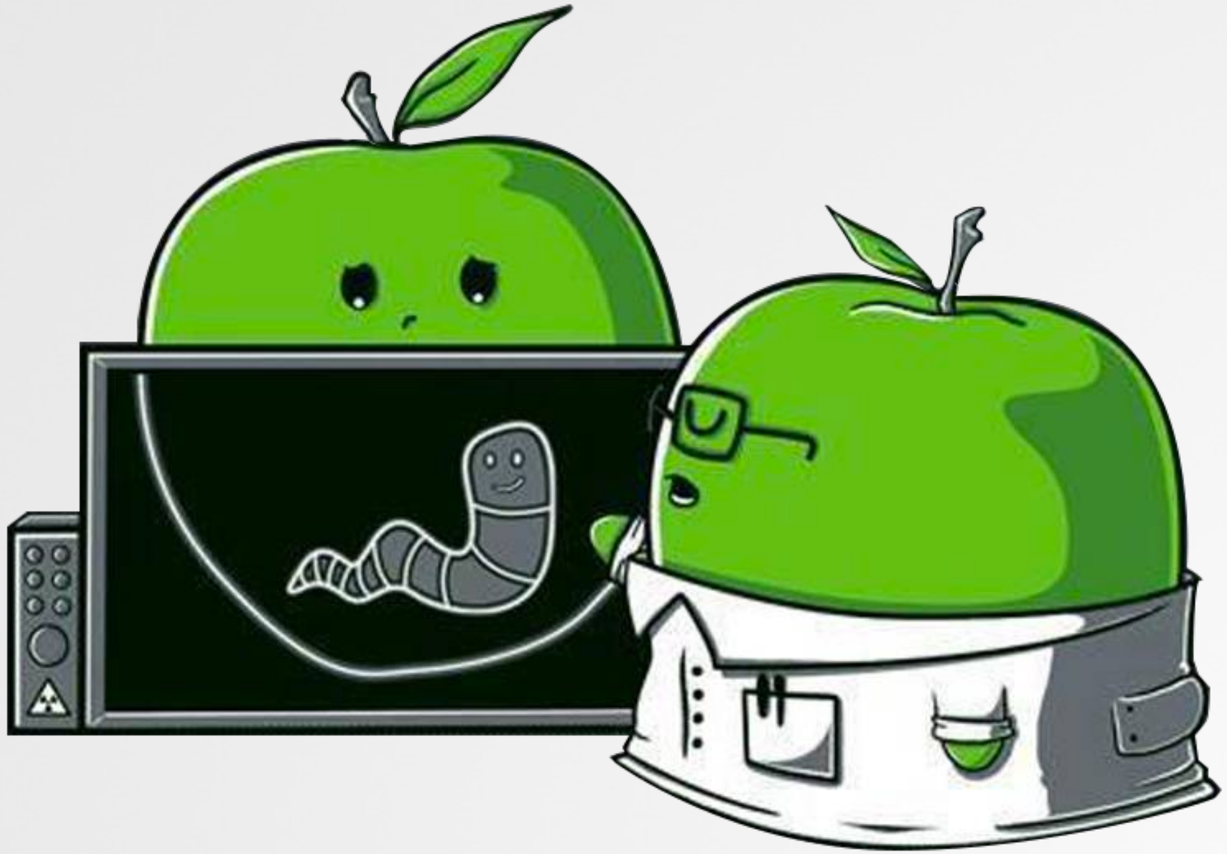
From Crash

...to trash? cash? or (/bin/)bash?



What You Will Learn



Although the talk is focused on understanding crash reports, we'll also touch on topics such as reverse-engineerings, vulnerabilities, & more.

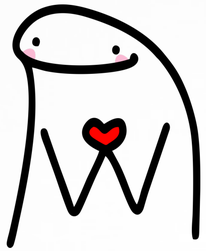


Understanding crash reports (on macOS)



Uncovering malware, bugs, & more!

...and come away with a deep appreciation (love?) crash reports!



Who Cares About Crash Report?

...(almost) everybody, really really should!



Users & Developers

to understand & get bugs fixed

> First Name: Patrick Wardle
> Email: patwardle@yahoo.com
>
>
> Crash on opening bundle
>
> Hopper: v5.11.0
> macOS: 13.4 (arm64)
>
> Repo: Open a bundle such as /System/Library/UserNotifications/Bundles/com.apple.battery-monitor.bundle
>
> Hopper will crash immediately w/ NULL-pointer exception?
>
> Translated Report (Full Report Below)
> ----------
>
> Process: Hopper Disassembler v4 [4923]
> Path: /Applications/Hopper Disassembler v4.app/Contents/MacOS/Hopper Disassembler v4
> Identifier: com.cryptic-apps.hopper-web-4
> Version: 5.11.0 (5.11.0)
> Code Type: ARM-64 (Native)
> Parent Process: launchd [1]
> User ID: 501
>
> Date/Time: 2023-06-05 09:05:53.2815 +0300
> OS Version: macOS 13.4 (22F66)
> Report Version: 12
>
>
> Time Awake Since Boot: 38000 seconds
> Time Since Wake: 855 seconds
>
> System Integrity Protection: enabled
>
> Crashed Thread: 0 Dispatch queue: com.apple.main-thread

bug report w/ crash report

"will be fixed in the next update" :)

Hello Patrick,

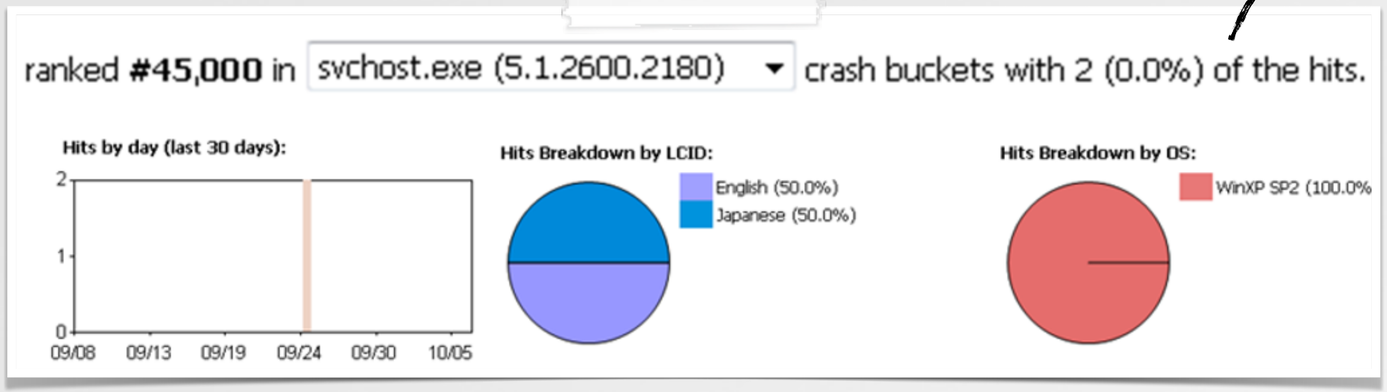
Thank you for your report.

I had a look, and I've found a way to reproduce the crash. In fact, it seems that you are using the menu File > Open which is reserved to opening Hopper databases, not executables. That being said, a few years ago, I've added a fallback for the case where a user would use this menu to load something else. The problem is that not all paths have been considered, and in your case, the bundle that you are trying to load only contain resources and no executable. In that case, Hopper crashes.
This will be fixed in the next update.

Regards,
Vincent Bénony

Security Teams
to find (failed) exploits ...0days?

ranked #45,000
with "exactly 2 hits ever"



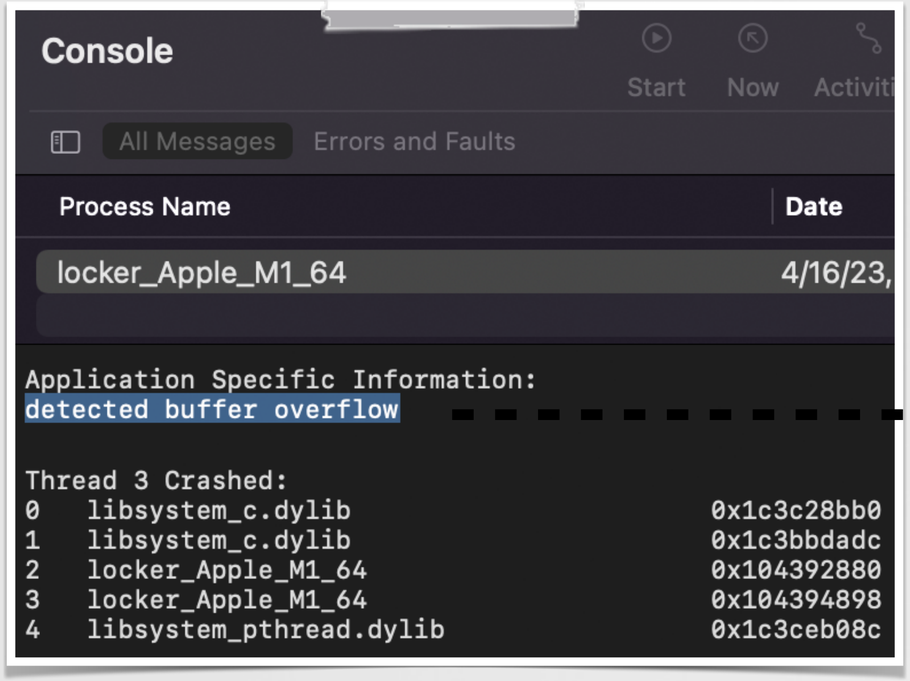
0-day
(patched as MS08-067)

"By September 2008 we had built a system that screened millions of crashes for security exploits. On September 25th a crash came in that got my attention--an exploit in netapi32.dll..."

