

Wow, PESSR has Eroded Apple in Blink

- Fun and Profit to Gain Dozens of iOS Vulnerabilities in Minutes by (P)ortable (E)xtensible (S)criptable (S)eed (R)eproducible Mobile Fuzzer

Ju Zhu
Moony Li
Lilang Wu

Agenda

- Who We Are
- PESSR Solution
 - Fuzz Strategy
 - Crash Monitor Strategy
 - Reproduce Strategy
 - Future Plan
- Hunt Practice via PESSR
 - Attack Interface



Ju Zhu

6+ years Security

Advanced Threat Research

Hunt 0Day/nDay

Vulnerability



Moony Li

@Flyic

Github: SilverMoonSecurity

8 years Security

OSX, Android, iOS Vulnerability Hunt and Exploit

Sandbox/Emulator Development



Lilang Wu

@Lilang_Wu

3 years Security

Mobile Advanced Threat Research

Mac/iOS Vulnerability/Malware

Agenda

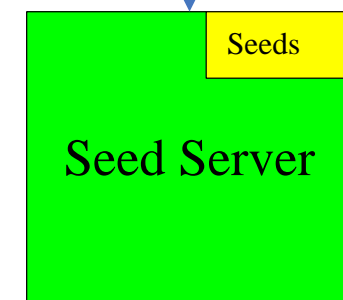
- Who We Are
- PESSR Solution
 - Fuzz Strategy
 - Crash Monitor Strategy
 - Reproduce Strategy
 - Future Plan
- Hunt Practice via PESSR
 - Attack Interface

Fuzzing Tool Compare

	Platform	Fuzz Interface (Extensibility)	Code Coverage	Drive by (Flexibility)	Repro by (Reducibility)	Portability Cost
Trinity	Linux	System Call (Middle)	No/Todo	Native Code + Config (Low)	Log (Log)	Middle
AFL	Any	Format Parse Logic (Middle)	Yes	Native Code + File (Middle)	File (Middle)	Middle
Triforce AFL	Qemu, Linux	Any in Guest OS (Middle)	Yes	Native Code + Config (Low)	Agent + Log (Middle/High)	High
PESSR	(Desktop, Mobile) Any	Any (High)	No/ToDo	All Script (High)	Seed (High)	Low

Solution Overview

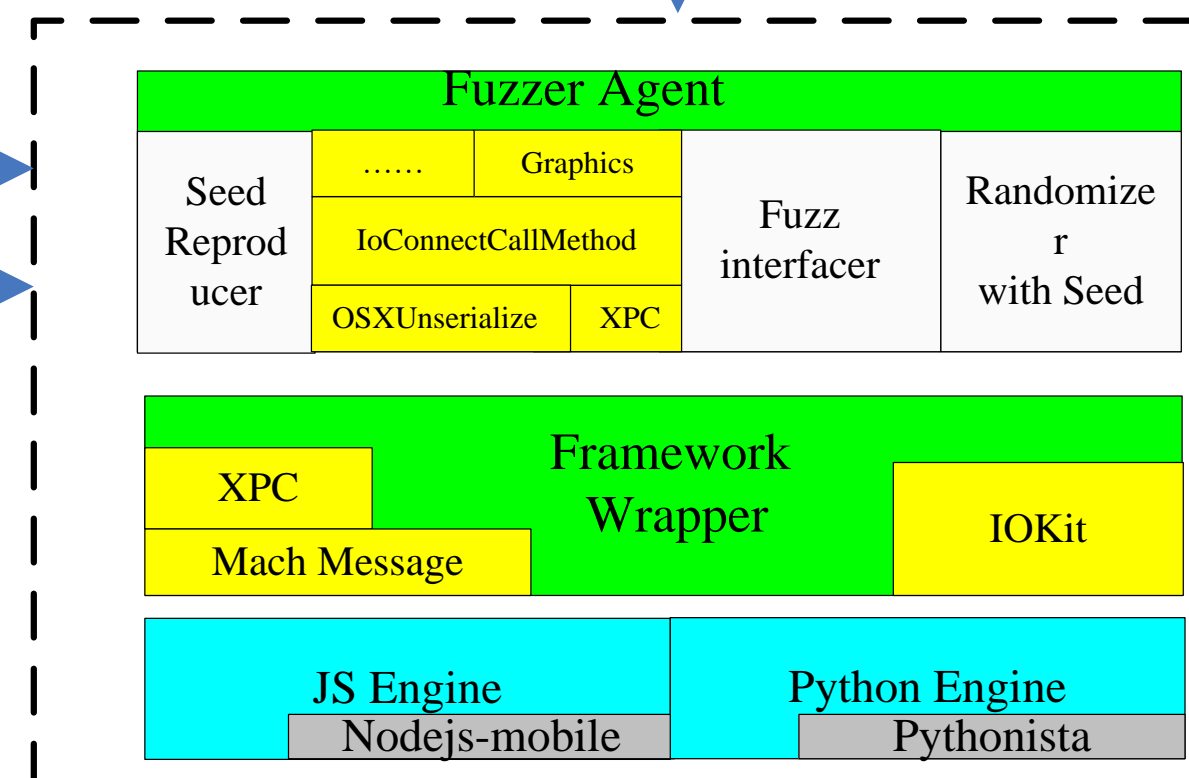
【1】 Update scripts



【4】 Reproduce by seed

Fuzzer as app

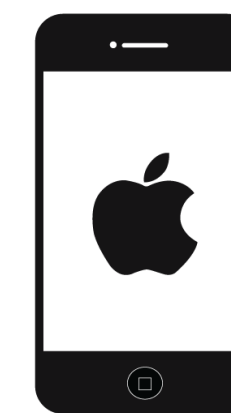
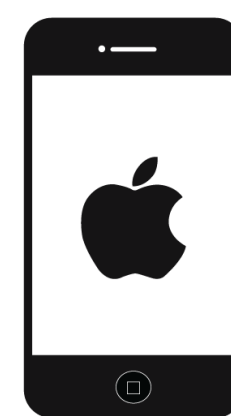
【2】 Run script driven fuzzing with seed



Crash Monitor
In JB mode

【3】 Crash monitor with modes

Target Service



Crash monitor host
Agent

Crash Monitor
Of iTunes log
in Non-JB mode

iTunes protocol via USB (libimobiledevice)

Solution Implementation

- Fuzzer Agent
 - Trigger fuzzing interface and reproduce crash case which is deployed as application normally.
 - Key and important frameworks (e.g. XPC, IOKit) or system library is wrapped in script as the middle level as fuzzing interface.
 - Integrate seed based randomization.

