苹果攻击面和漏洞挖掘自动化研究

Lilang Wu

About Me

- Lilang Wu
  - 4 years security experience
  - macOS/iOS malware/vulnerability
  - Fuzzing project
  - BH USA 2019, 2018, BH EU 2018, HITB, CodeBlue

Agenda

- Static Analysis for Kernel Extensions Attack Interfaces
- PassiveFuzz & ActiveFuzz
- Vulnerabilities Found
- Conclusion

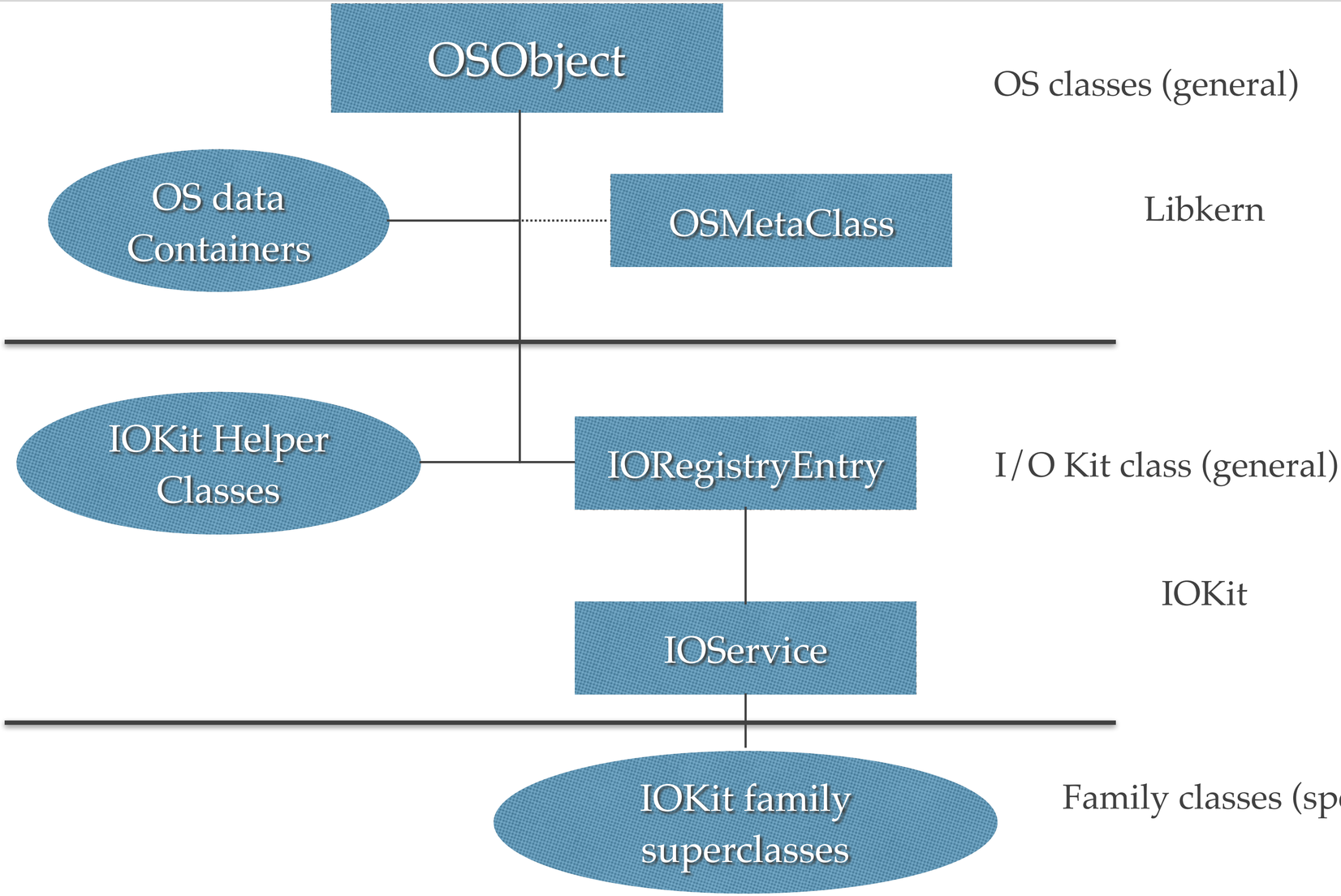
Agenda

- Static Analysis for Kernel Extensions Attack Interfaces
- PassiveFuzz & ActiveFuzz
- Vulnerabilities Found
- Conclusion

What is IOKit?

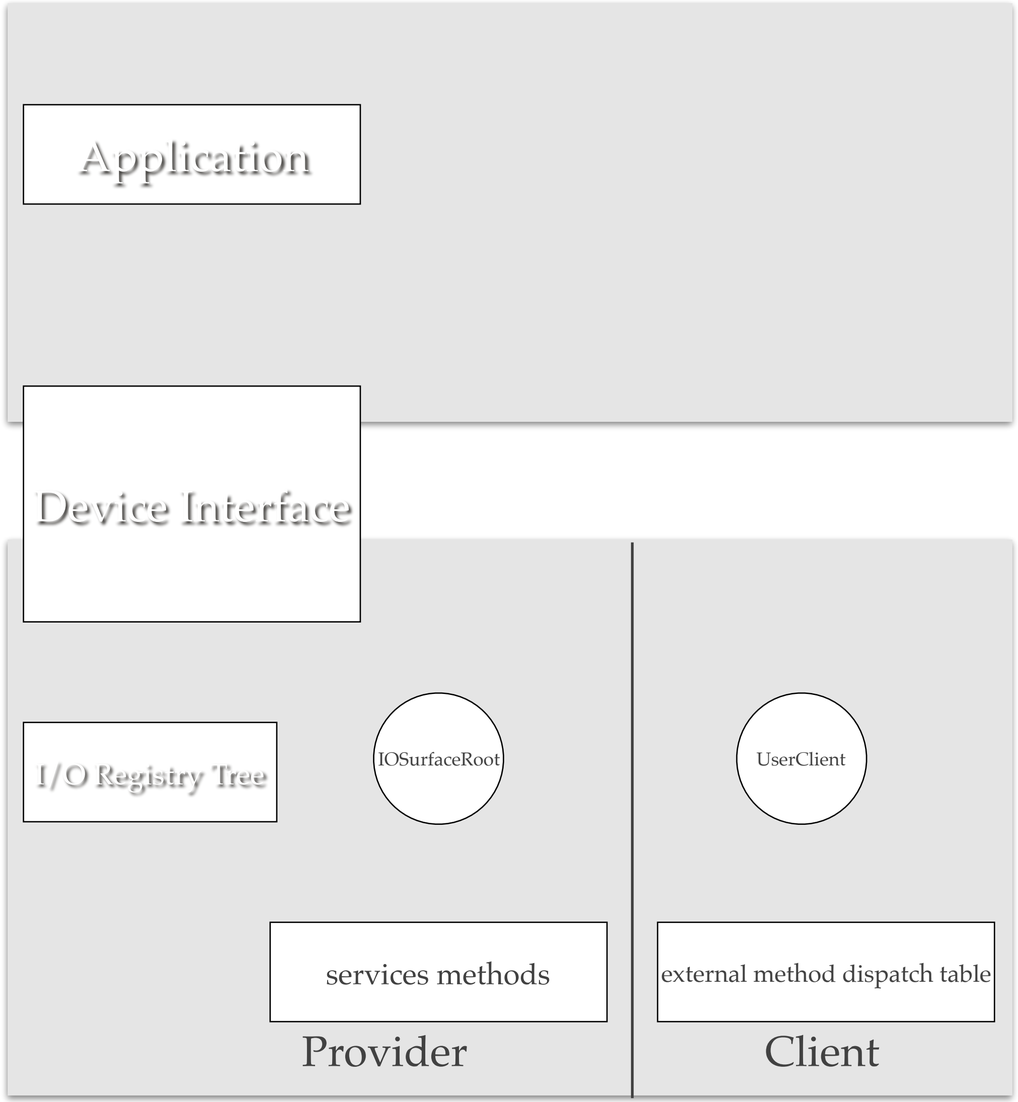
- The I/O Kit is a collection of system frameworks, libraries, tools, and other resources for creating device drivers in Apple system;
- modeling the hardware connected to an macOS system and abstracting common functionality for devices
- It is based on an object-oriented programming model implemented in a restricted form of C++;

I/O Kit Class Hierarchy



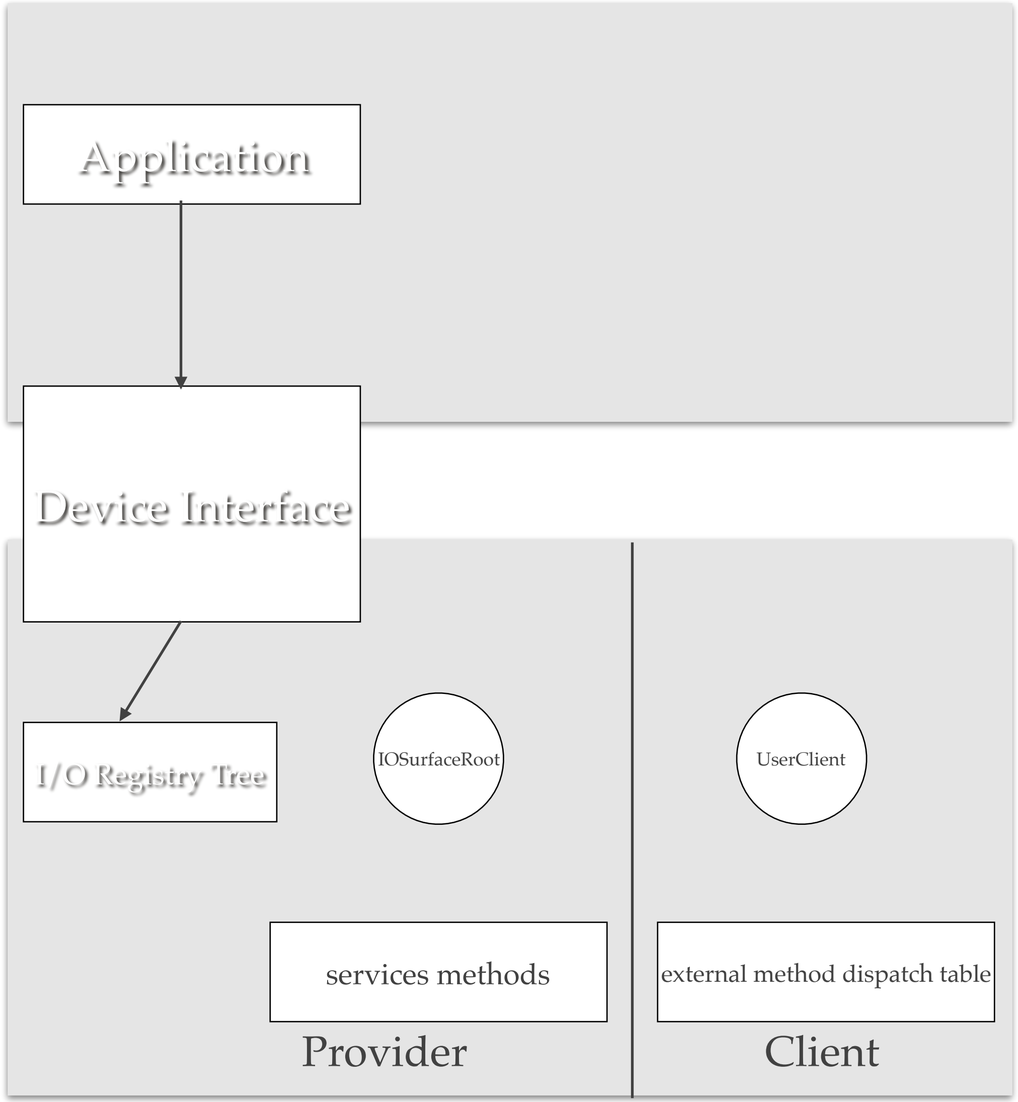
index	CanOpen	TOpenType	ServiceName	extends
0	False		IOSurfaceSharedEventNotification	OSObject::gMetaClass-->IOSurfaceSharedEventNotification
1	False		IOSurface	OSObject::gMetaClass-->IOSurface
2	False		IOSurfaceSharedEventNotificationPort	OSObject::gMetaClass-->IOSurfaceSharedEventNotificationPort
3	False		IOSurfaceSendRight	OSObject::gMetaClass-->IOSurfaceSendRight
4	False		IOSurfaceClient	OSObject::gMetaClass-->IOSurfaceClient
5	True	0	IOSurfaceRoot	IOService::gMetaClass-->IOSurfaceRoot
6	False		IOSurfaceSharedEventReference	OSObject::gMetaClass-->IOSurfaceSharedEventReference
7	False		IOSurfaceSharedEvent	OSObject::gMetaClass-->IOSurfaceEvent-->IOSurfaceSharedEvent
8	False		IOSurfaceRootUserClient	IOUserClient::gMetaClass-->IOSurfaceRootUserClient
9	False		IOSurfaceEvent	OSObject::gMetaClass-->IOSurfaceEvent
10	False		IOFence	OSObject::gMetaClass-->IOFence
11	False		IOSurfaceDeviceCache	OSObject::gMetaClass-->IOSurfaceDeviceCache

