07793/EC0155BA-3303-44FE-9DEB-17C98E73BF1E/iPhone14,7_17.1_21B74_Restore.ipsw
https://updates.cdn-apple.com/2023FallFCS/fullrestores/042-72634/BA4C0E1D-8910-4FC5-9AA4-8377864400C1/iPhone14,7_17.0.3_21A360_Restore.ipsw
https://updates.cdn-apple.com/2023SummerFCS/fullrestores/042-64894/586682CE-6B4B-4AC5-A7FC-DC515C4137E4/iPhone14,7_17.0.2_21A351_Restore.ipsw
https://updates.cdn-apple.com/2023FallFCS/fullrestores/042-55267/12DC014E-237D-4233-9376-F6B5DE479F1A/iPhone14,7_17.0.1_21A340_Restore.ipsw
https://updates.cdn-apple.com/2023FallFCS/fullrestores/042-49465/6E7047C4-6F6F-4EE4-B651-71D8D494FCFC/iPhone14,7_17.0_21A329_Restore.ipsw
https://updates.cdn-apple.com/2023SummerFCS/fullrestores/042-44329/17C4DE84-C4BB-492B-AE32-AA9D84354BB7/iPhone14,7_16.6.1_20G81_Restore.ipsw
https://updates.cdn-apple.com/2023SummerFCS/fullrestores/042-17533/9BEDD04F-DBE1-417A-B227-6AE642EAFCE8/iPhone14,7_16.6_20G75_Restore.ipsw
https://updates.cdn-apple.com/2023SpringFCS/fullrestores/042-02029/2D371D87-DF48-4B5A-8DE8-0CF324975AE2/iPhone14,7_16.5.1_20F75_Restore.ipsw
https://updates.cdn-apple.com/2023SpringFCS/fullrestores/032-85186/492CE2DB-EEE7-4294-939F-2E9C5CB3A9DF/iPhone14,7_16.5_20F66_Restore.ipsw
https://updates.cdn-apple.com/2023SpringFCS/fullrestores/032-71083/0C2AFDE5-7568-4757-8CCD-A52813BEE968/iPhone14,7_16.4.1_20E252_Restore.ipsw
https://updates.cdn-apple.com/2023SpringFCS/fullrestores/032-68785/C1C23EE4-ABC6-487A-842C-BB16F3E89EB7/iPhone14,7_16.4_20E247_Restore.ipsw
https://updates.cdn-apple.com/2023WinterFCS/fullrestores/032-49700/403A6142-5C40-4BF4-948B-75E11F1CF60E/iPhone14,7_16.3.1_20D67_Restore.ipsw
https://updates.cdn-apple.com/2023WinterFCS/fullrestores/032-36511/4D001B1A-729F-4CEB-A08E-AF40220CAADD/iPhone14,7_16.3_20D47_Restore.ipsw
```

- 16.5 (Patched)
  - `ipsw download ipsw --device iPhone14,7 -b 20F66`
- 16.4.1 (N-1)
  - `ipsw download ipsw --device iPhone14,7 -b 20E252`

Determine changes for update

```
ipsw diff
iPhone14,7_16.4.1_20E252_Restore.ipsw
iPhone14,7_16.5_20F66_Restore.ipsw
```

16.4.1 (20E252) .vs 16.5 (20F66)



iPhone14,7_16.4.1_20E252-iPhone14,7_16.5_20F66.ipsw.diff.min.md

- 16.4.1 (20E252) .vs 16.5 (20F66)
  - IPSWs
  - Kernel
    - Version
    - Kexts
      - ⬆️ Updated (98)
  - Machos
    - 🆕 NEW (3)
    - ❌ Removed (2)
    - ⬆️ Updated (478)
    - Entitlements
  - DSC
    - WebKit
    - Dylibs
      - 🆕 NEW (1)
      - ❌ Removed (1)
      - ⬆️ Updated (590)

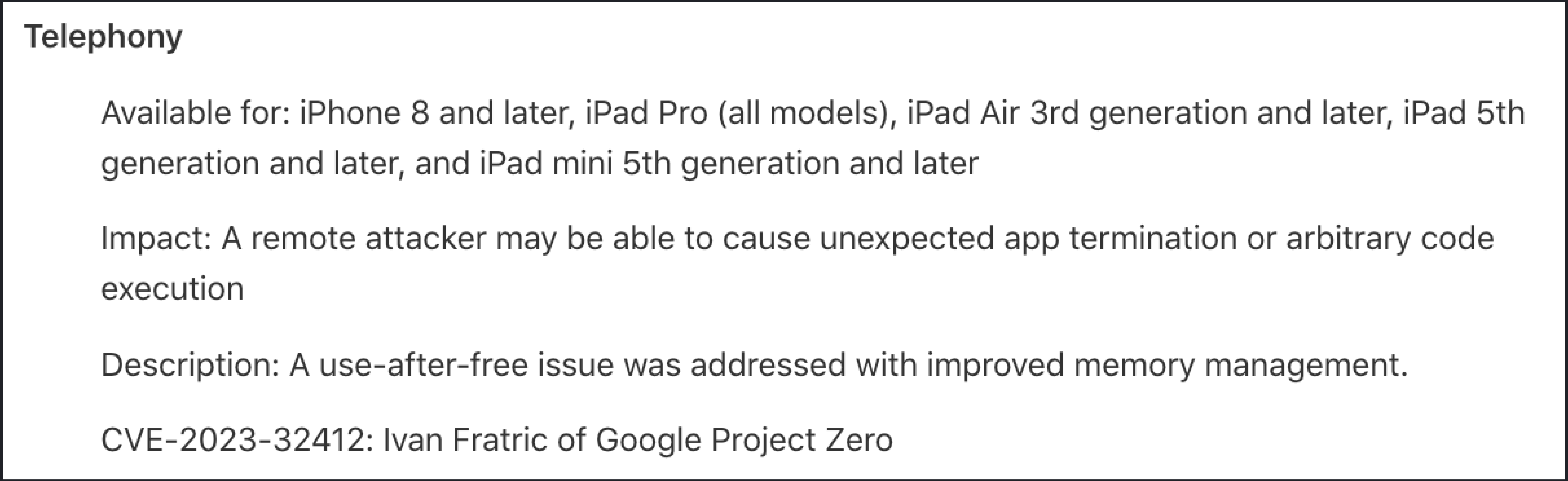
Map binaries to CVE

How do you know which binaries map to a CVE?

You don't

The Power of Diffing

- Will not tell you which functions/binaries are relevant
- Will focus you on the functions/binaries that could be



Telephony

Available for: iPhone 8 and later, iPad Pro (all models), iPad Air 3rd generation and later, iPad 5th generation and later, and iPad mini 5th generation and later

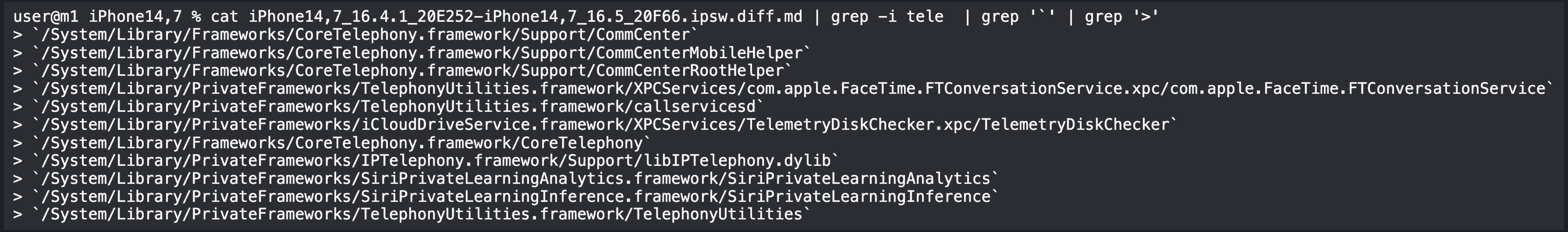
Impact: A remote attacker may be able to cause unexpected app termination or arbitrary code execution

Description: A use-after-free issue was addressed with improved memory management.

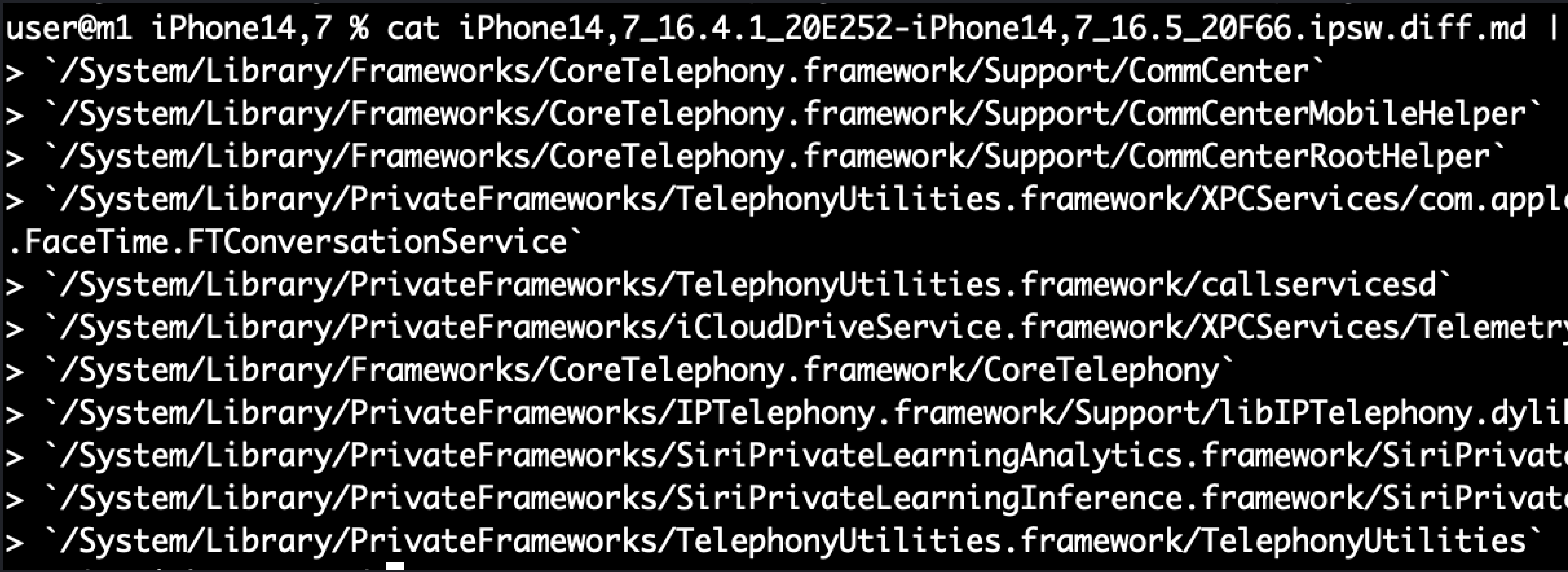
CVE-2023-32412: Ivan Fratric of Google Project Zero

https://support.apple.com/en-us/HT213757

`cat iPhone14,7_16.4.1_20E252-iPhone14,7_16.5_20F66.ipsw.diff.md | grep -i tele`



```
user@m1 iPhone14,7 % cat iPhone14,7_16.4.1_20E252-iPhone14,7_16.5_20F66.ipsw.diff.md | grep -i tele  | grep '`' | grep '>'
> `/System/Library/Frameworks/CoreTelephony.framework/Support/CommCenter`
> `/System/Library/Frameworks/CoreTelephony.framework/Support/CommCenterMobileHelper`
> `/System/Library/Frameworks/CoreTelephony.framework/Support/CommCenterRootHelper`
> `/System/Library/PrivateFrameworks/TelephonyUtilities.framework/XPCServices/com.apple.FaceTime.FTConversationService.xpc/com.apple.FaceTime.FTConversationService`
> `/System/Library/PrivateFrameworks/TelephonyUtilities.framework/callservicesd`
> `/System/Library/PrivateFrameworks/iCloudDriveService.framework/XPCServices/TelemetryDiskChecker.xpc/TelemetryDiskChecker`
> `/System/Library/Frameworks/CoreTelephony.framework/CoreTelephony`
> `/System/Library/PrivateFrameworks/IPTelephony.framework/Support/libIPTelephony.dylib`
> `/System/Library/PrivateFrameworks/SiriPrivateLearningAnalytics.framework/SiriPrivateLearningAnalytics`
> `/System/Library/PrivateFrameworks/SiriPrivateLearningInference.framework/SiriPrivateLearningInference`
> `/System/Library/PrivateFrameworks/TelephonyUtilities.framework/TelephonyUtilities`
```



```
user@m1 iPhone14,7 % cat iPhone14,7_16.4.1_20E252-iPhone14,7_16.5_20F66.ipsw.diff.md |
> `/System/Library/Frameworks/CoreTelephony.framework/Support/CommCenter`
> `/System/Library/Frameworks/CoreTelephony.framework/Support/CommCenterMobileHelper`
> `/System/Library/Frameworks/CoreTelephony.framework/Support/CommCenterRootHelper`
> `/System/Library/PrivateFrameworks/TelephonyUtilities.framework/XPCServices/com.appl
.FaceTime.FTConversationService`
> `/System/Library/PrivateFrameworks/TelephonyUtilities.framework/callservicesd`
> `/System/Library/PrivateFrameworks/iCloudDriveService.framework/XPCServices/Telemetr
> `/System/Library/Frameworks/CoreTelephony.framework/CoreTelephony`
> `/System/Library/PrivateFrameworks/IPTelephony.framework/Support/libIPTelephony.dyli
> `/System/Library/PrivateFrameworks/SiriPrivateLearningAnalytics.framework/SiriPrivat
> `/System/Library/PrivateFrameworks/SiriPrivateLearningInference.framework/SiriPrivat
> `/System/Library/PrivateFrameworks/TelephonyUtilities.framework/TelephonyUtilities`
```

Extract the related file(s)

```
cat iPhone14,7_16.4.1_20E252-iPhone14,7_16.5_20F66.ipsw.diff.md | egrep -i "tele|updated|kernel|machos|DSC" | egrep -i "#|`" > /tmp/changed-files.md
```

MachOs

⬆️ Updated

> `/System/Library/Frameworks/CoreTelephony.framework/Support/CommCenter`
> `/System/Library/Frameworks/CoreTelephony.framework/Support/CommCenterMobileHelper`
> `/System/Library/Frameworks/CoreTelephony.framework/Support/CommCenterRootHelper`

`softwareupdated`

> `/System/Library/PrivateFrameworks/MobileSoftwareUpdate.framework/Support/softwareupdated`
> `/System/Library/PrivateFrameworks/TelephonyUtilities.framework/XPCServices/com.apple.FaceTime.FTConversationService.xpc/com.apple.FaceTime.FTConversationService`
> `/System/Library/PrivateFrameworks/TelephonyUtilities.framework/callservicesd`

`TelemetryDiskChecker`

> `/System/Library/PrivateFrameworks/iCloudDriveService.framework/XPCServices/TelemetryDiskChecker.xpc/TelemetryDiskChecker`

DSC

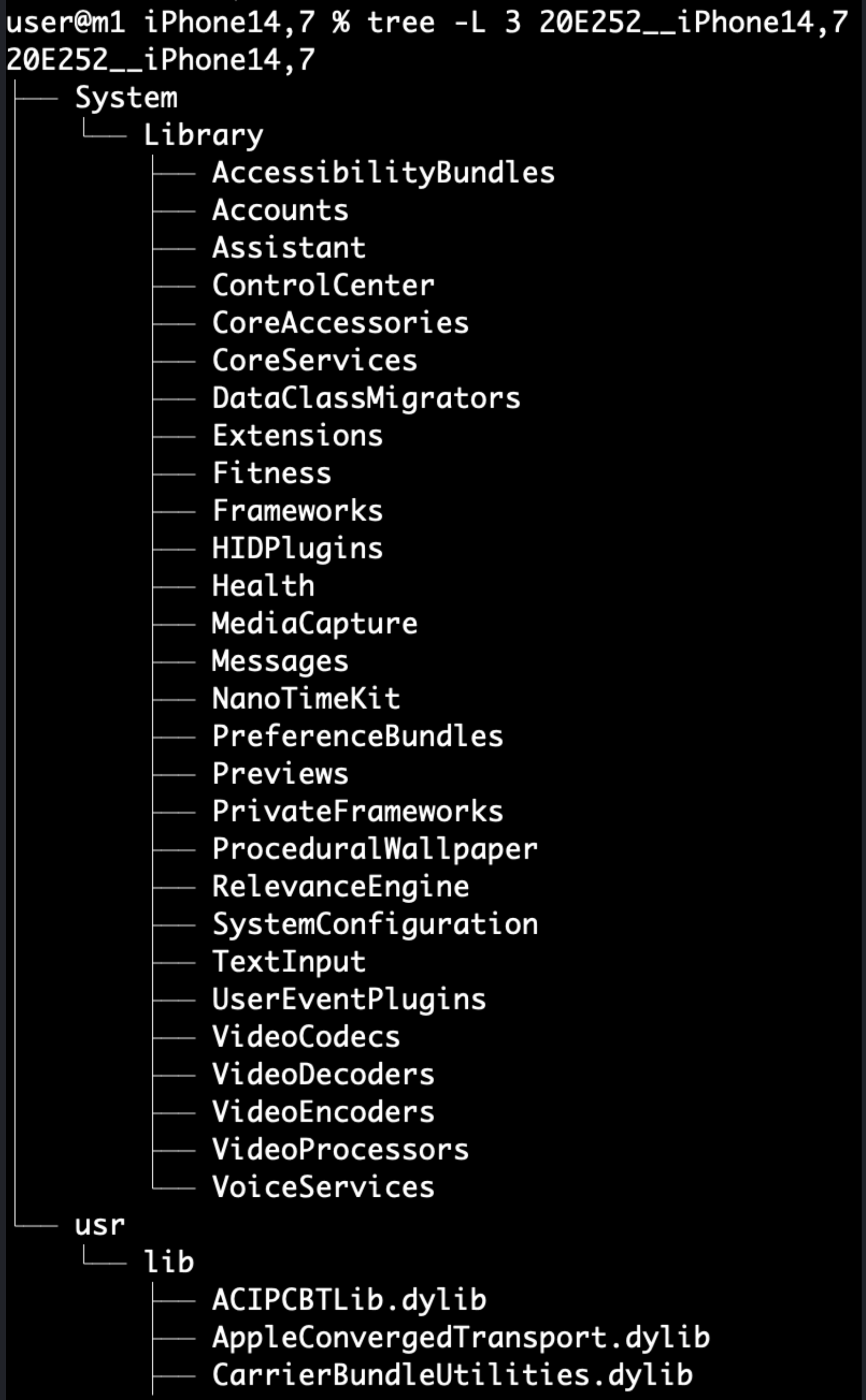
⬆️ Updated

`CoreTelephony`

> `/System/Library/Frameworks/CoreTelephony.framework/CoreTelephony`

`libIPTelephony.dylib`

- Extract the DSC
  - `ipsw extract -d <IPSW>`
- Extract all the files - `ipsw dyld split dyld_shared_cache_arm6`

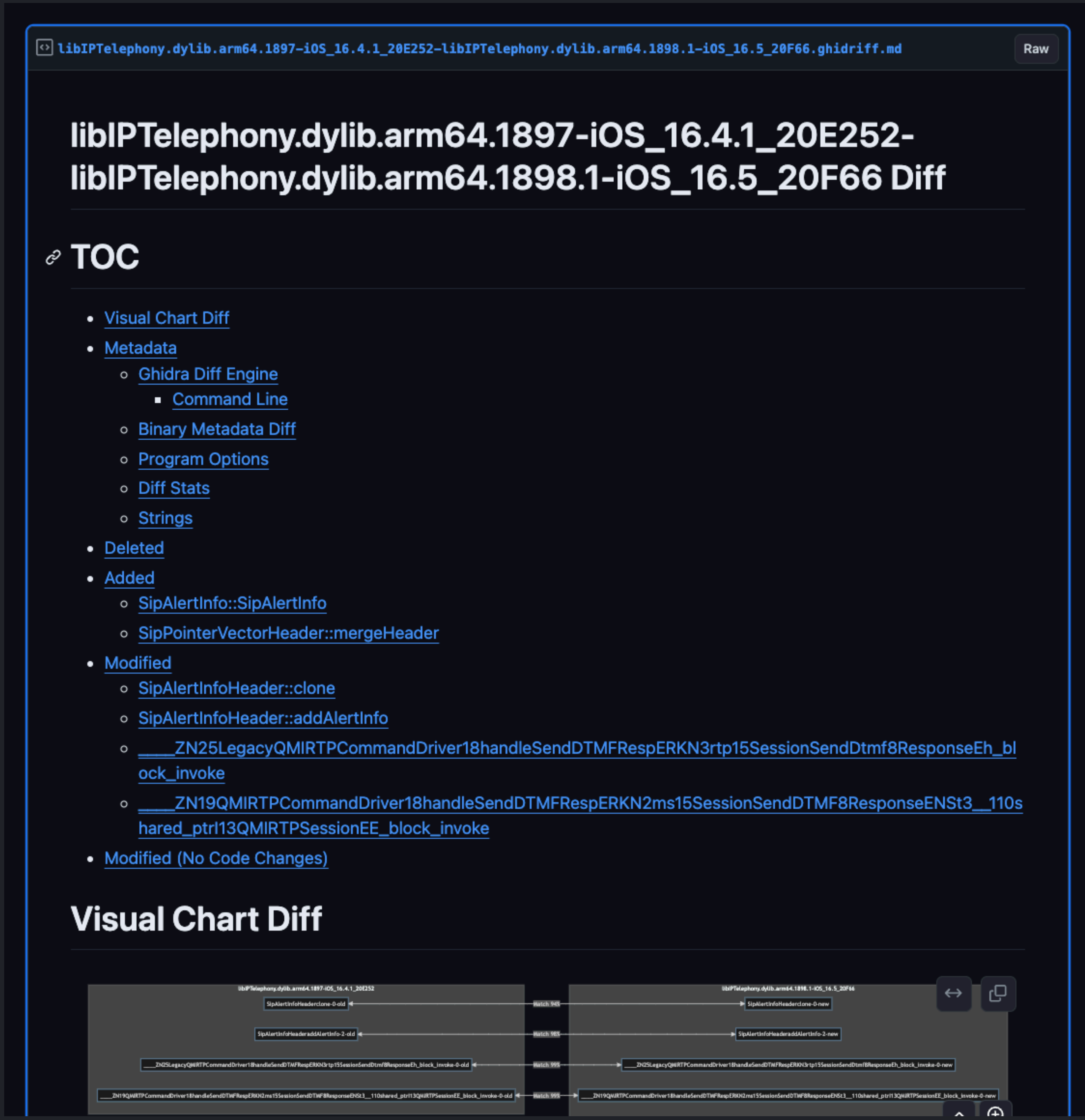


```
user@m1 iPhone14,7 % tree -L 3 20E252__iPhone14,7
20E252__iPhone14,7
├── System
│   └── Library
│       ├── AccessibilityBundles
│       ├── Accounts
│       ├── Assistant
│       ├── ControlCenter
│       ├── CoreAccessories
│       ├── CoreServices
│       ├── DataClassMigrators
│       ├── Extensions
│       ├── Fitness
│       ├── Frameworks
│       ├── HIDPlugins
│       ├── Health
│       ├── MediaCapture
│       ├── Messages
│       ├── NanoTimeKit
│       ├── PreferenceBundles
│       ├── Previews
│       ├── PrivateFrameworks
│       ├── ProceduralWallpaper
│       ├── RelevanceEngine
│       ├── SystemConfiguration
│       ├── TextInput
│       ├── UserEventPlugins
│       ├── VideoCodecs
│       ├── VideoDecoders
│       ├── VideoEncoders
│       ├── VideoProcessors
│       └── VoiceServices
└── usr
    └── lib
        ├── ACIPCBTLib.dylib
        ├── AppleConvergedTransport.dylib
        ├── CarrierBundleUtilities.dylib
```

Diff the CVE Binaries