Solution Implementation

- Script Server
 - The script server is mainly providing fuzzing script updating and downloading. Researchers could config fuzz script towards fuzzing interface or even update script at run time when fuzzer agent is running.

Solution Implementation

- Crash Monitor
 - Responsible for detecting the crash event for fuzzing target. Actually, by design, other fuzz event (e.g. kslide leakage) could also be detected by monitor.

Solution Implementation

- Seed Server
 - The seed reproducer is mainly for collect fuzzing seed for reproduction before target devices crashes.
 - To reduce the chance of seed reproducing info missing because of interference of target failure (e.g. kernel panic or hang), the seed reproducer is deployed outside of the device.

Seed Server

The screenshot displays a file explorer interface with a sidebar on the left containing a file named 'jsonSock...x_v1.2.py' and a folder named 'seeds'. The main area is divided into two columns. The left column lists various Apple system components, including 'com.apple.incoming-call-filter-server', 'com.apple.iohideventsystem', 'com.apple.KernelExtensionServer', 'com.apple.keyboardServices.textReplacementServer.aps', 'com.apple.KeyboardServices.TextReplacementService', 'com.apple.libquird', 'com.apple.LocalAuthentication.RemoteUIHost', 'com.apple.locationd.desktop.agent', 'com.apple.locationd.desktop.registration', 'com.apple.locationd.desktop.simulation', 'com.apple.locationd.desktop.synchronous', 'com.apple.locationd.routine', 'com.apple.logind', 'com.apple.lsd.advertisingidentifiers', 'com.apple.lsd.installation', 'com.apple.lsd.mapdb', 'com.apple.lsd.modifydb', 'com.apple.lsd.open', 'com.apple.lsd.openurl', 'com.apple.lsd.plugin', 'com.apple.lsd.trustedsignatures', 'com.apple.lsd.xpc', 'com.apple.mDNSResponder_Helper', 'com.apple.mDNSResponder.dnsctl', 'com.apple.mDNSResponder.dnsproxy', 'com.apple.mdworker.mail', 'com.apple.mediaemoted.xpc', 'com.apple.metadata.mds', 'com.apple.metadata.mds.cachedelete', 'com.apple.metadata.mds.index', 'com.apple.metadata.mds.index.xpc', 'com.apple.metadata.mds.legacy', 'com.apple.metadata.mds.xpcs', 'com.apple.mobile.keybagd.mach', 'com.apple.mobile.keybagd.UserManager.xpc', 'com.apple.mobile.keybagd.xpc', 'com.apple.mobileassetd', 'com.apple.mobileassetd.cache-delete', 'com.apple.MobileFileIntegrity', 'com.apple.network.EAPOLController', 'com.apple.network.IPConfiguration', 'com.apple.networking.captivenetworksupport', and 'com.apple.nsurlsessiond'. The right column lists a series of fuzz test files, including '621_FUZZ_MODE', '621_0_Crashed_FUZZ_MODE', '621_4_Crashed_FUZZ_MODE', '621_5_Crashed_FUZZ_MODE', '621_6_Crashed_FUZZ_MODE', '621_8_Crashed_FUZZ_MODE', '621_10_Crashed_FUZZ_MODE', '621_11_Crashed_FUZZ_MODE', '621_13_Crashed_FUZZ_MODE', '621_14_Crashed_FUZZ_MODE', '621_21_Crashed_FUZZ_MODE', '621_24_Crashed_FUZZ_MODE', '621_25_Crashed_FUZZ_MODE', '621_26_Crashed_FUZZ_MODE', '621_27_Crashed_FUZZ_MODE', '621_28_Crashed_FUZZ_MODE', '621_42_Crashed_FUZZ_MODE', '621_44_Crashed_FUZZ_MODE', '621_45_Crashed_FUZZ_MODE', '621_47_Crashed_FUZZ_MODE', '847_FUZZ_MODE', '847_28_Crashed_FUZZ_MODE', '847_29_Crashed_FUZZ_MODE', '847_30_Crashed_FUZZ_MODE', '847_31_Crashed_FUZZ_MODE', '847_32_Crashed_FUZZ_MODE', '847_34_Crashed_FUZZ_MODE', '847_35_Crashed_FUZZ_MODE', '847_36_Crashed_FUZZ_MODE', '847_37_Crashed_FUZZ_MODE', '847_38_Crashed_FUZZ_MODE', '847_39_Crashed_FUZZ_MODE', '847_41_Crashed_FUZZ_MODE', '847_42_Crashed_FUZZ_MODE', '847_43_Crashed_FUZZ_MODE', '847_44_Crashed_FUZZ_MODE', '847_45_Crashed_FUZZ_MODE', '847_46_Crashed_FUZZ_MODE', '847_47_Crashed_FUZZ_MODE', '979_FUZZ_MODE', '979_33_Crashed_FUZZ_MODE', '979_34_Crashed_FUZZ_MODE', and '979_36_Crashed_FUZZ_MODE'. The file '621_42_Crashed_FUZZ_MODE' is selected, and its details are shown in the bottom right panel.

621_42_Crashed_FUZZ_MODE
TextEdit - 2 KB
Created 2/11/18, 6:32 PM
Modified 2/11/18, 6:32 PM
Last opened --
[Add Tags...](#)

Agenda

- Who We Are
- PESSR Solution
 - Fuzz Strategy
 - Crash Monitor Strategy
 - Reproduce Strategy
 - Future Plan
- Hunt Practice via PESSR
 - Attack Interface

Fuzz Strategy

- Scriptable Fuzzing
 - Why Fuzzing with Script?
 - Porting Script Engine onto Mobile Devices
 - Mac OS/iOS Framework Wrapper
- Randomizer with Seed
- Fuzz Interface
- Fuzzing by Emulation
- Best Practice

Scriptable Fuzzing -- Why

- Comparison

	Performance	Compile Cost	Runtime Logic Modification	Portability Cost
Native Code (e.g. C/C++, Object-C, Swift)	High	High	Hard	High
Script (e.g. JS, Python)	Relatively High	No	Easy	Low

- The script execution performance of script is near to native code.
- No compilation and deploy time.
- Update its inner logic at runtime from script server if you like.
- We choose Python and JavaScript as the dev language.

