ACCUVANT

GoPro or GTFO

An (Incomplete) Tale of (Kinda) Reversing an Embedded System

Agenda

- Intro
- GoPro Overview
- Previous Research
- Methodology/Findings
- Future Research/Next Steps
- Conclusion

INTRO

About Us

- Todd Manning a.k.a. “El Isleño”
  - Sr. Research Consultant, Accuvant Labs’ Applied Research Consulting
  - Previously Mgr. of Security Research at BreakingPoint Systems
- Zach Lanier a.k.a. “quine”
  - Sr. Research Consultant, Accuvant Labs’ Applied Research Consulting
  - (Net | App | Web | Mobile) pen tester type

Obligatory Puffin Slide

Why the GoPro?

- Highly popular, consumer “rugged” camera
- WiFi-enabled
- Possible applicability to other Amberella-based devices
  - Including commercial IP-enabled CCTV installations
- We focused mainly on GoPro Hero3 Black Edition
  - So *most* details apply, but may be some HW differences
- Plus: IT’S EXTREEEEEEEEEEEEEEME!

GOPRO OVERVIEW

GoPro Overview

- Ambarella A770 camera SoC
  - ARMv6 1136J-S core (@528MHz)
- Sitronix ST7585 LCD
- Atheros AR6233GEAM2D 802.11n + BT controller (not used)
- and more...

GoPro Overview

- H3B runs two operating systems:
  - ITRON
    - Embedded RTOS
    - Manages most of the camera bits
    - Runs the “GoPro” Webserver on 80/tcp
    - “Internal” interface to Linux (10.9.9.9)
  - Linux 2.6.38
    - Actually runs as a task within ITRON
    - Resides on private/internal network (10.9.9.1)
    - Runs Cherokee webserver on 80/tcp, but port fwd’ed from 8080/tcp externally

PREVIOUS RESEARCH

Evil Wombat!

- O.G. contributor to GoPro forum
- ARM firmware developer (???)
- Discovered (and shared) autoexec.ash
  - Script that runs on boot, can enable such fun things as serial console, telnetd, etc.
- Wrote firmware parsers, camera “unbrick” tool, and techniques for direct booting Linux kernel

ambsh

- Amberella shell - limited shell accessible over serial/USB

```
**************************************************************
*                                                            *
*                          ambsh ;)                          *
*                                                            *
**************************************************************

BST (178034), HAL (178034), CHIP (a7)
rtos mesg disabled
dsp mesg disabled
type 'help' for help

a:\>
```

- Discovery courtesy of Evil Wombat
  - Drop the following into autoexec.ash on SD card, reboot camera:

```
sleep 4
t app test usb_rs232 1
```


Side note: what not to do

```
a:\> t nand_op erase 0 10
erase block 0 erase block 1 erase block 2 erase block 3 erase block 4 erase
 block 5 erase block 6 erase block 7 erase block 8 erase block 9 success
```

You have a successful failure, and now your camera is bricked.

lu_util

- ITRON uses IPC message queue for bi-directional, inter-OS messaging (more on this later)
- lu_util is iTRON-to-Linux utility
  - Execute commands within Linux, such as enabling telnetd
  - Once again, discovery courtesy of Evil Wombat
    - Drop the following into autoexec.ash on SD card:

```
sleep 30
lu_util exec 'pkill cherokee'
lu_util exec '/usr/sbin/telnetd -l /bin/sh -p 80'
```


Root shell ;)

With telnetd enabled, root shell!

```
user@hi:~$ telnet 10.5.5.9 8080
Trying 10.5.5.9...
Connected to 10.5.5.9.
Escape character is '^]'.

/ # id
uid=0(root) gid=0(root)
/ # uname -a
Linux buildroot 2.6.38.8 #1 PREEMPT Fri Mar 1 18:03:04 PST 2013 armv6l GNU/Linux
```