Users space



Kernel space

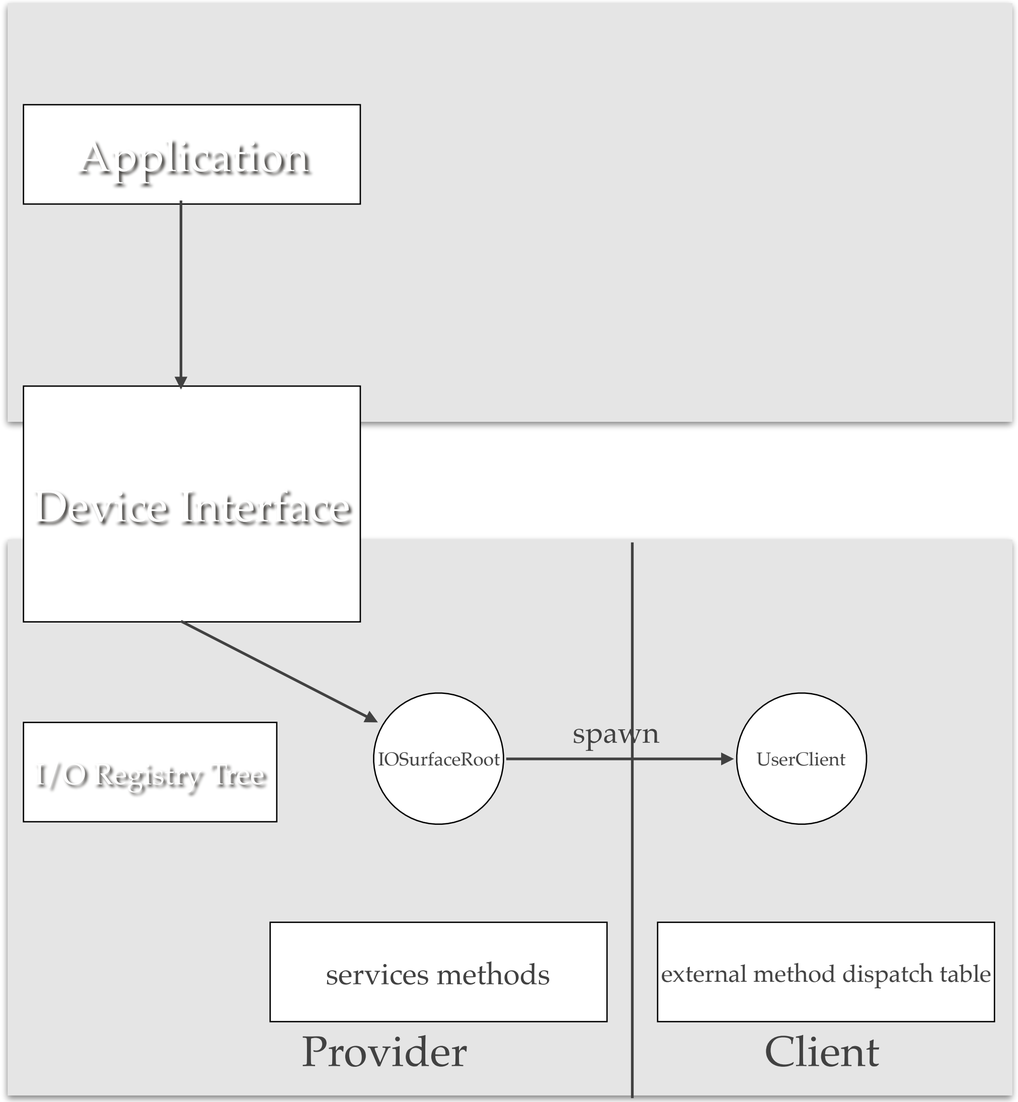
Users space



Kernel space

- IOServiceGetMatchingService(,class name)

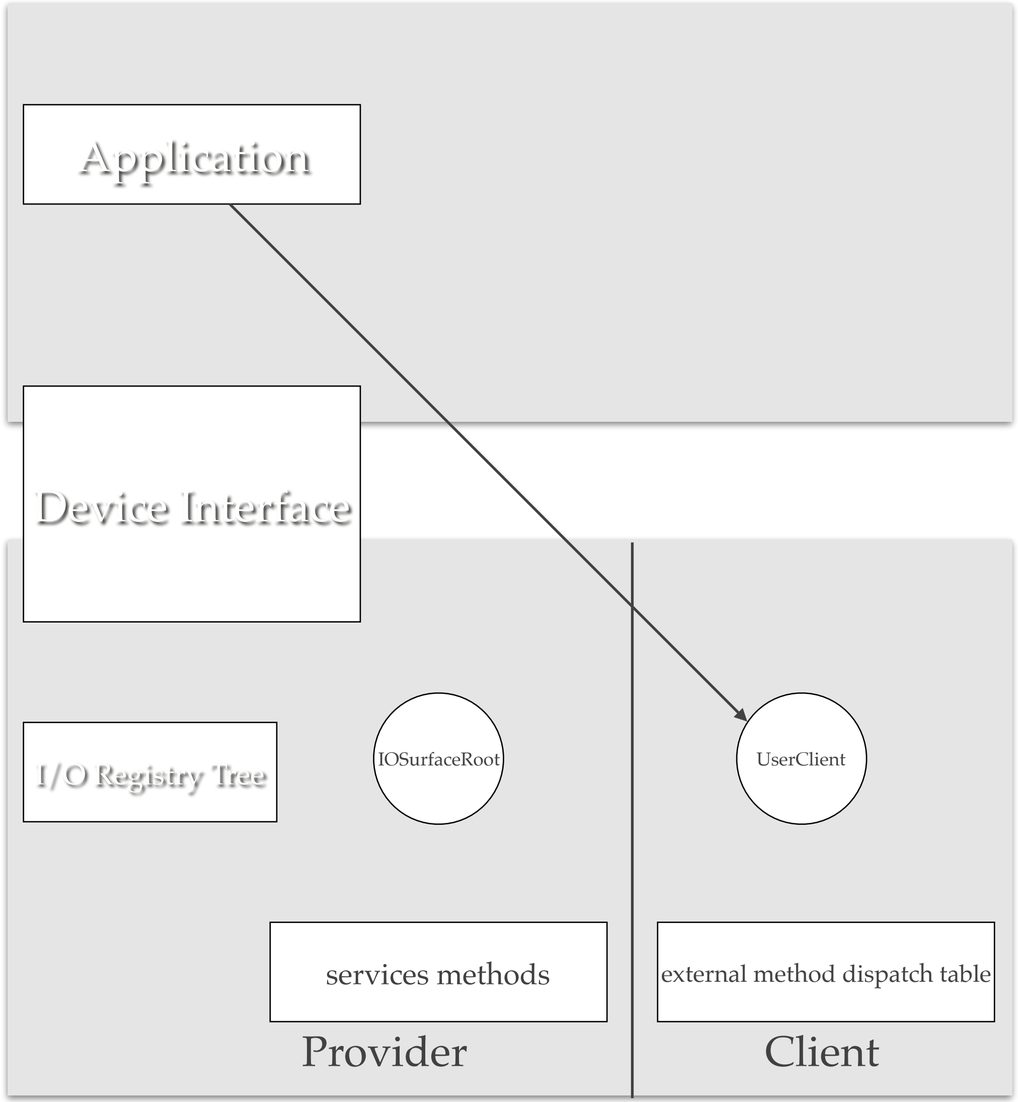
Users space



Kernel space

- IOServiceGetMatchingService(,class name)
- IOServiceOpen(service, , type, &conn)
  - newUserClient(, , , type)

