BLUEHAT SHANGHAI 2019

百度安全 有AI更安全

New Techniques for Discovering Smart Device Vulnerabilities

小灰灰 / Xiaohuihui

Baidu Security Lab Senior Security Researcher

30-May2019

Who am I

Senior Security Researcher at Baidu Security Lab

Research areas: IoT security / AI security / autopilot vehicle security

Experienced hardware cracker

Previously Responsible For:

- BSRC, incident response, 0-day analysis
- Baidu product security assessment
- Baidu security monitoring system construction

Traditional IoT Device Cracking

Router?

Firmware download

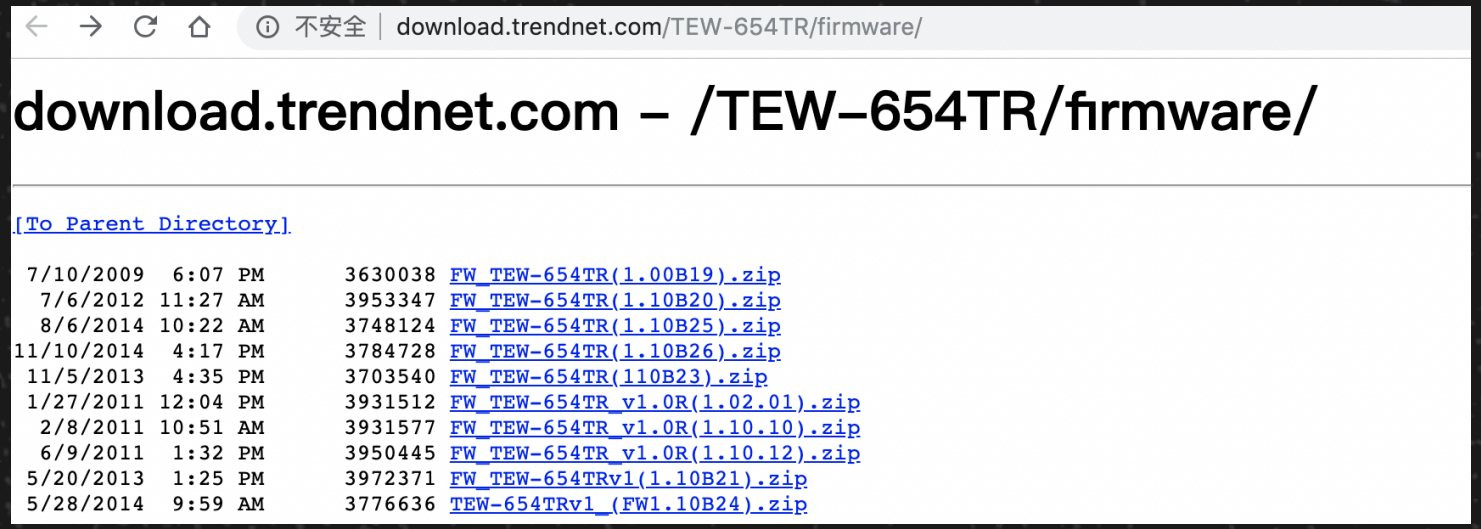
Binwalk unpacking

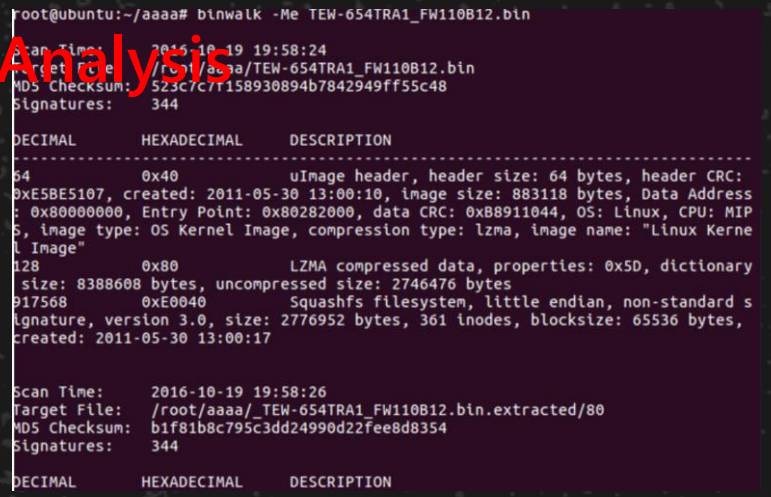
Finding problematic bin(why?)

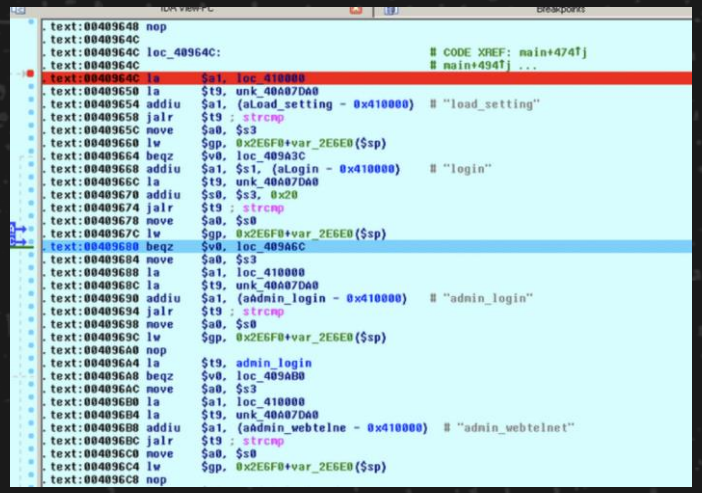
IDA analysis, WEB script file analysis

Vulnerability verification (real machine or QEMU test)

Most of them seem to be Vulnerability Analysis







Now ?

We have a lot more different IoT devices, not just routers

Large manufacturers won't let you easily crack the devices:

- Firmware is not available for download
- Telnet, serial port, ADB are all closed
- There's no way to get started

What do we do? ? ?

The Structures Are Similar

OS and Hardware	Full Android and Linux versions, ARMv5/6/7/x86 processors, EMMC/EMCP/NAND memory	Openwrt Linux with microkernel, ARM, MIPS processor, NAND/SPI Flash memory	RTOS Linux that has real-time operating systems with microkernel, ESP Lexin, Arduino system-on-chip, AVR, STM32 series, SPI Flash memory
Application Scenarios	Smart Speakers, Smart Watches, Vending Machines, TV boxes, Smart TVs, Smart Billboards, Vehicle System	Router, Mini Smart Speaker, Smart Camera	Smart Lock, Smart Rice Coker, Smart Socket, Smart Lamp, Smart Bracelet
Features	More features, more memory space, great platform for developing apps, most of them carry a large screen	Simple but have advanced features, large screen is not necessary/only small screen is needed	Function is simple but can be controlled via network, analogue electronics are not feasible.

Step 0: Teardown!

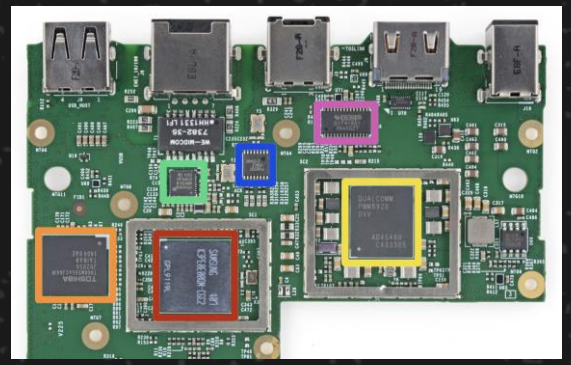
Be clear about what to do and where to go

- What are the chip solutions? What we can do?
- What to do when we get stuck?

Look for chip model information and datasheet

Pay attention to：

- Storage type and specifications
  - SPI Flash 8/16/Wide&Narrow
  - EMMC/EMCP 100/153/162/169/186/221/254
  - NandFlash TSOP32/40/48
- TTL and JTAG interface (how to find)
- Communications module (Ethernet, Bluetooth, wifi, 234G)



图片来自：https://www.crowdsupply.com/teardown/portland-2018

Online Teardown -- A good Way to Find Targets

Search for xxx teardown

Forum (teardown forum)

Ifixit.com

- Contains hardware devices from well-known manufacturers
- Clear picture, label

Fccid.io

- All devices with wireless capabilities and released internationally
- You can find many different devices
- Tips: Search site:fccid.io internal photos xxx

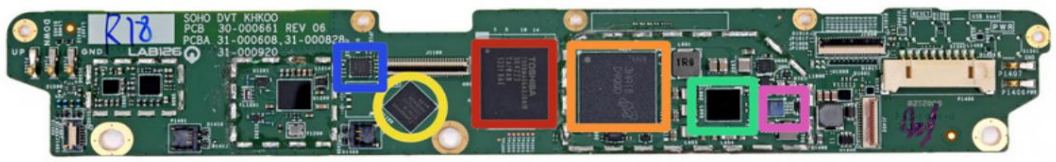
Tesla Motors, Car Key Fob - Found on fccid.io

Kindle Fire’s Logic Board - found on ifixit

安全 | https://www.ifixit.com/Teardown/Kindle+Fire+HD+2013+Teardown/18027

Step 10

Edit



THGBMAG6A2JBAIR Details - Toshiba | Datasheets
https://www.datasheets.com/details/thgbmag6a2jbair-toshiba-55307562 ▾ 翻译此页
Description: MLC NAND Flash Serial e-MMC 3.3V 64G-bit 153-Pin VFBGA. Taxonomy: Memory > Memory Chips > Flash. ECCN: 3A991.b.1.a. Supplier Cage ...

