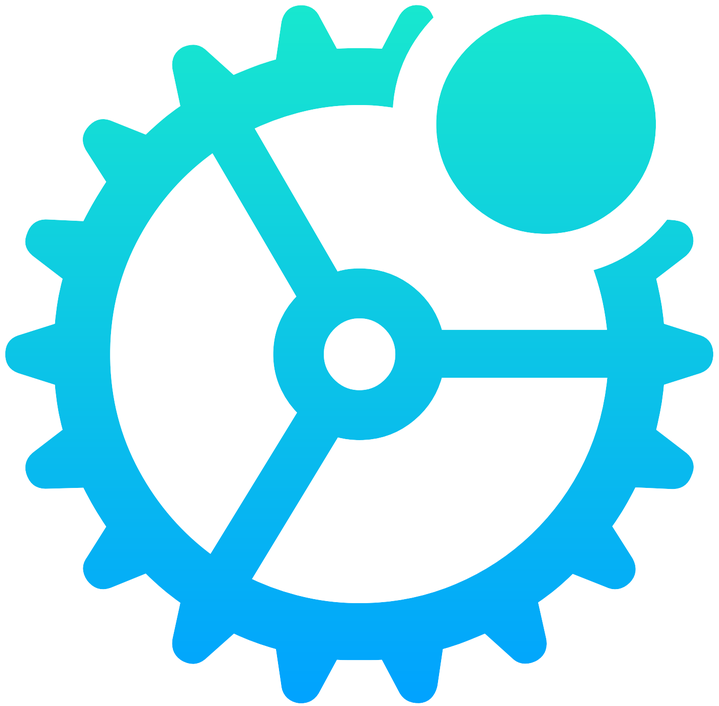
Apple's (not so) Rapid Security Response[1]

Objective by the Sea v7.0

Mykola Grymalyuk - December 6th, 2024

Bill of Materials

- What is a Rapid Security Response?
- What gaps does is it fill?
- What is the underlying technology?
- How do they work?
- Where do they fall short?
- Why are they no longer used?
- Where might they be secretly hiding today?



$ '/usr/bin/whoami'

```
> "Mykola Grymalyuk"
```

- Lead Security and Software Engineer at RIPEDA Consulting.
- Project lead of OpenCore Legacy Patcher.
- Breaking macOS internals on khronokernel.com.

What is a Rapid Security Response?

What is a Rapid Security Response?

A security update that comes out faster than other security updates...

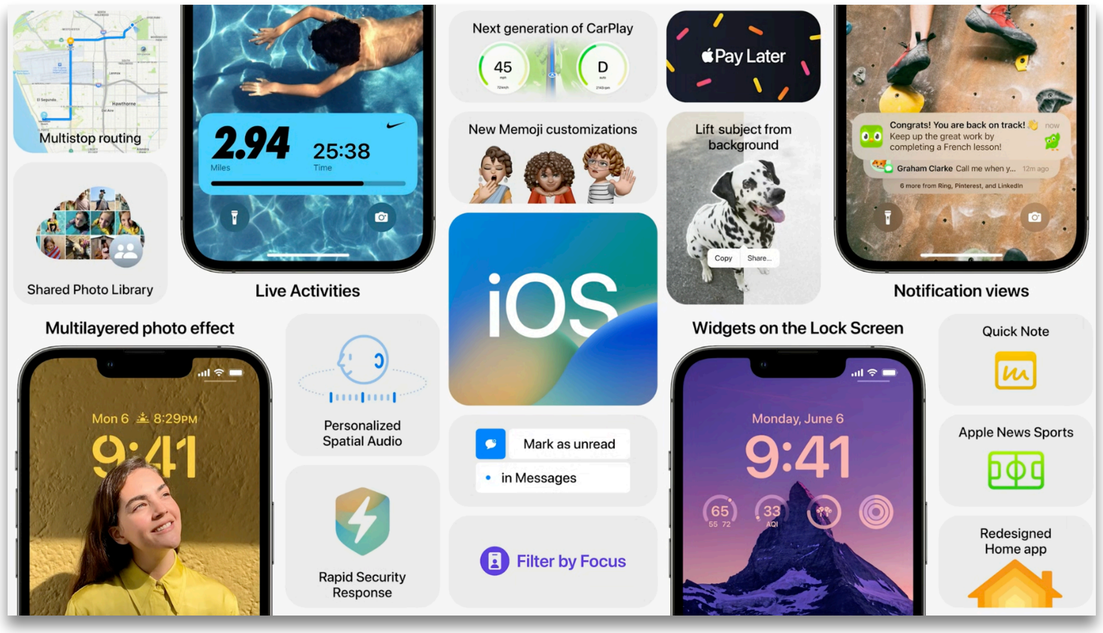
What is a Rapid Security Response?

A security update that comes out faster than other security updates...

Rapid Security Response

WWDC 2022

macOS 13, Ventura

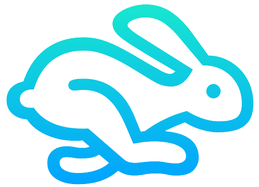


iOS 16

What gaps does it fill?

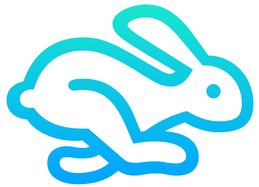
What gaps does it fill?

What gaps does it fill?



1. Quickly handle zero days

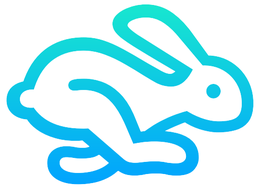
What gaps does it fill?



1. Quickly handle zero days

2. Reduce update times and reboots required

What gaps does it fill?



1. Quickly handle zero days

2. Reduce update times and reboots required

3. Ability to revert updates

RSR Versioning: X.Y.Z (a,b,c)

RSR Versioning: X.Y.Z (a,b,c)

macOS 15.1.1

iOS 18.1.1

RSR Versioning: X.Y.Z (a,b,c)

macOS 15.1.1 (a)

iOS 18.1.1 (a)

RSR Versioning: X.Y.Z (a,b,c)

macOS 15.1.1 (a)

iOS 18.1.1 (a)

Host: /System/Library/CoreServices/SystemVersion.plist

RSR: /System/Volumes/Preboot/Cryptexes/OS/System/Library/CoreServices/SystemVersion.plist

RSR Versioning: X.Y.Z (a,b,c)

macOS 15.1.1 (a)

iOS 18.1.1 (a)

Host: /System/Library/CoreServices/SystemVersion.plist	24B91
RSR: /System/Volumes/Preboot/Cryptexes/OS/System/Library/CoreServices/SystemVersion.plist	24B880910b

What is the underlying technology?

What is the underlying technology?

Let’s roll back to iOS 14

Apple Security Research Device Program

Apple Security Research Device Program

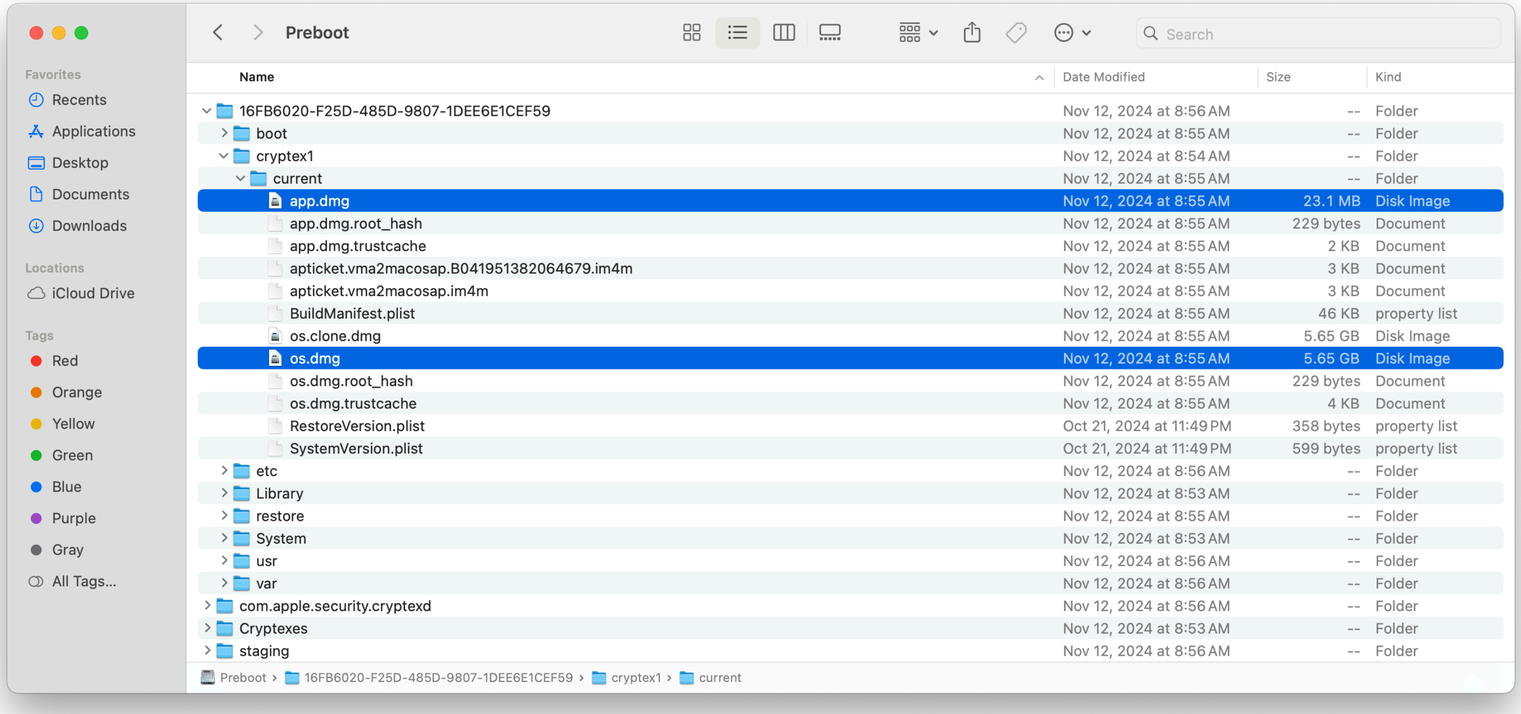
- Released in early 2021
- Launched with iOS 14 and an Apple provided iPhone 11 (SRD)
- Uses a new “Cryptex” system for research and development

🤔

How do RSRs/Cryptexes work?

How do RSRs/Cryptexes work?

Preboot Volume



/System/Volumes/Preboot/{OS UUID}/cryptex1/current

How do RSRs/Cryptexes work?

OS.dmg

Frameworks & dyld_shared_cache

Grafted to /System/Cryptexes/OS

App.dmg

Safari & Co.

Grafted to /System/Cryptexes/App

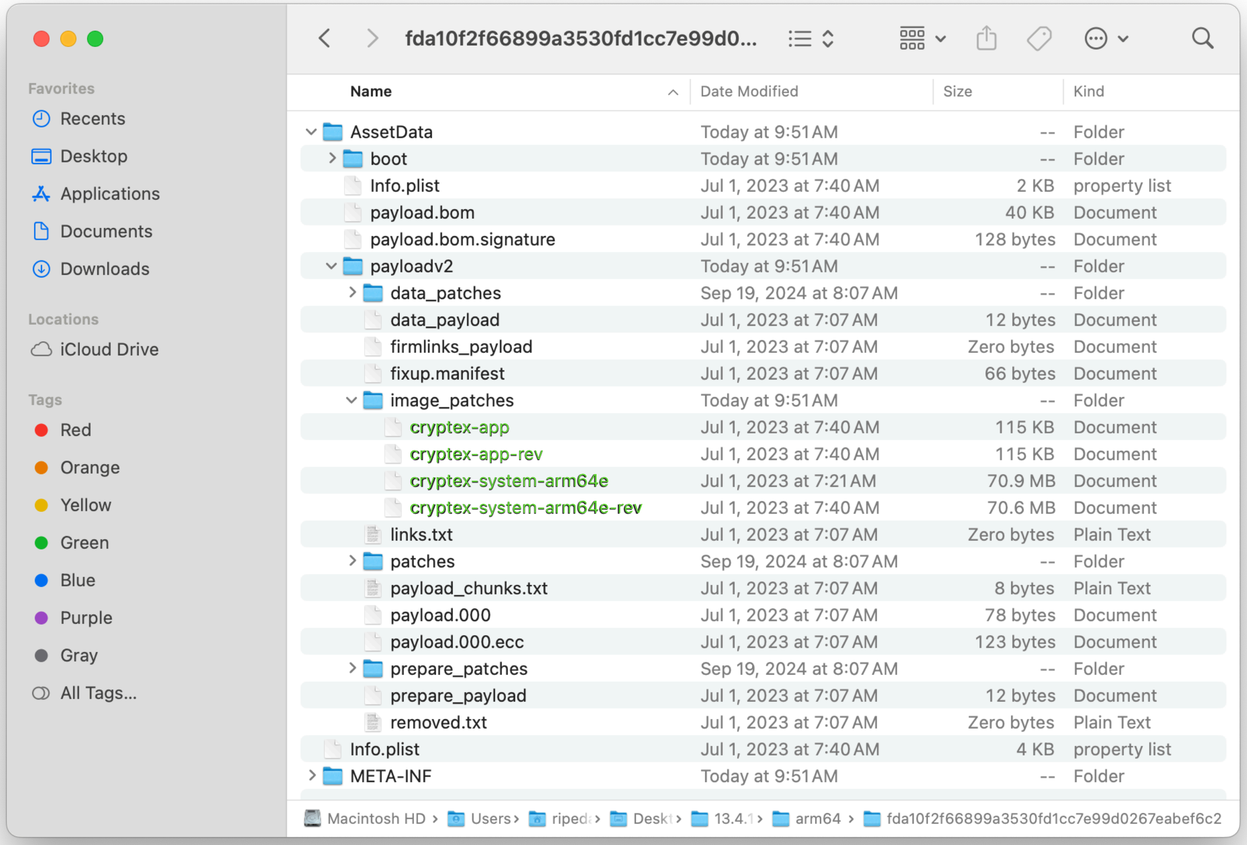
Cryptexes

Mounted via APFS.kext

Validated by root_hash & includes trushcache

RSRs applied via RIDIFF10

How do RSRs/Cryptexes work?