The attackers had a remote code execution vulnerability that affected every version of Windows, gave them full control at SYSTEM level rights, left almost no forensic footprint, and could be used anonymously from anywhere on the Internet. Their exploit was 95% reliable. Almost perfect. Almost." -John Lambert/MSRC

Malware Analysts

to help detect & analyze malware



I dug into this more today, and created a patched version of the ransomware that doesn't crash

(Basically I just NOP'd out the call to the `put_note` function ...so you won't see a note being created ...but also then no crash and the files are encrypted). I've attached it (⚠️ yes its malware, so treat appropriately), the password is: **infect3d**

patch ...to test



LockBit Ransomware

...crashes on macOS

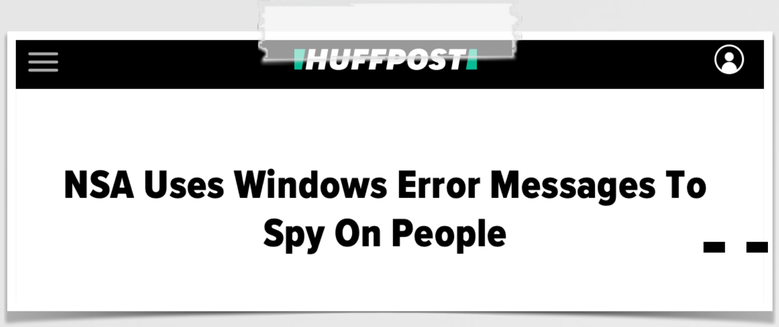
> Five months later a seemingly unrelated event occurred. A computer security firm in Belarus was called in to troubleshoot a series of computers in Iran that were crashing and rebooting repeatedly. Again, the cause of the problem was a mystery. That is, until the researchers found a handful of malicious files on one of the systems and discovered the world's first digital weapon.

Remember Stuxnet? ❤️

...uncovered via crashes

Hackers & Intelligence Agencies
to find exploitable vulnerabilities





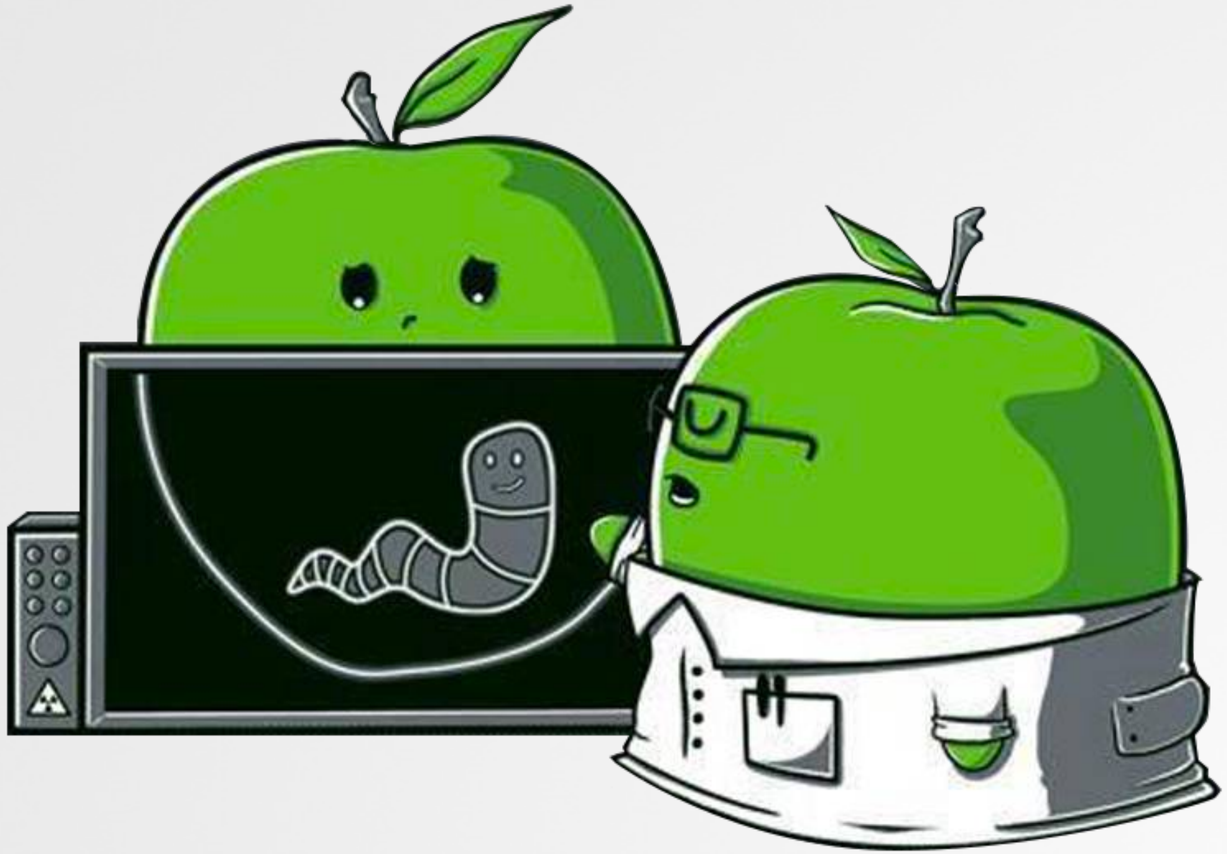


leaked image



"The automated crash reports are a 'neat way' to gain 'passive access' to a machine, the presentation continues. This passive access to error messages provides valuable insights into problems with a targeted person's computer and, thus, information on security holes that might be exploitable for planting malware or spyware on the unwitting victim's computer." -spiegel.de

Crash Report Basics



What is a Crash Report



What are Crash Reports?
"System & user reports about apps/processes that crash" -Apple

MacPaw

Mac Crashes: why they happen and how to fix them

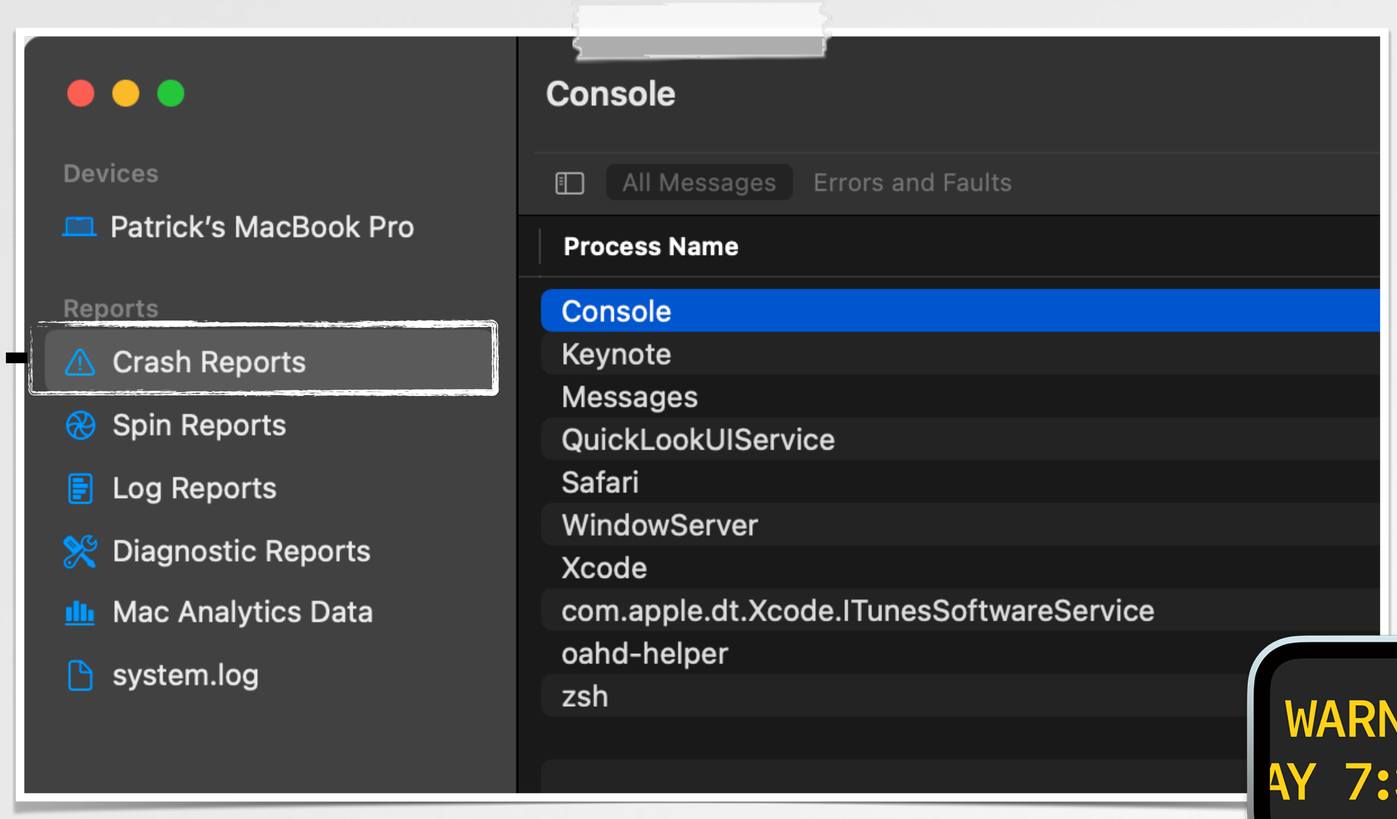
Mac crash report is an important file that can help developers investigate software imperfections and discover how to enhance both OS and computer programs.

On the other hand, crash reports are not that useful for users. So, if you can't understand the report, that's not your fault because it's generated mainly for developers.

Understanding reports
...takes some knowledge!

