

macOS Packet Capture tool (Senior Project)



Jared VanEnkevort

Introduction

- Native Application for macOS & Apple Silicon
- Enables packet capture in a relatively user friendly GUI
- Something that's related to the internet, but not web dev

Tech stack



Swift UI



Objective-C



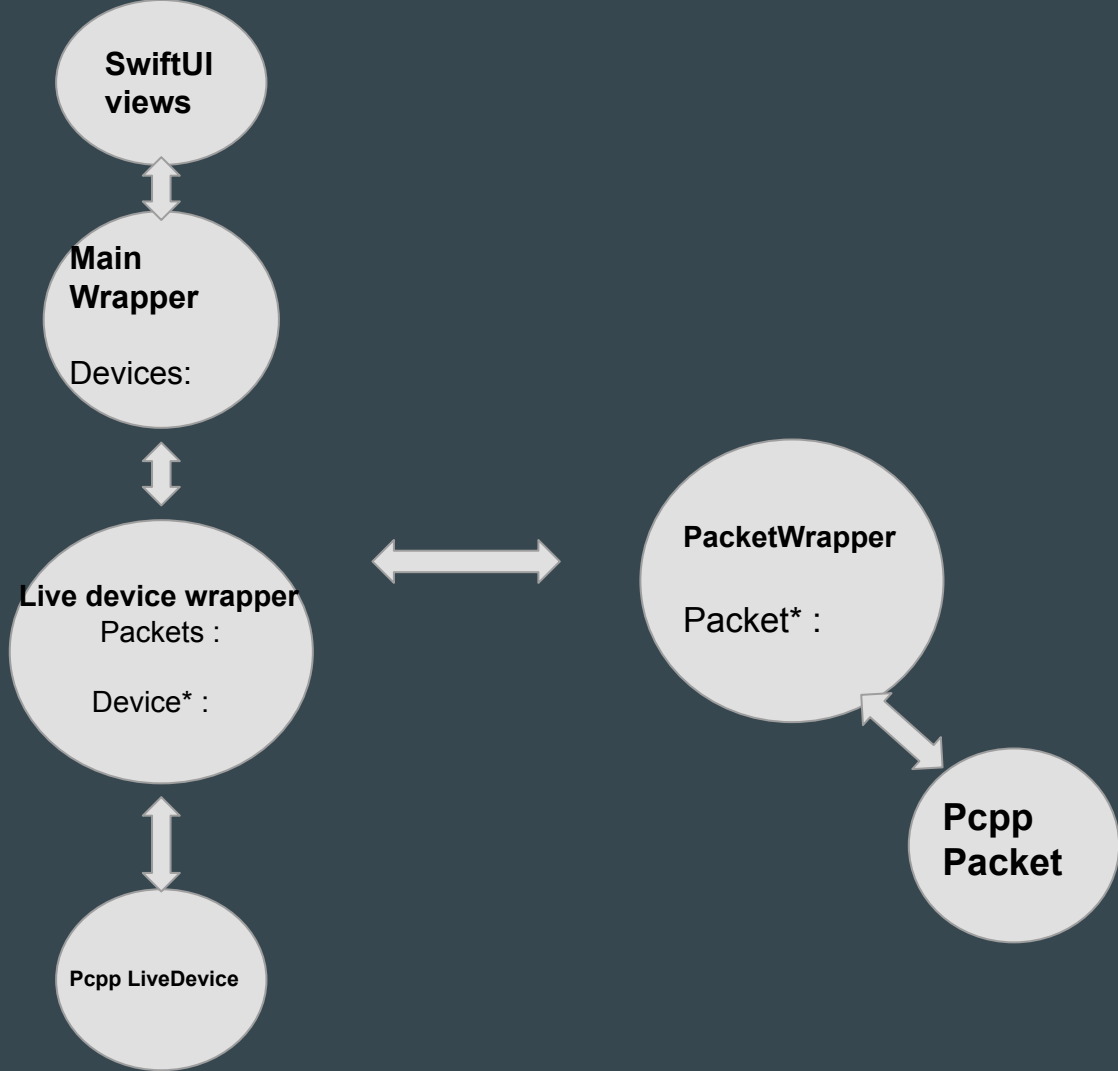
pcapPlusPlus



Concerns

- Is it even possible??
- Can I set the adapter to promiscuous mode?
- Utilizing three languages
- Objective C++??

