the hack



the crash

& two smoking barrels

and all the times I [almost] killed an engineer.

thomas sermpinis
@cr0wtom

They accused me of exposing vulnerabilities to “hostile nations”

$whoami

- Thomas Sermpinis - **@cr0wtom**
  - Technical Director by Day @ Auxilium Pentest Labs
  - Security Researcher by Night @ Cr0w's Place
  - *Hack Everything, Everywhere, All at Once (and Legally)*
    - Cinema aficionado. If you can help me with starting to shoot my first movie, **find me after my talk**
  - Major infosec complainer
- For more: **cr0wsplace.com**





What this talk IS about

- Analyse the state of cyber security in the automotive industry (in 2024)
- ~~Present unique (and hopefully intresting) stories, result of more than 200 pentest and research projects in the industry~~
  - *Talk about all the stupid things the automotive industry is still doing*
- Educate the new, the old and the bold
- Lurk more hackers to automotive security
- Raise and highlight the significance of safety related hardware

What this talk is NOT about

- Extreme exploit chains from the underworld
- Killing people for the shake of science
- Blowing up garages with EVs after a contractor failure
- Suffocating engineers after a hydrogen tank failure

Κεφάλαιο 0

The State


1 05
AM
22°
MUNICH | SUNNY


Pimp
My
Ride
22°
MUNICH
SUNNY

How Tech-Savvy Thieves Are Stealing Cars By Hacking Through Headlights

by **Nathan Ord** — Saturday, April 08, 2023, 02:37 PM EDT

2 Comments

BECOME A PATRON



Early last year, hackers were replaying remote keyless system codes to unlock and steal Honda or Acura vehicles. This year, criminals of TikTok have been showing people how to break into certain Hyundai and Kia models with some hotwiring. However, criminals are upping their thieving game as car companies come to the rescue with patches and security solutions for vehicles. With this forced advancement come car thefts through attacks on the car's central nervous system called the Controller Area Network (CAN) bus.

How Tech-Savvy Thieves Are Stealing Cars By Hacking Through Headlights

by **Nathan Ord** — Saturday, April 08, 2023, 02:37 PM EDT

How Thieves Are Stealing Hyundais and Kias With Just a USB Cable

This low-tech hack specifically targets the Korean cars that use a physical key.

BY ROB STUMPF | PUBLISHED AUG 2, 2022 3:28 PM EDT

NEWS



models with some hotwiring. However, criminals are upping their thieving game as car companies come to the rescue with patches and security solutions for vehicles. With this forced advancement come car thefts through attacks on the car's central nervous system called the Controller Area Network (CAN) bus.



How Tech-Savvy Thieves Are Stealing Cars By Hacking Through Headlights

Forget the Kia Boyz, a new exploit leaves Kias and Hyundais vulnerable

Story by James Ochoa • 1d • 3 min read

South Korean automaker Hyundai and its sister company Kia have been through the ringer. when it came to a notorious security vulnerability that was exposed through the power of social media.

The infamous "Kia Boyz" sprang a wave of auto thefts from 2021 to 2023, where Hyundai and Kia vehicles were targeted due to the lack of a vital security feature called an immobilizer, rendering them easy to drive without a key.

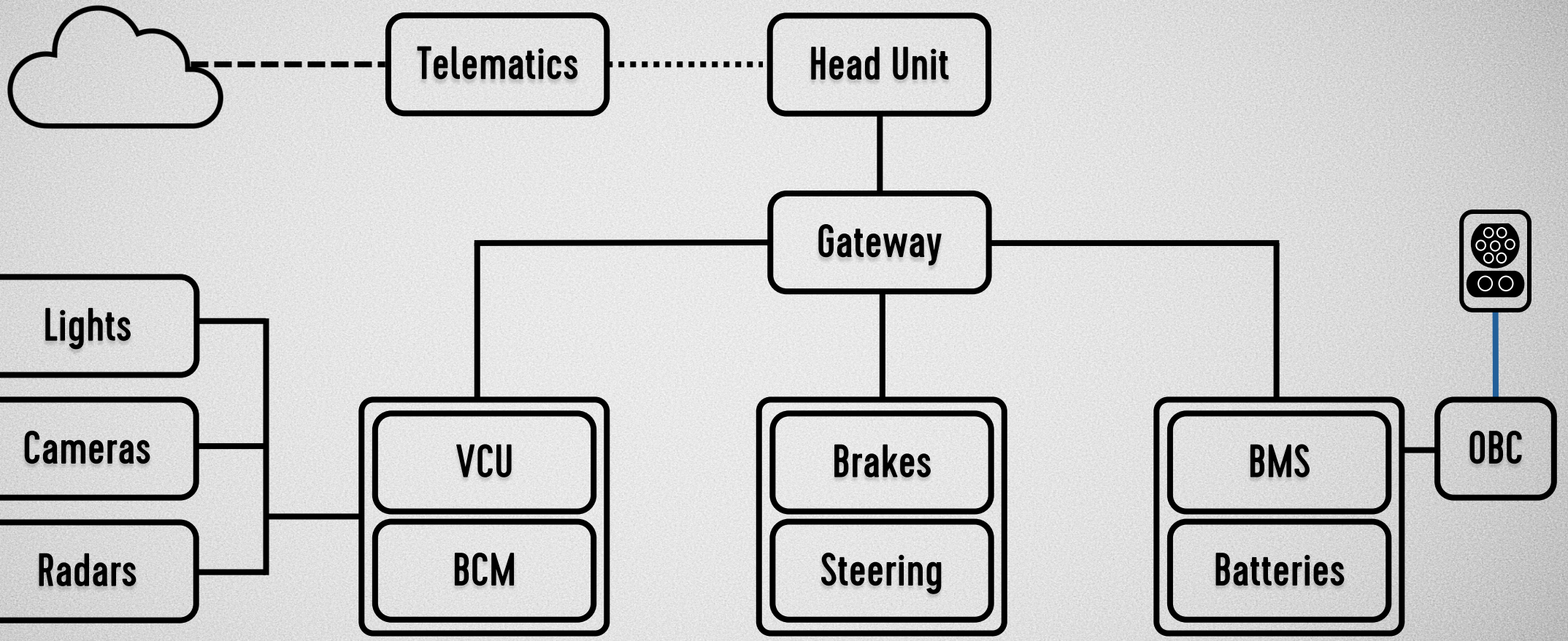
In "instructional" videos that have gone viral on social media platforms like TikTok, thieves demonstrated how easy it was to steal such vehicles, as it only required 35 seconds and rudimentary tools like a flat-head screwdriver and a USB charging cable for a smartphone.

While Kia and Hyundai have introduced fixes for vehicles targeted by the Kia Boyz, thieves have evolved and a new vulnerability is affecting vehicles that are crucial to the Korean automaking duo's future in the industry.

Related: CDK Global expects software outage to end soon; dealers lawyer up

rescue with patches and security solutions for vehicles. With this forced advancement come car thefts through attacks on the car's central nervous system called the Controller Area Network (CAN) bus.

Automotive Architecture



CAN – Constantly Annoying and iNsecure

- CAN - Physical and Data link layer communication in ECUs
- On "higher layers", data is exchanged without encryption
- Obscurity of proprietary messages acts as a layer of "protection"
- Protocols on top of CAN are also unencrypted
  - Interception, injection and replication are all applicable
- Protections like SecOC (Message Authentication) can be implemented

The State of Automotive Cybersecurity

- UN Regulation No. 155 - general requirements for Vehicle Cybersecurity
  - Provides a set of standards that must be met in order to ensure the safety of road vehicles
  - The regulation requires the operation of a certified cybersecurity management system (CSMS)
  - UN R155 is significant as it provides a set of standards that must be met in order to ensure the safety of road vehicles
  - Applicable to members of the United Nations Economic Commission for Europe (UNECE)
    - No equivalent in the US at a federal level
- **In summary:** Trying to shape the completely unregulated mess that exists right now
- **Biggest caveat?** Penetration testing is solely based on the Risk Assessment (TARA)

The State of Automotive Cybersecurity

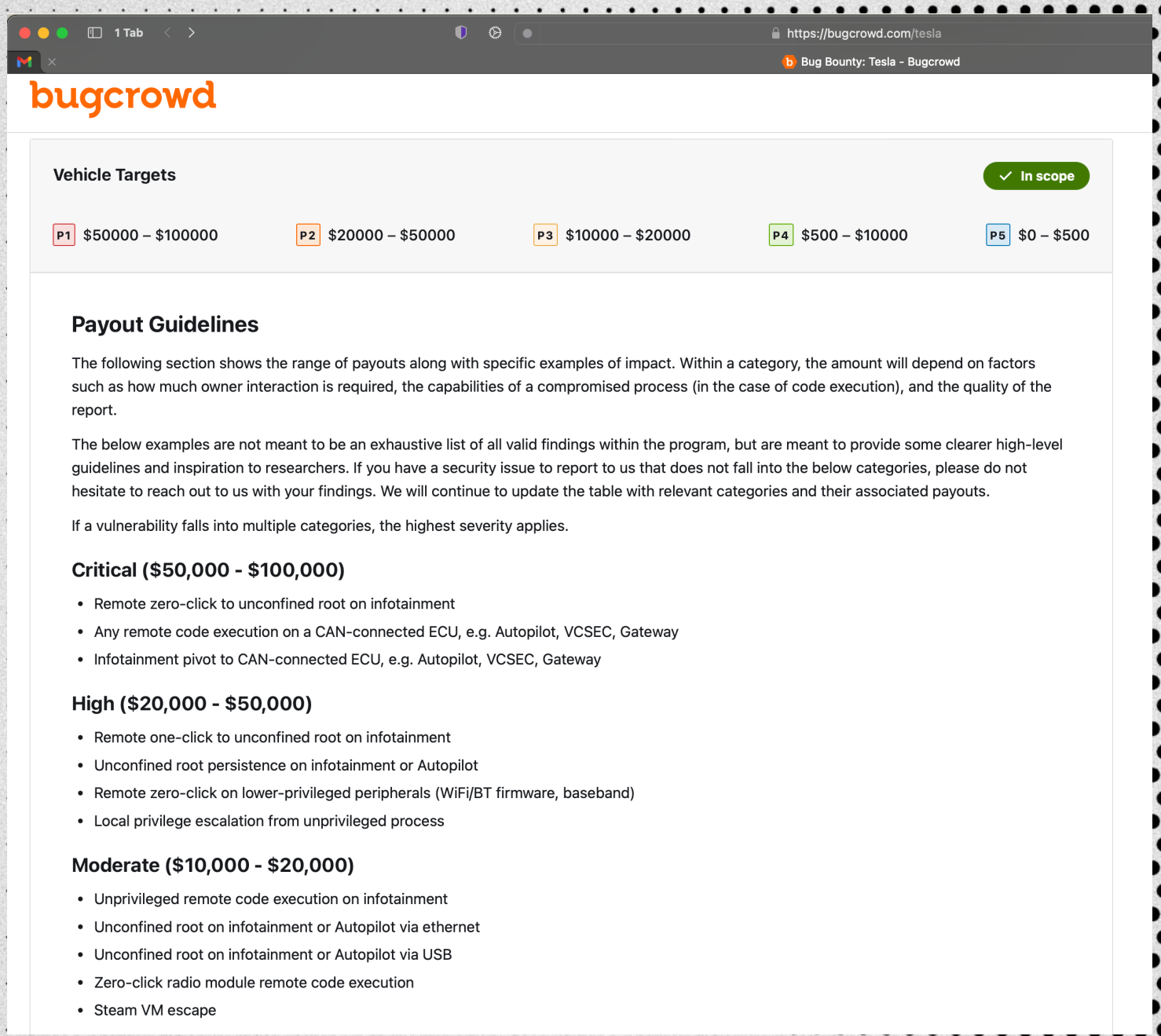
- Is there a light in the end of the tunnel?
- Automotive, hardware and embedded industries are not new by any means
- Wide adoption of personal vehicles and really short life-cycles, made development shorter and targets more accessible to the "average Joe"

Κεφάλαιο 1

The Hack

VDP - Vexingly Disorganized and Problematic!

- While VDPs are getting widespread, automotive is not following that trend
- Some large manufacturers have implemented VDPs, with no actual rewards for researchers
  - A big miss, considering the cost of the targets
- Company culture heavily affects how vulnerabilities are handled



Source: https://bugcrowd.com/tesla

In the last 2 years I discovered 25 safety related findings

2 were “accepted”

UDS - Unreliable, Dysfunctional, and Slow!

- UDS - Unified Diagnostic Services
  - Application layer protocol for diagnostics in Electronic Control Units
- Allows diagnostic functions such as reading and erasing fault codes, programming, testing, and monitoring of ECUs
- Consists of several “services" which can be used to perform specific actions
- A common one, ECUReset (0x11)
  - Results in different types of “resets” in an ECU, including Hard, Soft, and others
  - Usually, Hard ECUReset is equivalent to a complete powercycle




MENU
8:19
Call History
Phone
Audio
Connection
Info
Settings
8:19PM
93°F
Warning
Malfunction
Dismiss
173 miles
KALI


8:17PM
93°F
RPMx1000
Warning
Malfunction
Dismiss
26040 miles
P
173 miles
MPH
km/h
KALI

Was it that easy?

- While something was triggered, thorough analysis must be done to understand the underlying architecture
  - Remember architecture? :P

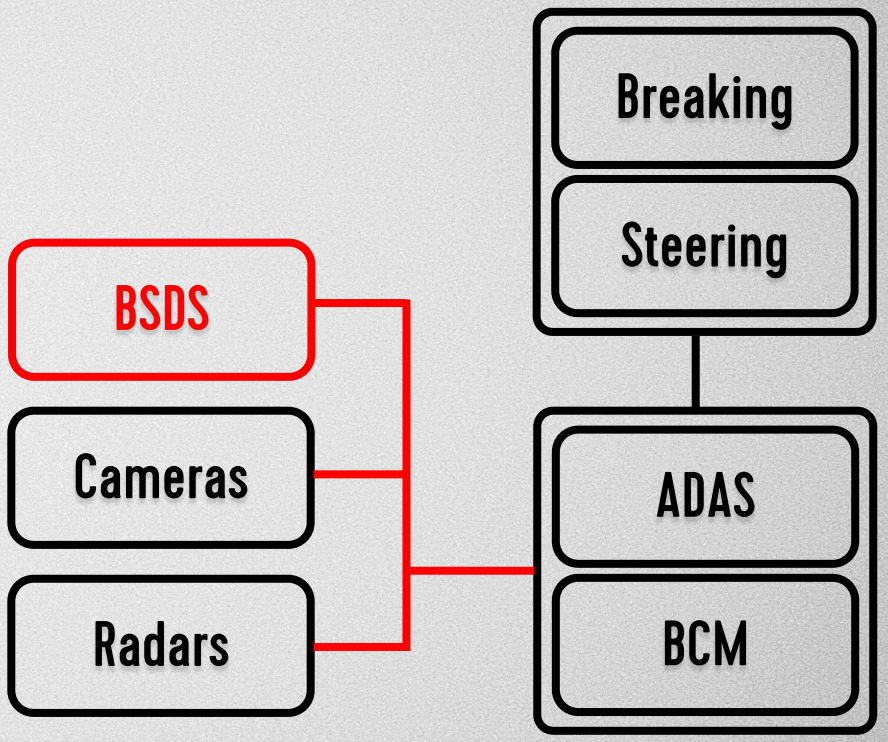
- Is this error “real”?
- Is it expected or not?
- Does it have a security impact?
- Does it have a safety impact?


1:20
Call History
Settings
TUNE·SCROLL
PUSH SOUND
CAMERA
FIXED
VMware Fusion File Edit View Virtual Machine Window Help
Kali Linux
File Actions Edit View Help
KALI
Demo 1

Remember Architecture?