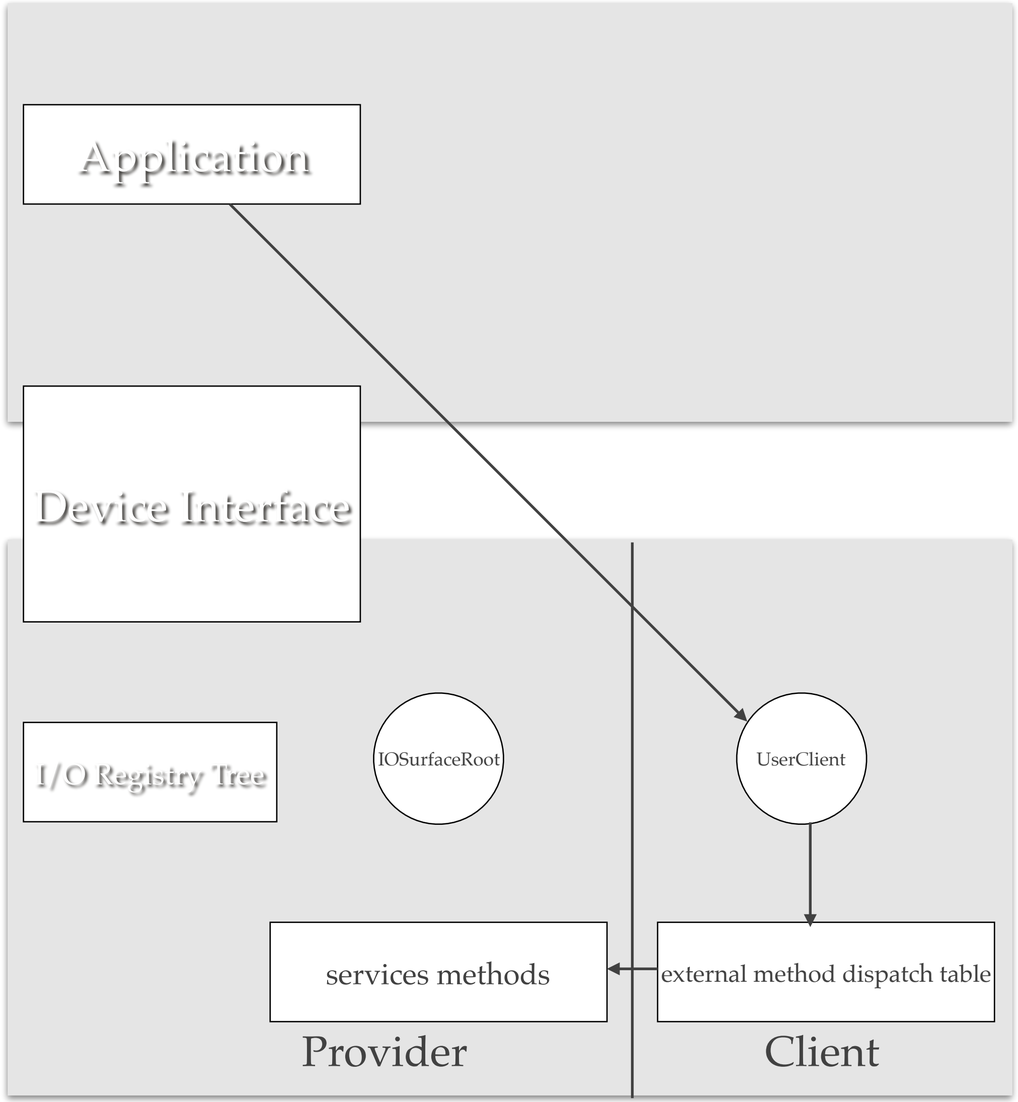
Users space



Kernel space

- IOServiceGetMatchingService(,class name)
- IOServiceOpen(service, , type, &conn)
  - newUserClient(, , , type)

Users space

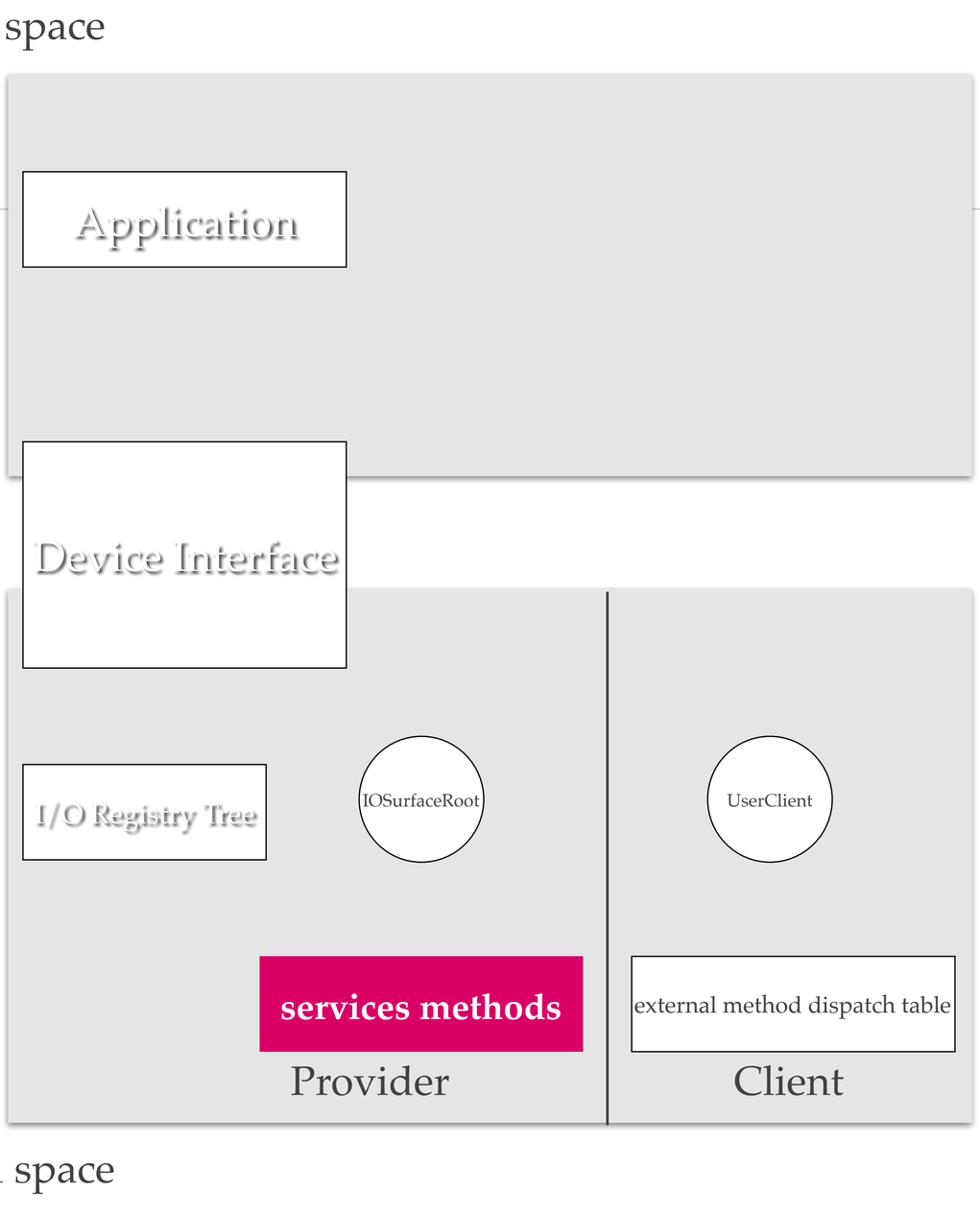


Kernel space

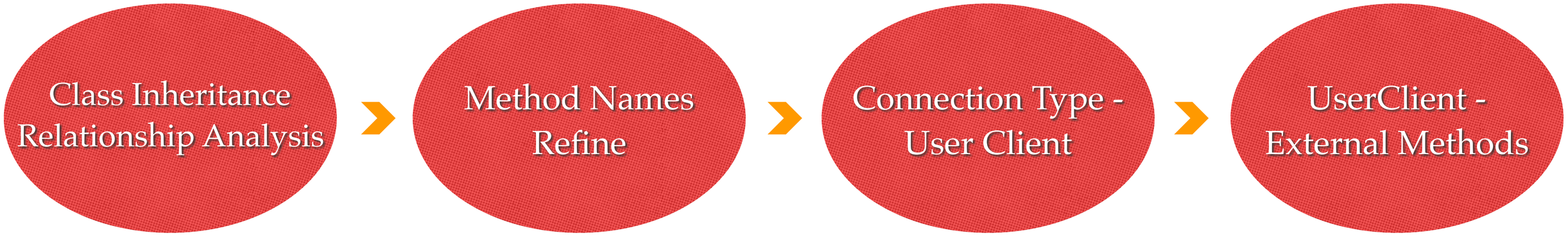
- IOServiceGetMatchingService(,class name)
- IOServiceOpen(service, , type, &conn)
  - newUserClient(, , , type)
- IOConnectCallMethod(conn, , , ,)
- IOConnectCallAsyncMethod(conn,)
  - XXuserclient::externalMethod
  - XXuserclient::getTargetAndMethodForIndex
  - XXuserclient::getAsyncTargetAndMethodForIndex

Fuzz What?

- Target Services
- Map<Connection Type, UserClient>
- Map<UserClient, [external methods]>



Kexts Interfaces Analysis Flow

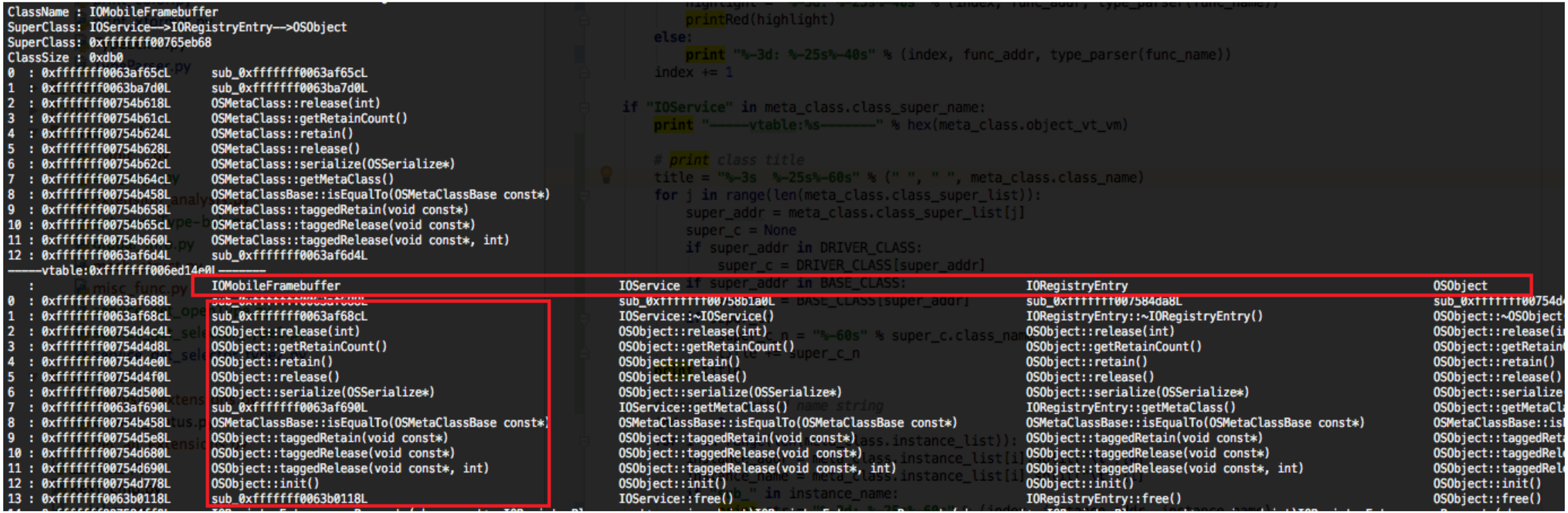


Class Inheritance Relationship

- rdi/x0: instance of register Meta class
- rsi/x1: Meta class name
- rdx/x2: instance of parent Meta class
- rcx/w3: size of register Meta class instance

```
__GLOBAL__sub_I_IOAccelMemory_cpp proc near
                                        ; DATA XREF: __mod_init_func:0000000000590E0↓o
                push    rbp
                mov     rbp, rsp
                lea     rdi, __ZN13IOAccelMemory10gMetaClassE ; IOAccelMemory::gMetaClass
                lea     rsi, aIoaccelmemory ; "IOAccelMemory"
                mov     rdx, cs:__ZN8OSObject10gMetaClassE_0 ; OSObject::gMetaClass
                mov     ecx, 0A0h ; '
                call    __ZN11OSMetaClassC2EPKcPKS_j ; OSMetaClass::OSMetaClass(char const*,OSMetaClass const*,uint)
                lea     rax, off_59550
                mov     cs:__ZN13IOAccelMemory10gMetaClassE, rax ; IOAccelMemory::gMetaClass
                pop     rbp
                retn
__GLOBAL__sub_I_IOAccelMemory_cpp endp
```