cryptex-app

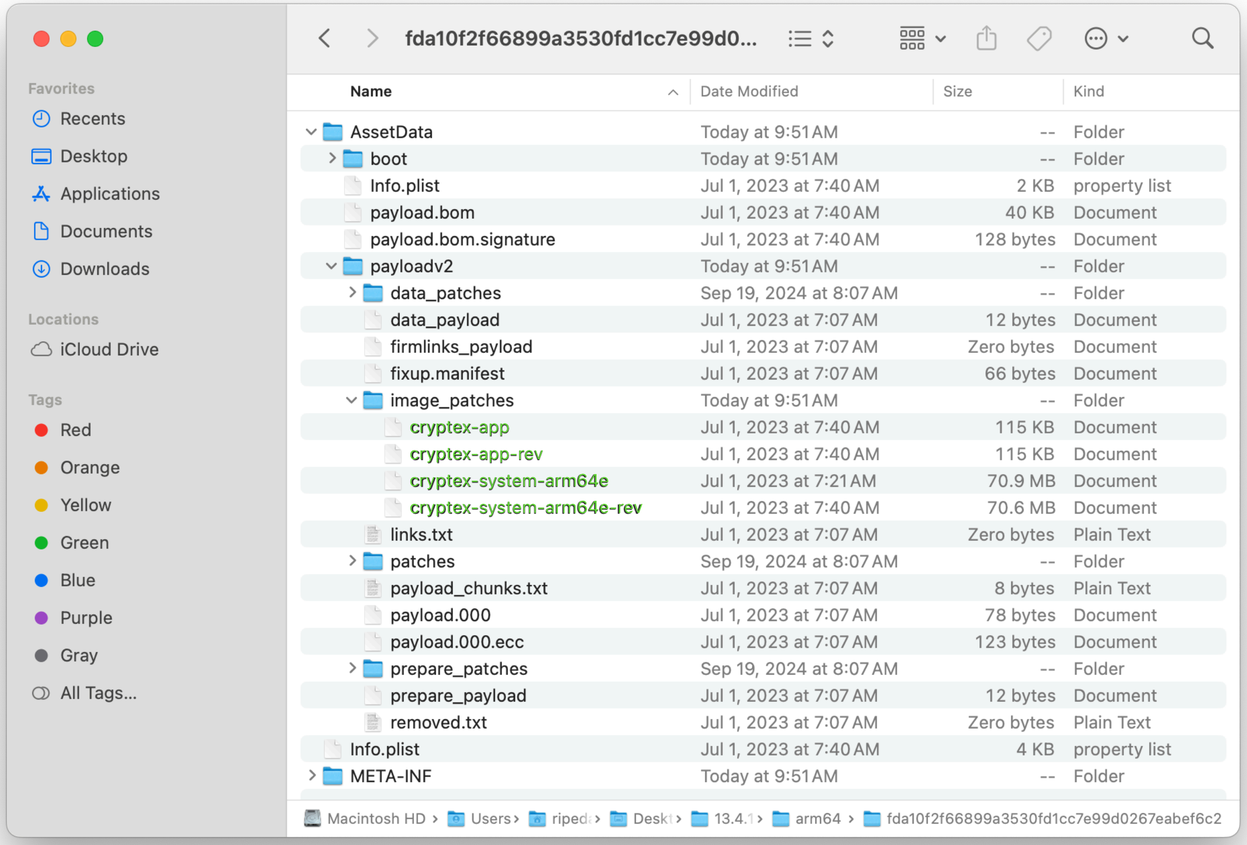
cryptex-app-rev

cryptex-system-{arch}

cryptex-system-{arch}-rev

https://updates.cdn-apple.com/2023SummerFCS/patches/042-11155/2C86E0CF-DB4F-4C20-8925-2F3F54F61A11/com_apple_MobileAsset_MacSplatSoftwareUpdate/fda10f2f66899a3530fd1cc7e99d0267eabef6c2.zip

How do RSRs/Cryptexes work?



cryptex-app

cryptex-app-rev

cryptex-system-{arch}

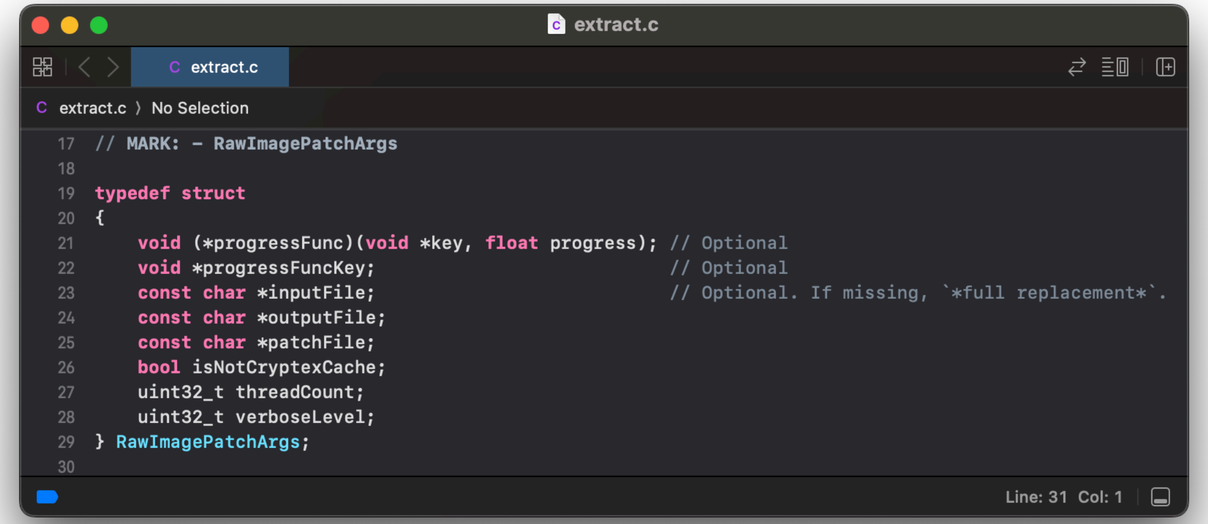
cryptex-system-{arch}-rev

RSRs can’t be applied on top each other

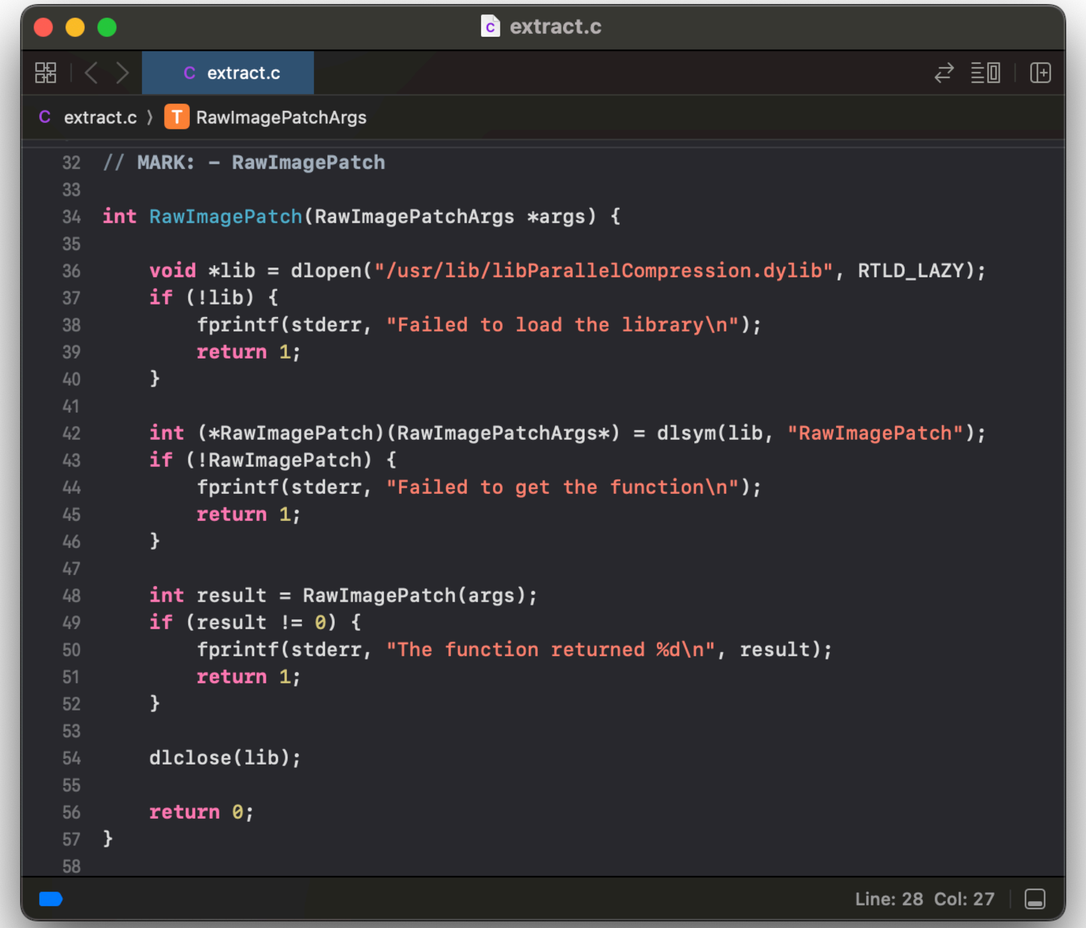
https://updates.cdn-apple.com/2023SummerFCS/patches/042-11155/2C86E0CF-DB4F-4C20-8925-2F3F54F61A11/com_apple_MobileAsset_MacSplatSoftwareUpdate/fda10f2f66899a3530fd1cc7e99d0267eabef6c2.zip

How do RSRs/Cryptexes work?

/usr/lib/libParallelCompression.dylib's RawImagePatch()



