

针对智能设备漏洞挖掘的 一些新方法

小灰灰

百度安全实验室 高级安全研究员

30-May2019



关于我

百度安全实验室 高级安全研究员

研究领域：IoT安全/AI安全/无人车安全

多次破解硬件设备

之前负责：

- BSRC、应急处置、0day分析
- 百度产品 安全评估
- 百度安全监控体系建设

传统IOT设备的破解

路由器?

固件下载

Binwalk解包

找到有问题的bin (why?)

IDA分析、WEB 脚本文件分析

漏洞验证 (真机or QEMU测试)

大多数好像都是漏洞分析



```
download.trendnet.com - /TEW-654TR/firmware/

[To Parent Directory].

7/10/2009 6:07 PM 3630038 FW_TEW-654TR(1.00B19).zip
7/6/2012 11:27 AM 3953347 FW_TEW-654TR(1.10B20).zip
8/6/2014 10:22 AM 3748124 FW_TEW-654TR(1.10B25).zip
11/10/2014 4:17 PM 3784728 FW_TEW-654TR(1.10B26).zip
11/5/2013 4:35 PM 3703540 FW_TEW-654TR(110B23).zip
1/27/2011 12:04 PM 3931512 FW_TEW-654TR_v1.0R(1.02.01).zip
2/8/2011 10:51 AM 3931577 FW_TEW-654TR_v1.0R(1.10.10).zip
6/9/2011 1:32 PM 3950445 FW_TEW-654TR_v1.0R(1.10.12).zip
5/20/2013 1:25 PM 3972371 FW_TEW-654TRv1(1.10B21).zip
5/28/2014 9:59 AM 3776636 TEW-654TRv1_(FW1.10B24).zip
```

```
root@ubuntu:~/aaaa# binwalk -Me TEW-654TRA1_FW110B12.bin

Scan Time: 2016-10-19 19:58:24
Target File: /root/aaaa/TEW-654TRA1_FW110B12.bin
MD5 Checksum: 523c7c7f158930894b7842949ff55c48
Signatures: 344

DECIMAL HEXADECIMAL DESCRIPTION
-----
54 0x40 uImage header, header size: 64 bytes, header CRC: 0xE5BE5107, created: 2011-05-30 13:00:10, image size: 883118 bytes, Data Address: 0x80000000, Entry Point: 0x802B2000, data CRC: 0xB8911044, OS: Linux, CPU: MIPS
Image type: OS Kernel Image, compression type: lzma, image name: "Linux Kernel Image"
58 0x80 LZMA compressed data, properties: 0x5D, dictionary size: 8388608 bytes, uncompressed size: 2746476 bytes
917568 0xE0040 Squashfs filesystem, little endian, non-standard signature, version 3.0, size: 2776952 bytes, 361 inodes, blocksize: 65536 bytes, created: 2011-05-30 13:00:17

Scan Time: 2016-10-19 19:58:26
Target File: /root/aaaa/TEW-654TRA1_FW110B12.bin.extracted/80
MD5 Checksum: b1f81b8c795c3dd24990d22fee8d8354
Signatures: 344

DECIMAL HEXADECIMAL DESCRIPTION
```

```
text:00409648 nop
text:0040964C
text:0040964C loc_40964C: # CODE XREF: main+474f]
text:0040964C
text:0040964C la $a1, loc_410000
text:00409650 la $t9, unk_40A07D40
text:00409654 addiu $a1, (aLoad_setting - 0x410000) # "load_setting"
text:00409658 jalr $t9, $truncp
text:0040965C move $a0, $s3
text:00409660 lw $gp, 0x2E6F0+var_2E6E0($sp)
text:00409664 beqz $v0, loc_409A3C
text:00409668 addiu $a1, $t1, (aLogin - 0x410000) # "login"
text:0040966C la $t9, unk_40A07D40
text:00409670 addiu $s0, $s3, 0x20
text:00409674 jalr $t9, $truncp
text:00409678 move $a0, $s0
text:0040967C lw $gp, 0x2E6F0+var_2E6E0($sp)
text:00409680 beqz $v0, loc_409A6C
text:00409684 move $a0, $s3
text:00409688 la $a1, loc_410000
text:0040968C la $t9, unk_40A07D40
text:00409690 addiu $a1, (aAdmin_login - 0x410000) # "admin_login"
text:00409694 jalr $t9, $truncp
text:00409698 move $a0, $s0
text:0040969C lw $gp, 0x2E6F0+var_2E6E0($sp)
text:004096A0 nop
text:004096A4 la $t9, admin_login
text:004096A8 beqz $v0, loc_409A60
text:004096AC move $a0, $s3
text:004096B0 la $a1, loc_410000
text:004096B4 la $t9, unk_40A07D40
text:004096B8 addiu $a1, (aAdmin_webtelnet - 0x410000) # "admin_webtelnet"
text:004096BC jalr $t9, $truncp
text:004096C0 move $a0, $s0
text:004096C4 lw $gp, 0x2E6F0+var_2E6E0($sp)
text:004096C8 nop
```

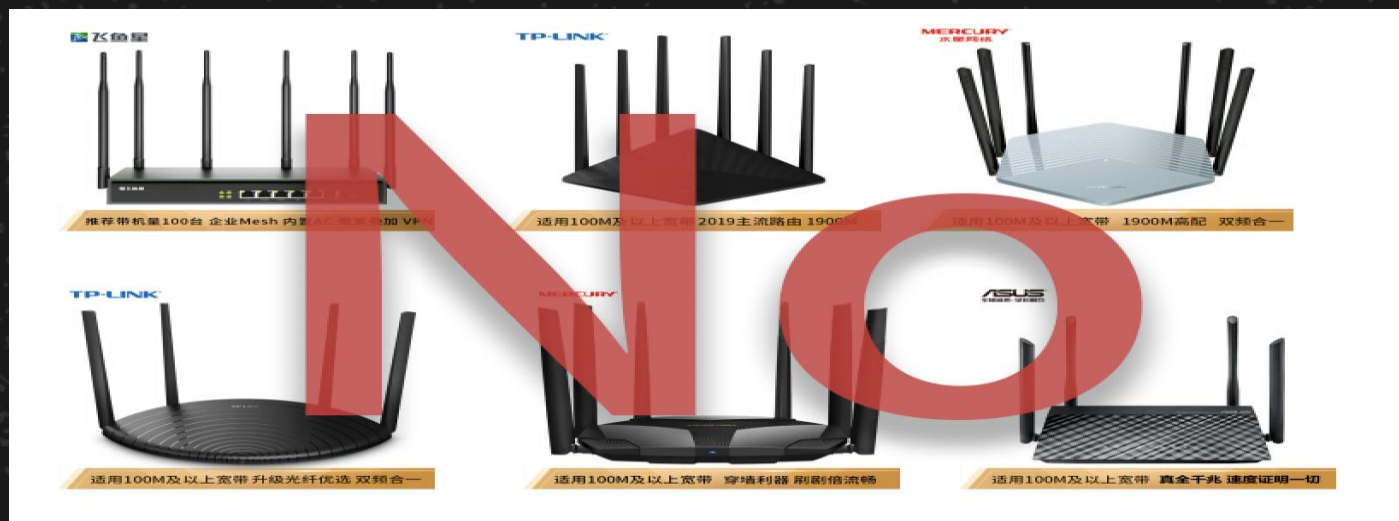
现在?

IOT设备越来越多种多样,
not only 路由器

大厂也不会让你轻易破解

- 固件不提供下载
- telnet、串口、ADB都关闭
- 甚至都无从下手

怎么办???



似乎有着相似的结构

操作系统及硬件	完整的Android、Linux发行版， ARMv5/6/7/x86处理器 EMMC/EMCP/NAND存储存储	Openwrt、精简内核的Linux，ARM、Mips处理器，NAND/SPI Flash存储	RTOS实时、精简内核的Linux，ESP乐鑫、Arduion片上系统、AVR、STM32系列，SPI Flash存储
应用场景	智能音箱、智能手表、自动售货机、电视盒子、智能电视、智能广告牌、车机	路由器、mini版智能音箱、智能摄像头	智能门锁、智能电饭煲、智能插座、智能灯、智能手环
特点	较多功能、较大的存储、易于开发APP的载体、大多有大屏幕	单一但高级功能、无需屏幕展示内容or小尺寸屏幕	功能单一简单但大多有通过网络进行简单控制，模拟电路无法实现

第0步：拆！

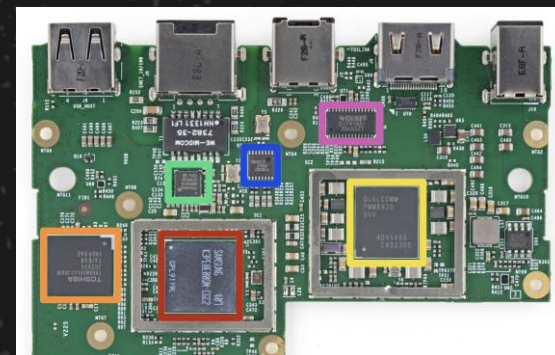
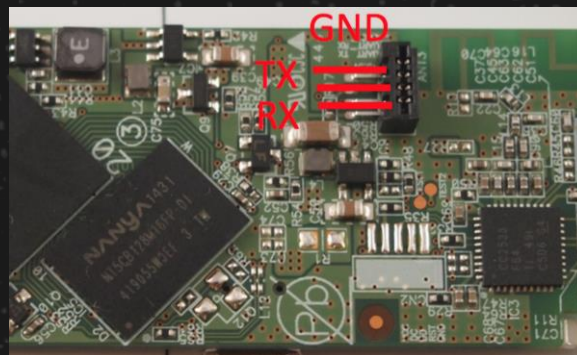
做到心中有数，减少弯路