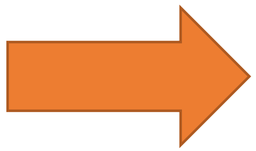
Method Name Refine

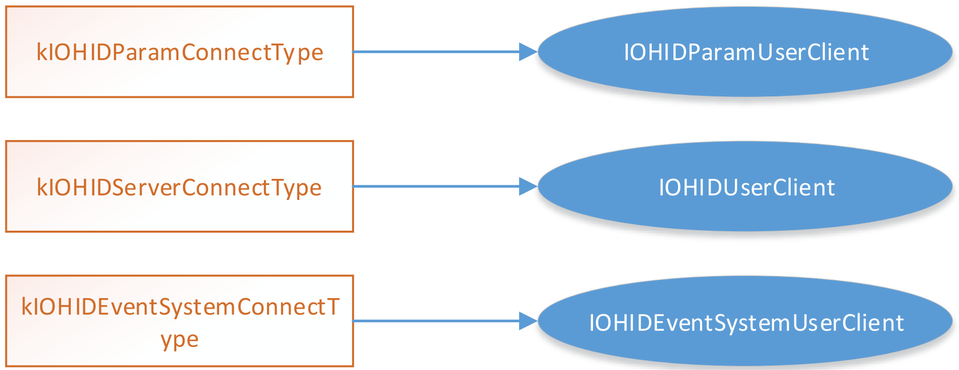


Connection Type - UserClients

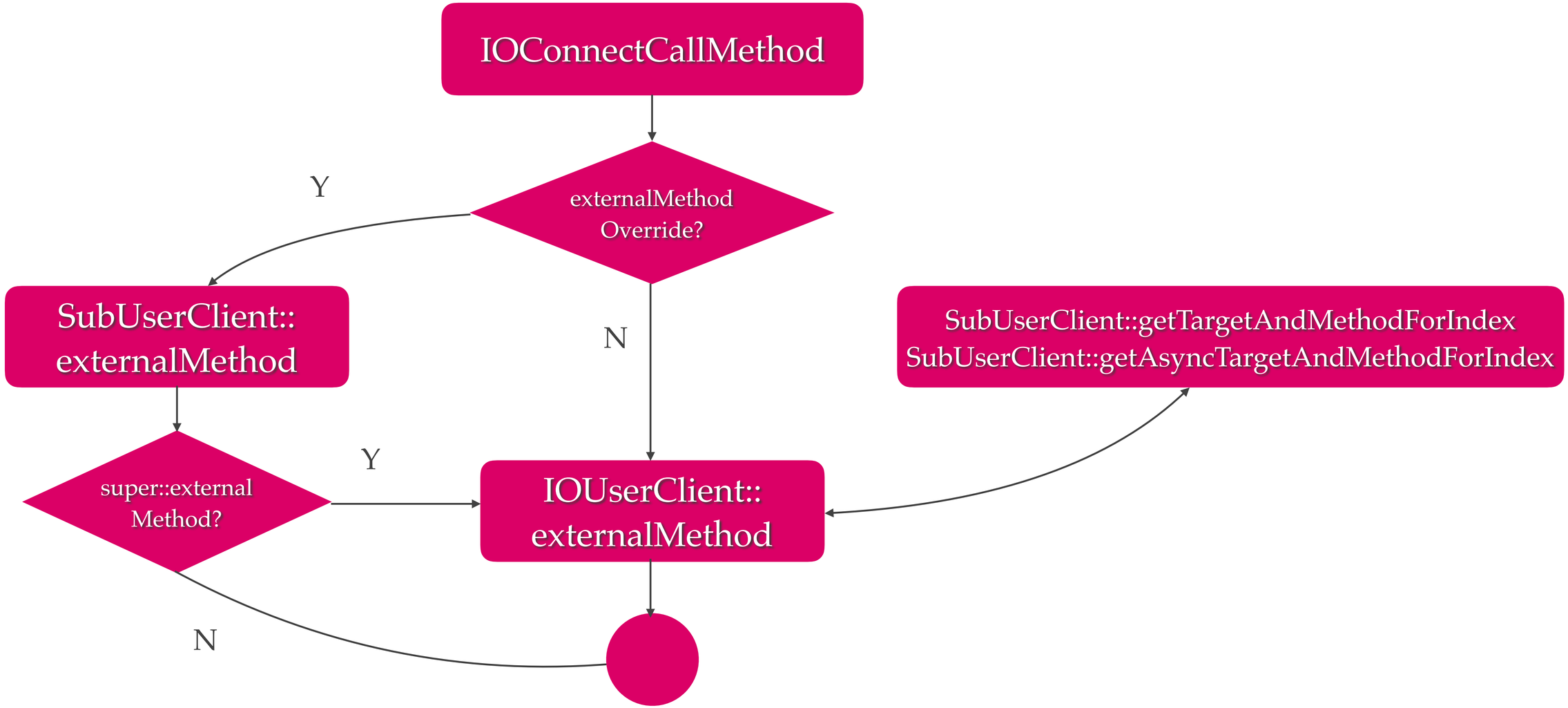
- Locate the newUserClient function address for IOServices
- Analyze the ASM instructions to enumerate the connection types
- Analyze the ASM instructions to get the corresponding user client for each connection type

```
do {
    if (type == kIOHIDParamConnectType) {
        if (eventsOpen) {
            newConnect = new IOHIDParamUserClient;
        } else {
            err = kIOReturnNotOpen;
            break;
        }
    }
    else if ( type == kIOHIDServerConnectType) {
        newConnect = new IOHIDUserClient;
    }
    else if ( type == kIOHIDStackShotConnectType ) {
        newConnect = new IOHIDStackShotUserClient;
    }
    else if ( type == kIOHIDEventSystemConnectType ) {
        newConnect = new IOHIDEventSystemUserClient;
    }
    else {
        err = kIOReturnUnsupported;
    }
```

