Security in IoT Ecosystem

Module 3

lP-Camera Pentest example

Prof.: Joaquín Recas

Table of Contents

First Step

- Attack Surface Mappin
  - Finding as much **information** as possible about the **device.**
  - Focus on the following categories
    1. Embedded device.
    2. Firmware, software, and applications.
    3. Radio communications.

Sricam SP009

- General specs:
  - 1.0MP color CMOS
  - 6 IR LEDs for up to 8m
  - Two-way audio
  - Support motion detection alert
  - SD card extension (128GB FAT32)
- Technical specs:
  - Hisilicon processor (ARM9 and a high-speed video co-processor)
  - Embedded Linux OS
  - Wireless: IEEE 802.11 b/g/n (WEP, WPA, WPA2)

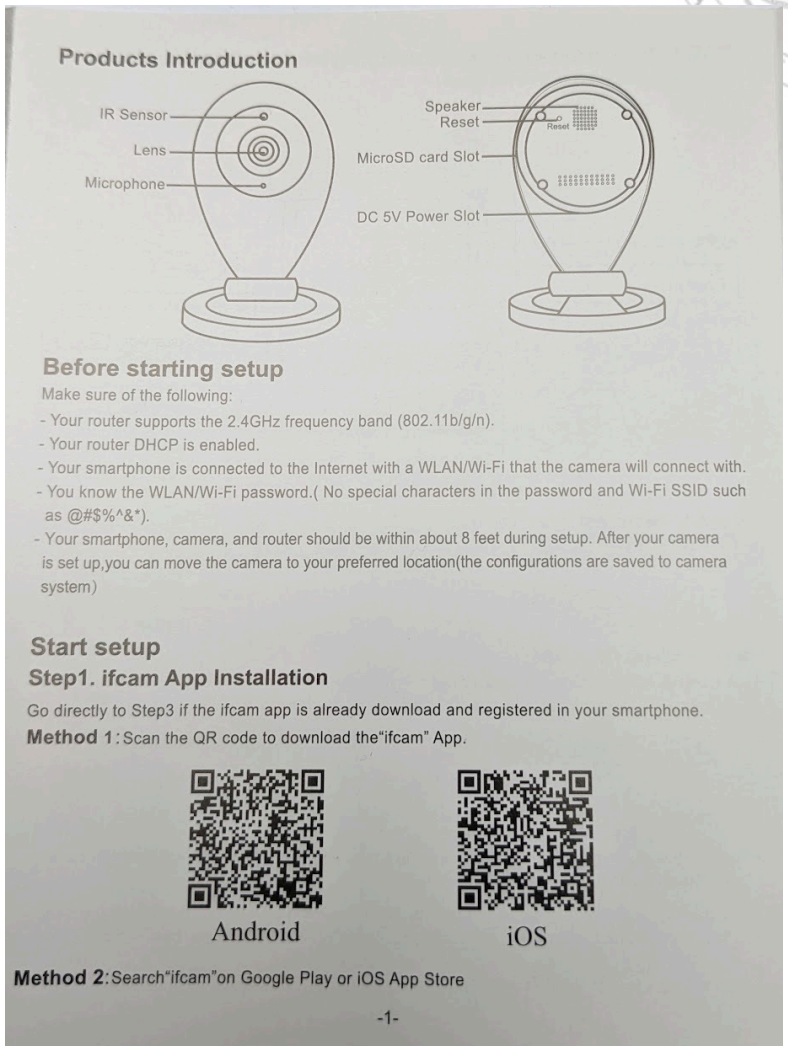
https://www.amazon.co.uk/Sricam-SP009-Wireless-Detection-Notification/dp/B015IIFR6K

Physical recognition

- Power supply 5v
- ID: 1871524
- Password: 888888
- SD card slot
- Security label
- Instructions
- Mobile App

Instructions

- Router with DHCP
  - SSID/Password
  - No special characters
- Smartphone
  - Register an user account
- Characteristics:
  - IR sensor
  - Lens
  - Microphone
  - Speaker
  - Reset
  - Micro SD card
  - 5V power supply



Products Introduction

IR Sensor
Lens
Microphone
Speaker
Reset
MicroSD card Slot
DC 5V Power Slot

Before starting setup

Make sure of the following:

- Your router supports the 2.4GHz frequency band (802.11b/g/n).
- Your router DHCP is enabled.
- Your smartphone is connected to the Internet with a WLAN/Wi-Fi that the camera will connect with.
- You know the WLAN/Wi-Fi password.(No special characters in the password and Wi-Fi SSID such as @#$%^&*).
- Your smartphone, camera, and router should be within about 8 feet during setup. After your camera is set up,you can move the camera to your preferred location(the configurations are saved to camera system)

Start setup

Step1. ifcam App Installation

Go directly to Step3 if the ifcam app is already download and registered in your smartphone.

Method 1:Scan the QR code to download the"ifcam" App.

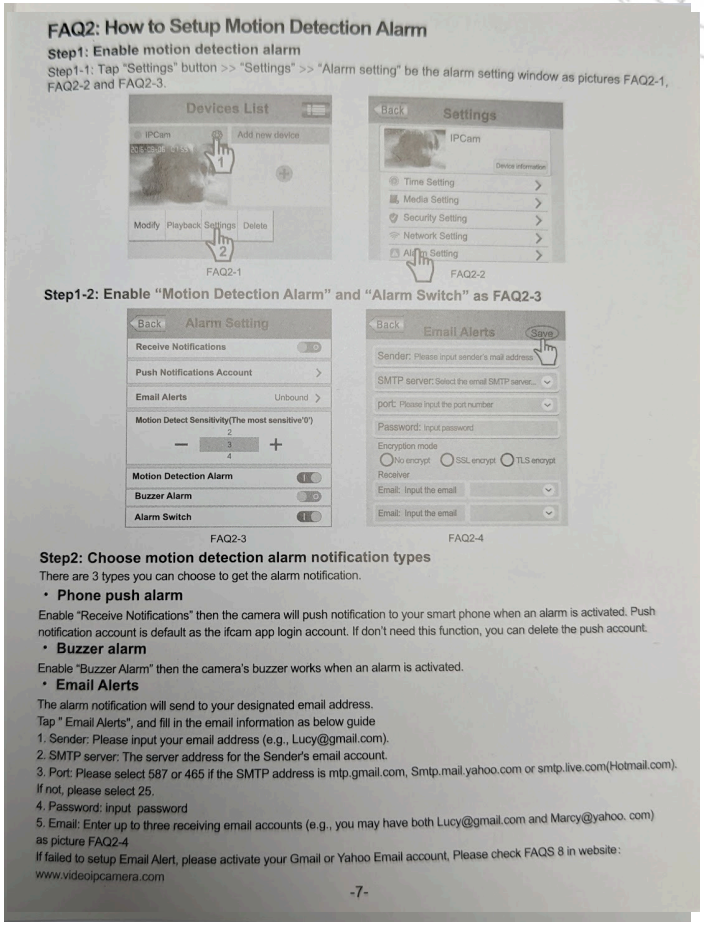
Android iOS

Method 2:Search"ifcam"on Google Play or iOS App Store

-1-

Instructions

- Record and playback
  - Micro SD card
  - FAT32
- Motion alarm
  - Configure the app
  - Push notifications by default
  - Buzzer
  - Email alerts: credentials



FAQ2: How to Setup Motion Detection Alarm

Step1: Enable motion detection alarm

Step1-1: Tap "Settings" button >> "Settings" >> "Alarm setting" be the alarm setting window as pictures FAQ2-1, FAQ2-2 and FAQ2-3.

FAQ2-1 FAQ2-2

Step1-2: Enable "Motion Detection Alarm" and "Alarm Switch" as FAQ2-3

FAQ2-3 FAQ2-4

Step2: Choose motion detection alarm notification types

There are 3 types you can choose to get the alarm notification.

- **Phone push alarm**

Enable "Receive Notifications" then the camera will push notification to your smart phone when an alarm is activated. Push notification account is default as the ifcam app login account. If don't need this function, you can delete the push account.

- **Buzzer alarm**

Enable "Buzzer Alarm" then the camera's buzzer works when an alarm is activated.

- **Email Alerts**

The alarm notification will send to your designated email address.
Tap " Email Alerts", and fill in the email information as below guide
1. Sender: Please input your email address (e.g., Lucy@gmail.com).
2. SMTP server: The server address for the Sender's email account.
3. Port: Please select 587 or 465 if the SMTP address is mtp.gmail.com, Smtp.mail.yahoo.com or smtp.live.com(Hotmail.com). If not, please select 25.
4. Password: input password
5. Email: Enter up to three receiving email accounts (e.g., you may have both Lucy@gmail.com and Marcy@yahoo. com) as picture FAQ2-4

If failed to setup Email Alert, please activate your Gmail or Yahoo Email account, Please check FAQS 8 in website: www.videoipcamera.com

-7-

Android App in PlayStore (Link)

ifcam

IP Camera manufacturer(深圳市施瑞安科技有限公司)

Tools

PEGI 3

This app is available for some of your devices

You can share this with your family. Learn more about

★★½☆☆ 319

REVIEWS

Review policy and info

2.6

★★½☆☆

319 total

5
4
3
2
1

Tharealmb

★☆☆☆☆ September 29, 2018

2

The app, since shortly, keeps running in the background. This can't be turned off. So I pressed home and the app kept running. Costing me 4GB in one night, or 50 Euros in data costs.... Thanks Ifcam!

Danny boy

★☆☆☆☆ November 6, 2019

3

I noticed the update for the app and also noticed an update available for my cameras through the app itself. Well, after i updated the app and each of my cameras, nothing works! I have a bunch of paperweights now and can't reverse the update.

APP ifcam support 13 languages : Simplified Chinese, traditional Chinese, English, Japanese, Korean, Italian, German, French, Russian, Spanish, Dutch, Portuguese, Arabic

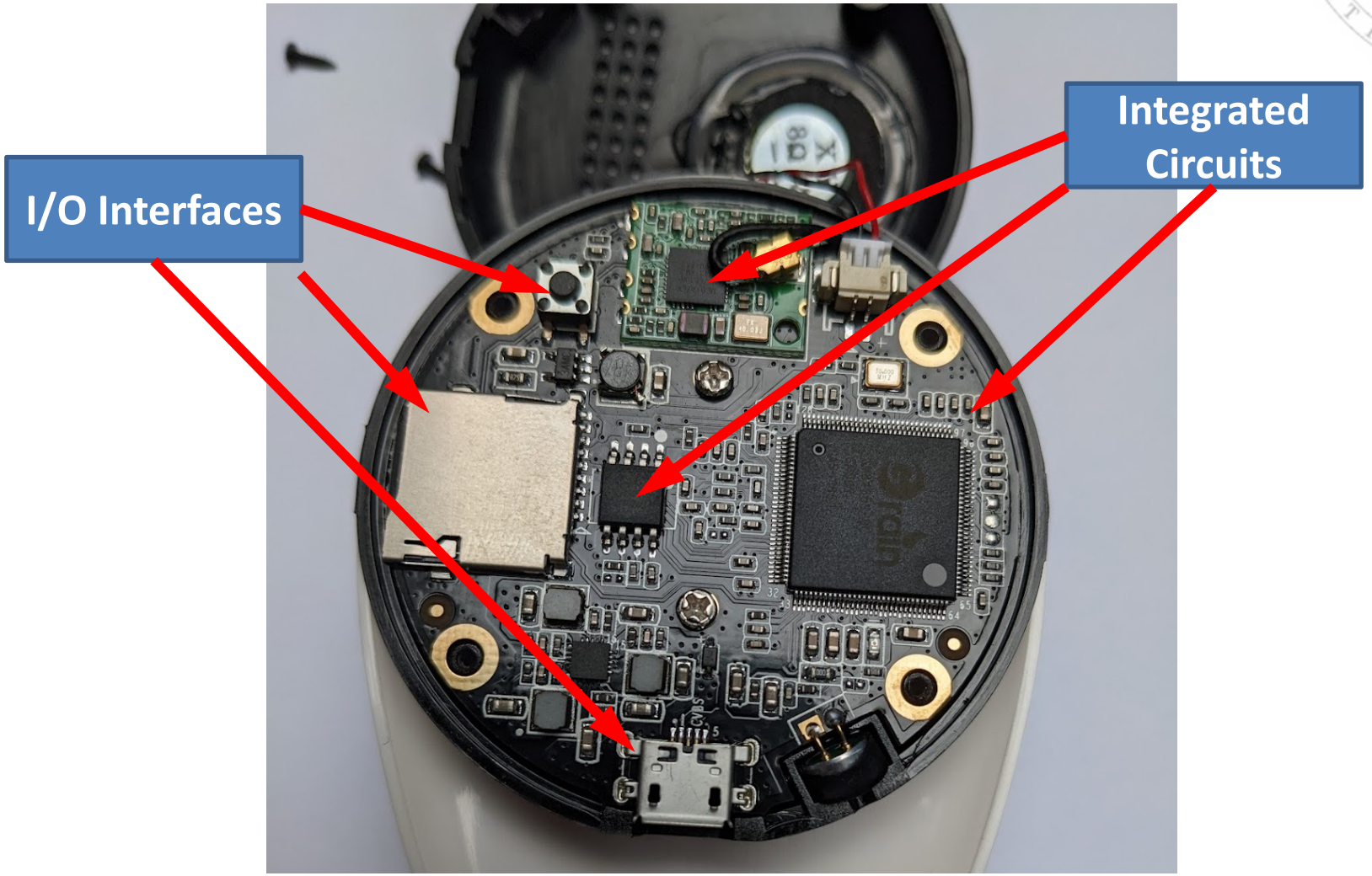
I have a bunch of paperweights now and can't reverse the update.