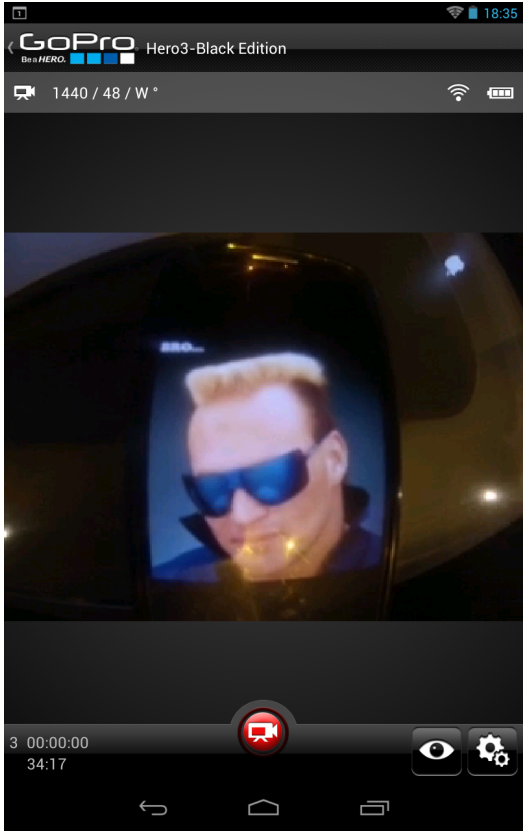
Konrad IT!

· Continued with the Autoexec.ash research
· Focused mainly on physical commands (button trigger, settings, loop…)
· Discovered and shared a way to extend the lapse in timelapse mode between 2 mins to 45 mins in HERO3 Black / HERO3+ Black / Silver cameras, allowing the camera to shoot for days, weeks, months or years!
Hack is free in github.
Web: chernowii.com
GitHub: @konradit

METHODOLOGY & FINDINGS

Overview - “GoPro App” Mode

- Camera acts as access point
- Mobile app connects to two webservers on camera:
  - “GoPro” Web Server for control / config
  - Cherokee for “real time” video preview (MPEG-TS via HLS)
    - App retrieves playlist from Cherokee with eight (8) 0.3 second clips for “streaming” preview
- WiFi Bacpac uses 10.5.5.9



Analysis - App Mode

- First step was traffic analysis
  - Captured comms between mobile device and camera
- Analyzed Android app using dex2jar, JD-GUI, apktool, & Androguard
  - Discovered additional URLs/commands
- Scanned all the network things!

Network Surface - App Mode

- Cherokee webserver (Linux)
  - Runs as root, despite listening on unpriv'ed port
  - No addt'l mitigations enabled (aside from NX & ASLR)

Bro, do you even PIE?

```
# cat /proc/1243/maps
00008000-0000c000 r-xp 00000000 00:0d 1278       /usr/sbin/cherokee
00013000-00014000 rw-p 00003000 00:0d 1278       /usr/sbin/cherokee
...
# cat /proc/1228/maps
00008000-0000c000 r-xp 00000000 00:0d 1278       /usr/sbin/cherokee
00013000-00014000 rw-p 00003000 00:0d 1278       /usr/sbin/cherokee
...
# cat /proc/1232/maps
00008000-0008a000 r-xp 00000000 00:0d 1276       /usr/sbin/cherokee-worker
00092000-00093000 rw-p 00082000 00:0d 1276       /usr/sbin/cherokee-worker
...
# cat /proc/1220/maps
00008000-0008a000 r-xp 00000000 00:0d 1276       /usr/sbin/cherokee-worker
00092000-00093000 rw-p 00082000 00:0d 1276       /usr/sbin/cherokee-worker
```

Network Surface - App Mode

- GoPro webserver (ITRON), in Mobile App mode
- Control of bacpac and camera
  - http://10.5.5.9/bacpac/...
  - http://10.5.5.9/camera/...
- Passes WPA2 passphrase as auth token
  - e.g. http://10.5.5.9/camera/cv?t=*MYWPA2KEY*

Some WS Commands - App Mode

- /bacpac/pw?t=*MYWPA2KEY*&p=%01
  - Powers camera on/off (01 or 00)
- /bacpac/sh?t=*MYWPA2KEY*&p=%01
  - Activates shutter (records or takes picture)
- /camera/LL?t=*MYWPA2KEY*&p=%01
  - “Locate” camera using its buzzer
- And more...

