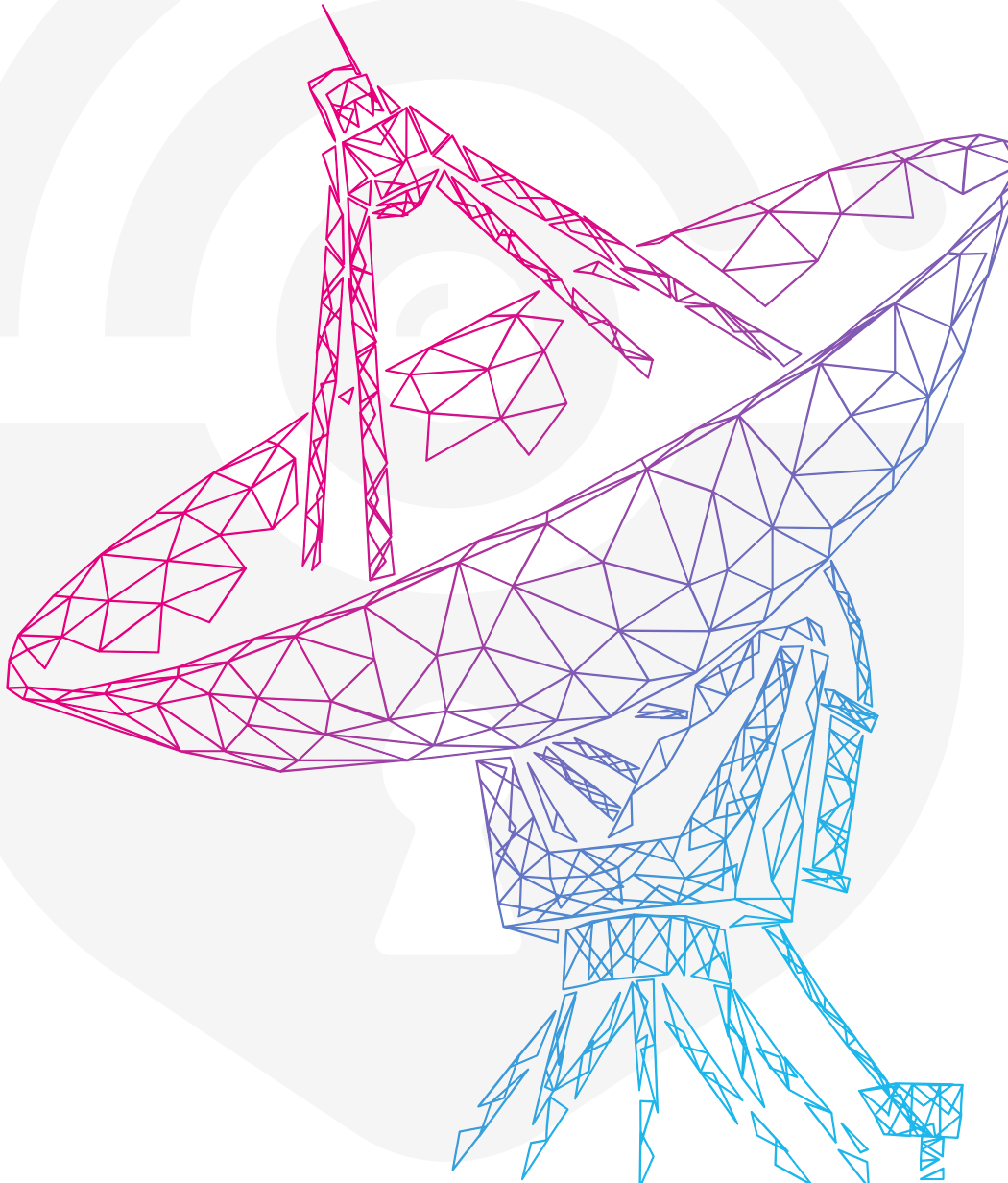
penthertz

Connected cars: With and without wires for tires

PraSec 2024

By Sébastien Dudek & Bogdan G.



Who are we?

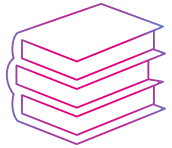
- Sébastien Dudek (@FlUxIuS)
- CEO of Penthertz
- Specialized in RF & telcoSec

- Bogdan G. (@djnn1337)
- Intern @Penthertz
- Automotive security apprentice

Main activities

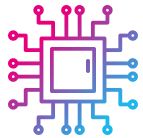
Security assessments

- Wireless communications (RFID, Wi-Fi, Mobile communications, Bluetooth, etc.)
- Embedded devices
- Backend servers
- Red Team



Trainings

- Software-Defined Radio Hacking
- Wi-Fi Red teaming
- RFID Hacking
- Mobile attacks (2G/3G/4G/5G), and more...



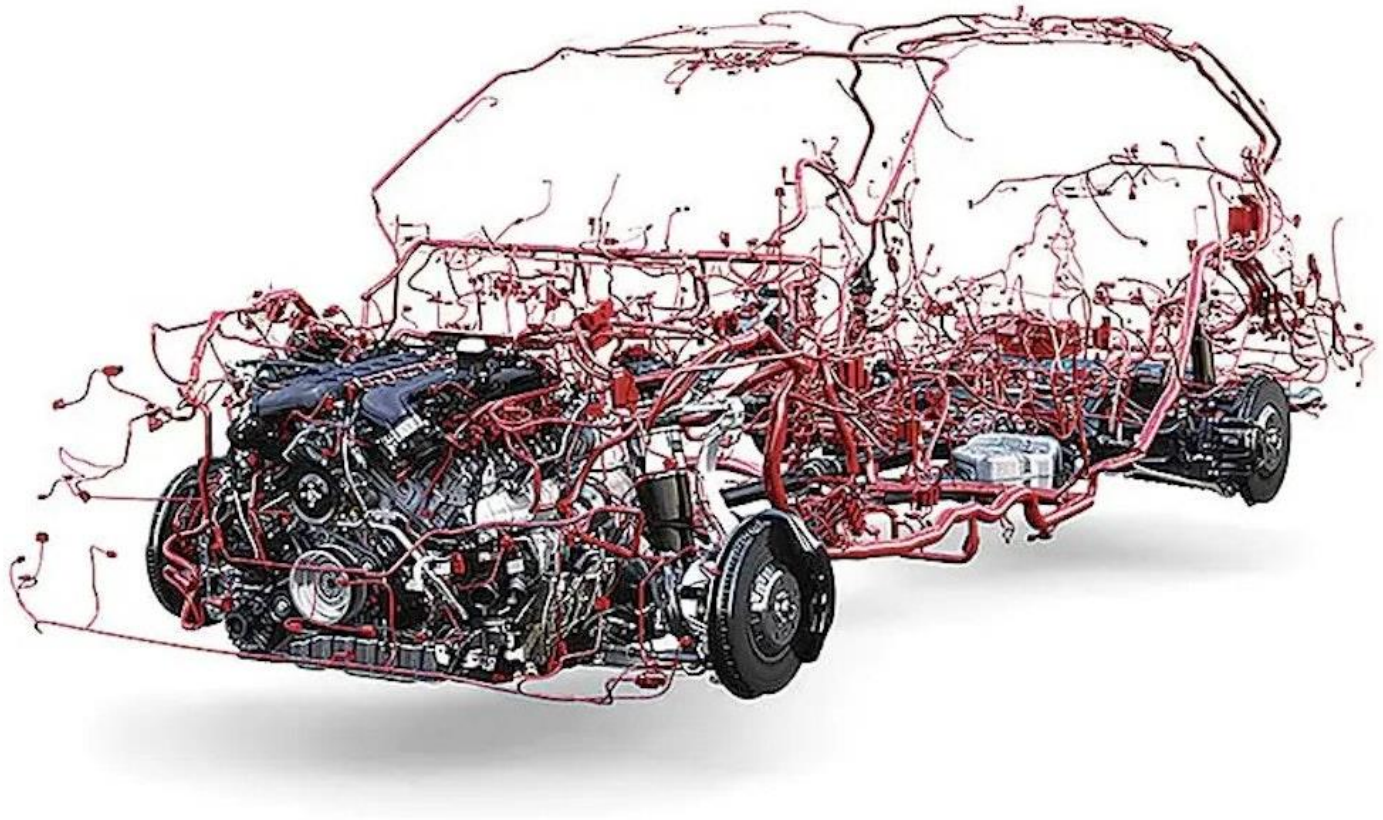
Hardware security

- Firmware extraction
- Chip off
- Secrets extraction
- Library's analysis
- Vulnerability hunting

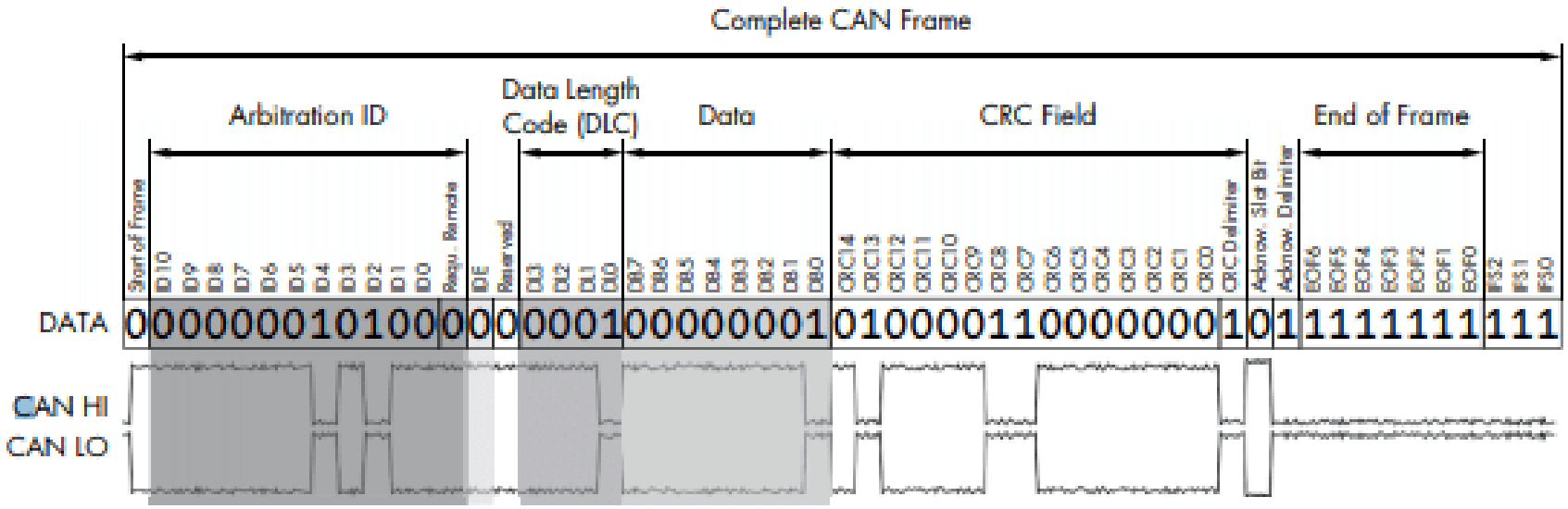
ECUs & interfaces

CANbus

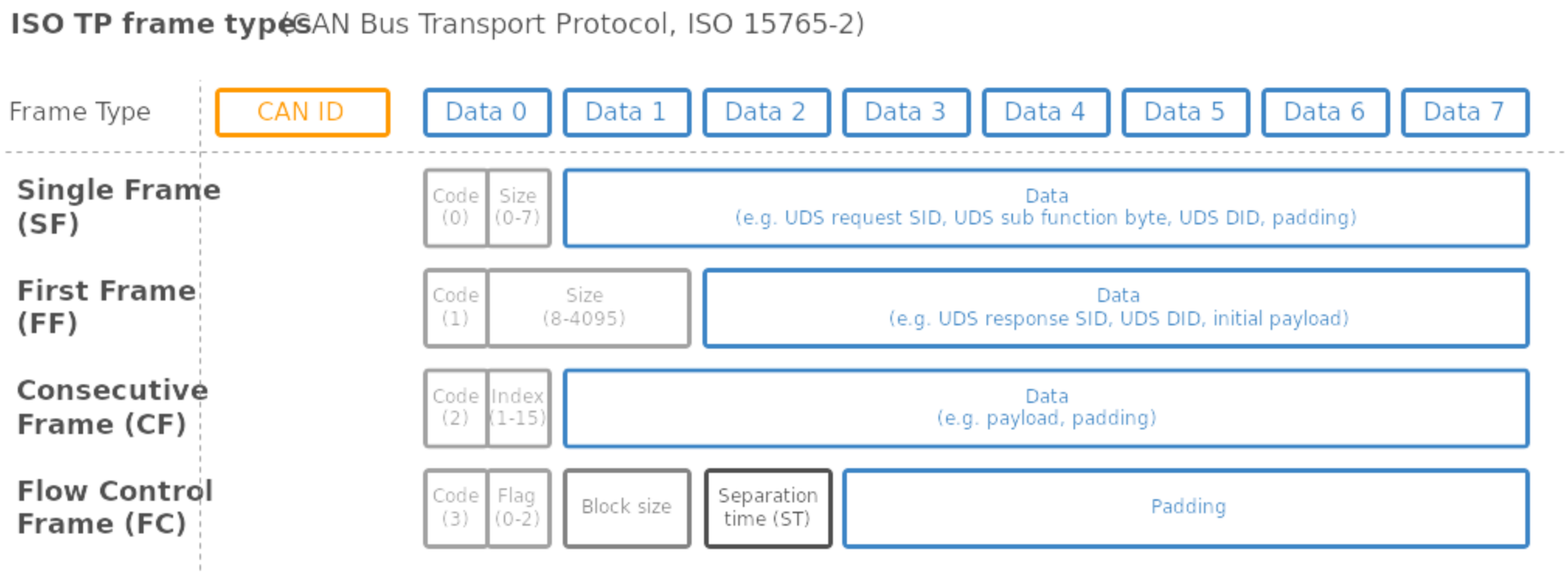
- Present everywhere in the cars to link one ECU to another
- Today:
  - Wiring is very complex
  - And includes a lot of sensors



What is a CAN frame ?



