All wireless communication stacks are equally broken

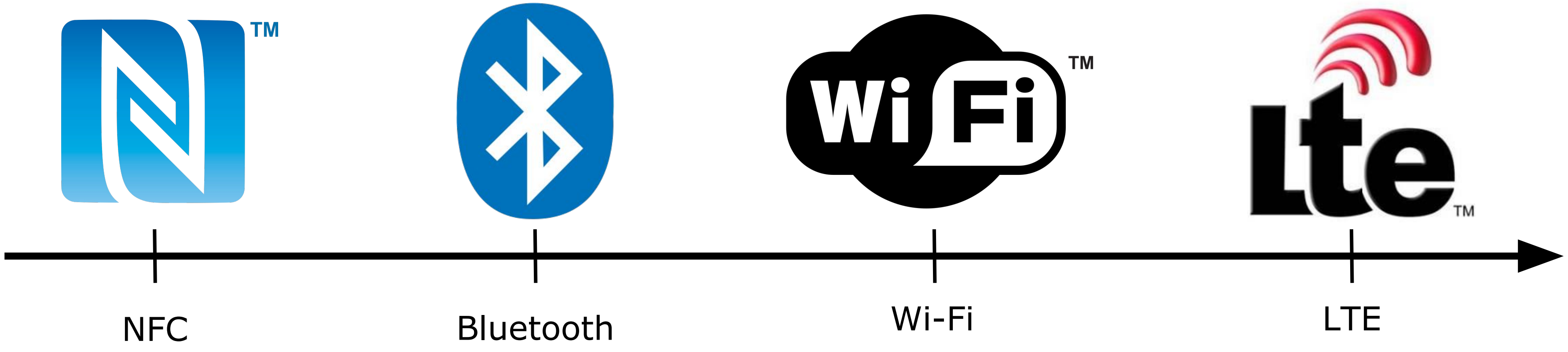
Jiska Classen
Secure Mobile Networking Lab - SEEMOO
Technische Universität Darmstadt, Germany

CROSSING
emergenCITY
Privacy and Trust for Mobile Users
ATHENE
National Research Center for Applied Cybersecurity
CYSEC
Cybersecurity
TU Darmstadt
TECHNISCHE UNIVERSITÄT DARMSTADT

A foundation talk???

Wireless communication is **fun**damentally broken…

…focus: everything in a **smartphone**.



NFC

Bluetooth

Wi-Fi

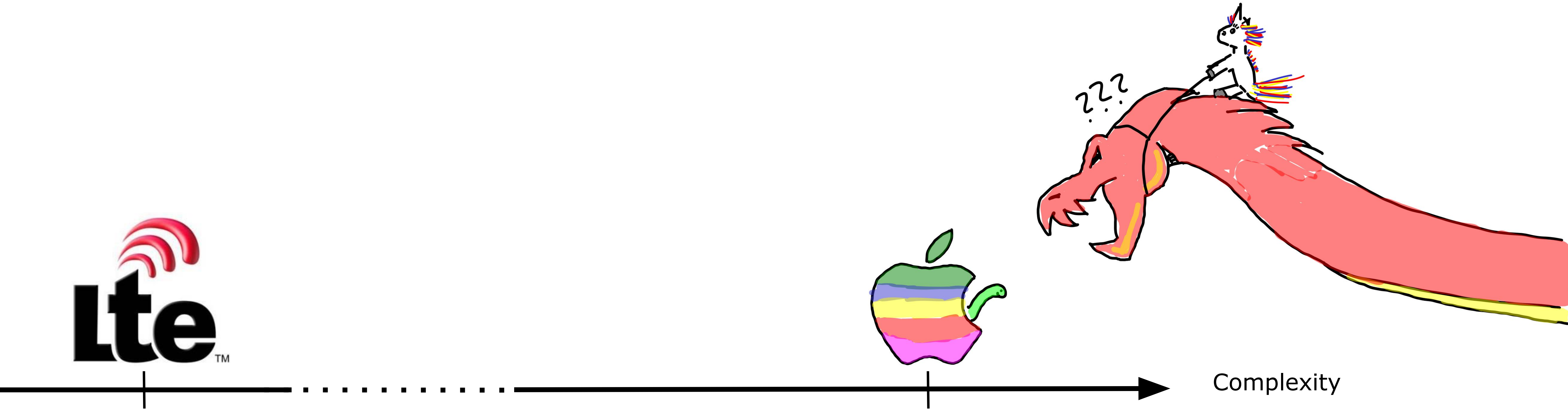
LTE

Communication range

Specification length

Lines of code

Higher complexity rises chance of issues with the specification and implementation!



LTE

Vendor-specific additions

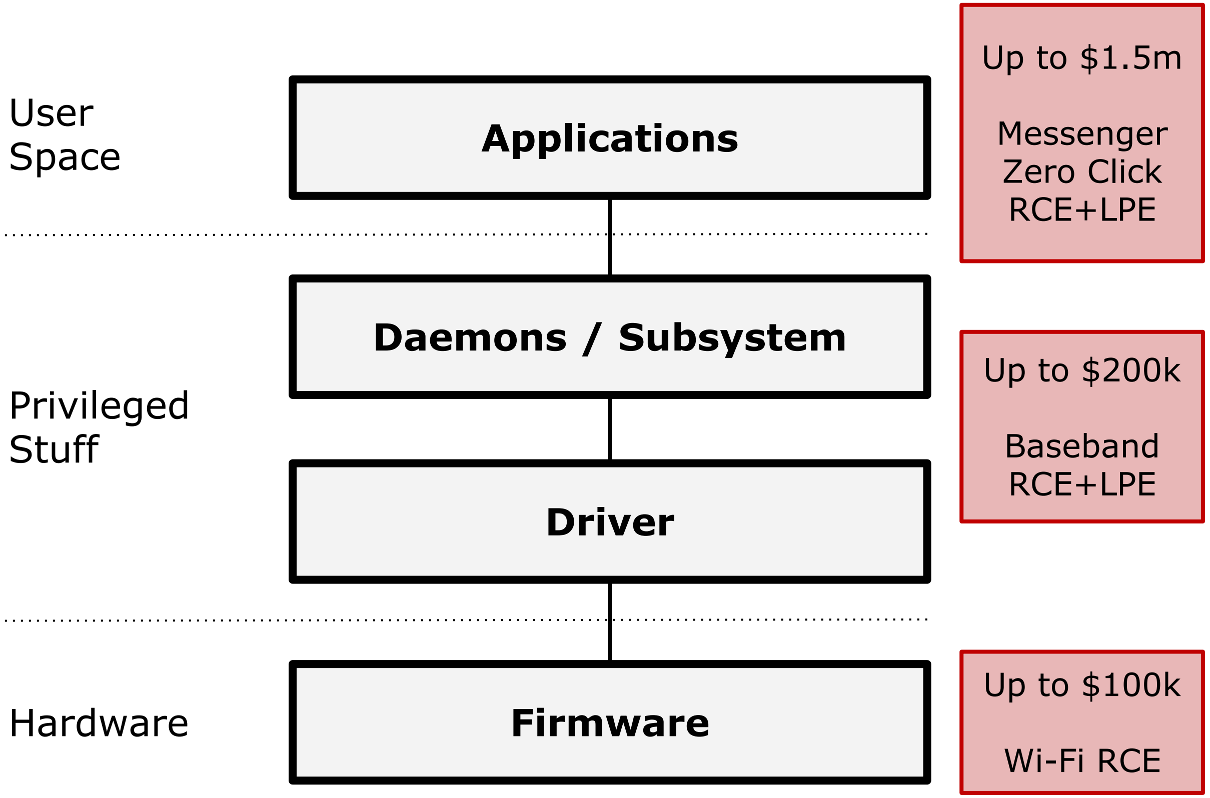
Complexity

WIRELESS EXPLOITATION

NEW Fuzzing Techniques

NEW Escalation Targets

Layers and Privileges



User Space

Privileged Stuff

Hardware

Applications

Daemons / Subsystem

Driver

Firmware

Up to $1.5m

Messenger Zero Click RCE+LPE

Up to $200k

Baseband RCE+LPE

Up to $100k

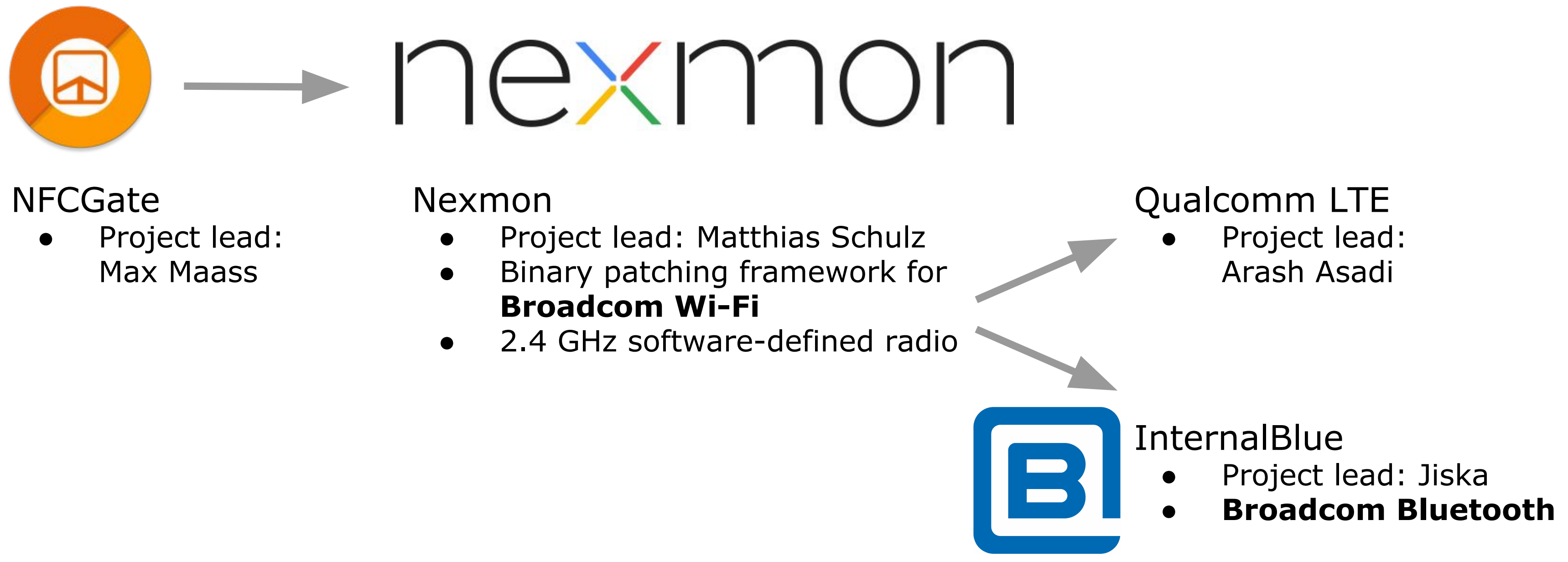
Wi-Fi RCE

- Execution within a component means security measures like **encryption** become **ineffective**.
- Less interaction / more distance / harder to find / privileged component / higher market demand

  → more expensive
- Attackers hate physical proximity!

