THE CURRENT STATE OF FULL DISK ENCRYPTION IS STILL NOT GOOD (2025)

GPN23

20.06.2025

INTRO

slides, links, further reads & can be found online:
https://debugging.works/blog/the-current-state-of-full-disk-encryption-is-still-not-good

MOTIVATION & THREAT MODEL

- tech people: fde works, just use it
- reality: why are cops able to unlock most phones?
- cops are real threat
- expectation
  - device is encrypted
  - they have physical access
  - my data stays safe
- from a technical point of view
  - physical access (device seized)
  - unattended physical access (backdoor case)

tagesschau

Sendung verpasst?

Startseite ▸ Inland ▸ Innenpolitik ▸ BKA soll heimlich Wohnungen betreten und durchsuchen dürfen

Bundesinnenministerium

BKA soll heimlich Wohnungen durchsuchen dürfen

Stand: 14.08.2024 15:00 Uhr

Das Innenministerium will einem Bericht zufolge dem Bundeskriminalamt ermöglichen, heimlich Wohnungen zu durchsuchen. Die Befugnis soll aber nur in Ausnahmefällen erteilt werden können. Grund für den Schritt ist auch die Terrorbekämpfung.

Das Bundesinnenministerium will dem Bundeskriminalamt (BKA) die Befugnis geben, künftig heimlich Wohnungen zu betreten und zu durchsuchen. Das BKA habe eine zentrale Position in der Strafverfolgung und zur Abwehr von Gefahren des internationalen Terrorismus, dafür benötige es wirksame und moderne Instrumente in der analogen wie digitalen Welt, heißt es aus Sicherheitskreisen. Zuerst hatten die Zeitungen des Redaktionsnetzwerks (RND) Deutschland berichtet.

Der Entwurf zur Reform des BKA-Gesetzes, der dem *ARD-Hauptstadtstudio* vorliegt, umfasst "die Befugnis zum verdeckten Betreten von Wohnungen als Begleitmaßnahme für die Online-Durchsuchung und Quellen-Telekommunikationsüberwachung", also das Anbringen von Spähsoftware auf Desktops oder Smartphones, sowie die Befugnis "zur verdeckten Durchsuchung von Wohnungen".

SMARTPHONE SECURITY

Let's talk about Cellebrite.

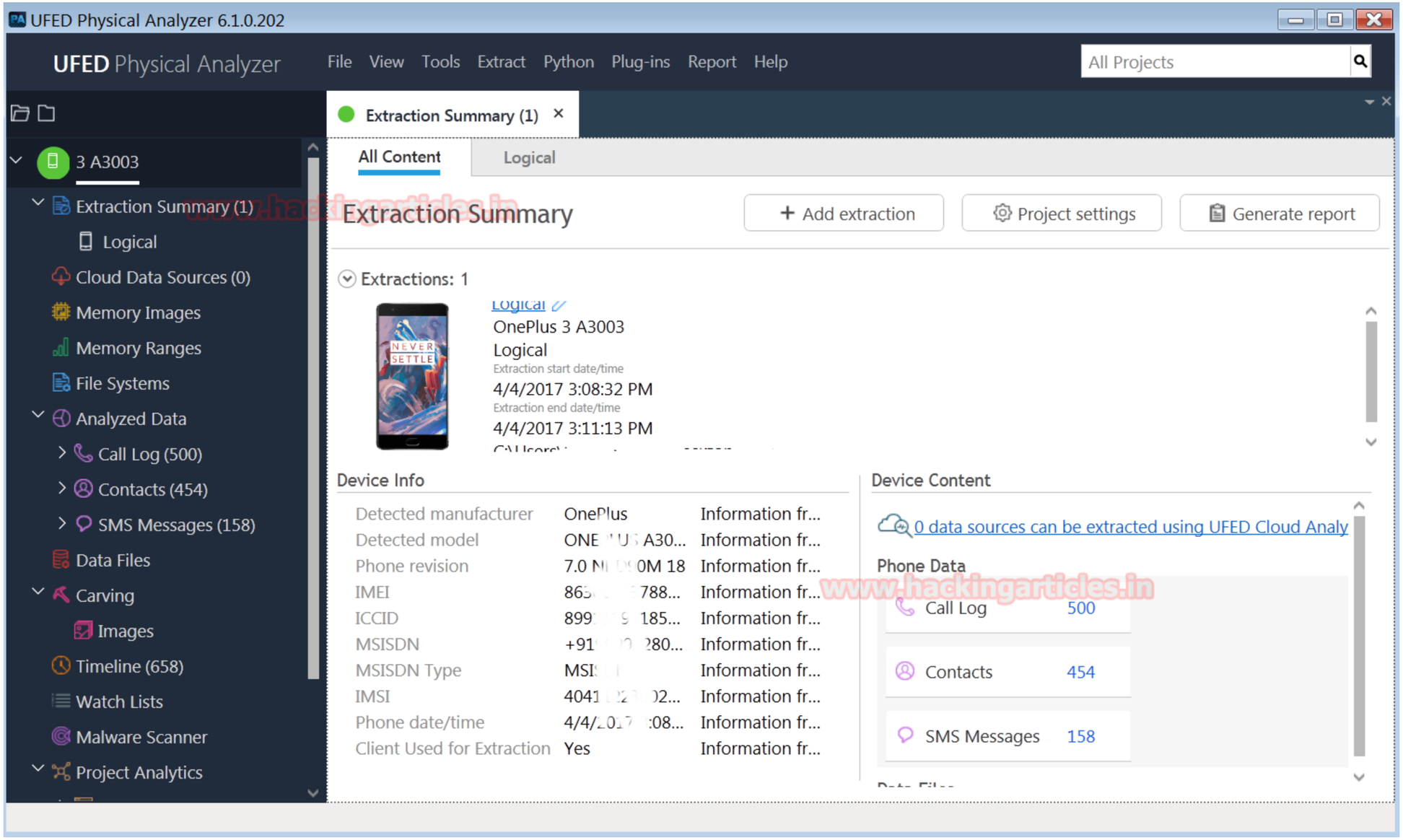


Table 1: iPhones Support Matrix 7.69.5 Locked

Newly added

iPhone	SoC	≤11	12.0-12.5.x	13.1-13.7.x	14.0-14.8.x	15-15.8.x	16.0-16.7.7	17.0 - 17.4.1	17.5-17.5.1
iPhone 5 | iPhone 5C | iPhone 5S | iPhone 6 | iPhone 6+	A6 |A7 |A8	BF	BF	N/A	N/A	N/A	N/A	N/A	N/A
iPhone 6S | iPhone 6S+ | iPhone SE gen 1 | iPhone 7 | iPhone 7+	A9 | A10	BF	Supersonic BF	Supersonic BF	Supersonic BF	Supersonic BF	N/A	N/A	N/A
iPhone 8 | iPhone 8+ | iPhone X	A11	BF	Supersonic BF	Supersonic BF	Supersonic BF	Supersonic BF	AFU + Supersonic BF (1)	N/A	N/A
iPhone XR | iPhone XS max | iPhone XS	A12	N/A	BF	Supersonic BF	Supersonic BF	Supersonic BF	AFU + Supersonic BF (1)	AFU + Supersonic BF (1)(3)	AFU + Supersonic BF (1) (3)
iPhone 11 | iPhone 11 pro | iPhone 11 pro max | iPhone SE gen 2	A13	N/A	N/A	Supersonic BF	Supersonic BF	Supersonic BF	AFU + Supersonic BF (1)(4)	AFU + Supersonic BF (1)(3)	AFU + Supersonic BF (1) (3)
iPhone 12 | iPhone 12 pro | iPhone 12 pro max| iPhone 12 mini	A14	N/A	N/A	N/A	Not supported	AFU (2)	AFU	AFU	AFU
iPhone 13 | iPhone 13 pro | iPhone 13 pro max | iPhone 13 mini | iPhone SE gen 3	A15	N/A	N/A	N/A	N/A	AFU (2)	AFU(4)	AFU	AFU
iPhone 14 | iPhone 14 Plus | iPhone 14 pro | iPhone 14 pro max	A16 | A15	N/A	N/A	N/A	N/A	N/A	AFU	AFU	AFU
iPhone 15 | iPhone 15 Pro | iPhone Pro Max	A16 |A17	N/A	N/A	N/A	N/A	N/A	N/A	Available in CAS	Available in CAS

Table 2: Android OS Access Support Matrix – Locked devices **7.73.1**

Vendor (Chipset)		Section 1: COLD - turned off (Secure startup or FBE)		Section 2: HOT (AFU or FDE without secure startup)		Comments and exceptions
		BFU extractions (for FBE devices)	Brute-Force Password to get the user data (CE) decrypted	All Extractions (Even without BF)	Brute-Force password (not needed for extraction)	
Samsung (Exynos / MTK / Qualcomm)	**Android 6**	✗	✗	✓	☑	Added support for *Galaxy* `M55s,M05,F05` .
	Android 7-14	✓	✓	✓	✓	
Huawei (Kirin / Qualcomm / MTK)		✓	☑	✓	✓	P40 family is supported for Brute-force only up to ~04-2021 SPL
Pixel	**Pixel, Pixel XL**	✓	✓	✓	✓	
	Pixel 3 - 5	✓	✓	✓	✓	
	Pixel 6 - 9	✓	✗	✓	✗	
Non-Samsung Qualcomm including Huawei, LG, Motorola, Xiaomi, Sony, OnePlus and many more		✓	✓	✓	✓	Added AFU, BFU & BF support for Snapdragon 8 Elite

✓ Fully Supported

☑ Partially Supported

✗ Not Supported

Huawei Kirin BF temporarily disabled.