Scriptable Fuzzing -- Port Script Engine to Mobile

- Script Solution Comparison

	Script	Platform	Porting Effort	Need Privilege
Kivy-iOS	Python	iOS	LOW	NO
Kivy/Python-for-Android	Python	Android	LOW	NO
NodeJS-Mobile	JS	Both	LOW	NO

Scriptable Fuzzing -- Framework Wrapper

- Python Binding (mach_msg, XPC...)

```
def xpcMachMsg(self, msg):  
    #prepare mach_msg_args  
    msgHeaderAddress = ctypes.addressof(msg.header)  
    option = MACH_SEND_MSG  
    send_size = msg.header.msgh_size  
    rcv_size = 0  
    rcv_name = MACH_PORT_NULL  
    timeout = MACH_MSG_TIMEOUT_NONE  
    notify = MACH_PORT_NULL  
    self.connection.mach_msg(  
        msgHeaderAddress,  
        option,  
        send_size,  
        rcv_size,  
        rcv_name,  
        timeout,  
        notify  
    )  
    logging.info("xpcMachMsg:\t headerAddress=0x{0:x},send_size=0x{1:x}".format(msgHeaderAddress, send_size))  
    logging.debug("xpcMachMsg:\t msgAddr=0x{0:x},headerAddr=0x{1:x}, headerSize=0x{3:x},bodyAddr=0x{2:x}, bodySize=0x{4:x},  
msgSize=0x{5:x}".format(  
        ctypes.addressof(msg),  
        ctypes.addressof(msg.header),  
        ctypes.addressof(msg.body),  
        ctypes.sizeof(msg.header),  
        ctypes.sizeof(msg.body),  
        ctypes.sizeof(msg)  
    ))
```

```
26  
27 BOOST_PYTHON_MODULE(xpcconnection) {  
28     PyEval_InitThreads();  
29  
30     class_<XpcConnection, boost::noncopyable>("XpcConnection", init<std::string>())  
31     //class_<XpcConnection>("XpcConnection")  
32     //xpc  
33     .def("XpcSendMessage", &XpcConnection::XPCSendMessage)  
34     .def("XpcCreateConnection", &XpcConnection::XpcCreateConnection)  
35     .def("XpcHandler", pure_virtual(&XpcConnection::handler))  
36  
37     //mach message  
38     .def("mach_msg", &XpcConnection::mach_msg_)  
39     .def("mach_connect", &XpcConnection::mach_connect_)  
40  
41     ;  
42  
43 }
```

Fuzz Interface

	Target Mode	Scope	Reason
System App	High Privilege in User Mode	Safari, Apple Store, iTunes	RCE
XPC Services	Root in User Mode	Launchd, *d...	1. Root or High Privilege 2. Called from Low Privilege App
IOKit Driver	Kernel	Graphic, HID...	Touched Inside Sandbox
XNU	Kernel	OSUnserializeBinary...	1. Ever Attacked by PEGASUS APT 2. Cross Platform
iBoot	BootLoader	UEFI Service	Source Leaked
Open Source Module	Any	FFMPEG...	1. Used by High Privilege Service 2. AFL Suitable 3. Cross Platform
New Added Feature	Any	New Hardware (e.g. Touchpad for MacPro), New System Call...	Bad Test Usually

Fuzzing by Emulation -- Why

- Increase Code Coverage
 - a Large Number of Cases
- iOS Device Sandbox System?
 - a Lot of Device, a Big Cost
- Arm Servers?
 - Very High Price
- Raspberry Pi
 - a Low-Cost Arm Device, **Cluster**



Fuzzing by Emulation -- Source of Inspiration

- Darling
 - Open Source on GitHub
 - Run MacOS Exe on Linux
- But
 - Only Support x86 Mach-O

Darwin/macOS emulation layer for Linux <http://www.darlinghq.org>

2,371 commits

6 branches

0 releases

26 contributors

GPL-3.0

Branch: master

New pull request

Find file

Clone or download

LubosD

dSYM generation improvements, thread name support for LLDB

Latest commit e8b0fd9 3 days ago

Developer

Liblaunch symlink

5 days ago

basic-headers

Adding sys/qos.h into basic-headers

2 years ago

cmake

dSYM generation improvements, thread name support for LLDB

3 days ago

debian

Initial fix to get dkms working on kernel upgrades

6 months ago

etc

Create a udev rule to set /dev/mach to 0666

2 years ago

kernel-include

Fix mach_error_string()