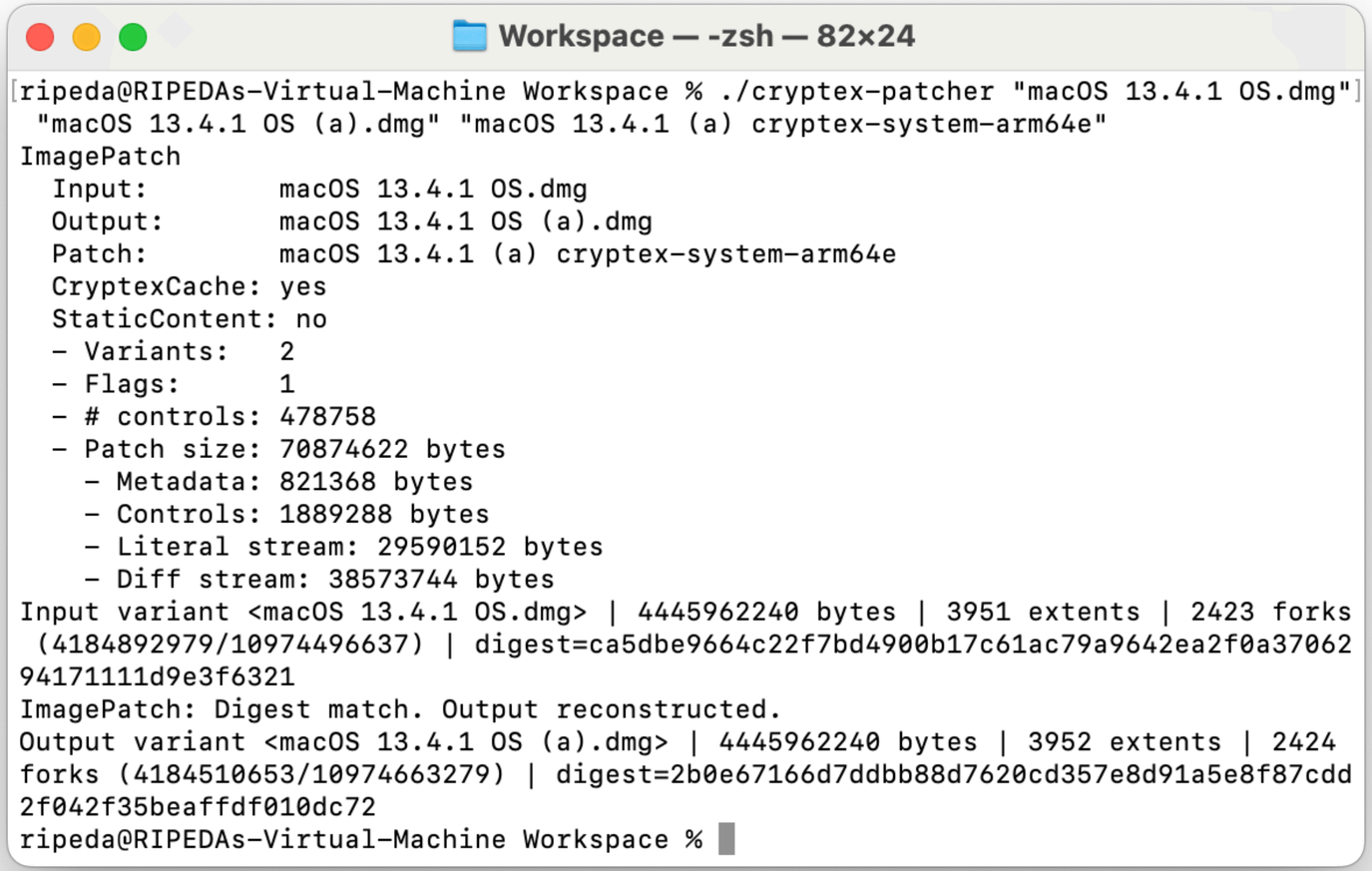
Credit to DhinakG



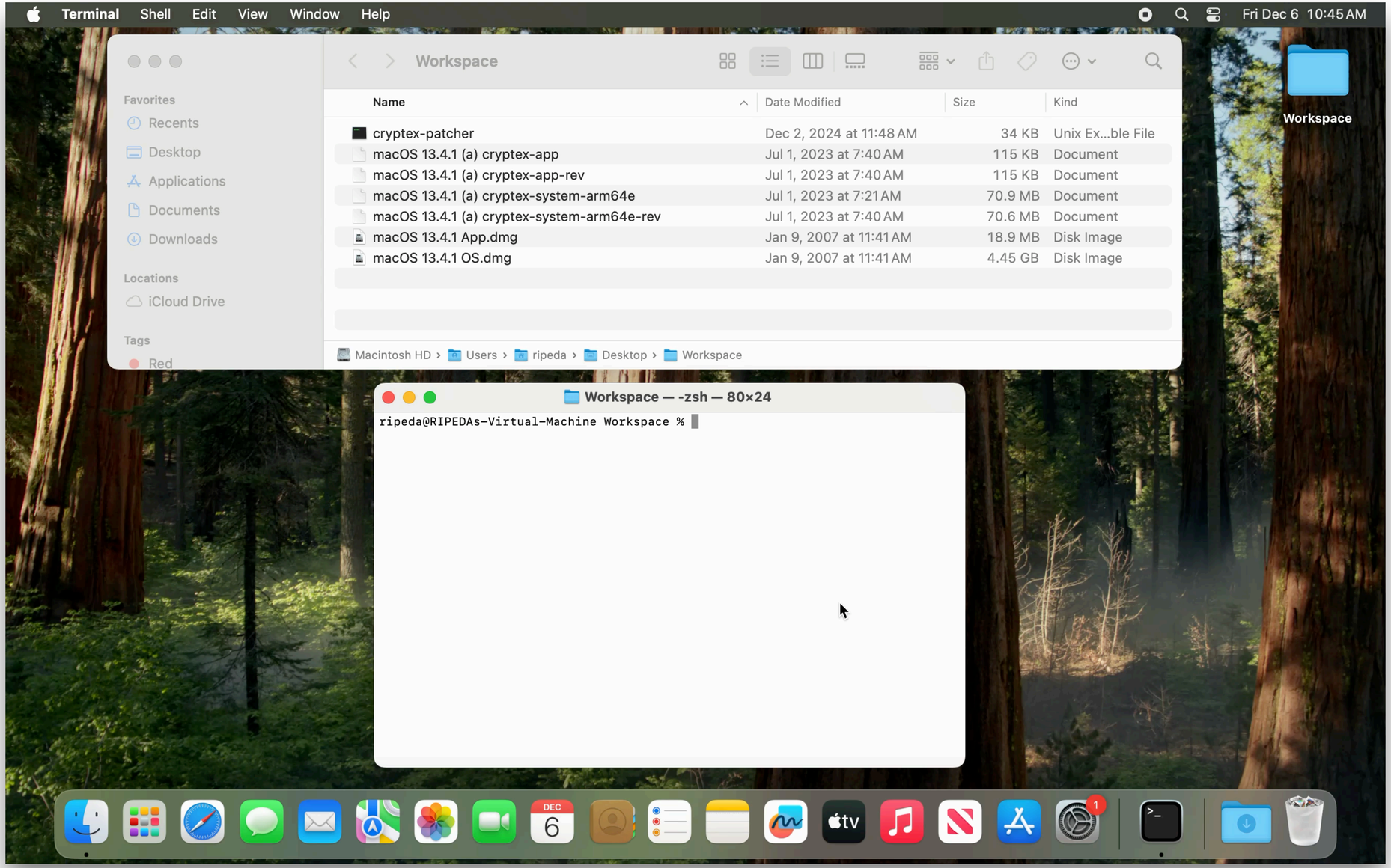
Credit to ASentientBot

How do RSRs/Cryptexes work?

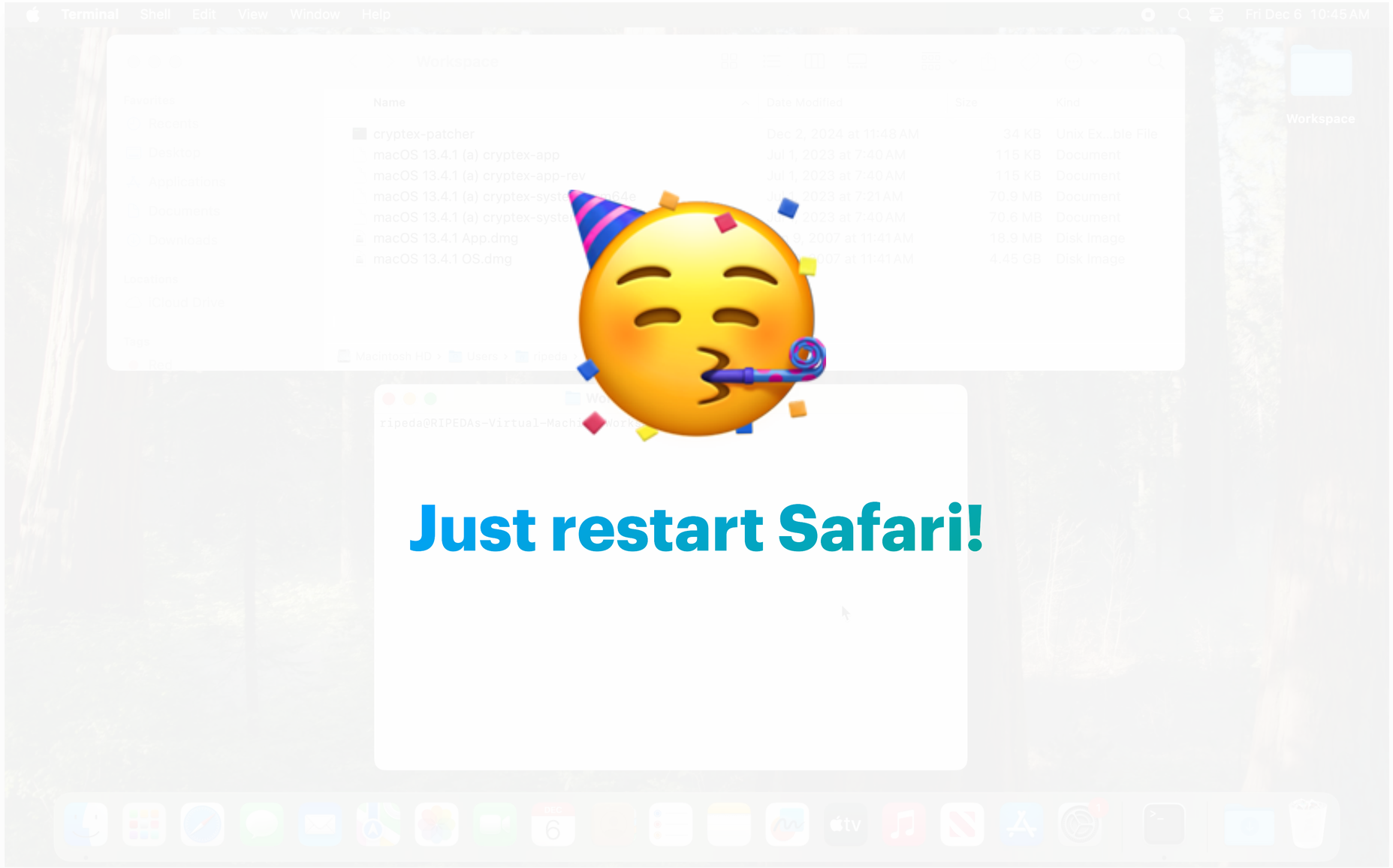
./cryptex-patcher <cryptex>.dmg <output>.dmg <rsr>



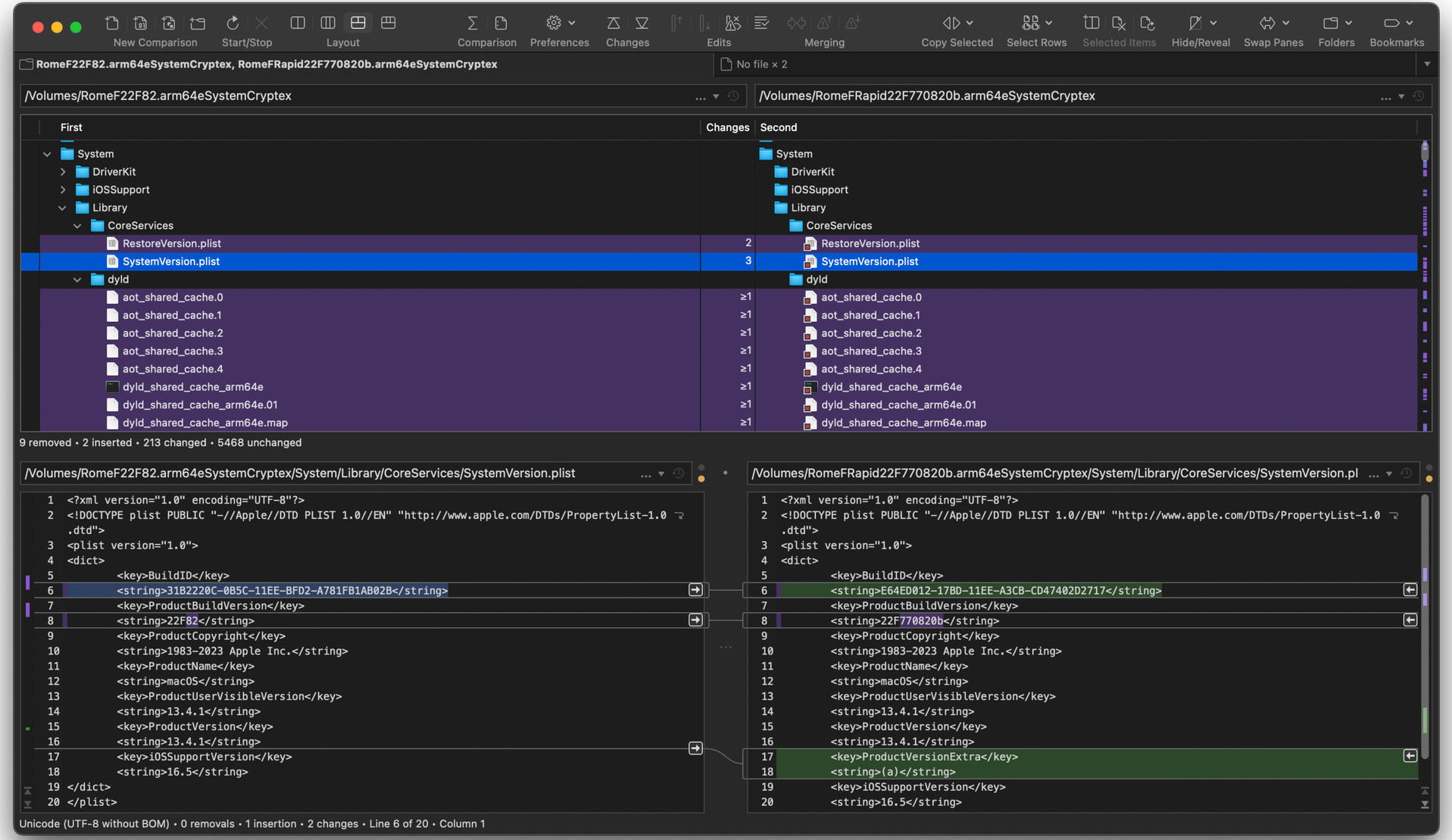
How do RSRs/Cryptexes work?



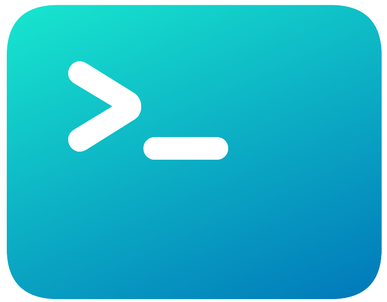
How do RSRs/Cryptexes work?



How do RSRs/Cryptexes work?



dyld_shared_cache system

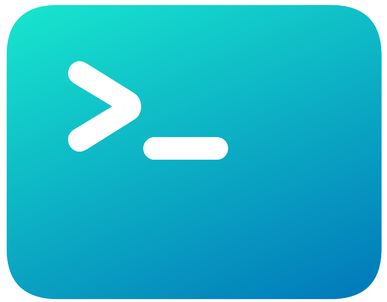


Test bin

Demo.framework

dyld

dyld_shared_cache system



Test bin

Demo.framework

dyld

System volume stand alone ?

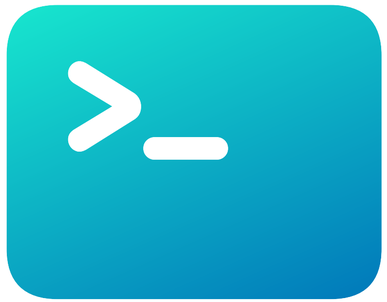
Cryptex stand alone?

System volume dyld_shared_cache ?

Cryptex dyld_shared_cache ?

Relative path?

dyld_shared_cache system



Test bin

Demo.framework

dyld

System volume stand alone ?