Overview - “WiFi RC” Mode

- Remote acts as access point, camera acts as mobile station
  - Remote/AP does not use any security - totally open
- Camera scans for HERO-RC-*XXXXXX* (where XX... are the last three octets of the BSSID/ MAC of the remote)
  - Prefers known BSSID, but can be configured to “pair” with new remote
- Remote uses 10.71.79.1

Analysis - RC Mode

- Remote acts as client, talks to camera(s) on 8484/udp
  - Discovers (newly) paired cameras via broadcast to 10.71.79.255
- Remote is basically a “dumb” client
  - Sends command, receives response, displays what’s on camera LCD

Conversation (UDP) - RC Mode

OO

se

lc

lc

lc + bitmap

se

lc

lc + bitmap

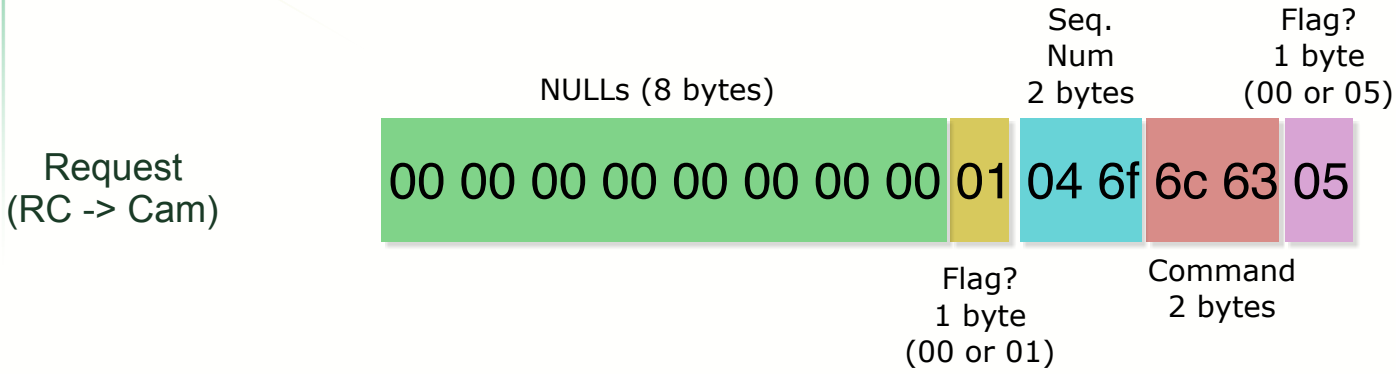
...