Where are Crash Reports Stored?

as .ips files in "DiagnosticReports" directories



macOS's Console app

crash reports file extension

/Library/Logs/DiagnosticReports/*.ips

~/Library/Logs/DiagnosticReports/*.ips

A (Demo) Crash

...and a walk thru

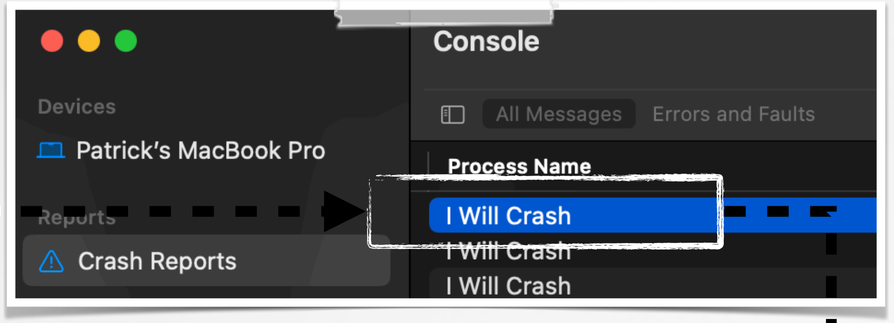
...run → crash → crash report

```
% ./"I Will Crash"
zsh: segmentation fault  ./"I Will Crash"
```

```
01  int main(int argc, const char * argv[]) {
02
03      char* a = NULL;
04      *a = 0x41;
05
06  }
```



Buggy code




```
Translated Report (Full Report Below)
------------------------------------

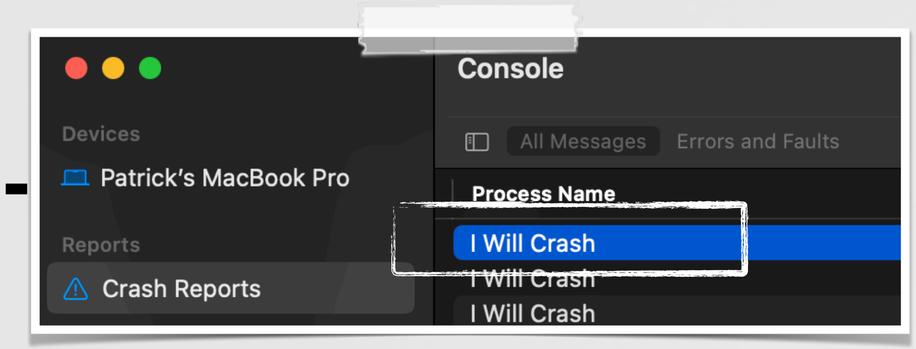
Process:                I Will Crash [30670]
Path:                   /Users/USER/Library/Developer/Xcode/.../Build/Products/Debug/I Will Crash
Identifier:             I Will Crash
Version:                1.0
Code Type:              ARM-64 (Native)
Parent Process:         zsh [26035]
Responsible:            Terminal [48970]
User ID:                501

Date/Time:              2023-08-26 10:41:16.0317 +0900
OS Version:             macOS 13.4 (22F66)
```

Crash report

Crashed Thread

...and exception information




```
Crashed Thread:        0  Dispatch queue: com.apple.main-thread

Exception Type:        EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:       KERN_INVALID_ADDRESS at 0x0000000000000000
Exception Codes:       0x0000000000000001, 0x0000000000000000
```

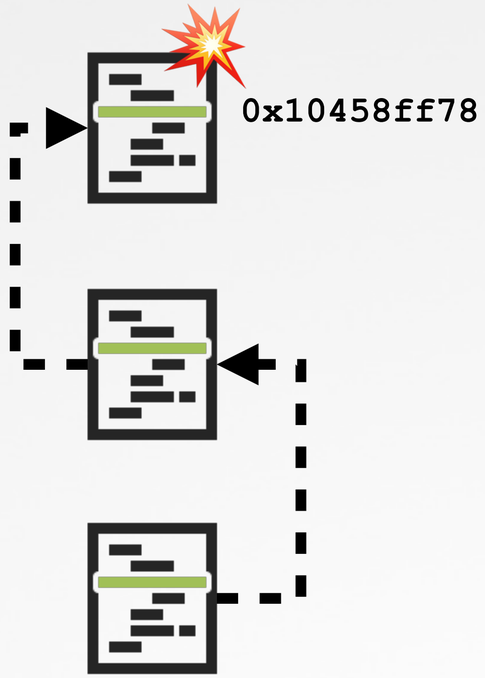
1. Crashed Thread
2. Exception Information
   - Exception type
   - Exception specifics

Stack Backtrace

from faulting address, backwards

"Faulting Address"
address of instruction that caused exception

```
Thread 0 Crashed::  Dispatch queue: com.apple.main-thread
0    I Will Crash       0x10458ff78 main + 44
1    dyld               0x18c867f28 start + 2236
....
```



Stack Frame #	Binary	Return Address*
0	I Will Crash	0x10458FF78
1	dyld	0x18C867F28

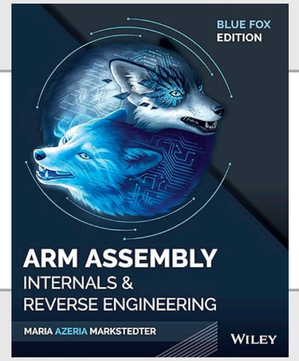
*Frame 0:
Faulting address

(CRASHED) THREAD STATE

which includes all registers

```
Thread 0 crashed with ARM Thread State (64-bit):
    x0: 0x0000000000000001   x1: 0x000000016b873818   x2: 0x000000016b873828   x3: 0x000000016b8738f0
    x4: 0x0000000000000000   x5: 0x0000000000000000   x6: 0x0000000000000000   x7: 0x0000000000000000
    ...
   x28: 0x0000000000000000   fp: 0x000000016b873580   lr: 0x000000018c867f28
    sp: 0x000000016b873550   pc: 0x000000010458ff78 cpsr: 0x80001000
   far: 0x0000000000000000  esr: 0x92000046 (Data Abort) byte write Translation fault
```

Register	Description
x0 - x7	arg 0, ...arg 7 (x0, also return value from function)
pc	(faulting) instruction
lr	return address
fp / sp	stack frame / stack pointer



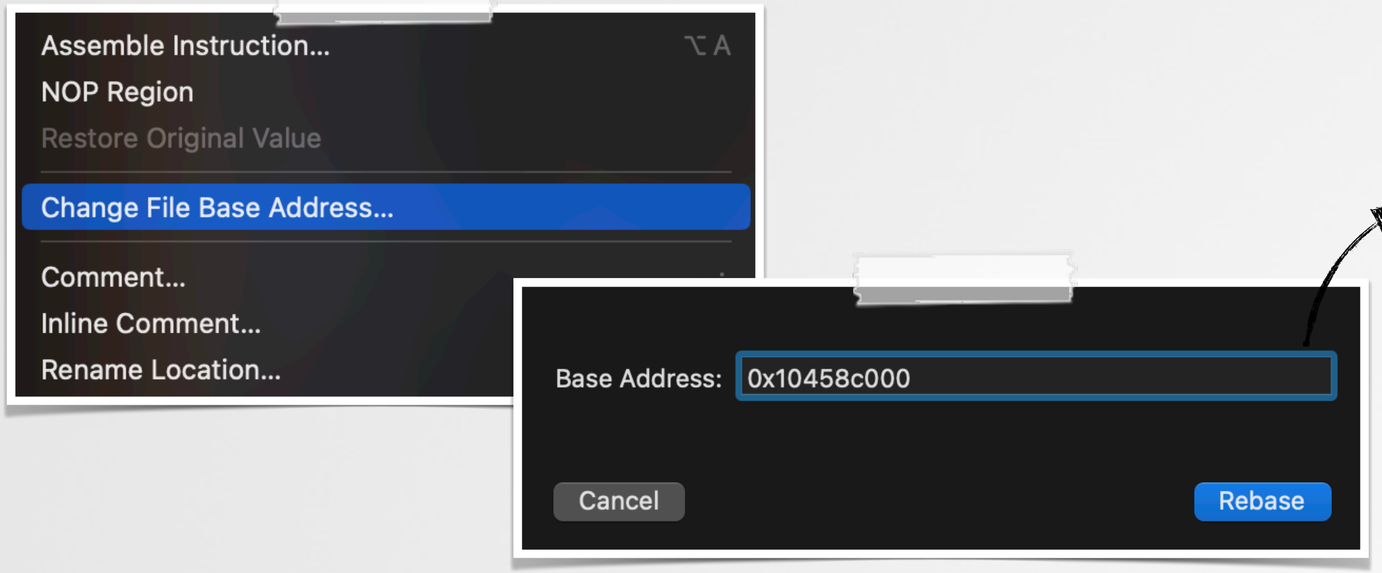
Arm Assembly Internals & Reverse Engineering
Author: Maria Markstedter (🦊)

Loaded Binaries

including their (ASLR'd) address & path

In-memory address (start, end)

```
Binary Images:
0x10458c000 - 0x10458ffff I Will Crash (*). ~/Library/Developer/Xcode/DerivedData/.../Debug/I Will Crash
0x18c862000 - 0x18c8f0553 dyld (*) /usr/lib/dyld
```



Rebase (to **0x10458c000**), so disassembly matches addresses in the crash report

The Faulting Address

address of instruction that caused the exception

```
Thread 0 Crashed::  Dispatch queue: com.apple.main-thread
0    I Will Crash          0x10458ff78 main + 44
1    dyld                  0x18c867f28 start + 2236
```

```
01  main:
02  0x10458ff4c          sub         sp, sp, #0x40
03  0x10458ff50          stp         fp, lr, [sp, #0x30]
04  0x10458ff54          add         fp, sp, #0x30
05  0x10458ff58          mov         w8, #0x0
06  0x10458ff5c          str         w8, [sp, #0x30 + var_1C]
07  0x10458ff60          stur        wzr, [fp, var_4]
08  0x10458ff64          stur        w0, [fp, var_8]
09  0x10458ff68          stur        x1, [fp, var_10]
10  0x10458ff6c          str         xzr, [sp, #0x30 + var_18]
11  0x10458ff70          ldr         x9, [sp, #0x30 + var_18]
12  0x10458ff74          mov         w8, #0x41
13
14  0x10458ff78          strb        w8, [x9]
15
16  ....
```

