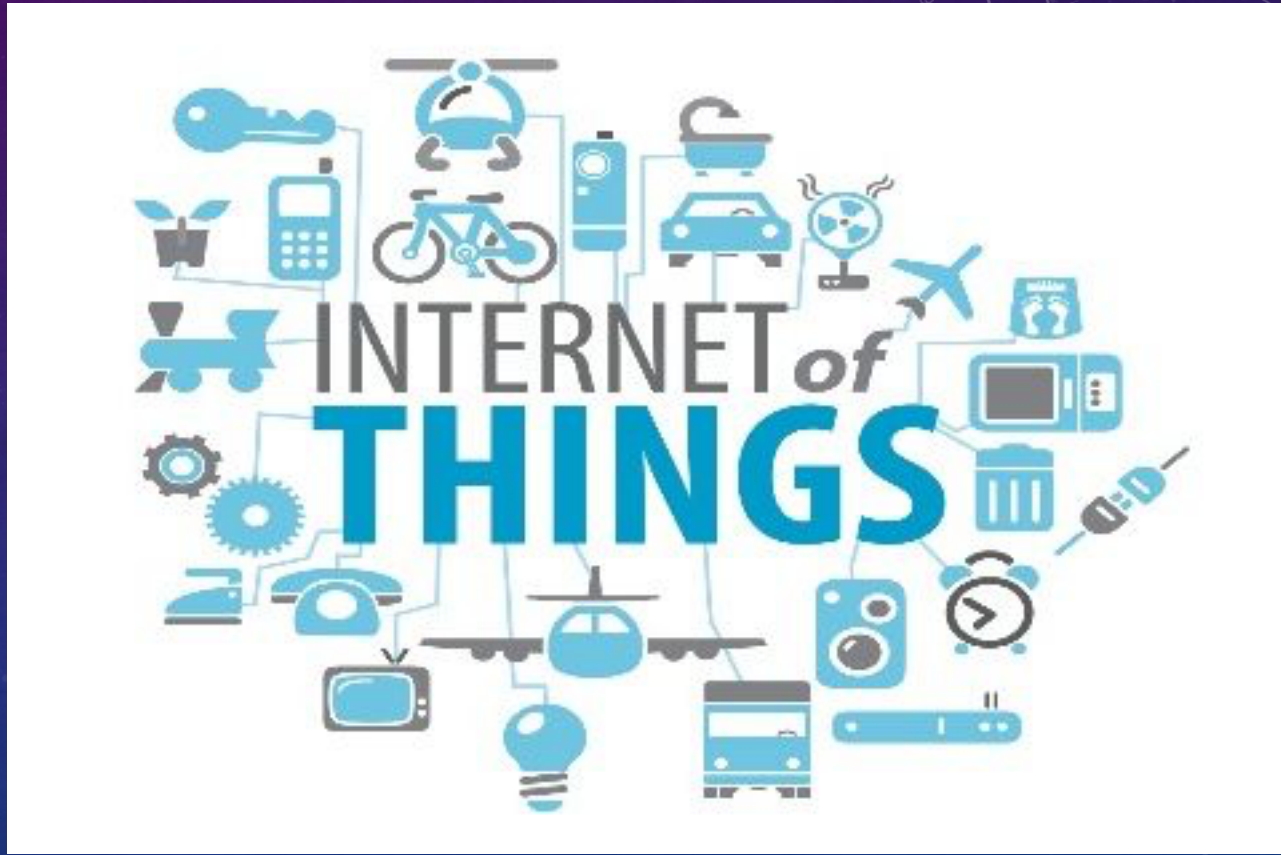
IOT SECURITY

ATUM

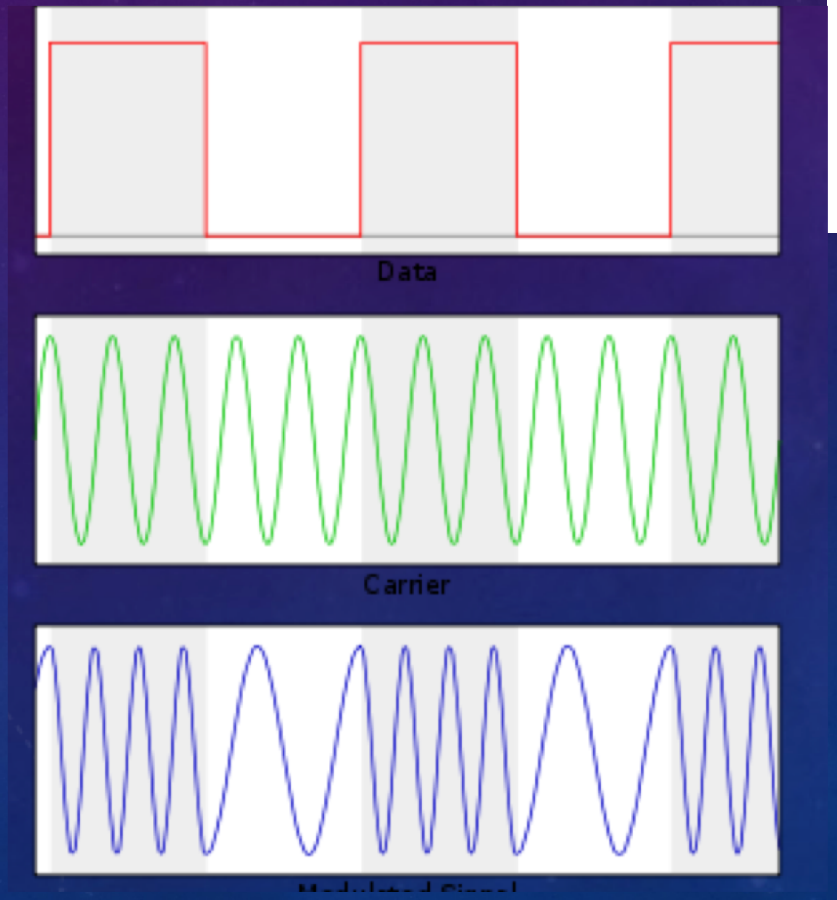
IOT

- Sensor, router, IOT printer…
- Skill Keyword：
  - Communication
  - Embedded System
  - Reverse Engineer
  - Exploitation
  - …



COMMUNICATION

- Signal sniffing/analysis/reversing
- Traffic hijacking/sniffing
- Devices:
  - HackRF/USRP/BladeRF
  - Ubertooth one
- Protocol
  - BT/BLE, ZigBee, GSM/GPRS, WIFI…
  - KNX over IP