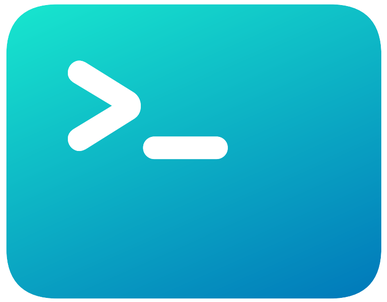
Cryptex stand alone!

System volume dyld_shared_cache ?

Cryptex dyld_shared_cache ?

Relative path?

dyld_shared_cache system



Test bin

/System/Cryptex/OS/System/Library
/Frameworks/Demo.framework

Lets follow the macOS Boot Process

macOS Boot Process

Intel Macs

1. boot.efi and XNU init

macOS Boot Process

Intel Macs

1. boot.efi and XNU init

2. Kernel Extensions load

macOS Boot Process

Intel Macs

1. boot.efi and XNU init

2. Kernel Extensions load

3. apfs.kext starts

macOS Boot Process

Intel Macs

1. boot.efi and XNU init

2. Kernel Extensions load

3. apfs.kext starts

4. Userspace starts

macOS Boot Process

Intel Macs

1. boot.efi and XNU init

2. Kernel Extensions load

3. apfs.kext starts

4. Userspace starts

5. /usr/lib/dyld starts

macOS Boot Process

Intel Macs

1. boot.efi and XNU init

2. Kernel Extensions load

3. apfs.kext starts

4. Userspace starts

5. /usr/lib/dyld starts

6. _apfs_graft executed

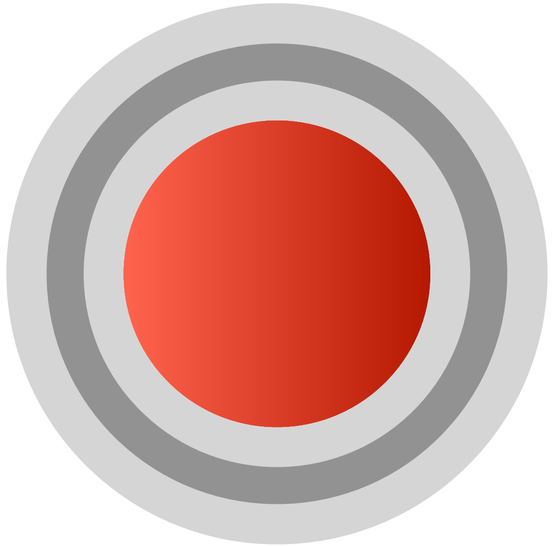
Where do RSRs fall short?

Where do RSRs fall short?

Kernel Space Updates

What is kernel space?

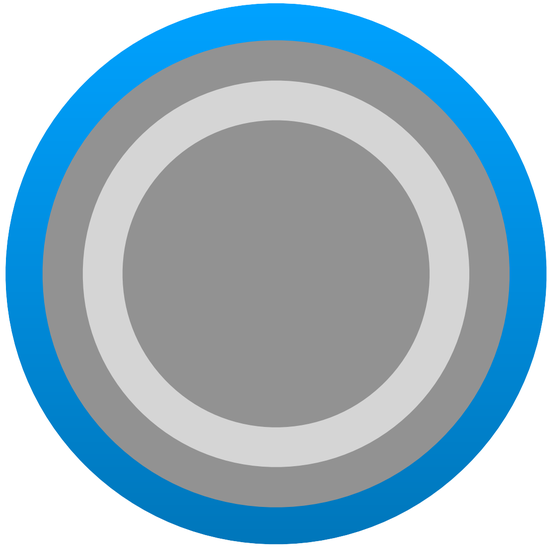
What is kernel space?



Ring 0

Kernel Space

- Closer to hardware
- Kernel and Kernel Extensions
- Higher danger when exploited



Ring 3

User Space

- Lower privileges
- Apps, CLIs, etc
- Goes through kernel space

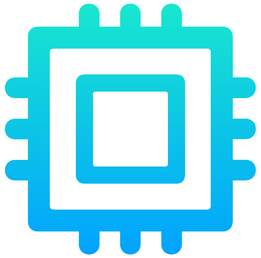
Least Privileged

Most Privileged

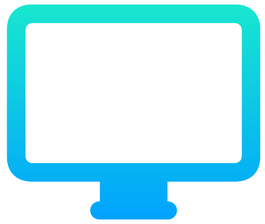
Apple's 3 stage kernel caching system

Apple's 3 stage kernel caching system

Introduced with macOS 11, Big Sur, at WWDC 2020



Boot Kernel Collection

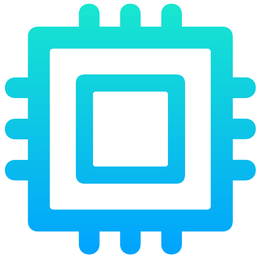


System Kernel Collection

Auxiliary Kernel Collection

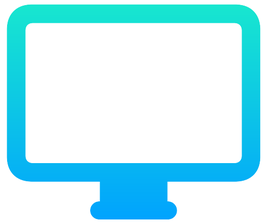
Apple's 3 stage kernel caching system

Introduced with macOS 11, Big Sur, at WWDC 2020



Boot Kernel Collection

- Critical to booting
- Kernel, ie. XNU
- Core Drivers (ex. NVMe)
- Preboot volume

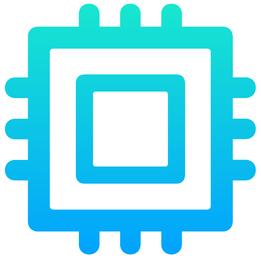


System Kernel Collection

Auxiliary Kernel Collection

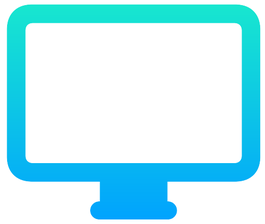
Apple's 3 stage kernel caching system

Introduced with macOS 11, Big Sur, at WWDC 2020



Boot Kernel Collection

- Critical to booting
- Kernel, ie. XNU
- Core Drivers (ex. NVMe)
- Preboot volume



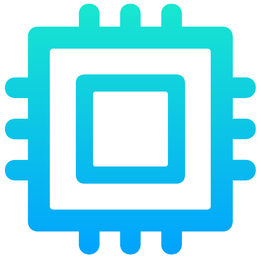
System Kernel Collection

- Pageable
- ex. Graphics & Audio
- N/A on Apple Silicon
- System volume

Auxiliary Kernel Collection

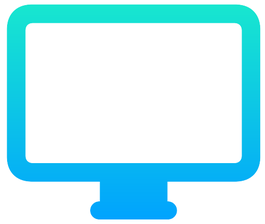
Apple's 3 stage kernel caching system

Introduced with macOS 11, Big Sur, at WWDC 2020



Boot Kernel Collection

- Critical to booting
- Kernel, ie. XNU
- Core Drivers (ex. NVMe)
- Preboot volume



System Kernel Collection

- Pageable
- ex. Graphics & Audio
- N/A on Apple Silicon
- System volume

Auxiliary Kernel Collection

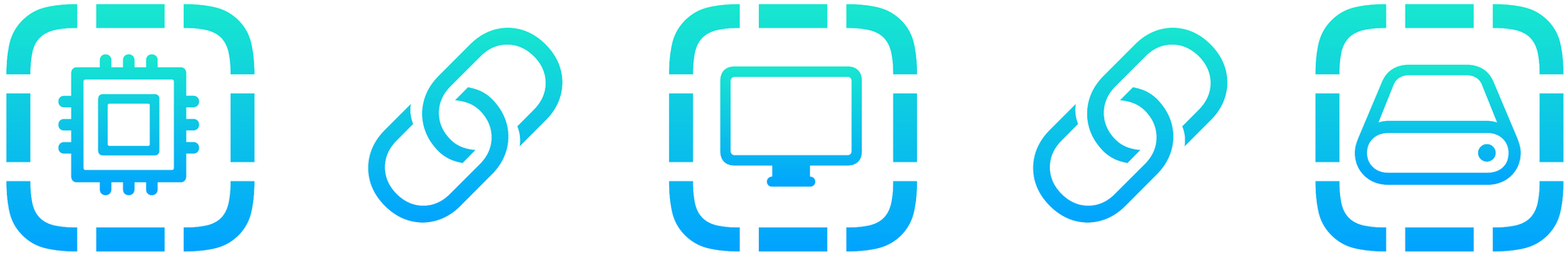
- 3rd party drivers
- ex. RAID Controllers
- Bridge as DriverKit developed
- Data volume

System Kernel Collection

- System volume

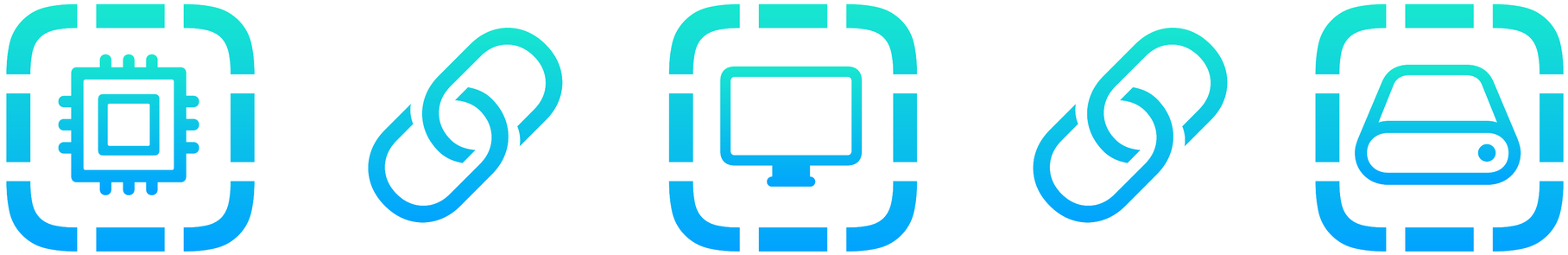
Kernel Collection UUIDs and APFS Snapshots

Kernel Collection UUIDs and APFS Snapshots



Linked at build time

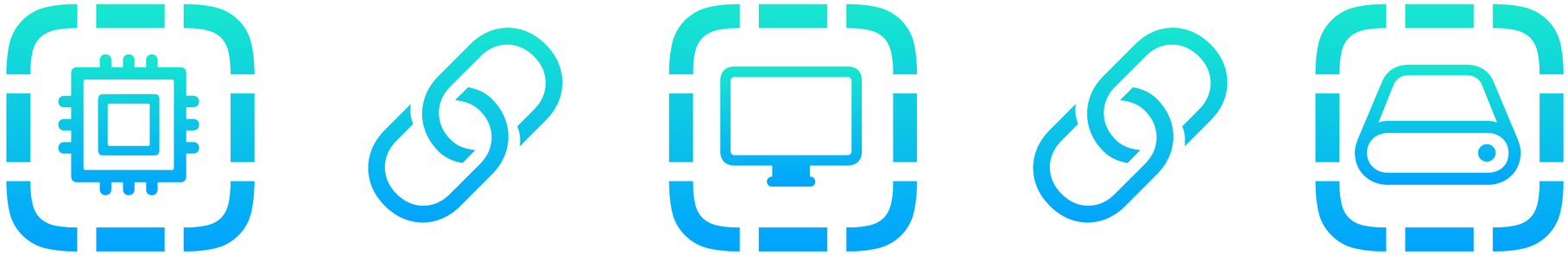
Kernel Collection UUIDs and APFS Snapshots



Linked at build time

Overlay/hot load kernel extensions from an RSR?

Kernel Collection UUIDs and APFS Snapshots



Killed with the Kernel Collections Architecture in Big Sur...

~~Overlay/hot load kernel extensions~~ from an RSR?

Kernel Collection UUIDs and APFS Snapshots

So why are snapshots an issue?

Killed with the Kernel Collections Architecture in Big Sur...

~~Overlay/hot load kernel extensions from an RSR?~~

Kernel Collection UUIDs and APFS Snapshots

So why are snapshots an issue?

Multiple reboots to apply...

Killed with the Kernel Collections Architecture in Big Sur...

Overlay/hot load kernel extensions from an RSR?

Why are RSRs no longer used?