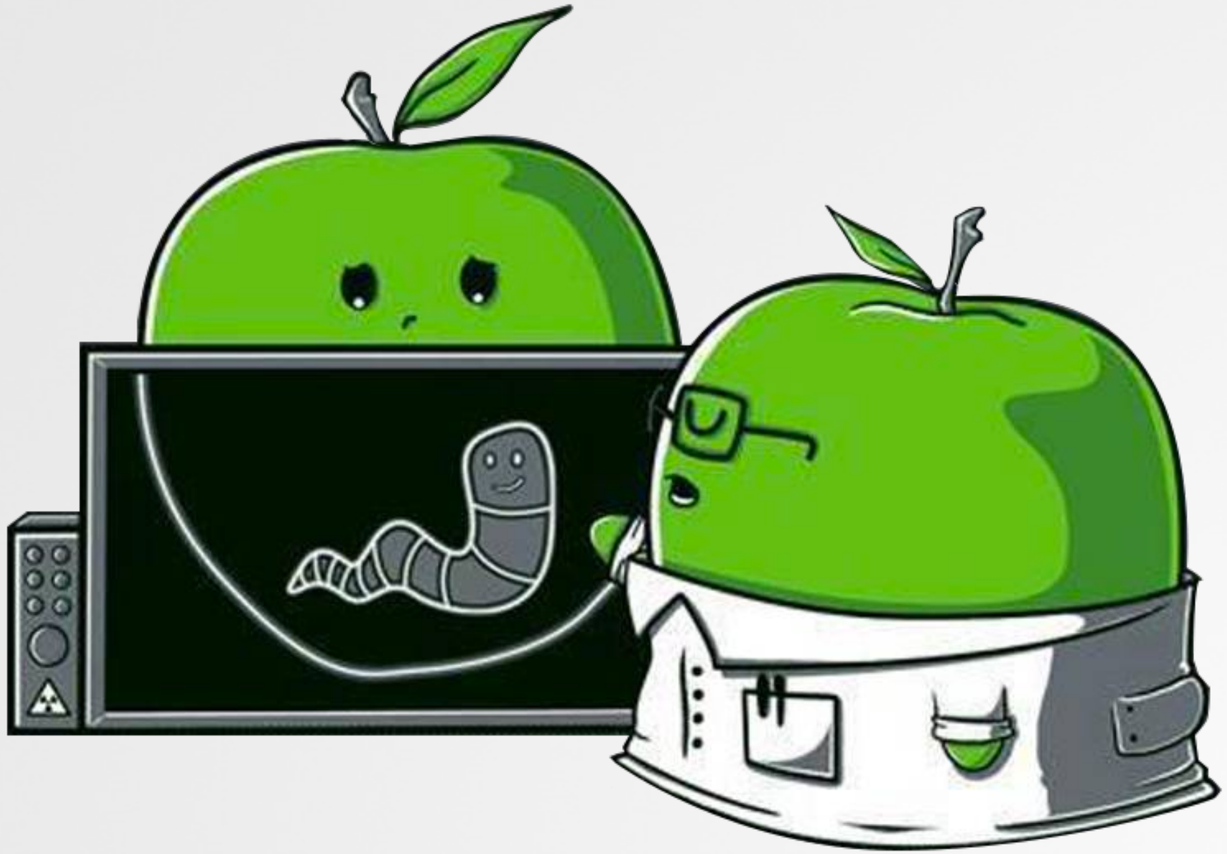
in C
(for reference)

```
01  int main(...) {
02
03      char* a = NULL;
04      *a = 0x41;
```

"I Will Crash" disassembly

solely from my computer
(...and solely from normal use!)

(Real) Crashes



CRASH: LULU

...with a EXC_BAD_ACCESS / SIGBUS

```
Process:             com.objective-see.lulu.extension [87832]
Path:                /Library/SystemExtensions/*/com.objective-see.lulu.extension

Crashed Thread:      5 com.apple.NSXPCConnection.user.VBG97UB4TA.com.objective-see.lulu.87821

Exception Type:      EXC_BAD_ACCESS (SIGBUS)
Exception Codes:     KERN_PROTECTION_FAILURE at 0x000000016ba9ffe0
Exception Codes:     0x0000000000000002, 0x000000016ba9ffe0
```

```
Thread 5 Crashed::  Dispatch queue: com.apple.NSXPCConnection.user.VBG97UB4TA.com.objective-see.lulu.87821
0   CoreFoundation                      0x18cc4eb60 __CFStringAppendBytes + 76
1   CoreFoundation                      0x18cc4dad8 __CFStringAppendFormatCore + 9652
2   CoreFoundation                      0x18cd8b0b8 _CFStringCreateWithFormatAndArgumentsReturningMetadata + 184
3   CoreFoundation                      0x18cc94840 CFStringCreateWithFormatAndArguments + 312
4   SystemConfiguration                 0x18d895b4c SCDynamicStoreKeyCreate + 56
5   SystemConfiguration                 0x18d8969e8 SCDynamicStoreCopyConsoleUser + 116
6   com.objective-see.lulu.extension    0x1043671c8 getConsoleUser + 28 (utilities.m:127)
7   com.objective-see.lulu.extension    0x104376f38 -[FilterDataProvider processEvent:] + 196 (FilterDataProvider.m:250)
8   com.objective-see.lulu.extension    0x10437a750 -[FilterDataProvider processRelatedFlow:] + 816 (FilterDataProvider.m:771)
9   com.objective-see.lulu.extension    0x10437962c -[FilterDataProvider processEvent:] + 10168 (FilterDataProvider.m:626)
10  com.objective-see.lulu.extension    0x10437a750 -[FilterDataProvider processRelatedFlow:] + 816 (FilterDataProvider.m:771)
11  com.objective-see.lulu.extension    0x10437962c -[FilterDataProvider processEvent:] + 10168 (FilterDataProvider.m:626)
12  com.objective-see.lulu.extension    0x10437a750 -[FilterDataProvider processRelatedFlow:] + 816 (FilterDataProvider.m:771)
13  com.objective-see.lulu.extension    0x10437962c -[FilterDataProvider processEvent:] + 10168 (FilterDataProvider.m:626)
14  com.objective-see.lulu.extension    0x10437a750 -[FilterDataProvider processRelatedFlow:] + 816 (FilterDataProvider.m:771)
15  com.objective-see.lulu.extension    0x10437962c -[FilterDataProvider processEvent:] + 10168 (FilterDataProvider.m:626)
16  com.objective-see.lulu.extension    0x10437a750 -[FilterDataProvider processRelatedFlow:] + 816 (FilterDataProvider.m:771)
17  com.objective-see.lulu.extension    0x10437962c -[FilterDataProvider processEvent:] + 10168 (FilterDataProvider.m:626)
18  com.objective-see.lulu.extension    0x10437a750 -[FilterDataProvider processRelatedFlow:] + 816 (FilterDataProvider.m:771)
```

deep, repeating call stack

```
323 libdispatch.dylib                   0x18ca2e624 _dispatch_mach_msg_async_reply_invoke + 344
324 libdispatch.dylib                   0x18ca17960 _dispatch_lane_serial_drain + 372
325 libdispatch.dylib                   0x18ca1862c _dispatch_lane_invoke + 436
326 libdispatch.dylib                   0x18ca23244 _dispatch_workloop_worker_thread + 648
327 libsystem_pthread.dylib             0x18cbbc074 _pthread_wqthread + 288
328 libsystem_pthread.dylib             0x18cbbad94 start_wqthread + 8
```

1️⃣ SIGBUS + 2️⃣ Repeating call stack = "Stack exhaustion" (due to recursion?)

RECURSION 

triggered stack exhaustion

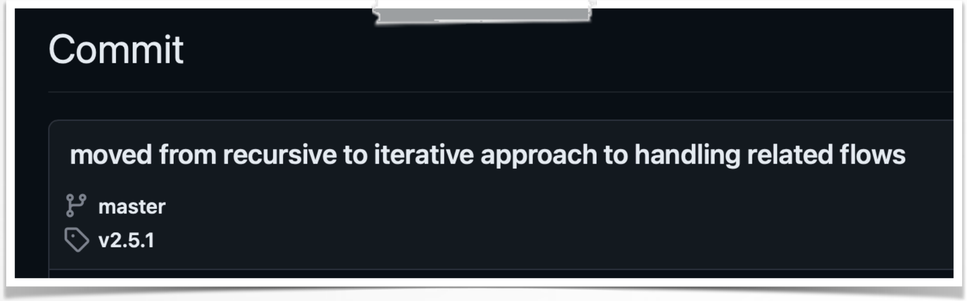
```
01  -(NEFilterNewFlowVerdict*)processEvent:(NEFilterFlow*)flow {
02    ...
03    [self processRelatedFlow:process.key];
```

```
01  -(void)processRelatedFlow:(NSString*)key {
02
03    flows = self.relatedFlows[key];
04    for(NSInteger i = flows.count - 1; i >= 0; i--) {
05
06      flow = flows[i];
07      [flows removeObjectAtIndex:i];
08
09      [self processEvent:flow]
10      ...
```



300+ flows
triggers stack exhaustion

"related" flows



fix: non-recursion

Crash: Do Not Disturb

...but in Apple's APIs?

```
Exception Type:  EXC_BAD_ACCESS (SIGSEGV)
Exception Codes: SEGV_NOOP at 0x0000000000000001
Crashed Thread:  6

Application Specific Information:
Attempted to dereference garbage pointer 0x1.
Originated at or in a subcall of -[FrameworkInterface initIdentity:]

Thread 6 Crashed:
  0   Security          0xfffe7192df0e    SecError
  1   Security          0xfffe7189b668    SecCDSAKeyCopyPublicKey(OpaqueSecKeyRef*)
  2   Security          0xfffe71751f69    SecKeyCopyPublicKey
  3   dnd               0x107464fc1       SecIdentity.deleteIdentity()
  4   dnd               0x10743d87d       DNDIdentity.deleteIdentity(deleteAssociatedCA:)
  5   Do Not Disturb    0x2073e765c       -[FrameworkInterface initIdentity:]
```

crash in Apple APIs ?

```
01  guard let key = privateKey,
02  if let pubKey = SecKeyCopyPublicKey(key) {
03      ....
04  }
```

DnD code
(occasionally) triggers bug

SecError Function

...garbage pointer?

```
Exception Type:  EXC_BAD_ACCESS (SIGSEGV)
Exception Codes: SEGV_NOOP at 0x0000000000000001

Attempted to dereference garbage pointer 0x1.
Thread 6 Crashed:
  0    Security    0xfffe7192df0e    SecError
```

SecError in Security.framework

```
01  SecError:
02       0xfffe7192deaf    mov         r11, rsi          ; 2nd arg into r11
03       0xfffe7192df06    test        r11, r11.         ; 2nd arg NULL? ...leave
04       0xfffe7192df09    je          leave
05
06       ...
07
08       0xfffe7192df0e    mov         rdx, qword [r11]  ; *(2nd arg)
```

faulting instruction

```
01  bool SecError(OSStatus status, CFErrorRef *error, CFStringRef format, ...) {
02  ...
03  if (error)
04      CFErrorRef previousError = *error;
```

SecError's source code
(instruction that triggers crash)