- 有没有搞头，用了哪些芯片解决方案
- 没有思路时 哪些地方还可以搞

搜索芯片型号信息、datasheet

重点关注：

- 存储类型及规格
 - SPI Flash 8/16/宽窄
 - EMMC/EMCP 100/153/162/169/186/221/254
 - NandFlash TSOP32/40/48
- TTL及JTAG接口（如何寻找）
- 通信模块（以太、蓝牙、wifi、234G）





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

云拆解--寻找攻击目标的好方法

Google xxx teardown

论坛（拆客论坛）

Ifixit.com

- 包含著名厂商硬件设备
- 图片清晰、标注

Fccid.io

- 所有带有无线功能、在国外发行的设备
- 种类繁多
- 技巧：搜索 `site:fccid.io internal photos xxx`

特斯拉钥匙使用的主控-在fccid.io网站上搜索到

<https://fccid.io/2AEIM-1133148/Internal-Photos/Internal-Photos-3989913>

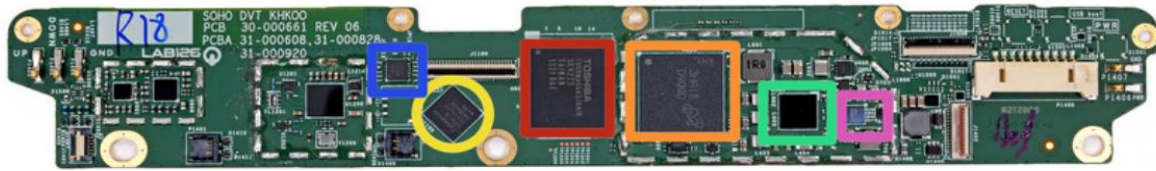


Kindle Fire 的存储结构-在ifixit网站上搜索到

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

- ① 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

第一步 准备工作：随心所欲的控制、获取

控制&获取

- 获取文件系统
- Getshell (更方便的分析, 查看网络、文件、进程)
- 获取、控制网络数据

最终根据这些已有内容, 进行综合分析, 寻找有效漏洞

Tips: 并没有完全的先后顺序, 同步穿插进行

- 例如getshell后直接就可以获取固件了, 或者dump获取固件进行修改后便getshell了
- 例如获取交互数据, 可以拿到升级连接, 直接获取固件下载地址

准备工作-获取固件

目的:

- 了解OS 及文件系统结构, 关注**关键目录** (/etc /home /usr/bin ...,如果是Android, /system/priv-app)
- 分析启动脚本 (/etc/inittab /etc/init.d), **加载的二进制**文件以及配置文件
- 分析**web目录**文件 (CGI、PHP、Lua.....)
- 方便恢复到老版本系统 (例如开启了telnet) , 分析更方便
- 固件也可能是新版本APK, 逆向分析之
- Chroot到对应处理器的QEMU, 方便分析二进制&web

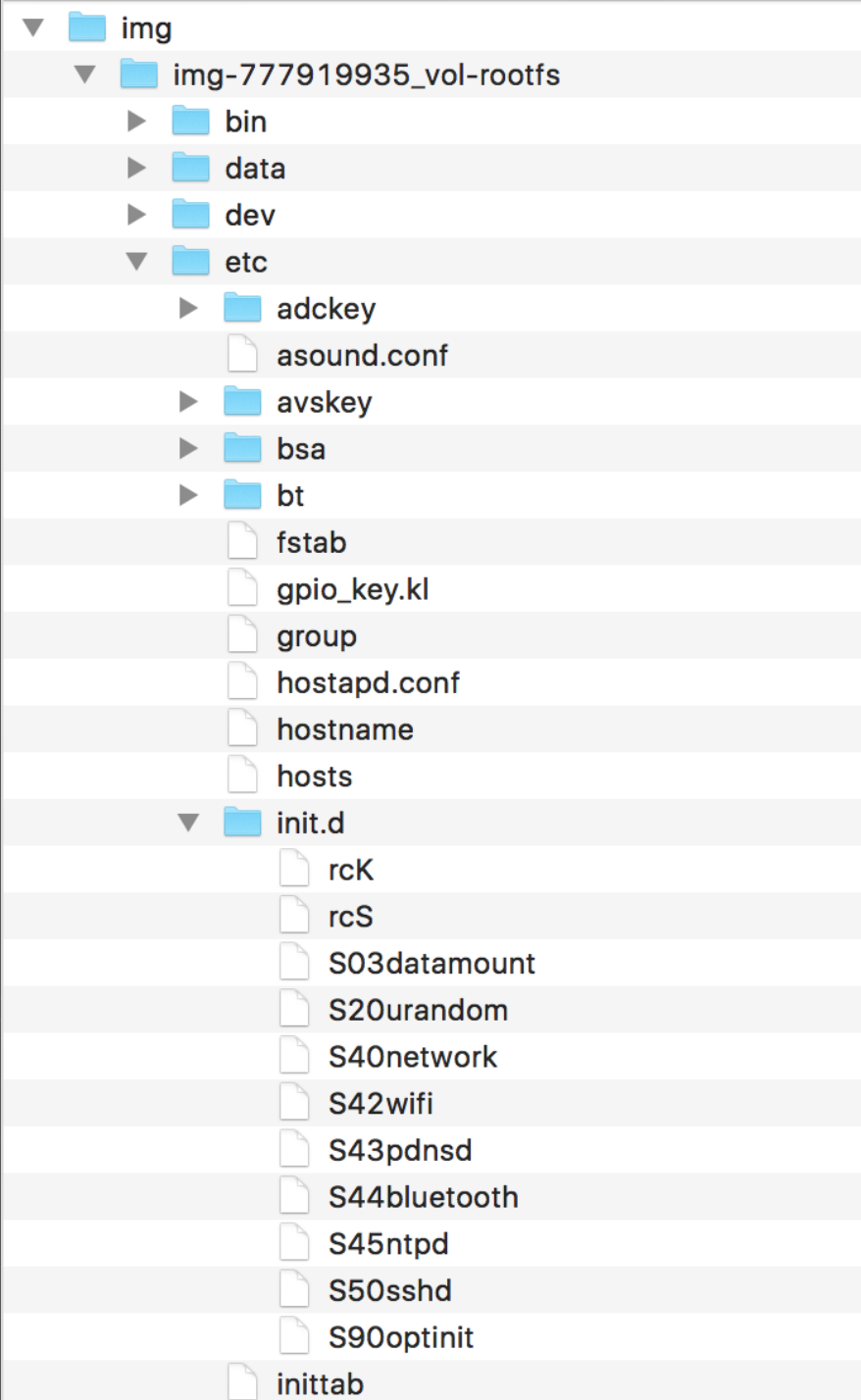
准备工作-获取固件

方法:

- 官网下载
- 自升级, 监听数据包 (如果查询版本, 需要截取修改低版本。特殊信道)
- 升级app逆向分析, 升级流程逆向分析 (访问ftp)
- 求助论坛、好心网友 (行业维修论坛)
- 万能的客服 (帮忙救砖)
- 获取shell (telnet、ssh、adb...), dump 固件 (dd、tar, nc转出)
- 进入BootLoader 读取存储器
- 特殊主控读取方式 (例如MTK、NXP系列, 可以通过数据线口获取/刷写文件系统)

