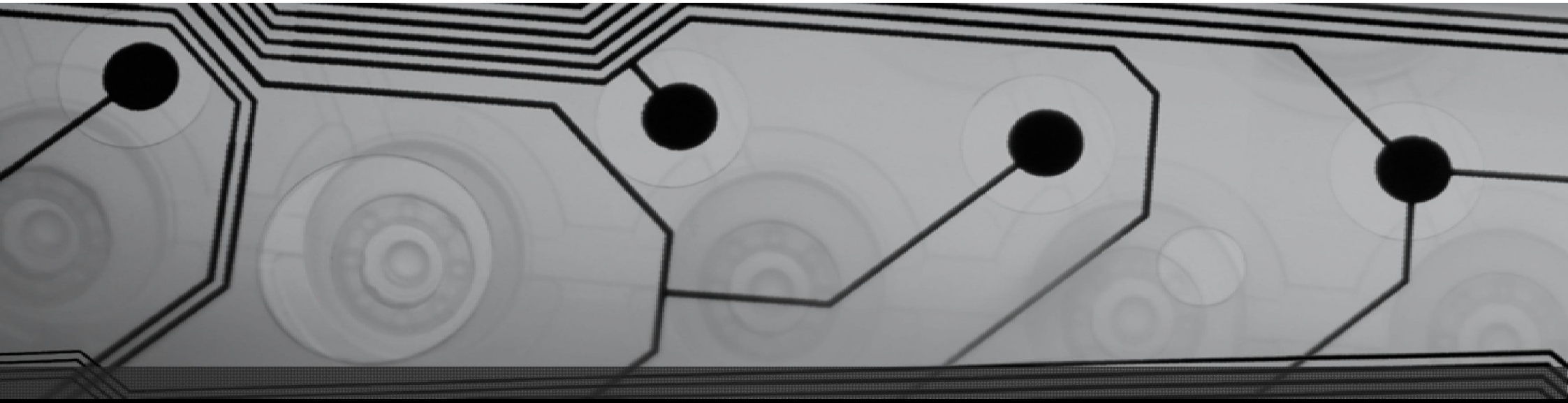
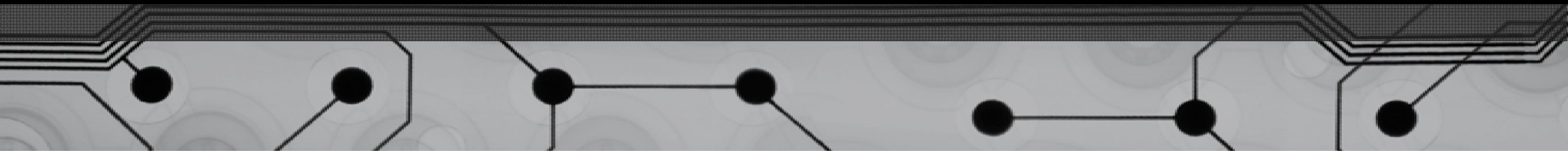


Making and breaking security in embedded devices

Yashin Mehaboobe
SDR Engineer, Bastille Networks



#whoami

- Security enthusiast
- SDR Engineer @ Bastille Networks
- Hardware tinkerer
- Speaker: CoCon 2013, Nullcon V, HITB Amsterdam and Kaspersky Cyberconference
- Organiser, Defcon Kerala

twitter.com/YashinMehaboobe
github.com/sp3ctr3
http://www.linkedin.com/pub/yashin-mehaboobe/38/a2/367

Why embedded?

- Large numbers
- Critical infrastructure dependent on embedded devices
- Network devices (both enterprise and SOHO)
- Even if it's not critical:
  - Botnet fodder
  - Pivoting
  - Storage for the bad guys
- On the internet and unsecured (Mostly)

But…..why?!

- REPRODUCTION:
  - Understand how the product works by reverse engineering it
  - Build a similar product
  - $$$$$Profit$$$$$$$
- FREE STUFF!:
  - Bypass restrictions
  - Get premium services
- UNLOCKING FEATURES:
  - Ex:Install dd-wrt
  - Don’t have to pay extra

But…..why?!

