Fresh Apples: Researching New Attack Interfaces on iOS and MacOS

Lilang Wu, Moony Li

Agenda

- About US
- Solution Overview
- Static Analysis for Kernel and KEXTs Attack Interfaces
  - KEXTs Interfaces Vector Automatic Generate
  - Kernel Interfaces Vector Automatic Generate
  - Kernel/KEXTs Interfaces Diff Analysis
- Dynamic Analysis for Kernel and UserMode Attack Interfaces
- Automatic Fuzzing solution
- 0Day vulnerabilities found

About us

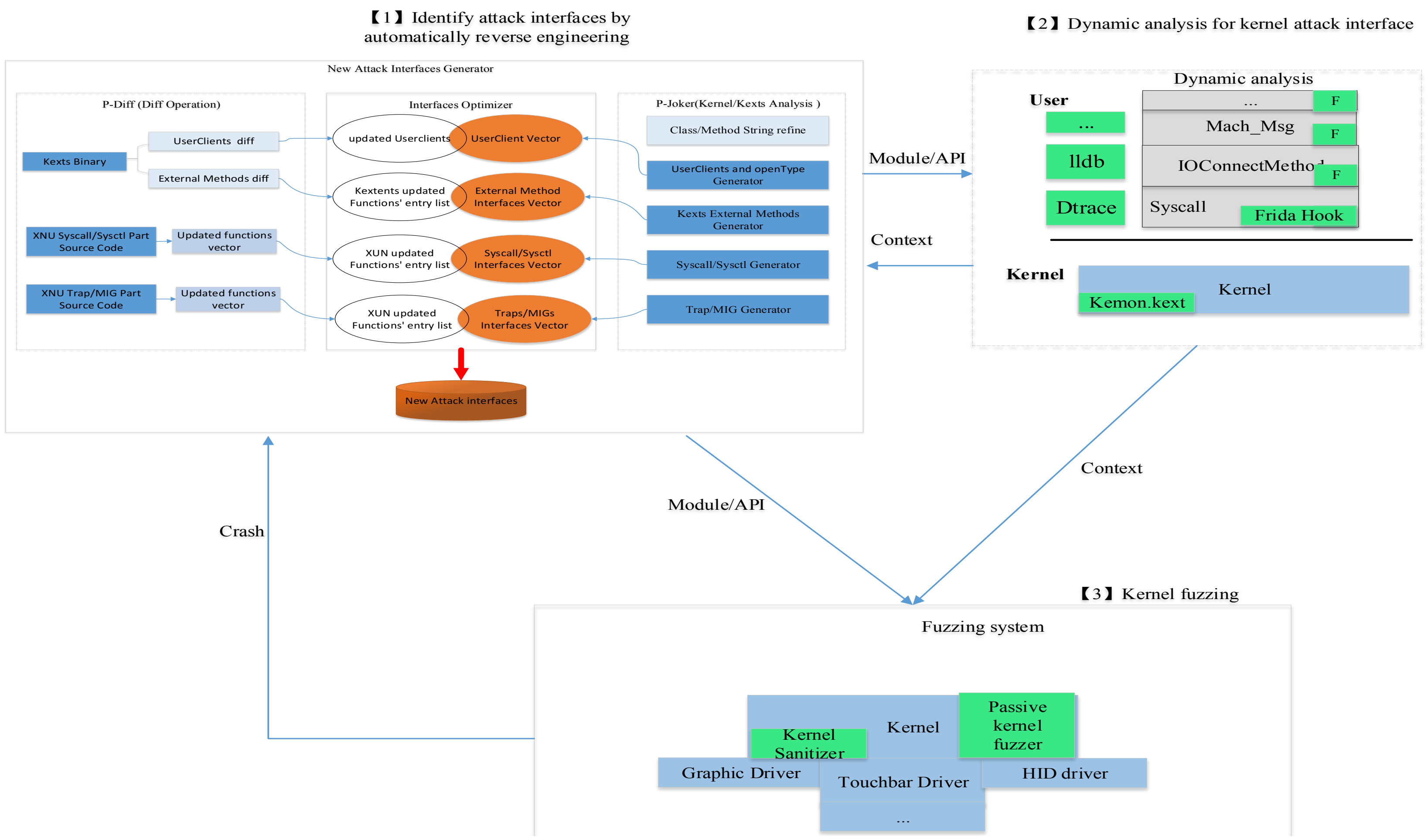
- Lilang Wu
  - 4 years security
  - Mobile advance threat research
  - MacOS/iOS Vulnerability/Malware
  - @Lilang_Wu

- Moony Li
  - 9 years security
  - MacOS/Android/iOS vulnerability
    hunt and exploit
  - Sandbox/Emulator Development
  - @Flyic

Agenda

- About US
- Solution Overview
- Static Analysis for Kernel and KEXTs Attack Interface
  - KEXTs Interfaces Vector Automatic Generate
  - Kernel Interfaces Vector Automatic Generate
  - Kernel/KEXTs Interfaces Diff Analysis
- Dynamic Analysis for Kernel and UserMode Attack Interface
- Automatic Fuzzing solution
- 0Day vulnerabilities found

Solution Overview


【1】Identify attack interfaces by automatically reverse engineering
New Attack Interfaces Generator
P-Diff (Diff Operation)
Kexts Binary
UserClients diff
External Methods diff
XNU Syscall/Sysctl Part Source Code
Updated functions vector
XNU Trap/MIG Part Source Code
Updated functions vector
Interfaces Optimizer
updated Userclients
UserClient Vector
Kextents updated Functions' entry list
External Method Interfaces Vector
XUN updated Functions' entry list
Syscall/Sysctl Interfaces Vector
XUN updated Functions' entry list
Traps/MIGs Interfaces Vector
New Attack interfaces
P-Joker(Kernel/Kexts Analysis)
Class/Method String refine
UserClients and openType Generator
Kexts External Methods Generator
Syscall/Sysctl Generator
Trap/MIG Generator
Module/API
Context
【2】Dynamic analysis for kernel attack interface
Dynamic analysis
User
...
lldb
Dtrace
...
F
Mach_Msg
F
IOConnectMethod
F
Syscall
Frida Hook
Kernel
Kernel
Kemon.kext
Context
Module/API
Crash
【3】Kernel fuzzing
Fuzzing system
Kernel
Kernel Sanitizer
Passive kernel fuzzer
Graphic Driver
Touchbar Driver
HID driver
...

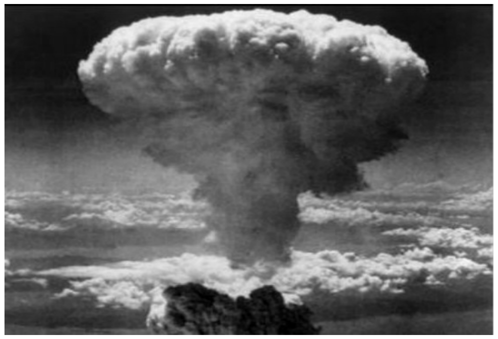
Agenda

- About US
- Solution Overview
- Static Analysis for Kernel and KEXTs Attack Interfaces
  - KEXTs Interfaces Vector Automatic Generate
  - Kernel Interfaces Vector Automatic Generate
  - Kernel/KEXTs Interfaces Diff Analysis
- Dynamic Analysis for Kernel and UserMode Attack Interfaces
- Automatic Fuzzing solution
- 0Day vulnerabilities found

Attack Surfaces

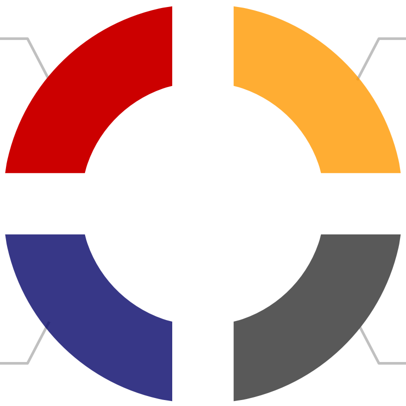
Factory

Military Base


Big City

Think about Apple System

Key Components



XPC Services

interprocess communication mechanism

Kernel

Core which control over everything in the system

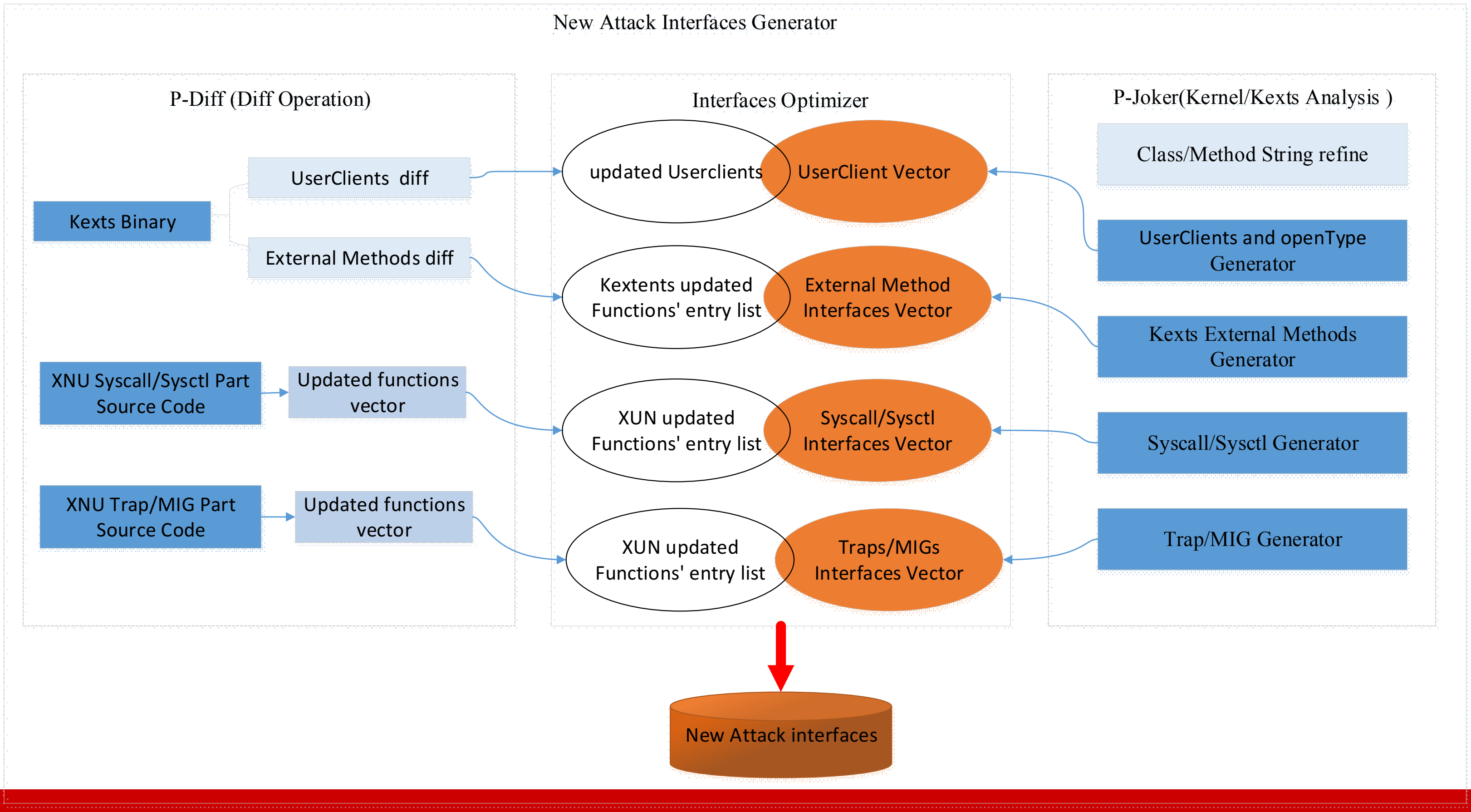
MIG system

generates RPC code for communication between a client and a server process

Kernel Extensions

operates or controls a particular type of device

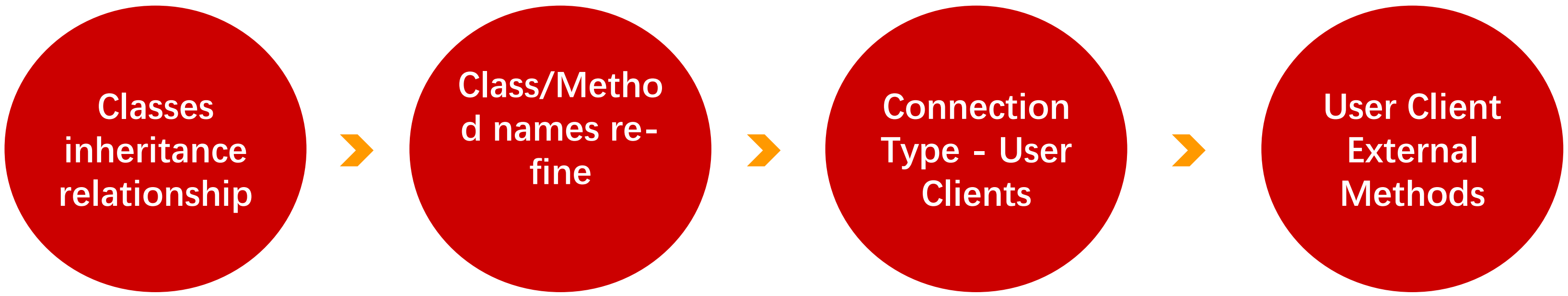
New Attack Interfaces Generator


New Attack Interfaces Generator
P-Diff (Diff Operation)
Kexts Binary
UserClients diff
External Methods diff
XNU Syscall/Sysctl Part Source Code
Updated functions vector
XNU Trap/MIG Part Source Code
Updated functions vector
Interfaces Optimizer
updated Userclients
UserClient Vector
Kextents updated Functions' entry list
External Method Interfaces Vector
XUN updated Functions' entry list
Syscall/Sysctl Interfaces Vector
XUN updated Functions' entry list
Traps/MIGs Interfaces Vector
P-Joker(Kernel/Kexts Analysis)
Class/Method String refine
UserClients and openType Generator
Kexts External Methods Generator
Syscall/Sysctl Generator
Trap/MIG Generator
New Attack interfaces

Agenda

- About US
- Solution Overview
- Static Analysis for Kernel and KEXTs Attack Interface
  - KEXTs Interfaces Vector Automatic Generate
  - Kernel Interfaces Vector Automatic Generate
  - Kernel/KEXTs Interfaces Diff Analysis
- Dynamic Analysis for Kernel and UserMode Attack Interface
- Automatic Fuzzing solution
- 0Day vulnerabilities found

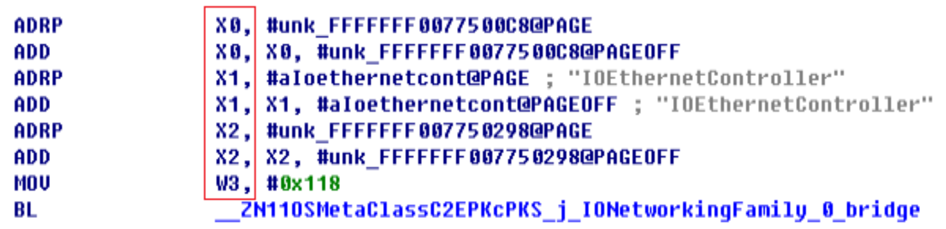
KEXTs Interfaces Analysis Flow


Classes inheritance relationship
Class/Metho d names re-fine
Connection Type - User Clients
User Client External Methods

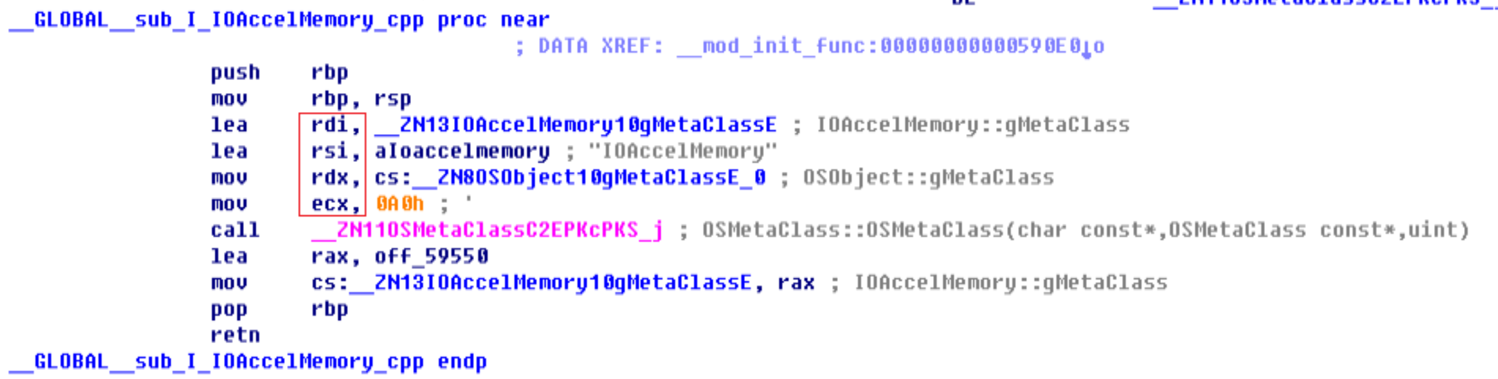
Classes inheritance relationship

OSMetaClass::OSMetaClass

- manages run-time type information for Libkern and I/O Kit C++ classes
- 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



```
ADRP            X0, #unk_FFFFFFF0077500C8@PAGE
ADD             X0, X0, #unk_FFFFFFF0077500C8@PAGEOFF
ADRP            X1, #aIoethernetcont@PAGE ; "IOEthernetController"
ADD             X1, X1, #aIoethernetcont@PAGEOFF ; "IOEthernetController"
ADRP            X2, #unk_FFFFFFF007750298@PAGE
ADD             X2, X2, #unk_FFFFFFF007750298@PAGEOFF
MOV             W3, #0x118
BL              __ZN11OSMetaClassC2EPKcPKS_j_IONetworkingFamily_0_bridge
```

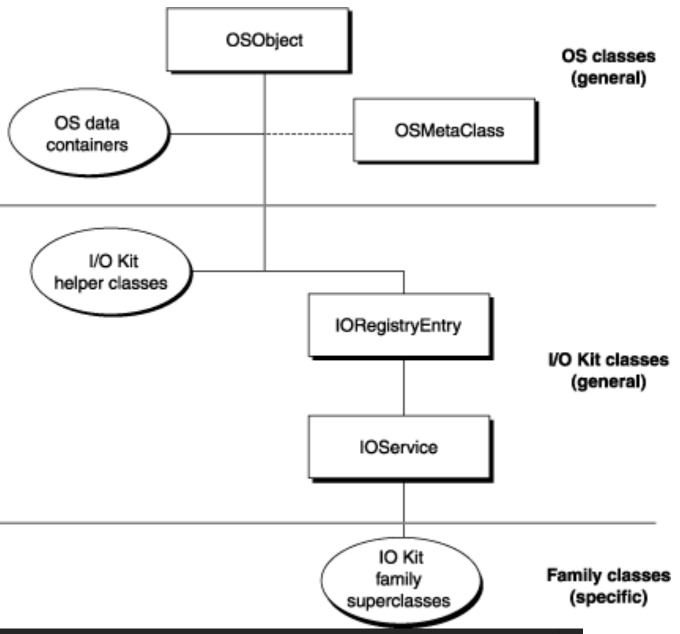


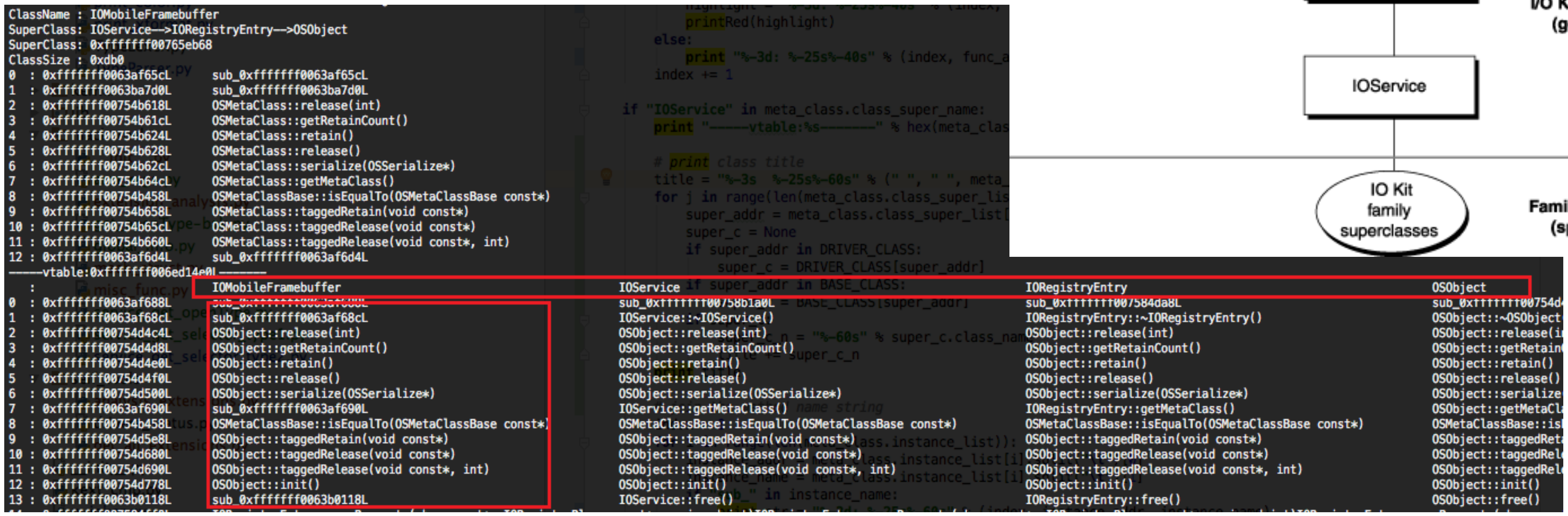
```
__GLOBAL__sub_I_IOAccelMemory_cpp proc near
                                        ; DATA XREF: __mod_init_func:00000000000590E0↓o
                push    rbp
                mov     rbp, rsp
                lea     rdi, __ZN13IOAccelMemory10gMetaClassE ; IOAccelMemory::gMetaClass
                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
```