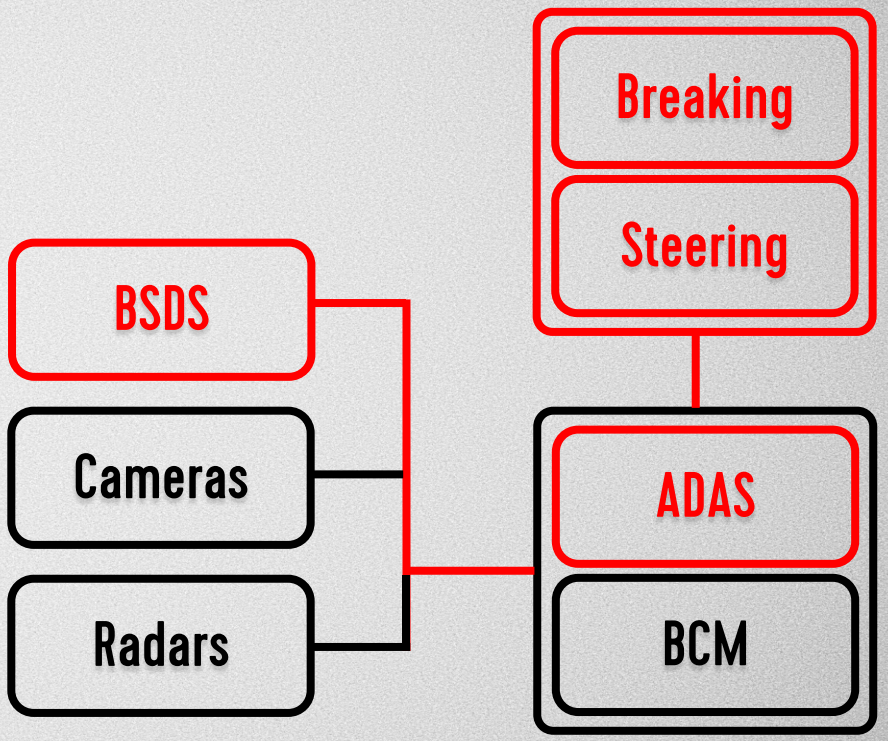
The Mighty Blind Spot Detection Sensor

- A lot of “educated guesses” can be made at this point
- The facts about BSDS:
  - Can be powercycled by the ECUReset service on UDS
  - Connected to the ADAS system
  - No preconditions implemented for access to the ECUReset service (e.g. vehicle speed)
  - No filtering implemented for access to the ECU by the OBD port



The Mighty Blind Spot Detection Sensor

- With no preconditions implemented, the sensor can be triggered mid-drive
- Potential security and safety implications can be assumed:
  - Steering interruption during a lane change
  - Braking engagement during injection of a safety related incident based on BSDS
- These assumptions have to get STRICTLY tested in a SAFE environment



Who has this environment?

VDP – A Common Timeline

1. Discover an applicable communication channel (Company website, VDP website, etc.)
2. Prepare a well structured vulnerability report, with reproduction steps and risk analysis
3. Submit the report to the contact point
4. Manufacturer acknowledges receipt and provide initial response
5. Vulnerability triage
6. Manufacturer moves to the appropriate steps for mitigation of the vulnerability
7. After the embargo period agreed by the two parties, vulnerability details are getting published, CVEs are getting assigned if applicable

VDP – My Timeline

1. Discover an applicable communication channel (Company website, VDP website, etc.)
2. ~~Prepare a well structured vulnerability report, with reproduction steps and risk analysis~~
3. ~~Submit the report to the contact point~~
4. ~~Manufacturer acknowledges receipt and provide initial response~~
5. ~~Vulnerability triage~~
6. ~~Manufacturer moves to the appropriate steps for mitigation of the vulnerability~~
7. ~~After the embargo period agreed by the two parties, vulnerability details are getting published, CVEs are getting assigned if applicable~~

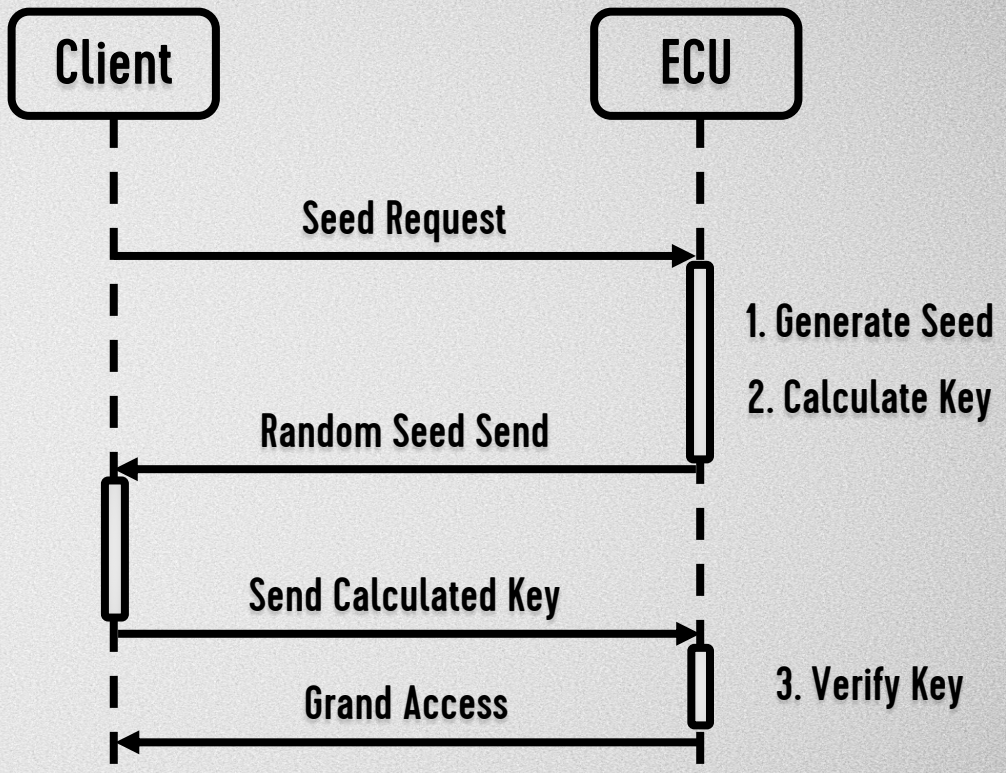
I didn’t give up.

I didn’t give up.

Kept searching for more vulnerabilities :P

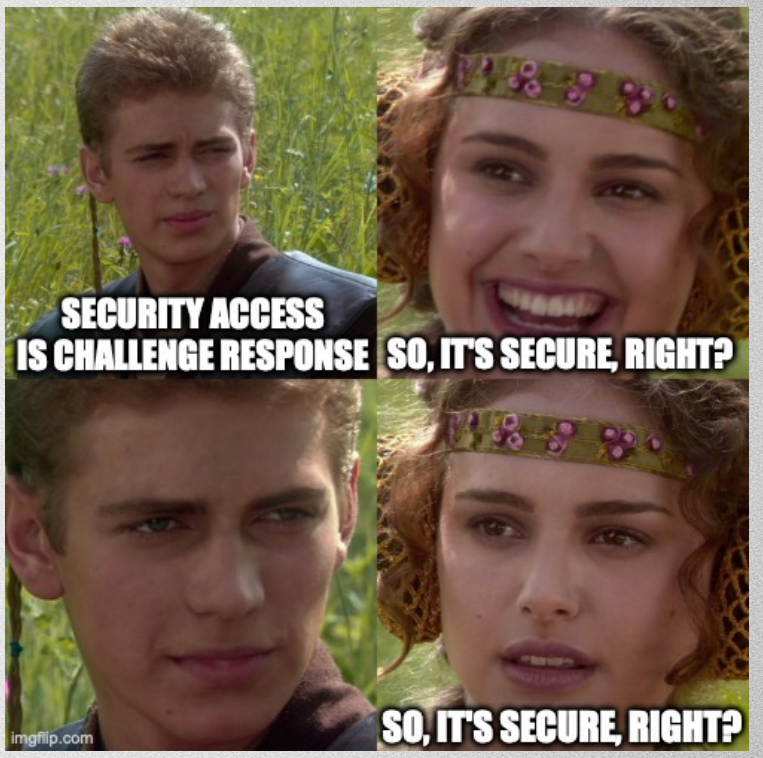
Access Control in UDS

- UDS service “Security Access” – 0x27
  - Most common authentication scheme in UDS
  - “Just” a simple challenge-response
- Implementation depends on cybersecurity requirements, set by the OEM
- Losely developed requirements can result in:
  - Sloppy security access implementations
  - Weak sources of randomness
  - Backdoors



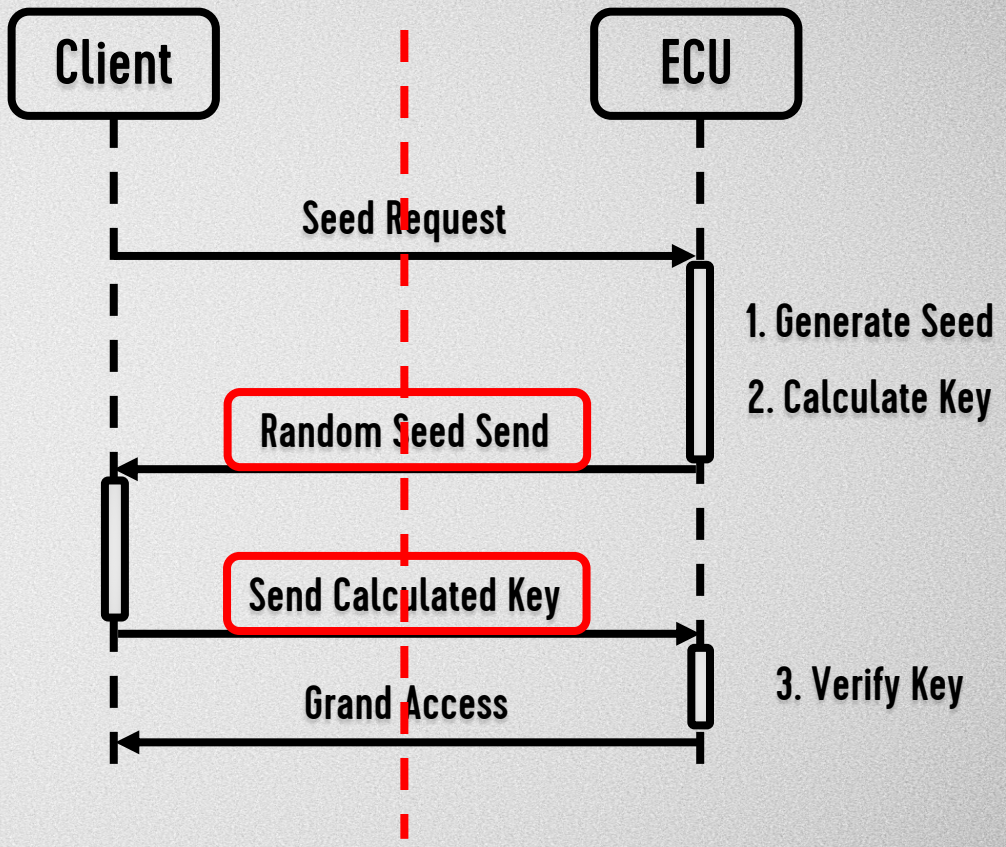
Access Control in UDS

- Security access can be used in several instances:
  - Software update from the main Head Unit
  - Reprogramming in service departments
- Process is not encrypted
- *Why would it be? It's a challenge response, right?*



The Ugly Truth

- If UDS is on top of CAN, communication is not encrypted
- Message authentication can be implemented, with its own issues
  - *99% of the ECUs we've tested did not implement message authentication for application layer protocols*
- Effectively allowing us to intercept any communication between the client and the ECU



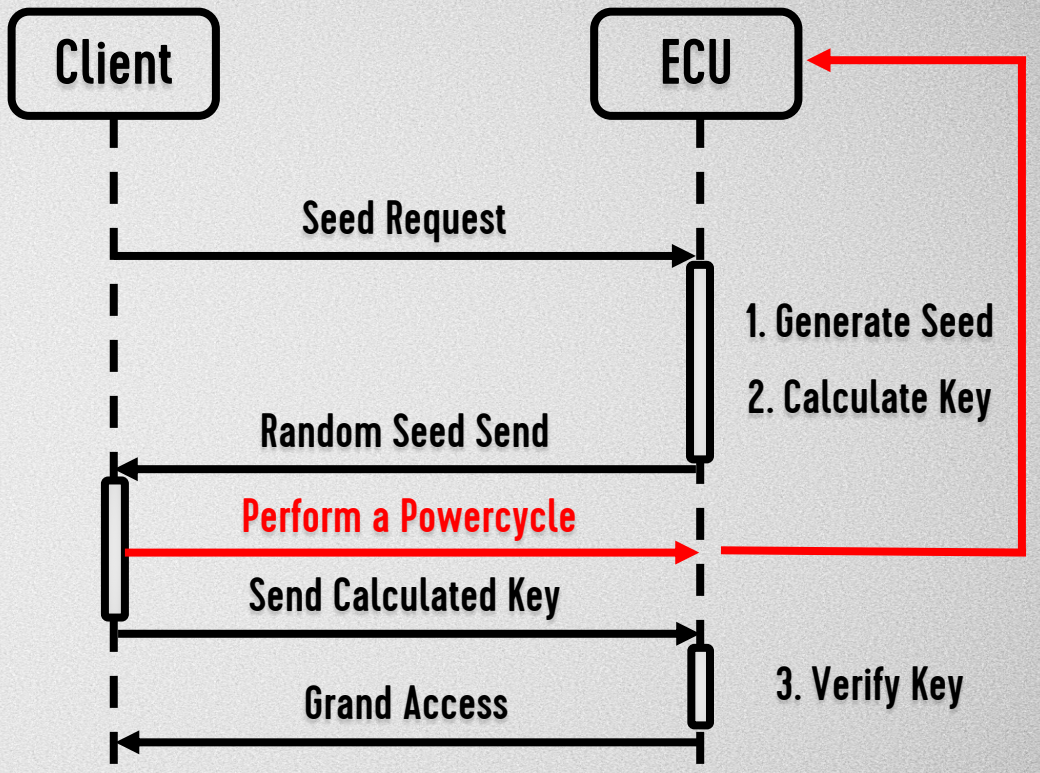
The Ugly Truth

- But even with access to the underlying communication, we still require:
  - The randomly generated seed
  - The secret algorithm for calculation of the calculated key
  - The secret key taking part in the process
- Without those, we cannot obtain elevated access to the target

Or can we?

The Path to Game Over

- Remember ECUReset?
- **Direct outcome:** Full ECU powercycle resulting in BSDS trigger
- **In-direct outcome:** Reset of the system clock



cr0wtom@kali-m1: ~/Tools/caringcaribou/tool

File Actions Edit View Help

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz seed_randomness_fuzzer -d 1.102 10032701 0x7d4 0x7d5
```

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ candump can0,7D5:7D4
```

Demo 2

[1] 0:zsh* "kali-m1" 13:44 04-Jun-22

The Ugly Truth

- *But even with access to the underlying communication, ~~we still require:~~*
  - The randomly generated seed
  - The secret algorithm for calculation of the calculated key
  - The secret key taking part in the process

- With that in mind, only one seed and calculated key pair is needed to get elevated access to the unit
  - Despite the size and complexity of the seed, success rate of requesting the desired seed is almost 100%

2 years later...

VDP – The Updated Timeline

1. Found someone that answered to my prayers

> If you have a report, we'd be interested in seeing what you've found.

2. ~~Prepare a well structured vulnerability report, with reproduction steps and risk analysis~~
3. ~~Submit the report to the contact point~~
4. ~~Manufacturer acknowledges receipt and provide initial response~~
5. ~~Vulnerability triage~~
6. ~~Manufacturer moves to the appropriate steps for mitigation of the vulnerability~~
7. ~~After the embargo period agreed by the two parties, vulnerability details are getting published, CVEs are getting assigned if applicable~~

VDP – The Updated Timeline

1. Found someone that answered to my prayers
2. Prepared a well structured vulnerability report, with reproduction steps and risk analysis
3. ~~Submit the report to the contact point~~
4. ~~Manufacturer acknowledges receipt and provide initial response~~
5. ~~Vulnerability triage~~
6. ~~Manufacturer moves to the appropriate steps for mitigation of the vulnerability~~
7. ~~After the embargo period agreed by the two parties, vulnerability details are getting published, CVEs are getting assigned if applicable~~

VDP – The Updated Timeline

1. Found someone that answered to my prayers
2. Prepared a well structured vulnerability report, with reproduction steps and risk analysis
3. Submitted the report to the contact point
4. ~~Manufacturer acknowledges receipt and provide initial response~~
5. ~~Vulnerability triage~~
6. ~~Manufacturer moves to the appropriate steps for mitigation of the vulnerability~~
7. ~~After the embargo period agreed by the two parties, vulnerability details are getting published, CVEs are getting assigned if applicable~~

VDP – The updated timeline

1. Found someone that answered to my prayers
2. Prepared a well structured vulnerability report, with reproduction steps and risk analysis
3. Submitted the report to the contact point
4. ~~Manufacturer~~ Contact Point acknowledges receipt and existence of vulnerabilities in more than one models, and several model years of the OEM

Privately, we've confirmed vulnerabilities #2 and #3.