RCE: Remote Code Execution, LPE: Local Privilege Escalation.
Zerodium price list in December 2019, actual prices on the black market might vary.

Advanced Wireless Tooling @ SEEMOO



NFCGate

- Project lead: Max Maass

Nexmon

- Project lead: Matthias Schulz
- Binary patching framework for **Broadcom Wi-Fi**
- 2.4 GHz software-defined radio

Qualcomm LTE

- Project lead: Arash Asadi

InternalBlue

- Project lead: Jiska
- **Broadcom Bluetooth**

OWL / OpenDrop

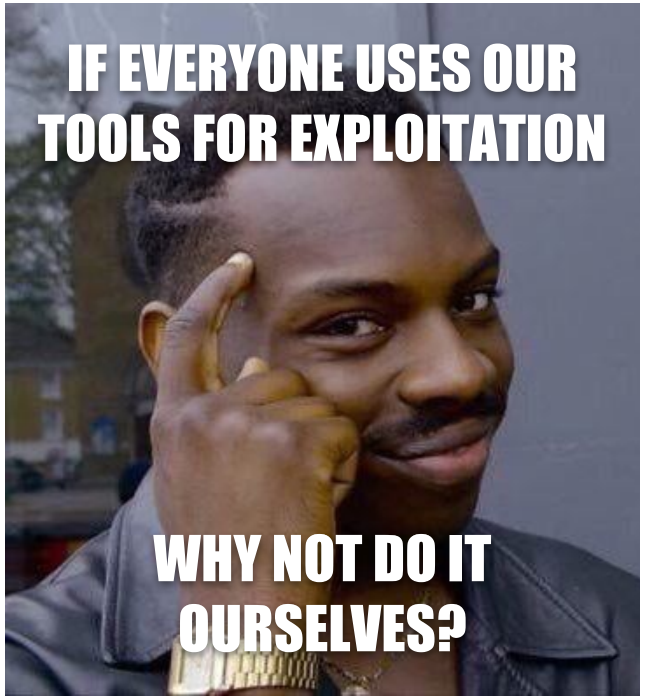
- Project lead: Milan Stute
- Open source **Apple AirDrop** implementation

Most of our projects are available online! https://github.com/seemoo-lab

Hackers gonna hack...

nexmon

- Google Project Zero (April 2017, Gal Beniamini)
- Broadpwn (July 2017, Nitay Artenstein)
- Quarkslab (April 2019, Hugues Anguelkov)

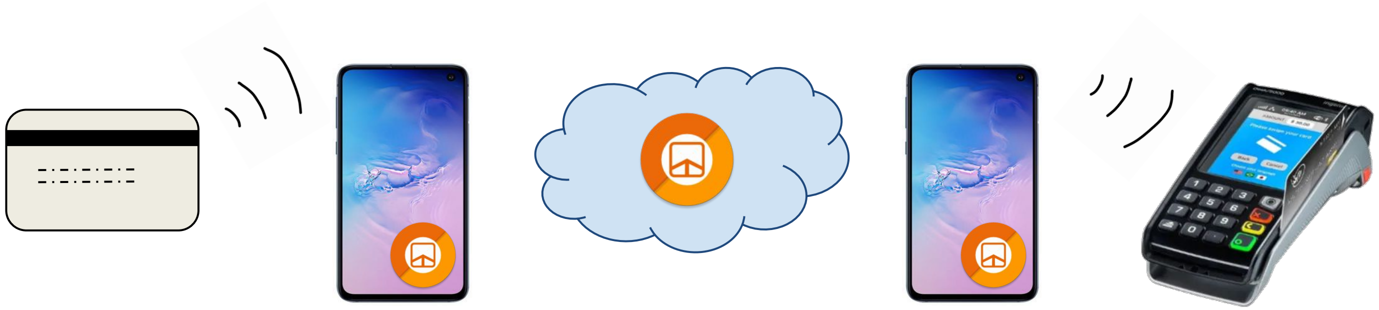


- Demonstration of the KNOB attack on Bluetooth key negotiation
  (August 2019, Daniele Antonioli et. al.)

- Honeypots @ Black Hat
- AirDos
  (December 2019, Kishan Bagaria)

~~NEAR FIELD~~ COMMUNICATION

NFCGate



- Wireless signals travel with speed of light, distance bounding is possible.
- NFC applications usually do not check any time constraints.
- Lab project:
  - Forward communication of an NFC-based payment system.
  - Vulnerable to **relays and even modification of messages** in some cases.
- Solution:
  - **3rd parties asked our students to stop testing :)**

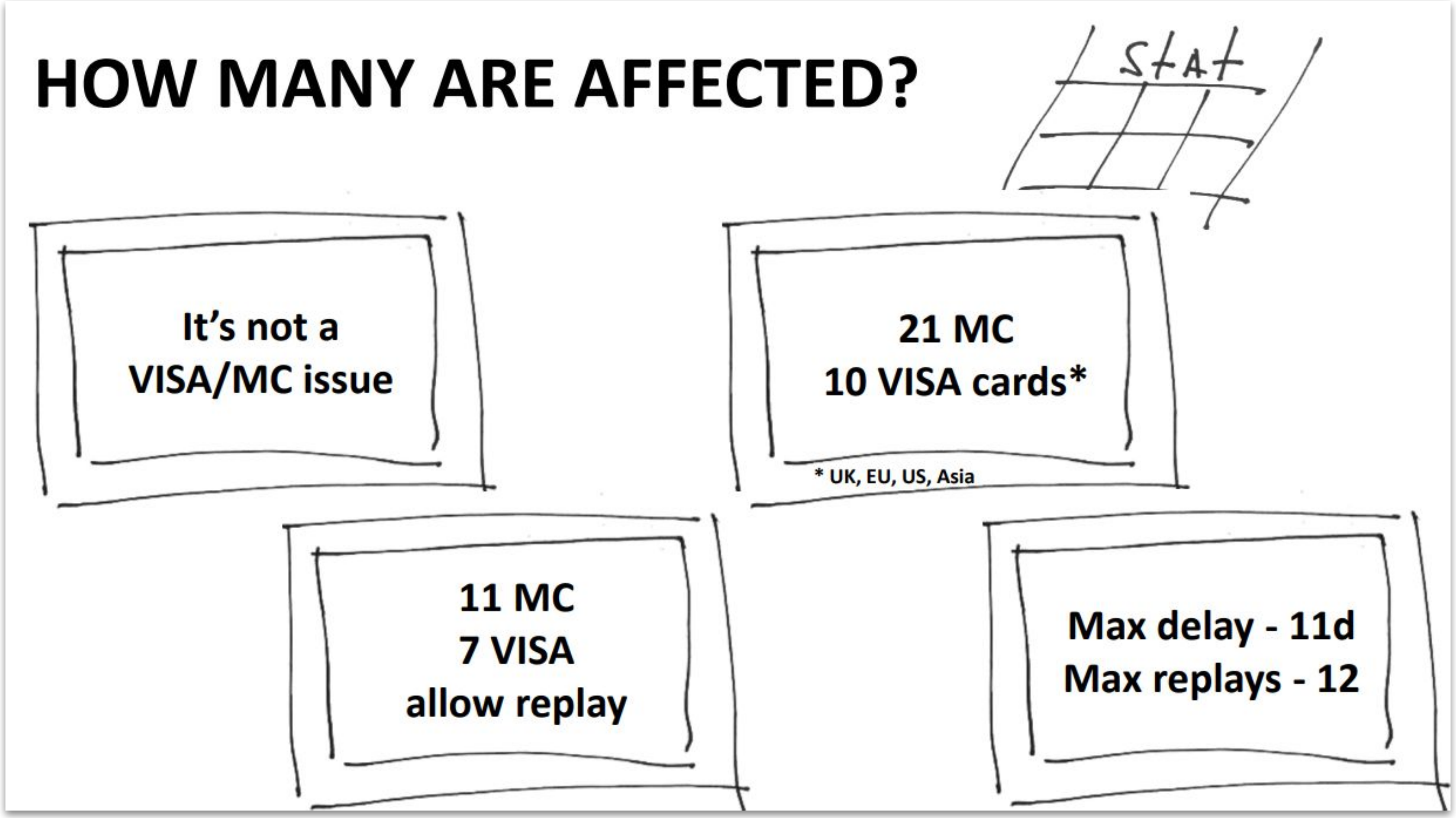
https://github.com/nfcgate/nfcgate (also by SEEMOO)

~~Near Field~~ Communication

VISA …

… specification compliant fraud \o/

Other 3rd parties continued analyzing NFC security.



HOW MANY ARE AFFECTED?

It's not a VISA/MC issue

21 MC
10 VISA cards*

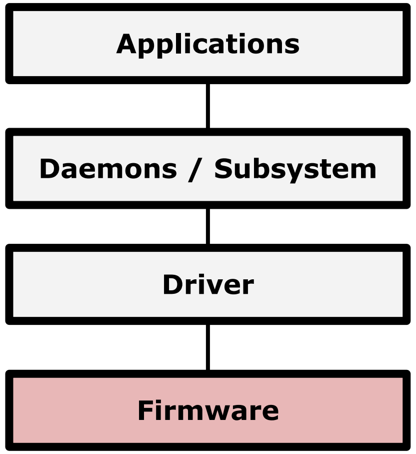
* UK, EU, US, Asia

11 MC
7 VISA
allow replay

Max delay - 11d
Max replays - 12

First Contact - Vulnerabilities in Contactless Payments: https://www.blackhat.com/eu-19/briefings/schedule/index.html#first-contact---vulnerabilities-in-contactless-payments-17454

BLUETOOTH CHIP REMOTE CODE EXECUTION



Applications

Daemons / Subsystem

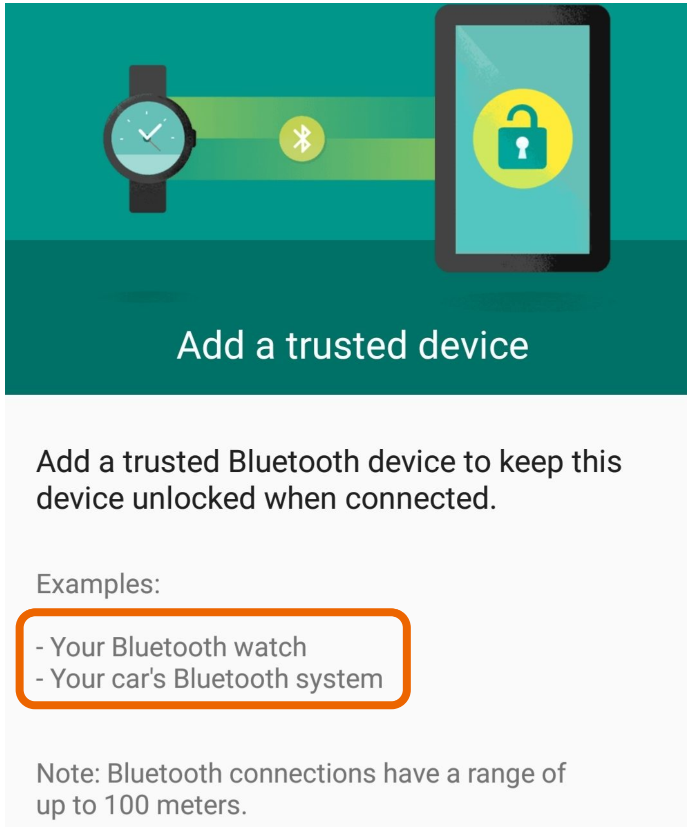
Driver

Firmware

Code Execution on a Bluetooth Chip

- Request the **encryption keys** for any MAC address.
  - Specification compliant request: `HCI_Link_Key_Request`.
  - Impersonate devices, overhear encrypted communication, …

    → Break Android **Smart Lock** and similar features!