Class/Method names re-fine

Re-fine two method table

- instance method table
- meta method table





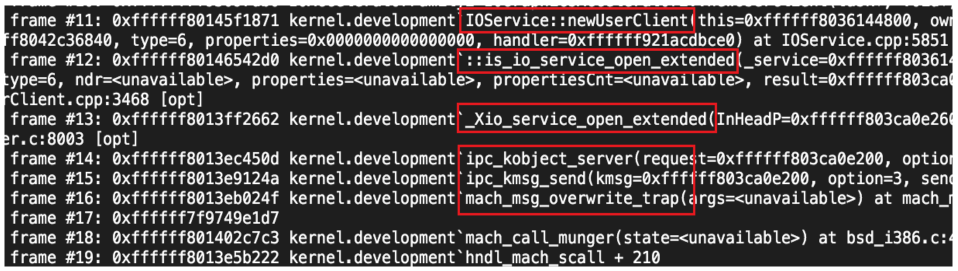
```
ClassName : IOMobileFramebuffer
SuperClass: IOService-->IORegistryEntry-->OSObject
SuperClass: 0xffffff00765eb68
ClassSize : 0xdb0
0  : 0xffffff0063af65cL    sub_0xffffff0063af65cL
1  : 0xffffff0063ba7d0L    sub_0xffffff0063ba7d0L
2  : 0xffffff00754b618L    OSMetaClass::release(int)
3  : 0xffffff00754b61cL    OSMetaClass::getRetainCount()
4  : 0xffffff00754b624L    OSMetaClass::retain()
5  : 0xffffff00754b628L    OSMetaClass::release()
6  : 0xffffff00754b62cL    OSMetaClass::serialize(OSSerialize*)
7  : 0xffffff00754b64cL    OSMetaClass::getMetaClass()
8  : 0xffffff00754b458L    OSMetaClassBase::isEqualTo(OSMetaClassBase const*)
9  : 0xffffff00754b658L    OSMetaClass::taggedRetain(void const*)
10 : 0xffffff00754b65cL    OSMetaClass::taggedRelease(void const*)
11 : 0xffffff00754b660L    OSMetaClass::taggedRelease(void const*, int)
12 : 0xffffff0063af6d4L    sub_0xffffff0063af6d4L
-----vtable:0xffffff006ed14e0L--------
   :                       IOMobileFramebuffer                                IOService                                          IORegistryEntry                                    OSObject
0  : 0xffffff0063af688L    sub_0xffffff0063af688L                             sub_0xffffff00758b1a0L                             sub_0xffffff007584da8L                             sub_0xffffff00754d
1  : 0xffffff0063af68cL    sub_0xffffff0063af68cL                             IOService::~IOService()                            IORegistryEntry::~IORegistryEntry()                OSObject::~OSObject
2  : 0xffffff00754d4c4L    OSObject::release(int)                             OSObject::release(int)                             OSObject::release(int)                             OSObject::release(i
3  : 0xffffff00754d4d8L    OSObject::getRetainCount()                         OSObject::getRetainCount()                         OSObject::getRetainCount()                         OSObject::getRetain
4  : 0xffffff00754d4e0L    OSObject::retain()                                 OSObject::retain()                                 OSObject::retain()                                 OSObject::retain()
5  : 0xffffff00754d4f0L    OSObject::release()                                OSObject::release()                                OSObject::release()                                OSObject::release()
6  : 0xffffff00754d500L    OSObject::serialize(OSSerialize*)                  OSObject::serialize(OSSerialize*)                  OSObject::serialize(OSSerialize*)                  OSObject::serialize
7  : 0xffffff0063af690L    sub_0xffffff0063af690L                             IOService::getMetaClass()                          IORegistryEntry::getMetaClass()                    OSObject::getMetaCl
8  : 0xffffff00754b458L    OSMetaClassBase::isEqualTo(OSMetaClassBase const*) OSMetaClassBase::isEqualTo(OSMetaClassBase const*) OSMetaClassBase::isEqualTo(OSMetaClassBase const*) OSMetaClassBase::is
9  : 0xffffff00754d5e8L    OSObject::taggedRetain(void const*)                OSObject::taggedRetain(void const*)                OSObject::taggedRetain(void const*)                OSObject::taggedRet
10 : 0xffffff00754d680L    OSObject::taggedRelease(void const*)               OSObject::taggedRelease(void const*)               OSObject::taggedRelease(void const*)               OSObject::taggedRel
11 : 0xffffff00754d690L    OSObject::taggedRelease(void const*, int)          OSObject::taggedRelease(void const*, int)          OSObject::taggedRelease(void const*, int)          OSObject::taggedRel
12 : 0xffffff00754d778L    OSObject::init()                                   OSObject::init()                                   OSObject::init()                                   OSObject::init()
13 : 0xffffff0063b0118L    sub_0xffffff0063b0118L                             IOService::free()                                  IORegistryEntry::free()                            OSObject::free()
```

```python
            printRed(highlight)
        else:
            print "%-3d: %-25s%-40s" % (index, func_a
        index += 1

    if "IOService" in meta_class.class_super_name:
        print "-----vtable:%s--------" % hex(meta_clas

        # print class title
        title = "%-3s  %-25s%-60s" % (" ", " ", meta_
        for j in range(len(meta_class.class_super_lis
            super_addr = meta_class.class_super_list[
            super_c = None
            if super_addr in DRIVER_CLASS:
                super_c = DRIVER_CLASS[super_addr]
            if super_addr in BASE_CLASS:
```

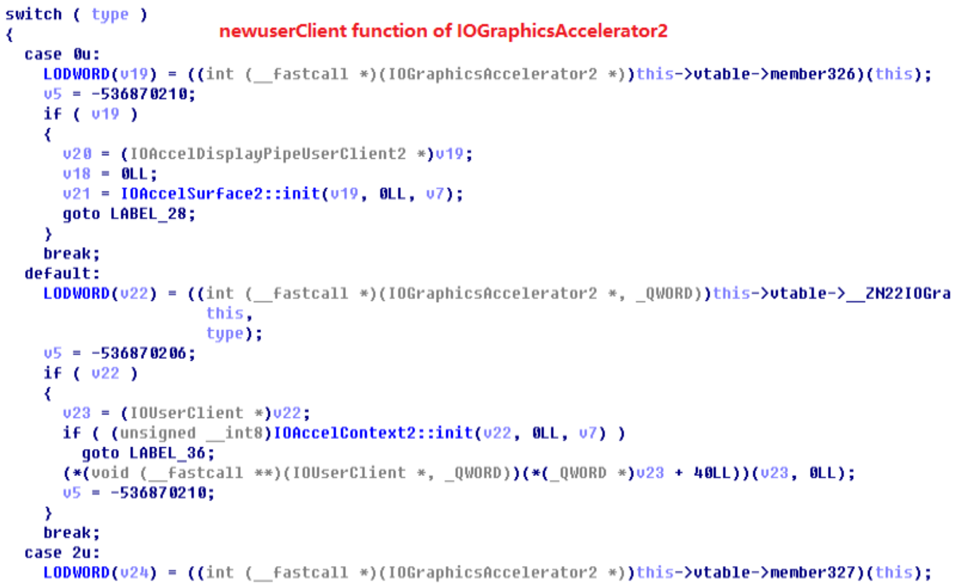
Connection Type - User Clients

IOService::newUserClient function

- creates an IOUserClient-based connection for communication with a non-kernel client
- invokes this function by calling the IOServiceOpen



```
 frame #11: 0xffffff80145f1871 kernel.development`IOService::newUserClient(this=0xffffff8036144800, own
ff8042c36840, type=6, properties=0x0000000000000000, handler=0xffffff921acdbce0) at IOService.cpp:5851
 frame #12: 0xffffff80146542d0 kernel.development`::is_io_service_open_extended(_service=0xffffff803614
type=6, ndr=<unavailable>, properties=<unavailable>, propertiesCnt=<unavailable>, result=0xffffff803ca0
rClient.cpp:3468 [opt]
 frame #13: 0xffffff8013ff2662 kernel.development`_Xio_service_open_extended(InHeadP=0xffffff803ca0e260
er.c:8003 [opt]
 frame #14: 0xffffff8013ec450d kernel.development`ipc_kobject_server(request=0xffffff803ca0e200, option
 frame #15: 0xffffff8013e9124a kernel.development`ipc_kmsg_send(kmsg=0xffffff803ca0e200, option=3, send
 frame #16: 0xffffff8013eb024f kernel.development`mach_msg_overwrite_trap(args=<unavailable>) at mach_r
 frame #17: 0xffffff7f9749e1d7
 frame #18: 0xffffff801402c7c3 kernel.development`mach_call_munger(state=<unavailable>) at bsd_i386.c:4
 frame #19: 0xffffff8013e5b222 kernel.development`hndl_mach_scall + 210
```



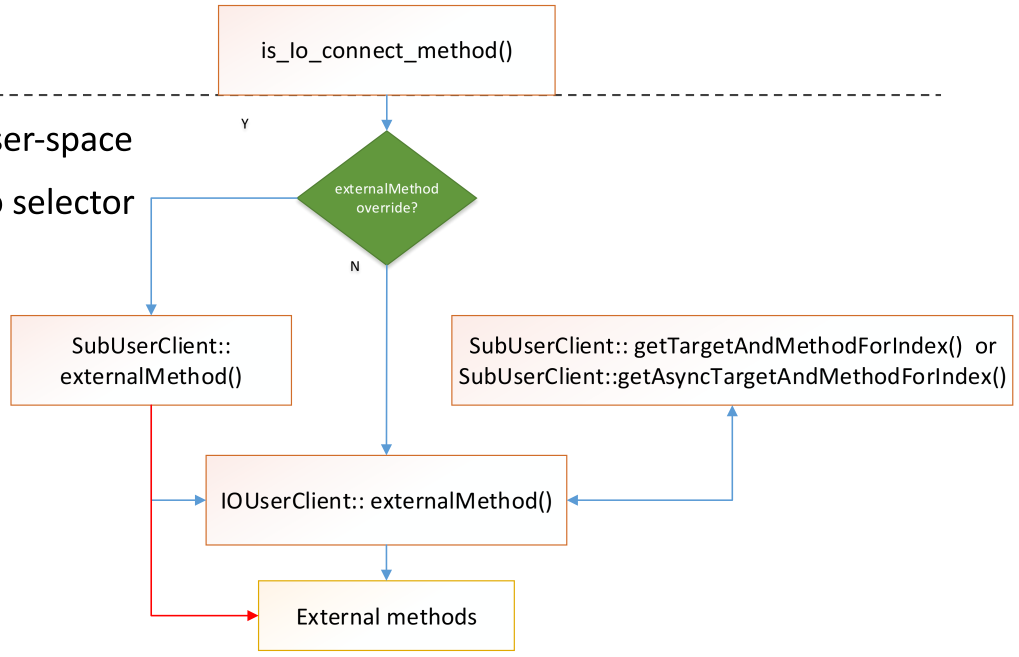
newuserClient function of IOGraphicsAccelerator2

```
switch ( type )
{
  case 0u:
    LODWORD(v19) = ((int (__fastcall *)(IOGraphicsAccelerator2 *))this->vtable->member326)(this);
    v5 = -536870210;
    if ( v19 )
    {
      v20 = (IOAccelDisplayPipeUserClient2 *)v19;
      v18 = 0LL;
      v21 = IOAccelSurface2::init(v19, 0LL, v7);
      goto LABEL_28;
    }
    break;
  default:
    LODWORD(v22) = ((int (__fastcall *)(IOGraphicsAccelerator2 *, _QWORD))this->vtable->__ZN22IOGra
                      this,
                      type);
    v5 = -536870206;
    if ( v22 )
    {
      v23 = (IOUserClient *)v22;
      if ( (unsigned __int8)IOAccelContext2::init(v22, 0LL, v7) )
        goto LABEL_36;
      (*(void (__fastcall **)(IOUserClient *, _QWORD))(*(_QWORD *)v23 + 40LL))(v23, 0LL);
      v5 = -536870210;
    }
    break;
  case 2u:
    LODWORD(v24) = ((int (__fastcall *)(IOGraphicsAccelerator2 *))this->vtable->member327)(this);
```

User Client External Methods

IOUserClient::externalMethod

- Can be call by is_io_connect_method from user-space
- execute related external method according to selector

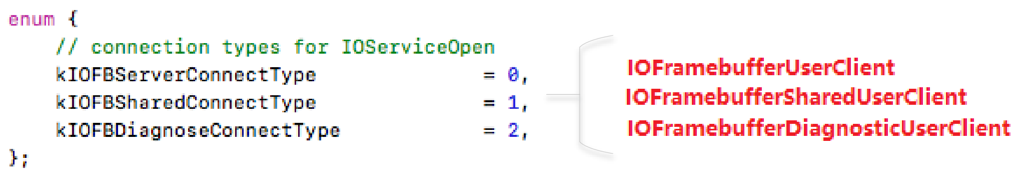


is_Io_connect_method()
Y
externalMethod override?
N
SubUserClient:: externalMethod()
SubUserClient:: getTargetAndMethodForIndex() or SubUserClient::getAsyncTargetAndMethodForIndex()
IOUserClient:: externalMethod()
External methods

How Automation?

CONNECTION TYPE - USER CLIENTS

- Find the <connection type – user client> tuple, like

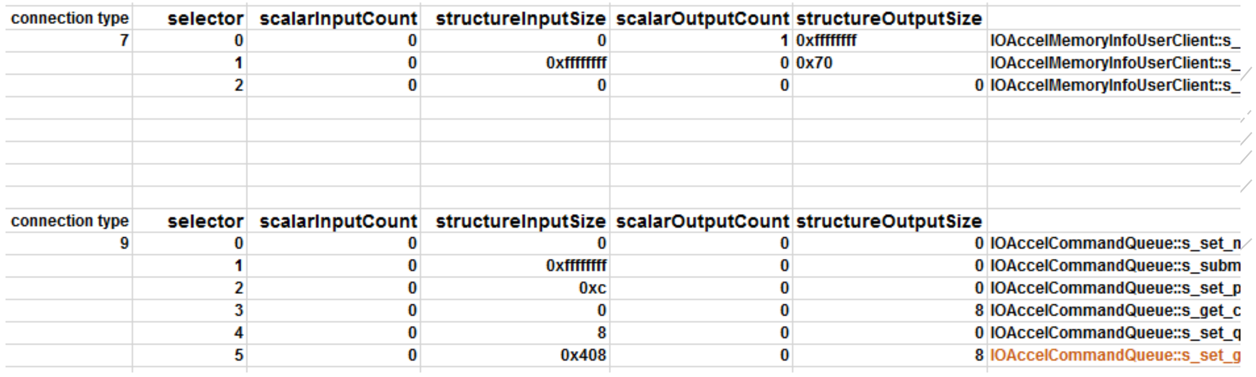


```
enum {
    // connection types for IOServiceOpen
    kIOFBServerConnectType             = 0,
    kIOFBSharedConnectType             = 1,
    kIOFBDiagnoseConnectType           = 2,
};
```

IOFramebufferUserClient
IOFramebufferSharedUserClient
IOFramebufferDiagnosticUserClient

USER CLIENT EXTERNAL METHODS

- Find or construct a external methods dispatch table, like



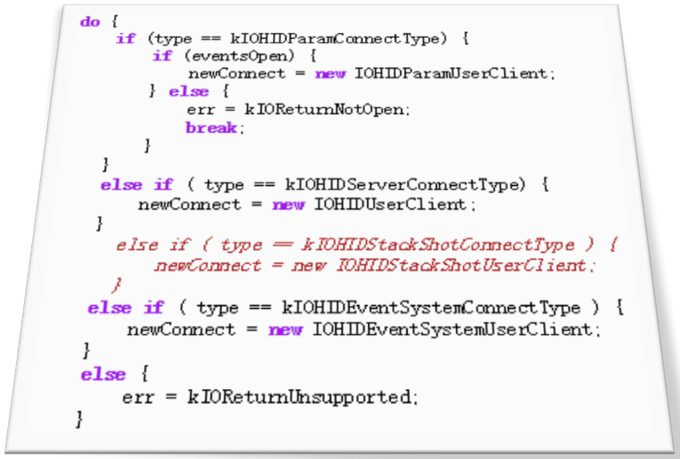
connection type	selector	scalarInputCount	structureInputSize	scalarOutputCount	structureOutputSize	
7	0	0	0	1	0xffffffff	IOAccelMemoryInfoUserClient::s_
	1	0	0xffffffff	0	0x70	IOAccelMemoryInfoUserClient::s_
	2	0	0	0	0	IOAccelMemoryInfoUserClient::s_

connection type	selector	scalarInputCount	structureInputSize	scalarOutputCount	structureOutputSize	
9	0	0	0	0	0	IOAccelCommandQueue::s_set_n
	1	0	0xffffffff	0	0	IOAccelCommandQueue::s_subm
	2	0	0xc	0	0	IOAccelCommandQueue::s_set_p
	3	0	0	0	8	IOAccelCommandQueue::s_get_c
	4	0	8	0	0	IOAccelCommandQueue::s_set_q
	5	0	0x408	0	8	IOAccelCommandQueue::s_set_g

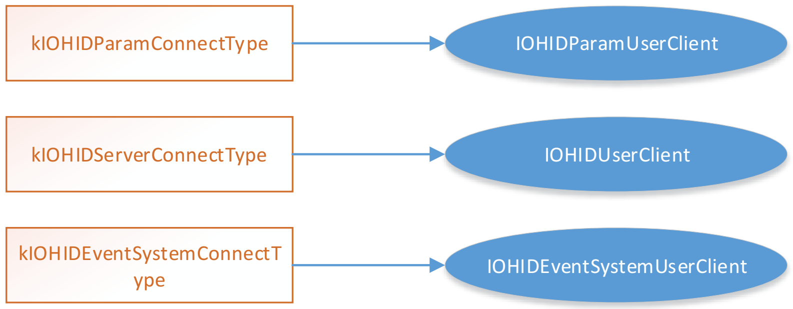
Connection Type - User Clients

Automation Methodology

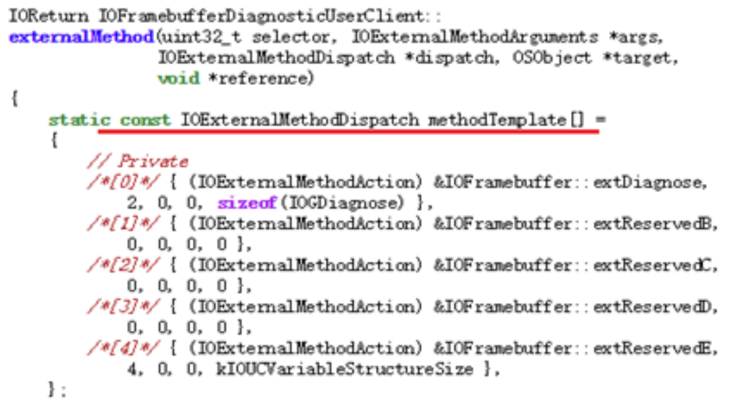
- Locate the newUserClient function address in the driver
- Analyze the ASM instructions to enumerate the connection types
- Analyze the ASM instructions to get the corresponding user client for each connection type



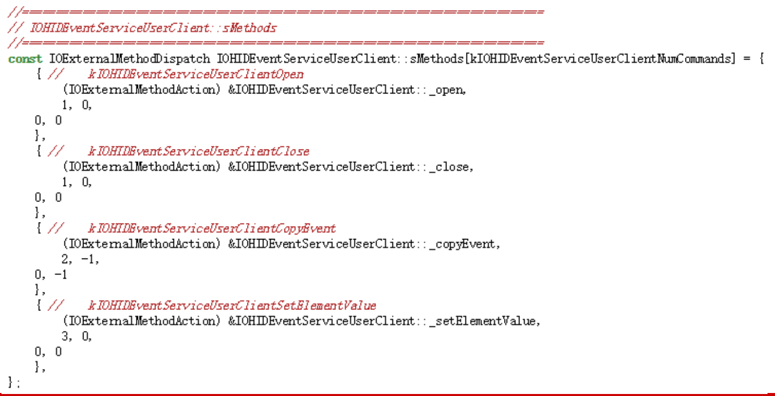
```
do {
    if (type == kIOHIDParamConnectType) {
        if (eventsOpen) {
            newConnect = new IOHIDParamUserClient;
        } else {
            err = kIOReturnNotOpen;
            break;
        }
    }
    else if ( type == kIOHIDServerConnectType) {
        newConnect = new IOHIDUserClient;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



User Client External Methods – Graceful