```
% ghidriff libIPTelephony.dylib.arm64.1897-iOS_16.4.1_20E252 libIPTelephony.dylib.arm64.1898.1-iOS_16.5_20F66
```

libIPTelephony.dylib.arm64.1897-iOS_16.4.1_20E252-libIPTelephony.dylib.arm64.1898.1-iOS_16.5_20F66.ghidriff.md


libIPTelephony.dylib.arm64.1897-iOS_16.4.1_20E252-libIPTelephony.dylib.arm64.1898.1-iOS_16.5_20F66.ghidriff.md
Raw

libIPTelephony.dylib.arm64.1897-iOS_16.4.1_20E252-libIPTelephony.dylib.arm64.1898.1-iOS_16.5_20F66 Diff

TOC

Visual Chart Diff

We're gonna do this.

Root Cause the CVE

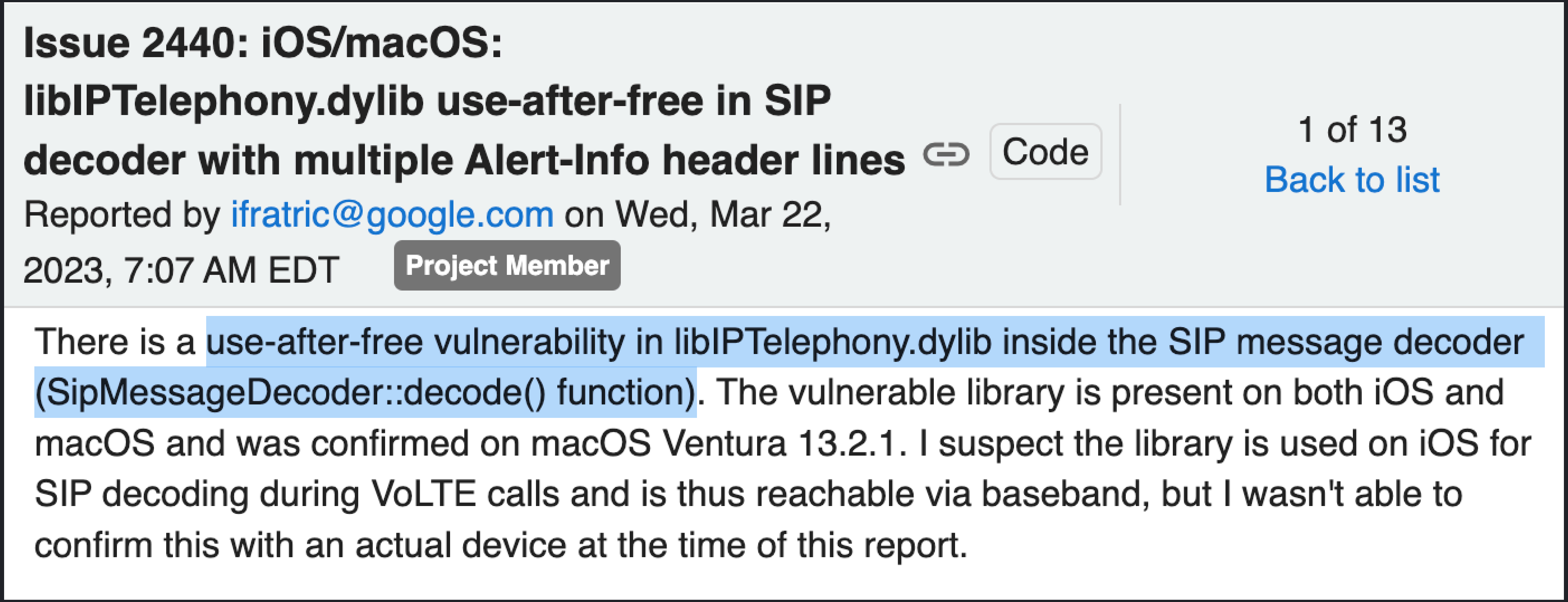
CVE-2023-32412

Telephony

Impact: A remote attacker may be able to cause unexpected app termination or arbitrary code execution

A use-after-free issue was addressed with improved memory management.

CVE-2023-32412: Ivan Fratric of Google Project Zero

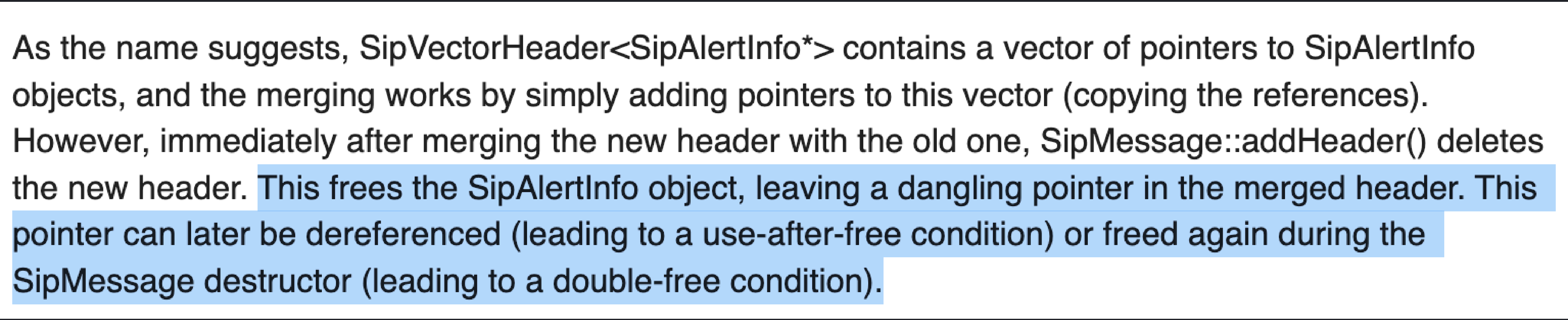


Issue 2440: iOS/macOS: libIPTelephony.dylib use-after-free in SIP decoder with multiple Alert-Info header lines

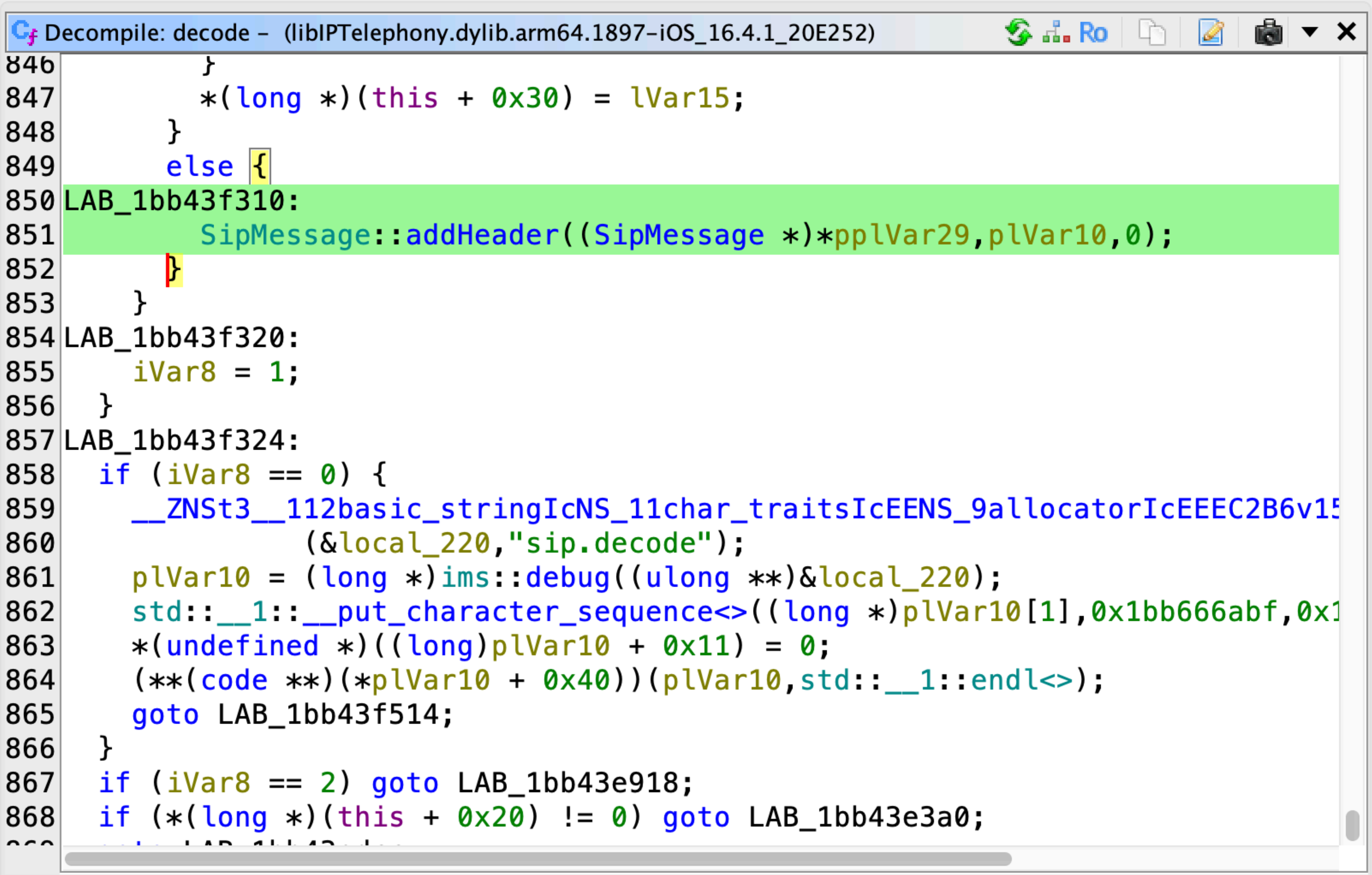
Reported by ifratric@google.com on Wed, Mar 22, 2023, 7:07 AM EDT

There is a use-after-free vulnerability in libIPTelephony.dylib inside the SIP message decoder (SipMessageDecoder::decode() function). The vulnerable library is present on both iOS and macOS and was confirmed on macOS Ventura 13.2.1. I suspect the library is used on iOS for SIP decoding during VoLTE calls and is thus reachable via baseband, but I wasn't able to confirm this with an actual device at the time of this report.

https://bugs.chromium.org/p/project-zero/issues/detail?id=2440

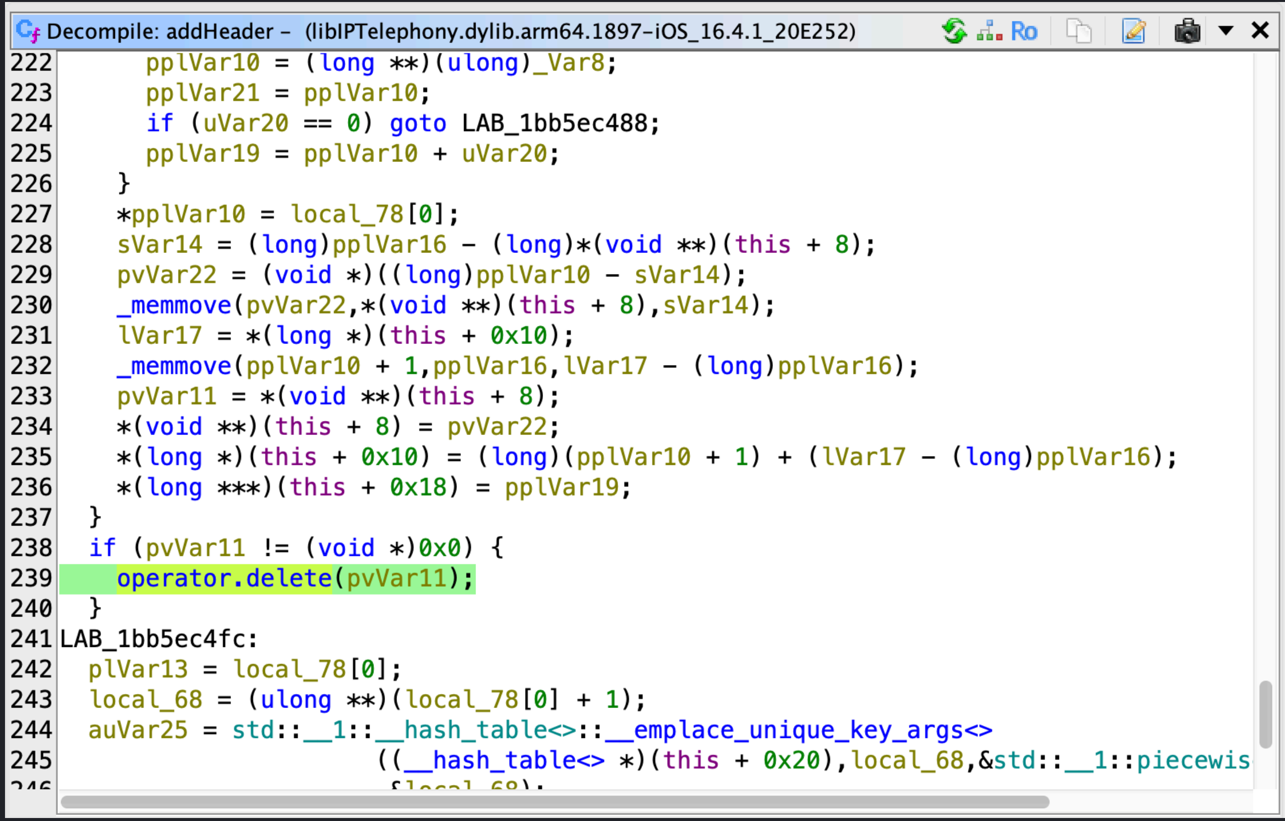


As the name suggests, SipVectorHeader<SipAlertInfo*> contains a vector of pointers to SipAlertInfo objects, and the merging works by simply adding pointers to this vector (copying the references). However, immediately after merging the new header with the old one, SipMessage::addHeader() deletes the new header. This frees the SipAlertInfo object, leaving a dangling pointer in the merged header. This pointer can later be dereferenced (leading to a use-after-free condition) or freed again during the SipMessage destructor (leading to a double-free condition).


Decompile: decode - (libIPTelephony.dylib.arm64.1897-iOS_16.4.1_20E252)

```
        }
        *(long *)(this + 0x30) = lVar15;
      }
      else {
LAB_1bb43f310:
        SipMessage::addHeader((SipMessage *)*pplVar29,plVar10,0);
      }
    }
LAB_1bb43f320:
    iVar8 = 1;
  }
LAB_1bb43f324:
  if (iVar8 == 0) {
    __ZNSt3__112basic_stringIcNS_11char_traitsIcEENS_9allocatorIcEEEC2B6v15
             (&local_220,"sip.decode");
    plVar10 = (long *)ims::debug((ulong **)&local_220);
    std::__1::__put_character_sequence<>((long *)plVar10[1],0x1bb666abf,0x1
    *(undefined *)((long)plVar10 + 0x11) = 0;
    (**(code **)(*plVar10 + 0x40))(plVar10,std::__1::endl<>);
    goto LAB_1bb43f514;
  }
  if (iVar8 == 2) goto LAB_1bb43e918;
  if (*(long *)(this + 0x20) != 0) goto LAB_1bb43e3a0;
```

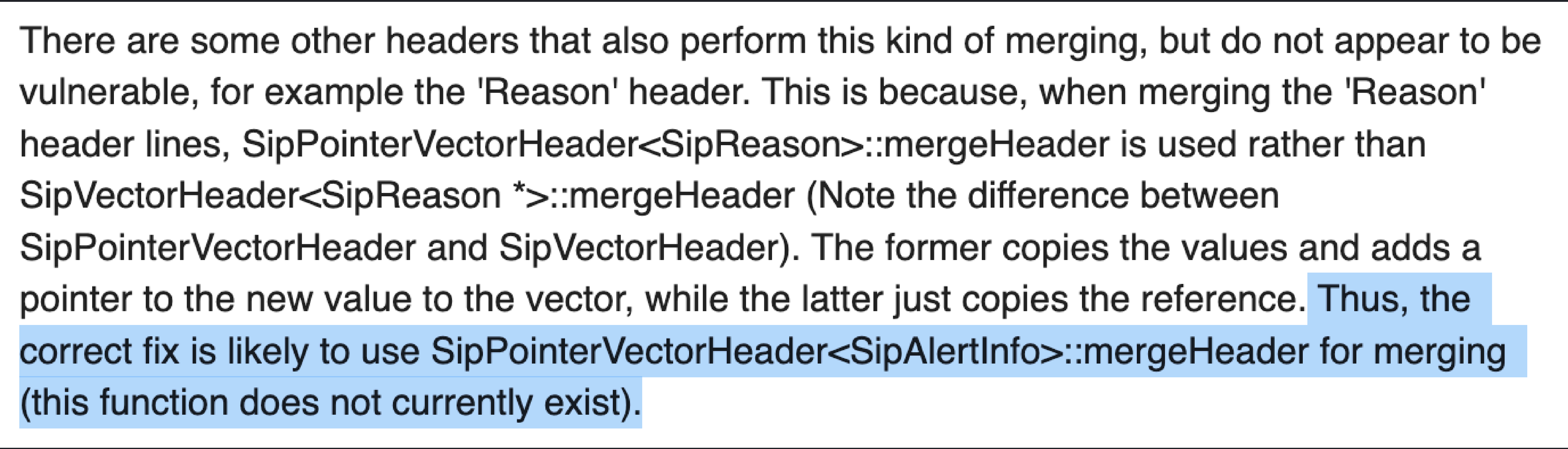
“ immediately after merging the new header `SipMessage::addHeader()` deletes the new header. ”



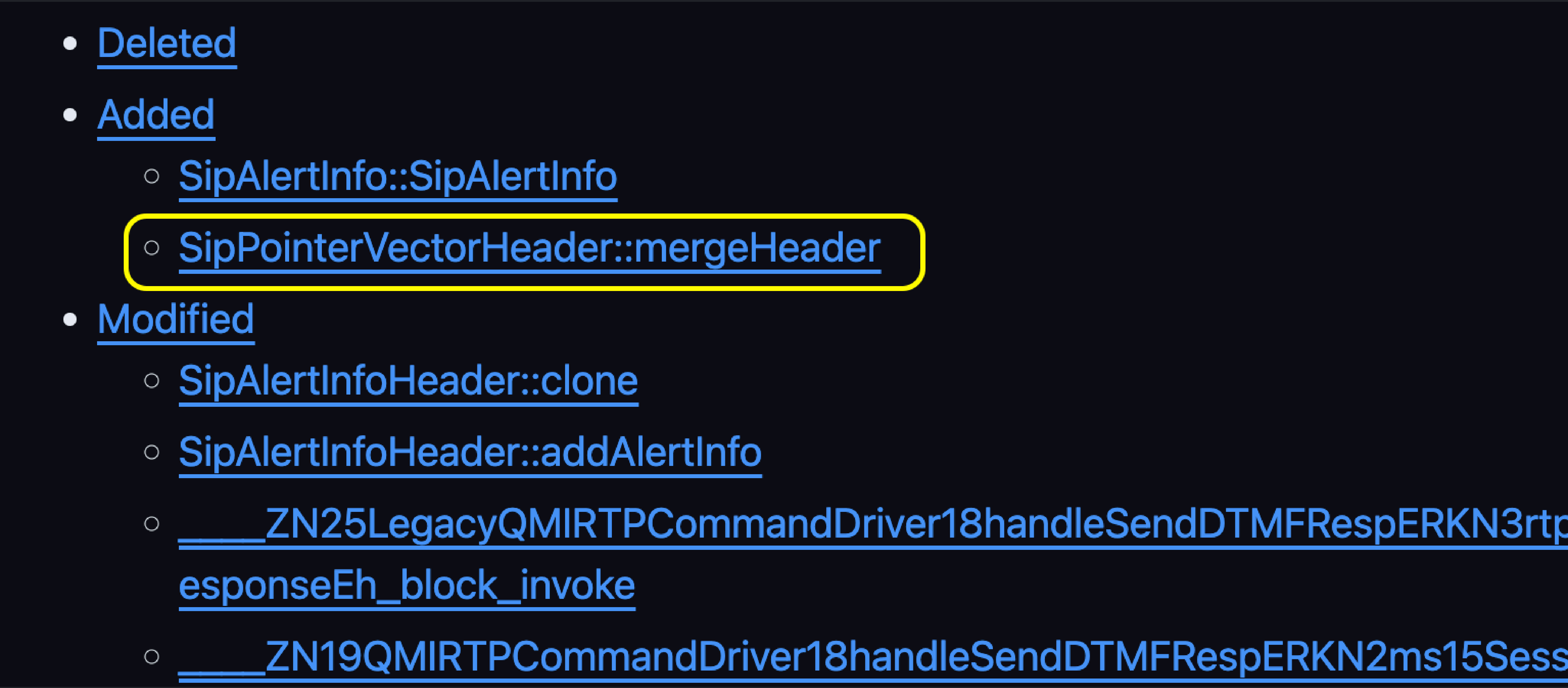
Decompile: addHeader – (libIPTelephony.dylib.arm64.1897-iOS_16.4.1_20E252)

```
      pplVar10 = (long **)(ulong)_Var8;
      pplVar21 = pplVar10;
      if (uVar20 == 0) goto LAB_1bb5ec488;
      pplVar19 = pplVar10 + uVar20;
    }
    *pplVar10 = local_78[0];
    sVar14 = (long)pplVar16 - (long)*(void **)(this + 8);
    pvVar22 = (void *)((long)pplVar10 - sVar14);
    _memmove(pvVar22,*(void **)(this + 8),sVar14);
    lVar17 = *(long *)(this + 0x10);
    _memmove(pplVar10 + 1,pplVar16,lVar17 - (long)pplVar16);
    pvVar11 = *(void **)(this + 8);
    *(void **)(this + 8) = pvVar22;
    *(long *)(this + 0x10) = (long)(pplVar10 + 1) + (lVar17 - (long)pplVar16);
    *(long ***)(this + 0x18) = pplVar19;
  }
  if (pvVar11 != (void *)0x0) {
    operator.delete(pvVar11);
  }
LAB_1bb5ec4fc:
  plVar13 = local_78[0];
  local_68 = (ulong **)(local_78[0] + 1);
  auVar25 = std::__1::__hash_table<>::__emplace_unique_key_args<>
                      ((__hash_table<> *)(this + 0x20),local_68,&std::__1::piecewis
```

👀



There are some other headers that also perform this kind of merging, but do not appear to be vulnerable, for example the 'Reason' header. This is because, when merging the 'Reason' header lines, SipPointerVectorHeader<SipReason>::mergeHeader is used rather than SipVectorHeader<SipReason *>::mergeHeader (Note the difference between SipPointerVectorHeader and SipVectorHeader). The former copies the values and adds a pointer to the new value to the vector, while the latter just copies the reference. Thus, the correct fix is likely to use SipPointerVectorHeader<SipAlertInfo>::mergeHeader for merging (this function does not currently exist).



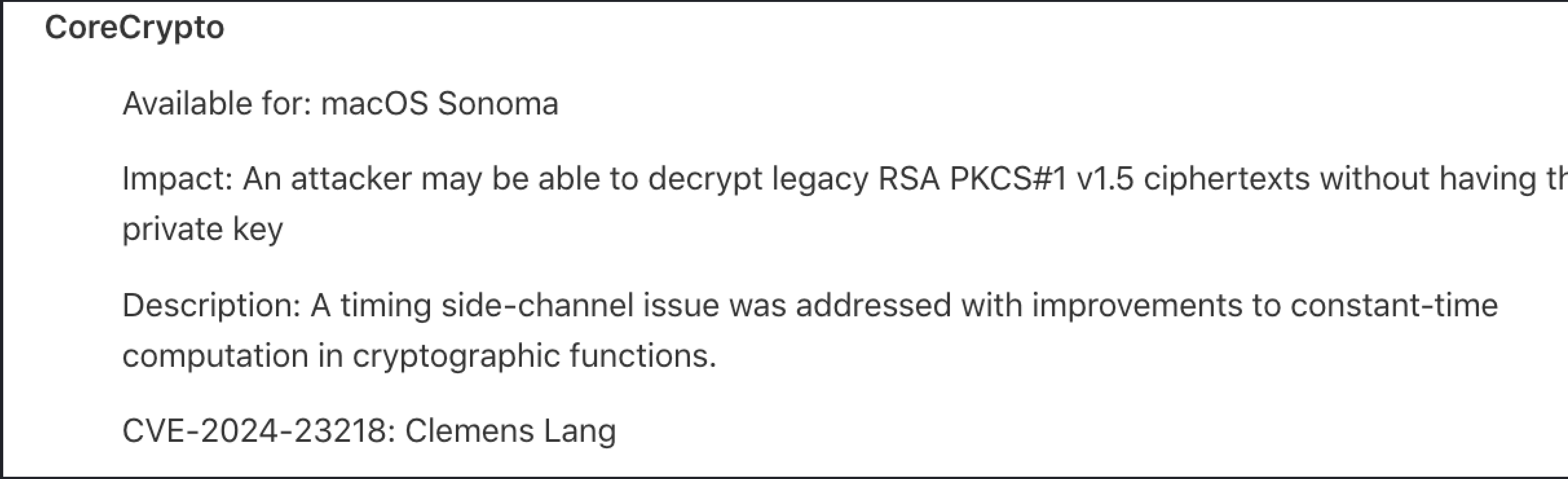
- Deleted
- Added
  - SipAlertInfo::SipAlertInfo
  - SipPointerVectorHeader::mergeHeader
- Modified
  - SipAlertInfoHeader::clone
  - SipAlertInfoHeader::addAlertInfo
  - _____ZN25LegacyQMIRTPCommandDriver18handleSendDTMFRespERKN3rtp
    esponseEh_block_invoke
  - _____ZN19QMIRTPCommandDriver18handleSendDTMFRespERKN2ms15Sess

CVE-2023-32412 - use after free

- Full details of the root cause in p0 post
- We proved that we found the right file
- That Ghidra actually handled the binary well
- That we could detect the differences
- Observed Apple followed Ivan's advice

CVE-2024-23218

(speed mode)



CoreCrypto

Available for: macOS Sonoma

Impact: An attacker may be able to decrypt legacy RSA PKCS#1 v1.5 ciphertexts without having th
private key

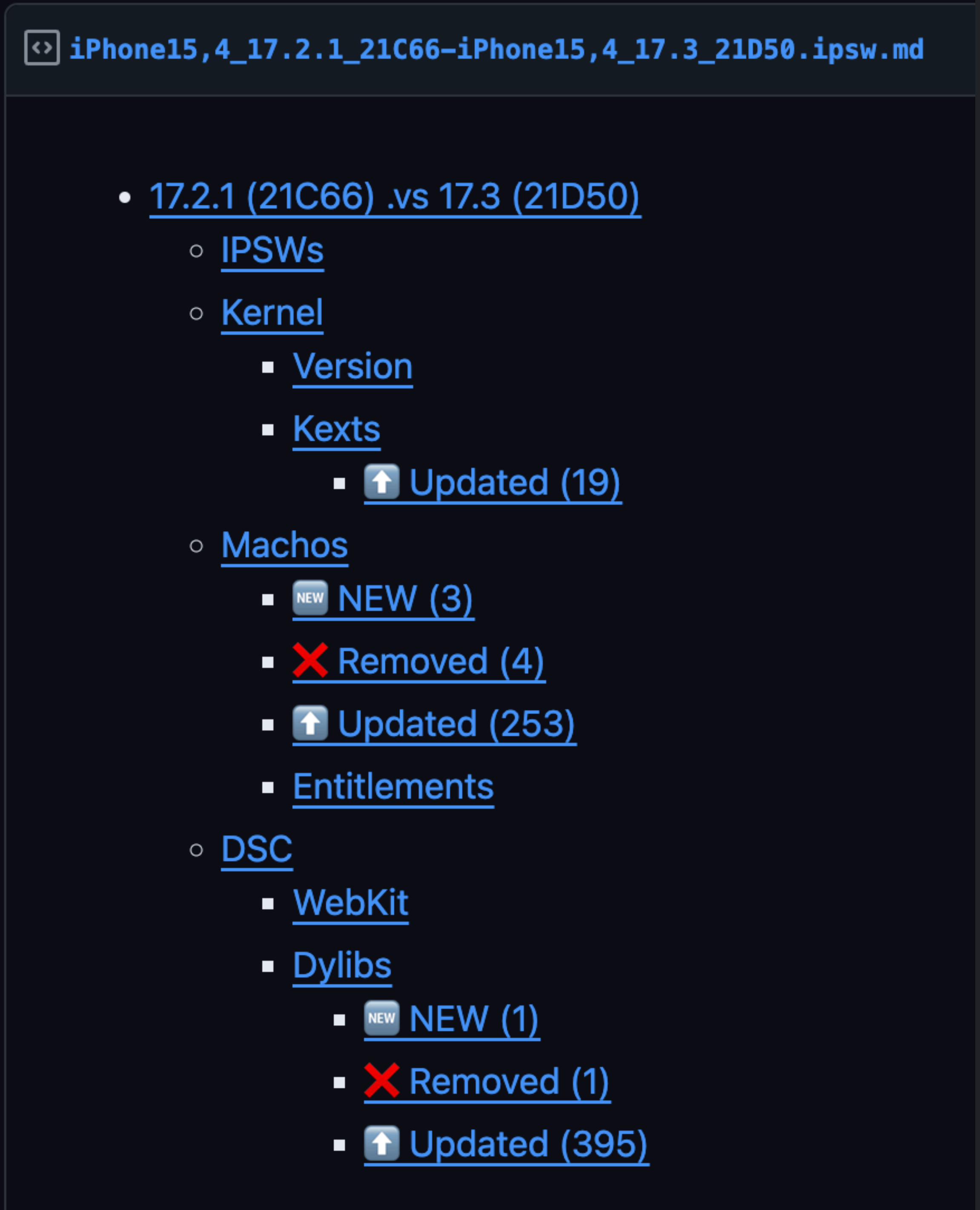
Description: A timing side-channel issue was addressed with improvements to constant-time computation in cryptographic functions.

CVE-2024-23218: Clemens Lang

- `ipsw download ipsw --device iPhone15,4 -V -b 21D50`
- `ipsw download ipsw --device iPhone15,4 -V -b 21C66`

```
ipsw diff iPhone15,4_17.2_21C62_Restore.ipsw
    iPhone15,4_17.2.1_21C66_Restore.ipsw
```

17.2.1 (21C66) .vs 17.3 (21D50)



iPhone15,4_17.2.1_21C66-iPhone15,4_17.3_21D50.ipsw.md

- 17.2.1 (21C66) .vs 17.3 (21D50)
  - IPSWs
  - Kernel
    - Version
    - Kexts
      - ⬆️ Updated (19)
  - Machos
    - 🆕 NEW (3)
    - ❌ Removed (4)
    - ⬆️ Updated (253)
    - Entitlements
  - DSC
    - WebKit
    - Dylibs
      - 🆕 NEW (1)
      - ❌ Removed (1)
      - ⬆️ Updated (395)

