Teaching Your Toaster New Tricks

Or doing cool things with IoT
About Me

- About me
  - Student Researcher at Cal Poly Pomona—Learn by doing!
  - Focus on Internet of Things and Embedded Devices
  - Participate in CCDC, CPTC, and CTF competitions regularly

- 3 years of active research in embedded devices
Agenda

- Look at the various types of devices that are available
- Find ways to make use of End of Life devices
- Find better ways to make “smart” devices

- Profit? Or end up with a IoToaster!!
Lets clear things up

https://www.technologyreview.com/s/400889/internet-on-a-chip/
Let's clear things up

Then there was....
Lets clear things up

And the future holds....
But this is all you get
The Victims...

- Routers
- Cameras
- NASes
- Travel Routers/Hotspots
- (WeMo) Coffee Maker
- Door Locks
- (WeMo/D-Link/TP-Link) Power Outlets
- (WeMo) Air Purifier / Cooler
- Drones (Parrot, Elfie, Generic)
- “Smart” TVs
Attack of the Clones

- Many IoT devices are based on reference models or are clones
- Cheaper to develop and release but doesn’t mean more secure
Dividing Everything Up

“Customizable Firmware”
- Asus **N16, N66, and AC88**
- GL.iNet **AR150 and 300N, AR300**
- WeMo *Outlet, Crockpot, Coffee Maker, and Air*
- TP Link **TL-WR710N and TL-WDR3600, HS100**
- HooToo **TM-02**
- Netgear **AC3200**
- Fosscam *Wifi Camera Clones*

“R/W Systems”
- *Parrot Drones*
- *WD My Cloud* (Pure Debian!)
- QNAP **TS-251**
Why Divide Up Devices?

- Ensure we know what we’re dealing with and what we will have to repair
- Level of Effort
- Identify what will be required to access the device
- Identify possible security issues as entry points
Parrot Drones

- Variety of drones available
- Relatively cheap
- Consistent Specs Advertized:
  - 1GB of RAM
  - 1ghz “Dual Core” Processor
- Actually:
  - 256-512MB of RAM and 400mhz Processor
- Great Marketing!

http://www.cpp.edu/~polysec/UAV/
Expectations
Expectations
Reality
Normal Use

- Phone App connects via WiFi
- Transfer data from the drone via FTP and AR-Stream Protocol
- Emergency Attack Mode?!
Gaining Access

bash-4.35 telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^[J'.

BusyBox v1.14.0 () built-in shell (ash)
Enter 'help' for a list of built-in commands.

# ls
README  dev  firmware  mnt  sbin  usr
Why is this still a thing?
So much is "right" with Parrot Drone Systems

As other talks have shown – it runs telnet and ftp and random other ports – as we see "bash proxy".

Factory reset doesn’t factory reset anything except config.ini.

Firmware modification should not be made 60ft in the air!
What does that mean?

- Easy modification and exploitation of drones
- Perform modification on any local Parrot drones
- Communicate between Drones (multiplayer)
  - Stop drones
  - File Transfer / Take-Over
- Malware Upload / Credential Theft
killall program.elf?

- Drone runs out of program.elf
- Everything else is just Linux.
- Pretty sure this is what they mean by fully upgradable

- If you upgrade the firmware or just stop program.elf....
Improvements?

- Use OpenWRT
  - Compiled...
- BuildRoot
  - Compiled
  - Upload Directories

And...
What went wrong?

- Build was set up after specific kernel / ulibc configurations
- No easy way to replace the system without taking up too much space
- Possibility of the brick
Try again!

- Compile Statically?
All components patched to run out of /opt/
Next Generation is: Entware-NG
Plenty of packages, works everywhere
Stand back

There's science in this shit.
Ideas!