ISO-TP


ISO TP frame types (CAN Bus Transport Protocol, ISO 15765-2)
Frame Type
CAN ID
Data 0
Data 1
Data 2
Data 3
Data 4
Data 5
Data 6
Data 7
Single Frame (SF)
Code (0)
Size (0-7)
Data (e.g. UDS request SID, UDS sub function byte, UDS DID, padding)
First Frame (FF)
Code (1)
Size (8-4095)
Data (e.g. UDS response SID, UDS DID, initial payload)
Consecutive Frame (CF)
Code (2)
Index (1-15)
Data (e.g. payload, padding)
Flow Control Frame (FC)
Code (3)
Flag (0-2)
Block size
Separation time (ST)
Padding

Unified Diagnostic Services

UDS service identifiers (SIDs)

	UDS SID (request)	UDS SID (response)	Service	Details
Diagnostic and Communications Management	0x10	0x50	Diagnostic Session Control	Control which UDS services are available
	0x11	0x51	ECU Reset	Reset the ECU ("hard reset", "key off", "soft reset")
	0x27	0x67	Security Access	Enable use of security-critical services via authentication
	0x28	0x68	Communication Control	Turn sending/receiving of messages on/off in the ECU
	0x29	0x69	Authentication	Enable more advanced authentication vs. 0x27 (PKI based exchange)
	0x3E	0x7E	Tester Present	Send a "heartbeat" periodically to remain in the current session
	0x83	0xC3	Access Timing Parameters	View/modify timing parameters used in client/server communication
	0x84	0xC4	Secured Data Transmission	Send encrypted data via ISO 15764 (Extended Data Link Security)
	0x85	0xC5	Control DTC Settings	Enable/disable detection of errors (e.g. used during diagnostics)
	0x86	0xC6	Response On Event	Request that an ECU processes a service request if an event happens
	0x87	0xC7	Link Control	Set the baud rate for diagnostic access
Data Transmission	0x22	0x62	Read Data By Identifier	Read data from targeted ECU - e.g. VIN, sensor data values etc.
	0x23	0x63	Read Memory By Address	Read data from physical memory (e.g. to understand software behavior)
	0x24	0x64	Read Scaling Data By Identifier	Read information about how to scale data identifiers
	0x2A	0x6A	Read Data By Identifier Periodic	Request ECU to broadcast sensor data at slow/medium/fast/stop rate
	0x2C	0x6C	Dynamically Define Data Identifier	Define data parameter for use in 0x22 or 0x2A dynamically
	0x2E	0x6E	Write Data By Identifier	Program specific variables determined by data parameters
	0x3D	0x7D	Write Memory By Address	Write information to the ECU's memory
DTCs	0x14	0x54	Clear Diagnostic Information	Delete stored DTCs
	0x19	0x59	Read DTC Information	Read stored DTCs, as well as related information
	0x2F	0x6F	Input Output Control By Identifier	Gain control over ECU analog/digital inputs/outputs
	0x31	0x71	Routine Control	Initiate/stop routines (e.g. self-testing, erasing of flash memory)
Upload/ Download	0x34	0x74	Request Download	Start request to add software/data to ECU (incl. location/size)
	0x35	0x75	Request Upload	Start request to read software/data from ECU (incl. location/size)
	0x36	0x76	Transfer Data	Perform actual transfer of data following use of 0x74/0x75
	0x37	0x77	Request Transfer Exit	Stop the transfer of data
	0x38	0x78	Request File Transfer	Perform a file download/upload to/from the ECU
		0x7F	Negative Response	Sent with a Negative Response Code when a request cannot be handled

- Enables diagnostic, firmware updates and overall testing,
- Applicative layer that works on top of CAN & ISO-TP in most cases,
- Present in every recent car (as far as we know)
- Relies on different level-access AKA sessions,
- Sessions usually protected by authentication mechanism (0x21, 0x27)

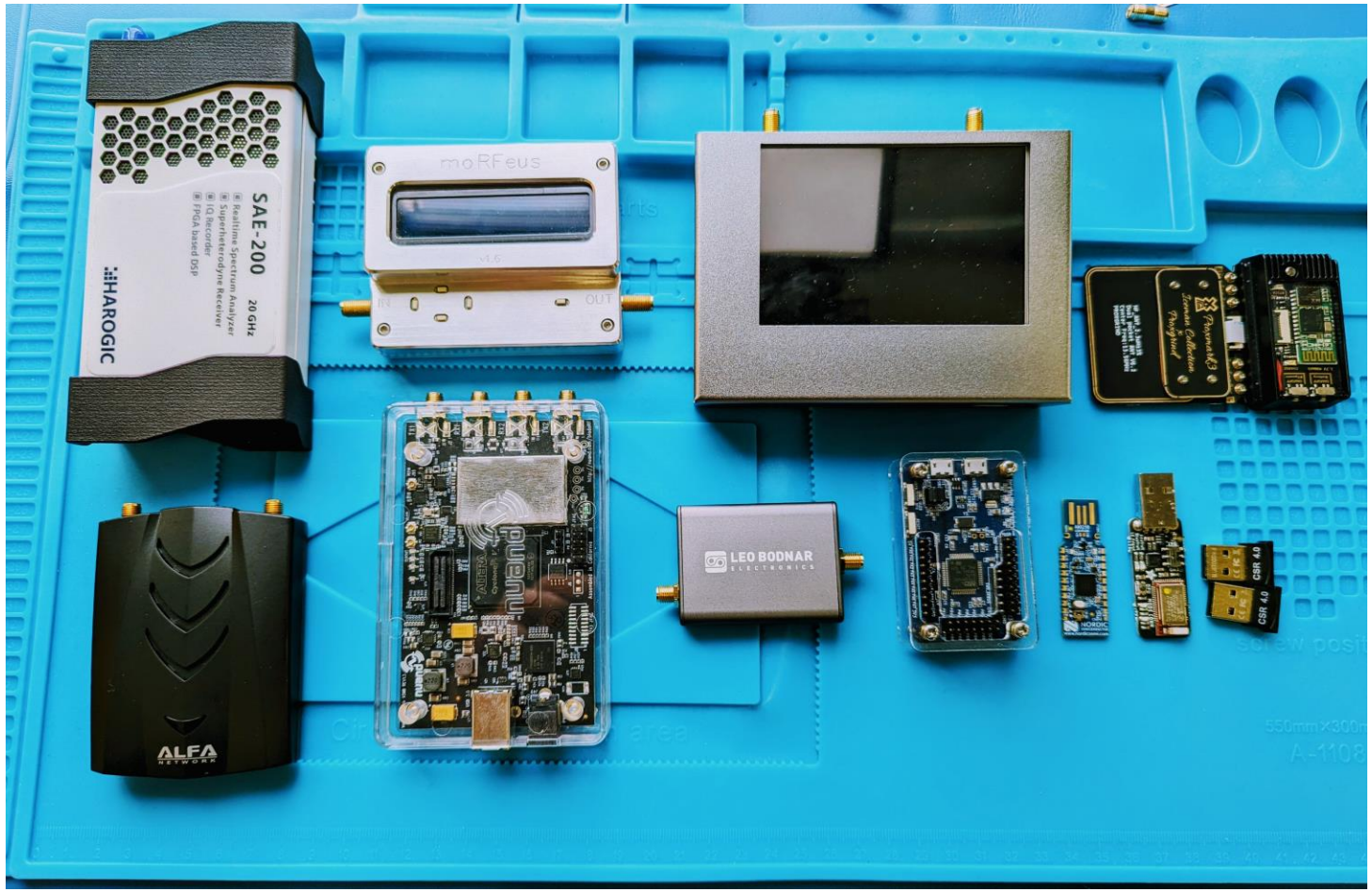
If it can be sent, it can be fuzzed

- Introducing CarZombie =)
- Release date TBD,
- Can fuzz CAN messages, UDS messages (not super well yet),
- Convenient UI, simple to understand,
- Planned to add support for SDR attacks as well,

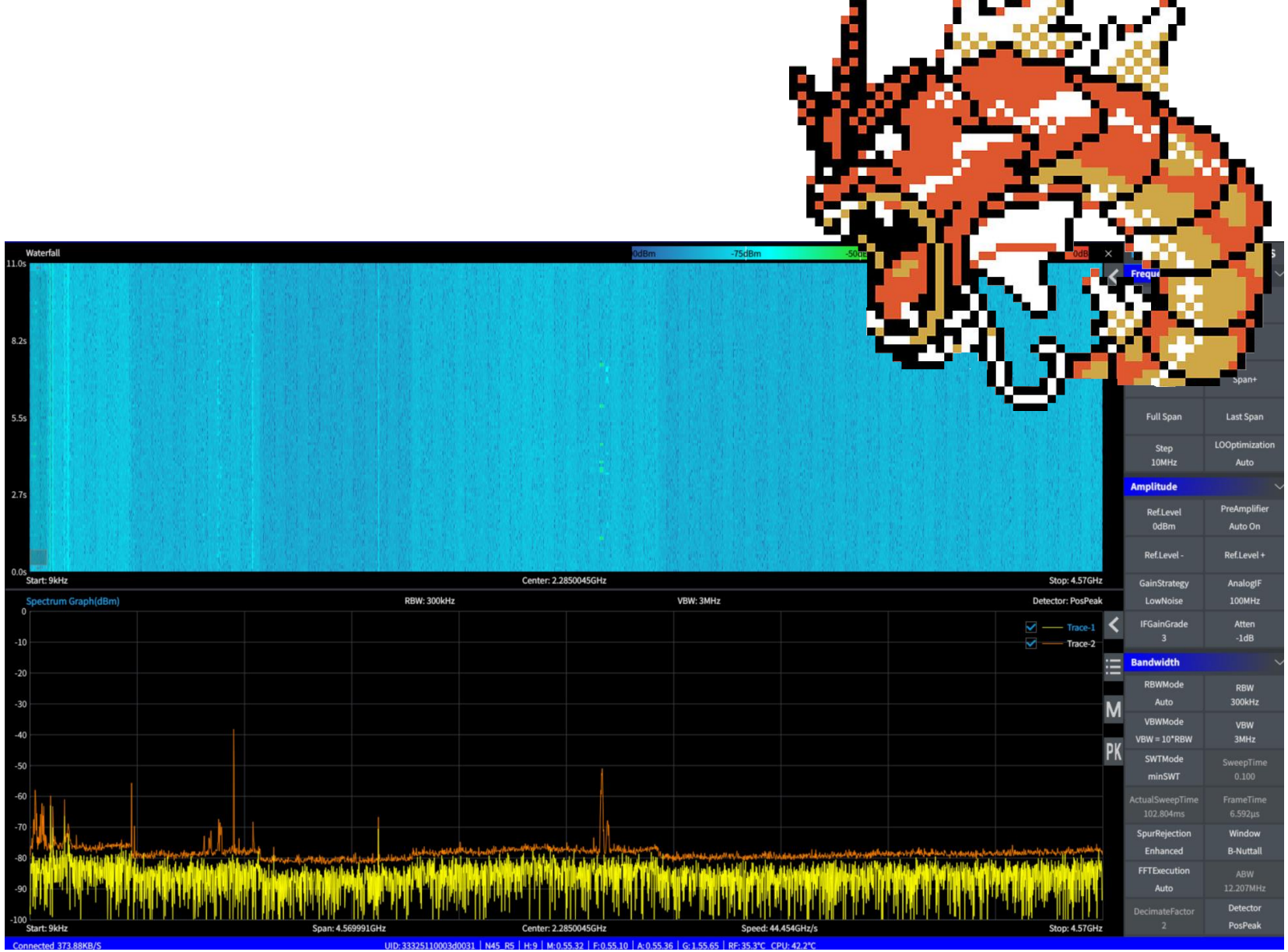
Wireless interfaces

Setup to PWN the radio

Essential kit: keep it simple



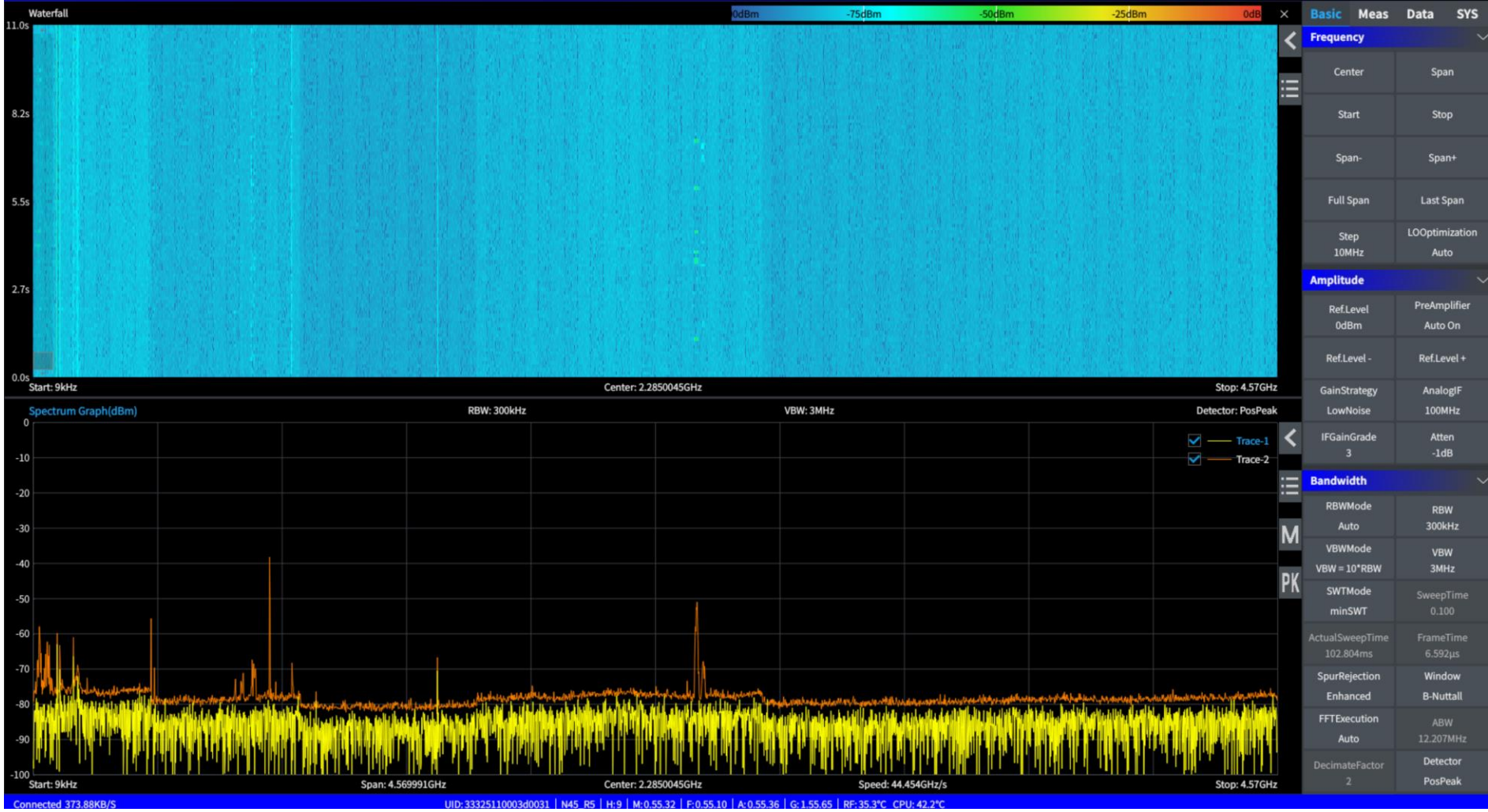
Using SA


Waterfall
11.0s
8.2s
5.5s
2.7s
0.0s
Start: 9kHz
Center: 2.2850045GHz
Stop: 4.57GHz
Spectrum Graph(dBm)
RBW: 300kHz
VBW: 3MHz
Detector: PosPeak
Trace-1
Trace-2
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
Start: 9kHz
Span: 4.569991GHz
Center: 2.2850045GHz
Speed: 44.454GHz/s
Stop: 4.57GHz
Connected 373.88KB/S
Full Span
Last Span
Step 10MHz
LOOptimization Auto
Amplitude
Ref.Level 0dBm
PreAmplifier Auto On
Ref.Level -
Ref.Level +
GainStrategy LowNoise
AnalogIF 100MHz
IFGainGrade 3
Atten -1dB
Bandwidth
RBWMode Auto
RBW 300kHz
VBWMode VBW = 10*RBW
VBW 3MHz
SWTMode minSWT
SweepTime 0.100
ActualSweepTime 102.804ms
FrameTime 6.592µs
SpurRejection Enhanced
Window B-Nuttall
FFTExecution Auto
ABW 12.207MHz
DecimateFactor 2
Detector PosPeak
M
PK