```
cat iPhone15,4_17.2.1_21C66-
iPhone15,4_17.3_21D50.ipsw.md | egrep -i
"crypt|updated|kernel|machos|DSC"
```

KEXTs

`com.apple.kec.corecrypto`

> `com.apple.kec.corecrypto`

Macho

> `/System/Library/CryptoTokenKit/usbsmartcardreaderd.slotd/usbsmartcardreaderd`

DSC

`CryptoTokenKit`

> `/System/Library/Frameworks/CryptoTokenKit.framework/CryptoTokenKit`

`CSExattrCrypto`

> `/System/Library/PrivateFrameworks/CSExattrCrypto.framework/CSExattrCrypto`

`libcommonCrypto.dylib`

> `/usr/lib/system/libcommonCrypto.dylib`

`libcorecrypto.dylib`

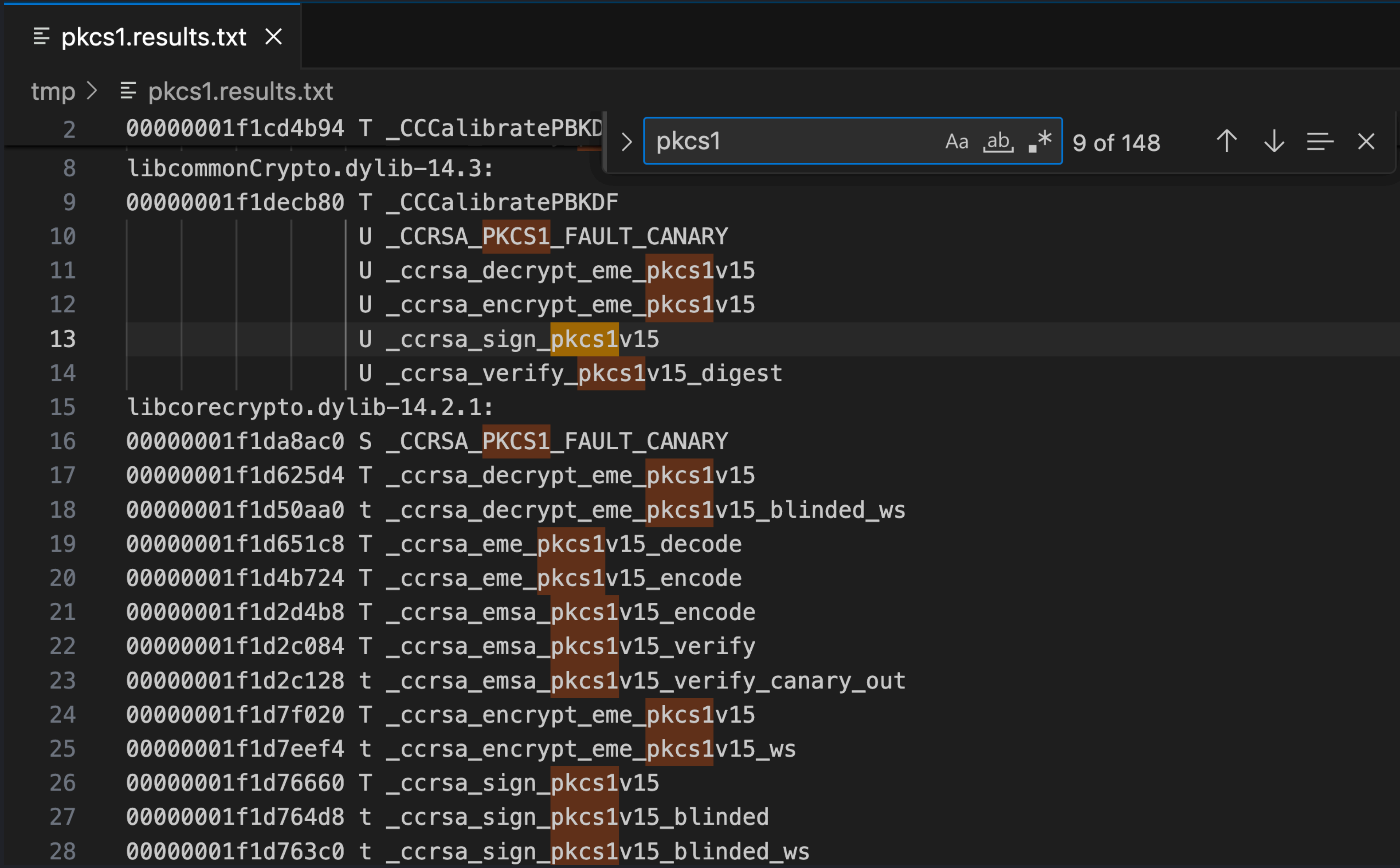
> `/usr/lib/system/libcorecrypto.dylib`

`libcorecrypto_noasm.dylib`

> `/usr/lib/system/libcorecrypto_noasm.dylib`

Which files have related symbols?

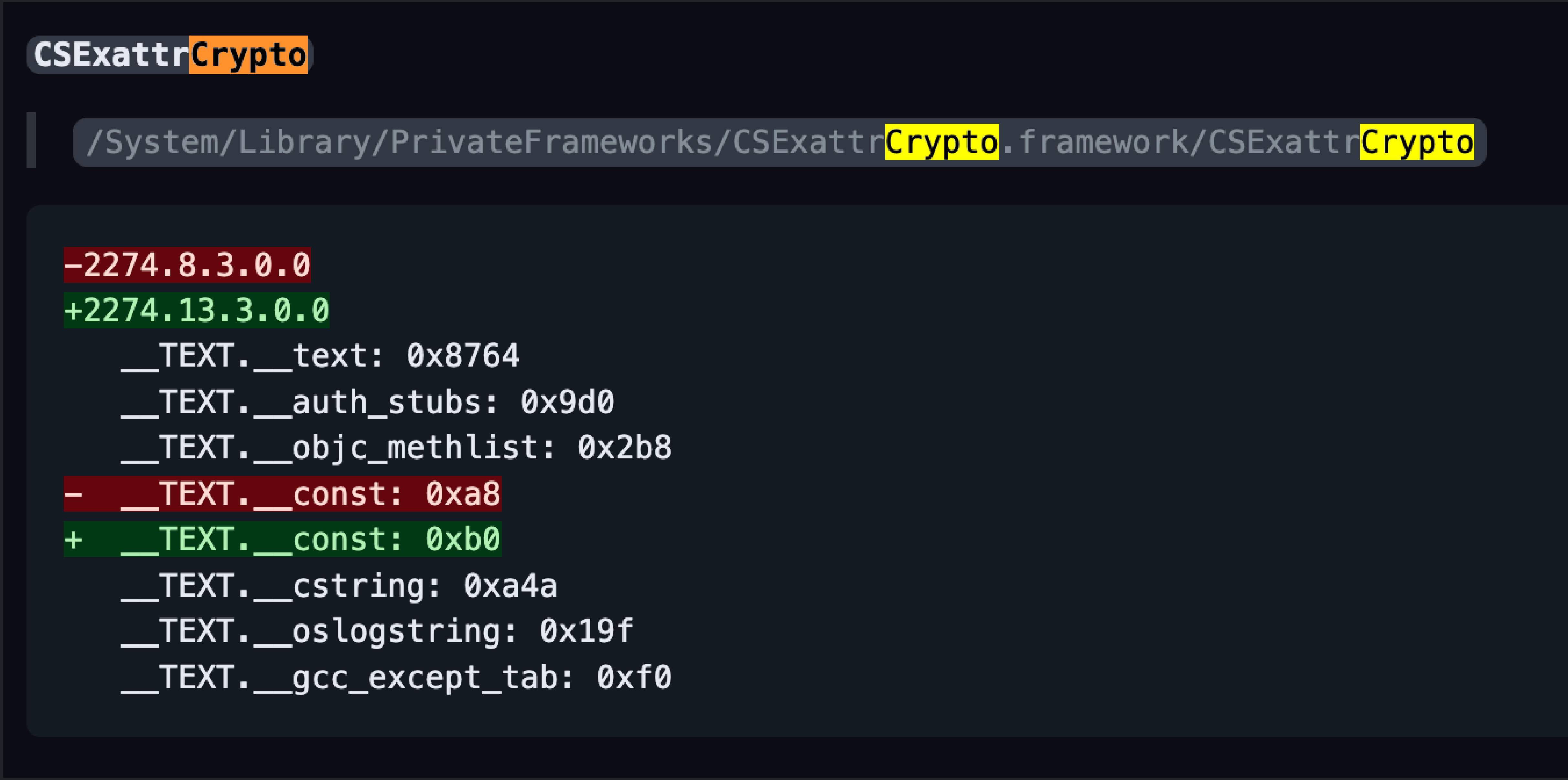
```
nm lib* | egrep -i "pkcs1|lib"
```



```
00000001f1cd4b94 T _CCCalibratePBKD
libcommonCrypto.dylib-14.3:
00000001f1decb80 T _CCCalibratePBKDF
                 U _CCRSA_PKCS1_FAULT_CANARY
                 U _ccrsa_decrypt_eme_pkcs1v15
                 U _ccrsa_encrypt_eme_pkcs1v15
                 U _ccrsa_sign_pkcs1v15
                 U _ccrsa_verify_pkcs1v15_digest
libcorecrypto.dylib-14.2.1:
00000001f1da8ac0 S _CCRSA_PKCS1_FAULT_CANARY
00000001f1d625d4 T _ccrsa_decrypt_eme_pkcs1v15
00000001f1d50aa0 t _ccrsa_decrypt_eme_pkcs1v15_blinded_ws
00000001f1d651c8 T _ccrsa_eme_pkcs1v15_decode
00000001f1d4b724 T _ccrsa_eme_pkcs1v15_encode
00000001f1d2d4b8 T _ccrsa_emsa_pkcs1v15_encode
00000001f1d2c084 T _ccrsa_emsa_pkcs1v15_verify
00000001f1d2c128 t _ccrsa_emsa_pkcs1v15_verify_canary_out
00000001f1d7f020 T _ccrsa_encrypt_eme_pkcs1v15
00000001f1d7eef4 t _ccrsa_encrypt_eme_pkcs1v15_ws
00000001f1d76660 T _ccrsa_sign_pkcs1v15
00000001f1d764d8 t _ccrsa_sign_pkcs1v15_blinded
00000001f1d763c0 t _ccrsa_sign_pkcs1v15_blinded_ws
```

Which files had actual code changes?

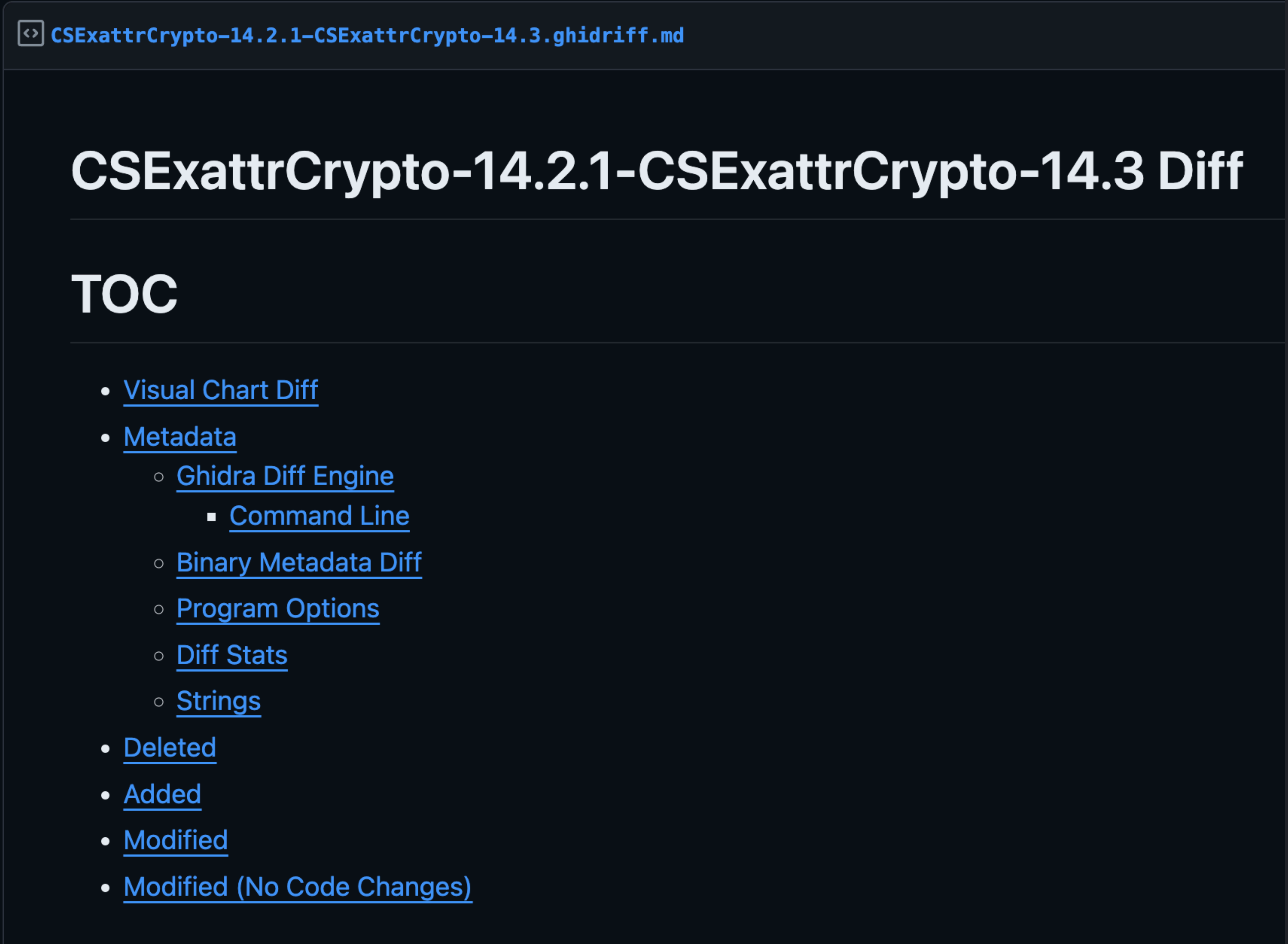
__TEXT.__text changed?



CSExattrCrypto

`/System/Library/PrivateFrameworks/CSExattrCrypto.framework/CSExattrCrypto`

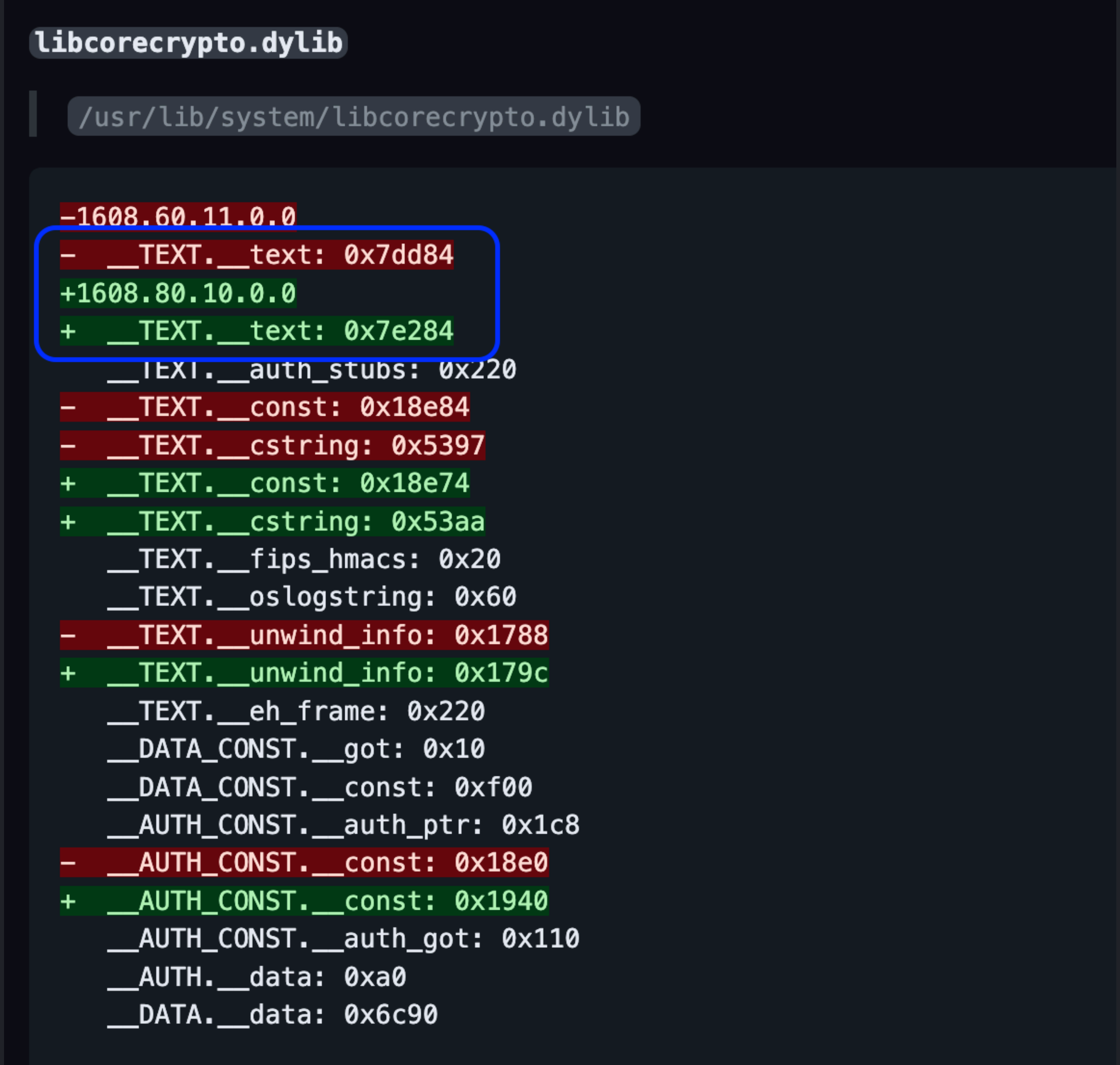
```
-2274.8.3.0.0
+2274.13.3.0.0
   __TEXT.__text: 0x8764
   __TEXT.__auth_stubs: 0x9d0
   __TEXT.__objc_methlist: 0x2b8
-  __TEXT.__const: 0xa8
+  __TEXT.__const: 0xb0
   __TEXT.__cstring: 0xa4a
   __TEXT.__oslogstring: 0x19f
   __TEXT.__gcc_except_tab: 0xf0
```



CSExattrCrypto-14.2.1-CSExattrCrypto-14.3.ghidriff.md

CSExattrCrypto-14.2.1-CSExattrCrypto-14.3 Diff

TOC

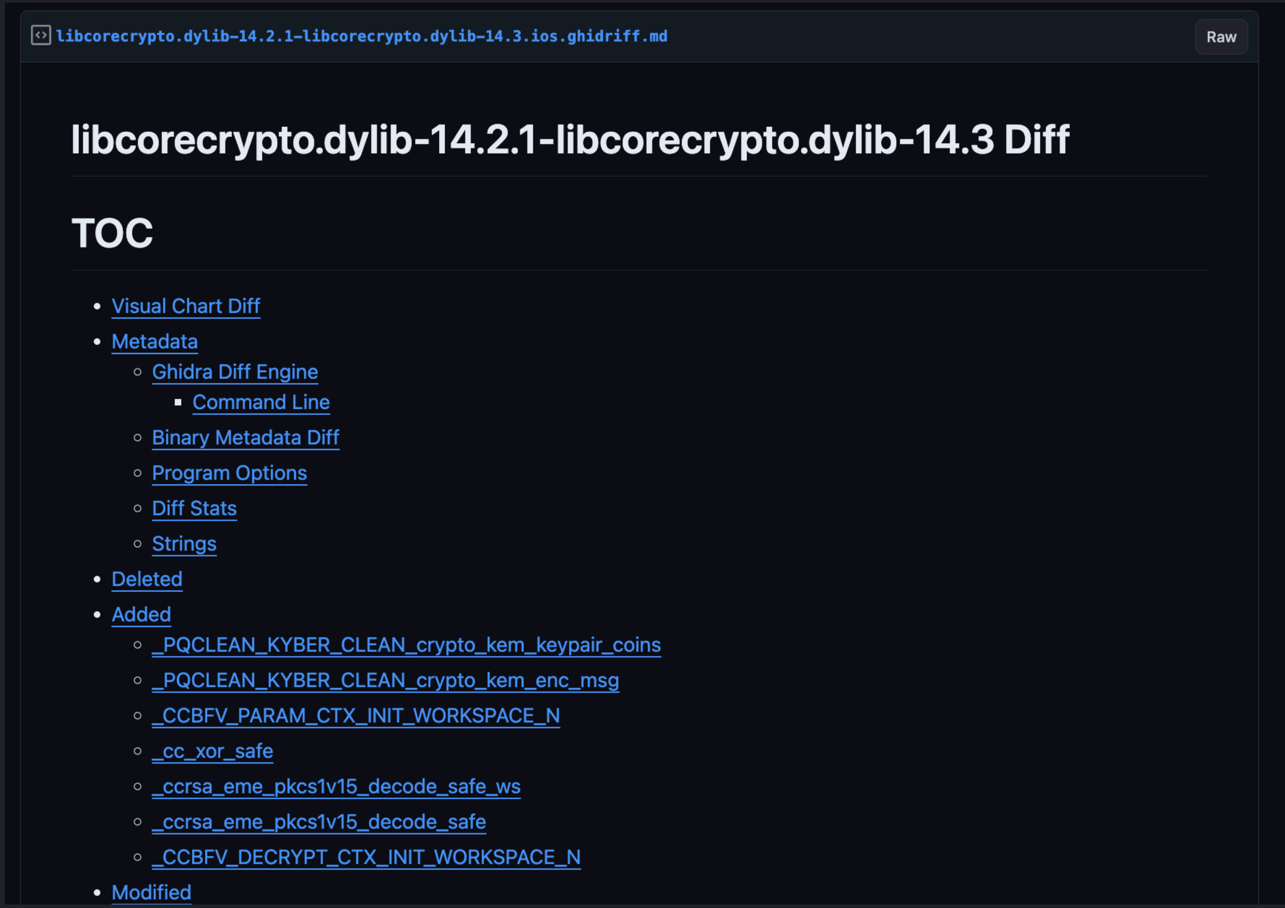


`libcorecrypto.dylib`

> `/usr/lib/system/libcorecrypto.dylib`

```diff
-1608.60.11.0.0
-   __TEXT.__text: 0x7dd84
+1608.80.10.0.0
+   __TEXT.__text: 0x7e284
    __TEXT.__auth_stubs: 0x220
-   __TEXT.__const: 0x18e84
-   __TEXT.__cstring: 0x5397
+   __TEXT.__const: 0x18e74
+   __TEXT.__cstring: 0x53aa
    __TEXT.__fips_hmacs: 0x20
    __TEXT.__oslogstring: 0x60
-   __TEXT.__unwind_info: 0x1788
+   __TEXT.__unwind_info: 0x179c
    __TEXT.__eh_frame: 0x220
    __DATA_CONST.__got: 0x10
    __DATA_CONST.__const: 0xf00
    __AUTH_CONST.__auth_ptr: 0x1c8
-   __AUTH_CONST.__const: 0x18e0
+   __AUTH_CONST.__const: 0x1940
    __AUTH_CONST.__auth_got: 0x110
    __AUTH.__data: 0xa0
    __DATA.__data: 0x6c90
```

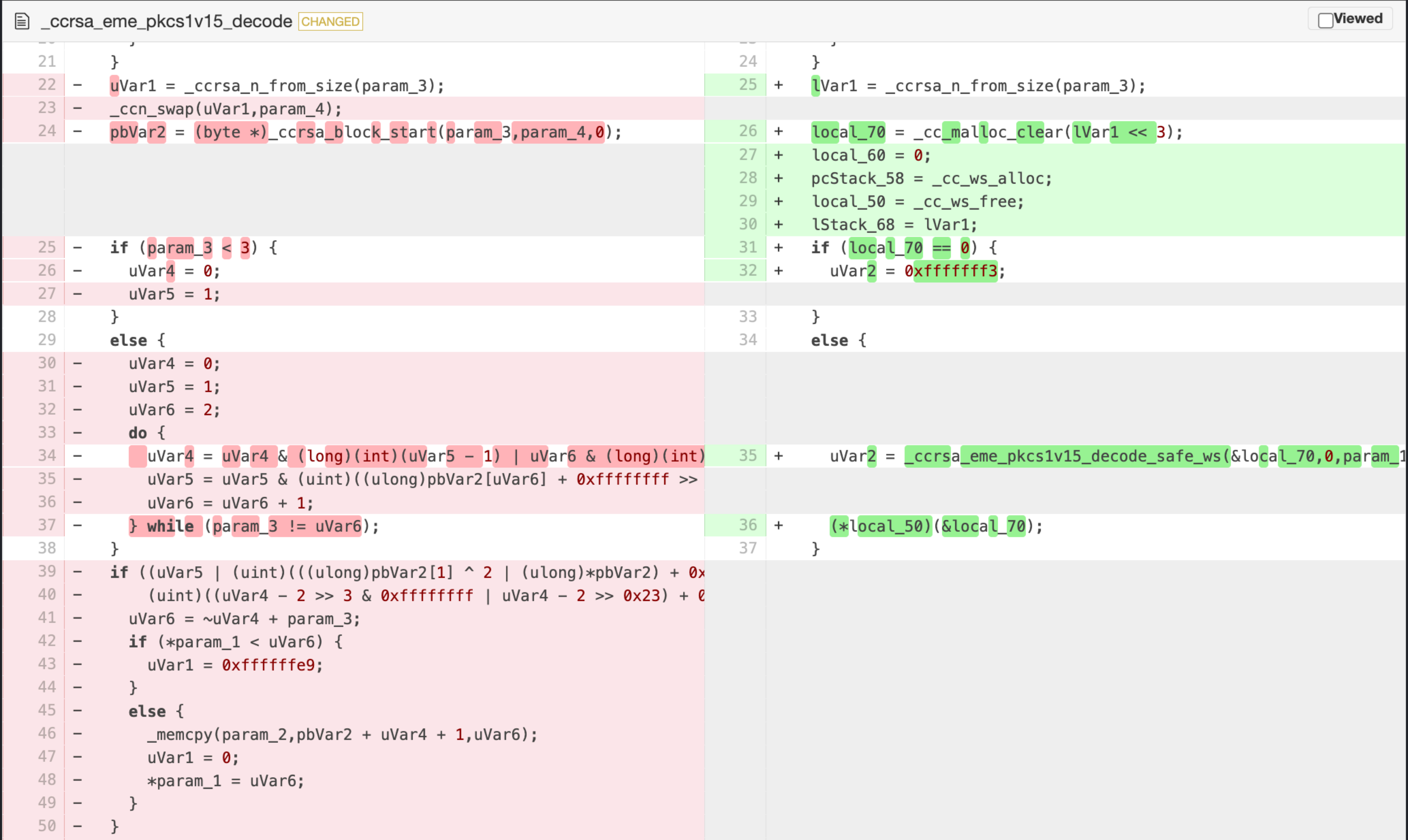
libcorecrypto.dylib-14.2.1-libcorecrypto.dylib-14.3.ios.ghidriff.md


libcorecrypto.dylib-14.2.1-libcorecrypto.dylib-14.3.ios.ghidriff.md

Raw

libcorecrypto.dylib-14.2.1-libcorecrypto.dylib-14.3 Diff

TOC

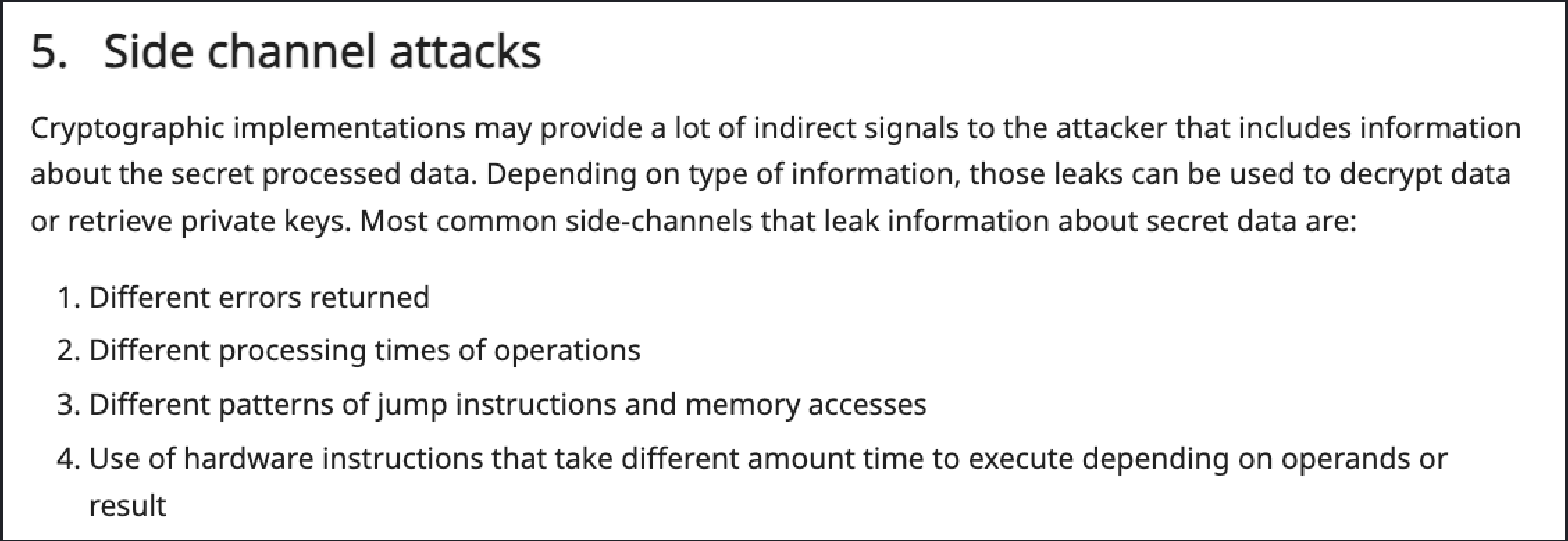


_ccrsa_eme_pkcs1v15_decode CHANGED

Viewed