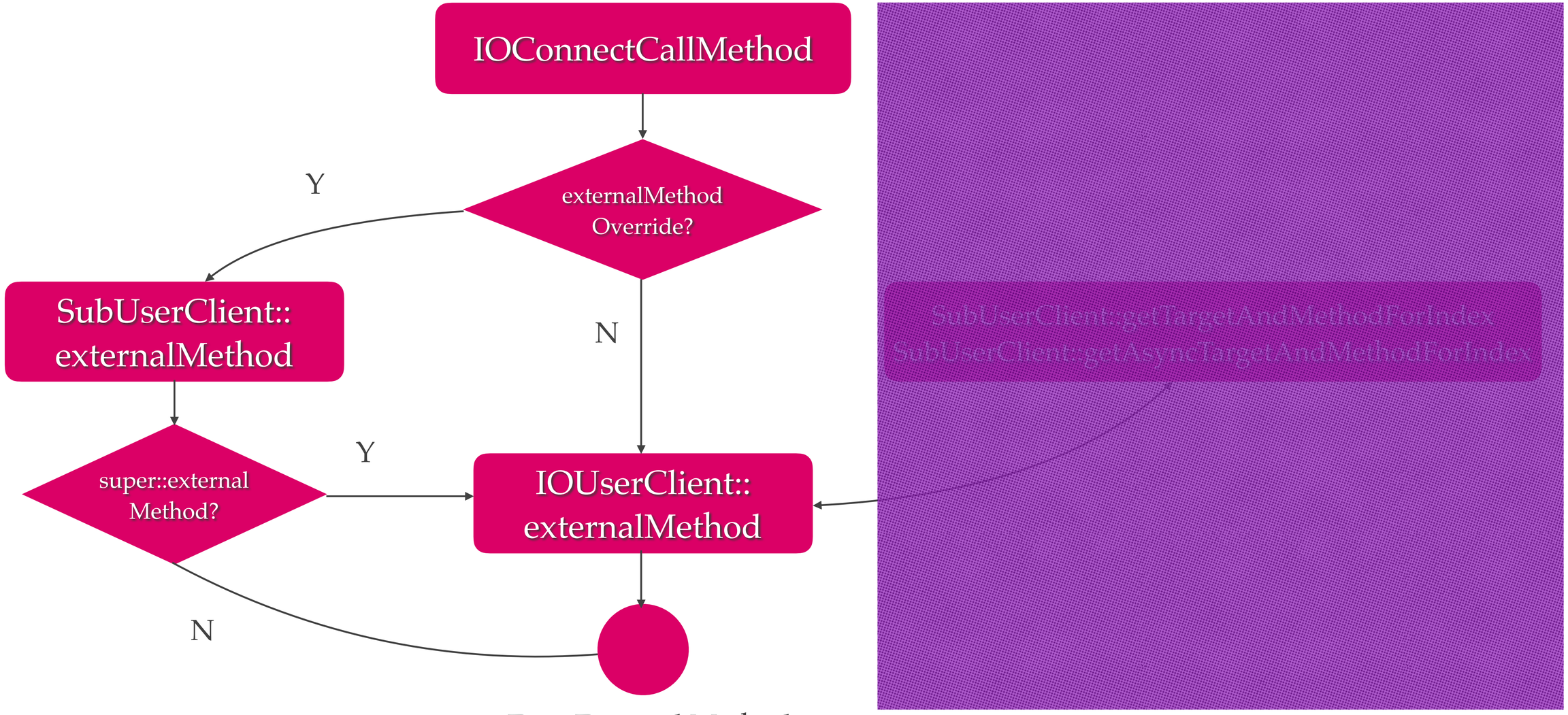




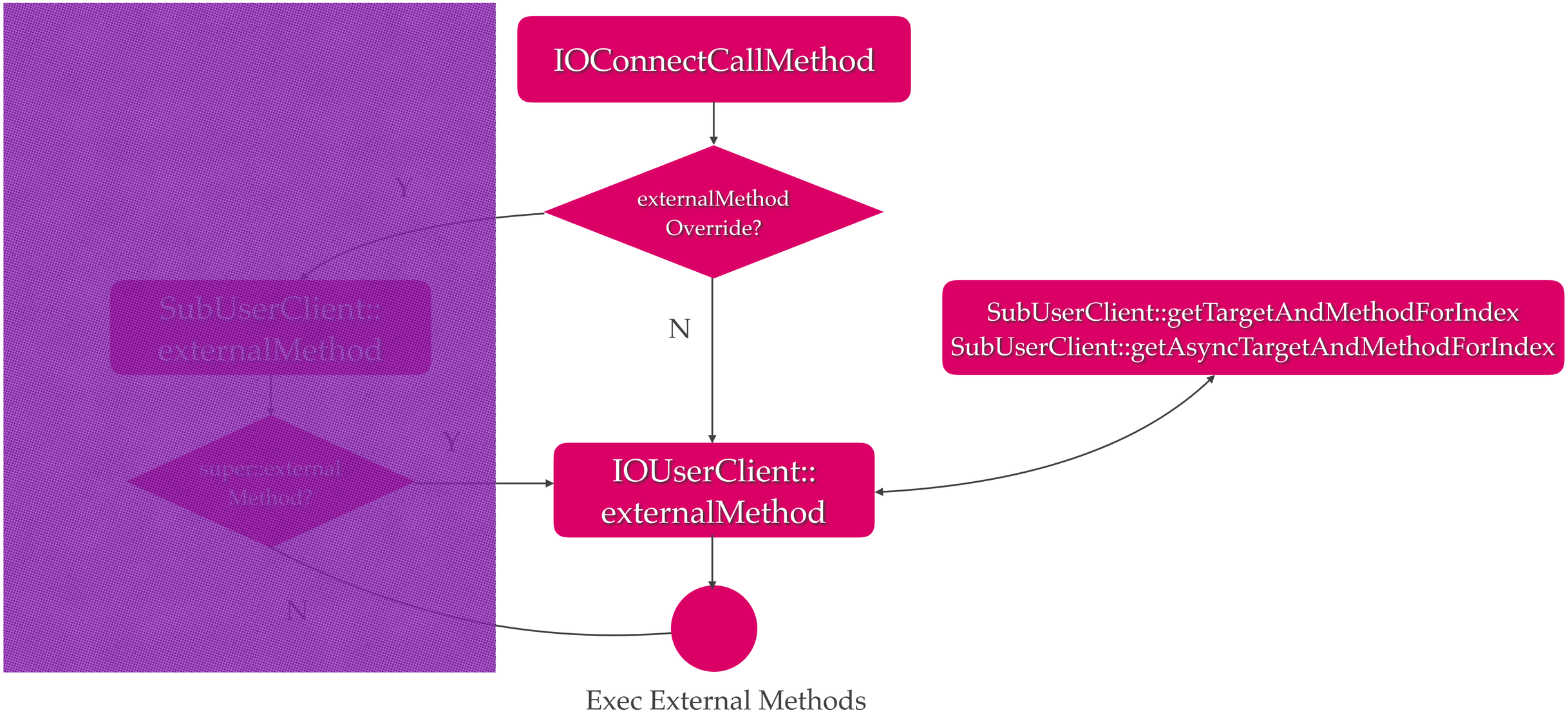
UserClient - External Methods



UserClient - External Methods



UserClient - External Methods



Two Graceful Implementation

```
struct IOExternalMethodDispatch
{
    IOExternalMethodAction function;
    uint32_t        checkScalarInputCount;
    uint32_t        checkStructureInputSize;
    uint32_t        checkScalarOutputCount;
    uint32_t        checkStructureOutputSize;
};
```

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

```
IOReturn IOHIDEventServiceUserClient::externalMethod(
                                uint32_t                    selector,
                                IOExternalMethodArguments * arguments,
                                IOExternalMethodDispatch *  dispatch,
                                OSObject *                  target,
                                void *                      reference)
{
    if (selector < (uint32_t) kIOHIDEventServiceUserClientNumCommands)
    {
        dispatch = (IOExternalMethodDispatch *) &sMethods[selector];

        if (!target)
            target = this;
    }

        return super::externalMethod(selector, arguments, dispatch, target, reference);
}
```

```
IOExternalMethod * IOI2CInterfaceUserClient::getTargetAndMethodForIndex(
    IOService ** targetP, UInt32 index )
{
    static const IOExternalMethod methodTemplate[] = {
                /* 0 */  { NULL, (IOMethod) &IOI2CInterfaceUserClient::extAcquireBus,
                           kIOUCScalarIScalarO, 0, 0 },
                /* 1 */  { NULL, (IOMethod) &IOI2CInterfaceUserClient::extReleaseBus,
                           kIOUCScalarIScalarO, 0, 0 },
                /* 3 */  { NULL, (IOMethod) &IOI2CInterfaceUserClient::extIO,
                           kIOUCStructIStructO, 0xffffffff, 0xffffffff },
        };

    if (index >= (sizeof(methodTemplate) / sizeof(methodTemplate[0])))
        return (NULL);

    *targetP = this;
    return ((IOExternalMethod *)(methodTemplate + index));
}
```

```
//============================================================================
// IOHIDEventServiceUserClient::sMethods
//============================================================================
const IOExternalMethodDispatch IOHIDEventServiceUserClient::sMethods[kIOHIDEventServiceUserClientNumCommands] = {
    { //    kIOHIDEventServiceUserClientOpen
        (IOExternalMethodAction) &IOHIDEventServiceUserClient::_open,
        1, 0,
    0, 0
    },
    { //    kIOHIDEventServiceUserClientClose
        (IOExternalMethodAction) &IOHIDEventServiceUserClient::_close,
        1, 0,
    0, 0
    },
    { //    kIOHIDEventServiceUserClientCopyEvent
        (IOExternalMethodAction) &IOHIDEventServiceUserClient::_copyEvent,
        2, -1,
    0, -1
    },
    { //    kIOHIDEventServiceUserClientSetElementValue
        (IOExternalMethodAction) &IOHIDEventServiceUserClient::_setElementValue,
        3, 0,
    0, 0
    },
};
```

Parse Method

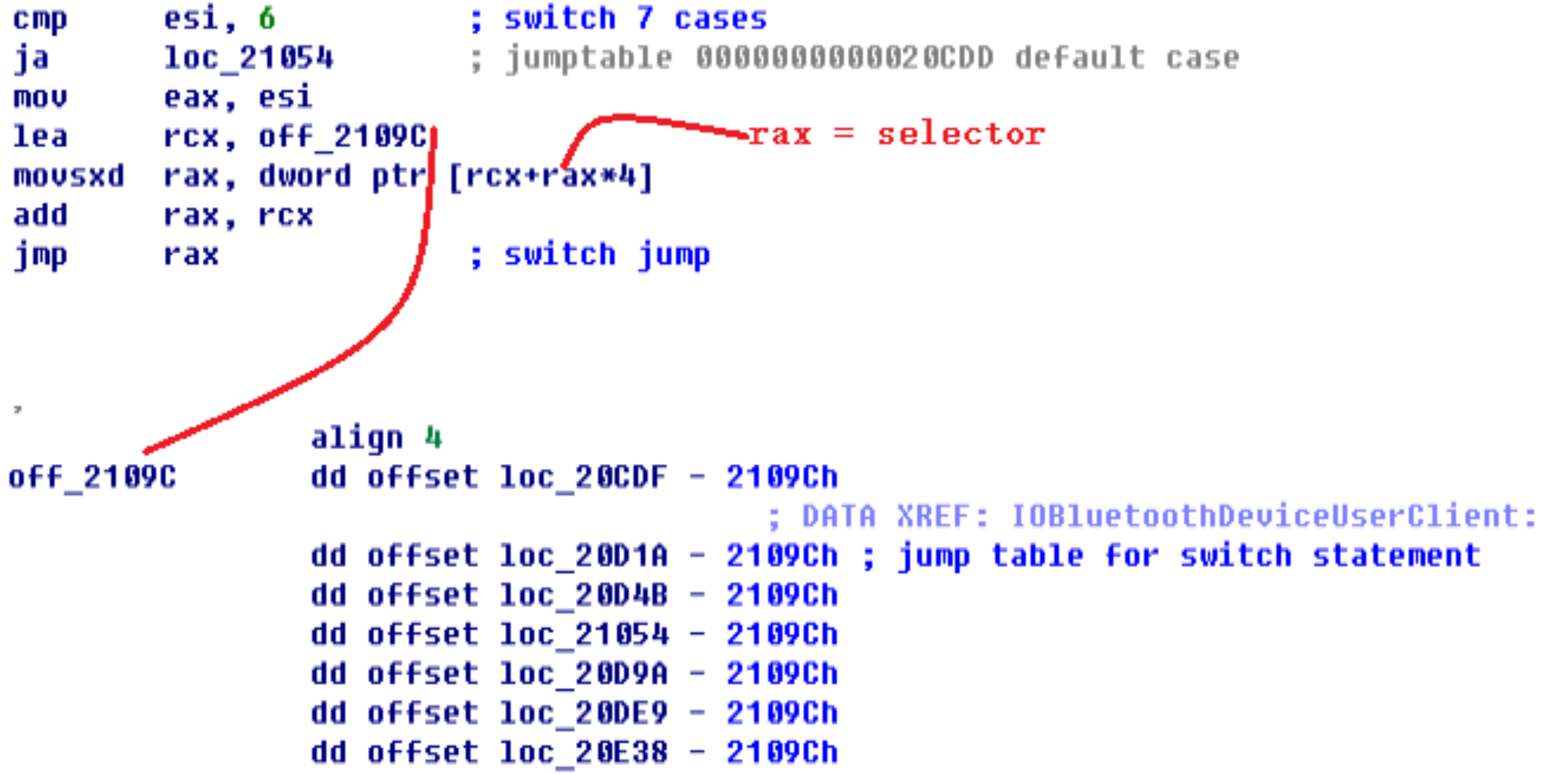
- Parse "Symbol Table“ section
- Search Constant Array name, shown as "String Table Index“
- Start with “__ZZN” or “__ZN”
- Locate the address, shown as "value"

```
String Table Index      __ZZN23IOFramebufferUserClient14externalMethodEjP25IOExternalMethodArgumentsP24IOExternalMethodDispatchP8OSObjectPvE14methodTemplate
Type
0E                      N_SECT
Section Index           7 (__DATA,__const)
Description
Value                   205360 ($+41072)
```

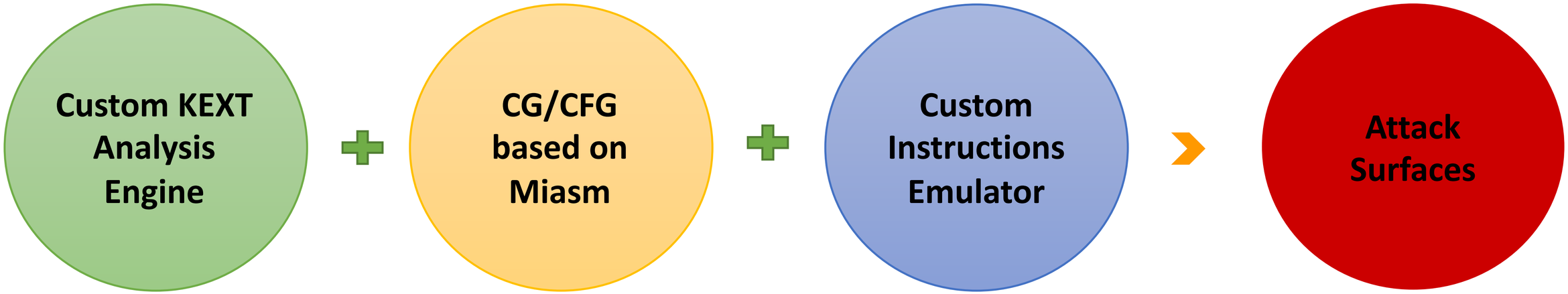
```
000627A0  0000C05F              String Table Index      __ZN18IOHIDLibUserClient8sMethodsE
000627A4  0F                    Type
                                0E                      N_SECT
                                01                      N_EXT
000627A5  08                    Section Index           8 (__DATA,__const)
000627A6  0000                  Description
000627A8  0000000000042F10      Value                   274192 ($+11216)
```