But, 有时这些都不奏效





```
# 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",
```

```
1 <?
2 if ($_POST["act"] == "ping")
3 {
4     set("/runtime/diagnostic/ping", $_POST["dst"]);
5     $result = "OK";
6 }
```

```
printf((char *)&v31, "USER %s\r\n", aWan);
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) )
    {
        printf((char *)&v31, "PASS %s\r\n", aWYf);
        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
8961881	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
6501172	1.21 MB	2018-03-29 14:20:26

物理Dump

当常规方法无法轻易获取固件

- 大厂设备固件都是加密的，binwalk等无法解开
- 没有固件升级流程，固件写死不变
- 固件通过GPRS升级，无法干预（实际我们可以干预☺）
- TTL关闭、telnet关闭、BootLoader无法停止进入

那么就开拆，物理dump

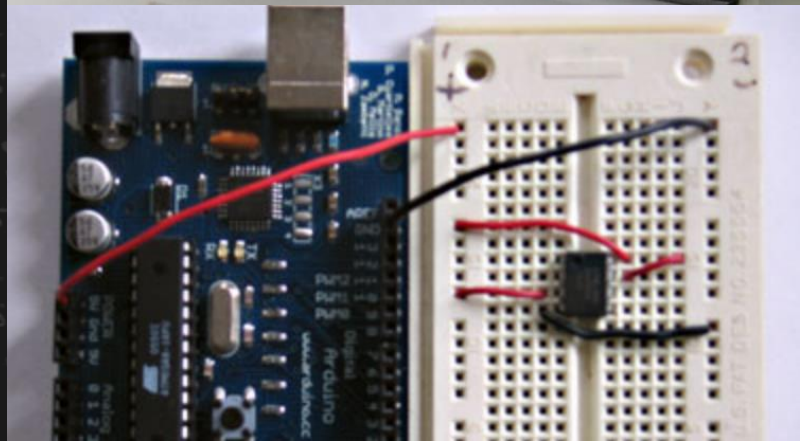
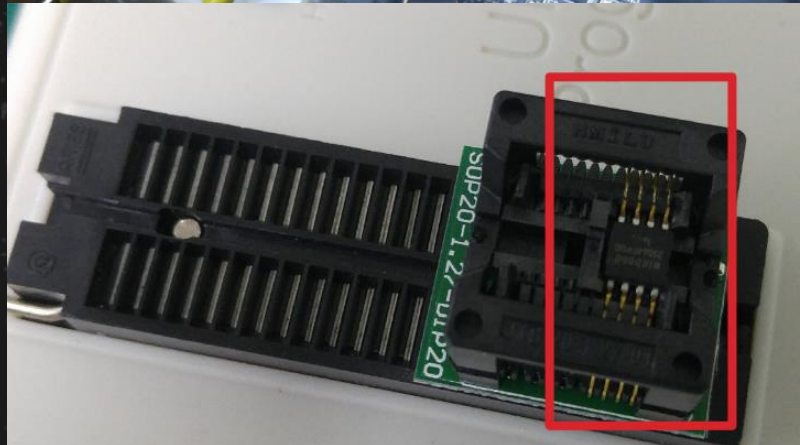
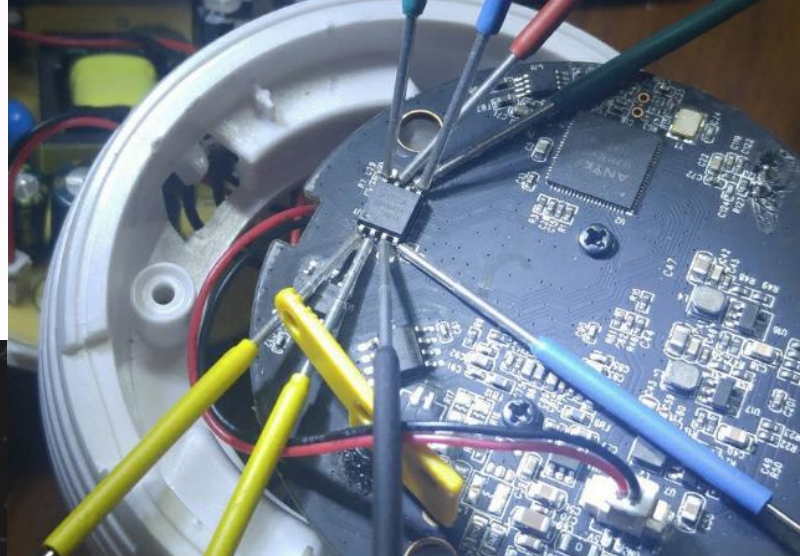
- 针对不同型号，采取不同读取方式
- 步骤：
 - 选择读取设备、方式（在线 or 离线）
 - 对固件进行处理

物理Dump-SPI Flash



针对SPI Flash (对应简单设备、路由设备)

- 串行读写设备，常见容量4/8/16MB，8针脚 SPI接口
- 结构：
 - 完整操作系统：Bootloader+内核+文件系统，大多采用压缩
 - 仅仅存储数据、配置文件等
- 读写方法：
 - Arduino+EEPROM 库
 - Raspberry SPI接口+ [flashrom](#)
 - 编程器读取更快捷 (RT809H)
- 免拆焊 (勾针、夹子)，但有时不奏效 (CPU被加电)，建议拆下来
- 可直接修改固件、getshell，注意文件结构
 - 文件系统、偏移：启动信息获取、binwalk获取
 - 焊接下来->解包->修改->重打包->dd偏移、合并->刷写回去->焊接回去



物理Dump-SPI Flash-获取文件系统结构