Object Design



Difficulties/Challenges

- Getting it to link and compile
- Accessing devices
- Objective C++ restrictions
- Keeping headers clean
- Swift value semantics vs Objective C reference semantics

More Problems



Thread 1: EXC_BAD_ACCESS (code=2, address=0x7fff514147b0)

Weird Stuff

```
//hand nsobj over to corefoundation framework from obj c land
- (void) asyncCaptureStart {
    pcap::PcapLiveDevice *tempDev = (pcap::PcapLiveDevice*) dev;
    tempDev->startCapture(onPacketArrives, (void *)CFBridgingRetain(self));
    return;
}
```

```
//Add to packet array when packet arrives
//Note : Temp raw should be free'd by Packet class destructor automatically |
static void onPacketArrives (pcap::RawPacket *rawPacket, pcap::PcapLiveDevice *dev, void *cookie) {
    PcapCppDevWrapper *aDev = (__bridge PcapCppDevWrapper*)cookie;
    pcap::RawPacket* tempRawCopy = new pcap::RawPacket;
    tempRawCopy->setRawData(rawPacket->getRawData(), rawPacket->getRawDataLen(),
        rawPacket->getPacketTimeStamp(), rawPacket->getLinkLayerType(), rawPacket->getFrameLength());
    pcap::Packet *nonRawPacket = new pcap::Packet(tempRawCopy);
    PcapCppPacketWrappper *newPacketWrapper = [[PcapCppPacketWrappper alloc] initWithPacket:nonRawPacket];
    [aDev addToPacketArray:newPacketWrapper];
}

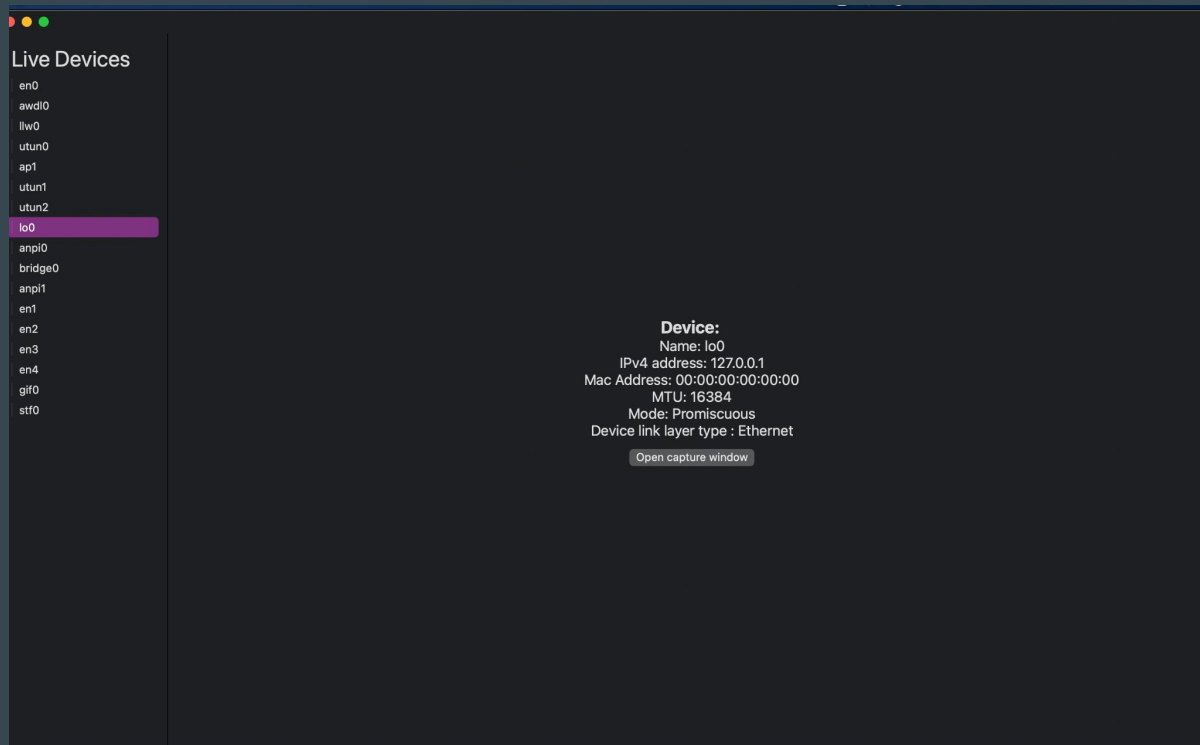
@end
```