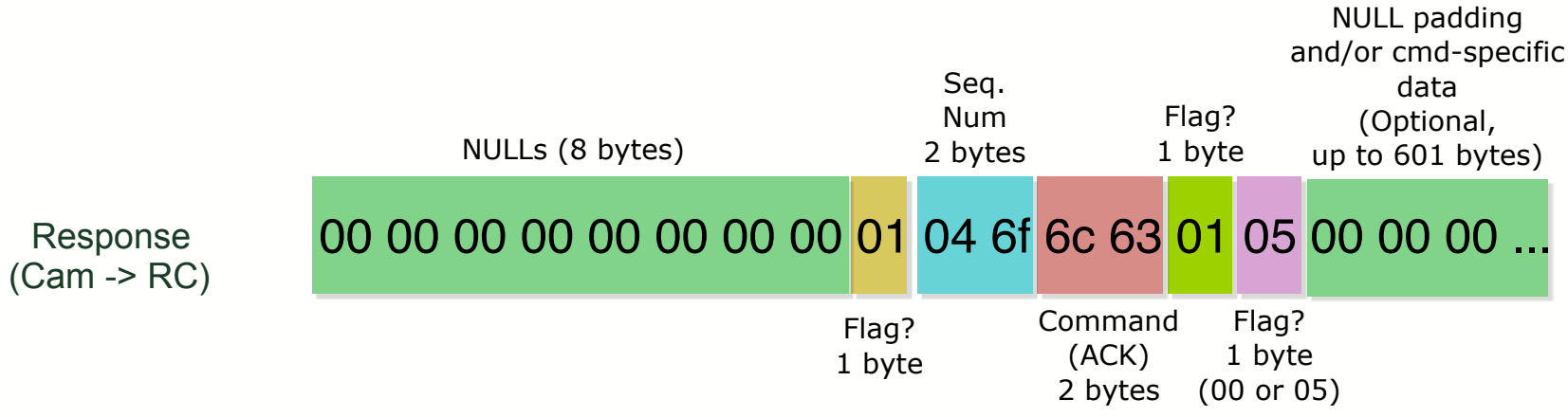
sh

sh

8484/udp

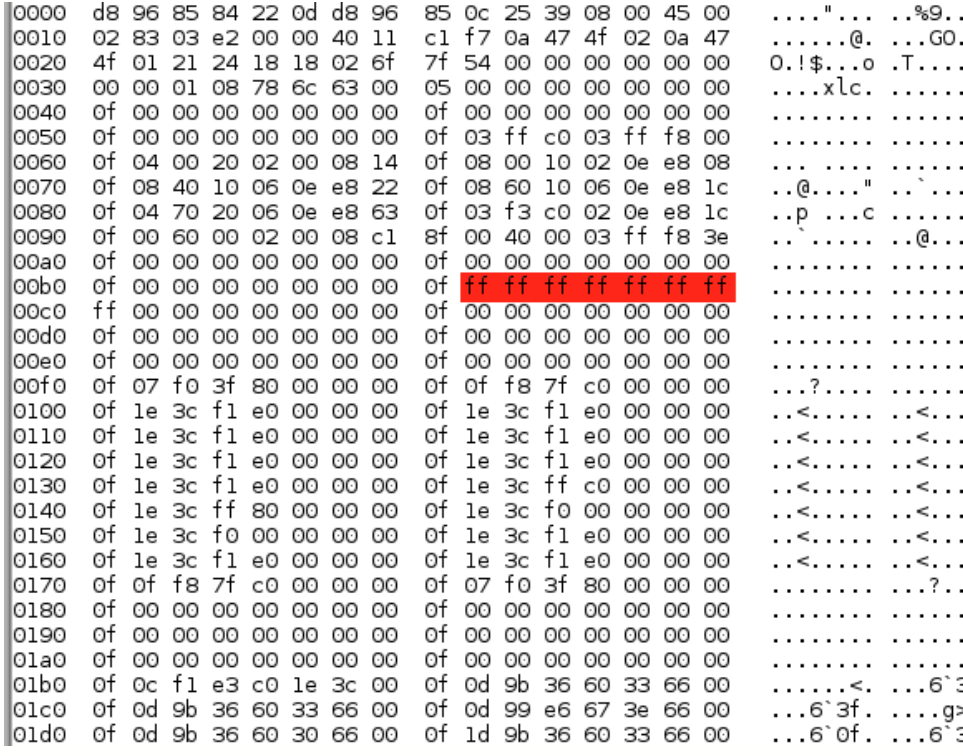
Protocol





“LC” command

- *LC* response is interesting/neat
- Kept noticing “larger”, 615 byte responses after *LC* command request