Table 2: Android OS Access Support Matrix – Locked devices (cont.) **7.73.1**

Vendor (Chipset)		Section 1: COLD - turned off (Secure startup or FBE)		Section 2: HOT (AFU or FDE without secure startup)		More Info
		BFU extractions (for FBE devices)	Brute-Force Password and then All Extractions	All Extractions (Even without BF)	Brute-Force password (not needed for extraction)	
Non-Samsung MTK	Xiaomi, Huawei, LG, Motorola, ...	✓	✓	✓	✓	Added BF for many Xiaomi MTKs: Redmi A3/A3+, Redmi Note 13 5G and more.
	Vivo, Oppo, Realme, OnePlus, Tecno, Infinix	✓	✓	✓	✓	Most of these devices are supported. Added AFU & BFU support for `Dimensity 9400`
Non-Samsung Exynos Example Motorola, Vivo, …		✓	✓	✓	✓	
Tracfone brand of LG, etc… Spreadtrum based devices Non-Android devices MDM devices						**Samsung MDM partially supported in CAS**
Unisoc devices		✓	✓	✓	✓	Added AFU & BFU support for Unisoc T760, T820

Table 3: Android OS Access Support Matrix – Google Pixel 1-5 | **7.73.1**

Model / State	Standard Android OS, BFU	Standard Android OS, AFU	Standard Android OS, Unlocked	GrapheneOS, BFU *	GrapheneOS, AFU *	GrapheneOS, Unlocked
Pixel / Pixel XL	BFU Yes BF yes, no SPL limitation	FFS Yes BF Yes, no SPL limitation	FFS Yes	NA	NA	NA
Pixel 2 / Pixel 2XL	BFU Yes BF No	FFS Yes BF No	FFS Yes	NA	NA	NA
Pixel 3 / Pixel 3XL / Pixel 3a	BFU Yes BF Yes, no SPL limitation	FFS Yes BF Yes, no SPL limitation	FFS Yes	BFU Yes, up to late 2022 SPL BF Yes, up to late 2022 SPL	FFS Yes, up to late 2022 SPL BF Yes, up to late 2022 SPL	FFS Yes
Pixel 4 / Pixel 4XL / Pixel 4a	BFU Yes BF Yes, no SPL limitation	FFS Yes BF Yes, no SPL limitation	FFS Yes	BFU Yes, up to late 2022 SPL BF Yes, up to late 2022 SPL	FFS Yes, up to late 2022 SPL BF Yes, up to mid 2022 SPL	FFS Yes
Pixel 5 / Pixel 5XL / Pixel 5a	BFU Yes BF Yes, no SPL limitation	FFS Yes BF Yes, no SPL limitation	FFS Yes	BFU Yes, up to late 2022 SPL BF Yes, up to late 2022 SPL	FFS Yes, up to late 2022 SPL BF Yes, up to late 2022 SPL	FFS Yes

Table 3: Android OS Access Support Matrix – Google Pixel **7.73.1**

Model / State	Standard Android OS, BFU	Standard Android OS, AFU	Standard Android OS, Unlocked	GrapheneOS, BFU *	GrapheneOS, AFU *	GrapheneOS, Unlocked
Pixel 6 / Pixel 6 Pro / Pixel 6a	BFU Yes BF No	FFS Yes BF No	FFS Yes	BFU Yes, up to late 2022 SPL BF No	FFS Yes, up to late 2022 SPL BF No	FFS Yes
Pixel 7 / Pixel 7 Pro / Pixel 7a / Pixel Tablet / Pixel Fold	BFU Yes BF No	FFS Yes BF No	FFS Yes	BFU Yes, up to late 2022 SPL BF No	FFS Yes, up to late 2022 SPL BF No	FFS Yes
Pixel 8 / Pixel 8a / Pixel 8 Pro	BFU Yes BF No	FFS Yes BF No	FFS Yes	BFU Yes, up to late 2022 SPL BF No	FFS Yes, up to late 2022 SPL BF No	FFS Yes
Pixel 9 / Pixel 9 Pro / Pixel 9 Pro XL / Pixel 9 Pro Fold	BFU Yes BF No	FFS Yes BF No	FFS Yes	BFU No BF No	FFS No BF No	FFS Yes

GRAPHENEOS AUTO REBOOT

Auto reboot

GrapheneOS provides an auto-reboot feature which reboots locked devices after a set period of time to put data at rest. A countdown timer is started each time the device is locked, and the device will reboot if a successful unlock doesn't occur before the timer reaches zero. Unlocking any profile cancels the timer, not just the Owner profile.

The timer is set to 18 hours by default, but can be set to values between 10 minutes and 72 hours, or turned off.

This feature doesn't apply when the device is in "Before First Unlock" state, meaning that it will not lead to the device continuously rebooting, as data is already at rest.

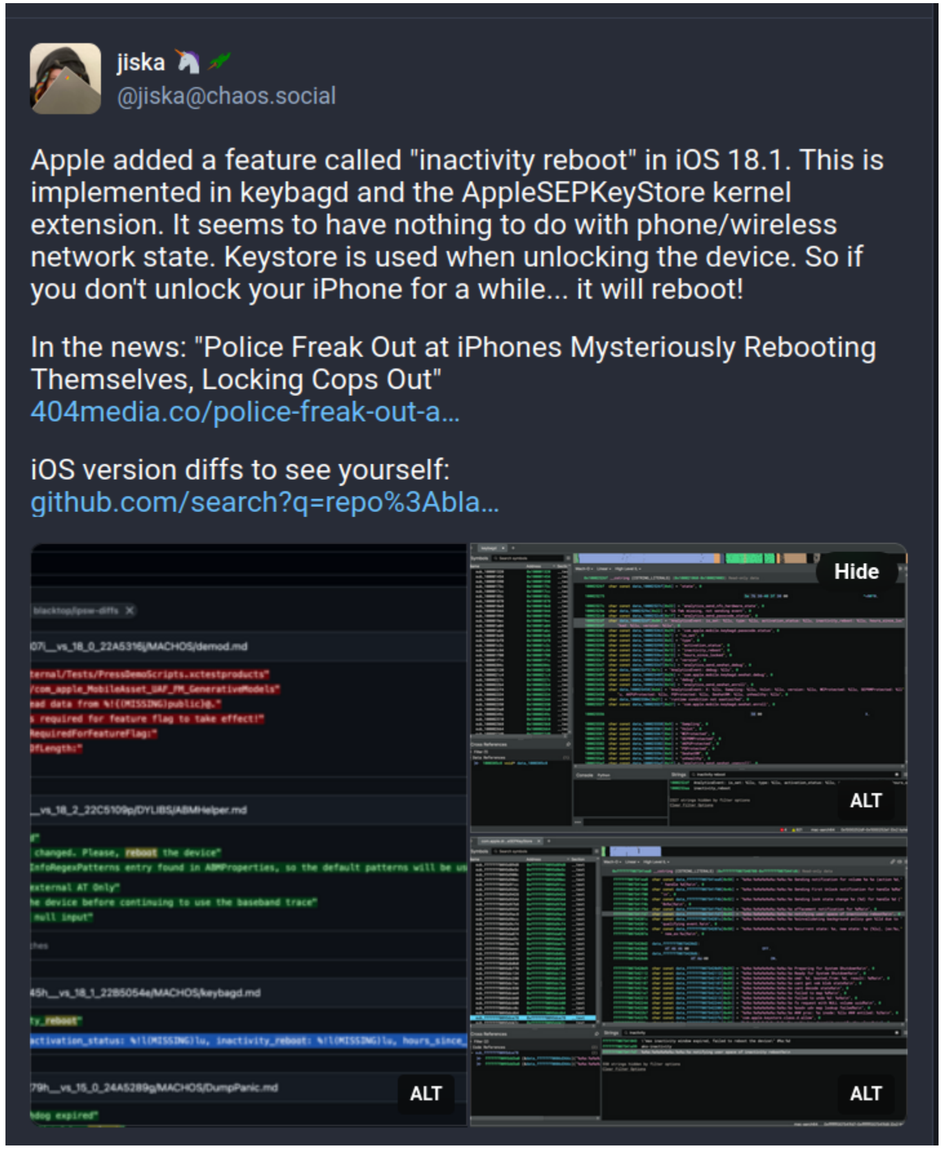
The feature is implemented in the init process, preventing it from being bypassed through system process crashes since an init crash causes a kernel panic which leads to a reboot.

