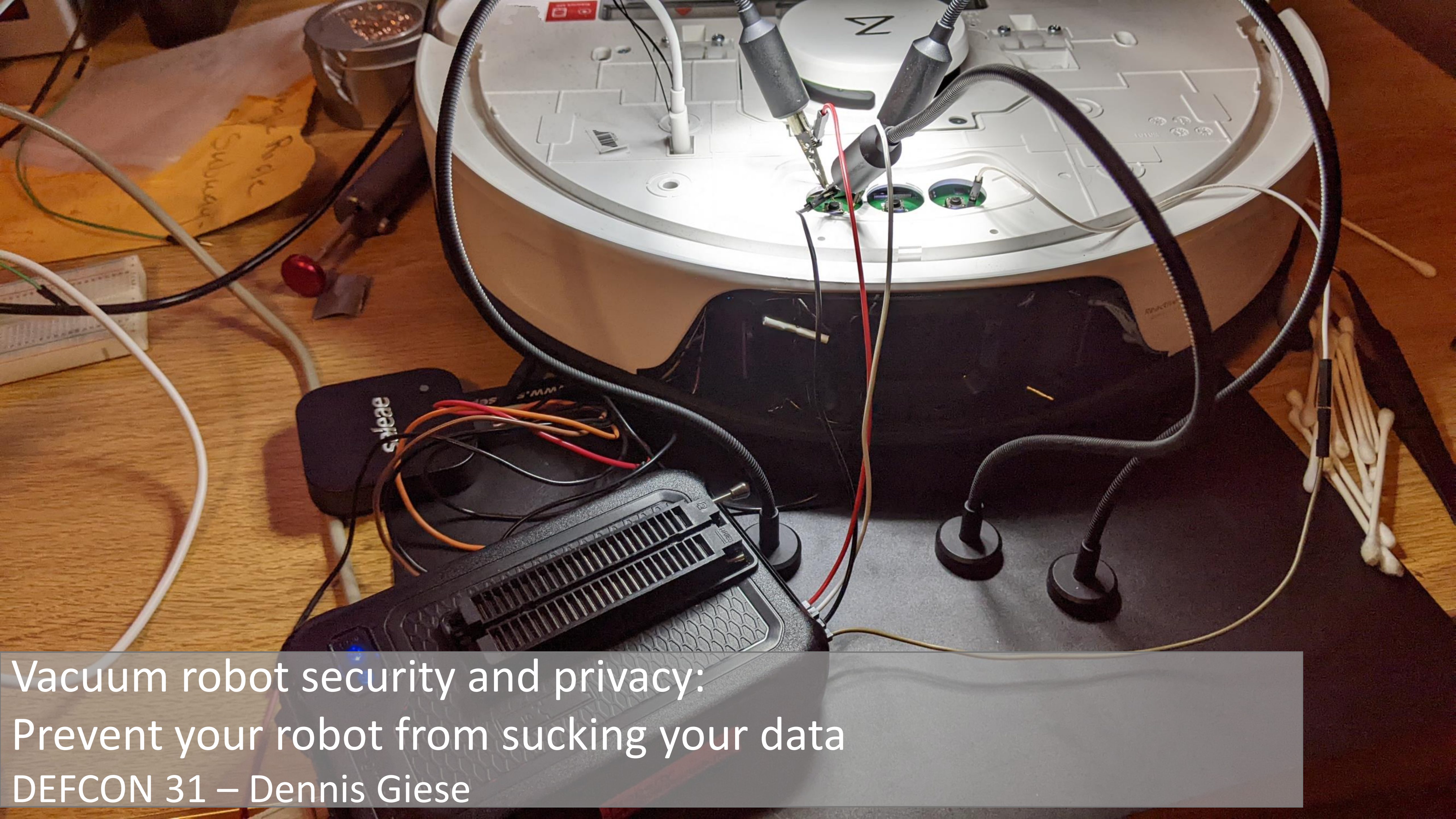


Vacuum robot security and privacy: Prevent your robot from sucking your data

DEFCON 31 – Dennis Giese

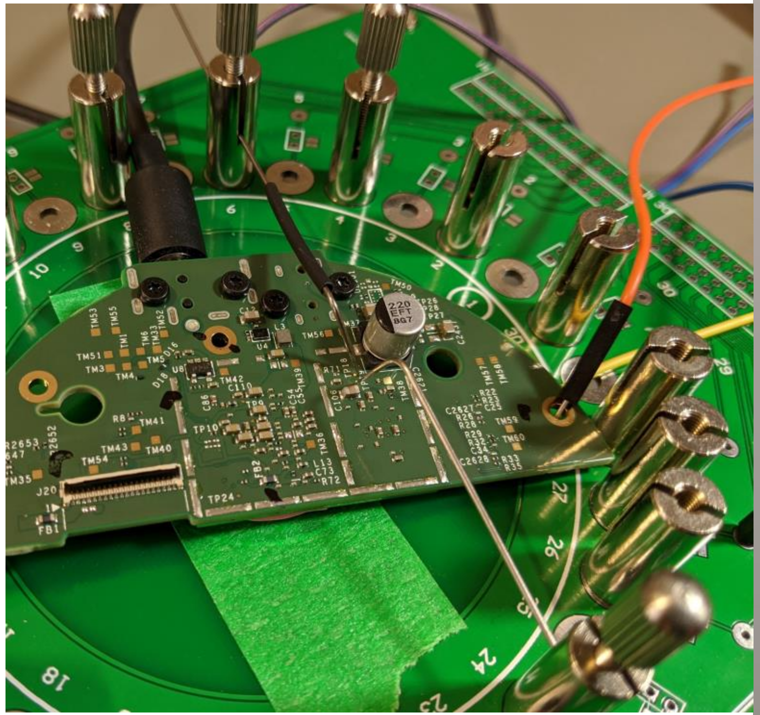
About me

Northeastern University
Khoury College of
Computer Sciences

- PhD student at Northeastern University, USA
  - Research field: Wireless and embedded Security&Privacy
- Vacuum Robot collector
- Interests: Reverse engineering of interesting devices
  - Current research: Robots, Smart Speakers, Flash memory

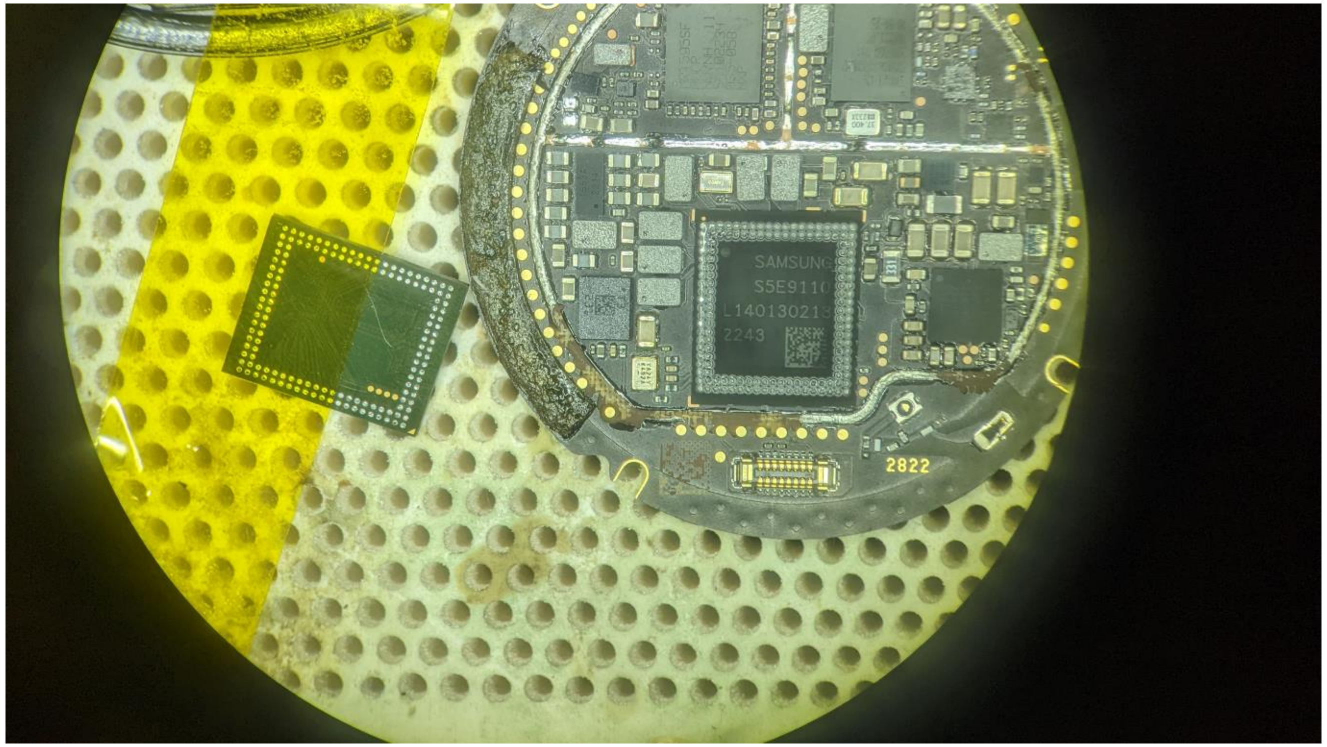
Recent projects

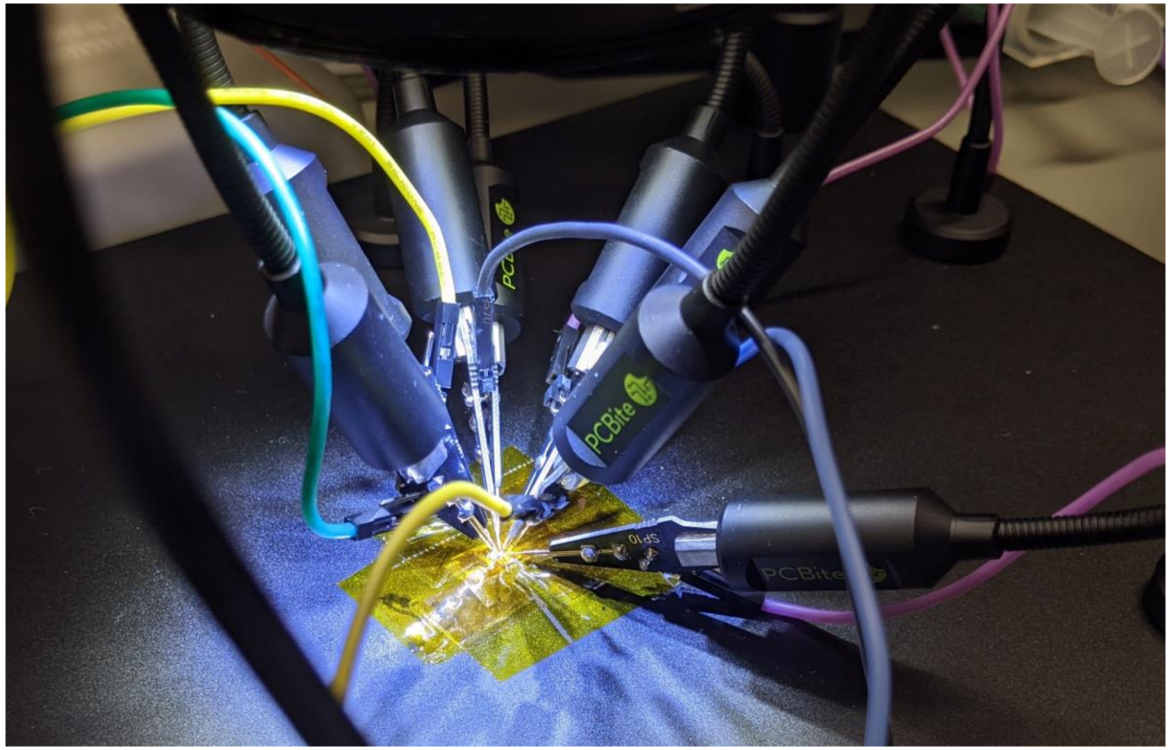
- “Amazon Echo Dot or the reverberating secrets of IoT devices”
  - Authors: Dennis Giese and Guevara Noubir



Recent projects

- Flash forensics
  - Analysis of embedded devices and flash memory itself





Recent projects

- Robotinfo.dev
  - Systematic analysis of robots
    - OS
    - Sensors
    - Vulnerabilities
  - Focus: security and privacy
  - Tracking of firmware changes
  - Source: emulated devices, app
  - Base for further research

Goals of this talk

- Get an overview of the development of vacuum robot hacking
- Learn about vulnerabilities and backdoors
- Understand methods to root current robots
  – Ultimate goal: get root access without disassembly

Side note: Generally, a friendly, but competitive relationship with vendors is maintained

https://www.dreame-technology.com/
https://www.roborock.com

Devices covered in this Talk

- Roborock S8 (a51) (04/2023)
- Roborock S8 Pro Ultra (a70) (04/2023)
- Roborock G10s (a46) (04/2022)
- Roborock Q Revo (a75) (06/2023)*
- Roborock S7 Max Ultra (a65) (06/2023)*

- Dreame L10s Ultra (r2228) (09/2022)
- Dreame D10s Pro (r2250) (11/2022)
- Dreame D10s Plus (r2240) (11/2022)
- Dreame L10s Pro (r2216) (10/2022)*
- Dreame L10 Ultra (r2257) (03/2023)*
- Xiaomi X10+ (p2114a) (10/2022)

- New rooting method for:
  - Dreame F9, D9, W10, Z10, L10
  - Xiaomi Vacuum-Mop 2 Ultra

- Indirectly:
  - Narwal Freo
  - Shark AI Robot**
  - Segway Navimow Mower

* At the time of the talk: tooling in preparation

** Applies to some modes, primarily Allwinner H3 based ones

About this talk

- Result of 15 months of research and experiments
- This talk is a collaborative effort between me and Sören Beye
- Sören Beye (aka hypfer) is the developer of Valetudo

- We would not be here if it wasn't for our community!
- The vendors have been unaware of our findings until today

  … expect patches and firmware updates in the next days/weeks

MOTIVATION

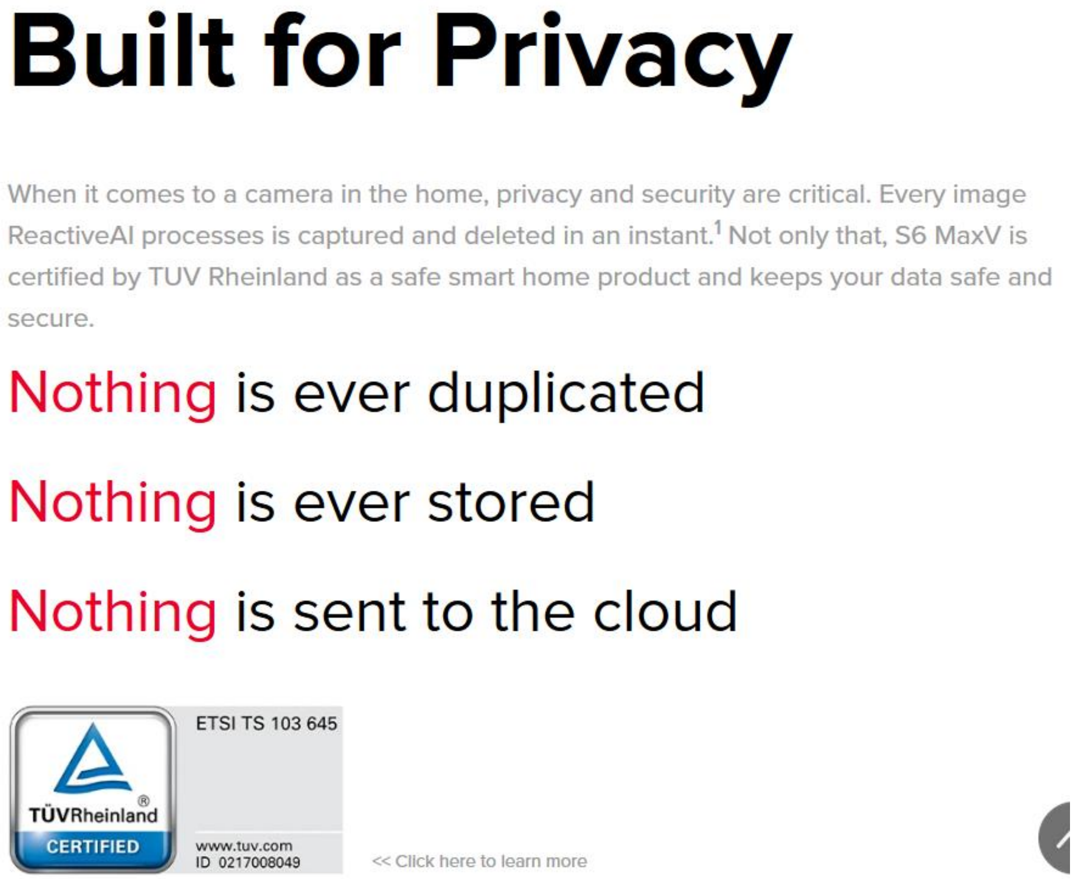
Why do we want to root devices?

- Play with cool hardware
- Stop devices from constantly phoning home
- Use custom Smart Home Software
- Diagnosis of broken devices
- Verification of privacy claims

Why do we not trust IoT?

- Devices are connected to the home network
- Communication to the cloud is encrypted, content unclear
- Developing secure hardware and software is hard
- Vendor claims contradict each other

“Nothing is sent to the cloud”?



Built for Privacy

When it comes to a camera in the home, privacy and security are critical. Every image ReactiveAI processes is captured and deleted in an instant.[1] Not only that, S6 MaxV is certified by TUV Rheinland as a safe smart home product and keeps your data safe and secure.