Why are RSRs no longer used?

macOS 13.4.1 (a)…(b)…(c)…

iOS 16.5.1 (a)…(b)…(c)…

macOS 13.4.1 (& iOS 16.5.1) Timeline

macOS 13.4.1 (& iOS 16.5.1) Timeline

macOS 13.4.1 (& iOS 16.5.1) release

June 21, 2023

macOS 13.4.1 (& iOS 16.5.1) Timeline

macOS 13.4.1 (& iOS 16.5.1) Timeline

1. macOS 13.4.1 release

CVE-2023-37450 discovered

June - July ~, 2023

macOS 13.4.1 (& iOS 16.5.1) Timeline

macOS 13.4.1 (& iOS 16.5.1) Timeline

1. macOS 13.4.1 release
2. CVE-2023-37450 discovered

13.4.1 (a) released

July 10, 2023

macOS 13.4.1 (& iOS 16.5.1) Timeline

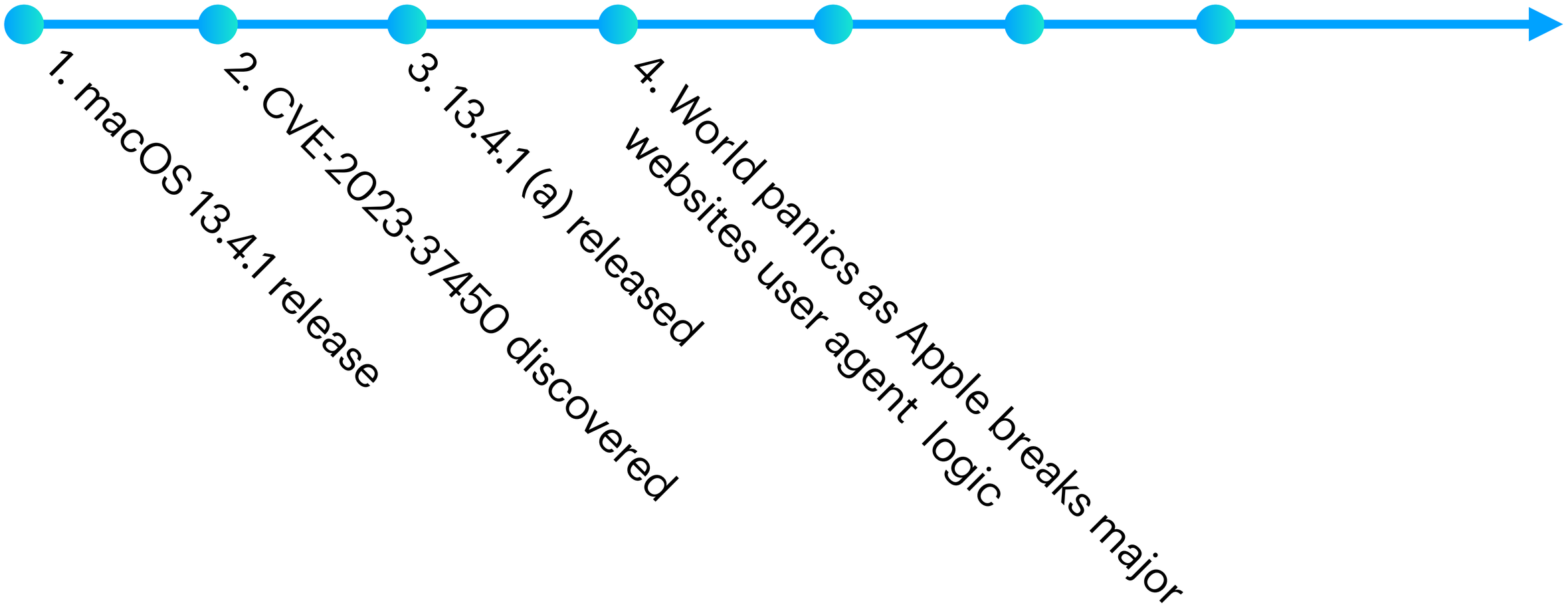
macOS 13.4.1 (& iOS 16.5.1) Timeline

1. macOS 13.4.1 release
2. CVE-2023-37450 discovered
3. 13.4.1(a) released

World panics as Apple breaks major websites user agent logic.

July 10, 2023

macOS 13.4.1 (& iOS 16.5.1) Timeline



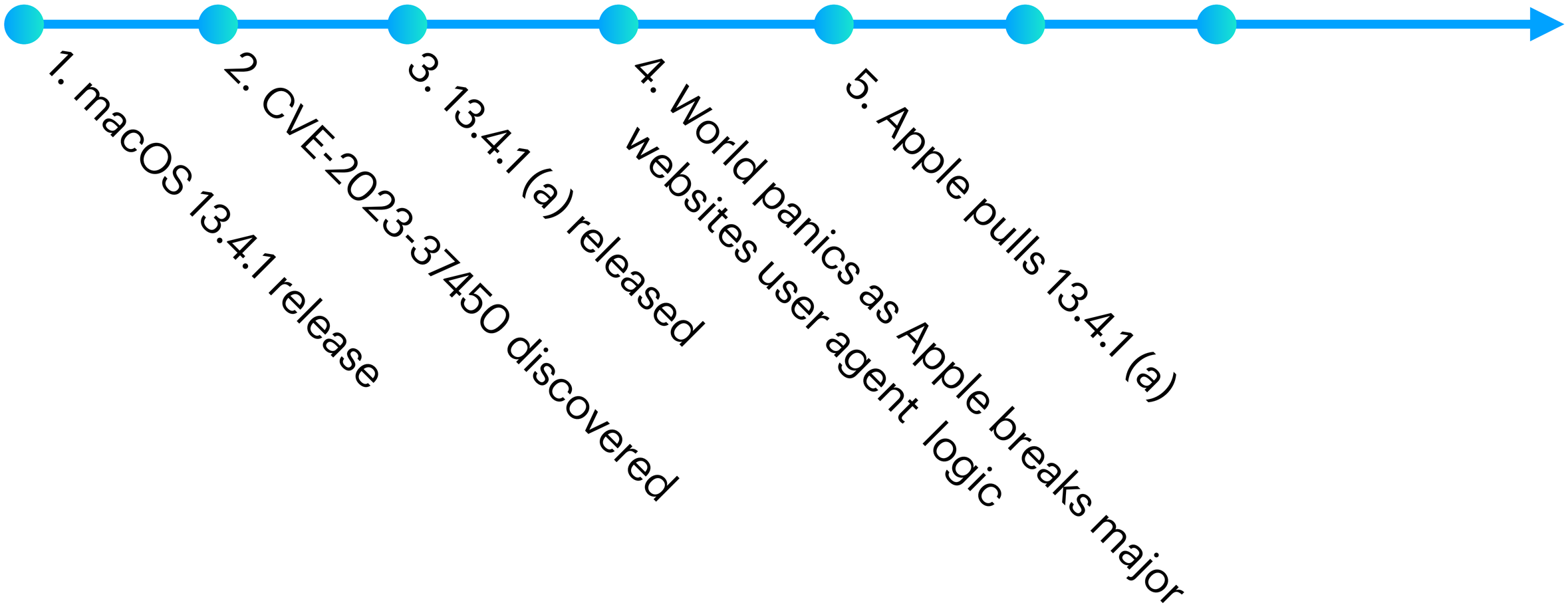
macOS 13.4.1 (& iOS 16.5.1) Timeline

1. macOS 13.4.1 release
2. CVE-2023-37450 discovered
3. 13.4.1 (a) released
4. Major parties as Apple breaks major websites user agent logic

Apple pulls 13.4.1 (a) and recommends users to uninstall the update if affected

July 10, 2023

macOS 13.4.1 (& iOS 16.5.1) Timeline



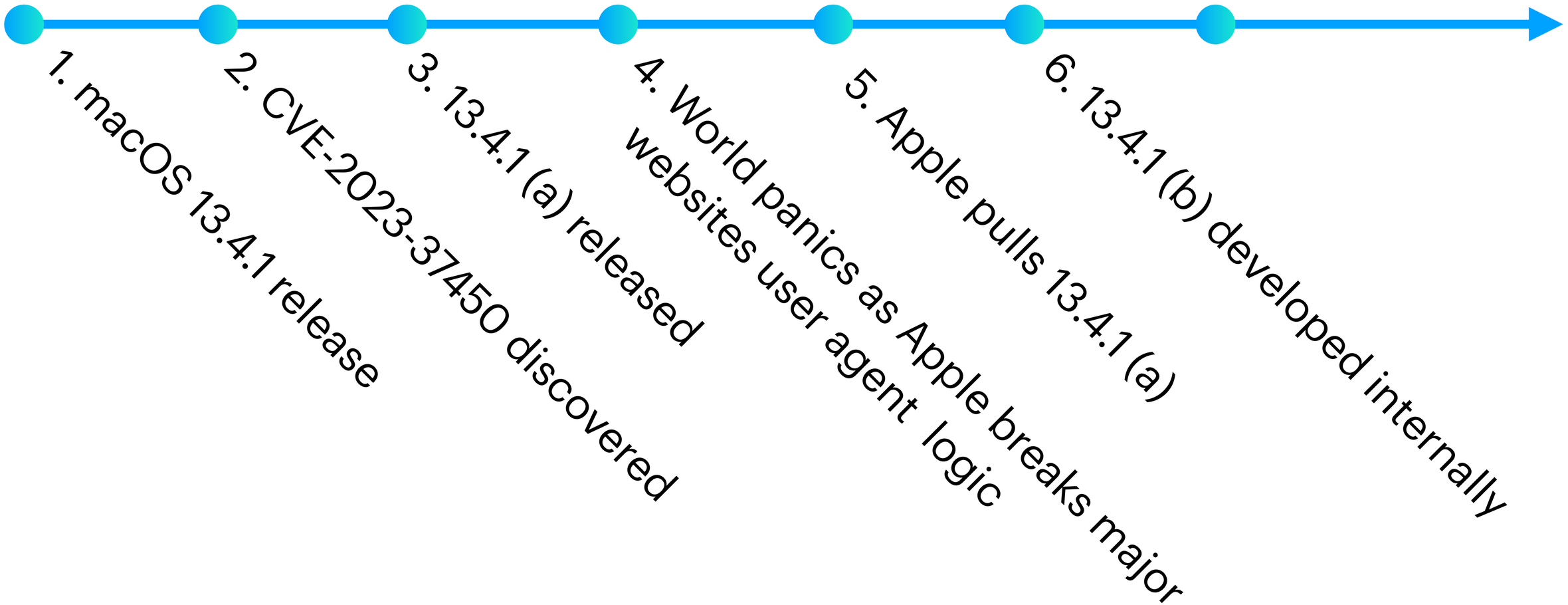
macOS 13.4.1 (& iOS 16.5.1) Timeline

1. macOS 13.4.1 release
2. CVE-2023-37450 discovered
3. 13.4.1 (a) released
4. WebKit changes user agent logic
5. Apple breaks major websites as Apple pulls 13.4.1 (a)

13.4.1 (b) developed internally

July 10-11~, 2023

macOS 13.4.1 (& iOS 16.5.1) Timeline



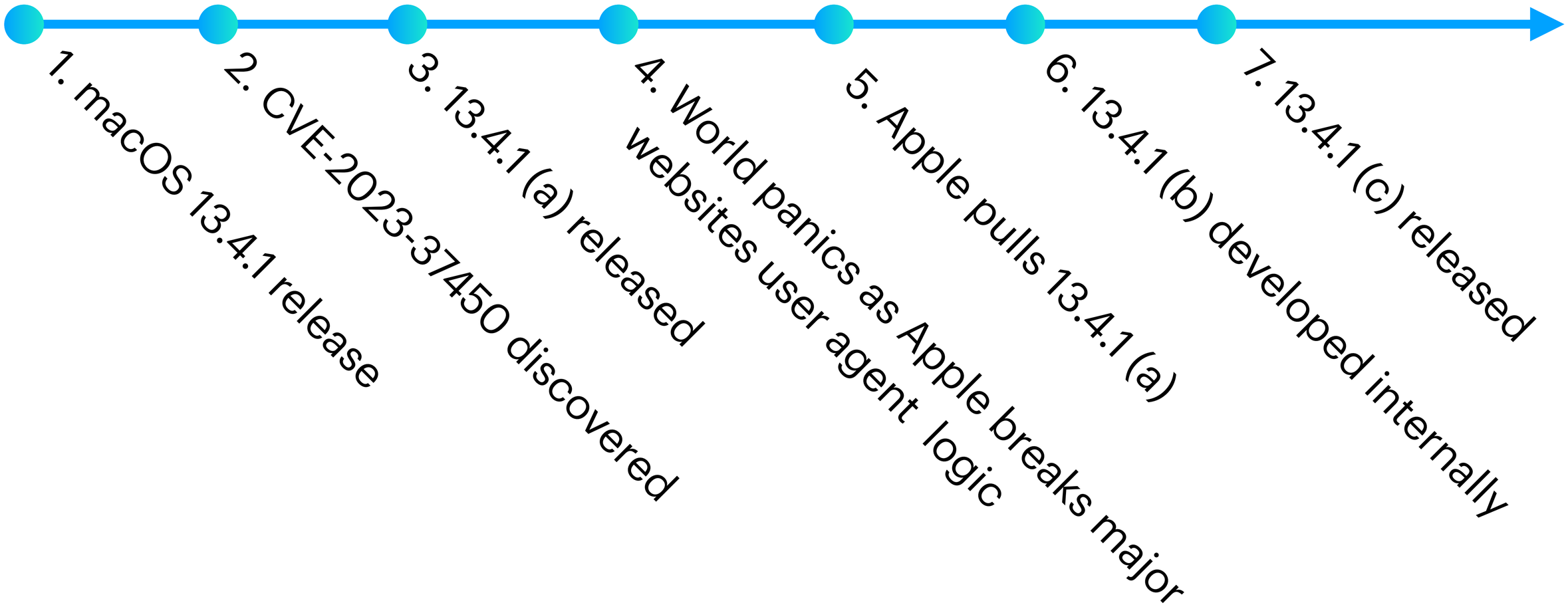
macOS 13.4.1 (& iOS 16.5.1) Timeline

13.4.1 (c) released 🎉

July 12, 2023

1. macOS 13.4.1 release
2. CVE-2023-37450 discovered
3. 13.4.1 (a) released
4. Websites user agent logic
5. Apple pulls 13.4.1 (a)
6. 13.4.1 (b) developed internally

macOS 13.4.1 (& iOS 16.5.1) Timeline



macOS 13.4.1 (& iOS 16.5.1) Timeline

1. macOS 13.4.1 release
2. CVE-2023-37450 discovered
3. 13.4.1 (a) released
4. Websites using WebKit Graphics as Apple breaks major user agent logic
6. Apple pulls 13.4.1 (a)
6. 13.4.1 (b) developed internally
7. 13.4.1 (c) released

macOS 13.4.1 (& iOS 16.5.1) Timeline

What was the fix?

1. macOS 13.4.1 release
2. CVE-2023-37450 discovered
3. 13.4.1 (a) released
4. Websites using Apple browser agent logic
5. Apple pulls 13.4.1 (a)
6. 13.4.1 (b) developed internally
7. 13.4.1 (c) released

macOS 13.4.1 (& iOS 16.5.1) Timeline

What was the fix?

Removing letters from user-agent versioning...