```
  }
- uVar1 = _ccrsa_n_from_size(param_3);
- _ccn_swap(uVar1,param_4);
- pbVar2 = (byte *)_ccrsa_block_start(param_3,param_4,0);
- if (param_3 < 3) {
-   uVar4 = 0;
-   uVar5 = 1;
  }
  else {
-   uVar4 = 0;
-   uVar5 = 1;
-   uVar6 = 2;
-   do {
-     uVar4 = uVar4 & (long)(int)(uVar5 - 1) | uVar6 & (long)(int)
-     uVar5 = uVar5 & (uint)((ulong)pbVar2[uVar6] + 0xffffffff >>
-     uVar6 = uVar6 + 1;
-   } while (param_3 != uVar6);
  }
- if ((uVar5 | (uint)(((ulong)pbVar2[1] ^ 2 | (ulong)*pbVar2) + 0x
-     (uint)((uVar4 - 2 >> 3 & 0xffffffff | uVar4 - 2 >> 0x23) + 0
-   uVar6 = ~uVar4 + param_3;
-   if (*param_1 < uVar6) {
-     uVar1 = 0xffffffe9;
-   }
-   else {
-     _memcpy(param_2,pbVar2 + uVar4 + 1,uVar6);
-     uVar1 = 0;
-     *param_1 = uVar6;
-   }
- }
```

```
  }
+ lVar1 = _ccrsa_n_from_size(param_3);
+ local_70 = _cc_malloc_clear(lVar1 << 3);
+ local_60 = 0;
+ pcStack_58 = _cc_ws_alloc;
+ local_50 = _cc_ws_free;
+ lStack_68 = lVar1;
+ if (local_70 == 0) {
+   uVar2 = 0xfffffff3;
  }
  else {
+   uVar2 = _ccrsa_eme_pkcs1v15_decode_safe_ws(&local_70,0,param_1
+   (*local_50)(&local_70);
  }
```



5. Side channel attacks

Cryptographic implementations may provide a lot of indirect signals to the attacker that includes information about the secret processed data. Depending on type of information, those leaks can be used to decrypt data or retrieve private keys. Most common side-channels that leak information about secret data are:

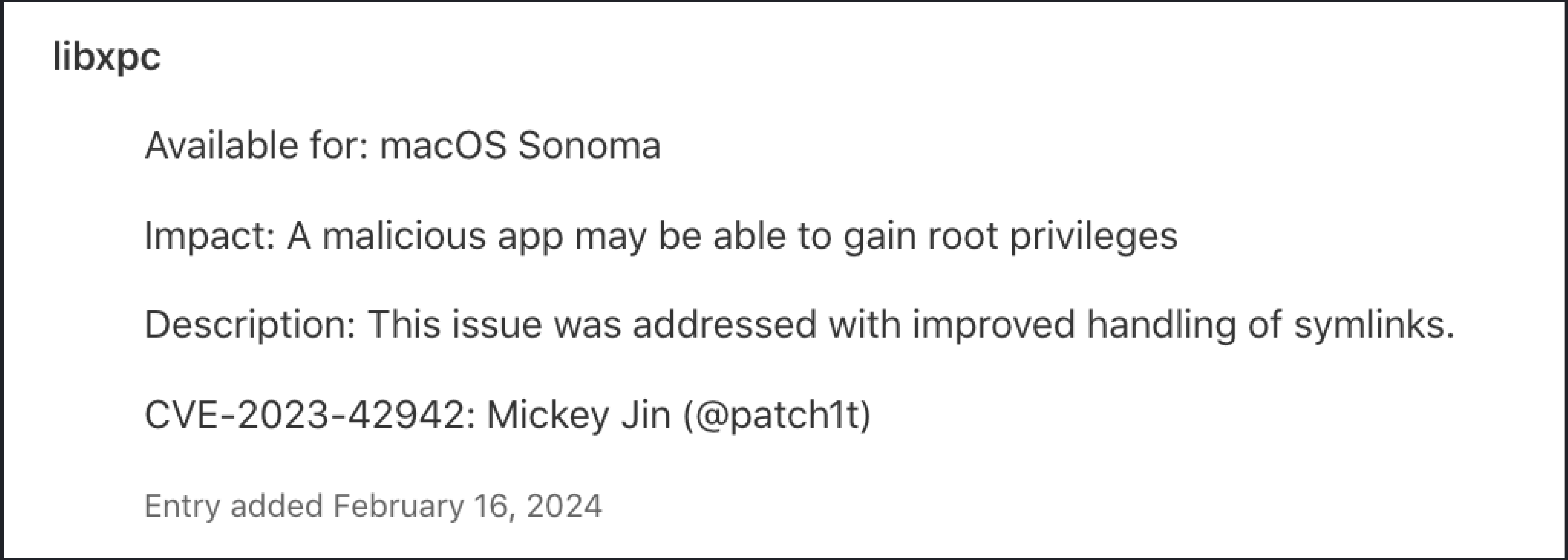
1. Different errors returned
2. Different processing times of operations
3. Different patterns of jump instructions and memory accesses
4. Use of hardware instructions that take different amount time to execute depending on operands or result

https://www.ietf.org/archive/id/draft-kario-rsa-guidance-01.html

CVE-2024-23218 - timing side-channel

- Several crypto files were updated
  - Some had no code changes!
- Ghidra can handle *OS symbols
- The `decode` function now uses a new "safe" API
- Several "PKCS" related function updates
- More effort needed to understand the root cause

CVE-2023-42942


libxpc

Available for: macOS Sonoma

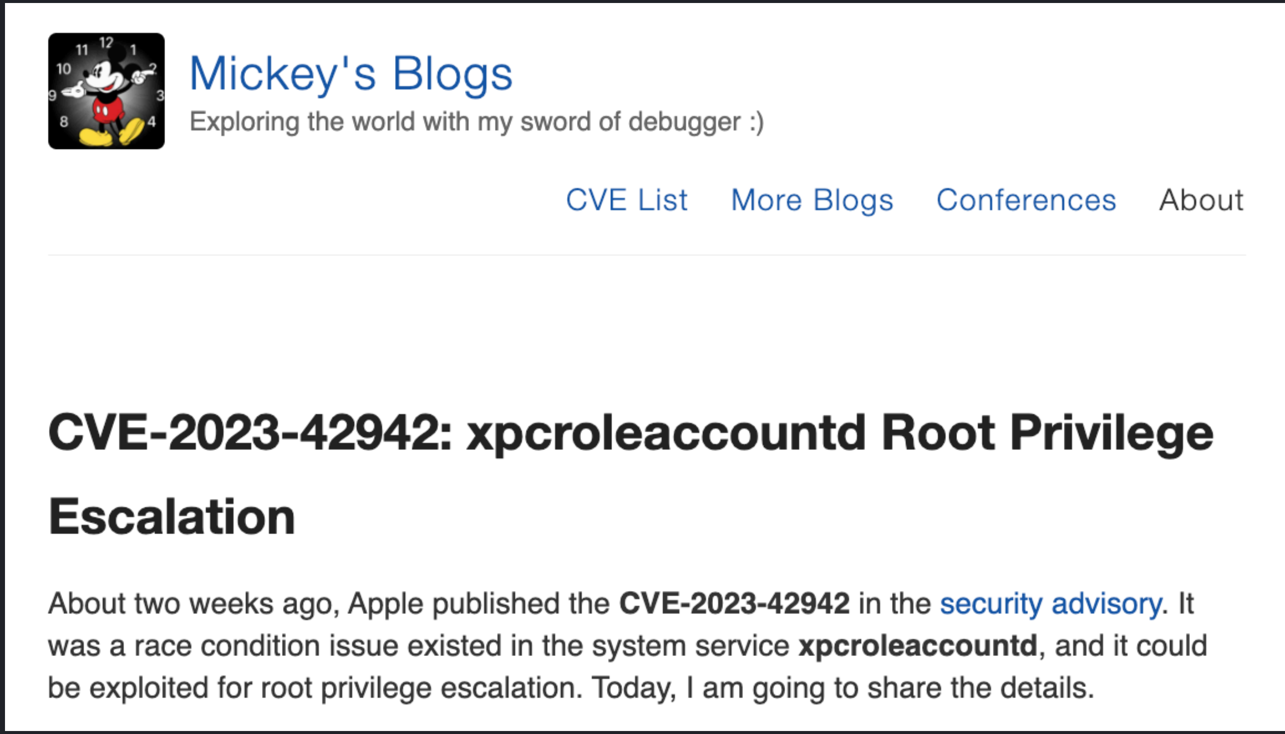
Impact: A malicious app may be able to gain root privileges

Description: This issue was addressed with improved handling of symlinks.

CVE-2023-42942: Mickey Jin (@patch1t)

Entry added February 16, 2024

https://support.apple.com/en-us/HT213984 jhftss.github.io - CVE-202-42942



CVE-2023-42942: xpcroleaccountd Root Privilege Escalation

About two weeks ago, Apple published the **CVE-2023-42942** in the security advisory. It was a race condition issue existed in the system service **xpcroleaccountd**, and it could be exploited for root privilege escalation. Today, I am going to share the details.

A Perfect Example 👌

- detailed blog post
- exploit POC
- comments on the patch

TOCTOU LPE

1. Copy legitimate xpc service to /tmp
2. Create xpc connection and call `add_bundle` with legit service
3. Pass symlink mapped to legitimate service
4. Once validation is passed, switch symlink to malicious package
5. Gain arbitrary remote execution as root

Apple’s patch

The issue has been patched in **macOS 14.1**:

```
void handle_xpc_message(void *, void *msg) {
...
        v70 = copyfile_state_alloc();
    copyfile_state_set(v70, 6u, &copyfile_callback); // COPYFILE_ST
    v71 = copyfile(from, to, v70, 0xC800Fu);
    copyfile_state_free(v70);
    if ( v71 ) {
       // "pid[%d]: copyfile(3) failed on service: %d, (errno %{err
       goto EXIT;
    }
...
}
```

In the `copyfile_callback` function, it will check the **destination path**:

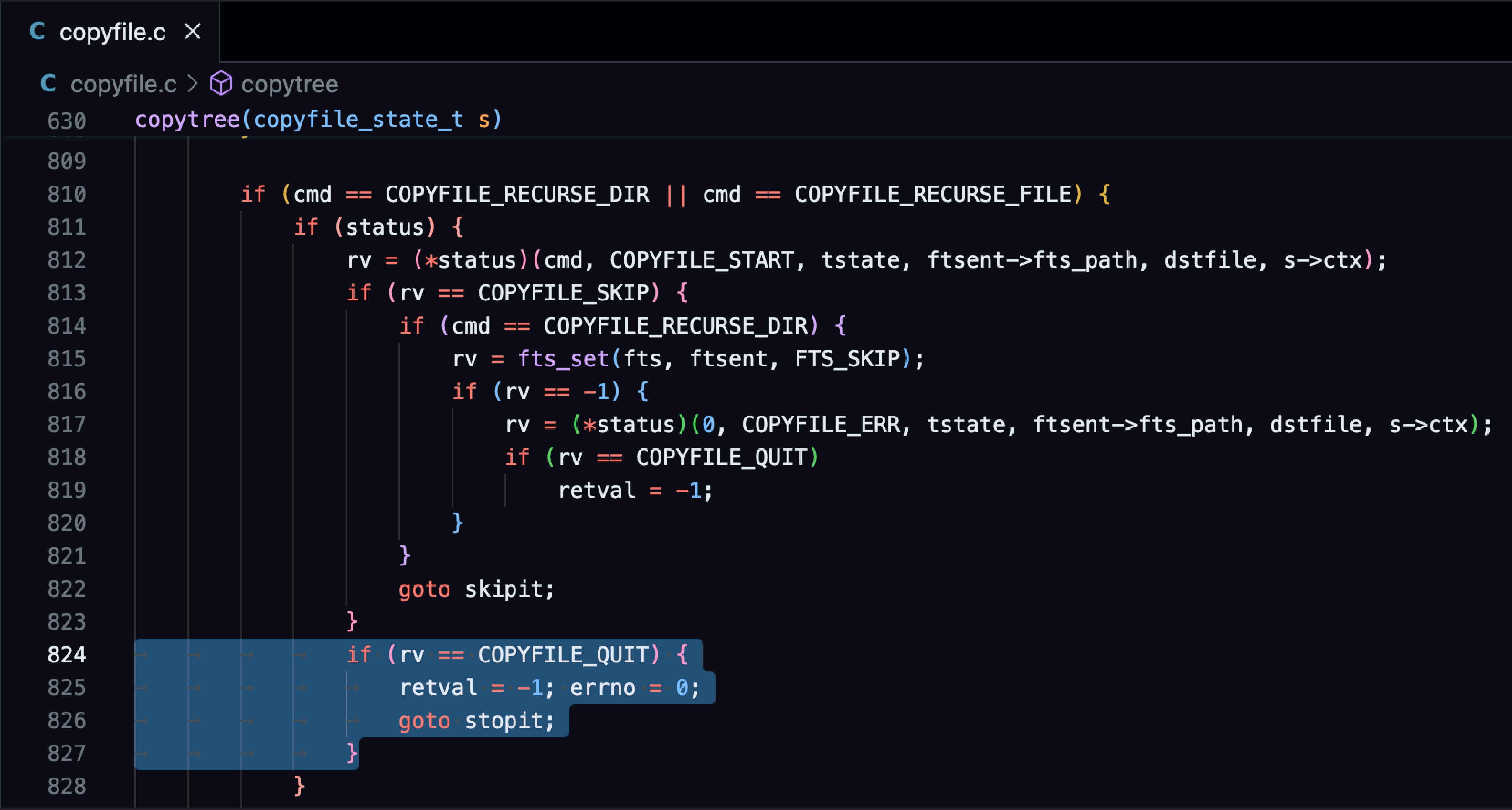
```
int copyfile_callback(int what, int stage, copyfile_state_t state,
{
  memset(&v11, 170, sizeof(v11));
  result = 0LL;
  if ( stage == COPYFILE_FINISH )
  {
```

Apple's Patch in 14.1

> If the destination path is a symbolic link, copyfile_callback will return 2 (COPYFILE_QUIT), the entire copy will be aborted at this stage. Next, copyfile will return -1, but errno will be unmodified. Finally, xpcroleaccountd exits and replies with an error.

`copyfile` handling a directory copy

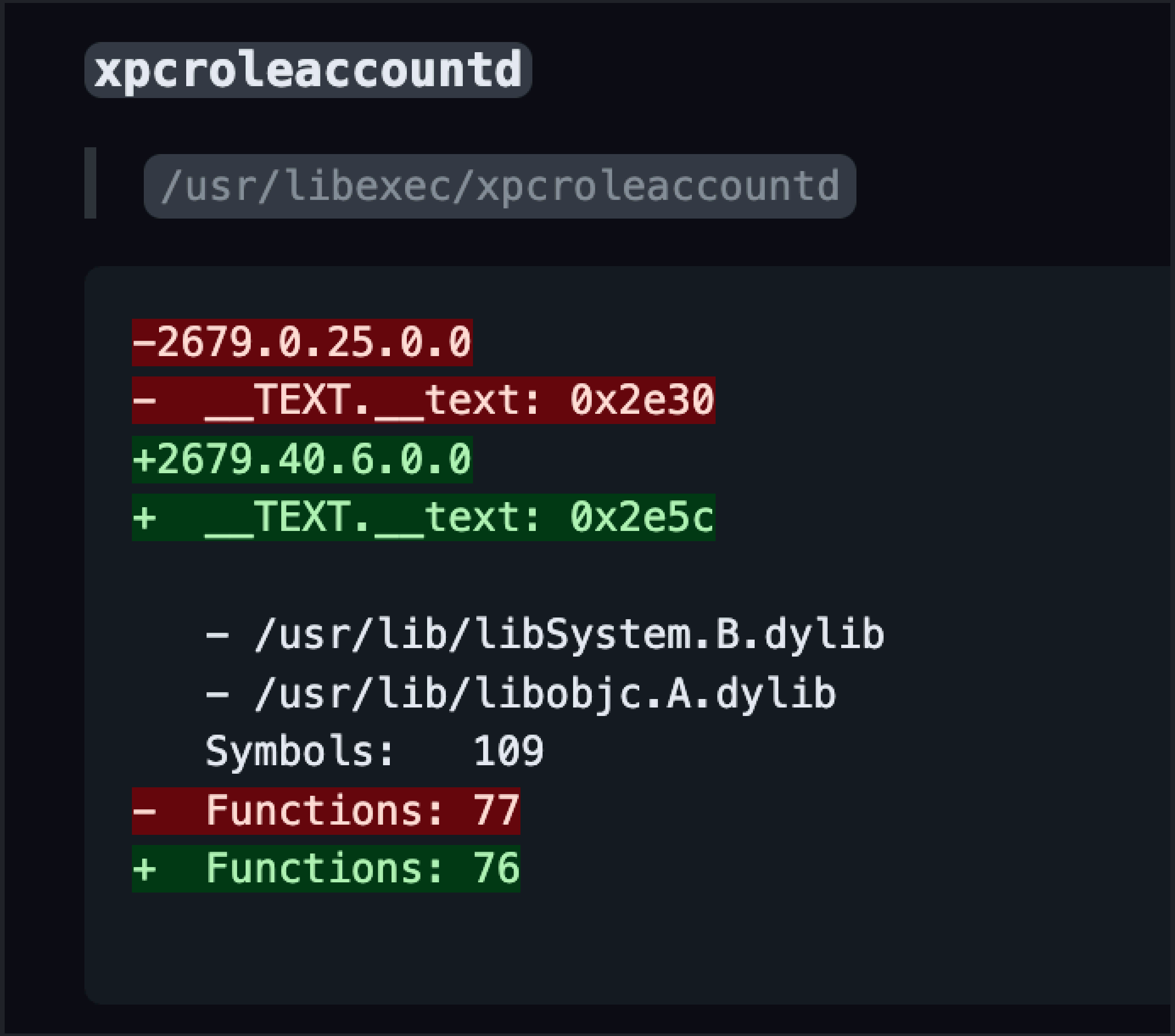
https://github.com/apple-oss-distributions/copyfile/blob/main/copyfile.c#L824-L827



```c
copytree(copyfile_state_t s)

        if (cmd == COPYFILE_RECURSE_DIR || cmd == COPYFILE_RECURSE_FILE) {
            if (status) {
                rv = (*status)(cmd, COPYFILE_START, tstate, ftsent->fts_path, dstfile, s->ctx);
                if (rv == COPYFILE_SKIP) {
                    if (cmd == COPYFILE_RECURSE_DIR) {
                        rv = fts_set(fts, ftsent, FTS_SKIP);
                        if (rv == -1) {
                            rv = (*status)(0, COPYFILE_ERR, tstate, ftsent->fts_path, dstfile, s->ctx);
                            if (rv == COPYFILE_QUIT)
                                retval = -1;
                        }
                    }
                    goto skipit;
                }
                if (rv == COPYFILE_QUIT) {
                    retval = -1; errno = 0;
                    goto stopit;
                }
            }
```

Assumed patch

The copyfile callback function added a check for symbolic link and fails the LPE

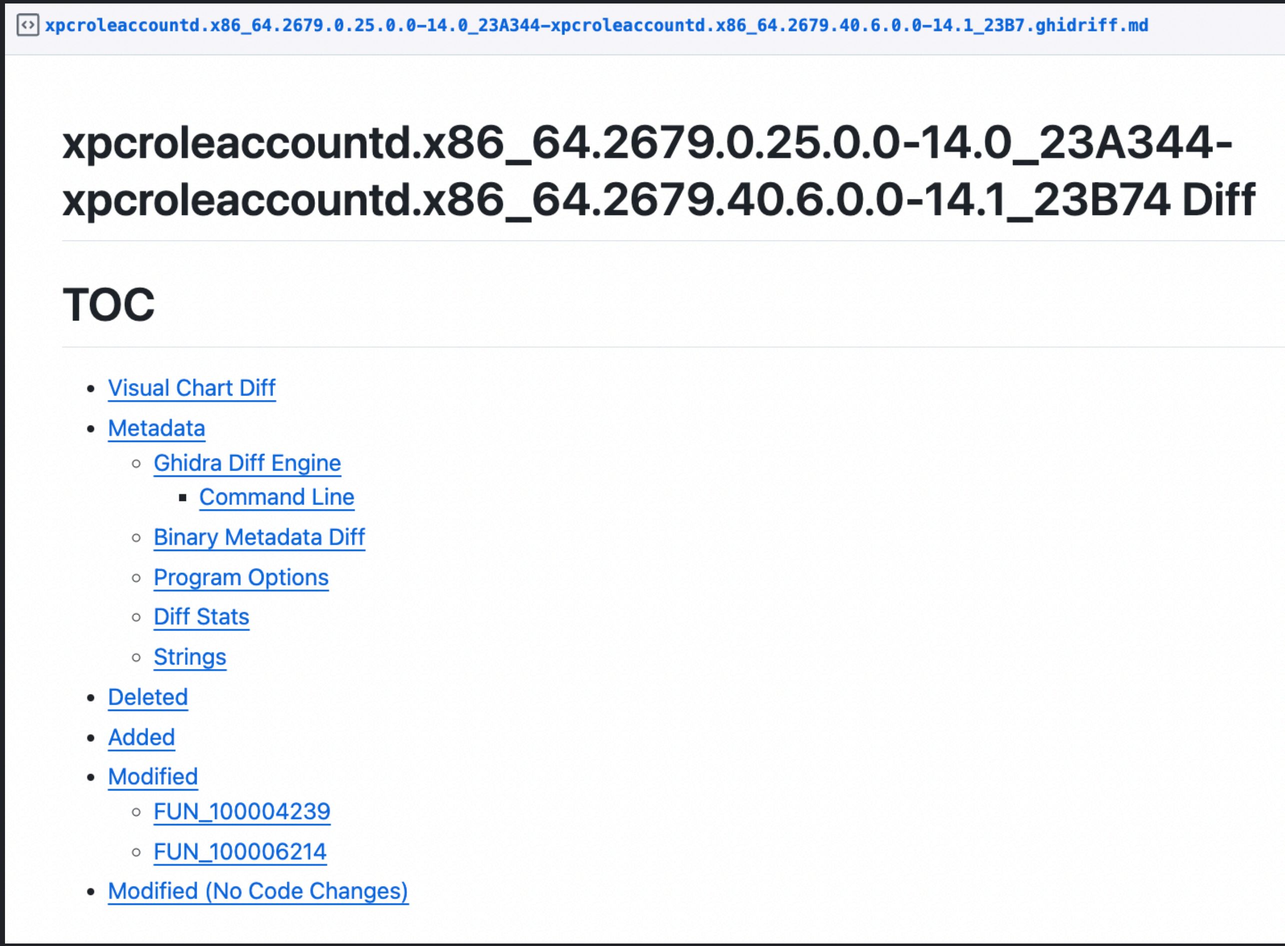


`xpcroleaccountd`

> `/usr/libexec/xpcroleaccountd`

```diff
-2679.0.25.0.0
-   __TEXT.__text: 0x2e30
+2679.40.6.0.0
+   __TEXT.__text: 0x2e5c

    - /usr/lib/libSystem.B.dylib
    - /usr/lib/libobjc.A.dylib
    Symbols:   109
-   Functions: 77
+   Functions: 76
```

ipsw diff 14.0 (23A344) .vs 14.1 (23B74) xpcroleaccountd.x86_64.2679.14.0_23A344-xpcroleaccountd.x86_64.2679.40.6.0.0-14.1_23B7.ghidriff.md

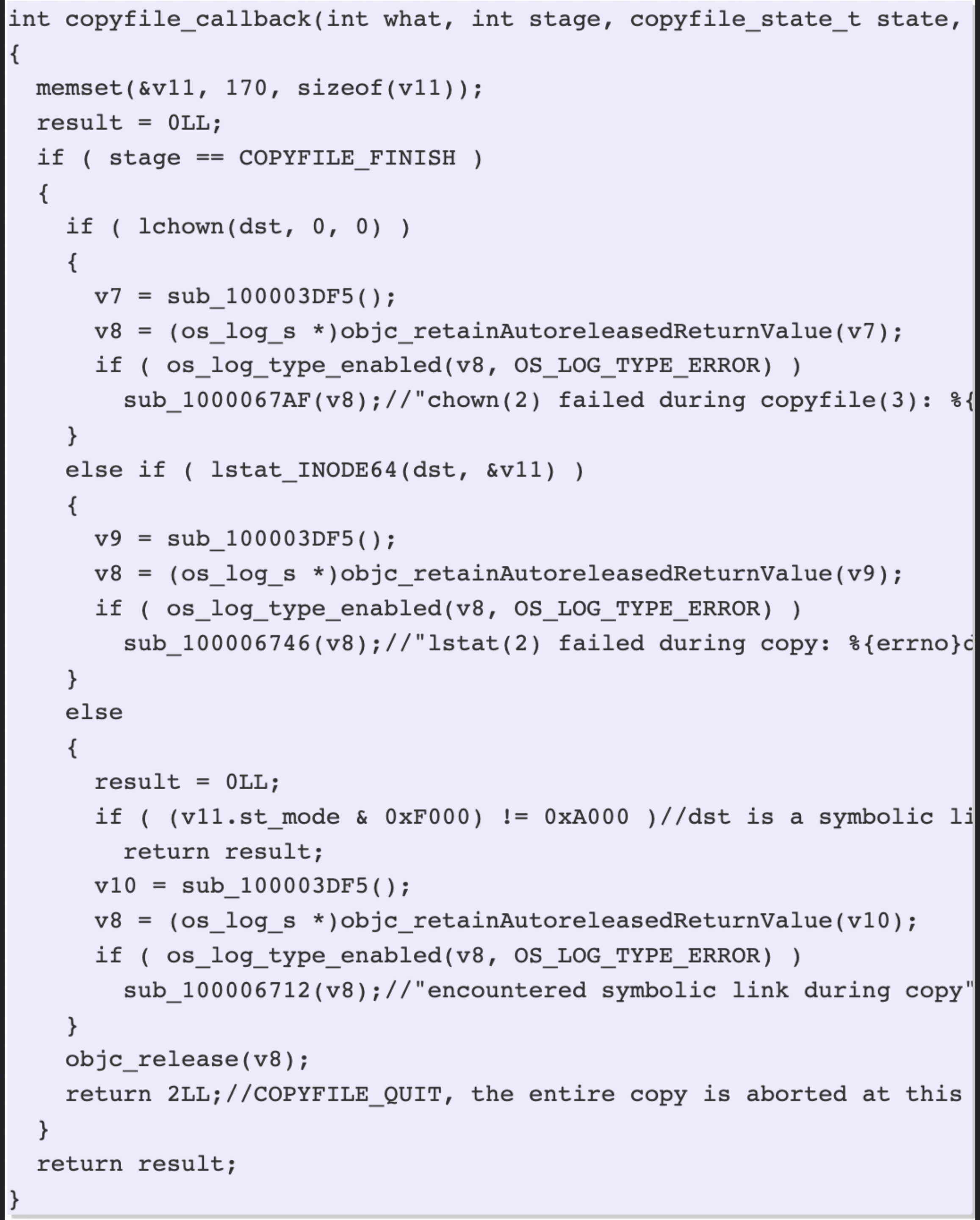


xpcroleaccountd.x86_64.2679.0.25.0.0-14.0_23A344-xpcroleaccountd.x86_64.2679.40.6.0.0-14.1_23B7.ghidriff.md

xpcroleaccountd.x86_64.2679.0.25.0.0-14.0_23A344-xpcroleaccountd.x86_64.2679.40.6.0.0-14.1_23B74 Diff

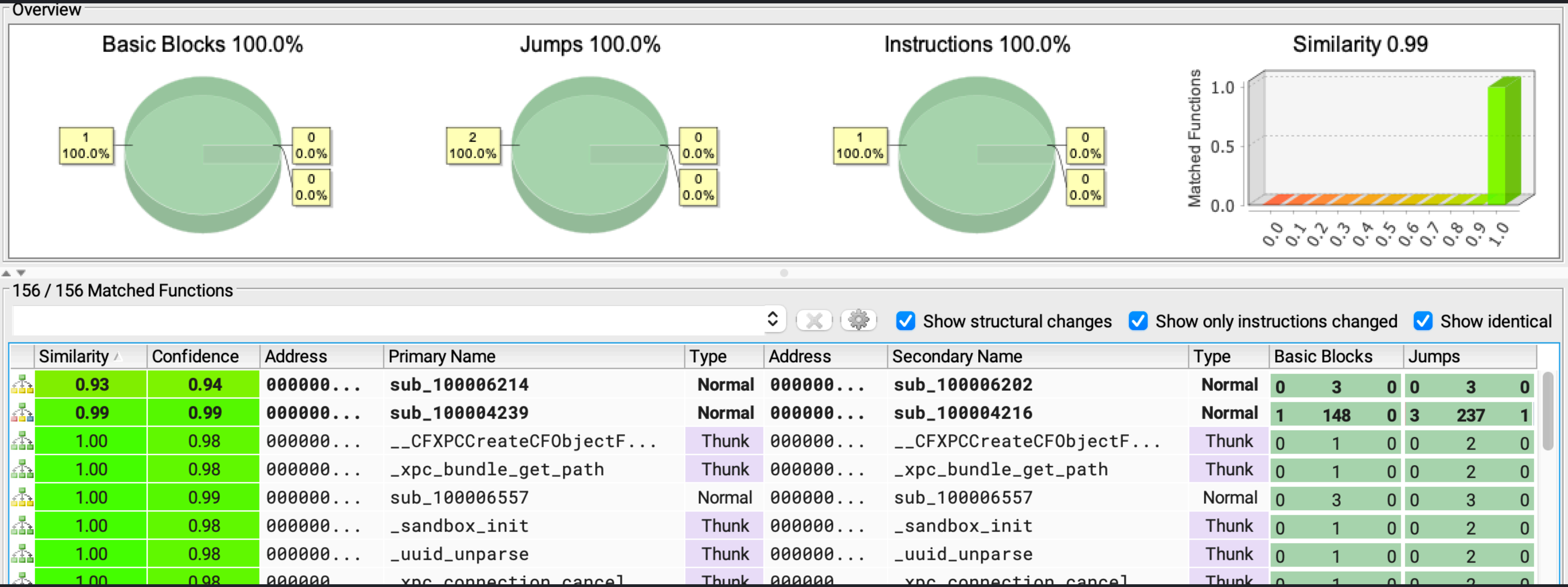
TOC

`ghidriff` didn't find assumed fix... `copyfile_callback` was unchanged



