GoPro or GTFO

A Tale of Reversing an Embedded System
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
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 EXTREEEEEEEEEEEEEEEEME!
GOPRO OVERVIEW
GoPro Overview

- Ambarella A770 camera SoC
  - ARMv6 1136J-S core (@528MHz)
- Sitronix ST7585 LCD
- Atheros AR6233GEAM2D 802.11n + BT controller
- 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
- If you’re in the audience, plz to be letting us buy you a drink
• Amberella shell - limited shell accessible over serial/USB

```bash
* *
* 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

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

```bash
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
```
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
```
METHODOLOGY AND FINDINGS
Analysis - “GoPro App” Mode

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

- Remote acts as access point, camera acts as mobile station/client
  - Remote/AP does not use any security - totally open
- Camera scans for HERO-RC-XXXXXXXX (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
Network Attack Surface

- Cherokee webserver (Linux)
  - Runs as root, despite listening on unpriv’ed port
  - No addt’l mitigations enabled (aside from NX & ASLR)
    - Exec base is not randomized
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
Local Attack Surface - Linux

- No priv separation - everything runs as root
- ASLR enabled system wide
- Decent slew of useful commands
  - Busybox
  - GoPro-specific tools
- Numerous “interesting” commands/daemons
  - amba_mq_handler
  - ombra
  - 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
```
IPC - ITRON side

Numerous registered IPC programs (viewable in ambsh with ipcprog command)
FUTURE RESEARCH & NEXT STEPS
Future Research

- Remote monitoring
  - Legitimate, bespoke 3rd party clients
  - Using the camera to spy
    - Following up on accessibility of MPEG-TS streaming
- 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
- Backdoors, persistence, blah blah blah
Code, notes, etc.

https://github.com/quine/GoProGTFO

Watch this space!
Will drop public scripts, tools, etc. here soon
Questions / Contact

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

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

Greetz:

bNull, jono, aloria, cji, d0c_s4vage, KF, cmulliner, natron, tigerbeard, jduck, m0nk_dot, drspringfield, zek, marcinw, sl0w, drraid, amberalla, solareclipse, katalyst, cd, sbit, awr, tkrpata, kingpin, thegrugq, eas, rumble, ddz, sa7ori, HockeyInJune, pof, oxff, zenofex, hustlelabs, redpantz, cmillerchrisko, mcalias, rfp

And the rest of the jerks in
#busticati & #aha

And to anyone we forgot: sorry.