Table 1: iPhones Support Matrix 7.69.5 Locked

Newly added

iPhone	SoC	≤11	12.0-12.5.x	13.1-13.7.x	14.0-14.8.x	15-15.8.x	16.0-16.7.7	17.0 - 17.4.1	17.5-17.5.1
iPhone 5 | iPhone 5C | iPhone 5S | iPhone 6 | iPhone 6+	A6 |A7 |A8	BF	BF	N/A	N/A	N/A	N/A	N/A	N/A
iPhone 6S | iPhone 6S+ | iPhone SE gen 1 | iPhone 7 | iPhone 7+	A9 | A10	BF	Supersonic BF	Supersonic BF	Supersonic BF	Supersonic BF	N/A	N/A	N/A
iPhone 8 | iPhone 8+ | iPhone X	A11	BF	Supersonic BF	Supersonic BF	Supersonic BF	Supersonic BF	AFU + Supersonic BF (1)	N/A	N/A
iPhone XR | iPhone XS max | iPhone XS	A12	N/A	BF	Supersonic BF	Supersonic BF	Supersonic BF	AFU + Supersonic BF (1)	AFU + Supersonic BF (1)(3)	AFU + Supersonic BF (1) (3)
iPhone 11 | iPhone 11 pro | iPhone 11 pro max | iPhone SE gen 2	A13	N/A	N/A	Supersonic BF	Supersonic BF	Supersonic BF	AFU + Supersonic BF (1)(4)	AFU + Supersonic BF (1)(3)	AFU + Supersonic BF (1) (3)
iPhone 12 | iPhone 12 pro | iPhone 12 pro max| iPhone 12 mini	A14	N/A	N/A	N/A	Not supported	AFU (2)	AFU	AFU	AFU
iPhone 13 | iPhone 13 pro | iPhone 13 pro max | iPhone 13 mini | iPhone SE gen 3	A15	N/A	N/A	N/A	N/A	AFU (2)	AFU(4)	AFU	AFU
iPhone 14 | iPhone 14 Plus | iPhone 14 pro | iPhone 14 pro max	A16 | A15	N/A	N/A	N/A	N/A	N/A	AFU	AFU	AFU
iPhone 15 | iPhone 15 Pro | iPhone Pro Max	A16 |A17	N/A	N/A	N/A	N/A	N/A	N/A	Available in CAS	Available in CAS

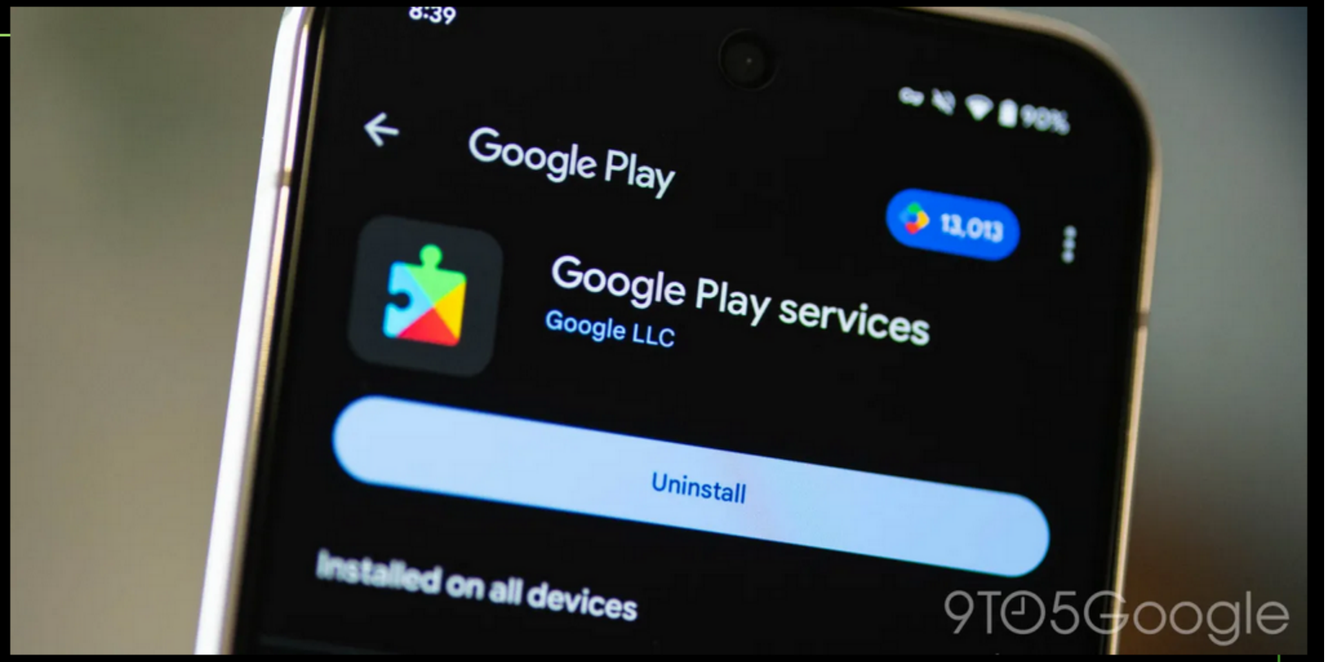
APPLE: WE CAN ALSO DO THAT



ANDROID: MAYBE WE SHOULD HAVE THIS, TOO

[Update: Optional] Google rolling out auto-restart security feature to Android

Abner Li | Apr 16 2025 - 8:24 am PT 25 Comments



LINUX: I WANT TO HAVE THIS, TOO!

- GrapheneOS's auto reboot feature for Linux laptops
- https://debugging.works/blog/grapheneos-auto-reboot-feature-for-linux/

Das Mobiltelefon [redacted] (IMEI: [redacted] besitzt eine Gerätesperre in Form eines Musters. Dieses ist nicht bekannt. Eine Entsperrung des Mobiltelefons ist Voraussetzung für die Sicherung der auf dem Mobiltelefon befindlichen Daten. Zwecks Ermittlung des Sperrcodes wurde das Mobiltelefon mit der hier verwendeten Auswerteeinheit UFED Premium verbunden und eine sog. BFU Brute-Force-Attacke initiiert. Der hieraus resultierende Gerätecode lautet „1458". Daraufhin konnte eine logische Sicherung der auf dem Telefon befindlichen Daten erzeugt werden. Das hieraus resultierende Abbild des Speichers wurde zwecks Decodierung in die forensische Auswertesoftware Cellebrite UFED importiert und die Berichtsdatei „[redacted].ufdr", die mit dem Reader geöffnet werden muss, im Ordner „UFED Bericht" gespeichert.

CHIP-OFF

No video with supported format and
0:00 / 0:00


minigrip
AST

BUDAPEST KOMPLEX

3. Majas Festnahme: Telefonüberwachung einer Maja nahestehenden Person. Diese soll am Telefon zu einer dritten Person gesagt haben sie treffe sich bald mit Maja in Berlin. Daraufhin Observation, Verfolgung bis nach Berlin, dort nimmt das MEK Maja in einem Hotel fest.
4. J. wurde im Regio festgenommen. Dabei hatte er einen Schlüssel und eine Stempelkarte eines Cafés bei sich. Mathe und Co. probierten den Schlüssel systematisch in den Straßen um das Café aus und konnten so Js Wohnung finden.
5. Alle Smartphones mit Graphene OS Betriebssystem konnten bis zum gegenwärtigen Zeitpunkt technisch nicht ausgelesen werden.

https://alleantifa.noblogs.org/post/2025/05/26/21-05-2025-bericht-vom-17-prozesstag/

INTERIM CONCLUSION

- "fde just works"?
- not the case for smartphones

LET'S TALK ABOUT LINUX

Disk setup

How do you want to install Ubuntu?

Erase disk and install Ubuntu
Start from scratch on your selected disk.

Advanced features... None selected

Manual installation
For advanced users seeking customized disk setups.