```
int copyfile_callback(int what, int stage, copyfile_state_t state,
{
  memset(&v11, 170, sizeof(v11));
  result = 0LL;
  if ( stage == COPYFILE_FINISH )
  {
    if ( lchown(dst, 0, 0) )
    {
      v7 = sub_100003DF5();
      v8 = (os_log_s *)objc_retainAutoreleasedReturnValue(v7);
      if ( os_log_type_enabled(v8, OS_LOG_TYPE_ERROR) )
        sub_1000067AF(v8);//"chown(2) failed during copyfile(3): %{
    }
    else if ( lstat_INODE64(dst, &v11) )
    {
      v9 = sub_100003DF5();
      v8 = (os_log_s *)objc_retainAutoreleasedReturnValue(v9);
      if ( os_log_type_enabled(v8, OS_LOG_TYPE_ERROR) )
        sub_100006746(v8);//"lstat(2) failed during copy: %{errno}d
    }
    else
    {
      result = 0LL;
      if ( (v11.st_mode & 0xF000) != 0xA000 )//dst is a symbolic li
        return result;
      v10 = sub_100003DF5();
      v8 = (os_log_s *)objc_retainAutoreleasedReturnValue(v10);
      if ( os_log_type_enabled(v8, OS_LOG_TYPE_ERROR) )
        sub_100006712(v8);//"encountered symbolic link during copy"
    }
    objc_release(v8);
    return 2LL;//COPYFILE_QUIT, the entire copy is aborted at this
  }
  return result;
}
```

Maybe ghidriff is having issues? Bindiff?



156 / 156 Matched Functions

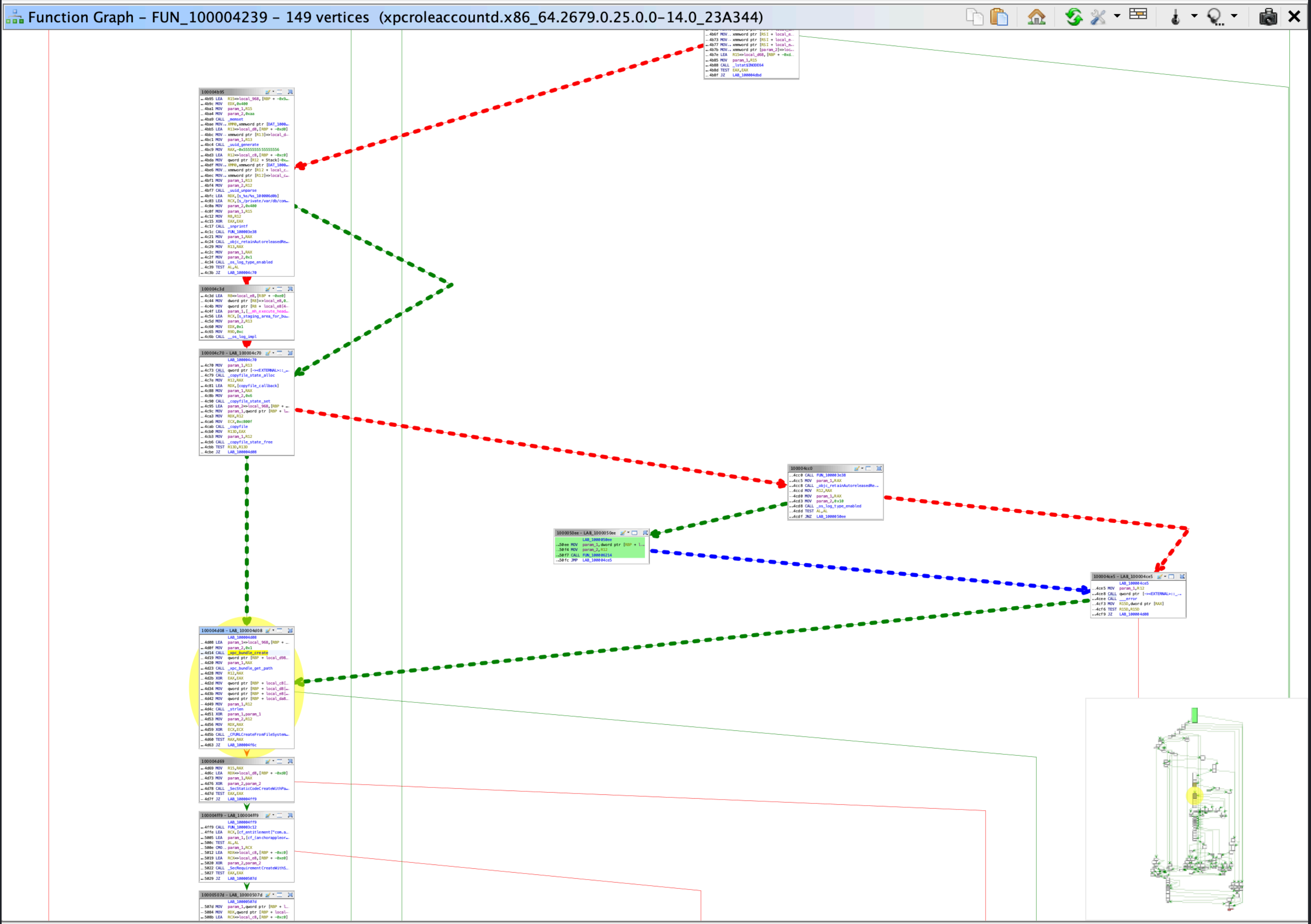
[✓] Show structural changes [✓] Show only instructions changed [✓] Show identical

Similarity	Confidence	Address	Primary Name	Type	Address	Secondary Name	Type	Basic Blocks			Jumps		
0.93	**0.94**	**000000...**	**sub_100006214**	**Normal**	**000000...**	**sub_100006202**	**Normal**	**0**	**3**	**0**	**0**	**3**	**0**
0.99	**0.99**	**000000...**	**sub_100004239**	**Normal**	**000000...**	**sub_100004216**	**Normal**	**1**	**148**	**0**	**3**	**237**	**1**
1.00	0.98	000000...	__CFXPCCreateCFObjectF...	Thunk	000000...	__CFXPCCreateCFObjectF...	Thunk	0	1	0	0	2	0
1.00	0.98	000000...	_xpc_bundle_get_path	Thunk	000000...	_xpc_bundle_get_path	Thunk	0	1	0	0	2	0
1.00	0.99	000000...	sub_100006557	Normal	000000...	sub_100006557	Normal	0	3	0	0	3	0
1.00	0.98	000000...	_sandbox_init	Thunk	000000...	_sandbox_init	Thunk	0	1	0	0	2	0
1.00	0.98	000000...	_uuid_unparse	Thunk	000000...	_uuid_unparse	Thunk	0	1	0	0	2	0
1.00	0.98	000000...	_xpc_connection_cancel	Thunk	000000...	_xpc_connection_cancel	Thunk	0	1	0	0	2	0

`ghidriff` correctly identified the changed functions. There was no additional check for SYMLINK files. It existed in both 14.0 and 14.1

From the post I assumed that added an addtional check for symlinks in the `copyfile` callback function

The truth is a bit more subtle



Function Graph – FUN_100004239 – 149 vertices (xpcroleaccountd.x86_64.2679.0.25.0.0–14.0_23A344)

```
...4b6f MOV... xmmword ptr [RSI + local_e...
...4b73 MOV... xmmword ptr [RSI + local_e...
...4b77 MOV... xmmword ptr [RSI + local_e...
...4b7b MOV... xmmword ptr [param_2]=>loc...
...4b7e LEA    R15=>local_d68,[RBP + -0xd...
...4b85 MOV    param_1,R15
...4b88 CALL   _lstat$INODE64
...4b8d TEST   EAX,EAX
...4b8f JZ     LAB_100004dbd
```

```
100004b95
...4b95 LEA    R15=>local_968,[RBP + -0x9...
...4b9c MOV    EDX,0x400
...4ba1 MOV    param_1,R15
...4ba4 MOV    param_2,0xaa
...4ba9 CALL   _memset
...4bae MOV... XMM0,xmmword ptr [DAT_1000...
...4bb5 LEA    R13=>local_d8,[RBP + -0xd0]
...4bbc MOV... xmmword ptr [R13]=>local_d...
...4bc1 MOV    param_1,R13
...4bc4 CALL   _uuid_generate
...4bc9 MOV    RAX,-0x5555555555555556
...4bd3 LEA    R12=>local_c8,[RBP + -0xc0]
...4bda MOV    qword ptr [R12 + Stack[-0x...
...4bdf MOV... XMM0,xmmword ptr [DAT_1000...
...4be6 MOV... xmmword ptr [R12 + local_c...
...4bec MOV... xmmword ptr [R12]=>local_c...
...4bf1 MOV    param_1,R13
...4bf4 MOV    param_2,R12
...4bf7 CALL   _uuid_unparse
...4bfc LEA    RDX,[s_%s/%s_100006d0b]
...4c03 LEA    RCX,[s_/private/var/db/com...
...4c0a MOV    param_2,0x400
...4c0f MOV    param_1,R15
...4c12 MOV    R8,R12
...4c15 XOR    EAX,EAX
...4c17 CALL   _snprintf
...4c1c CALL   FUN_100003e38
...4c21 MOV    param_1,RAX
...4c24 CALL   _objc_retainAutoreleasedRe...
...4c29 MOV    R13,RAX
...4c2c MOV    param_1,RAX
...4c2f MOV    param_2,0x1
...4c34 CALL   _os_log_type_enabled
...4c39 TEST   AL,AL
...4c3b JZ     LAB_100004c70
```

```
100004c3d
...4c3d LEA    R8=>local_e8,[RBP + -0xe0]
...4c44 MOV    dword ptr [R8]=>local_e8,0...
...4c4b MOV    qword ptr [R8 + local_e8[4...
...4c4f LEA    param_1,[__mh_execute_head...
...4c56 LEA    RCX,[s_staging_area_for_bu...
...4c5d MOV    param_2,R13
...4c60 MOV    EDX,0x1
...4c65 MOV    R9D,0xc
...4c6b CALL   __os_log_impl
```

```
100004c70 – LAB_100004c70
               LAB_100004c70
...4c70 MOV    param_1,R13
...4c73 CALL   qword ptr [-><EXTERNAL>::_...
...4c79 CALL   _copyfile_state_alloc
...4c7e MOV    R12,RAX
...4c81 LEA    RDX,[copyfile_callback]
...4c88 MOV    param_1,RAX
...4c8b MOV    param_2,0x6
...4c90 CALL   _copyfile_state_set
...4c95 LEA    param_2=>local_968,[RBP + ...
...4c9c MOV    param_1,qword ptr [RBP + l...
...4ca3 MOV    RDX,R12
...4ca6 MOV    ECX,0xc800f
...4cab CALL   _copyfile
...4cb0 MOV    R13D,EAX
...4cb3 MOV    param_1,R12
...4cb6 CALL   _copyfile_state_free
...4cbb TEST   R13D,R13D
...4cbe JZ     LAB_100004d08
```

```
100004cc0
...4cc0 CALL   FUN_100003e38
...4cc5 MOV    param_1,RAX
...4cc8 CALL   _objc_retainAutoreleasedRe...
...4ccd MOV    R12,RAX
...4cd0 MOV    param_1,RAX
...4cd3 MOV    param_2,0x10
...4cd8 CALL   _os_log_type_enabled
...4cdd TEST   AL,AL
...4cdf JNZ    LAB_1000050ee
```

```
1000050ee – LAB_1000050ee
               LAB_1000050ee
...50ee MOV    param_1,dword ptr [RBP + l...
...50f4 MOV    param_2,R12
...50f7 CALL   FUN_100006214
...50fc JMP    LAB_100004ce5
```

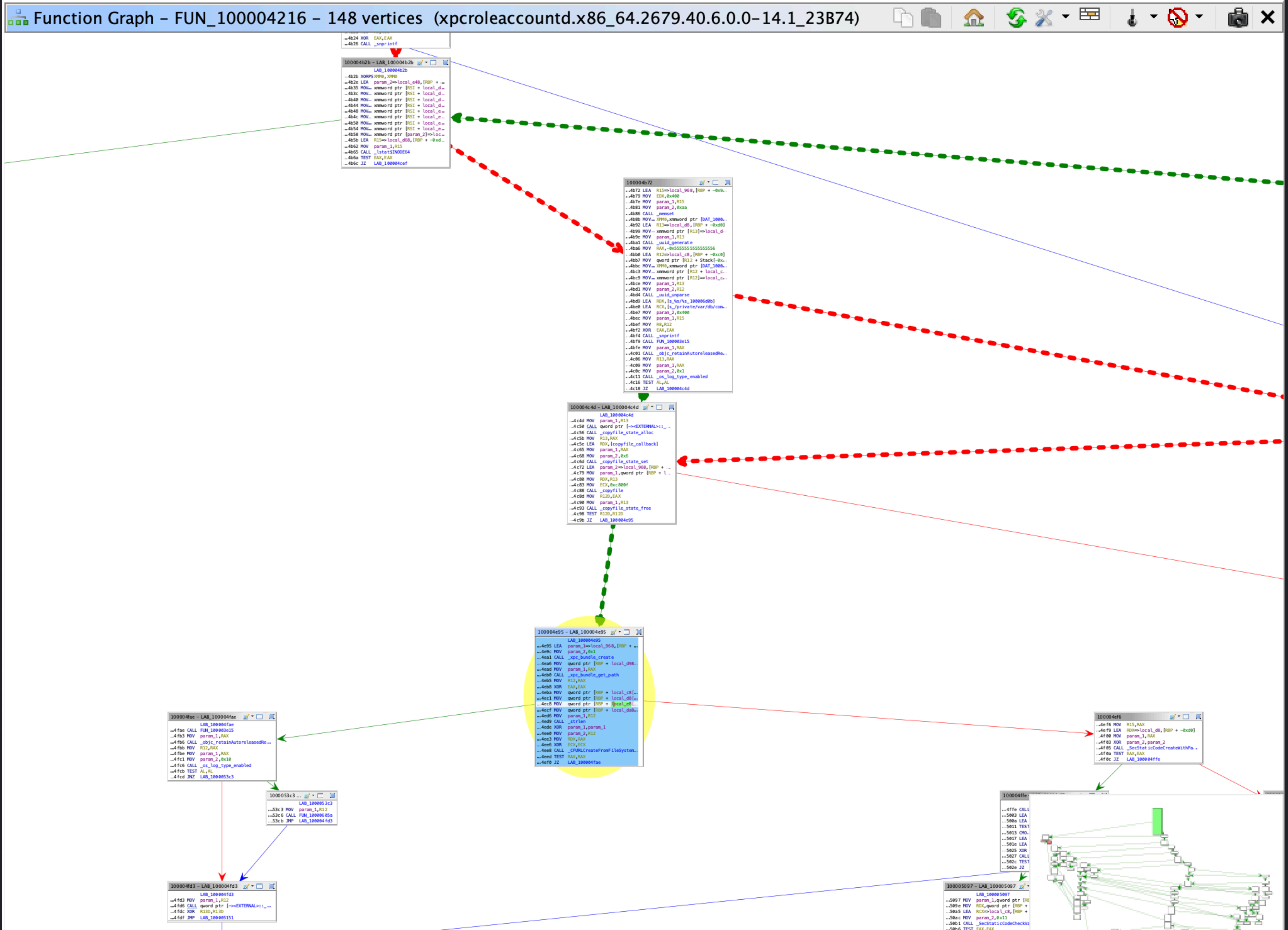
```
100004ce5 – LAB_100004ce5
               LAB_100004ce5
...4ce5 MOV    param_1,R12
...4ce8 CALL   qword ptr [-><EXTERNAL>::_...
...4cee CALL   ___error
...4cf3 MOV    R15D,dword ptr [RAX]
...4cf6 TEST   R15D,R15D
...4cf9 JZ     LAB_100004d08
```

```
100004d08 – LAB_100004d08
               LAB_100004d08
...4d08 LEA    param_1=>local_968,[RBP + ...
...4d0f MOV    param_2,0x1
...4d14 CALL   _xpc_bundle_create
...4d19 MOV    qword ptr [RBP + local_d98...
...4d20 MOV    param_1,RAX
...4d23 CALL   _xpc_bundle_get_path
...4d28 MOV    R12,RAX
...4d2b XOR    EAX,EAX
...4d2d MOV    qword ptr [RBP + local_c8[...
...4d34 MOV    qword ptr [RBP + local_d8[...
...4d3b MOV    qword ptr [RBP + local_e8[...
...4d42 MOV    qword ptr [RBP + local_da8...
...4d49 MOV    param_1,R12
...4d4c CALL   _strlen
...4d51 XOR    param_1,param_1
...4d53 MOV    param_2,R12
...4d56 MOV    RDX,RAX
...4d59 XOR    ECX,ECX
...4d5b CALL   _CFURLCreateFromFileSystem...
...4d60 TEST   RAX,RAX
...4d63 JZ     LAB_100004f6c
```

```
100004d69
...4d69 MOV    R15,RAX
...4d6c LEA    RDX=>local_d8,[RBP + -0xd0]
...4d73 MOV    param_1,RAX
...4d76 XOR    param_2,param_2
...4d78 CALL   _SecStaticCodeCreateWithPa...
...4d7d TEST   EAX,EAX
...4d7f JZ     LAB_100004ff9
```

```
100004ff9 – LAB_100004ff9
               LAB_100004ff9
...4ff9 CALL   FUN_100003c12
...4ffe LEA    RCX,[cf_entitlement["com.a...
...5005 LEA    param_1,[cf_(anchorappleor...
...500c TEST   AL,AL
...500e CMO... param_1,RCX
...5012 LEA    RDX=>local_c8,[RBP + -0xc0]
...5019 LEA    RCX=>local_e8,[RBP + -0xe0]
...5020 XOR    param_2,param_2
...5022 CALL   _SecRequirementCreateWithS...
...5027 TEST   EAX,EAX
...5029 JZ     LAB_10000507d
```

```
10000507d – LAB_10000507d
               LAB_10000507d
...507d MOV    param_1,qword ptr [RBP + l...
...5084 MOV    RDX,qword ptr [RBP + local...
...508b LEA    RCX=>local_c8,[RBP + -0xc0]
```


Function Graph – FUN_100004216 – 148 vertices (xpcroleaccountd.x86_64.2679.40.6.0.0–14.1_23B74)

```
…4b24 XOR   EAX,EAX
…4b26 CALL  _snprintf
```

```
100004b2b – LAB_100004b2b
            LAB_100004b2b
…4b2b XORPS XMM0,XMM0
…4b2e LEA   param_2=>local_e48,[RBP + …
…4b35 MOV…  xmmword ptr [RSI + local_d…
…4b3c MOV…  xmmword ptr [RSI + local_d…
…4b40 MOV…  xmmword ptr [RSI + local_d…
…4b44 MOV…  xmmword ptr [RSI + local_d…
…4b48 MOV…  xmmword ptr [RSI + local_e…
…4b4c MOV…  xmmword ptr [RSI + local_e…
…4b50 MOV…  xmmword ptr [RSI + local_e…
…4b54 MOV…  xmmword ptr [RSI + local_e…
…4b58 MOV…  xmmword ptr [param_2]=>loc…
…4b5b LEA   R15=>local_d68,[RBP + -0xd…
…4b62 MOV   param_1,R15
…4b65 CALL  _lstat$INODE64
…4b6a TEST  EAX,EAX
…4b6c JZ    LAB_100004cef
```

```
100004b72
…4b72 LEA   R15=>local_968,[RBP + -0x9…
…4b79 MOV   EDX,0x400
…4b7e MOV   param_1,R15
…4b81 MOV   param_2,0xaa
…4b86 CALL  _memset
…4b8b MOV…  XMM0,xmmword ptr [DAT_1000…
…4b92 LEA   R13=>local_d8,[RBP + -0xd0]
…4b99 MOV…  xmmword ptr [R13]=>local_d…
…4b9e MOV   param_1,R13
…4ba1 CALL  _uuid_generate
…4ba6 MOV   RAX,-0x5555555555555556
…4bb0 LEA   R12=>local_c8,[RBP + -0xc0]
…4bb7 MOV   qword ptr [R12 + Stack[-0x…
…4bbc MOV…  XMM0,xmmword ptr [DAT_1000…
…4bc3 MOV…  xmmword ptr [R12 + local_c…
…4bc9 MOV…  xmmword ptr [R12]=>local_c…
…4bce MOV   param_1,R13
…4bd1 MOV   param_2,R12
…4bd4 CALL  _uuid_unparse
…4bd9 LEA   RDX,[s_%s/%s_100006d0b]
…4be0 LEA   RCX,[s_/private/var/db/com…
…4be7 MOV   param_2,0x400
…4bec MOV   param_1,R15
…4bef MOV   R8,R12
…4bf2 XOR   EAX,EAX
…4bf4 CALL  _snprintf
…4bf9 CALL  FUN_100003e15
…4bfe MOV   param_1,RAX
…4c01 CALL  _objc_retainAutoreleasedRe…
…4c06 MOV   R13,RAX
…4c09 MOV   param_1,RAX
…4c0c MOV   param_2,0x1
…4c11 CALL  _os_log_type_enabled
…4c16 TEST  AL,AL
…4c18 JZ    LAB_100004c4d
```

```
100004c4d – LAB_100004c4d
            LAB_100004c4d
…4c4d MOV   param_1,R13
…4c50 CALL  qword ptr [-><EXTERNAL>::_…
…4c56 CALL  _copyfile_state_alloc
…4c5b MOV   R13,RAX
…4c5e LEA   RDX,[copyfile_callback]
…4c65 MOV   param_1,RAX
…4c68 MOV   param_2,0x6
…4c6d CALL  _copyfile_state_set
…4c72 LEA   param_2=>local_968,[RBP + …
…4c79 MOV   param_1,qword ptr [RBP + l…
…4c80 MOV   RDX,R13
…4c83 MOV   ECX,0xc800f
…4c88 CALL  _copyfile
…4c8d MOV   R12D,EAX
…4c90 MOV   param_1,R13
…4c93 CALL  _copyfile_state_free
…4c98 TEST  R12D,R12D
…4c9b JZ    LAB_100004e95
```

```
100004e95 – LAB_100004e95
            LAB_100004e95
…4e95 LEA   param_1=>local_968,[RBP + …
…4e9c MOV   param_2,0x1
…4ea1 CALL  _xpc_bundle_create
…4ea6 MOV   qword ptr [RBP + local_d98…
…4ead MOV   param_1,RAX
…4eb0 CALL  _xpc_bundle_get_path
…4eb5 MOV   R12,RAX
…4eb8 XOR   EAX,EAX
…4eba MOV   qword ptr [RBP + local_c8[…
…4ec1 MOV   qword ptr [RBP + local_d8[…
…4ec8 MOV   qword ptr [RBP + local_e8[…
…4ecf MOV   qword ptr [RBP + local_da8…
…4ed6 MOV   param_1,R12
…4ed9 CALL  _strlen
…4ede XOR   param_1,param_1
…4ee0 MOV   param_2,R12
…4ee3 MOV   RDX,RAX
…4ee6 XOR   ECX,ECX
…4ee8 CALL  _CFURLCreateFromFileSystem…
…4eed TEST  RAX,RAX
…4ef0 JZ    LAB_100004fae
```