Add a trusted device

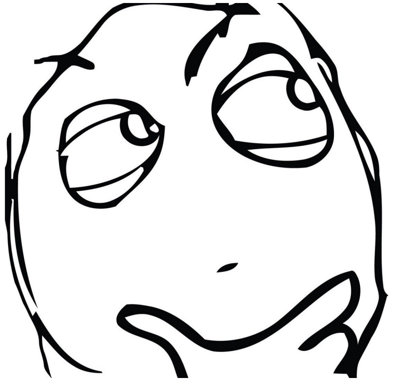
Add a trusted Bluetooth device to keep this device unlocked when connected.

Examples:

- Your Bluetooth watch
- Your car's Bluetooth system

Note: Bluetooth connections have a range of up to 100 meters.

LAST TIME I UPDATED BLUETOOTH IN MY CAR?

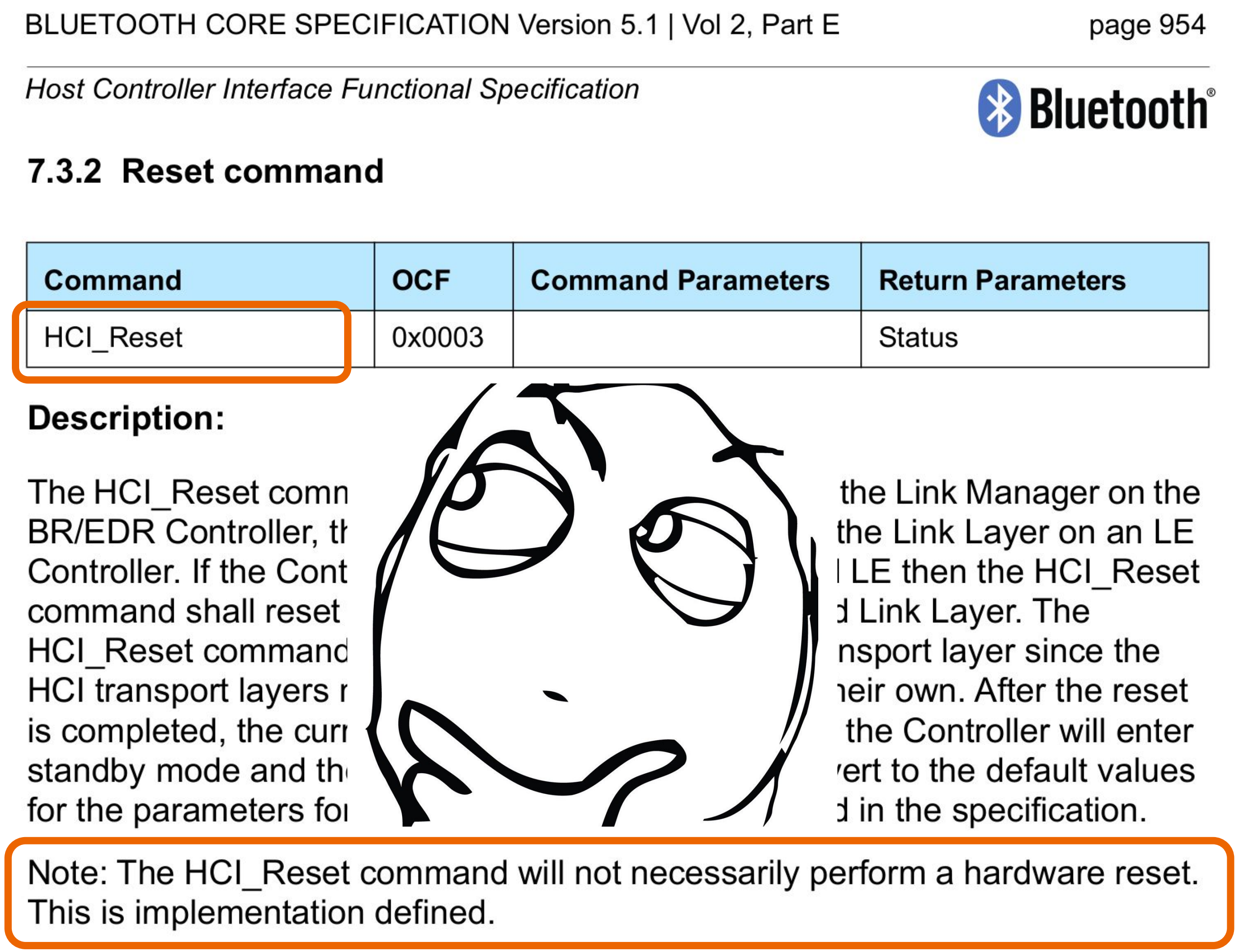


- More possibilities to **escalate** into other components.

Exploit Persistence

- Broadcom/Cypress chips only flush queues, connections, etc. upon reset.
  No full hardware reset.
- Many operating systems will only issue a `HCI_Reset` command.

- ~~**Flight mode**~~ might hard reset the chip.
- ~~**Reboot**~~ might hard reset the chip.
  (Coexistence behavior sometimes persists…)
- **Turning off your smartphone will hard reset the chip.**



BLUETOOTH CORE SPECIFICATION Version 5.1 | Vol 2, Part E page 954

Host Controller Interface Functional Specification

7.3.2 Reset command

Command	OCF	Command Parameters	Return Parameters
HCI_Reset	0x0003		Status

Description:

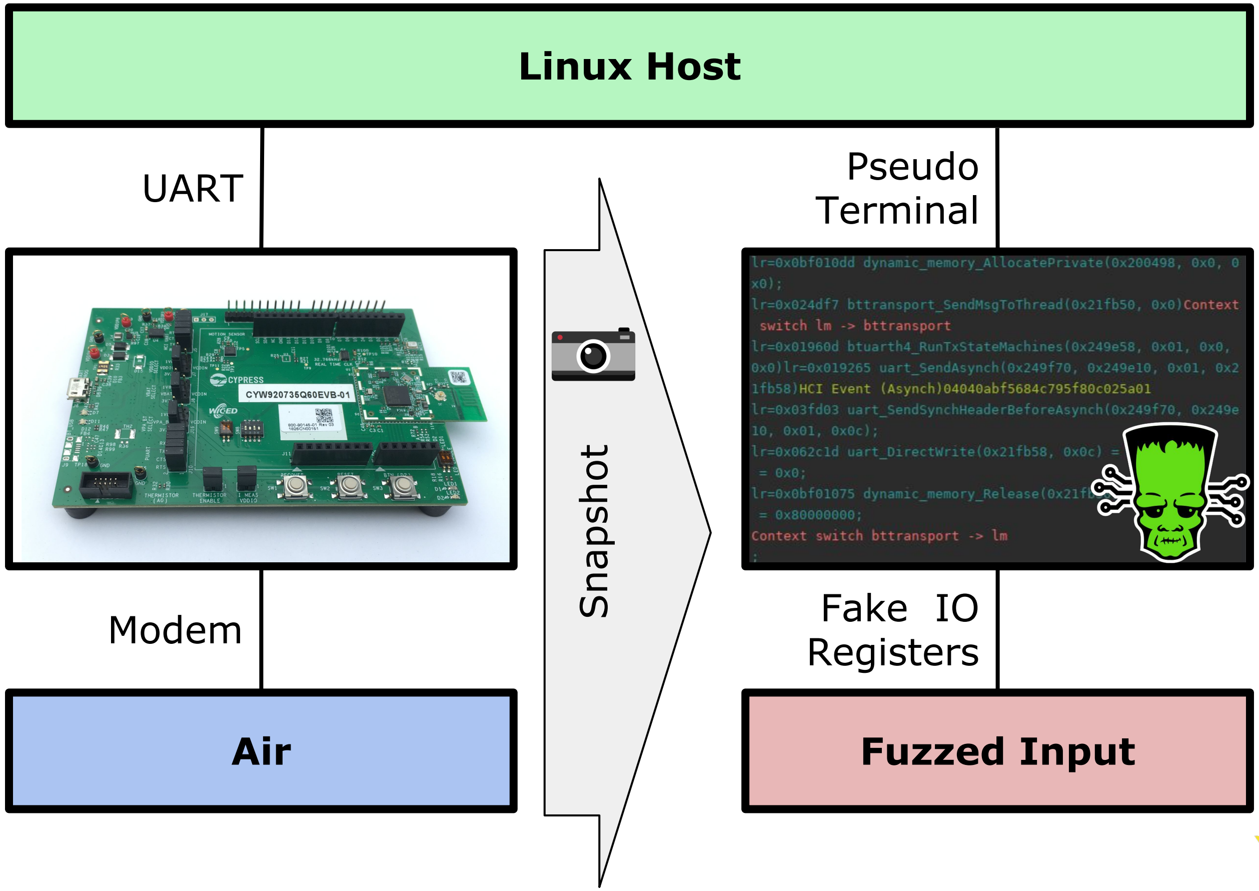
The HCI_Reset comn the Link Manager on the
BR/EDR Controller, tl the Link Layer on an LE
Controller. If the Cont l LE then the HCI_Reset
command shall reset d Link Layer. The
HCI_Reset command nsport layer since the
HCI transport layers r ıeir own. After the reset
is completed, the curr the Controller will enter
standby mode and th ert to the default values
for the parameters fo d in the specification.

Note: The HCI_Reset command will not necessarily perform a hardware reset. This is implementation defined.

NEW

Fuzzing Techniques

Frankenstein



```
lr=0x0bf010dd dynamic_memory_AllocatePrivate(0x200498, 0x0, 0
x0);
lr=0x024df7 bttransport_SendMsgToThread(0x21fb50, 0x0)Context
 switch lm -> bttransport
lr=0x01960d btuarth4_RunTxStateMachines(0x249e58, 0x01, 0x0,
0x0)lr=0x019265 uart_SendAsynch(0x249f70, 0x249e10, 0x01, 0x2
1fb58)HCI Event (Asynch)04040abf5684c795f80c025a01
lr=0x03fd03 uart_SendSynchHeaderBeforeAsynch(0x249f70, 0x249e
10, 0x01, 0x0c);
lr=0x062c1d uart_DirectWrite(0x21fb58, 0x0c) =
 = 0x0;
lr=0x0bf01075 dynamic_memory_Release(0x21fb
 = 0x80000000;
Context switch bttransport -> lm
;
```

Emulate Bluetooth firmware with the same speed as in hardware for realistic **full-stack fuzzing**.

- The Linux host can run a full Bluetooth stack on a desktop setup.
- Add an `xmit_state` 📷 hook to the Bluetooth firmware function of interest, e.g., device scanning, active connection, …
- Reattach emulated snapshot with `btattach`, enter a similar state on the desktop, and start fuzzing.