Back

Next

Ready to install

Review your choices

General

Disk setup	Erase disk and install Ubuntu
Installation disk	VBOX HARDDISK **sda**
Applications	Default selection

Security & more

Disk encryption	None
Proprietary software	None

Partitions

partition **sda1** formatted as **fat32** used for **/boot/efi**
partition **sda2** formatted as **ext4** used for **/**

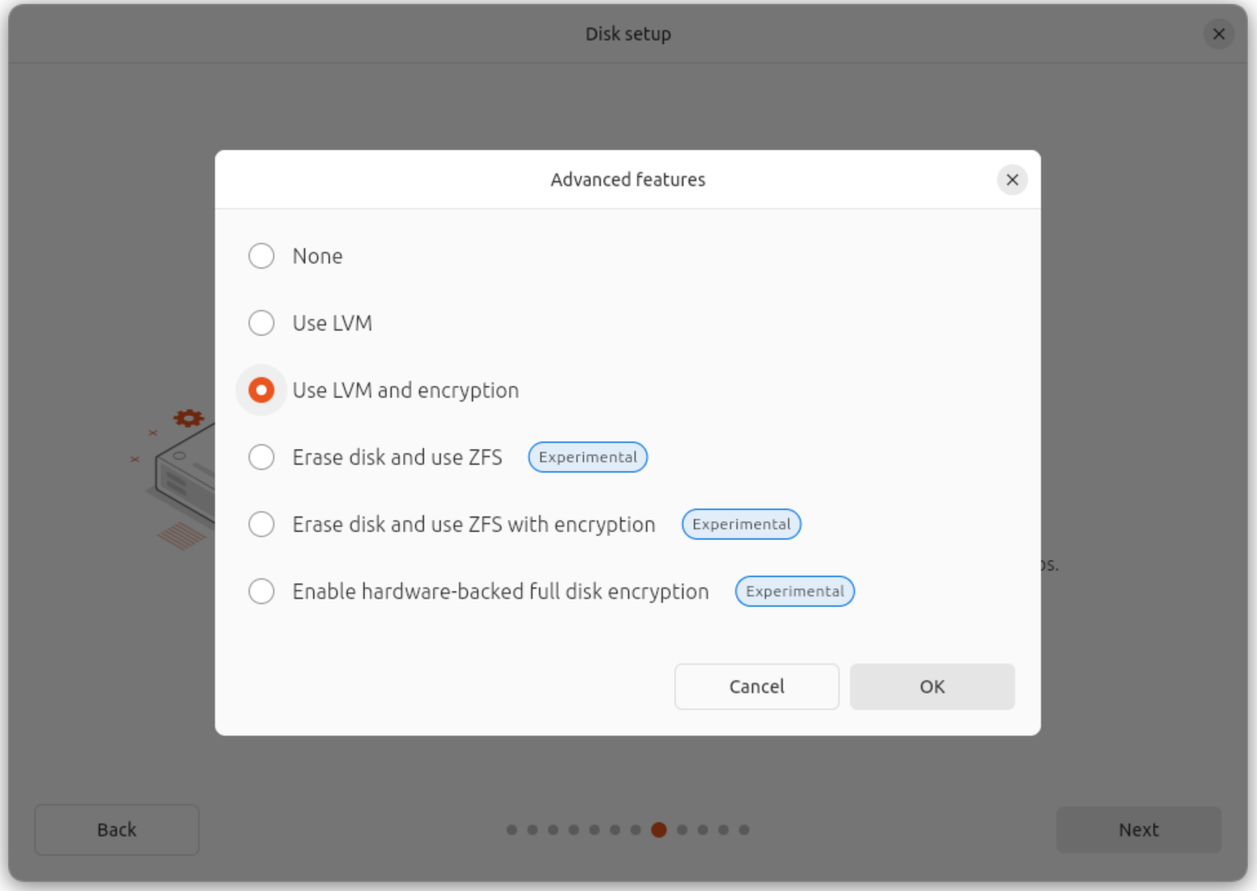
Back

Install

ENCRYPT EXISTING LINUX SYSTEM

1. Boot Linux from USB
2. `resize2fs -p /dev/nvme0n1p7 511968M`
3. `cryptsetup reencrypt --encrypt --reduce-device-size 16M /dev/nvme0n1p7`

https://srijan.ch/encrypting-an-existing-linux-systems-root-partition


Disk setup
Advanced features
None
Use LVM
Use LVM and encryption
Erase disk and use ZFS
Experimental
Erase disk and use ZFS with encryption
Experimental
Enable hardware-backed full disk encryption
Experimental
Cancel
OK
Back
Next

Ready to install

Review your choices

General

Disk setup	Erase disk and install Ubuntu
Installation disk	VBOX HARDDISK **sda**
Applications	Default selection

Security & more

Disk encryption	LUKS (LVM)
Proprietary software	None

Partitions

partition **sda1** formatted as **fat32** used for **/boot/efi**
partition **sda2** formatted as **ext4** used for **/boot**
partition **sda3** created

Back

Install

BOOT PROCESS W/ UNENCRYPTED /BOOT

1. UEFI finds a partition of type EFI System
2. UEFI understands fat file system
3. UEFI can run efi applications
    - for example boot loader
    - /boot/efi/EFI/ubuntu/grubx64.efi
    - which one? EFI variables
4. grubx64.efi mounts unencrypted /boot
5. and runs kernel + initrd
6. initrd: decrypt disk, mount it and run init system

DEMO: BACKDOORING INITRD

NEXT STEP: ENCRYPTED /BOOT

BOOT PROCESS W/ ENCRYPTED /BOOT

- encrypted /boot is supported by grub
- not supported by Ubuntu installer
- what changes:
  - initrd and kernel are protected/encrypted
  - decryption of /boot takes place in grubx64.efi
- but: grubx64.efi can still be backdoored
- problem remains: code needs to be verified
- way harder to implement

POC - EVIL MAID ON ENCRYPTED /BOOT

(BIOS boot, not UEFI)

https://media.ccc.de/v/gpn20-32-poc-implementing-evil-maid-attack-on-encrypted-boot

https://github.com/kmille/evil-maid-attack-on-encrypted-boot

SECURE BOOT

1. Boot into UEFI firmware
    1. enable Secure Boot
    2. put it into "Setup Mode"
    3. set a password
2. Boot into your Linux system, then
    1. Create CA: `sbctl create-keys`
    2. Enroll CA: `sbctl enroll-keys`
    3. Sign your efi application: `sbctl sign /boot/efi/EFI/ubuntu/grubx64.efi`

SECURE BOOT

- now we are in a good state:
  - data is encrypted
  - integrity of (unencrypted) code is verified
- good: only trust your CA
- good: hard fail
- bad: unnoticed when disabled
  - bypass firmware password
  - change firmware settings with programmer

COLD BOOT ATTACK - BASICS

- well known for a very long time
- full disk encryption key is always stored in memory
  - even if system is suspended
- general assumption: remove power supply, memory loses its content
- but: it keeps its data for some time
  - even longer when the RAM bar is cooled (up to a few seconds)

COLD BOOT ATTACK - HOW IT WORKS

1. Freeze the RAM bar with a cool spray
2. Reset the device: Remove the power supply for a short time
3. Boot from USB stick
4. Read the whole memory and extract the AES master key to decrypt the whole disk

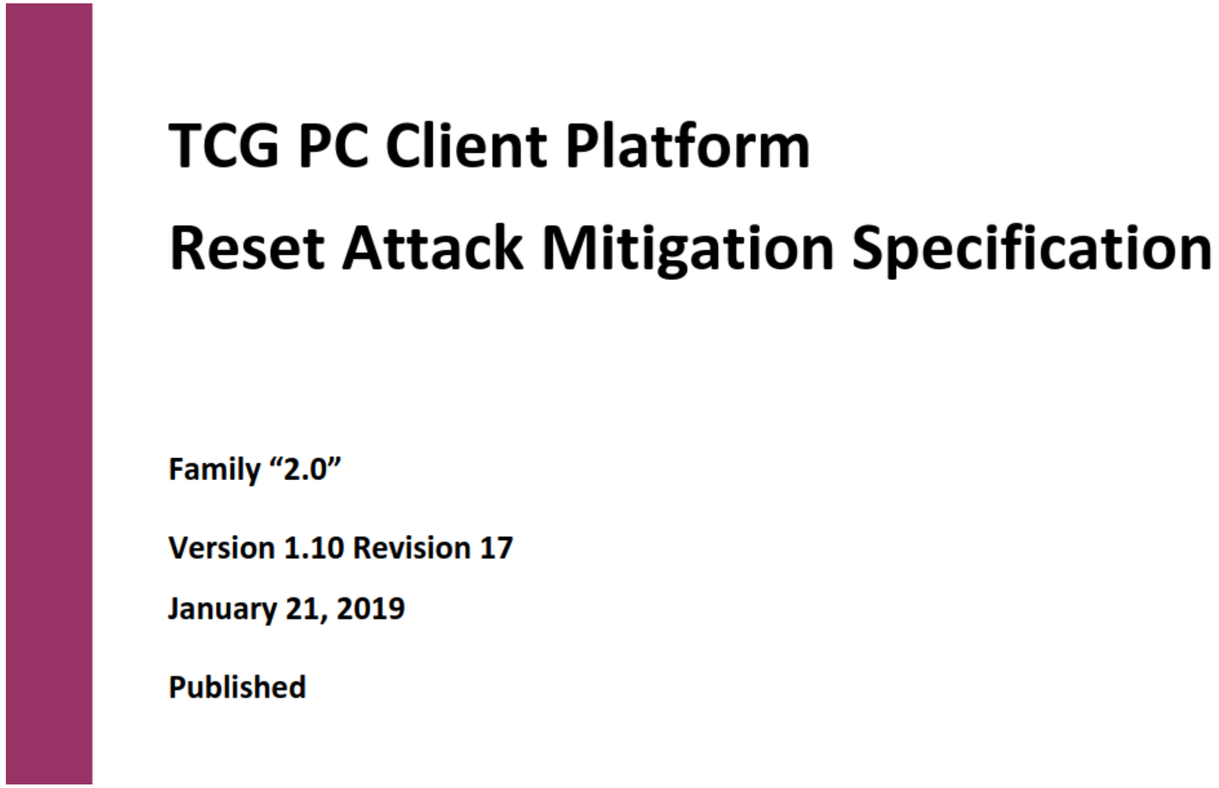
DEMO: COLD BOOT ATTACK (2008)