Data types

```
- (NSMutableArray<PcapCppDevWrapper*> *) getDevices {
    PcapMain pcapMainClass;
    std::vector<pcpp::PcapLiveDevice*> devices = pcapMainClass.getDevices();
    NSMutableArray<PcapCppDevWrapper*> *devicesArray = [NSMutableArray arrayWithCapacity:devices.size()];
    for (pcpp::PcapLiveDevice* dev : devices) {
        PcapCppDevWrapper *newDevWrapper = [[PcapCppDevWrapper alloc] initWithDev:dev];
        [devicesArray addObject:newDevWrapper];
    }
    return devicesArray;
}
```

```
1
2 //TODO: return string from timespec struct
3 - (NSString *) getTimeStamp {
4     pcpp::Packet *tempPacket = (pcpp::Packet*) packet;
5     pcpp::RawPacket *rawPacket = tempPacket->getRawPacket();
6     timespec timeStampStruct = rawPacket->getPacketTimeStamp();
7     std::string timeStr = std::to_string(timeStampStruct.tv_nsec);
8     NSString *finalTimeStr = [NSString stringWithCString: timeStr.c_str() encoding:[NSString defaultCStringEncoding]];
9     return finalTimeStr;
10 }
11
```

Screenshots



Start Capture

stop capture

Save as file

View Packets

Discard Packets

exit

No capture active on device lo0

No of Packets captured : 0

es

Total Packets captured : 260 Select by protocol: SSH Select by size: 0 exit

Packet No.	Arrival Time	Total packet len	Protocol
Packet No: 103	Arrival Time : 80959000	Total packet len: 54	Protocol : TCP
Packet No: 104	Arrival Time : 110800000	Total packet len: 78	Protocol : TCP
Packet No: 105	Arrival Time : 114275000	Total packet len: 74	Protocol : TCP
Packet No: 106	Arrival Time : 114387000	Total packet len: 66	Protocol : TCP
Packet No: 107	Arrival Time : 114966000	Total packet len: 87	Protocol : SSH
Packet No: 108	Arrival Time : 118973000	Total packet len: 66	Protocol : TCP
Packet No: 109	Arrival Time : 123246000	Total packet len: 107	Protocol : SSH
Packet No: 110	Arrival Time : 123336000	Total packet len: 66	Protocol : TCP
Packet No: 111	Arrival Time : 125480000	Total packet len: 1514	Protocol : SSH
Packet No: 112	Arrival Time : 125526000	Total packet len: 130	Protocol : SSH
Packet No: 113	Arrival Time : 127477000	Total packet len: 1122	Protocol : SSH
Packet No: 114	Arrival Time : 127554000	Total packet len: 66	Protocol : TCP
Packet No: 115	Arrival Time : 129000000	Total packet len: 66	Protocol : TCP
Packet No: 116	Arrival Time : 130176000	Total packet len: 98	Protocol : SSL
Packet No: 117	Arrival Time : 130291000	Total packet len: 54	Protocol : TCP
Packet No: 118	Arrival Time : 130926000	Total packet len: 114	Protocol : SSH

Info for packet no: 1

Frame length: 136
Link type : Ethernet
layer 1 : Packet length: 136 [Bytes], Arrival time: 2021-12-10 11:40:35.778225000

layer 2 : Ethernet II Layer, Src: 3c:57:31:bc:5b:c2, Dst: a0:78:17:b4:c0:b0

layer 3 : IPv4 Layer, Last fragment [offset= 35288], Src: 64.9.8.28, Dst: 0.0.0.0

layer 4 : TCP Layer, Src port: 0, Dst port: 8195

layer 5 : Unknown Layer, Application Data

Packet length: 136 [Bytes], Arrival time:
2021-12-10 11:40:35.778225000
Ethernet II Layer, Src: 3c:57:31:bc:5b:c2,
Dst: a0:78:17:b4:c0:b0
IPv4 Layer, Last fragment [offset= 35288],
Src: 64.9.8.28, Dst: 0.0.0.0
TCP Layer, Src port: 0, Dst port: 8195
Unknown Layer, Application Data

Conclusion

- Probably would've spent a more considerable amount of time formally learning Swift & SwiftUI
- Done more research into the technologies I chose to utilize
- Overall satisfied