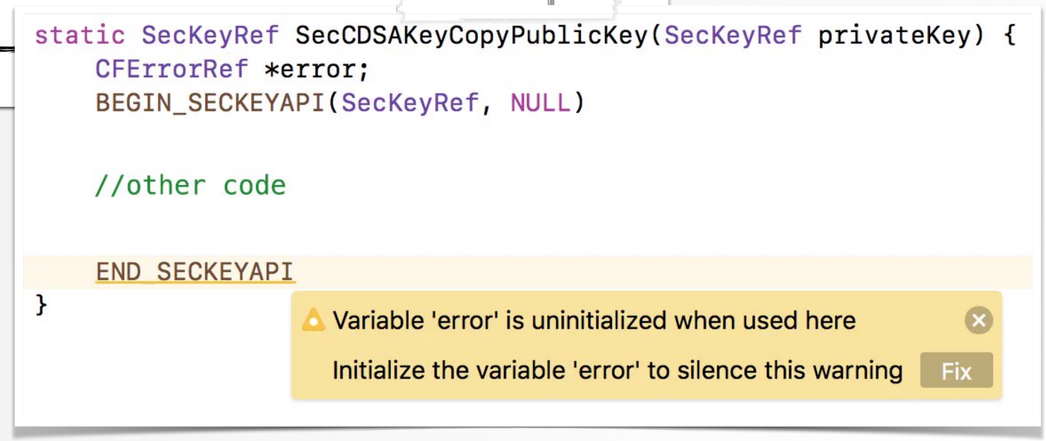
SecError Function

...uninitialized (pointer) variable

```
01  static SecKeyRef SecCDSAKeyCopyPublicKey(SecKeyRef privateKey) {
02
03      CFErrorRef *error;
04      BEGIN_SECKEYAPI(SecKeyRef, NULL)
05      ...
06      END_SECKEYAPI
```

```
01  #define END_SECKEYAPI }\
02    ...
03    catch (const CommonError &err) { \
04      if (err.osStatus() != CSSMERR_CSP_INVALID_DIGEST_ALGORITHM) { \
05        ...
06
07        SecError(status, error, CFSTR("%s"), ...
08    ...
```

As Xcode points out,
pointer is never initialized!



SecError Function

...fix/CVE, & (unofficial) Apple commentary 😅

- [x] **Fix + CVE**

> **Security**
>
> Impact: A local user may be able to cause a denial of service
>
> Description: This issue was addressed with improved checks.
>
> CVE-2018-4395: Patrick Wardle of Digita Security

- [x] **Blog post**



"An Insecurity in Apple's Security Framework"
https://objective-see.org/blog/blog_0x2D.html

- [x] **An Apple employee weighs in**



CRASH: KERNEL

...via RansomWhere?

```
*** Panic Report ***
panic(cpu 1 caller 0xffffff8008605ecd):
Kernel trap at 0xffffff800892544b, type 14=page fault

registers:
CR0: 0x0000000080010033, CR2: 0xffffff803db4f000, CR3: 0x000000044d97f05c, CR4: 0x00000000001626e0
RAX: 0x0000000000000001, RBX: 0xffffff803db4eff0, RCX: 0x0000000000000000, RDX: 0x0000000000000010
RSP: 0xffffff9222ac3d20, RBP: 0xffffff9222ac3e60, RSI: 0xffffff803db4f000, RDI: 0xffffff803433a2e8
R8: 0x0000000000000000, R9: 0xffffff80448b16e8, R10: 0x0000700001f2f4c0, R11: 0xffffff802f59d4e8
R12: 0xffffff802813e458, R13: 0x000000000000000e, R14: 0xffffff8034339db0, R15: 0x1575312836070096
RFL: 0x0000000000010202, RIP: 0xffffff800892544b, CS: 0x0000000000000008, SS: 0x0000000000000010
Fault CR2: 0xffffff803db4f000, Error code: 0x0000000000000000, Fault CPU: 0x1, PL: 0, VF: 1

Backtrace (CPU 1), Frame : Return Address
0xffffff9222ac39b0 : 0xffffff80084f210c
0xffffff9222ac3a30 : 0xffffff8008605ecd
0xffffff9222ac3c10 : 0xffffff80084a3743
0xffffff9222ac3c30 : 0xffffff800892544b
0xffffff9222ac3e60 : 0xffffff80089de83b
0xffffff9222ac3ee0 : 0xffffff80089de47a
0xffffff9222ac3f50 : 0xffffff8008a273ea
0xffffff9222ac3fb0 : 0xffffff80084a3f46

BSD process name corresponding to current thread: syslogd
```

triggered via **syslog!?**

CRASH: KERNEL

...checking socket path?

```
*** Panic Report ***
panic(cpu 1 caller 0xffffff8008605ecd):
Kernel trap at 0xffffff800892544b, type 14=page fault
```

```
01  audit_arg_sockaddr:
02
03   ...
04   0xffffff800892544b      cmp byte ptr[rbx+r13+2], 0
```

mapping disasm
to the source code

```
01  void audit_arg_sockaddr(struct kaudit_record *ar, struct vnode *cwd_vp, struct sockaddr *sa) {
02
03  int slen;
04  struct sockaddr_un *sun;
05
06  bcopy(sa, &ar->k_ar.ar_arg_sockaddr, sa->sa_len);
07  switch(sa->sa_family) {
08
09    case AF_UNIX:
10        sun = (struct sockaddr_un *)sa;
11        slen = sun->sun_len - offsetof(struct sockaddr_un, sun_path);
12
13        if (sun->sun_path[slen] != 0) {
14          ....
```

Crash: Kernel

...register values

```
*** Panic Report ***

registers:
RAX: 0x0000000000000001, RBX: 0xffffff803db4eff0
...
R12: 0xffffff802813e458, R13: 0x000000000000000e

Fault CR2: 0xffffff803db4f000
```

```
cmp byte ptr[rbx+r13+2], 0
in C: if(sun->sun_path[slen] == 0)
```

Reg.	Value	Meaning
RBX	0xffffff803db4eff0	"sun": ptr to a sockaddr_un structure
R13	0x0e	"slen": length of the structure, minus path offset (2)

RBX + R13 + 2 =

0xffffff803db4eff0 + 0x0E + 0x2 = 0xffffff803db4f000

CRASH: KERNEL

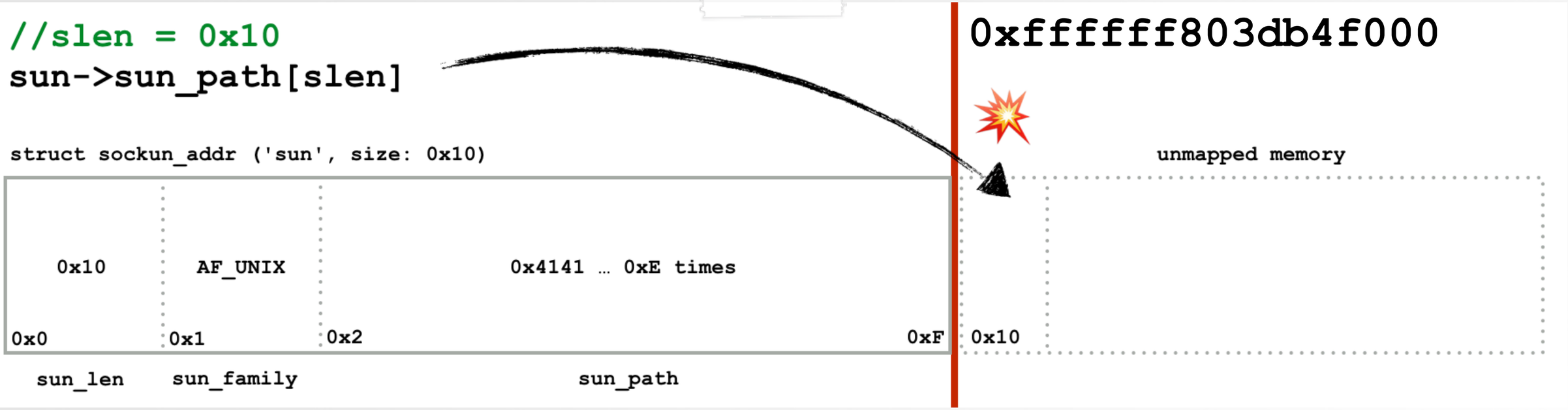
...off by one (into unmapped page)

```
*** Panic Report ***
...

Fault CR2: 0xffffff803db4f000
```

```
//slen = 0x10
sun->sun_path[slen]
```