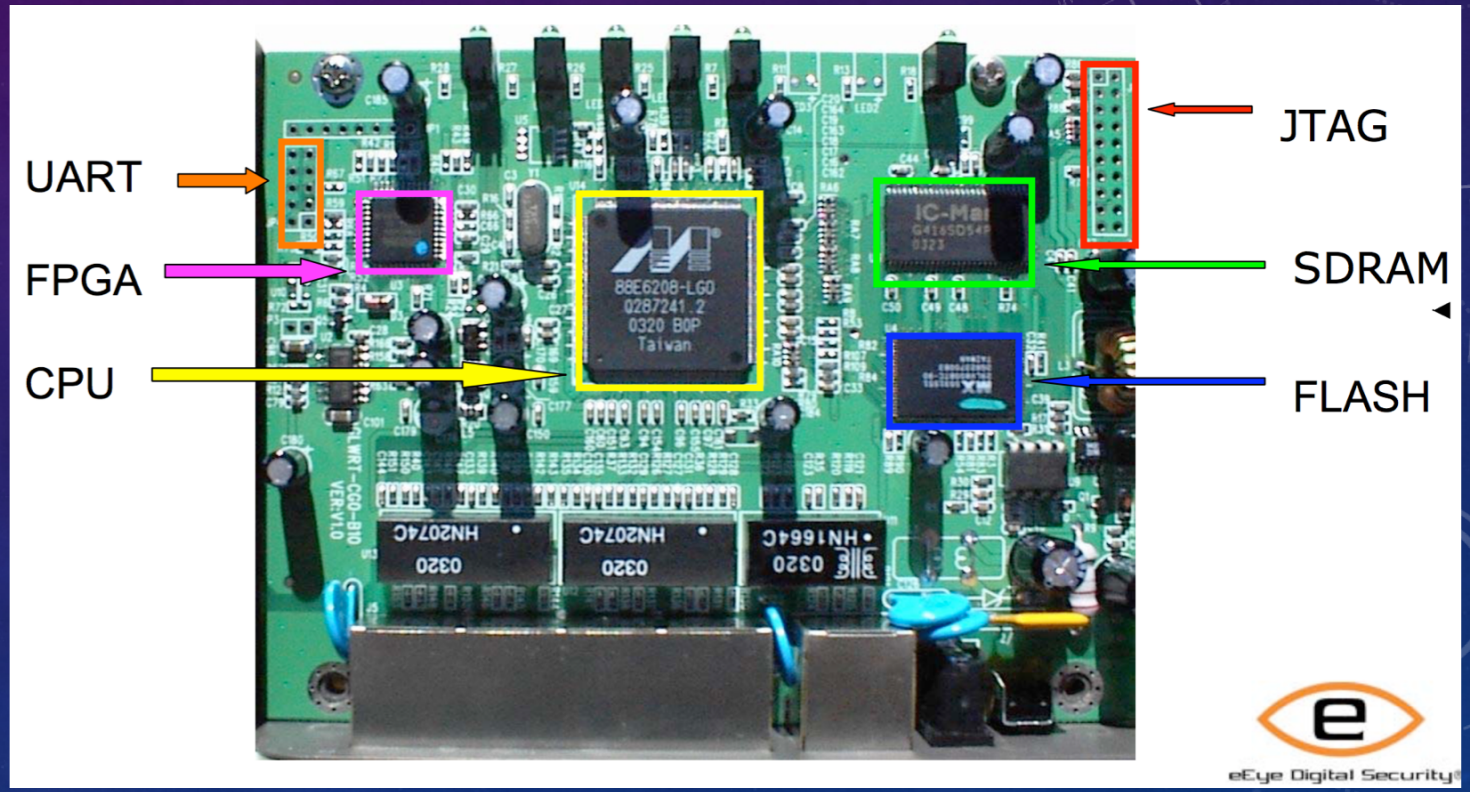


USERFUL TOOLKIT

- btlejuice(https://github.com/DigitalSecurity/btlejuice), useful BLE MITM framework
- OPENBTS(https://github.com/RangeNetworks/openbts), SDR implementation of GSM base station
- GNU Radio (https://github.com/gnuradio), SDR Toolkit

EMBEDDED SYSTEM

- Embedded SYSTEM Keyword
  - Microcontroller
  - Microprocessor
  - RAM
  - FLASH
  - JTAG/UART
- OS:
  - Uclinux, UCOS, OS-9
  - Vxworks, ThreadX , VxWorks, RTLinux