DEMO!

Fuzzinating!

Jan: Fuzzes **early connection states**, finds **heap overflows** in basic packet types.

iPhone 11: Can you hear me? I come with Bluetooth 5.1!

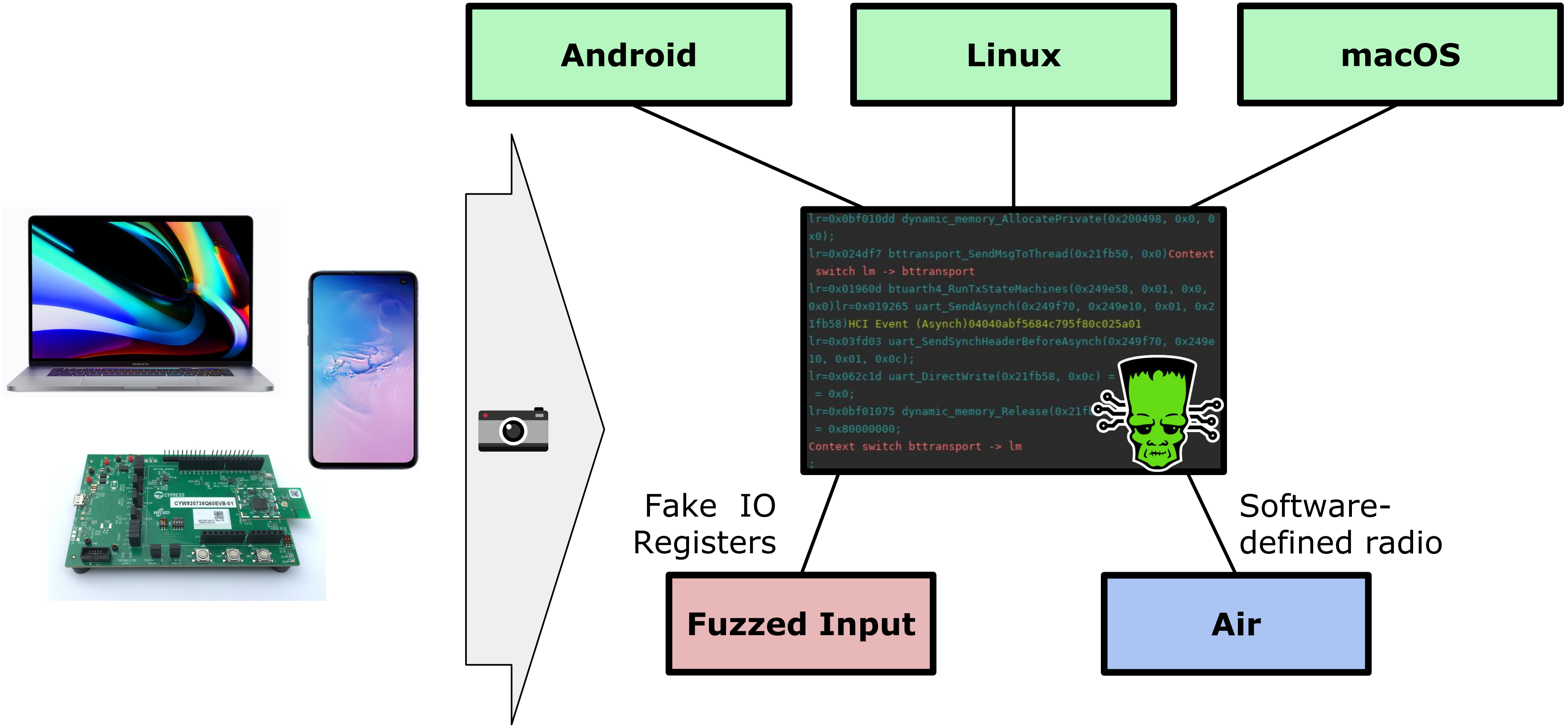
Me: The connection state can be paired and encrypted, any user or app interaction is valid input, as long as I can get code execution on that chip.

… enters a more complicated state for fuzzing …

Me: Oh noes, they misconfigured the heap on this one specific evaluation board. Fixing the heap hard-bricked one evaluation board.

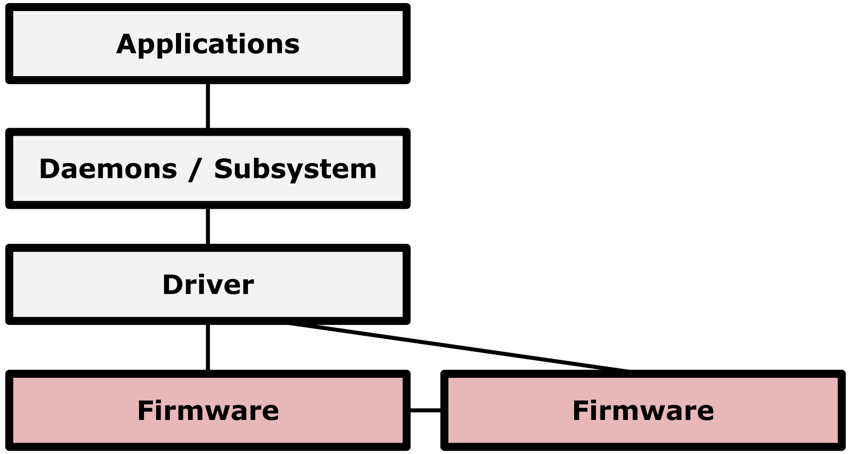
… porting ~200 handwritten hooks to another evaluation board with correct heap …

Fuzz next?

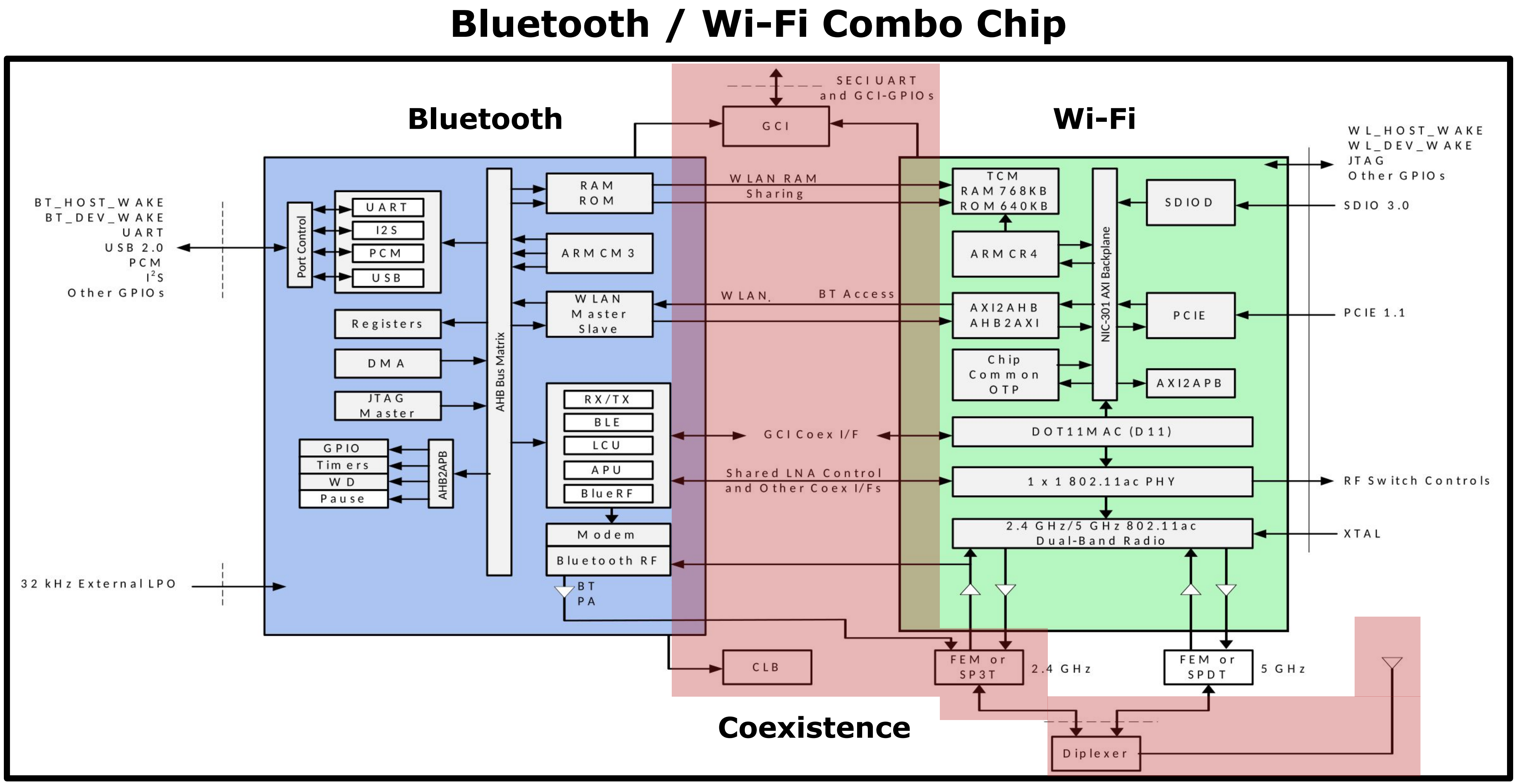


```
lr=0x0bf010dd dynamic_memory_AllocatePrivate(0x200498, 0x0, 0
x0);
lr=0x024df7 bttransport_SendMsgToThread(0x21fb50, 0x0)Context
 switch lm -> bttransport
lr=0x01960d btuarth4_RunTxStateMachines(0x249e58, 0x01, 0x0,
0x0)lr=0x019265 uart_SendAsynch(0x249f70, 0x249e10, 0x01, 0x2
1fb58)HCI Event (Asynch)04040abf5684c795f80c025a01
lr=0x03fd03 uart_SendSynchHeaderBeforeAsynch(0x249f70, 0x249e
10, 0x01, 0x0c);
lr=0x062c1d uart_DirectWrite(0x21fb58, 0x0c) =
 = 0x0;
lr=0x0bf01075 dynamic_memory_Release(0x21f
 = 0x80000000;
Context switch bttransport -> lm
;
```

CHIP LEVEL ESCALATION



Coexistence: Escalation within the Chip



Francesco Gringoli and me worked on this.

Coexistence???

Francesco: I guess it’s just a marketing feature.

Me: Aww it must be an exploitation feature!

… traveling to Italy for eating some gelato …

Reality: Hard-coded blacklisting and traffic classes for Bluetooth and Wi-Fi.

Tons of patents.

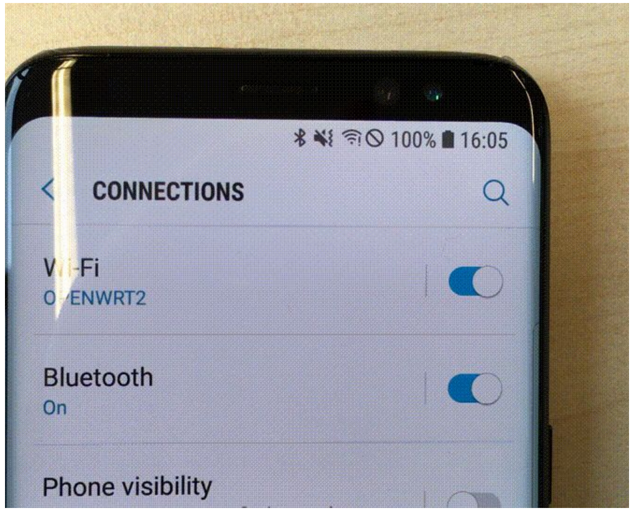
Proprietary deluxe!