READ ALL REVIEWS

Embedded Device Exploitation

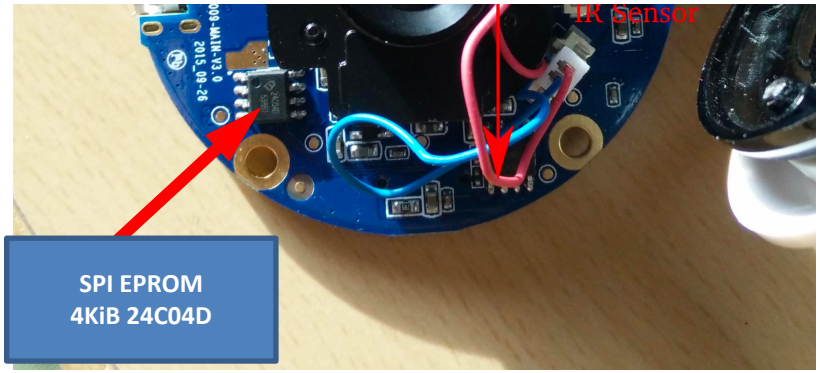
- Some of the vulnerabilities found in embedded devices:
  i. Serial ports exposed.
  ii. Insecure authentication mechanism used in the serial ports.
  iii. Ability to dump the firmware over JTAG or via Flash chips.
  iv. External media-based attacks.
  v. Power analysis and side channel-based attacks.

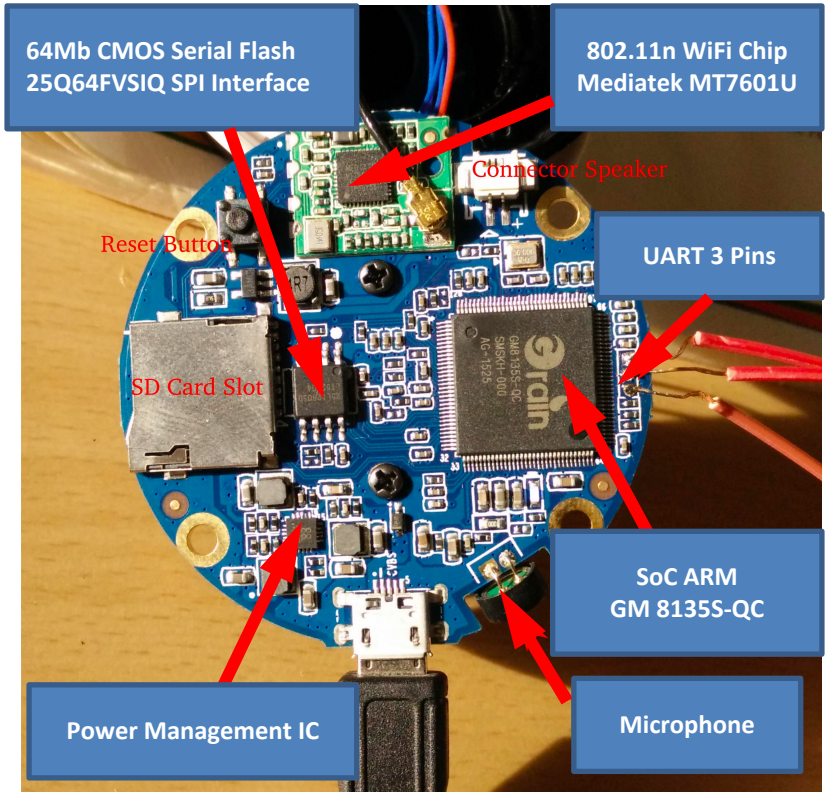
Opening the camera



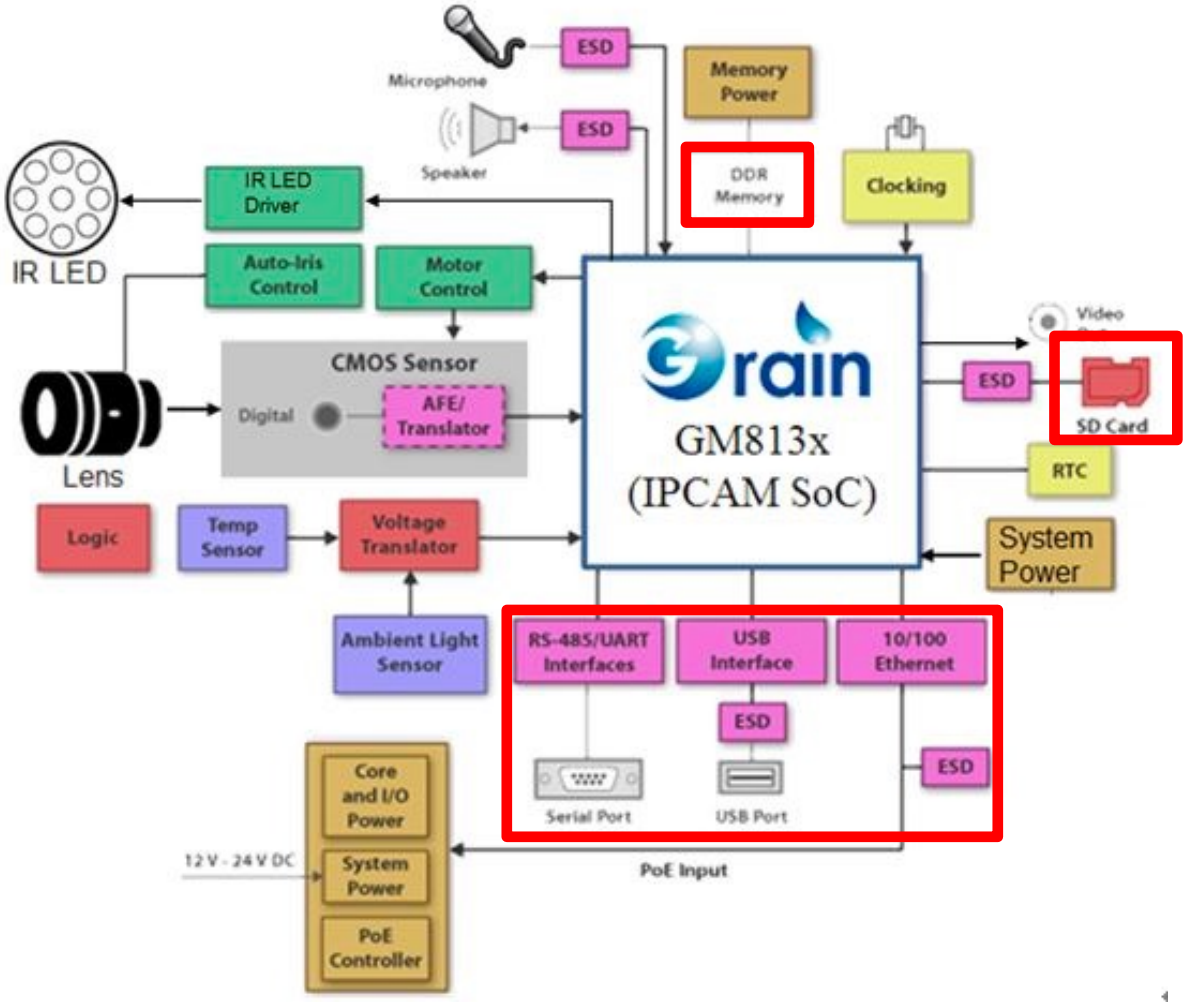
Sricam SP009

- SoC Grain Media GM8135S
- Communications:
  - USB
  - UART 3 pines
  - WiFi Mediatek MT7601U
- External memory
  - Winbond Serial Flash 25Q64FVSIQ
  - EEPROM ABLIC 24C04D
  - Micro SD card reader



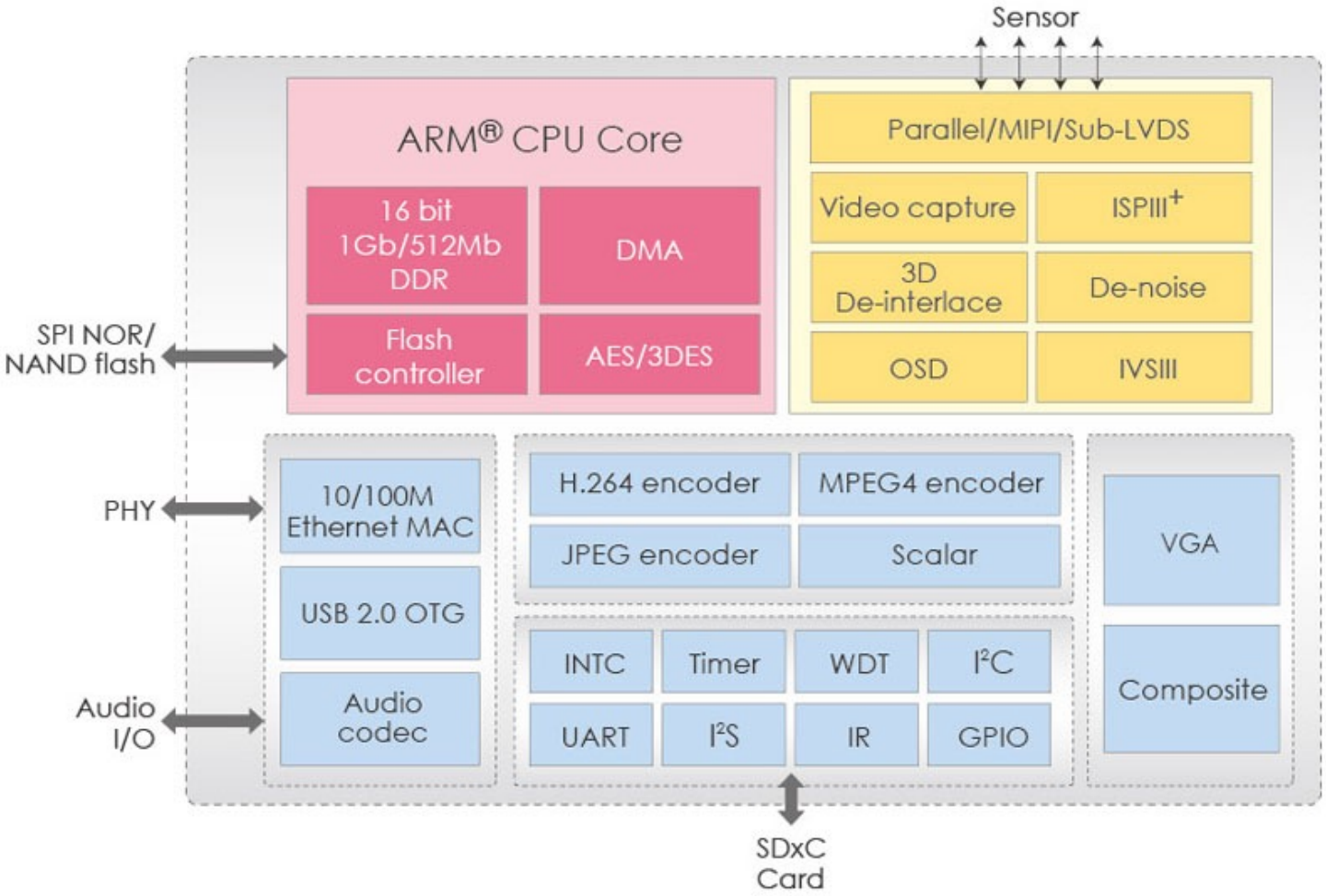


Sricam SP009



[SoC Video]

GM8135S SoC Block Diagram



https://datasheetspdf.com/datasheet/GM8135S.html

Feature List

Embedded Processor	ARM® CPU core - 32-bit RISC, up to 600MHz
Memory Interface	• GM8136S: SiP 1Gb DDR GM8135S: SiP 512Mb DDR Support SPI NOR/NAND flash boot
H.264 Codec	• Support the high profile encoding - GM8136S: 720P@60fps - GM8135S: 720P@45fps
Video Input	• MIPI, Parallel, Sub-LVDS - YUV x 2 for 2-channel 720P
Video Capture	• BT.656 input (up to 148.5 MHz) • BT.1120 input (up to 148.5 MHz) • Support 54MHz/108MHz and 72MHz/144MHz byte/frame interleave modes
Image Signal Processing	• Temporal and spatial 3D-Denoise filtering • Support Digital-WDR • Support WDR sensor • Support tone mapping • Low lux sensitivity to ISO6400 • Embedded IVS engine
Display Interface	• Built-in BT.1120 digital output • Support composite/parallel up to FHD output
Peripherals Interface	10/100M Ethernet MAC USB 2.0 OTG USB 1.1 Device SDIO x 2 I²S I²C GPIO
Operating Voltage	• Core: 1.1 V • DDR: 1.5/1.8 V • I/O: 3.3 V
Package	• 128-pin EPADLQFP