  - While(1) OS(2333)

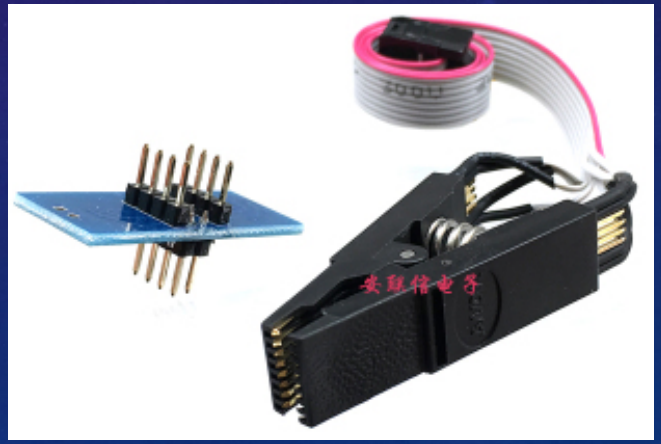


MICROCONTROLLER

- RAM+ROM+CPU in one CHIP
- Arch:
  - MIPS, **ARM,** PowerPC, X86..
- Firm:
  - STM32, 51, PIC, Freescale, MC9S12G etc..

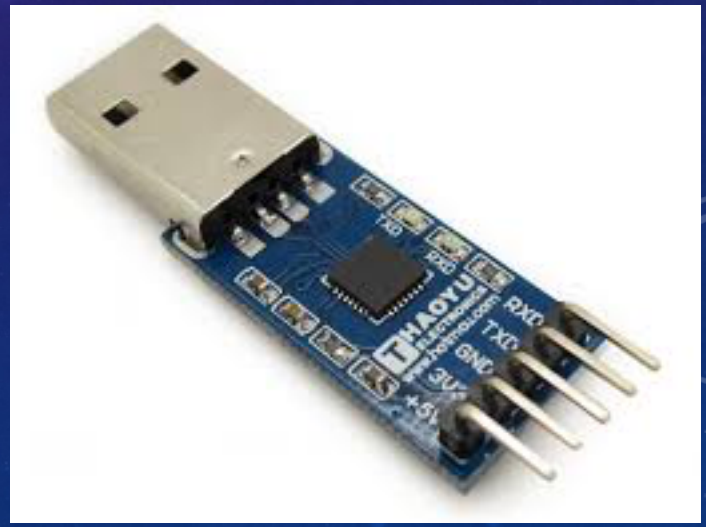
FIRMWARE DUMP

- Via Internet
  - By Manufacturer, Maybe Encrypted
- Via Sniffing
  - Sniffing Updating(USB/Network)
- Via Read Flash Directly
  - IC Test Clips + Flash Reader
  - Disordering + Flash Reader



