**Deliverable #5: Reverse Engineering: Software Analysis**

**Project: IoT Vulnerability Research Project**

**Participants:**

* Vikram Kher
* Varin Jaggi
* Tarik Wilkinson
* Patrick Kakooza
* Liza Moody
* Haoda Wang

**Introduction:**

For this deliverable, we switched our focus from hardware analysis to software analysis. Our primary tool that we used for this deliverable is the open source software Ghidra (https://github.com/NationalSecurityAgency/ghidra). Ghidra allows us to reconstruct the source code of several files from an apk. We managed to locate an apk file corresponding to our La-View camera and ran it through Ghidra. We present some of our findings below. Besides using Ghidra, we had wanted to use wireshark to analyze the network packets from the La-View Camera. Unfortunately, we do not currently have access to the La-View camera hardware due to activities surrounding a previous deliverable. We will work on the wireshark analysis for next deliverable.

**APK Extraction**

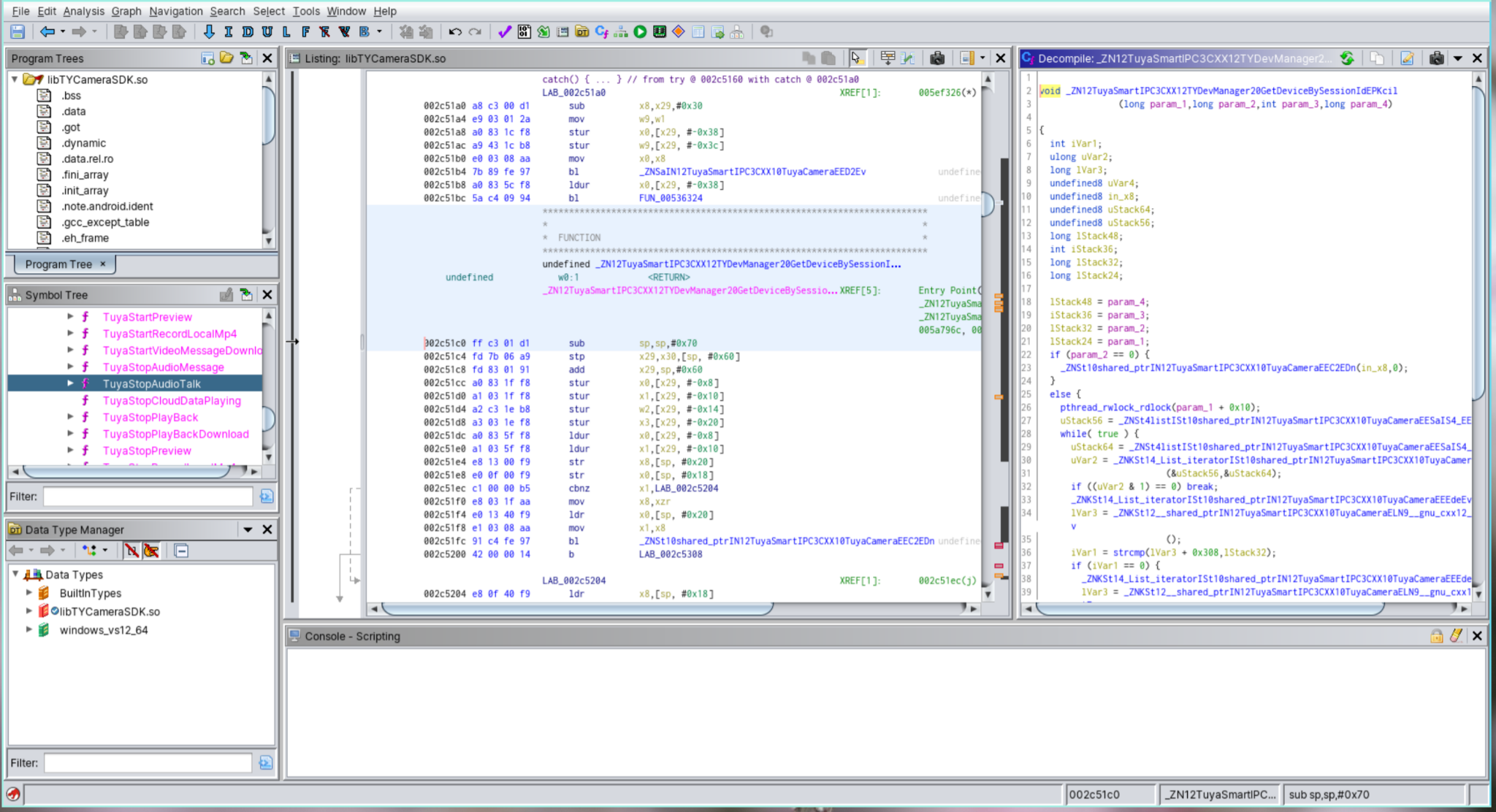
APK has same compression format as a .zip file, so just extract. Result layout:



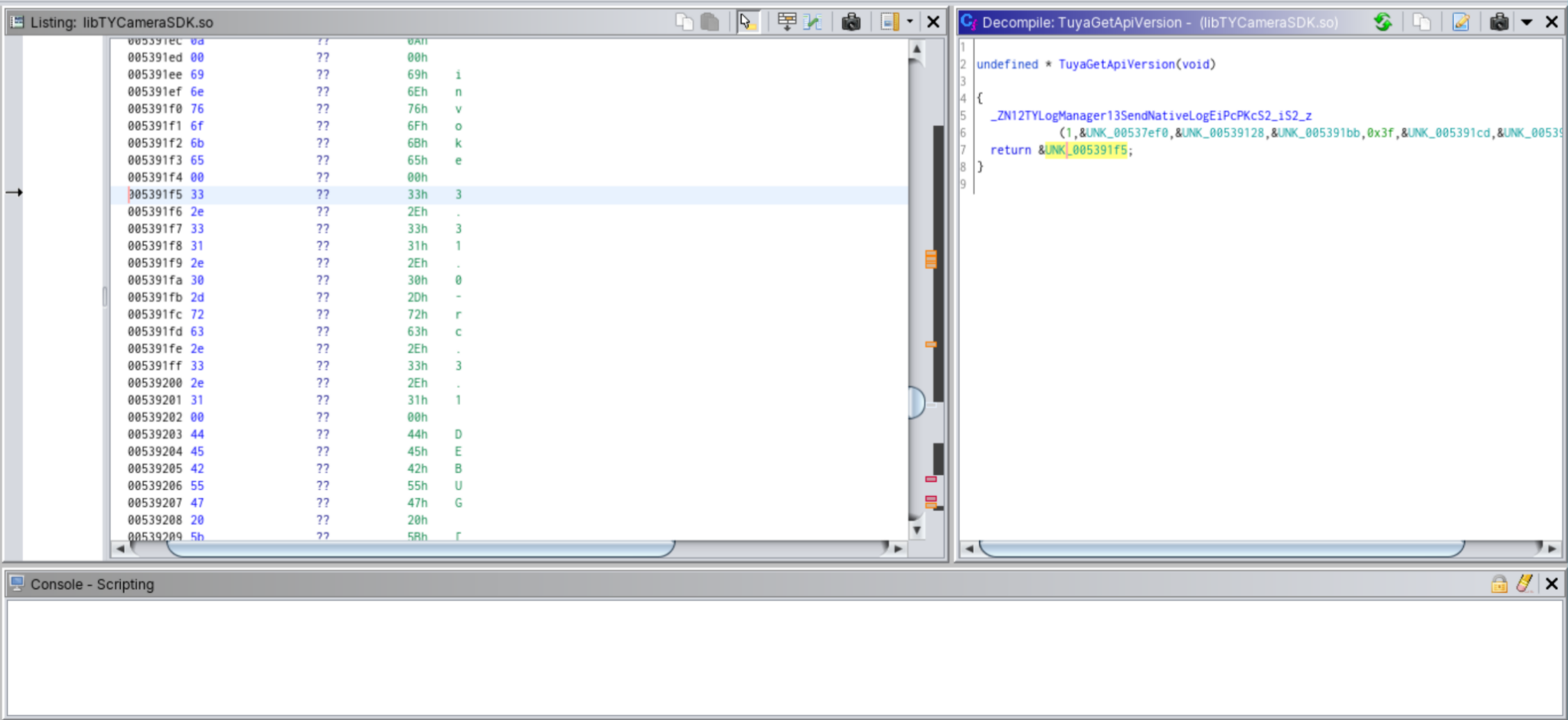
From this directory structure we could tell:

* User interface is made with Flutter
* Control logic done with Kotlin
* Two sets of libraries for ARM32 and ARM64
* Uses some pretty standard libraries: libav, LLVM libc++, etc.
* libTuya

**Disassembling Tuya Libraries**



All functions here make a HTTP request through “Tuya cloud,” very suspect



Tuya API version at 3.31.2 - possible exploits?

**Tuya**

* This is a IOT development platform (https://www.tuya.com/)
* Allows you to buy “off the shelf software” for IOT
* Seems like La-View Camera uses this platform
* Perhaps, we can find an exploit for Tuya’s technology platform and it will carry over to the La-View camera
* Additionally, we could attempt to find out what versions of libraries they are using and see if there are any known exploits for those versions of the libraries (which has presumable been patched in more recent editions)