5. ~~Vulnerability triage~~
6. ~~Manufacturer moves to the appropriate steps for mitigation of the vulnerability~~
7. ~~After the embargo period agreed by the two parties, vulnerability details are getting published, CVEs are getting assigned if applicable~~

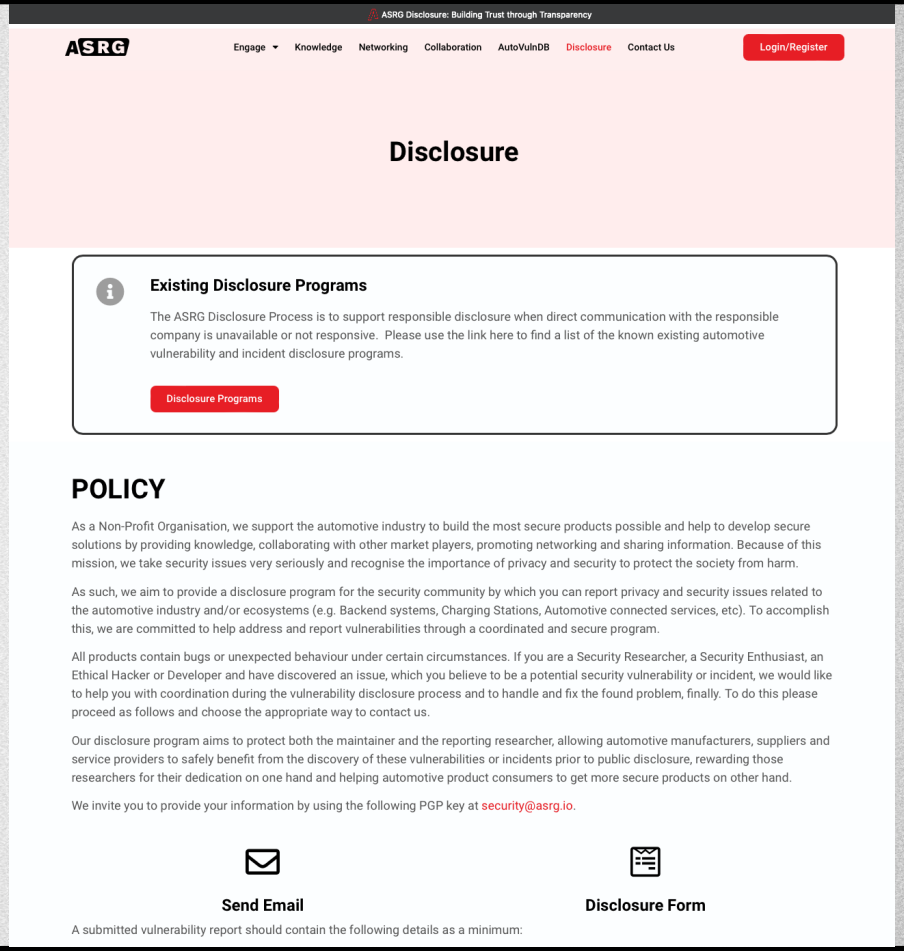
Now what?

- Despite the help from the contact point, OEM decided that cost of fixing the issue exceeds the expectations for these models
- **The argument:** *The new E/E architecture will fix those issues in future models – MY25/MY26*
  - Meaning that the actual architecture which is **currently sold**, is still vulnerable, and will be
  - **OEM is not willing to fix or disclose the findings**
- Due to the safety aspect of the vulnerabilities, I decided to **NOT** disclose them publicly and search other avenues of escalation



Automotive Security Research Group

- ASRG – Automotive Security Research Group
- Out of many:
  - Supports Responsible Disclosure in Automotive
  - Handles discovery and communication with responsible parties
    - If not directly available
  - Officially recognized CVE Central Naming Authority - CNA



Source: https://asrg.io/disclosure/

VDP – A Series of Unfortunate Events

1. Found someone that answered to my prayers
2. Prepared a well structured vulnerability report, with reproduction steps and risk analysis
3. Submitted the report to the contact point
4. ~~Manufacturer~~ Contact Point acknowledges receipt and existence of vulnerabilities in more than one models, and several model years of the OEM
5. Vulnerability triage
6. ~~Manufacturer moves to the appropriate steps for mitigation of the vulnerability~~
7. ~~After the embargo period agreed by the two parties, vulnerability details are getting published, CVEs are getting assigned if applicable~~

VDP – A Series of Unfortunate Events

1. Found someone that answered to my prayers
2. Prepared a well structured vulnerability report, with reproduction steps and risk analysis
3. Submitted the report to the contact point
4. ~~Manufacturer~~ Contact Point acknowledged receipt and existence of vulnerabilities in more than one models, and several model years of the OEM
5. Vulnerability triage
6. ~~Manufacturer moves to the appropriate steps for mitigation of the vulnerability~~
7. ~~After the embargo period agreed by the two parties, vulnerability details are getting published CVEs are getting assigned if applicable~~

OEM STOPS RESPONDING

Is this the end?

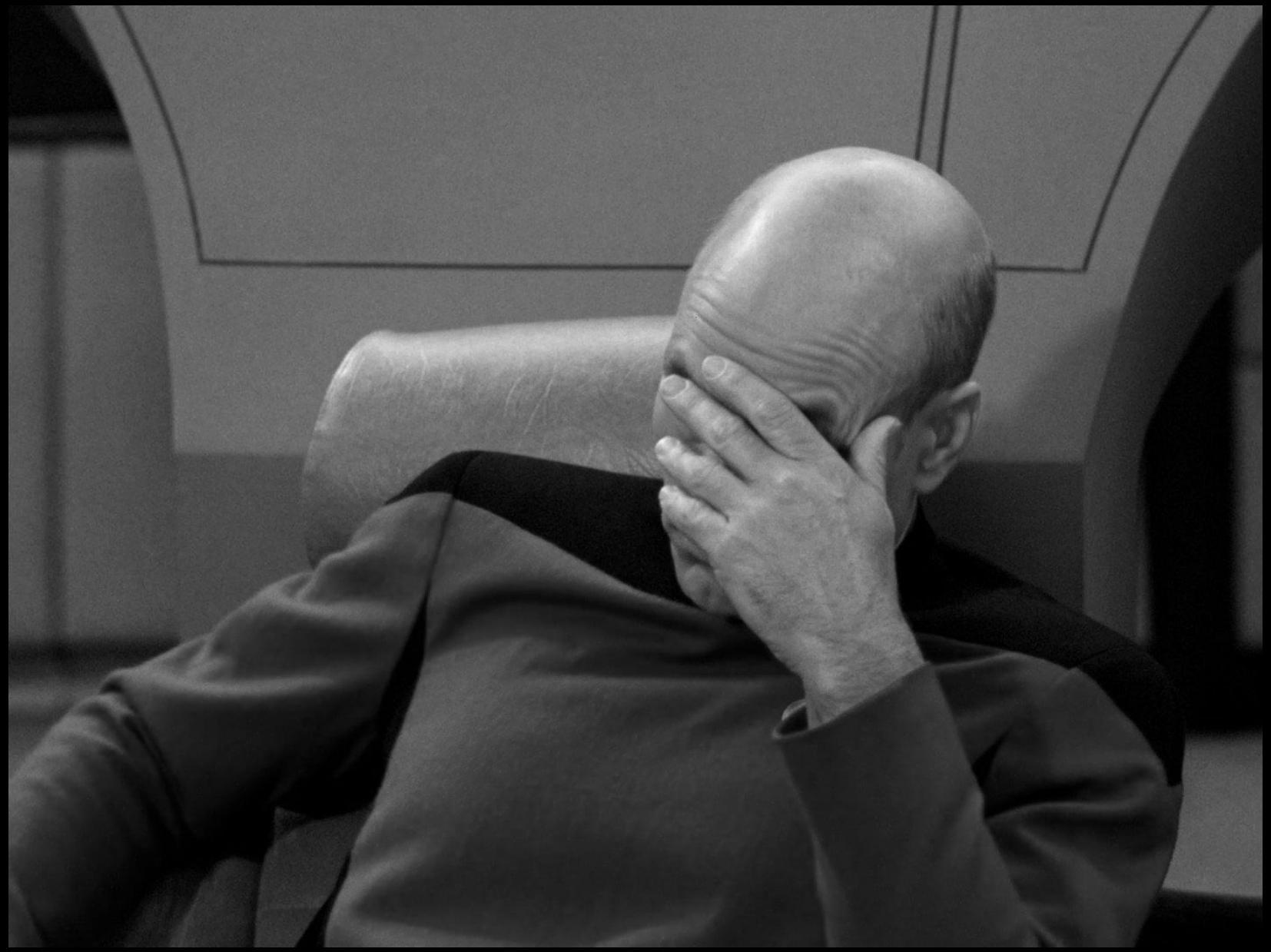
- Despite our continuous effort to resolve the issue in the best way possible, **OEM stopped responding to me and the CNA**
- The reason - "***they cannot collaborate with hostile nations***"
- Clearly, I am not from one, so wtf is happening here?



My Train Wreck Timeline

- July 2022 – Vulnerabilities Discovered
- January 2024– Contact point discovered
- 30th January 2024 – Report shared with OEM
- 5th February 2024 – Report shared with CNA after agreement with OEM
- 2nd March 2024 – Vulnerabilities get confirmed by OEM
  - 12th March – OEM decides to accept risk / not fix the issues **unofficially**
  - From here on, no further responses by the OEM to me or to the CNA
- 1st April 2024 – Embargo period ends
  - No responses by the OEM to our requests
- June 2024 - Several people outside of the **PRIVATE** disclosure, mention that I shared findings with ***hostile nations***
- DEF CON day – **Public Release of vulnerabilities**

What really happened?



My Train Wreck Timeline

- **While I thought that it's over and that OEM will never give an official statement, this happened:**

> Thank you for providing information and feedback regarding your analysis of a potential vehicle cybersecurity concern. We value the safety of our customers and our vehicles and strive to improve customer satisfaction by continuously improving vehicle safety. We have considered your comments carefully as we continue to strive to that goal. After thorough study, we do not consider any specific action is necessary at this time. As stated in the owner's manual, customers are advised not to drive with a dongle inserted into the OBD port.

“As stated in the owner's manual, customers are advised not to drive with a dongle inserted into the OBD port.”

- OEM that values the safety of their customers and their vehicles and strive to improve customer satisfaction by continuously improving vehicle safety.

The Aftermath

- 2 Vulnerabilities get published today:
  - CVE-2024-6348 - Predictable seed generation after ECU reset
  - CVE-2024-6347 - Unauthorized access to ECU functionality

- Instead of fixing an issue of a current MY vehicle, OEM decided to play politics
- Researchers are not the ones to blame here with falsified accusations

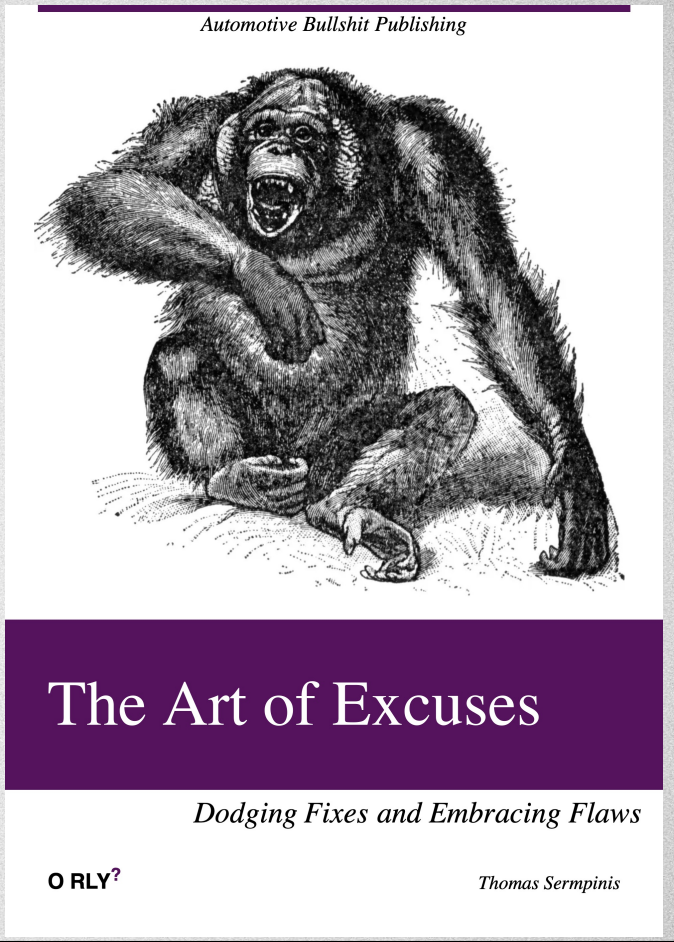
- While OEMs/suppliers try to hide vulnerabilities and minimize their security and safety impact, malicious users are out there using them in more sophisticated exploit chains

Κεφάλαιο 2

The Crash

Politics, I Love Politics

- Politics play a big role
- ~~Reasons~~ / Excuses:
  - Loosely developed requirements
  - Strict deadlines
  - Long product lifecycles
  - etc.

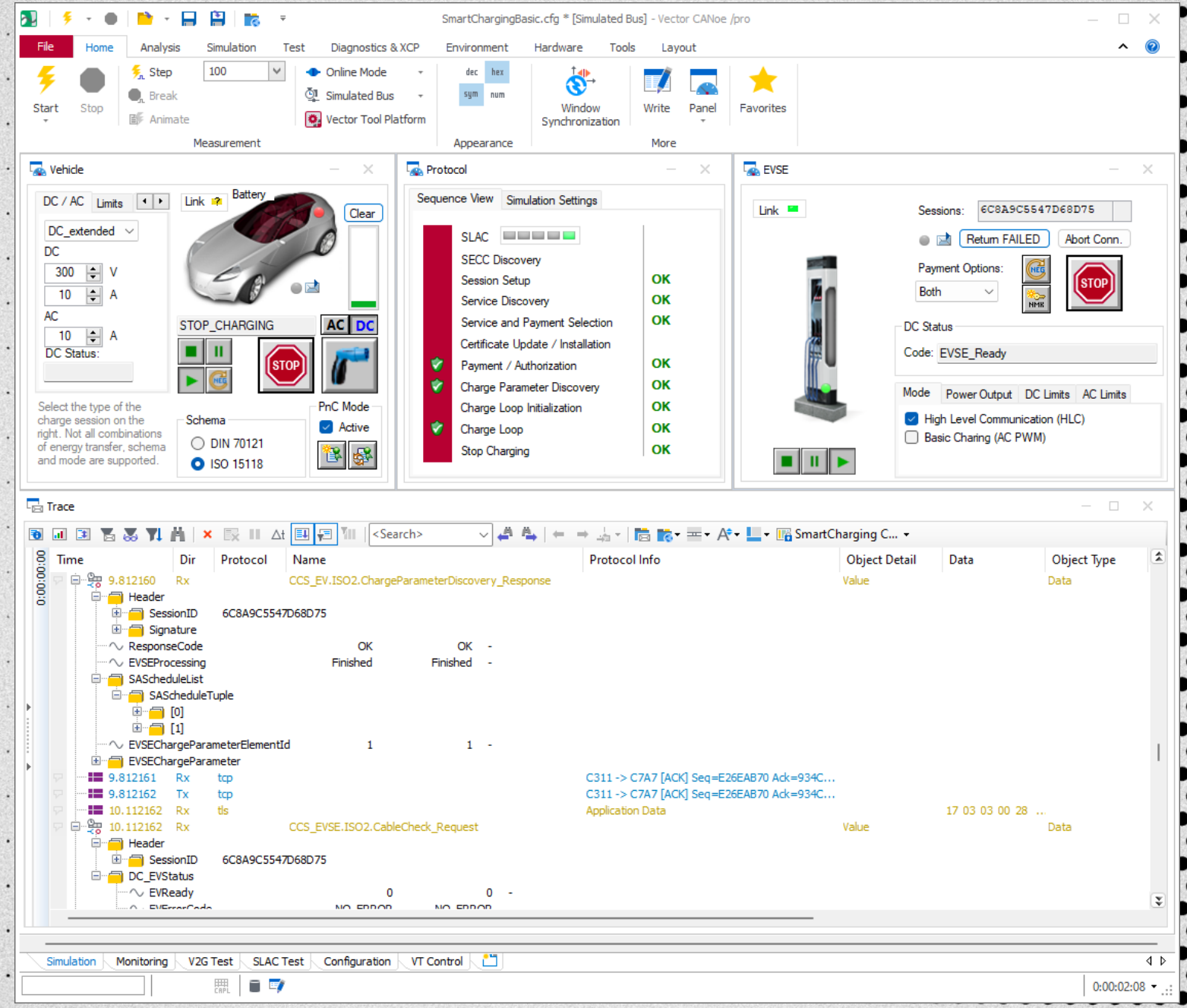


Politics – The Debate

- Culture is the main aspect resulting to politics in pentesting and research
  - Company culture
  - Local culture
    - *Countries of "low transparency" and "weak governance"*[1]
- Engineers follow standards that are "supposed" to be secure
- They never want to be wrong
  - Being wrong = I am bad at my job

1. Proper term is shady, but I wanted to avoid more legal troubles after DEFCON :P

But what happens when something does not go as planned?



Source: https://www.vector.com




OLDHAM

My Magical Exploit Chain

1. Spoof preconditions with non authenticated messagers over CAN



My Magical Exploit Chain

1. Spoof [illegible] with non-authenticated messagers over CAN

Unencrypted CAN



My Magical Exploit Chain

1. Spoof [illegible] with non-authenticated messagers over CAN
2. Perform a powercycle to the safety critical ECU

Unencrypted CAN



My Magical Exploit Chain

1. Spoof [illegible] with non-authenticated messagers over CAN
2. Perform a power cycle to the safety critical ECU

Unencrypted CAN

Unauthenticated ECUReset



My Magical Exploit Chain

1. Spoof [illegible] with non-authenticated messagers over CAN
2. Perform a power cycle to the safety critical ECU
3. Find a way to access the affected ECU from publicly accessible interfaces

Unencrypted CAN

Unauthenticated ECUReset



M

1.