- Why couldn’t we return this? With “improved” firmware?
- Download files to people’s phones or tables.
- Mobile Captive Portal
- Drive by Drone Capture and Pivot
Captive Portals: Things Learned

- Most operating systems now have built in handling of captive portals.
- On latest platforms this interface is restricted

- However, on Windows and iOS you can have links that will allow people to open up an unrestricted browser
- Time to send some files!
Parrot Drones have a unused feature called "Multi-Player"

Allows drones to connect to a shared network or each other easily

This also allows us to connect to drones and take them over

- Drones are configured with IPTables but only flight control is blocked
- Telnet and ftp are enabled and not blocked, allowing us to transfer and run payloads
WD MyCloud

- "With its robust software...

- Its Just Debian!

- Really..
  "Firmware Updates" are .deb packages!
BUT WAIT

THERE'S MORE
Root?

- We don’t even have to try
- Web UI is fully optimized PHP (still)
- Multiple vulnerabilities in the Web UI.
  - Old: Status Checker run arbitrary Commands
  - http://wdmycloud.local/api/1.0/rest/safepoint_getstatus?handle="$(telnetd)"
- New: Firmware Updater still allows command injection
Fun with Debian

- Restore the Debian repos, you have a fully functional arm Debian box.
- Upgrade or install anything you would like!
- Want to use Kali Tools? Sure thing!
No such thing as factory!

One thing we’ve seen so far with all these R/W devices.

- Factory Reset is just a name. IT DOES NOTHING... EVER...
- WD MyCloud factory reset does not restore Web UI files, does not reset most content on the drive.
- You want persistence... This is how you get persistence.
How did we find out?
Great News for Us!

- Remove WD’s features
- Low-Powered Server
- Network Monitor?

Possibilities are almost endless with one caveat - the kernel has been customized
Great News for Us!

- Remove WD’s features
- Low-Powered Server
- Network Monitor?

Possibilities are almost endless with one caveat - the kernel has been customized.

240 days continuous uptime running bro via a tap.
The other option...

- DD-WRT, OpenWRT, LEDE
- Firmware compresses extremely well
- (Usually) Easily unbricked, easily updated, easy maintenance
- Deploy to one system or dozens of all types, sizes, and kinds
Good and Bad

- The good: You can setup packages, resources to always run, and restore on failure.

- The bad: You are stuck with a set of packages and resources.

- The really bad: Not all devices are the same – even if they have the same chip! Fixes often required to setup a device (but upgrades are easier)
RA RT5350(F)
Why?

- Used by WeMo and dozens of other IoT platforms
- Usually has accessible UART (Serial)

Specs:

- 16MB flash, 32MB ram
- ~360mhz processor
- 802.11n 2.4ghz
- 4 port 10/100 switch (support)
- 1 usb
- GPIO
Plenty of Open Devices

■ VoCore 1

- Runs OpenWRT from the start, no need to provide additional patches

■ HooToo Devices (TM-02)

- Fully supported by OpenWRT, simply needs an initial “factory image”
Back to this...
$ nmap 10.10.10.254 -p 22,23; telnet 10.10.10.254

Starting Nmap 7.12 (https://nmap.org) at 2016-07-18 22:21 PDT
Nmap scan report for 10.10.10.254
Host is up (0.0022s latency).
PORT    STATE    SERVICE
22/tcp   closed   ssh
23/tcp   open     telnet

Nmap done: 1 IP address (1 host up) scanned in 13.03 seconds
Trying 10.10.10.254...
Connected to 10.10.10.254.
Escape character is 'A'.

TM03 login: admin
Password:
login: can't chdir to home directory '/data/UsbDisk1/Volume1'
$ echo $USER
admin
$ cat /etc/shadow
root:$1$0cz0b934g$hLY0jyeFPlfEXVMdqUFsEj/:15386:0:99999:7::
bin:*:13341:0:99999:7::
daeon:*:13341:0:99999:7::
A better way?

- Pretty much all run OpenWRT
- They’re REALLY AWESOME for price
  - $30 $25 gets you either:
    - 256mb of RAM, 500mhz processor, and 64mb of flash, microSD Slot
    - 64mb of RAM, 400mhz processor, 16mb of flash, PoE
- Pretty sweet specs for a cheap device that fits in your palm
- Time to put them to use!
One small problem: Value Add
Stratum-1 GPS NTP Server

- High Accuracy
- No need to connect to the internet
- Self contained and very low power!
  - ~300 mA/h
  - PoE Capable

- Gl.iNet AR150
  - 400mhz
  - 16MB ROM / 64MB RAM
  - 4 pins GPIO
Final Result:

- RTC (DS3231)
- External Ant.
- DHT11/22
- GPS Module
- POE Module
Getting there...