2 months ago

misc

Cleanup benchmarks, misc and tools directories

3 years ago

platform-include

Build dSYM files for all binaries

4 days ago

rpm

Added missing rtsig.h file

14 days ago

src

dSYM generation improvements, thread name support for LLDB

3 days ago

tests/src

Remove unused stuff from tests/

3 years ago


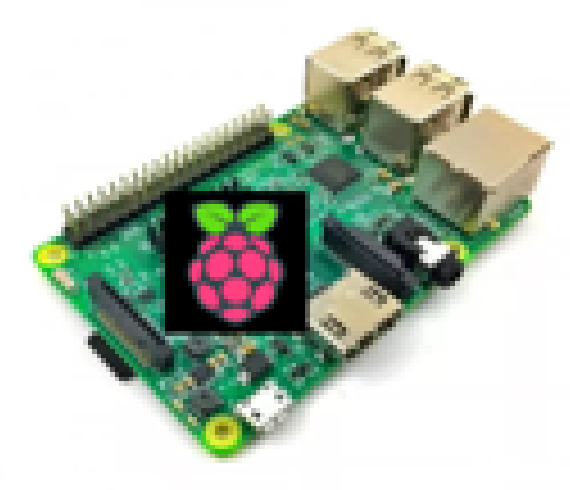
tools

Create tool to rename lines recursively

2 months ago

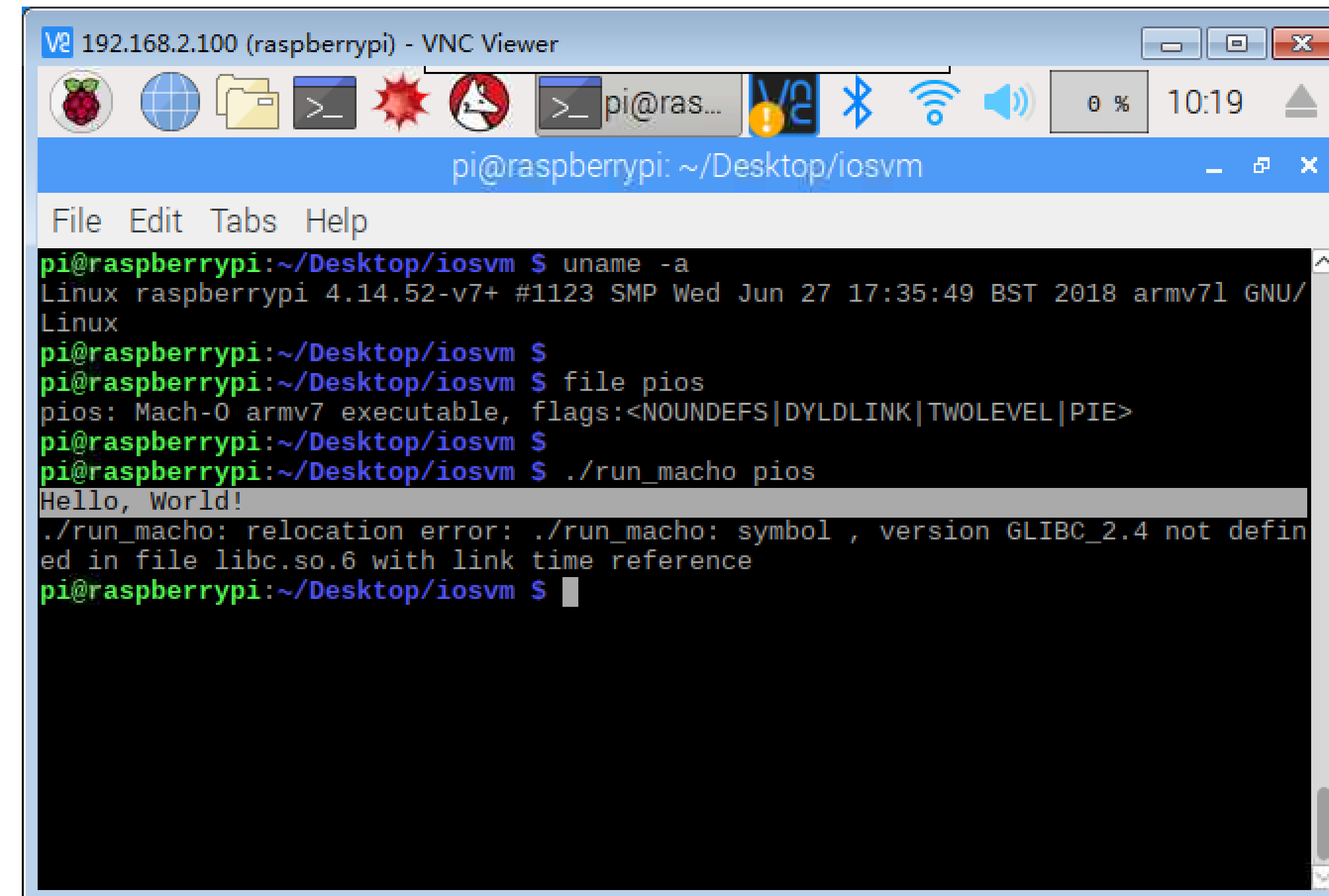
Fuzzing by Emulation -- Transplant and Rebuild

- Darling-Pi
 - Support ARM Mach-O
 - Run iOS Exe on Raspberry Pi
- But Rebuild The Code

Mach-O (x86) ↗	↗	Mach-O (ARM) ↗
Darling (x86) ↗	--> ↗	Darling-Pi (ARM) ↗
Linux (x86) ↗	--> ↗	 Raspbian Stretch (ARM) ↗
PC or Server (x86) ↗	--> ↗	 Raspberry Pi (ARM) ↗

Fuzzing by Emulation -- Architecture

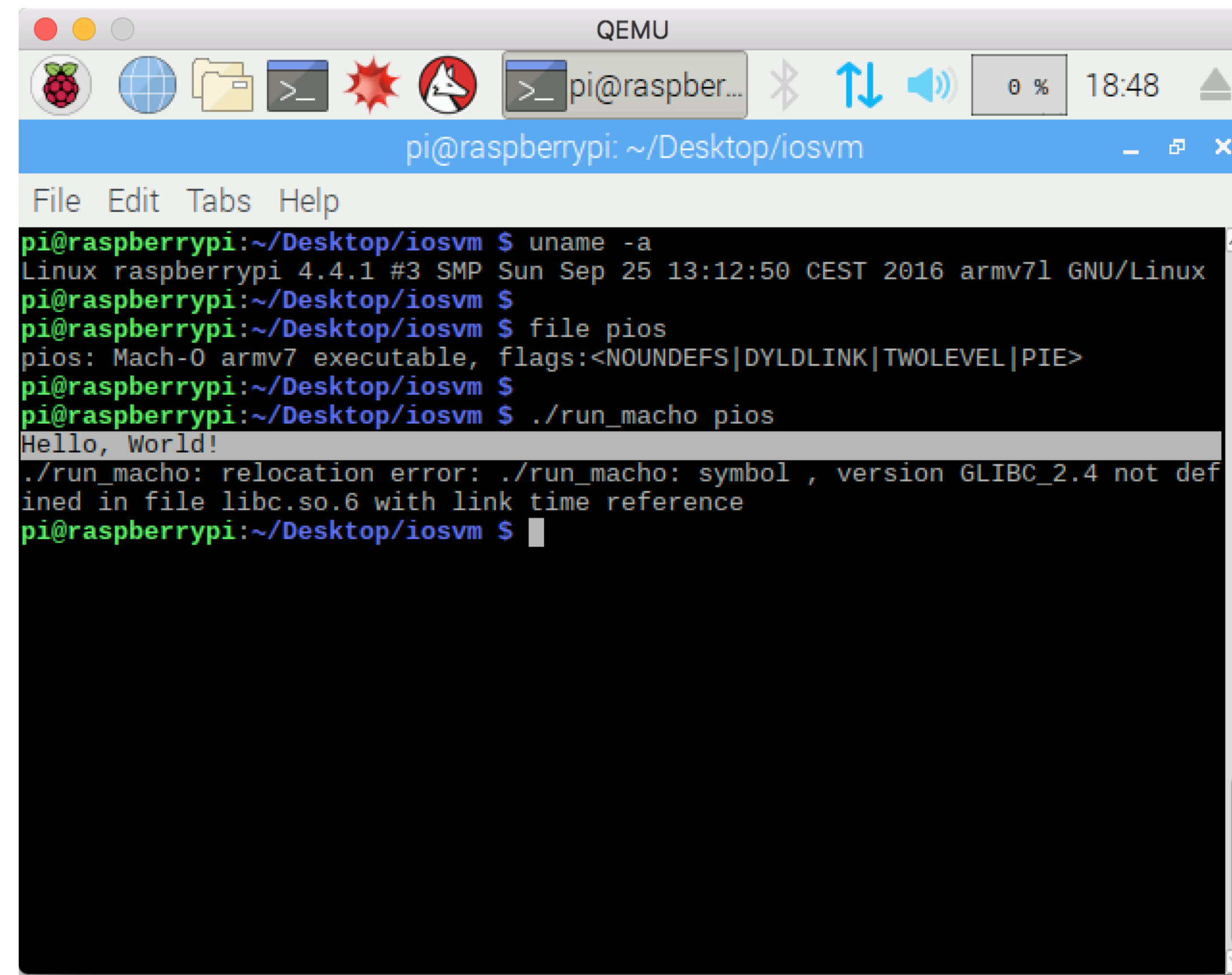
dyld Mach-O							
libc++.so		libCocoa.so		libIOKit.so		...	
IOKit	IPC	...		File System	Networking	...	
Mach Driver							
Linux Kernel							
Hardware							



```
192.168.2.100 (raspberrypi) - VNC Viewer
pi@raspberrypi: ~/Desktop/iosvm
File Edit Tabs Help
pi@raspberrypi:~/Desktop/iosvm $ uname -a
Linux raspberrypi 4.14.52-v7+ #1123 SMP Wed Jun 27 17:35:49 BST 2018 armv7l GNU/Linux
pi@raspberrypi:~/Desktop/iosvm $ file pios
pios: Mach-0 armv7 executable, flags:<NOUNDEFS|DYLDLINK|TWOLEVEL|PIE>
pi@raspberrypi:~/Desktop/iosvm $ ./run_macho pios
Hello, World!
./run_macho: relocation error: ./run_macho: symbol , version GLIBC_2.4 not defined in file libc.so.6 with link time reference
pi@raspberrypi:~/Desktop/iosvm $
```