Nothing is ever duplicated

Nothing is ever stored

Nothing is sent to the cloud

TÜVRheinland® CERTIFIED

ETSI TS 103 645

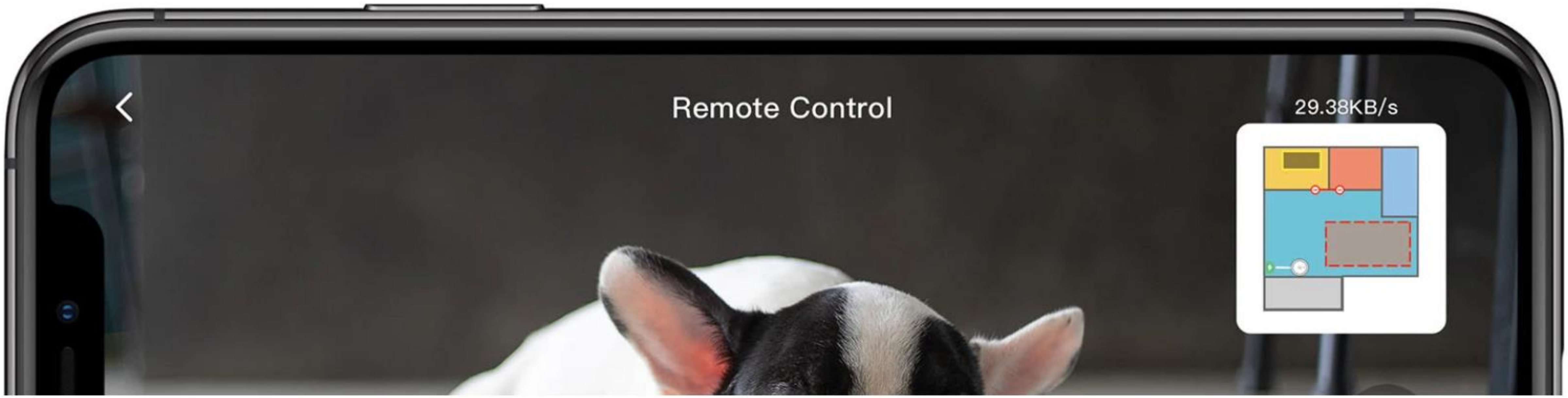
www.tuv.com
ID 0217008049

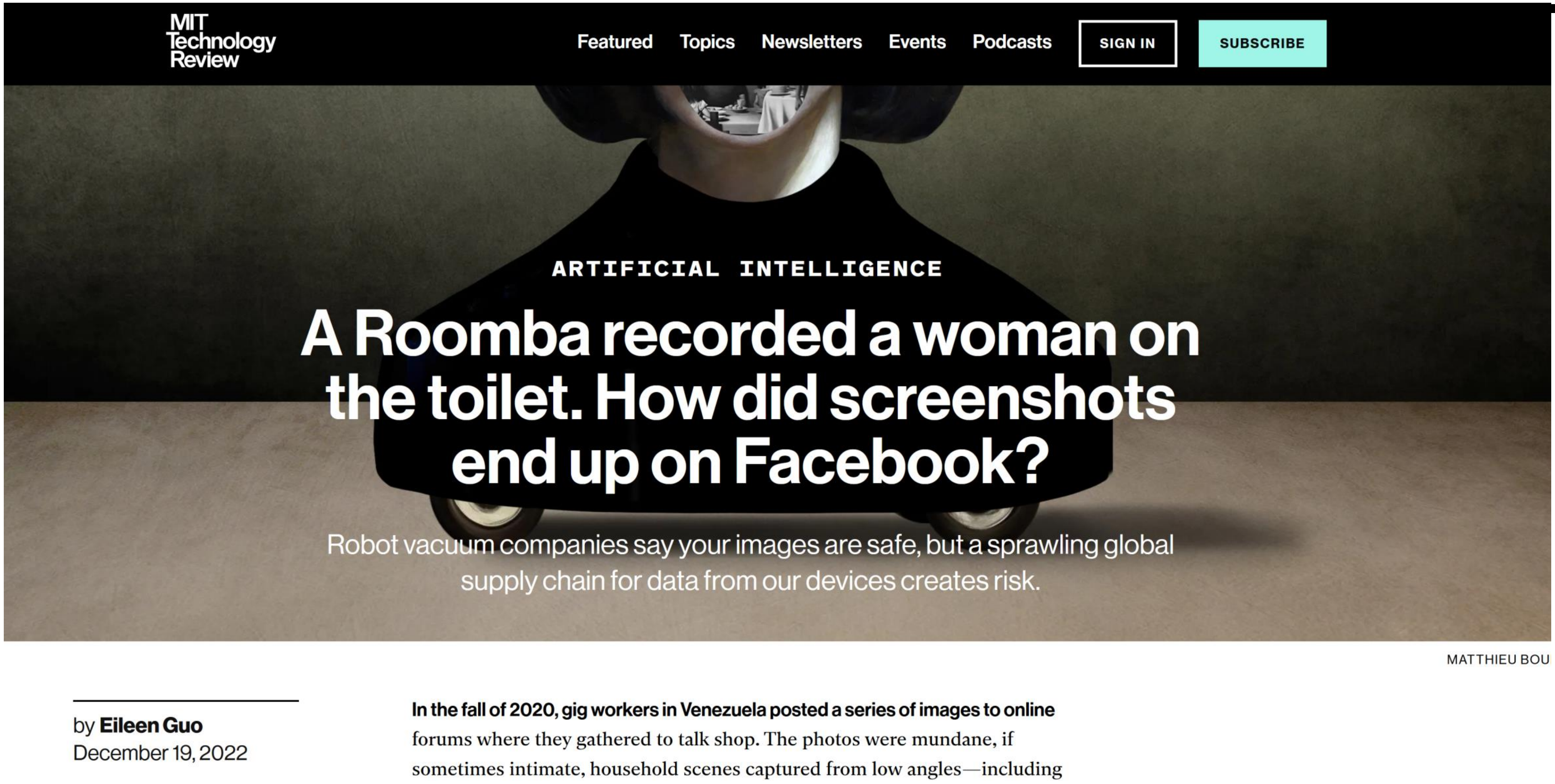
<< Click here to learn more

https://global.roborock.com/pages/roborock-s6-maxv

... but you can access the camera?

Look around your home even when you're away. Fire up the Roborock app and drive around seeing what S6 MaxV sees. Make sure you've closed your doors, reassure yourself that your home is as you left it, or check in on the mischief your pets are up to. Even send a voice message to tell them you'll be home soon.[7]





ARTIFICIAL INTELLIGENCE

A Roomba recorded a woman on the toilet. How did screenshots end up on Facebook?

Robot vacuum companies say your images are safe, but a sprawling global supply chain for data from our devices creates risk.

MATTHIEU BOU

by **Eileen Guo**
December 19, 2022

In the fall of 2020, gig workers in Venezuela posted a series of images to online forums where they gathered to talk shop. The photos were mundane, if sometimes intimate, household scenes captured from low angles—including

https://www.technologyreview.com/2022/12/19/1065306/roomba-irobot-robot-vacuums-artificial-intelligence-training-data-privacy/

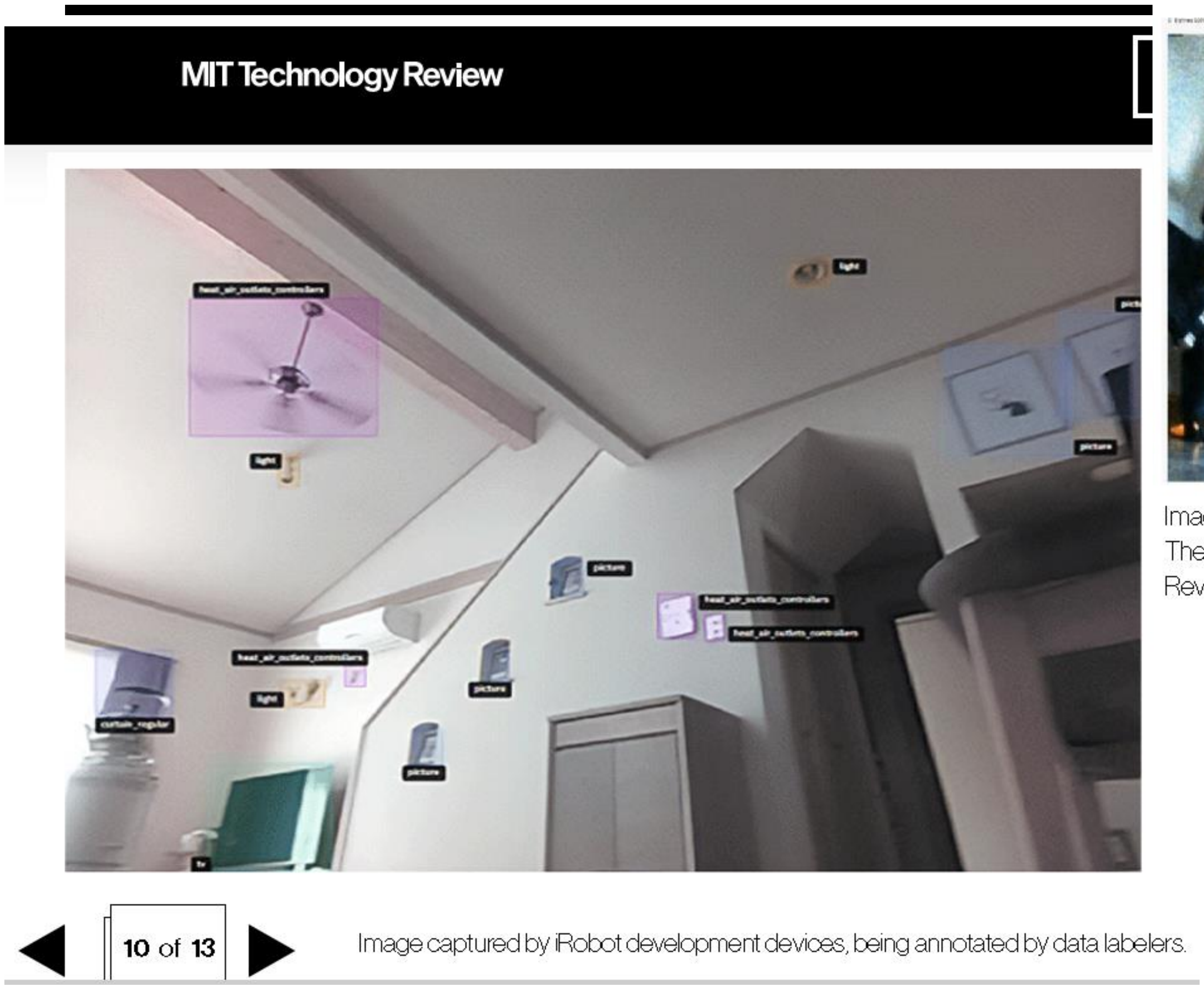

MIT Technology Review
heat_air_outlets_controllers
light
light
picture
picture
picture
picture
picture
heat_air_outlets_controllers
heat_air_outlets_controllers
heat_air_outlets_controllers
light
curtain_regular
tv
10 of 13

Image captured by iRobot development devices, being annotated by data labelers.

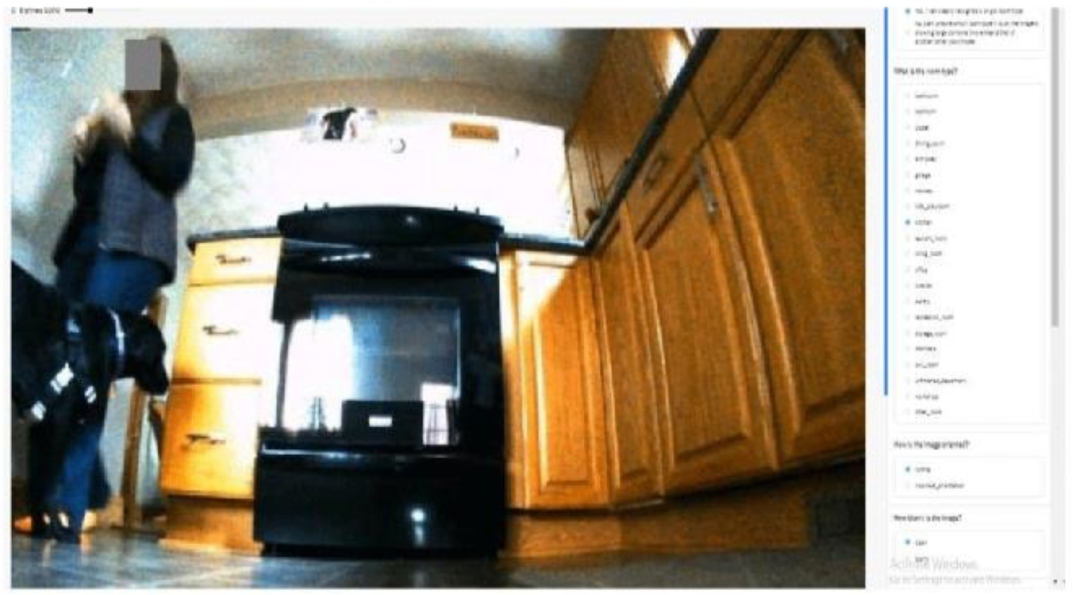
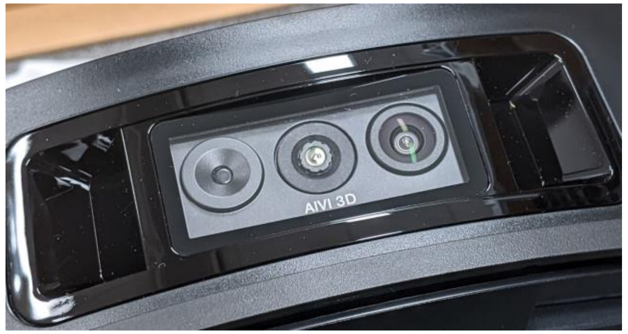
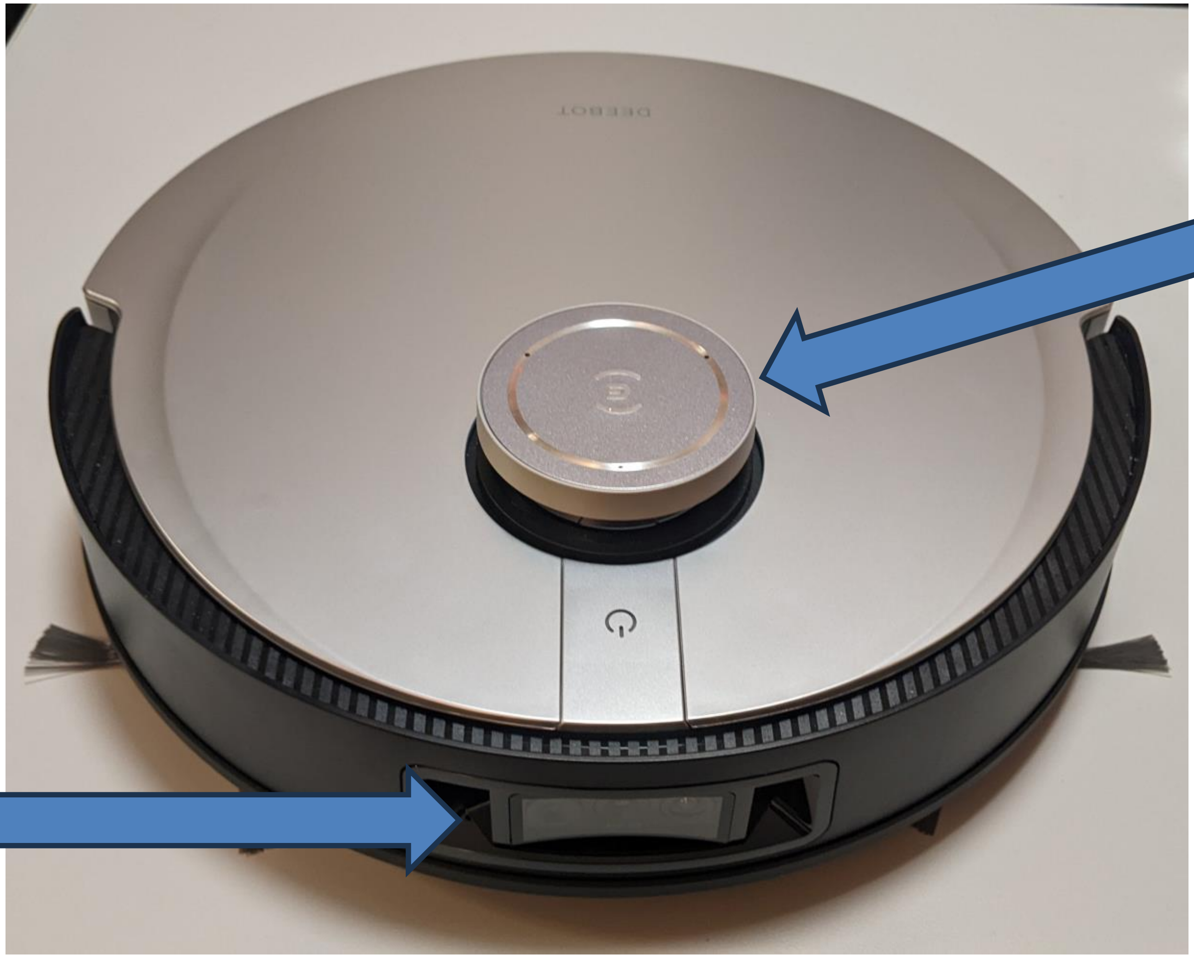


Image captured by iRobot development devices, being annotated by data labelers. The woman's face was originally visible, but was obscured by MIT Technology Review. The Roomba J7's front light is reflected on the oven.

Fun fact:
Vendors panicked and started to change firmwares, apps and privacy policies

More sensors?





Microphones??

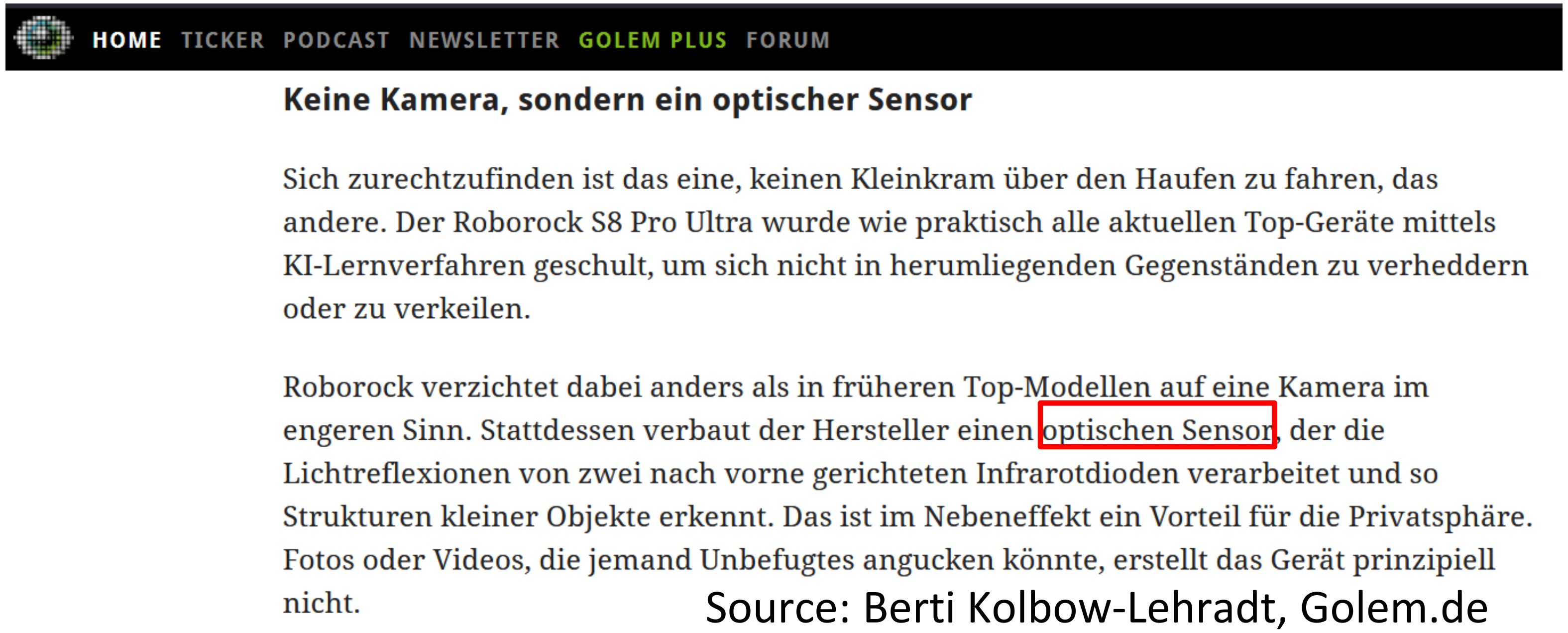
Cameras

Risks of devices with cameras

- Devices might store pictures indefinitely… and some do. both cloud and local
- Used devices might be problematic
  - Previous owner installed rootkit
  - New owner cannot verify software
  - Result: Device might behave maliciously on your network
- Rooting is the only way to verify that a device is „clean“

Optical sensors instead of camera?