```
IOReturn IOFramebufferDiagnosticUserClient::
externalMethod(uint32_t selector, IOExternalMethodArguments *args,
               IOExternalMethodDispatch *dispatch, OSObject *target,
               void *reference)
{
    static const IOExternalMethodDispatch methodTemplate[] =
    {
        // Private
        /*[0]*/ { (IOExternalMethodAction) &IOFramebuffer::extDiagnose,
            2, 0, 0, sizeof(IOGDiagnose) },
        /*[1]*/ { (IOExternalMethodAction) &IOFramebuffer::extReservedB,
            0, 0, 0, 0 },
        /*[2]*/ { (IOExternalMethodAction) &IOFramebuffer::extReservedC,
            0, 0, 0, 0 },
        /*[3]*/ { (IOExternalMethodAction) &IOFramebuffer::extReservedD,
            0, 0, 0, 0 },
        /*[4]*/ { (IOExternalMethodAction) &IOFramebuffer::extReservedE,
            4, 0, 0, kIOUCVariableStructureSize },
    };
```



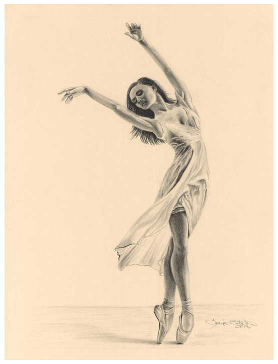
```
//==============================================================================
// IOHIDEventServiceUserClient::sMethods
//==============================================================================
const IOExternalMethodDispatch IOHIDEventServiceUserClient::sMethods[kIOHIDEventServiceUserClientNumCommands] = {
    { //    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
    },
};
```

Defined as Global or Local Constant Array

- externalMethod

OR

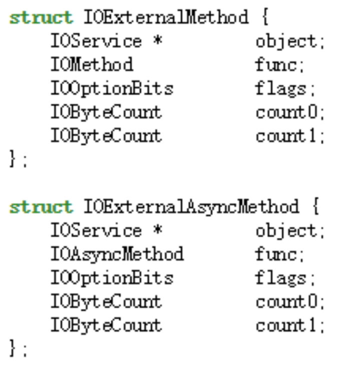
- getTargetAndMethodForIndex
- getAsyncTargetAndMethodForIndex



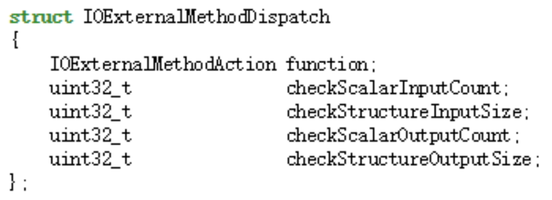
User Client External Methods – Graceful

Automation Methodology

- Locate the starting address for each constant array in the symbol table
- Parse the contents according to the IOExternalMethodDispatch or IOExternalMethod structure from starting address

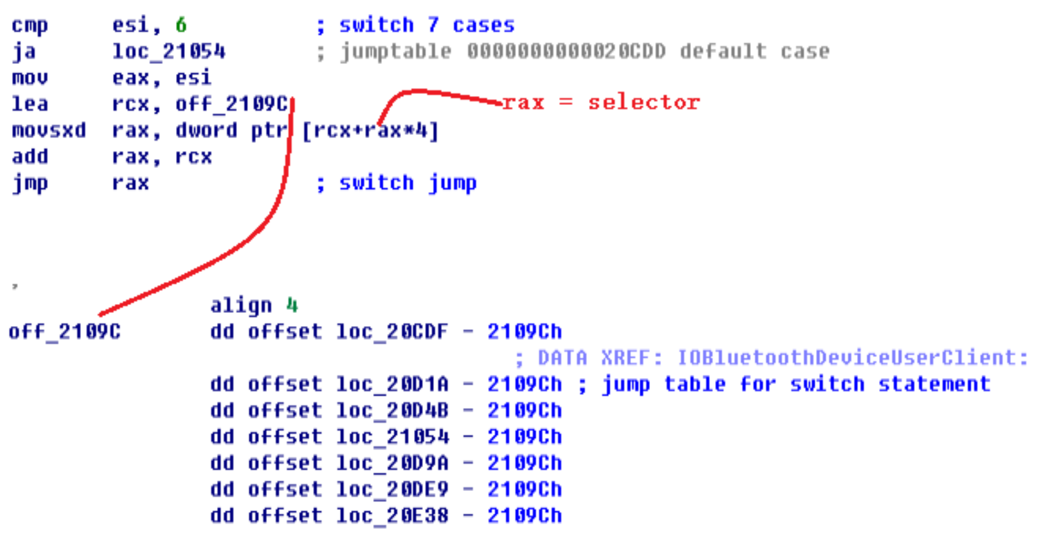


```
struct IOExternalMethod {
    IOService *         object;
    IOMethod            func;
    IOOptionBits        flags;
    IOByteCount         count0;
    IOByteCount         count1;
};

struct IOExternalAsyncMethod {
    IOService *         object;
    IOAsyncMethod       func;
    IOOptionBits        flags;
    IOByteCount         count0;
    IOByteCount         count1;
};
```



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

User Client External Methods - Ugly



```
cmp     esi, 6          ; switch 7 cases
ja      loc_21054       ; jumptable 0000000000020CDD default case
mov     eax, esi
lea     rcx, off_2109C
movsxd  rax, dword ptr [rcx+rax*4]
add     rax, rcx
jmp     rax             ; switch jump

;
            align 4
off_2109C   dd offset loc_20CDF - 2109Ch
                                        ; DATA XREF: IOBluetoothDeviceUserClient:
            dd offset loc_20D1A - 2109Ch ; jump table for switch statement
            dd offset loc_20D4B - 2109Ch
            dd offset loc_21054 - 2109Ch
            dd offset loc_20D9A - 2109Ch
            dd offset loc_20DE9 - 2109Ch
            dd offset loc_20E38 - 2109Ch
```

Automation Methodology

- Locate the address of override externalMethod/getTarget…/getAsyncTarget... Function
- Analyze the ASM instructions to get selector and external methods

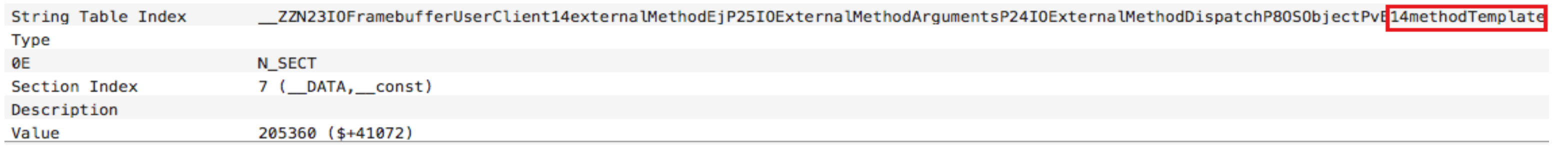
Automation Implementation

- Locate the Address of Key Const Array and Parse their Content
  - For global const external method dispatch
  - For local const external method dispatch
- Analyze the ASM Instructions
  - For newUserClient method
  - For externalMethod/getTar…/getAsynTar… method

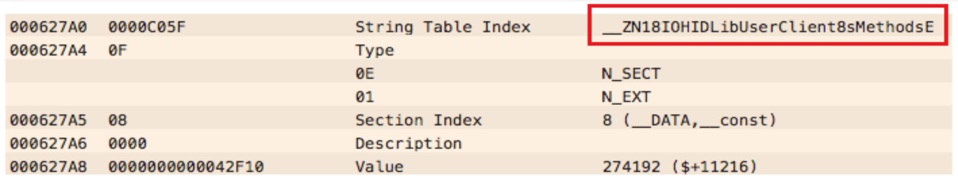
Parse the External Method Dispatch Array

Global/Local Const Array

- 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)
```

Analyze the ASM Instructions

Methods Compare

Angr/Miasm

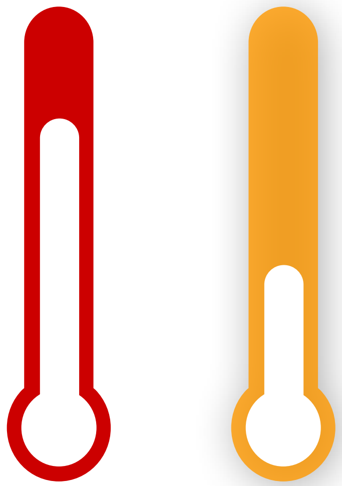
Support binary to ASM

Support ASM CG/CFG

Support Emulating using JIT

Need to Map Multi code into SE VM

Construct conditions to support all paths



Start From Scratch

Can only Simulate part of Instruction Operation

Can only care about certain registers' data flow

Need to construct a kext analysis engine

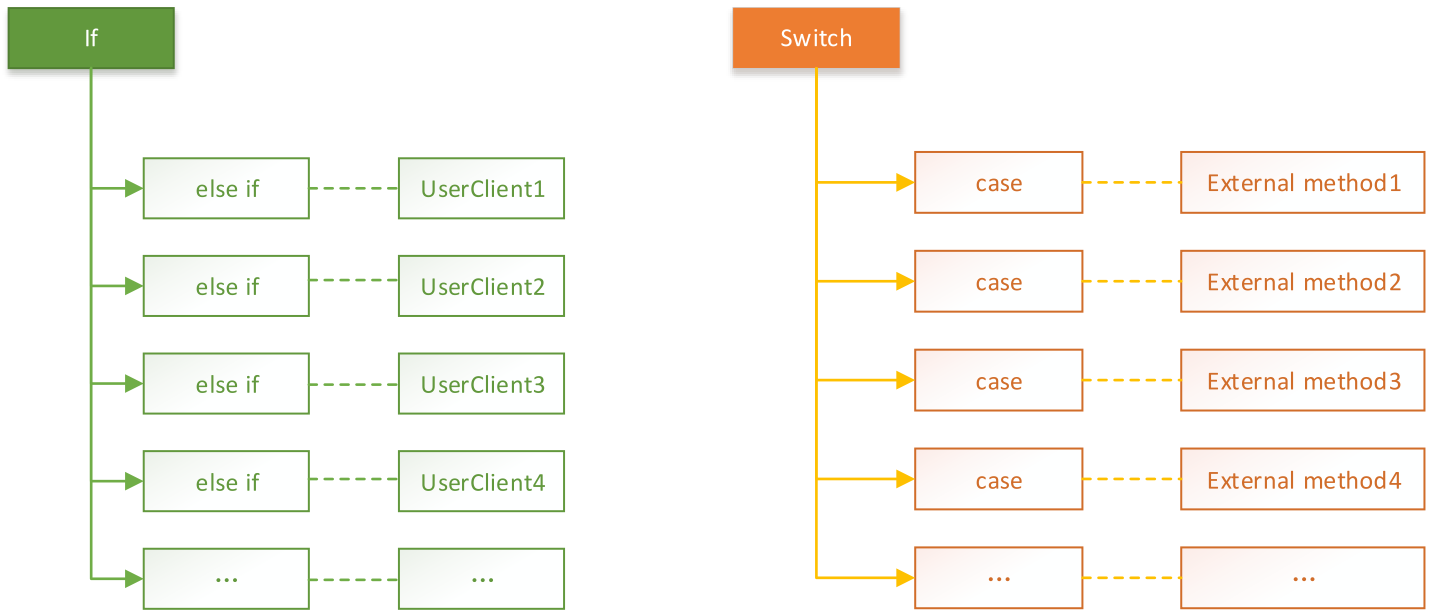
Need disassembling, CG/CFG

Need to vm address operation

…

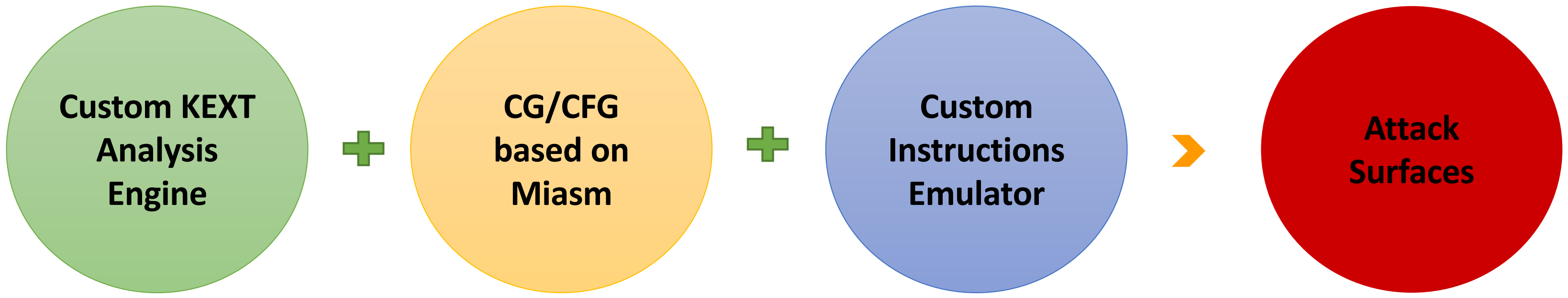
Analyze the ASM Instructions

What we want?



Analyze the ASM Instructions

- Combination Method

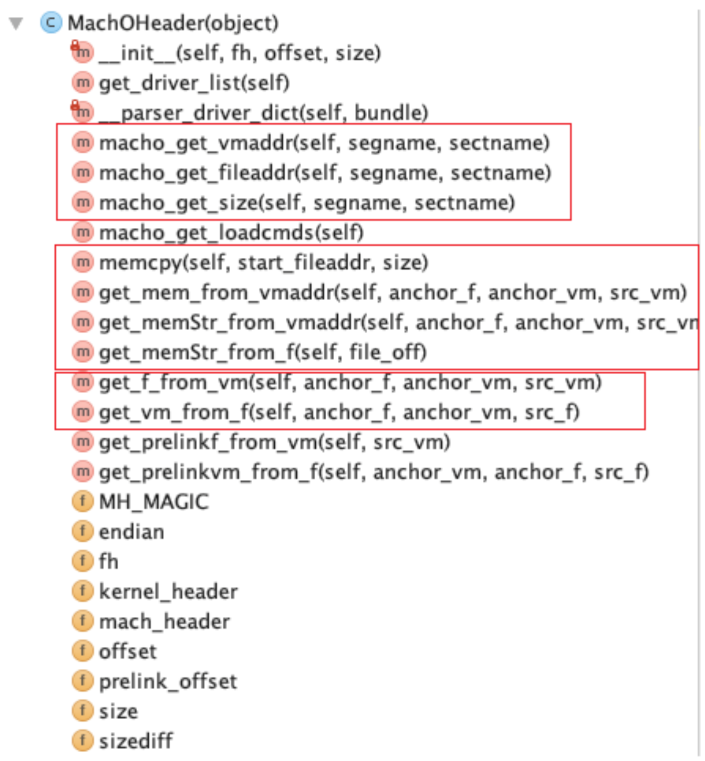


Automation Practice

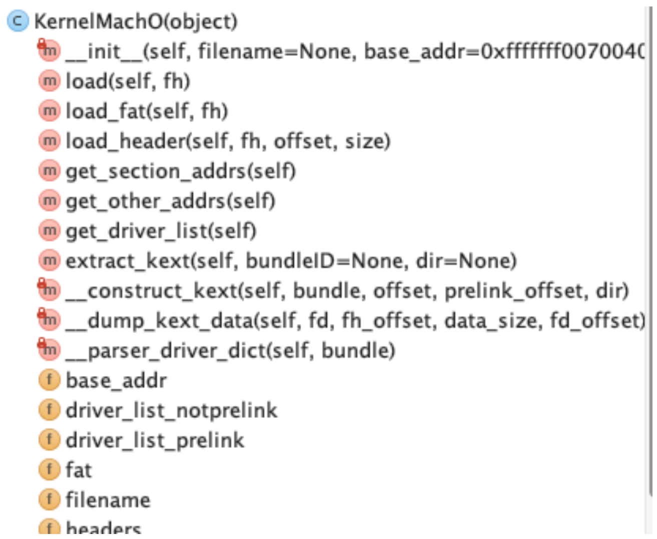
Custom KEXT Analysis Engine

- Init a MachO structure handler
- Parse key sections and segments
- Vm addr and file addr operation
- Vm memory operation based on vm and file addr

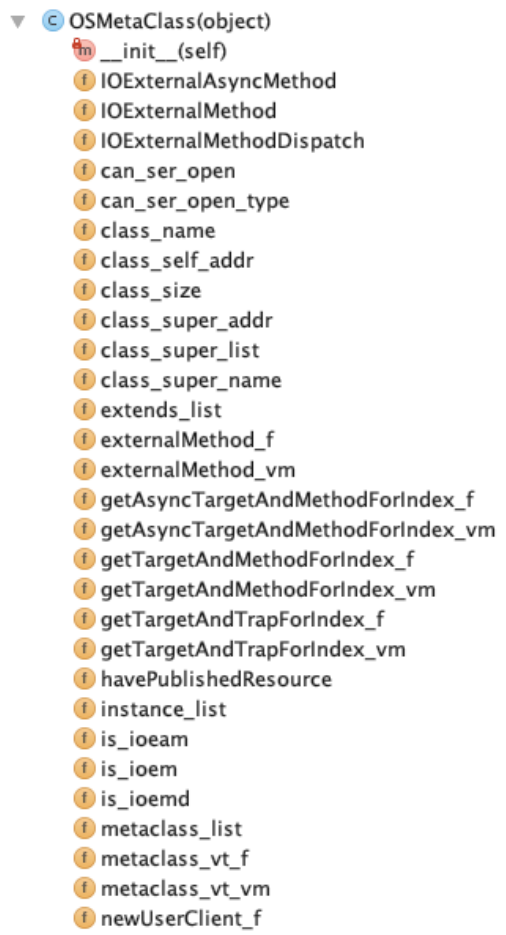
Custom KEXT Analysis Engine



```
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_vr
    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=0xffffff0070040
    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
```

Automation Practice

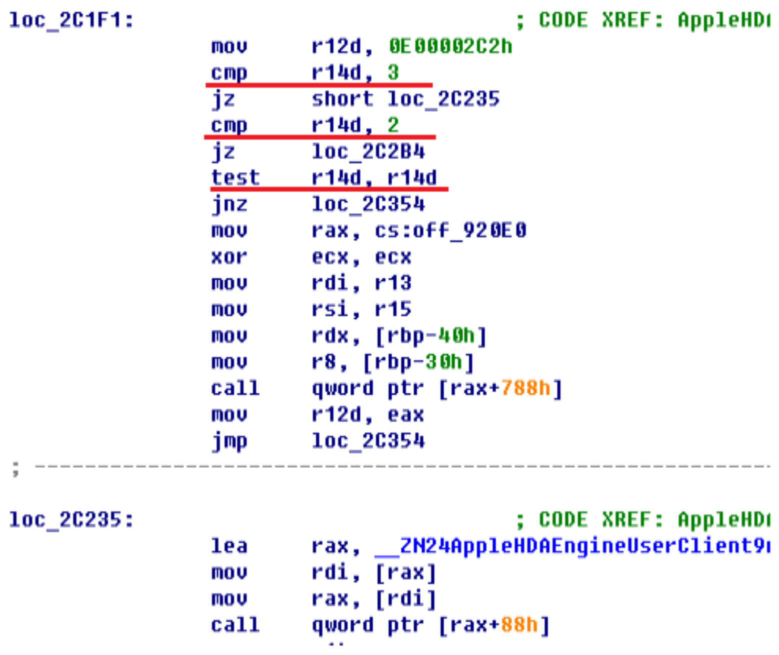
CG/CFG BASED ON MIASM

- Init a disassembling engine, like capstone
- Init Miasm machine
- Gernerate CFG local information

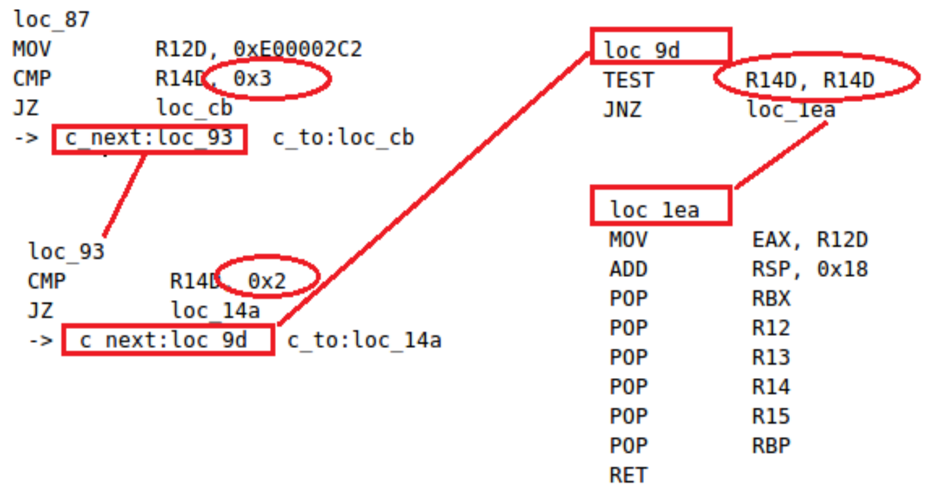
https://github.com/cea-sec/miasm

Generate CFG local information

AppleHDAEngine::newUserClient



```
loc_2C1F1:                              ; CODE XREF: AppleHD
                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: AppleHD
                lea     rax, __ZN24AppleHDAEngineUserClient9
                mov     rdi, [rax]
                mov     rax, [rdi]
                call    qword ptr [rax+88h]
```