The Ugly Implementation

- Locate the address of override externalMethod Function
- Analyze the ASM instructions to get selector and external methods



Analyze the ASM Instructions



Custom KEXTs Analysis Engine

- MachOHeader(object)
  - __init__(self, fh, offset, size)
  - get_driver_list(self)
  - __parser_driver_dict(self, bundle)
  - macho_get_vmaddr(self, segname, sectname)
  - macho_get_fileaddr(self, segname, sectname)
  - macho_get_size(self, segname, sectname)
  - macho_get_loadcmds(self)
  - memcpy(self, start_fileaddr, size)
  - get_mem_from_vmaddr(self, anchor_f, anchor_vm, src_vm)
  - get_memStr_from_vmaddr(self, anchor_f, anchor_vm, src_vm
  - get_memStr_from_f(self, file_off)
  - get_f_from_vm(self, anchor_f, anchor_vm, src_vm)
  - get_vm_from_f(self, anchor_f, anchor_vm, src_f)
  - get_prelinkf_from_vm(self, src_vm)
  - get_prelinkvm_from_f(self, anchor_vm, anchor_f, src_f)
  - MH_MAGIC
  - endian
  - fh
  - kernel_header
  - mach_header
  - offset
  - prelink_offset
  - size
  - sizediff

- KernelMachO(object)
  - __init__(self, filename=None, base_addr=0xffffff007004(
  - load(self, fh)
  - load_fat(self, fh)
  - load_header(self, fh, offset, size)
  - get_section_addrs(self)
  - get_other_addrs(self)
  - get_driver_list(self)
  - extract_kext(self, bundleID=None, dir=None)
  - __construct_kext(self, bundle, offset, prelink_offset, dir)
  - __dump_kext_data(self, fd, fh_offset, data_size, fd_offset)
  - __parser_driver_dict(self, bundle)
  - base_addr
  - driver_list_notprelink
  - driver_list_prelink
  - fat
  - filename
  - headers

- OSMetaClass(object)
  - __init__(self)
  - IOExternalAsyncMethod
  - IOExternalMethod
  - IOExternalMethodDispatch
  - can_ser_open
  - can_ser_open_type
  - class_name
  - class_self_addr
  - class_size
  - class_super_addr
  - class_super_list
  - class_super_name
  - extends_list
  - externalMethod_f
  - externalMethod_vm
  - getAsyncTargetAndMethodForIndex_f
  - getAsyncTargetAndMethodForIndex_vm
  - getTargetAndMethodForIndex_f
  - getTargetAndMethodForIndex_vm
  - getTargetAndTrapForIndex_f
  - getTargetAndTrapForIndex_vm
  - havePublishedResource
  - instance_list
  - is_ioeam
  - is_ioem
  - is_ioemd
  - metaclass_list
  - metaclass_vt_f
  - metaclass_vt_vm
  - newUserClient_f
  - newUserClient_vm
  - object_vt_f
  - object_vt_vm

Custom KEXTs Analysis Engine

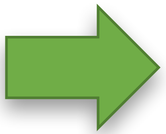
- Parse the Kexts MachO structure
- Arbitrary memory read with the virtual/file address
- Parse all basic information for each IOService sub classes
- Parse all virtual/file address for all critical functions
- Judge whether the service can be opened and its open type if yes
- …

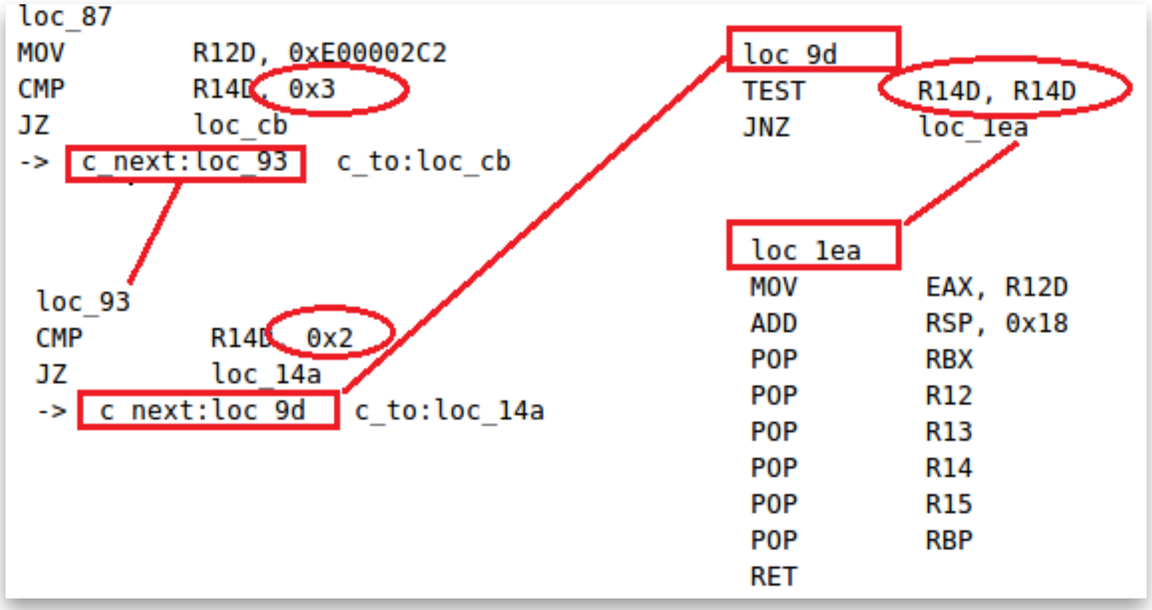
Generate CFG using Miasm

AppleHDAEngine::newUserClient(, , type,)

```
loc_2C1F1:                              ; CODE XREF: AppleHDA
                mov     r12d, 0E00002C2h
                cmp     r14d, 3
                jz      short loc_2C235
                cmp     r14d, 2
                jz      loc_2C2B4
                test    r14d, r14d
                jnz     loc_2C354
                mov     rax, cs:off_920E0
                xor     ecx, ecx
                mov     rdi, r13
                mov     rsi, r15
                mov     rdx, [rbp-40h]
                mov     r8, [rbp-30h]
                call    qword ptr [rax+788h]
                mov     r12d, eax
                jmp     loc_2C354
; ---------------------------------------------------------------------------

loc_2C235:                              ; CODE XREF: AppleHDA
                lea     rax, __ZN24AppleHDAEngineUserClient9
                mov     rdi, [rax]
                mov     rax, [rdi]
                call    qword ptr [rax+88h]
```

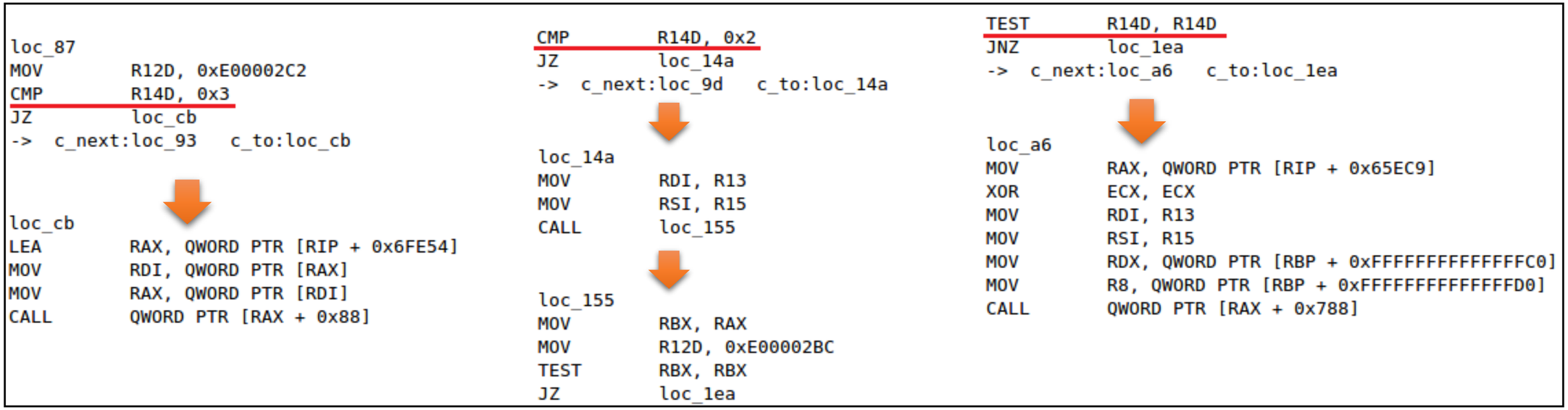




https://github.com/cea-sec/miasm

Analysis key paths based on CFG

- Key Paths based on Key registers
  - RCX register in “newUserClient“ function
  - RSI register in “externalMethod“ function
  - Tracking data flow between registers, as shown below, RCX move to R14D register