```
[ 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"
```

```
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"
```

通过console信息输出获取

→ ~ 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

通过binwalk获取

通过shell命令获取

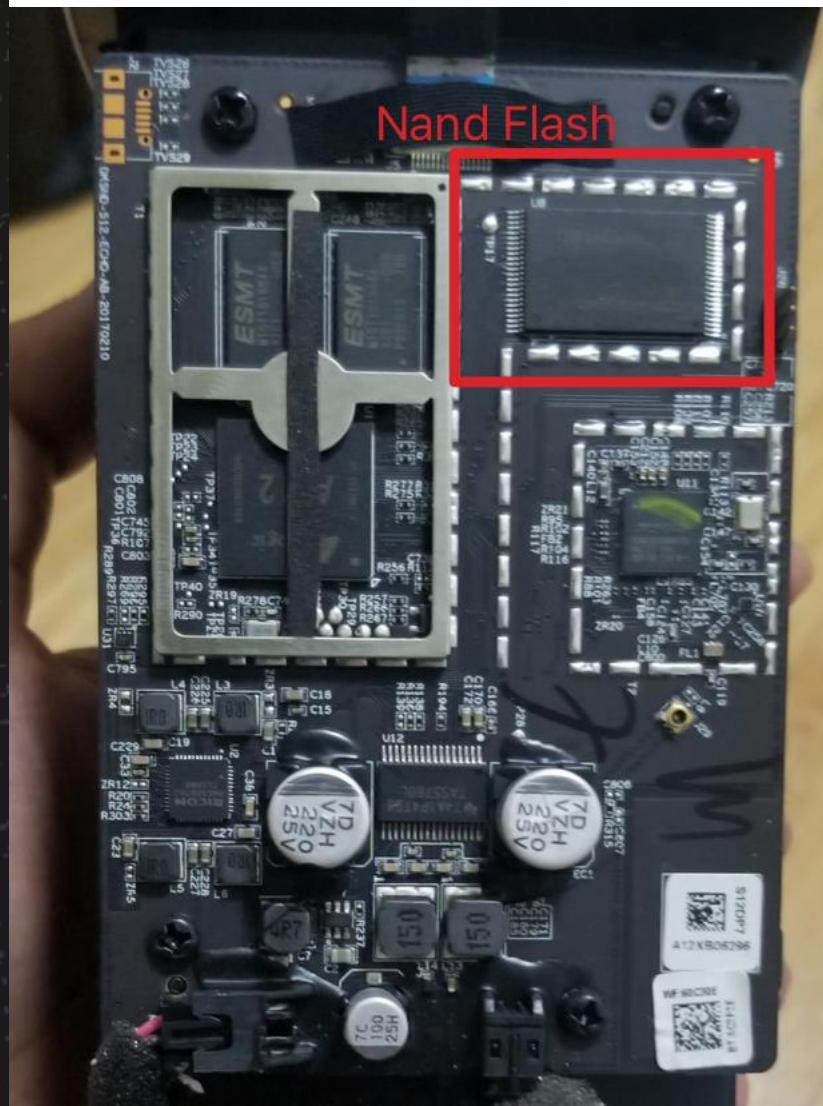
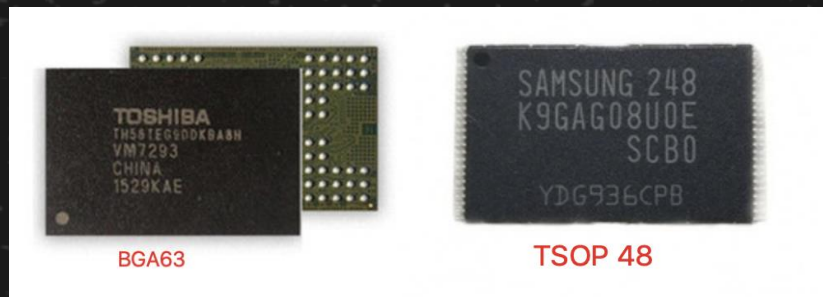
物理Dump-NandFlash

针对NandFlash（对应较复杂设备例如高级路由、智能音箱）

- 16MB-2GB容量，TSOP48/BGA封装，按块读写
- 拖焊新手经常出问题，推荐使用热风枪拆焊（注意保护周围元件）
- 结构：完整Linux/Android系统，大多不需要压缩解压
- 读写方法：
 - 有效针脚17+，需使用编程器读取，例如RT809H
 - 有坏块管理，但是管理较低级，写入比较繁琐

坑：获取的bin固件 通常binwalk无法解开，需对binwalk进行修改，或者去除ECC校验位数据

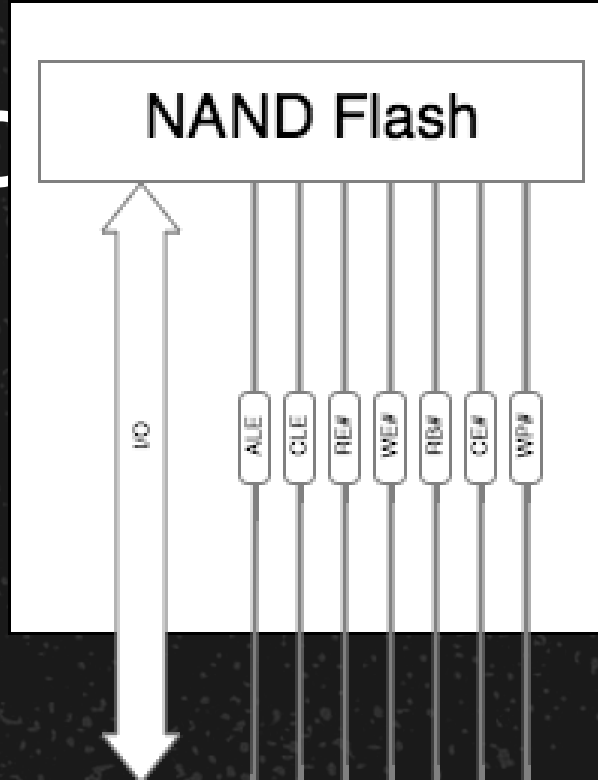
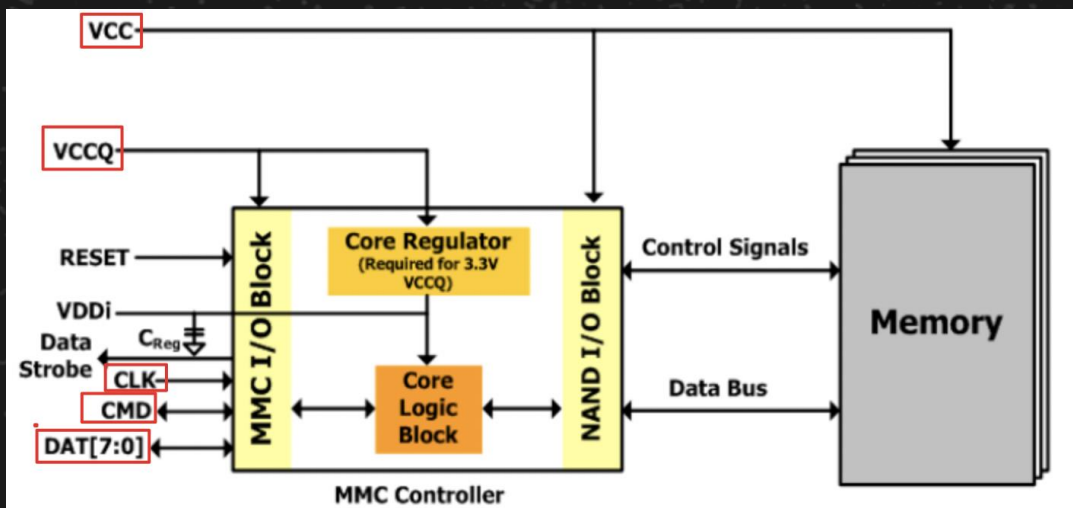
此类设备处理比SPI Flash和EMMC麻烦，且文件系统格式各家不统



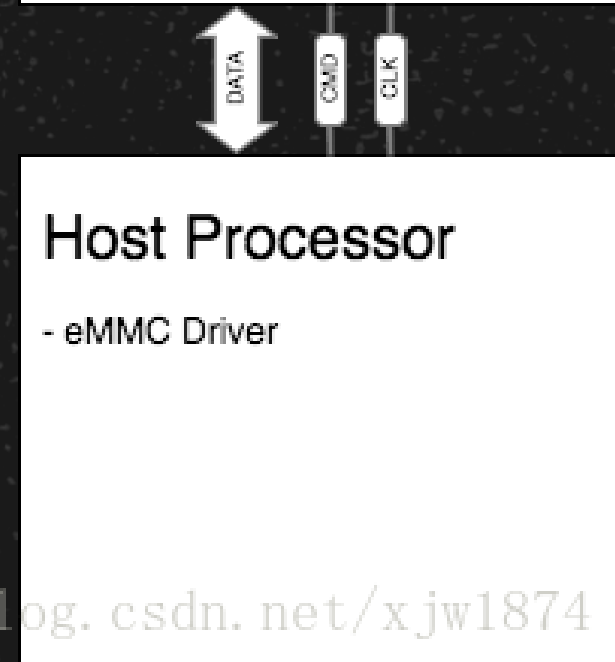
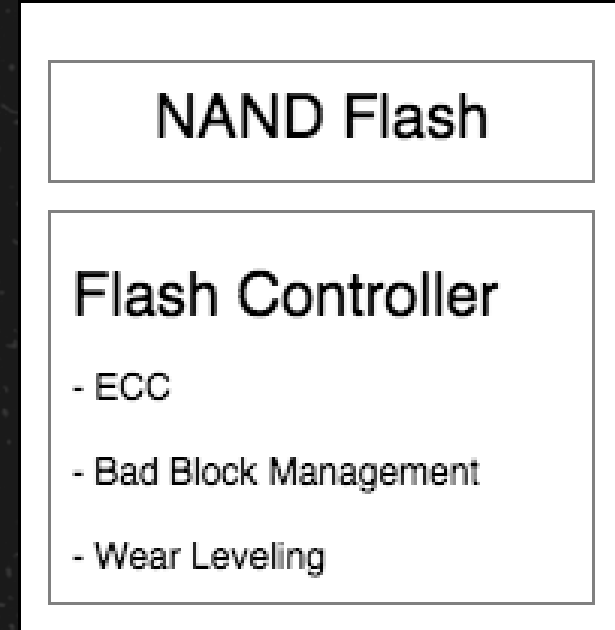
物理Dump-EMMC/EMC

EMMC与Nand Flash的关系

EMMC=NAND闪存+闪存控制
芯片+标准接口封装



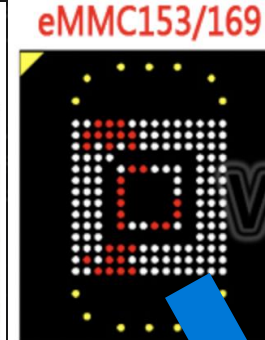
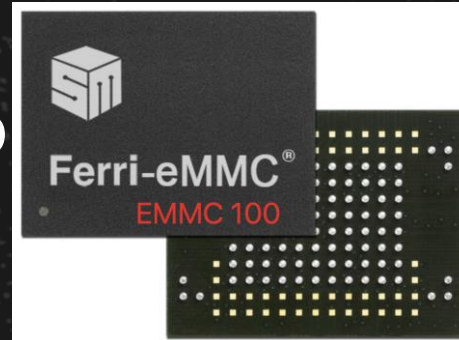
Nand Flash



EMMC

<https://blog.csdn.net/xjw1874>

物理Dump-EMMC/EMCP



针对EMMC/EMCP (对应复杂设备例如智能电视、手机)

可近似理解成SD卡

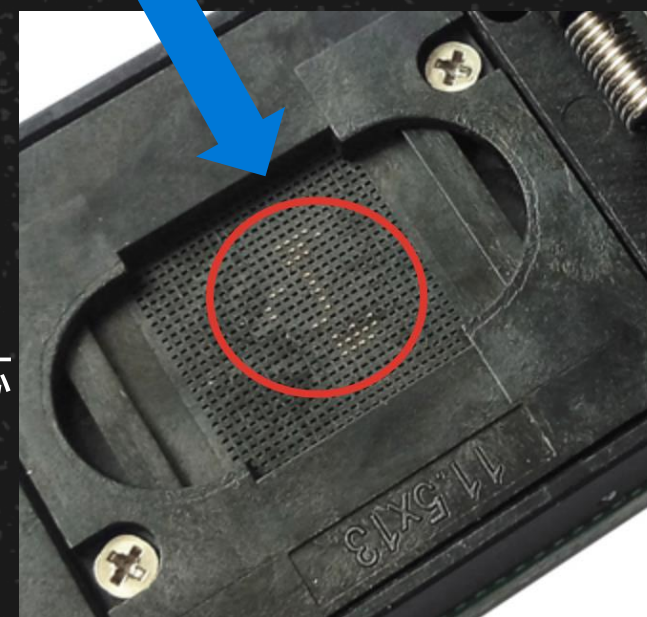