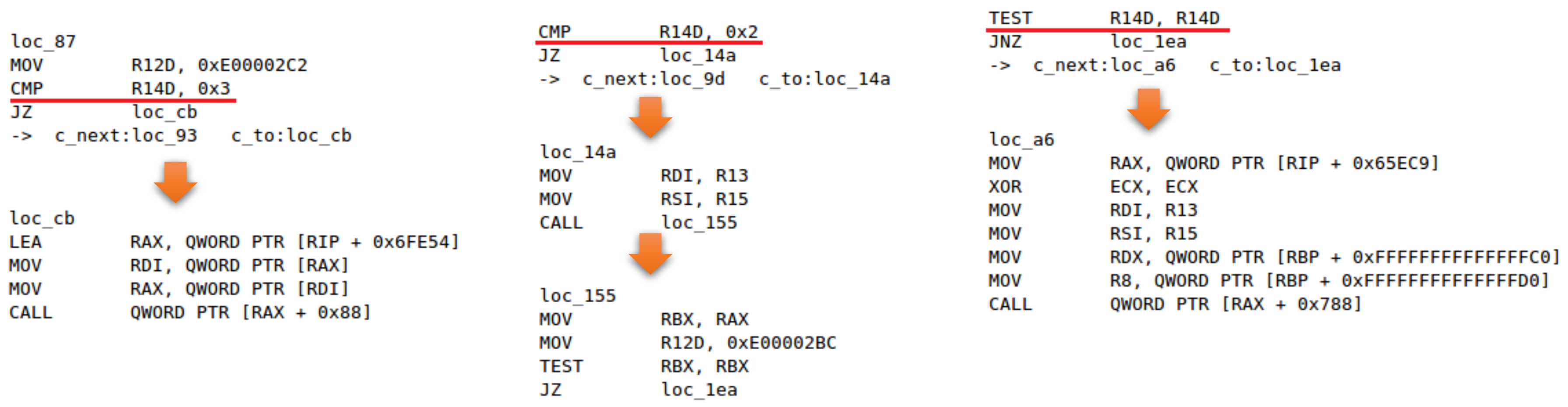


```
loc_87                                          loc_9d
MOV     R12D, 0xE00002C2                        TEST    R14D, R14D
CMP     R14D, 0x3                               JNZ     loc_1ea
JZ      loc_cb
->  c_next:loc_93   c_to:loc_cb
                                                loc_1ea
                                                MOV     EAX, R12D
 loc_93                                         ADD     RSP, 0x18
 CMP     R14D, 0x2                              POP     RBX
 JZ      loc_14a                                POP     R12
 ->  c_next:loc_9d   c_to:loc_14a               POP     R13
                                                POP     R14
                                                POP     R15
                                                POP     RBP
                                                RET
```

Analyze 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

AppleHDAEngine::newUserClient



```
loc_87
MOV         R12D, 0xE00002C2
CMP         R14D, 0x3
JZ          loc_cb
->  c_next:loc_93    c_to:loc_cb

loc_cb
LEA         RAX, QWORD PTR [RIP + 0x6FE54]
MOV         RDI, QWORD PTR [RAX]
MOV         RAX, QWORD PTR [RDI]
CALL        QWORD PTR [RAX + 0x88]
```

```
CMP         R14D, 0x2
JZ          loc_14a
->  c_next:loc_9d    c_to:loc_14a

loc_14a
MOV         RDI, R13
MOV         RSI, R15
CALL        loc_155

loc_155
MOV         RBX, RAX
MOV         R12D, 0xE00002BC
TEST        RBX, RBX
JZ          loc_1ea
```

```
TEST        R14D, R14D
JNZ         loc_1ea
->  c_next:loc_a6    c_to:loc_1ea

loc_a6
MOV         RAX, QWORD PTR [RIP + 0x65EC9]
XOR         ECX, ECX
MOV         RDI, R13
MOV         RSI, R15
MOV         RDX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFC0]
MOV         R8, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFD0]
CALL        QWORD PTR [RAX + 0x788]
```

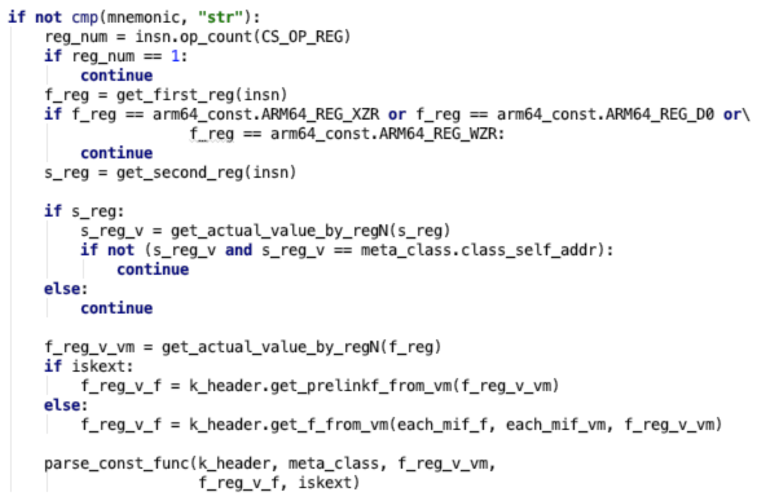
Automation Practice

CUSTOM INSTRUCTIONS EMULATOR

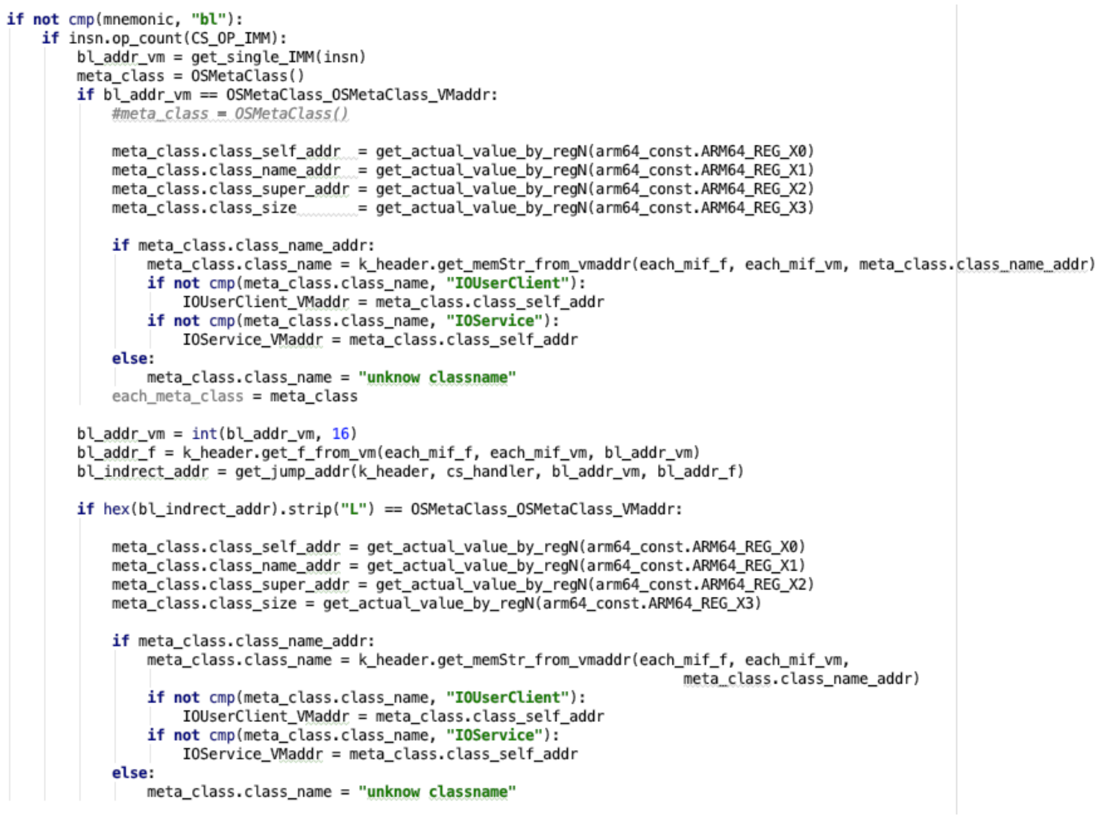
- Analysis key functions based on CFG
- Emulate key instructions operation
- Output User Client or external method information

Emulate key instructions operation

- ARM Emulator
  - adrp/adr, add, mov/movz, orr, ldr, bl…
- X86_64 Emulator
  - lea, mov, call, cmp, jz, je…

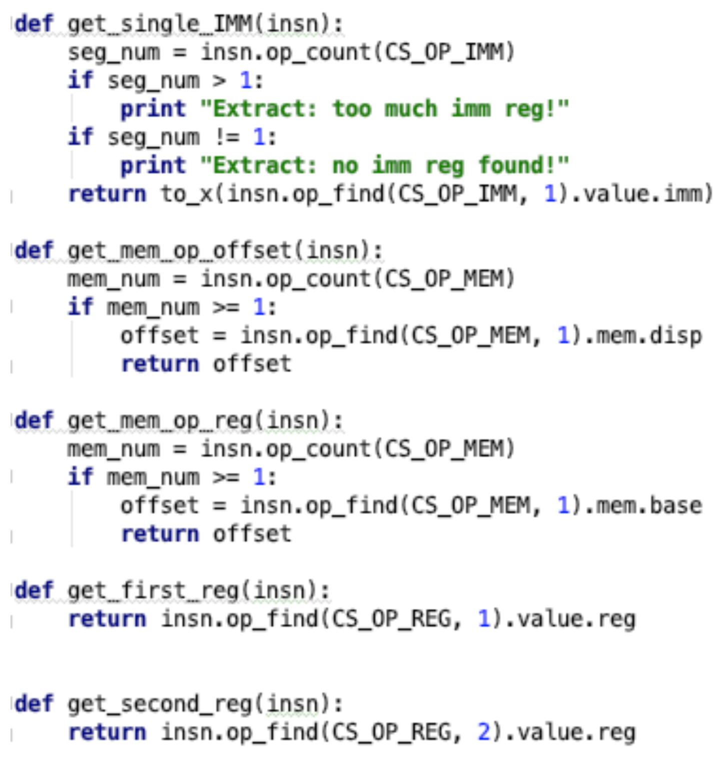


```python
if not cmp(mnemonic, "str"):
    reg_num = insn.op_count(CS_OP_REG)
    if reg_num == 1:
        continue
    f_reg = get_first_reg(insn)
    if f_reg == arm64_const.ARM64_REG_XZR or f_reg == arm64_const.ARM64_REG_D0 or\
                f_reg == arm64_const.ARM64_REG_WZR:
        continue
    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_self_addr):
            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_vm)
    else:
        f_reg_v_f = k_header.get_f_from_vm(each_mif_f, each_mif_vm, f_reg_v_vm)

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

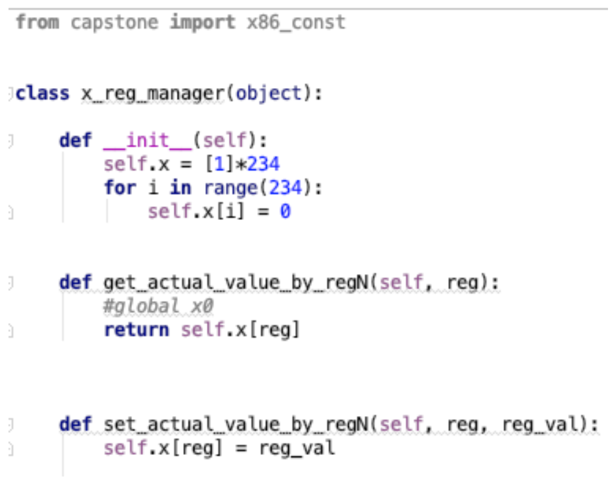


```python
if not cmp(mnemonic, "bl"):
    if insn.op_count(CS_OP_IMM):
        bl_addr_vm = get_single_IMM(insn)
        meta_class = OSMetaClass()
        if bl_addr_vm == OSMetaClass_OSMetaClass_VMaddr:
            #meta_class = OSMetaClass()

            meta_class.class_self_addr  = get_actual_value_by_regN(arm64_const.ARM64_REG_X0)
            meta_class.class_name_addr  = get_actual_value_by_regN(arm64_const.ARM64_REG_X1)
            meta_class.class_super_addr = get_actual_value_by_regN(arm64_const.ARM64_REG_X2)
            meta_class.class_size       = get_actual_value_by_regN(arm64_const.ARM64_REG_X3)

            if meta_class.class_name_addr:
                meta_class.class_name = k_header.get_memStr_from_vmaddr(each_mif_f, each_mif_vm, meta_class.class_name_addr)
                if not cmp(meta_class.class_name, "IOUserClient"):
                    IOUserClient_VMaddr = meta_class.class_self_addr
                if not cmp(meta_class.class_name, "IOService"):
                    IOService_VMaddr = meta_class.class_self_addr
            else:
                meta_class.class_name = "unknow classname"
            each_meta_class = meta_class

        bl_addr_vm = int(bl_addr_vm, 16)
        bl_addr_f = k_header.get_f_from_vm(each_mif_f, each_mif_vm, bl_addr_vm)
        bl_indrect_addr = get_jump_addr(k_header, cs_handler, bl_addr_vm, bl_addr_f)

        if hex(bl_indrect_addr).strip("L") == OSMetaClass_OSMetaClass_VMaddr:

            meta_class.class_self_addr = get_actual_value_by_regN(arm64_const.ARM64_REG_X0)
            meta_class.class_name_addr = get_actual_value_by_regN(arm64_const.ARM64_REG_X1)
            meta_class.class_super_addr = get_actual_value_by_regN(arm64_const.ARM64_REG_X2)
            meta_class.class_size = get_actual_value_by_regN(arm64_const.ARM64_REG_X3)

            if meta_class.class_name_addr:
                meta_class.class_name = k_header.get_memStr_from_vmaddr(each_mif_f, each_mif_vm,
                                                                        meta_class.class_name_addr)
                if not cmp(meta_class.class_name, "IOUserClient"):
                    IOUserClient_VMaddr = meta_class.class_self_addr
                if not cmp(meta_class.class_name, "IOService"):
                    IOService_VMaddr = meta_class.class_self_addr
            else:
                meta_class.class_name = "unknow classname"
```

Emulate register operation



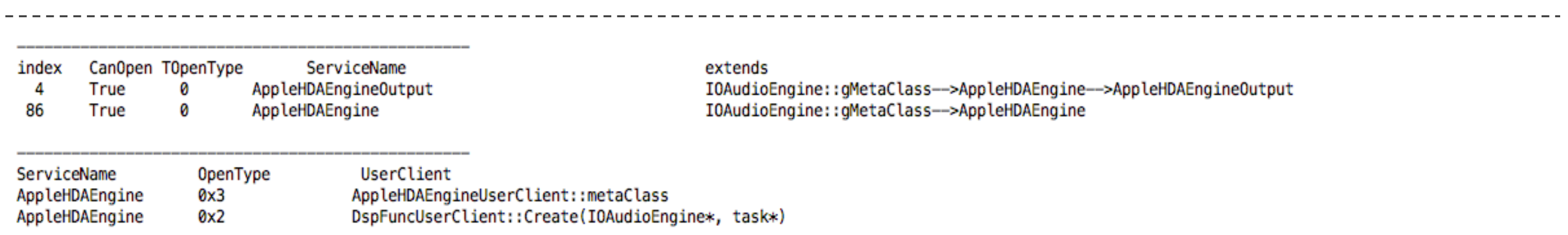
```python
def get_single_IMM(insn):
    seg_num = insn.op_count(CS_OP_IMM)
    if seg_num > 1:
        print "Extract: too much imm reg!"
    if seg_num != 1:
        print "Extract: no imm reg found!"
    return to_x(insn.op_find(CS_OP_IMM, 1).value.imm)

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
```



```python
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
```

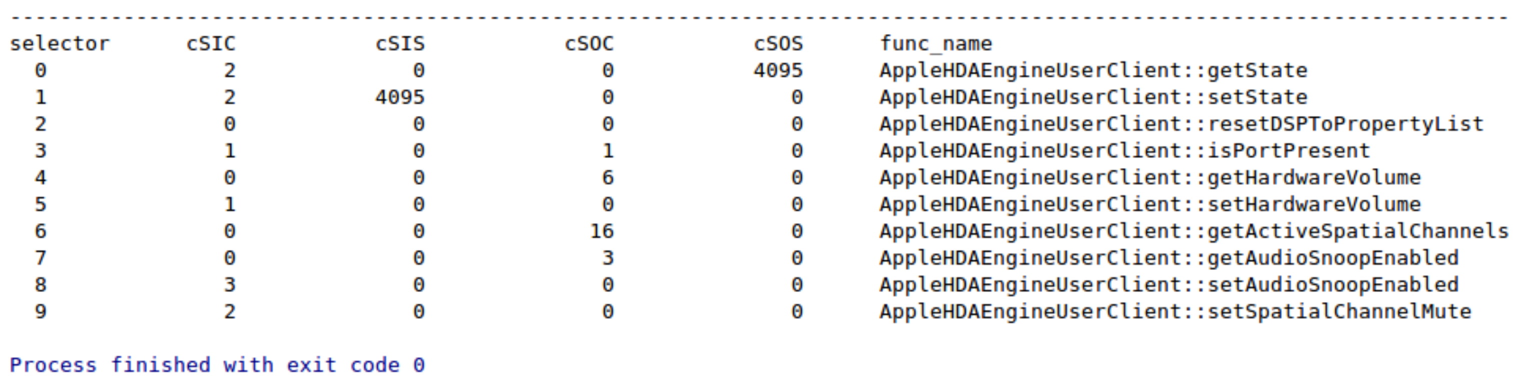
Output User Client or external method information

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
```

Agenda

- About US
- Solution Overview
- Static Analysis for Kernel and KEXTs Attack Interface
  - KEXTs Interfaces Vector Automatic Generate
  - Kernel Interfaces Vector Automatic Generate
  - Kernel/KEXTs Interfaces Diff Analysis
- Dynamic Analysis for Kernel and UserMode Attack Interface
- Automatic Fuzzing solution
- 0Day vulnerabilities found

Kernel Interfaces

Implementation files in XNU source code

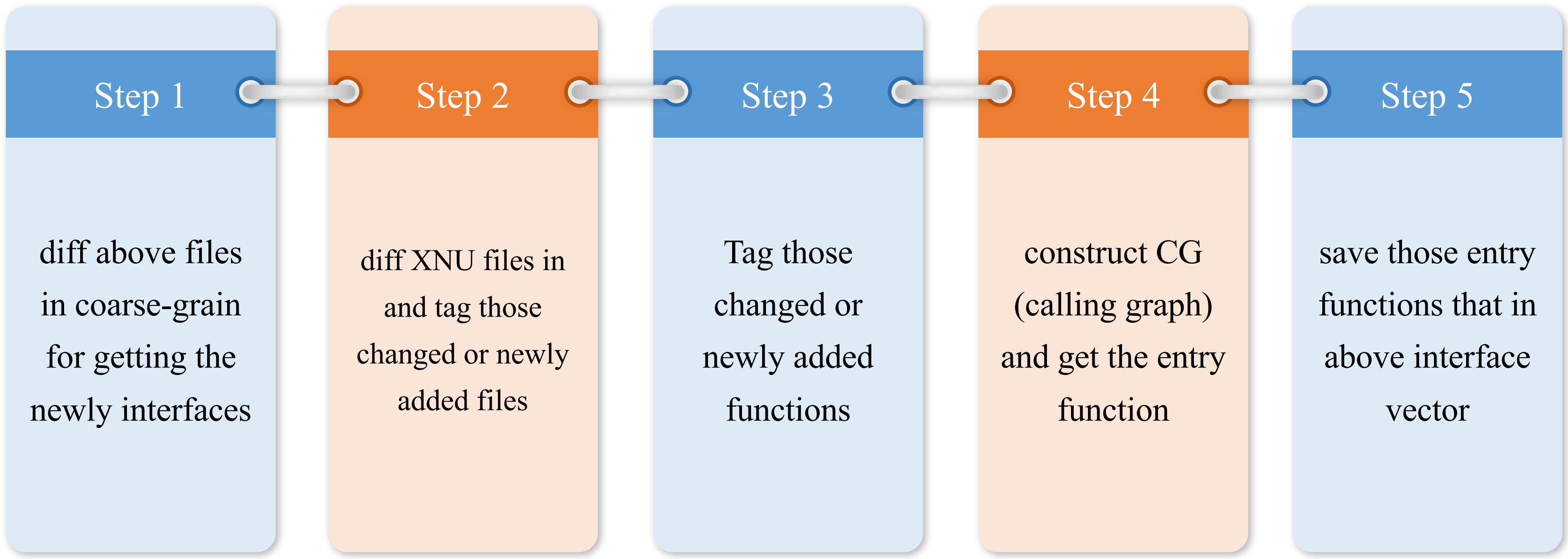
```
BUILD
 └──── OBJ
     ├──── DEBUG_X86_64
     │     └──── bsd
     │         └──── DEBUG
     │             └──── init_sysent.c
     └──── EXPORT_HDRS
         └──── bsd
             └──── sys
                 ├──── syscall.h
                 ├──── sysproto.h
                 └──── systm.h
```

```
-- osfmk
   |-- device
   |   `-- device.defs            //mig
   |-- kern
   |   `-- syscall_sw.c           //traps
   `-- mach
       `-- mach_traps.h           //traps
```

Agenda

- About US
- Solution Overview
- Static Analysis for Kernel and KEXTs Attack Interface
  - KEXTs Interfaces Vector Automatic Generate
  - Kernel Interfaces Vector Automatic Generate
  - Kernel/KEXTs Interfaces Diff Analysis
- Dynamic Analysis for Kernel and UserMode Attack Interface
- Automatic Fuzzing solution
- 0Day vulnerabilities found

Kernel Diff Methodology

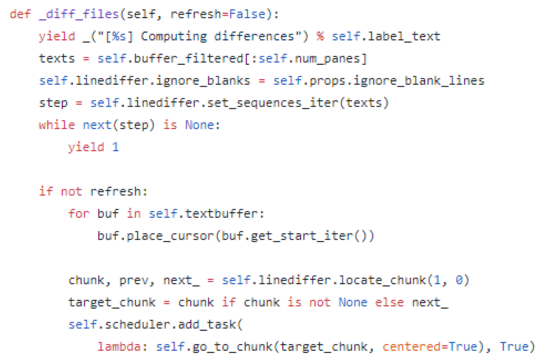


Step 1
diff above files in coarse-grain for getting the newly interfaces
Step 2
diff XNU files in and tag those changed or newly added files
Step 3
Tag those changed or newly added functions
Step 4
construct CG (calling graph) and get the entry function
Step 5
save those entry functions that in above interface vector