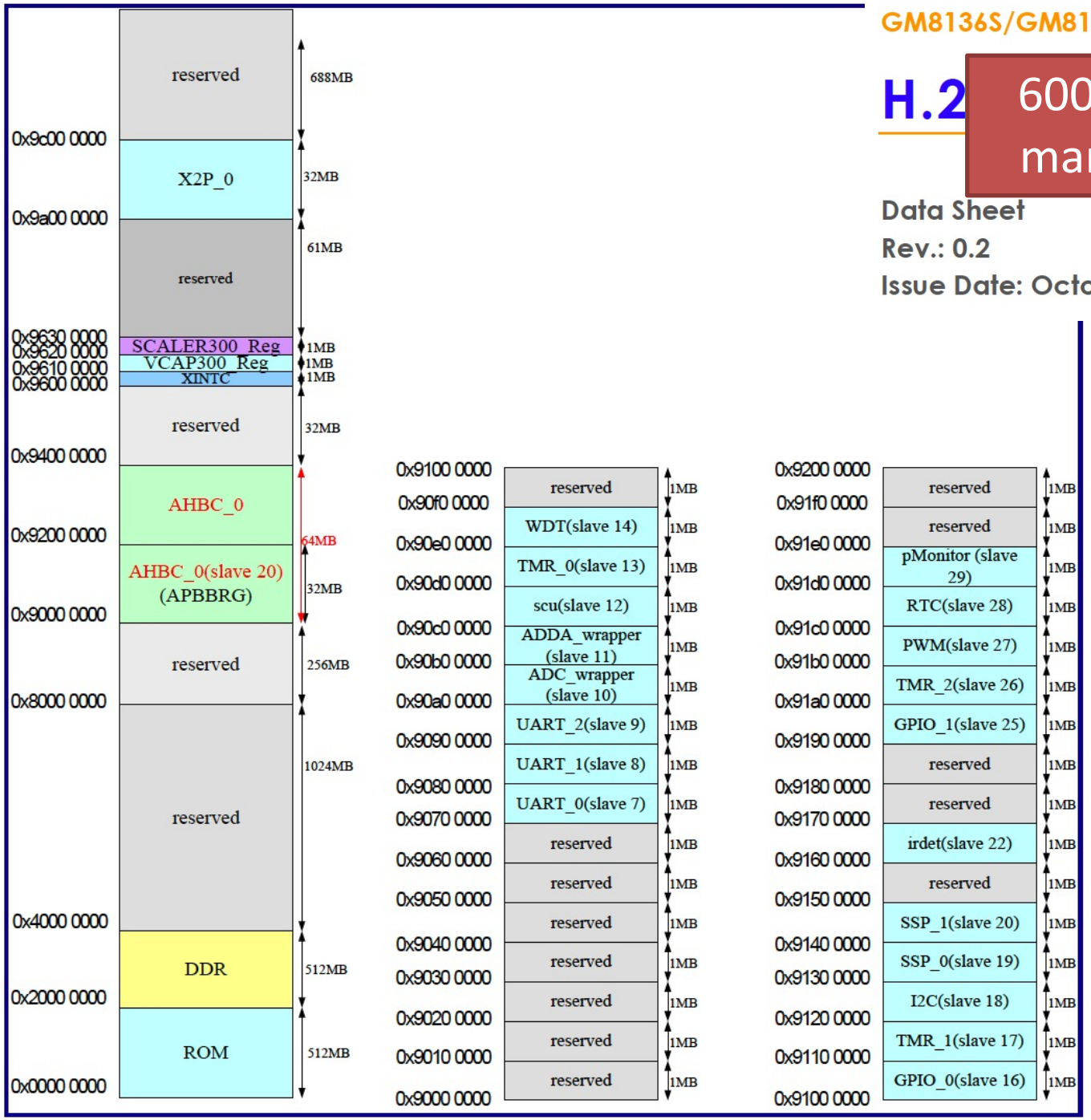
http://www.grain-media.com/html/8136S_8135S.htm

GM8136S/GM8135S

H.2 SoC

600 pages manual!!!

Data Sheet
Rev.: 0.2
Issue Date: October 2014

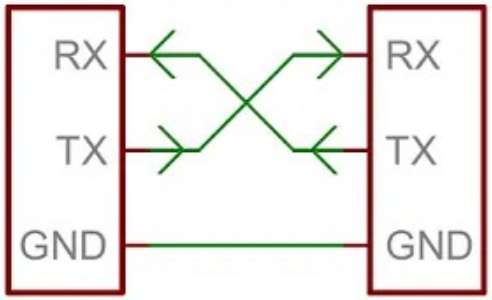


Attack Surface Mappin

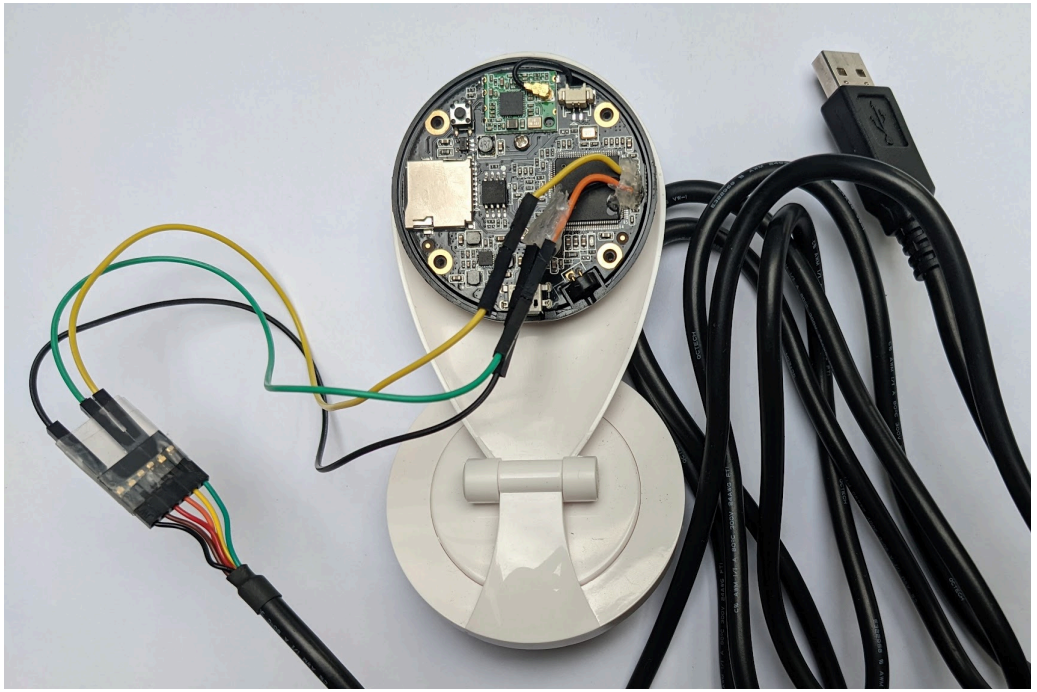
- We know enough to create an attack surface map:
  - We have identified the components present in the IP camera
  - We have access to the mobile app for reverse engineering
  - We know the communication protocols
  - We have access to the hardware for firmware extraction
- Identify attack vectors for each component:
  - We will try to extract and analyse the firmware.

Serial pins (UART)

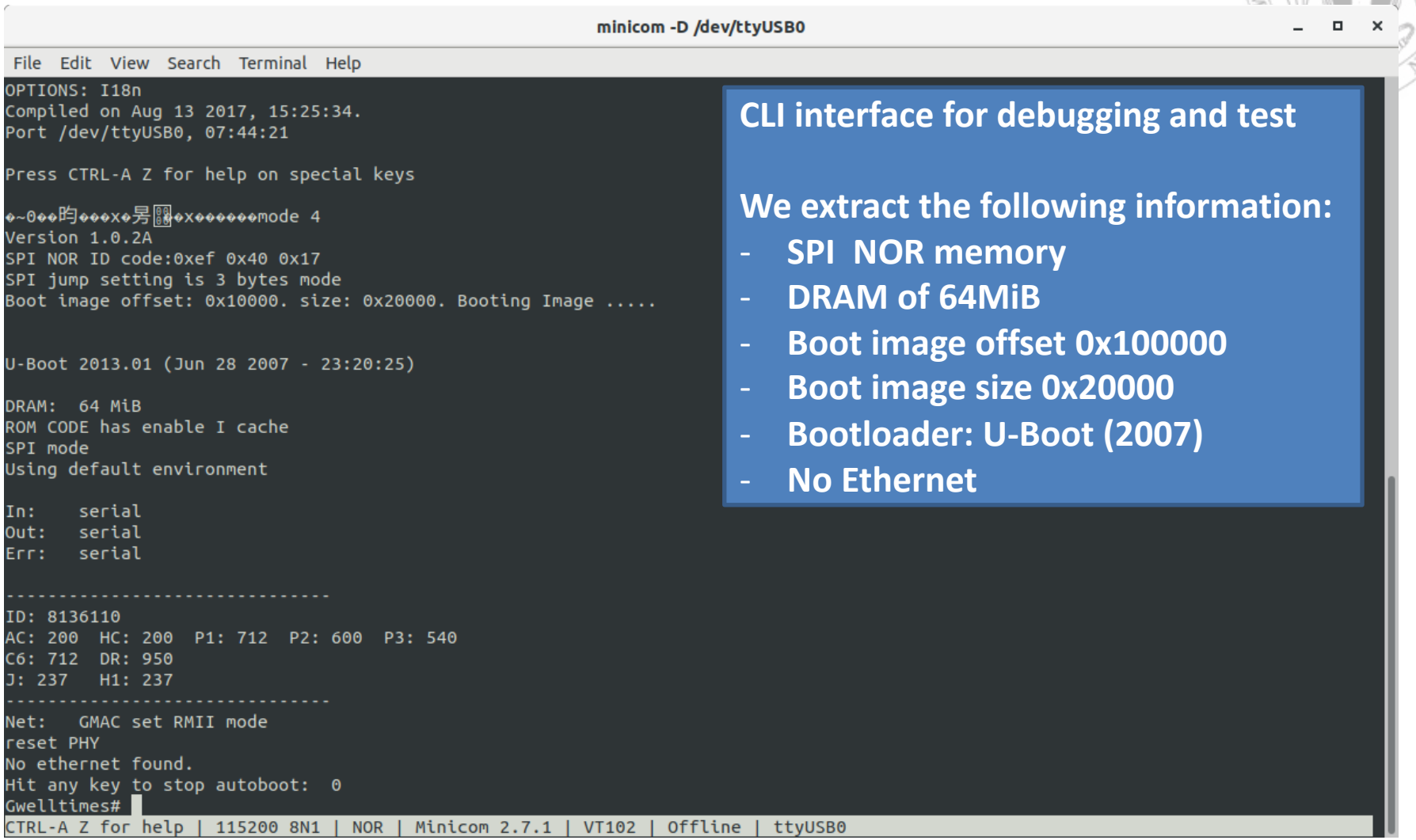
- We can solder 3 (TX, RX, gnd) wires on the exposed pins to see if they give access to a serial console



- We need a TTL to USB converter
- Voltage?:
  - 3.3V
  - 5V



Serial console



CLI interface for debugging and test

We extract the following information:

- **SPI NOR memory**
- **DRAM of 64MiB**
- **Boot image offset 0x100000**
- **Boot image size 0x20000**
- **Bootloader: U-Boot (2007)**
- **No Ethernet**

Das U-Boot

- The "Universal Bootloader" (**"Das U-Boot")** is a monitor program
- Free Software: full source code under GPL
- Hosted on SourceForge: http://sourceforge.net/projects/u-boot
- Production quality: used as default boot loader by several board vendors
- Portable and easy to port and to debug
- Many supported architectures:
  - PPC, ARM, MIPS, x86, m68k, NIOS, Microblaze
- More than 216 boards supported by public source tree
- many, many features

https://www.denx.de/wiki/U-Boot/Documentation

Video

Das U-Boot: help

```
minicom -D /dev/ttyUSB0
File  Edit  View  Search  Terminal  Help
Net:   GMAC set RMII mode
reset PHY
No ethernet found.
Hit any key to stop autoboot:  0
Gwelltimes# help
?       - alias for 'help'
boot    - boot default, i.e., run 'bootcmd'
bootd   - boot default, i.e., run 'bootcmd'
bootm   - boot application image from memory
bootp   - boot image via network using BOOTP/TFTP protocol
cmp     - memory compare
cp      - memory copy
crc32   - checksum calculation
env     - environment handling commands
erase   - erase FLASH memory
flinfo  - print FLASH memory information
go      - start application at address 'addr'
help    - print command description/usage
l2cache_test- Perform test of L2 cache
md      - memory display
memtester- memory tester
mm      - memory modify (auto-incrementing address)
mtest   - simple RAM read/write test
mw      - memory write (fill)
nm      - memory modify (constant address)
printenv- print environment variables
protect - enable or disable FLASH write protection
reset   - Perform RESET of the CPU
run     - run commands in an environment variable
setenv  - set environment variables
sf      - SPI flash sub-system
sspi    - SPI utility command
tftpboot- boot image via network using TFTP protocol
version - print monitor, compiler and linker version
Gwelltimes#
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
```

Controls the boot sequence:

- **Locate the firmware**
- **Load it**
- **Run it**

We can extract it for reverse engineering

Das U-Boot: printenv

minicom -D /dev/ttyUSB0

File Edit View Search Terminal Help

```
Gwelltimes# printenv
baudrate=115200
bootcmd=sf probe 0:0;run lm;bootm 0x2000000
bootdelay=1
cmd1=mem=64M gmmem=30M console=ttyS0,115200 user_debug=31 init=/squashfs_init root=/dev/mtdblock2 rootfstype=squashfs
cmd2=mem=128M gmmem=90M console=ttyS0,115200 user_debug=31 init=/squashfs_init root=/dev/mtdblock2 rootfstype=squashfs
cmd3=mem=256M gmmem=190M console=ttyS0,115200 user_debug=31 init=/squashfs_init root=/dev/mtdblock2 rootfstype=squashfs
cmd4=mem=512M gmmem=432M console=ttyS0,115200 user_debug=31 init=/squashfs_init root=/dev/mtdblock2 rootfstype=squashfs
lm=sf read 0x02000000 z
stderr=serial
stdin=serial
stdout=serial

Environment size: 637/65532 bytes
Gwelltimes#
```

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | tty

Printenv displays the values of environment variables:

- **Baudrate**
- **Bootcmd**
- **Several cmds (squashfs)**
- **Redirections**

Serial console: Embedded Linux

minicom -D /dev/ttyUSB0

File Edit View Search Terminal Help

```
[MSC] set chl, chan_id:1, width:2, chan_flags:0x0
80211> cfg80211_scan_done ==> NULL

drwxr-xr-x   21 1015     1015          279 Jun 30  2007 .
drwxr-xr-x   21 1015     1015          279 Jun 30  2007 ..
drwxr-xr-x    2 root     root          860 Jan  1  1970 bin
-rwxrwxrwx    1 1015     1015         5383 Jun 30  2007 boot.sh
drwxr-xr-x    5 root     root         1600 Jan  1 01:01 dev
drwxr-xr-x    7 root     root         1024 Jan  1 01:03 etc
drwxrwxrwx    5 1015     1015           54 Jun 30  2007 gm
drwxrwxrwx    2 1015     1015            3 Jun 30  2007 home
-rwxrwxrwx    1 1015     1015          371 Jun 30  2007 init
drwxrwxrwx    3 1015     1015          612 Jun 30  2007 lib
drwxrwxrwx    9 1015     1015           99 Jun 30  2007 mnt
drwxr-xr-x    9 root     root            0 Jan  1  1970 npc
drwxrwxrwx    2 1015     1015            3 Jun 30  2007 opt
drwxrwxrwx    4 1015     1015           37 Jun 30  2007 patch
dr-xr-xr-x  116 root     root            0 Jan  1  1970 proc
drwxr-xr-x    3 root     root            0 Jan  1  1970 rom
drwxrwxrwx    2 1015     1015            3 Jun 30  2007 root
drwxr-xr-x    2 root     root          500 Jan  1  1970 sbin
drwxrwxrwx    2 1015     1015            3 Jun 30  2007 share
-rwxrwxrwx    1 1015     1015          946 Jun 30  2007 squashfs_init
drwxr-xr-[MSC] msc_cmd_handler result 7
x   11 root     [MSC] AM=0, ssid=, pwd=, user=, cust_data_len=0, cust_data=,
root             0 Jan  1  1970 sys
drwxr-xr-x    7 root     root         1024 Jan  1 01:03 tmp
drwxr-xr-x    4 root     root           80 Jan  1  1970 usr
drwxr-xr-x    4 root     root           80 Jan  1  1970 var
/ # Sending discover...
[MSC] set chl, chan_id:6, width:2, chan_flags:0x0
80211> cfg80211_scan_done ==> NULL
[MSC] set chl, chan_id:11, width:8, chan_flags:0x0
80211> cfg80211_scan_done ==> NULL
```

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0

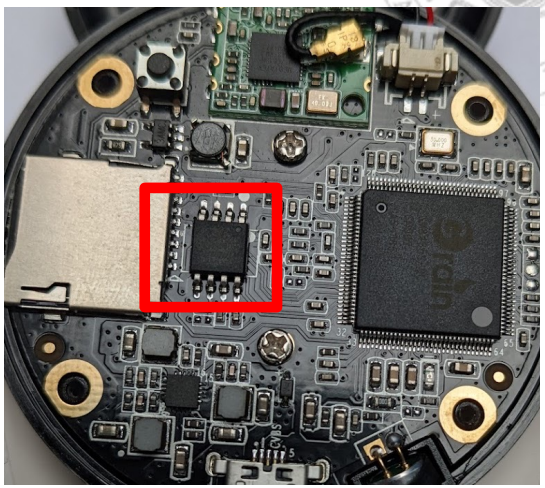
Root access to the Operative System!!

Firmware Extraction

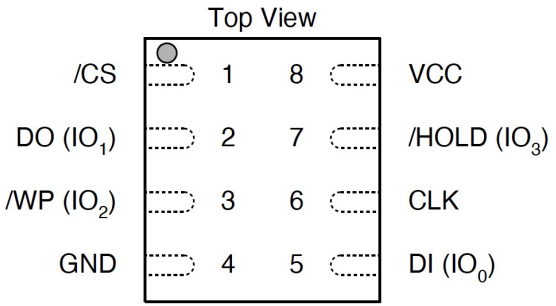
- We can extract the firmware for analysis using:
  - Reading it from the SPI NOR memory
  - U-Boot console
  - Linux console
  - From the vendor web page?

W25Q64FV

- **Family of SpiFlash Memories**
  - W25Q64FV: 64M-bit / 8M-byte (8,388,608)
  - Standard SPI: CLK, /CS, DI, DO, /WP, /Hold
  - Dual SPI: CLK, /CS, IO_0, IO_1, /WP, /Hold
  - Quad SPI: CLK, /CS, IO_0, IO_1, IO_2, IO_3
  - QPI: CLK, /CS, IO_0, IO_1, IO_2, IO_3
- **Highest Performance Serial Flash**
  - 104MHz Standard/Dual/Quad SPI clocks
  - 208/416MHz equivalent Dual/Quad SPI
  - 50MB/S continuous data transfer rate
  - More than 100,000 erase/program cycles
  - More than 20-year data retention
- **Efficient "Continuous Read" and QPI Mode**
  - Continuous Read with 8/16/32/64-Byte Wrap
  - As few as 8 clocks to address memory
  - Quad Peripheral Interface (QPI) reduces instruction overhead
  - Allows true XIP (execute in place) operation
  - Outperforms X16 Parallel Flash
- **Low Power, Wide Temperature Range**
  - Single 2.7 to 3.6V supply
  - 4mA active current, <1μA Power-down (typ.)
  - -40°C to +85°C operating range
- **Flexible Architecture with 4KB sectors**
  - Uniform Sector Erase (4K-bytes)
  - Uniform Block Erase (32K and 64K-bytes)
  - Program 1 to 256 byte per programmable page
  - Erase/Program Suspend & Resume
- **Advanced Security Features**
  - Software and Hardware Write-Protect
  - Top/Bottom, 4KB complement array protection
  - Power Supply Lock-Down and OTP protection
  - 64-Bit Unique ID for each device
  - Discoverable Parameters (SFDP) Register
  - 3X256-Bytes Security Registers with OTP locks
  - Volatile & Non-volatile Status Register Bits
- **Space Efficient Packaging**
  - 8-pin SOIC/VSOP 208-mil
  - 8-pad WSON 6x5-mm/8x6-mm
  - 16-pin SOIC 300-mil
  - 8-pin PDIP 300-mil
  - 24-ball TFBGA 8x6-mm
  - Contact Winbond for KGD and other options



Top View



3.4 Pin Description SOIC/VSOP 208-mil, WSON 6x5/8x6-mm and PDIP 300-mil

PIN NO.	PIN NAME	I/O	FUNCTION
1	/CS	I	Chip Select Input
2	DO (IO1)	I/O	Data Output (Data Input Output 1)*1
3	/WP (IO2)	I/O	Write Protect Input (Data Input Output 2)*2
4	GND		Ground
5	DI (IO0)	I/O	Data Input (Data Input Output 0)*1
6	CLK	I	Serial Clock Input
7	/HOLD (IO3)	I/O	Hold Input (Data Input Output 3)*2
8	VCC		Power Supply

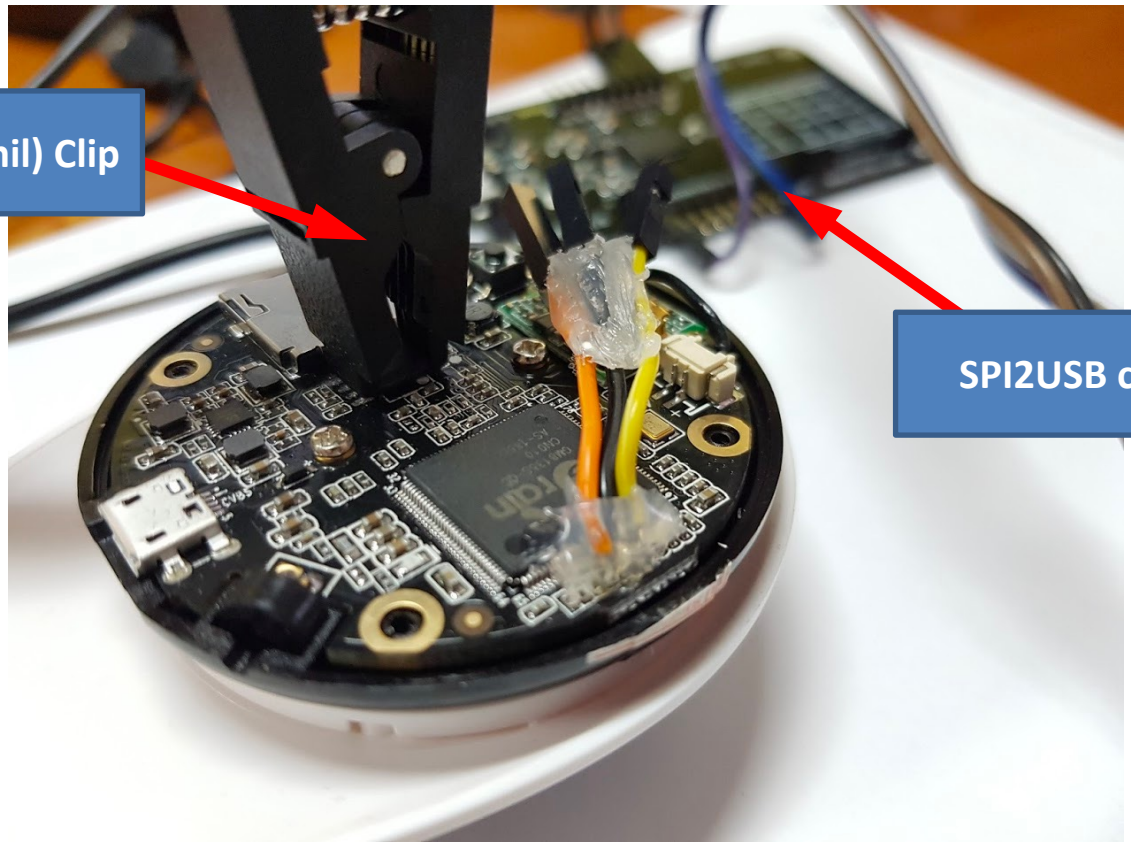
*1 IO0 and IO1 are used for Standard and Dual SPI instructions

*2 IO0 – IO3 are used for Quad SPI instructions

[Datasheet]

Firmware: Flash

- SPI to USB clip IC (W25Q64FV):



Firmware: Flash

```
File  Edit  View  Search  Terminal  Help
→ examples git:(master) ✗ python spiflash.py -s 15000000 -r ipcam_firmware
FT232H Future Technology Devices International, Ltd initialized at 15000000 hertz
Reading 15000000 bytes starting at address 0x0...saved to ipcam_firmware.
→ examples git:(master) ✗ hexdump -n 512 -C ipcam_firmware
00000000  47 4d 38 31 33 36 00 00  00 00 01 00 00 00 02 00  |GM8136..........|
00000010  00 00 00 00 00 00 01 00  00 00 02 00 55 2d 42 4f  |............U-BO|
00000020  4f 54 00 00 00 00 00 00  00 00 03 00 00 00 1f 00  |OT..............|
00000030  4b 45 52 4e 45 4c 00 00  00 00 00 00 00 00 22 00  |KERNEL........".|
00000040  00 00 32 00 52 4f 4f 54  46 53 00 00 00 00 00 00  |..2.ROOTFS......|
00000050  00 00 54 00 00 00 05 00  52 4f 4d 00 00 00 00 00  |..T.....ROM.....|
00000060  00 00 00 00 00 00 59 00  00 00 27 00 41 50 50 00  |......Y...'.APP.|
00000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000000d0  00 00 00 00 00 00 00 00  00 00 00 00 08 00 00 00  |................|
000000e0  0c 00 00 00 17 00 00 00  08 00 00 00 00 00 00 00  |................|
000000f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00000100  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
*
00000200
→ examples git:(master) ✗
```

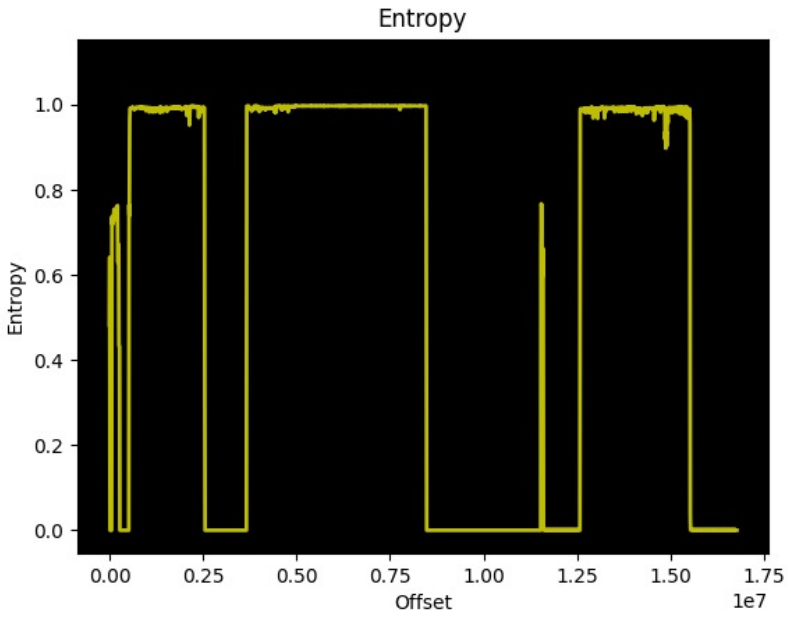
Binwalk

- Binwalk is a fast, easy to use tool for analysing, reverse engineering, and extracting firmware images.

```
iot@attifyos: ~/Desktop
File  Edit  View  Search  Terminal  Help
→ Desktop binwalk -e ip_cam_attify.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
217452        0x3516C         CRC32 polynomial table, little endian
524288        0x80000         uImage header, header size: 64 bytes, header CRC: 0xFE93070D, cr
eated: 2005-07-24 11:35:15, image size: 2025296 bytes, Data Address: 0x2000000, Entry Point: 0
x2000040, data CRC: 0xEF715CB9, OS: Linux, CPU: ARM, image type: OS Kernel Image, compression
type: none, image name: "gm8136"
524352        0x80040         Linux kernel ARM boot executable zImage (little-endian)
542452        0x846F4         gzip compressed data, maximum compression, from Unix, last modif
ied: 1970-01-01 00:00:00 (null date)
3670016       0x380000        Squashfs filesystem, little endian, version 4.0, compression:xz,
 size: 4803481 bytes, 177 inodes, blocksize: 131072 bytes, created: 2005-08-05 07:41:04
11534956      0xB0026C        Zlib compressed data, compressed
11536772      0xB00984        Zlib compressed data, compressed
11539856      0xB01590        Zlib compressed data, compressed
11541576      0xB01C48        Zlib compressed data, compressed
11541896      0xB01D88        Zlib compressed data, compressed
11542184      0xB01EA8        Zlib compressed data, compressed
11542472      0xB01FC8        Zlib compressed data, compressed
11542760      0xB020E8        Zlib compressed data, compressed
11543048      0xB02208        Zlib compressed data, compressed
11543336      0xB02328        Zlib compressed data, compressed
11543624      0xB02448        Zlib compressed data, compressed
11543912      0xB02568        Zlib compressed data, compressed
11544200      0xB02688        Zlib compressed data, compressed
11544388      0xB02744        Zlib compressed data, compressed
11544676      0xB02864        Zlib compressed data, compressed
11544964      0xB02984        Zlib compressed data, compressed
11545252      0xB02AA4        Zlib compressed data, compressed
```