全部为BGA封装, 规格100/153/162/169/186/221 (实际这4种占95%)

离线读写:

- 采用热风枪吹焊 (注意保护周围元件), 专用读取座+编程器or 直接在芯片上飞线读取
- 焊接采用植锡法, 对焊接技术要求高, 需要多练习

在线读写 (不需要焊接下来):

- 需要寻找or已知关键焊点, 非常细小, 焊接要求高 (寻找方法?)
- 直接飞线最少DAT0、CMD、CLK、GND、(VCC、VCCQ) 到SD读卡器, 不需拆焊, 注意需要短接晶振



物理Dump-植锡过程视频



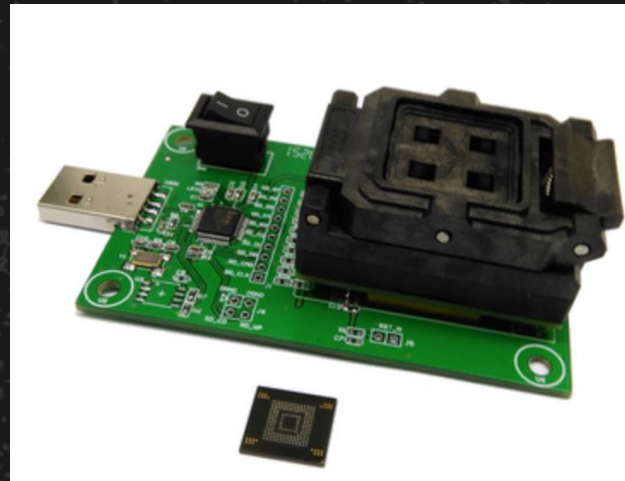
离线读写



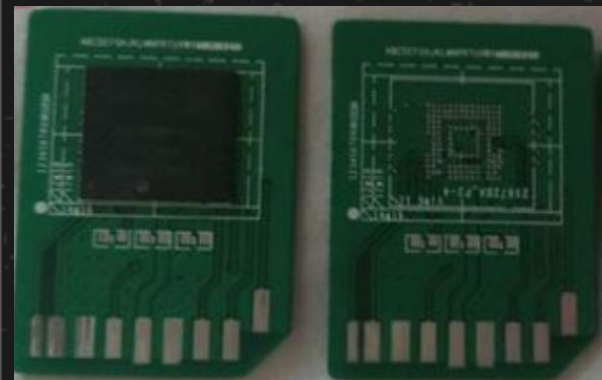
保护芯片



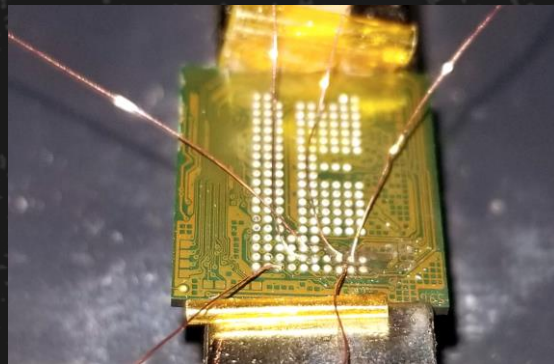
使用编程器



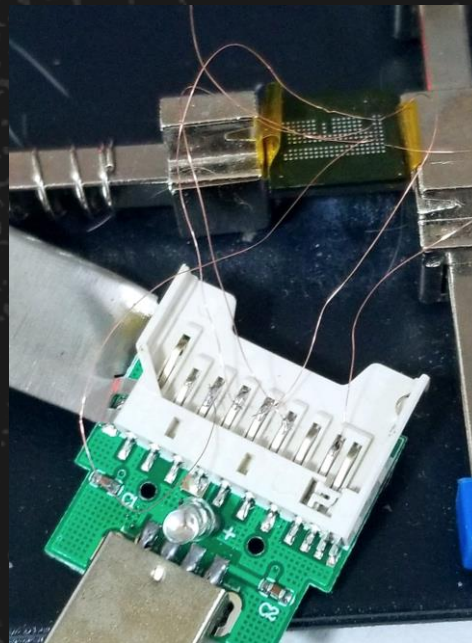
使用专用读写座



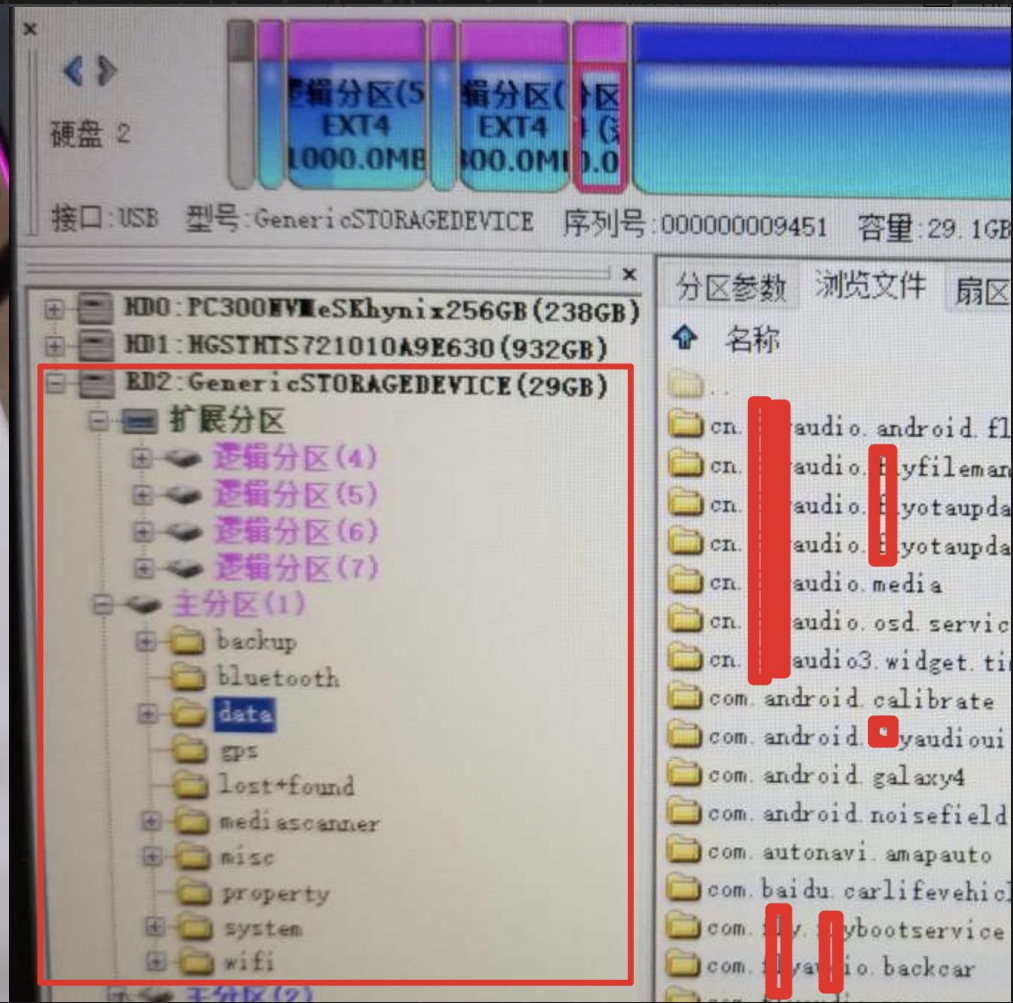
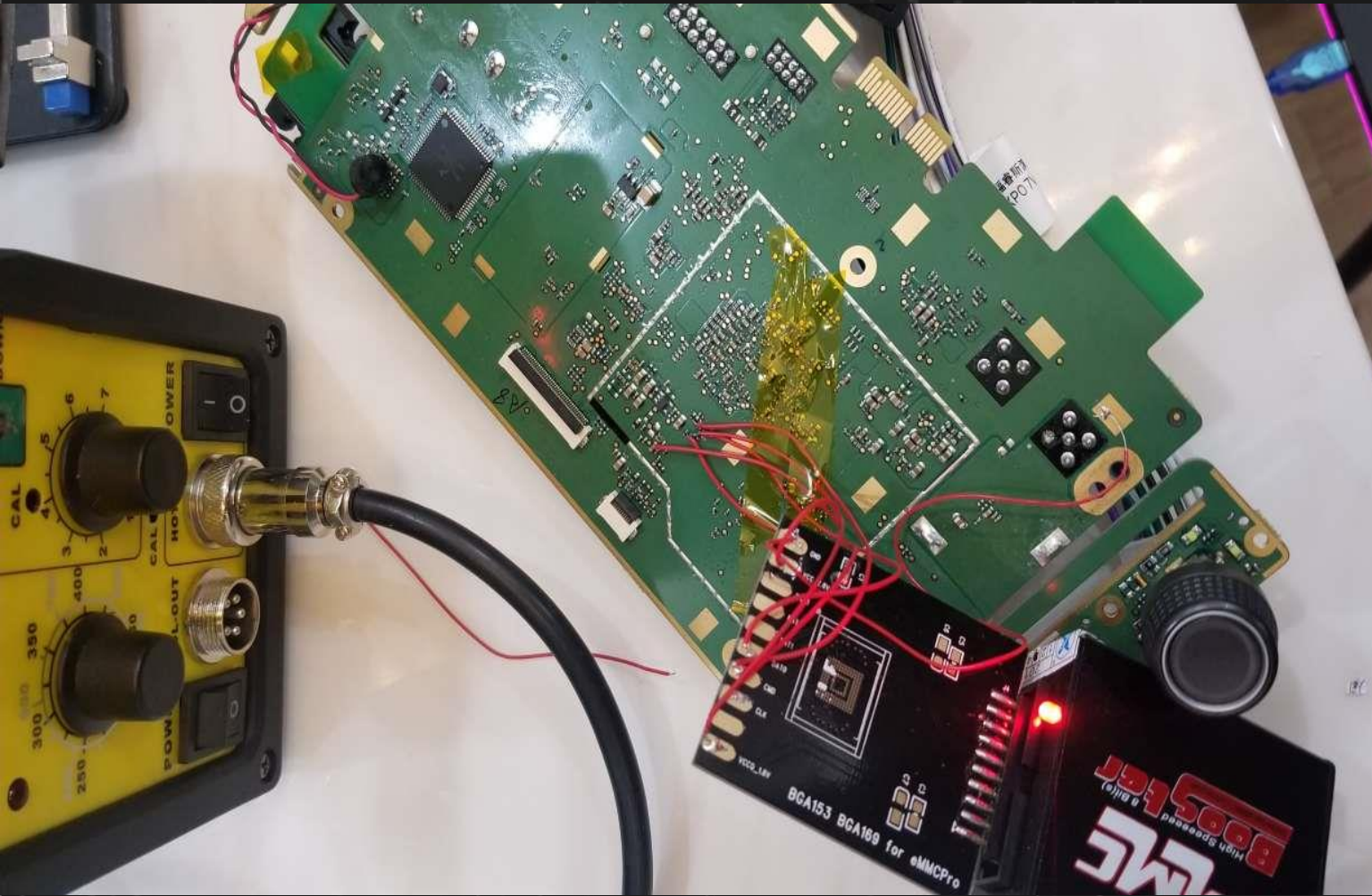
直接焊接到SD卡上



直接在存储芯片上飞线，连接到SD读卡器



在线读写



某款车机系统在线读写

准备工作-Getshell

目的:

- Getshell后直接tar导出文件系统（固件），分析bin、web脚本
- 更方便的查看数据，例如查看端口、进程、网络、文件
- 搭建测试环境，编译好的测试工具
- 方便的在线调试（有时lib库、硬件限制，QEMU很难离线run起来）

总之，能getshell就是最理想的破解前提环境

- 如果通过远程getshell了，其实已经完成了破解

准备工作-Getshell

方法:

扫描端口, 寻找是否开启telnet、ssh、adb服务等

- 使用快速扫描, masscan、nmap -sS
- 密码&hash可以在固件里找, 离线破解 (如何加速破解?)

lsusb, 查看是否开启usb adb

寻找web上传漏洞、命令注入漏洞等

在线or离线修改存储

- 例如init启动项中添加: busybox telnetd -l /bin/sh &
- 对于采用EMMC存储结构, 修改非常方便

寻找板上TTL针脚

- 明显标注
- 根据CPU datasheet

7 [redacted]_Bk:y2 [redacted]Um ← 破解结果