- The front side of the Kindle Fire HD's motherboard is occupied by the following ICs:
  - Toshiba THGBMAG6A2JBAIR 64 Gb (8 GB) e-MMC NAND Flash
  - Micron 3HAI8 D9QQD 8 Gb (1 GB) Mobile LPDDR2 SDRAM
    - We believe that the 1.5 Ghz Dual Core TI OMAP4 (4470) HS processor is nested underneath the Micron SDRAM IC.
  - Synaptics S7301B Touchscreen Controller
  - Texas Instruments TWL6032 Fully Integrated Power Management with Power Path and Battery Charger
  - InvenSense MPU-6500 6-axis gyroscope
  - 347 CB307

Step 1 Premise：Control and Acquire as You Like

Control & Acquire

- Acquire file system
- Getshell (easy for analysis and viewing network/files/processes)
- Acquire and control network data

Eventually we can use the acquired information to conduct a comprehensive analysis and find valid vulnerabilities.

Tips: it is not necessary to follow a specific order here, these steps can be mixed

- For example, you can get the firmware directly after running getshell, or dump the firmware, make modifications, and then run getshell.
- For example, to acquire interactive data, you can have connection upgrades to get the firmware download address directly.

Preparation - Acquire Firmware

Purpose：

- Understand the OS and file system structure, focus on the key directory (/etc /home /usr/bin ..., if it is Android, /system/priv-app)
- Analyze the startup script (/etc/inittab /etc/init.d), loaded binary and configuration file
- Analyze web directory files (CGI, PHP, Lua.....)
- Easy for restoring the old system version (e.g. activate telnet), and easy for analysis
- Make reverse engineering easy as it's possible that the firmware is a new version of APK
- Chroot to the corresponding processor's QEMU for analyzing the binary & web

Preparation - Acquire Firmware

Methods：

- Download from the official website
- Self-upgrade, monitor the packets (if you query the version, you need to intercept the modified older version. Special channel.)
- Reverse engineer the App update and the update process (ftp access)
- Get help from forum and friends (industry maintenance forum)
- Contact customer service (help device recovery)
- Get the shell (telnet, ssh, adb...) and dump firmware (dd, tar, nc out)
- Enter BootLoader to read memory
- Special access to master console (such as the MTK, NXP series - you can read/write file system via data line port)

But, sometimes none of these works.

- img
  - img-777919935_vol-rootfs
    - bin
    - data
    - dev
    - etc
      - adckey
      - asound.conf
      - avskey
      - bsa
      - bt
      - fstab
      - gpio_key.kl
      - group
      - hostapd.conf
      - hostname
      - hosts
      - init.d
        - rcK
        - rcS
        - S03datamount
        - S20urandom
        - S40network
        - S42wifi
        - S43pdnsd
        - S44bluetooth
        - S45ntpd
        - S50sshd
        - S90optinit
      - inittab

```
# Startup the system
::sysinit:/bin/mount -t proc proc /proc
::sysinit:/bin/mkdir /dev/shm
::sysinit:/bin/mkdir /dev/pts
::sysinit:/bin/mount -o remount,rw /
::sysinit:/bin/mount -a
::sysinit:/bin/hostname -F /etc/hostname
::sysinit:/sbin/ifconfig lo 127.0.0.1 up
::sysinit:/sbin/route add -net 127.0.0.0 netmask 255.0.0.0 lo
# now run any rc scripts
::sysinit:/etc/init.d/rcS

tty3::respawn:/usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
tty3::respawn:/usr/bin/hardware
tty3::respawn:/usr/bin/gateway
tty3::once:/usr/bin/rockcli hardware hardware.led_play mode=flash rgb=0066ed on_time=

# Put a getty on the serial port
#ttyS0::respawn:/sbin/getty -L ttyS0 115200 vt100 # UNSUPPORT GENERIC_SERIAL

#todo use /usr/bin/rlogin at production release
ttyS0::respawn:-/bin/sh  # AMLOGIC_GENERAL_SERIAL

# Logging junk
null::sysinit:/bin/touch /var/log/messages
null::respawn:/sbin/syslogd -n
null::respawn:/sbin/klogd -n
```

http://192.168.1.1/cgi-bin/webproc?getpage=html/gui/APIS/returnWifiJSON.txt&var:p


```
{ "RETURN":{ "success": true }, "WIFI": { "status":"1", "ssidName":"Amelia", "ssidVisibility":"1",
"channelMode":"MANUAL", "channel":"4", "SECURITY":{ "cipherAlgorithm": "WPA" , "algVersion": "WPA1" ,
"passwordWEP":"12345", "passwordWPA":"GUSS1986", "passwordWPA2":"GUSS1986", "passwordAUTO":"GUSS1986" } },
"DHCP": { "status":"1", "poolStart":"192.168.1.33", "poolEnd":"192.168.1.254" }, "LAN": { "ip": "192.168.1.1"
, "mask": "255.255.255.0",
```


```
<?
if ($_POST["act"] == "ping")
{
    set("/runtime/diagnostic/ping", $_POST["dst"]);
    $result = "OK";
}
```

```
{
  sprintf((char *)&v31, "USER %s\r\n", aWan[redacted]);
  v9 = strlen((const char *)&v31);
  if ( send(v1, &v31, v9, 0x4000) > 0 && read(v1, &s, 0x400u) != -1 )
  {
    v10 = strlen("331");
    if ( !strncmp(&s, "331", v10) )
    {
      sprintf((char *)&v31, "PASS %s\r\n", aWyf[redacted]);
      v11 = strlen((const char *)&v31);
      if ( send(v1, &v31, v11, 0x4000) > 0 && read(v1, &s, 0x400u) != -1 )
      {
        v12 = strlen("230");
        if ( !strncmp(&s, "230", v12) )
        {
          v13 = strlen("PASV\r\n");
          if ( send(v1, "PASV\r\n", v13, 0x4000) > 0 && read(v1, &s, 0x400u) != -1 )
          {
```

192.168.1.1/form2saveConf.cgi?submit.htm?saveconf.h


```
</chain>
<chain N="USERNAME_PASSWORD">
<V N="FLAG" V="0x0"/>
<V N="USERNAME" V="1234"/>
<V N="PASSWORD" V="1234"/>
<V N="BACKDOOR" V="0x0"/>
<V N="PRIORITY" V="0x2"/>
</chain>
<chain N="USERNAME_PASSWORD">
<V N="FLAG" V="0x0"/>
<V N="USERNAME" V="admin"/>
<V N="PASSWORD" V="7449airocon"/>
```


/

名称	大小	修改时间
public	0	2019-03-06 20:08:00
root	0	2017-06-14 18:03:05
74083	1.21 MB	2018-06-27 14:06:52
428489	285 KB	2017-06-26 14:52:31
512601	1.21 MB	2018-05-08 16:46:01
9575483	1.18 MB	2017-05-31 15:49:09
0489284	284 KB	2017-06-19 17:06:33
5774639	273 KB	2017-05-16 15:50:52
4771076	1.18 MB	2017-05-27 12:34:36
4367697	265 KB	2017-04-28 11:22:28
7012604	285 KB	2017-06-29 11:37:09
9925053	266 KB	2017-04-28 11:22:28
7139746	273 KB	2017-05-08 15:22:24
3961881	1.22 MB	2019-03-05 11:33:05
1307502	1.21 MB	2018-03-26 14:50:02
2920649	1.22 MB	2018-12-04 14:41:27
2878292	264 KB	2017-04-28 11:22:29
1539983	1.21 MB	2018-05-18 18:04:42
1720848	259 KB	2017-04-28 11:22:29
8319901	1.21 MB	2018-01-09 16:47:46
5501172	1.21 MB	2018-03-29 14:20:26

Physical Dump

When it’s impossible to acquire firmware through normal channels

- Large manufacture's firmware is encrypted and cannot be decrypted by using binwalk
- There is no firmware upgrade process so the firmware remains unchanged
- The firmware is upgraded via GPRS and cannot be intervened (actually we can intervene ☺)
- TTL off, telnet off, unable to stop BootLoader and enter

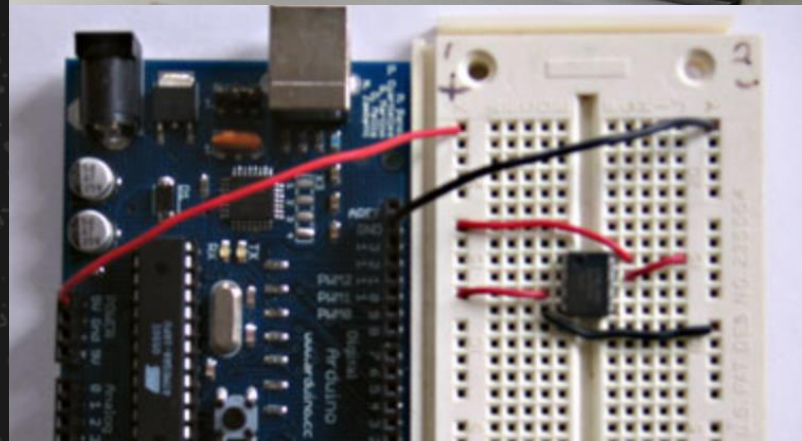
Then tear it apart, physical dump

- Different read/write methods for different models
- Steps:
  - Choose a device and the read/write method (online or offline)
  - Hack the firmware

Physical Dump-SPI Flash

SPI Flash (for simple devices/routing devices)

- Serial read/write device, common capacity is 4/8/16MB, 8-pin SPI interface
- Structure:
  - Full operating system: Bootloader+kernel+file system, mostly compressed
  - Only for storing data, configuration files, etc.
- Read and write methods:
  - Arduino+EEPROM library
  - Raspberry SPI interface + flashrom
  - Programmer can R/W faster (RT809H)
- Soldering is not required (hook and clip is enough). However, sometimes it doesn't work (when the CPU is powered), so it's better to take it out
- Can directly modify the firmware and run getshell. Need to pay attention to the file structure.
  - File system, offset: acquire start information and binwalk
  - Soldering->Unpack->Modify->Repackage->dd offset, merge->brush write back->Soldering