Binwalk: entropy

- The entropy will be high when the bytes in the image look random, the image is
  - Encrypted,
  - Compressed or
  - obfuscated file

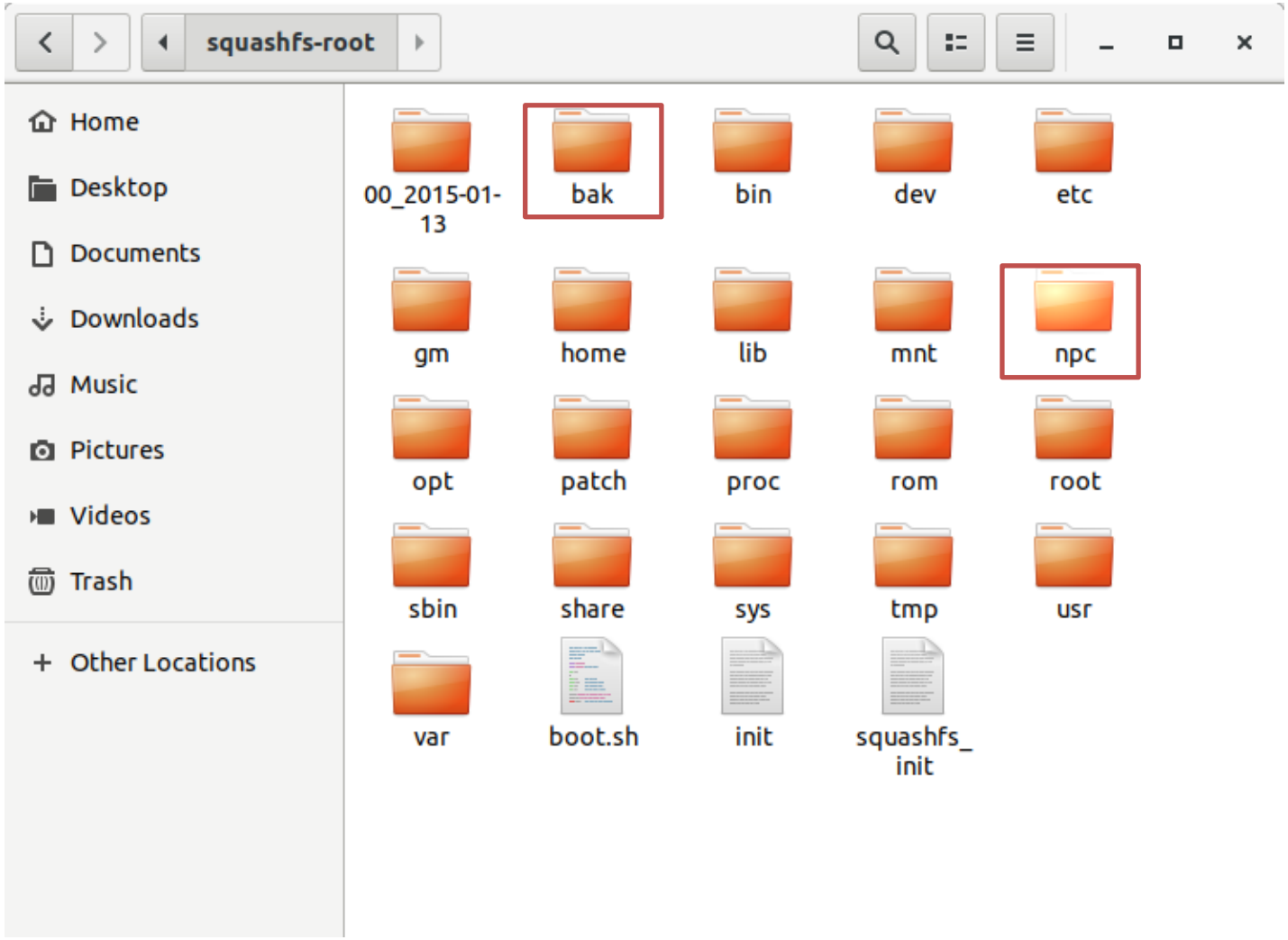



```
ubuntu@ubuntu2004: ~/Security/IPCAM
ubuntu@ubuntu2004:~/Security/IPCAM$ binwalk -E ip_cam_attify.bin

DECIMAL       HEXADECIMAL     ENTROPY
--------------------------------------------------------------------------------
0             0x0             Falling entropy edge (0.482174)
540672        0x84000         Rising entropy edge (0.961417)
2547712       0x26E000        Falling entropy edge (0.326261)
3670016       0x380000        Rising entropy edge (0.996336)
8470528       0x814000        Falling entropy edge (0.524907)
12582912      0xC00000        Rising entropy edge (0.987825)
14909440      0xE38000        Rising entropy edge (0.964547)
15523840      0xECE000        Falling entropy edge (0.036562)
```

Binwalk: extract filesystem

```
ubuntu@ubuntu2004: ~/Security/IPCAM
ubuntu@ubuntu2004:~/Security/IPCAM$ binwalk -e ip_cam_attify.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
217452        0x3516C         CRC32 polynomial table, little endian
524288        0x80000         uImage header, header size: 64 bytes, header CRC: 0xF
E93070D, created: 2005-07-24 11:35:15, image size: 2025296 bytes, Data Address: 0x2
000000, Entry Point: 0x2000040, data CRC: 0xEF715CB9, OS: Linux, CPU: ARM, image ty
pe: OS Kernel Image, compression type: none, image name: "gm8136"
524352        0x80040         Linux kernel ARM boot executable zImage (little-endia
n)
542452        0x846F4         gzip compressed data, maximum compression, from Unix,
 last modified: 1970-01-01 00:00:00 (null date)
3670016       0x380000        Squashfs filesystem, little endian, version 4.0, comp
ression:xz, size: 4803481 bytes, 177 inodes, blocksize: 131072 bytes, created: 2005
-08-05 07:41:04
11534956      0xB0026C        Zlib compressed data, compressed
11536772      0xB00984        Zlib compressed data, compressed
11539856      0xB01590        Zlib compressed data, compressed
11541576      0xB01C48        Zlib compressed data, compressed
11541896      0xB01D88        Zlib compressed data, compressed
11542184      0xB01EA8        Zlib compressed data, compressed
11542472      0xB01FC8        Zlib compressed data, compressed
11542760      0xB020E8        Zlib compressed data, compressed
11543048      0xB02208        Zlib compressed data, compressed
11543336      0xB02328        Zlib compressed data, compressed
11543624      0xB02448        Zlib compressed data, compressed
```

Binwalk: filesystem review



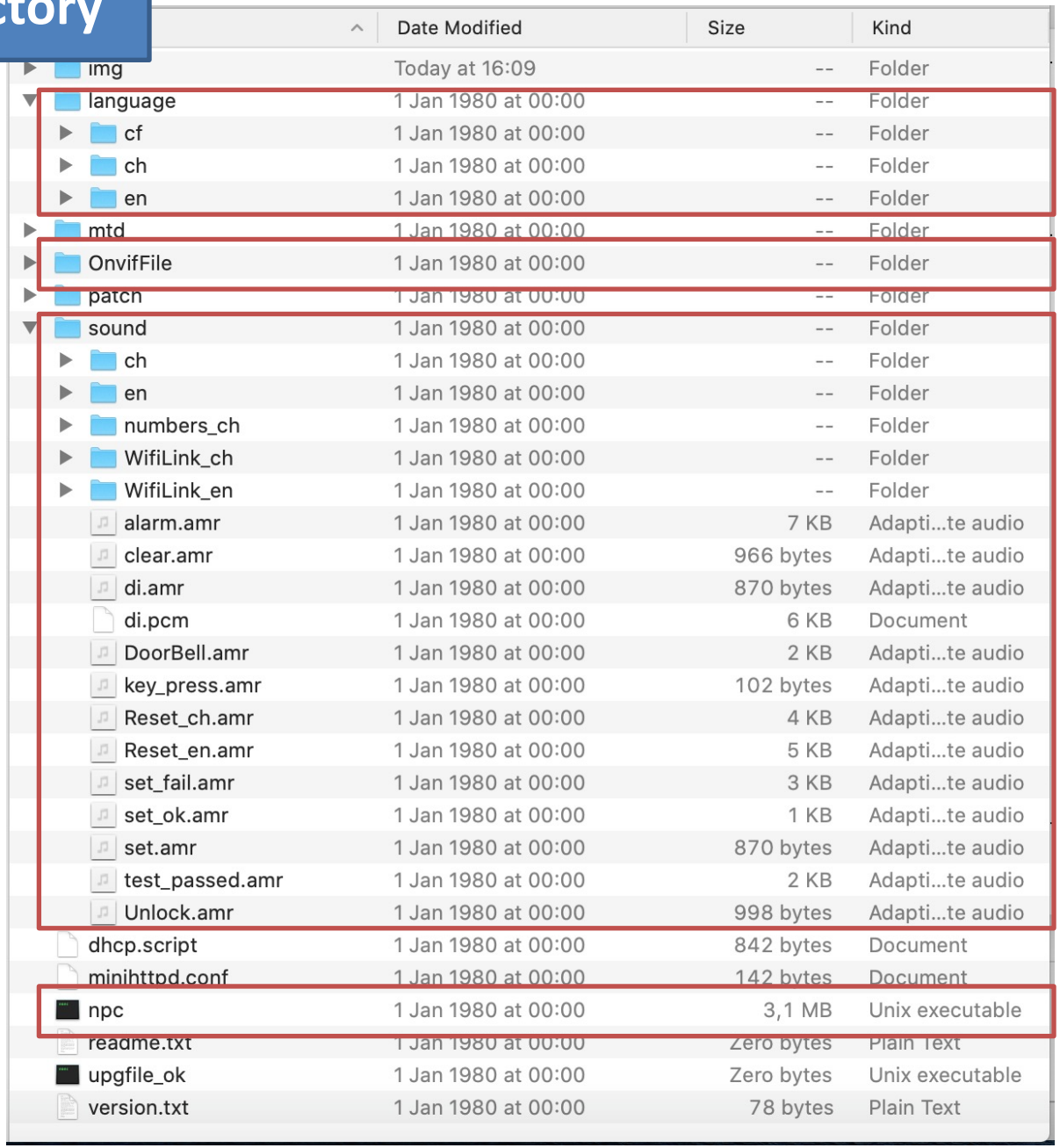
Firmware: CLI

- We insert a uSD and copy the content:

```
cp -r / /mnt/disc1
```

```
File  Edit  View  Search  Terminal  Help
##### Apply #58 #####
audio info: change rec sample rate to 8000
-----Func:vSC2145SwitchDayToNight Line:6864 echo w reload_cfg /npc/mtd/isp328_sc2145_night.cfg > /proc/isp328/command
[ISP_WARN]: [AWB_M2] $grey_cx_boundary$ is not found!!
-----Func:vSC2145SwitchDayToNight brightness:128 contrast:128 sharpness:128 saturation0 target_y:100 denoise:148
Sending discover...

##### Apply #59 #####
vVIControl dwChID= 1 fgEnable=0 fgEn=0
The CmosType(19) support power down
in CMOSPowerDown fgLevelCtl = 1
Performing a DHCP renew
Sending discover...
mount
rootfs on / type rootfs (rw)
/dev/root on / type squashfs (ro,relatime)
devtmpfs on /dev type devtmpfs (rw,relatime,size=30484k,nr_inodes=7621,mode=755)
tmpfs on /dev type tmpfs (rw,relatime,mode=755)
tmpfs on /tmp type tmpfs (rw,relatime,mode=777)
tmpfs on /var type tmpfs (rw,relatime,mode=755)
tmpfs on /bin type tmpfs (rw,relatime,mode=755)
tmpfs on /usr type tmpfs (rw,relatime,mode=755)
tmpfs on /sbin type tmpfs (rw,relatime,mode=755)
/dev/sys on /sys type sysfs (rw,relatime)
none on /proc type proc (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600)
tmpfs on /dev/shm type tmpfs (rw,relatime,size=20480k)
/dev/mtdblock3 on /rom type jffs2 (rw,relatime)
/dev/ram0 on /mnt/ramdisk type ext2 (rw,relatime,errors=continue,user_xattr,acl)
/dev/ram0 on /etc type ext2 (rw,relatime,errors=continue,user_xattr,acl)
/dev/ram0 on /tmp type ext2 (rw,relatime,errors=continue,user_xattr,acl)
/dev/mtdblock4 on /npc type jffs2 (rw,relatime)
/dev/mmcblk0p1 on /mnt/disc1 type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=cp437,iocharset=iso8859-1,shortname=mixed,error)
/ # Sending discover...

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
```