- Vendors react on worries of users:
  - Avoidance of the word “camera” for privacy aware users
  - Usage of “optical sensor” instead

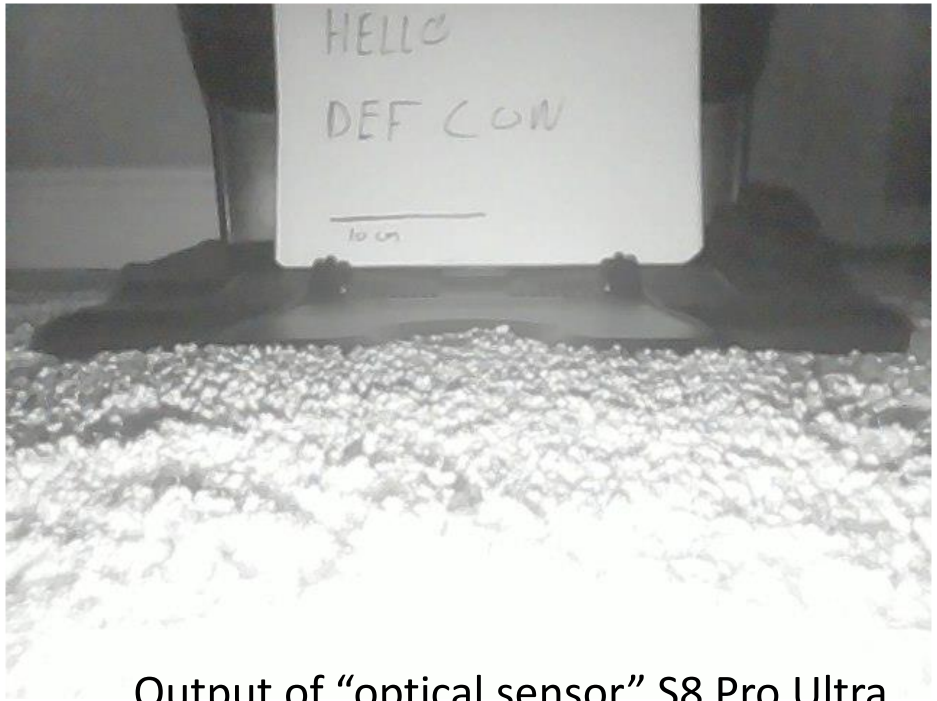


Keine Kamera, sondern ein optischer Sensor

Sich zurechtzufinden ist das eine, keinen Kleinkram über den Haufen zu fahren, das andere. Der Roborock S8 Pro Ultra wurde wie praktisch alle aktuellen Top-Geräte mittels KI-Lernverfahren geschult, um sich nicht in herumliegenden Gegenständen zu verheddern oder zu verkeilen.

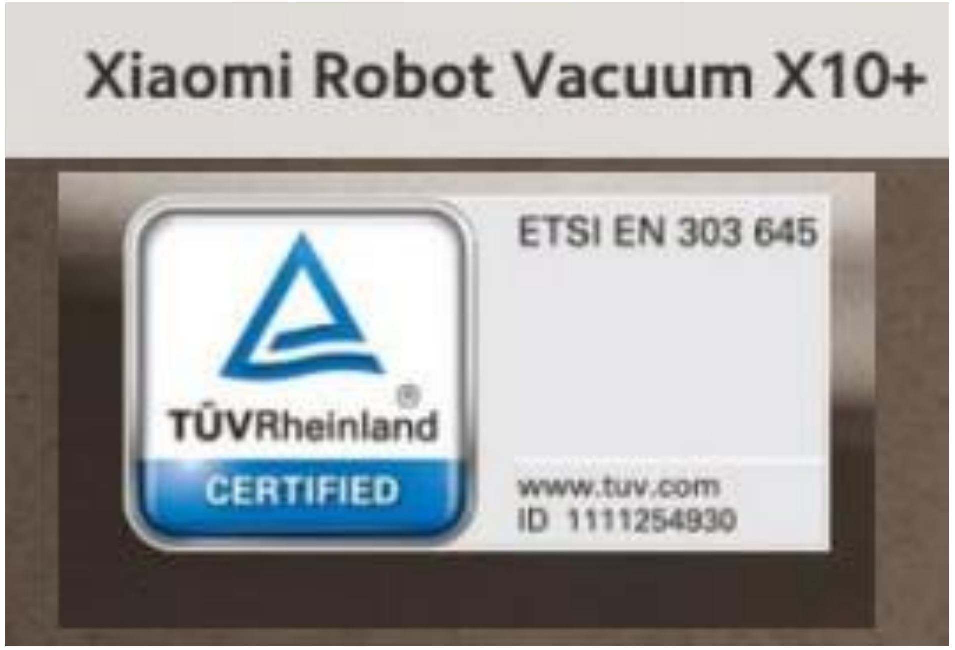
Roborock verzichtet dabei anders als in früheren Top-Modellen auf eine Kamera im engeren Sinn. Stattdessen verbaut der Hersteller einen optischen Sensor, der die Lichtreflexionen von zwei nach vorne gerichteten Infrarotdioden verarbeitet und so Strukturen kleiner Objekte erkennt. Das ist im Nebeneffekt ein Vorteil für die Privatsphäre. Fotos oder Videos, die jemand Unbefugtes angucken könnte, erstellt das Gerät prinzipiell nicht.

Source: Berti Kolbow-Lehradt, Golem.de



Output of “optical sensor” S8 Pro Ultra

Certifications are important

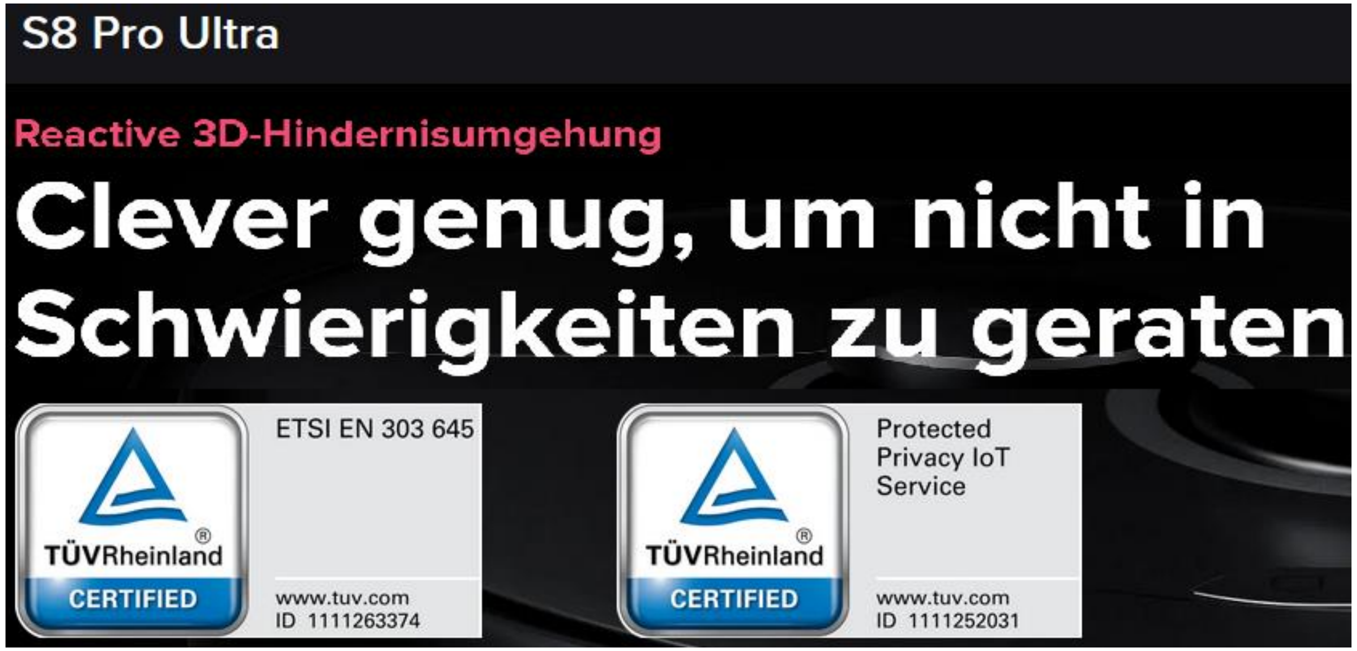

Xiaomi Robot Vacuum X10+

ETSI EN 303 645

TÜVRheinland CERTIFIED

www.tuv.com
ID 1111254930

Source: https://www.mi.com/global/product/xiaomi-robot-vacuum-x10-plus/


S8 Pro Ultra

Reactive 3D-Hindernisumgehung

Clever genug, um nicht in Schwierigkeiten zu geraten

TÜVRheinland CERTIFIED

ETSI EN 303 645

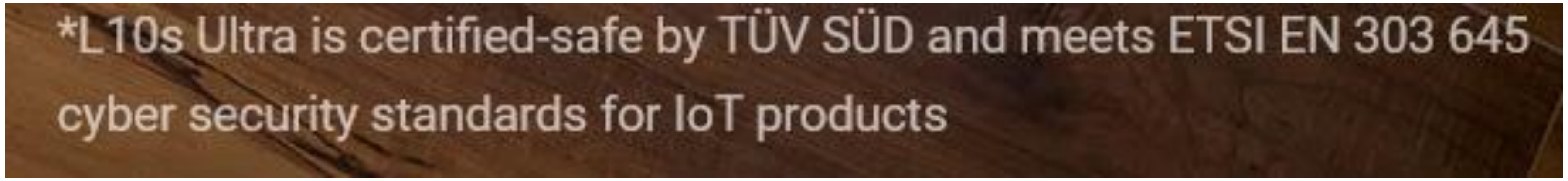
www.tuv.com
ID 1111263374

TÜVRheinland CERTIFIED

Protected Privacy IoT Service

www.tuv.com
ID 1111252031

Source: https://de.roborock.com/pages/roborock-s8-pro-ultra


*L10s Ultra is certified-safe by TÜV SÜD and meets ETSI EN 303 645 cyber security standards for IoT products

Source: https://www.dreametech.com/products/dreamebot-l10s-ultra

WHAT HAPPENED SO FAR

General observation

- Every time a rooting method gets released, the vendors react
  - Sometimes they even break things in the process ☹
- Best case for us: rooting method just fails
- Worse cases for us:
  - Rooting succeeds, but device breaks “randomly”
  - Device destroys itself

First work in 2017

- Work together with Daniel Wegemer
- Xiaomi Vacuum Robot / Roborock S5
- Findings:
  - Firmware images: unsigned and encrypted with weak key
  - Custom firmware could be pushed from local network
- Result:
  - Rooting without disassembly
  - Development of custom Software and Voice packages
- Publication: 34C3 (2017) and DEF CON 26 (2018)

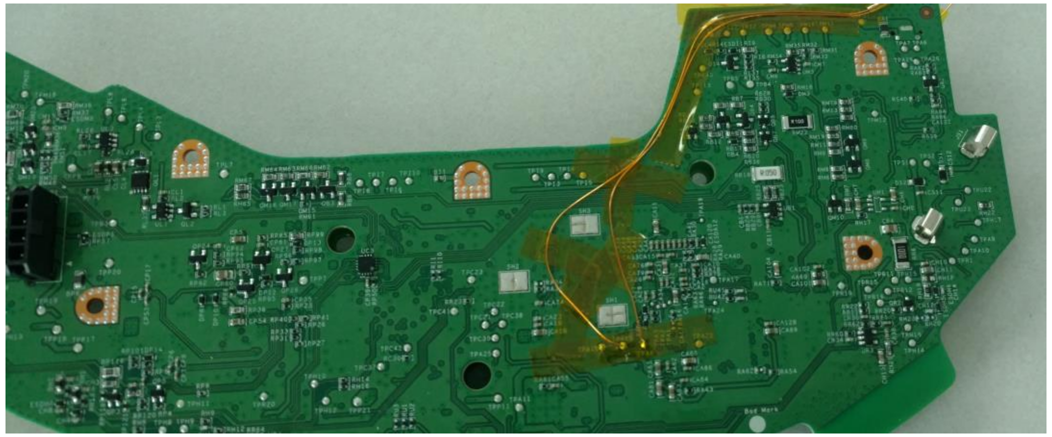
rockrobo.vacuum.v1 (End of 2016), roborock.vacuum.s5 (End of 2017)

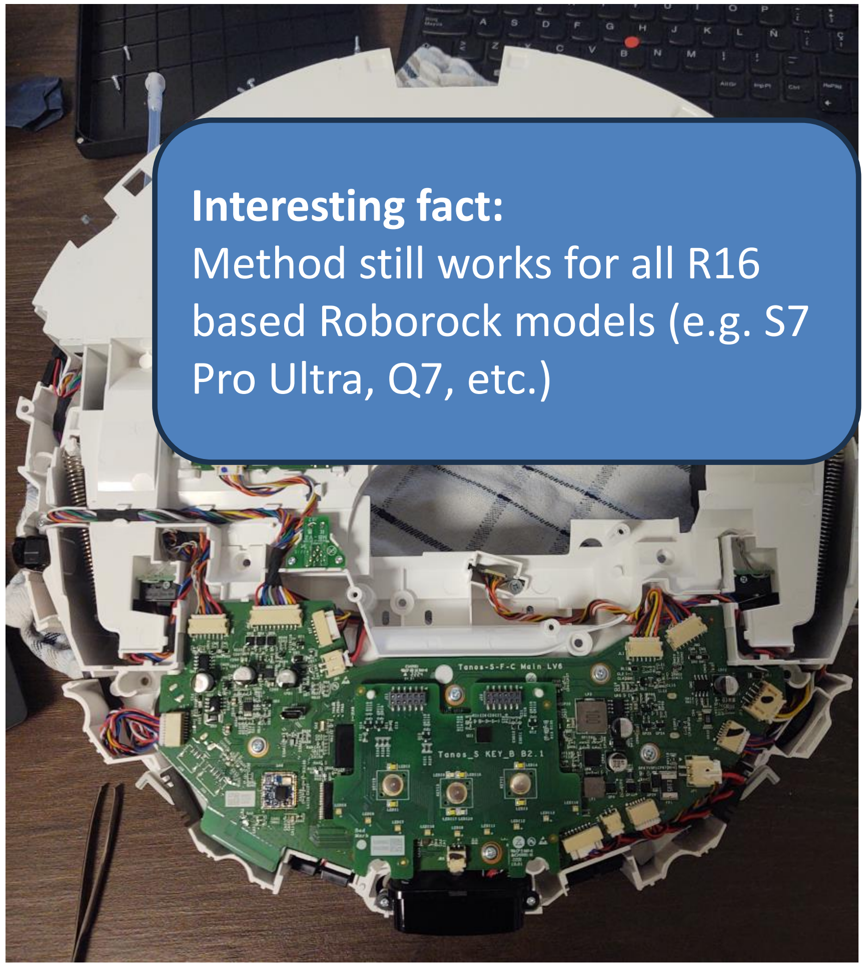
Roborock is unhappy

- Newer Roborock S5 firmware: local updates blocked
- With introduction of Roborock S6 (2019):
  - Signed firmware and voice packages
  - Each model uses different encryption keys
  - Signed configuration files to enforce region locks
  - However: Hardware remains mostly the same
- Disassembly of devices was required

Rooting methods for new models (2019)

- Works for Roborock S6, S5 Max, S7 and others
- Initial idea: Access to U-Boot shell via UART
- New approach:
  - push device into FEL Bootrom mode
  - patch filesystem
- Disadvantage: requires tear down of device





Interesting fact:
Method still works for all R16 based Roborock models (e.g. S7 Pro Ultra, Q7, etc.)

Roborock reacts again

- U-Boot gets locked down and shell is removed
- SecureBoot, SELinux, DM-verity is introduced
- New models use LUKS filesystem encryption
  - User data and Application partition is encrypted
  - Keys are stored in OPTEE/Trustzone
- Custom ELF binary signature checks in Kernel
  - Unsigned binaries cannot be executed

Hackers fight back (2021)

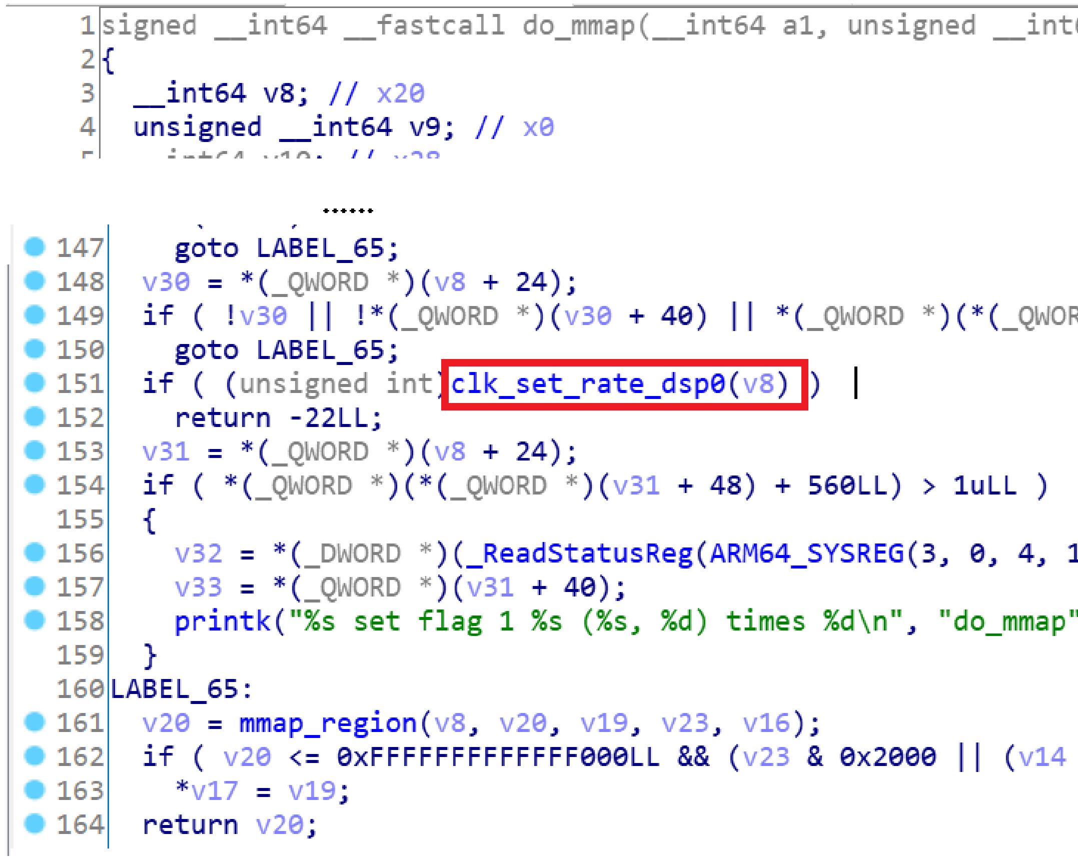
- Presented at DEFCON 31
- Method to bypass ELF binary verification and security
  - Idea: modification of configuration partition
  - Disadvantage: eMMC desoldering or ISP access required
- New Vendor: Dreame Technologies
  - Powerful devices with cameras
  - Easy rooting method without disassembly via USB
  - Problem: In some cases devices get soft-bricked

Roborocks reaction

- Email from Roborock CEO: „Thank you for the talk, our engineers watched the talk live and are fixing right now all vulnerabilities“
- All partitions, except system, are now encrypted
- ELF binary signature verification is obfuscated
- Custom code in libcurl to silently bypass DNS redirections
- Lots of obfuscation (XOR ftw!)