2.

3.





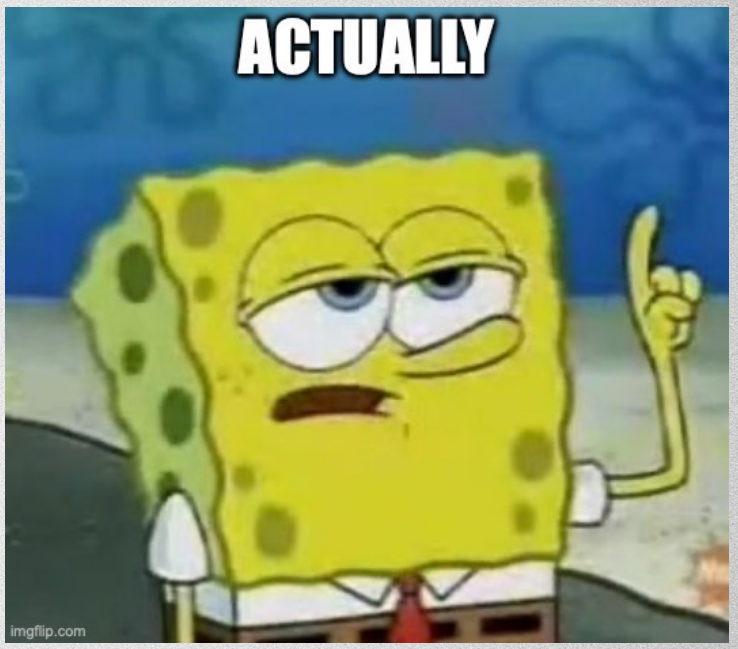
Source: Ken Tindell Linkedin

My Magical Exploit Chain

1. Spoof unconditional and non authenticated messagers over CAN
2. Perform a power cycle to the safety critical ECU
3. Find a way to access the affected ECU from publicly accessible interfaces
4. Listen to clients excuses on how this is not realistic

Unencrypted CAN

Unauthenticated ECUReset



My Magical Exploit Chain

1. Spoof preconditions with non authenticated messagers over CAN
2. Perform unauthenticated ECUReset on safety critical ECU
3. Find a way to access the affected ECU from publicly accessible interfaces
4. Listen to clients excuses on how this is not realistic
5. Put them inside the vehicle for some real life proof

Unencrypted CAN

Unauthenticated ECUReset



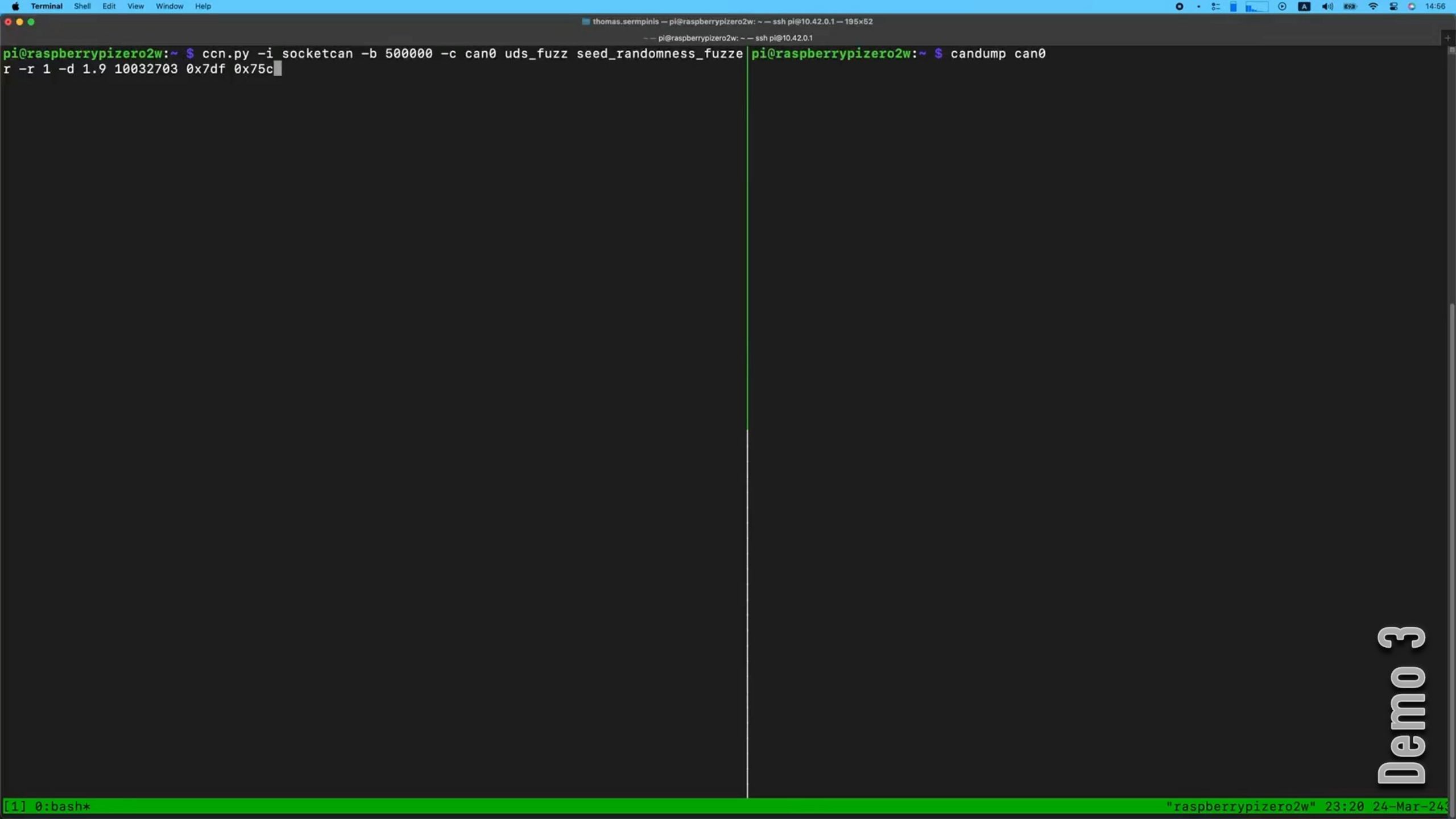
For safety reasons, they didn’t let me do it…

The next day...

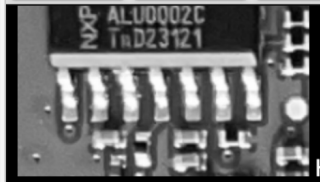




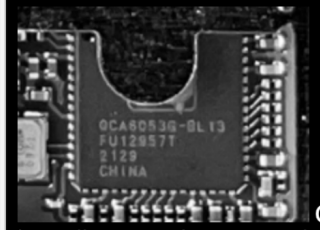

TA2013/1167
APPROVED
0682
ACTIA NORDIC
MADE IN TUNISIA
E5 10 R-04182


Terminal Shell Edit View Window Help
14:56
thomas.sermpinis — pi@raspberrypizero2w: ~ — ssh pi@10.42.0.1 — 195×52
~ — pi@raspberrypizero2w: ~ — ssh pi@10.42.0.1
pi@raspberrypizero2w:~ $ ccn.py -i socketcan -b 500000 -c can0 uds_fuzz seed_randomness_fuzzer -r 1 -d 1.9 10032703 0x7df 0x75c
pi@raspberrypizero2w:~ $ candump can0
Demo 3
[1] 0:bash*
"raspberrypizero2w" 23:20 24-Mar-24


dnt
QUALCOMM®
MDM6200
BN79
E241200
OK
JBC
OK
multicomp PRO



High-Speed CAN Transceiver with Standby and Sleep Mode
https://www.nxp.com/docs/en/data-sheet/TJA1043.pdf



Qualcomm® APQ8053 SoCs for IoT
https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/apq8053-socs-product-brief_87-pg760-1-b.pdf

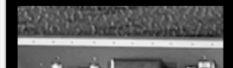
---



LTE/4G Modem Chip



CMOS QUAD 2-INPUT NAND SCHMITT TRIGGER
https://www.ti.com/lit/ds/symlink/cd4093b-q1.pdf?ts=1694496622695