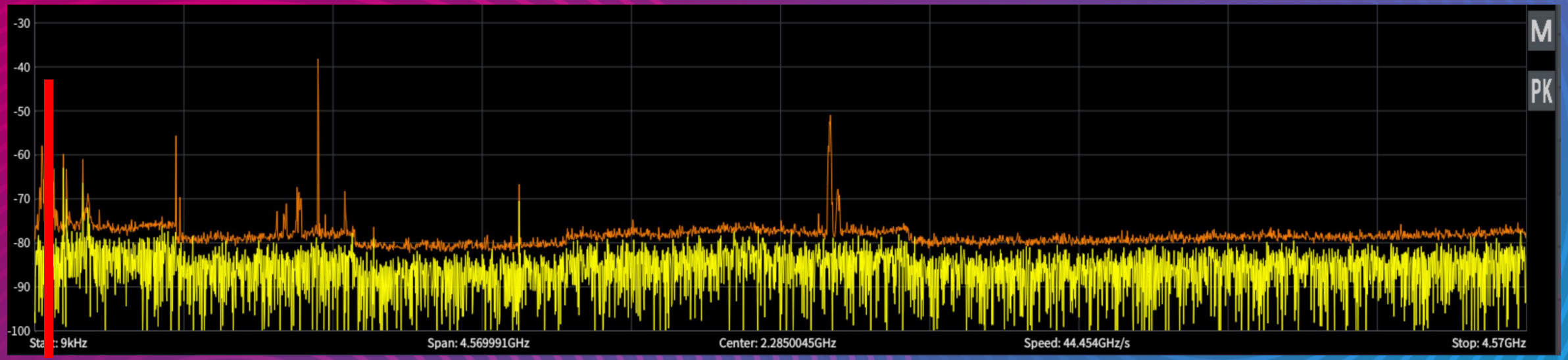
Setup

Using SA

GOTTA CATCH EM ALL!



125 KHz


-30
-40
-50
-60
-70
-80
-90
-100
M
PK
Sta : 9kHz
Span: 4.569991GHz
Center: 2.2850045GHz
Speed: 44.454GHz/s
Stop: 4.57GHz

1215 kHz

TPMS triggers

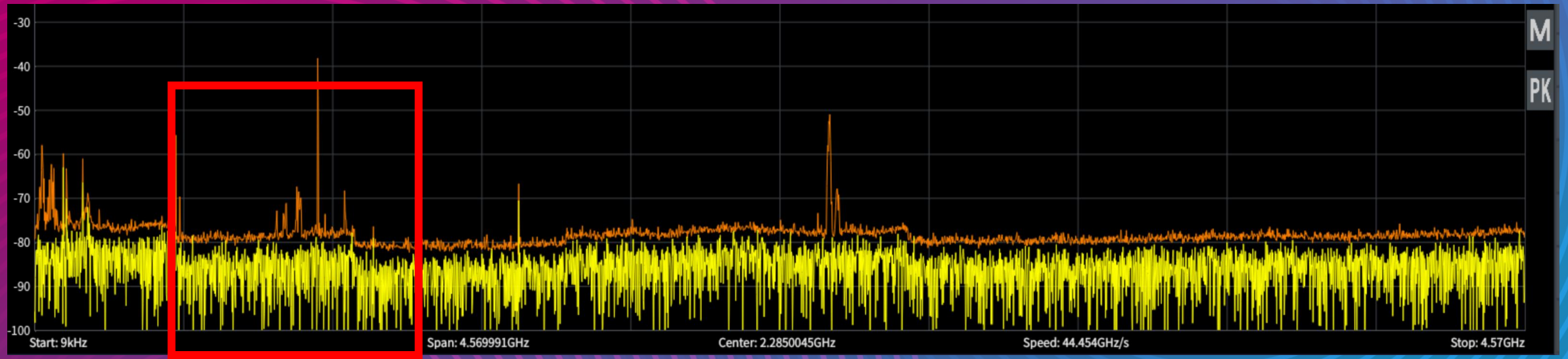
- Used to wake-up TPMS sensors
- Sensors: Frequency bands -> ISM bands of the country mostly:
  - 433 MHz / 868 MHz in EU
  - 315 MHZ / 433 MHz in US
  - Etc.
- Modulations:
  - ASK: Amplitude Shift Key
  - or 2-FSK/BFSK
  - or both (hybrid)

TPMS reader/trigger

Car transponders on Hitag2 crypto

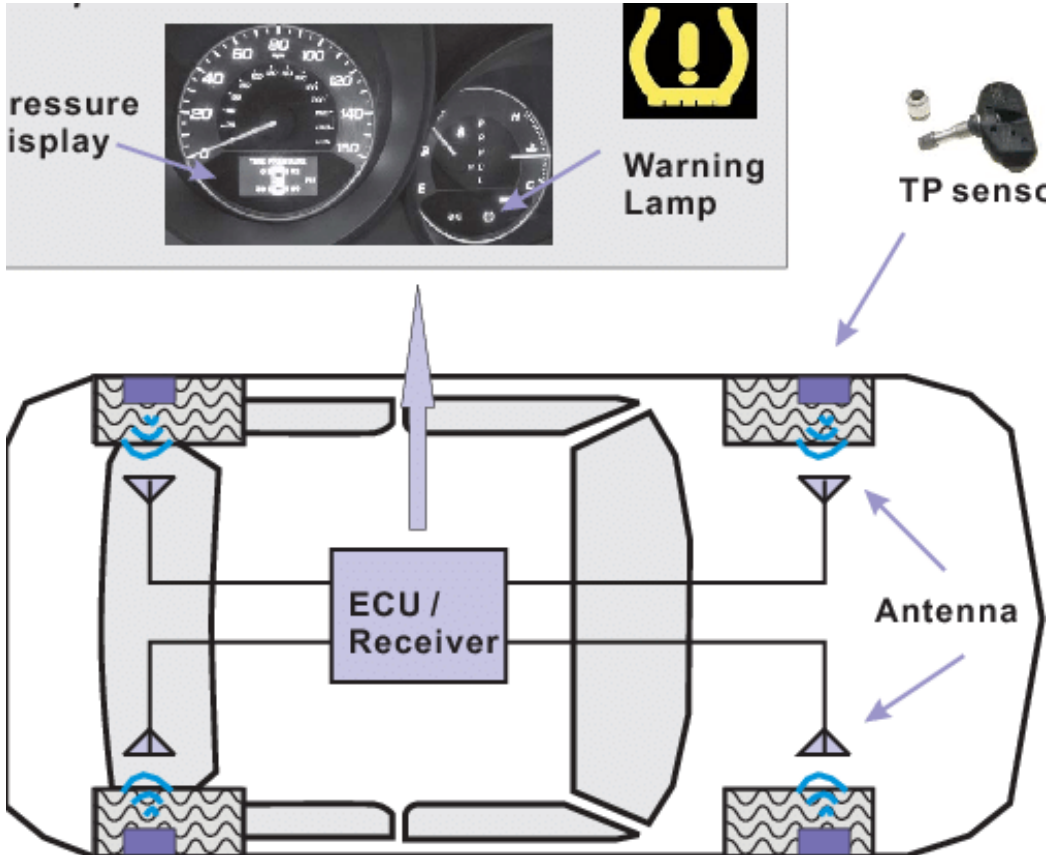
- Authentication to run the car
- State of the art:
  - Gone in 360 Seconds: Hijacking with Hitag2 by Roel Verdult, Flavio D. Garcia, and Josep Balasch (https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final95.pdf)
  - Newer content on Hitag2:
    - Cracking HiTag2 Crypto – Weaponising Academic Attacks for Breaking and Entering by Kevin Sheldrake by Kevin Sheldrake

Sub-GHz ISM bands



TPMS

- TPMS (Tire Pressure Monitoring System)
- 2 types/technologies:
  - Indirect → measurement of each wheel rate revolution
  - Direct → actual pressure level measurement

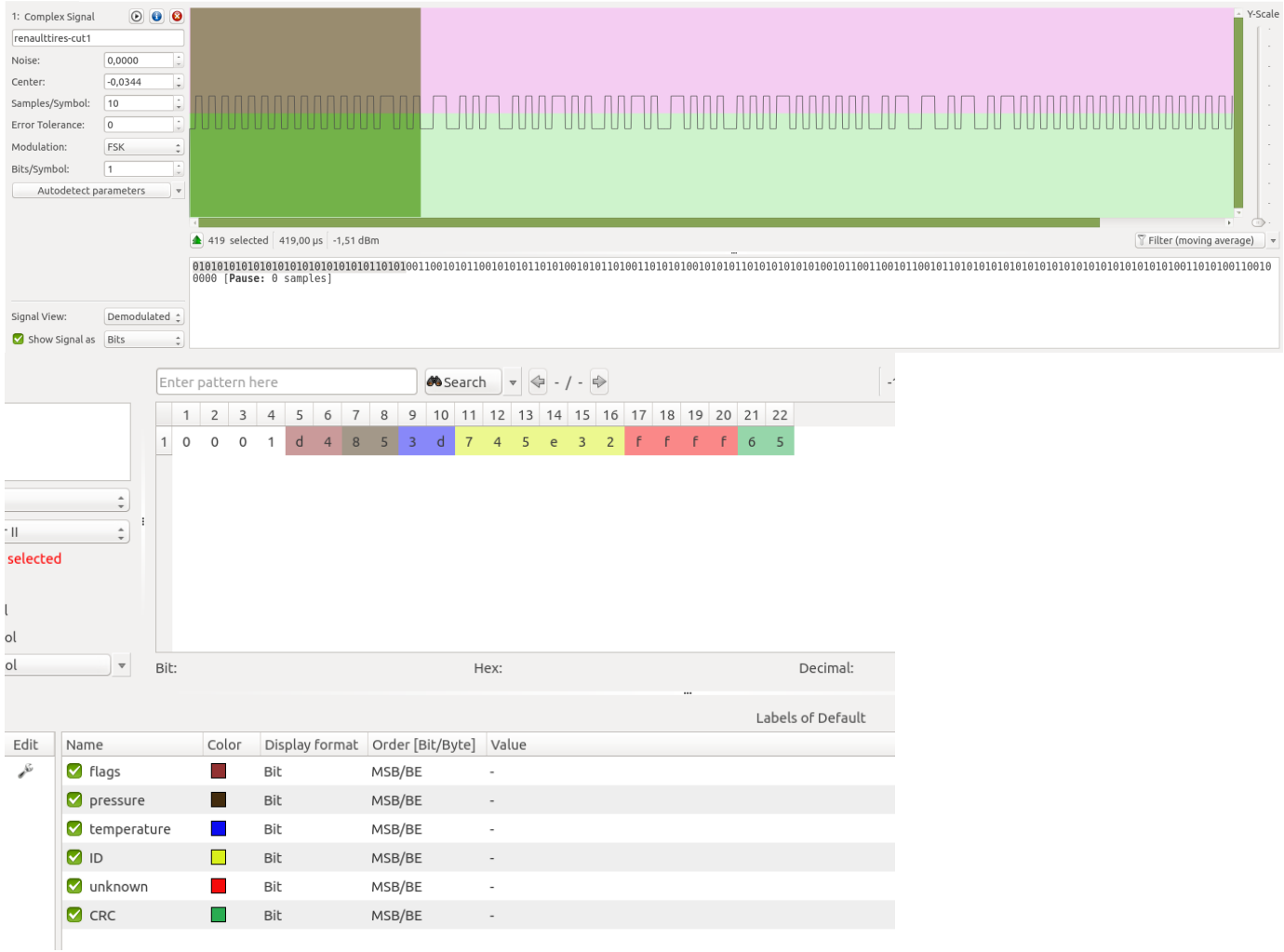


TPMS architecture with four antennas (source: [1])

[1] Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Stud byIshtiaq Rouf et al.

Demodulating data

- Quick way with URH:



Tesla != uses BLE = more fun?