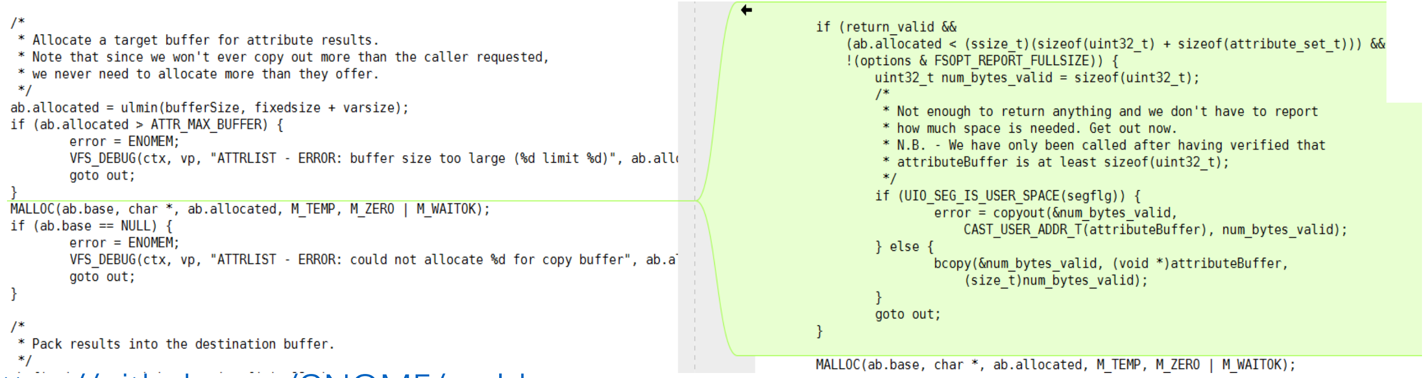
Kernel Diff Analysis Practice (1/2)

Updated Function "getvolattrlist ()" in vfs_attrlist.c

- Diff kernel source code files and find the difference functions
- Meld: A open source visual diff and merge tool
- CVE-2018-4243 Patch



```python
def _diff_files(self, refresh=False):
    yield _("[%s] Computing differences") % self.label_text
    texts = self.buffer_filtered[:self.num_panes]
    self.linediffer.ignore_blanks = self.props.ignore_blank_lines
    step = self.linediffer.set_sequences_iter(texts)
    while next(step) is None:
        yield 1

    if not refresh:
        for buf in self.textbuffer:
            buf.place_cursor(buf.get_start_iter())

        chunk, prev, next_ = self.linediffer.locate_chunk(1, 0)
        target_chunk = chunk if chunk is not None else next_
        self.scheduler.add_task(
            lambda: self.go_to_chunk(target_chunk, centered=True), True)
```



```c
/*
 * Allocate a target buffer for attribute results.
 * Note that since we won't ever copy out more than the caller requested,
 * we never need to allocate more than they offer.
 */
ab.allocated = ulmin(bufferSize, fixedsize + varsize);
if (ab.allocated > ATTR_MAX_BUFFER) {
        error = ENOMEM;
        VFS_DEBUG(ctx, vp, "ATTRLIST - ERROR: buffer size too large (%d limit %d)", ab.all
        goto out;
}
MALLOC(ab.base, char *, ab.allocated, M_TEMP, M_ZERO | M_WAITOK);
if (ab.base == NULL) {
        error = ENOMEM;
        VFS_DEBUG(ctx, vp, "ATTRLIST - ERROR: could not allocate %d for copy buffer", ab.a
        goto out;
}

/*
 * Pack results into the destination buffer.
 */
```

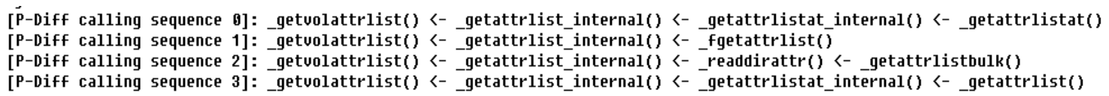
```c
if (return_valid &&
    (ab.allocated < (ssize_t)(sizeof(uint32_t) + sizeof(attribute_set_t))) &&
    !(options & FSOPT_REPORT_FULLSIZE)) {
        uint32_t num_bytes_valid = sizeof(uint32_t);
        /*
         * Not enough to return anything and we don't have to report
         * how much space is needed. Get out now.
         * N.B. - We have only been called after having verified that
         * attributeBuffer is at least sizeof(uint32_t);
         */
        if (UIO_SEG_IS_USER_SPACE(segflg)) {
                error = copyout(&num_bytes_valid,
                    CAST_USER_ADDR_T(attributeBuffer), num_bytes_valid);
        } else {
                bcopy(&num_bytes_valid, (void *)attributeBuffer,
                    (size_t)num_bytes_valid);
        }
        goto out;
}

MALLOC(ab.base, char *, ab.allocated, M_TEMP, M_ZERO | M_WAITOK);
```

https://github.com/GNOME/meld

Kernel Diff Analysis Practice (2/2)

getvolattrlist () for example

- P-diff: A tool Implemented by IDA pro script
  - ✓ Construct CG for getvolattrlist function
  - ✓ List and report calling sequence that entry function in attack vector
  - ✓ CodeRefsTo(ea, flow) and CodeRefsFrom(ea, flow)



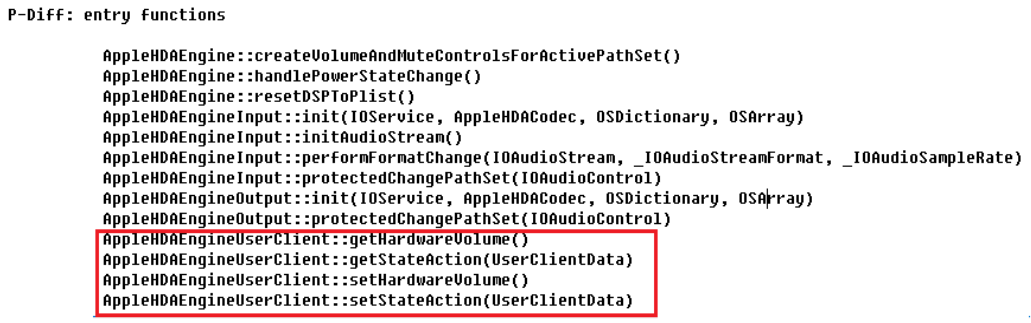
```
[P-Diff calling sequence 0]: _getvolattrlist() <- _getattrlist_internal() <- _getattrlistat_internal() <- _getattrlistat()
[P-Diff calling sequence 1]: _getvolattrlist() <- _getattrlist_internal() <- _fgetattrlist()
[P-Diff calling sequence 2]: _getvolattrlist() <- _getattrlist_internal() <- _readdirattr() <- _getattrlistbulk()
[P-Diff calling sequence 3]: _getvolattrlist() <- _getattrlist_internal() <- _getattrlistat_internal() <- _getattrlist()
```

220 AUE_GETATTRLIST ALL { int getattrlist(const char *path, struct attrlist *alist, void *attributeBuffer, size_t bufferSize, u_long options) NO_SYSCALL_STUB; }
461 AUE_GETATTRLISTBULK ALL { int getattrlistbulk(int dirfd, struct attrlist *alist, void *attributeBuffer, size_t bufferSize, uint64_t options); }
228 AUE_FGETATTRLIST ALL { int fgetattrlist(int fd, struct attrlist *alist, void *attributeBuffer, size_t bufferSize, u_long options); }
476 AUE_GETATTRLISTAT ALL { int getattrlistat(int fd, const char *path, struct attrlist *alist, void *attributeBuffer, size_t bufferSize, u_long options); }

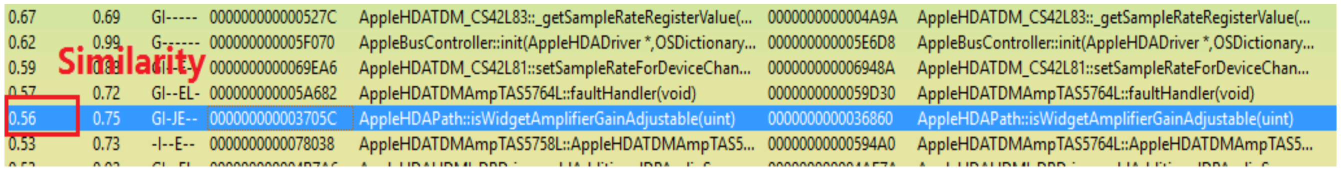
KEXTs Diff Analysis

Difference with Kernel and KEXTs

- KEXTs are closed source
- Using IDA pro script with plugin "Bindiff“



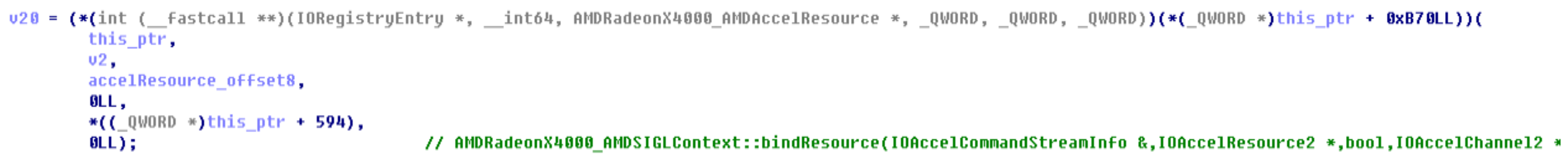
```
P-Diff: entry functions

         AppleHDAEngine::createVolumeAndMuteControlsForActivePathSet()
         AppleHDAEngine::handlePowerStateChange()
         AppleHDAEngine::resetDSPToPlist()
         AppleHDAEngineInput::init(IOService, AppleHDACodec, OSDictionary, OSArray)
         AppleHDAEngineInput::initAudioStream()
         AppleHDAEngineInput::performFormatChange(IOAudioStream, _IOAudioStreamFormat, _IOAudioSampleRate)
         AppleHDAEngineInput::protectedChangePathSet(IOAudioControl)
         AppleHDAEngineOutput::init(IOService, AppleHDACodec, OSDictionary, OSArray)
         AppleHDAEngineOutput::protectedChangePathSet(IOAudioControl)
         AppleHDAEngineUserClient::getHardwareVolume()
         AppleHDAEngineUserClient::getStateAction(UserClientData)
         AppleHDAEngineUserClient::setHardwareVolume()
         AppleHDAEngineUserClient::setStateAction(UserClientData)
```



0.67	0.69	GI-----	000000000000527C	AppleHDATDM_CS42L83::_getSampleRateRegisterValue(...	0000000000004A9A	AppleHDATDM_CS42L83::_getSampleRateRegisterValue(...
0.62	0.99	G------	000000000005F070	AppleBusController::init(AppleHDADriver *,OSDictionary...	000000000005E6D8	AppleBusController::init(AppleHDADriver *,OSDictionary...
0.59	[illegible]	[illegible]	0000000000069EA6	AppleHDATDM_CS42L81::setSampleRateForDeviceChan...	000000000006948A	AppleHDATDM_CS42L81::setSampleRateForDeviceChan...
0.57	0.72	GI--EL-	000000000005A682	AppleHDATDMAmpTAS5764L::faultHandler(void)	0000000000059D30	AppleHDATDMAmpTAS5764L::faultHandler(void)
0.56	0.75	GI-JE--	000000000003705C	AppleHDAPath::isWidgetAmplifierGainAdjustable(uint)	0000000000036860	AppleHDAPath::isWidgetAmplifierGainAdjustable(uint)
0.53	0.73	-I--E--	0000000000078038	AppleHDATDMAmpTAS5758L::AppleHDATDMAmpTAS5...	00000000000594A0	AppleHDATDMAmpTAS5764L::AppleHDATDMAmpTAS5...

Disadvantages about KEXTs static analysis

- KEXTs are closed source, many method strings are stripped
- Function call usually use *(object_ptr + offset) type
    - ✓ difficult to construct CFG, which bring noise for KEXTs vector analysis
    - ✓ difficult to construct CG, which bring noise for KEXTs diff analysis



```
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 *
```

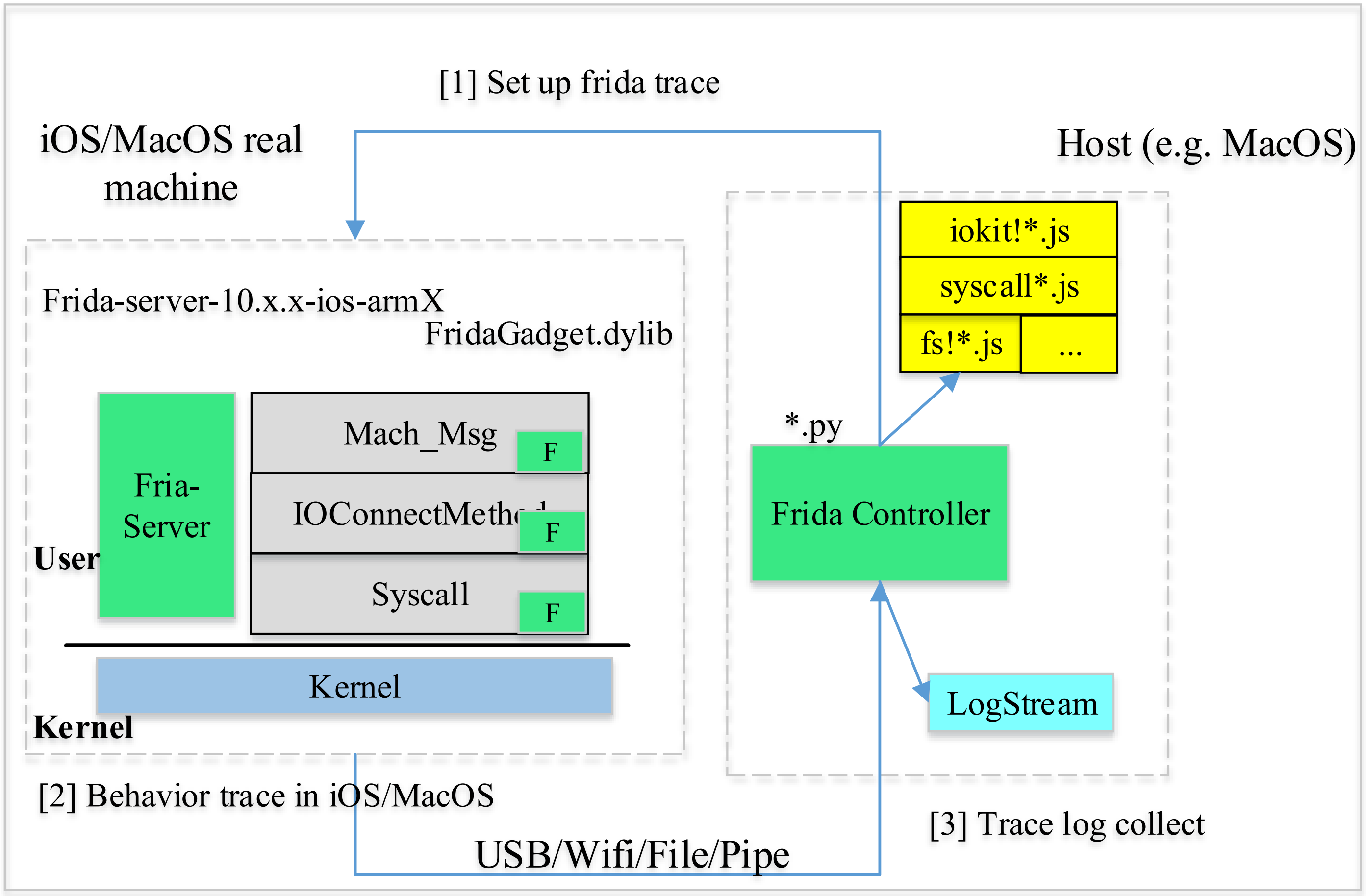
Agenda

- About US
- Solution Overview
- Static Analysis for Kernel and KEXTs Attack Interface
  - KEXTs Interfaces Vector Automatic Generate
  - Kernel Interfaces Vector Automatic Generate
  - Kernel/KEXTs Interfaces Diff Analysis
- Dynamic Analysis for Kernel and UserMode Attack Interface
- Automatic Fuzzing solution
- 0Day vulnerabilities found

Comparison of dynamic trace

	User Trace	Kernel Trace	Embedded in OS	Any priviledge?	Support script?	Performance	Platform
Frida	Yes	No	No	Root or Repack	Yes	Middle	iOS/Osx
Dtrace	No	Yes	Yes	Root	Yes	High	Osx
lldb	Yes	Yes	Yes	Root	Yes	Low	iOS/Osx
Kernel hook	---	Yes	No	Root	No	Middle	Osx

Frida Hook in User Mode



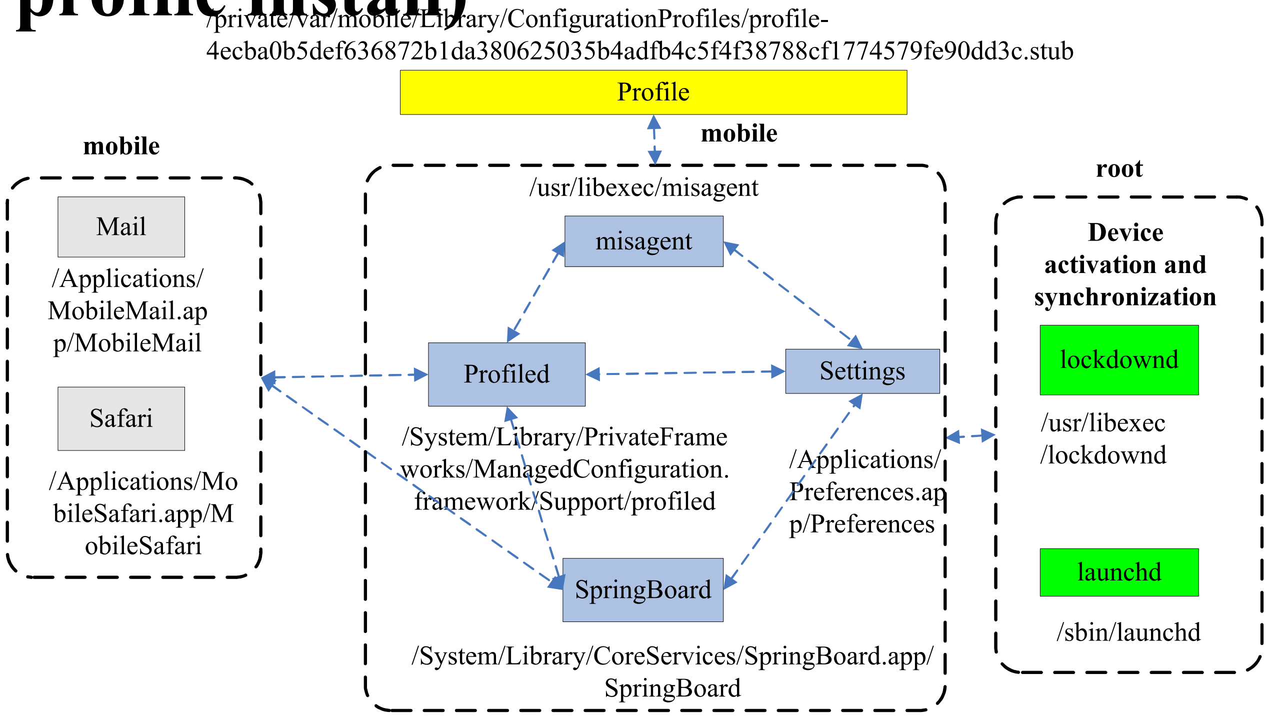
xpc_connection_send_message API context (e.g.)