Physical Dump-SPI Flash- Acquire the File System Structure

```
[   0.500000] m25p80 spi0.0: s25fl064k (8192 Kbytes)
[   0.510000] 5 tp-link partitions found on MTD device spi0.0
[   0.510000] Creating 5 MTD partitions on "spi0.0":
[   0.520000] 0x000000000000-0x000000020000 : "u-boot"
[   0.520000] 0x000000020000-0x00000013f5dc : "kernel"
[   0.530000] 0x00000013f5dc-0x0000007f0000 : "rootfs"
[   0.530000] mtd: device 2 (rootfs) set to be root filesystem
[   0.540000] 1 squashfs-split partitions found on MTD device rootfs
[   0.540000] 0x000000370000-0x0000007f0000 : "rootfs_data"
[   0.550000] 0x0000007f0000-0x000000800000 : "art"
[   0.550000] 0x000000020000-0x0000007f0000 : "firmware"
```

Obtained through console information output

```
root@OpenWrt:/# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00020000 00010000 "u-boot"
mtd1: 000f0000 00010000 "kernel"
mtd2: 006e0000 00010000 "rootfs"
mtd3: 00010000 00010000 "art"
mtd4: 007d0000 00010000 "firmware"
```

Obtained by shell command

```
➜  ~ binwalk /Volumes/Untitled/tplink.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
23728         0x5CB0          CRC32 polynomial table, big endian
25184         0x6260          uImage header, header size: 64 bytes, header CRC: 0xEAE8B8C1, created
0010000, data CRC: 0xBBDF4C08, OS: Linux, CPU: MIPS, image type: Firmware Image, compression type:
25248         0x62A0          LZMA compressed data, properties: 0x6D, dictionary size: 33554432 byt
131584        0x20200         LZMA compressed data, properties: 0x6D, dictionary size: 8388608 byte
1308124       0x13F5DC        Squashfs filesystem, little endian, version 4.0, compression:xz, size
3604480       0x370000        JFFS2 filesystem, big endian
```

Obtained through binwalk

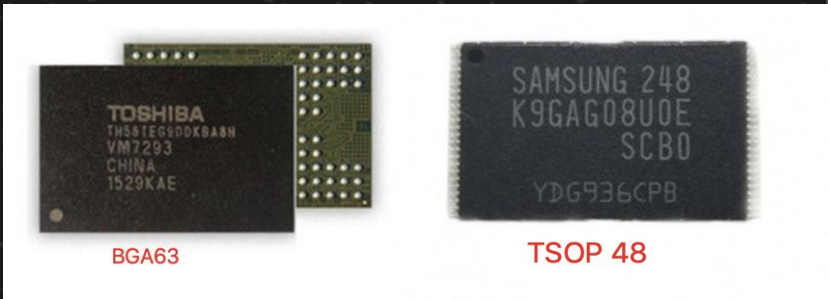
Physical Dump-NandFlash

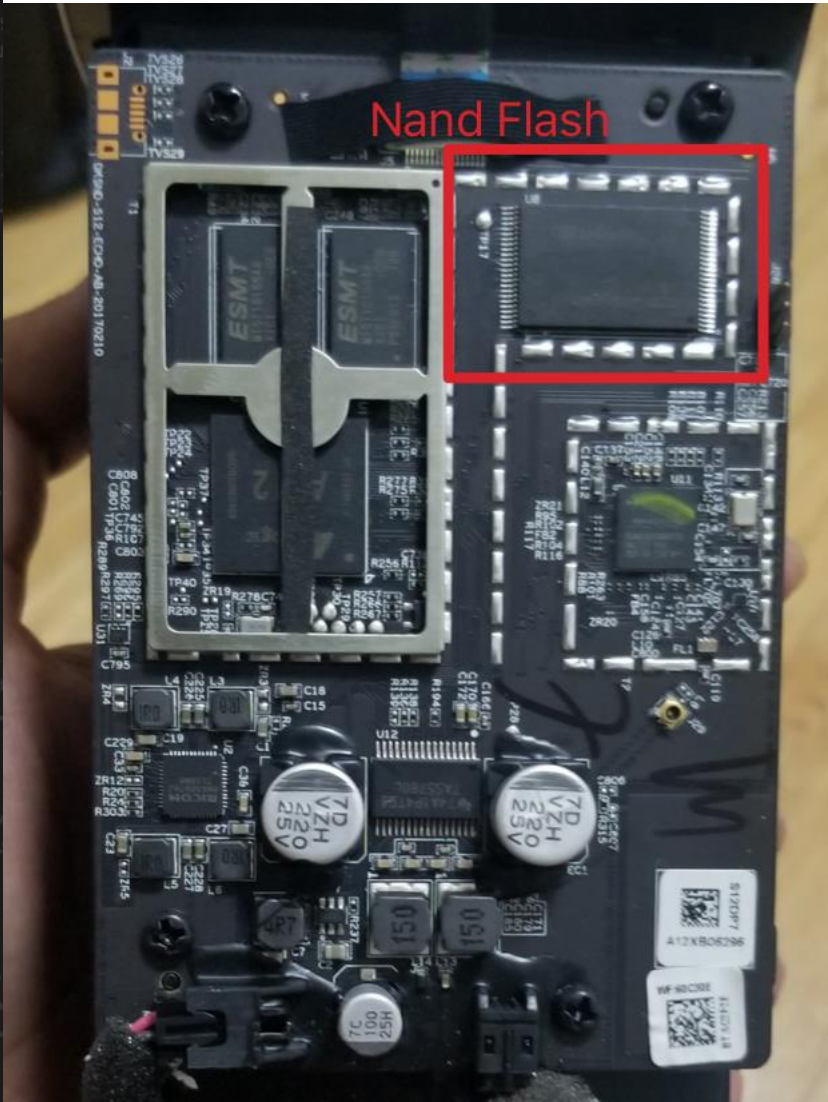
NandFlash (for more complex devices such as advanced routing, smart speakers)

- 16MB-2GB capacity, TSOP48/BGA package, read and write by block
- People new to drag soldering often run into issues. It is recommended to use a heat gun for de-soldering (be careful about the surrounding components)
- Structure: most of the full Linux/Android system do not require compression and decompression
- Read and write methods:
  - 17+ valid pins require programmer for read/write (e.g. RT809H)
  - Bad block management exists but is not very advanced. Write is more complicated here.

Warning: usually binwalk cannot decrypt the acquired bin firmware. Modifying binwalk or removing ECC check digit data is needed.

Handling this type of device is more challenging than SPI Flash and EMMC. Also the file system format is always different.