FIRMWARE DUMP

- Via Debug interface
  - JTAG: Interface with R&W permission to ROM/RAM (20pins, 14pins)
  - UART: Depended on Developer (4-6pins)
- Via Chip decryption
  - Side Channel, Probe, etc.. Bypass Read-Protect of MCU



USERFUL REFERENCE

- http://www.taylorkillian.com/2013/01/retrieving-st-linkv2-firmware-from.html Retrieve Firmware via USB
- https://www.slideshare.net/Synack/internet-of-things-51400317 IOT Security pre on Europe black hat
- http://jcjc-dev.com/2016/04/08/reversing-huawei-router-1-find-uart/ Firmware dump

REVERSE ENGINEER

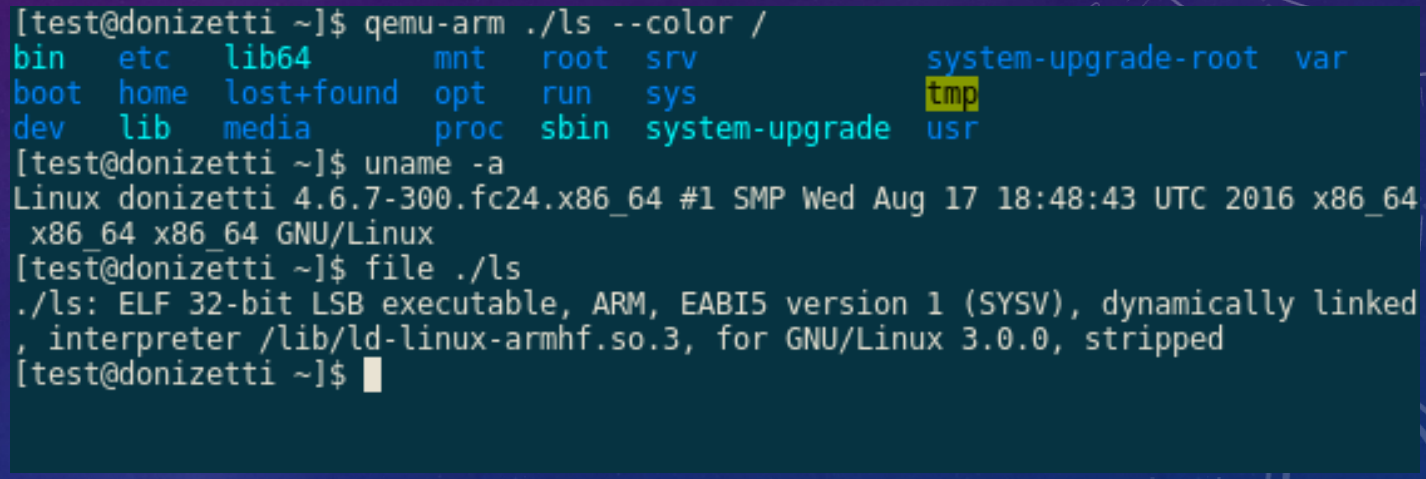
- Unpack Firmware
- Embedded Filesystem Analysis
  - SquashFS, YAFFS, JFFS2, cramfs, etc..
    - Just Binwalk it!
  - Unknown Filesystem
    - Parse Manually
  - No Filesystem
    - Put into IDAPRO

```
$ binwalk firmware.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             TRX firmware header, little endi
28            0x1C            LZMA compressed data, properties
2319004       0x23629C        Squashfs filesystem, little endi
```

REVERSE ENGINEER

- Static analysis
  - FOR MCU, Maybe more drivers code
  - No F5 on MIPS PowerPC ☹
- Dynamic analysis
  - Using Qemu/Unicorn
  - Cross Compiling & Copy to Devices
  - Using Development Board



USEFUL REFERENCES

- http://hexblog.com/files/recon%202010%20Skochinsky.pdf From PC Reverse to Embedded Reverse
- http://www.devttys0.com/ Useful Blog on Embedded System
- http://www.qemu.org QEMU
- https://github.com/MurphysChaos/VirtualFirmware Virtual Firmware
- https://retdec.com/idaplugin/ IDA MIPS F5 plugin

EXPLOITATION

- TO BE A DOC READER!!!
- SECURITY Mitigations depend on OS and Architectures
  - NO NX on MIPS
- Exploit Techniques depend on OS Architectures
  - Harvard Arch.
  - RUN on Flash & Store File on RAM?
  - RTOS PWN?

USERFUL REFERENCE

- https://security.stackexchange.com/questions/107361/arduino-buffer-overflow-and-arbitrary-code-execution MCU PWN
- https://www.slideshare.net/44Con/44con-london-attacking-vxworks-from-stone-age-to-interstellar vxworks PWN