Fuzzing by Emulation -- Full System Emulation

- Virtualize Raspberry Pi
 - Qemu



```
QEMU
pi@raspberrypi: ~/Desktop/iosvm
File Edit Tabs Help
pi@raspberrypi:~/Desktop/iosvm $ uname -a
Linux raspberrypi 4.4.1 #3 SMP Sun Sep 25 13:12:50 CEST 2016 armv7l GNU/Linux
pi@raspberrypi:~/Desktop/iosvm $
pi@raspberrypi:~/Desktop/iosvm $ file pios
pios: Mach-O armv7 executable, flags:<NOUNDEFS|DYLDLINK|TWOLEVEL|PIE>
pi@raspberrypi:~/Desktop/iosvm $
pi@raspberrypi:~/Desktop/iosvm $ ./run_macho pios
Hello, World!
./run_macho: relocation error: ./run_macho: symbol , version GLIBC_2.4 not defined in file libc.so.6 with link time reference
pi@raspberrypi:~/Desktop/iosvm $
```


Fuzzing by Emulation -- Work Flow

- Fuzz
 - Mach-0
 - Fragment Code
- All Attack Surfaces

Best Practice

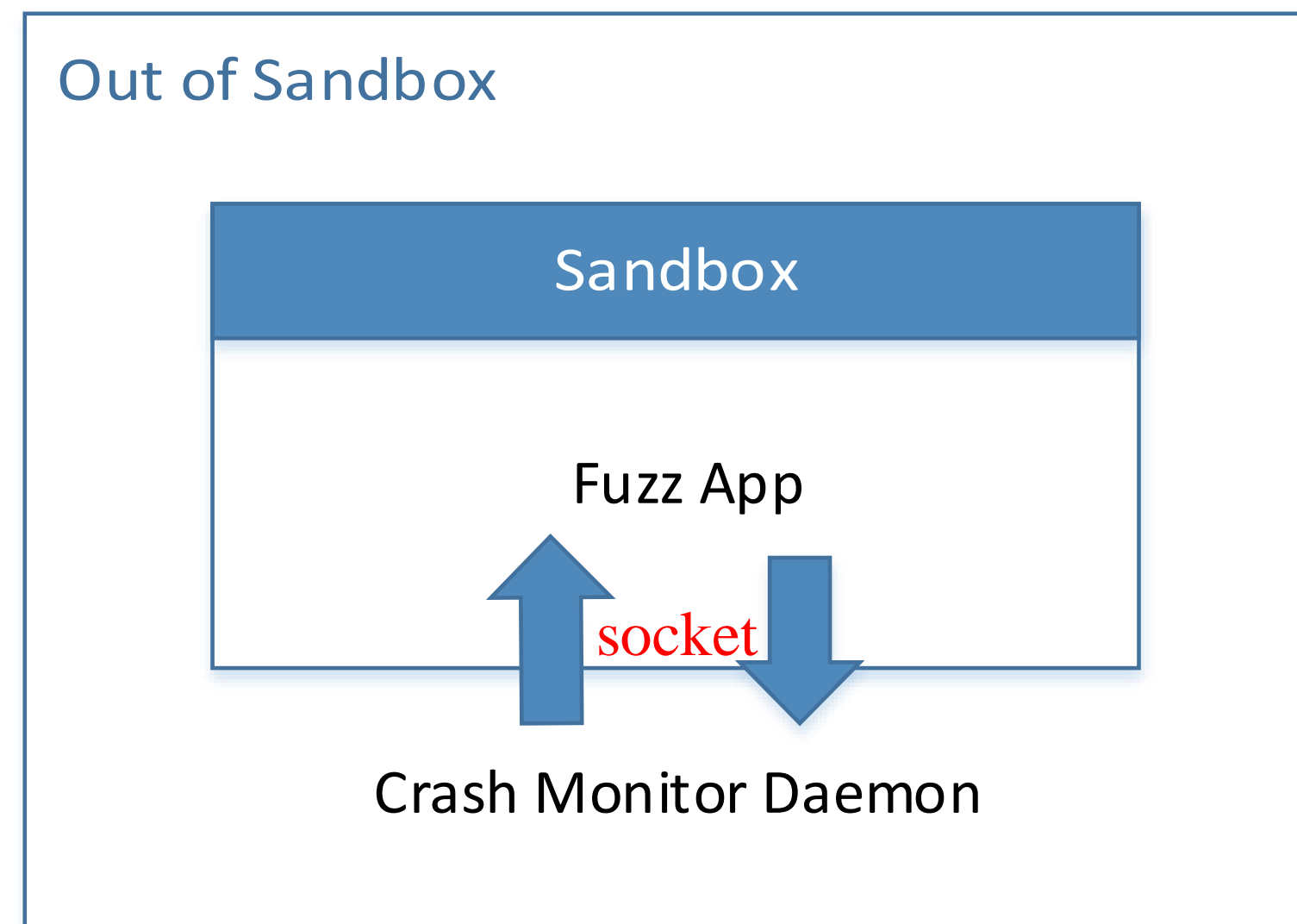
- Treat Fuzz as Development Work
 - It would reduce your much effort of fuzzing is the project is well designed implemented.
- Collect Attack Scenario POC as Repository
 - The more scenario POC towards attack interface you gain, the more possibility you would trigger vulnerability.
 - Implement your attack scenario in script.
- Automatic Whitelisting Ever Crashed or Hanged Cases
 - High possibility the ever crashed issue would happen again and again to interrupt your fuzzing.