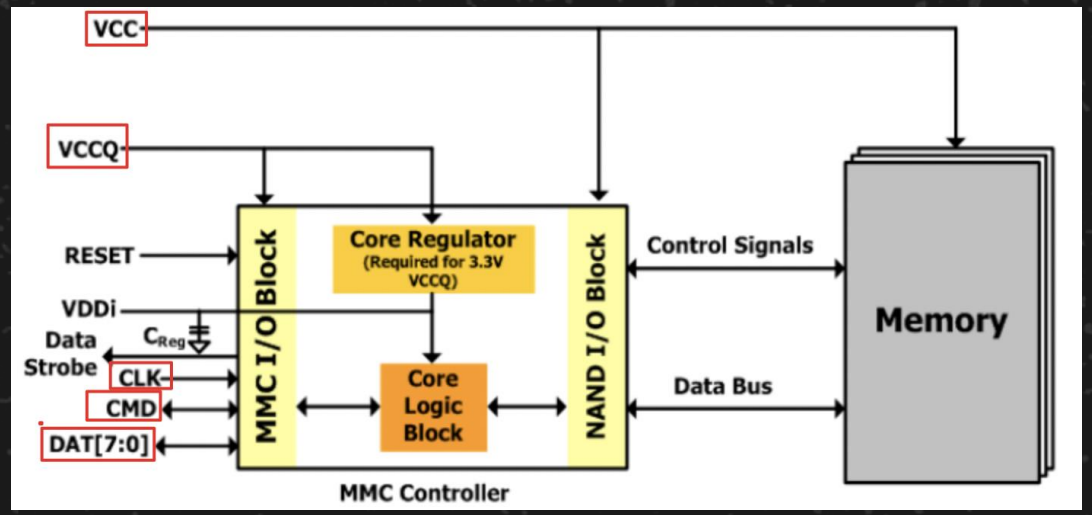


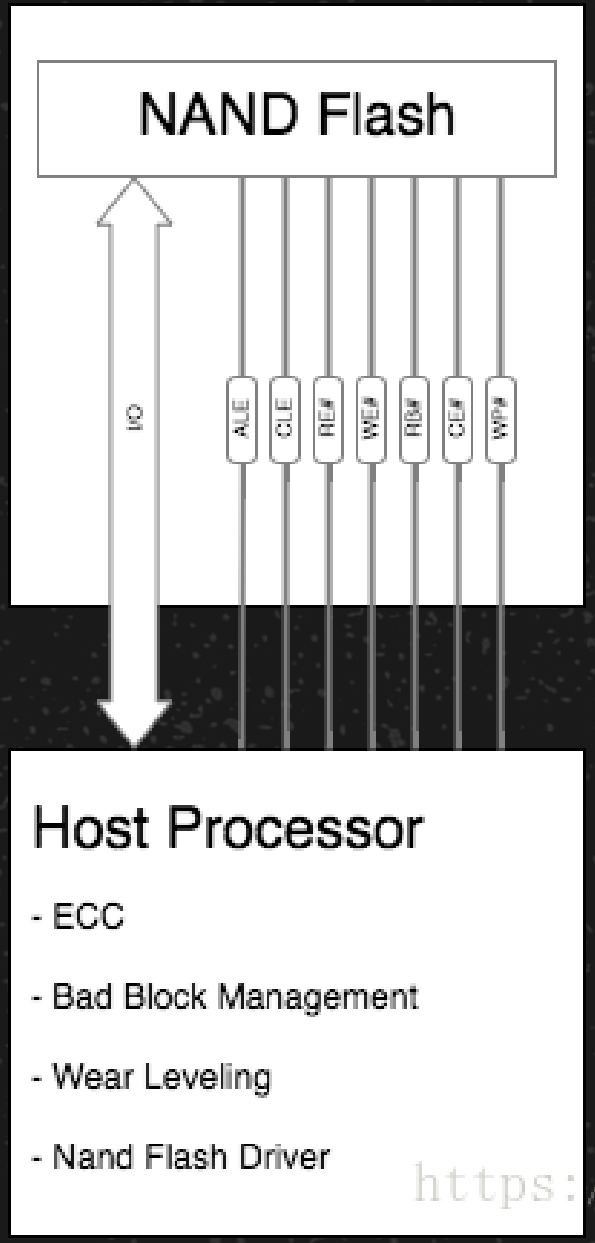


Physical Dump-EMMC/EMCP

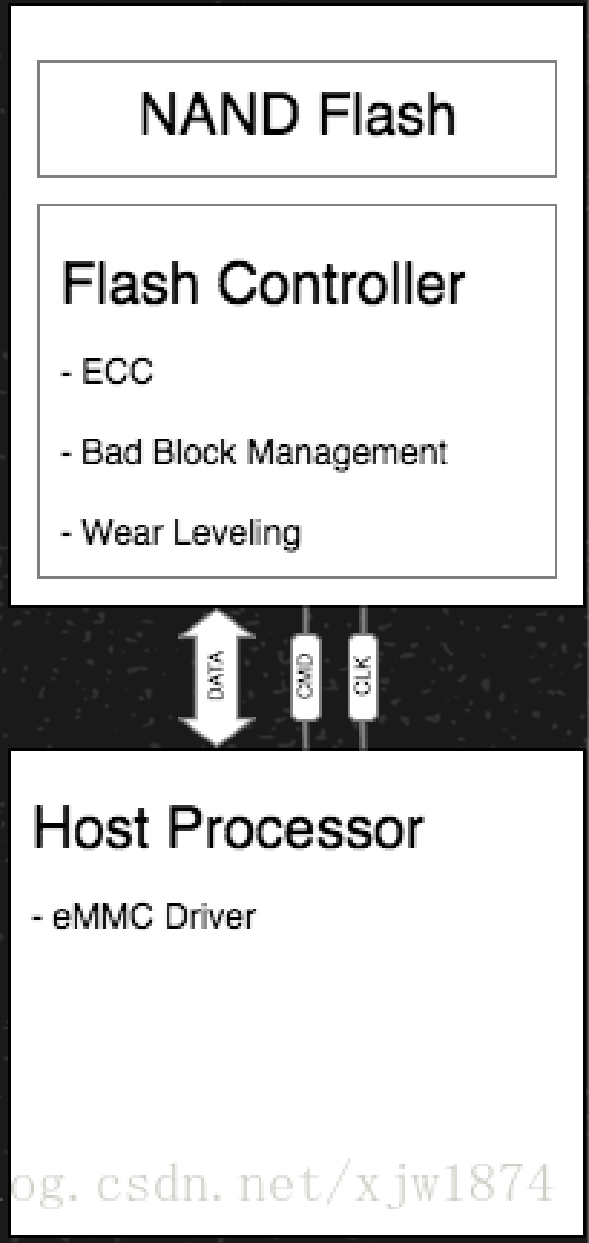
The relationship between EMMC and Nand Flash

EMMC=NAND Flash+Flash Control Chip+Standard Interface Package



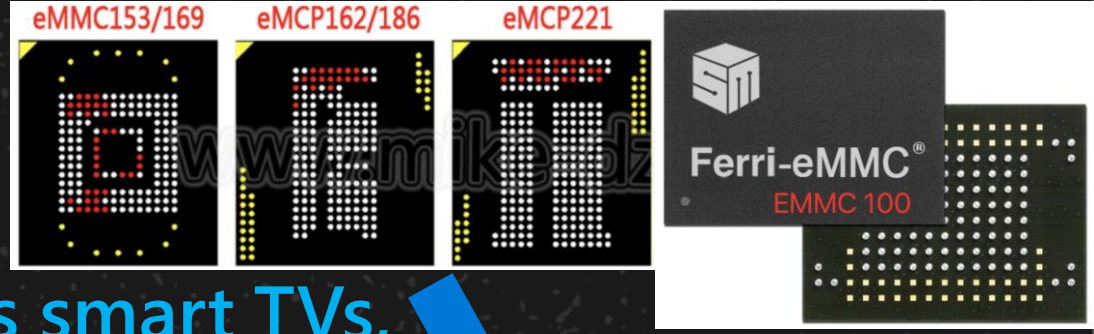


Nand Flash



EMMC

Physical Dump-EMMC/EMCP



EMMC/EMCP (for complex devices such as smart TVs, mobile phones)

Similar to an SD card

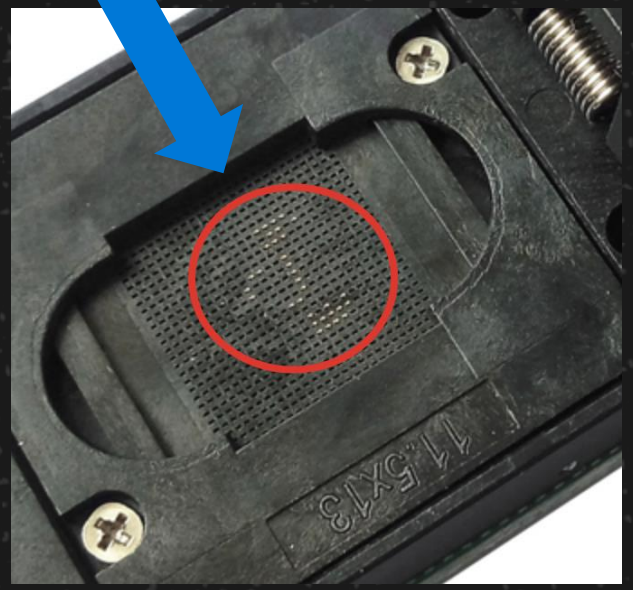
All in BGA package, specification 100/153/162/169/186/221 (account for 95%)

Offline read and write:

- Use hot air gun for arc blow (be careful about the surrounding components), special reader + programmer OR read directly from the chip
- The tinning method could be challenging so it requires more practice.

Online read and write (soldering is not needed):

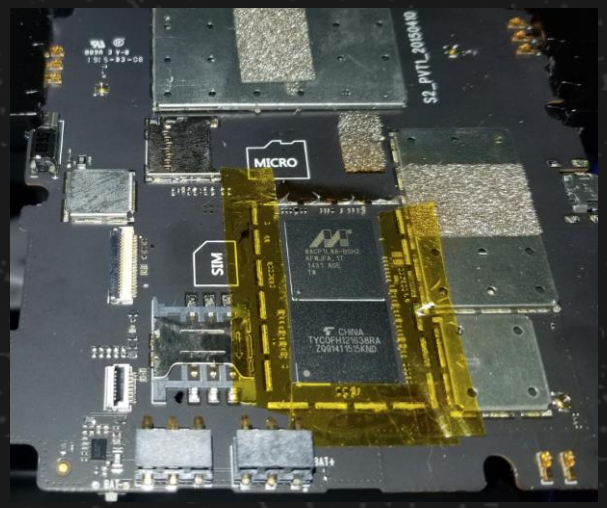
- Need to find or understand key solder joints, which are tiny/. Soldering can be challenging here (identify a good method?)
- Jump wire DAT0, CMD, CLK, GND, (VCC, VCCQ) to the SD card reader. De-soldering is not necessary but will need to pay attention not to to short the crystal oscillator



The acquired firmware can be recognized by the file system and is easy for read and write.

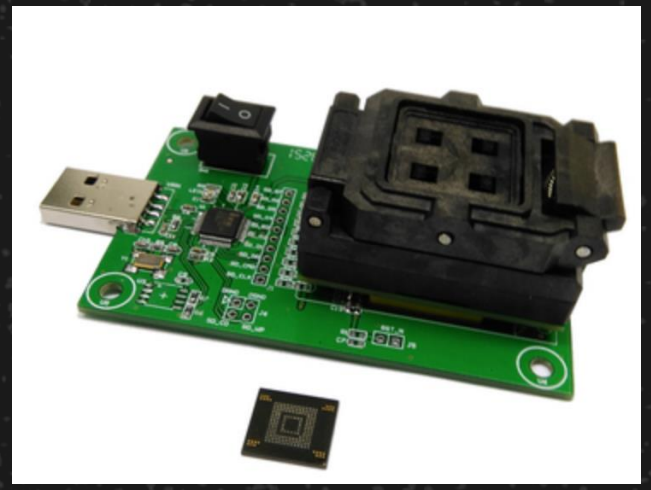
Physical Dump – Video of the Tinning Process