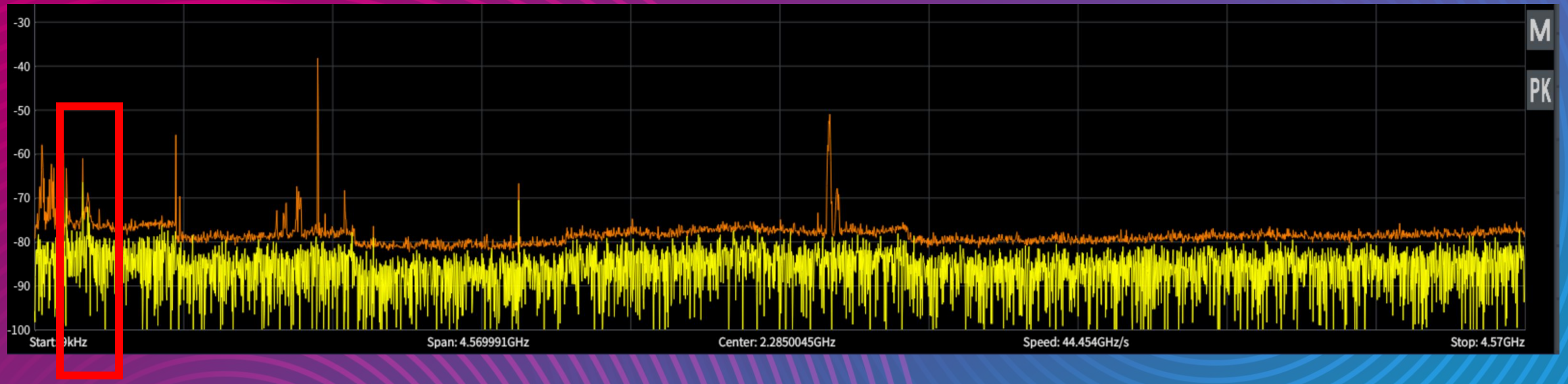
Risks on traditional TPMS

- Mostly Tracking
- Impersonating sensors → stopping the vehicle
- or raising (crazy) notifications → driver in pain
- Crashes with unknown elements, or unwell handled values
- But not easy to trigger on the road:
  - Need to be in range, or transmit a signal with a decent gain → directional antenna + LNA

Attacks on Remote keyless Entry

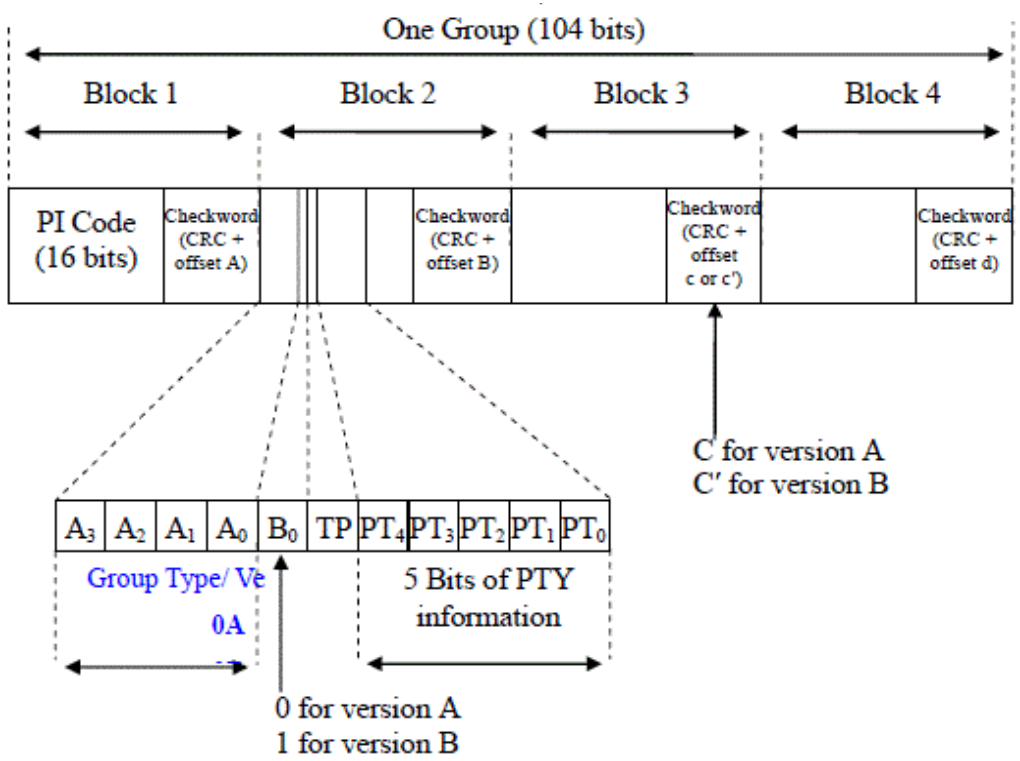
- Different modes:
  - Fixed code → old & rare today --> replay
  - Rolling code --> breaking the manufacturer key
    - Attacks:
      - Hitag2: From Academia to Real World : a Practical Guide to Hitag-2 RKE System Analysis by Ryad Benadjila, Mathieu Renard, José Lopes-Esteves, Chaouki Kasmi
      - DST80: Dismantling DST80-based Immobiliser Systems by Lennert Wouters, Jan Van den Herrewegen, Flavio D. Garcia, DavidOswald, Benedikt Gierlichs and Bart Preneel
      - Rollback attacks: https://rollingpwn.github.io/rolling-pwn/
  - IFF (Identify Friend or Foe)
  - UWB --> Interference Attacks
    - GoGoBark:Interference Attacks onUWBRangingfor IEEE 802.15.4z Standard by Yuqiao Yang & Zhongjie Wu

Radio FM bands



RDS

- Radio Data System (Radio Broadcast Data System (RBDS) for the U.S. version)
- Embeds digital information in FM radio broadcast
- Uses BPSK
- Structure:
  - PI: Program ID code
  - TP: Traffic Program code
  - PTY: Program Type code
  - **TA: Traffic Announcement**
  - Etc.



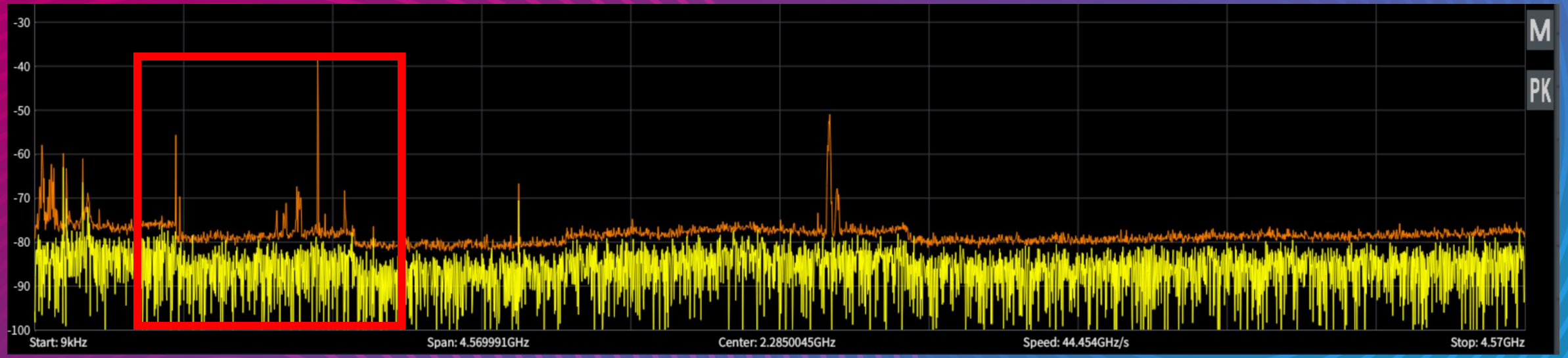
Go further by Friedt Jean-Michel: https://connect.ed-diamond.com/GNU-Linux-Magazine/glmf-204/radio-data-system-rds-analyse-du-canal-numerique-transmis-par-les-stations-radio-fm-commerciales-introduction-aux-codes-correcteurs-d-erreur

RDS Alerts

- With a modified version → fuzzing:
- PI
- TP
- PTY
- Etc.
- **But also, TMC events ☺**

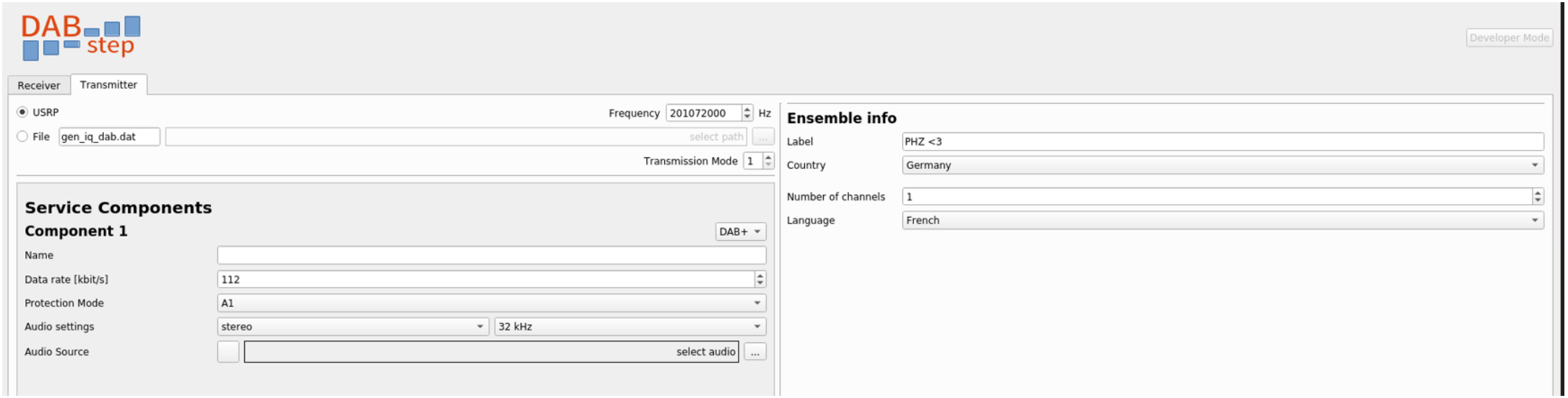

```
{"1168","security alert","1515"," "},
{"1169","security incident","1476"," "},
{"1170","police checkpoint","1477"," "},
{"1171","bomb alert","1516"," "},
{"1172","terrorist incident","1478"," "},
{"1173","gunfire on roadway, danger","1479"," "},
{"1174","civil emergency","1480"," "},
{"1175","air raid, danger","1481"," "},
{"1176","evacuation","1494"," "},
{"1177"," "," "," "},
{"1178","air raid warning cancelled","1587"," "},
{"1179","security alert withdrawn","1492"," "},
{"1180","civil emergency cancelled","1588"," "},
{"1181"," "," "," "},
```

Band III & L-bands



DAB

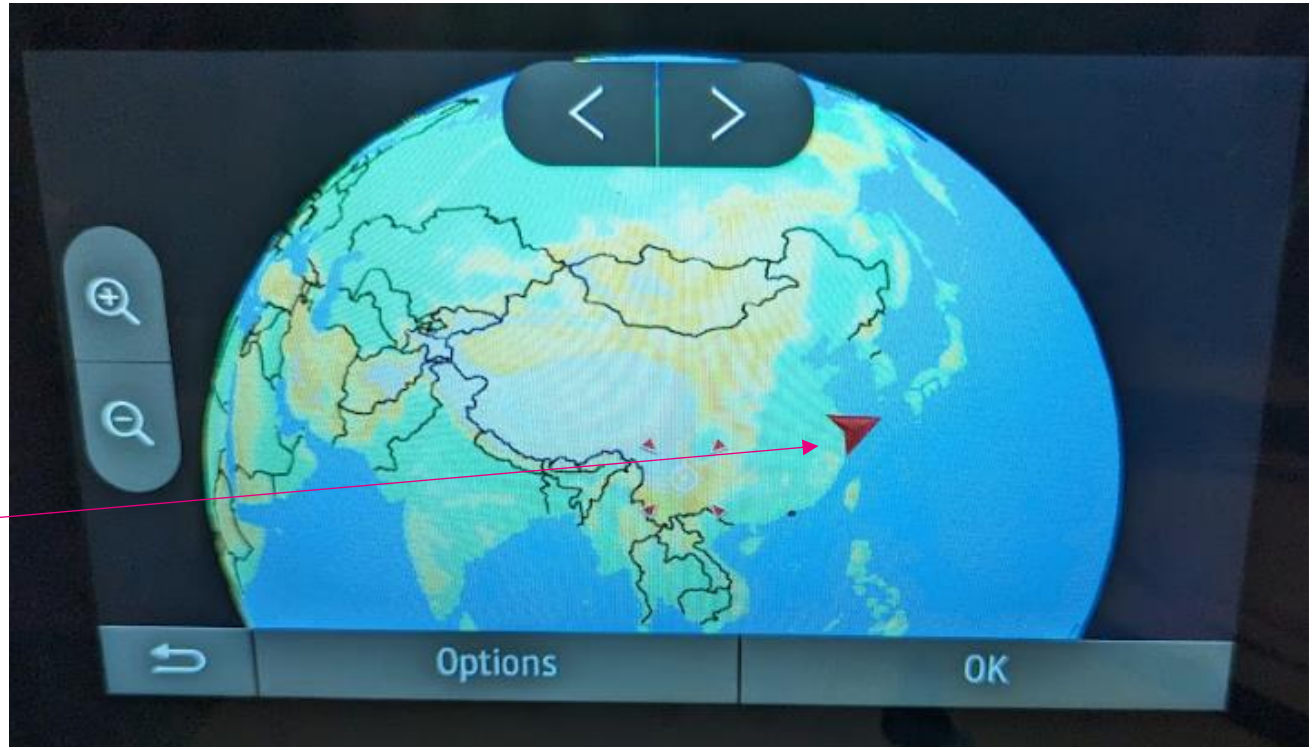
- Digital Audio Broadcasting
- DAB+ → upgrades for more stations with HD quality
- Tool for injection to modify:



Hijacking in action

Hijacking in action (2)

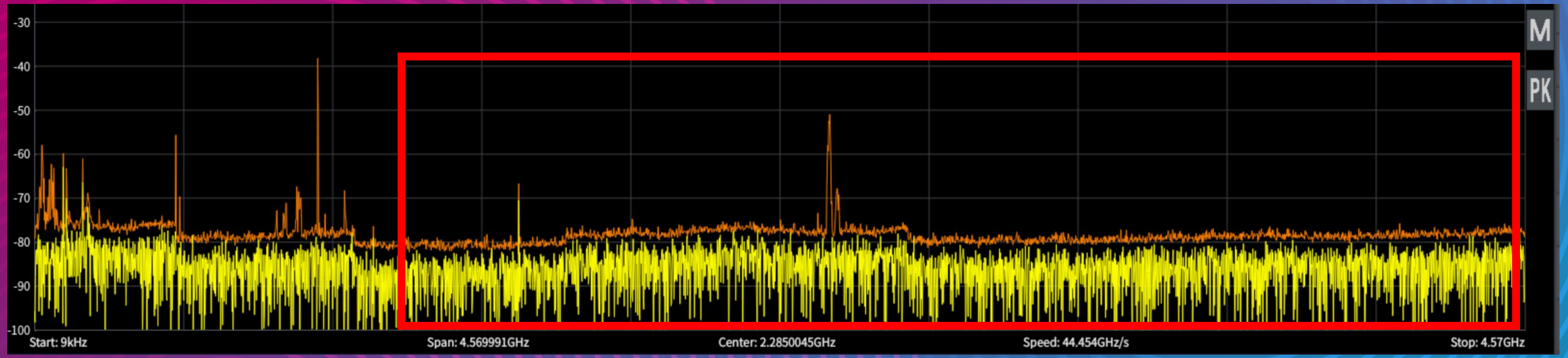
- The signal GPS can be hijacked
- Some GPS receivers look at how strong the signal is + other mechanisms to avoid this
- But doing that in the right way, it's still possible to teleport!