0xffffff803db4f000

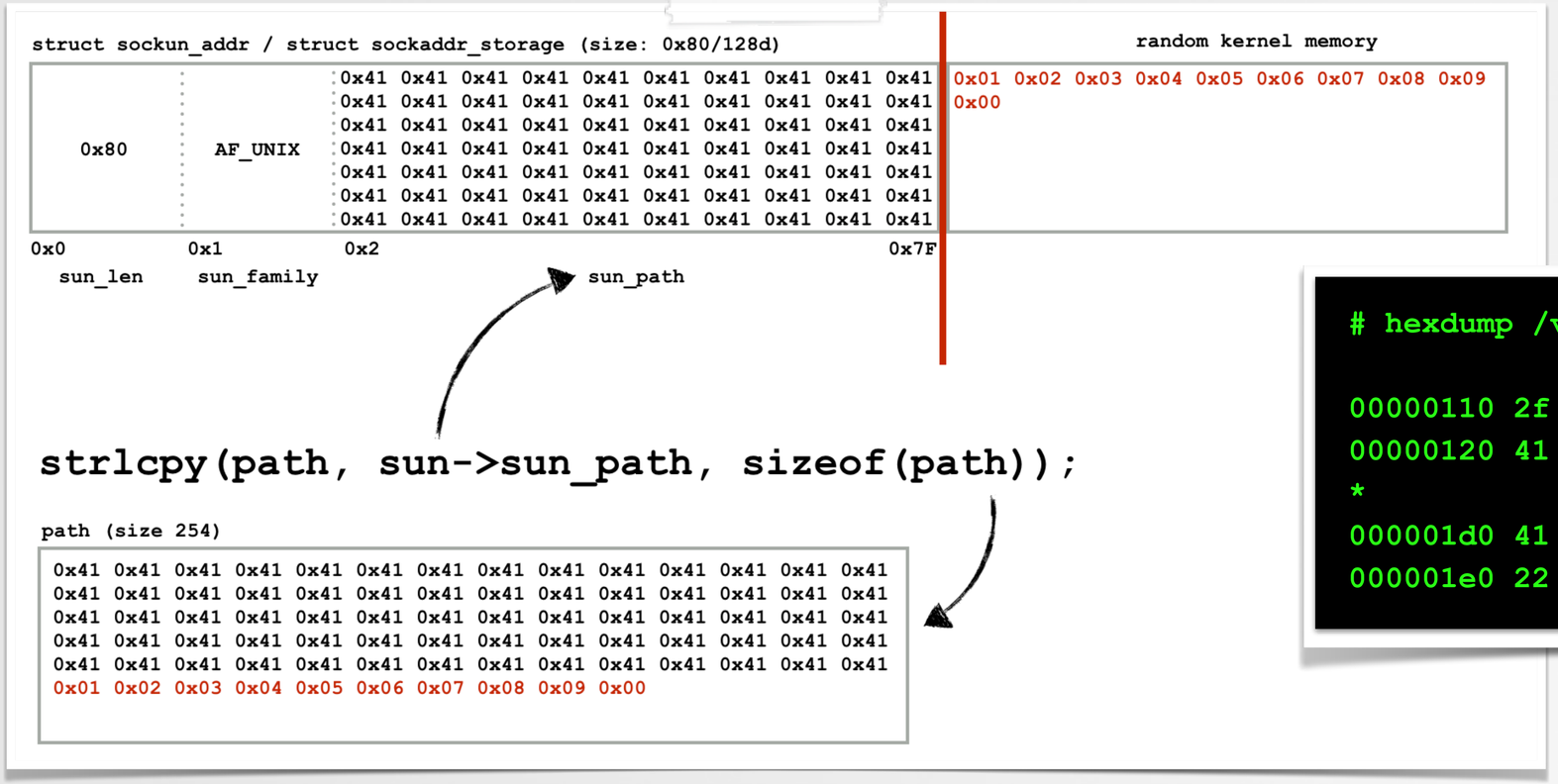


Off by One
(💥 when +1 hits an unmapped page)

CRASH: KERNEL

Apple's attempted "fix"

```
01  void audit_arg_sockaddr(struct kaudit_record *ar, struct vnode *cwd_vp, struct sockaddr *sa) {
02  ...
03    case AF_UNIX:
04      if (sun->sun_len > offsetof(struct sockaddr_un, sun_path)) {
05      /* Make sure the path is NULL-terminated */
06      strlcpy(path, sun->sun_path, sizeof(path));
07        ...
```




```
# hexdump /var/audit/20170406055225.not_terminated

00000110 2f 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
00000120 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
*
000001d0 41 41 41 41 41 41 41 90 99 0b 0f 07 54 38 c4 ba
000001e0 22 83 3b 9e 56 d5 e0 00
```

kernel memory leak!

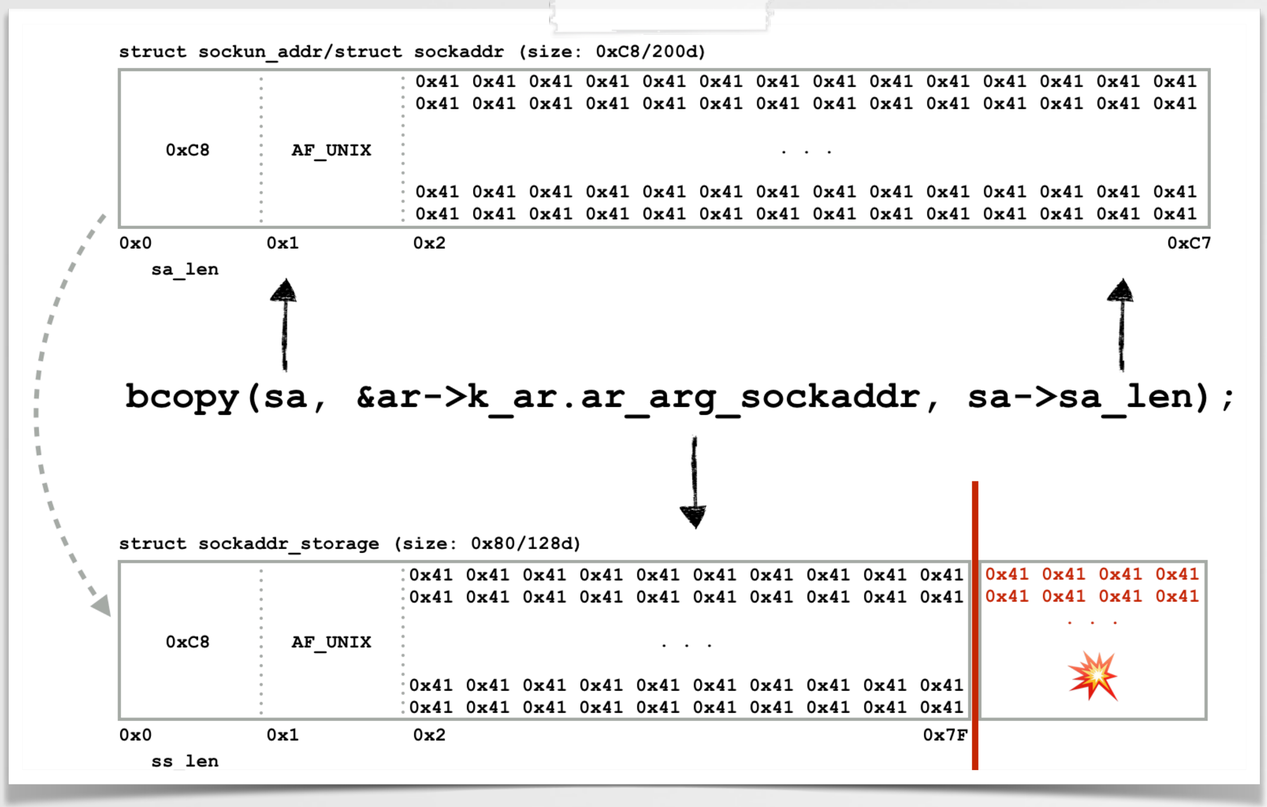
Another Bug Close By?

yes! a kernel heap overflow

```
01  void audit_arg_sockaddr(struct kaudit_record *ar, struct vnode *cwd_vp, struct sockaddr *sa) {
02  ...
03
04    bcopy(sa, &ar->k_ar.ar_arg_sockaddr, sa->sa_len);
05
06    case AF_UNIX:
```

128 bytes

can be > than 128 bytes!




```
[(lldb) x/100xb 0xffffff801a4c26f8
0xffffff801a4c26f8: 0xfa 0x01 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2700: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2708: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2710: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2718: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2720: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2728: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2730: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2738: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2740: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2748: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2750: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41
0xffffff801a4c2758: 0x41 0x41 0x41 0x41
[(lldb) x/i $pc
 ->  0xffffff80063eb6da: 48 8b 00  movq   (%rax), %rax
[(lldb) reg read $rax
     rax = 0x4141414141414141
```

The Fix

closed as "duplicate" ...no CVE

30824217 **Exploitable Kernel Heap Overflow: sockaddr_un bcopy'd into sockaddr_storage**

State: Open **Product:** macOS + SDK

Rank: No Value

Duplicate of 29173030 (*Closed*)

Add new information...

Attach a file Hide | Resolved Revert Save

Patrick Wardle

Area:
Something not on this list

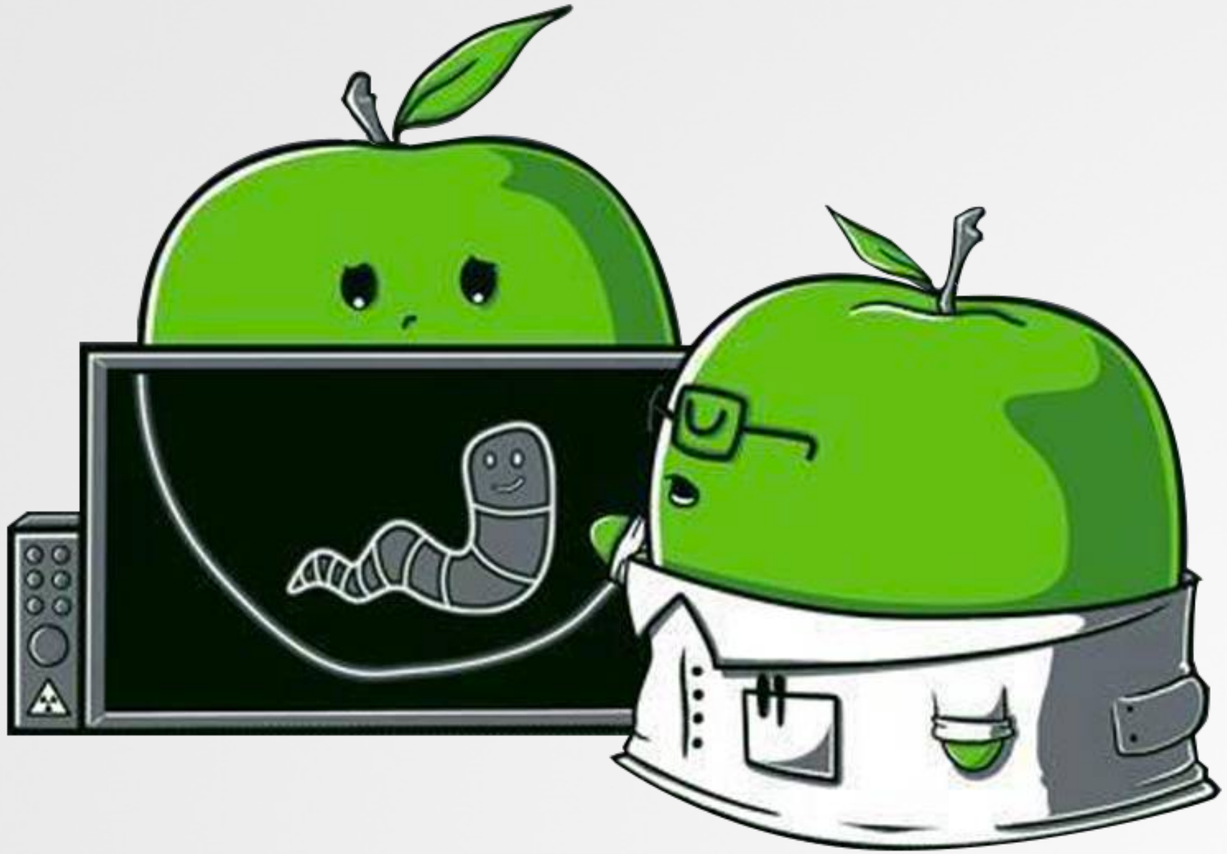
Summary:
The audit_arg_sockaddr() function (in bsd/security/audit/audit_arg.c) copies a struct sockaddr (passed in as a parameter, 'sa') into struct sockaddr_storage ('ar_arg_sockaddr'):

bcopy(sa, &ar->k_ar.ar_arg_sockaddr, sa->sa_len);