Offline Read and Write

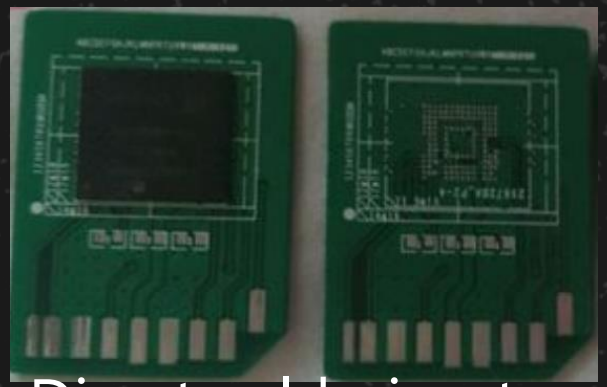


Protect Chip

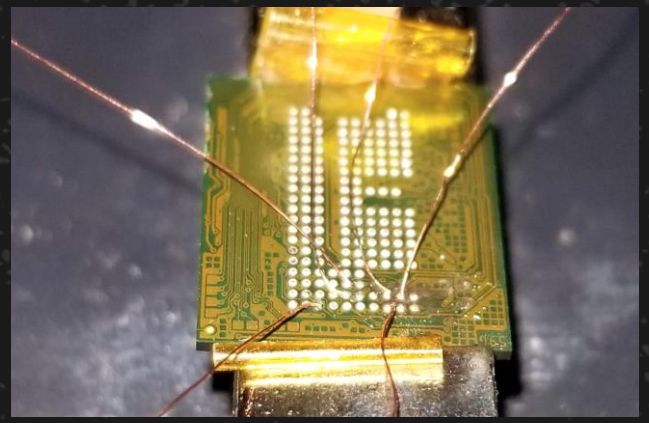
Use the programmer



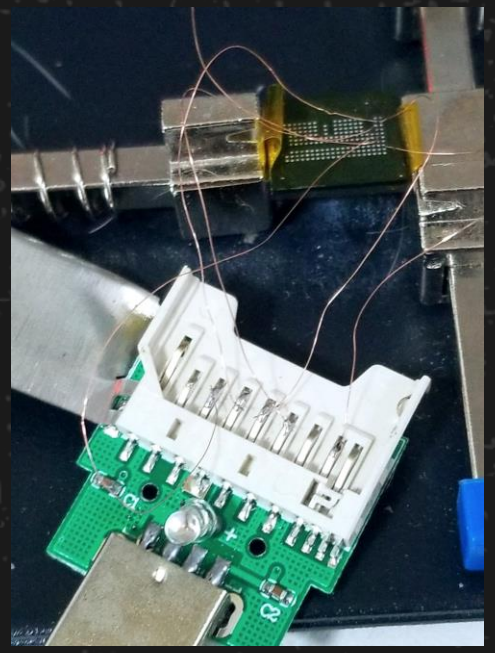
Use a dedicated reader stand



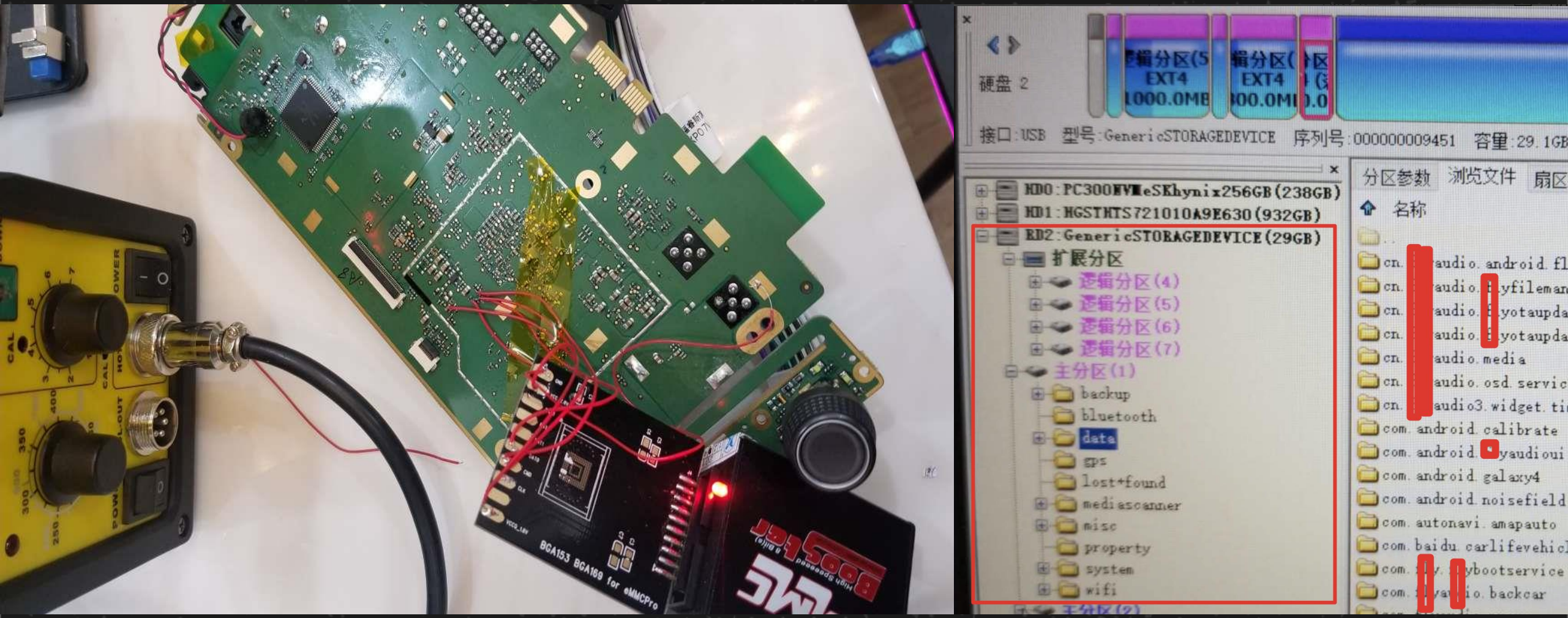
Direct soldering to the SD card



Jump wire on the memory chip to connect the SD card reader



Online Read and Write



Online read and write of a car system

Preparation-Getshell

Purpose：

- tar extract files (firmware) after running Getshell and analyze bin & web script
- Easy to review the data, including ports, processes, networks, and files
- Build a test environment and compile test tools
- Convenient online debugging (sometimes it’s hard to run QEMU offline due to the lib library and hardware limitations)

In short, the ideal environment for cracking is to enable getshell

- If you are able to remove getshell, it means you have cracked the device

Preparation-Getshell

Method：

Scan the port for telnet, ssh, adb services, etc.

- Use fast scan, masscan, nmap -sS
- Password & hash can be found in the firmware, offline cracking (how to speed up the crack?)

lsusb, check whether the usb adb is enabled

Look for web upload vulnerabilities, command injection vulnerabilities, etc.

Online or offline memory modificaiton

- For example, add busybox telnetd -l /bin/sh & to init startup items
- Easy to modify for EMMC memory structure

Look for TTL pins on the board

- Obviously labeled
- Use CPU datasheet

```
7[redacted]Bk:y2[redacted]Um
```

破解结果

```
Session..........: hashcat
Status...........: Cracked
Hash.Type........: descrypt, DES (Unix), Traditional DES
Hash.Target......: 7H[redacted]Bk
Time.Started.....: Thu Apr 11 16:17:27 2019 (1 day, 4 hours)
Time.Estimated...: Fri Apr 12 21:03:28 2019 (0 secs)
Guess.Mask.......: ?2?2?2?2?2?2?2?2 [8]
Guess.Charset....: -1 Undefined, -2 ?l?d?u, -3 Undefined, -4 Undefin
Guess.Queue......: 1/1 (100.00%)
```

使用显卡进行hash破解

Use hashcat to crack ssh, telnet password

```
➜ ~ sudo nmap 192.168.43.94 -p 1-20000 -T5 -PN
Password:
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-18 18:11
Warning: 192.168.43.94 giving up on port because retransmiss
Nmap scan report for android-8222553185129195 (192.168.43.94
Host is up (0.013s latency).
Not shown: 19987 closed ports
PORT      STATE    SERVICE
2357/tcp  filtered unihub-server
6216/tcp  filtered unknown
7014/tcp  filtered microtalon-com
8503/tcp  filtered lsp-self-ping
8663/tcp  filtered unknown
10001/tcp open     scp-config
10002/tcp open     documentum
```

Quick scan with nmap

```
➜ ~ masscan 192.168.225.1 -p 1-65000 --rate=800

Starting masscan 1.0.4 (http://bit.ly/14GZzcT) at 2019-04-17 09:56:26 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65000 ports/host]
Discovered open port 38888/tcp on 192.168.225.1
Discovered open port 80/tcp on 192.168.225.1
Discovered open port 53/tcp on 192.168.225.1
Discovered open port 28888/tcp on 192.168.225.1
```

Quick scan with masscan

Preparation-Getshell

Method：

Modify the Bootloader startup parameters

- Force the uboot configuration mode and modify the kernel parameters. For example, add <space> 1 to enter single-user mode.
- Modify kernel parameters using the JTAG interface

```
[    2.603121@0] Freeing unused kernel memory: 320K
(none) login:
(none) login: root
login[1]: root login on 'console'
-sh: can't access tty; job control turned off
[    9.744360@1] meson_uart ff803000.serial: ttyS0 use xtal(8M) 24000000 change 115200 to 115200
# [   11.268772@0] random: fast init done

# cat /etc/hostname
buildroot
```

```
] Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 20320
] Kernel command line: root=/dev/mtdblock1 mem=80M console=1 rootfstype=squashfs user_debug=31 init=/bin/sh
] PID hash table entries: 512 (order: 9, 2048 bytes)
```

```
0.000000] Kernel command line: root=/dev/mtdblock1 mem=80M console=1 rootfstype=squashfs user_debug=31
0.000000] PID hash table entries:
```

Preparation-Getshell

Method：

Modify kernel parameters using JTAG to get shell

- The device needs a JTAG port, and a corresponding JTAG device and CPU configuration file.
- OpenOCD is a good software option, which can support a variety of CPUs. As to hardware, jlink is a good option.
- Modify startup parameters
  - Find the location of the boot parameters in the firmware
  - Add a breakpoint
  - Modify the startup parameters, such as adding <space> 1 to enter single-user mode.
  - Boot the kernel, the console serial port gets the shell

Preparation - Get Communication Data

Purpose：

- Understand the workflow logic (to support analysis, such as looking for encrypted code based on http request)
- Get cookie, token and other authentication information, and sensitive private data
- Get the server interface to infiltrate the server (authorization penetration)
- Intercept and modify packet or construct a replay based on the known packet
- Finally, issue legal instructions, construct a poc, get critical keys, etc.