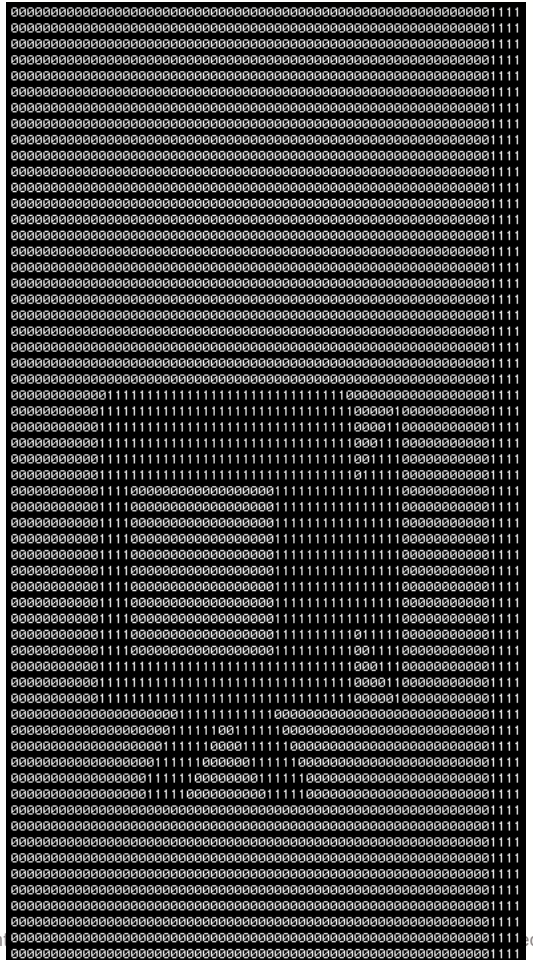


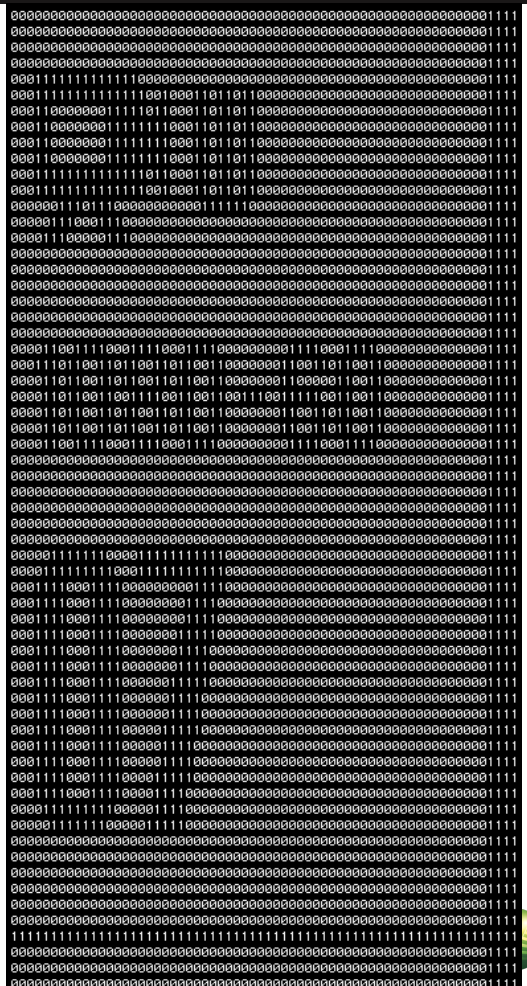
“LC” command

- *LC* short for “LCD” (?)
  - Yep.
- 615 byte response contains bitmap of camera LCD
- Quickly hacked up script to extract *LC* response body, “display” contents

“LC” command

=





Local Attack Surface - Linux

- No priv separation - everything runs as root
- ASLR enabled system wide, but no PIE or stack guard
- Decent slew of useful commands
  - Busybox & GoPro-specific tools
- Numerous “interesting” commands/daemons
  - amba_mq_handler
  - ombra
  - local_message_daemon
    - Prepares/sends and receives/parses JSON messages
  - network_message_daemon
    - Amongst other things, parses JSON messages passed on 7878/tcp (not remotely accessible)

IPC - Linux side

Message queue

```
/ # ipcs -p

------ Shared Memory Creator/Last-op --------
shmid      owner      cpid       lpid


------ Message Queues PIDs --------
msqid      owner      lspid      lrpid
0          root        788       753
32769      root          0         0
65538      root          0         0
```

Points to queue used by amba_mq_handler which handles IPC from Linux <-> ITRON

```
753 root       0:00 amba_mq_handler
```