Roborocks reaction

- ELF signature checks in Kernel
  - Favorite hook: do_mmap
  - Result: seg-fault if signature invalid
  - Creative names for functions:
    - clk_set_rate_dsp0
    - clk_get_rate_dsp0_info
    - clk_set_rate_dsp_common
    - clk_load_new_signals
    - ex_mpi_ ….
    - And many more

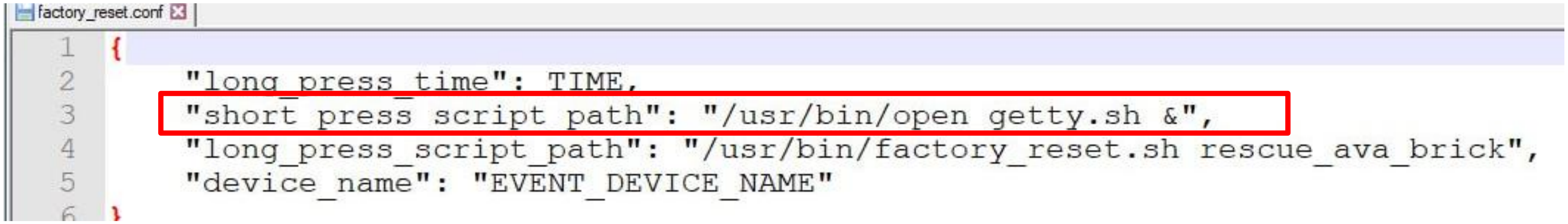


```
signed __int64 __fastcall do_mmap(__int64 a1, unsigned __int
{
  __int64 v8; // x20
  unsigned __int64 v9; // x0
```

```
......
    goto LABEL_65;
  v30 = *(_QWORD *)(v8 + 24);
  if ( !v30 || !*(_QWORD *)(v30 + 40) || *(_QWORD *)(*(_QWOR
    goto LABEL_65;
  if ( (unsigned int)clk_set_rate_dsp0(v8) )
    return -22LL;
  v31 = *(_QWORD *)(v8 + 24);
  if ( *(_QWORD *)(*(_QWORD *)(v31 + 48) + 560LL) > 1uLL )
  {
    v32 = *(_DWORD *)(_ReadStatusReg(ARM64_SYSREG(3, 0, 4, 1
    v33 = *(_QWORD *)(v31 + 40);
    printk("%s set flag 1 %s (%s, %d) times %d\n", "do_mmap"
  }
LABEL_65:
  v20 = mmap_region(v8, v20, v19, v23, v16);
  if ( v20 <= 0xFFFFFFFFFFFFF000LL && (v23 & 0x2000 || (v14
    *v17 = v19;
  return v20;
```

Dreame starts to panic (2021)

- After release of the talk
  - Lots of changes in firmwares
  - UART login shell gets removed, U-Boot shell is patched
  - Some changes seem to brick robots via OTA
  - Fun fact: patches reveal a simpler rooting method


factory_reset.conf

```
{
    "long_press_time": TIME,
    "short_press_script_path": "/usr/bin/open_getty.sh &",
    "long_press_script_path": "/usr/bin/factory_reset.sh rescue_ava_brick",
    "device_name": "EVENT_DEVICE_NAME"
}
```

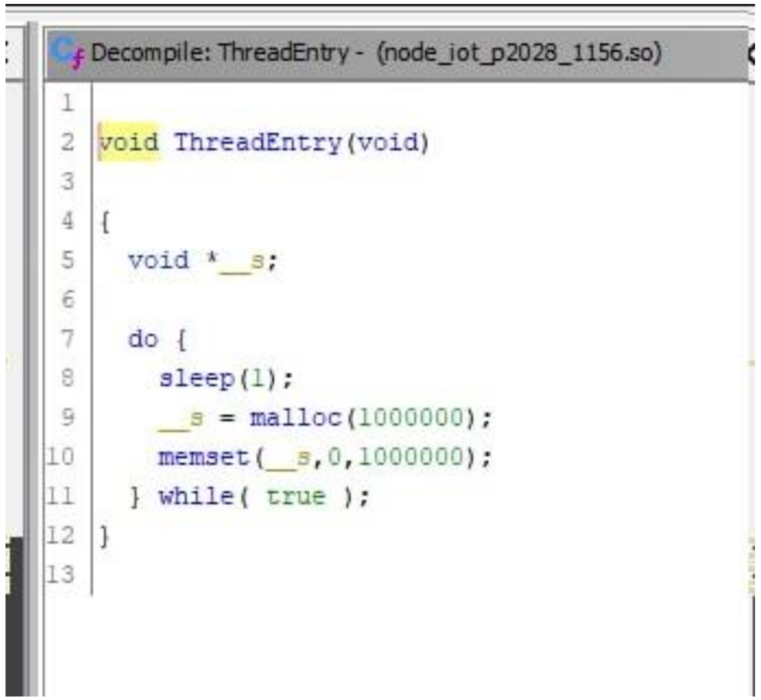
Dreame panics even more (2022)

- New countermeasures are introduced
  - SecureBoot is enforced and eFuses are set
  - System partition is signed and verified by U-Boot
  - Kernel is paired with a specific system partition
  - „Judge“ countermeasure is introduced into the software

Dreame countermeasure „Judge“

- Introduced in all new firmwares and models in 2022
- Rooted firmwares started to randomly crash

- Expected SHA256 of the rootfs is baked into the Kernel
- Main process executes a new function on startup:
  - Compute hash of system partition
  - If the hash doesn't match it spawns a new thread
  - Thread exclusively contains an endless malloc loop

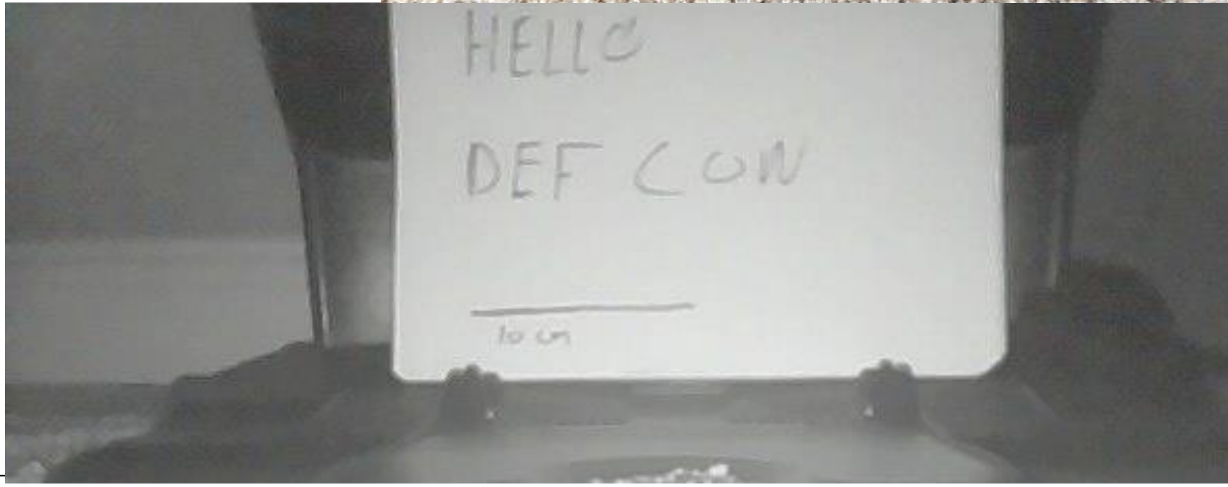


```

void ThreadEntry(void)

{
  void *__s;

  do {
    sleep(1);
    __s = malloc(1000000);
    memset(__s,0,1000000);
  } while( true );
}

```

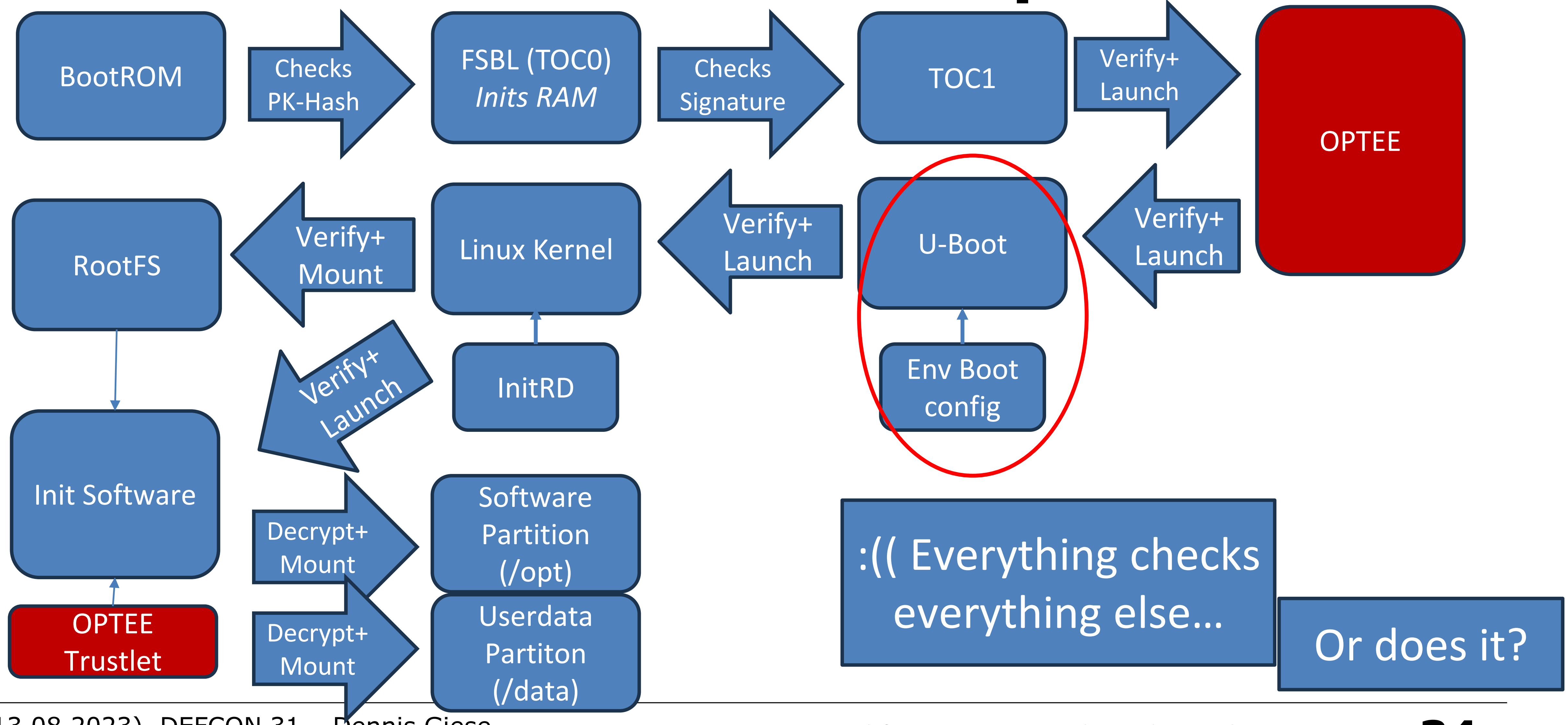
CURRENT STATE OF ROBOT SECURITY

Example device: Roborock S8 Pro Ultra

- Current flagship model
- Allwinner MR813 SoC + 2x STM32 MCUs
- 512MByte/ 1Gbyte RAM
- 4GByte Flash
- 2 Cameras, LIDAR sensor, 2 Linelasers
- Security:
  - SecureBoot
  - DM-Verity protected rootfs
  - LUKS encrypted partitions
  - SELinux, ELF signatures



Roborock S8 Pro Ultra boot process



:((Everything checks everything else…

Or does it?

Simplified version, technical details omitted

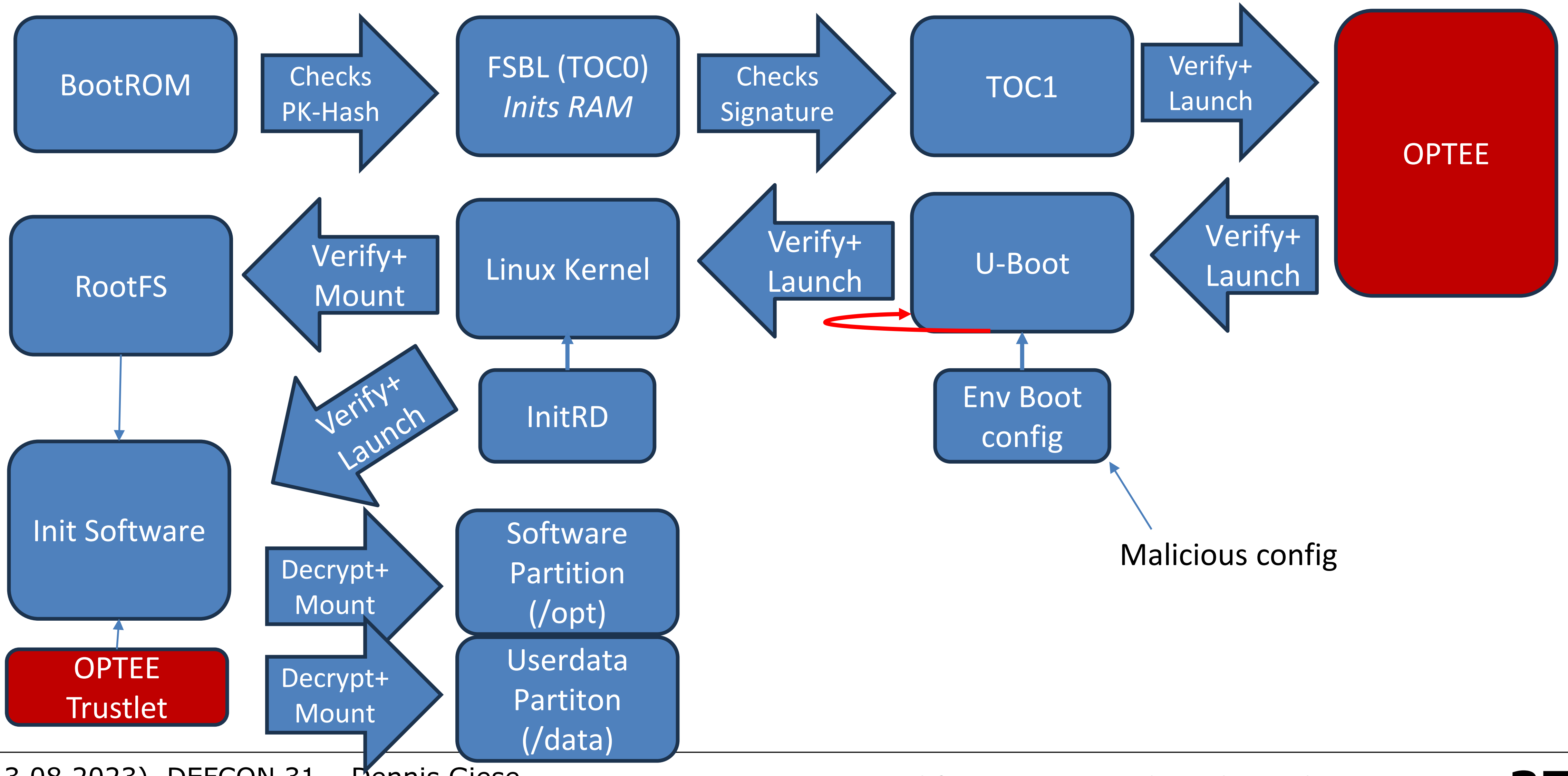
U-Boot Bootloader

- Defacto default bootloader for embedded devices
- Powerful software
- Sets up some hardware
- Sets command line arguments for Kernel
- Verifies Kernel and boots it
- Uses “env” Environment to configure itself
  - Env is used for A/B booting and OTA updates
  - Powerful command set
  - Allows loading of partitions, accessing memory, changing memory

Idea

- Question: Can we use U-Boot to modify itself?
- Theory: Use memory read+write commands to overwrite instructions
  - Math:
    - Figure out where the signature verification function is
    - Figure out U-Boot relocation offset
    - Calculate place of function in memory
- Patch: “mw.w 7ff2021a e016”
  - writes e016 at 0x7ff2021a
  - Disables verification

Roborock S8 Pro Ultra hacked boot



Simplified version, technical details omitted

What did we archive?

- Bypass the SecureBoot process ☑
- Modification of the Kernel ☑
  - Removing DM-Verity checks
  - Removing ELF binary verification
  - Disabling SELinux
- Modification of the root filesystem ☑
- Install custom software and get SSH access ☑

<u>Approach is not limited on Allwinner SOCs, but works for many other devices (other robots, smart speakers, media devices, etc)</u>

HOW CAN WE MODIFY ENV?

We have a problem

- Without root access, we cannot modify the “env” partition ☹
- We also cannot modify the rootfs
- We do not want to desolder the flash
- Good news: We have found solutions for each model
  - Dreame L10, Z10, W10 and other “p” models: USB stick method
  - Dreame L10sU, D10s Pro/Plus and other “r” models: FEL root
  - Roborock G10s, S8: scary FEL root

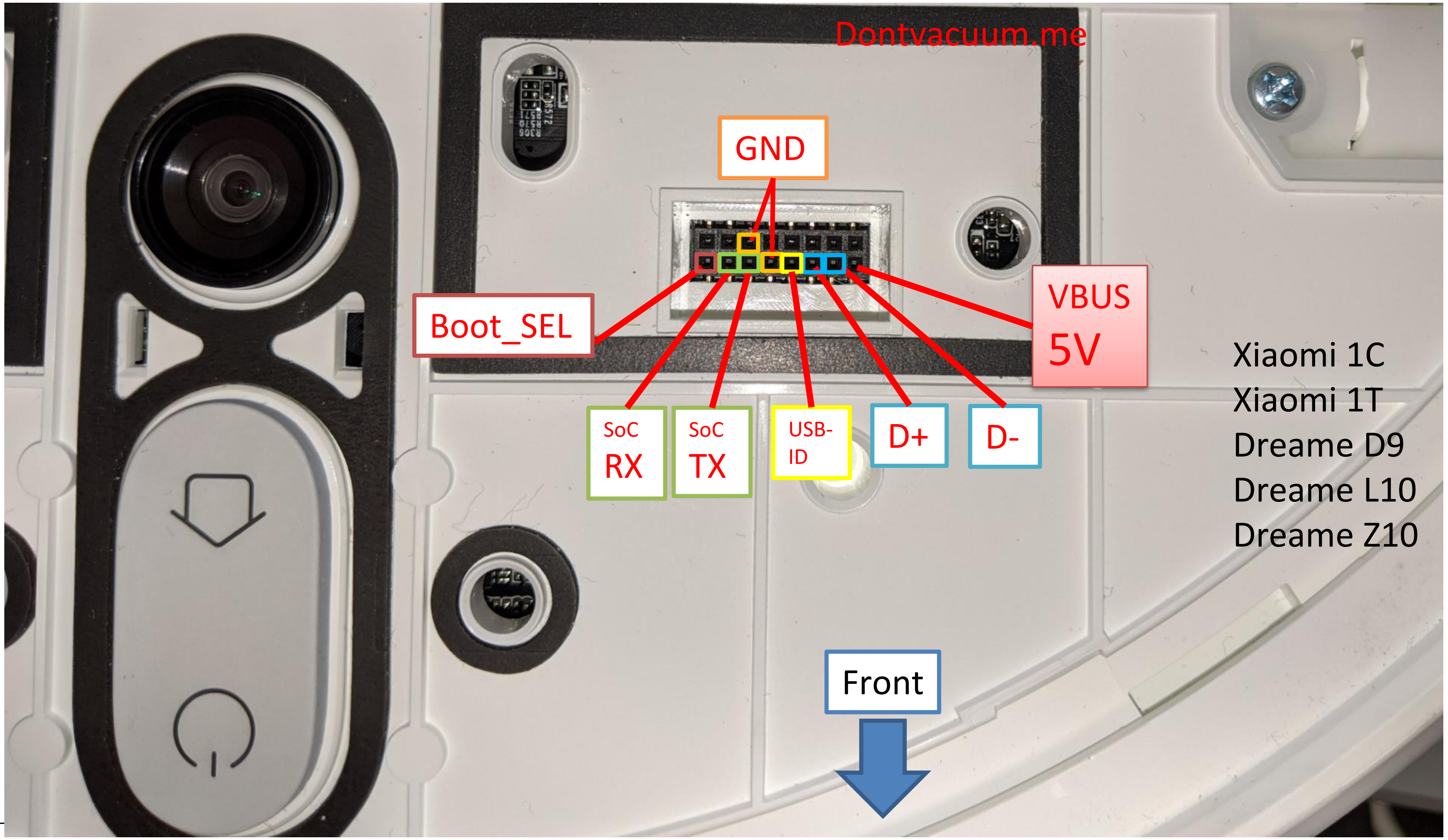
Recap: Dreame Debug pinout

- Debug interface
  - 2x8 pins
  - 2mm pitch size

Warning:
2mm pitch size is way smaller than the usual 2.54 mm

Warning:
Make sure you connect to the correct pins!