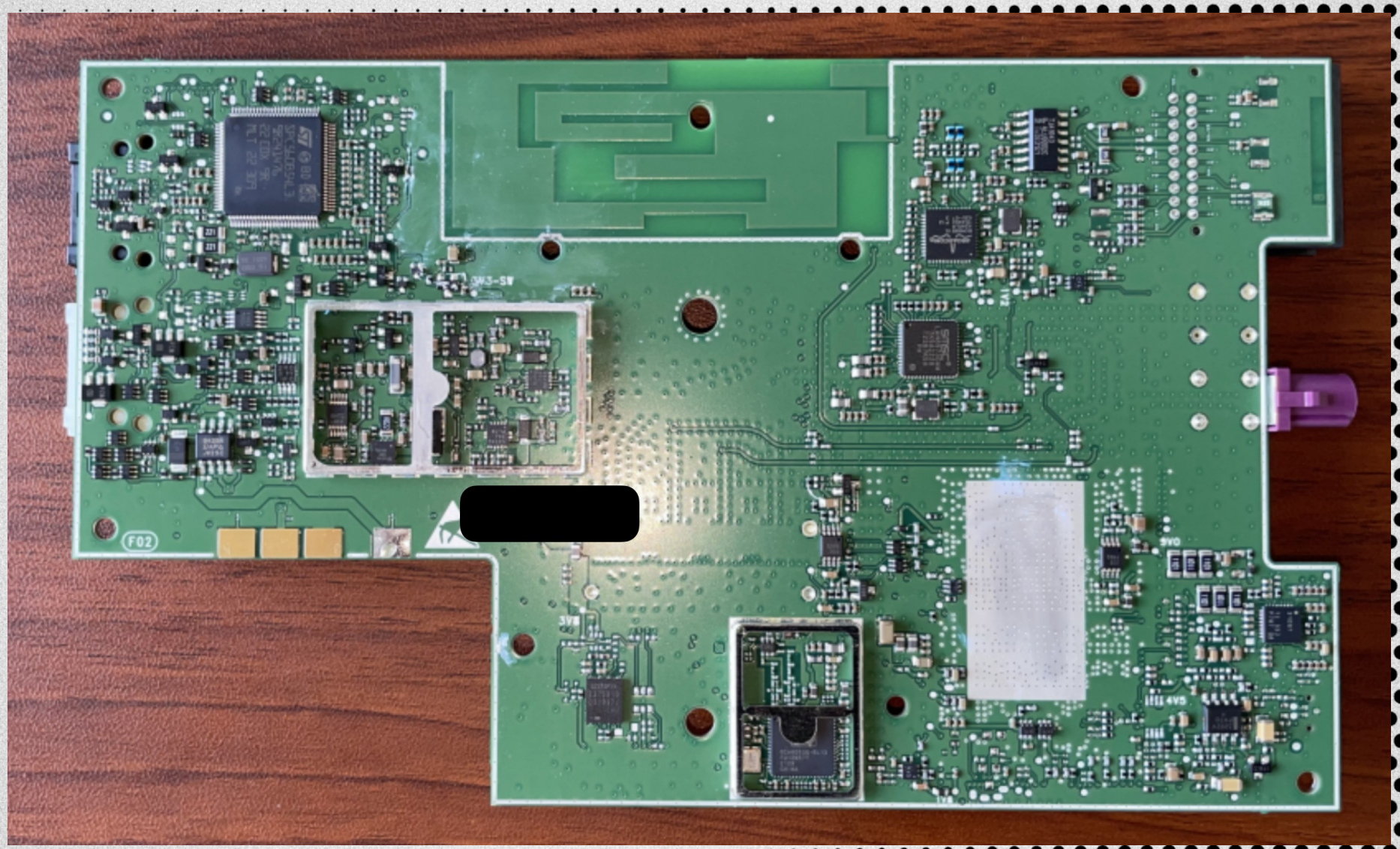


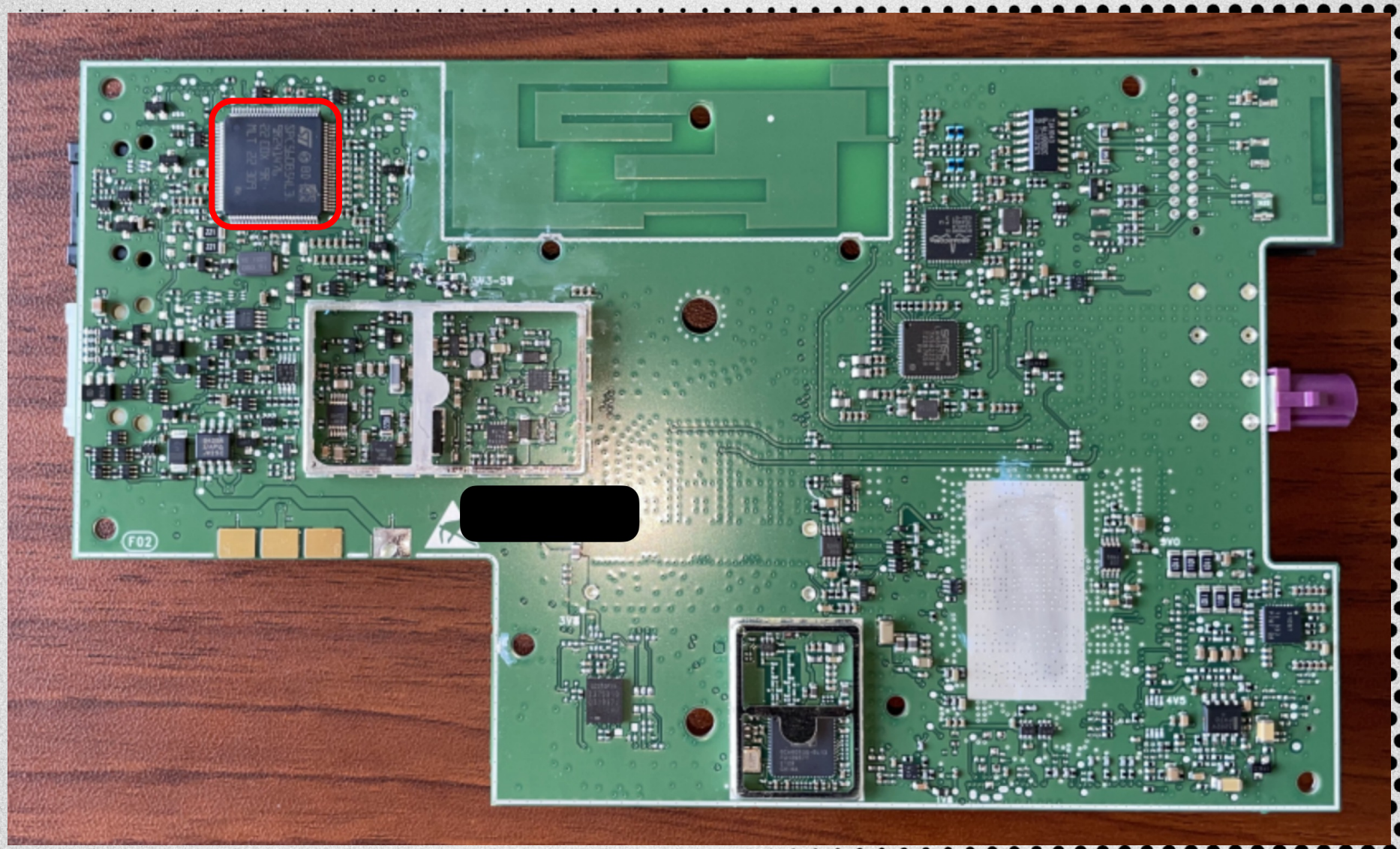
Automotive Specific Architectures

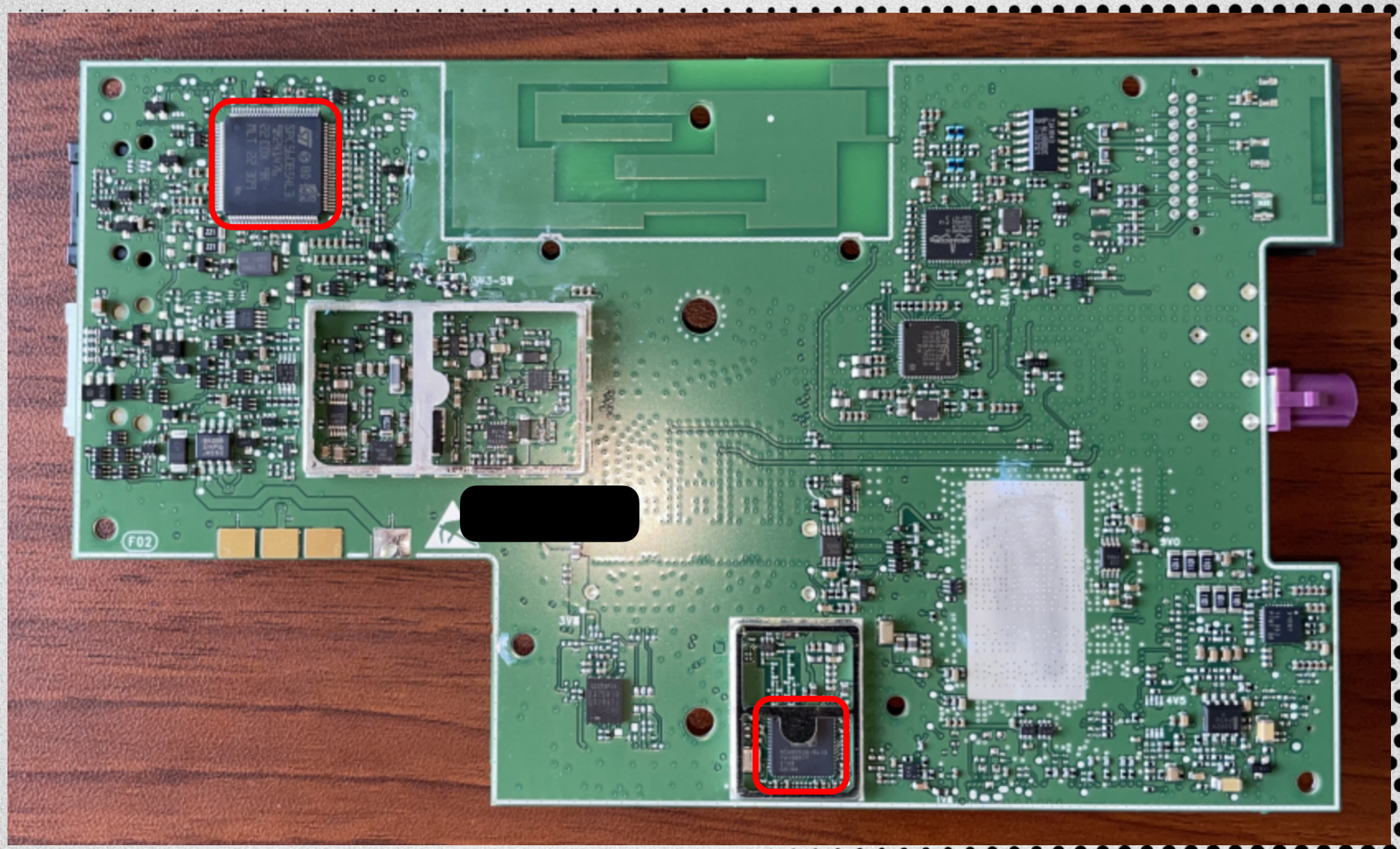
- Most Electronic Control Units (ECUs) use custom automotive specific architectures
  - e.g. Tricore, Renesas, etc.
  - Safety related functionality, with "real-time" responsiveness is required in automotive systems



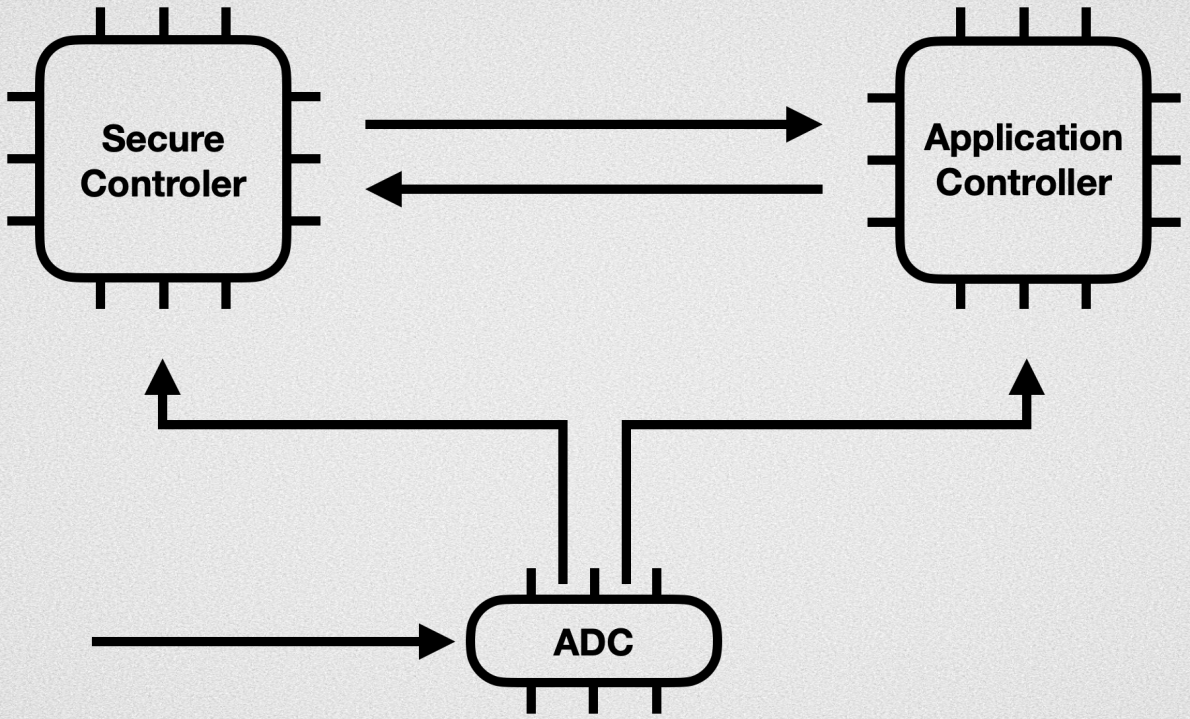



F02
3V3
5VO
4V5


F02
3V3
5V0



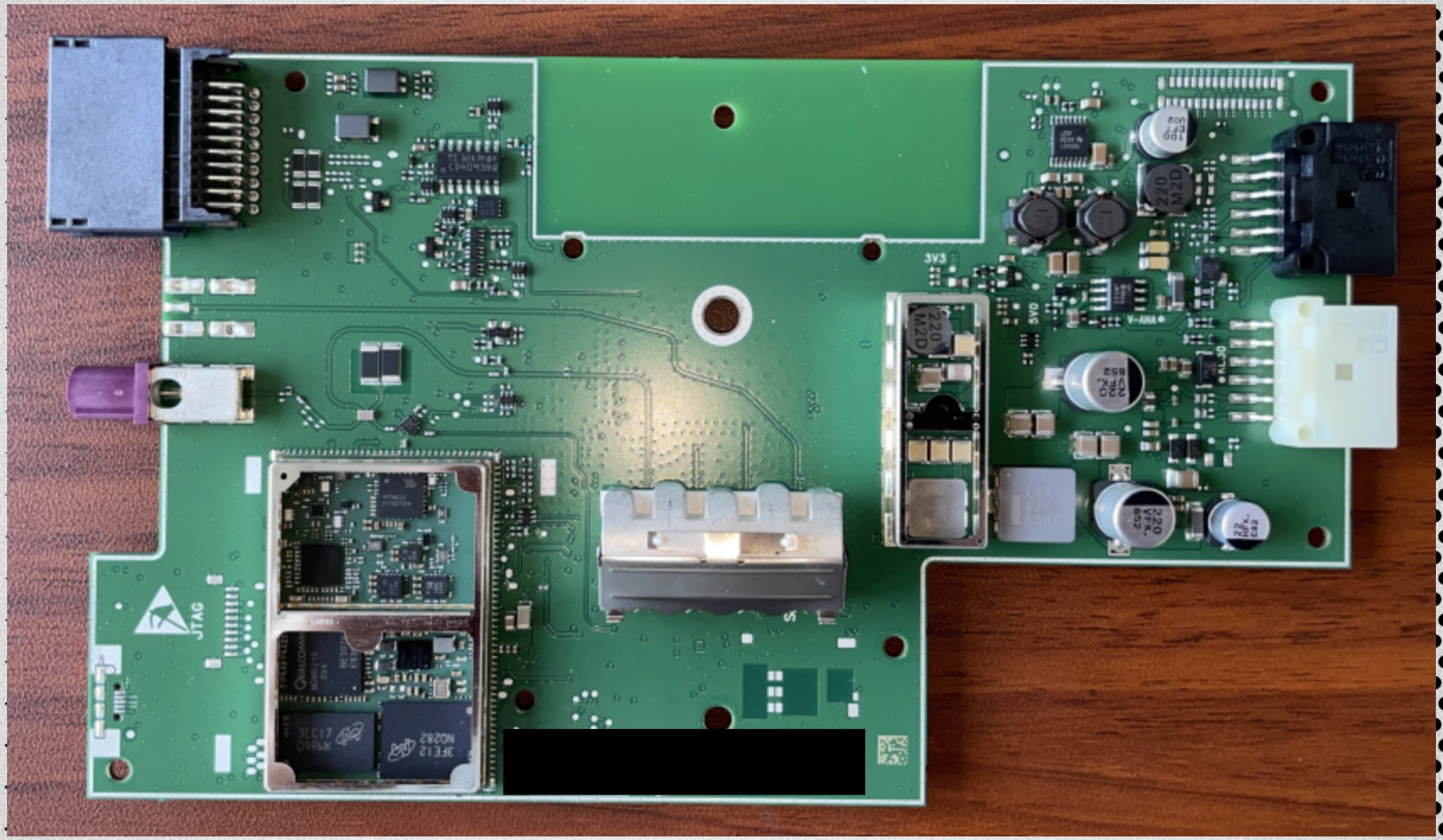
The Two SoCs - Stressful, Overcomplicated, and Confining

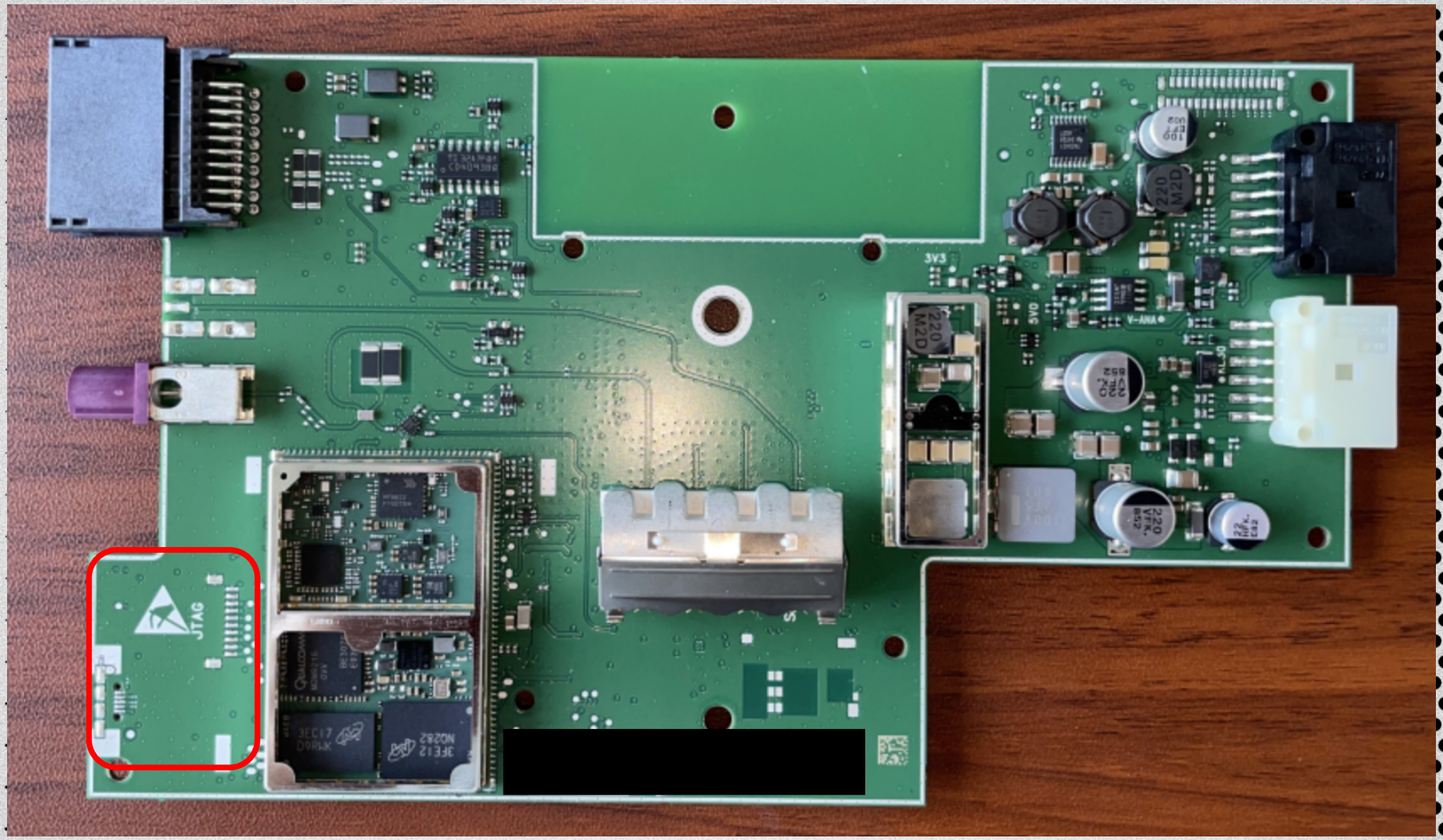


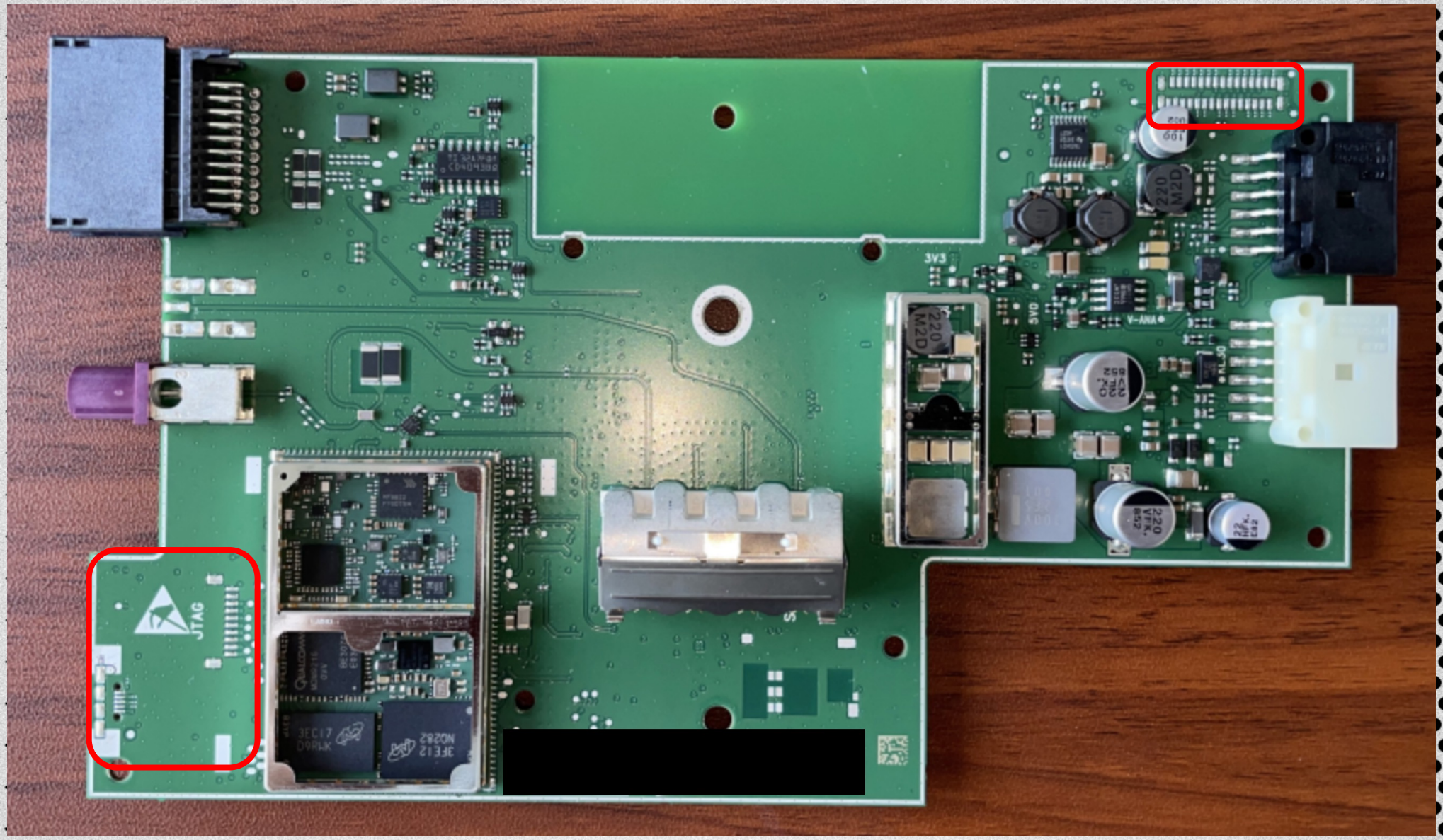
“Safety”-Critical Systems

- Many ECUs are considered "Safety-critical systems”
  - Their failure may result in loss or damage to equipment, death or injury, and environmental harm
    - NOT only applicable to the Automotive Industry
  - In automotive, it mostly applies to ECUs which supply critical functionality or can directly affect safety
    - e.g. Airbag, Braking, Power Steering, and others