Safari 13.4.1 (a) -> Safari 13.4.1

Where are RSRs* secretly hiding today?

Where are RSR technologies secretly hiding today?

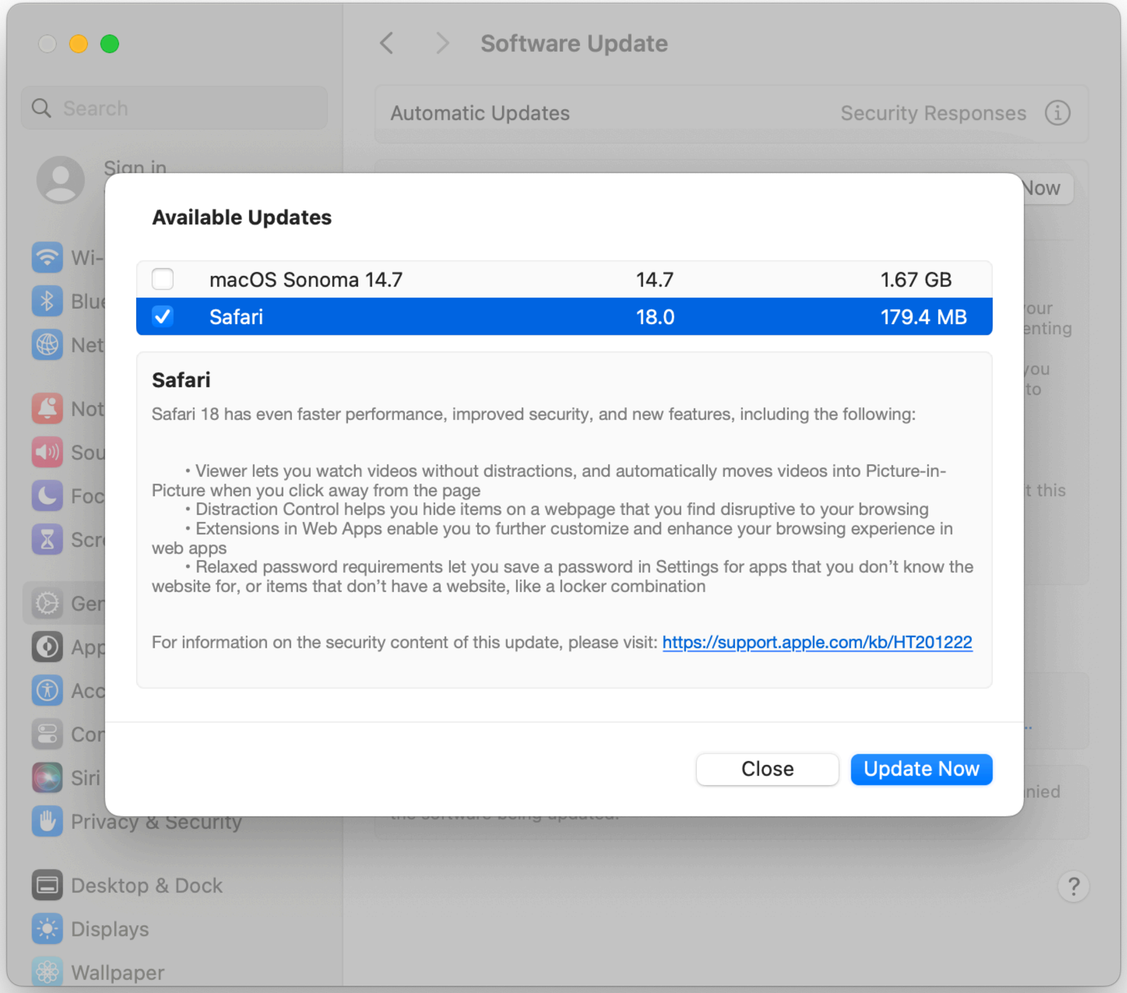
Where are RSR technologies secretly hiding today?

Safari Updates

Private Cloud Compute

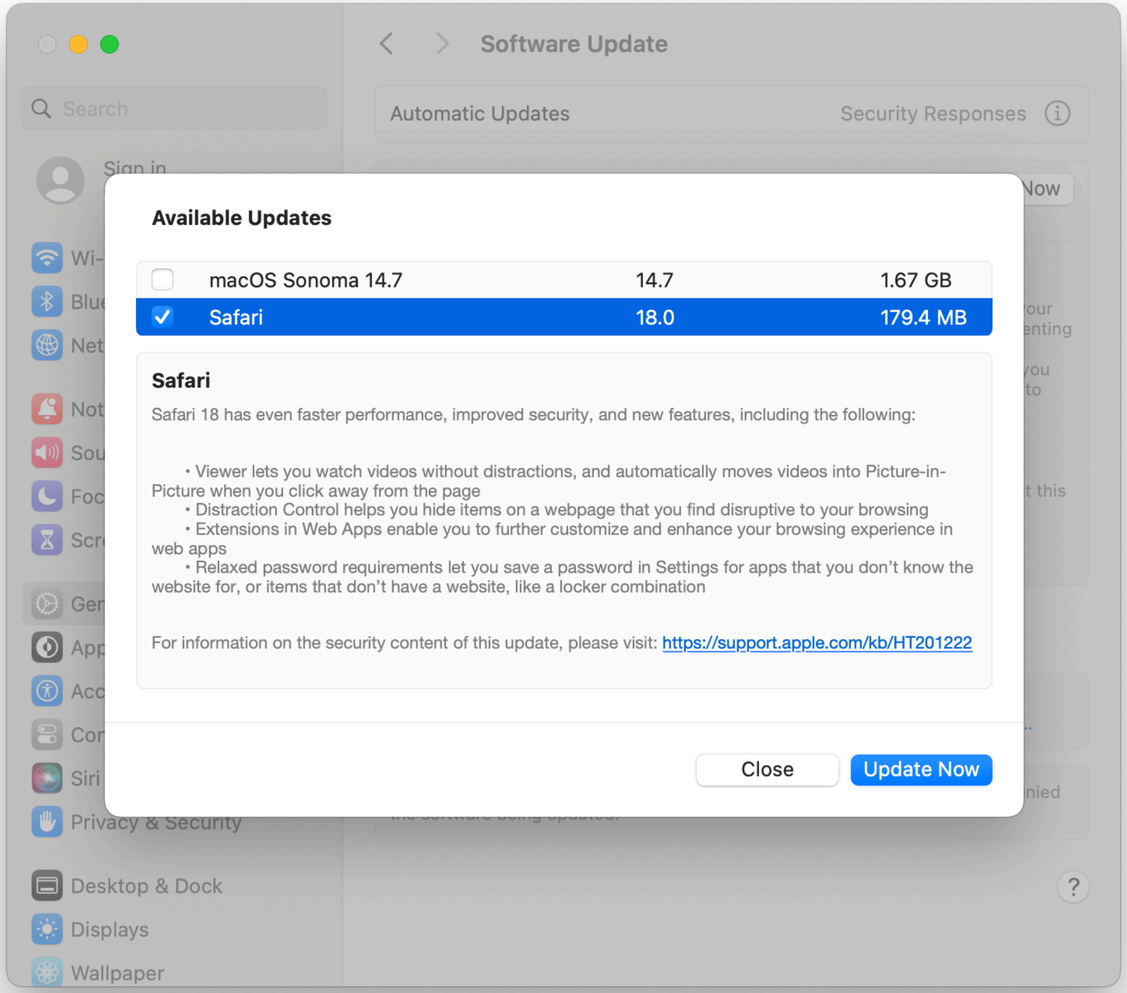
Safari Updates

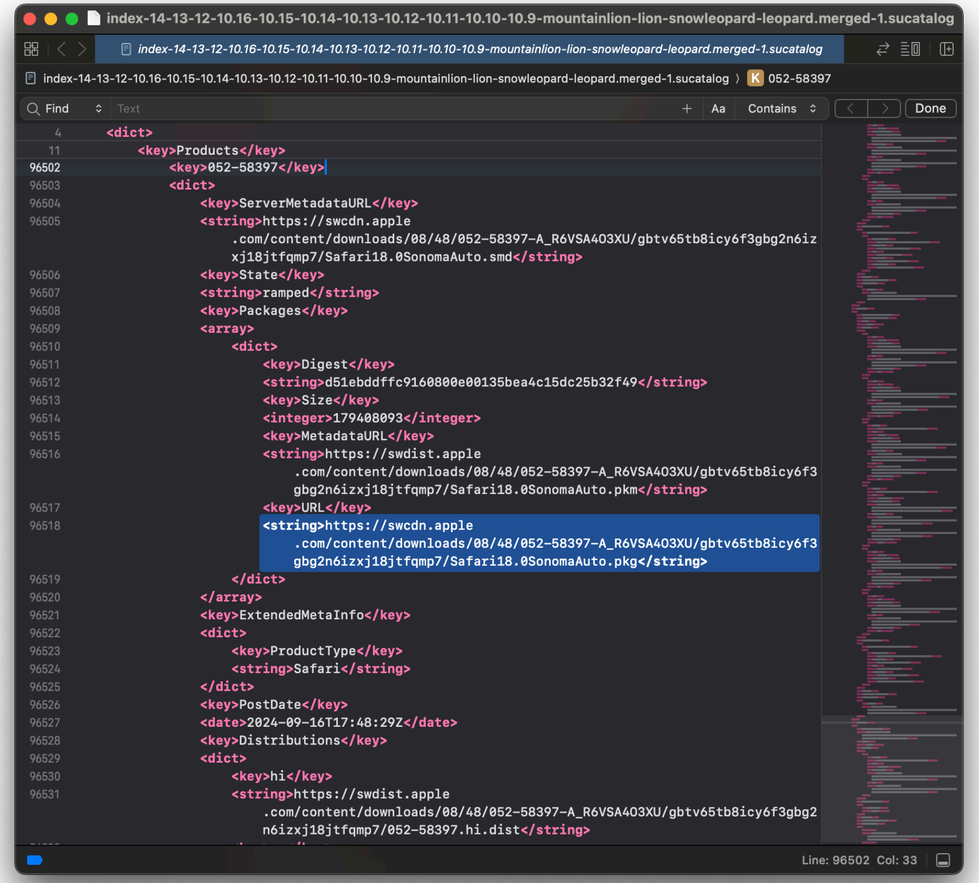
N-1 & N-2
Security Updates



Safari Updates

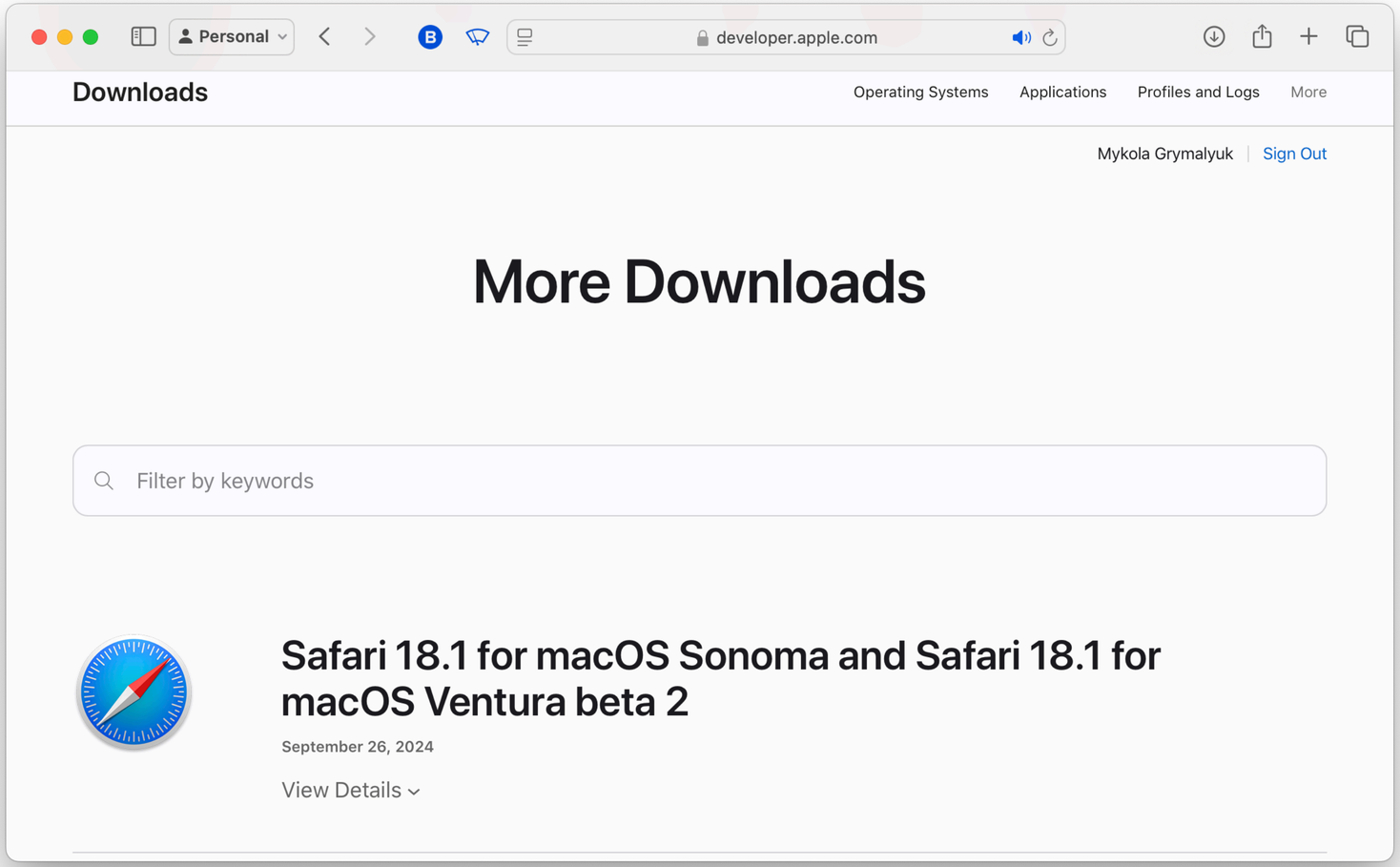
https://swscan.apple.com/content/catalogs/others/index-14-13-12-10.16-10.15-10.14-10.13-10.12-10.11-10.10-10.9-mountainlion-lion-snowleopard-leopard.merged-1.sucatalog