Hijacking vs Autopilot

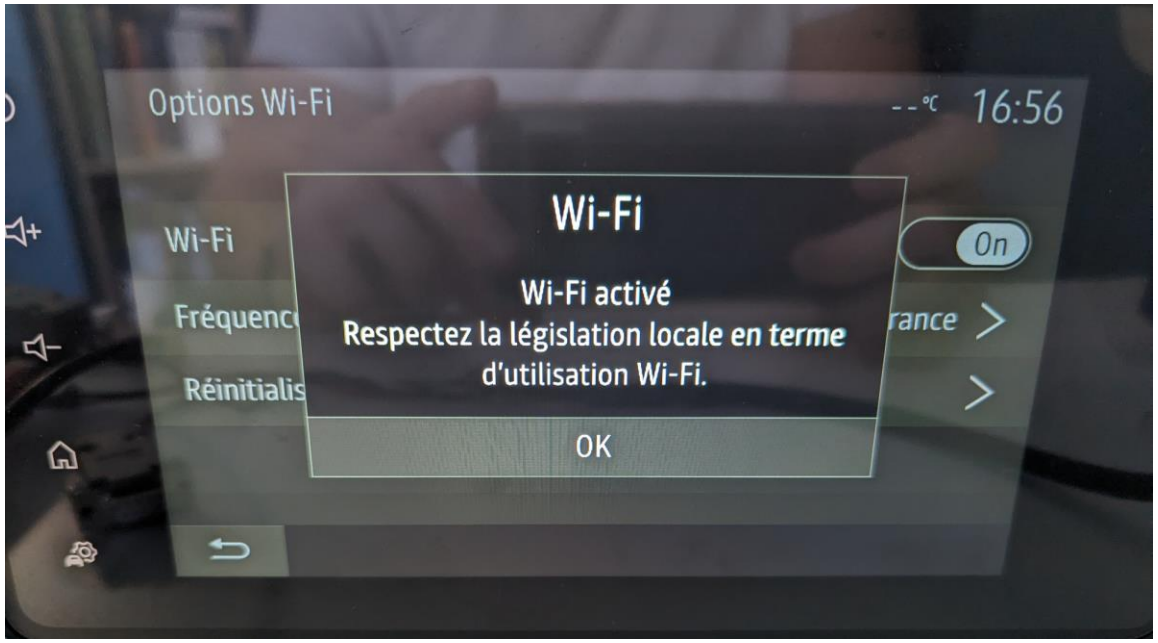
- Question: What about Autopilot?

> 1 GHz – ISM bands



Wi-Fi 2.4/5 GHz and more

- IVI gives a hotspot
- A WPA2 PSK is randomly generated
- After pairing the mobile phone in BT classic -> PSK key exchange through BT
- The hotspot exposes some interesting service
  - MirrorLink like servers (e.g: https://www.usenix.org/conference/woot16/workshop-program/presentation/mazloom)
  - Services also available on mobile, USB OTG, and/others...



Recurrent candidates

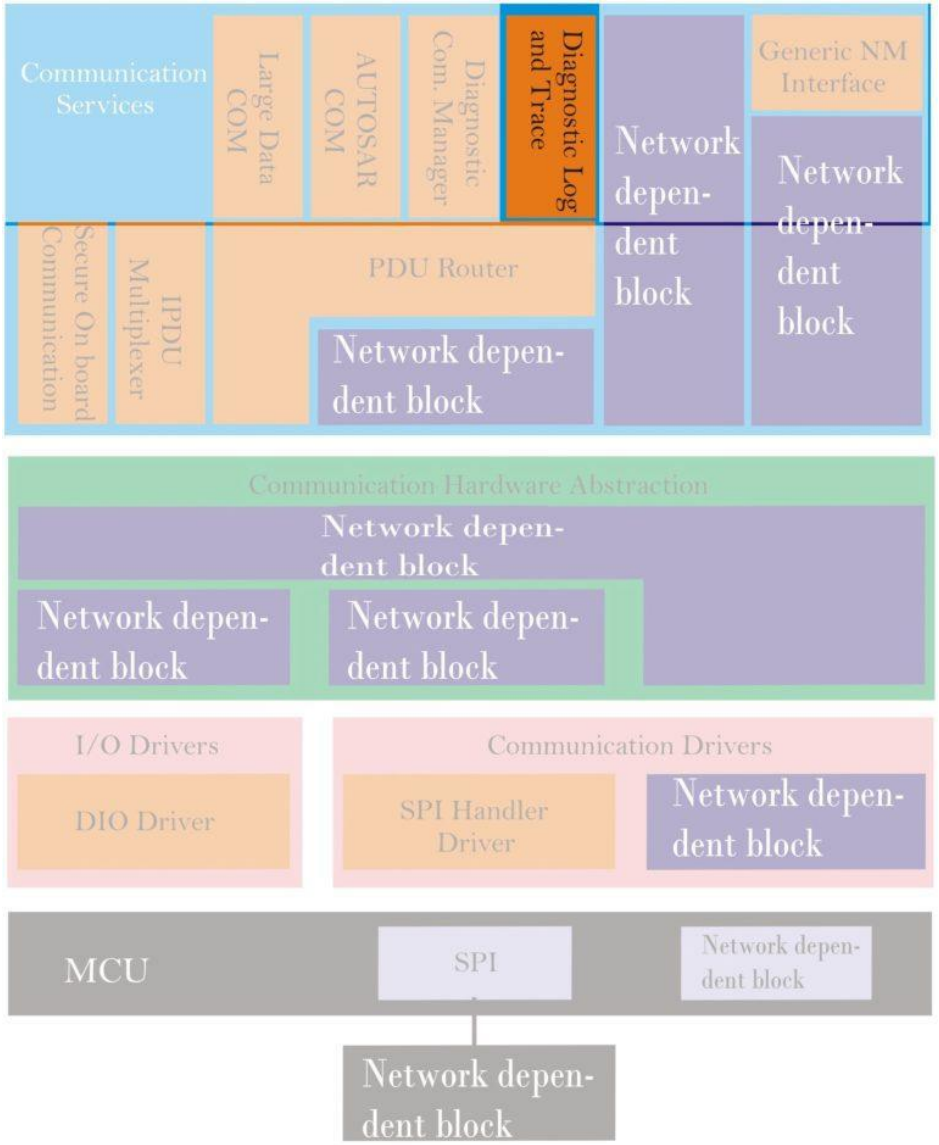
- QNX in uses:
  - Look at exposed qconn service ☺ (good old trick! But with a little update)

```
user@testlab:~$ telnet
telnet> open 192.168.86.125 8000 # target's IP address
Trying 192.168.86.125...
Connected to 192.168.86.125.
Escape character is '^]'.
QCONN
<qconn-broker> service launcher
OK
<qconn-launcher> start/flags run /sbin/shutdown -b
OK 970775
^[[3~^M^MConnection closed by foreign host.
```

DLT?

- Diagnostic Log and Trace
- Sender-receiver communication
- See more:

https://autosartutorials.com/diagnostic-log-and-trace/



DLT traces

- Can trace:
  - Events
  - Crashes
  - Running processes

Index	Time	Timestamp	Ecuid	Apid	Ctid	Type	Payload
1 615	202	5.5...	936.2404	RADI	RADI	log	void quint16, const QString&, int, bool) UpdatePresetView: freq 9840 PI 65158 PSN YVELINES
1 616	202	5.5...	936.2405	RADI	RADI	log	QList hannelsInPreset(quint16, quint16, bool) FindChannelsInPreset: Freg = 9840 - PI = 65158 - bRdsEnabled = 1 - AF op
1 617	202	5.5...	936.2406	RADI	RADI	log	void (bool) Set RemovePresetByPiUpdated: 1 -> 1
1 618	202	5.5...	936.2406	RADI	RADI	log	QList hannelsInPreset(quint16, quint16, bool) FindChannelsInPreset: Freg = 9840 - PI = 65158 - bRdsEnabled = 1 - AF op
1 619	202	5.5...	936.2407	RADI	RADI	log	void quint16, const QString&, int, bool) Not found match item preset
1 620	202	5.5...	936.2407	RADI	RADI	log	void , bool) YVELINES
1 621	202	5.5...	936.2407	RADI	RADI	log	void , bool) Delay 200ms to sending media info
1 622	202	5.5...	936.2408	RADI	RADI	log	void FM: YVELINES -> 98.4 FM
1 623	202	5.5...	936.2409	RADI	RADI	log	Radic nList(quint16, quint16, bool) FindChannelsInList: Freg = 9840 - PI = 65158 - bRdsEnabled = 1 - AF opt = 1
1 624	202	5.5...	936.2409	RADI	RADI	log	void quint16, const QString&, int, bool) UpdatePresetView: freq 9840 PI 65158 PSN 98.4 FM
1 625	202	5.5...	936.2411	RADI	RADI	log	QList hannelsInPreset(quint16, quint16, bool) FindChannelsInPreset: Freg = 9840 - PI = 65158 - bRdsEnabled = 1 - AF op
1 626	202	5.5...	936.2411	RADI	RADI	log	void , bool) 98.4 FM
1 627	202	5.5...	936.2412	RADI	RADI	log	void , bool) Last media info sending was not finished for 200ms, wait
1 628	202	5.5...	936.2542	MM...	MM...	log	Micor d9 00 7d f6 00 00
1 629	202	5.6...	936.3758			control	[]
1 630	202	5.7...	936.4209	MM...	MM...	log	READ)
1 631	202	5.7...	936.4211	MM...	MM...	log	Radic
1 632	202	5.7...	936.4315	RADI	RADI	log	void el info: {"info":"98.4 FM","launch":"com.lge.bavn.appradio","source":"Radio"}
1 633	202	5.7...	936.4335	HO...	INFO	log	[boo String&, const QString&)] isEnable: 1 ~~ source_audio: FM ~~ name_played: 98.4 FM
1 634	202	5.7...	936.4338	HO...	INFO	log	[boo String&, const QString&)] m_listActiveAudioSource: count= 1
1 635	202	5.7...	936.4339	HO...	INFO	log	[boo String&, const QString&)] PopupSystem is displayed, save data to cache!!!
1 636	202	5.7...	936.4363	RADI	RADI	log	void DBusPendingCallWatcher*) Send channel info to home screen successfully
1 637	202	5.7...	936.4374	MIPV	MIPC	log	hand nfo":"98.4 FM","launch":"com.lge.bavn.appradio","source":"Radio"}
1 638	202	5.7...	936.4375	MIPV	MIPC	log	sendI h":"com.lge.bavn.appradio","source":"Radio"}
1 639	202	5.7...	936.4375	MIPV	MIPC	log	Medi unch":"com.lge.bavn.appradio","source":"Radio"})
1 640	202	5.7...	936.4411	RADI	RADI	log	void BusPendingCallWatcher*) Send channel info to navi successfully
1 641	202	5.8...	936.5810	MM...	MM...	log	READ(23) a

- Perfect to debug fuzzing when it's exposed! ☺

DLT RCE?

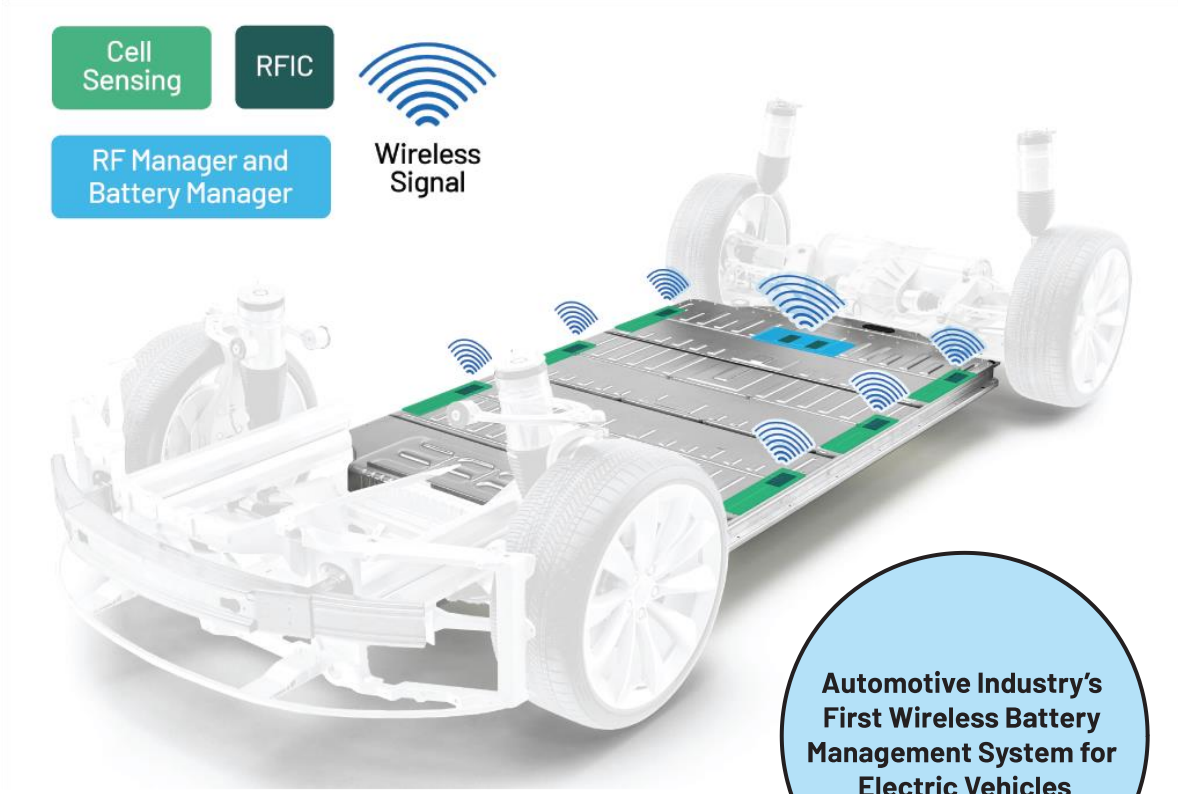
- Interesting function:
  - Possible to reach with right ECU ID + Service ID if the configuration allows!

dlt-daemon / src / system / dlt-system-shell.c

Code | Blame | 121 lines (106 loc) · 4.82 KB

```
87              DLT_STRING("dlt-system-shell, injection data:"),
88              DLT_STRING(text));
89
90      switch (service_id) {
91      case 0x1001:
92
93          if ((syserr = system(text)) != 0)
94              DLT_LOG(shellContext, DLT_LOG_ERROR,
95                      DLT_STRING("dlt-system-shell, abnormal exit status."),
96                      DLT_STRING(text),
97                      DLT_INT(syserr));
98          else
99              DLT_LOG(shellContext, DLT_LOG_INFO,
100                     DLT_STRING("Shell command executed:"),
101                     DLT_STRING(text));
102
103         break;
104     default:
105         DLT_LOG(shellContext, DLT_LOG_ERROR,
106                 DLT_STRING("dlt-system-shell, unknown command received."),
107                 DLT_UINT32(service_id),
108                 DLT_STRING(text));
109         break;
110     }
111
```

Looking forward target: wBMS



https://www.analog.com/en/resources/analog-dialogue/articles/in-the-new-era-of-wireless-battery-management-systems-wbms-security-takes-the-spotlight.html