The problem is, the sockaddr_storage has a maximum size of 128 (see bsd/sys/socket.h)

Recent (unfixed) Crashes

...macOS Ventura (+ some, Sonoma)



Crash: Safari (WebKit)

WKPageCopyPageConfiguration

```
-------------------------------------

Translated Report (Full Report Below)

-------------------------------------
Process:                Safari [1105]
Path:                   /Volumes/VOLUME/*/Safari.app/Contents/

Exception Type:         EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:        KERN_INVALID_ADDRESS at 0x0000000000000068
Exception Codes:        0x0000000000000001, 0x0000000000000068

Crashed Thread:         0  Dispatch queue: com.apple.main-thread

Thread 0 Crashed::  Dispatch queue: com.apple.main-thread
0   WebKit  0x1adeb10c4 WKPageCopyPageConfiguration + 20
1   Safari  0x1b7dc0a4c Safari::WebArchiveGenerationController::websiteDataStoreForOriginalPage() const + 40
```

```
01  WKPageCopyPageConfiguration:
02  0x00000001adeb10b0    pacibsp
03  0x00000001adeb10b4    sub     sp, sp, #0x20
04  0x00000001adeb10b8    stp     fp, lr, [sp, #0x10]
05  0x00000001adeb10bc    add     fp, sp, #0x10
06  0x00000001adeb10c0    bl      objc_msgSend$_apiObject
07  0x00000001adeb10c4    ldr     x0, [x0, #0x68]
```

Faulting instruction (**0x1adeb10c4**):
an object returned by "**apiObject**"
triggers a crash when it is dereferenced.

Crash: Safari (WebKit)

NULL pointer dereference

```
01  WKPageCopyPageConfiguration:
02
03  ...
04  0x00000001adeb10c0          bl          _objc_msgSend$_apiObject
05  0x00000001adeb10c4          ldr         x0, [x0, #0x68]
06  0x00000001adeb10c8          add         x8, sp, #0x8
07  0x00000001adeb10cc          bl          API::PageConfiguration::copy()
```

```
Exception Type:        EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:       KERN_INVALID_ADDRESS at 0x0000000000000068

Thread 0 crashed with ARM Thread State (64-bit):
    x0: 0x0000000000000000   x1: 0x0000000000000000   x2: 0x000000016dc9d4c0   x3: 0x000000016dc9d4c8
    x4: 0x0000000000002800   x5: 0x0000000000000000   x6: 0x0000000000000000   x7: 0x0000000000000403
    ...
    far: 0x0000000000000068  esr: 0x92000006 (Data Abort) byte read Translation fault
```

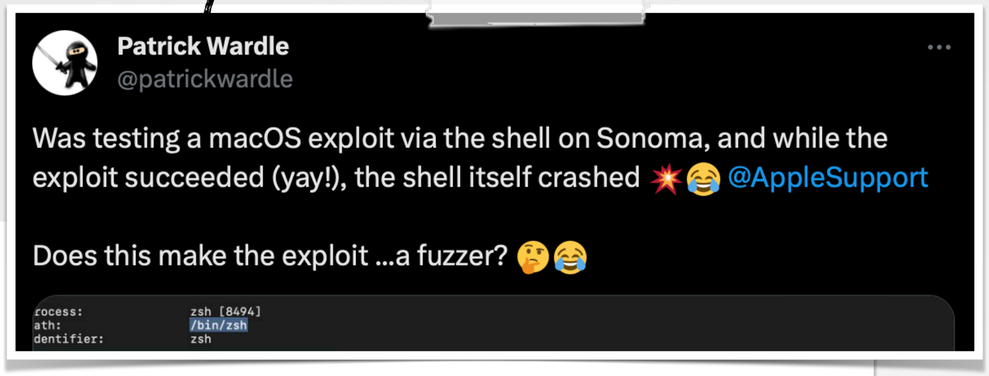
As x0 is NULL:

ldr x0, [x0, #0x68] → *(0x0 + 0x68) → 💥

Crash: Shell (zsh)

invalid memory access

exploit → shell crashes!?




```
----------------------------------------
Translated Report (Full Report Below)
----------------------------------------
Process:               zsh [17819]
Path:                  /bin/zsh

Crashed Thread:        0  Dispatch queue: com.apple.main-thread

Exception Type:        EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:       KERN_INVALID_ADDRESS at 0x0000c0dc0f430130 -> 0x000040dc0f430130
                                        (possible pointer authentication failure)
Exception Codes:       0x0000000000000001, 0x0000c0dc0f430130

Thread 0 Crashed::  Dispatch queue: com.apple.main-thread
0    zsh                                    0x1022b5a64 getlinknode + 28
1    zsh                                    0x1022ae518 deletefilelist + 36
2    zsh                                    0x1022ae248 deletejob + 28
3    zsh                                    0x1022adc98 printjob + 2388
4    zsh                                    0x1022ad074 update_job + 1328
5    zsh                                    0x1022dc2cc wait_for_processes + 840
6    zsh                                    0x1022dbbf4 zhandler + 328
7    libsystem_platform.dylib               0x18444ea24 _sigtramp + 56
```

signal handler

Crash: Shell (ZSH)

invalid "first node" pointer

```
01  getlinknode:
02  0x00000001022b5a48      pacibsp
03  0x00000001022b5a4c      stp     x20, x19, [sp, #-0x20]!
04  0x00000001022b5a50      stp     fp, lr, [sp, #0x10]
05  0x00000001022b5a54      add     fp, sp, #0x10
06  0x00000001022b5a58      mov     x8, x0
07  0x00000001022b5a5c      ldr     x0, [x0]
08  0x00000001022b5a60      cbz     x0, loc_1022b5a94
09  0x00000001022b5a64      ldr     x19, [x0, #0x10]
```

… to source code

```
01  mod_export void * getlinknode(LinkList list) {
02      void *dat;
03      LinkNode node;
04
05      if (!(node = firstnode(list)))
06          return NULL;
07
08      dat = node->dat;
```

```
Exception Type:         EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:        KERN_INVALID_ADDRESS at 0x0000c0dc0f430130 -> 0x000040dc0f430130
                                                (possible pointer authentication failure)
Exception Codes:        0x0000000000000001, 0x0000c0dc0f430130

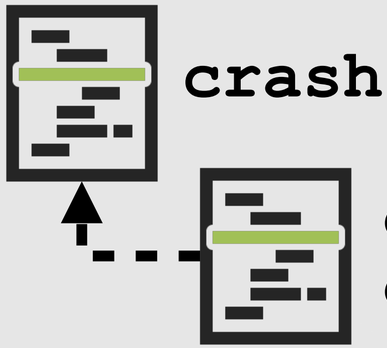
Thread 0 crashed with ARM Thread State (64-bit):
    x0: 0x0000c0dc0f430120   x1: 0x0000000000000000   x2: 0x0000000000000000   x3: 0xfffffffffffffff0
   ...
```

x0 + 0x10 =

0x0000c0dc0f430120 + 0x10 = 0x0000c0dc0f430130 → 

Crash: Xcode / Notarization Service

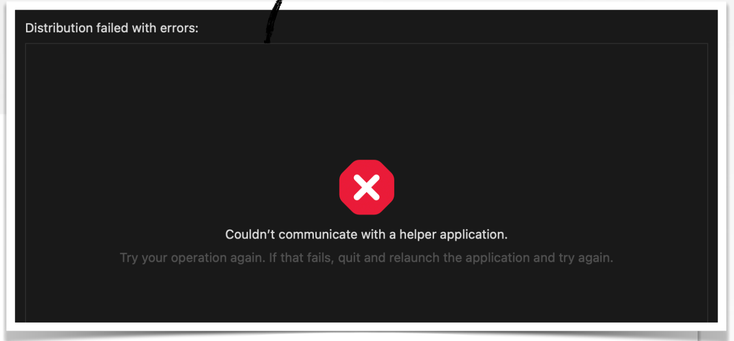
com.apple.dt.Xcode.ITunesSoftwareService.xpc



crash report: Xcode

crash report:
com.apple.dt.Xcode.ITunesSoftwareService

broke distribution




```
----------------------------------
Translated Report (Full Report Below)
----------------------------------

Process:     com.apple.dt.Xcode.ITunesSoftwareService [22094]
Path:       /Applications/Xcode.app/Contents/SharedFrameworks/DVTITunesSoftware.framework/Versions/A/
XPCServices/com.apple.dt.Xcode.ITunesSoftwareService.xpc/Contents/MacOS/
com.apple.dt.Xcode.ITunesSoftwareService

Crashed Thread:        2  Dispatch queue: com.apple.NSXPCConnection.user.anonymous.21620
Exception Type:        EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:       KERN_INVALID_ADDRESS at 0x0000cdaf567ea620 -> 0x00004daf567ea620
                                          (possible pointer authentication failure)

Exception Codes:       0x0000000000000001, 0x0000cdaf567ea620
```

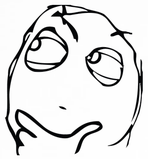
Crash: Notarization Service

com.apple.dt.Xcode.ITunesSoftwareService.xpc

```
Thread 2 Crashed::  Dispatch queue: com.apple.NSXPCConnection.user.anonymous.1014
0    libobjc.A.dylib                        0x18b311c20 objc_msgSend + 32
1    DVTITunesSoftwareServiceFoundation     0x102fd0cb8
     NotarizationWrapper.getStatus(submissionID:authenticationServiceIdentifier:accountCredentials:teamID:extraHeaders:) + 1344
2    DVTITunesSoftwareServiceFoundation     0x102fd141c @objc
     NotarizationWrapper.getStatus(submissionID:authenticationServiceIdentifier:accountCredentials:teamID:extraHeaders:) + 252
3    com.apple.dt.Xcode.ITunesSoftwareService     0x102e56ee4 -[DVTITunesSoftwareService
     notarizationStatusResponse:accountCredentials:teamID:authenticationServiceIdentifier:extraHeaders:callback:] + 260
```

```
Thread 2 Crashed::  Dispatch queue: com.apple.NSXPCConnection.user.anonymous.21620
0    libobjc.A.dylib    0x1806505f8 objc_release + 16
1    DVTITunesSoftwareServiceFoundation     0x102614cc8
     NotarizationWrapper.getStatus(submissionID:authenticationServiceIdentifier:accountCredentials:teamID:extraHeaders:) + 1360
2    DVTITunesSoftwareServiceFoundation     0x10261541c @objc
     NotarizationWrapper.getStatus(submissionID:authenticationServiceIdentifier:accountCredentials:teamID:extraHeaders:) + 252
3    com.apple.dt.Xcode.ITunesSoftwareService     0x10249aee4 -[DVTITunesSoftwareService
     notarizationStatusResponse:accountCredentials:teamID:authenticationServiceIdentifier:extraHeaders:callback:] + 260
```