NEW

Escalation Targets

Almost a Demo :)

- You can disable Wi-Fi via Bluetooth and Bluetooth via Wi-Fi.
- Sometimes requires manual reboot to get wireless stuff working again.
- Buggy **driver panics** some older Androids and all **up-to-date iPhones**.
- Broadcom says **six months** might be sufficient to **fix firmware**.
- … but exploitation requires code execution on the Bluetooth or Wi-Fi chip.



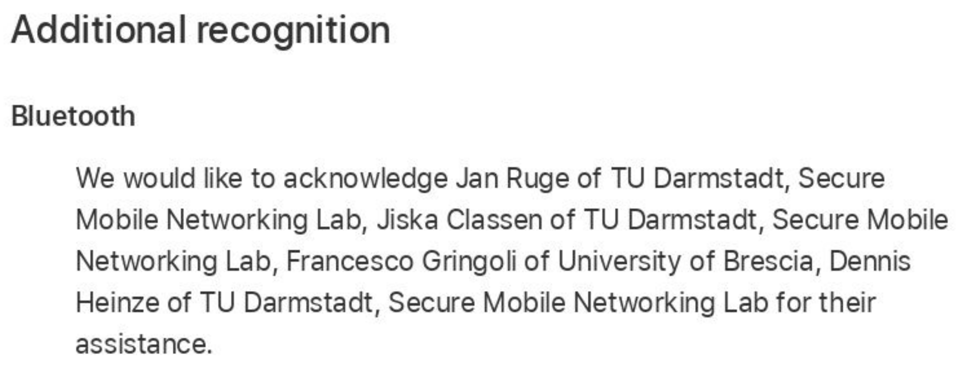
CONNECTIONS

Wi-Fi
O ENWRT2

Bluetooth
On

Phone visibility

iOS 13 Release Notes



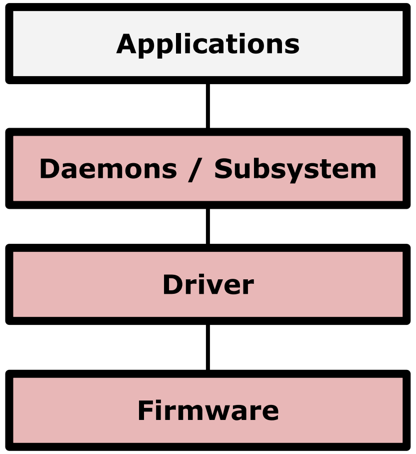
Additional recognition

Bluetooth

We would like to acknowledge Jan Ruge of TU Darmstadt, Secure Mobile Networking Lab, Jiska Classen of TU Darmstadt, Secure Mobile Networking Lab, Francesco Gringoli of University of Brescia, Dennis Heinze of TU Darmstadt, Secure Mobile Networking Lab for their assistance.

Tested iOS 12.4 (reported end of August), still not fixed in 13.3...

ESCALATE ALL THE STACKS



Applications

Daemons / Subsystem

Driver

Firmware

Attacking Bluetooth Hosts

- BlueBorne: Various attacks on Android, Windows, Linux, iOS.
- … okay but that was 2017?

If someone looked into it, it must be secure now!

Vulnerability with a logo!

- **IoT** gadgets, wireless headphones, fitness trackers, …
- **Apple** ecosystem: Bluetooth is almost everywhere and always enabled.
- **Web Bluetooth**: BLE support within various browsers.

2020 might bring a couple of BlueBorne like attacks.

BlueBorne by Armis https://www.armis.com/blueborne/

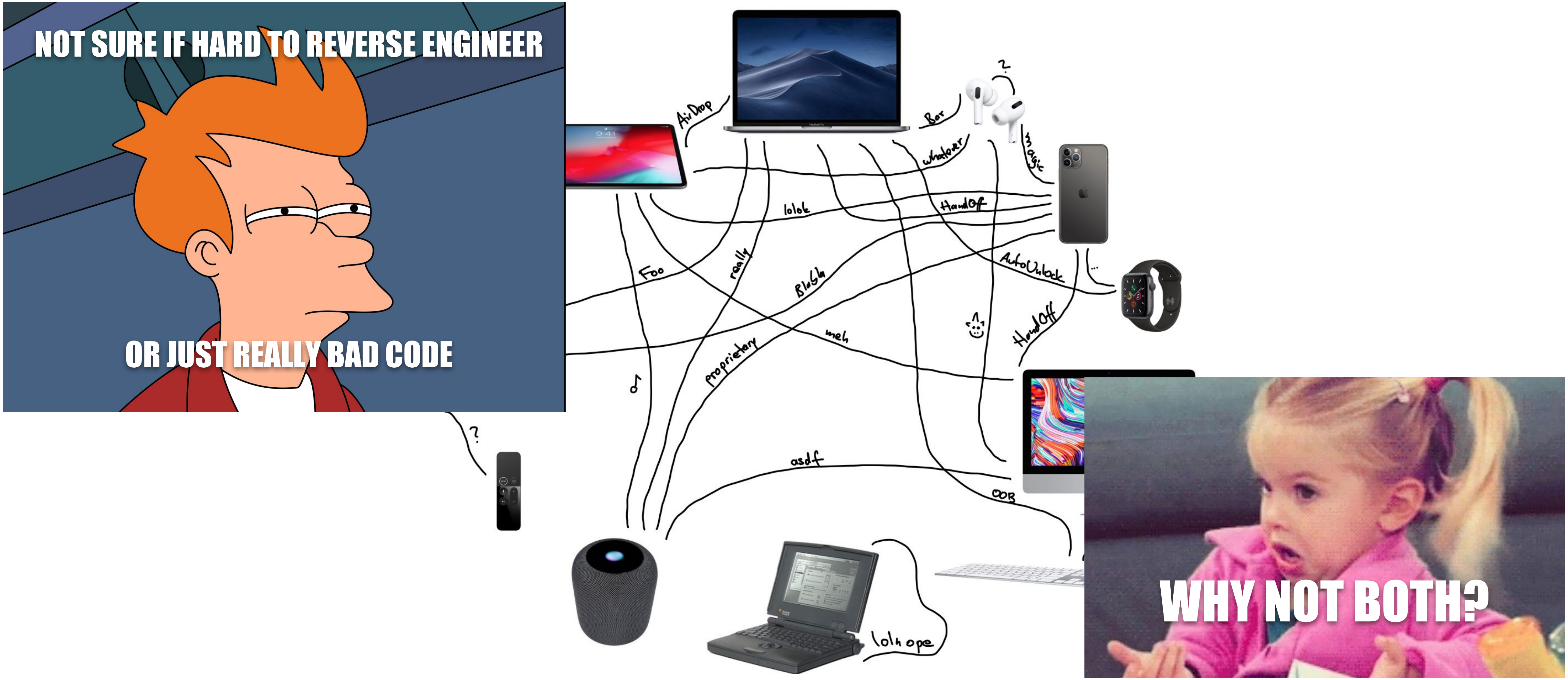
The Linux Bluetooth Stack

Number of commits in BlueZ:

- 23% Committer #1
- 17% Committer #2
- 15% Committer #3
- 5% Committer #4

Tim Walter did some fuzzing in the Linux Bluetooth Stack :)

The Apple Bluetooth Stack(s)



Dennis Heinze (@ttdennis) ported *InternalBlue* to iOS, Davide Toldo (@unixb0y) ported it to macOS. They enjoyed it a lot!
Alexander Heinrich (@Sn0wfreeze) supervised by Milan Stute looked into Handoff.

The Android Bluetooth Stack

Because all our chip reversing projects start on Android.

Bluetooth for Bluescreens???

:(

Sadly I couldn’t find any student who wants to work on this yet.

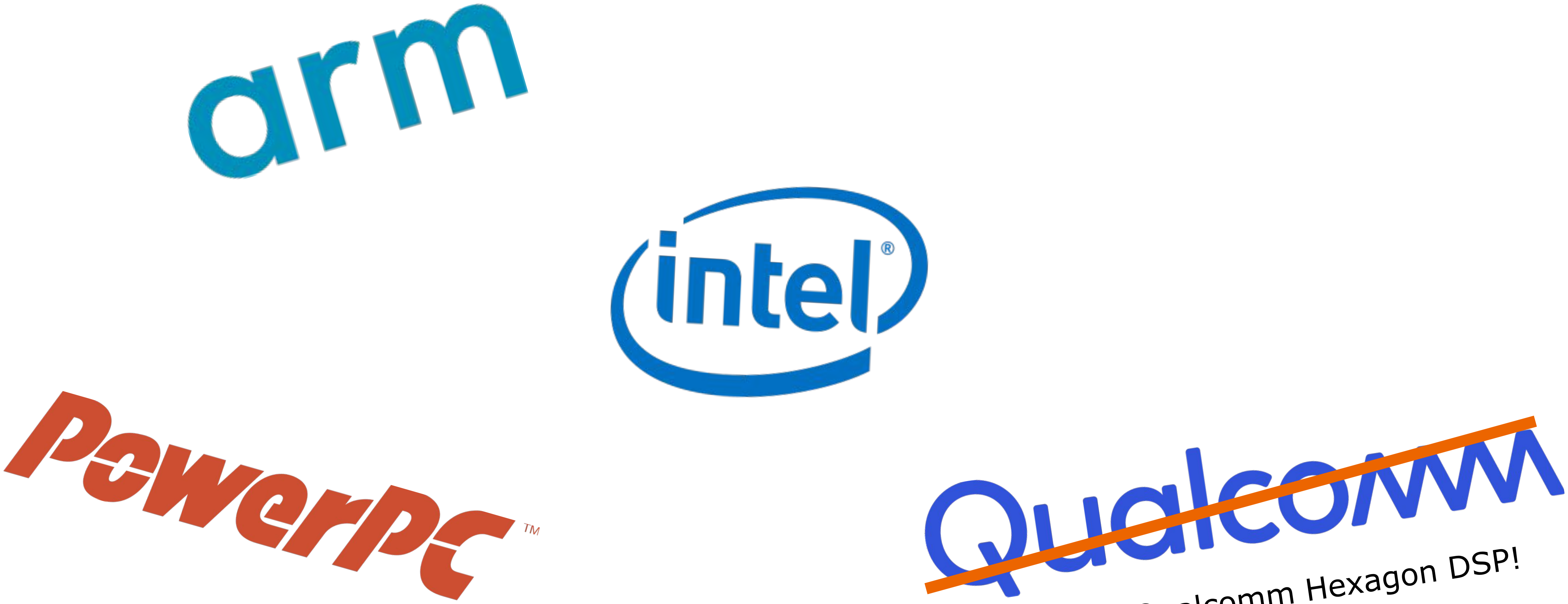
0% complete

But if you are really into pain, consider this as a job offer
for a student thesis @ SEEMOO :D

LTE*

* LONG TERM EXPLOITATION

All Assembly is Beautiful!