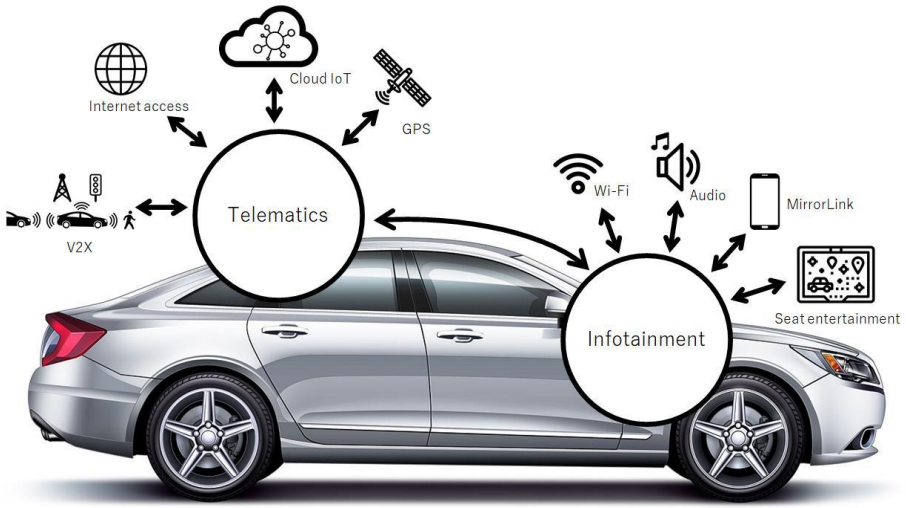
ALCAR BLE sensors for Tesla

- An epic presentation to come:
  - 0-click RCE on Tesla Model 3 through TPMS Sensors by David Berard & Thomas Imbert from Synacktiv
  - https://www.hexacon.fr/conference/speakers/#tesla_model_3

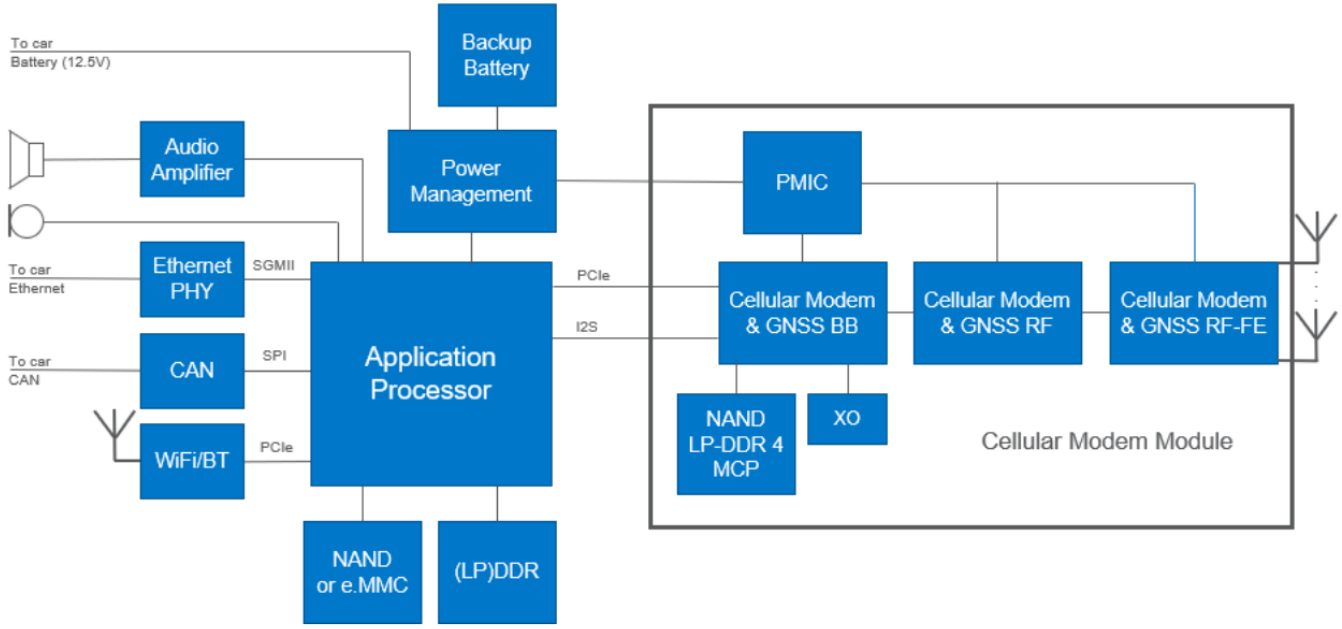
Mobile interface

TCUs with 3G-5G stacks used in cars

- 5G → not very common, but starting to be developed



Source: https://www.i-pex.com/



Source: https://media-www.micron.com/

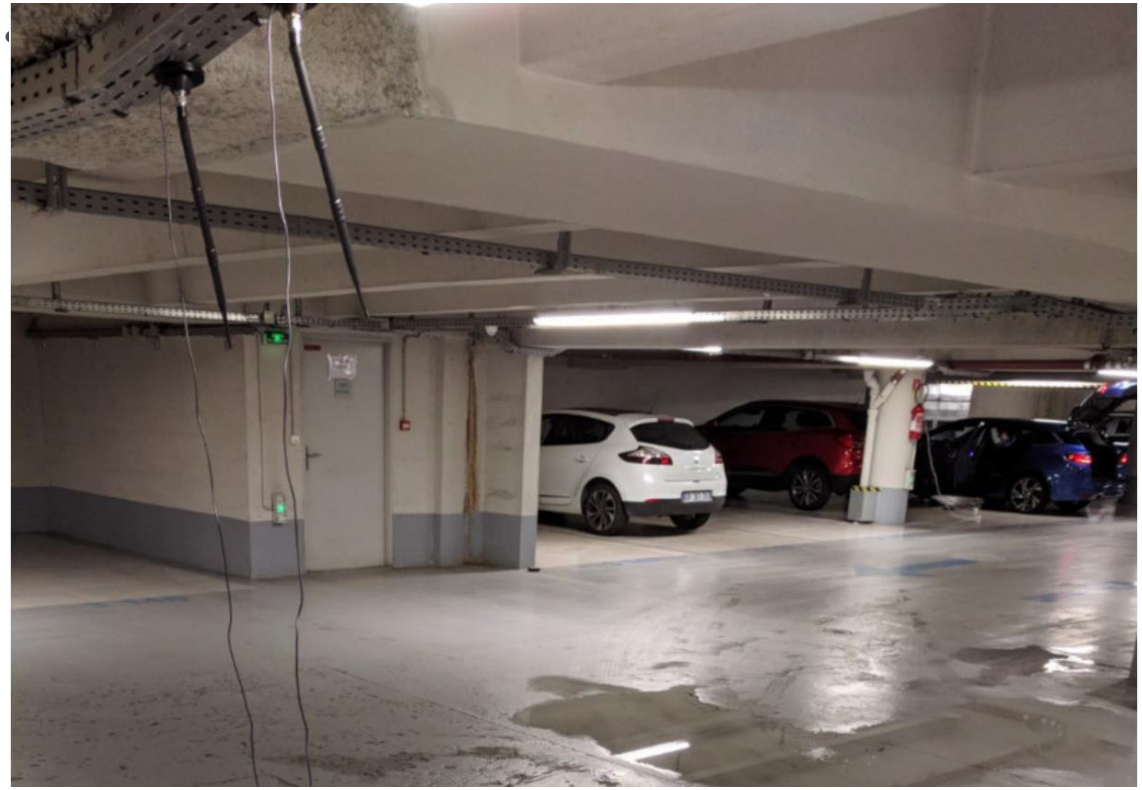
IVI and telematic systems in cars

- Usually use the mobile network:
  - Updates
  - Applications (Twitter, Facebook, etc.)
  - In-car internet
  - Streaming
  - Etc.
- Use GSM/GPRS, 3G, 4G stacks
- New 5G stacks are comming

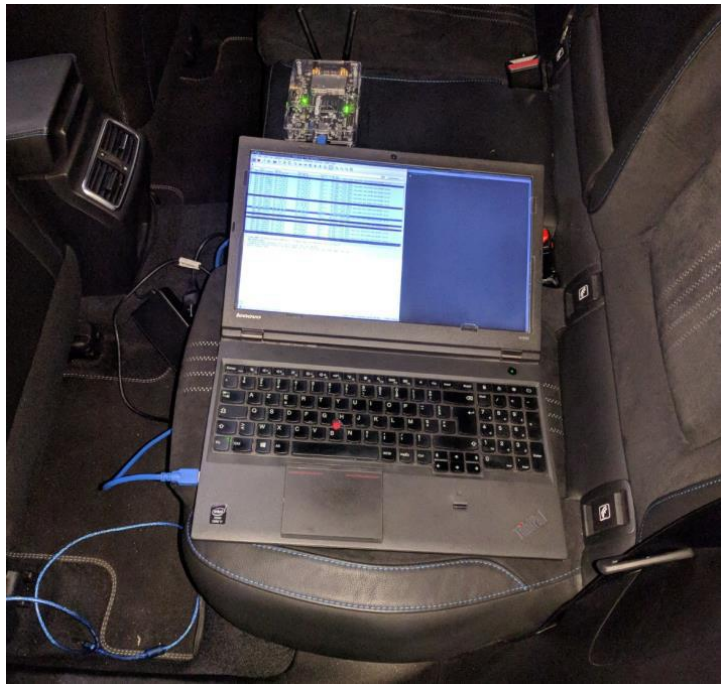
Interception

- Eavesdropping in 2G:
  - no mutual authentication
  - A5/0 can be enforced
- Downgrading from 4G/3G to 2G:
  - Jamming (https://github.com/PentHertz/Modmobjam)
  - Parking places
  - Or protocol attacks (even in 5G, see: Never Let Me Down Again: Bidding-Down Attacks and Mitigations in 5G and 4G" by Bedran Karakoc, Nils Fürste, David Rupprecht, Katharina Kohls from Radix-security)

Or good old parking places

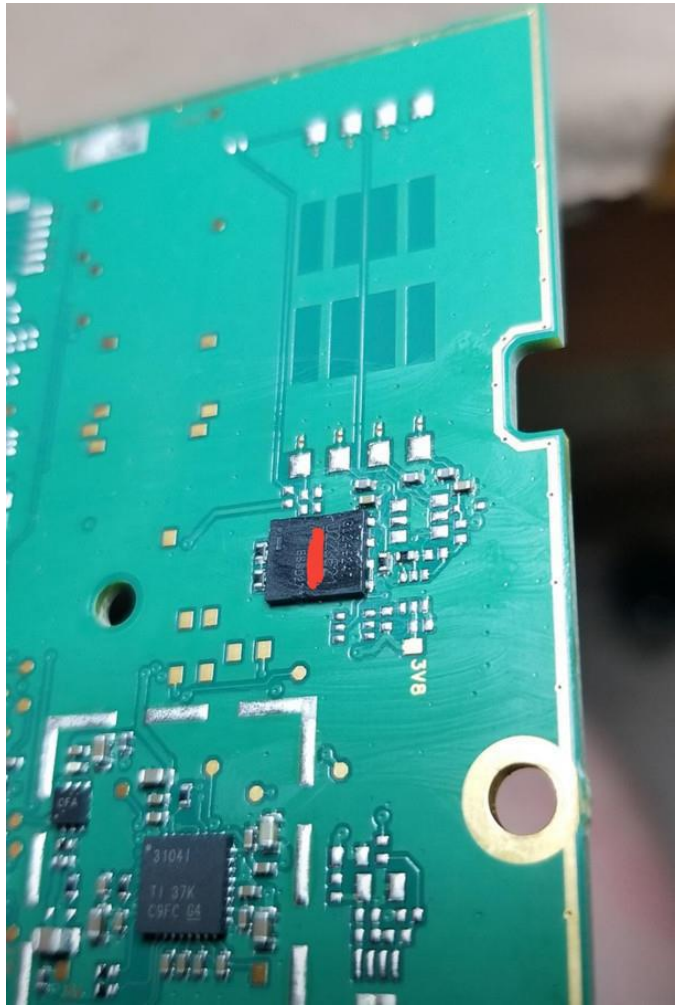


10	1.459318826	192.168.99.2	192.168.99.254	HTTP	913	POST /Service/InitSession/I HTTP/1.1 (applicat
19	7.536599505	192.168.99.2	10.91.80.203	HTTP	52	HEAD http://master.coyoterts.com HTTP/1.1
26	13.660617735	192.168.99.2	10.91.80.203	HTTP	52	HEAD http://master.coyoterts.com HTTP/1.1
65021	922.704281910	192.168.99.2	10.91.80.203	HTTP	52	HEAD http://master.coyoterts.com HTTP/1.1
66923	946.703883356	192.168.99.2	10.91.80.203	HTTP	52	HEAD http://master.coyoterts.com HTTP/1.1
69066	974.461373298	192.168.99.254	192.168.99.2	HTTP	173	HTTP/1.0 404 File not found
69093	974.818419668	192.168.99.2	192.168.99.254	HTTP	52	HEAD http://master.coyoterts.com HTTP/1.1
70396	990.503915759	192.168.99.2	192.168.99.254	HTTP	406	POST /api/app/call HTTP/1.1 (application/x-protobuf)
70401	990.504770592	192.168.99.254	192.168.99.2	HTTP	390	HTTP/1.0 501 Unsupported method ('POST') (text/html)
70459	991.484062985	192.168.99.2	192.168.99.254	HTTP	406	POST /api/app/call HTTP/1.1 (application/x-protobuf)
70462	991.484923306	192.168.99.254	192.168.99.2	HTTP	390	HTTP/1.0 501 Unsupported method ('POST') (text/html)
70530	992.483719425	192.168.99.2	192.168.99.254	HTTP	406	POST /api/app/call HTTP/1.1 (application/x-protobuf)
70533	992.484544176	192.168.99.254	192.168.99.2	HTTP	390	HTTP/1.0 501 Unsupported method ('POST') (text/html)
1048…	1590.1445388…	192.168.99.2	192.168.99.254	HTTP	406	POST /api/app/call HTTP/1.1 (application/x-protobuf)
1048…	1590.1450970…	192.168.99.254	192.168.99.2	HTTP	390	HTTP/1.0 501 Unsupported method ('POST') (text/html)
1048…	1591.0455681…	192.168.99.2	192.168.99.254	HTTP	406	POST /api/app/call HTTP/1.1 (application/x-protobuf)
1048…	1591.0462935…	192.168.99.254	192.168.99.2	HTTP	390	HTTP/1.0 501 Unsupported method ('POST') (text/html)
1049…	1591.8855224…	192.168.99.2	192.168.99.254	HTTP	406	POST /api/app/call HTTP/1.1 (application/x-protobuf)