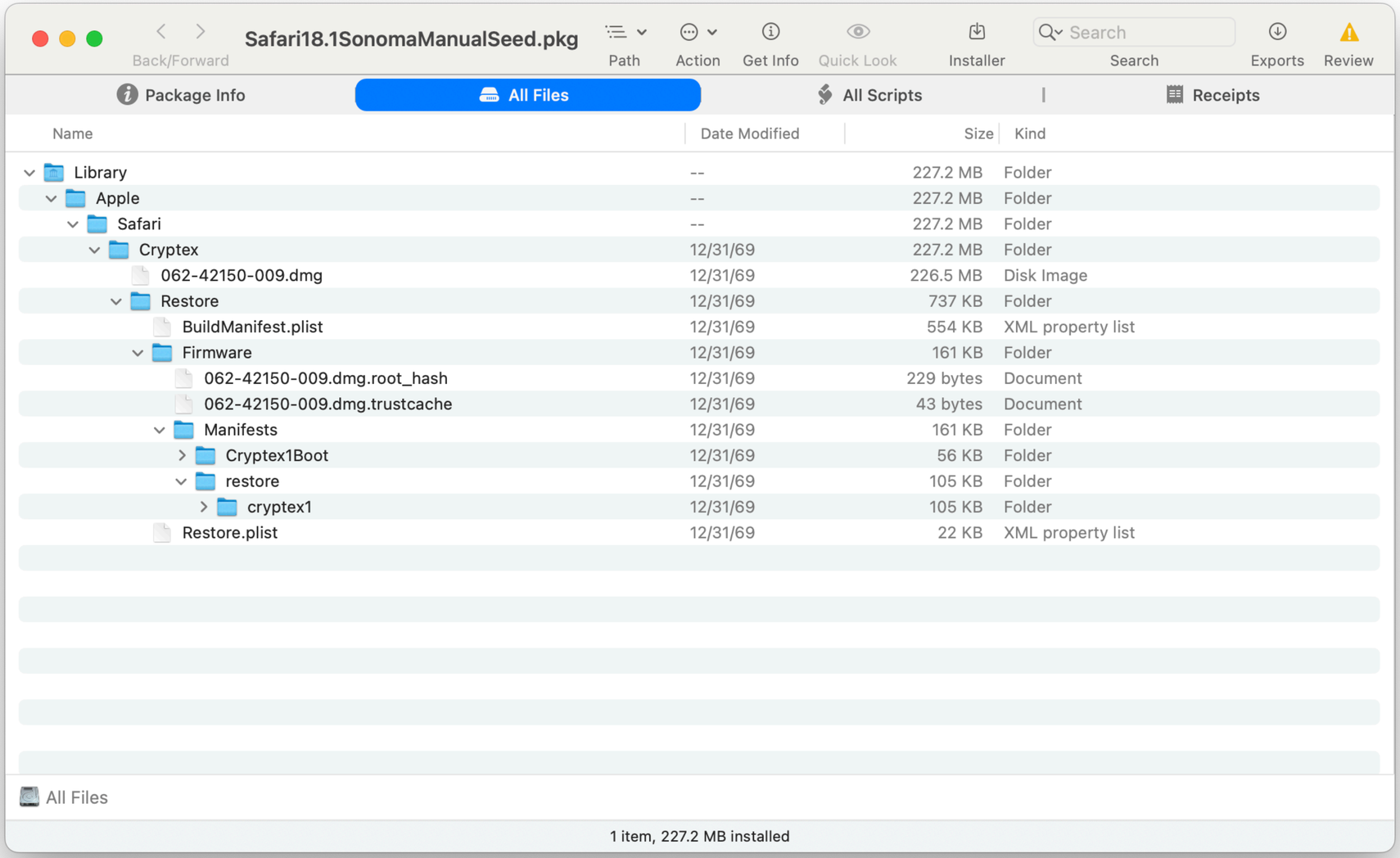


Safari Updates

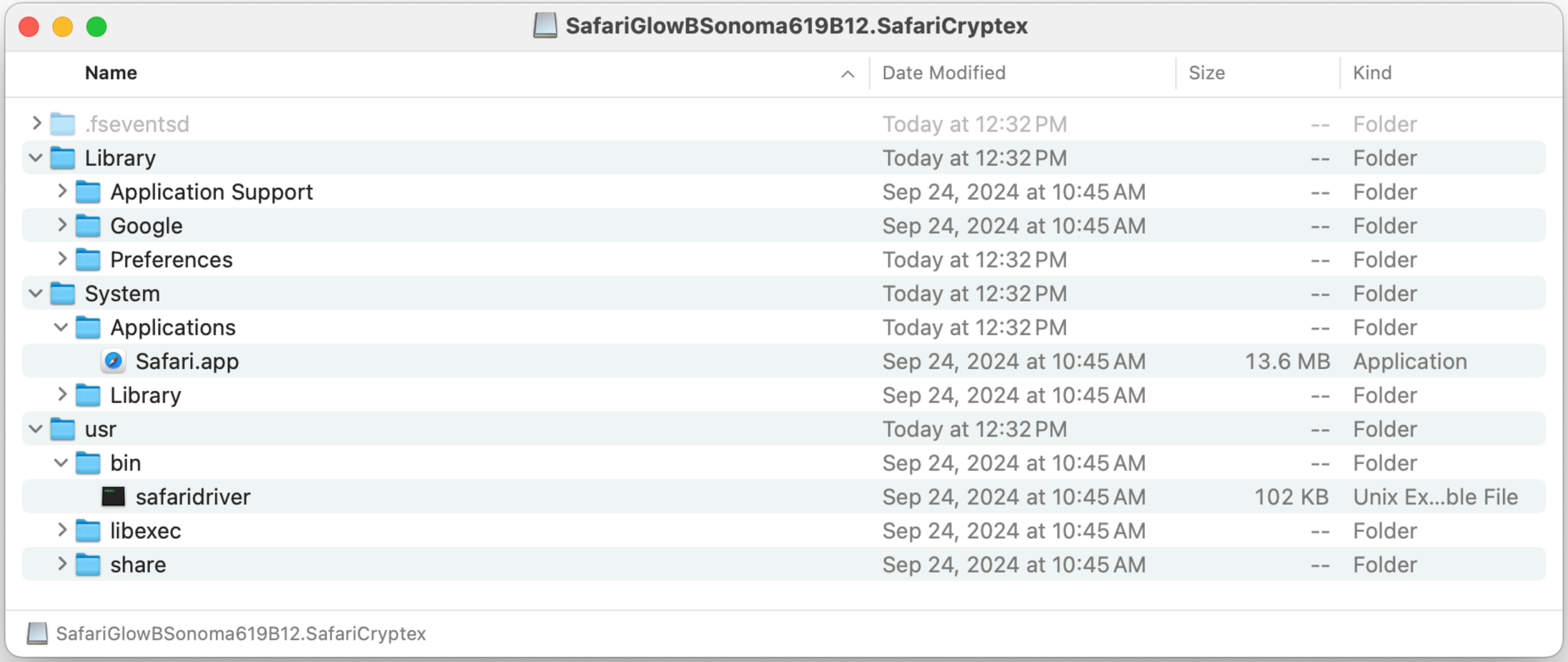
https://developer.apple.com/download/all/?q=Safari



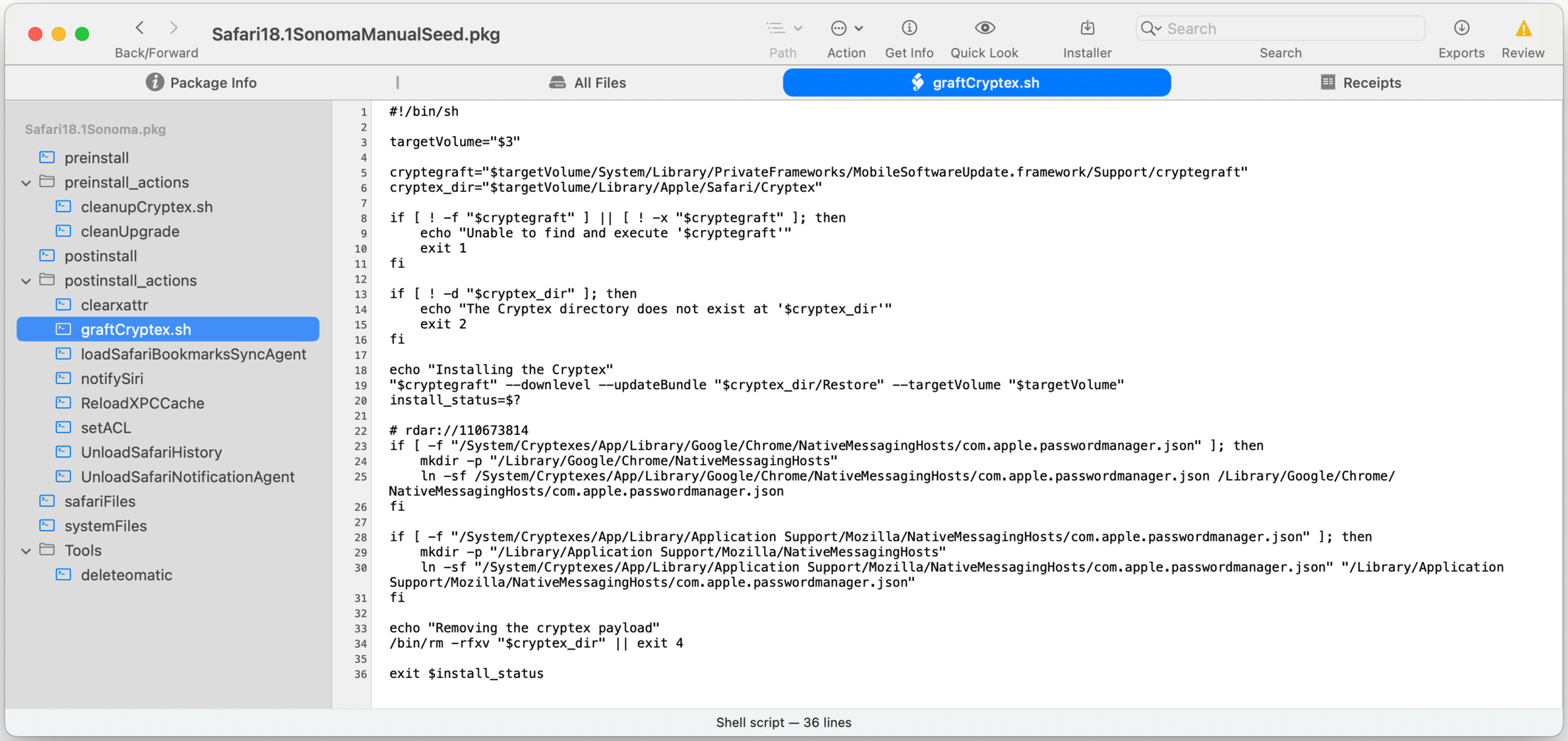
Safari Updates



Safari Updates



Safari Updates



Safari Updates

Deltas?

Ability to revert?

Private Cloud Compute

Private Cloud Compute

Standard & Codeless Cryptexes

- Debug Shell Cryptex
- Model Weight Cryptex (data only)

Private Cloud Compute

Standard & Codeless Cryptexes

- Debug Shell Cryptex
- Model Weight Cryptex (data only)

Please read the Private Cloud Compute Documentation, really neat stuff:
https://security.apple.com/documentation/private-cloud-compute/

RSR technologies live on 🎉

Even if not directly through “Rapid Security Response” updates

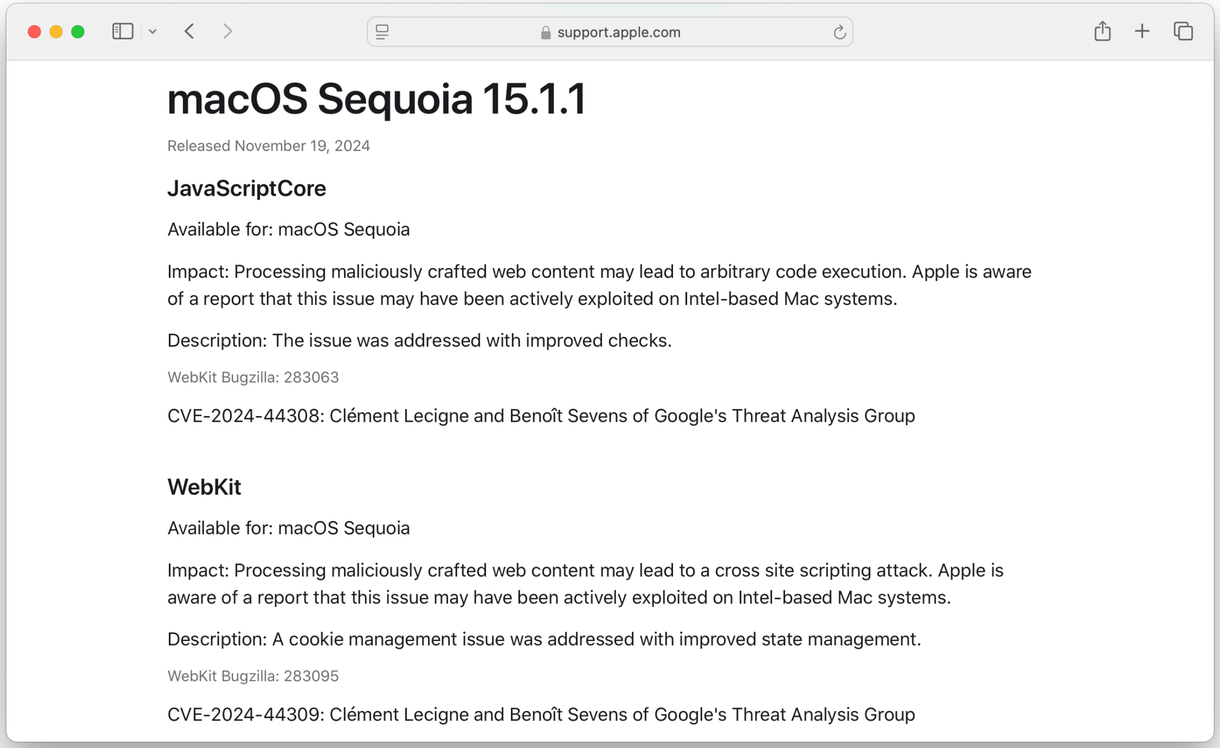
“Is Rapid Security Response a Failure?”