```
100004fae – LAB_100004fae
            LAB_100004fae
…4fae CALL  FUN_100003e15
…4fb3 MOV   param_1,RAX
…4fb6 CALL  _objc_retainAutoreleasedRe…
…4fbb MOV   R12,RAX
…4fbe MOV   param_1,RAX
…4fc1 MOV   param_2,0x10
…4fc6 CALL  _os_log_type_enabled
…4fcb TEST  AL,AL
…4fcd JNZ   LAB_1000053c3
```

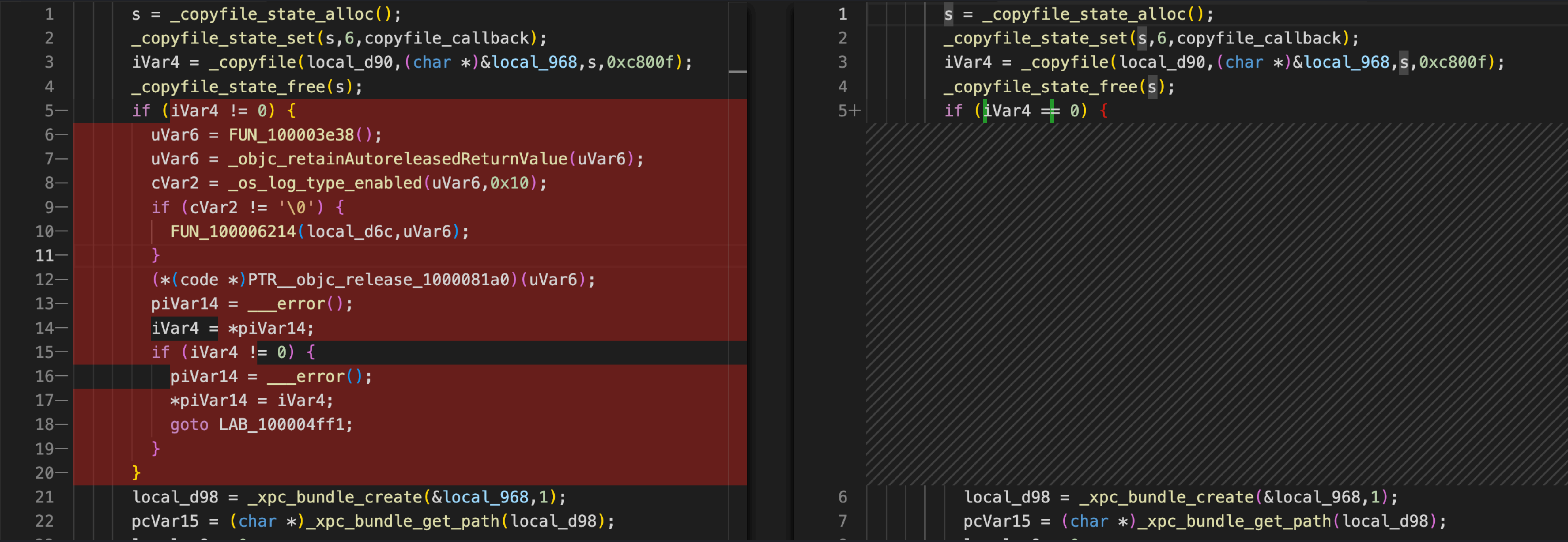
```
1000053c3 …
            LAB_1000053c3
…53c3 MOV   param_1,R12
…53c6 CALL  FUN_10000605a
…53cb JMP   LAB_100004fd3
```

```
100004fd3 – LAB_100004fd3
            LAB_100004fd3
…4fd3 MOV   param_1,R12
…4fd6 CALL  qword ptr [-><EXTERNAL>::_…
…4fdc XOR   R13D,R13D
…4fdf JMP   LAB_100005151
```

```
100004ef6
…4ef6 MOV   R15,RAX
…4ef9 LEA   RDX=>local_d8,[RBP + -0xd0]
…4f00 MOV   param_1,RAX
…4f03 XOR   param_2,param_2
…4f05 CALL  _SecStaticCodeCreateWithPa…
…4f0a TEST  EAX,EAX
…4f0c JZ    LAB_100004ffe
```

```
100004ffe
…4ffe CALL
…5003 LEA
…500a LEA
…5011 TEST
…5013 CMO…
…5017 LEA
…501e LEA
…5025 XOR
…5027 CALL
…502c TEST
…502e JZ
```

```
100005097 – LAB_100005097
            LAB_100005097
…5097 MOV   param_1,qword ptr [RB
…509e MOV   RDX,qword ptr [RBP +
…50a5 LEA   RCX=>local_c8,[RBP +
…50ac MOV   param_2,0x11
…50b1 CALL  _SecStaticCodeCheckVa
…50b6 TEST  EAX,EAX
```



```
s = _copyfile_state_alloc();
_copyfile_state_set(s,6,copyfile_callback);
iVar4 = _copyfile(local_d90,(char *)&local_968,s,0xc800f);
_copyfile_state_free(s);
if (iVar4 != 0) {
  uVar6 = FUN_100003e38();
  uVar6 = _objc_retainAutoreleasedReturnValue(uVar6);
  cVar2 = _os_log_type_enabled(uVar6,0x10);
  if (cVar2 != '\0') {
    FUN_100006214(local_d6c,uVar6);
  }
  (*(code *)PTR__objc_release_1000081a0)(uVar6);
  piVar14 = ___error();
  iVar4 = *piVar14;
  if (iVar4 != 0) {
    piVar14 = ___error();
    *piVar14 = iVar4;
    goto LAB_100004ff1;
  }
}
local_d98 = _xpc_bundle_create(&local_968,1);
pcVar15 = (char *)_xpc_bundle_get_path(local_d98);
```

```
s = _copyfile_state_alloc();
_copyfile_state_set(s,6,copyfile_callback);
iVar4 = _copyfile(local_d90,(char *)&local_968,s,0xc800f);
_copyfile_state_free(s);
if (iVar4 == 0) {
  local_d98 = _xpc_bundle_create(&local_968,1);
  pcVar15 = (char *)_xpc_bundle_get_path(local_d98);
```

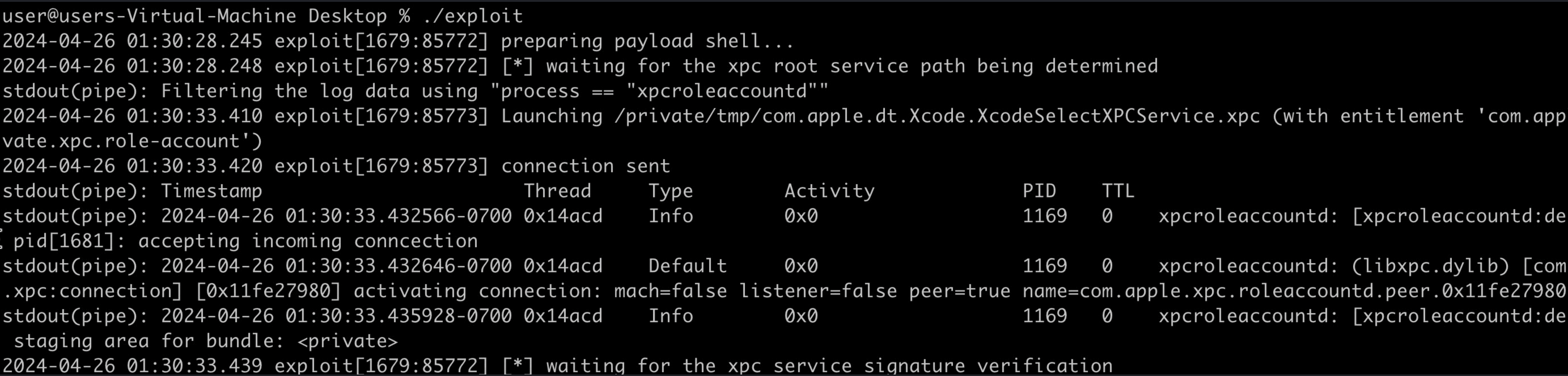
The root issue is that the code had a path to get back to "success" after the copyfile failure!

```
        // Use copyfile to copy the file to /tmp
        int result = copyfile([sourcePath UTF8String], [destinationPath UTF8String], NULL, COPYFILE_ALL);
        if (result != 0) {
            NSLog(@"Error: Failed to copy file. errno: %d", errno);

            // Check errno one more time just to make sure
            if (errno == 0):
              goto: success;  // Root Cause!!



            return 1;
        }

success:

        NSLog(@"File copied successfully to %@", destinationPath);
```

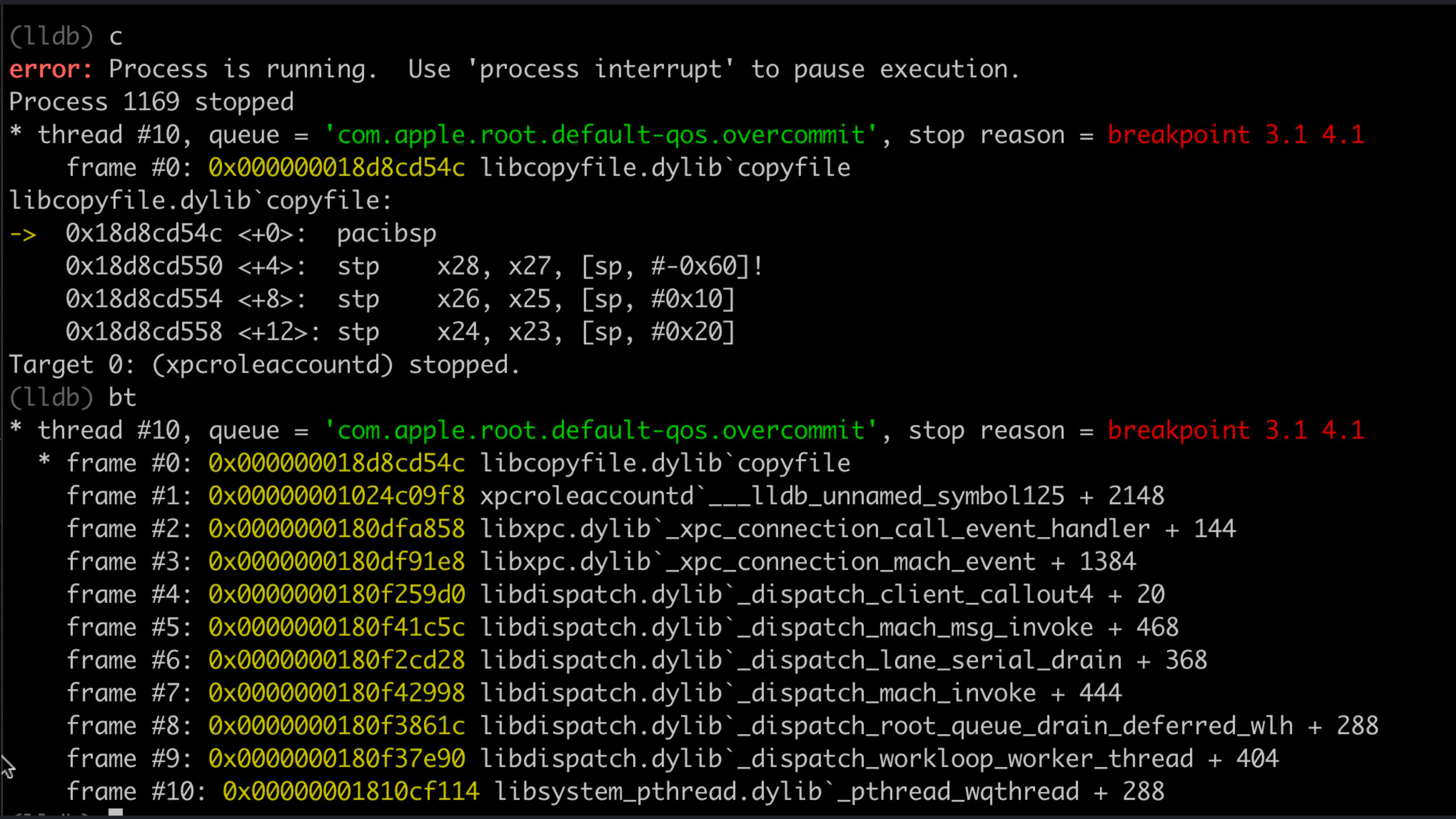
Static Analysis

Dynamic Analysis

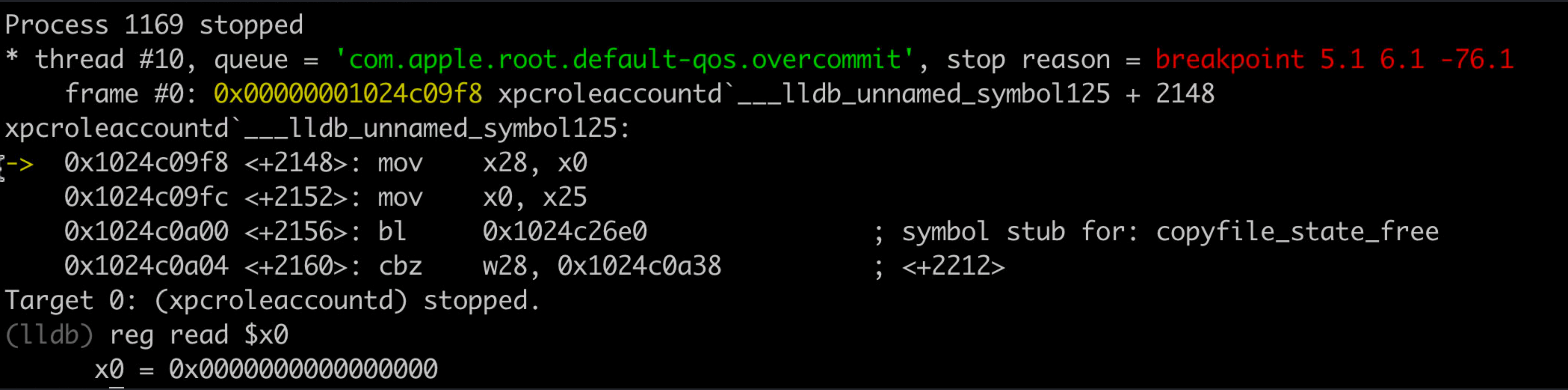
https://github.com/jhftss/POC/blob/main/CVE-2023-42942/



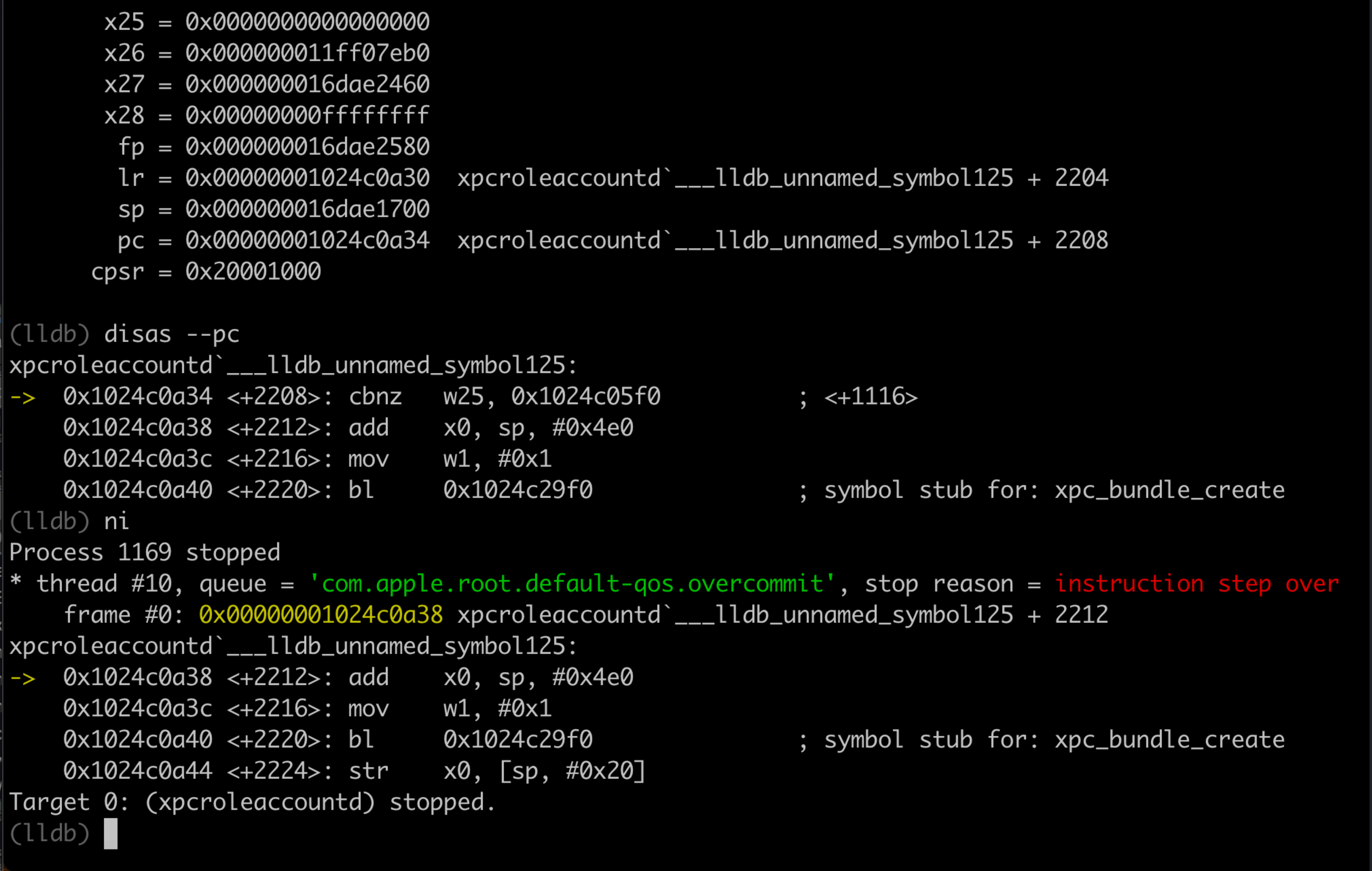
```
user@users-Virtual-Machine Desktop % ./exploit
2024-04-26 01:30:28.245 exploit[1679:85772] preparing payload shell...
2024-04-26 01:30:28.248 exploit[1679:85772] [*] waiting for the xpc root service path being determined
stdout(pipe): Filtering the log data using "process == "xpcroleaccountd""
2024-04-26 01:30:33.410 exploit[1679:85773] Launching /private/tmp/com.apple.dt.Xcode.XcodeSelectXPCService.xpc (with entitlement 'com.app
vate.xpc.role-account')
2024-04-26 01:30:33.420 exploit[1679:85773] connection sent
stdout(pipe): Timestamp                       Thread     Type        Activity             PID    TTL
stdout(pipe): 2024-04-26 01:30:33.432566-0700 0x14acd    Info        0x0                  1169   0    xpcroleaccountd: [xpcroleaccountd:de
 pid[1681]: accepting incoming conncection
stdout(pipe): 2024-04-26 01:30:33.432646-0700 0x14acd    Default     0x0                  1169   0    xpcroleaccountd: (libxpc.dylib) [com
.xpc:connection] [0x11fe27980] activating connection: mach=false listener=false peer=true name=com.apple.xpc.roleaccountd.peer.0x11fe27980
stdout(pipe): 2024-04-26 01:30:33.435928-0700 0x14acd    Info        0x0                  1169   0    xpcroleaccountd: [xpcroleaccountd:de
 staging area for bundle: <private>
2024-04-26 01:30:33.439 exploit[1679:85772] [*] waiting for the xpc service signature verification
```



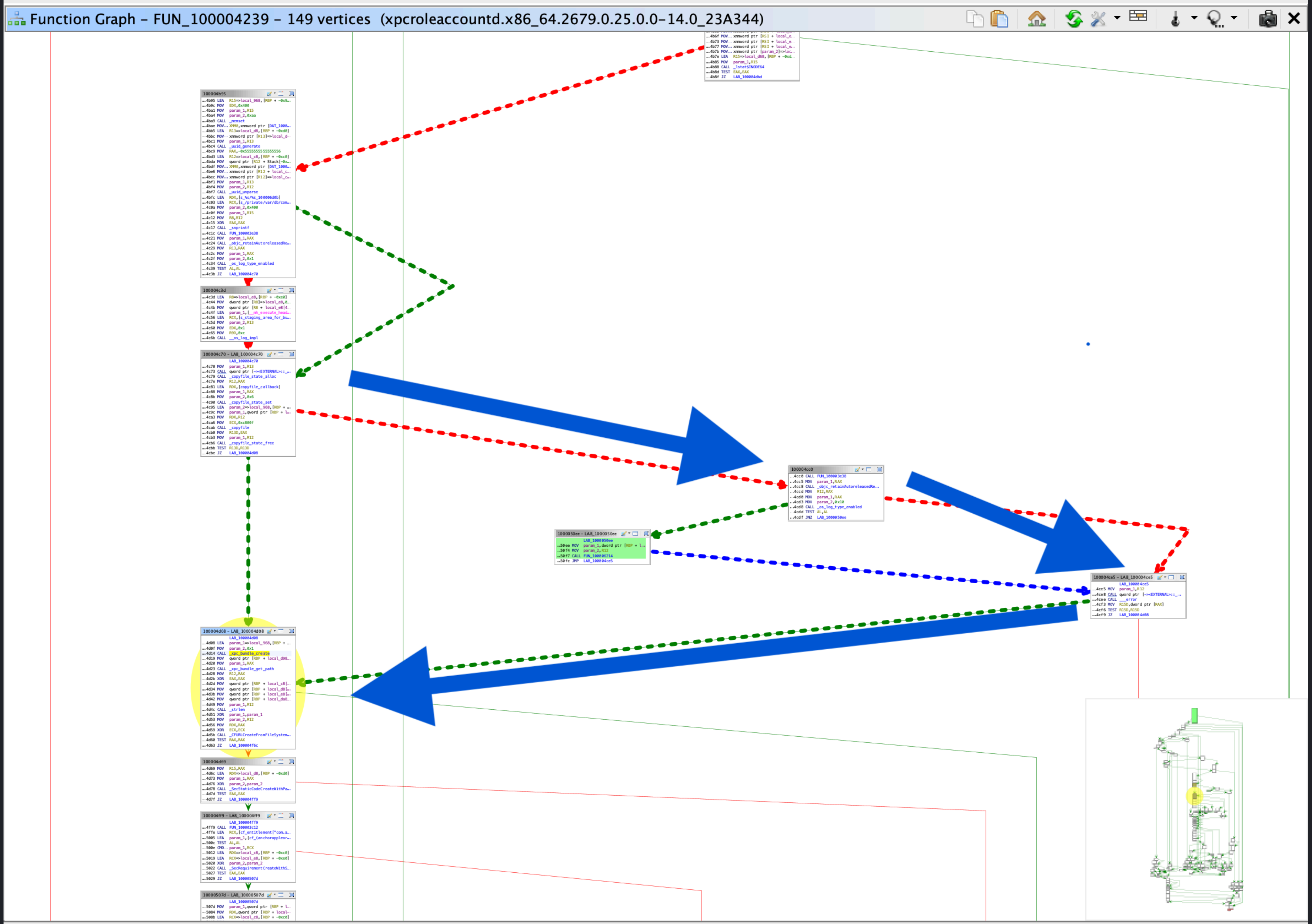
```
(lldb) c
error: Process is running.  Use 'process interrupt' to pause execution.
Process 1169 stopped
* thread #10, queue = 'com.apple.root.default-qos.overcommit', stop reason = breakpoint 3.1 4.1
    frame #0: 0x000000018d8cd54c libcopyfile.dylib`copyfile
libcopyfile.dylib`copyfile:
->  0x18d8cd54c <+0>:  pacibsp
    0x18d8cd550 <+4>:  stp    x28, x27, [sp, #-0x60]!
    0x18d8cd554 <+8>:  stp    x26, x25, [sp, #0x10]
    0x18d8cd558 <+12>: stp    x24, x23, [sp, #0x20]
Target 0: (xpcroleaccountd) stopped.
(lldb) bt
* thread #10, queue = 'com.apple.root.default-qos.overcommit', stop reason = breakpoint 3.1 4.1
  * frame #0: 0x000000018d8cd54c libcopyfile.dylib`copyfile
    frame #1: 0x00000001024c09f8 xpcroleaccountd`___lldb_unnamed_symbol125 + 2148
    frame #2: 0x0000000180dfa858 libxpc.dylib`_xpc_connection_call_event_handler + 144
    frame #3: 0x0000000180df91e8 libxpc.dylib`_xpc_connection_mach_event + 1384
    frame #4: 0x0000000180f259d0 libdispatch.dylib`_dispatch_client_callout4 + 20
    frame #5: 0x0000000180f41c5c libdispatch.dylib`_dispatch_mach_msg_invoke + 468
    frame #6: 0x0000000180f2cd28 libdispatch.dylib`_dispatch_lane_serial_drain + 368
    frame #7: 0x0000000180f42998 libdispatch.dylib`_dispatch_mach_invoke + 444
    frame #8: 0x0000000180f3861c libdispatch.dylib`_dispatch_root_queue_drain_deferred_wlh + 288
    frame #9: 0x0000000180f37e90 libdispatch.dylib`_dispatch_workloop_worker_thread + 404
    frame #10: 0x00000001810cf114 libsystem_pthread.dylib`_pthread_wqthread + 288
```



```
Process 1169 stopped
* thread #10, queue = 'com.apple.root.default-qos.overcommit', stop reason = breakpoint 5.1 6.1 -76.1
    frame #0: 0x00000001024c09f8 xpcroleaccountd`___lldb_unnamed_symbol125 + 2148
xpcroleaccountd`___lldb_unnamed_symbol125:
->  0x1024c09f8 <+2148>: mov    x28, x0
    0x1024c09fc <+2152>: mov    x0, x25
    0x1024c0a00 <+2156>: bl     0x1024c26e0               ; symbol stub for: copyfile_state_free
    0x1024c0a04 <+2160>: cbz    w28, 0x1024c0a38          ; <+2212>
Target 0: (xpcroleaccountd) stopped.
(lldb) reg read $x0
      x0 = 0x0000000000000000
```



```
       x25 = 0x0000000000000000
       x26 = 0x000000011ff07eb0
       x27 = 0x000000016dae2460
       x28 = 0x00000000ffffffff
        fp = 0x000000016dae2580
        lr = 0x00000001024c0a30  xpcroleaccountd`___lldb_unnamed_symbol125 + 2204
        sp = 0x000000016dae1700
        pc = 0x00000001024c0a34  xpcroleaccountd`___lldb_unnamed_symbol125 + 2208
      cpsr = 0x20001000

(lldb) disas --pc
xpcroleaccountd`___lldb_unnamed_symbol125:
->  0x1024c0a34 <+2208>: cbnz   w25, 0x1024c05f0          ; <+1116>
    0x1024c0a38 <+2212>: add    x0, sp, #0x4e0
    0x1024c0a3c <+2216>: mov    w1, #0x1
    0x1024c0a40 <+2220>: bl     0x1024c29f0               ; symbol stub for: xpc_bundle_create
(lldb) ni
Process 1169 stopped
* thread #10, queue = 'com.apple.root.default-qos.overcommit', stop reason = instruction step over
    frame #0: 0x00000001024c0a38 xpcroleaccountd`___lldb_unnamed_symbol125 + 2212
xpcroleaccountd`___lldb_unnamed_symbol125:
->  0x1024c0a38 <+2212>: add    x0, sp, #0x4e0
    0x1024c0a3c <+2216>: mov    w1, #0x1
    0x1024c0a40 <+2220>: bl     0x1024c29f0               ; symbol stub for: xpc_bundle_create
    0x1024c0a44 <+2224>: str    x0, [sp, #0x20]
Target 0: (xpcroleaccountd) stopped.
(lldb) 
```



Function Graph - FUN_100004239 - 149 vertices (xpcroleaccountd.x86_64.2679.0.25.0.0-14.0_23A344)

```
...4b6f MOV... xmmword ptr [RSI + local_e...
...4b73 MOV... xmmword ptr [RSI + local_e...
...4b77 MOV... xmmword ptr [RSI + local_e...
...4b7b MOV... xmmword ptr [param_2]=>loc...
...4b7e LEA    R15=>local_d68,[RBP + -0xd...
...4b85 MOV    param_1,R15
...4b88 CALL   _lstat$INODE64
...4b8d TEST   EAX,EAX
...4b8f JZ     LAB_100004dbd
```

```
100004b95
...4b95 LEA    R15=>local_968,[RBP + -0x9...
...4b9c MOV    EDX,0x400
...4ba1 MOV    param_1,R15
...4ba4 MOV    param_2,0xaa
...4ba9 CALL   _memset
...4bae MOV... XMM0,xmmword ptr [DAT_1000...
...4bb5 LEA    R13=>local_d8,[RBP + -0xd0]
...4bbc MOV... xmmword ptr [R13]=>local_d-...
...4bc1 MOV    param_1,R13
...4bc4 CALL   _uuid_generate
...4bc9 MOV    RAX,-0x5555555555555556
...4bd3 LEA    R12=>local_c8,[RBP + -0xc0]
...4bda MOV    qword ptr [R12 + Stack[-0x...
...4bdf MOV... XMM0,xmmword ptr [DAT_1000...
...4be6 MOV... xmmword ptr [R12 + local_c...
...4bec MOV... xmmword ptr [R12]=>local_c...
...4bf1 MOV    param_1,R13
...4bf4 MOV    param_2,R12
...4bf7 CALL   _uuid_unparse
...4bfc LEA    RDX,[s_%s/%s_100006d0b]
...4c03 LEA    RCX,[s_/private/var/db/com...
...4c0a MOV    param_2,0x400
...4c0f MOV    param_1,R15
...4c12 MOV    R8,R12
...4c15 XOR    EAX,EAX
...4c17 CALL   _snprintf
...4c1c CALL   FUN_100003e38
...4c21 MOV    param_1,RAX
...4c24 CALL   _objc_retainAutoreleasedRe...
...4c29 MOV    R13,RAX
...4c2c MOV    param_1,RAX
...4c2f MOV    param_2,0x1
...4c34 CALL   _os_log_type_enabled
...4c39 TEST   AL,AL
...4c3b JZ     LAB_100004c70
```