{"time":"2017-09-18T10:38:32.807Z",
"txnType":"moony?",
"lib":"libxpc.dylib",
"method":"**xpc_connection_send_message**",
"artifact":[
	{"name":"connection",
	"value":"0x1658d090",
	"argSeq":0},

	{"name":"connectioninfo",
	"value":"\tconnection=0x1658d090\tconnectionName=\tconnectionPid=2312\tconnectionProcName=Preferences",
	"argSeq":0},

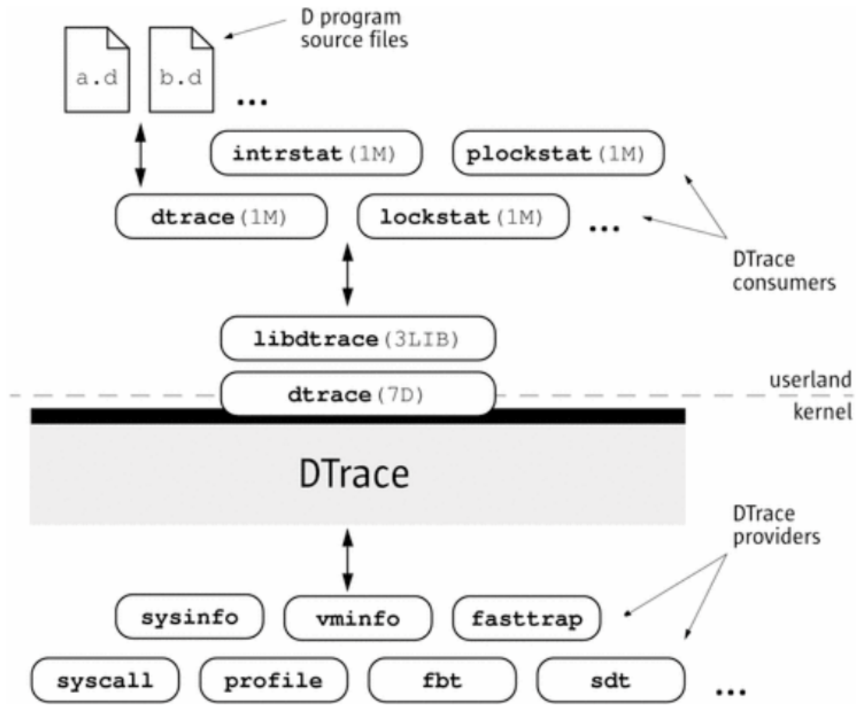
	{"name":"retval","value":374477440,"argSeq":-1}
]}

Hunt more dynamic relation if you like (e.g. profile install)


/private/var/mobile/Library/ConfigurationProfiles/profile-4ecba0b5def636872b1da380625035b4adfb4c5f4f38788cf1774579fe90dd3c.stub
Profile
mobile
mobile
root
Mail
/Applications/MobileMail.app/MobileMail
Safari
/Applications/MobileSafari.app/MobileSafari
/usr/libexec/misagent
misagent
Profiled
Settings
/System/Library/PrivateFrameworks/ManagedConfiguration.framework/Support/profiled
/Applications/Preferences.app/Preferences
SpringBoard
/System/Library/CoreServices/SpringBoard.app/SpringBoard
Device activation and synchronization
lockdownd
/usr/libexec/lockdownd
launchd
/sbin/launchd

```
Profiled←--→misagent
xpc_connection_get_name_by_address:     com.apple.misagent
[2916:289299   (profiled)]:   libxpc.dylib!xpc_connection_send_message_with_reply_sync      :      connection=0x1575c100
connectionName=com.apple.misagent      connectionPid=3115      connectionProcName=misagent
[2916:205566   (profiled)]:   libsystem_kernel.dylib!__read_nocancel  call stack:
0x26032dcb Foundation!_NSReadFromFileDescriptorWithProgress,
0x26032c11 Foundation!_NSReadBytesFromFileWithExtendedAttributes,
```

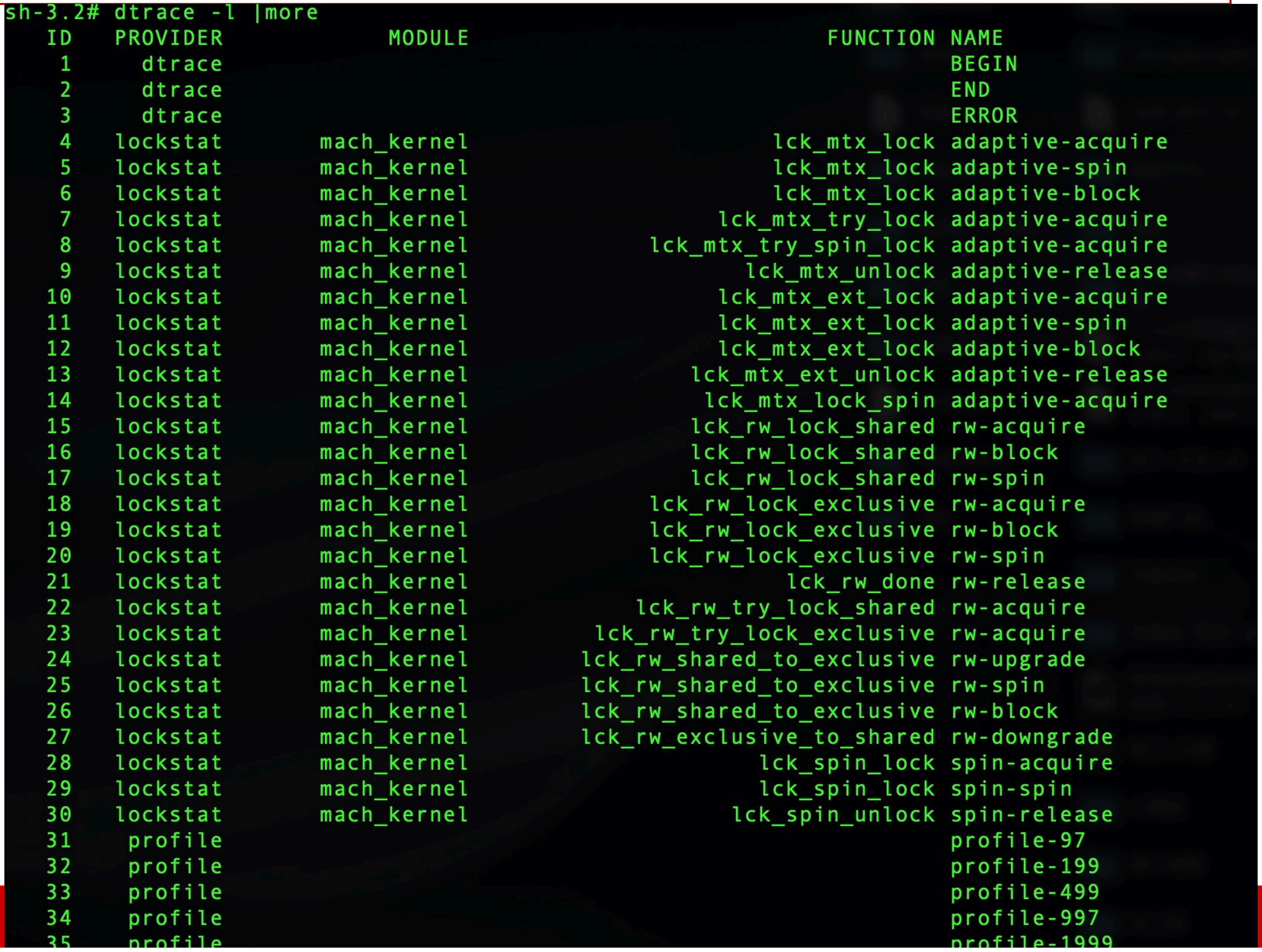
Dtrace introduction



Source: Solaris Dynamic Tracing Guide

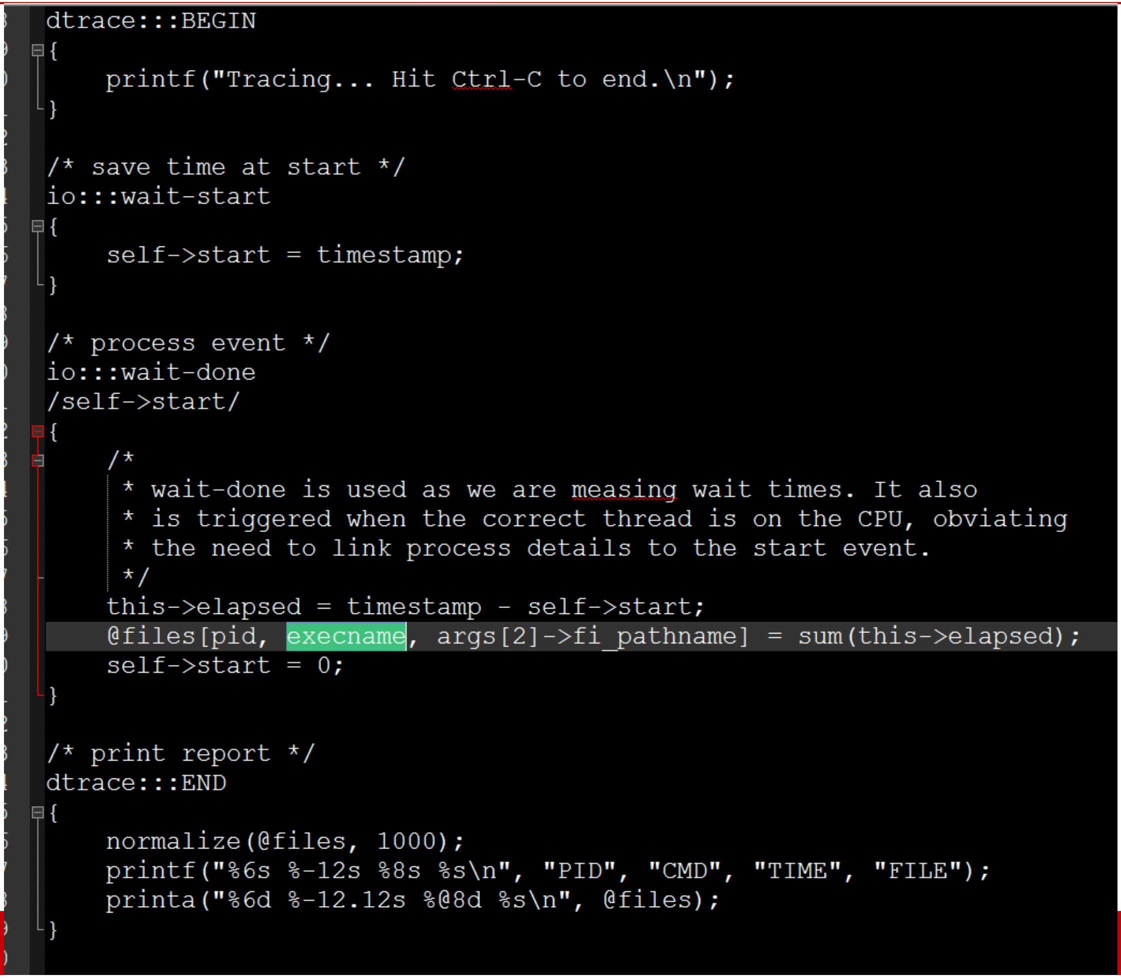
Dtrace providers list

The typical interface for fuzzing:
system call,
IOKit,
mach msg,
network,
Disk,
XPC
…



```
sh-3.2# dtrace -l |more
   ID   PROVIDER            MODULE                          FUNCTION NAME
    1     dtrace                                                     BEGIN
    2     dtrace                                                     END
    3     dtrace                                                     ERROR
    4   lockstat          mach_kernel                     lck_mtx_lock adaptive-acquire
    5   lockstat          mach_kernel                     lck_mtx_lock adaptive-spin
    6   lockstat          mach_kernel                     lck_mtx_lock adaptive-block
    7   lockstat          mach_kernel                 lck_mtx_try_lock adaptive-acquire
    8   lockstat          mach_kernel            lck_mtx_try_spin_lock adaptive-acquire
    9   lockstat          mach_kernel                   lck_mtx_unlock adaptive-release
   10   lockstat          mach_kernel                 lck_mtx_ext_lock adaptive-acquire
   11   lockstat          mach_kernel                 lck_mtx_ext_lock adaptive-spin
   12   lockstat          mach_kernel                 lck_mtx_ext_lock adaptive-block
   13   lockstat          mach_kernel               lck_mtx_ext_unlock adaptive-release
   14   lockstat          mach_kernel                lck_mtx_lock_spin adaptive-acquire
   15   lockstat          mach_kernel               lck_rw_lock_shared rw-acquire
   16   lockstat          mach_kernel               lck_rw_lock_shared rw-block
   17   lockstat          mach_kernel               lck_rw_lock_shared rw-spin
   18   lockstat          mach_kernel            lck_rw_lock_exclusive rw-acquire
   19   lockstat          mach_kernel            lck_rw_lock_exclusive rw-block
   20   lockstat          mach_kernel            lck_rw_lock_exclusive rw-spin
   21   lockstat          mach_kernel                      lck_rw_done rw-release
   22   lockstat          mach_kernel           lck_rw_try_lock_shared rw-acquire
   23   lockstat          mach_kernel        lck_rw_try_lock_exclusive rw-acquire
   24   lockstat          mach_kernel       lck_rw_shared_to_exclusive rw-upgrade
   25   lockstat          mach_kernel       lck_rw_shared_to_exclusive rw-spin
   26   lockstat          mach_kernel       lck_rw_shared_to_exclusive rw-block
   27   lockstat          mach_kernel       lck_rw_exclusive_to_shared rw-downgrade
   28   lockstat          mach_kernel                    lck_spin_lock spin-acquire
   29   lockstat          mach_kernel                    lck_spin_lock spin-spin
   30   lockstat          mach_kernel                  lck_spin_unlock spin-release
   31    profile                                                     profile-97
   32    profile                                                     profile-199
   33    profile                                                     profile-499
   34    profile                                                     profile-997
   35    profile                                                     profile-1999
```

Dtrace script (e.g. file probe)

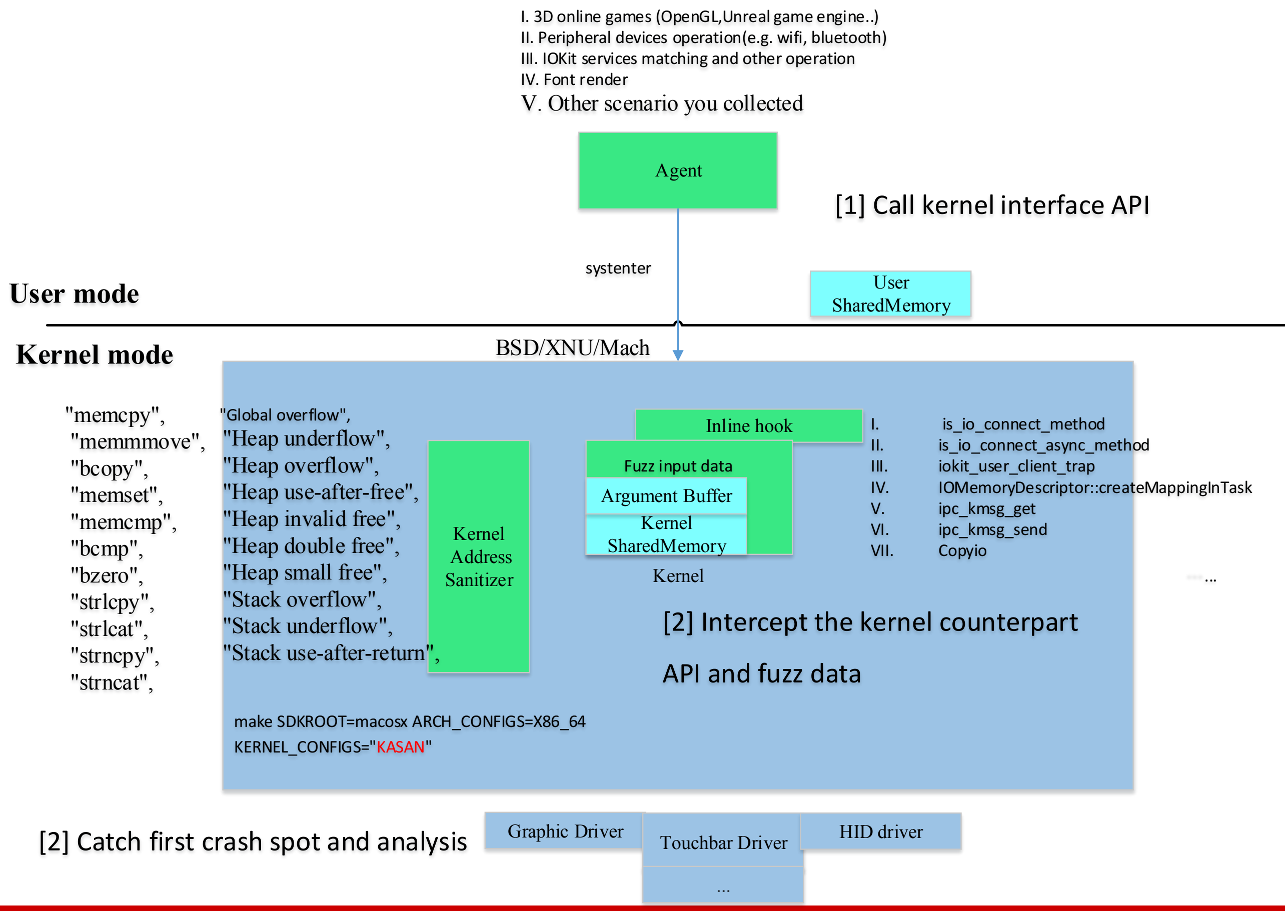


```
dtrace:::BEGIN
{
    printf("Tracing... Hit Ctrl-C to end.\n");
}

/* save time at start */
io:::wait-start
{
    self->start = timestamp;
}

/* process event */
io:::wait-done
/self->start/
{
    /*
     * wait-done is used as we are measing wait times. It also
     * is triggered when the correct thread is on the CPU, obviating
     * the need to link process details to the start event.
     */
    this->elapsed = timestamp - self->start;
    @files[pid, execname, args[2]->fi_pathname] = sum(this->elapsed);
    self->start = 0;
}

/* print report */
dtrace:::END
{
    normalize(@files, 1000);
    printf("%6s %-12s %8s %s\n", "PID", "CMD", "TIME", "FILE");
    printa("%6d %-12.12s %@8d %s\n", @files);
}
```

Agenda

- About US
- Solution Overview
- Static Analysis for Kernel and KEXTs Attack Interface
  - KEXTs Interfaces Vector Automatic Generate
  - Kernel Interfaces Vector Automatic Generate
  - Kernel/KEXTs Interfaces Diff Analysis
- Dynamic Analysis for Kernel and UserMode Attack Interface
- Automatic Fuzzing solution
- 0Day vulnerabilities found

Enhanced kernel fuzz

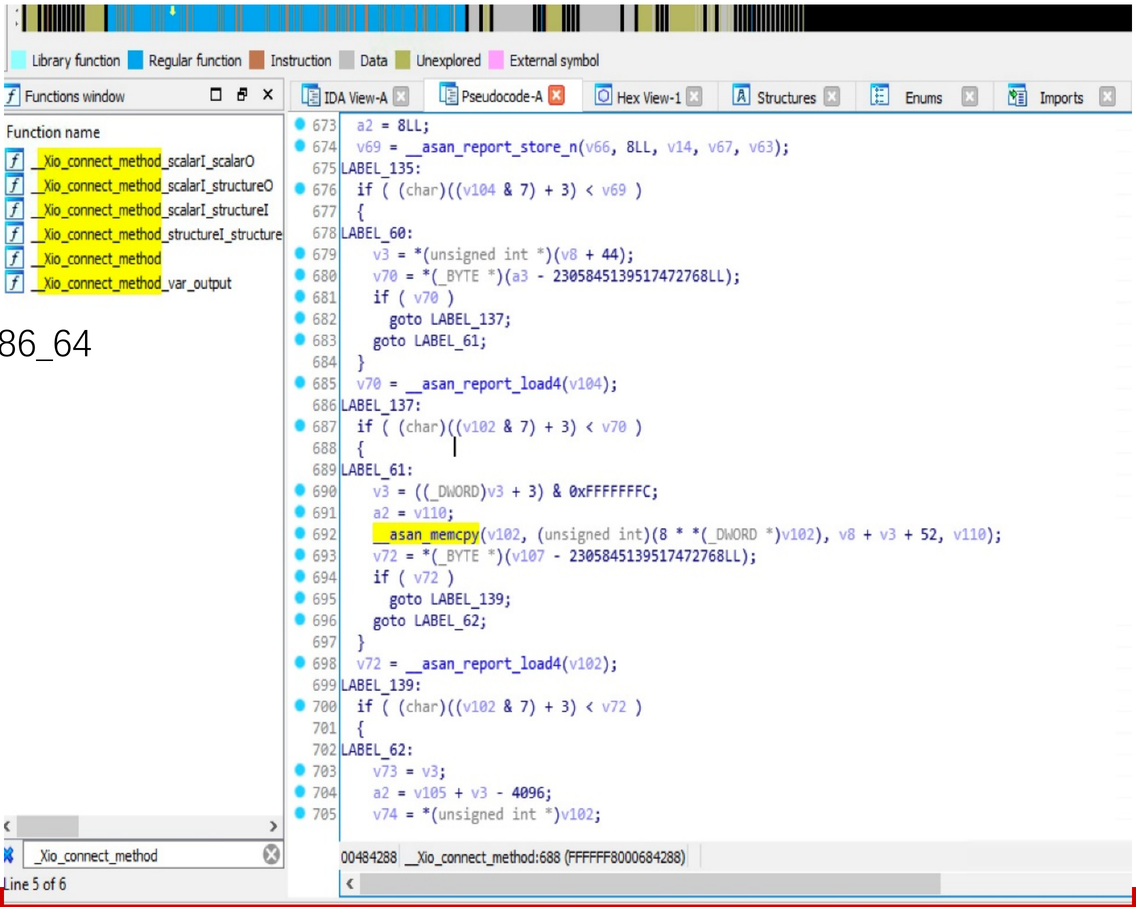


```
make SDKROOT=macosx ARCH_CONFIGS=X86_64
KERNEL_CONFIGS="KASAN"
```

KASAN in iOS/OSX Kernel

```
make SDKROOT=macosx ARCH_CONFIGS=X86_64
KERNEL_CONFIGS="KASAN  "
```

```
/System/Library/Kernenls/kernel*
```



```
__Xio_connect_method_scalarI_scalarO
__Xio_connect_method_scalarI_structureO
__Xio_connect_method_scalarI_structureI
__Xio_connect_method_structureI_structure
__Xio_connect_method
__Xio_connect_method_var_output
```

```
  a2 = 8LL;
  v69 = __asan_report_store_n(v66, 8LL, v14, v67, v63);
LABEL_135:
  if ( (char)((v104 & 7) + 3) < v69 )
  {
LABEL_60:
    v3 = *(unsigned int *)(v8 + 44);
    v70 = *(_BYTE *)(a3 - 2305845139517472768LL);
    if ( v70 )
      goto LABEL_137;
    goto LABEL_61;
  }
  v70 = __asan_report_load4(v104);
LABEL_137:
  if ( (char)((v102 & 7) + 3) < v70 )
  {
LABEL_61:
    v3 = ((_DWORD)v3 + 3) & 0xFFFFFFFC;
    a2 = v110;
    __asan_memcpy(v102, (unsigned int)(8 * *(_DWORD *)v102), v8 + v3 + 52, v110);
    v72 = *(_BYTE *)(v107 - 2305845139517472768LL);
    if ( v72 )
      goto LABEL_139;
    goto LABEL_62;
  }
  v72 = __asan_report_load4(v102);
LABEL_139:
  if ( (char)((v102 & 7) + 3) < v72 )
  {
LABEL_62:
    v73 = v3;
    a2 = v105 + v3 - 4096;
    v74 = *(unsigned int *)v102;
```

```
00484288 __Xio_connect_method:688 (FFFFFF8000684288)
```

Future plan

- **Syzkaller-like Fuzzing in Kernel Mode**

    There would be existing many environment preparation or initialization(e.g. Open correct service , initialize the target devices and send the correct mach message id) before one special kernel API (e.g. IOConnectionCallMethod) could work properly in kernel.