- ACCESS TO OTHERWISE SECURE NETWORKS
  - No one expects the embedded inquisition!
  - SOHO/Enterprise routers are not audited most times
  - No one checks the firmware

TOOLS OF THE TRADE

Choose your weapons!

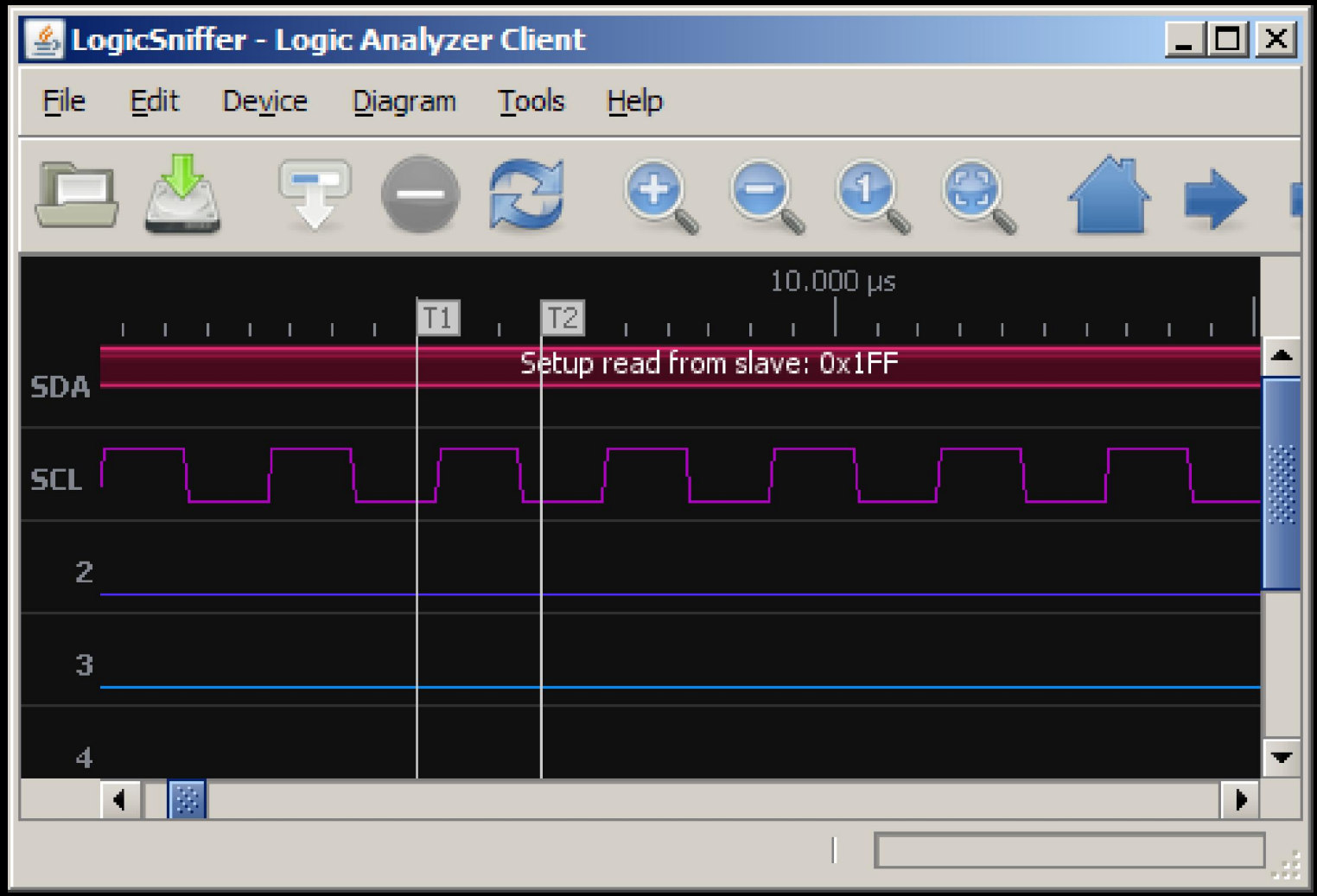

BUY
ALL THE HARDWARE
memegenerator.net

A word about equipment

- Good equipment = $$$$
- Use open source equipment such as the bus pirate, hackRF, OpenBench etc…
- Commercial tools work better in most of the cases
  - Would be a good investment
- Have at least one each of the separate categories of tools
  - Logic Analyzer
  - RF Spectrum Analyzer
  - Oscilloscope
  - JTAG debugger
  - Etc……

LOGIC ANALYZERS

- Monitor communication
- Decode protocols
- Replay (in some cases)
- Cheap (44$ to 500$++)
- Open source ones:
  - Open Bench
  - Bus Pirate