Not you, Qualcomm Hexagon DSP!

Simjacker and WIBAttack

- Purpose of a SIM card: **Protect sensitive key material**. (??!!)

- SIM cards can be configured remotely by your telecommunication provider.
- … SIM cards including eSIMs …
  - Receiving a victim’s location,
  - fraud by dialing premium numbers,
  - … launch browser.

Simjacker

Vulnerability with a logo!

SIM card technology from A-Z
LaF0rge
36C3

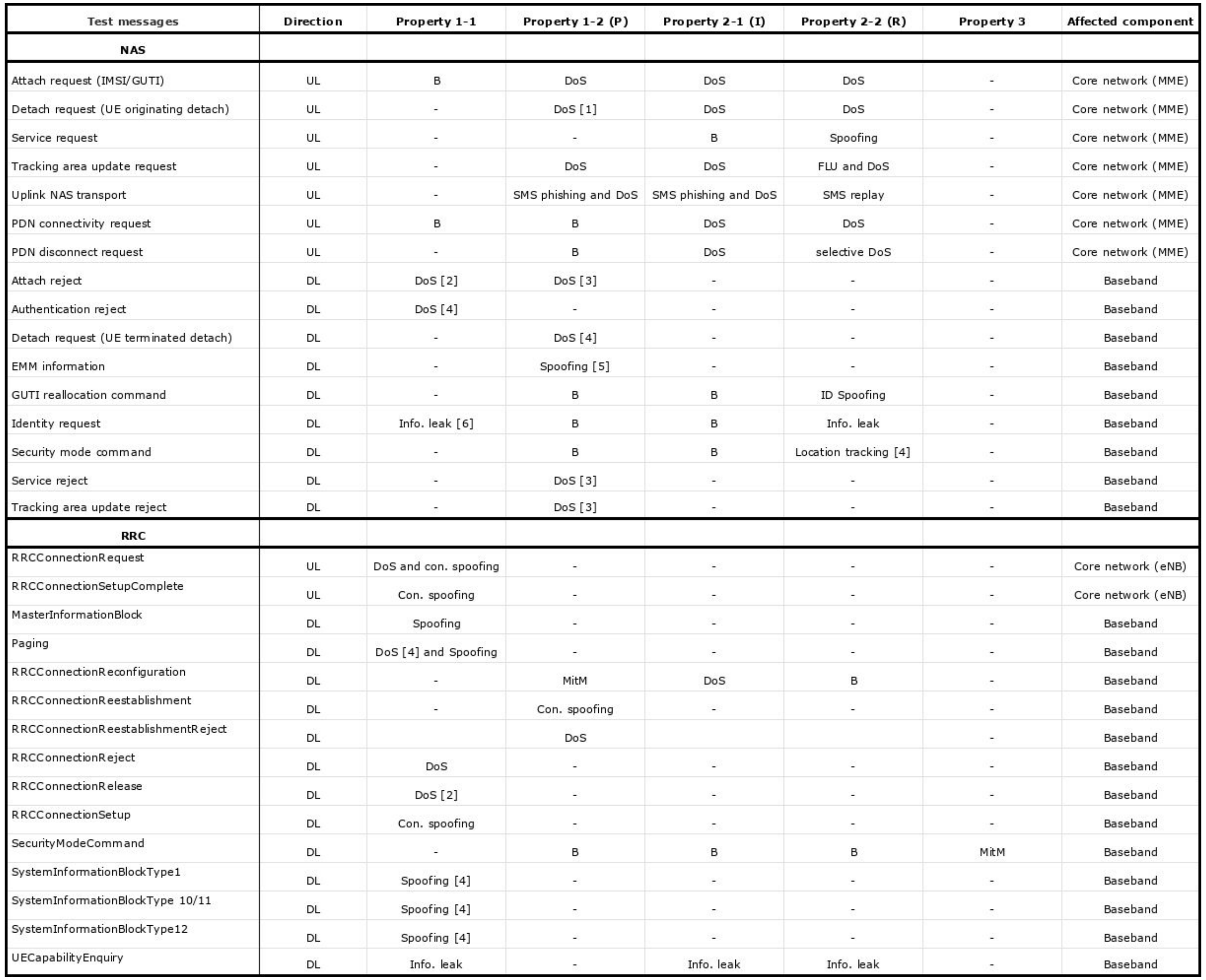
- I’m a Telekom business customer, making a call to the support hotline takes less than 3 minutes.
- Phone: 13.9. 2x, 19.9., 21.9., 27.9., 1.10., 17.10.
- Mails: 3x …
- **Still no answer what is running on my SIM cards.**

https://simjacker.com/

LTEFuzz

- Highly complex LTE state machines.
- Implementation failures in backends and mobile devices.

SigOver + alpha
CheolJun Park,
Mincheol Son



Test messages	Direction	Property 1-1	Property 1-2 (P)	Property 2-1 (I)	Property 2-2 (R)	Property 3	Affected component
NAS							
Attach request (IMSI/GUTI)	UL	B	DoS	DoS	DoS	-	Core network (MME)
Detach request (UE originating detach)	UL	-	DoS [1]	DoS	DoS	-	Core network (MME)
Service request	UL	-	-	B	Spoofing	-	Core network (MME)
Tracking area update request	UL	-	DoS	DoS	FLU and DoS	-	Core network (MME)
Uplink NAS transport	UL	-	SMS phishing and DoS	SMS phishing and DoS	SMS replay	-	Core network (MME)
PDN connectivity request	UL	B	B	DoS	DoS	-	Core network (MME)
PDN disconnect request	UL	-	B	DoS	selective DoS	-	Core network (MME)
Attach reject	DL	DoS [2]	DoS [3]	-	-	-	Baseband
Authentication reject	DL	DoS [4]	-	-	-	-	Baseband
Detach request (UE terminated detach)	DL	-	DoS [4]	-	-	-	Baseband
EMM information	DL	-	Spoofing [5]	-	-	-	Baseband
GUTI reallocation command	DL	-	B	B	ID Spoofing	-	Baseband
Identity request	DL	Info. leak [6]	B	B	Info. leak	-	Baseband
Security mode command	DL	-	B	B	Location tracking [4]	-	Baseband
Service reject	DL	-	DoS [3]	-	-	-	Baseband
Tracking area update reject	DL	-	DoS [3]	-	-	-	Baseband
RRC							
RRCConnectionRequest	UL	DoS and con. spoofing	-	-	-	-	Core network (eNB)
RRCConnectionSetupComplete	UL	Con. spoofing	-	-	-	-	Core network (eNB)
MasterInformationBlock	DL	Spoofing	-	-	-	-	Baseband
Paging	DL	DoS [4] and Spoofing	-	-	-	-	Baseband
RRCConnectionReconfiguration	DL	-	MitM	DoS	B	-	Baseband
RRCConnectionReestablishment	DL	-	Con. spoofing	-	-	-	Baseband
RRCConnectionReestablishmentReject	DL		DoS			-	Baseband
RRCConnectionReject	DL	DoS	-	-	-	-	Baseband
RRCConnectionRelease	DL	DoS [2]	-	-	-	-	Baseband
RRCConnectionSetup	DL	Con. spoofing	-	-	-	-	Baseband
SecurityModeCommand	DL	-	B	B	B	MitM	Baseband
SystemInformationBlockType1	DL	Spoofing [4]	-	-	-	-	Baseband
SystemInformationBlockType 10/11	DL	Spoofing [4]	-	-	-	-	Baseband
SystemInformationBlockType12	DL	Spoofing [4]	-	-	-	-	Baseband
UECapabilityEnquiry	DL	Info. leak	-	Info. leak	Info. leak	-	Baseband

https://sites.google.com/view/ltefuzz

RESPONSIBLE DISCLOSURE

Fixing the Heap

Jan: Your heap implementation does not provide any protection against exploitation, here is how you could fix it …

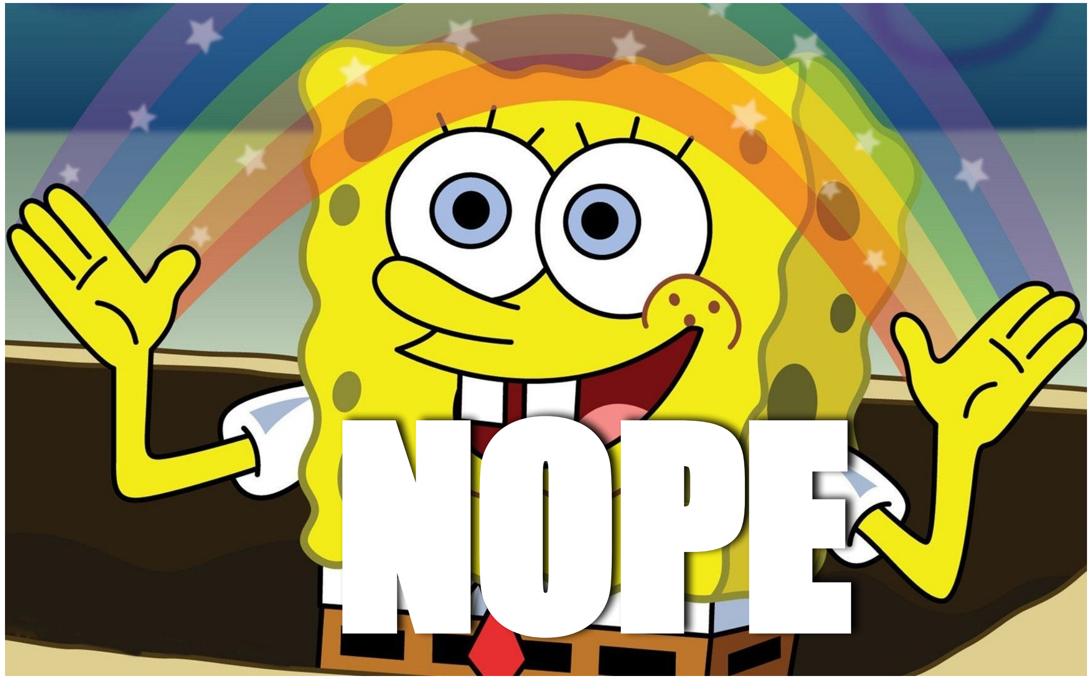
ThreadX: Our heap has already been exploited, see the following blog post.
Please note that **it is up to the application to use the heap correctly**.

https://embedi.org/blog/remotely-compromise-devices-by-using-bugs-in-marvell-avastar-wi-fi-from-zero-knowledge-to-zero-click-rce/ (offline)
https://2018.zeronights.ru/wp-content/uploads/materials/19-Researching-Marvell-Avastar-Wi-Fi.pdf

Is the vulnerability still there?

Me: Your smartphones are still vulnerable.
Samsung: We cannot send you any patches for testing without an NDA.

Broadcom: We can send you patches in advance for testing.
Me: Great!
Broadcom: Can you give us a list of devices?
Me: Sends detailed list, as they can guess it from our infos anyway.
Broadcom: Before sending you the patches, we need an NDA.
Me: LOLNOPE…



Zero day? Some dollars :)

- **Broadcom’s PSIRT**
  - does not hand out CVEs, and
  - not getting into legal trouble is sufficient amount of “bug bounty”.