```
Session.....: hashcat      使用显卡进行hash破解
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 Undefined
Guess.Queue....: 1/1 (100.00%)
```

使用hashcat破解ssh、telnet密码

```
→ ~ 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
```

使用nmap进行快速扫描

```
→ ~ 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
```

使用masscan进行快速扫描

准备工作-Getshell

方法:

修改Bootloader启动参数

- 强制进入uboot配置模式, 修改内核参数 例如添加<空格> 1 ,进入单用户模式
- 使用JTAG接口修改内核参数

```
[ 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: 512 (order: 9, 2048 bytes)
```

准备工作-Getshell

方法:

使用JTAG修改内核参数 获取shell

- 设备需要具有JTAG口, 并且有对应JTAG设备、CPU配置文件
- 软件推荐OpenOCD, 支持CPU种类多 硬件推荐jlink
- 修改启动参数
 - 固件中寻找启动参数位置
 - 添加断点
 - 修改启动参数, 例如添加<空格> 1 ,进入单用户模式
 - 引导内核, console 串口获取shell



准备工作-获取通信数据

目的:

- 了解工作逻辑 (辅助分析, 例如根据http请求寻找加密代码)
- 获取cookie、token等认证信息, 敏感隐私数据
- 获取服务器接口, 以便渗透服务器 (授权渗透)
- 截获修改数据包, 或根据已知数据包构造重放
- 最终下发合法指令、构造poc、拿到关键key等

通常使用Wifi/234G/蓝牙/低功耗蓝牙/红外/有线/其他频段无线电

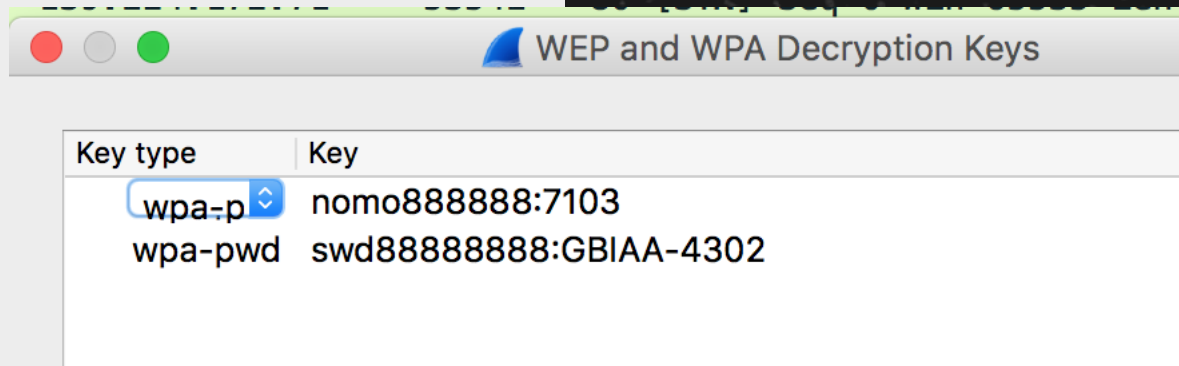
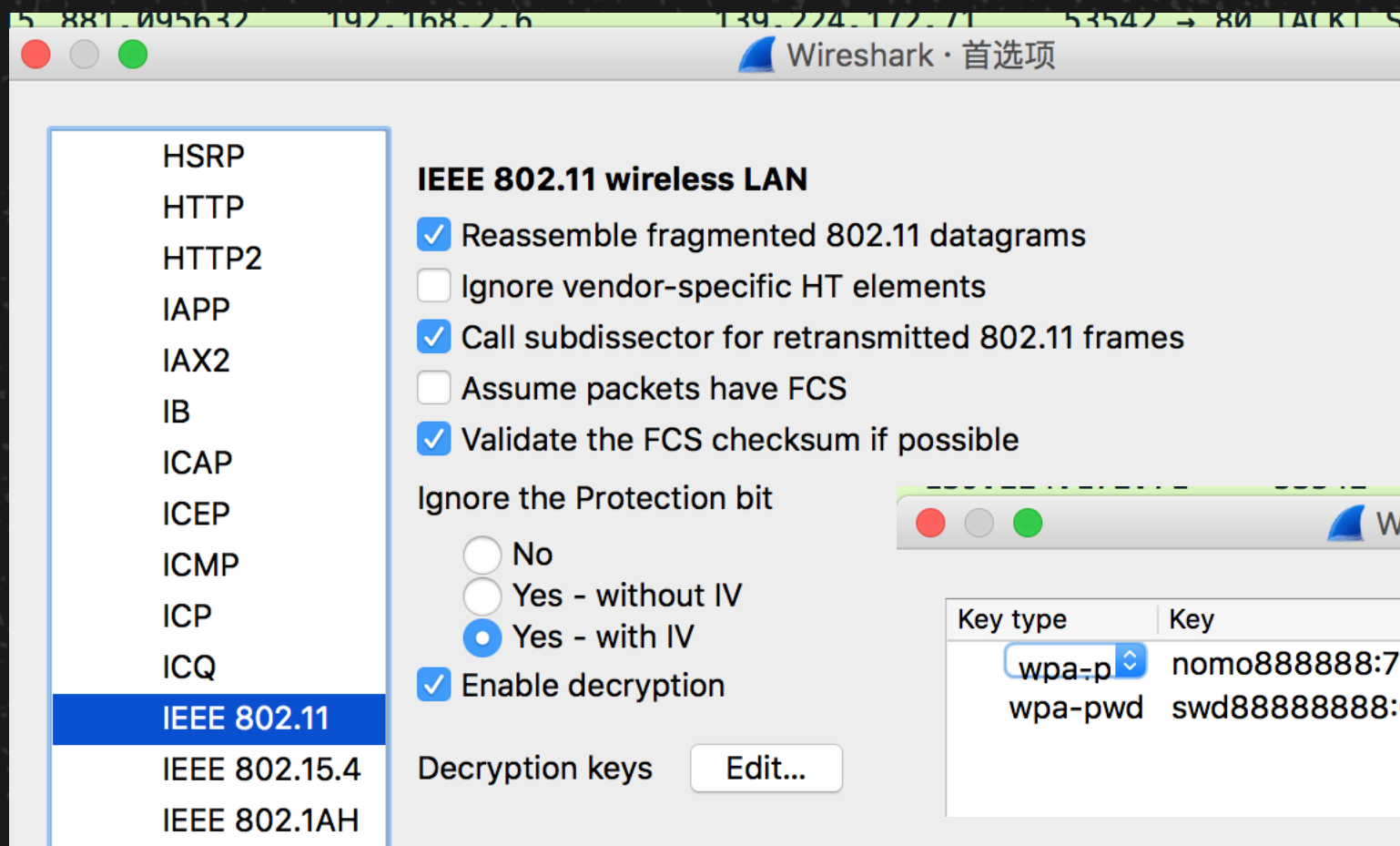
准备工作-获取通信数据

方法：针对IP数据（TCP、UDP、HTTP、MQTT等）

Wifi:

- 实时wireshark:
 - 开启无线热点并连接，wireshark直接监听这块网卡
 - Android adb forward + tcpdump + 管道 给 PC机wireshark
- 在路由设备上抓包
- 如果是Android APP，直接在本机模拟器运行，监听本网卡
- 如果是HTTP、HTTPS，设置代理，
- 交叉编译tcpdump (arm、mips) ， -A选项or -w
- 如果远端设备，arp中间人
- 如果远端设备，且动作小：**wifi实时解密**（强大网卡支持，例如RTL8812U）

设置wpa/wpa2实时解密



如果通讯数据全是密文

SSL/TLS 加密信道

- https代理
- 如果验证证书, 导入burp根证书
- Android中:
 - Xposed bypass 强制不验证证书
 - Hook大法 (okhttp)

AES\DES等对称加密, 采用TCP传输

- 逆向分析APP、二进制, 获取秘钥
- Android中: Hook大法 (Crypto)

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

对常用的http操作库okhttp进行hook

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

对java自带的加解密库crypto进行hook

关于hook (针对Android)

框架

- Xposed:
 - 仅支持java层面hook
 - 适合批量部署 安装
- CydiaSubstrate:
 - 支持java/native
 - 不开源, 且不更新无法适配新Android系统
- Frida:
 - 适合破解使用
 - 支持java/native, 支持多平台, 适配最新系统

工具

集成了http、加解密、sql查询、文件操作、IPC、自定义hook的功能

- 基于Xposed:
 - Inspeckage
 - <https://github.com/ac-pm/Inspeckage>