Agenda

- Who We Are
- PESSR Solution
 - Fuzz Strategy
 - **Crash Monitor Strategy**
 - Reproduce Strategy
 - Future Plan
- Hunt Practice via PESSR
 - Attack Interface

Crash Monitor Strategy -- Jail Broken

- More privilege can get on jailed system.
- Socket mechanism is used in the communication between fuzzer and daemon server for its high efficient and performance.



Crash Monitor Strategy -- Non Jail Broken

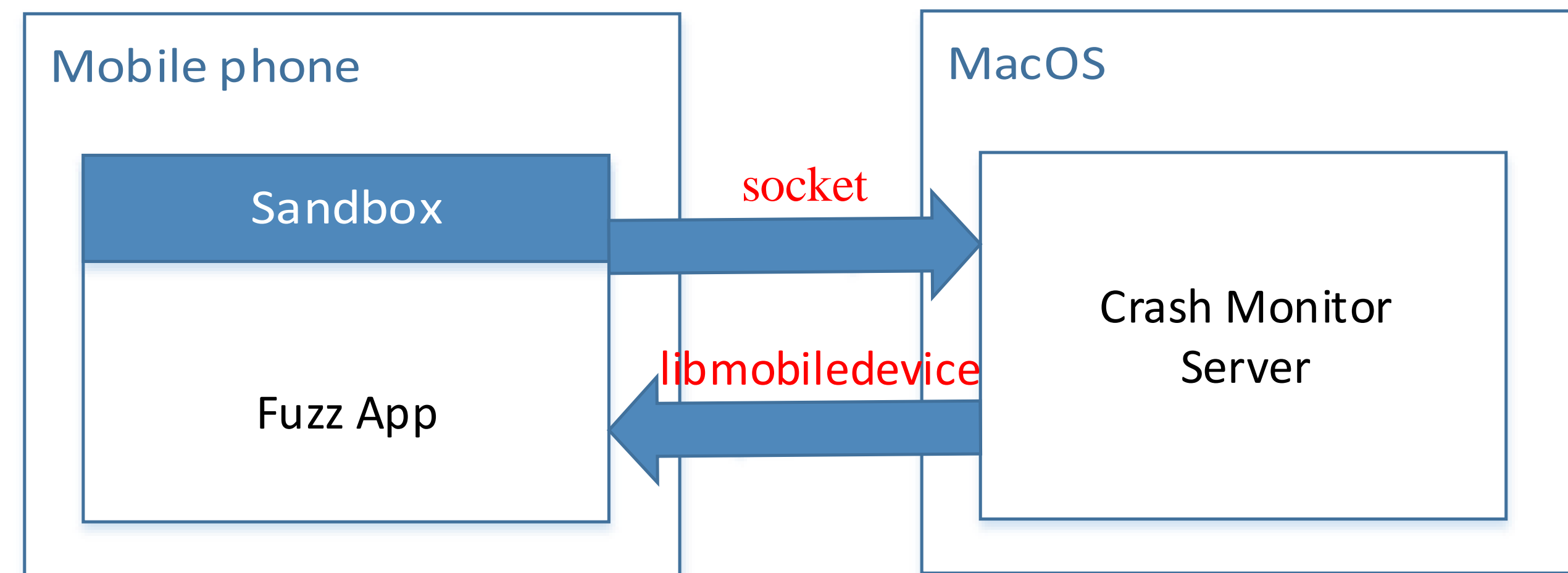
- Libmobiledevice toolset is an open source project which implement the iTunes protocol on different platform.

- 1) `idevicesyslog`:

- Real-time Logs on iDevice

- 2) `idevicecrashreport`:

- `/var/mobile/Library/Logs/CrashReporter`
 - Contains all diagnostics information including panic, crash and hang.



Best Practice

- Checking Process Cluster if Possible
 - For some scenarios, the target process or service is not one but cluster (for example mdworker).
- Add System or Service Hang as Monitor Signal Also

Agenda

- Who We Are
- PESSR Solution
 - Fuzz Strategy
 - Crash Monitor Strategy
 - **Reproduce Strategy**
 - Future Plan
- Hunt Practice via PESSR
 - Attack Interface

What Interference Reproduction?

- Race Condition
 - Mutex competition for sharing resources (e.g. socks, service port), multiple threads condition, callback method and so forth would cause instruction execution disorder.
- API Calls Correlation
 - A series APIs must be called in specific order and the input parameter must be provided correctly by previous API.
- Environment Dependency
 - The run time environment would be volatile and easy to be changed in fuzzing life cycle.

Reproduce Method

- Comparison

	Typical Example	Storage Cost	Speed Cost	Support Complex Scenario	Reproduce Rate	Dev Effort
Log	Trinity	High (Execution Log)	High	Low	Low	Low
Case (File)	AFL	Middle (Files Causing Crash)	Low	Middle	Middle	High
Crash Dump	-	High (Every Crash Context)	High	-	Very Low	No
Seed	JS Fun Fuzz	Low (Integer)	Low	High	High	Low

- (e. g. Pseudo-Random Number Generator with Mersenne Twister Algorithm) with a seed which is usually a long integer (64bit).
- All data could be reproduced exactly in your same code no matter how many times you call the code.

Best Practice -- Client-Server Mode via Socket

```
1
2 def on_new_client(self, server, client, client_addr):
3     if SingletonGlobalConfig().gMode == Mode.FUZZ:
4         excCmd("clear")
5         logging.error("on_new_client:\tEnter{0}:{1} ".format(client, client_addr))
6         clientLastPID = -1
7         clientLastRunCounter = -1
8         recvCount = 0
9         while True:
10            data = {}
11            try:
12                data = server.recv(client)
13            except Exception as e:
14                traceback.print_exc()
15                logging.error("on_new_client:\trecv client force to disconnect.....{0}:{1} ".format(client, client_addr ))
16                #save latest seed and history
17                self.saveSeedsFileByKey(clientLastPID, clientLastRunCounter)
18                break
19            clientPID = data["clientPID"]
20            runCounter = data["runCounter"]
21            seed = data["seed"]
22            mode = data["mode"]
23            isTargetCrashed = data["isTargetCrashed"]
24            targetName = data["targetName"]
25            clientLastRunCounter = runCounter
26            clientLastPID = clientPID
27            jsonReply = {
28                "clientPID":clientPID,
29                "runCounter":runCounter,
30                "seed":seed,
31                "mode":mode
32            }
33            logging.info(data)
34            newPlistItem = data
35            dirKey = (clientPID, runCounter)
36            #for new dir Key
37            if not self.seedsDir.has_key(dirKey):
38                logging.error(data)
39                #Just keep the last two
40                self.seedsDir[dirKey] = []
41                self.seedsDir[dirKey].append(newPlistItem)
42                if self.seedsDir.has_key((clientPID, runCounter-1)):
43                    pass
44                if self.seedsDir.has_key((clientPID, runCounter-2)):
45                    del self.seedsDir[(clientPID, runCounter-2)]
46                #Save the last two seed
47                self.saveSeedsFileByKey(clientPID, runCounter )
48            #for unknown dir Key
49            else:
50                self.seedsDir[dirKey].append(newPlistItem)
51            #print "xxxxxxxxxxxxself.seedsDir len = {0} xxxxxx".format(len(self.seedsDir))
52            if isTargetCrashed:
53                logging.error(data)
54                self.saveSeedsFileByKey(clientPID, runCounter, str(runCounter)+"_Crashed")
55            #log it
56            #logging.info(jsonReply)
57            try:
58                server.send(client, jsonReply)
59            except Exception as e:
60                traceback.print_exc()
61                logging.error("on_new_client:\t send client force to disconnect.....{0}:{1} ".format(client, client_addr, targetName))
62                #save latest seed and history
63                self.saveSeedsFileByKey(clientPID, runCounter)
64                break
65            recvCount = recvCount +1
66        #end of while
```