Usually uses Wifi/234G/Bluetooth/Low Power Bluetooth/Infrared/Wired/Other Band Radio

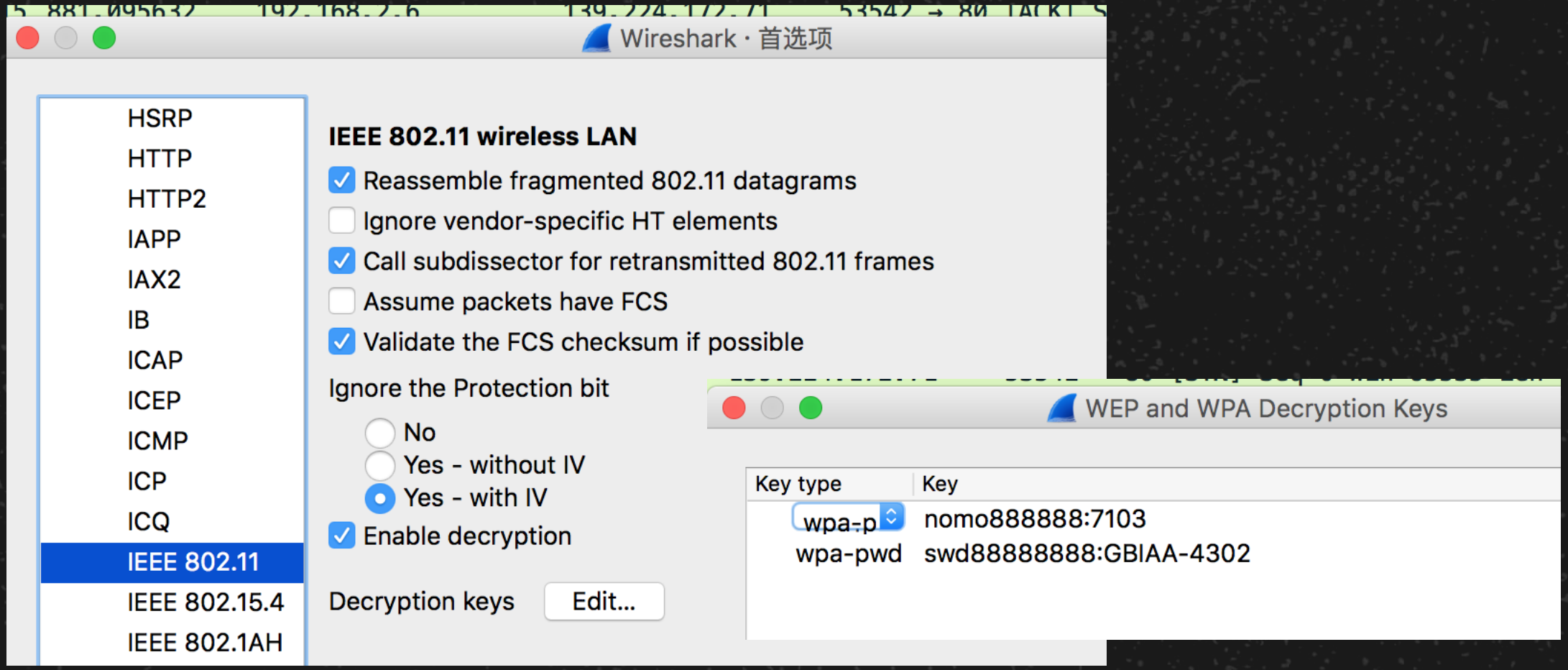
Preparation - Get Communication Data

Method: IP data (TCP, UDP, HTTP, MQTT, etc.)

WiFi:

- Real-time wireshark:
  - Turn on the wireless hotspot and connect. Wireshark will sniff this network card directly.
  - Android adb forward + tcpdump + pipe to PC wireshark
- Capture packets on the routing device
- If it is an Android APP, run directly in the native emulator, sniff this network card
- If it is HTTP, HTTPS, set proxy
- Cross compile tcpdump (arm, mips), -A option or –w
- If it is a remote device, try arp middleman
- If it is a remote device and nd need subtle action: try WiFi real-time decryption (strong network card support, such as RTL8812U))

Set WPA/WPA2 Real-Time Decryption



If Commutation Data is Encrypted

SSL/TLS encryption

- HTTPS agent
- If you verify the certificate, import the burp root certificate
- Android:
  - Xposed bypass forces certificate verification bypass
  - Hook (okhttp)

```
var send_data = {};
send_data.time = new Date();
send_data.txnType = 'HTTP';
send_data.lib = 'com.android.okhttp.internal.http.HttpURLConnectionImpl';
send_data.method = 'getInputStream';
```

Hook the commonly used http operation library okhttp

```
var send_data = {};
send_data.time = new Date();
send_data.txnType = 'Crypto';
send_data.lib = 'javax.crypto.Cipher';
send_data.method = 'getInstance';
```

Hook the java encryption and decryption library crypto

Symmetric encryption such as AES\DES, using TCP transmission

- Reverse analysis APP, binary, get the key
- Android: Hook (Crypto)

About Hook (for Android)

Framework

- Xposed:
  - Hook is supported at Java level only
  - Good for batch deployment
- CydiaSubstrate:
  - Support java/native
  - Not open source and there's no update to support new Android system
- Frida:
  - Good for cracking
  - Support java/native, support multi platforms, and can adapt to the latest system

Tools

Integrated http, encryption and decryption, SQL query, file operation, IPC, and custom hook function

- Xposed based：
  - Inspeckage
  - https://github.com/ac-pm/Inspeckage
- Frida based：
  - appmon
  - https://github.com/dpnishant/appmon

What to Hook?

Sensitive operation hook

- Symmetric encryption key, plaintext ciphertext
- Sqlite query (determine whether there is injection and help understand the logic)
- HTTP, HTTPS request content
- Hash call status
- Other (webView, serialization, file system operations, SharedPreference, IPC, etc.)

Target function custom hook

- Get the return value
- Modify the return value
- How to determine the class, method(trace) of a hook

Package Information | Shared Preferences 257 | Serialization | Crypto | Hash 525 | SQLite 2404 | HTTP | File System | Misc. | WebView

IPC 92 | + Hooks

Clear

721 HttpURLConnection: http://pzb.winpos.mve.cn/mpos-api-disasterrecovery/mpos/api/disasterrecovery/261353-103/record/shoplogup2
720 HttpURLConnection: http://b.mvee.cn/services/b_api_log/addClientRunningInfo
719 HttpURLConnection: http://b.mvee.cn/shop/queversion.php?VerCode=276&Type=57&shopId=261353&DeviceID=8XV7N15A30002497
718 HttpURLConnection: http://b.mvee.cn/shop/queversion.php?VerCode=276&Type=57&shopId=261353&DeviceID=8XV7N15A30002497
717 HttpURLConnection: http://b.mvee.cn/shop/queversion.php?VerCode=276&Type=57&shopId=261353&DeviceID=8XV7N15A30002497
716 HttpURLConnection: http://pdd.winpos.mve.cn/posapi/shop/261353-103/shopapi/modifyXmppState
715 HttpURLConnection: http://pdd.winpos.mve.cn/posapi/shop/261353-103/shopapi/getuntreatedorder
714 HttpURLConnection: http://pdd.winpos.mve.cn/posapi/shop/261353-103/shopapi/express/pub/shop/search
713 HttpURLConnection: http://pdd.winpos.mve.cn/posapi/shop/261353-103/shopapi/getuntreatedorder
712 HttpURLConnection: http://pdd.winpos.mve.cn/posapi/shop/261353-103/shopapi/getExchangeListBystate

Use Inspeckage Hook http request

IPC 27 | + Hooks

New hooks | Replaces | Clear

6 {"class":"cn[redacted]","method":"getAccount","result":"\"pphappy\""}
5 {"class":"cn.pospal.www.mo.PospalAccount","method":"getAccount","result":"\"pphappy\""}

Customize hooks with Inspeckage

127.0.0.1:8008

Download | OFF | LogCat | clipboard | App is running: true | Module enable: true

[redacted]

2.4.5.276

UID: 10094 | Debuggable: false

GIDs: 3002-3003-3001

Allow Backup: true

Package: com.[redacted]

Data dir: /data/user/0[redacted] Tree View

Package Information | Shared Preferences 257 | Serialization | Crypto | Hash 525 | SQLite 2404 | HTTP | File System | Misc. | WebView