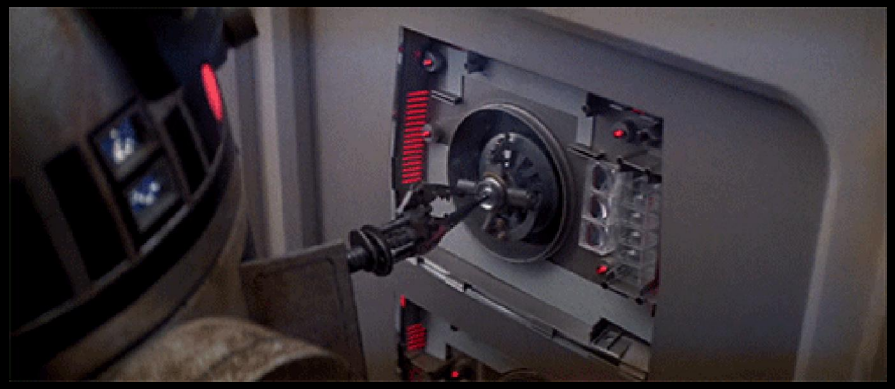

LogicSniffer - Logic Analyzer Client
File
Edit
Device
Diagram
Tools
Help
10.000 µs
T1
T2
Setup read from slave: 0x1FF
SDA
SCL
2
3
4

RF Analysis tools

- For scanning the RF frequencies
- Recognizing signals
- Storing and replay
- SDRs are your friends!
- Example:
  - RFExplorer
  - RTL-SDR
  - HackRF/BladeRF/USRP

Oscilloscope

- Digital/Analog
- Useful for noting timing
- Can also help in recognition of communication protocol
- Very much needed



Debug Ports

Debug ports FTW

- Ports setup to allow developer/engineer access during testing/repairing
- Loved by hackers because of the access it provides
- Different types:
  - JTAG
  - Serial
  - LPC (Xbox/TPM)
- Allow access to boot messages
- Allows you to log in without authentication
- Sometimes you can even du