No video with supported format and

0:00 / 0:00

COLD BOOT ATTACK - MITIGATION

- hardware mitigation: Reset Attack Mitigation (2018)
- introduced the MOR bit (Memory Overwrite Request)



MOR BIT - HOW DOES IT WORK?

1. On a clean OS shutdown, the OS wipes the memory
2. OS tells the firmware: memory was cleared
3. firmware sets MOR bit
4. Next boot
    - firmware checks MOR bit
    - if memory was not cleared before, firmware clears the RAM
    - then, it boots the system

BYPASSING MOR BIT

- In the same year (2018), F-Secure was able to bypass the Platform Reset Attack Mitigation
- They used a programmer to change the NVRAM before booting from USB

DEMO: BYPASSING MOR BIT

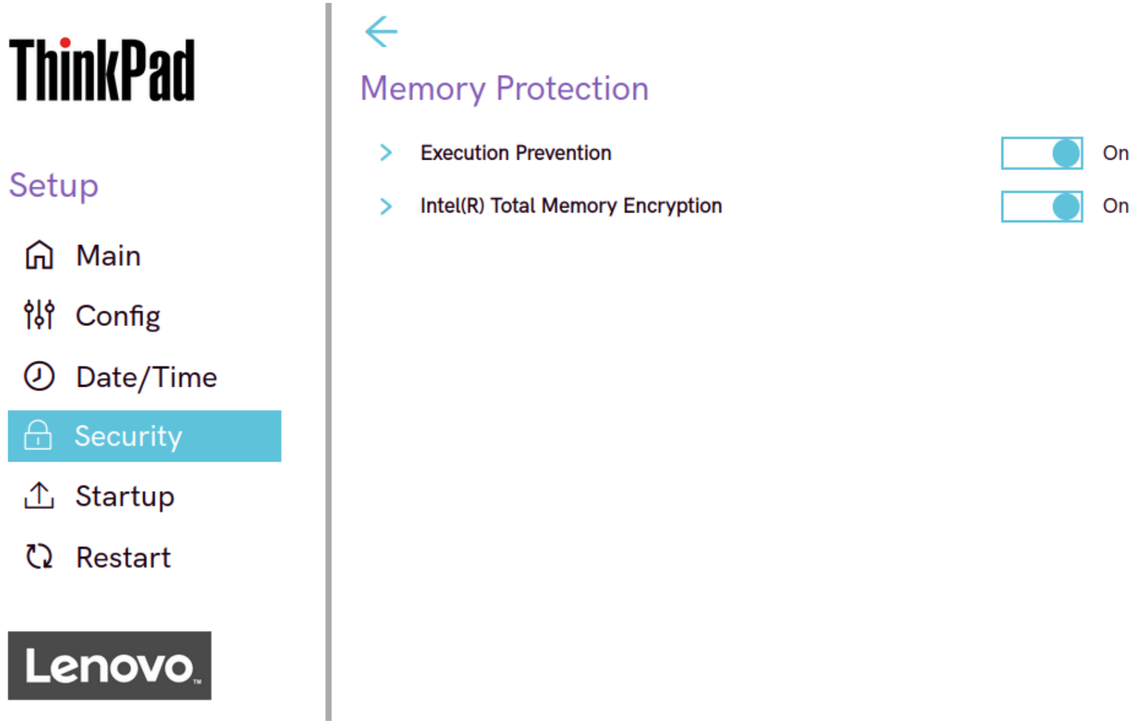
No video with supported format and
0:00 / 0:00

SOME NOT SO GREAT MITIGATIONS

- don't left your device unattended
- use Secure Boot with your own CA
  - attacker can remove the RAM and put it own device
    - use a device with soldered RAM
- firmware config hardening
  - password required to boot system in general
  - password required to change boot order
  - password prevents attacker from disabling Secure Boot
  - but: attacker can change these values with a programmer

COLD BOOT ATTACK: REAL MITIGATIONS

- turn off your device when your are on the way/go to bed
- don't use suspend, use hibernate instead
- hardware memory encryption



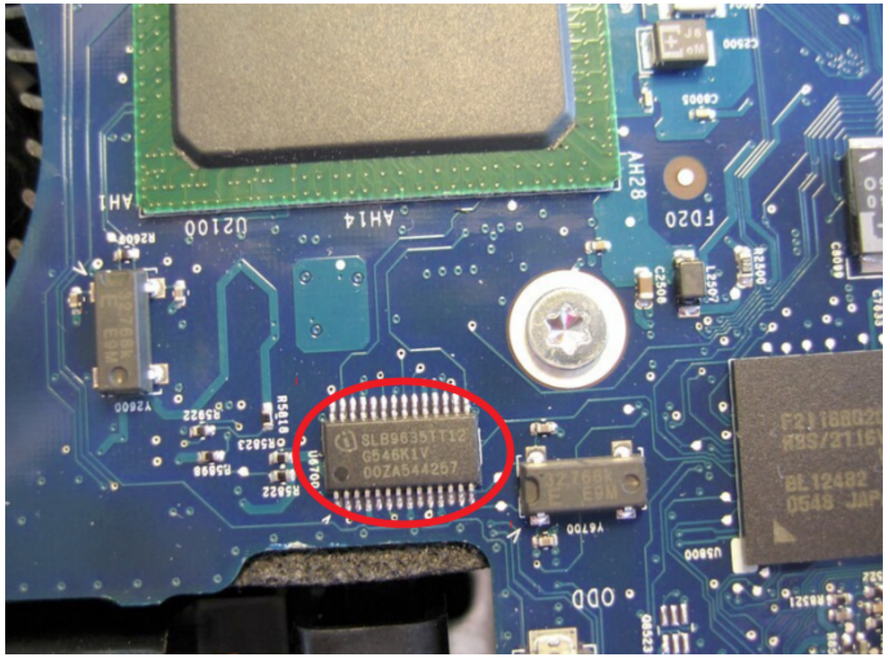
MITIGATION: BUSKILL

No video with supported format and
0:00 / 0:00

COLD BOOT ATTACK: SUMMARY

- I do not know about latest research (high density memory DDR5/ECC)
- not a physical problem
- you have to
  - bypass MOR bit with a programmer
  - or put the RAM in a different device
- you only have a single try
- cops perform cold boot attacks (though not as default)
  - maybe they are quite happy with Cellebrite + Smartphones

LET'S TALK ABOUT TPMS



https://debugging.works/blog/tpm-explained/

WHAT IS A TPM?

- Trusted Platform Module
- dedicated security chip
- Windows requirement
  - almost all devices have one
- version 2.0
- functions
  - secure random number generator
  - private key storage, e.g. it can hold you ssh keys
- passive device

TPM FOR FDE

- fde key stored inside the TPM
- TPM can be protected by a PIN (can be alphanumeric)
- as a backup, a recovery key should be used
- TPM uses a single luks key slot
- During boot, user enter's the correct PIN to decrypt the disk
- TPM implements lockout mechanism, brute forcing is not possible

TPM FOR FDE

- disk gets automatically decrypted during boot (like on Windows)
- but only if the device was not modified
- during boot, each component is hashed and verified (PCR)
- also known as Measured Boot
- optionally: protect the TPM with a PIN
  - so called pre-boot authentication

PCR

- Platform Control Register
- used to check integrity of the system
- 24 PCR registers (0-23)
- each register
  - holds a sha256 value
  - is initialized with zeros
  - can not be resetted
  - has a dedicated purpose
- data can only be extended

HOW ARE COMPONENTS MEASURED?