npc directory



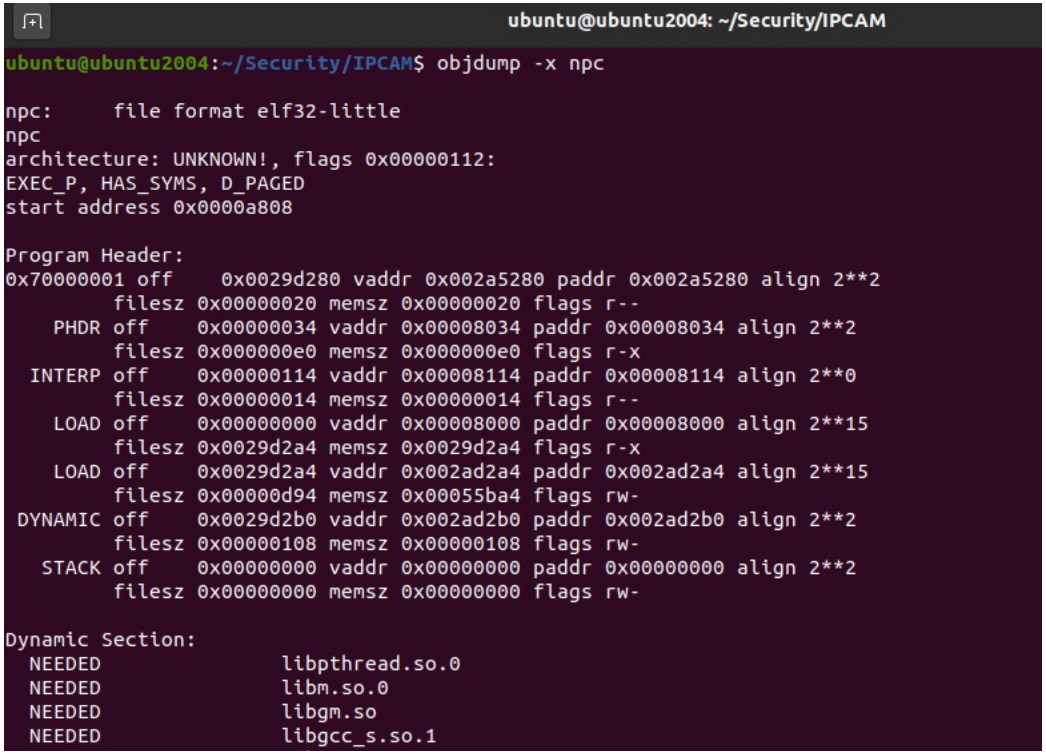
	Date Modified	Size	Kind
img	Today at 16:09	--	Folder
language	1 Jan 1980 at 00:00	--	Folder
cf	1 Jan 1980 at 00:00	--	Folder
ch	1 Jan 1980 at 00:00	--	Folder
en	1 Jan 1980 at 00:00	--	Folder
mtd	1 Jan 1980 at 00:00	--	Folder
OnvifFile	1 Jan 1980 at 00:00	--	Folder
patch	1 Jan 1980 at 00:00	--	Folder
sound	1 Jan 1980 at 00:00	--	Folder
ch	1 Jan 1980 at 00:00	--	Folder
en	1 Jan 1980 at 00:00	--	Folder
numbers_ch	1 Jan 1980 at 00:00	--	Folder
WifiLink_ch	1 Jan 1980 at 00:00	--	Folder
WifiLink_en	1 Jan 1980 at 00:00	--	Folder
alarm.amr	1 Jan 1980 at 00:00	7 KB	Adapti...te audio
clear.amr	1 Jan 1980 at 00:00	966 bytes	Adapti...te audio
di.amr	1 Jan 1980 at 00:00	870 bytes	Adapti...te audio
di.pcm	1 Jan 1980 at 00:00	6 KB	Document
DoorBell.amr	1 Jan 1980 at 00:00	2 KB	Adapti...te audio
key_press.amr	1 Jan 1980 at 00:00	102 bytes	Adapti...te audio
Reset_ch.amr	1 Jan 1980 at 00:00	4 KB	Adapti...te audio
Reset_en.amr	1 Jan 1980 at 00:00	5 KB	Adapti...te audio
set_fail.amr	1 Jan 1980 at 00:00	3 KB	Adapti...te audio
set_ok.amr	1 Jan 1980 at 00:00	1 KB	Adapti...te audio
set.amr	1 Jan 1980 at 00:00	870 bytes	Adapti...te audio
test_passed.amr	1 Jan 1980 at 00:00	2 KB	Adapti...te audio
Unlock.amr	1 Jan 1980 at 00:00	998 bytes	Adapti...te audio
dhcp.script	1 Jan 1980 at 00:00	842 bytes	Document
minihttpd.conf	1 Jan 1980 at 00:00	142 bytes	Document
npc	1 Jan 1980 at 00:00	3,1 MB	Unix executable
readme.txt	1 Jan 1980 at 00:00	Zero bytes	Plain Text
upgfile_ok	1 Jan 1980 at 00:00	Zero bytes	Unix executable
version.txt	1 Jan 1980 at 00:00	78 bytes	Plain Text

Firmware: Resume

- using the SoC manual we get:
  - Memory map, supported devices, etc..
- Access to boot system
  - Firmware, attached devices, partitions (FSs), etc.
- CLI and analyzing firmware, we see the directory tree and look for:
  - OS configuration files
  - Binary files that are not part of the OS:
    - `squashfs-root/bak/npc/npc.tar.gz`
    - `npc directory`

npc file

- Lets analyze the npc binary



- Lets disassemble it!!

Ghidra

- A software for engineering (SRE). Suite of tools developed by NSA's Research Directorate in support of the Cybersecurity mission
- Analyse malicious code and malware like viruses, and can give cybersecurity professionals a better understanding of potential vulnerabilities in their networks and systems
- Key features:
  - Software analysis tools for analysing compiled code
  - Disassembly, assembly, decompilation, graphing and scripting
  - Supports a wide variety of processor instruction sets and executable formats

Ghidra analysis

- Ghidra allows us to do the following:
  - Search and renaming of character strings, variables and functions
    - Variables: definition and use (RW)
    - Functions: tree-shaped analysis and disassembly
  - Once we have located what interests us, we can modify the file and reprogram the Flash