Debug ports identification

1. Identify the ports
2. Connect the debugger/communication device
3. Profit!

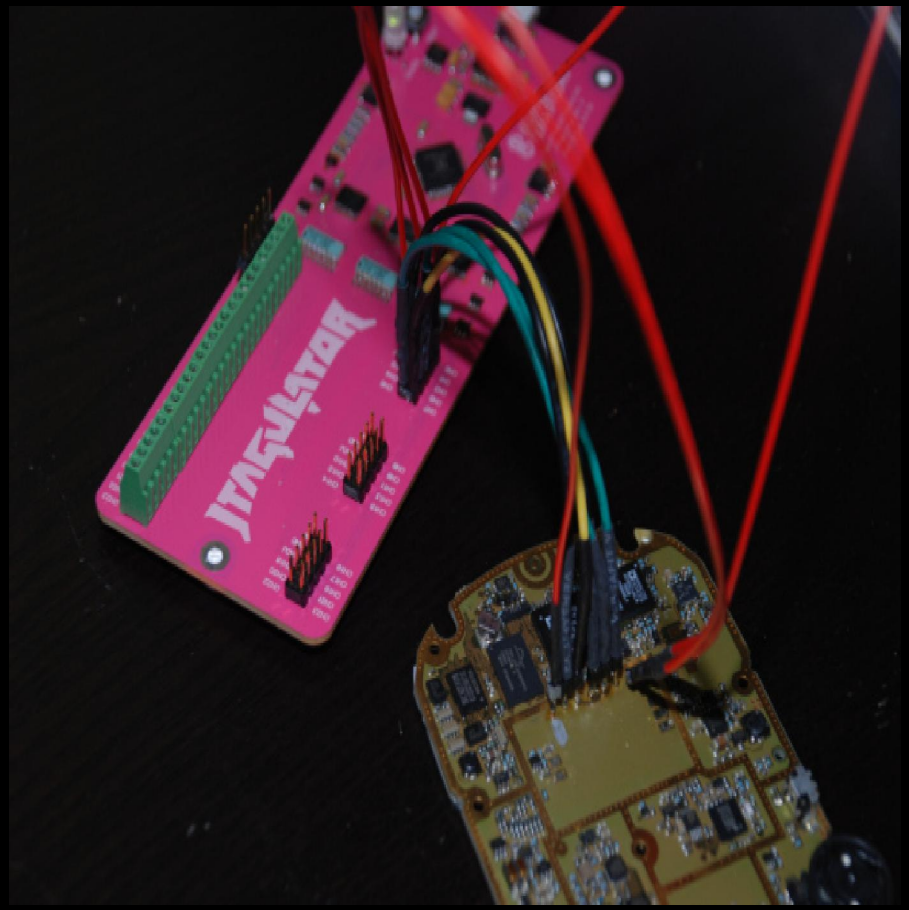
- First step is the most complex
- Methodology varies from protocol to protocol
- Number of points is a good indication

Identifying Serial ports

- Serial has 4 lines:
  - Vcc
  - Ground
  - Rx
  - Tx
- Identify ground pin with a multimeter continuity test
- Find vcc by powering up and checking vcc + ground with multimeter
- Tx will be the pin with high activity
- Rx will be the other
- Identify baudrate by trial and error
- JTAGulator has support for serial

JTAG ports

- Joint Test Action Group
- Used for debugging, updating firmware etc…
- Running homebrew on Xbox
- Dumping firmware
- Use JTAGulator for finding JTAG ports
- OpenOCD has support for a large number of JTAG debuggers



Defending against debug port attacks

- Disable unneeded ports
- Use authentication for the debug ports
- Shell access should not be given without authentication
- Unfortunately these defenses may not be practical in some cases

Electronic bus attacks

SPI,UART and I2C

- SPI, UART and I2c are some of the more commonly used protocols in embedded devices
- There is no authentication or authorization
- It is trivial to sniff traffic
- Very easy to replay attacks
- Bus pirate would be a good tool
  - Hardware hackers swiss army knife
  - Developed by Dangerous Prototypes

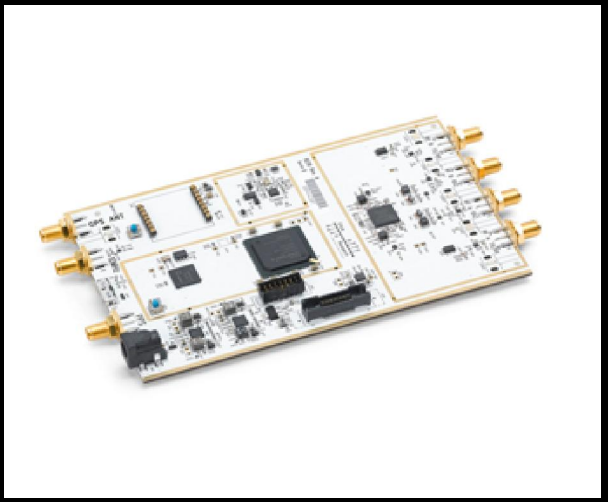
Radio communication

Sniffing radio signals

- Use to be hard and expensive
- With the arrival of SDRs the situation changed
- Now you can RX and TX with hardware ranging from 20$ RTL SDR to 1000$ devices
- Most signals aren't encrypted
- Some rely on FHSS (Not a good idea)