Warning:
Double-check the orientation
Of the pinout!



Xiaomi 1C
Xiaomi 1T
Dreame D9
Dreame L10
Dreame Z10

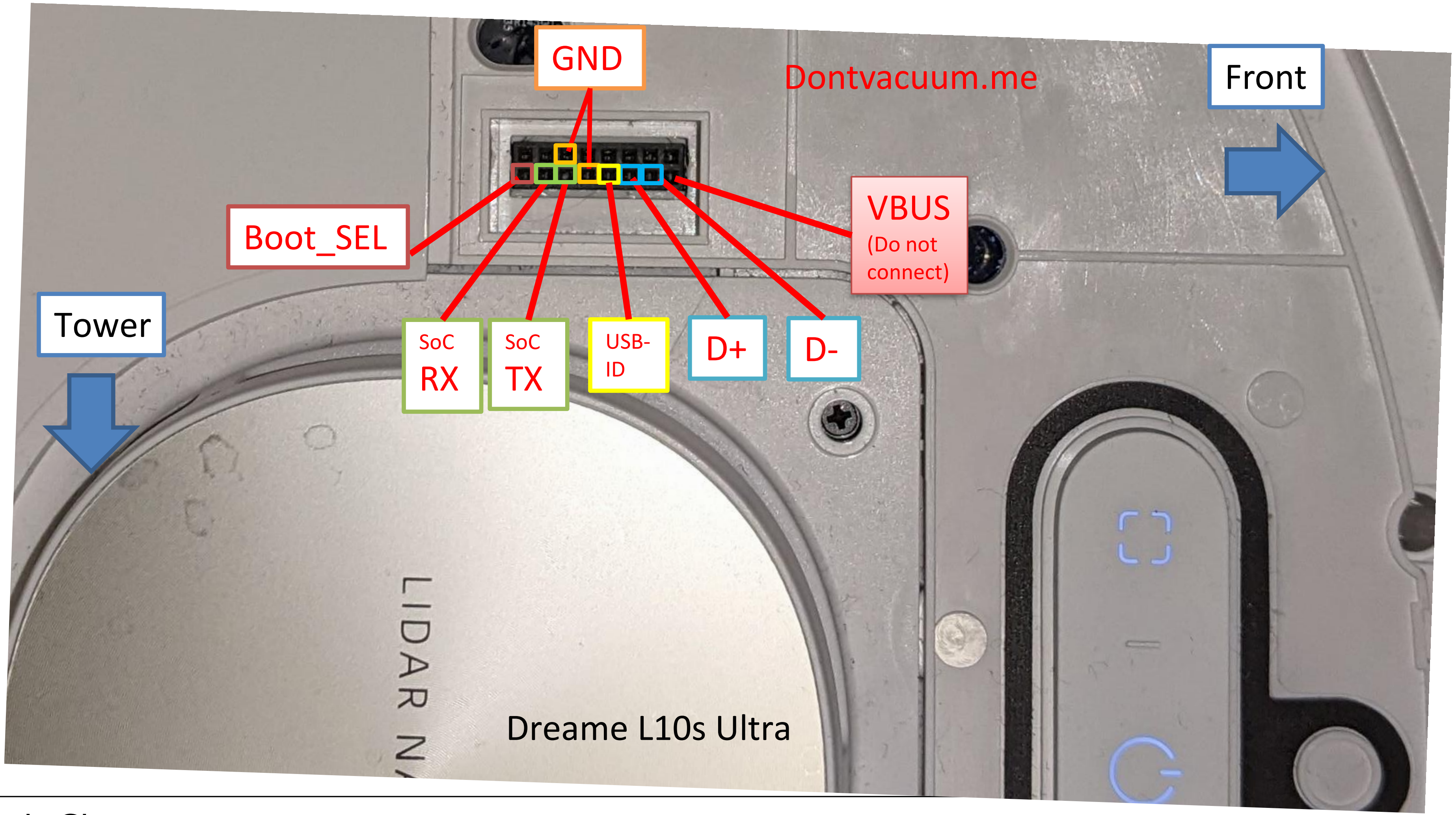
Debug pinout for Dreame “r” models

- Debug interface
  - 2x8 pins
  - 2mm pitch size

Warning:
2mm pitch size is way smaller than the usual 2.54 mm

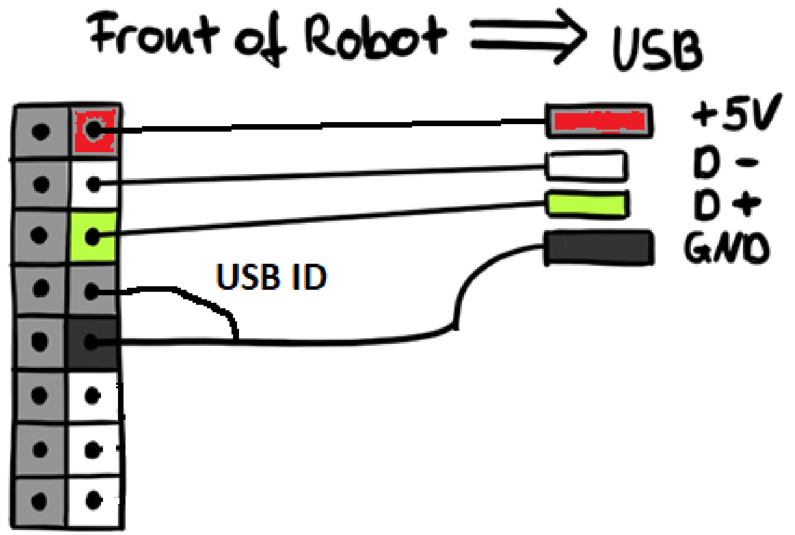
Warning:
Make sure you connect to the correct pins!

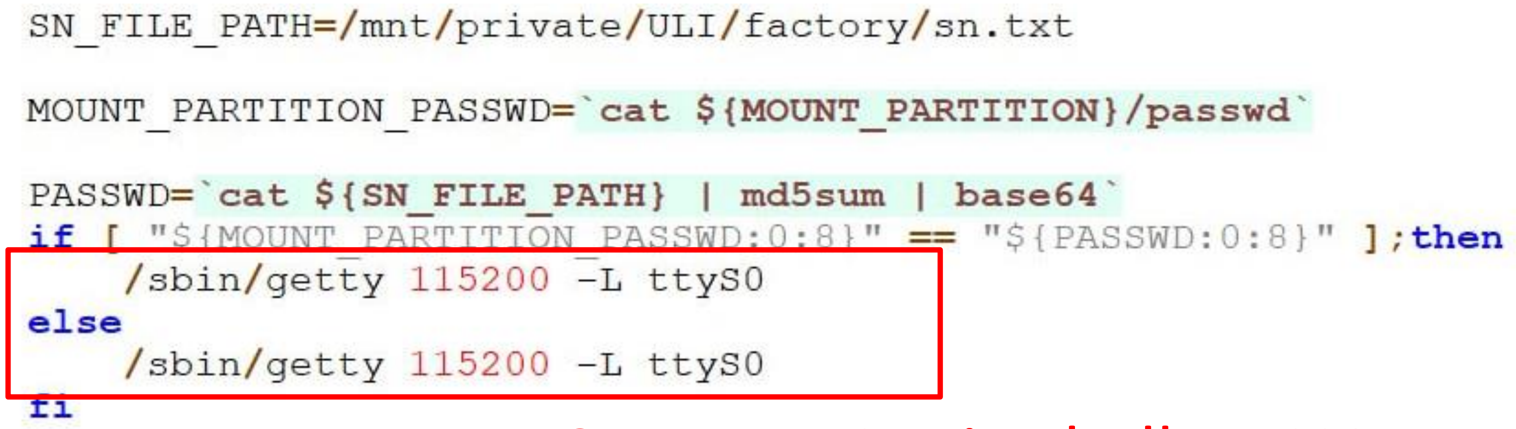
Warning:
Double-check the orientation
Of the pinout!



USB Stick method for Dreame

- Dreame left a backdoor in all „p“ models (e.g Z10, L10, etc)
  - Script gets executed when an USB stick is attached





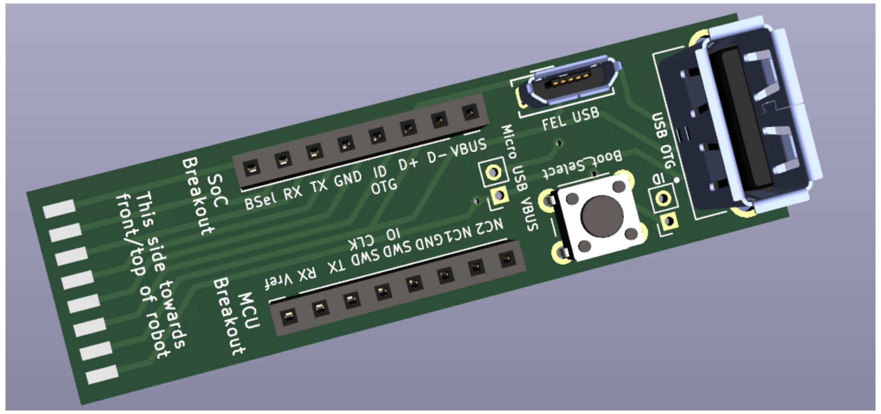
```
SN_FILE_PATH=/mnt/private/ULI/factory/sn.txt

MOUNT_PARTITION_PASSWD=`cat ${MOUNT_PARTITION}/passwd`

PASSWD=`cat ${SN_FILE_PATH} | md5sum | base64`
if [ "${MOUNT_PARTITION_PASSWD:0:8}" == "${PASSWD:0:8}" ];then
    /sbin/getty 115200 -L ttyS0
else
    /sbin/getty 115200 -L ttyS0
fi
```

Creates a Login shell on UART

- After login, we can patch „env“ and install custom firmware
- Sad news: „r“ models check a signed file ☹

FEL root

- Solution for Dreame „r" models
- Boot robot in Bootloader mode
  - Short BOOT_SEL to GND
- Usage of custom Phoenixsuit image
  - Does not actually flash any firmware
  - Gives us a fastboot interface via USB
  - Automatically patch bootloader
  - Allows us to transfer rootfs and kernel images onto the robot
- Howto: https://builder.dontvacuum.me/nextgen/dreame_gen3.pdf

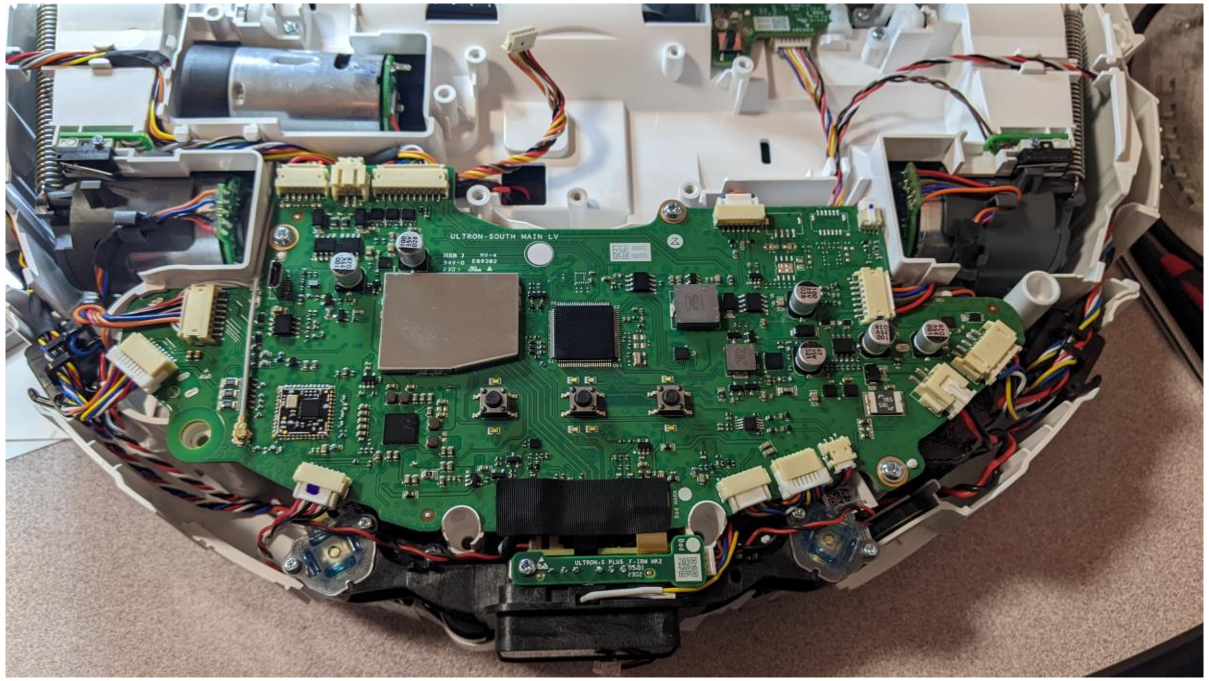
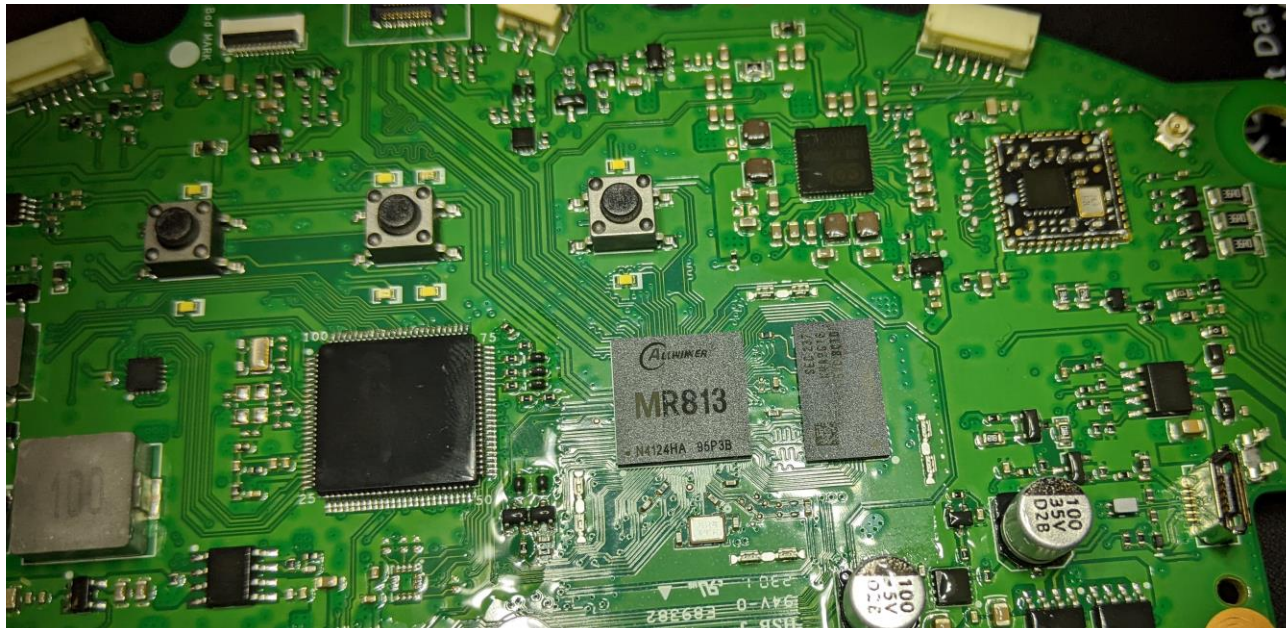


This side towards front/top of robot
SoC Breakout
BSel RX TX GND ID D+ D- VBUS
OTG
FEL USB
Micro USB VBUS
Boot_Select
USB OTG
MCU Breakout
RX Vref XX TX SWD SWD GND NC1 NC2
IO CLK

Get the Gerber files for the custom PCB here: https://github.com/Hypfer/valetudo-dreameadapter

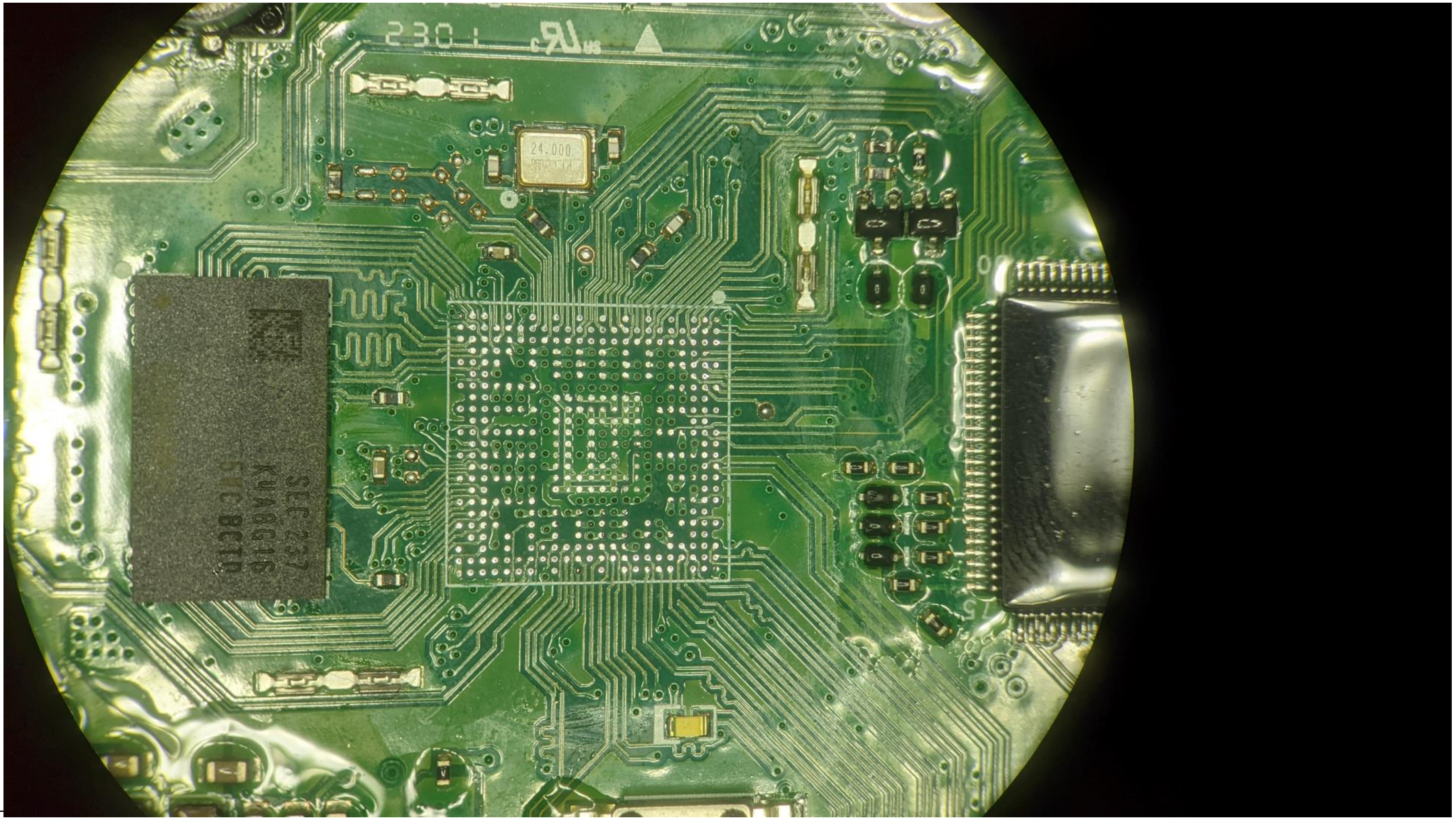
Scary FEL root

- Problem with Roborock: we have no exposed debug pins
  - Only USB is available without teardown
- First approach: complete teardown