- We need:
  - Serial to be free (for GPS to use)
  - PPS via GPIO (Pulse Per Second)
  - Easy deployment
  - i2C Support and DHT Support
Building Made Easy

● Tips:
  ○ Make menuconfig - good for configuring packages, resources, and anything “optional”
  ○ Make kernel_menuconfig - Internal modules built into the kernel - RTC, PPS, GPIO modules are here.
  ○ When done, always make defconfig
Machine selection
- [ ] OpenNRT specific image command line hack
- [ ] Endianness selection (Big endian)
- [ ] CPU selection
- [ ] Kernel type
- [ ] General setup
- [ ] Enable modules module support
- [ ] Enable the block layer
- [ ] Bus options (PCI, POMCSA, EISA, ISA, TI)
- [ ] Executable file formats
- [ ] Power management options
- [ ] CPU Power Management
- [ ] Networking support
- [ ] Device Drivers
- [ ] Firmware Drivers
- [ ] File systems
- [ ] Kernel hacking
- [ ] Security options
- [ ] Cryptographic API
- [ ] Library routines

<Select>  < Exit >  < Help >  < Save >  < Load >
Building Made Easy

- **Files:**
  - Full root structure in ./files/
  - Configurations:
    - Rc.local - Runs at boot, good for some settings
    - Init Scripts - Better, runs at specific target
    - Inittab - By default responds on serial interfaces
What to include?

- Chrony has built in support for RTCs and PPS
- GPIO-PPS
- Lsof
- NTP Utils
- GPSD
- Custom GPIO-PPS “driver”
  - By default driver has no settings
  - You must write mappings to support each device IO type
  - AR7XXX has IRQ so we can use that
Why?

- ImageBuilder / Source is significantly smaller than adding packages after install
- Allows us to deploy settings, configurations, again and again
  - Mesh networks
  - Cheap APs
  - Easy restore
- My current uses:
  - Low Power Emergency Box
  - NTP Server
  - Travel Hotspot/Router
  - Network Tap
Time to build something!
Fosscam (Clones)

- Runs Linux 2.4-uc0
- Very modern with full IPv4 networking stack!
- Not a lot of space to customize, but easily accessible serial
- Some clones are implemented poorly, have vulnerabilities and telnet
- Some clones can swap firmware with other manufacturers
- API is based on a SDK
- We can use this to connect and use the camera features
TP-Link HS100

- Like other “Smart Plugs” has no authentication
- Designed to be used “locally” or “in the cloud”
- Protocol is just static-key rotation, easy json on decode
- No obvious way to reflash (unlike WeMo), UART accessible

- Not ideal But: Just put on its own its own WiFi
- How to use it though?
Smart WiFi

- Now have an isolated network, but how do we use it?
Light Dude
Light Dude

- Amazon Dash Buttons are fun
  - Connects to WiFi
  - Uses AA battery to power SOC
  - Very low power
- Performs DHCP request and TLS connection to Amazon
  - We can listen to DHCP
  - Sadly it makes multiple requests...
Light Duder

- Taking multiple IoT devices and using them for good!
  - Smart Camera (From before)
  - Amazon Dash Buttons
  - Real Time / Sunrise / Sunset Data

- Automatically turn on lights when:
  - motion is detected
  - Multiple rules trigger
  - Sunrise/Sunset
  - Weather
Light Duder
SOUNDS GOOD TO ME!
One last Note
Great! But...

- I actually have a hybrid of these suggestions
- I have a bridge router to connect my network and the IoT
  - Allows access to weather reports
  - Allows access to syslog (out)

This allows me to keep the risk relatively low but provide all the features I need without the IFTTT / Internet.
Any questions?

Feel free to contact me:

On Twitter: @spiceywasabi