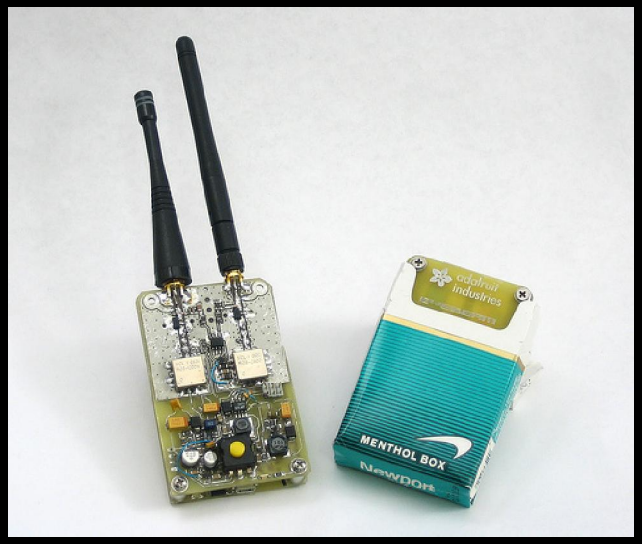
Tools used

- For most radio communication attacks an SDR would suffice
- Mainly because they can TX and RX in a wide range of frequencies
- Some examples are
  - USRP B210
  - HackRF
  - BladeRF
  - RTLSDR
- You can also use RFCat (cc1111 based attack toolkit)
- Ubertooth One can be used for Bluetooth sniffing
- OR sniff the buses of the transmitter



RF attacks

- Jamming
  - Basically DoS at RF level
  - Decreases SNR
  - Techniques differ
  - Some even disrupt handshakes
- Replay
  - Capture signal
  - Store it
  - Replay at some other time



Defenses against RF attacks

- FHSS is effective against jamming
- Use of encryption will defeat most sniffing attacks
- Encryption is built into most transmitters
- Unfortunately it is not used as much as it should be
- Rolling code system is a good defense against replay attack

Flash memory forensics

Flash memory

- Nonvolatile
- Used to store data
- Firmware is usually stored in flash memory
- Usually uses SPI for communication
- Usually does not have any protection

Extracting data from flash memory

- In circuit:
  - Don’t remove the chip
  - Use a chip programmer or bus pirate to read data
- Desoldering
  - The chip should be removed by desoldering it.
  - It is then accessed using a chip programmer to get the data
- Firmware can be extracted in this manner

Defenses against Flash memory forensics

- OTP memory protection bits
  - Doesn’t allow the modification of flash memory
  - Only useful against modification attacks
- Encryption
  - Storing the firmware/data encrypted would defeat memory forensics
- Also not storing confidential info on the chip