- each boot component measures the next component
- measuring: hash config/code into a dedicated PCR
- firmware executable code is hashed in PCR 0
- firmware configuration (UEFI settings) is hashed in PCR 1
- firmware hashes Secure Boot state and the trusted CA certificates in PCR 7
- Linux kernel measures all initrds it receives to PCR 9
- boot loader measures the kernel command line in PCR 12
- If a UKI is used, systemd-stub measures the UKI in PCR 11
- systemd-pcrphase measures boot phase strings like “enter-initrd” in PCR 11
- docs: TPM2 PCR Measurements Made by systemd

LINUX HANDS-ON #1

How to use a TPM for FDE?
commands and output:
https://gist.github.com/kmille/1bc2e4b84adac13f4cc529e9f0b6391a

WHICH PCRS SHOULD I USE?

- the more the better, as the attack surface shrinks
- but what if they change? Re-enroll TPM
- in practice
  - PCR 7 (Secure Boot): always constant
  - PCR 0 (firmware updates): rarely
  - PCR 1 (firmware configuration): rarely
  - Kernel/initrd: often

WHAT IS A SIGNED PCR POLICY?

- two technical ways to use a PCR:
  - by comparing PCR values (current and the previous ones when enrolled)
  - the other mechanism works like this:
    1. You once generate an asymmetric key pair
    2. When enrolling the TPM, you bind the public key to the TPM
    3. The UKI is signed with the key bound to the TPM
    4. Update the kernel => UKI is re-generated and signed automatically

LINUX HANDS-ON #2

How to use a signed PCR policy

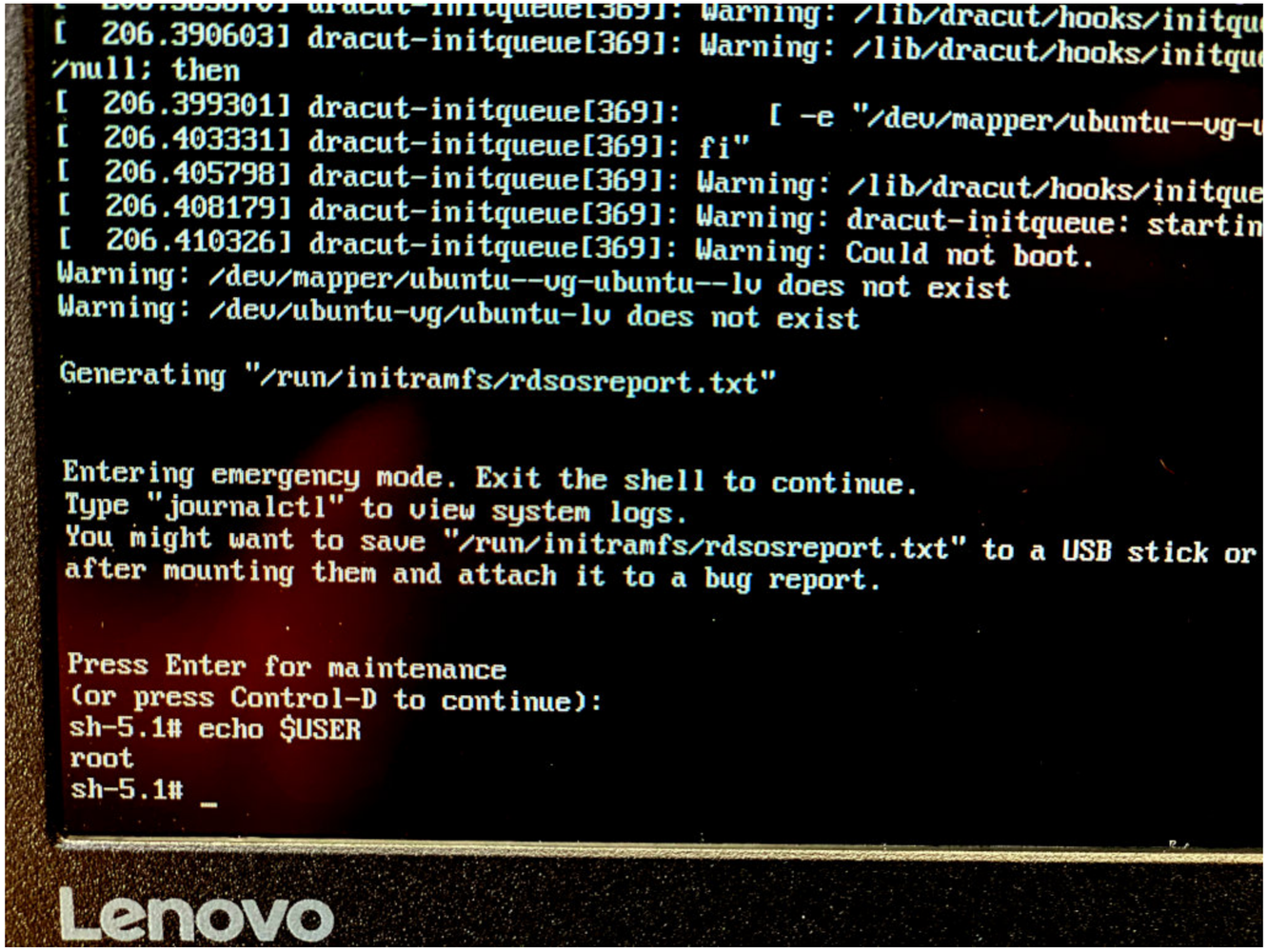
WHAT CAN GO WRONG?

- a lot when
  - not all relevant boot components are measured
  - especially when not using pre-boot authentication
- worst case: decrypt the disk and get a root shell

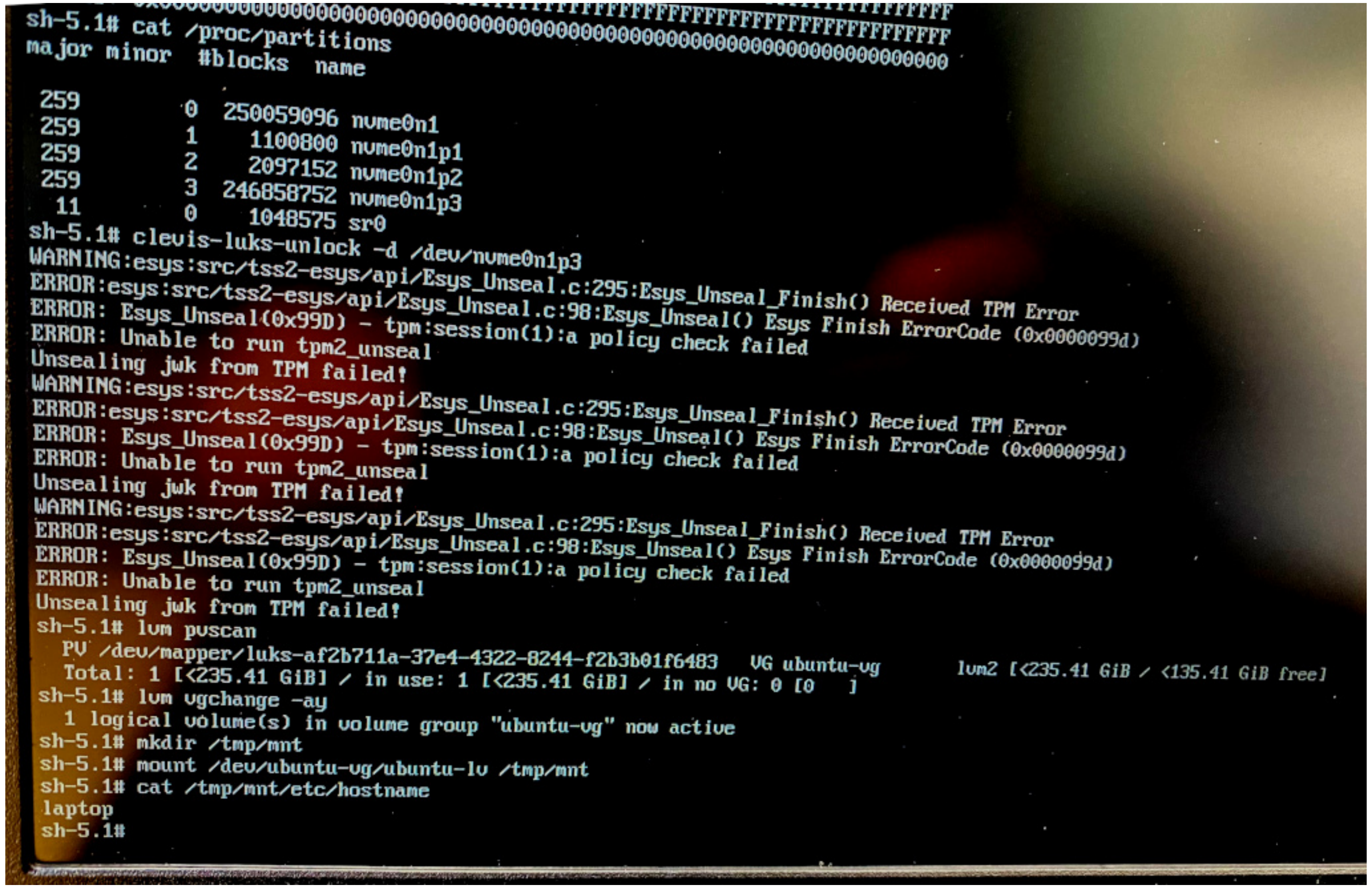
RECOVERY SHELL

- https://pulsesecurity.co.nz/advisories/tpm-luks-bypass
- used a microcontroller to simulate a keyboard
- used to hammer the "enter" key
- suddenly, a recovery shell popped up

RECOVERY SHELL



RECOVERY SHELL



KERNEL PARAMETERS

kernel command line not measured?