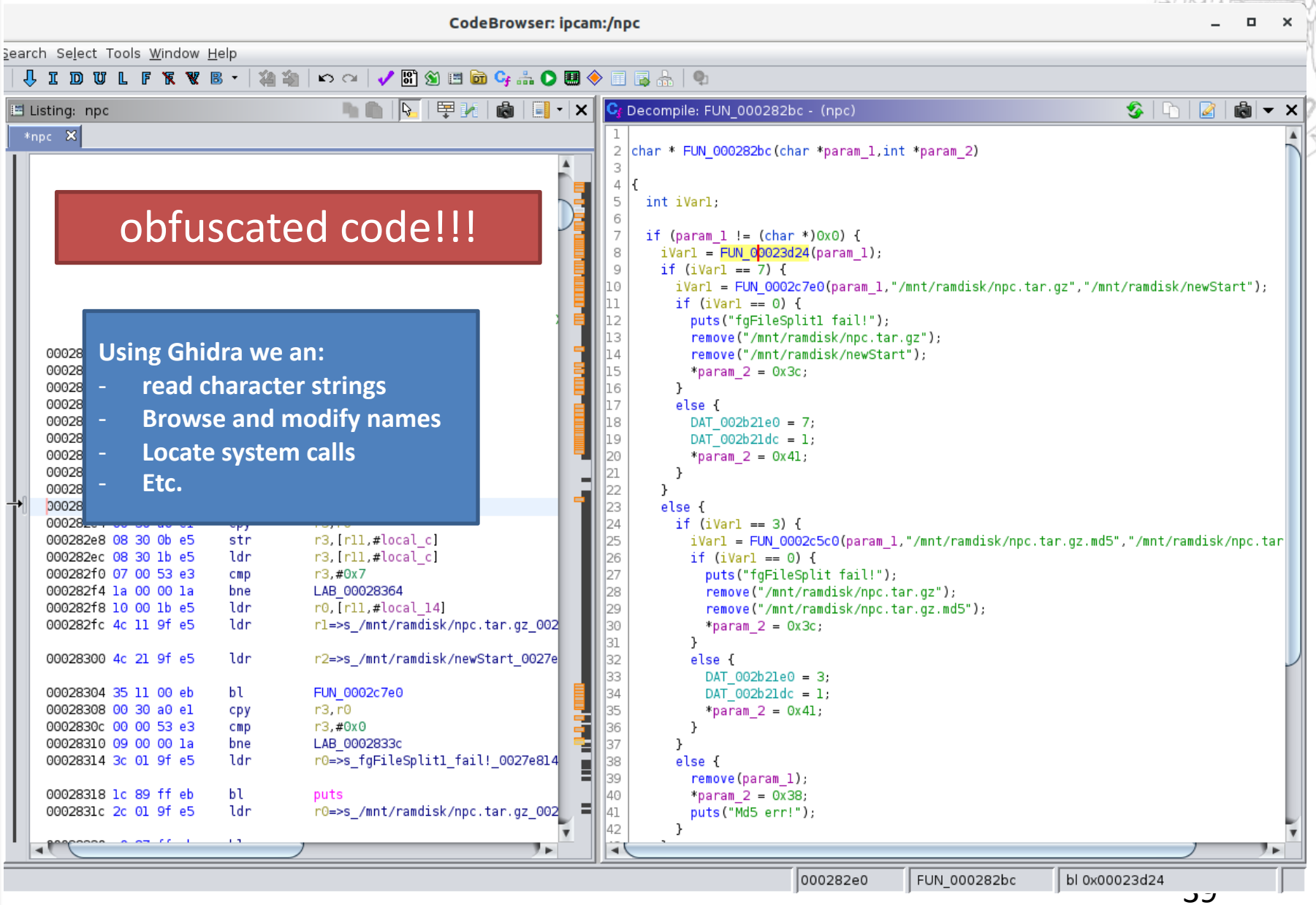
video

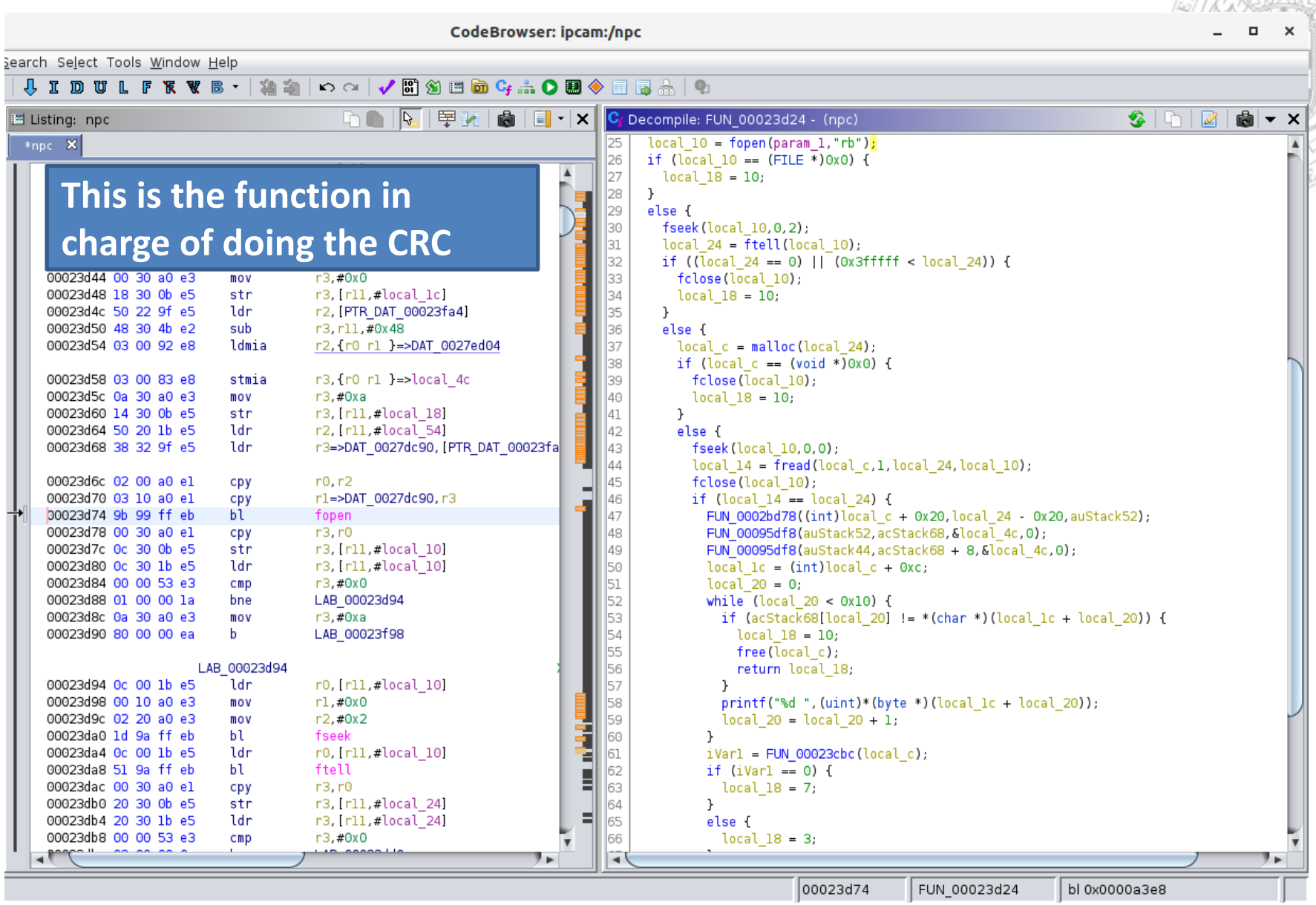
Tasks

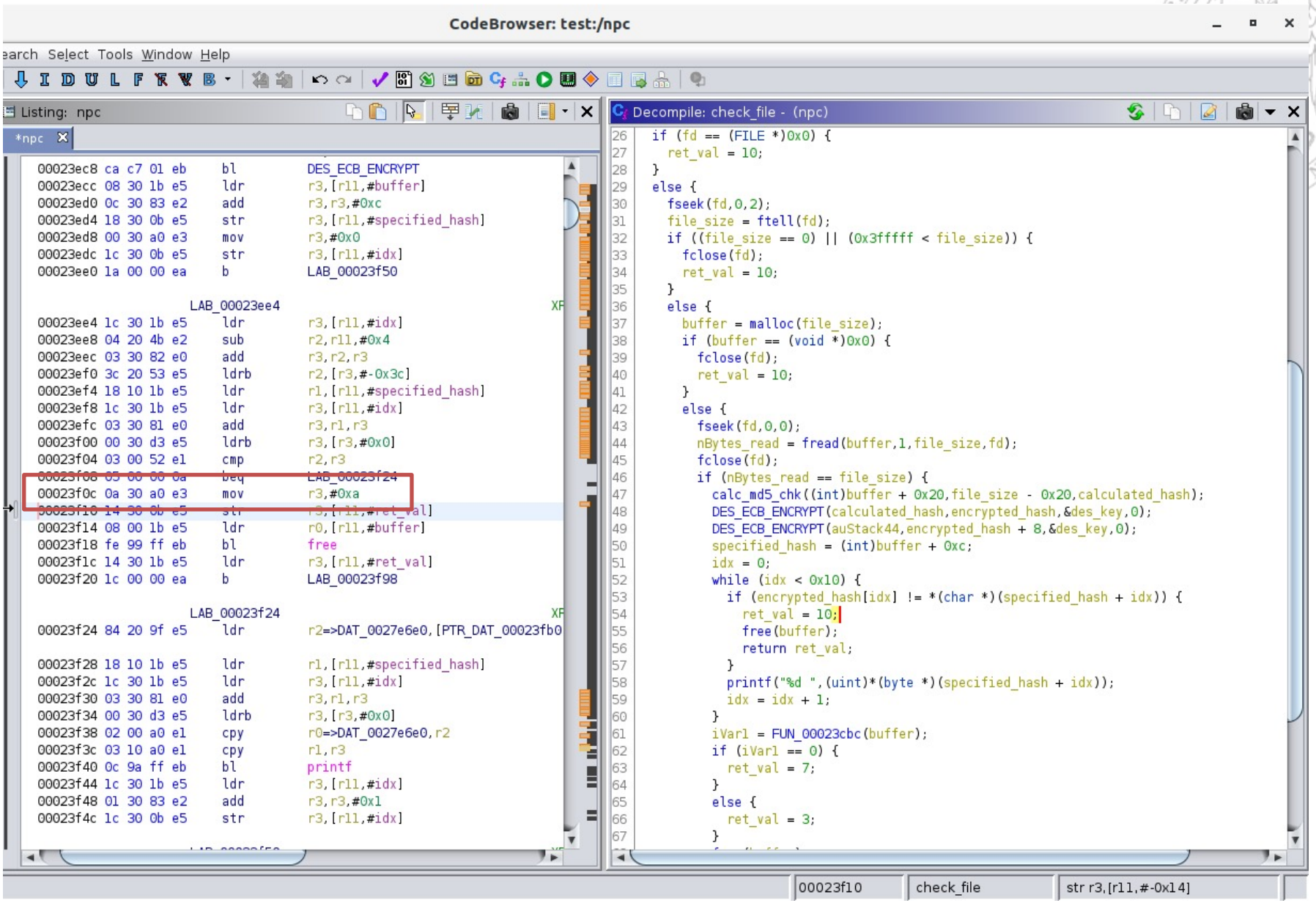
1. Download and import the IOTNA Ubuntu Virtual Machine available in the link
2. Download the firmware `ip_cam_attify.bin` and use `binwalk` to extract its content.
3. Locate the `npc.tar.gz` and extract its content

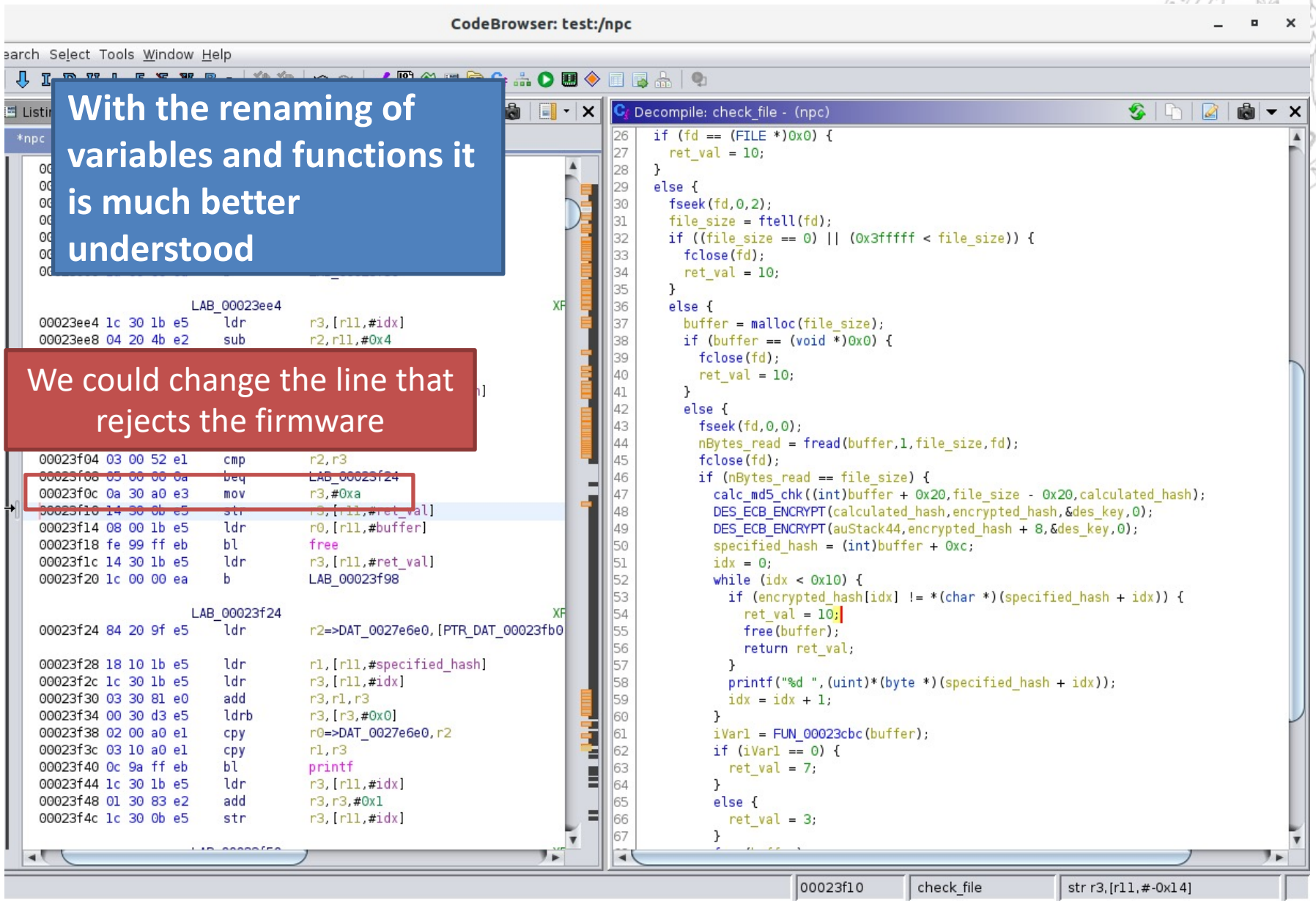
Tasks

4. Analyze the npc file with Ghidra:
   a) `ghidra_10.1.2_PUBLIC/ghidraRun`
   b) Create a project, import the npc file and analyze it
   c) The camera rejects modified Firmwares with the message: "Md5 err!”
   d) Find the String and locate the functions in which it is used


CodeBrowser: ipcam:/npc
Search Select Tools Window Help
Listing: npc
*npc
Decompile: FUN_000282bc - (npc)
obfuscated code!!!
Using Ghidra we an:
- read character strings
- Browse and modify names
- Locate system calls
- Etc.
000282e8 08 30 0b e5 str r3,[r11,#local_c]
000282ec 08 30 1b e5 ldr r3,[r11,#local_c]
000282f0 07 00 53 e3 cmp r3,#0x7
000282f4 1a 00 00 1a bne LAB_00028364
000282f8 10 00 1b e5 ldr r0,[r11,#local_14]
000282fc 4c 11 9f e5 ldr r1=>s_/mnt/ramdisk/npc.tar.gz_002
00028300 4c 21 9f e5 ldr r2=>s_/mnt/ramdisk/newStart_0027e
00028304 35 11 00 eb bl FUN_0002c7e0
00028308 00 30 a0 e1 cpy r3,r0
0002830c 00 00 53 e3 cmp r3,#0x0
00028310 09 00 00 1a bne LAB_0002833c
00028314 3c 01 9f e5 ldr r0=>s_fgFileSplit1_fail!_0027e814
00028318 1c 89 ff eb bl puts
0002831c 2c 01 9f e5 ldr r0=>s_/mnt/ramdisk/npc.tar.gz_002
char * FUN_000282bc(char *param_1,int *param_2)
{
int iVar1;
if (param_1 != (char *)0x0) {
iVar1 = FUN_00023d24(param_1);
if (iVar1 == 7) {
iVar1 = FUN_0002c7e0(param_1,"/mnt/ramdisk/npc.tar.gz","/mnt/ramdisk/newStart");
if (iVar1 == 0) {
puts("fgFileSplit1 fail!");
remove("/mnt/ramdisk/npc.tar.gz");
remove("/mnt/ramdisk/newStart");
*param_2 = 0x3c;
}
else {
DAT_002b21e0 = 7;
DAT_002b21dc = 1;
*param_2 = 0x41;
}
}
else {
if (iVar1 == 3) {
iVar1 = FUN_0002c5c0(param_1,"/mnt/ramdisk/npc.tar.gz.md5","/mnt/ramdisk/npc.tar
if (iVar1 == 0) {
puts("fgFileSplit fail!");
remove("/mnt/ramdisk/npc.tar.gz");
remove("/mnt/ramdisk/npc.tar.gz.md5");
*param_2 = 0x3c;
}
else {
DAT_002b21e0 = 3;
DAT_002b21dc = 1;
*param_2 = 0x41;
}
}
else {
remove(param_1);
*param_2 = 0x38;
puts("Md5 err!");
}
000282e0
FUN_000282bc
bl 0x00023d24


CodeBrowser: ipcam:/npc
Search Select Tools Window Help
Listing: npc
*npc
Decompile: FUN_00023d24 - (npc)
This is the function in charge of doing the CRC
00023d44 00 30 a0 e3 mov r3,#0x0
00023d48 18 30 0b e5 str r3,[r11,#local_1c]
00023d4c 50 22 9f e5 ldr r2,[PTR_DAT_00023fa4]
00023d50 48 30 4b e2 sub r3,r11,#0x48
00023d54 03 00 92 e8 ldmia r2,{r0 r1 }=>DAT_0027ed04
00023d58 03 00 83 e8 stmia r3,{r0 r1 }=>local_4c
00023d5c 0a 30 a0 e3 mov r3,#0xa
00023d60 14 30 0b e5 str r3,[r11,#local_18]
00023d64 50 20 1b e5 ldr r2,[r11,#local_54]
00023d68 38 32 9f e5 ldr r3=>DAT_0027dc90,[PTR_DAT_00023fa
00023d6c 02 00 a0 e1 cpy r0,r2
00023d70 03 10 a0 e1 cpy r1=>DAT_0027dc90,r3
00023d74 9b 99 ff eb bl fopen
00023d78 00 30 a0 e1 cpy r3,r0
00023d7c 0c 30 0b e5 str r3,[r11,#local_10]
00023d80 0c 30 1b e5 ldr r3,[r11,#local_10]
00023d84 00 00 53 e3 cmp r3,#0x0
00023d88 01 00 00 1a bne LAB_00023d94
00023d8c 0a 30 a0 e3 mov r3,#0xa
00023d90 80 00 00 ea b LAB_00023f98
LAB_00023d94
00023d94 0c 00 1b e5 ldr r0,[r11,#local_10]
00023d98 00 10 a0 e3 mov r1,#0x0
00023d9c 02 20 a0 e3 mov r2,#0x2
00023da0 1d 9a ff eb bl fseek
00023da4 0c 00 1b e5 ldr r0,[r11,#local_10]
00023da8 51 9a ff eb bl ftell
00023dac 00 30 a0 e1 cpy r3,r0
00023db0 20 30 0b e5 str r3,[r11,#local_24]
00023db4 20 30 1b e5 ldr r3,[r11,#local_24]
00023db8 00 00 53 e3 cmp r3,#0x0
25 local_10 = fopen(param_1,"rb");
26 if (local_10 == (FILE *)0x0) {
27 local_18 = 10;
28 }
29 else {
30 fseek(local_10,0,2);
31 local_24 = ftell(local_10);
32 if ((local_24 == 0) || (0x3fffff < local_24)) {
33 fclose(local_10);
34 local_18 = 10;
35 }
36 else {
37 local_c = malloc(local_24);
38 if (local_c == (void *)0x0) {
39 fclose(local_10);
40 local_18 = 10;
41 }
42 else {
43 fseek(local_10,0,0);
44 local_14 = fread(local_c,1,local_24,local_10);
45 fclose(local_10);
46 if (local_14 == local_24) {
47 FUN_0002bd78((int)local_c + 0x20,local_24 - 0x20,auStack52);
48 FUN_00095df8(auStack52,acStack68,&local_4c,0);
49 FUN_00095df8(auStack44,acStack68 + 8,&local_4c,0);
50 local_1c = (int)local_c + 0xc;
51 local_20 = 0;
52 while (local_20 < 0x10) {
53 if (acStack68[local_20] != *(char *)(local_1c + local_20)) {
54 local_18 = 10;
55 free(local_c);
56 return local_18;
57 }
58 printf("%d ",(uint)*(byte *)(local_1c + local_20));
59 local_20 = local_20 + 1;
60 }
61 iVar1 = FUN_00023cbc(local_c);
62 if (iVar1 == 0) {
63 local_18 = 7;
64 }
65 else {
66 local_18 = 3;
00023d74
FUN_00023d24
bl 0x0000a3e8