Firmware/Code Analysis

(In)Security

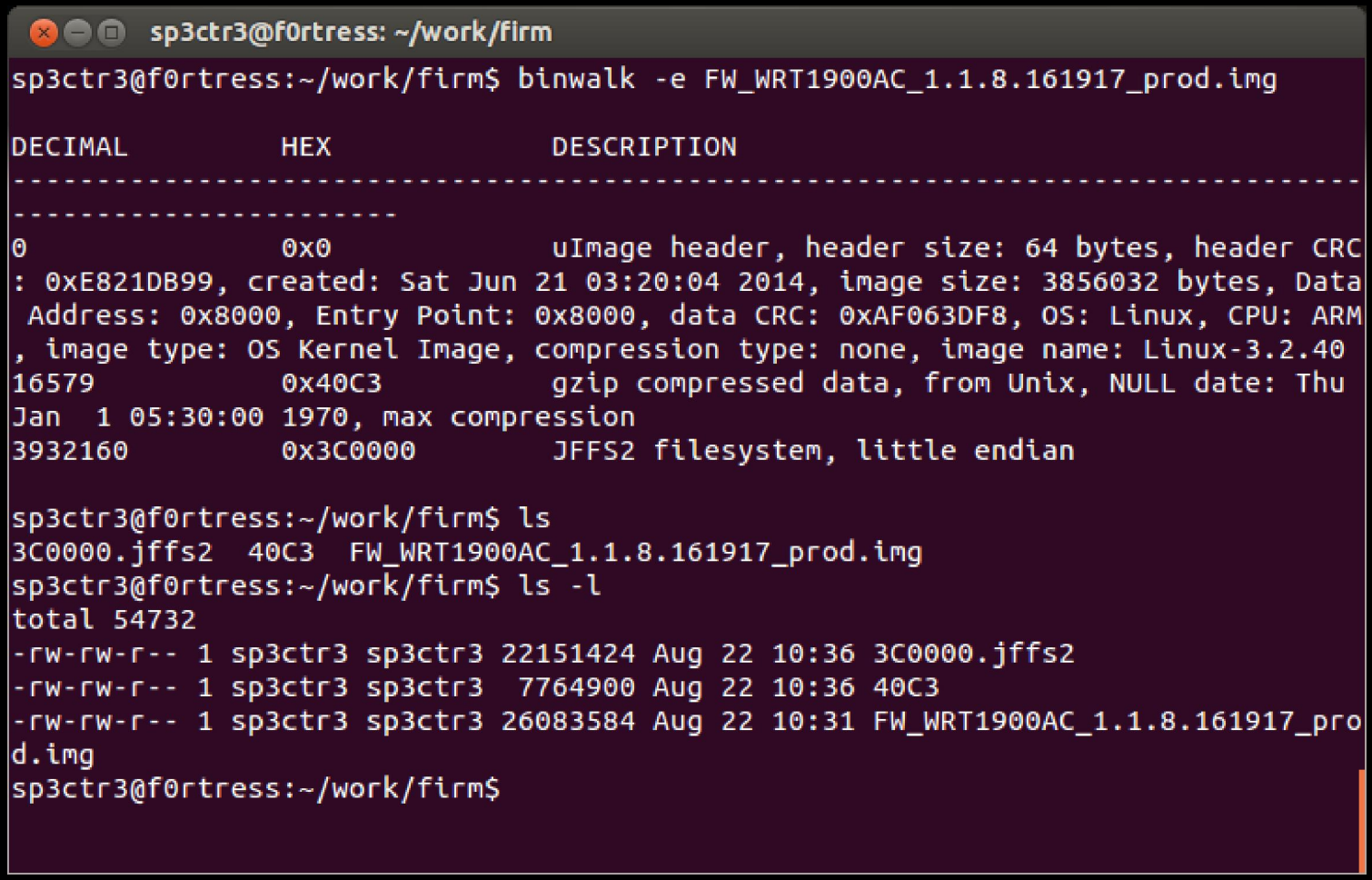
- Code is outdated in most devices
- Routers are the worst transgressors
- Most are internet facing
- Have more vulns than a CTF challenge
- Code is available for us to check and find vulns

Firmware

- Almost always linux
- Bootloader is usually Uboot
- Serial output usually gives you hints about the device
- Some may be obfuscated
- Can be obtained by either:
  - JTAG dump
  - Flash dump via Serial
  - Flash dump via chip desoldering
  - From the company website