1. Edit parameters: `init=/bin/bash`
2. Get a root shell
3. Just use initrd's tools to decrypt the disk
4. Profit

PCR 9 - INITRD

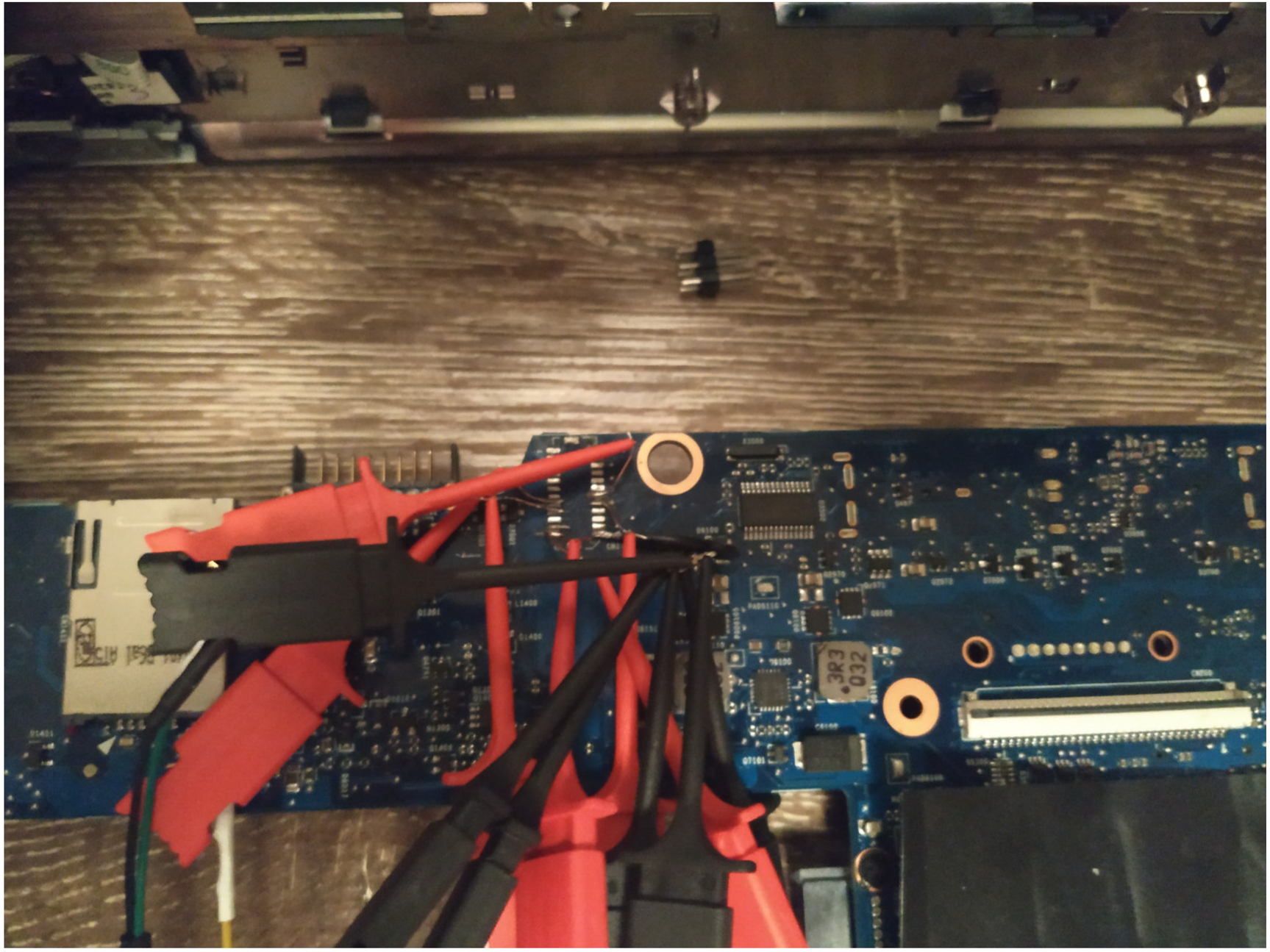
- https://blog.securityinnovation.com/preventing-initramfs-attacks-tpm
- in this setup:
  - initrd was not measured
  - initrd stored in unencrypted /boot
  - Secure Boot is enabled
    - initrd is not verified by Secure Boot
  - UKI was not used
- attack: just modify initrd
  - pop a shell, unlock disk & profit

TPMS & COLD BOOT ATTACKS

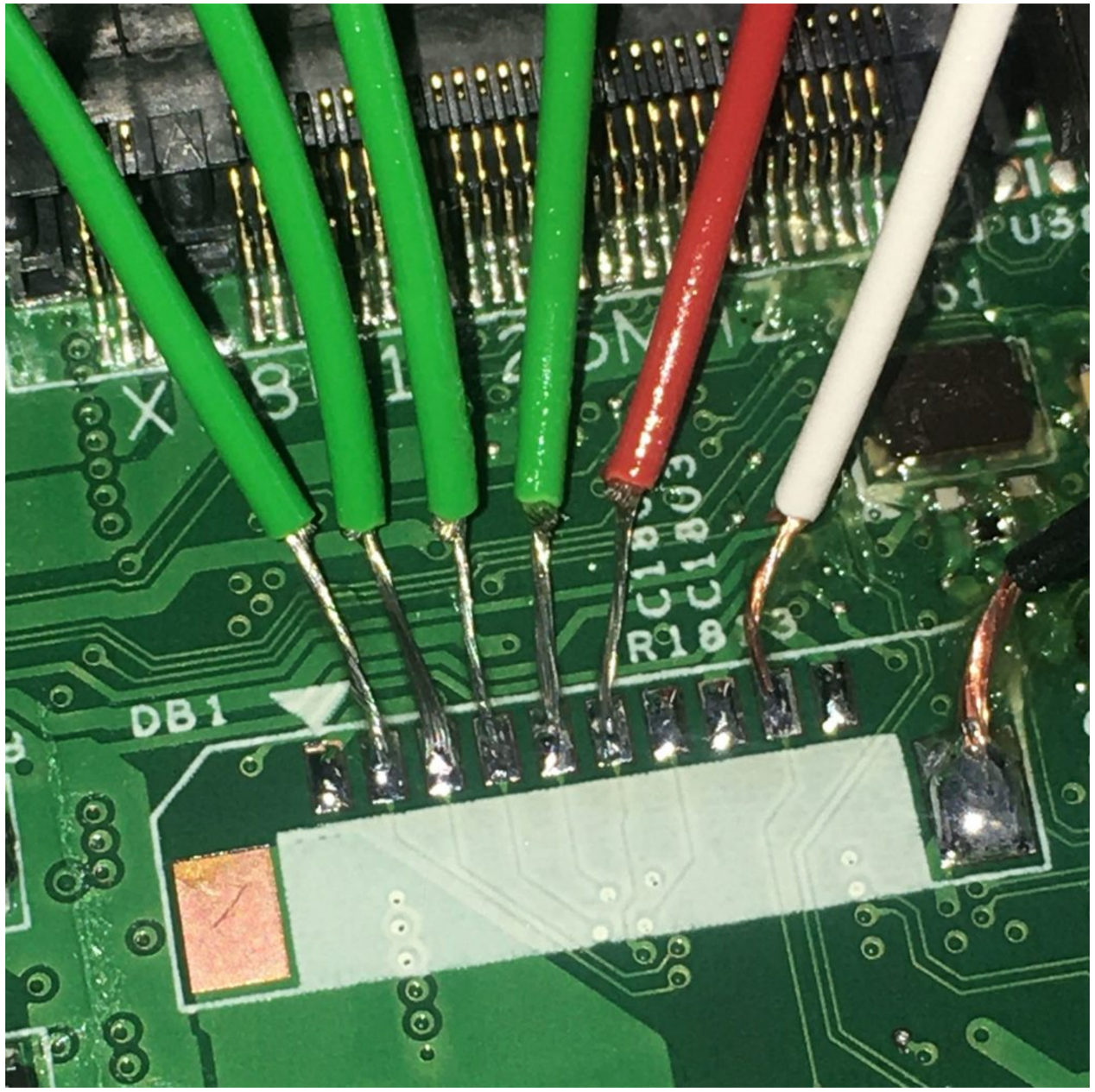
TPM SNIFFING

- During decryption, CPU asks the TPM for the key
- TPM checks integrity and sends key back to the CPU
- The traffic on the bus can be sniffed by an attacker
- Without pre-boot authentication:
  - unlimited number of tries for the attacker