    So why do not we intercept the kernel API at the proper time under proper state and fuzz it like AFL in kernel mode directly?

- **Porting KASAN/KMSAN for close-source driver**

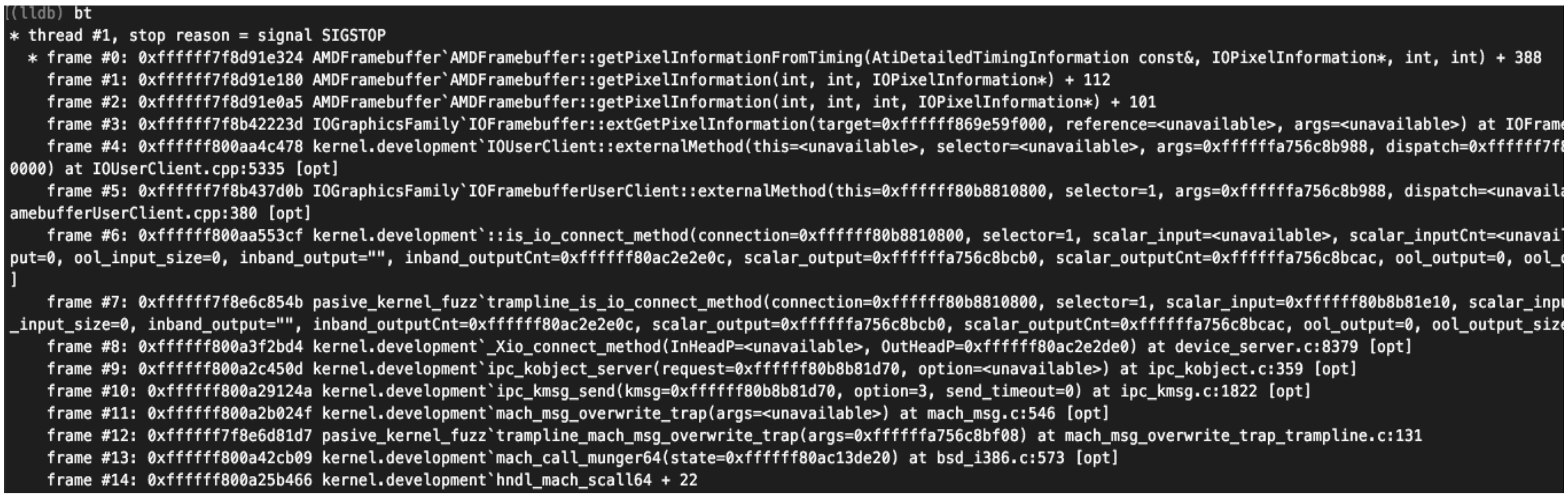
    Modifying the memory API to asan_* API in import table in driver module, or patch the code with memory management in driver to support kernel sanitizer could be as further research.

Agenda

- About US
- Solution Overview
- Static Analysis for Kernel and KEXTs Attack Interface
  - KEXTs Interfaces Vector Automatic Generate
  - Kernel Interfaces Vector Automatic Generate
  - Kernel/KEXTs Interfaces Diff Analysis
- Dynamic Analysis for Kernel and UserMode Attack Interface
- Automatic Fuzzing solution
- 0Day vulnerabilities found

CVE-2018-4462

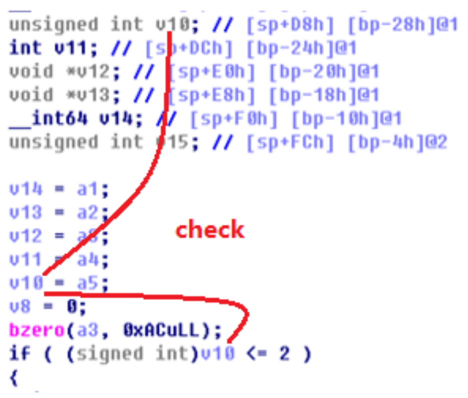
Integer overflow vulnerability in AMDFramebuffer driver



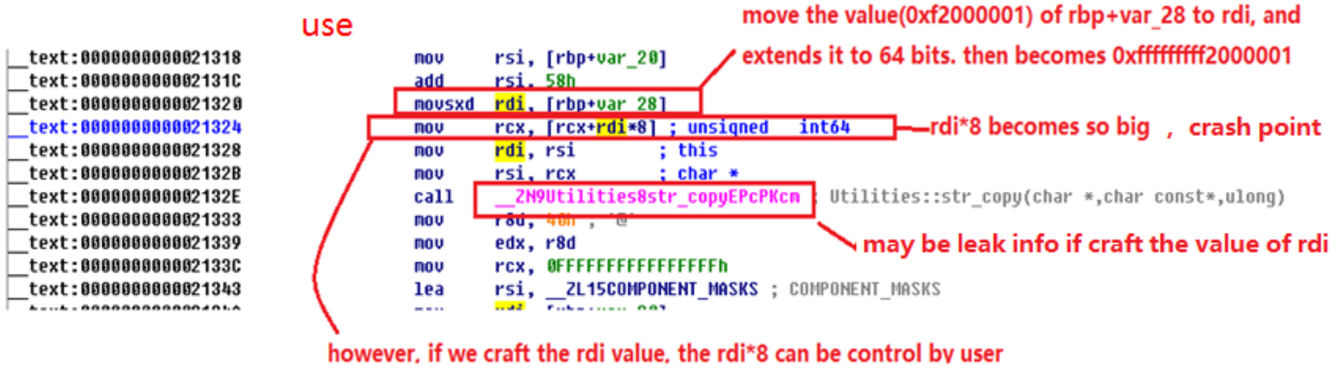
```
[(lldb) bt
* thread #1, stop reason = signal SIGSTOP
  * frame #0: 0xffffff7f8d91e324 AMDFramebuffer`AMDFramebuffer::getPixelInformationFromTiming(AtiDetailedTimingInformation const&, IOPixelInformation*, int, int) + 388
    frame #1: 0xffffff7f8d91e180 AMDFramebuffer`AMDFramebuffer::getPixelInformation(int, 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
```

CVE-2018-4462 – Root Cause

```
frame #0: 0xffffff7f8d91e324 AMDFramebuffer`AMDFramebuffer::getPixelInformationFromTiming(AtiDetailedTimingInformation const&, IOPixelInformation*, int, int)
+ 388
AMDFramebuffer`AMDFramebuffer::getPixelInformationFromTiming:
->  0xffffff7f8d91e324 <+388>: movq   (%rcx,%rdi,8), %rcx
    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 = 0xfffffffff2000001
```



```
unsigned int v10; // [sp+D8h] [bp-28h]@1
int v11; // [sp+DCh] [bp-24h]@1
void *v12; // [sp+E0h] [bp-20h]@1
void *v13; // [sp+E8h] [bp-18h]@1
__int64 v14; // [sp+F0h] [bp-10h]@1
unsigned int v15; // [sp+FCh] [bp-4h]@2

v14 = a1;
v13 = a2;
v12 = a3;
v11 = a4;
v10 = a5;
v8 = 0;
bzero(a3, 0xACuLL);
if ( (signed int)v10 <= 2 )
{
```



```
__text:0000000000021318     mov     rsi, [rbp+var_20]
__text:000000000002131C     add     rsi, 58h
__text:0000000000021320     movsxd  rdi, [rbp+var_28]
__text:0000000000021324     mov     rcx, [rcx+rdi*8] ; unsigned   int64
__text:0000000000021328     mov     rdi, rsi         ; this
__text:000000000002132B     mov     rsi, rcx         ; char *
__text:000000000002132E     call    __ZN9Utilities8str_copyEPcPKcm ; Utilities::str_copy(char *,char const*,ulong)
__text:0000000000021333     mov     r8d, 40h ; '@'
__text:0000000000021339     mov     edx, r8d
__text:000000000002133C     mov     rcx, 0FFFFFFFFFFFFFFFFh
__text:0000000000021343     lea     rsi, __ZL15COMPONENT_MASKS ; COMPONENT_MASKS
```

use
move the value(0xf2000001) of rbp+var_28 to rdi, and
extends it to 64 bits. then becomes 0xfffffffff2000001
rdi*8 becomes so big , crash point
may be leak info if craft the value of rdi
however, if we craft the rdi value, the rdi*8 can be control by user

OOB read in AMDRadeonX4000 Extension

OOB Read Vulnerability Found in AMDRadeonX4000_AMDAccelResource Initialize Process

----

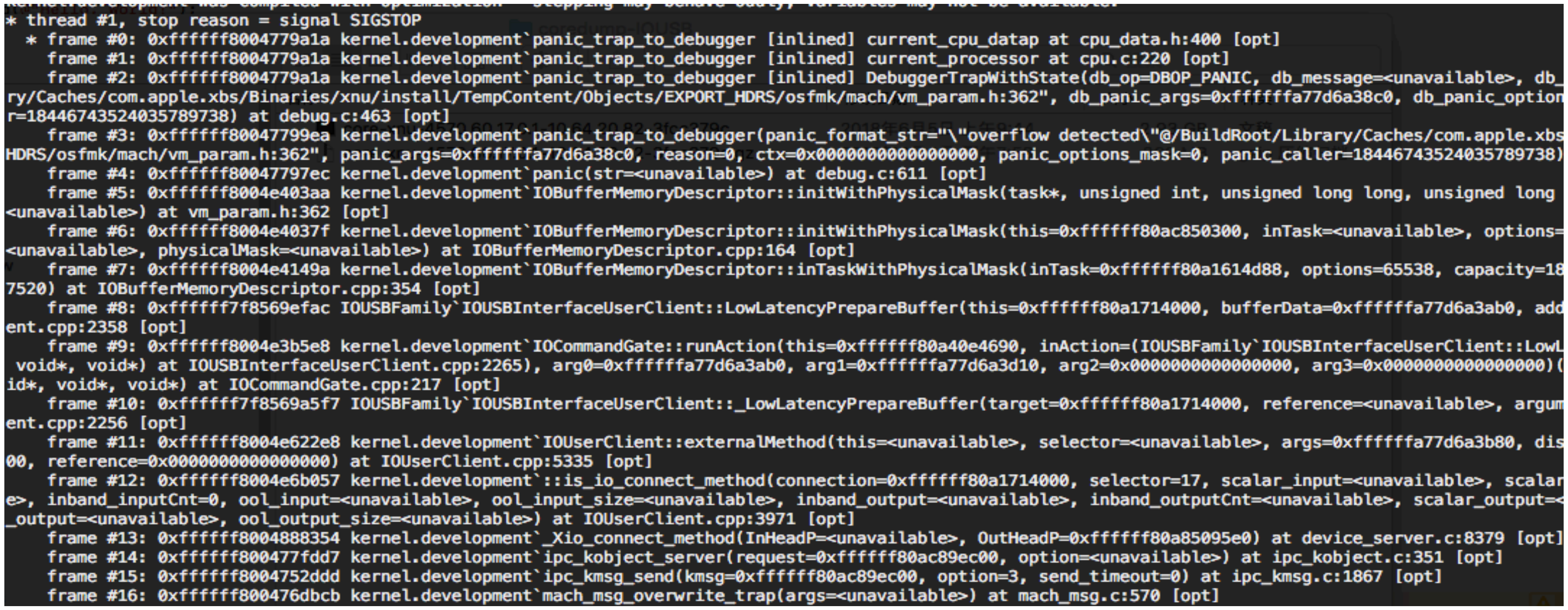
```
* thread #1, stop reason = signal SIGSTOP
  * frame #0: 0xffffff7fa00965d3 AMDRadeonX4000`AMDRadeonX4000_AMDAccelResource::initialize(IOAccelNewResourceArgs*, unsigned long long) + 1525
    frame #1: 0xffffff7f9fea346b IOAcceleratorFamily2`IOAccelSharedUserClient2::new_resource(IOAccelNewResourceArgs*, IOAccelNewResourceReturnData*, unsigned long long, unsigned int*) + 1893
    frame #2: 0xffffff7f9fea4a41 IOAcceleratorFamily2`IOAccelSharedUserClient2::s_new_resource(IOAccelSharedUserClient2*, void*, IOExternalMethodArguments*) + 151
    frame #3: 0xffffff801d625ab8 kernel.development`IOUserClient::externalMethod(this=<unavailable>, selector=<unavailable>, args=0xffffff83dd4b3b58, dispatch=0xffffff7f9fee8260,
target=0xffffff80854fd780, reference=0x0000000000000000) at IOUserClient.cpp:5358 [opt]
    frame #4: 0xffffff7f9fea4d98 IOAcceleratorFamily2`IOAccelSharedUserClient2::externalMethod(unsigned int, IOExternalMethodArguments*, IOExternalMethodDispatch*, OSObject*, void*) + 120
    frame #5: 0xffffff801d62eb7f kernel.development`::is_io_connect_method(connection=0xffffff80854fd780, selector=0, scalar_input=<unavailable>, scalar_inputCnt=<unavailable>,
inband_input=<unavailable>, inband_inputCnt=2424, ool_input=0, ool_input_size=0, inband_output="", inband_outputCnt=0xffffff806ba03e0c, scalar_output=0xffffff83dd4b3ce0,
scalar_outputCnt=0xffffff83dd4b3cdc, ool_output=0, ool_output_size=0xffffff8085919d5c) at IOUserClient.cpp:3994 [opt]
    frame #6: 0xffffff801cfbbce4 kernel.development`_Xio_connect_method(InHeadP=<unavailable>, OutHeadP=0xffffff806ba03de0) at device_server.c:8379 [opt]
    frame #7: 0xffffff801ce8d27d kernel.development`ipc_kobject_server(request=0xffffff8085919000, option=<unavailable>) at ipc_kobject.c:359 [opt]
    frame #8: 0xffffff801ce59465 kernel.development`ipc_kmsg_send(kmsg=0xffffff8085919000, option=3, send_timeout=0) at ipc_kmsg.c:1832 [opt]
    frame #9: 0xffffff801ce78a75 kernel.development`mach_msg_overwrite_trap(args=<unavailable>) at mach_msg.c:549 [opt]
    frame #10: 0xffffff801cff6323 kernel.development`mach_call_munger64(state=0xffffff806ca9c480) at bsd_i386.c:573 [opt]
    frame #11: 0xffffff801ce23486 kernel.development`hndl_mach_scall64 + 22
```

OOB read – Root Cause

```
__text:000000000000E58E loc_E58E:                ; CODE XREF: AMDRadeonX4000_AMDAccelResource::initialize(IOAccelNewResourceArgs *,ulong long)+58Dj
__text:000000000000E58E          mov    ecx, [r15+0F8h]
__text:000000000000E595          test   rcx, rcx
__text:000000000000E598          jz     short loc_E603
__text:000000000000E59A          shl    rcx, 3
__text:000000000000E59E          lea    rdi, [rcx+rcx*2]
__text:000000000000E5A2          call   _IOMalloc
__text:000000000000E5A7          mov    [r12+178h], rax  --- rax== buffer address which create by IOMalloc        ------- (a)
__text:000000000000E5AF          test   rax, rax
__text:000000000000E5B2          jz     short loc_E62A
__text:000000000000E5B4          or     byte ptr [r12+186h], 8
__text:000000000000E5BD          mov    ecx, [r15+0F8h]  --------r15==structureInput,    ecx=( (uint32_t*) structureInput+62)     ---------(b)
__text:000000000000E5C4          mov    [r12+180h], ecx
__text:000000000000E5CC          test   rcx, rcx         ------ test rcx, if zero, break        ----------(c)
__text:000000000000E5CF          jz     short loc_E639
__text:000000000000E5D1          xor    edx, edx         ------- index                          ---------(d)
__text:000000000000E5D3
__text:000000000000E5D3 loc_E5D3:                ; CODE XREF: AMDRadeonX4000_AMDAccelResource::initialize(IOAccelNewResourceArgs *,ulong long)+621j
__text:000000000000E5D3          mov    rsi, [r15+rdx+98h]     ---- mov structureInput+rdx+0x98 to rsi
__text:000000000000E5DB          mov    [rax+rdx], rsi         ----mov rsi to rax+rdx, rax== buffer address which create by IOMalloc  ----------(g)
__text:000000000000E5DF          mov    rsi, [r15+rdx+0A0h]
__text:000000000000E5E7          mov    [rax+rdx+8], rsi
__text:000000000000E5EC          mov    esi, [r15+rdx+0A8h]
__text:000000000000E5F4          mov    [rax+rdx+10h], esi
__text:000000000000E5F8          add    rdx, 18h                                                -------(e)
__text:000000000000E5FC          dec    rcx                    ----decrease rcx value           --------(f)
__text:000000000000E5FF          jnz    short loc_E5D3
```

OverFlow in IOUSBFamliy Extension

OverFlow bug due to No Boundary Check



```
* thread #1, stop reason = signal SIGSTOP
  * frame #0: 0xffffff8004779a1a kernel.development`panic_trap_to_debugger [inlined] current_cpu_datap at cpu_data.h:400 [opt]
    frame #1: 0xffffff8004779a1a kernel.development`panic_trap_to_debugger [inlined] current_processor at cpu.c:220 [opt]
    frame #2: 0xffffff8004779a1a kernel.development`panic_trap_to_debugger [inlined] DebuggerTrapWithState(db_op=DBOP_PANIC, db_message=<unavailable>, db_
ry/Caches/com.apple.xbs/Binaries/xnu/install/TempContent/Objects/EXPORT_HDRS/osfmk/mach/vm_param.h:362", db_panic_args=0xffffffa77d6a38c0, db_panic_option
r=18446743524035789738) at debug.c:463 [opt]
    frame #3: 0xffffff80047799ea kernel.development`panic_trap_to_debugger(panic_format_str="\"overflow detected\"@/BuildRoot/Library/Caches/com.apple.xbs
HDRS/osfmk/mach/vm_param.h:362", panic_args=0xffffffa77d6a38c0, reason=0, ctx=0x0000000000000000, panic_options_mask=0, panic_caller=18446743524035789738)
    frame #4: 0xffffff80047797ec kernel.development`panic(str=<unavailable>) at debug.c:611 [opt]
    frame #5: 0xffffff8004e403aa kernel.development`IOBufferMemoryDescriptor::initWithPhysicalMask(task*, unsigned int, unsigned long long, unsigned long
<unavailable>) at vm_param.h:362 [opt]
    frame #6: 0xffffff8004e4037f kernel.development`IOBufferMemoryDescriptor::initWithPhysicalMask(this=0xffffff80ac850300, inTask=<unavailable>, options=
<unavailable>, physicalMask=<unavailable>) at IOBufferMemoryDescriptor.cpp:164 [opt]
    frame #7: 0xffffff8004e4149a kernel.development`IOBufferMemoryDescriptor::inTaskWithPhysicalMask(inTask=0xffffff80a1614d88, options=65538, capacity=18
7520) at IOBufferMemoryDescriptor.cpp:354 [opt]
    frame #8: 0xffffff7f8569efac IOUSBFamily`IOUSBInterfaceUserClient::LowLatencyPrepareBuffer(this=0xffffff80a1714000, bufferData=0xffffffa77d6a3ab0, add
ent.cpp:2358 [opt]
    frame #9: 0xffffff8004e3b5e8 kernel.development`IOCommandGate::runAction(this=0xffffff80a40e4690, inAction=(IOUSBFamily`IOUSBInterfaceUserClient::LowL
 void*, void*) at IOUSBInterfaceUserClient.cpp:2265), arg0=0xffffffa77d6a3ab0, arg1=0xffffffa77d6a3d10, arg2=0x0000000000000000, arg3=0x0000000000000000)(
id*, void*, void*) at IOCommandGate.cpp:217 [opt]
    frame #10: 0xffffff7f8569a5f7 IOUSBFamily`IOUSBInterfaceUserClient::_LowLatencyPrepareBuffer(target=0xffffff80a1714000, reference=<unavailable>, argum
ent.cpp:2256 [opt]
    frame #11: 0xffffff8004e622e8 kernel.development`IOUserClient::externalMethod(this=<unavailable>, selector=<unavailable>, args=0xffffffa77d6a3b80, dis
00, reference=0x0000000000000000) at IOUserClient.cpp:5335 [opt]
    frame #12: 0xffffff8004e6b057 kernel.development`::is_io_connect_method(connection=0xffffff80a1714000, selector=17, scalar_input=<unavailable>, scalar
e>, inband_inputCnt=0, ool_input=<unavailable>, ool_input_size=<unavailable>, inband_output=<unavailable>, inband_outputCnt=<unavailable>, scalar_output=<
_output=<unavailable>, ool_output_size=<unavailable>) at IOUserClient.cpp:3971 [opt]
    frame #13: 0xffffff8004888354 kernel.development`_Xio_connect_method(InHeadP=<unavailable>, OutHeadP=0xffffff80a85095e0) at device_server.c:8379 [opt]
    frame #14: 0xffffff800477fdd7 kernel.development`ipc_kobject_server(request=0xffffff80ac89ec00, option=<unavailable>) at ipc_kobject.c:351 [opt]
    frame #15: 0xffffff8004752ddd kernel.development`ipc_kmsg_send(kmsg=0xffffff80ac89ec00, option=3, send_timeout=0) at ipc_kmsg.c:1867 [opt]
    frame #16: 0xffffff800476dbcb kernel.development`mach_msg_overwrite_trap(args=<unavailable>) at mach_msg.c:570 [opt]
```

