390R Final Writeup on the Wyze Cam V2

Ian Anderson & Nam Nyguen

May 23, 2023

1 Overview

The Wyze Cam V2 is a low cost security camera (MSRP $26) developed by Wyze Labs, Inc. As with most Wyze cameras, it is largely a licenced copy of an existing Xiaomi camera, in this case being based on the Xiaomi Xiaofang 1S. `\use this for commands`

Some notable features of this camera include:

- 1080p sensor with a 110 degree wide angle lens
- F2.0 aperture + 4 infrared LEDs, good for low light
- Speaker & microphone for two way communication, so it can be used as an intercom
- Accompanying mobile app to see live video feed, capture images & bitrate information
- microSD card slot for storage

Our goal with this project was to see if we could find a way to exploit security vulerabilities in this camera through the process of extracting and examining it's firmware.

2 Technical Details

The Wyze Cam V2 is powered by the Ingenic T20 processor, an efficient SOC designed for video related IOT devices. Some of it's specs include:

- "XBurst" 1GHz single core CPU based on the MIPS32 architecture
- 128MB of DDR2 memory
- Hardware H.264 encoder supporting 1080p@30fps
- 600mw power draw

Something that stood out on the product info page was "Linux BSP, GCC tool chain, Glibc", confirming that this device most likely ran Linux.

3 Attempting to dump the firmware

We obtained two cameras on eBay, and after testing one of them to make sure it was fully working, went and disassembled it until we were able to take out the main board and the attached camera board. To gain shell access to the Linux system, we did some research and found that a hobbyist had figured out that three pins on the edge of the main corresponded to GND, RX and TX, and used an Arduino to gain serial access to the device. Using tools in the CICS Makerspace, we were able to replicate this and get to the login, but none of the passwords we found worked for the root account, so we were not able to directly get the firmware off the device itself. However, Wyze has versions of the device's firmware avaliable for download, though these are not the standard versions. We will examine the webcam firmware found on the Wyze website. (We later found a version of the RTSP firmware that was removed from the website some time ago, but most of this writeup will be focued on the webcam version.)

4 Basic examination

To start off, one of the first things we tried was using the `strings` command to see if we could find any useful data, as trying to read the binary just with `cat` would just result in a bunch of garbage being outputted to the screen, and `xxd` would take forever to go through with a binary of this size (11MB). To reduce the chance of simply getting a bunch of random valid characters in a row, we used the `-n16` flag to only print out ”words” of 16 characters or greater, which resulted in this output:

```
Fn.record_auto_list
AppVer=4.15.2.82
wpa_supplicant -D nl80211 -iwlan0 -c/system/bin/wpa.conf -B &
udhcpc -i wlan0 -p /var/run/udhcpc.pid -b &
+m=DGEhostapd_cli
hostapd_wpa2.conf
'_ _ _$_$_"_"_&_&_!_!_%_%_#_#_'_'
'? ? ?$?$?"?"?&?&?!?!?%?%?#?#?'?'
'_ _ _$_$_"_"_&_&_!_!_%_%_#_#_'_'
'? ? ?$?$?"?"?&?&?!?!?%?%?#?#?'?'
mount -o nolock,rw 10.0.0.167:/home/xuxuequan/Ingenicwork/sharenfs /mnt
restart_wlan0.sh
az#=f02`F##f22af#3f1
:*Fp"`D'"Fr"ad'2Fq
az%=fP2`F%#fR2af%3fQ
zISA_VERSION=5.6.1.32
root:x:0:0:root:/:/bin/sh
x:root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
webrtc_profile.ini
wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
model=isa.camera.isc5c1
mac=34:CE:00:E3:FD:D7
key=V2RX3WeYFLdWeEZc
&PsD7p5STEDiLK4DV
miio_client_helper_nomqtt.sh
videobuf2-vmalloc.ko
WH@XXH@HPDLFBDTZTPHRQRZVNAAAXBYUI^UF^A
```

```
+8$4,<"2*9%5-=#3
}Rlibsysutils.so
dongle_network_add_failed.wav
dongle_network_add_success.wav
E6rdongle_network_start.wav
Jdongle_sensor_delete.wav
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
7root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
=root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
iroot:rJ0FHsG0ZbyZo:10933:0:99999:7:::
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
Kroot:rJ0FHsG0ZbyZo:10933:0:99999:7:::
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
K!uvc_f22.config
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
G8uvc_jxf22.config
uvc_jxf23.config
[Qbnjp~iQr*V=s&:1$Lv
@VYSGFQ]KJ^UCRNYMBFQELZ^ITRVADBFNX\JFHTRJPD\B@XL
+8$4,<"2*9%5-=#3
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::
croot:rJ0FHsG0ZbyZo:10933:0:99999:7:::
libaudioProcess.so
```

While a few of these were still garbage, most of this was useful information that was telling about the development of this system. For example, `f`