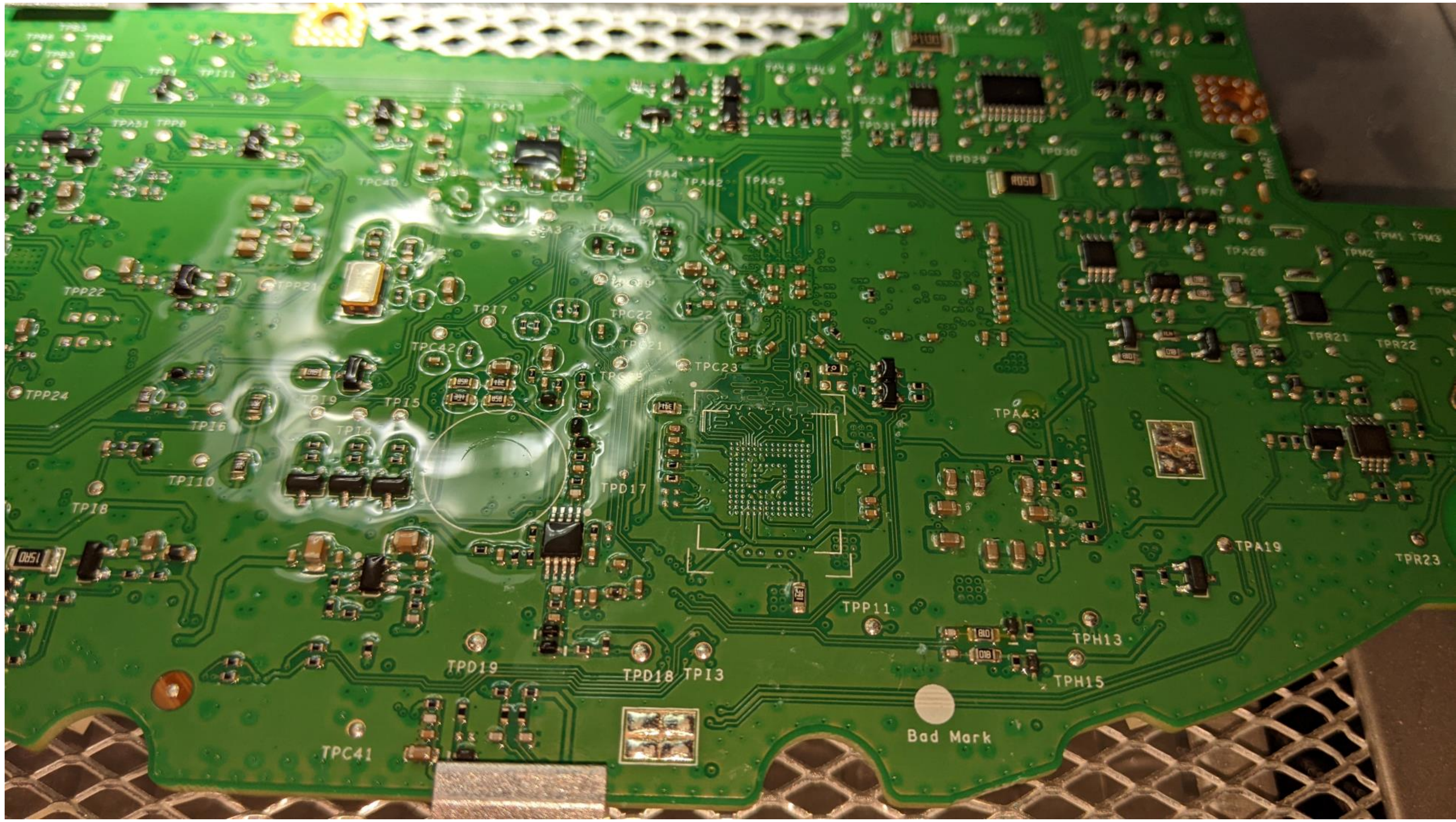
Where are the debug pins?

Removal of the SOC to track traces according to the datasheet

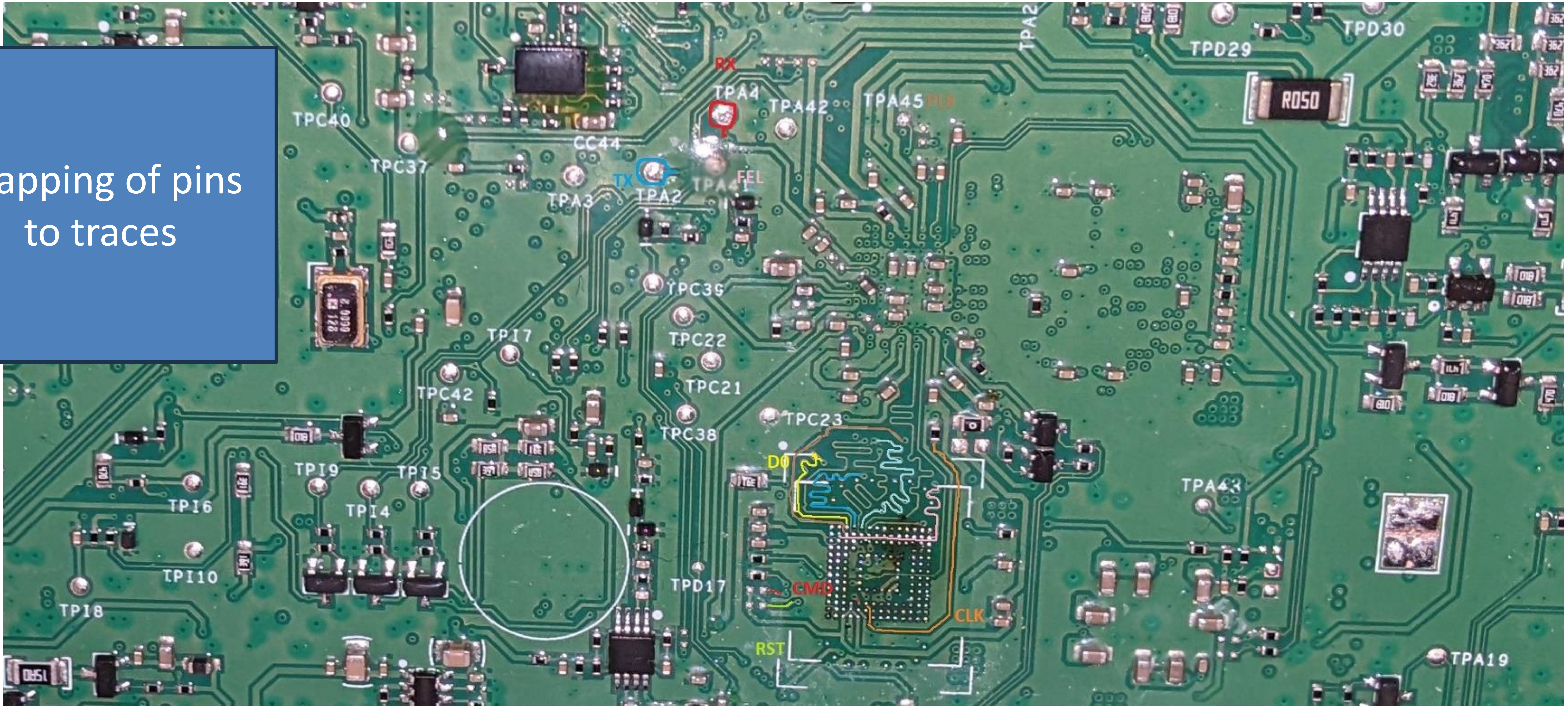


Where are the debug pins?

Removal of the eMMC Flash to track traces

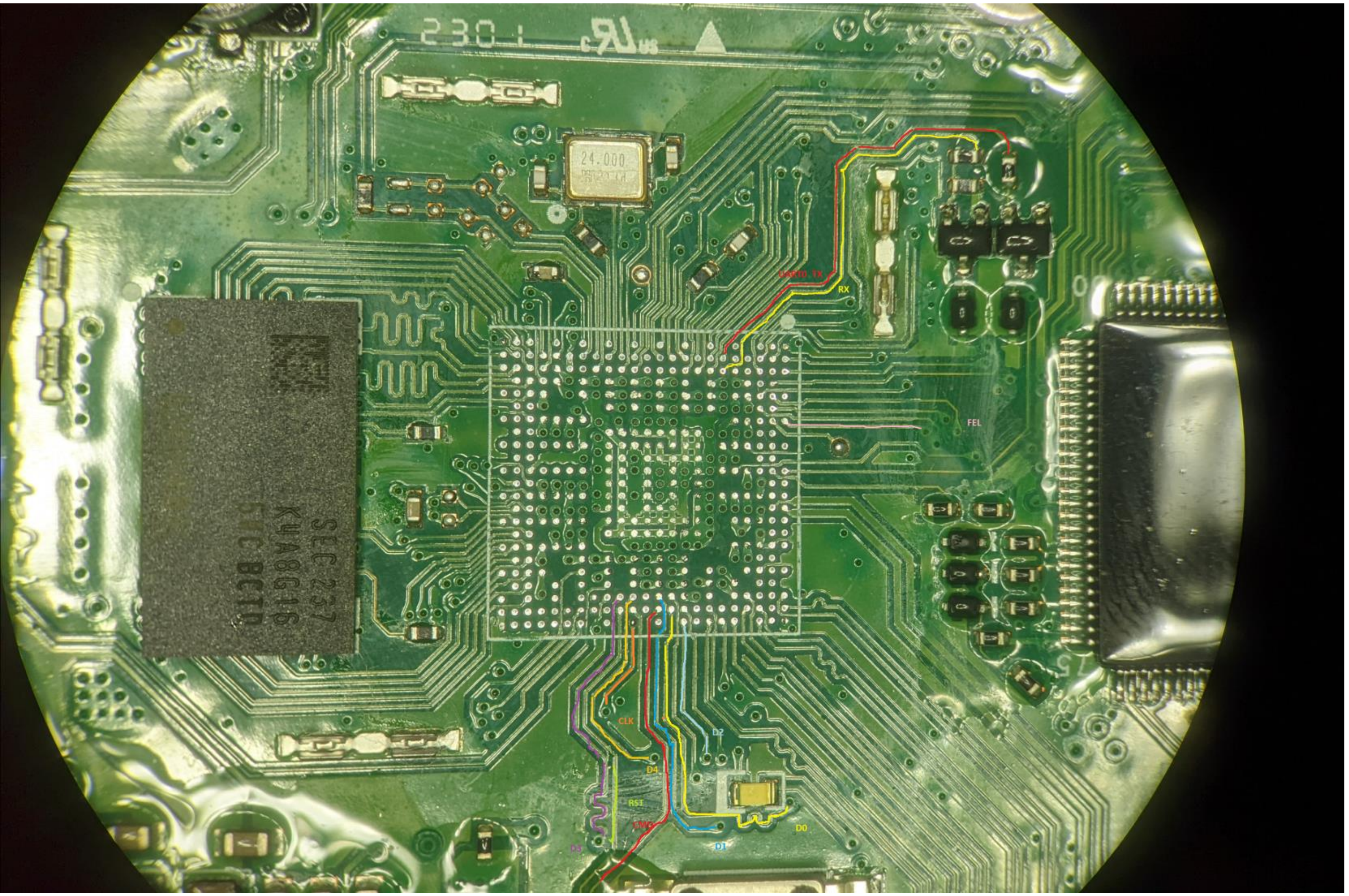


Where are the debug pins?



Mapping of pins to traces

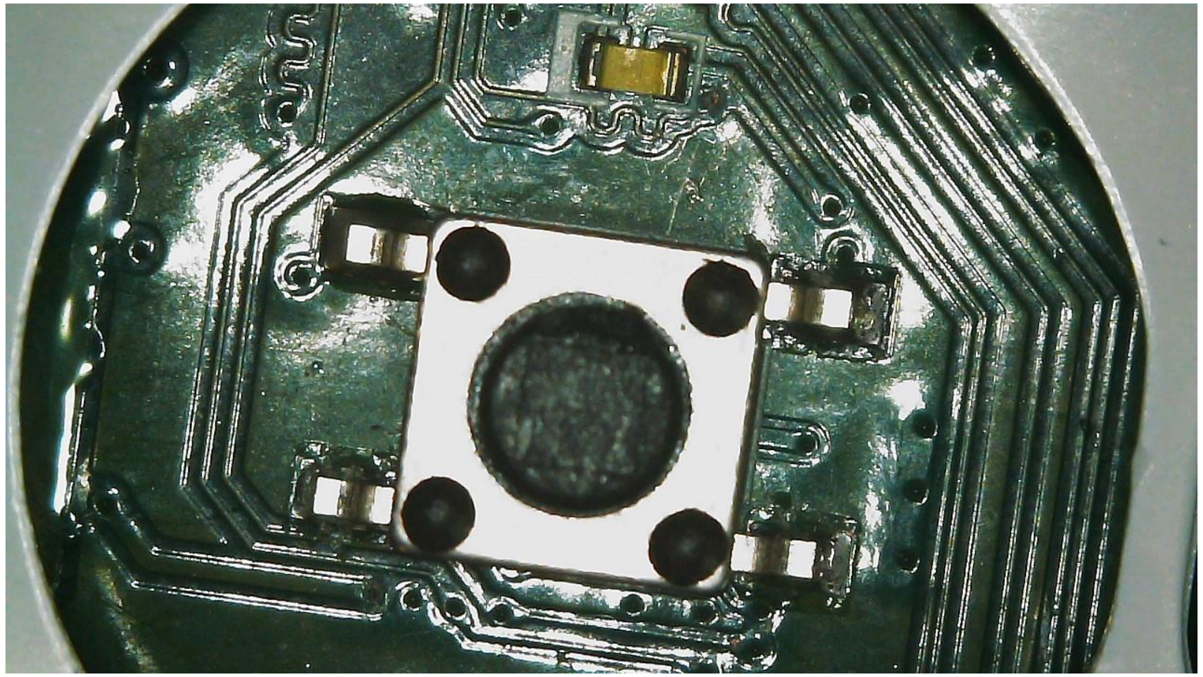
Where are the debug pins?