Analysing firmware

- Usually various sections wrapped into one bin file
- You can use dd to separate
- Best option is to use binwalk
- Binwalk is a tool by Craig (of devttyso blog(great resource for hw reversing))
- Automatically analyze and extract firmware files


sp3ctr3@f0rtress: ~/work/firm
sp3ctr3@f0rtress:~/work/firm$ binwalk -e FW_WRT1900AC_1.1.8.161917_prod.img
DECIMAL HEX DESCRIPTION
0 0x0 uImage header, header size: 64 bytes, header CRC: 0xE821DB99, created: Sat Jun 21 03:20:04 2014, image size: 3856032 bytes, Data Address: 0x8000, Entry Point: 0x8000, data CRC: 0xAF063DF8, OS: Linux, CPU: ARM, image type: OS Kernel Image, compression type: none, image name: Linux-3.2.40
16579 0x40C3 gzip compressed data, from Unix, NULL date: Thu Jan 1 05:30:00 1970, max compression
3932160 0x3C0000 JFFS2 filesystem, little endian
sp3ctr3@f0rtress:~/work/firm$ ls
3C0000.jffs2 40C3 FW_WRT1900AC_1.1.8.161917_prod.img
sp3ctr3@f0rtress:~/work/firm$ ls -l
total 54732
-rw-rw-r-- 1 sp3ctr3 sp3ctr3 22151424 Aug 22 10:36 3C0000.jffs2
-rw-rw-r-- 1 sp3ctr3 sp3ctr3 7764900 Aug 22 10:36 40C3
-rw-rw-r-- 1 sp3ctr3 sp3ctr3 26083584 Aug 22 10:31 FW_WRT1900AC_1.1.8.161917_prod.img
sp3ctr3@f0rtress:~/work/firm$

Defenses

- Review your code!
- Obfuscate your firmware
- Review your code again!

Invasive attacks

Invasion of chips

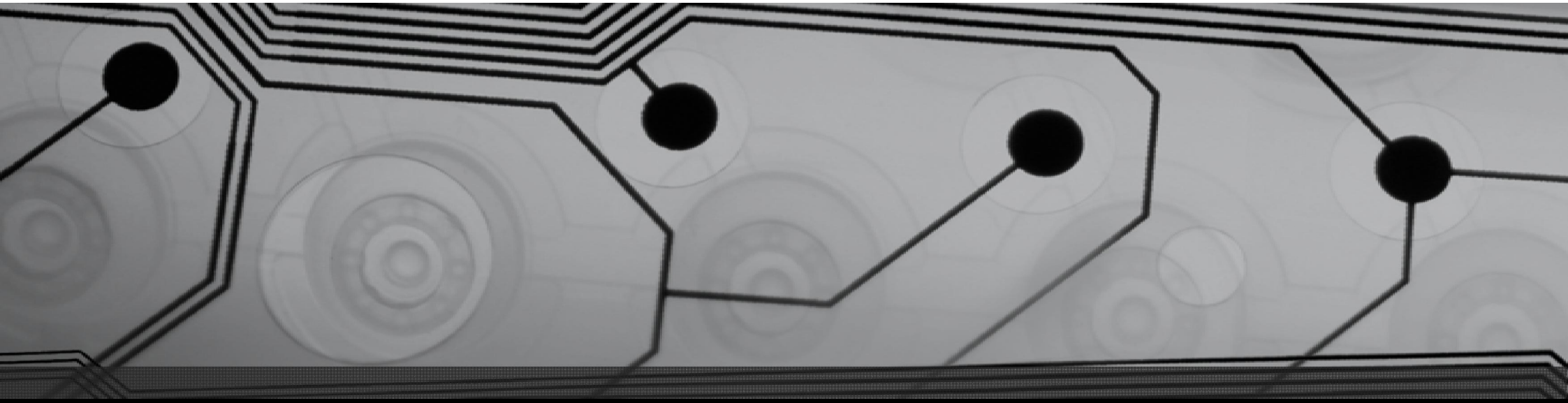
- Pretty easy to notice
- Chips will be desoldered and/or destroyed in the process
- Processors are mapped using microscopes
- Very complicated attacks
- Usually done for replication of chips

Resources

Stuff that helped me and may help you

Blogs

- http://www.devttyso.com/
- http://www.bunniestudios.com/
- http://travisgoodspeed.blogspot.com
- http://www.grandideastudio.com/



Thanks !

Questions?