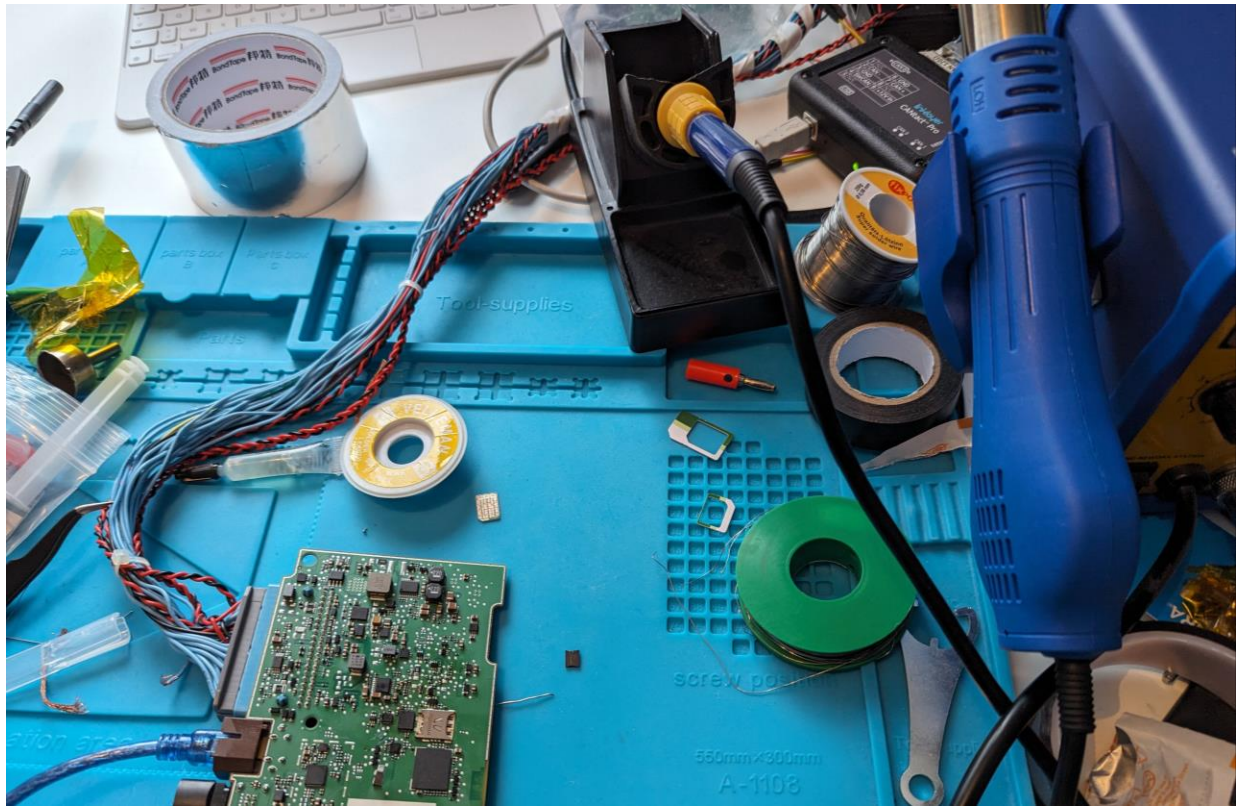
Old Android are also used → choice of RCE

Soldered eUICC



https://f30.bimmerpost.com/forums/showthread.php?t=1642417

Soldered eUICC -> reworking

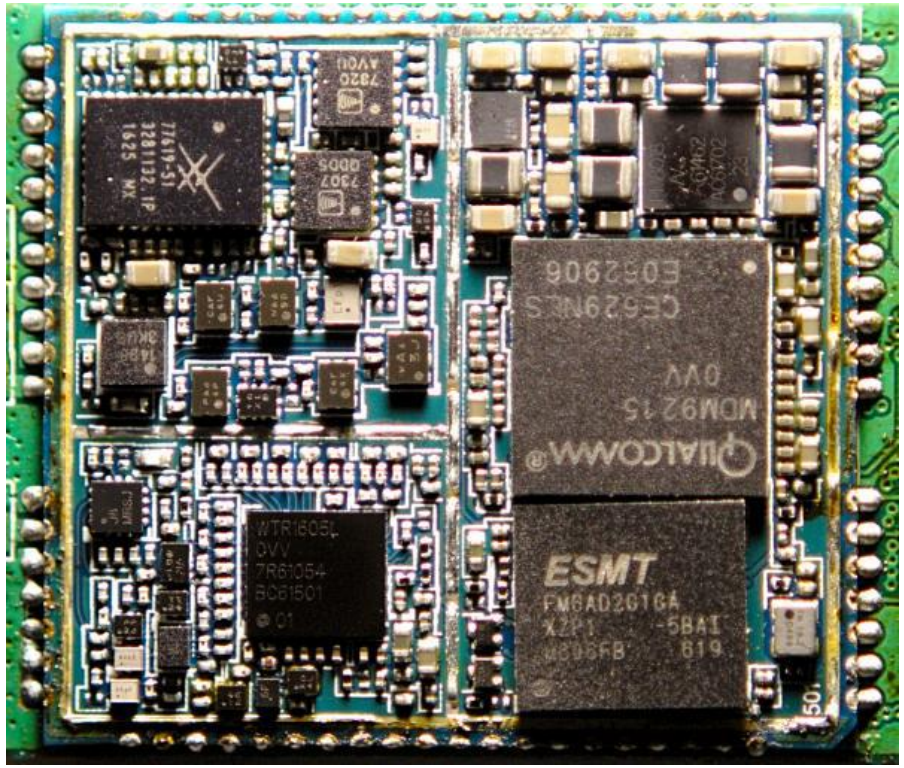


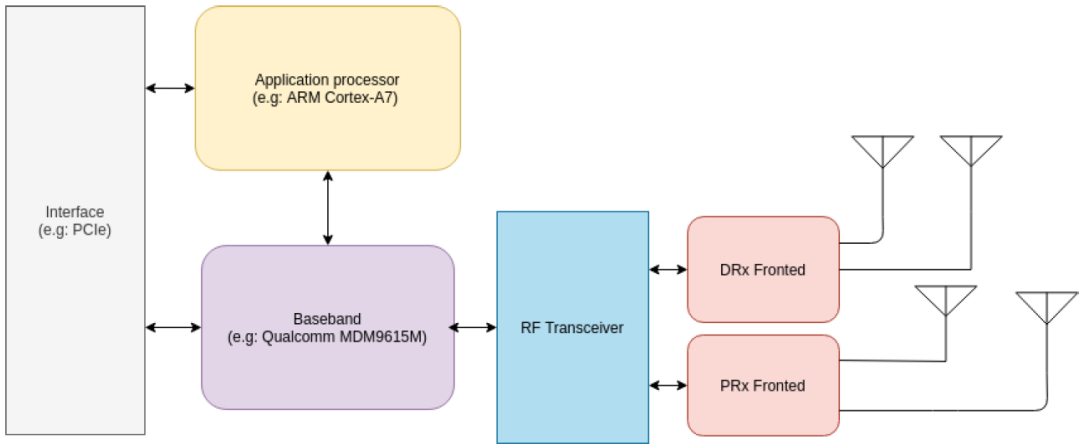
Interception with soldered eUICC

- After desoldering, we can put our custom SIM card
- If IP is whitelisted, we can use the legitimate SIM card with a computer to forward accesses:

Mobile modules

- Used in IoT and cars to communicate with the mobile network





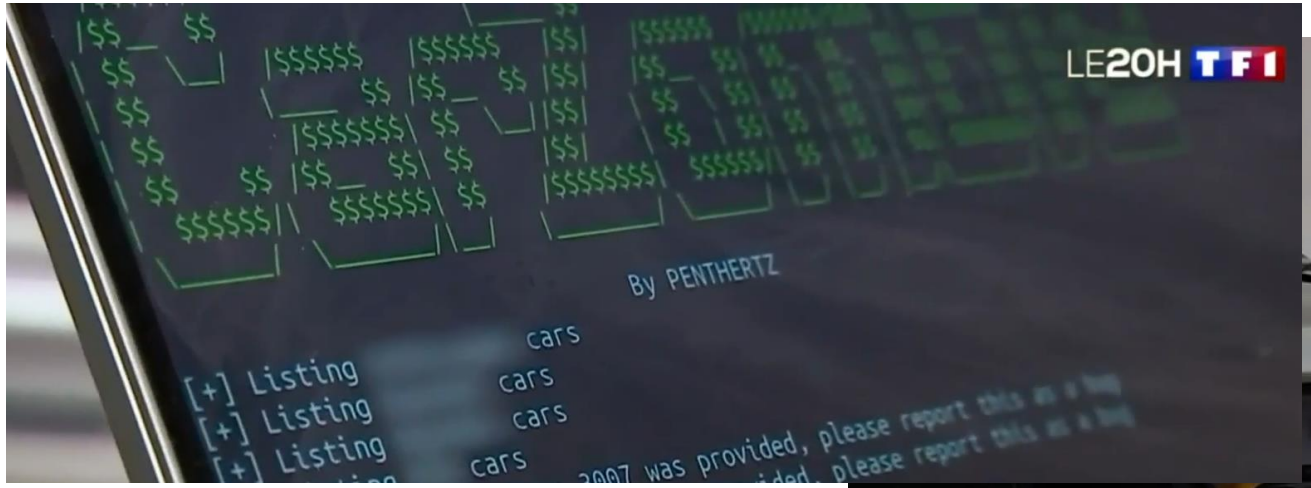
Backdooring servers for FOTA: https://penthertz.com/blog/mobile-iot-modules-FOTA-backdooring-at-scale.html

Attacking backends

Car apps

- Sometimes simpler than cracking RKEs hacking around Object IDs:
  - Remotely flashing the victim's vehicle's headlights
  - Honking the horn
  - Starting or stopping the engine
  - Locking or unlocking the car
  - Changing a PIN
  - Unlocking the boot

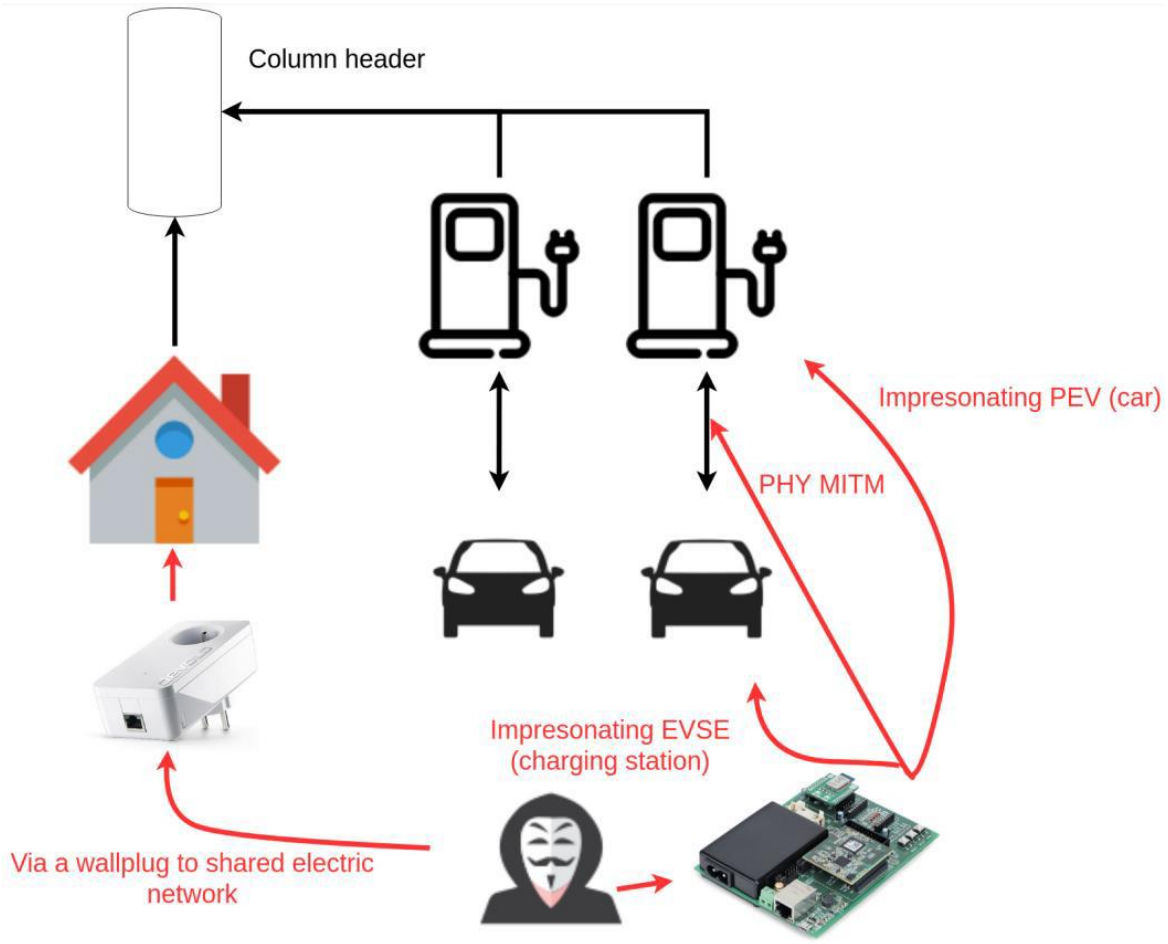
Apps to PWN them all!