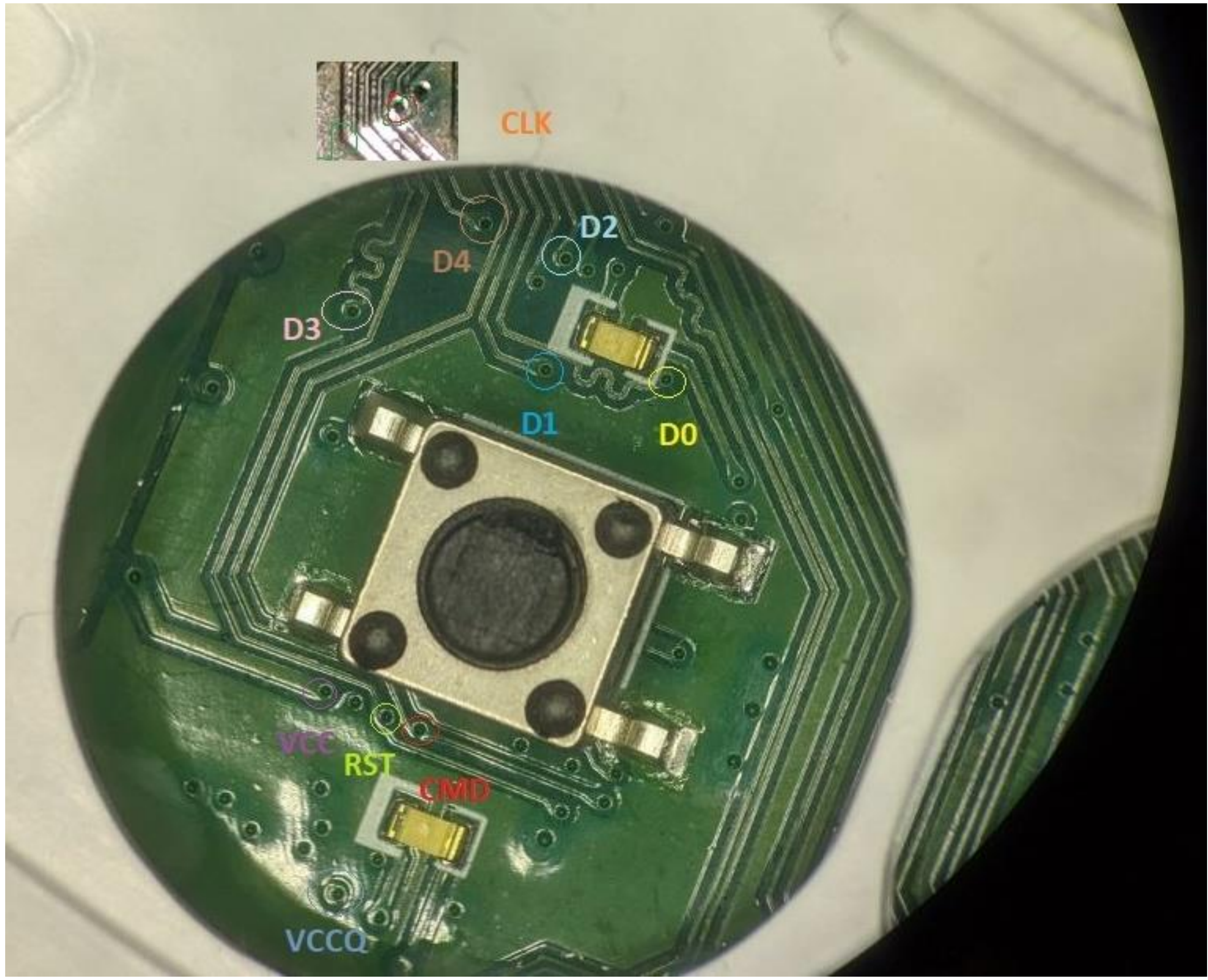


Connecting the traces of the back side of the PCB to the front side

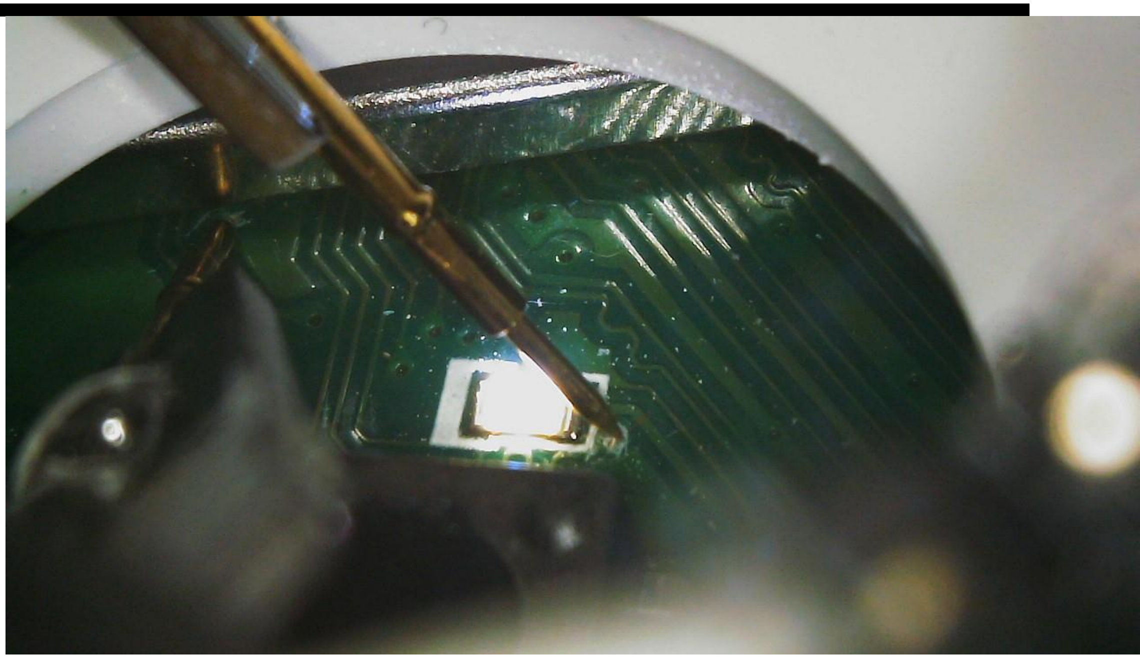
Suprising finding

All eMMC pins are accessible from the holes of the buttons





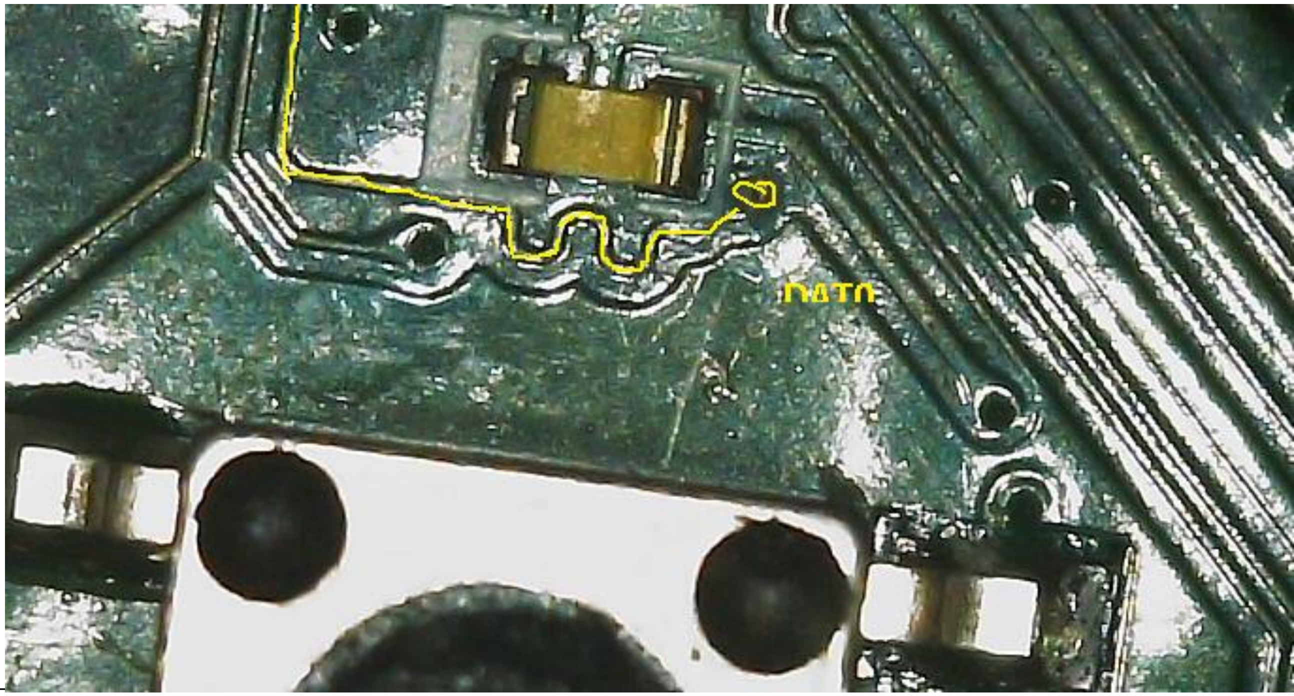
Scary FEL root



- Allows root without tear down
  - accessible under rubber cover
  - Applies to S8 and S8 Pro Ultra
- Boot robot in Bootloader mode
  - Short DAT0 pin to GND
- Usage of custom Phoenixsuit image
  - Same approach as for Dreame L10s Ultra („R“ models)
- Howto: https://builder.dontvacuum.me/nextgen/roborock_s8.pdf

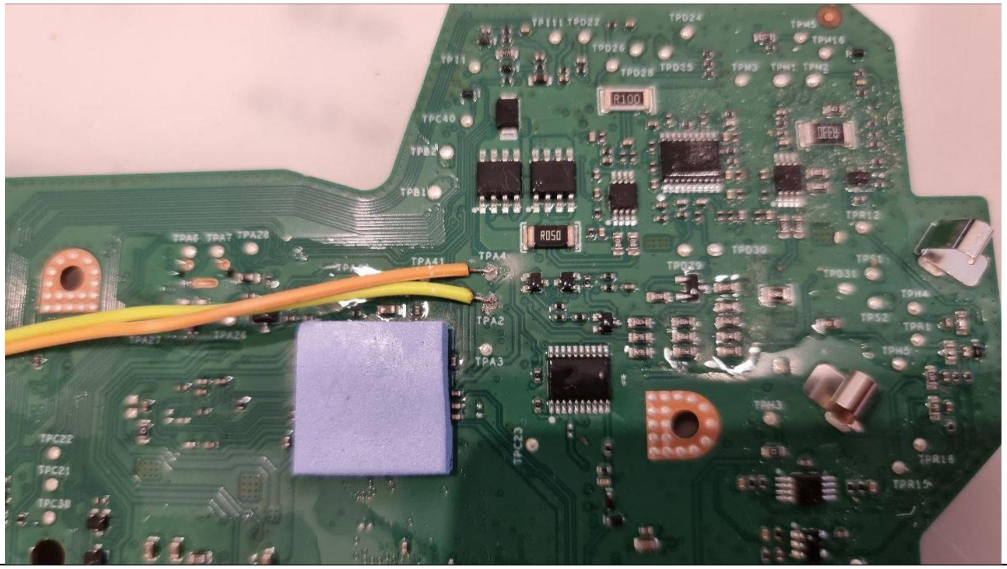
Why is it scary?

- Requires removal of solder mask to access the copper trace
- Do not do this method if you are uncomfortable
- Alternate: Teardown and using FEL pin on the back side of the PCB
- If unsure: ask other users



Root of other Roborock vacuums

- The same process works for most newer Roborock vacuums*
- Not all devices have reachable flash or debug pins
  - Sadly, they require a tear down to access these ☹
- Alternative way for FEL mode (via UART): hold “2” while powering on
- Check robotinfo.dev for pinouts
- Ask in the community for help



*Exceptions apply

WHAT TO DO WITH ROOT ACCESS

Now what?

- Secure boot is defeated
- We can run custom software
- But what software?
- Do we have to build our own OS, SLAM, Navigation, AI Models?

- Idea:

  Can we just disconnect the cloud and keep the vendor cleaning logic?

How to get rid of the cloud

- Block cloud traffic?
  - ✔ Robot still works
  - ✕ No live maps
  - ✕ No advanced features (Map editing, per-room-cleaning, etc.)
  - Why did we even root for that?

- Redirect cloud traffic?
  - Extract secrets from the robot
  - Point it to a cloud emulation

Replacing the cloud

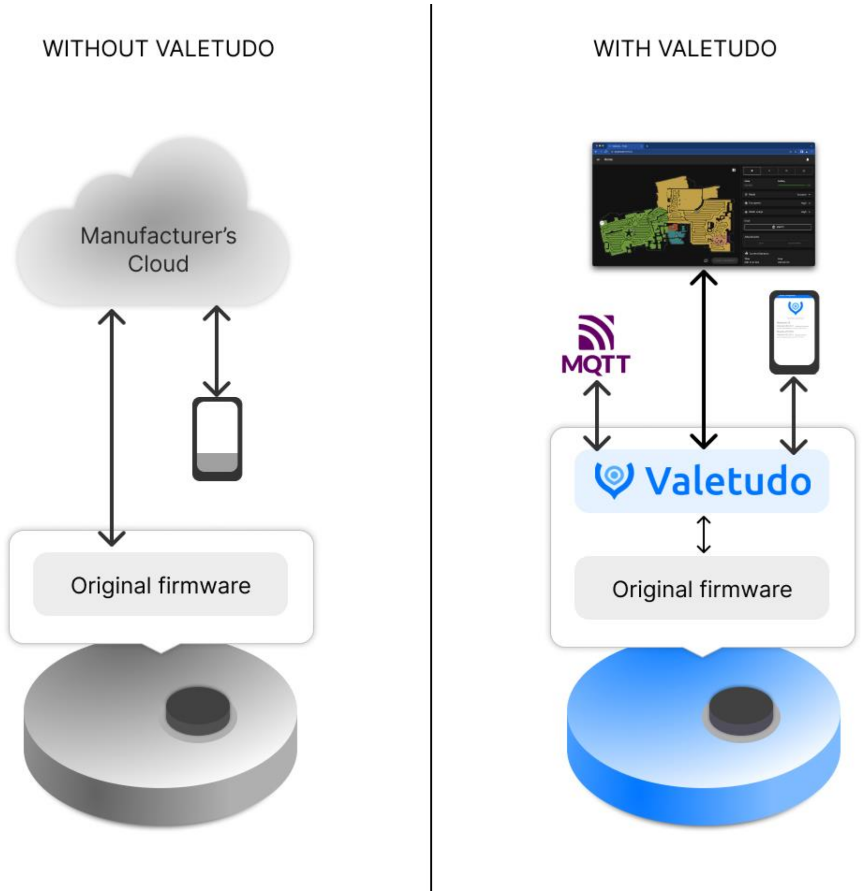
- Initial approach: Redirect DNS
  - miio_client 3.3.x requires cloud IPs to be part of a different subnet
  - Solution:
    - Use /etc/hosts to point cloud to different IP
    - Use IPTables to reroute that traffic back to localhost
- Xiaomi reacts:
  - miio_client 3.5.x introduces "http_dns"
  - Fetches cloud endpoints via http from hardcoded IPs
  - Solution: Replace their hardcoded IPs with ours ☺

Valetudo

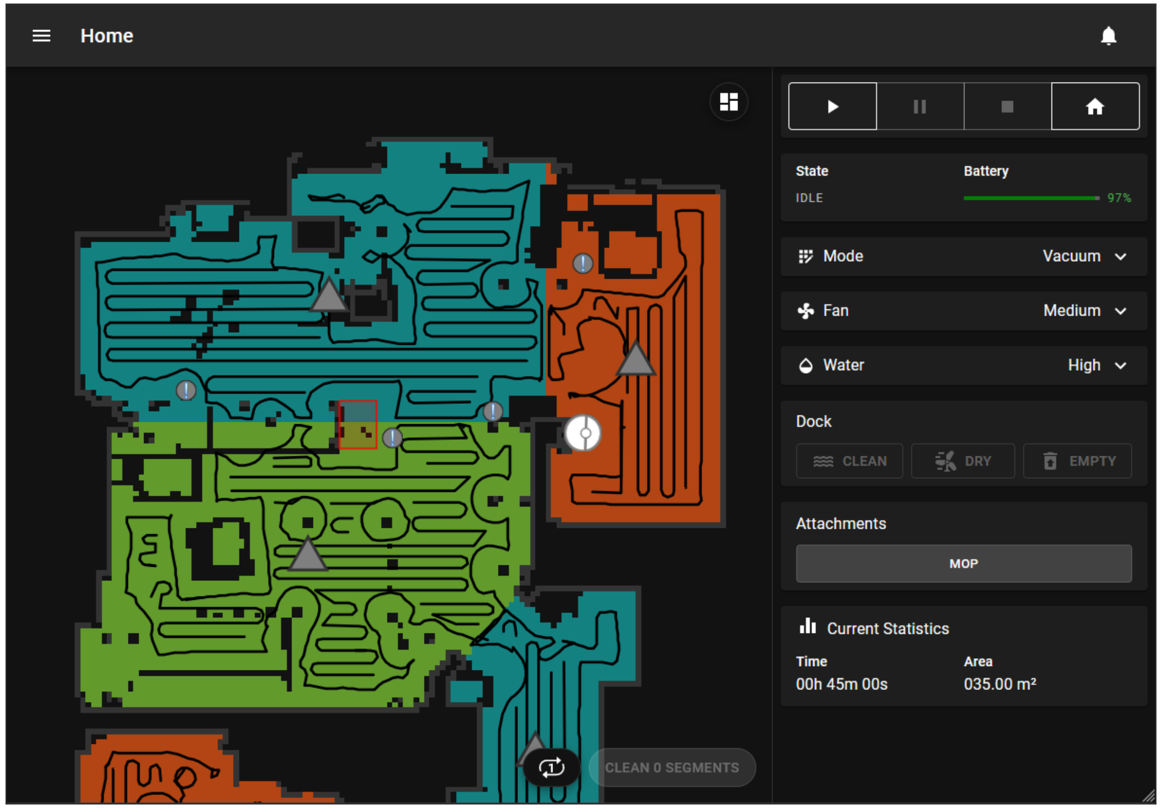
URL: https://valetudo.cloud

GitHub: https://github.com/Hypfer/Valetudo

- Completely replaces the cloud and vendor apps
- Runs on the robot
- Features:
  - Full Robot Control
    - Live Maps
    - Map Editing
    - Robot Configuration
  - Responsive Webinterface
    - Works best on both Mobile *and* Desktop
  - REST-API
  - MQTT-Connectivity
  - Embedded JavaScript :-)



Screenshots



Home

State: IDLE

Battery: 97%

Mode: Vacuum

Fan: Medium

Water: High

Dock: CLEAN, DRY, EMPTY

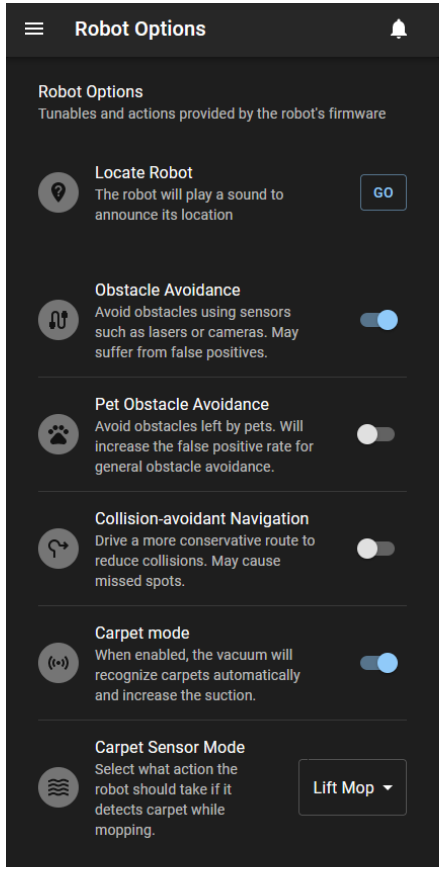
Attachments: MOP

Current Statistics

Time: 00h 45m 00s

Area: 035.00 m²

CLEAN 0 SEGMENTS



Robot Options

Robot Options
Tunables and actions provided by the robot's firmware

Locate Robot
The robot will play a sound to announce its location
GO

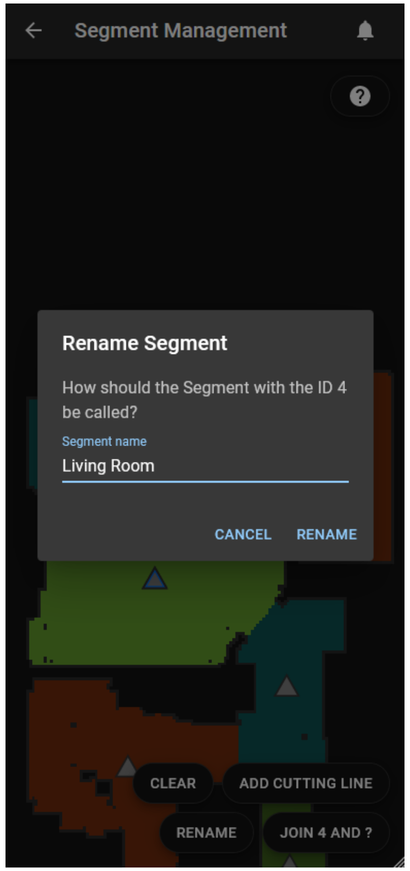
Obstacle Avoidance
Avoid obstacles using sensors such as lasers or cameras. May suffer from false positives.

Pet Obstacle Avoidance
Avoid obstacles left by pets. Will increase the false positive rate for general obstacle avoidance.

Collision-avoidant Navigation
Drive a more conservative route to reduce collisions. May cause missed spots.

Carpet mode
When enabled, the vacuum will recognize carpets automatically and increase the suction.

Carpet Sensor Mode
Select what action the robot should take if it detects carpet while mopping.
Lift Mop



Segment Management

Rename Segment

How should the Segment with the ID 4 be called?

Segment name: Living Room

CANCEL RENAME

CLEAR ADD CUTTING LINE RENAME JOIN 4 AND ?