Best Practice

- Seed is The Only Data as Traffic
 - The seed is small (usually 64bits) enough, network transformation speed is relative not low compare to disk storage serialization.
- Seed Server Design Out Side of Running Target
 - (Flexible) Redirect the traffic to any other server.
 - (Steady) Target process or even kernel crash has no any interference to seed server.

Best Practice

- Multiple Fuzz Agent with Multiple Server Ports
 - You can establish mapping of one server port (max number is 65535) to one fuzz agent.
 - And every fuzzing session owns its separated folders on server side.

Best Practice

- Keep Last N Seeds Info with Last N-M Logs
 - Last N Seeds Info for Reproduction
 - N Should Not be Small
 - N Should Not be Big
 - Previous M Seeds for Redundancy
 - Last N-M Detail Logs
 - Usually detail input parameter content and return value and so forth.
 - Help you verify the crash as soon as possible.

102
103

Agenda

- Who We Are
- PESSR Solution
 - Fuzz Strategy
 - Crash Monitor Strategy
 - Reproduce Strategy
 - **Future Plan**
- Hunt Practice via PESSR
 - Attack Interface

Future Plan -- Code Coverage Feedback Support

- Integrate code coverage feedback support.
 - User mode fuzzing like AFL, Honggfuzz, Kernel mode fuzzing like syzkaller.
- Dynamic binary instrumentation or static patch for close-source library.
- Keep seed generated data method in the corpus mutation.
- “persistent” mode as one of in-process fuzzing mode in AFL like fuzzing.

Future Plan -- Attach and Hook and Fuzz

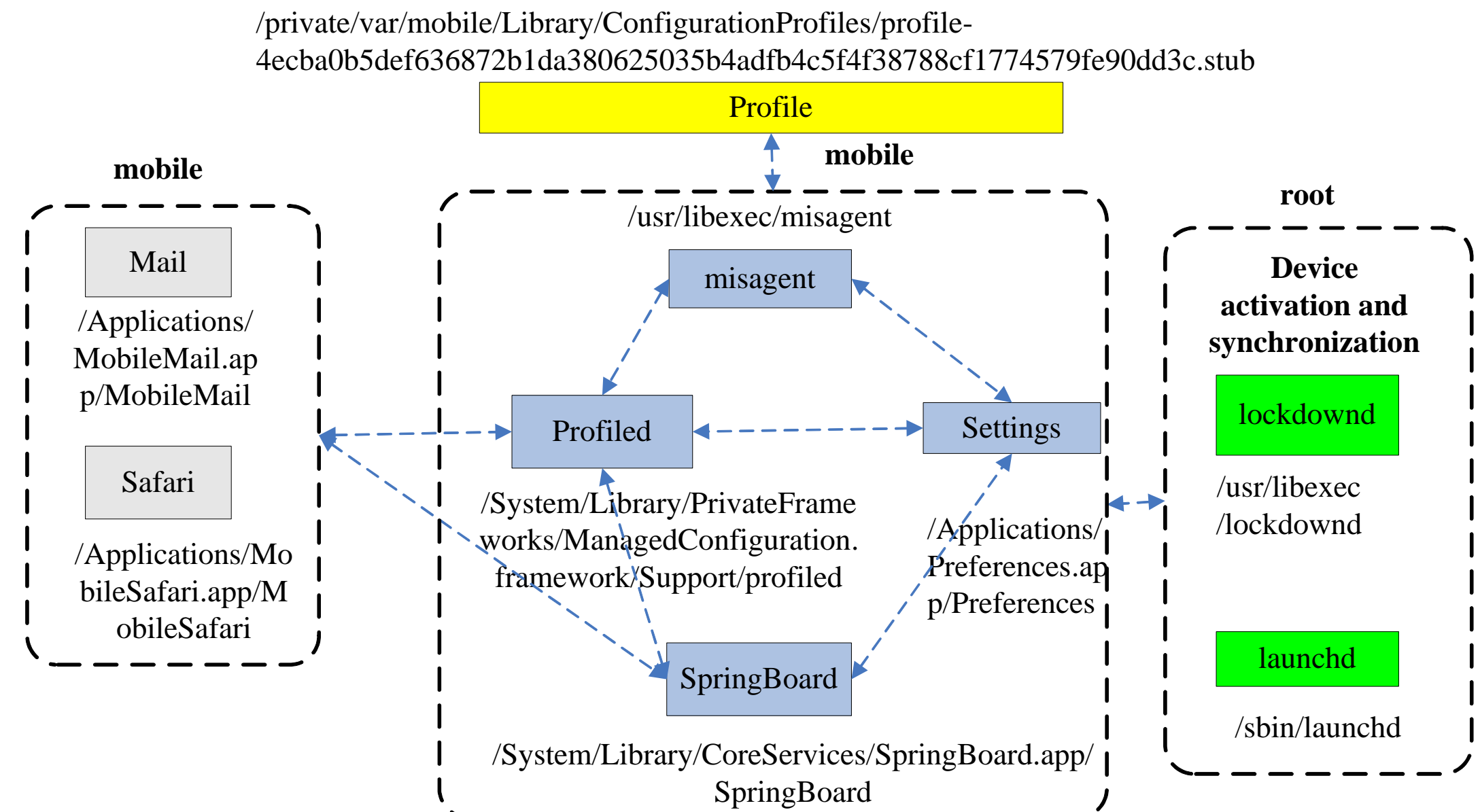
- Some fuzzing interfaces could not be permitted to be touched because of privilege restriction or security policy check.
- One possible fuzzing method is to inject to the target process with enough privilege (e.g. launchd) and launch fuzz at specific time and specific code location.
- Already supported in Honggfuzz (in Linux platform).