- 基于Frida:
 - appmon
 - <https://github.com/dpnishant/appmon>

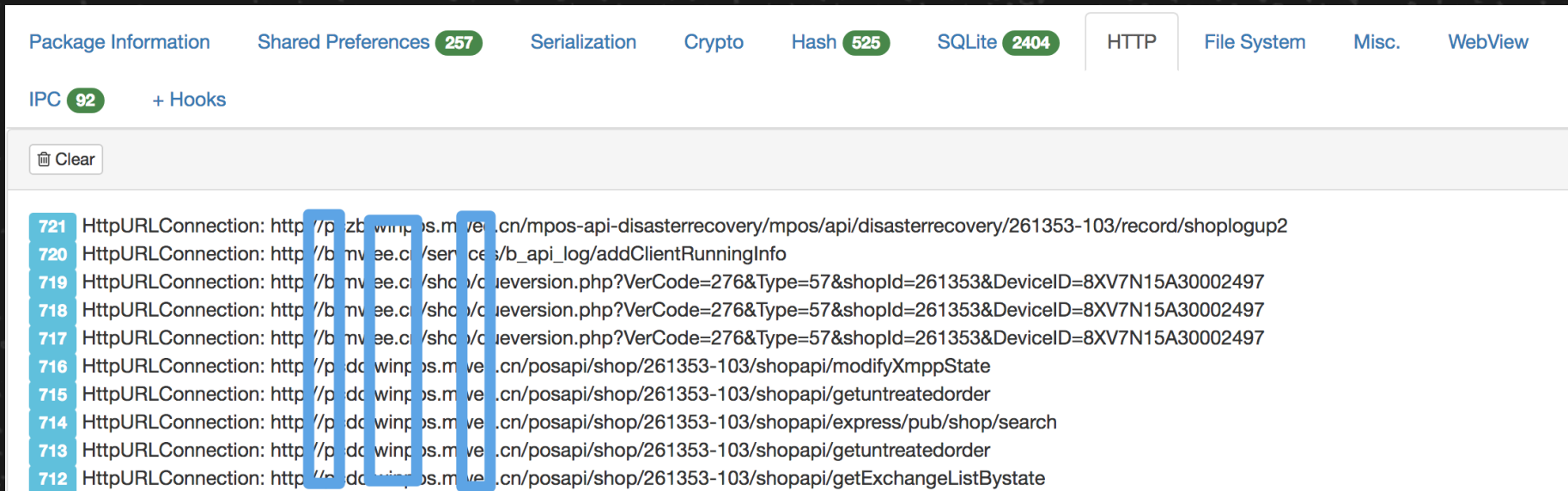
Hook什么?

敏感操作hook

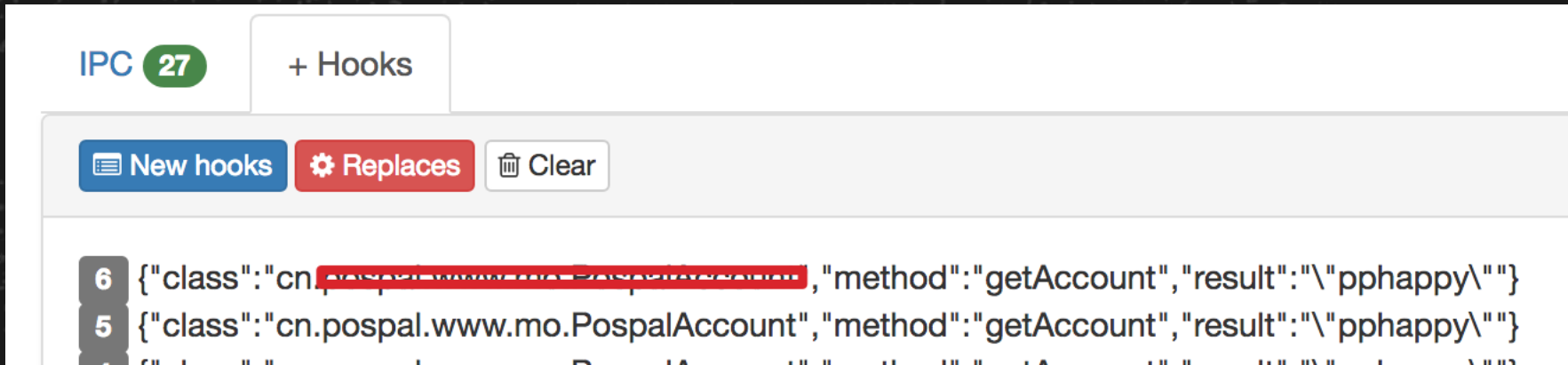
- 对称加密key、明文密文
- Sqlite查询 (判断是否有注入、有助理解逻辑)
- http、https请求内容
- Hash调用情况
- 其他 (webView、序列化、文件系统操作、SharedPreferences、IPC等)

目标函数自定义hook

- 获取返回值
- 修改返回值
- 如何确定hook的class、method (trace)



使用Inspeckage Hook http请求



使用Inspeckage 自定义hook

127.0.0.1:8008

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

UID: 10094 | Debuggable: false Package: com. [redacted]
GIDs: 3002-3003-3001 Data dir: /data/user/0/[redacted] Tree View
Allow Backup: true

2.4.5.276

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] (NcJWkqmfIS6s7I7PdrPSTNRNyl7uCYTnQvi4K+zWM3ijSGoQdB5anKGoY
JfcDP74NkMZ2nk5aP1/oC/BeGx0z11jJpyGxNZ8Q5I1byNDieAqiWVW8ng+y64BqcJgPEavr5hJVBRvKxN2JvOa+teB7os/NMNfzbvppJffw6P37NRQCh1TK9cwbwNJEpCYU4PAbXu
U1RUL9wu8TbjrbTR8EiQatSKAXRpOZBX5WpMrNLbchfIASLNdjoa4bwGIlFICZz0s/3C+9npiUBqBf7HKu95GCV03UCVNZBcCALkpeP67H/fgFpxUCFMZnAwW2Xv12EcTIIYICai
rYV0jLkPrRi86HgKTAMMhuEoZxKMTBm3a9bnljVURNXkX/Blv5uBvqoFhD9mXfHUMcSV08+fzdXIHJLa2/011FgZJSivRQ1XfEJD7639ad4xok8rjgRk3wgjgTZLHS1FzUpHiqonDaFx
etNjE8kyiwjmj7jUY+kZiU3Pt5TUgyUvbat0uPxN220VeKkN8YS/vwkC2kpAOX1EeB65mrt0JmHJat21pc2GKI9VBoeFKHmSz7NDMPduadk48/HjWYauQWZu99MB1n6j14y/NOmd
23nox7D+oTGCoTSN6yEpwp0loSrlU0pkQ4j64zguNWJL+PT6egXVNXi9dGjL4w1RGD1cp5XbJwulPVdO+W4o5HPVfJoyl09B/c/LhBCc1tLeg1IVhEqyNiicVV+Xgu8shpC+d/wjrsoV
KFwoCDqOIQ434MHtse4Uz81gbdnjmWGShlUJ+S6CJdqrDUDfrf0dZs+ifuphcA/Ovf9s5OpGAFtMA1O139mdfuRQLaODy1NSmGc/3MY0zdP6knUojdqdz302Tr+ldBBCKonJzM
SecddoWgG1nlgCLXx9vHV9fzUyB+bBhZvjZjskNjVJysct9zAwagFSN9LBDYus6Alrei41Om7/7LkYBCqIRx9SY/pjtbPB7ozjELIM1v0am1UxRPvITWFjsihYNvowpiRG+hvFGOZ2gxE
ViyAlzDkwdstUrJawONEj3FEOI6opylWED/be2VbjfWxCj2aUeo5TiXHbwS6MV14/5DvJzxKlrzsyTYmR5733trbj7n2zXg8a6UoskOhtdXebheQQJatS4f0lbz8QyEY8bctU94W78bjah
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":"....."}}
```

使用Inspeckage hook AES加密

获取通信数据-其他信道

234G:

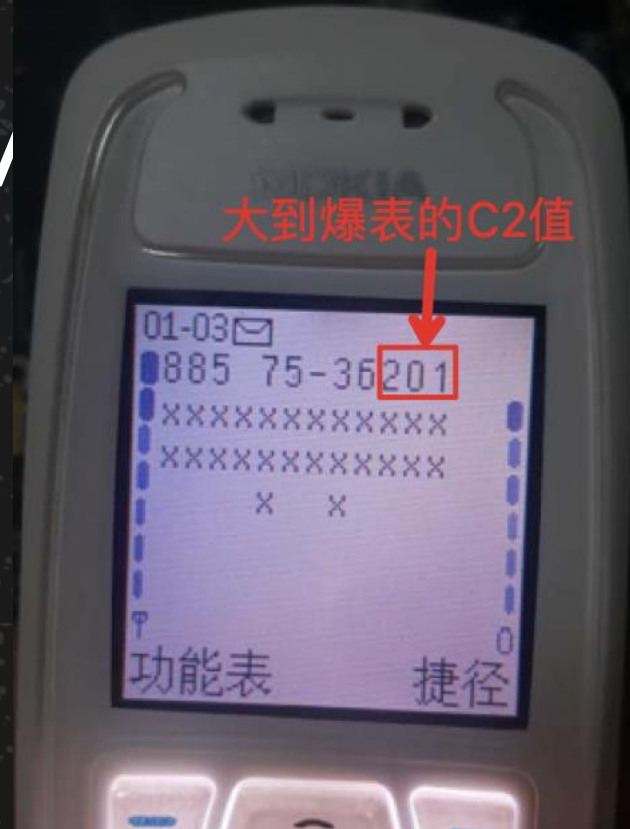
- 需要远端通信的设备，例如自动售货、共享单车锁等等
- 开发人员通常认为这个信道十分安全，很少考虑加固
- 通过假基站GPRS劫持，可以**完全控制与基站连接的网络流量**
- 根据运营商网络互通问题，也可以进行远程访问触发漏洞

蓝牙:

- 现阶段主要以低功耗蓝牙为主，例如运动手环、智能温度计、蓝牙开锁等
- 开发人员通常认为这个信道十分安全，基本很少考虑加固，多存在密钥泄露
- 传统蓝牙分析只能跟踪广播包，山寨设备可以跟踪跳频
- 手机调试模式**开启蓝牙log**，简单稳定

针对234G设备的流量访问、嗅探、MITM

- 2G网络由于手机无法对基站进行认证，存在假基站风险
- 搭建GSM基站系统（合法条件下测试）
 - 硬件：Bladerf（相对其他SDR设备，精度高）
 - 软件：YateBTS（图形化界面/安装方便）
- 如何让智能设备自动连接到假基站
 - 借鉴伪基站给手机发短信的思路：增大小区重选参数C1、C2
 - 修改YateBTS源码实现
 - 详细内容可以参考我将要再Defcon China上关于这方面的议题
- 攻击：获取流量、MITM、访问端口触发...
- 其他简单方法：
 - 运营商网络互通（10 or 172网段），买两张sim卡，可以触发基于端口的漏洞



```
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 {
```

第二步-分析

结合已有文件、网络请求、shell

- Netstat -tunlp 看监听端口对应进程，分析之
 - 命令注入，例如fopen()中的内容可以控制
 - 危险函数导致溢出，例如strcpy()
- 如果没有shell，就端口扫描，无状态扫描
- 如果有web
 - 确定配置文件、web源文件
 - 对web页面进行漏洞挖掘（php、cgi、lua脚本等）
- 根据网络访问定位关键代码位置（反编译、关键词、trace），获取加密逻辑，获取接口参数格式

最终获取到关键数据，或者下发指令

分析举例：某自动售货机 核心逻辑问题

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 url、用户名密码）

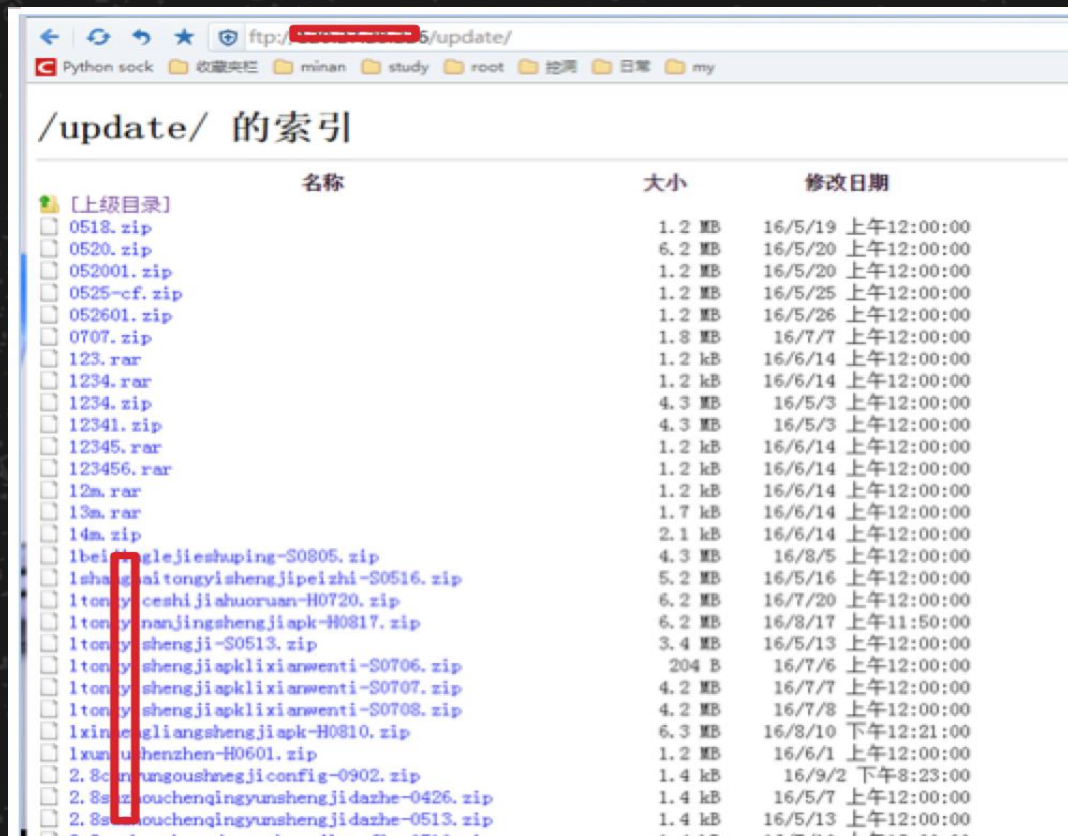
```
root@ubuntu-14:~# nc [redacted] 6001
[redacted]
05710031ftp://etftp:yichu[redacted]5/update/[redacted].zip
```

Bin文件中泄露FTP升级服务器密码

```
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.getByOrderId(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");
            }
        }
    }
}
```

支付服务未验签导致0元支付



可以控制其他售货机任意更新固件

两款手表 信息泄露及配置修改

```
..{"Version":"00030000","SN":1074096116,"CID":10211,"PL":  
{"Name":"863412030", "Password":"7805303461C5E33FC8", "Type":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":
```

云端登录过程泄露密钥

10463	http://watch.okii.com	GET	/smartwatch/watchinit	200	2342	JSON
10463	http://watch.okii.com	PUT	/smartwatch/watchaccount/battery	200	235	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_acleph", "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": "ed3009a56a1a4b8db61afe535dc509a9", "watchId": "b5c2dc7065cc46bf90e602af13e1589808955307", "title": "禁用时间段", "classSwitch": 0, "amTime": { "start": "08:00:00", "end": "11:30:00", "pmSwitch": 1, "nmSwitch": 0, "classWeek": 31, "type": 0, "createTime": 1493888530000 }, "autoRecordSportTime": 3600, "guards": [ { "rate": 300, "start": "07:30:00", "end": "08:40:00", "schoolWeek": 31 }, { "rate": 300, "start": "16:23:00", "end": "18:10:00", "schoolWeek": 31 }, "contactMobiles": [ { "contactId": "b58ea4d6ad324c96852cd6744c1bbc3b", "mobileId": "077e6fd1473842deb318759802ab6c29", "imAccountInfo": { "accountId": "077e6fd1473842deb318759802ab6c29", "imAccountId": "66529961", "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": "http://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", "waiterStopTime": 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": "451bea68cd2d4064ac1da2cda16c377e", "error": [], "errorSN": null, "serverGreyCode": null }
```

云端交互过程中MITM修改配置

某共享车锁 信息泄露及解密

```
PUST /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+ijHq59dUwrBdxcv0DqDQ"

qLUAAQAvMUhcMTcxNDAzOTExAYAE+B0mKUwiw0girvY5Sax1sLQy45XaioAu38M6gJxkpyYnAQ==
```

与云端加密传输

```
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('d31d1ef31dcebada585ae71bd5a3c0365d66b5d6de92b320b18bd32cc6d332313'.decode('hex'))
41
42
```

Run aes_test

/opt/local/bin/python /Users/gaoshupeng/PycharmProjects/work/of0/aes_test.py

n:0"00'07 vG~WY
01234000000000Y0 NY0 N000000000

解锁密码

分析固件，获取密钥及升级协议

某通讯模块 FTP server 协议存在命令注入

```
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 }
```

一些必备技能、小tips

- 焊接技能
 - 烙铁焊接，拆焊、拖焊、吸锡、洗板，不连焊&脱焊
 - 热风枪拆焊焊接，植球植锡（低温锡浆）
 - 飞线
 - 买正品白光烙铁，调温&8秒升温不老化
- APK反编译、hook、动态调试、Java代码阅读
- Web攻防，源码审计能力
- Python、Java编码能力
- 简单的二进制逆向分析
- Wireshark TCP、HTTP数据包分析
- 熟悉跨平台、交叉编译

一些必备技能、小tips

- 常用工具:

- 准备好多平台下的gdb、tcpdump、telnetd、nmap、masscan...
- 好朋友 多平台下的busybox

- 常用命令:

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

Q&A