...multiple (related) crashes
(**getStatus+0x1360** vs. **getStatus+0x1344**)
invoking **objc_msgSend** / **objc_release** ...same(?) object



Crash: Notarization Service

com.apple.dt.Xcode.ITunesSoftwareService.xpc

...in Obj-C

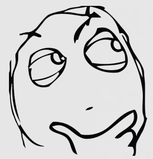
```
01  [someObj closeFile];
02
03  [someObj release];
```

```
01  DVTITunesSoftwareServiceFoundation.NotarizationWrapper.getStatus:
02
03  ...
04  0x0000000102fd0cac          ldr         x1, closefile       ;@selector(closeFile)
05  0x0000000102fd0cb0          mov         x0, x20             ;"instance" for objc_msgSend
06  0x0000000102fd0cb4          bl          objc_msgSend        ; objc_msgSend -> "closeFile"
07
08  ...
09  0x0000000102fd0cc0          mov         x0, x20             ;"instance" for objc_release
    0x0000000102fd0cc4          bl          objc_msgSend        ; objc_msgSend -> "objc_release"
```

Crash #1

Crash #2

crashes in DVTITunesSoftwareServiceFoundation's NotarizationWrapper.getStatus



Crashes in **objc_msgSend** & **objc_release** are (almost always) due to either a corrupted or an already freed object

Crash: Notarization Service

com.apple.dt.Xcode.ITunesSoftwareService.xpc

```
01  DVTITunesSoftwareServiceFoundation.NotarizationWrapper.getStatus:
02  ...
03  0x0000000102fd08a4    bl
04                         DVTITunesSoftwareServiceFoundation.NotarizationWrapper.majecticLogFileHandle()
05                         throws -> __C.NSFileHandle
06  ...
07
08  0x0000000102fd0ca0          ldur        x20, [x8, #-0x100]
09  0x0000000102fd0ca4          cbz         x20, loc_102fd1014
10  0x0000000102fd0cac          ldr         x1, =aClosefile      ;@selector(closeFile) for _objc_msgSend
11  0x0000000102fd0cb0          mov         x0, x20              ;"instance" for objc_msgSend
12  0x0000000102fd0cb4          bl          objc_msgSend         ; objc_msgSend -> "closeFile"
```

```
(lldb) x/i $pc
->  0x0000000102fd0cac: 0x580ce8a1   ldr    x1, #0x19d14  ; "closeFile"

(lldb) po $x0
<NSConcreteFileHandle: 0x6000000d4e00>
```

Object is a "majestic" file handle
(related to notarization log)

Crash: Notarization Service

com.apple.dt.Xcode.ITunesSoftwareService.xpc

Thread #2

```
Thread 2 Crashed::  Dispatch queue: com.apple.NSXPCConnection.user.anonymous.21620
0    libobjc.A.dylib    objc_release
1    DVTITunesSoftwareServiceFoundation NotarizationWrapper.getStatus(...)
...
3    com.apple.dt.Xcode.ITunesSoftwareService -[DVTITunesSoftwareService notarizationStatusResponse: ...] + 260
```



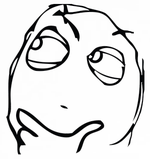
Thread #1

```
Thread 1::  Dispatch queue: com.apple.NSXPCConnection.user.anonymous.21620
...
6    DVTITunesSoftwareServiceFoundation    DVTNotarizationStatusResponse.description.getter + 1248
...
12   DVTFoundation   -[DVTLogAspect _logAtLogLevel:withFormat:] + 128
13   com.apple.dt.Xcode.ITunesSoftwareService -[DVTITunesSoftwareService notarizationStatusResponse: ...] + 484
```

Thread #4

```
Thread 4::  Dispatch queue: com.apple.NSXPCConnection.user.anonymous.21620
...
7    DVTITunesSoftwareServiceFoundation    DVTNotarizationStatusResponse.description.getter + 1248
...
13   DVTFoundation   -[DVTLogAspect _logAtLogLevel:withFormat:] + 128
14   com.apple.dt.Xcode.ITunesSoftwareService -[DVTITunesSoftwareService notarizationStatusResponse: ...] + 484
```

...other threads are concurrently executing logic within **notarizationStatusResponse** (and have already invoked **NotarizationWrapper.getStatus** calls **release** on the file handle).



Crash: Notarization Service

com.apple.dt.Xcode.ITunesSoftwareService.xpc

```
(lldb) b 0x0000000100ddafcc
Breakpoint 1:
DVTITunesSoftwareServiceFoundation.NotarizationWrapper.majecticLogFileHandle() throws -> __C.NSFileHandle

(lldb) c
Process stopped

* thread #2, queue = 'com.apple.NSXPCConnection.user.anonymous.645', stop reason = breakpoint 1.1
    frame #0: 0x0000000100ddafcc DVTITunesSoftwareServiceFoundation.NotarizationWrapper.majecticLogFileHandle

thread #3, queue = 'com.apple.NSXPCConnection.user.anonymous.645', stop reason = breakpoint 1.1
    frame #0: 0x0000000100ddafcc DVTITunesSoftwareServiceFoundation.NotarizationWrapper.majecticLogFileHandle
```

multiple threads
...hit (same) breakpoint at same time !

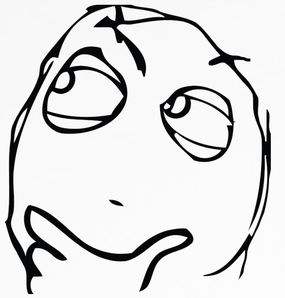
```
01  NotarizationWrapper.majecticLogFileHandle:
02  ...
03  type metadata accessor for __C.NSFileHandle();
04  r28 = @nonobjc __C.NSFileHandle.__allocating_init(r25);
05
```

over-released (?) file handle

And More

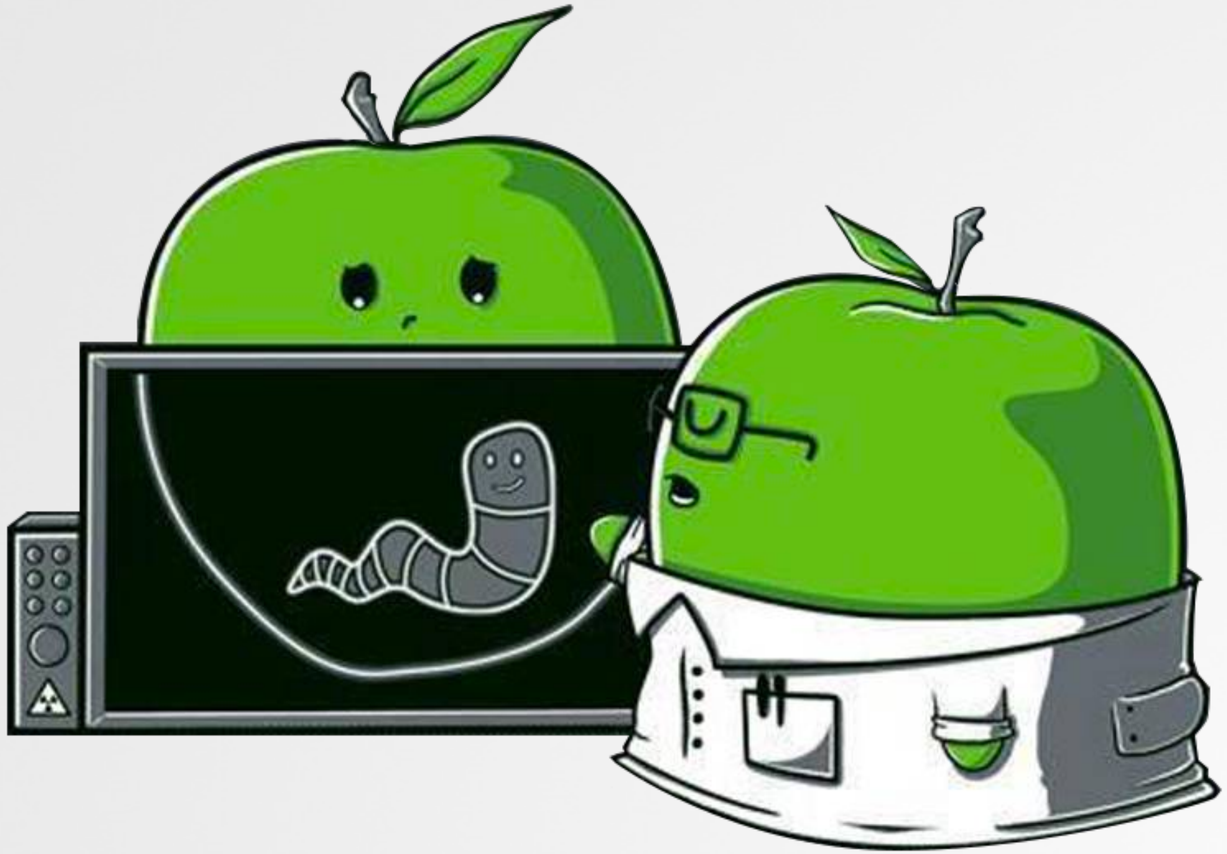
...so many more 😬

- bluetoothd-2023-09-10-175203.ips
- Console-2023-06-27-164823.ips
- Keynote-2023-08-02-103229.ips
- Messages-2023-06-13-115221.ips
- oahd-helper-2023-07-06-133104.000.ips
- QuickLookUIService-2023-06-16-154233.ips
- WindowServer-2023-06-12-084344.ips



Conclusions

...& take aways



Takeaways

Crash reports
are your friends!





Find bugs
...and fix || exploit



Uncover
malware & exploits

...so who loves crash reports now!?