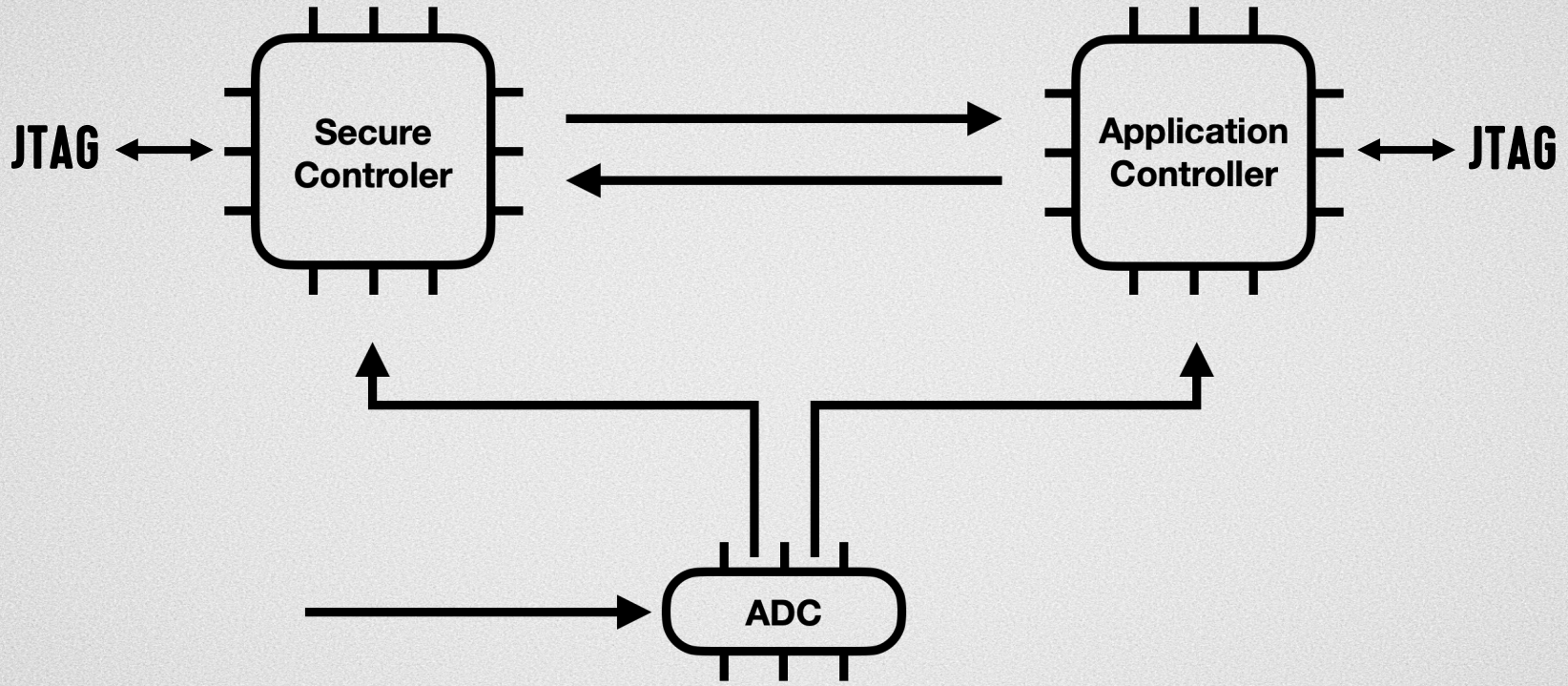



JTAG
3V3

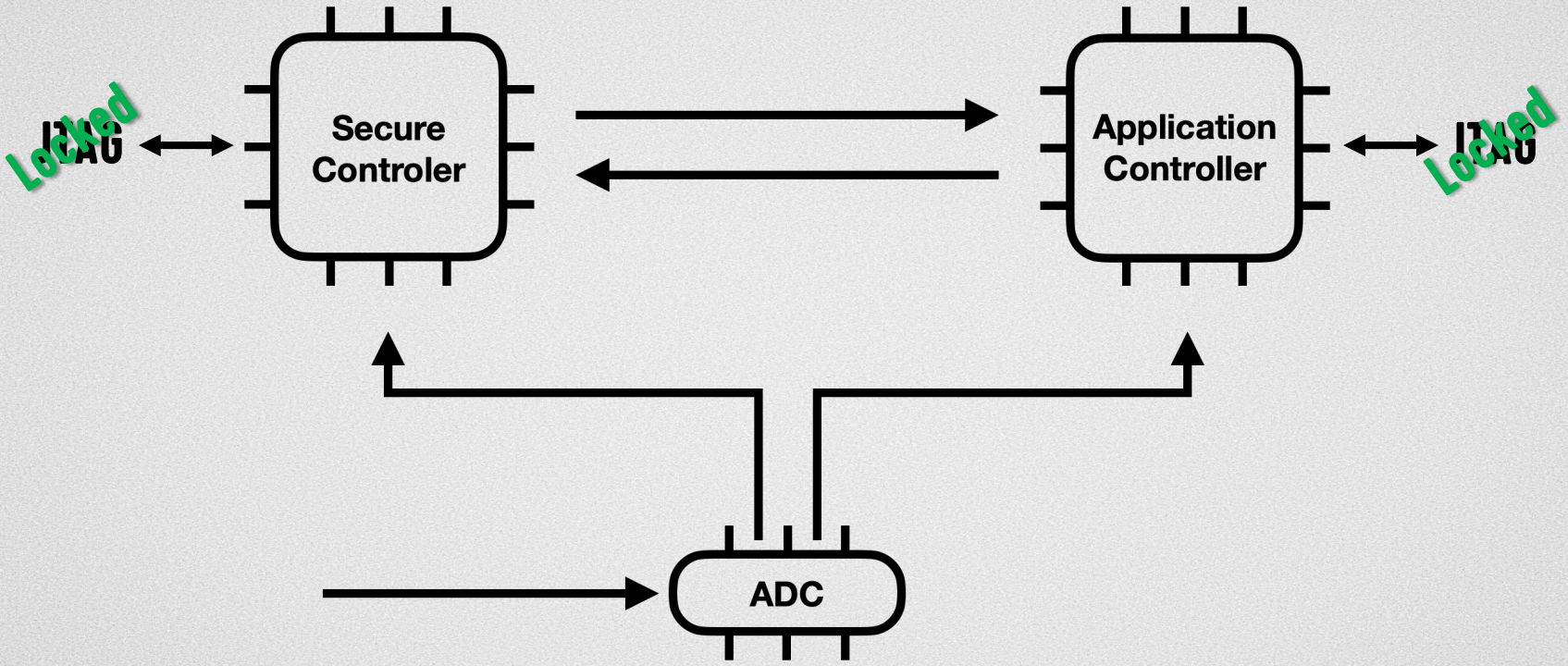

JTAG
3V3

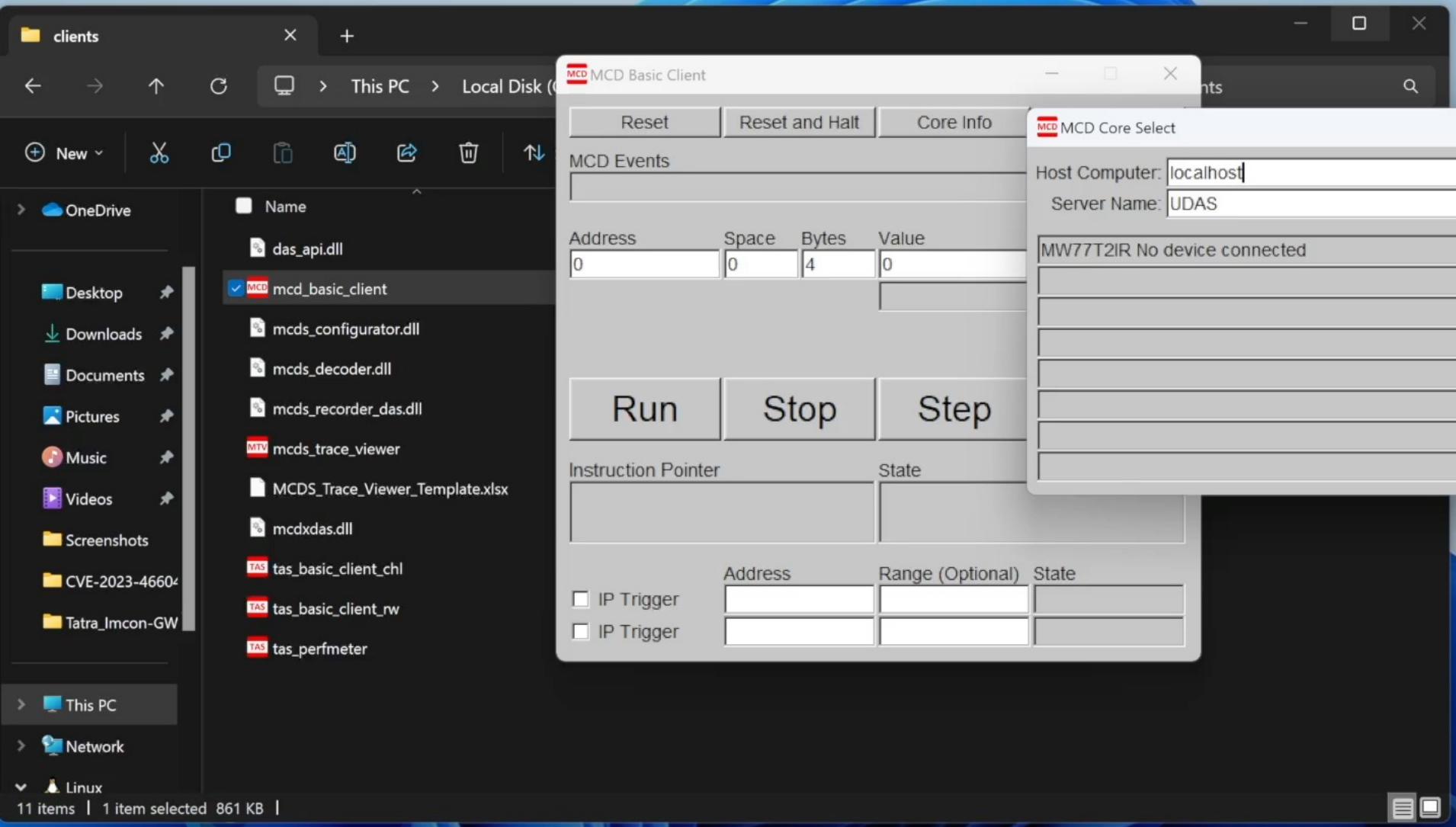

JTAG
3V3

The two SoCs - Stressful, Overcomplicated, and Confining



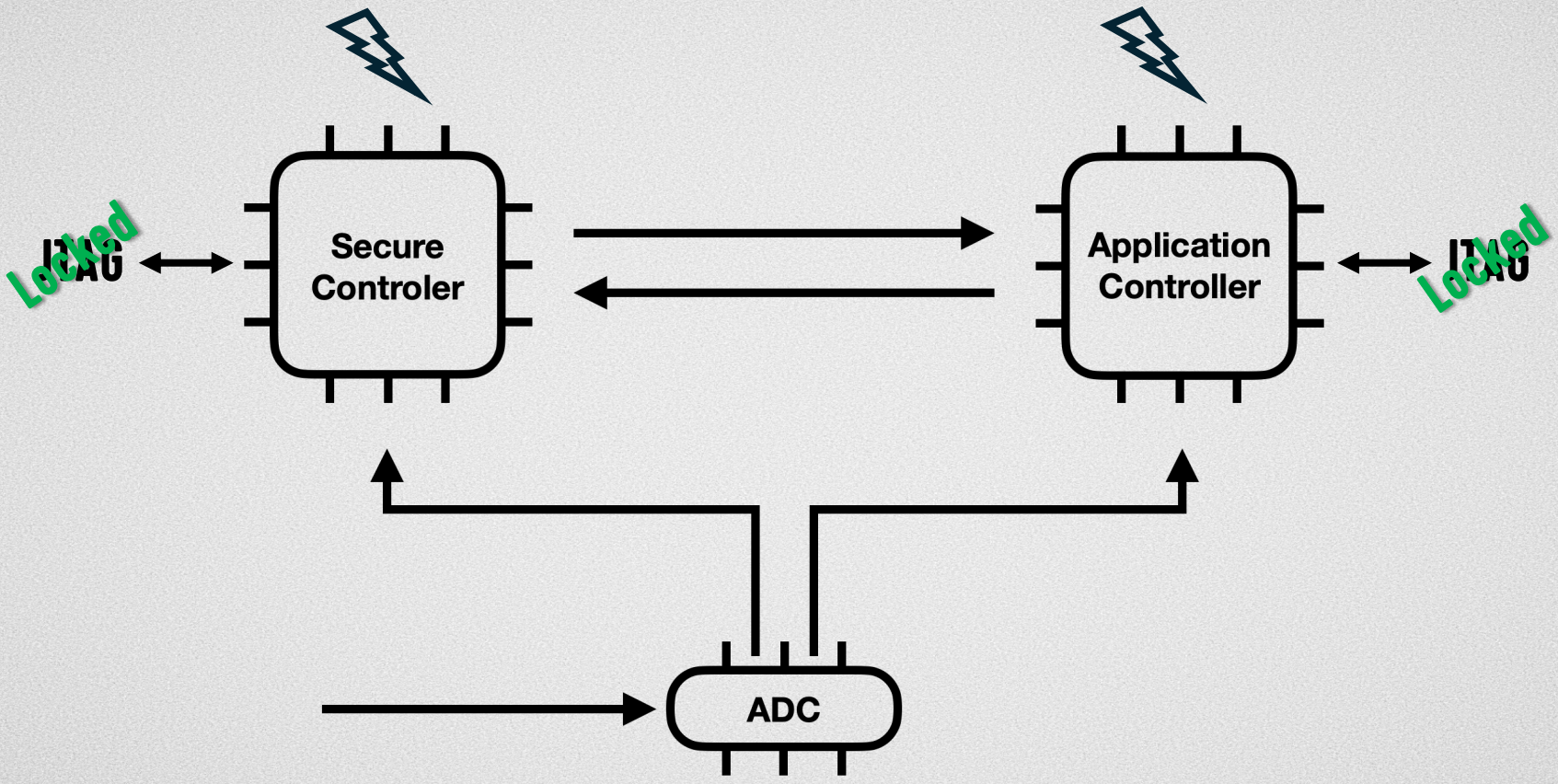
The two SoCs - Stressful, Overcomplicated, and Confining