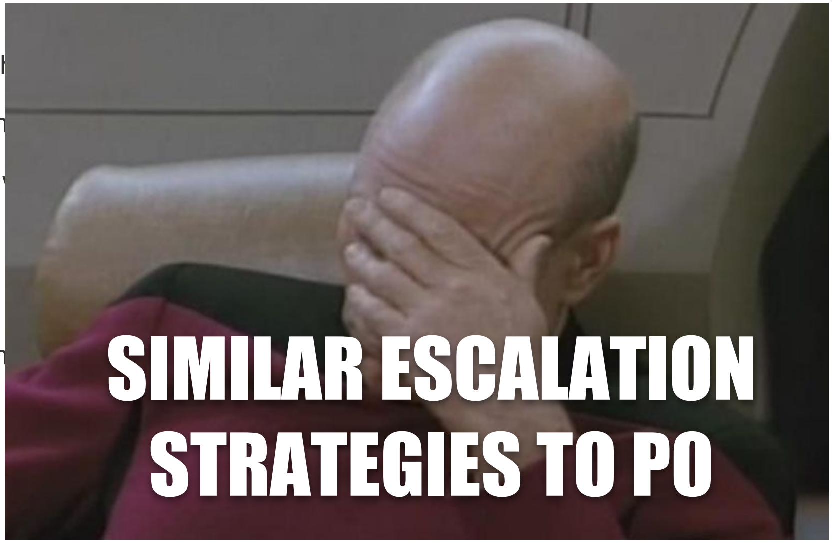
… but their customers and partners (Samsung, Apple, Cypress, …) might still value the work of my students?

- One company which does not want to be mentioned sponsored a **flight to DEF CON**.
- **Samsung** gave a bounty of **$1000**.

Responsible Disclosure Timeline

Quarkslab Responsible Disclosure Timeline for Broadcom Wi-Fi Chips

- 2018-09-13: Email sent to Broadcom detailing the vulnerabilities
- 2018-09-13: Reply from Broadcom acknowledging the report.
- 2018-09-19: Broadcom asks if Quarkslab has a communication plan for the bugs.
- 2018-09-20: Quarkslab replies it plans to publish a blog post and provides URLs to prior publications as example. Asks if Broadcom could reproduce the bugs and if they were already known to them.
- 2018-09-20: **Broadcom** replies that they have "limited ability to share our plans and findings" because there isn't a **Non-Disclosure Agreement (NDA)** signed between the companies and they would be sharing non-public information.
- 2018-09-20: Quarkslab replies that it is not possible to coordinate disclosure if one of the involved parties -the reporter- h
  and when the vendor plans to issue fixes, and that it cannot agree to sign an NDA that would prevent reporting to custon
  the disclosure process was handled. Finally, Quarkslab asks if there is any information that Broadcom may provide that
- 2018-09-20: Broadcom asks if there is a date set for the publication.
- 2018-09-20: Reply indicating the date is not set and it is not entirely dependent on Quarkslab.
- 2018-10-28: Email set to CERT/CC asking for help to coordinate with Broadcom (since it is a US-based vendor) given th
  details about the bugs and a brief timeline of previous communications is provided.



https://blog.quarkslab.com/reverse-engineering-broadcom-wireless-chipsets.html

Responsible Disclosure Timeline

- 2018-10-30: CERT/CC reply asking for further details such as list of vulnerable devices, proof-of-concept program and the planned date of publication.
- 2018-10-30: Quarkslab replies pointing to page 2 of the report which lists versions of firmware confirmed vulnerable. Indicates that PoC is not available at the moment but may be sent the following week and that the publication date is not set, and that both things may not be easy to do since they also depend on availability of a former intern.
- 2018-11-01: Email from Apple saying that Broadcom shared Quarkslab's report with them, they are investigating one of the vulnerabilities and would like to coordinate disclosure. Asks if disclosure date has been set.
- 2018-11-06: Reply from Quarkslab informing Apple that CERT/CC is on the loop as well, explains that Broadcom said it will not provide information unless an NDA is signed, and that the publication date is not set but would likely be before the end of the year. Quarkslab asks if a CVE
- 2018-11-13: Apple replies that a CVE ID will be assigned closer to patch release date and that the
- 2019-01-09: Email from CERT/CC requesting a status update
- 2019-01-10: Reply saying that Quarkslab has not received any communication from Broadcom sir CERT/CC has any news.
- 2019-03-08: Email to CERT/CC asking if there are any news.
- 2019-03-26: CERT/CC replies that it received a response from Broadcom that did not confirm nor 2019. CERT/CC asks Quarkslab if there is any new information.
- 2019-03-28: Apple informs they will be releasing a patch on April 14th, 2019 and asks if Quarkslab
- 2019-04-08: Quarkslab sends mail to Apple and CERT/CC asking if the fix will be for one or more Linux kernel driver on February 14th, 2019 without a CVE ID nor a security notice, and asks if App

Responsible Disclosure Timeline

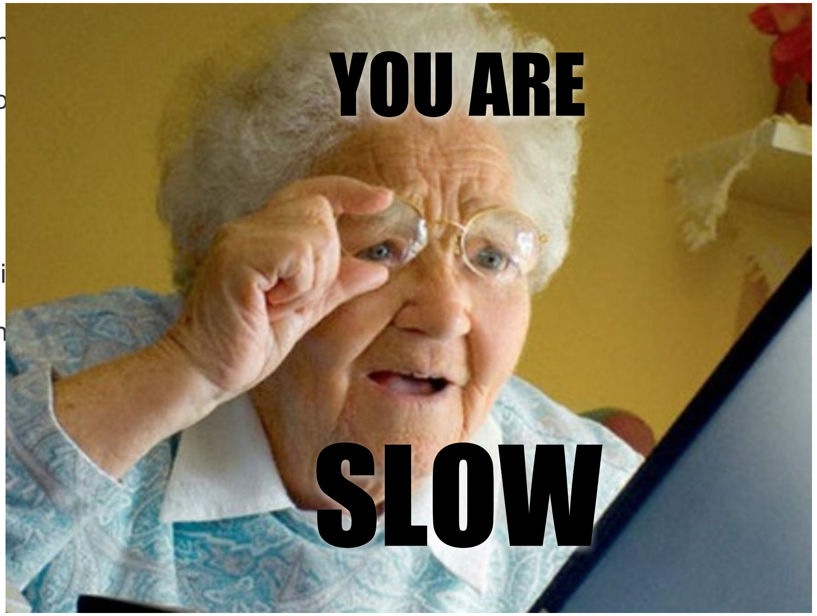
- 2019-04-08: Quarkslab sends mail to Apple and CERT/CC asking if the fix will be for one or more vulnerabilities. Points out that **Broadcom committed a fix** to a bug in their open source **Linux kernel** driver on February 14th, 2019 **without a CVE ID nor a security notice**, and asks if Apple will be fixing the same bug.
- 2019-04-08: CERT/CC asks for permission to send a general notification that includes the report originally sent by Quarkslab in September 2018, says it will assign CVE IDs and that it is drafting a security note that will send for comments.
- 2019-04-08: Quarkslab agrees to have the vulnerability report disseminated.
- 2019-04-10: CERT/CC sends draft vulnerability note and asks if any of the heap overflows could result in code execution. Also asks for URL to Quarkslab blog post.
- 2019-04-11: Apple sends CVE ID and draft of paragraph describing their bugfix. States they are fixing a bug different than the one Broadcom patched in the brcmfmac Linux kernel driver
- 2019-04-12: Quarkslab replies that the GTK bugs could result in remote code execution either on the Linux kernel or on th

  FullMAC). Remote heap layout manipulation is very complicated but RCE should not be discarded as worst case scenario

  Quarkslab will provide publication URL on the week of April 14th.
- 2019-04-12: Apple asks for draft of our blog post. Quarkslab replies that is not yet ready.
- 2019-04-12: CERT/CC sends update vulnerability note with summary description of each vuln and assigned CVE IDs. Poi

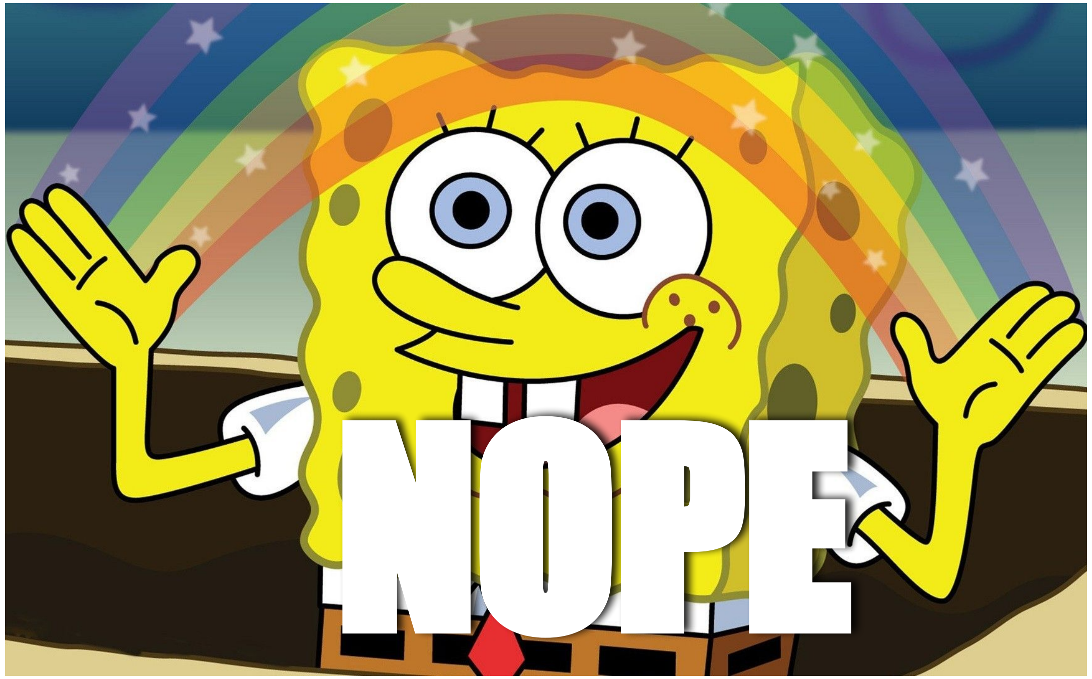
  another bug that was described in the report Quarkslab sent in September 2018. The original response from the vendor in

  though they apparently later supplied patches), and they would not provide information about the wl driver.
- 2019-04-15: CERT/CC asks if Quarkslab will publish on this date. Corrects one of the CVE IDs previously provided.
- 2019-04-15: Quarkslab replies that blog post will very likely go live on the 16th.
- 2019-04-15: Apple sends link to Security Update 2019-002 that fixes CVE-2019-8564
- 2019-04-16: This blog post is published.