CodeBrowser: test:/npc
Search Select Tools Window Help
Listing: npc
*npc
00023ec8 ca c7 01 eb bl DES_ECB_ENCRYPT
00023ecc 08 30 1b e5 ldr r3,[r11,#buffer]
00023ed0 0c 30 83 e2 add r3,r3,#0xc
00023ed4 18 30 0b e5 str r3,[r11,#specified_hash]
00023ed8 00 30 a0 e3 mov r3,#0x0
00023edc 1c 30 0b e5 str r3,[r11,#idx]
00023ee0 1a 00 00 ea b LAB_00023f50
LAB_00023ee4
00023ee4 1c 30 1b e5 ldr r3,[r11,#idx]
00023ee8 04 20 4b e2 sub r2,r11,#0x4
00023eec 03 30 82 e0 add r3,r2,r3
00023ef0 3c 20 53 e5 ldrb r2,[r3,#-0x3c]
00023ef4 18 10 1b e5 ldr r1,[r11,#specified_hash]
00023ef8 1c 30 1b e5 ldr r3,[r11,#idx]
00023efc 03 30 81 e0 add r3,r1,r3
00023f00 00 30 d3 e5 ldrb r3,[r3,#0x0]
00023f04 03 00 52 e1 cmp r2,r3
00023f08 05 00 00 0a beq LAB_00023f24
00023f0c 0a 30 a0 e3 mov r3,#0xa
00023f10 14 30 0b e5 str r3,[r11,#ret_val]
00023f14 08 00 1b e5 ldr r0,[r11,#buffer]
00023f18 fe 99 ff eb bl free
00023f1c 14 30 1b e5 ldr r3,[r11,#ret_val]
00023f20 1c 00 00 ea b LAB_00023f98
LAB_00023f24
00023f24 84 20 9f e5 ldr r2=>DAT_0027e6e0,[PTR_DAT_00023fb0
00023f28 18 10 1b e5 ldr r1,[r11,#specified_hash]
00023f2c 1c 30 1b e5 ldr r3,[r11,#idx]
00023f30 03 30 81 e0 add r3,r1,r3
00023f34 00 30 d3 e5 ldrb r3,[r3,#0x0]
00023f38 02 00 a0 e1 cpy r0=>DAT_0027e6e0,r2
00023f3c 03 10 a0 e1 cpy r1,r3
00023f40 0c 9a ff eb bl printf
00023f44 1c 30 1b e5 ldr r3,[r11,#idx]
00023f48 01 30 83 e2 add r3,r3,#0x1
00023f4c 1c 30 0b e5 str r3,[r11,#idx]
Decompile: check_file - (npc)
26 if (fd == (FILE *)0x0) {
27 ret_val = 10;
28 }
29 else {
30 fseek(fd,0,2);
31 file_size = ftell(fd);
32 if ((file_size == 0) || (0x3fffff < file_size)) {
33 fclose(fd);
34 ret_val = 10;
35 }
36 else {
37 buffer = malloc(file_size);
38 if (buffer == (void *)0x0) {
39 fclose(fd);
40 ret_val = 10;
41 }
42 else {
43 fseek(fd,0,0);
44 nBytes_read = fread(buffer,1,file_size,fd);
45 fclose(fd);
46 if (nBytes_read == file_size) {
47 calc_md5_chk((int)buffer + 0x20,file_size - 0x20,calculated_hash);
48 DES_ECB_ENCRYPT(calculated_hash,encrypted_hash,&des_key,0);
49 DES_ECB_ENCRYPT(auStack44,encrypted_hash + 8,&des_key,0);
50 specified_hash = (int)buffer + 0xc;
51 idx = 0;
52 while (idx < 0x10) {
53 if (encrypted_hash[idx] != *(char *)(specified_hash + idx)) {
54 ret_val = 10;
55 free(buffer);
56 return ret_val;
57 }
58 printf("%d ",(uint)*(byte *)(specified_hash + idx));
59 idx = idx + 1;
60 }
61 iVar1 = FUN_00023cbc(buffer);
62 if (iVar1 == 0) {
63 ret_val = 7;
64 }
65 else {
66 ret_val = 3;
67 }
00023f10
check_file
str r3,[r11,#-0x14]

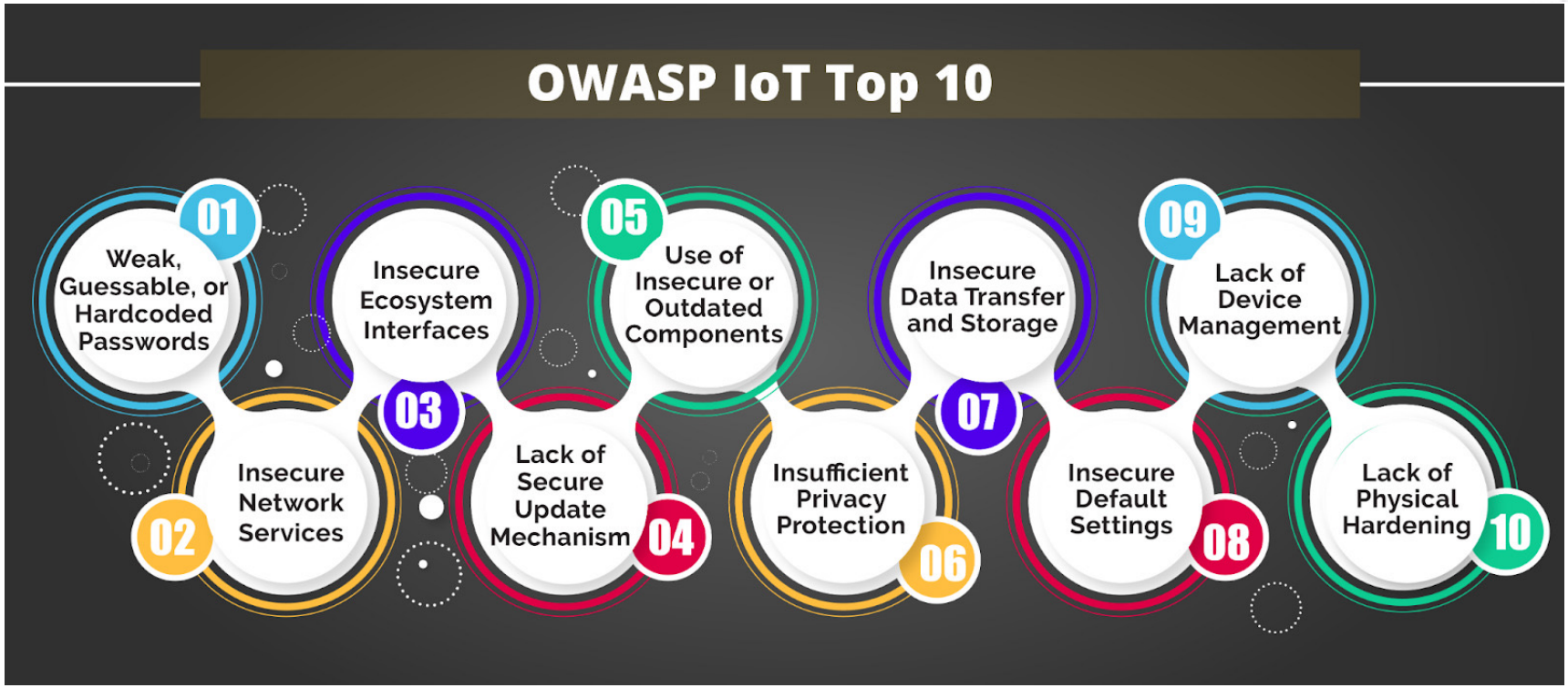

CodeBrowser: test:/npc
Search Select Tools Window Help
With the renaming of variables and functions it is much better understood
We could change the line that rejects the firmware
Decompile: check_file - (npc)
LAB_00023ee4
00023ee4 1c 30 1b e5 ldr r3,[r11,#idx]
00023ee8 04 20 4b e2 sub r2,r11,#0x4
00023f04 03 00 52 e1 cmp r2,r3
00023f08 05 00 00 0a beq LAB_00023f24
00023f0c 0a 30 a0 e3 mov r3,#0xa
00023f14 08 00 1b e5 ldr r0,[r11,#buffer]
00023f18 fe 99 ff eb bl free
00023f1c 14 30 1b e5 ldr r3,[r11,#ret_val]
00023f20 1c 00 00 ea b LAB_00023f98
LAB_00023f24
00023f24 84 20 9f e5 ldr r2=>DAT_0027e6e0,[PTR_DAT_00023fb0
00023f28 18 10 1b e5 ldr r1,[r11,#specified_hash]
00023f2c 1c 30 1b e5 ldr r3,[r11,#idx]
00023f30 03 30 81 e0 add r3,r1,r3
00023f34 00 30 d3 e5 ldrb r3,[r3,#0x0]
00023f38 02 00 a0 e1 cpy r0=>DAT_0027e6e0,r2
00023f3c 03 10 a0 e1 cpy r1,r3
00023f40 0c 9a ff eb bl printf
00023f44 1c 30 1b e5 ldr r3,[r11,#idx]
00023f48 01 30 83 e2 add r3,r3,#0x1
00023f4c 1c 30 0b e5 str r3,[r11,#idx]
if (fd == (FILE *)0x0) {
ret_val = 10;
}
else {
fseek(fd,0,2);
file_size = ftell(fd);
if ((file_size == 0) || (0x3fffff < file_size)) {
fclose(fd);
ret_val = 10;
}
else {
buffer = malloc(file_size);
if (buffer == (void *)0x0) {
fclose(fd);
ret_val = 10;
}
else {
fseek(fd,0,0);
nBytes_read = fread(buffer,1,file_size,fd);
fclose(fd);
if (nBytes_read == file_size) {
calc_md5_chk((int)buffer + 0x20,file_size - 0x20,calculated_hash);
DES_ECB_ENCRYPT(calculated_hash,encrypted_hash,&des_key,0);
DES_ECB_ENCRYPT(auStack44,encrypted_hash + 8,&des_key,0);
specified_hash = (int)buffer + 0xc;
idx = 0;
while (idx < 0x10) {
if (encrypted_hash[idx] != *(char *)(specified_hash + idx)) {
ret_val = 10;
free(buffer);
return ret_val;
}
printf("%d ",(uint)*(byte *)(specified_hash + idx));
idx = idx + 1;
}
iVar1 = FUN_00023cbc(buffer);
if (iVar1 == 0) {
ret_val = 7;
}
else {
ret_val = 3;
}
00023f10
check_file
str r3,[r11,#-0x14]

OWASP Recommendations

- If we had followed the OWAS recommendations, would we have avoided these security holes?

- Lets review the recommendations....

OWASP IoT Top 10



https://www.appsealing.com/owasp-iot-top-10/

TOP 1:
Weak, Guessable, or Hardcoded Pass.

- *"Use of easily bruteforced, publicly available (defpass) or unchangeable credentials, including **backdoors in firmware** or client software that **grants unauthorized access to deployed systems**."*
- To combat the first vulnerability, manufacturers must take the following steps:
  - Every device must have a unique set of credentials
  - Disable weak passwords
  - **Removing backdoors created during debugging**

TOP 3:
Insecure Ecosystem Interfaces

- *"Insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components. Common issues include: a **lack of authentication/authorization**, **lacking or weak encryption**, and a lack of input and output filtering."*
- To address insecure interfaces, the following tips are useful:
  - Adhering to the principle of least privilege
  - Block public access to private resources (S3 bucket AWS)
  - **Strong authentication of IoT endpoints**

TOP 4:
Lack of Secure Update Mechanism

- *"**Lack of ability to securely update the device**. This includes lack of firmware validation on device, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanisms, and lack of notifications of security changes due to updates."*
- For secure delivery of updates to IoT devices, manufacturers must:
  - **Only implement updates that are digitally signed**
  - Implement anti-rollback mechanisms
  - **Secure and verify access to updates**

TOP 10:
Lack of Physical Hardening

- *"**Lack of physical hardening measures**, allowing potential attackers to gain sensitive information that can help in a future remote attack or take local control of the device."*
- To counter physical threats to IoT devices, manufacturers should:
  - **Understand how a user may modify the device**
  - Proactively anticipate what damages any user may inflict on the device
  - **Devise solutions and build an IoT device that can withstand all the possible attacks**