Message processing API provided by libmsgprocess.so

IPC - ITRON side

Numerous registered IPC programs (viewable in ambsh with ipcprog command)

```
a:\> ipcprog
======================
Registered IPC programs
servers: i_status i_util i_ffs i_sdresp i_mq i_wifi_display i_dvf2web i_streamer i_heapmem i_example_util i_example_framer
clients: lk_util lu_util lu_net_config lu_wifi_config lu_dvf2web lu_streamer lu_rappctrl lk_sdresp lu_mq lu_lnxfio_stream lk_example_util lu_example_util lu_example_framer
======================


i_status (S) P:0x10000002, V:1 N:3
0xc0342bcc    1    4    0 00000000  0 00000000  0      500 lk_status_report
0xc0342c14    2    4    0 00000000  0 00000000  0      500 lk_boss_version_report
0xc0342da0    3   28    0 00000000  0 00000000  0      500 lk_time_event

i_util (S) P:0x10000001, V:1 N:17
0xc0342ee0    1    0    8 00000000  0 00000000  0      500 itron_gettimeofday
0xc0342f00    2    4    0 c0912f5c  1 00000000  0      500 itron_printk
0xc0342fa0    3  256    0 00000000  0 00000000  0      500 itron_fixed_printk
0xc0343088    4    4    0 00000000  0 00000000  0      500 itron_async_ipc
0xc03430d0    5    0    0 00000000  0 00000000  0      500 itron_pm_suspend
0xc0343118    6    0    0 00000000  0 00000000  0      500 itron_pm_resume
0xc0343138    7    0    0 00000000  0 00000000  0      500 lk_report_ready
0xc0343168    8    0    0 00000000  0 00000000  0      500 lk_report_resume
0xc0343198    9    0    0 00000000  0 00000000  0      500 lk_request_suspend
0xc03431c8   10    0    0 00000000  0 00000000  0      500 lk_request_shutdown
0xc03431fc   11    0    8 00000000  0 c0912f60  1      500 lk_get_exfb
0xc0343244   12    0    0 00000000  0 00000000  0      500 lk_report_fb_owned
0xc0343274   13    0    0 00000000  0 00000000  0      500 lk_report_fb_released
0xc03432a4   14    0    4 00000000  0 c0912f64  1      500 lk_absuspend_check
0xc03432bc   15    0    0 00000000  0 00000000  0      500 lk_absuspend_enter
0xc03432d4   16    0    0 00000000  0 00000000  0      500 lk_absuspend_exit
0xc0343344   17    8    0 00000000  0 00000000  0      500 lk_set_device_owner
```

FUTURE RESEARCH & NEXT STEPS

Future Research

- Remote monitoring
  - Legitimate, bespoke 3rd party clients
  - Using the camera to spy
- Dumping firmware from WiFi Remote
- GoPro 30-pin bus interface
  - Remarkably similar to Apple i-device connector
  - Used for interfacing with product add-on devices
- Further analysis of IPC stuff and ITRON

Code, notes, etc.

https://github.com/quine/GoProGTFO

Watch this space!

Will drop public scripts, tools, etc. here soon

Questions / Contact

- zlanier@accuvant.com
- tmanning@accuvant.com
- https://twitter.com/quine
- https://twitter.com/tmanning

Greetz:

snare, bNull, jono, aloria, cji, d0c_s4vage, KF, donb, k8em0 cmulliner, natron, tigerbeard, jduck, m0nk_dot, drspringfield, zek, marcinw, sl0w, drraid, amberalla, solareclipse, mckeay, katalyst, cd, sbit, awr, tkrpata, kingpin, thegrugq, eas, rumble, ddz, sa7ori, HockeyInJune, pof, rmogull, oxff, zenofex, hustlelabs, redpantz **(Todd says he's sorry)**, cmiller, chrisko, mcalias, rfp

All of NORDICSEC!

And the rest of the jerks in
#busticati & #aha
And to anyone we forgot: sorry.

www.accuvant.com