Custom Instruction Emulator

- ARM Emulator
  - adrp/adr, add, mov/mov
- nX86_64 Emulator
  - lea, mov, call, cmp, jz, je…

```
if not cmp(mnemonic, "str"):
    reg_num = insn.op_count(CS_OP_REG)
    if reg_num == 1:
        continue
    f_reg = get_first_reg(insn)
    if f_reg == arm64_const.ARM64_REG_XZR or f_reg == arm64
                f_reg == arm64_const.ARM64_REG_WZR:
        continue
    s_reg = get_second_reg(insn)

    if s_reg:
        s_reg_v = get_actual_value_by_regN(s_reg)
        if not (s_reg_v and s_reg_v == meta_class.class_sel
            continue
    else:
        continue

    f_reg_v_vm = get_actual_value_by_regN(f_reg)
    if iskext:
        f_reg_v_f = k_header.get_prelinkf_from_vm(f_reg_v_v
    else:
        f_reg_v_f = k_header.get_f_from_vm(each_mif_f, each

    parse_const_func(k_header, meta_class, f_reg_v_vm,
                     f_reg_v_f, iskext)
```

```
def get_single_IMM(insn):
    seg_num = insn.op_count(CS_OP_IMM)
    if seg_num > 1:
        print "Extract: too much imm reg!"
    if seg_num != 1:
        print "Extract: no imm reg found!"
    return to_x(insn.op_find(CS_OP_IMM, 1).value.imm)

def get_mem_op_offset(insn):
    mem_num = insn.op_count(CS_OP_MEM)
    if mem_num >= 1:
        offset = insn.op_find(CS_OP_MEM, 1).mem.disp
        return offset

def get_mem_op_reg(insn):
    mem_num = insn.op_count(CS_OP_MEM)
    if mem_num >= 1:
        offset = insn.op_find(CS_OP_MEM, 1).mem.base
        return offset

def get_first_reg(insn):
    return insn.op_find(CS_OP_REG, 1).value.reg


def get_second_reg(insn):
    return insn.op_find(CS_OP_REG, 2).value.reg
```

```
if not cmp(mnemonic, "bl"):
    if insn.op_count(CS_OP_IMM):
        bl_addr_vm = get_single_IMM(insn)
        meta_class = OSMetaClass()
        if bl_addr_vm == OSMetaClass_OSMetaClass_VMaddr:
            #meta_class = OSMetaClass()

  = get_actual_value_by_regN(arm64_const.ARM64_REG_X0)
  = get_actual_value_by_regN(arm64_const.ARM64_REG_X1)
r = get_actual_value_by_regN(arm64_const.ARM64_REG_X2)
  = get_actual_value_by_regN(arm64_const.ARM64_REG_X3)

ddr:
= k_header.get_memStr_from_vmaddr(each_mif_f, each_mif_vm, meta_class.class_name_addr)
class_name,
r = meta_cl
class_name,
 meta_class

= "unknow c
ss

6)
m_vm(each_m
r(k_header,

"L") == OSM

 = get_actu
 = get_actu
r = get_act
t_actual_va

ddr:
= k_header.

class_name,
r = meta_cl
class_name,
 meta_class

= "unknow classname"
```

```
from capstone import x86_const


class x_reg_manager(object):

    def __init__(self):
        self.x = [1]*234
        for i in range(234):
            self.x[i] = 0


    def get_actual_value_by_regN(self, reg):
        #global x0
        return self.x[reg]



    def set_actual_value_by_regN(self, reg, reg_val):
        self.x[reg] = reg_val
```

Attack Interfaces

AppleHDAEngine::newUserClient

```
_____________________________________________________

index   CanOpen TOpenType       ServiceName                                  extends
  4     True      0       AppleHDAEngineOutput                             IOAudioEngine::gMetaClass-->AppleHDAEngine-->AppleHDAEngineOutput
 86     True      0       AppleHDAEngine                                   IOAudioEngine::gMetaClass-->AppleHDAEngine

_____________________________________________________

ServiceName         OpenType          UserClient
AppleHDAEngine      0x3               AppleHDAEngineUserClient::metaClass
AppleHDAEngine      0x2               DspFuncUserClient::Create(IOAudioEngine*, task*)
```

AppleHDAEngineUserClient::externalMethod

```
-------------------------------------------------------------------------------------------------------------------
selector      cSIC            cSIS            cSOC            cSOS      func_name
  0              2               0               0            4095      AppleHDAEngineUserClient::getState
  1              2            4095               0               0      AppleHDAEngineUserClient::setState
  2              0               0               0               0      AppleHDAEngineUserClient::resetDSPToPropertyList
  3              1               0               1               0      AppleHDAEngineUserClient::isPortPresent
  4              0               0               6               0      AppleHDAEngineUserClient::getHardwareVolume
  5              1               0               0               0      AppleHDAEngineUserClient::setHardwareVolume
  6              0               0              16               0      AppleHDAEngineUserClient::getActiveSpatialChannels
  7              0               0               3               0      AppleHDAEngineUserClient::getAudioSnoopEnabled
  8              3               0               0               0      AppleHDAEngineUserClient::setAudioSnoopEnabled
  9              2               0               0               0      AppleHDAEngineUserClient::setSpatialChannelMute

Process finished with exit code 0
```

Shortage

- KEXTs are closed source, many method strings are stripped
- Function call usually use *(object_ptr + offset) type

```
v20 = (*(int (__fastcall **)(IORegistryEntry *, __int64, AMDRadeonX4000_AMDAccelResource *, _QWORD, _QWORD, _QWORD))(*(_QWORD *)this_ptr + 0xB70LL))(
        this_ptr,
        v2,
        accelResource_offset8,
        0LL,
        *((_QWORD *)this_ptr + 594),
        0LL);                           // AMDRadeonX4000_AMDSIGLContext::bindResource(IOAccelCommandStreamInfo &,IOAccelResource2 *,bool,IOAccelChannel2 *
```

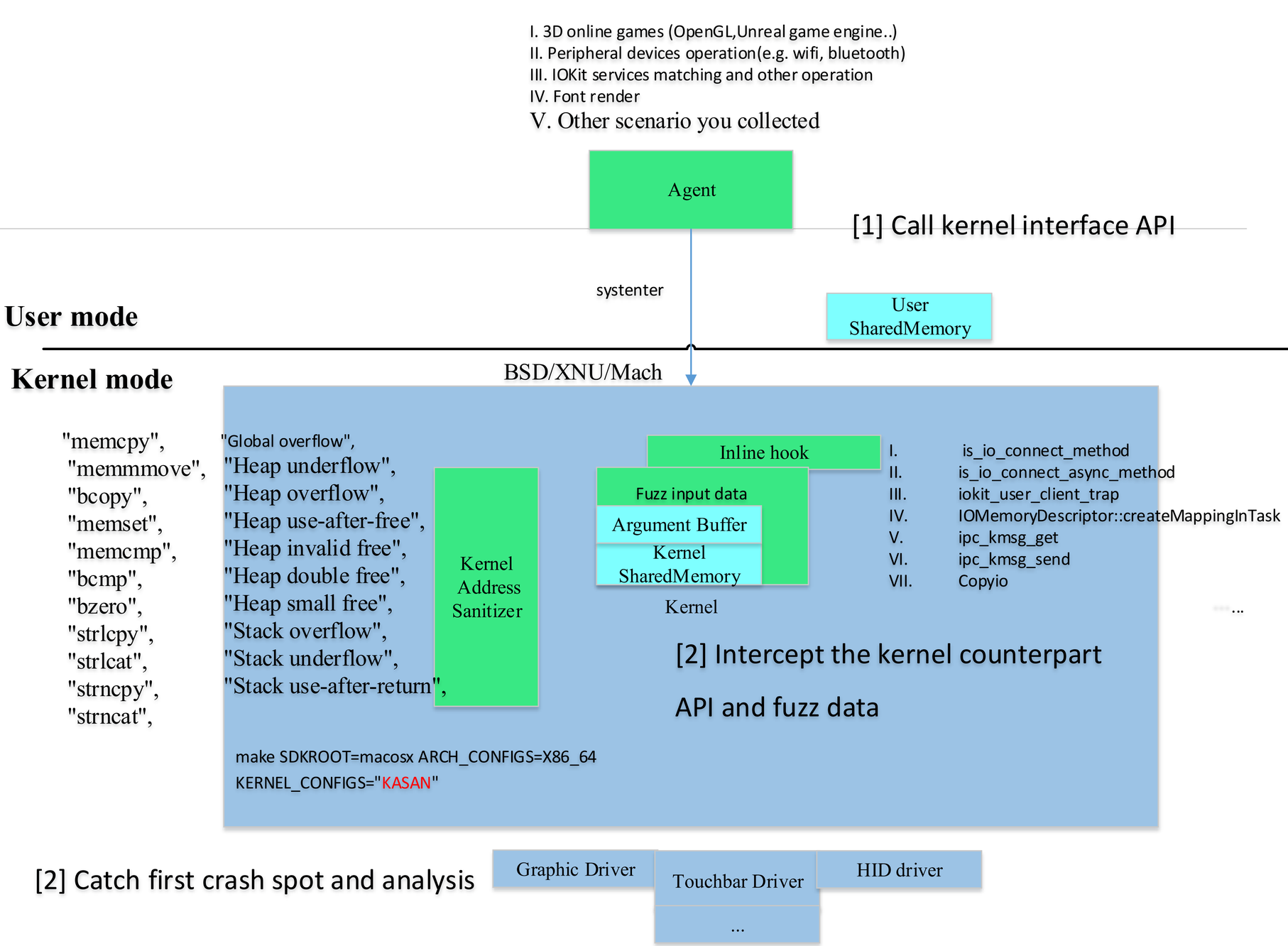
LLDB Debug is your choose

Agenda

- Static Analysis for Kernel Extensions Attack Interfaces
- PassiveFuzz & ActiveFuzz
- Vulnerabilities Found
- Conclusion

PassiveFuzz

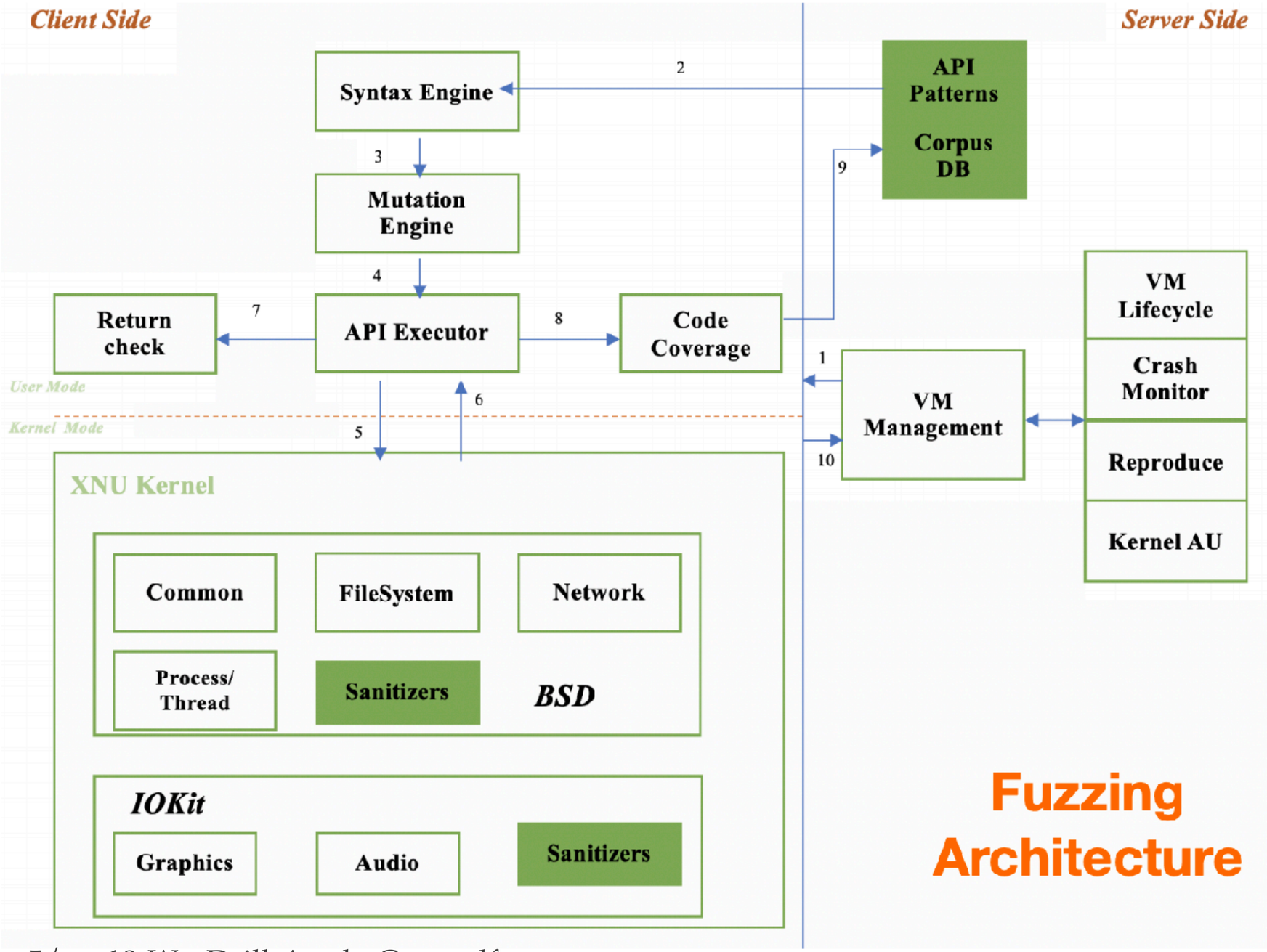
- Inline HOOK
- Probe Installation
- Mutation
- KASAN
- Agent
- Automation



https://conference.hitb.org/hitbsecconf2019ams/materials/D1T2%20-%20Fresh%20Apples%20-%20Researching%20New%20Attack%20Interfaces%20on%20iOS%20and%20OSX%20-%20Moony%20Li%20&%20Lilang%20Wu.pdf