```
100004c3d
...4c3d LEA    R8=>local_e8,[RBP + -0xe0]
...4c44 MOV    dword ptr [R8]=>local_e8,0...
...4c4b MOV    qword ptr [R8 + local_e8[4-...
...4c4f LEA    param_1,[__mh_execute_head...
...4c56 LEA    RCX,[s_staging_area_for_bu...
...4c5d MOV    param_2,R13
...4c60 MOV    EDX,0x1
...4c65 MOV    R9D,0xc
...4c6b CALL   __os_log_impl
```

```
100004c70 - LAB_100004c70
               LAB_100004c70
...4c70 MOV    param_1,R13
...4c73 CALL   qword ptr [-><EXTERNAL>::_...
...4c79 CALL   _copyfile_state_alloc
...4c7e MOV    R12,RAX
...4c81 LEA    RDX,[copyfile_callback]
...4c88 MOV    param_1,RAX
...4c8b MOV    param_2,0x6
...4c90 CALL   _copyfile_state_set
...4c95 LEA    param_2=>local_968,[RBP + ...
...4c9c MOV    param_1,qword ptr [RBP + l...
...4ca3 MOV    RDX,R12
...4ca6 MOV    ECX,0xc800f
...4cab CALL   _copyfile
...4cb0 MOV    R13D,EAX
...4cb3 MOV    param_1,R12
...4cb6 CALL   _copyfile_state_free
...4cbb TEST   R13D,R13D
...4cbe JZ     LAB_100004d08
```

```
100004cc0
...4cc0 CALL   FUN_100003e38
...4cc5 MOV    param_1,RAX
...4cc8 CALL   _objc_retainAutoreleasedRe...
...4ccd MOV    R12,RAX
...4cd0 MOV    param_1,RAX
...4cd3 MOV    param_2,0x10
...4cd8 CALL   _os_log_type_enabled
...4cdd TEST   AL,AL
...4cdf JNZ    LAB_1000050ee
```

```
1000050ee - LAB_1000050ee
               LAB_1000050ee
...50ee MOV    param_1,dword ptr [RBP + l...
...50f4 MOV    param_2,R12
...50f7 CALL   FUN_100006214
...50fc JMP    LAB_100004ce5
```

```
100004ce5 - LAB_100004ce5
               LAB_100004ce5
...4ce5 MOV    param_1,R12
...4ce8 CALL   qword ptr [-><EXTERNAL>::_...
...4cee CALL   ___error
...4cf3 MOV    R15D,dword ptr [RAX]
...4cf6 TEST   R15D,R15D
...4cf9 JZ     LAB_100004d08
```

```
100004d08 - LAB_100004d08
               LAB_100004d08
...4d08 LEA    param_1=>local_968,[RBP + ...
...4d0f MOV    param_2,0x1
...4d14 CALL   _xpc_bundle_create
...4d19 MOV    qword ptr [RBP + local_d98...
...4d20 MOV    param_1,RAX
...4d23 CALL   _xpc_bundle_get_path
...4d28 MOV    R12,RAX
...4d2b XOR    EAX,EAX
...4d2d MOV    qword ptr [RBP + local_c8[...
...4d34 MOV    qword ptr [RBP + local_d8[...
...4d3b MOV    qword ptr [RBP + local_e8[...
...4d42 MOV    qword ptr [RBP + local_da8...
...4d49 MOV    param_1,R12
...4d4c CALL   _strlen
...4d51 XOR    param_1,param_1
...4d53 MOV    param_2,R12
...4d56 MOV    RDX,RAX
...4d59 XOR    ECX,ECX
...4d5b CALL   _CFURLCreateFromFileSystem...
...4d60 TEST   RAX,RAX
...4d63 JZ     LAB_100004f6c
```

```
100004d69
...4d69 MOV    R15,RAX
...4d6c LEA    RDX=>local_d8,[RBP + -0xd0]
...4d73 MOV    param_1,RAX
...4d76 XOR    param_2,param_2
...4d78 CALL   _SecStaticCodeCreateWithPa...
...4d7d TEST   EAX,EAX
...4d7f JZ     LAB_100004ff9
```

```
100004ff9 - LAB_100004ff9
               LAB_100004ff9
...4ff9 CALL   FUN_100003c12
...4ffe LEA    RCX,[cf_entitlement["com.a...
...5005 LEA    param_1,[cf_(anchorappleor...
...500c TEST   AL,AL
...500e CMO... param_1,RCX
...5012 LEA    RDX=>local_c8,[RBP + -0xc0]
...5019 LEA    RCX=>local_e8,[RBP + -0xe0]
...5020 XOR    param_2,param_2
...5022 CALL   _SecRequirementCreateWithS...
...5027 TEST   EAX,EAX
...5029 JZ     LAB_10000507d
```

```
10000507d - LAB_10000507d
               LAB_10000507d
...507d MOV    param_1,qword ptr [RBP + L...
...5084 MOV    RDX,qword ptr [RBP + local-...
...508b LEA    RCX=>local_c8,[RBP + -0xc0]
```

CVE-2023-42942 - TOCTOU LPE

- Started with a ton of info (@patch1t 🙏)
- Used Ghidra analysis and diffing to dig deeper 🧐
- RE'd code I wrote for understanding++
- Leveraged dynamic analysis to confirm possible root cause path

CVE-2022-46718

Sensitive Information Leak

TCC

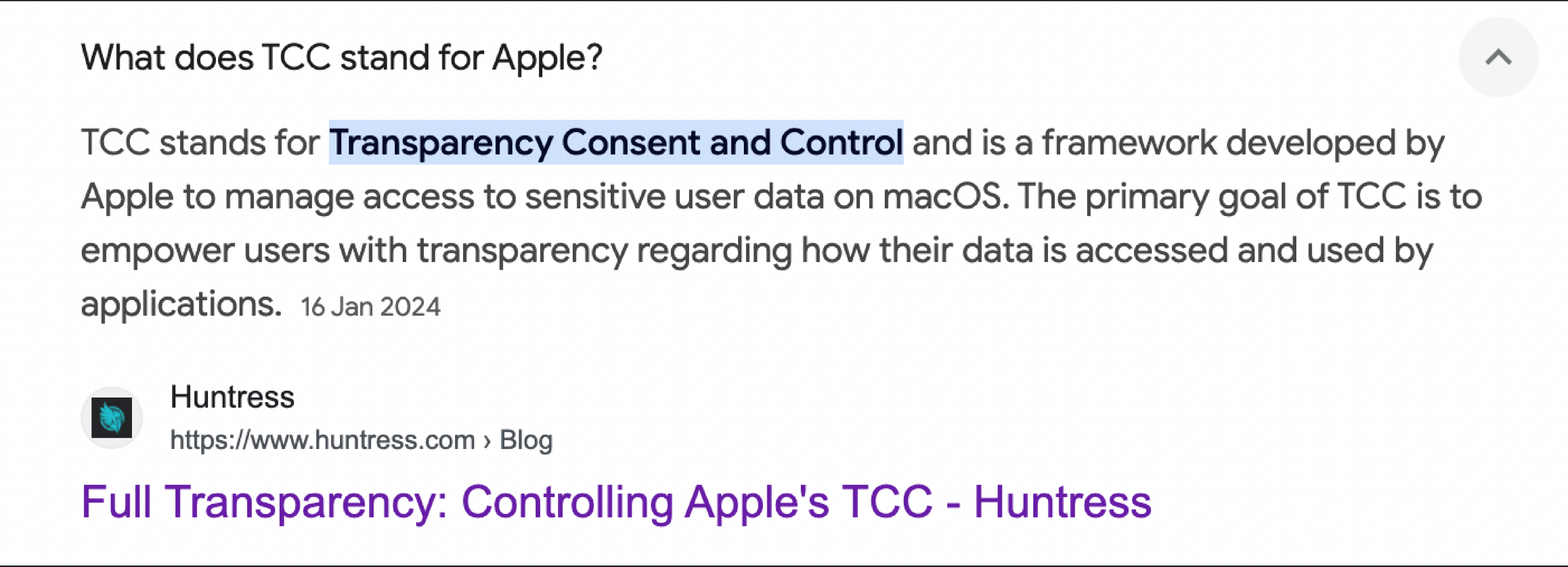
Available for: iPhone 8 and later, iPad Pro (all models), iPad Air 3rd generation and later, iPad 5th generation and later, and iPad mini 5th generation and later

Impact: An app may be able to read sensitive location information

Description: A logic issue was addressed with improved restrictions.

CVE-2022-46718: Michael (Biscuit) Thomas

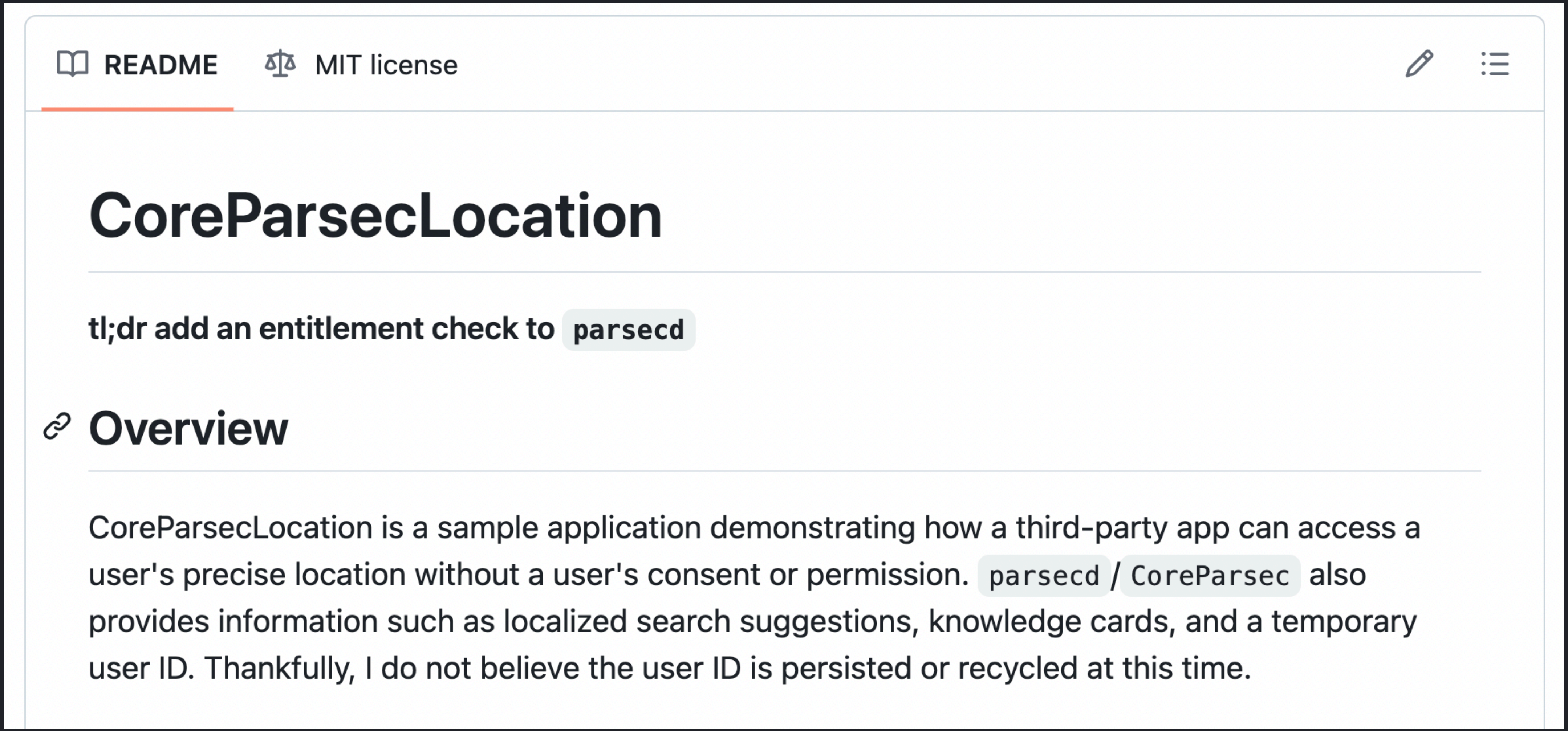
Entry added May 1, 2023


What does TCC stand for Apple?

TCC stands for Transparency Consent and Control and is a framework developed by Apple to manage access to sensitive user data on macOS. The primary goal of TCC is to empower users with transparency regarding how their data is accessed and used by applications. 16 Jan 2024

Huntress
https://www.huntress.com › Blog

Full Transparency: Controlling Apple's TCC - Huntress


README MIT license

CoreParsecLocation

tl;dr add an entitlement check to `parsecd`

Overview

CoreParsecLocation is a sample application demonstrating how a third-party app can access a user's precise location without a user's consent or permission. `parsecd` / `CoreParsec` also provides information such as localized search suggestions, knowledge cards, and a temporary user ID. Thankfully, I do not believe the user ID is persisted or recycled at this time.

https://github.com/biscuitehh/cve-2022-46718-leaky-location

🆕 SearchPoirotExtension

`/System/Library/PrivateFrameworks/CoreParsec.framework/PlugIns/SearchPoirotExtension.appex/SearchPoirotExtension`

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>com.apple.intents.extension.discovery</key>
    <true/>
    <key>com.apple.runningboard.launchprocess</key>
    <true/>
    <key>com.apple.security.app-sandbox</key>
    <true/>
    <key>com.apple.security.application-groups</key>
    <array>
        <string>group.com.apple.PegasusConfiguration</string>
    </array>
    <key>com.apple.security.exception.mach-lookup.global-name</key>
    <array>
        <string>com.apple.parsecd</string>
    </array>
</dict>
</plist>
```

parsec-fbf

`/System/Library/PrivateFrameworks/CoreParsec.framework/parsec-fbf`

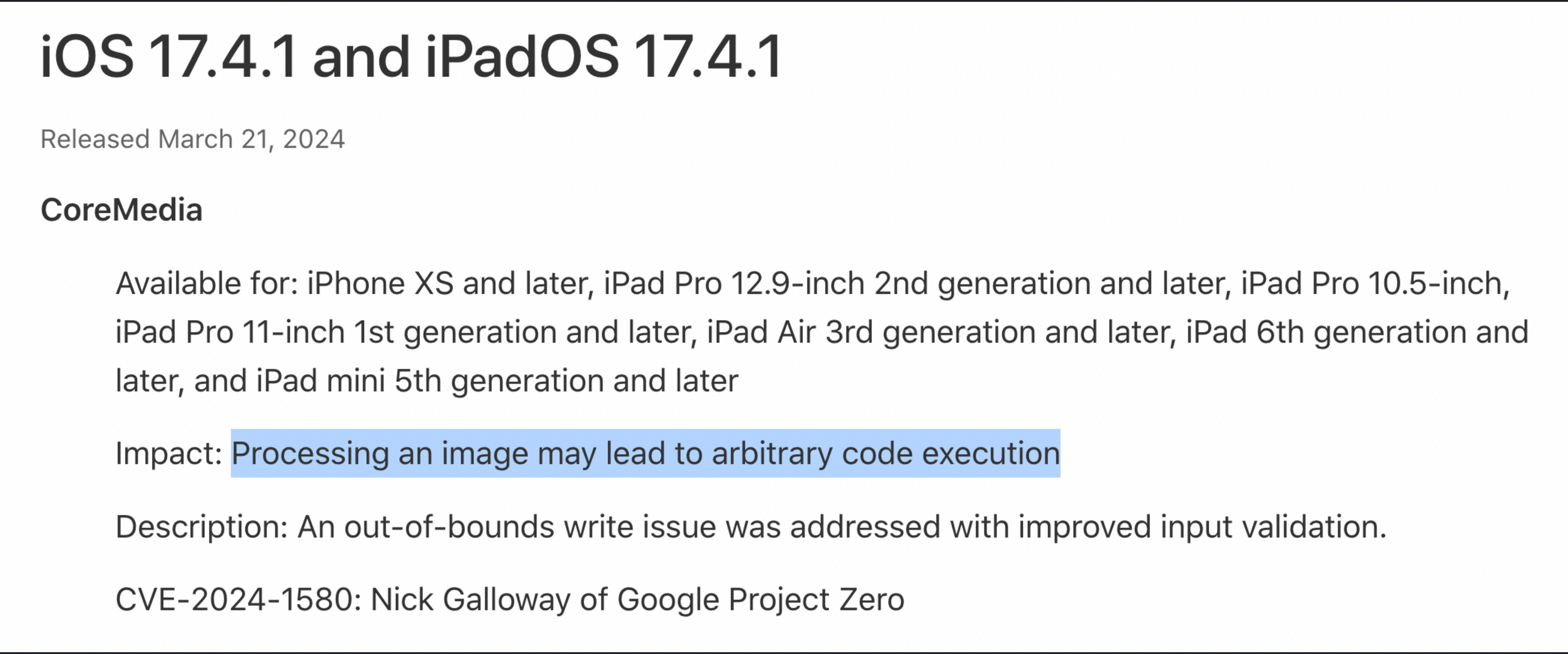
```
    <string>com.apple.parsec-fbf</string>
    <key>com.apple.developer.networking.multipath_extended</key>
    <true/>
+   <key>com.apple.extensionkit.host.extension-point-identifiers</key>
+   <array>
+       <string>com.apple.mlruntime.extension-point-ondemand</string>
+   </array>
    <key>com.apple.intents.extension.discovery</key>
    <true/>
+   <key>com.apple.mlruntime.host.ondemand</key>
+   <true/>
+   <key>com.apple.mlruntime.host.ondemandplugin</key>
+   <array>
+       <string>com.apple.siri.parsec.CoreParsec.SearchPoirotExtension</string>
+   </array>
    <key>com.apple.private.feedbacklogger</key>
    <true/>
    <key>com.apple.security.application-groups</key>
    <array>
        <string>group.com.apple.PegasusConfiguration</string>
    </array>
+   <key>com.apple.security.exception.mach-lookup.global-name</key>
+   <array>
+       <string>com.apple.extensionkitservice</string>
+       <string>com.apple.siri.parsec.CoreParsec.SearchPoirotExtension</string>
+   </array>
    <key>com.apple.security.exception.shared-preference.read-only</key>
```

CVE-2022-46718 - Sensitive Info Leak

- No easy way to map binaries for this CVE
- Too many potential changes for me to root cause
- Diffing can't overcome inexperience, but it can reveal it
- CVE fixed with an entitlement change

CVE-2024-1580 - int overflow

(super speed mode)



iOS 17.4.1 and iPadOS 17.4.1

Released March 21, 2024

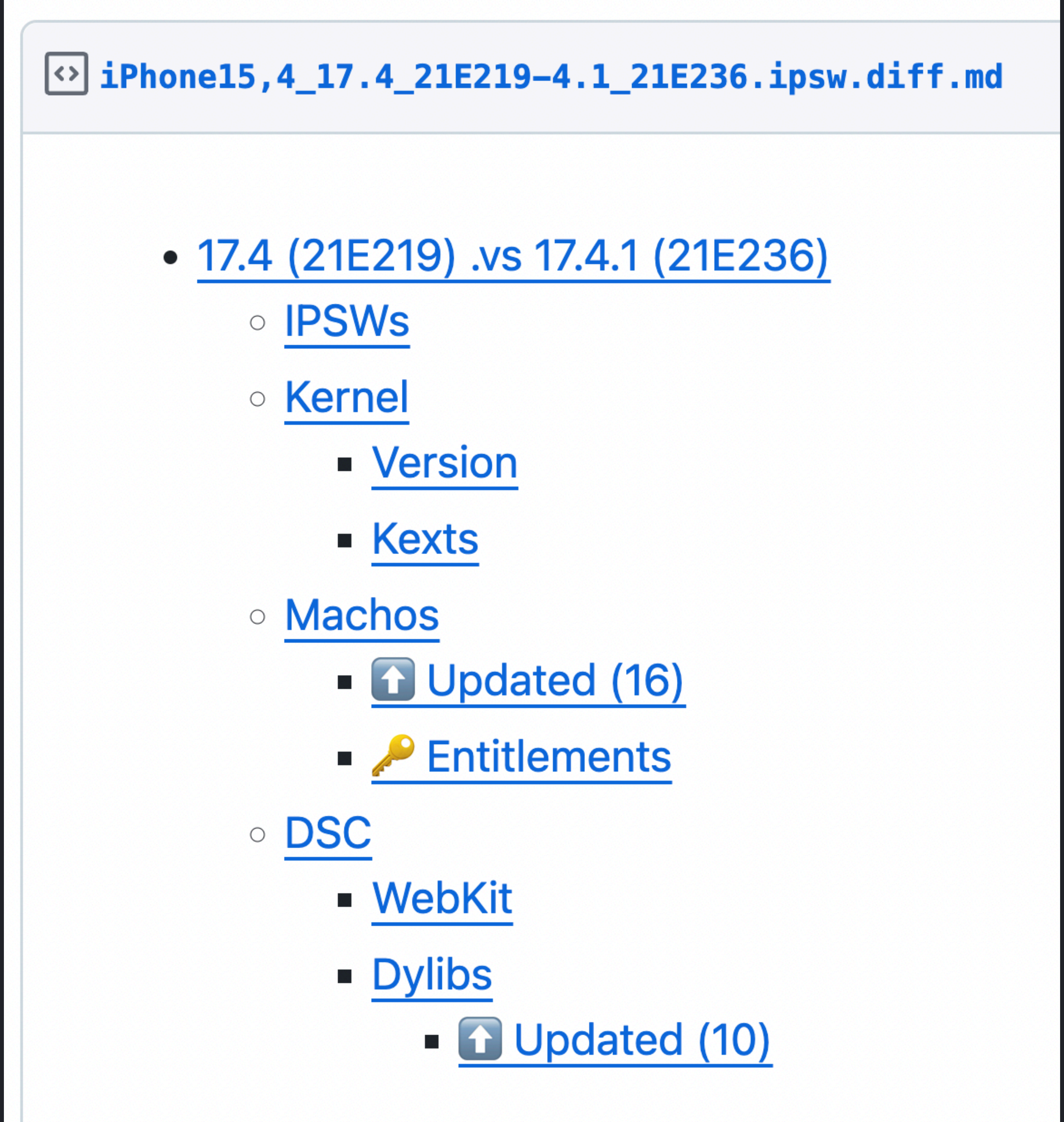
CoreMedia

Available for: iPhone XS and later, iPad Pro 12.9-inch 2nd generation and later, iPad Pro 10.5-inch, iPad Pro 11-inch 1st generation and later, iPad Air 3rd generation and later, iPad 6th generation and later, and iPad mini 5th generation and later

Impact: Processing an image may lead to arbitrary code execution

Description: An out-of-bounds write issue was addressed with improved input validation.

CVE-2024-1580: Nick Galloway of Google Project Zero



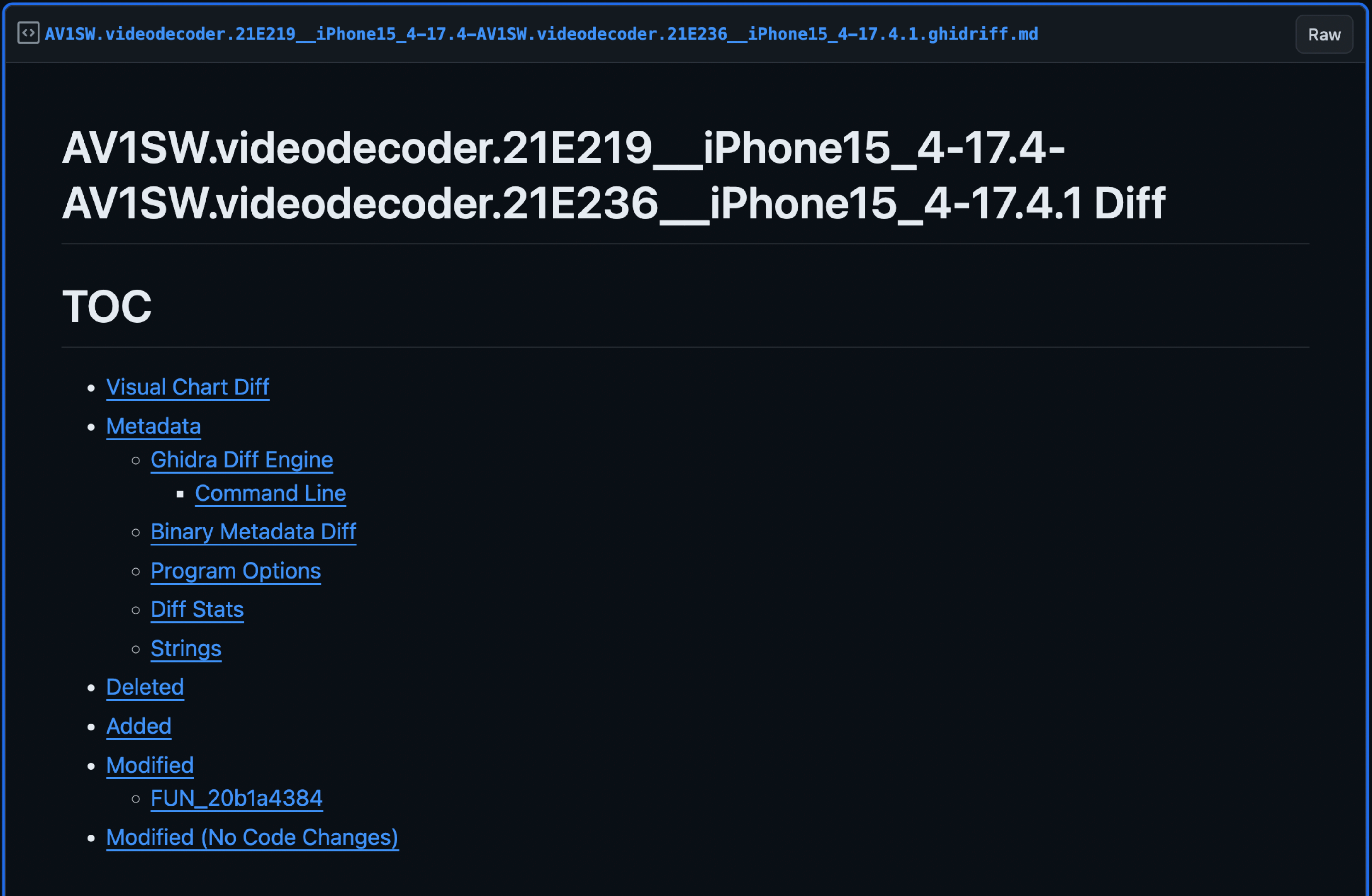
iPhone15,4_17.4_21E219-4.1_21E236.ipsw.diff.md

- 17.4 (21E219) .vs 17.4.1 (21E236)
  - IPSWs
  - Kernel
    - Version
    - Kexts
  - Machos
    - ⬆️ Updated (16)
    - 🔑 Entitlements
  - DSC
    - WebKit
    - Dylibs
      - ⬆️ Updated (10)

`AV1SW.videodecoder`

> `/System/Library/VideoDecoders/AV1SW.videodecoder`

```diff
-54.5.0.0.0
-  __TEXT.__text: 0x91888
+80.1.0.0.0
+  __TEXT.__text: 0x91880
   __TEXT.__auth_stubs: 0x990
   __TEXT.__objc_methlist: 0xc8
   __TEXT.__const: 0x187cc
```