V2G

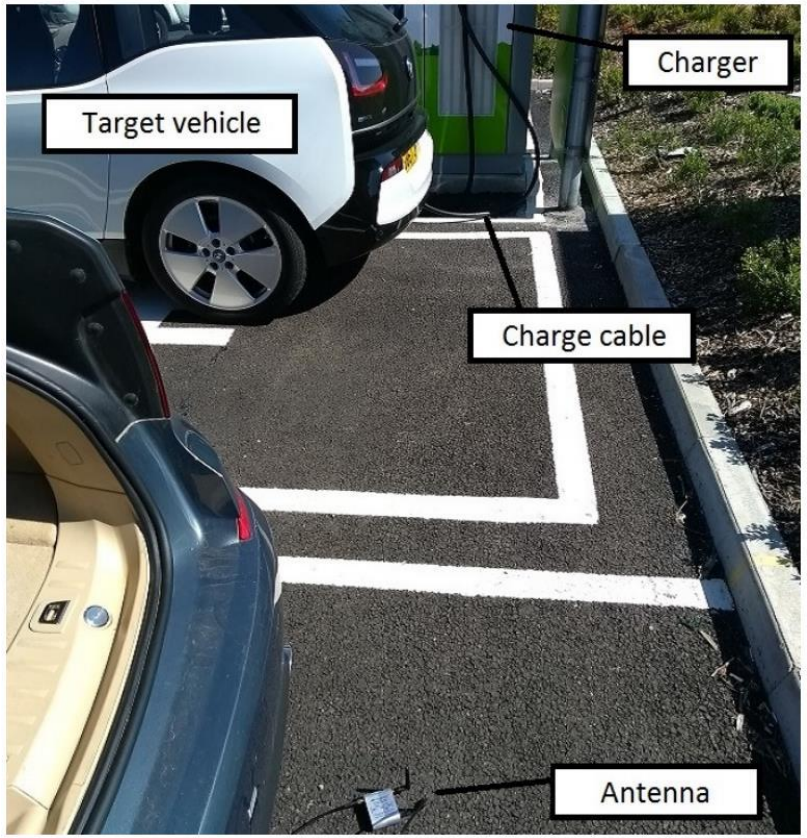
V2G flaws

- Uses HPGP→vulnerable to key collection on powerline
- Security mode not enforced by default→MITM and injection possible
- Downgrade opportunities depending on the configuration/implementation
- Tools:
  - V2G Injector: https://github.com/FlUxIuS/V2GInjector
  - HomePlugPWN: https://github.com/FlUxIuS/HomePlugPWN
- Some other fun triggering Log4shell: https://www.youtube.com/watch?v=k7koOa_S44Y



V2G key collection in radio

- HomePlug AV: hard to get the whole bandwidth with a cheap device
- But HomePlug GreenPHY as less data rate →possible with bladeRF :)

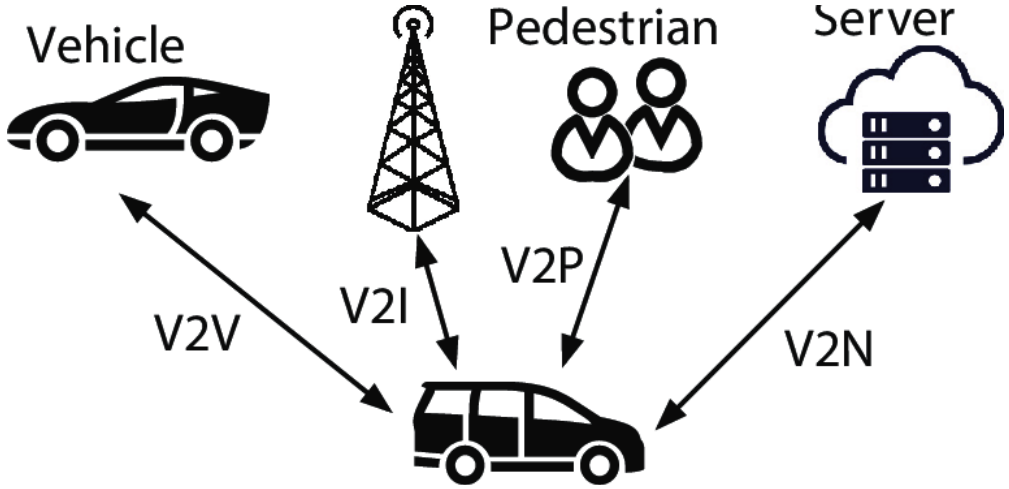


Awesome research!:https://www.usenix.org/system/files/sec19-baker.pdf

(C-)V2X: forward looking research, still

V2X

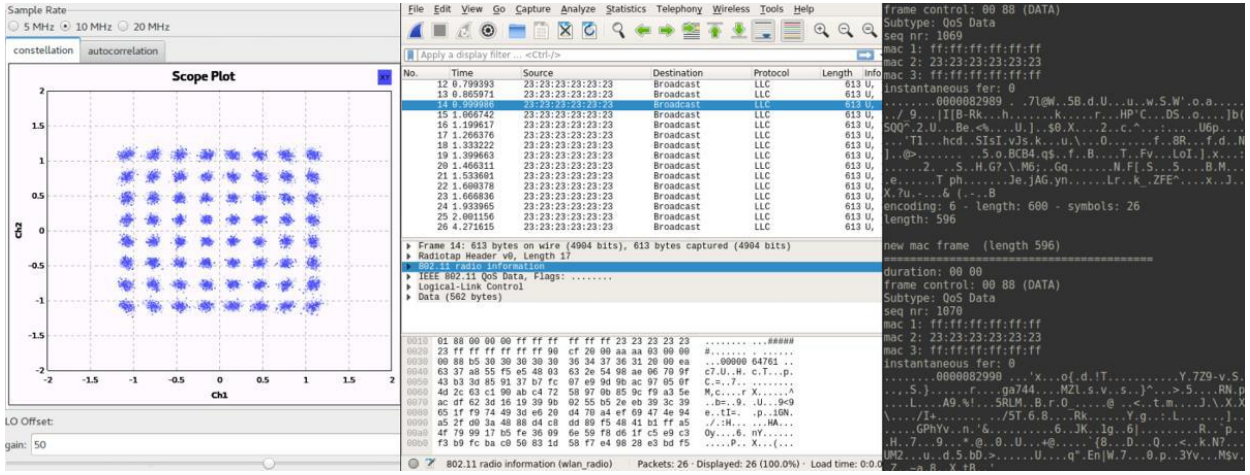
- Vehicle-to-everything
- For autonomous driving → safety, efficiency, and comfort
- C-ITS (Cooperative Intelligent Transport Systems)→ standardize Connected Automated Driving (CAD)
- Type of communications→
  - V2I
  - V2N
  - V2V
  - V2P
  - V2D
- 802.11p → first deployed

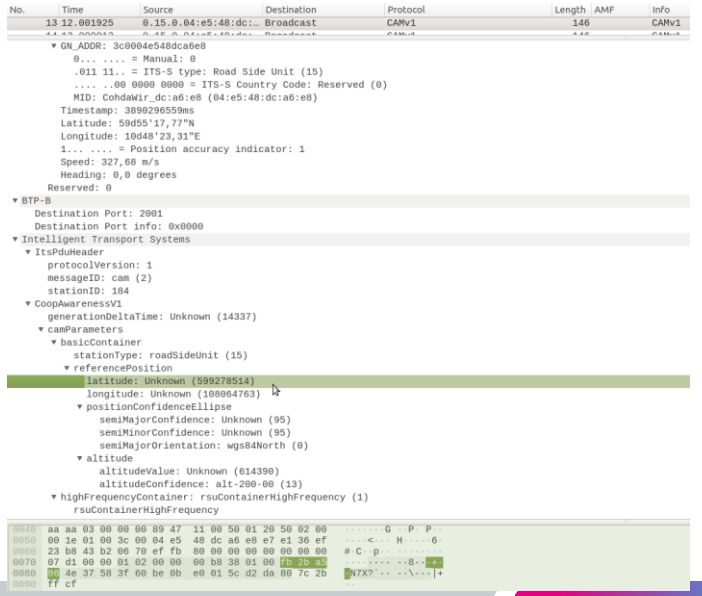


Source: An Overview of 3GPP Cellular Vehicle-to-Everything Standards by Xuyu Wang, Shiwen Mao, Michelle X. Gong

Capturing 802.11p data

- Based on Wi-Fi
- DSRC in US
- ITS-G5 in EU
- Capturing CAMv1 messages and more:
  - Using a dedicated dongle with a modified kernel[1]
  - Using and adapting Openwifi projects [2], or bladerf-wiphy[3]
  - Or still using at least a USRP B with WIME (allows also TX!):





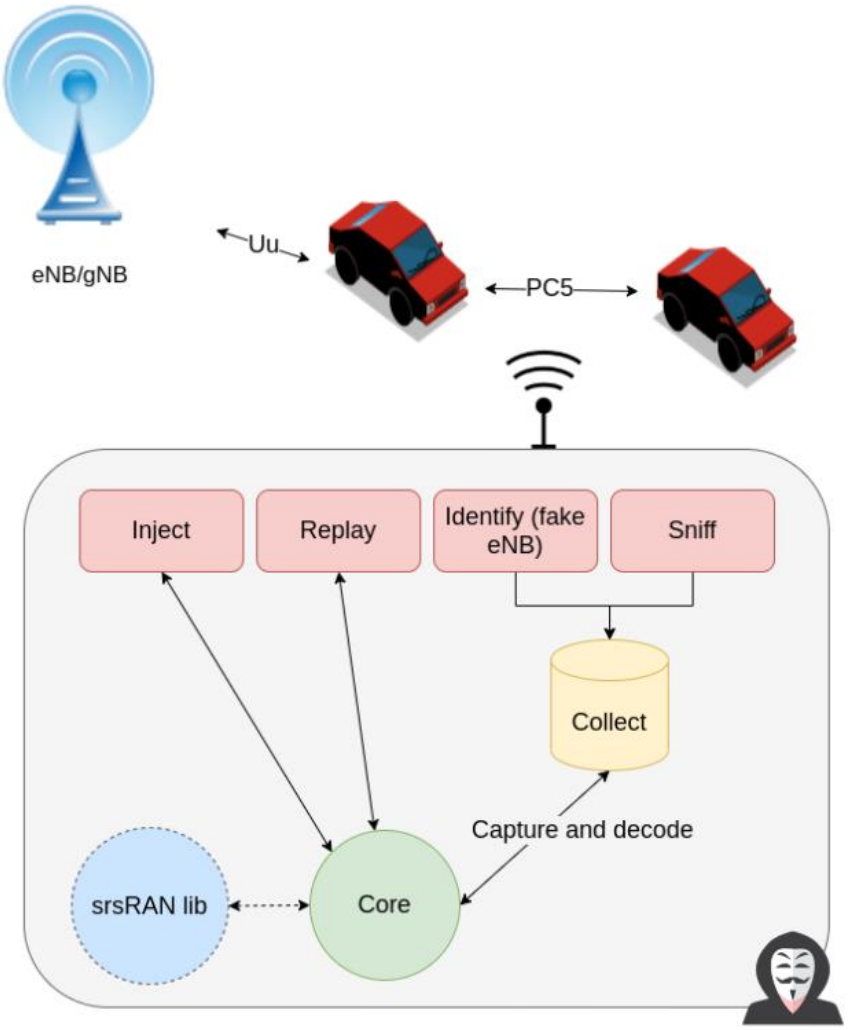
[1]https://harrisonsand.com/posts/802-11p-v2x-hunting/
[2]https://github.com/open-sdr/openwifi
[3]https://www.nuand.com/bladerf-wiphy/

C- V2X

- Cellular V2X → LTE-V2X for the moment
- 2 modes of communications: Direct short-range & Network
- Powerful alternative to 802.11p (but 802.11bd is on its way!)
- Introduction of ProSe (Proximity Service)→Side Link→PC5 interface
- Defined by 3GPP
  - LTE: Rel. 12 & Rel. 13 → D2D and eD2D→ Hazard warning
  - LTE Basic V2X by Rel. 14→ safety use case
  - 3GPP Release 15 → enhanced V2X→ Enhanced Navigation & Infotainment
  - and 3GPP Release 16 includes work on 5G-NR→Cooperative auto. driving
- Current problem to solve → privacy protection and usurpation → use of PKI→ handled by ETSI only not 3GPP

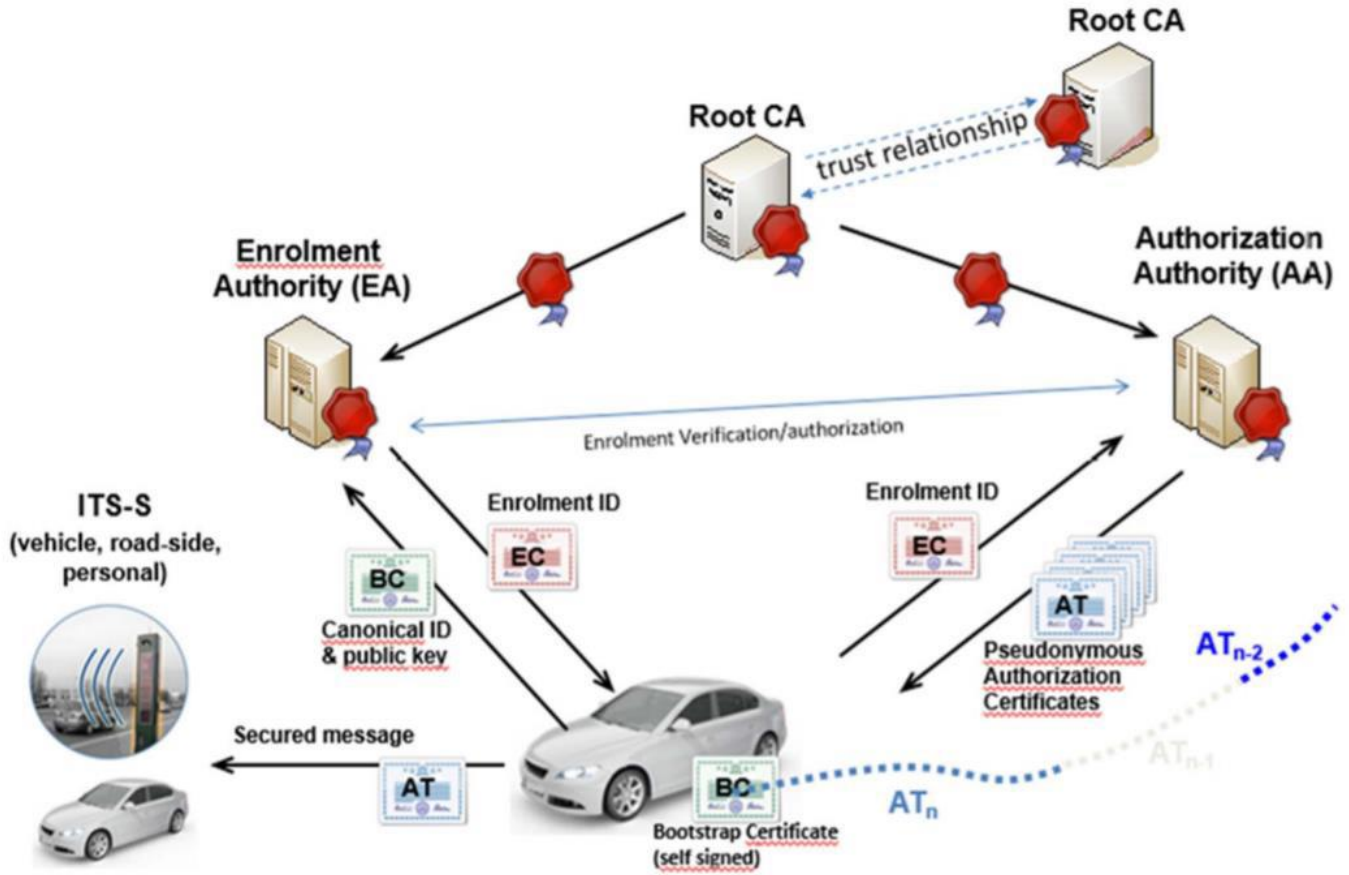
Our tools in LTE-V2X

- Based on srsRAN
- Focuses on PC5 mode 4
- Features:
  - Detection of capable V2X devices
  - Intercept and inspect SL messages
  - Injection of messages in current dev.



The current state of this research: still looking for real producs to test...

V2V/V2I PKI: What is the real state?



Source: ETSI TR 103 415 V1.1.1 (2018-04)

Conclusion

To conclude

- A lot of angle we couldn't cover --> different models = different techs
- Vehicles embed more and more technologies
- Some of these technologies are using RF to communicates → fewer cables
- RF is getting more accessible to attackers
- But without proper security mechanisms:
  - Inject message to trigger bugs or fake alerts
  - Track users
  - Inject frames on CAN → needs to bypass associated gateways

penthertz

Thank You

Please contact us:

contact@penthertz.com

+33 1 73 13 82 77

penthertz.com

Watch us on YouTube