ActiveFuzz

- Porting SyzKaller to Darwin
- API Patterns&Corpus
- Syntax Engine
- Mutation Engine
- API Executor
- Code Coverage
- Sanitizers
- Automation



https://i.blackhat.com/eu-18/Wed-Dec-5/eu-18-Wu-Drill-Apple-Core.pdf

Agenda

- Static Analysis for Kernel Extensions Attack Interfaces
- PassiveFuzz & ActiveFuzz
- Vulnerabilities Found
- Conclusion

Apple CVEs Found

- CVE-2019-8519 : OOB
- CVE-2019-8529 : EoP
- CVE-2019-8635 : EoP
- CVE-2019-8616 : EoP
- CVE-2019-8691 : OOB
- CVE-2019-8692 : OOB

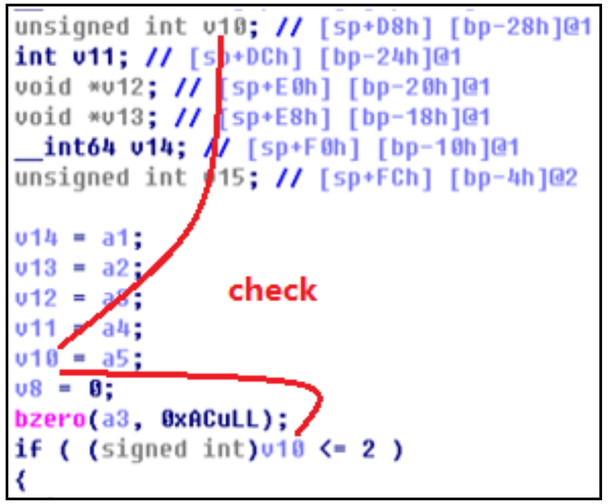
CVE-2018-4462

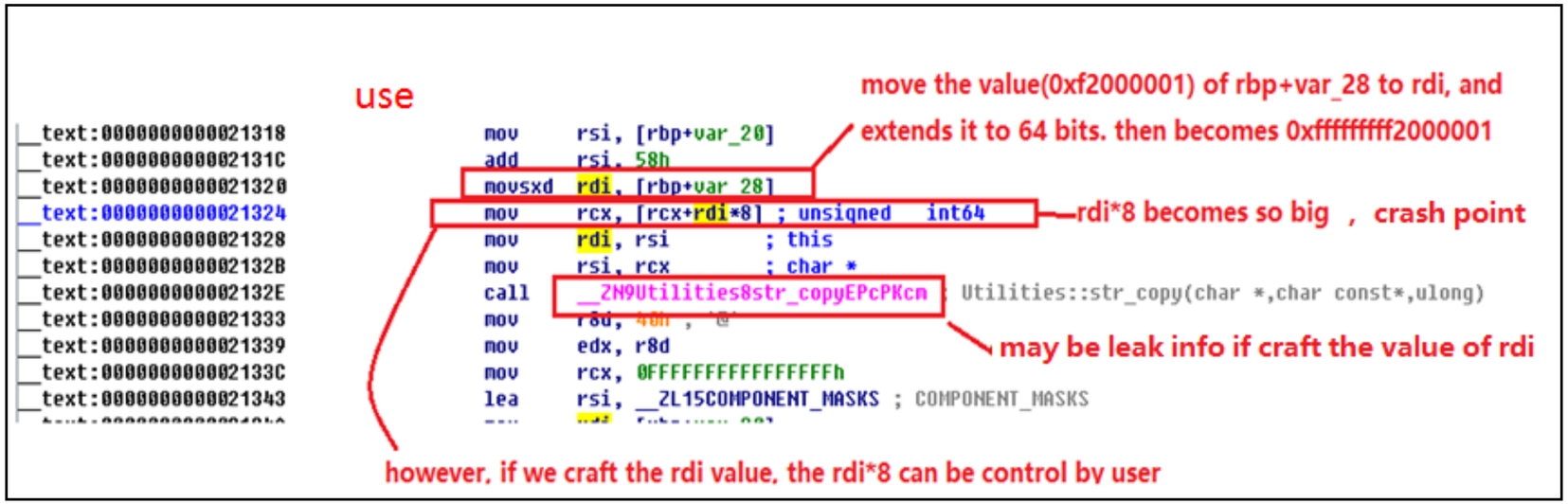
- Integer overflow vulnerability in AMDFramebuffer driver

```
(lldb) bt
* thread #1, stop reason = signal SIGSTOP
  * frame #0: 0xffffff7f8d91e324 AMDFramebuffer`AMDFramebuffer::getPixelInformationFromTiming(AtiDetailedTimingInformation const&, IOPixelInformation*, int, int) + 388
    frame #1: 0xffffff7f8d91e180 AMDFramebuffer`AMDFramebuffer::getPixelInformation(int, int, IOPixelInformation*) + 112
    frame #2: 0xffffff7f8d91e0a5 AMDFramebuffer`AMDFramebuffer::getPixelInformation(int, int, int, IOPixelInformation*) + 101
    frame #3: 0xffffff7f8b42223d IOGraphicsFamily`IOFramebuffer::extGetPixelInformation(target=0xffffff869e59f000, reference=<unavailable>, args=<unavailable>) at IOFrame
    frame #4: 0xffffff800aa4c478 kernel.development`IOUserClient::externalMethod(this=<unavailable>, selector=<unavailable>, args=0xffffffa756c8b988, dispatch=0xffffff7f8
0000) at IOUserClient.cpp:5335 [opt]
    frame #5: 0xffffff7f8b437d0b IOGraphicsFamily`IOFramebufferUserClient::externalMethod(this=0xffffff80b8810800, selector=1, args=0xffffffa756c8b988, dispatch=<unavaila
amebufferUserClient.cpp:380 [opt]
    frame #6: 0xffffff800aa553cf kernel.development`::is_io_connect_method(connection=0xffffff80b8810800, selector=1, scalar_input=<unavailable>, scalar_inputCnt=<unavai
put=0, ool_input_size=0, inband_output="", inband_outputCnt=0xffffff80ac2e2e0c, scalar_output=0xffffffa756c8bcb0, scalar_outputCnt=0xffffffa756c8bcac, ool_output=0, ool_
]
    frame #7: 0xffffff7f8e6c854b pasive_kernel_fuzz`trampline_is_io_connect_method(connection=0xffffff80b8810800, selector=1, scalar_input=0xffffff80b8b81e10, scalar_inpu
_input_size=0, inband_output="", inband_outputCnt=0xffffff80ac2e2e0c, scalar_output=0xffffffa756c8bcb0, scalar_outputCnt=0xffffffa756c8bcac, ool_output=0, ool_output_size
    frame #8: 0xffffff800a3f2bd4 kernel.development`_Xio_connect_method(InHeadP=<unavailable>, OutHeadP=0xffffff80ac2e2de0) at device_server.c:8379 [opt]
    frame #9: 0xffffff800a2c450d kernel.development`ipc_kobject_server(request=0xffffff80b8b81d70, option=<unavailable>) at ipc_kobject.c:359 [opt]
    frame #10: 0xffffff800a29124a kernel.development`ipc_kmsg_send(kmsg=0xffffff80b8b81d70, option=3, send_timeout=0) at ipc_kmsg.c:1822 [opt]
    frame #11: 0xffffff800a2b024f kernel.development`mach_msg_overwrite_trap(args=<unavailable>) at mach_msg.c:546 [opt]
    frame #12: 0xffffff7f8e6d81d7 pasive_kernel_fuzz`trampline_mach_msg_overwrite_trap(args=0xffffffa756c8bf08) at mach_msg_overwrite_trap_trampline.c:131
    frame #13: 0xffffff800a42cb09 kernel.development`mach_call_munger64(state=0xffffff80ac13de20) at bsd_i386.c:573 [opt]
    frame #14: 0xffffff800a25b466 kernel.development`hndl_mach_scall64 + 22
```

Root Cause

frame #0: 0xffffff7f8d91e324 AMDFramebuffer`AMDFramebuffer::**getPixelInformationFromTiming**(AtiDetailedTimingInformation const&, IOPixelInformation*, int, int) + 388
AMDFramebuffer`AMDFramebuffer::getPixelInformationFromTiming:
-> 0xffffff7f8d91e324 <+388>: movq (%rcx,%rdi,8), %rcx ——— a)
0xffffff7f8d91e328 <+392>: movq %rsi, %rdi
0xffffff7f8d91e32b <+395>: movq %rcx, %rsi
0xffffff7f8d91e32e <+398>: **callq 0xffffff7f8ccbcfe0** ; **Utilities::str_copy**(char*, char const*, unsigned long)
(lldb) register read rcx
rcx = 0xffffff7f8d926030 AMDFramebuffer::getPixelInformationFromTiming(AtiDetailedTimingInformation const&, IOPixelInformation*, int, int)::PIXEL_ENCODINGS
(lldb) register read rdi
rdi = 0xffffffff2000001





Other Bugs

- https://i.blackhat.com/USA-19/Thursday/us-19-Lilang-Debug-For-Bug-Crack-And-Hack-Apple-Core-By-Itself-Fun-And-Profit-To-Debug-And-Fuzz-Apple-Kernel-By-LLDB-Script.pdf

- https://documents.trendmicro.com/images/TEx/infographics/Technical%20Brief-Debug%20for%20Bug%20Crack%20and%20Hack%20Apple%20Core%20by%20Itself.pdf

- Follow me on Twitter: @Lilang_Wu
- Email me: 574407955@qq.com

Agenda

- Static Analysis for Kernel Extensions Attack Interfaces
- PassiveFuzz & ActiveFuzz
- Vulnerabilities Found
- Conclusion

Conclusion

- Introduce a method to analyze the attack surfaces of kernel extensions, then introduce two fuzz architecture on Apple system. Finally, we study one CVE case

Questions?