OverFlow – Root Cause

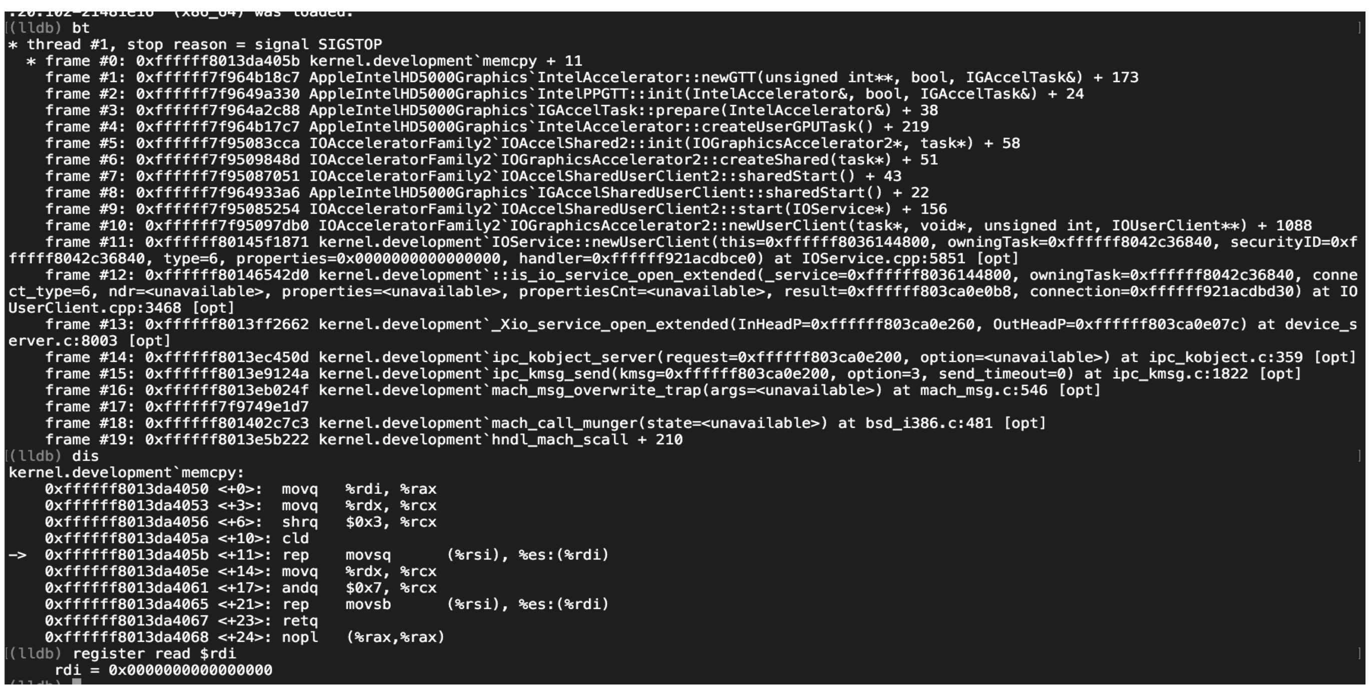
```
frame #7: 0xffffff8004a4149a kernel.development`IOBufferMemoryDescriptor::inTaskWithPhysicalMask(inTask=0xffffff80a10ebd88,
options=65538, capacity=18446744073709551615, physicalMask=18446744073709547520) at
IOBufferMemoryDescriptor.cpp:354 [opt]
```

18446744073709551615 = 0xffffffffffffffff

```
(lldb) f 8
IOUSBFamily was compiled with optimization - stepping may behave oddly; variables may not be available.
frame #8: 0xffffff7f8529efac IOUSBFamily`IOUSBInterfaceUserClient::LowLatencyPrepareBuffer(this=0xffffff80a5dbd800,
bufferData=0xffffffa750b2bab0, addrOut=0xffffffa750b2ba00) at IOUSBInterfaceUserClient.cpp:2358 [opt]
(lldb) fr v
(IOUSBInterfaceUserClientV3 *) this = 0xffffff80a5dbd800
(LowLatencyUserBufferInfoV3 *) bufferData = 0xffffffa750b2bab0
(uint64_t *) addrOut = 0xffffffa750b2ba00
(bool) preparedIOMD = false
(uint64_t) dataBuffer = 0
(IOReturn) ret = <variable not available>

(IOUSBDevice *) device = <variable not available>

(IOOptionBits) optionBits = 0
(mach_vm_address_t) physicalMask = <variable not available>

(IOBufferMemoryDescriptor *) dataBufferDescriptor = <register rax is not available>
 …
 …
(lldb) memory read --size 8 --format x --count 12 0xffffffa750b2bab0
0xffffffa750b2bab0: 0x0e38340d99444cc1 0x0000000000000049
0xffffffa750b2bac0: 0xffffffffffffffff 0x0000000000000000
0xffffffa750b2bad0: 0x0000000000000008 0xffffff8004cc0590
0xffffffa750b2bae0: 0x0000000000000000 0xffffffa750b2bbac
0xffffffa750b2baf0: 0xffffff8004cc08f0 0xffffff7f852ede68
0xffffffa750b2bb00: 0xffffffa750b2bb80 0x0000000000000000
(lldb)
```

NULL PAGE Reference in IntelAccelerator

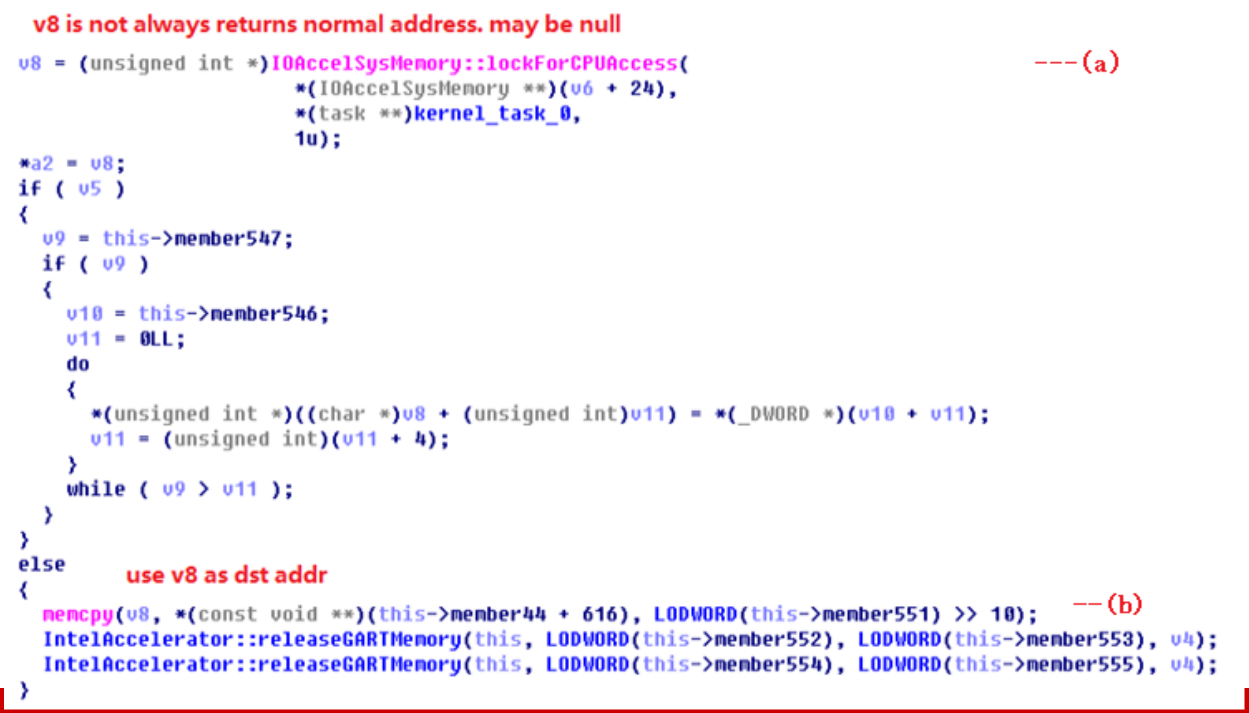
NULL PAGE Reference bug found in IntelAccelerator



```
.20.102-21481e10  (x86_64) was loaded.
(lldb) bt
* thread #1, stop reason = signal SIGSTOP
  * frame #0: 0xffffff8013da405b kernel.development`memcpy + 11
    frame #1: 0xffffff7f964b18c7 AppleIntelHD5000Graphics`IntelAccelerator::newGTT(unsigned int**, bool, IGAccelTask&) + 173
    frame #2: 0xffffff7f9649a330 AppleIntelHD5000Graphics`IntelPPGTT::init(IntelAccelerator&, bool, IGAccelTask&) + 24
    frame #3: 0xffffff7f964a2c88 AppleIntelHD5000Graphics`IGAccelTask::prepare(IntelAccelerator&) + 38
    frame #4: 0xffffff7f964b17c7 AppleIntelHD5000Graphics`IntelAccelerator::createUserGPUTask() + 219
    frame #5: 0xffffff7f95083cca IOAcceleratorFamily2`IOAccelShared2::init(IOGraphicsAccelerator2*, task*) + 58
    frame #6: 0xffffff7f9509848d IOAcceleratorFamily2`IOGraphicsAccelerator2::createShared(task*) + 51
    frame #7: 0xffffff7f95087051 IOAcceleratorFamily2`IOAccelSharedUserClient2::sharedStart() + 43
    frame #8: 0xffffff7f964933a6 AppleIntelHD5000Graphics`IGAccelSharedUserClient::sharedStart() + 22
    frame #9: 0xffffff7f95085254 IOAcceleratorFamily2`IOAccelSharedUserClient2::start(IOService*) + 156
    frame #10: 0xffffff7f95097db0 IOAcceleratorFamily2`IOGraphicsAccelerator2::newUserClient(task*, void*, unsigned int, IOUserClient**) + 1088
    frame #11: 0xffffff80145f1871 kernel.development`IOService::newUserClient(this=0xffffff8036144800, owningTask=0xffffff8042c36840, securityID=0xf
fffff8042c36840, type=6, properties=0x0000000000000000, handler=0xffffff921acdbce0) at IOService.cpp:5851 [opt]
    frame #12: 0xffffff80146542d0 kernel.development`::is_io_service_open_extended(_service=0xffffff8036144800, owningTask=0xffffff8042c36840, conne
ct_type=6, ndr=<unavailable>, properties=<unavailable>, propertiesCnt=<unavailable>, result=0xffffff803ca0e0b8, connection=0xffffff921acdbd30) at IO
UserClient.cpp:3468 [opt]
    frame #13: 0xffffff8013ff2662 kernel.development`_Xio_service_open_extended(InHeadP=0xffffff803ca0e260, OutHeadP=0xffffff803ca0e07c) at device_s
erver.c:8003 [opt]
    frame #14: 0xffffff8013ec450d kernel.development`ipc_kobject_server(request=0xffffff803ca0e200, option=<unavailable>) at ipc_kobject.c:359 [opt]
    frame #15: 0xffffff8013e9124a kernel.development`ipc_kmsg_send(kmsg=0xffffff803ca0e200, option=3, send_timeout=0) at ipc_kmsg.c:1822 [opt]
    frame #16: 0xffffff8013eb024f kernel.development`mach_msg_overwrite_trap(args=<unavailable>) at mach_msg.c:546 [opt]
    frame #17: 0xffffff7f9749e1d7
    frame #18: 0xffffff801402c7c3 kernel.development`mach_call_munger(state=<unavailable>) at bsd_i386.c:481 [opt]
    frame #19: 0xffffff8013e5b222 kernel.development`hndl_mach_scall + 210
(lldb) dis
kernel.development`memcpy:
    0xffffff8013da4050 <+0>:  movq   %rdi, %rax
    0xffffff8013da4053 <+3>:  movq   %rdx, %rcx
    0xffffff8013da4056 <+6>:  shrq   $0x3, %rcx
    0xffffff8013da405a <+10>: cld
->  0xffffff8013da405b <+11>: rep    movsq      (%rsi), %es:(%rdi)
    0xffffff8013da405e <+14>: movq   %rdx, %rcx
    0xffffff8013da4061 <+17>: andq   $0x7, %rcx
    0xffffff8013da4065 <+21>: rep    movsb      (%rsi), %es:(%rdi)
    0xffffff8013da4067 <+23>: retq
    0xffffff8013da4068 <+24>: nopl   (%rax,%rax)
(lldb) register read $rdi
     rdi = 0x0000000000000000
(lldb)
```

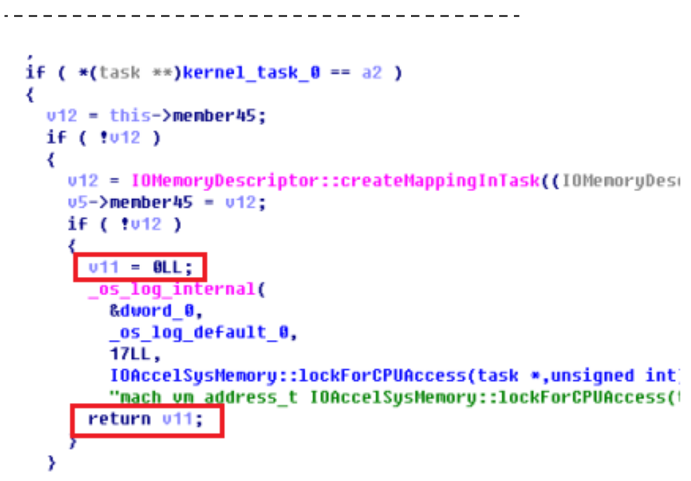
NULL PAGE Reference – Root Cause

Pseudo Code snippet of IntelAccelerator::newGTT function



v8 is not always returns normal address. may be null

```
v8 = (unsigned int *)IOAccelSysMemory::lockForCPUAccess(                        ---(a)
                       *(IOAccelSysMemory **)(v6 + 24),
                       *(task **)kernel_task_0,
                       1u);
*a2 = v8;
if ( v5 )
{
  v9 = this->member547;
  if ( v9 )
  {
    v10 = this->member546;
    v11 = 0LL;
    do
    {
      *(unsigned int *)((char *)v8 + (unsigned int)v11) = *(_DWORD *)(v10 + v11);
      v11 = (unsigned int)(v11 + 4);
    }
    while ( v9 > v11 );
  }
}
else        use v8 as dst addr
{
  memcpy(v8, *(const void **)(this->member44 + 616), LODWORD(this->member551) >> 10);   --(b)
  IntelAccelerator::releaseGARTMemory(this, LODWORD(this->member552), LODWORD(this->member553), v4);
  IntelAccelerator::releaseGARTMemory(this, LODWORD(this->member554), LODWORD(this->member555), v4);
}
```



```
  ;
  if ( *(task **)kernel_task_0 == a2 )
  {
    v12 = this->member45;
    if ( !v12 )
    {
      v12 = IOMemoryDescriptor::createMappingInTask((IOMemoryDes
      v5->member45 = v12;
      if ( !v12 )
      {
        v11 = 0LL;
        _os_log_internal(
          &dword_0,
          _os_log_default_0,
          17LL,
          IOAccelSysMemory::lockForCPUAccess(task *,unsigned int
          "mach_vm_address_t IOAccelSysMemory::lockForCPUAccess(
        return v11;
      }
    }
```

Divide Zero in AMDRadeonX4000 Extension

Divide Zero BUG found in IOAccelCommandQueue class

```
--------------------------------------------------------------------------------------------------------------------
(lldb) bt
* thread #1, stop reason = signal SIGSTOP
    frame #0: 0xffffff7f88b04941 AMDRadeonX4000`BltMgr::HwlOptimizeBufferBltRects(BltInfo*, unsigned int) + 879
    frame #1: 0xffffff7f88b1c474 AMDRadeonX4000`SiBltMgr::Adjust3dBltInfo(BltInfo*) + 662
    frame #2: 0xffffff7f88b1bfe2 AMDRadeonX4000`SiBltMgr::Execute3dBlt(BltInfo*) + 76
    frame #3: 0xffffff7f88b04241 AMDRadeonX4000`BltMgr::Memset(BltDevice*, _UBM_MEMSETINFO*) + 753
    frame #4: 0xffffff7f88a75506 AMDRadeonX4000`AMDRadeonX4000_AMDBltMgr::Memset(_UBM_MEMSETINFO*, _UBM_E_RETURNCODE*) + 28
    frame #5: 0xffffff7f88a73d29 AMDRadeonX4000`AMDRadeonX4000_AMDAtomicBlitManager::doMemset(_UBM_MEMSETINFO*, ABM_OPTIONS const*) + 263
    frame #6: 0xffffff7f88a56478 AMDRadeonX4000`AMDRadeonX4000_AMDAccelResource::initFillRegions() + 390
    frame #7: 0xffffff7f88a5c9ac AMDRadeonX4000`AMDRadeonX4000_AMDAccelResource::prepare() + 108
    frame #8: 0xffffff7f889b7c3e IOAcceleratorFamily2`IOAccelSegmentResourceList::prepare() + 48
    frame #9: 0xffffff7f889cbe94 IOAcceleratorFamily2`IOAccelCommandQueue::coalesceSegment(IOAccelCommandQueueSegment*, unsigned int*, IOAccelSegmentResourceList*, IOAccelKernelCommand
const*, IOAccelKernelCommand const*) + 78
    frame #10: 0xffffff7f889cc1ce IOAcceleratorFamily2`IOAccelCommandQueue::processCommandBuffer(unsigned int, unsigned int) + 666
    frame #11: 0xffffff7f889cd188 IOAcceleratorFamily2`IOAccelCommandQueue::process_command_buffer(unsigned int, unsigned int) + 924
    frame #12: 0xffffff7f889cb4c0 IOAcceleratorFamily2`IOAccelCommandQueue::submit_command_buffer(unsigned int, unsigned int, unsigned long long, unsigned long long) + 252
    frame #13: 0xffffff7f889cb2b9 IOAcceleratorFamily2`IOAccelCommandQueue::submit_command_buffers(IOAccelCommandQueueSubmitArgs const*) + 827
    frame #14: 0xffffff7f889ca2f4 IOAcceleratorFamily2`IOAccelCommandQueue::s_submit_command_buffers(IOAccelCommandQueue*, void*, IOExternalMethodArguments*) + 250
    frame #15: 0xffffff8006224978 kernel.development`IOUserClient::externalMethod(this=<unavailable>, selector=<unavailable>, args=0xffffffa753d3b9b8, dispatch=0xffffff7f889fac68, target=<unavailable>,
reference=<unavailable>) at IOUserClient.cpp:5689 [opt]
  * frame #16: 0xffffff800622da02 kernel.development`::is_io_connect_method(connection=<unavailable>, selector=1, scalar_input=<unavailable>, scalar_inputCnt=<unavailable>, inband_input=<unavailable>,
inband_inputCnt=32, ool_input=0, ool_input_size=0, inband_output="", inband_outputCnt=0xffffff80b137560c, scalar_output=0xffffffa753d3bce0, scalar_outputCnt=0xffffffa753d3bcdc, ool_output=0,
ool_output_size=0xffffff80b1ee3138) at IOUserClient.cpp:4304 [opt]
    frame #17: 0xffffff8005bbc386 kernel.development`_Xio_connect_method(InHeadP=<unavailable>, OutHeadP=0xffffffa753d3bce0) at device_server.c:8379 [opt]
    frame #18: 0xffffff8005a948fd kernel.development`ipc_kobject_server(request=0xffffff80b1ee3050, option=3) at ipc_kobject.c:361 [opt]
    frame #19: 0xffffff8005a6088e kernel.development`ipc_kmsg_send(kmsg=0xffffff80b1ee3050, option=3, send_timeout=0) at ipc_kmsg.c:1868 [opt]
    frame #20: 0xffffff8005a800e3 kernel.development`mach_msg_overwrite_trap(args=<unavailable>) at mach_msg.c:553 [opt]
    frame #21: 0xffffff8005bf702b kernel.development`mach_call_munger64(state=0xffffff80acb77100) at bsd_i386.c:580 [opt]
    frame #22: 0xffffff8005a2a476 kernel.development`hndl_mach_scall64 + 22
```

Divide Zero – Root Cause

ASM Code snippet of AMDRadeonX4000`BltMgr::HwlOptimizeBufferBltRects Function

```
------------------------------------------------------------------------------------------------
    __text:00000000000BB75B          div     esi
    __text:00000000000BB75D          mov     r14d, 0
    __text:00000000000BB763          mov     r12d, 0        -----init r12d with 0               --(a)
    __text:00000000000BB769          test    edx, edx
     …..
     -----omitted code ----
     …..
    __text:00000000000BB93C  loc_BB93C:                                           ; CODE  XREF:
    BltMgr::HwlOptimizeBufferBltRects(BltInfo *,uint)+3E1j
    __text:00000000000BB93C          xor     edx, edx
    __text:00000000000BB93E          mov     eax, r13d
    __text:00000000000BB941          div     r12d           -----r12d is not always nonzero     ---(b)
    __text:00000000000BB944          cmp     eax, r14d
    __text:00000000000BB947          jbe     short loc_BB95B
    __text:00000000000BB949          mov     dword ptr [rsi+rbx-0Ch], 0
    __text:00000000000BB951          mov     [rsi+rbx-4], r12d
    __text:00000000000BB956          mov     eax, r14d
    __text:00000000000BB959          jmp     short loc_BB97C
```

https://github.com/dongyangwu/p-joker

new version will release later