Dustbuilder

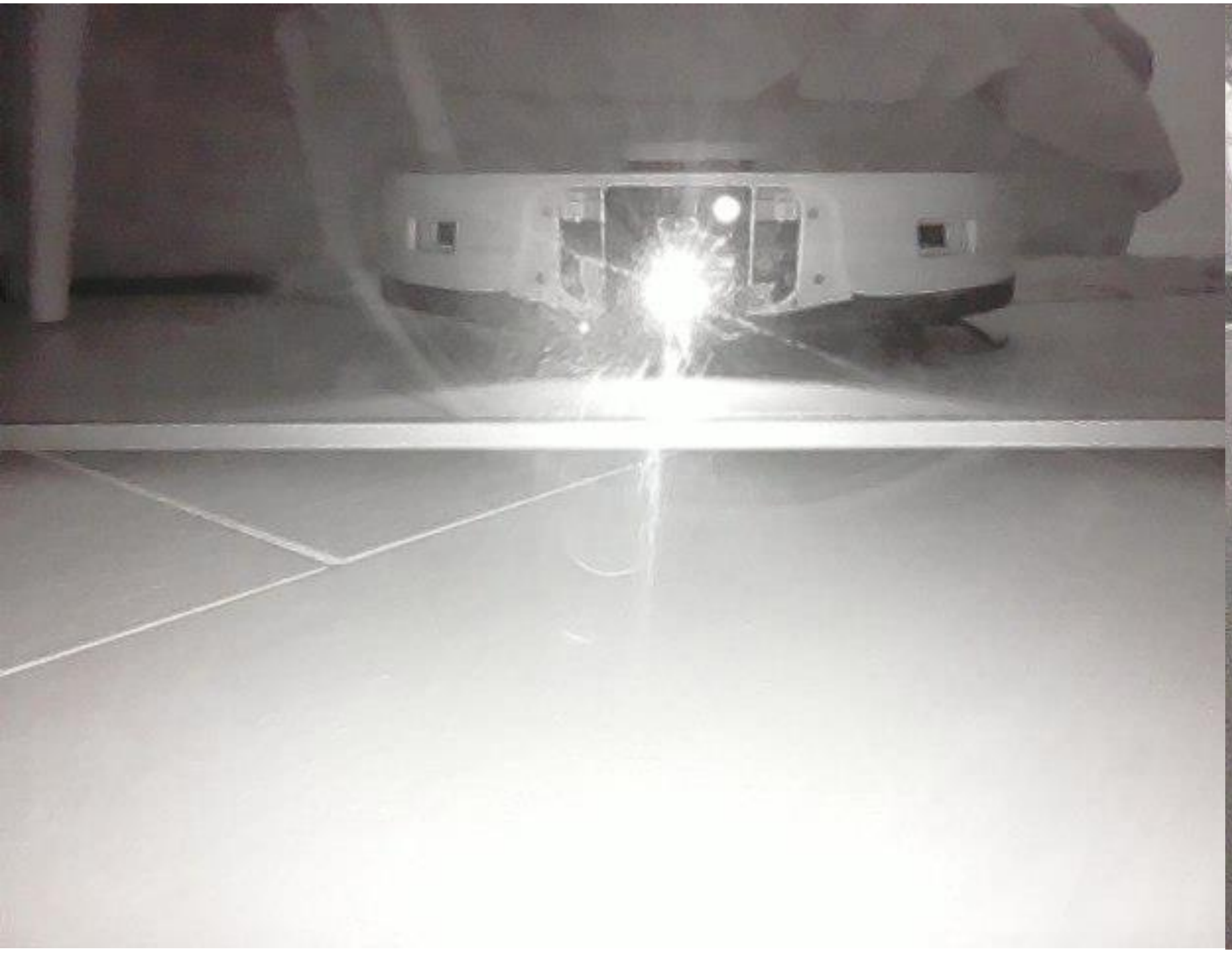
- Website for building your own custom robot firmwares
  - Reproducible builds
  - Easy to use
  - Works for Dreame, Roborock and Viomi
- Alternative to local building
  - All tools are published on GitHub
- URL: https://builder.dontvacuum.me/

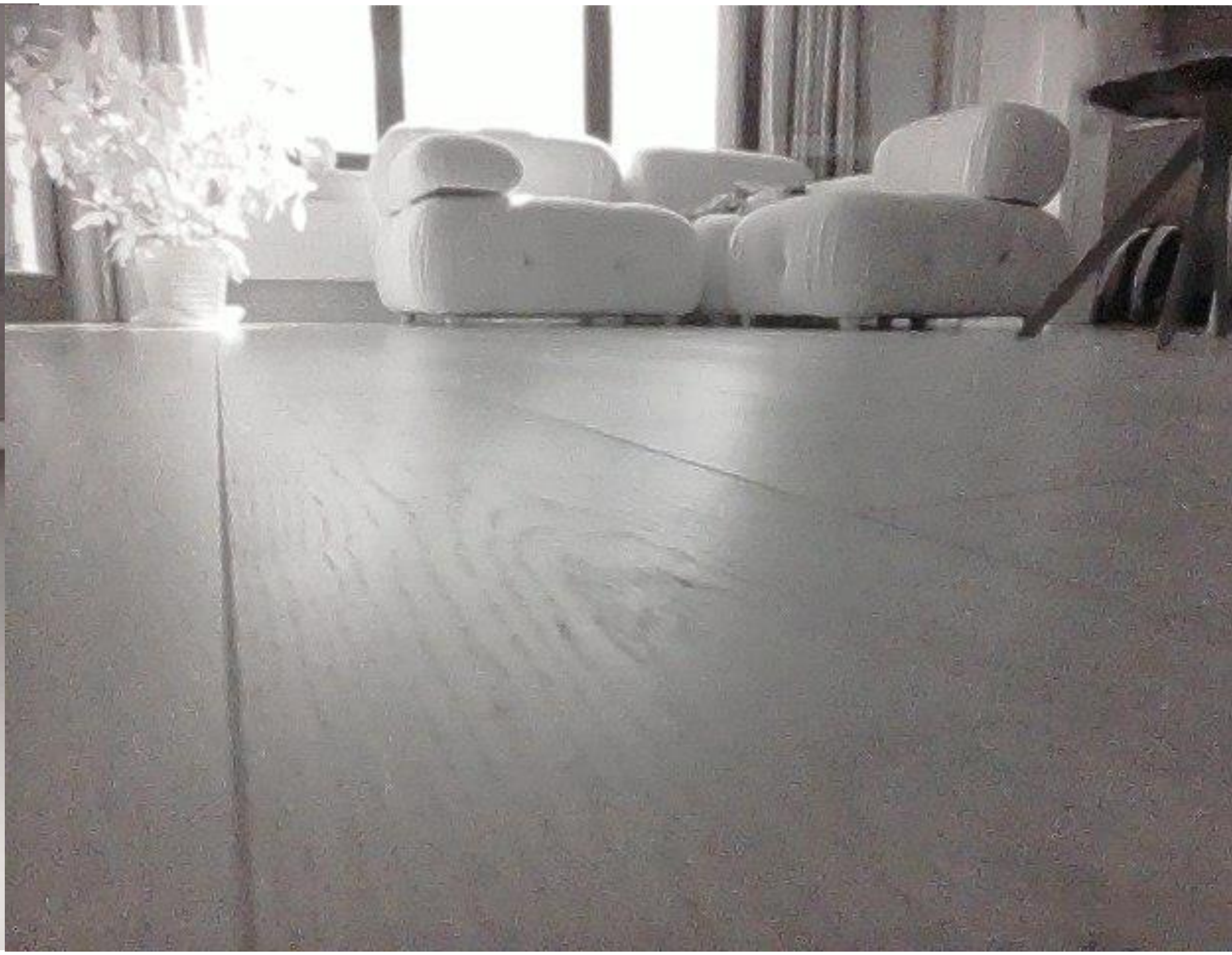
OTHER INTERESTING FINDINGS

Let’s have some fun: camera access

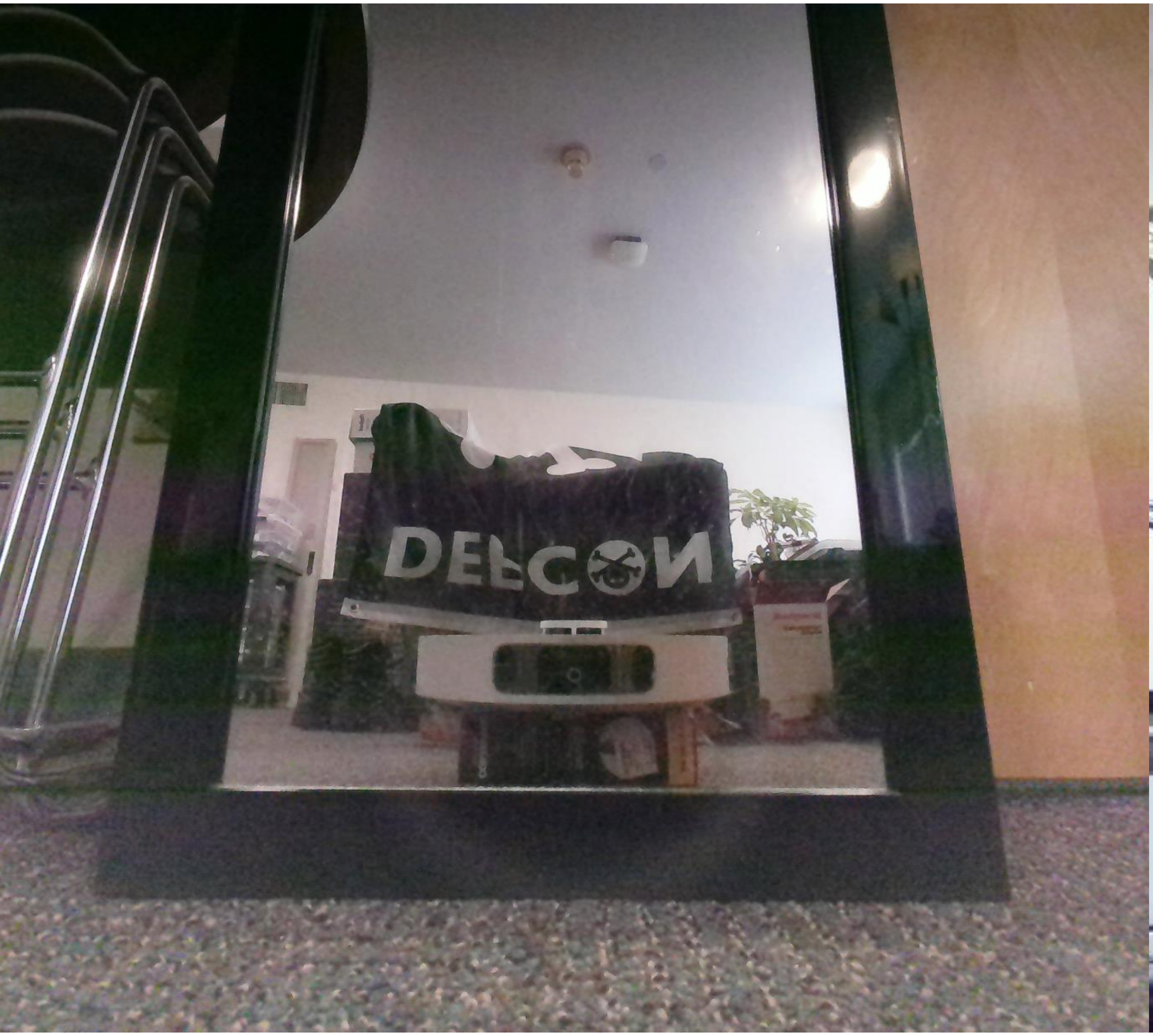
- All devices use the Video4Linux subsystem
- Cameras are accessible via /dev/video0, /dev/video1, etc.
- Vendors left tools on robot like “camerademo”

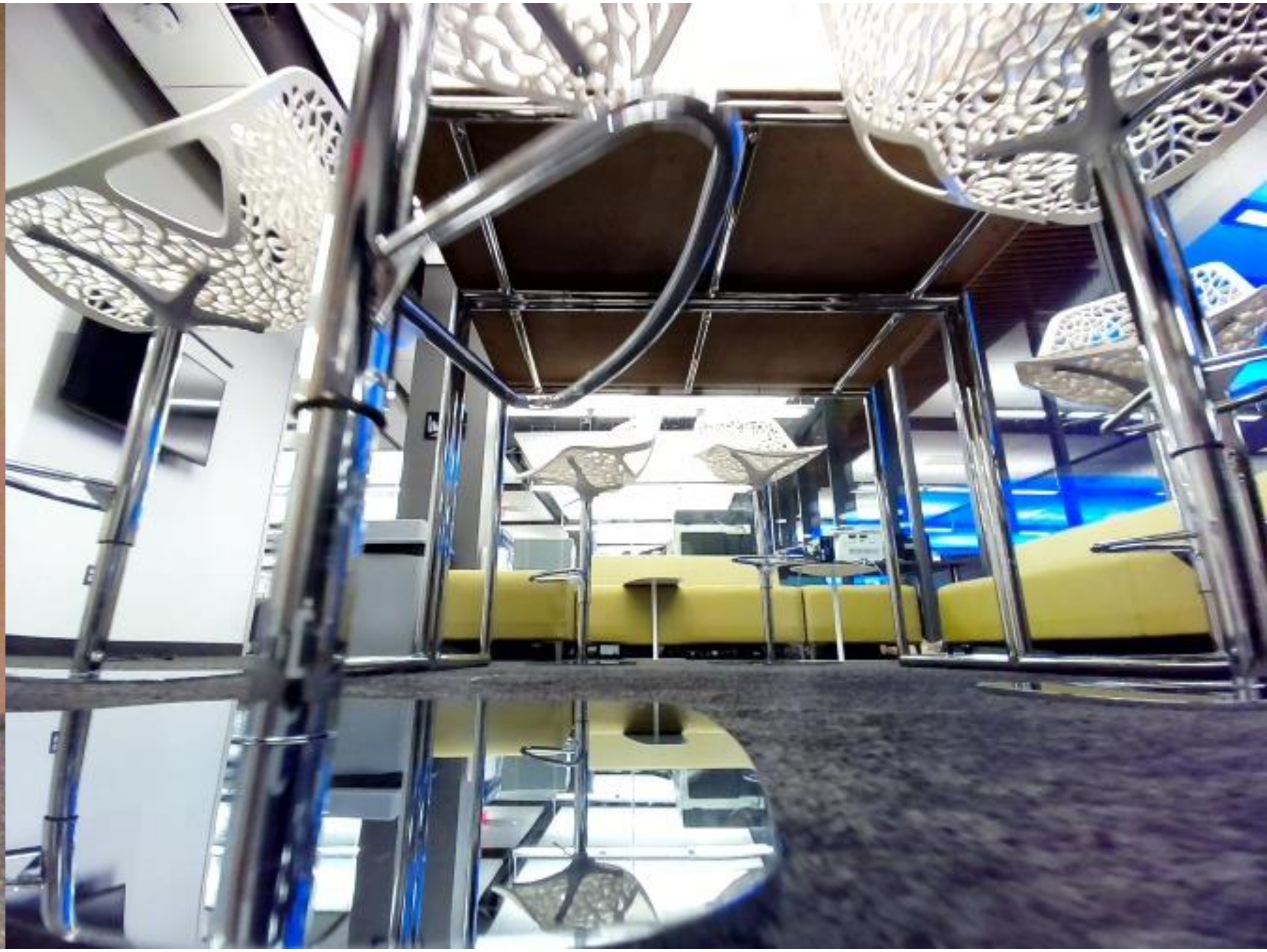
Camera access Roborock G10S





Camera access Dreame L10s Ultra





Camera access Roborock S8

Dreame

- Good news: this year, no SSH credentials in the firmware
- Made lots of improvements of the software
- Bad news:
  - introduced lots of “calling-home” functions
  - Enforce geo-blocking by IP address

Dreame: Camera Monitoring alert

- Most robots with cameras will alert the user via voice prompt when the live view is enabled
- Legal requirement in many countries
- Voice prompts need to be localized

  => Audio files are part of externally downloaded audio packages
- Problem: Audio packages are not signed or encrypted
  – Audio prompt can be overwritten with an empty file
  – Works also on non-rooted devices

Dreame: Firmware signature fail

- New generations of Dreame robots encrypt, then sign firmware
- Firmware update payload:
  - Outside Zip archive, encrypted with static password
  - Random file, signed with private key
  - Inside Zip archive, encrypted with random file as password

- Problem:
  - The actual firmware is not signed, only its password is
  - Password can be recycled for fake firmware updates

Summary

- We have rooting methods for most of the currently released Dreame and Roborock robots
  - Bypass of SecureBoot and any other security features
  - Persistence and operation of custom firmware
- We can validate and verify vendors claims
- The bootloader attack is appliable to a wide range of devices, not only robots
- Work allows further research into IoT and AI

Final notes

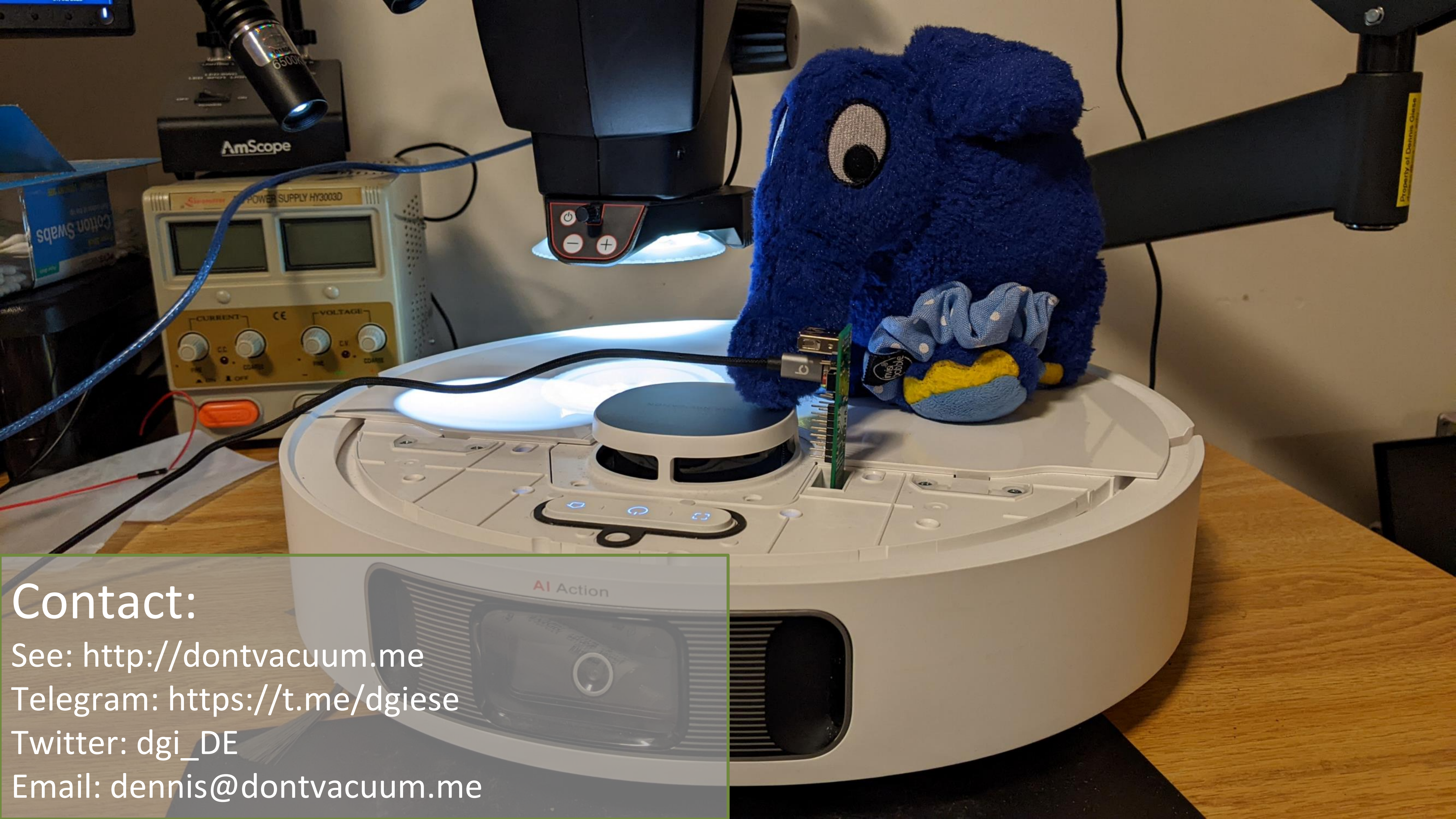
- Do not use the knowledge for bad things!
- Help others if they need help with rooting. Share rooting PCBs
- Tools will be published over the next days. Please be patient
- Join me in robot hacking and flash forensics at hardwear.io NL

Secure Your Hardware
Hardwear.io Security Trainings and Conference
Netherlands 2023
Date : 30th October - 3rd November 2023
Venue: Marriott Hotel, The Hague, Netherlands

Acknowledgements

- Daniel Wegemer
- Guevara Noubir
- Sören Beye
- Mikael Kolkinn

- And all the testers in the community!



Contact:

See: http://dontvacuum.me
Telegram: https://t.me/dgiese
Twitter: dgi_DE
Email: dennis@dontvacuum.me