IPC 92 | + Hooks

Clear

11 SecretKeySpec(eO4bfYqGfzRpTjdEwXyvuw==,AES) , Cipher[AES/CBC/PKCS5PADDING] (NcJWkqmflS6s7I7PdrPSTNRNyIi7uCYTnQvi4K+zWM3ijSGoQdB5anKGoY
JfcDP74NkMZ2nk5aP1/oC/BeGx0zl11jJpyGxNZ8Q5I1byNDieAqiWV8ng+y64BqcJgPEavr5hJVBRvkxN2JvOa+tEB7os/NMNfzbvpJffw6P37NRQCh1TK9cwbtNJEpCYU4PAbXu
U1RUL9wu8TbjrbTR8EiQatSKAXRpOZBX5WpMrNLbchfIASLNdoja4bwGIIFICZz0s/3C+9nplUBqBf7HKu95GCV03UCVNZBcCALkpeP67H/fgFpxUCFMZnAwW2Xv12EcTlYiYCai
rYV0jLkPrRi86HgKTAMMhuEoZxKMtBm3a9bnIjVUrNXkX/Blv5uBvqoFhD9mXfHUMcSV08+fzdXIHJLa2/011FgZJSivRQ1XfEJD7639ad4xok8rjgRk3wgjgTZLHS1FzUpHiqonDaFx
etNjE8kyiwmjz7jUY+kZiU3Pt5TUgyUbvat0uPxN220VeKkN8YS/wvkC2kpAOX1EeB65mrt0JmHJat21pc2GKlI9VBoeFKHmSz7NDMpduadk48/HjWYauQWZu99MB1n6j14y/NOmd
23nox7D+oTGCoTSN6yEpwp0loSrlU0pkQ4j64zguNWjL+PT6egXVNXi9dGjL4w1RGD1cp5XbJwuIPVdO+W4o5HPVfJoyl09B/c/LhBCc1tLeg1IVhEqyNiicVV+Xgu8shpC+d/wjrsoV
KFwoCDqOIQ434MHtse4Uz81gbdnjmWGShlUJ+S6CJdqrrDUDfrf0dZs+ifuphcA/OfV9s5OpGAFtMA1O139mdfuRQLaODy1NSmGc/3MY0zdP6knUojdqdz302Tr+IdBBCKonJzM
SecddoWgG1nlgCLXx9vHV9fzUyB+bBhZvjZjskNjVJysct9zAwagFSN9LBDYus6Alrei41Om7/7LkYBCqiRx9SY/pjtbPB7ozjELIM1v0am1UxRPvlTWFjsihYNvowpiRG+hvFGOZ2gxE
ViyAlzDkwdstUrJawONEj3FEOI6opylwED/be2/VbjfWxCj2aUeo5TiXHbwS6MVI4/5DvJzxKlrzsycTYmR5733trbj7n2zXg8a6UoskOhtdXebheQQljatS4f0lbz8QyEY8bctU94W78bjah
Sg6yKf0XxpTSdR7p6xkUs2m3DA= , {"code":0,"data":{"areaMessage":{"1":0,"all":0},"hintTableList":[],"tableStatus":{"1":{"fdExpAmt":0,"fiCustSum":1,"fioccupyflag":2,"fiopenjo
b":0,"fisharebills":0,"fistatus":1,"fiwxmsgflag":0,"flag":0,"fsmareaid":"1","fsmtableid":"1","fsmtablesteid":"2","fsopenhstime":"2018-08-13 19:25:20","fsopenusernam
e":".....","fssellno":"201808070003","fsupdatetime":"2018-08-13 19:25:20","fsupdateuserid":"admin","fsupdateusername":".....","hasPayInfo":0,"lockedHostId":"","lockedS
tatus":0,"lockedUserID":"","lockedUserName":"","ordersource":0,"prePayFlag":0,"prestatmentstatus":0}}},"head":{"device":"8XV7N15A30002497","dv":"2018-08-13 20:20:1
3","exe":0,"hd":"Cashier","ot":"5564f0eb-398a-43c9-98a2-841d87bda1b11808140812298","requestId":"table/refreshTableBizData_1534208645037","shopid":"261353","us":"3
f25811c-e032-4ffd-877f-86dd6bbf9f5e1808140904035","version":100},"message":"....."})

Encrypted with Inspeckage hook AES

Get Communication Data - Other Channels

234G：

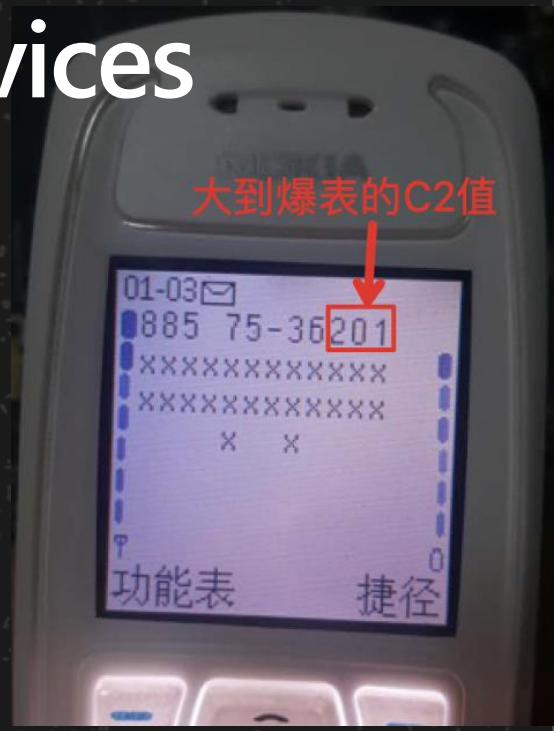
- Devices that require remote communication, such as vending, shared bicycle locks, etc.
- Developers often think that this channel is very secure and rarely take hardening into consideration.
- Through the fake base station GPRS hijacking, the network traffic connected to the base station can be completely controlled.
- Remote access triggering vulnerability can also be performed according to carrier network interoperability issues

Bluetooth：

- At this stage, mainly based on low-power Bluetooth, such as sports bracelets, smart thermometers, Bluetooth unlocking, etc.
- Developers generally think that this channel is very secure, and there are few hard-to-reinforce reinforced, and there are many key leaks.
- Traditional Bluetooth analysis can only track broadcast packets, and cottage devices can track frequency hopping.
- Mobile phone debugging mode to enable Bluetooth log, simple and stable

Traffic Access, Sniffing, MITM for 234G Devices

- Fake base stations can be a risk for 2G networks as the mobile phone is unable to authenticate the base stations
- Build a GSM base station system (tested legally)
  - Hardware: Bladerf (higher accuracy in comparison with other SDR devices)
  - Software: YateBTS (graphical interface / easy to install)
- How to auto connect a smart device to a fake base station
  - Same thought process as sending text message to a cellphone from fake base station: increase cell reselection parameters C1, C2
  - Modify the YateBTS source code
  - For details, please refer to my Defcon China talk
- Attack: Get traffic, MiTM, access port trigger....
- Other simple methods:
  - The intranet of carrier is interconnected (10 or 172 network segments). Simply purchase two sim cards to trigger port-based vulnerabilities.



GSML3RRElements.cpp

```
48
49
50  L3SI3RestOctets::L3SI3RestOctets()
51      :L3RestOctets(),
52      mHaveSI3RestOctets(false),
53      mHaveSelectionParameters(false),
54      mCBQ(0),mCELL_RESELECT_OFFSET(0),
55      mTEMPORARY_OFFSET(0),
56      mPENALTY_TIME(0),
57      mRA_COLOUR(0),
58      mHaveGPRS(false)
59  {
```

Step 2 - Analysis

Combine existing files, web requests, shells

- Use Netstat –tunlp to monitor and analyze the corresponding process
  - Command injection, such as the contents of fopen() can be controlled
  - Dangerous functions may cause overflows, such as strcpy()
- If shell is not an option, use port scan or stateless scanning
- If web is available:
  - Determine configuration files and web source files
  - Try web page vulnerability mining (php, cgi, lua scripts, etc.)
- Locate critical code locations (Reverse compilation, keyword, trace) based on network access to obtain the encryption logic and interface parameter format

Finally get key data, or issue instructions

Example: Core Logic Problem of A Vending Machine

例如 FTP 泄露，里面包含大量其他配置文件（通过逆向协议获知 ftp 下载 ulr、用户名密码）

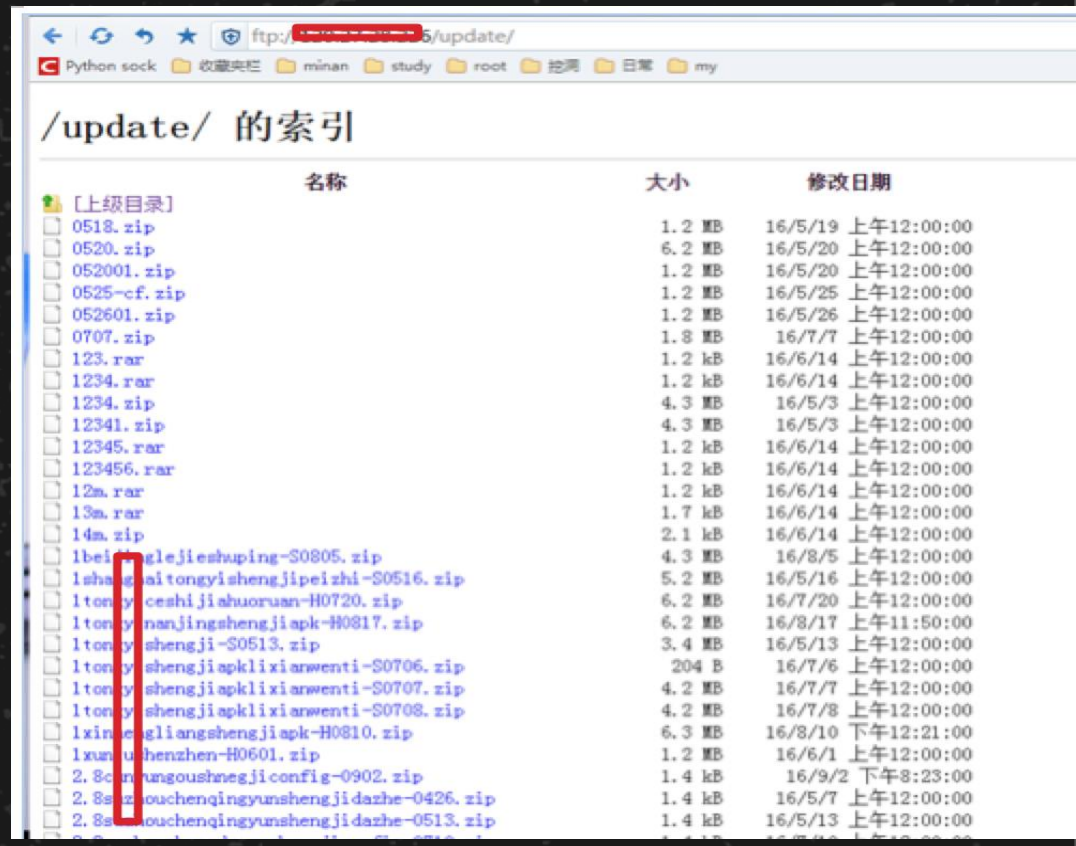
```
root@ubuntu-14:~# nc [redacted] 6001
[redacted]
05710031ftp://etftp:yichu[redacted]6/update/[redacted].zip
```