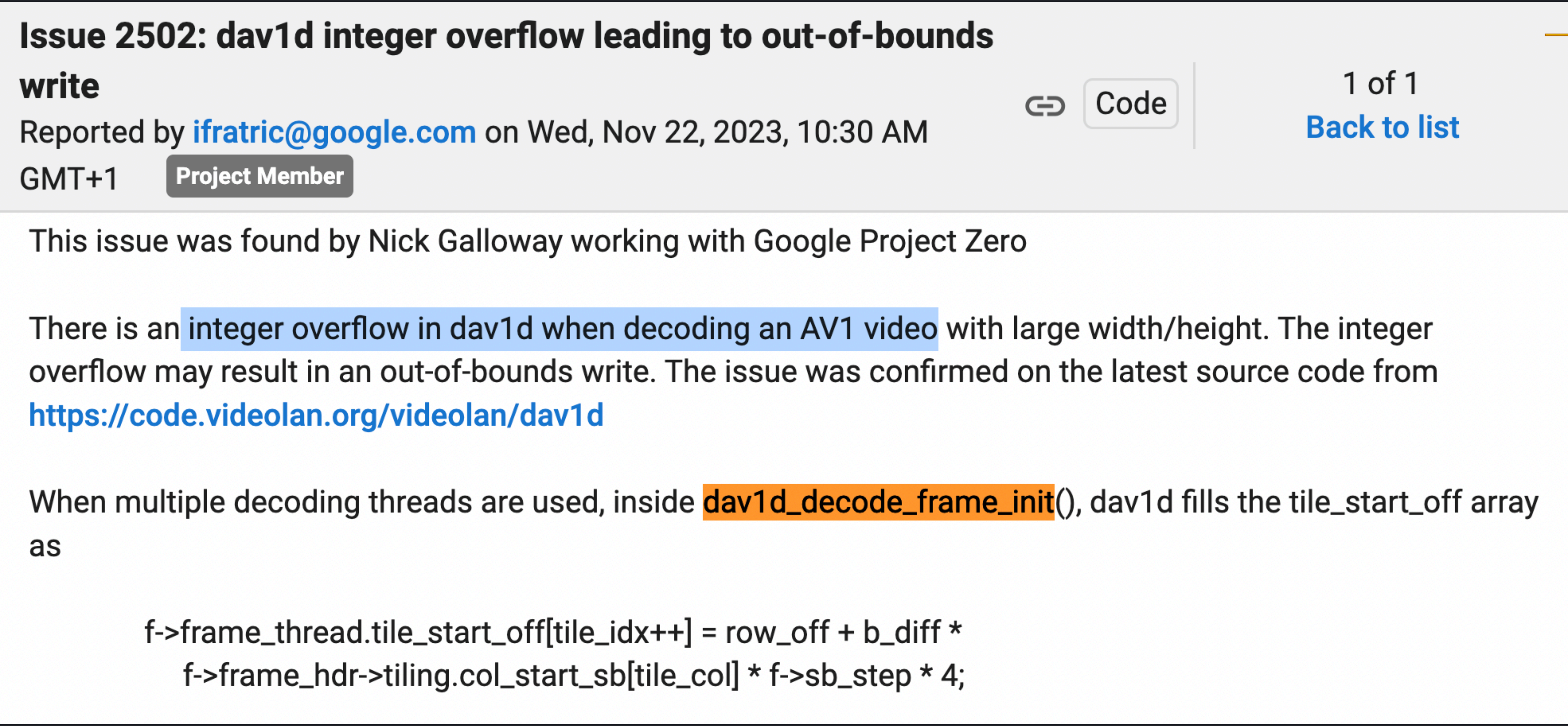


AV1SW.videodecoder.21E219__iPhone15_4-17.4-AV1SW.videodecoder.21E236__iPhone15_4-17.4.1.ghidriff.md

Raw

AV1SW.videodecoder.21E219__iPhone15_4-17.4-AV1SW.videodecoder.21E236__iPhone15_4-17.4.1 Diff

TOC



Issue 2502: dav1d integer overflow leading to out-of-bounds write

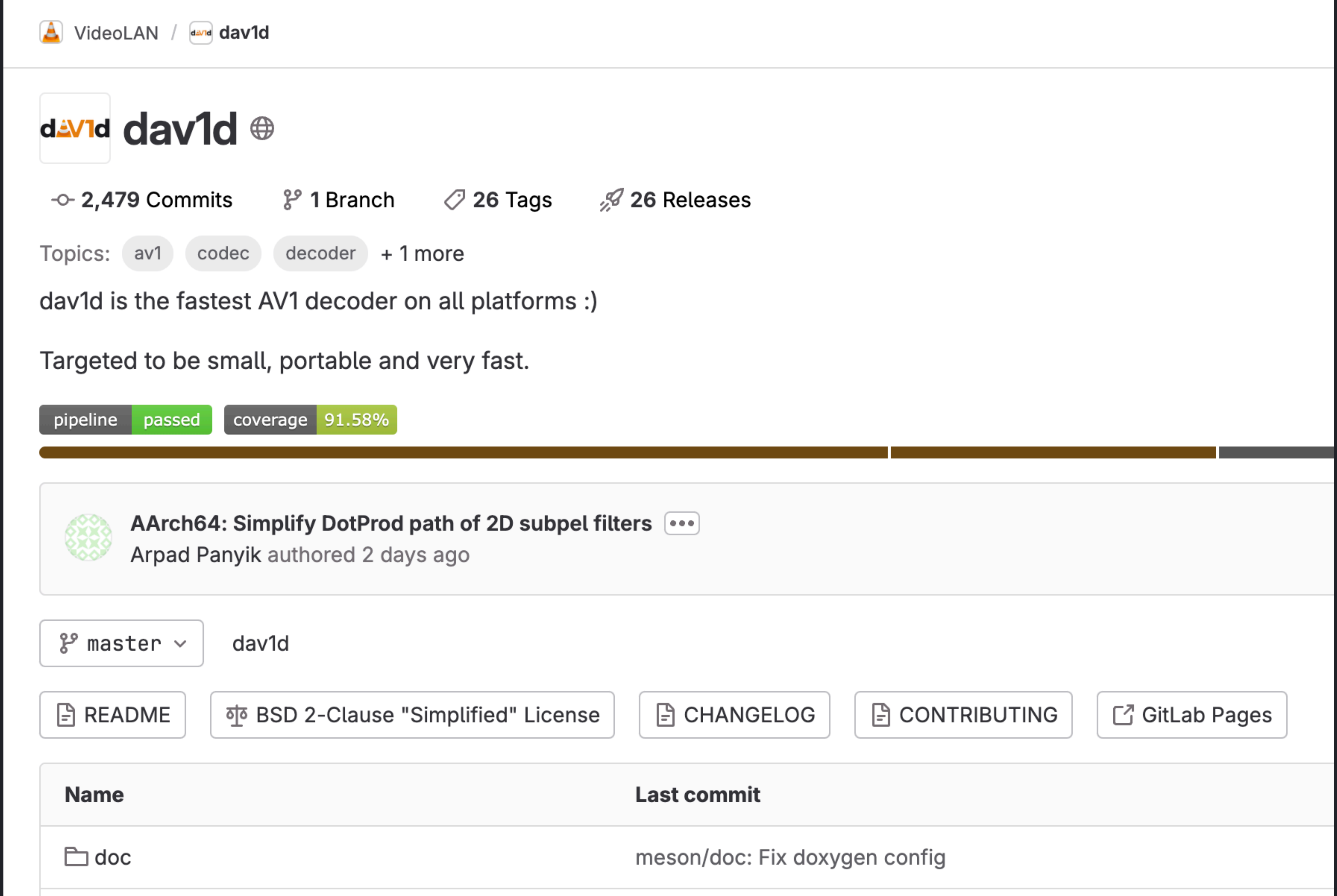
Reported by ifratric@google.com on Wed, Nov 22, 2023, 10:30 AM GMT+1 Project Member

This issue was found by Nick Galloway working with Google Project Zero

There is an integer overflow in dav1d when decoding an AV1 video with large width/height. The integer overflow may result in an out-of-bounds write. The issue was confirmed on the latest source code from https://code.videolan.org/videolan/dav1d

When multiple decoding threads are used, inside dav1d_decode_frame_init(), dav1d fills the tile_start_off array as

```
        f->frame_thread.tile_start_off[tile_idx++] = row_off + b_diff *
            f->frame_hdr->tiling.col_start_sb[tile_col] * f->sb_step * 4;
```


VideoLAN / dav1d

dav1d

2,479 Commits   1 Branch   26 Tags   26 Releases

Topics: av1 codec decoder + 1 more

dav1d is the fastest AV1 decoder on all platforms :)

Targeted to be small, portable and very fast.

pipeline passed   coverage 91.58%

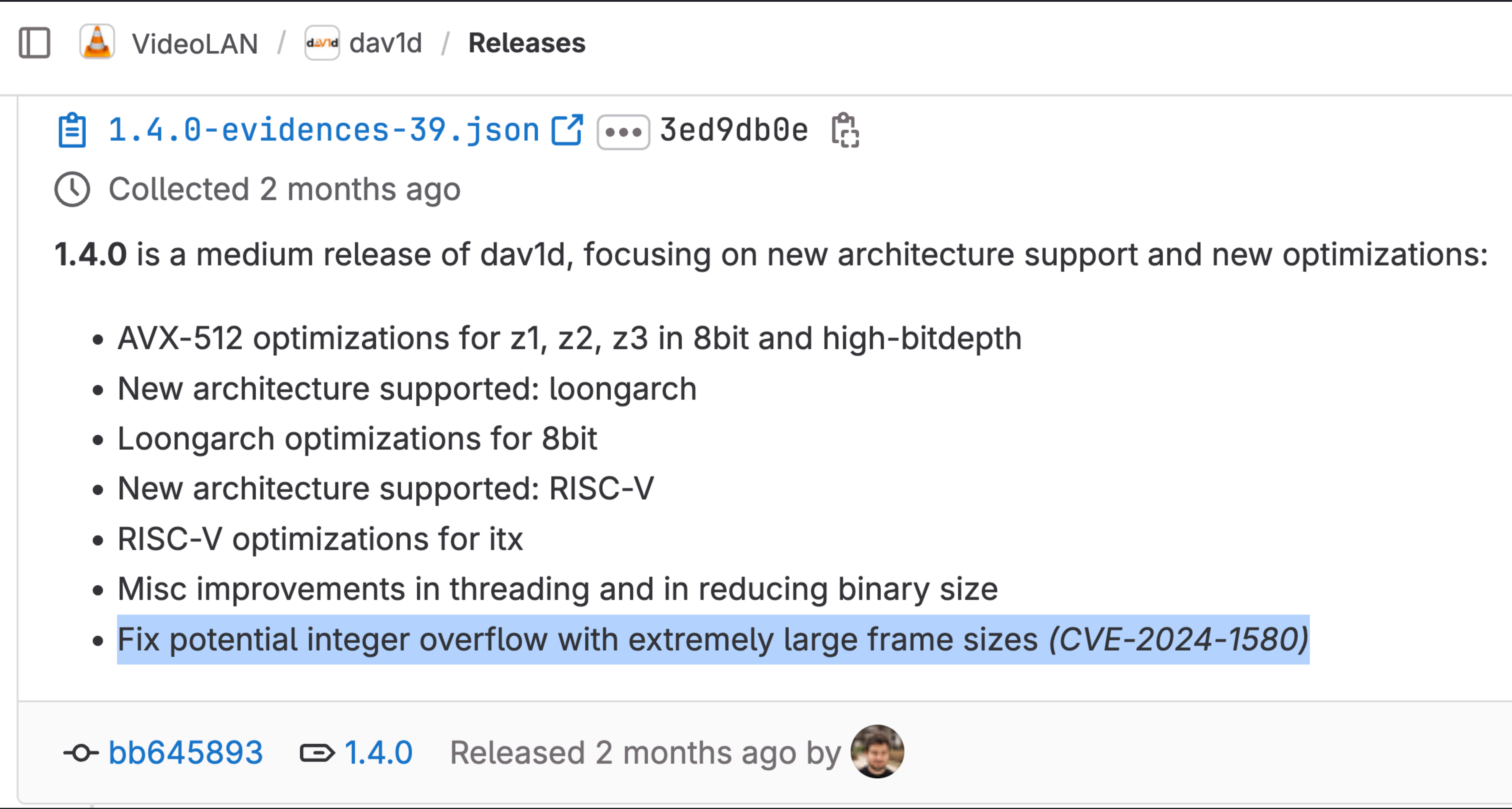
AArch64: Simplify DotProd path of 2D subpel filters

Arpad Panyik authored 2 days ago

master   dav1d

README   BSD 2-Clause "Simplified" License   CHANGELOG   CONTRIBUTING   GitLab Pages

Name	Last commit
doc	meson/doc: Fix doxygen config



VideoLAN / dav1d / **Releases**

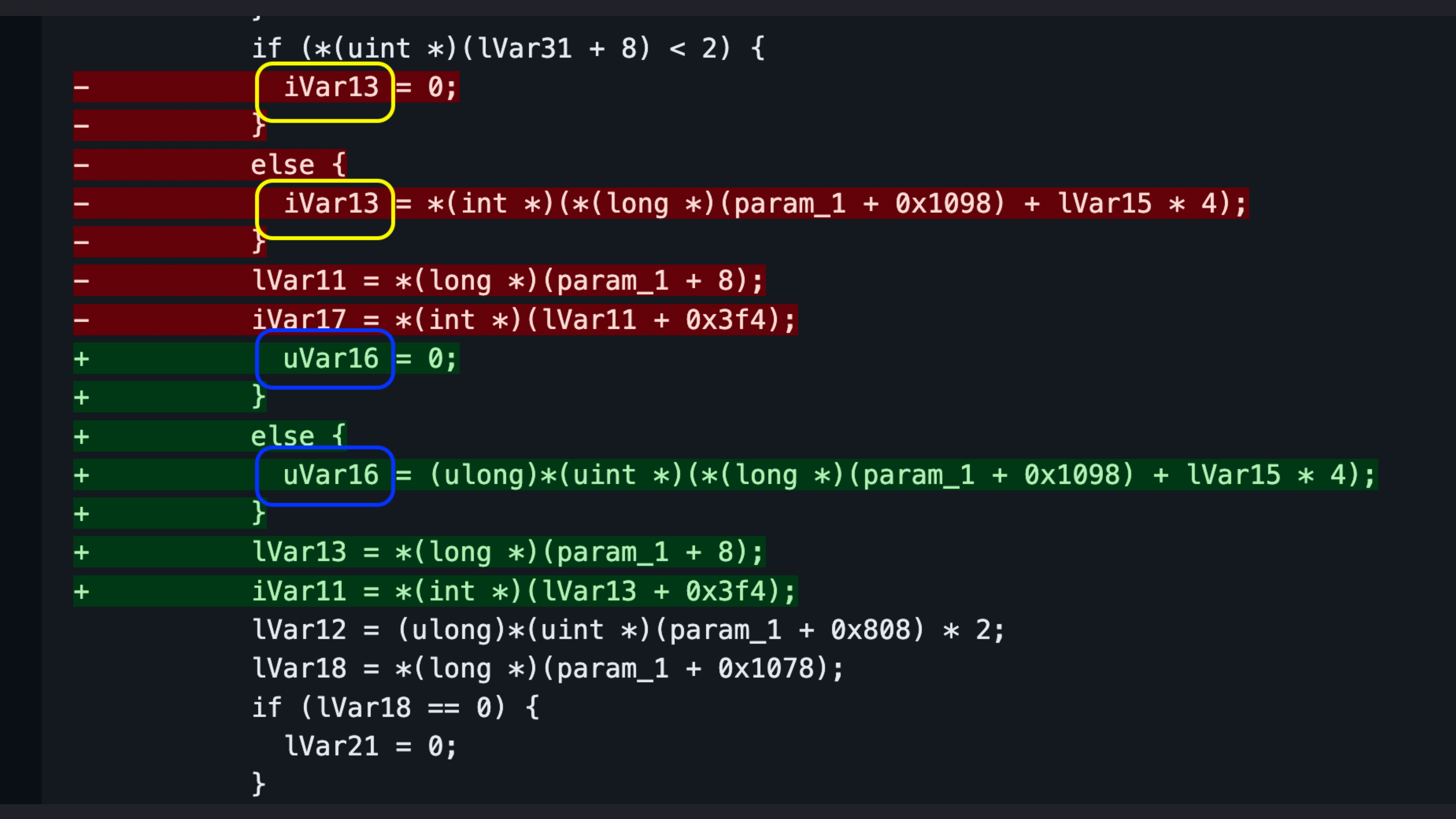
1.4.0-evidences-39.json 3ed9db0e

Collected 2 months ago

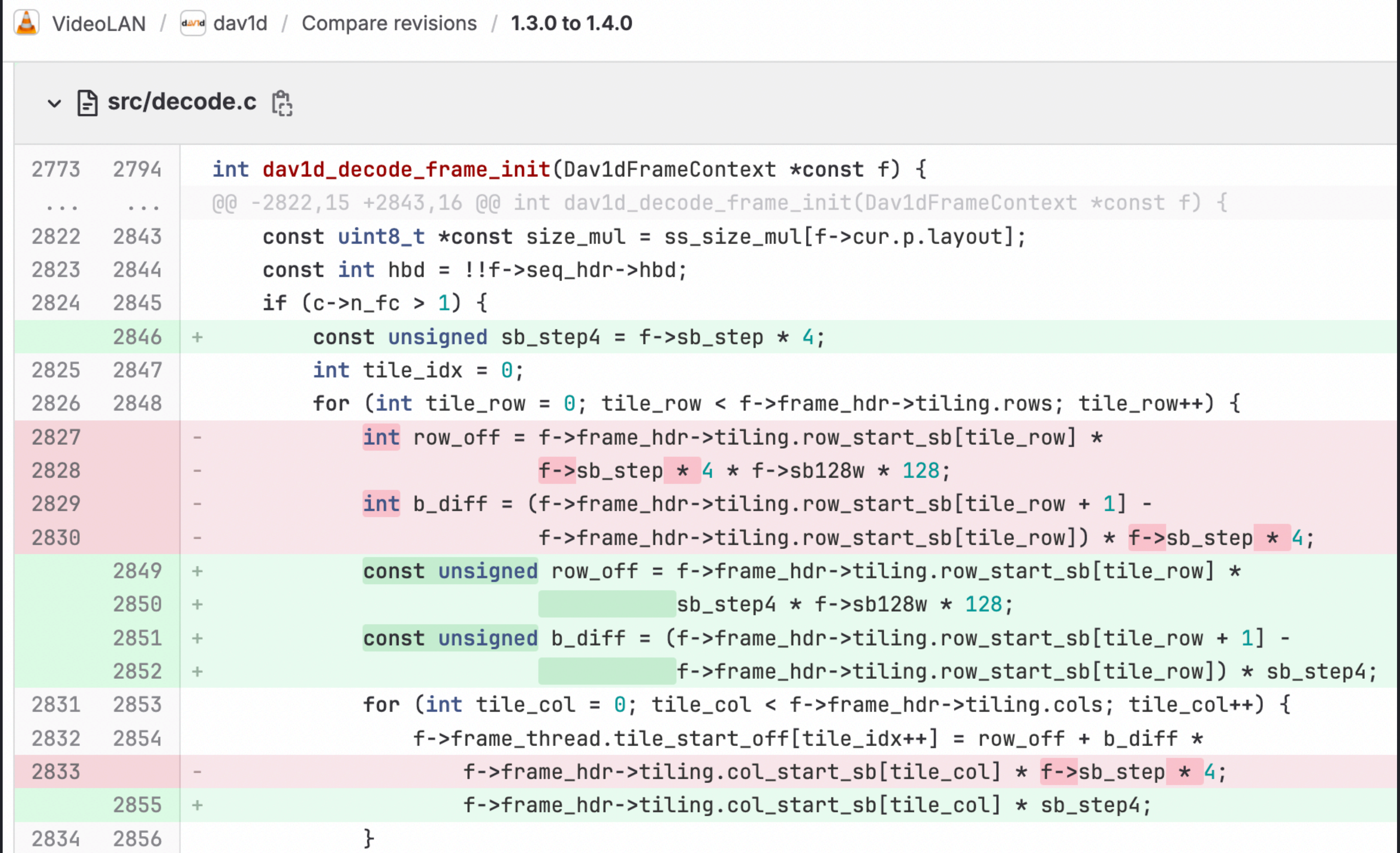
1.4.0 is a medium release of dav1d, focusing on new architecture support and new optimizations:

- AVX-512 optimizations for z1, z2, z3 in 8bit and high-bitdepth
- New architecture supported: loongarch
- Loongarch optimizations for 8bit
- New architecture supported: RISC-V
- RISC-V optimizations for itx
- Misc improvements in threading and in reducing binary size
- Fix potential integer overflow with extremely large frame sizes *(CVE-2024-1580)*

bb645893 1.4.0 Released 2 months ago by



```
        if (*(uint *)(lVar31 + 8) < 2) {
-         iVar13 = 0;
-       }
-       else {
-         iVar13 = *(int *)(*(long *)(param_1 + 0x1098) + lVar15 * 4);
-       }
-       lVar11 = *(long *)(param_1 + 8);
-       iVar17 = *(int *)(lVar11 + 0x3f4);
+         uVar16 = 0;
+       }
+       else {
+         uVar16 = (ulong)*(uint *)(*(long *)(param_1 + 0x1098) + lVar15 * 4);
+       }
+       lVar13 = *(long *)(param_1 + 8);
+       iVar11 = *(int *)(lVar13 + 0x3f4);
        lVar12 = (ulong)*(uint *)(param_1 + 0x808) * 2;
        lVar18 = *(long *)(param_1 + 0x1078);
        if (lVar18 == 0) {
          lVar21 = 0;
        }
```


VideoLAN / dav1d / Compare revisions / **1.3.0 to 1.4.0**

src/decode.c

```
 int dav1d_decode_frame_init(Dav1dFrameContext *const f) {
 @@ -2822,15 +2843,16 @@ int dav1d_decode_frame_init(Dav1dFrameContext *const f) {
     const uint8_t *const size_mul = ss_size_mul[f->cur.p.layout];
     const int hbd = !!f->seq_hdr->hbd;
     if (c->n_fc > 1) {
+        const unsigned sb_step4 = f->sb_step * 4;
         int tile_idx = 0;
         for (int tile_row = 0; tile_row < f->frame_hdr->tiling.rows; tile_row++) {
-            int row_off = f->frame_hdr->tiling.row_start_sb[tile_row] *
-                          f->sb_step * 4 * f->sb128w * 128;
-            int b_diff = (f->frame_hdr->tiling.row_start_sb[tile_row + 1] -
-                          f->frame_hdr->tiling.row_start_sb[tile_row]) * f->sb_step * 4;
+            const unsigned row_off = f->frame_hdr->tiling.row_start_sb[tile_row] *
+                                     sb_step4 * f->sb128w * 128;
+            const unsigned b_diff = (f->frame_hdr->tiling.row_start_sb[tile_row + 1] -
+                                     f->frame_hdr->tiling.row_start_sb[tile_row]) * sb_step4;
             for (int tile_col = 0; tile_col < f->frame_hdr->tiling.cols; tile_col++) {
                 f->frame_thread.tile_start_off[tile_idx++] = row_off + b_diff *
-                    f->frame_hdr->tiling.col_start_sb[tile_col] * f->sb_step * 4;
+                    f->frame_hdr->tiling.col_start_sb[tile_col] * sb_step4;
             }
```

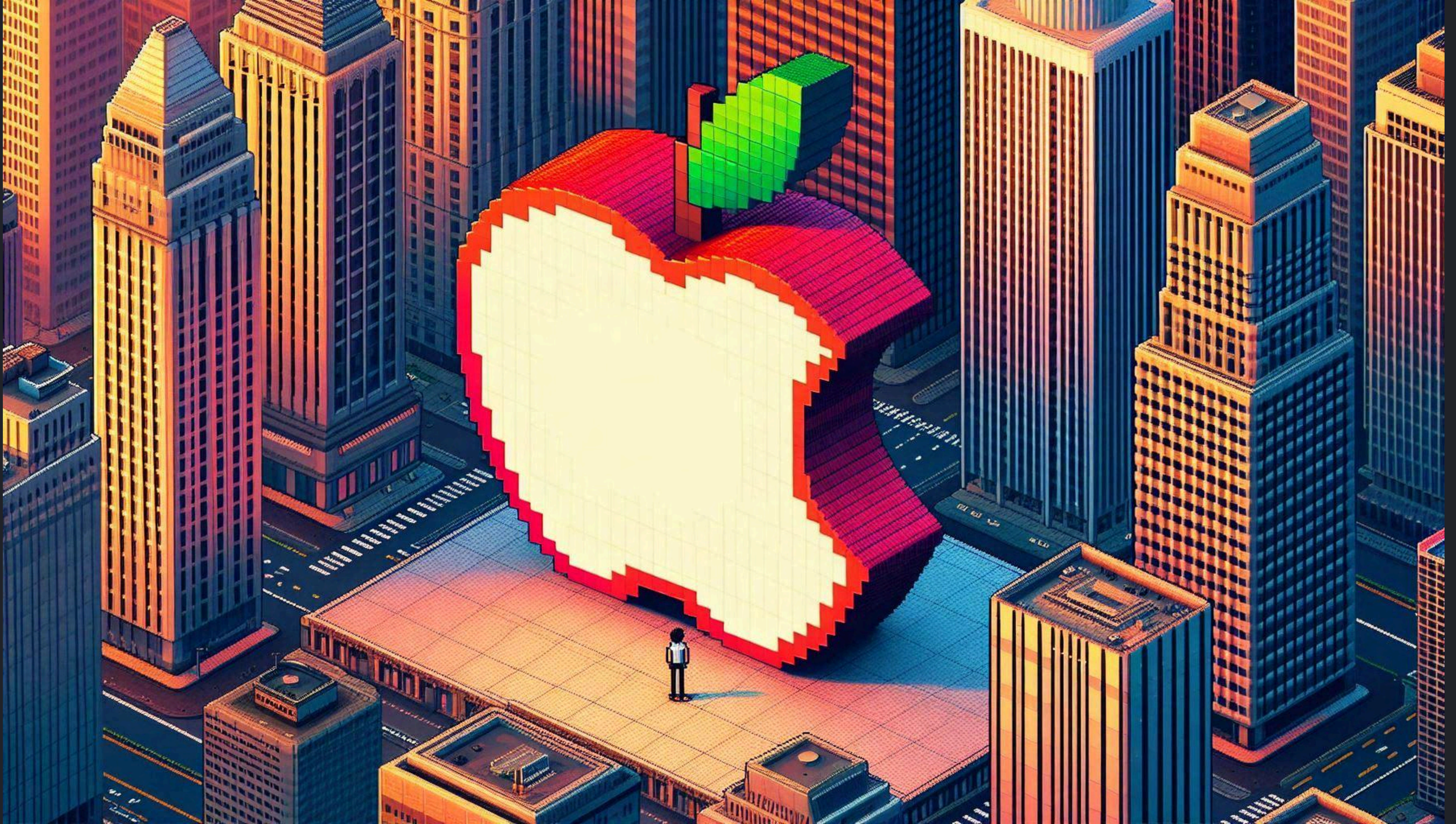
Links

- https://support.apple.com/en-us/HT214097
- https://bugs.chromium.org/p/project-zero/issues/detail?id=2502&q=label%3ACVE-2024-1580&can=1
- https://code.videolan.org/videolan/dav1d/-/releases
- https://blacktop.github.io/ipsw/
- https://gist.github.com/clearbluejar/0f221ceef88212eca69
- **AV1SW.videodecoder.21E219__iPhone15_4-17.4-AV1SW.videodecoder.21E236__iPhone15_4-17.4.1.ghid**
- https://diffpreview.github.io/?6b5bd4cd2eda17f4e34c924a
- https://code.videolan.org/videolan/dav1d/-/compare/1.3.0...from_project_id=198&straight=false

CVE-2024-1580 - int overflow

- Apple leverages 3rd party libraries in OS
- Issues found in 3rd party apply to Apple
- Can leverage open source library to confirm details / improve RE

Conclusion



Questions?

Patch Different on *OS

John McIntosh

contact: @clearbluejar

CLEARSEC LABS