## A Hackers View to ISO/SAE 21434

Or, how not to get pwned while driving 250 km/h on the Autobahn

## \$whoami

- Hi, I'm Martin!
- @Fr333k
- By day: security engineer in the automotive industry
- By night: digital forensics & applied privacy
- In between: old Land Rover



### \$whoami



#### Disclaimer

- This is me, not my employer!
- I'm not actively involved in the standarization process
- My view is biased, your mileage may vary!

Ignory my sarcasm:

• This gunna be good!





© Andy Greenberg, wired.com  $^5$ 

• 2

62 10



-



Camera 01

#### New Vehicle Security Research by KeenLab: Experimental Security Assessment of BMW Cars

by Tencent Keen Security Lab

#### **Vulnerability Findings**

After conducting the intensive security analysis of multiple BMW cars' electronic control units, Keen Security Lab has found 14 vulnerabilities with local and remote access vectors in BMW connected cars. And 7 of these vulnerabilities were assigned CVE (Common Vulnerabilities and Exposures) numbers.

All the following vulnerabilities and CVEs have been confirmed by BMW after we submitted the full report and collaborated with them on technical details:

No.	Vulnerability Description	Access	Affected Components	Reference
1	All the detail information has been reserved due to security concerns.	Local (USB)	HU_NBT	CVE-2018-9322
2		Local (USB/OBD)	HU_NBT	
3		Remote	HU_NBT	Logic Issue
4		Remote	HU_NBT	Reserved
5		Local (USB)	HU_NBT	CVE-2018-9320
6		Local (USB)	HU_NBT	CVE-2018-9312
7		Remote (Bluetooth)	HU_NBT	CVE-2018-9313
8		Physical	HU_NBT	CVE-2018-9314
9		Physical	TCB	Reserved
10		Remote	TCB	Logic Issue
11		Remote	TCB	CVE-2018-9311
12		Remote	TCB	CVE-2018-9318
13		Indirect Physical	BDC/ZGW	Logic Issue
14		Indirect Physical	BDC/ZGW	Logic Issue



Vehicles are complex!

- 100+ micro controllers
- Strongly coupled, closely connected
- Real-time requirements
- Security not a priority so far!



Software updates?

- Weakest part are often customers
- Common fear is rise in insurance costs



# Background

#### Common Interfaces

#### CAN bus:

- Differential power signal
- Very robust
- Allows arbitration
- Small payload: 64bit
- Bandwidth <= 1 Mbit/s
- No security!



#### Attack surface

Attack surface, traditionally:

- Physical, aka chip tuning
- Wireless protocols i.e., key fobs, tire pressure
- Infotainment system
- OBD-II port
- (Backends)

#### Common Interfaces

Wired/wireless:

- CAN-FD, Flexray, MOST, LIN
- BroadR-Reach
- Automotive Ethernet
- 433/868 Mhz
- NFC, Wifi, Bluetooth, ...

No security, or optional!

#### Common Interfaces

GSM/LTE:

- Mandatory, for eCall
- Not only once, but multiple times

V2X in the future:

- C-V2X, based on LTE
- IST-5G, based on 802.11p

#### Best security measure so far:



#### Doomed?



## ISO/SAE 21434

Current status:

- Committee draft, summer baseline
- About 120 pages
- International collaboration: DIN, VDA, ISO, SAE, ...
- Publication sometime 2020

Some fun facts:

• V-model development process is still a thing!

Wordly occurences:

- "cybersecurity": 822
- "cyber security": 1
- "risk": 180
- "agile": 1
- "Lorem ipsum": none

What is it all about:

- Risk management!
- Development process & beyond!
- Vulnerability management



#### Scope:

- Automotive componentes
- E/E (electrical and electronic) architecture

Related documents:

- ISO 26262: Functional safety
- SAE J3061, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems



Also relevant:

- ISO 31000: Risk management
- 2016: NHTSA Cybersecurity Best Practices for Modern Vehicles
- 2018: UNECE World Forum for Harmonization of Vehicle Regulations (WP.29)
- 2018: California IoT Security Law

Whats inside:

- Assets
- Vulnerabilities
- Threats
- Risk = Impact \* Feasibility
- Security goals: Confidentiality, Integrity, Availability

Standard risk management!

Example:

- Asset = software on specific ECU, say engine control unit
- Goal = integrity
- Impact = severe
- Feasibility = high

Outcome:

• Better do something about it!

Example:

- Asset = Infotainment
- Goal = availability
- Impact = low
- Feasibility = high

Outcome:

• Could be ok, but is probably not

Next step: security concept

- Define security measures & mitigations
- Can be requirements, or recommendations

Can be things like:

- MISRA C
- Fuzzing
- Proper training of developers

Going forward:

- Develop
- Verify
- Validate e.g., pentest



Post-development:

• Residual risks should be clear

Production!

- Some aspects can be surprisingly hard
- Like key management: Individual keys, certificates, signatures, ...
- Usually not a one-shot

Operations & maintenance:

- Monitoring
- Triage
- Incident response

But for how long?

- End of support?
- End of production?
- End of expected lifetime?



What it boils down to:

- Updates!
- What about supply chain?

FIN:

Decommission



## So, whats my take?

On the standarization process



My 2 cents:

- Closed standardization process?
- NDAs? Really???
- I thought we are over this!



To quote Kenny Paterson:

 "You're a bit late to the party. We're metaphorically speaking at the stage of emptying the ash trays and hunting for the not quite empty beer cans."

I would love to see more openness here!



Devil is in the details:

- "Sure we use secure boot" (but we store the key in unprotected memory)
- "Sure we use TLS" (but we don't pin on CA, or certificate)
- "Sure we have individual keys" (but no TRNG)

Don't wanna jinx it, but:

- This could be good!
- Move in the right direction

Possible issues:

- Vehicles as a whole are sooooooo complex
- Details, details, details ...



## Questions?

Contact me: @Fr333k