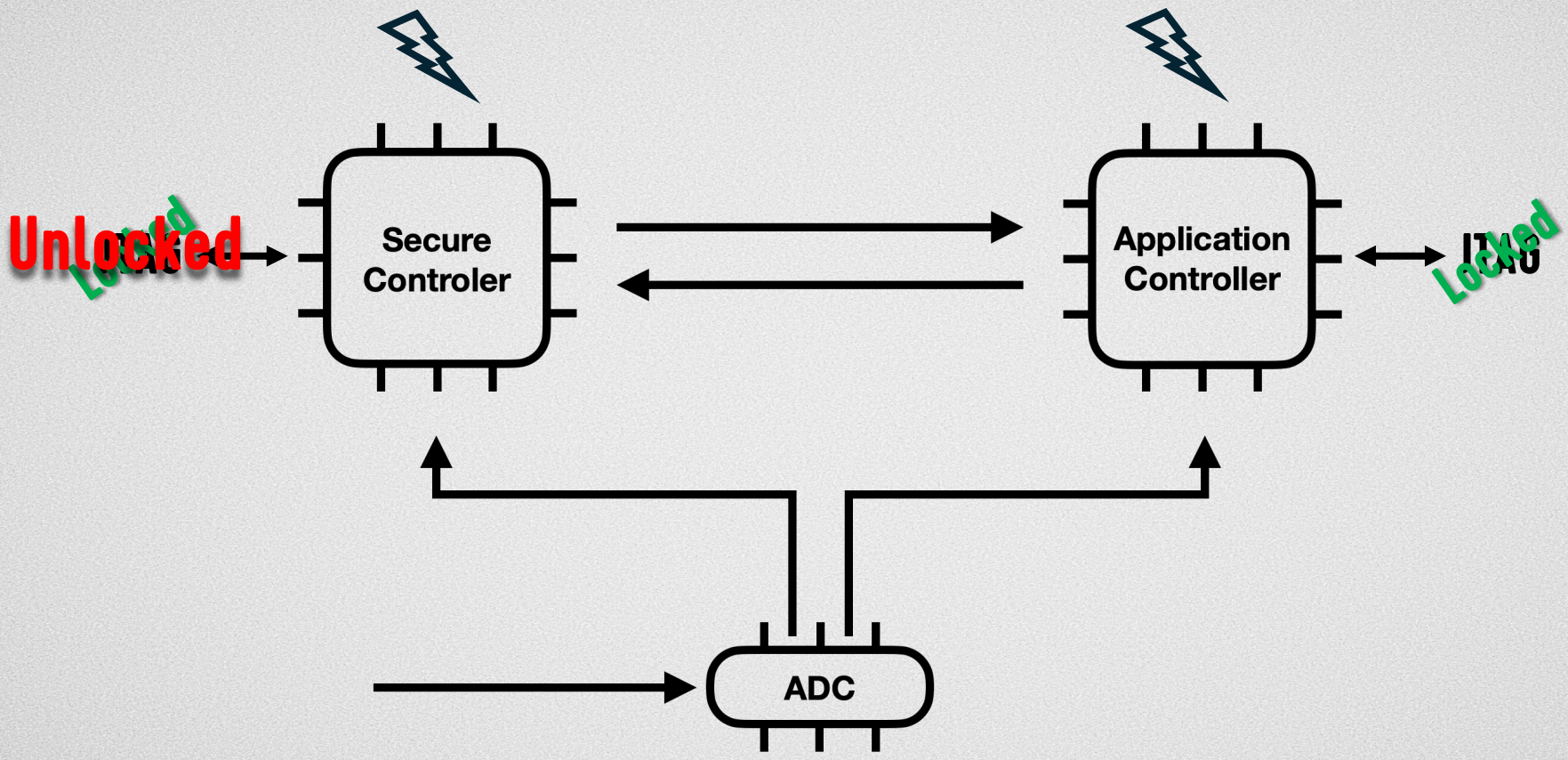

clients
This PC
Local Disk
New
OneDrive
Name
das_api.dll
mcd_basic_client
mcds_configurator.dll
mcds_decoder.dll
mcds_recorder_das.dll
mcds_trace_viewer
MCDS_Trace_Viewer_Template.xlsx
mcdxdas.dll
tas_basic_client_chl
tas_basic_client_rw
tas_perfmeter
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
CVE-2023-46604
Tatra_Imcon-GW
This PC
Network
Linux
11 items | 1 item selected 861 KB
MCD Basic Client
Reset
Reset and Halt
Core Info
MCD Events
Address
Space
Bytes
Value
0
0
4
0
Run
Stop
Step
Instruction Pointer
State
Address
Range (Optional)
State
IP Trigger
IP Trigger
MCD Core Select
Host Computer:
localhost
Server Name:
UDAS
Change
MW77T2IR No device connected
Info
Select

Demo 4

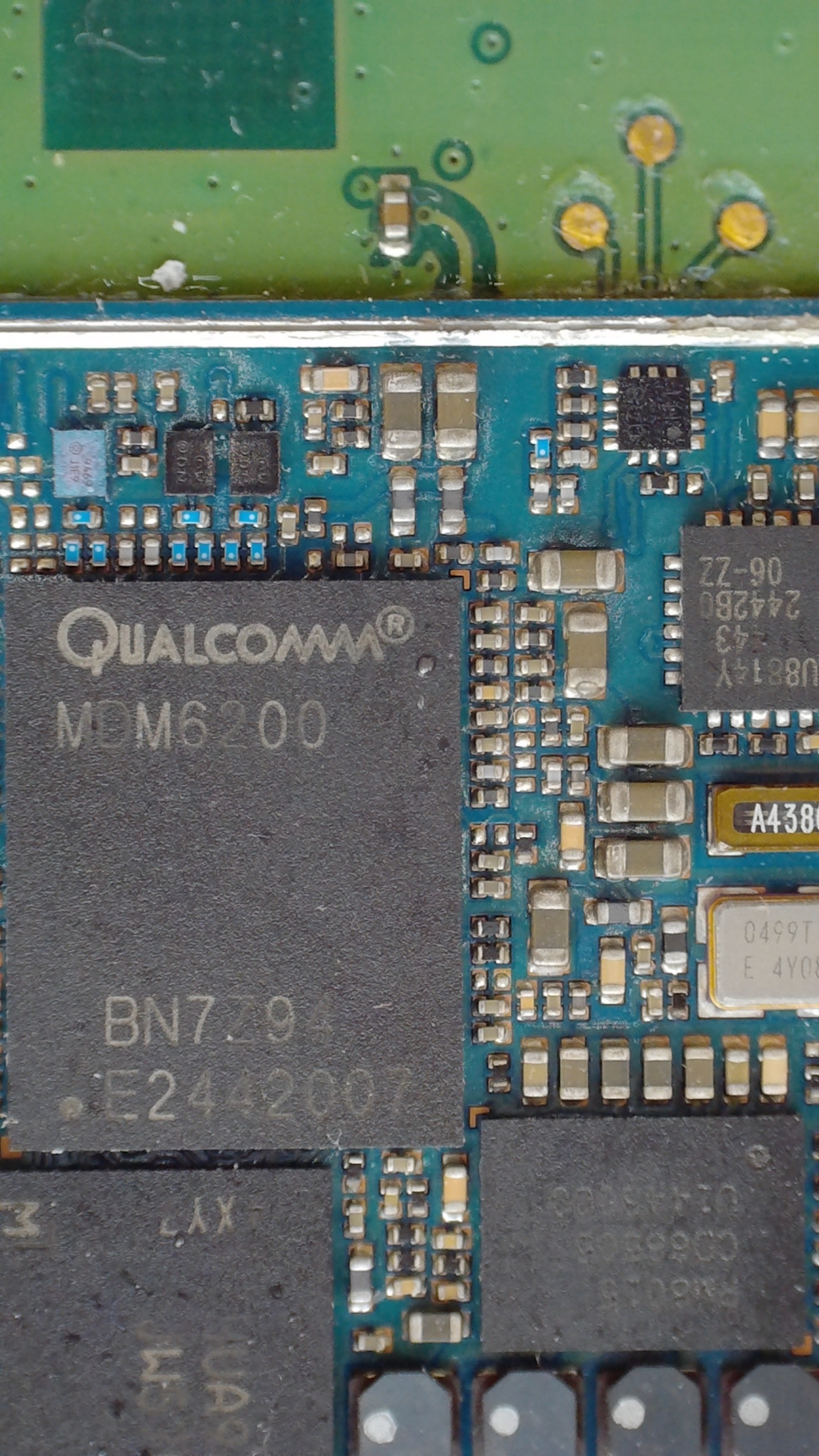
The two SoCs – Forgotten Gems



The two SoCs – Forgotten Gems

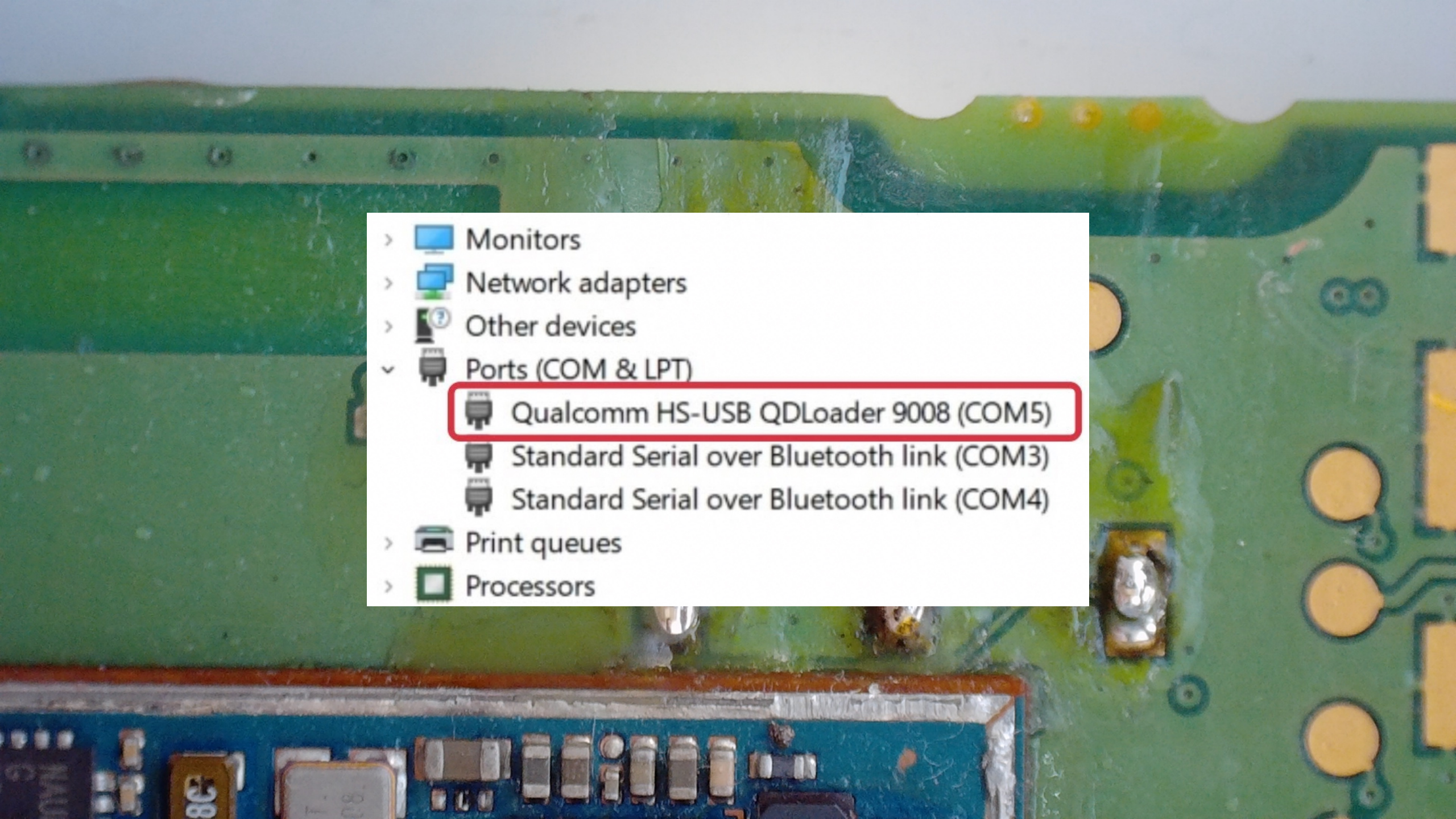



SECURITY LEVEL?
MAXIMUM!