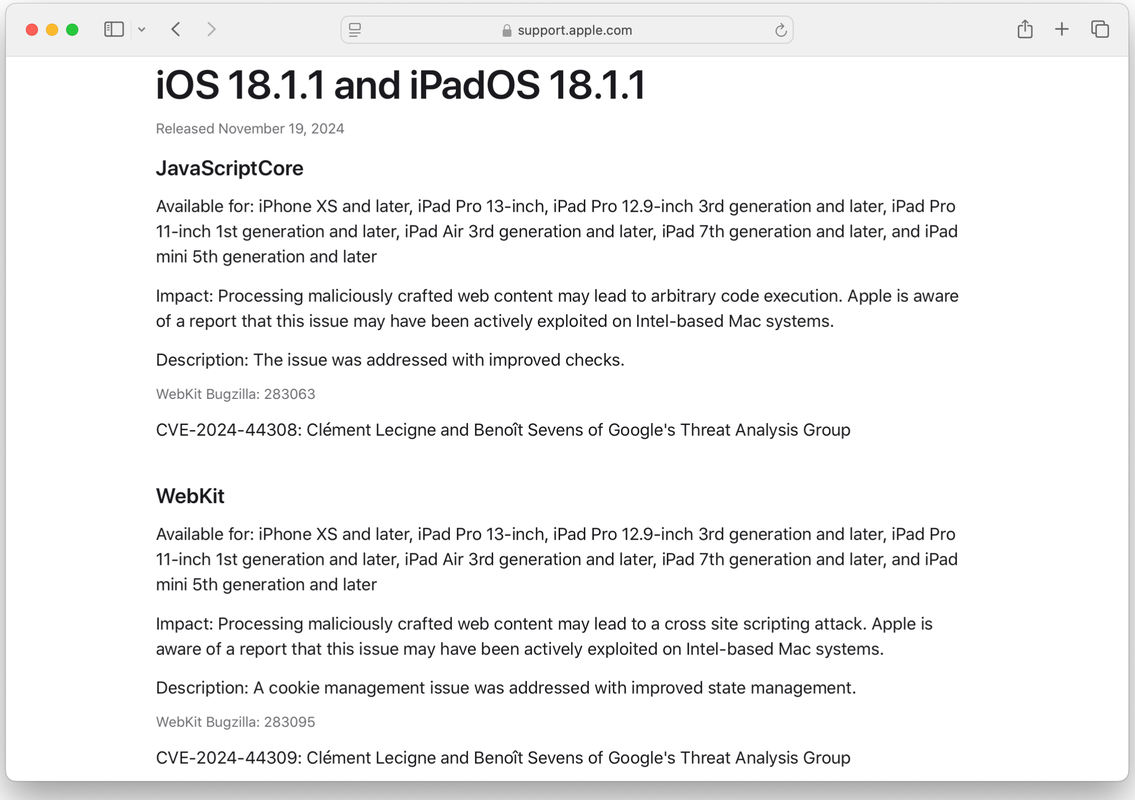
- Me, complaining on the internet (April, 2023)

“Is Rapid Security Response a Failure?”

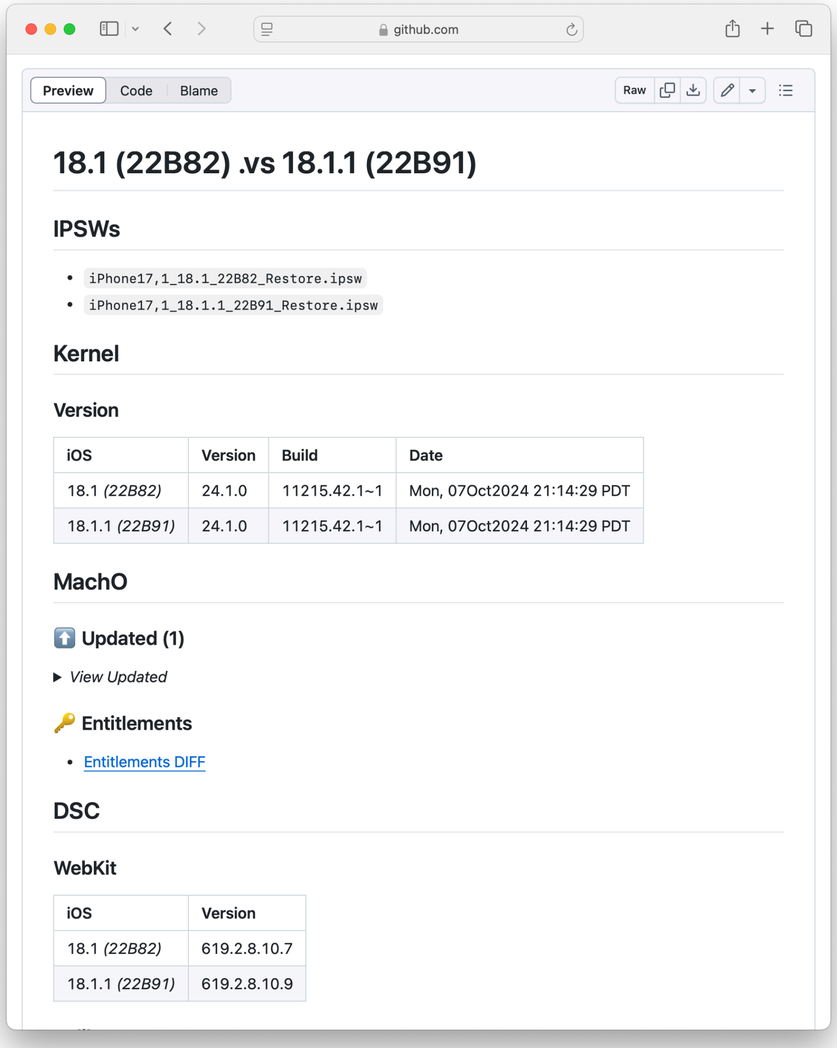
macOS 15.1.1 & iOS 18.1.1…

macOS 15.1.1 & iOS 18.1.1...





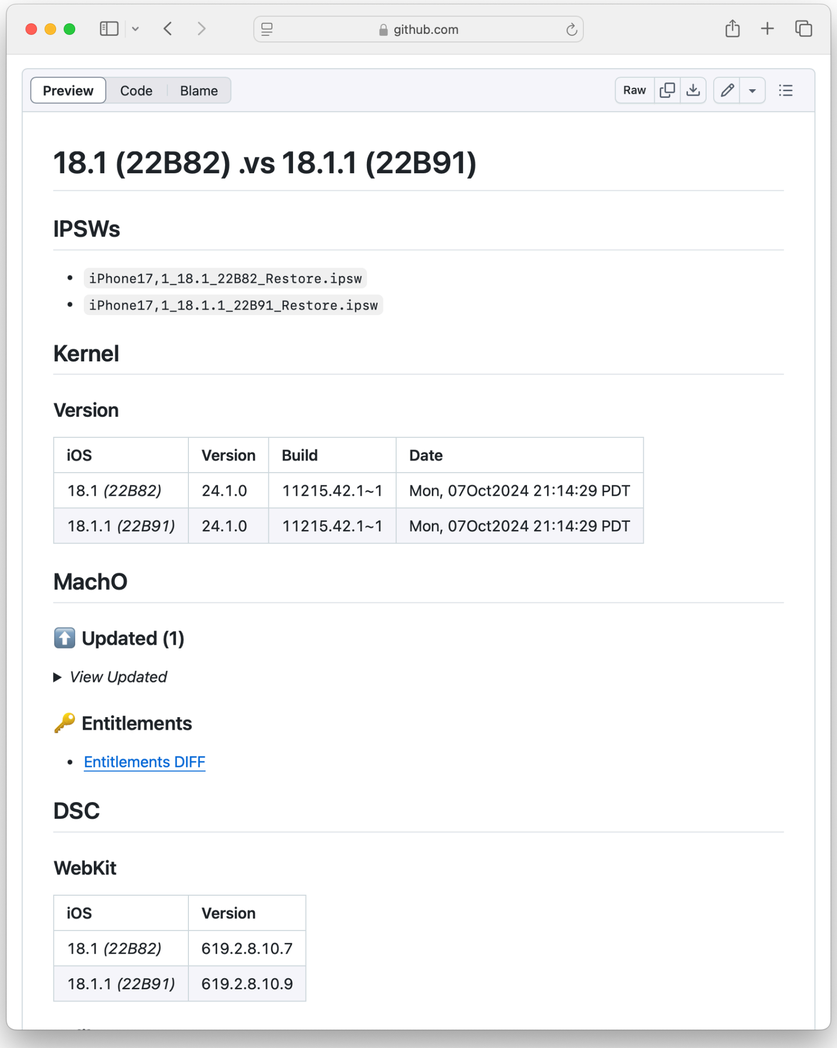
macOS 15.1.1 & iOS 18.1.1...



Thank you blacktop_ for the diff!

Userspace-only fixes...

macOS 15.1.1 & iOS 18.1.1...



Thank you blacktop_ for the diff!

Userspace-only fixes...

This ~~*meeting*~~ security update could've been ~~*an email*~~ a Rapid Security Response

Conclusion


RSR & Cryptexes
I JUST THINK THEY'RE NEAT!

Thanks for listening to my rambles!

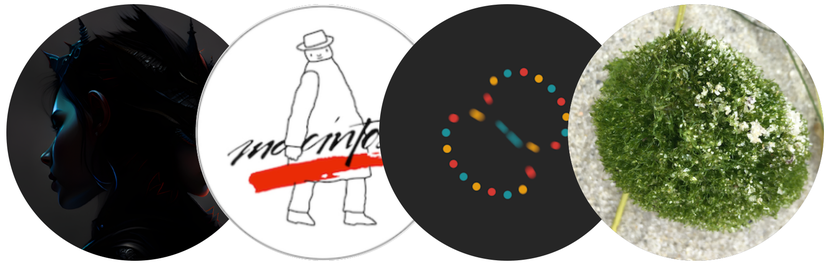
Mirrored on khronokernel.com

Socials

- Twitter: https://twitter.com/khronokernel
- GitHub: https://github.com/khronokernel
- LinkedIn: https://www.linkedin.com/in/mykola-grymalyuk

Credits to those who directly/indirectly helped make this possible!

- Blacktop_: IPSW Diffs
- MrMacintosh: Archival of Software Updates
- ASentientBot & DhinakG: Reverse Engineering Cryptexes & OpenCore Legacy Patcher work



References

References:

- Security Research Device:
  - archive.org (August 3rd, 2020): https://developer.apple.com/programs/security-research-device/
- CVE-2023-37450:
  - Security Contents: https://support.apple.com/en-il/106354
- Aaron's report on 13.4.1 (a) being pulled:
  - Twitter: https://x.com/aaronp613/status/1678603028113289217
- Private Cloud Compute Cryptex Usage:
  - Software Layering: https://security.apple.com/documentation/private-cloud-compute/softwarelayering#Cryptexes
- macOS 15.1.1 Security Contents:
  - https://support.apple.com/en-ca/121753
- iOS 18.1.1 Security Contents:
  - https://support.apple.com/en-ca/121752
- Blacktop_'s iOS 18.1.1 diff against 18.1:
  - https://github.com/blacktop/ipsw-diffs/blob/main/18_1_22B82_vs_18_1_1_22B91/README.md
- macOS 13.4.1 (c) User agent:
  - https://tidbits.com/2023/07/13/rapid-security-responses-for-ios-ipados-16-5-1-a-and-macos-ventura-13-4-1-a-2/
- macOS 13.4.1 IPSW:
  - https://updates.cdn-apple.com/2023SpringFCS/fullrestores/042-01877/2F49A9FE-7033-41D0-9D0C-64EFCE6B4C22/UniversalMac_13.4.1_22F82_Restore.ipsw
- macOS 13.4.1 (a) RSR payload:
  - x86_64: https://updates.cdn-apple.com/2023SummerFCS/patches/042-11156/A193B5BA-AC1C-4074-A4EB-18BE0E10DBAD/com_apple_MobileAsset_MacSplatSoftwareUpdate/d89d81737950136e2d6106ecbfbff16aa024e8e6.zip
  - arm64: https://updates.cdn-apple.com/2023SummerFCS/patches/042-11155/2C86E0CF-DB4F-4C20-8925-2F3F54F61A11/com_apple_MobileAsset_MacSplatSoftwareUpdate/fda10f2f66899a3530fd1cc7e99d0267eabef6c2.zip
- Shameless plug:
  - https://khronokernel.com/macos/2023/04/18/RSR.html