Future Plan -- Symbolic Execution Guided Fuzz

- Help to guide the fuzzer to find proper input fuzz data instead of blind way so as to improved code coverage.
- Symbolic execution (e.g. Angr) is good at deciding the input scope and content towards uncovered code location especially in relatively small target code size (e.g. inner one function).
- Such Driller (Angr+AFL)

Future Plan -- Call Back Fuzz

- An iOS XPC Service always needs to interact with other XPC Services.
- Using Darling-Pi, we can simulate other XPC Services to implement and communicate with our target XPC Service and fuzz.



Agenda

- Who We Are
- PESSR Solution
 - Fuzz Strategy
 - Crash Monitor Strategy
 - Reproduce Strategy
 - Future Plan
- Hunt Practice via PESSR
 - Attack Interface

Attack Interface

- Enumerate Fuzz Interface towards XPC Daemon Services
- Enumerate Reachable Drivers within Sandboxed App

XPC Daemon Services

- `launchctl DumpState`

Process Name	Service Name
AirPlayUIAgent	com.apple.AirPlayAgent.xpc
	com.apple.AirPlayUIAgent.xpc
AirPlayXPCHelper	com.apple.AirPlayXPCHelper
	com.apple.coremedia.endpoint.xpc
	com.apple.coremedia.endpointpicker.xpc
	com.apple.coremedia.endpointplaybacksession.xpc
	com.apple.coremedia.volumecontroller.xpc
	com.apple.coremedia.routingcontext.xpc
	com.apple.coremedia.endpointstream.xpc
	com.apple.coremedia.endpointuiagent.xpc
	com.apple.coremedia.endpointremotecontrolsession.xpc
	com.apple.coremedia.endpointmanager.xpc
	com.apple.coremedia.endpointgroup.xpc
	com.apple.coremedia.routediscoverer.xpc

XPC Daemon Services

```
services = {
    0      -      com.apple.rpmuxd
    (dp)   0      com.apple.wifiFirmwareLoader
    48     -      com.apple.uninstalld
    49     -      com.apple.kextd
    0      -      com.apple.diagnostics.extensions.osx.spotlight.helper
    0      78     com.apple.duetknowledge
    0      -      com.apple.tzlinkd
    0      -      com.apple.diagnostics.extensions.osx.timemachine.helper
    0      -      com.apple.kcproxy
    50     -      com.apple.fsevents
    0      -      com.apple.storedownloadd.daemon
    264    -      com.apple.CoreAuthentication.daemon
    0      0      com.apple.Kerberos.digest-service
    (dp)   0      com.apple.CoreRAID
    0      -      com.apple.systempreferences.cacheAssistant
    0      0      com.apple.automountd
    0      -      com.apple.TrustEvaluationAgent.system
    119    -      com.apple.coreservicesd
    0      -      com.apple.newsyslog
    52     -      com.apple.mediaremoted
    (dp)   0      com.apple.applefileutil
    371    -      com.apple.adid
    281    0      com.apple.AmbientDisplayAgent
    0      -      com.vix.cron
    0      -      com.apple.storeagent.daemon
    (dp)   0      com.apple.MRTd
    220    -      com.apple.touchbarserver
    0      -      com.apple.mbusertrampoline
    213    -      com.apple.thermald
    333    -      com.apple.FileCoordination
    377    -      com.apple.taskgated
    284    -      com.apple.GSSCred
    272    -      com.apple.audio.systemsoundserverd
    0      -      com.apple.RemoteDesktop.PrivilegeProxy
    222    -      com.apple.colorsync.displayservices
    0      -      com.apple.avbdeviced
    0      -      org.macosforge.xquartz.privileged_startx
    55     -      com.apple.coreservices.appleevents
}
```


XPC Daemon Services

```
domain = com.apple.xpc.launchd.user.domain.501.100008.Aqua
asid = 100008
minimum runtime = 10
exit timeout = 5
runs = 1
successive crashes = 0
pid = 655
immediate reason = ipc (mach)
forks = 0
execs = 1
initialized = 1
trampolined = 1
started suspended = 0
proxy started suspended = 0
last exit code = (never exited)

event triggers = {
}

endpoints = {
    "com.apple.FontObjectsServer" = {
        port = 0x73d03
        active = 1
        managed = 1
        reset = 0
        hide = 0
    }
}

dynamic endpoints = {
}

pid-local endpoints = {
}

instance-specific endpoints = {
}

event channels = {
```

e.g. com.apple.FontObjectsServer

```
{
  u'body': [
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -236443597, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, -2584486, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1024772044, 0, 0, 0, 196103729, 0, -1410287190, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 1440320692, 0, 0, -236640535, 0, 1636347732, 0, 0, -1733774177, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    1716914514, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1605381140, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1328446666, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 1455149345, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -489620296, 0, -920350586, 0, 0, 0, 0, 0, 0, 0,
    -884749930, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 167377220, 53552196, 1183905232, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1789160880, 0, 0, 0, 0, 0, 0, 0, 0, -2051169477, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -794602644, 0, 0, 0, 0, 0, 0, 0, 0, -1392271967, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0
  ],
  u'clientPID': 4015,
  u'targetName': u'com.apple.FontObjectsServer',
  u'runCounter': 0,
  u'uMachMsgTry': 0,
  u'seed': 1161363155,
  u'mode': u'Mode.CRASH_BY_CASE',
  u'serviceName': u'com.apple.FontObjectsServer',
  u'clientIP': u'192.168.43.52',
  u'msg_id': 16
}
```

Reachable Drivers within Sandboxed App in iOS

Service Name	iPhone X	iPhone 7	iPad Air
AGXAccelerator	√	√	√
AGXAcceleratorG10	√		
AGXAcceleratorG10P_B0	√		
AGXFamilyAccelerator	√	√	√
AppleCLCD	√	√	√
AppleHIDTransportHIDDevice	√	√	
AppleJPEGDriver	√	√	√
AppleKeyStore	√	√	√
AppleM2ScalerCSCDriver	√	√	√
AppleSPUHIDDevice	√	√	
IOAcceleratorES	√	√	√
IOGraphicsAccelerator2	√	√	√
IOHIDDevice	√	√	
IOMobileFramebuffer	√	√	√
IOSurfaceRoot	√	√	√
UnifiedPipeline	√		

Q&A