QUALCOMM®
MDM6200

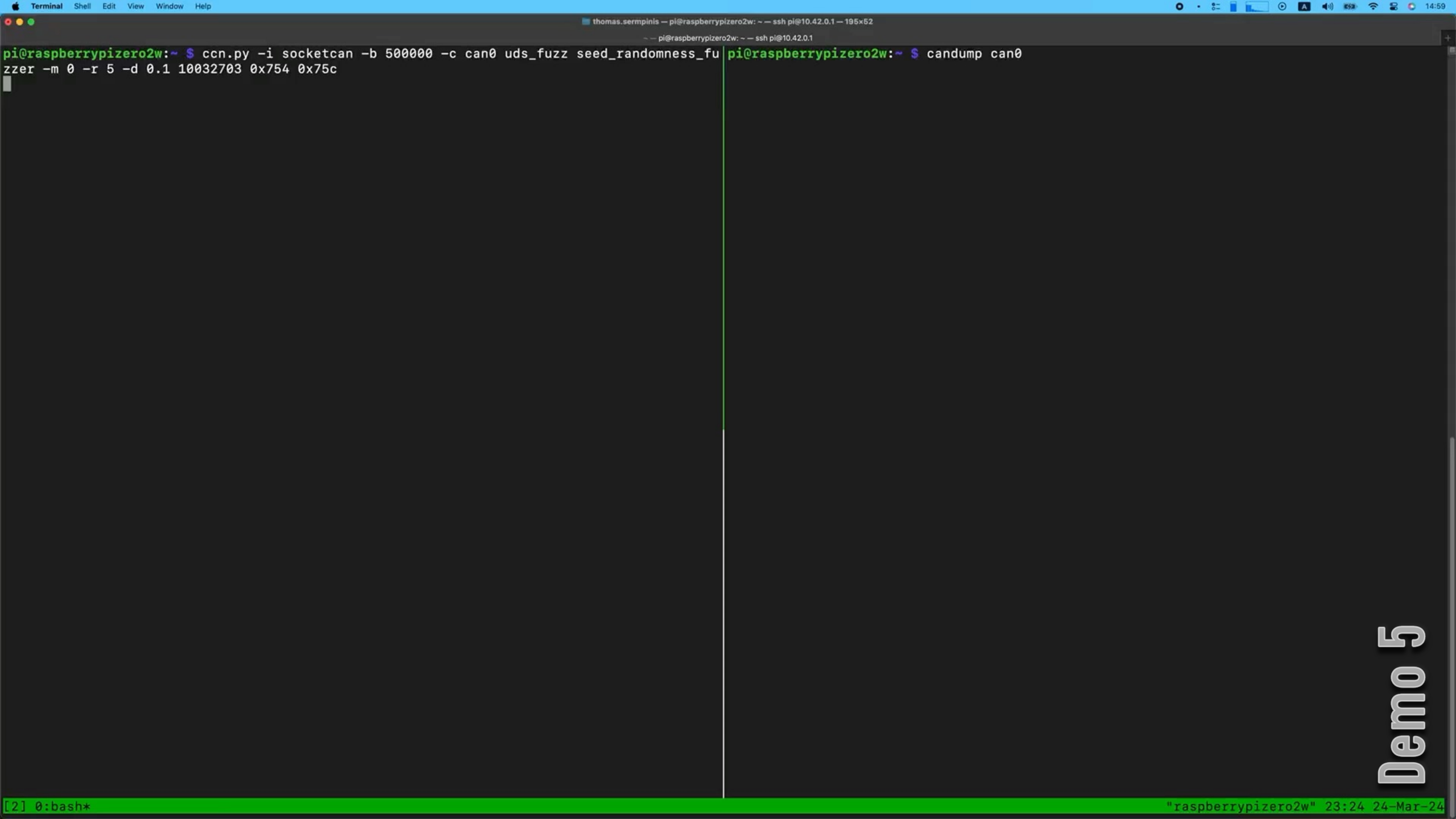
A438G
0469T
E 4Y08

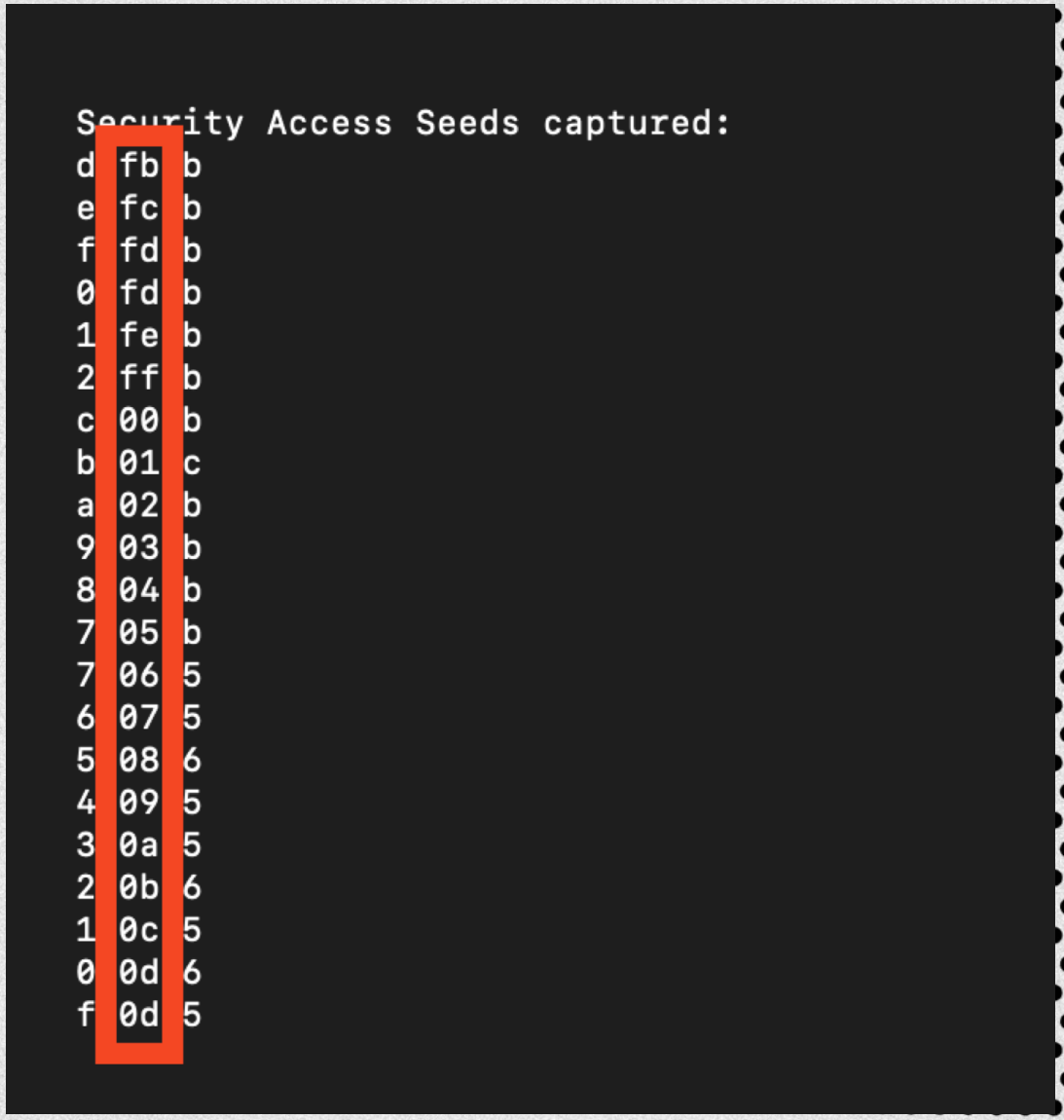


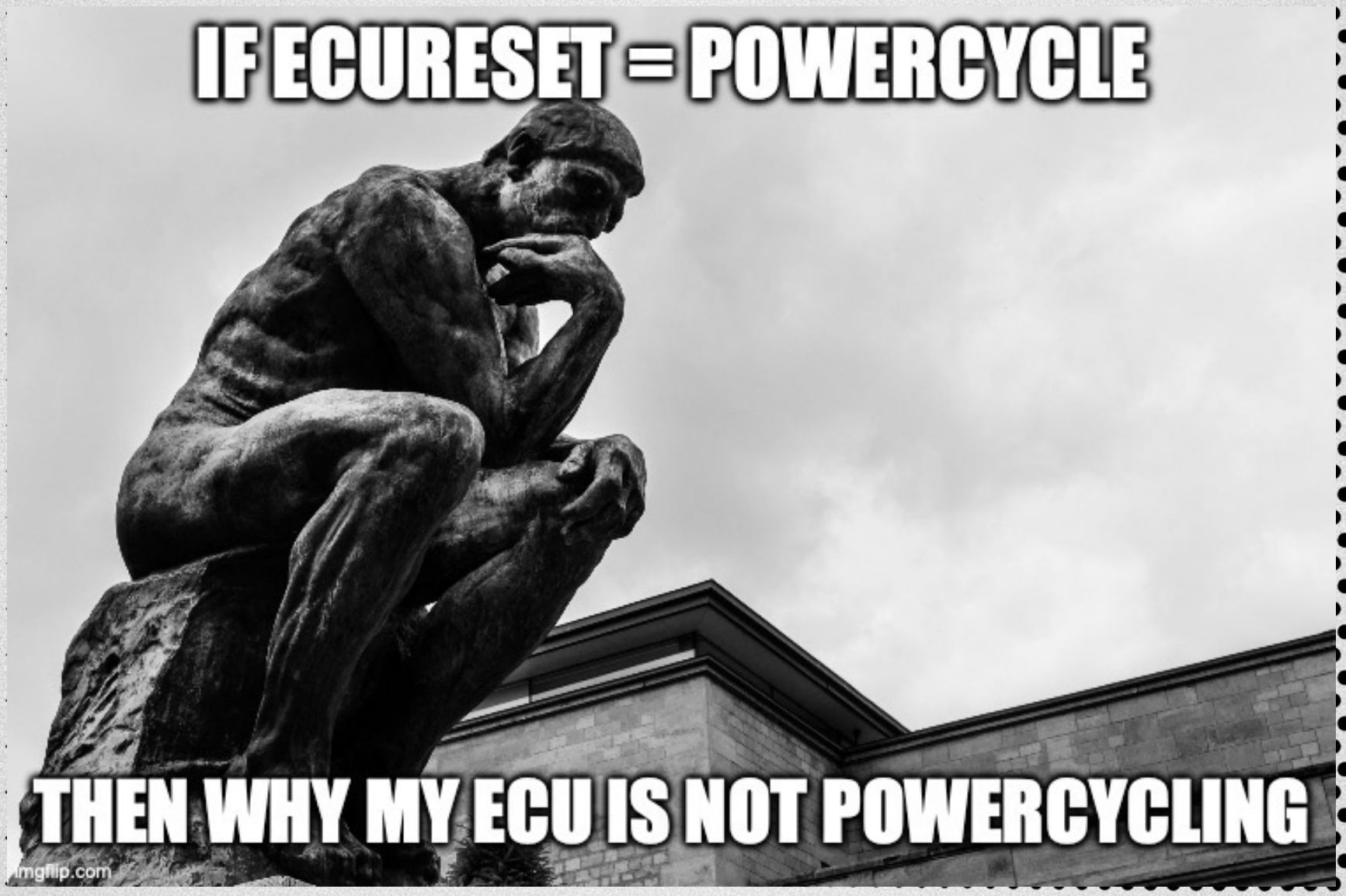

Monitors
Network adapters
Other devices
Ports (COM & LPT)
Qualcomm HS-USB QDLoader 9008 (COM5)
Standard Serial over Bluetooth link (COM3)
Standard Serial over Bluetooth link (COM4)
Print queues
Processors

Decompile: FUN_000192f6 – ([redacted].bin)

```
1
2  void FUN_000192f6(void)
3
4  {
5    undefined4 uVar1;
6    ushort uVar2;
7    undefined2 uVar3;
8    char cVar4;
9    char cVar5;
10   byte bVar6;
11   int iVar7;
12   int iVar8;
13   uint uVar9;
14
15   cVar4 = DAT_d000df5f;
16   bVar6 = DAT_d000dfd8;
17   cVar5 = DAT_d000df60;
18   if ((byte)(cVar4 - 1U) < 3) {
19     iVar7 = STM_TIM0;
20     iVar8 = STM_CAP;
21     uVar9 = STM_CLC;
22     uVar9 = uVar9 >> 8 & 7;
23     uVar2 = DAT_d000df58;
24     DAT_d000df50 = 0;
25     uVar1 = Ramd0000fcc;
26     DAT_d000df5a = ~uVar2;
27     DAT_d000df54 = uVar1;
28     DAT_d000df40 = CONCAT44(iVar8 * uVar9,iVar7 * uVar9);
29   }
30   uVar3 = DAT_d000df58;
31   DAT_d000df60 = cVar5 + '\x01';
32   DAT_d000dfd8 = bVar6 + 1 & 7;
33   (&DAT_d000dfc8)[bVar6] = uVar3;
34   return;
35 }
```


Terminal Shell Edit View Window Help
14:59
thomas.sermpinis — pi@raspberrypizero2w: ~ — ssh pi@10.42.0.1 — 195×52
~ — pi@raspberrypizero2w: ~ — ssh pi@10.42.0.1
pi@raspberrypizero2w:~ $ ccn.py -i socketcan -b 500000 -c can0 uds_fuzz seed_randomness_fu
zzer -m 0 -r 5 -d 0.1 10032703 0x754 0x75c
pi@raspberrypizero2w:~ $ candump can0
Demo 5
[2] 0:bash*
"raspberrypizero2w" 23:24 24-Mar-24

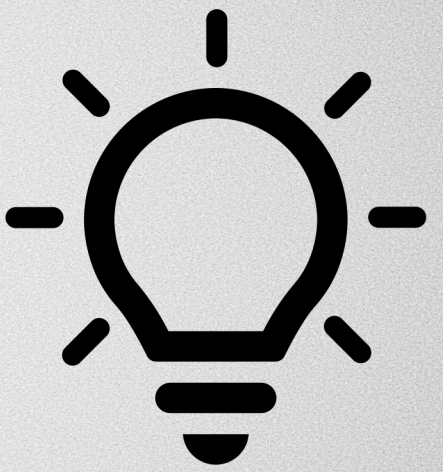

Security Access Seeds captured:
d fb b
e fc b
f fd b
0 fd b
1 fe b
2 ff b
c 00 b
b 01 c
a 02 b
9 03 b
8 04 b
7 05 b
7 06 5
6 07 5
5 08 6
4 09 5
3 0a 5
2 0b 6
1 0c 5
0 0d 6
f 0d 5


IF ECURESET = POWERCYCLE
THEN WHY MY ECU IS NOT POWERCYCLING
imgflip.com


SECURITY THROUGH OBSCURITY
imgflip.com

Reasonable Questions

- What if the ECU is restricting software powercycles for safety reasons?
- What if there were more ways to control a powercycle?
- *What if we had a nice relay?*




thomas.sermpinis — pi@raspberrypizero2w: ~/Tools/caringcaribounext — ssh pi@10.42.0.1 — 223×62
~ — pi@raspberrypizero2w: ~/Tools/caringcaribounext — ssh pi@10.42.0.1
pi@raspberrypizero2w:~/Tools/caringcaribounext $ ccn.py -i socketcan -b 500000 -c can0 uds_fuzz def
con_poc -r 0 -g 7 -il 1 -d 1.9 10032703 d3adb3 0x754 0x75c
pi@raspberrypizero2w:~/Tools/caringcaribounext $ candump can0
[1] 0:bash*
"raspberrypizero2w" 23:12 24-Mar-24
The Final Demo

What about real life scenarios?

- With this method we can effectively bypass several other mitigations:
  - Restricted ECUReset[1]
  - Restricted bootloader access[1]
  - Restricted debug access ports
- It is mainly applicable to test-bench setups, where we can effectively control the power supply of the ECU
- Can be applied to full vehicle, if it uses battery isolators
  - Mainly utilized in heavy duty vehicles



1.https://troopers.de/troopers22/agenda/tr22-993-uds-fuzzing-and-the-path-to-game-over



Source: https://www.youtube.com/watch?v=9rsloDEERoM


2
1
0
OFF
12V BATT
RESET

The Aftermath - Vol 2

- Applicable Findings:
  1. *Unauthenticated access to bootloader*
  2. *Unauthenticated access to sensitive services*
  3. *Unauthenticated access to safety related functionality*
  4. *Hardware debug interface restriction bypass*
  5. *Unauthorized access to debug functionality*
  6. *Week source of seed randomness*
  7. *Security access method bypass resulting in full ECU reprogramming*

Guess who did not receive any response?

Κεφάλαιο 3

Two Smoking Barrels

The Disturbing Truth

- These are only some, out of several other flaws that we see daily, during our testing
- Decades old vulnerabilities and misconfigurations, applicable to “modern” vehicles
- Several "zero-days" and CVEs are currently "stuck" in VDPs
  - The automotive industry is not ready for this (for the last 15 years :P)
  - Usually, 100+ year old industries, trying to catch up with young start-ups

Do they even care?

The Small OEM

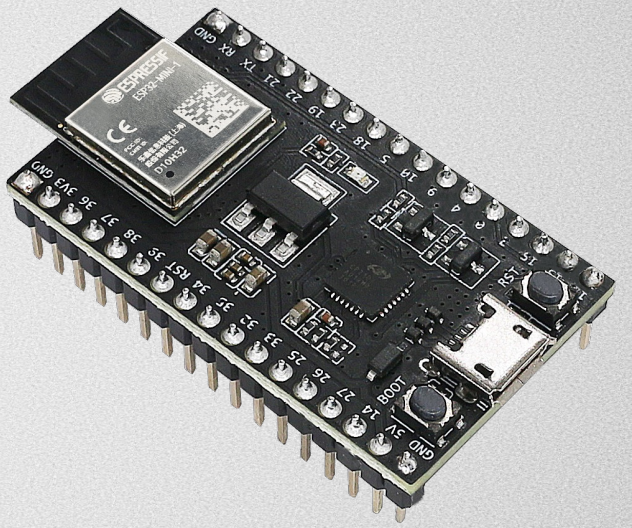
- Surprising results from one of the smallest OEMs we worked with
- While testing, none of the “common” vulnerabilities were applicable
- Device was properly protected both from hardware and software standpoint
- The underlying hardware was one of the cheapest, off-the-shelf hardware a developer can buy

Their magical solution?

One of the cheapest, off-the-shelf hardware a developer can buy :P

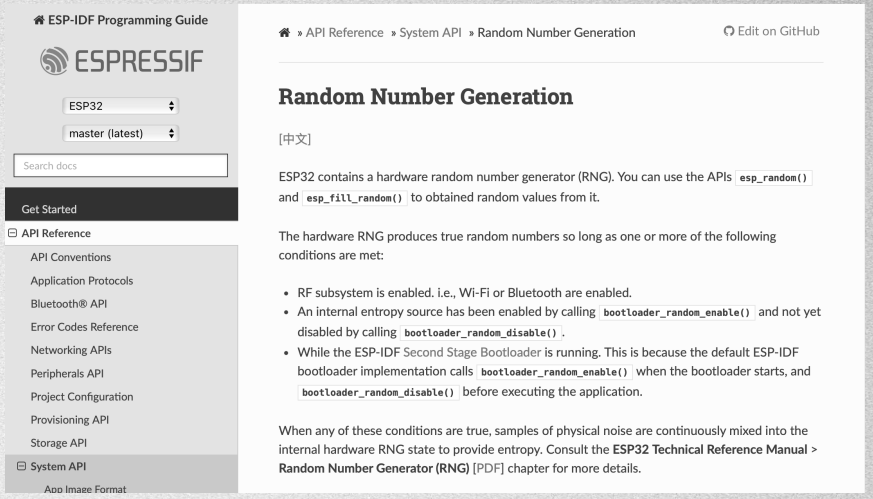
Saving the day

- ESP32 controller
  - Low-cost, low-power system on a chip microcontroller with integrated Wi-Fi and Bluetooth
  - *Not built with safety-critical functionality*
  - Mainly targeting the "home automation" industry



Saving the day

- 10$ ESP32, includes internal RNG[1]
- Uses random thermal noise, sourced from hardware ADCs[1]
- Also uses the asynchronous clock mismatch[1]
- Combines the inputs for a true random number[1]



ESP-IDF Programming Guide

ESPRESSIF

ESP32

master (latest)

Search docs

Get Started

API Reference

API Conventions

Application Protocols

Bluetooth® API

Error Codes Reference

Networking APIs

Peripherals API

Project Configuration

Provisioning API

Storage API

System API

App Image Format

» API Reference » System API » Random Number Generation

Edit on GitHub

Random Number Generation

[中文]

ESP32 contains a hardware random number generator (RNG). You can use the APIs `esp_random()` and `esp_fill_random()` to obtained random values from it.

The hardware RNG produces true random numbers so long as one or more of the following conditions are met:

- RF subsystem is enabled. i.e., Wi-Fi or Bluetooth are enabled.
- An internal entropy source has been enabled by calling `bootloader_random_enable()` and not yet disabled by calling `bootloader_random_disable()`.
- While the ESP-IDF Second Stage Bootloader is running. This is because the default ESP-IDF bootloader implementation calls `bootloader_random_enable()` when the bootloader starts, and `bootloader_random_disable()` before executing the application.

When any of these conditions are true, samples of physical noise are continuously mixed into the internal hardware RNG state to provide entropy. Consult the **ESP32 Technical Reference Manual** > **Random Number Generator (RNG)** [PDF] chapter for more details.

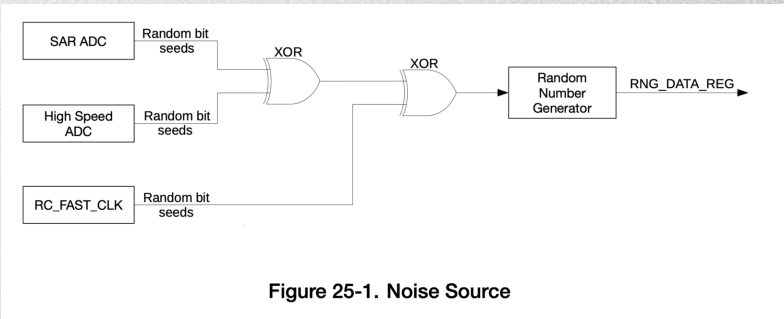


Figure 25-1. Noise Source

1. https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf#rng

Problem solved?

No one is perfect

- As said, randomness is not perfect in deterministic machines
- Several steps can be taken to isolate and control the sources of randomness
  - Out of scope of this talk
- Still better than the single, vulnerable source used in expensive safety critical components (especially the system clock)

```
* @warning This function is not safe to use if any other subsystem is accessing the RF subsystem or
*          the ADC at the same time!
```


Reference: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf#rng

Is there light in the end of the tunnel?

THE END

THANK YOU FOR YOUR ATTENTION



@cr0wtom | https://cr0wsplace.com