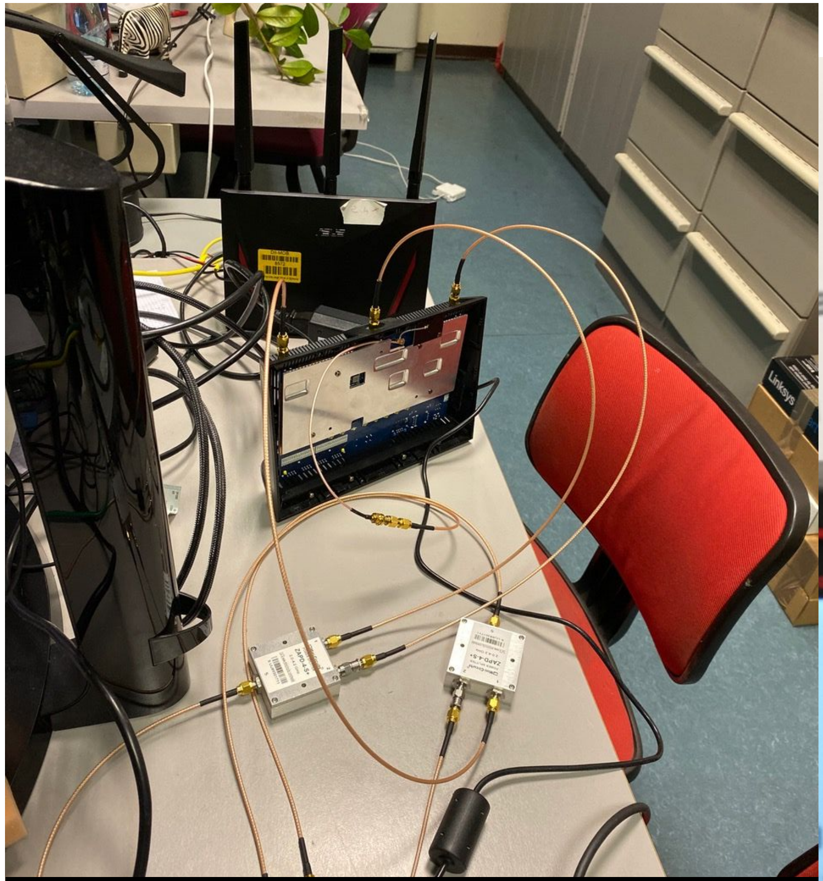
Is this just Broadcom?

- **Cypress** (who acquired Broadcom IoT and is now acquired by Infineon) also has very slow response times.
- … people told me **Qualcomm** responsible disclosure timelines are even worse.
- Luckily we didn't find something in an **Intel** CPU ;)
- And there are so many other vendors to become friends with :)

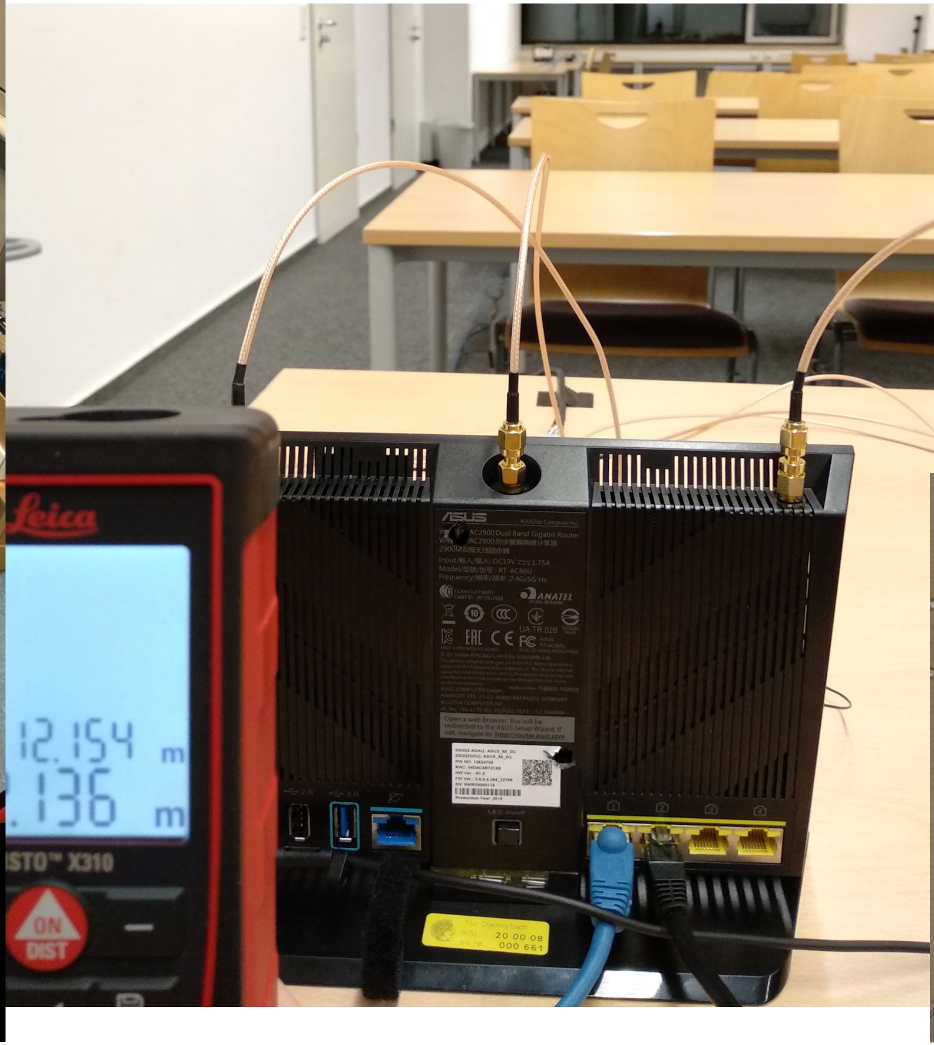


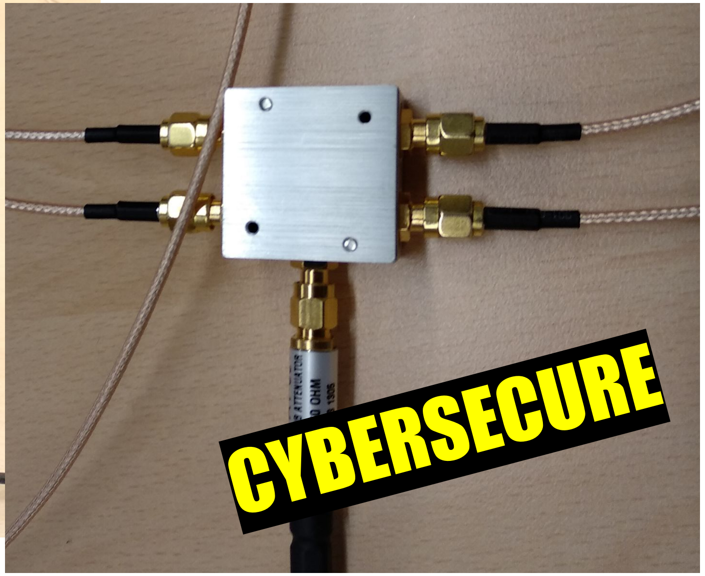
PRACTICAL SOLUTIONS

Hope: The Secure Wi-Fi Setup



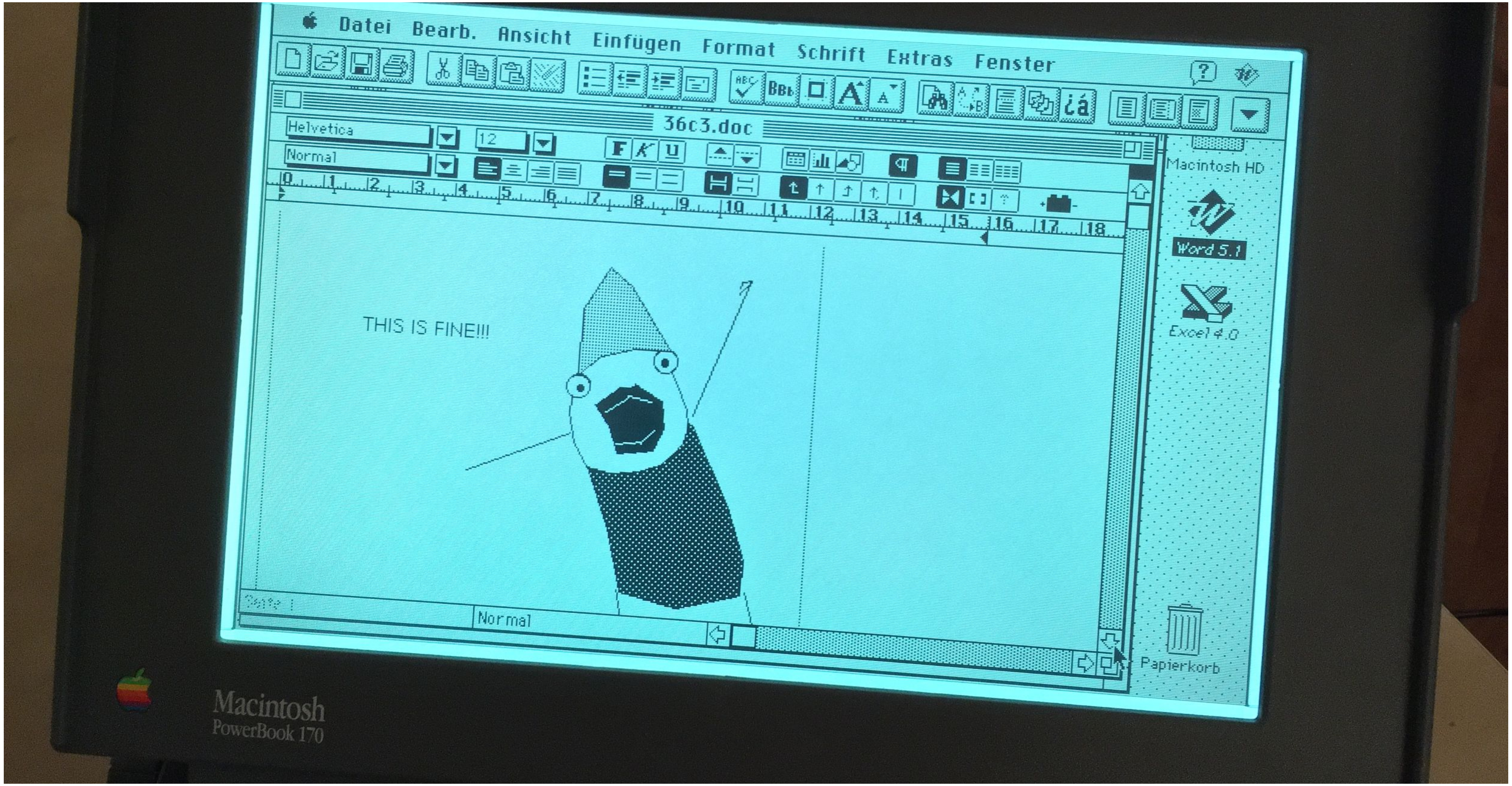
I liked really much “The Secure Wi-Fi Setup” it’s right now on my desktop :-)
-Francesco





Proudly presented by Felix Kosterhon under almost realistic lab conditions.

The Air-Gapped Device



THIS IS FINE!!!

ASK ALL THE QUESTIONS

!!! ???

Twitter: @naehrdine, @seemoolab

jiska@bluetooth.lol