The FTP upgrade server password is leaked in the Bin file.

```
public void zfbnotify()
{
  try
  {
    String sign = this.request.getParameter("sign");
    String sign_type = this.request.getParameter("sign_type");
    String gmt_create = this.request.getParameter("gmt_create");
    String seller_email = this.request.getParameter("seller_email");
    String seller_id = this.request.getParameter("seller_id");
    String quantity = this.request.getParameter("quantity");
    String notify_action_type1 = this.request.getParameter("notify_action_type");
    String notify_action_type = this.request.getParameter("trade_status");
    String out_trade_no = this.request.getParameter("out_trade_no");
    String trade_no = this.request.getParameter("trade_no");
    String price = this.request.getParameter("price");
    String total_fee = this.request.getParameter("total_fee");

    this.log.info("接收到支付宝通知:" + out_trade_no);
    Orderform orderform = this.orderformService.getByOrderFormId(out_trade_no);
    ZFBPay zfbpay = new ZFBPay();
    if (orderform == null)
    {
      orderform = new Orderform();
      orderform.setAmount(Integer.valueOf((int)(Float.valueOf(total_fee).floatValue() * 100.0F)));
      orderform.setDanjia(Integer.valueOf((int)(Float.valueOf(price).floatValue() * 100.0F)));
      orderform.setDazhe(Integer.valueOf(100));
      orderform.setOrderid(out_trade_no);
      orderform.setMachineid(Integer.valueOf(out_trade_no.substring(0, 8)));
      orderform.setBorderid(trade_no);
      orderform.setAppid(seller_id);
      if (notify_action_type.equals("WAIT_BUYER_PAY"))
      {
        orderform.setStatus("1");
        DBTools.MYMap.put(orderform.getOrderid().toString().substring(0, 22), "1");
      }
      else if (notify_action_type.equals("TRADE_SUCCESS"))
      {
        orderform.setStatus("2");
      }
```

Payment service not verified, resulting in 0 CNY payment



Can control other vending machines to update firmware arbitrarily

Information Disclosure and Configuration Modification for Two Smart Watches


```
..{"Version":"00030000","SN":1074096116,"CID":10211,"PL":
{"Name":"863412030[redacted]","Password":"7805303461C5E33FC8[redacted]","Ty
pe":200,"machSerialNo":"15183/00035289"}}..{"RC":
1,"Version":"00030000","SN":1074096116,"PL":
{"EID":"B0C2116A04126B9122919095E6BA24FD","BIND":0,"GID":
["888B31A71E5B75376A97EBD8A0010429"],"GMT":"20170501184949080"},"CID":
10212,"SID":"89D4229CEB9C4128A0DC45F20BE7399B"}..{"CID":
80041,"Version":"00030000","SN":
```


Key was disclosed during the cloud login process


```
10463   http://watch.okii.com   GET   /smartwatch/watchinit/b5c2dc7065...   ☑   ☐   200   2342   JSON

Request | Original response | Auto-modified response
Raw | Headers | Hex
Content-Length: 2162

{"code":"000001","desc":"success","data":{"guardSwitch":0,"consOnceTime":1,"dialPlateInfo":{"dialPlateBuildTime":1493909574,"dialPlate":
"digit_acaleph"},"schoolTime":"{\"week\":31,\"morningStart\":\"08:00:00\",\"morningEnd\":\"11:30:00\",\"afternoonStart\":\"14:00:00\",\"
afternoonEnd\":\"16:30:00\"}","conTime":2,"homeTime":"\"18:00:00\"","classmode":[{"id":10882317,"classId":"ed3009a56a1a4b8db61afe535dc50
9a9","watchId":"b5c2dc7065cc46bf90e602af13e1589808955307","title":"禁用时间段","classSwitch":0,"amTime":"{\"startTime\":\"08:00:00\",\"e
ndTime\":\"11:30:00\"}","amSwitch":1,"pmTime":"{\"startTime\":\"14:00:00\",\"endTime\":\"16:30:00\"}","pmSwitch":1,"nmSwitch":0,"classWe
ek":31,"type":0,"createTime":1493888530000}],"autoRecordSportTime":3600,"guards":[{"rate":300,"start":"07:30:00","end":"08:40:00","schoo
lWeek":31},{"rate":300,"start":"16:23:00","end":"18:10:00","schoolWeek":31}],"contactMobiles":[{"contactId":"b58ea4d6ad324c96852cd6744c1
bbc3b","mobileId":"077e6fd1473842deb318759802ab6c29","imAccountInfo":{"accountId":"077e6fd1473842deb318759802ab6c29","imAccountId":66529
961,"singlePushDialogId":183898509}}],"curTotalSteps":0,"guardSwitchWifi":0,"lastestSportLogTime":1494604800,"imHeartConf":"{\"minHeart\
": 30,\"maxHeart\": 240,\"curHeart\": 240,\"heartStep\":
10}","passiveRecordSportTime":300,"sportTimeSliceSize":300,"respTime":{"startResponseTime":360,"endResponseTime":1350},"qnURL":{"upL":"h
ttp://uptx.qiniu.com:80","downSY":"http://bbksmartwatch.qiniucdn.com:80","downSL":"http://bbksmartwatch.qiniucdn.com:80","downGT":["http
://smartwatch.qiniucdn.com:80"],"upT":["http://uptx.qiniu.com:80","http://up.qiniu.com:80"],"downST":["http://bbksmartwatch.qiniucdn.com
:80"],"downGY":"http://smartwatch.qiniucdn.com:80","upY":"http://upyd.qiniu.com:8888","downGL":"http://smartwatch.qiniucdn.com:80"},"wat
erStopTime":600,"legalHolidaySwitch":0,"consCounts":2,"contacts":[{"id":"b58ea4d6ad324c96852cd6744c1bbc3b","mobileNumber":"19999999999",
"type":0,"salutation":"爸爸","status":1,"hfars":261644}],"consTotalTime":30},"pushError":{"code":0,"identify":"451bea68cd2d4064ac1da2cda
16c377e","error":[]},"errorSN":null,"serverGreyCode":null}
```


Configuration was modified by MiTM during interaction in the cloud

Information Leakage and Decryption in A Lock for Shared Bicycle

```
POST /gsmlock HTTP/1.1
Host:[redacted]fo.so
Content-Type: text/plain
Content-Length: 116
Cache-Control: no-cache

qlUAAQBPMUhCMTcxNDAzOTExAYA4r5S1QtHfGB7GxD44V9bml02K5xSxAhz3Mg2z00Lem9qIXY6eo
LEPTdYTRrEfQHT7EyG6TUPhBay44z5BeawX80YQHTTP/1.1 200 OK
Date: Fri, 04 Aug 2017 22:11:53 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 76
Connection: keep-alive
X-Powered-By: Express
ETag: W/"4c-o+ijHq59dUwrBdxcvODqDQ"

qlUAAQAvMUhCMTcxNDAzOTExAYAe+B0mKUwiwOgirvY5Sax1sLQy45XaioAu38M6gJxkpyYnAQ==
```

Encrypted cloud transmission

```
29      def decrypt(self,txt):
30          key = hashlib.md5('[redacted]' + self.devide_id).hexdigest().decode("HEX")
31          print key
32          cryptor = AES.new(key, self.mode)
33
34          plain_text = cryptor.decrypt(txt)
35          return plain_text
36
37  aes_encrypt = AES_ENCRYPT()  # 初始化密钥
38  aes_encrypt.devide_id = '1HB1715[redacted]'
39
40  print aes_encrypt.decrypt('d31d1ef31dcebad585ae71bd5a3c0365d66b5d6de92b320b18bd32cc6d332313'.decode('hex'))
41
42
```

key

```
Run aes_test
/opt/local/bin/python /Users/gaoshupeng/PycharmProjects/work/ofo/aes_test.py
n;?"??'?7 vG~Wy
□?1234□□□□□□□□□Y? NY? N□□□□□□□□
```

解锁密码

Analyze firmware, get keys and upgrade agreements

FTP Server Protocol Command Injection in a Communications Module

```
20  addr_len = 16;
21  v3 = accept(dword_15230, &addr, &addr_len);
22  if ( v3 == -1 )
23  {
24    perror("accept error");
25  }
26  else
27  {
28    memset(&s, 0, 0x64u);
29    memset(&v10, 0, 0x64u);
30    v4 = getcwd(&s, 0x64u);
31    snprintf(&v10, 0x64u, "ls -l %s", v4);
32    v5 = popen(&v10, "r");
33    if ( v5 )
34    {
35      printf("pipe open successfully!, cmd is %s\n", &v10);
36      while ( 1 )
37      {
38        v6 = fgetc(v5);
39        putchar(v6);
40        write(v3, &v6, 1u);
41      }
42    }
43    puts("pipe open error in cmd_list");
44  }
45 }
```

Some Essential Skills and Tips

- Soldering skills
  - Soldering, de-soldering, drag soldering, tinning, board washing, non-welding & de-soldering
  - Use hot air gun for de-soldering, ball planting tin (low temperature tin paste)
  - Jump Wire
  - Buy genuine white soldering iron with adjustable temperature, which takes 8 seconds to heat up without aging
- APK decompile, hook, dynamic debugging, Java code reading
- Web attack and defense and source code auditing capabilities
- Ability to code Python/Java
- Simple binary reverse analysis
- TCP, HTTP packet analysis by using Wireshark
- Familiar with cross-platform cross compilation

Some Essential Skills and Tips

- Common tools:
  - Prepare gdb, tcpdmp, telnetd, nmap, masscan...
  - Busybox in multiple platforms

- Common commands:
  - busybox netstat -tunlp
  - busybox telnetd -l /bin/sh &
  - tcpdump -i xxx not tcp port xxxx -A
  - nmap -sS -PN -T5

Q&A