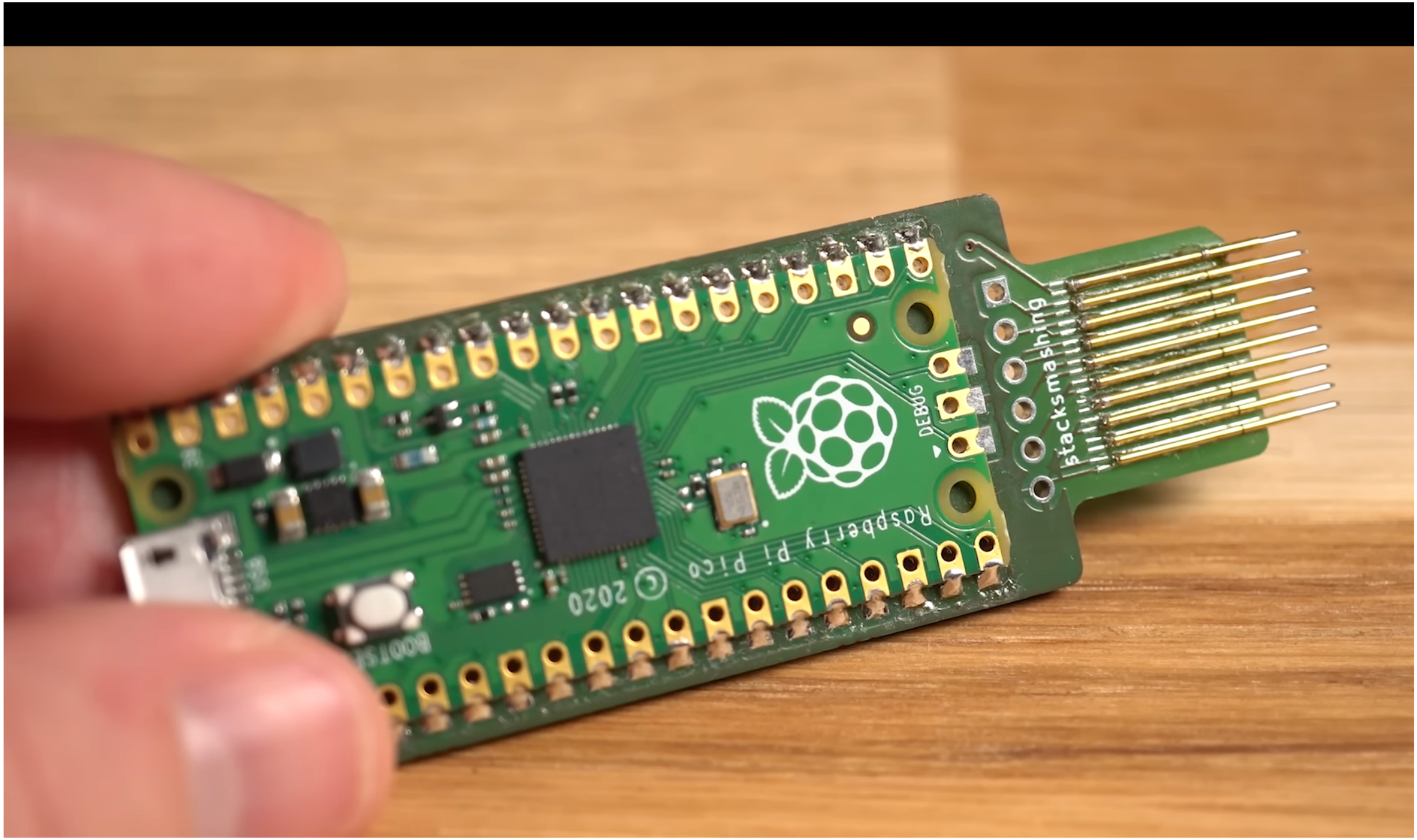
TPM SNIFFING

- 

TPM SNIFFING

- 
DB1

TPM SNIFFING - STACKSMASHING (FEB 2024)


DEBUG
stacksmashing
Raspberry Pi Pico
© 2020

TPM SNIFFING - DEMO

No video with supported format and

0:00 / 0:00

TPM SNIFFING - MITIGATION

- firmware TPMs (fTPM)
  - TPM embedded inside the CPU
  - Intel PTT (Intel Platform Trust Technology)
  - “AMD Firmware TPM”
- TPM traffic can also be encrypted
  - so called "parameter encryption"

A FEW WORDS ABOUT THE BROKEN STATE OF WINDOWS' BITLOCKER

- uses TPM to decrypt the disk automatically during boot
  - default: no PIN required
- does not implement parameter encryption

Why does Windows not enable TPM 2.0 parameter encryption to protect against bus sniffing of Bitlocker key?

Asked 3 years, 10 months ago Modified 3 years, 4 months ago Viewed 2k times

TPM SNIFFING ON WINDOWS

Indeed, the way we perform the attack nowadays allows us to break the BitLocker protection in only a few minutes on the three major enterprise-grade laptop manufacturers (i.e. Lenovo, HP, and Dell).

https://blog.scrt.ch/2024/10/28/privilege-escalation-through-tpm-sniffing-when-bitlocker-pin-is-enabled/

TPM SNIFFING - WORKSHOPS /O\

Join us for an exclusive TPM sniffing workshop

We are pleased to invite you to attend our workshop: **"Exploiting TPM sniffing"** given by **Julien Oberson**, our Head of Offensive Security, in our office in Bern.

We'll dive deep into TPM sniffing and its implication for BitLocker encryption—even when additional protections like a PIN are in place. This hands-on demo session is crafted for cybersecurity and technology enthusiasts interested in cutting-edge insights on workstation data protection.

Highlights

- Analysis of BitLocker's transparent mode vulnerabilities and how they can be exploited.
- Live demonstration of TPM sniffing and its potential to escalate privileges.
- Practical strategies to secure against data extraction from a workstation.
- Dinner and networking with other cybersecurity enthusiasts.

BITLOCKER - SUMMARY

- bad: unlimited tries for cold boot attacks
- bad: unlimited tries for TPM sniffing attacks
- BitLocker recovery key is often stored online in the user's Microsoft account
- bitpixie vulnerability (CVE-2023-21563): a TPM related issue
  - not fixed, Proof of Concepts are on Github
- the good: all of the issues can be fixed by enabling pre-boot authentication
- but: can only be done by using the command line or Group Policy Editor
- You should expect cops to be able to decrypt Windows devices

CONCLUSION: ENCRYPTION (STILL) WORKS

Die Daten auf dem im Untersuchungsgegenstand mit der Asservatennummer [redacted] eingebauten Datenträger waren verschlüsselt und konnten mit den Mitteln des Sachgebiets [redacted] nicht entschlüsselt werden. Das Gerät wurde ohne weitere Untersuchung der Daten an den Antragsteller der Untersuchung zurückgegeben.

Verantwortlicher Sachverständiger

CONCLUSION

- Smartphone situation: very bad
- Windows: flaws everywhere
  - activate pre-boot authentication
- Mac/macOS: ???
- Linux
  - fde disabled by default
  - hard to encrypt existing system
- cold boot attacks
  - especially a problem on auto-unlock systems
- Secure Boot + TPM: very powerful in combination

NON TECHNICAL SOLUTIONS (FOR COPS)

BY ROLE PLAY: DHL COURIER

tagesschau

Sendung verpasst?

3 Min

IT-Sicherheitslücken

Unerwünschte Zivilcourage bei Hackern?

Stand: 18.05.2025 11:56 Uhr

In einer digitalen Welt ist die IT-Sicherheit zentral. Trotzdem droht Strafe, wenn Hacker solche Lücken ehrenamtlich finden und meldet.

Von Philip Raillon, ARD-Rechtsredaktion

Vor vier Jahren klingelt es bei Hendrik Heinle. Er ist Software-Entwickler, damals selbständig. Heinle sieht zwei DHL-Boten vor der Tür. Nichts Ungewöhnliches, also öffnet er. "Paket für Herrn Heinle. Sind Sie das?", sollen die beiden Männer gefragt haben. Heinle bejaht und wird direkt an die Wand gedrückt. "Polizei", ruft einer der - vermeintlichen - Paketboten. "Das war ein einschneidendes Ereignis", erinnert sich Heinle.

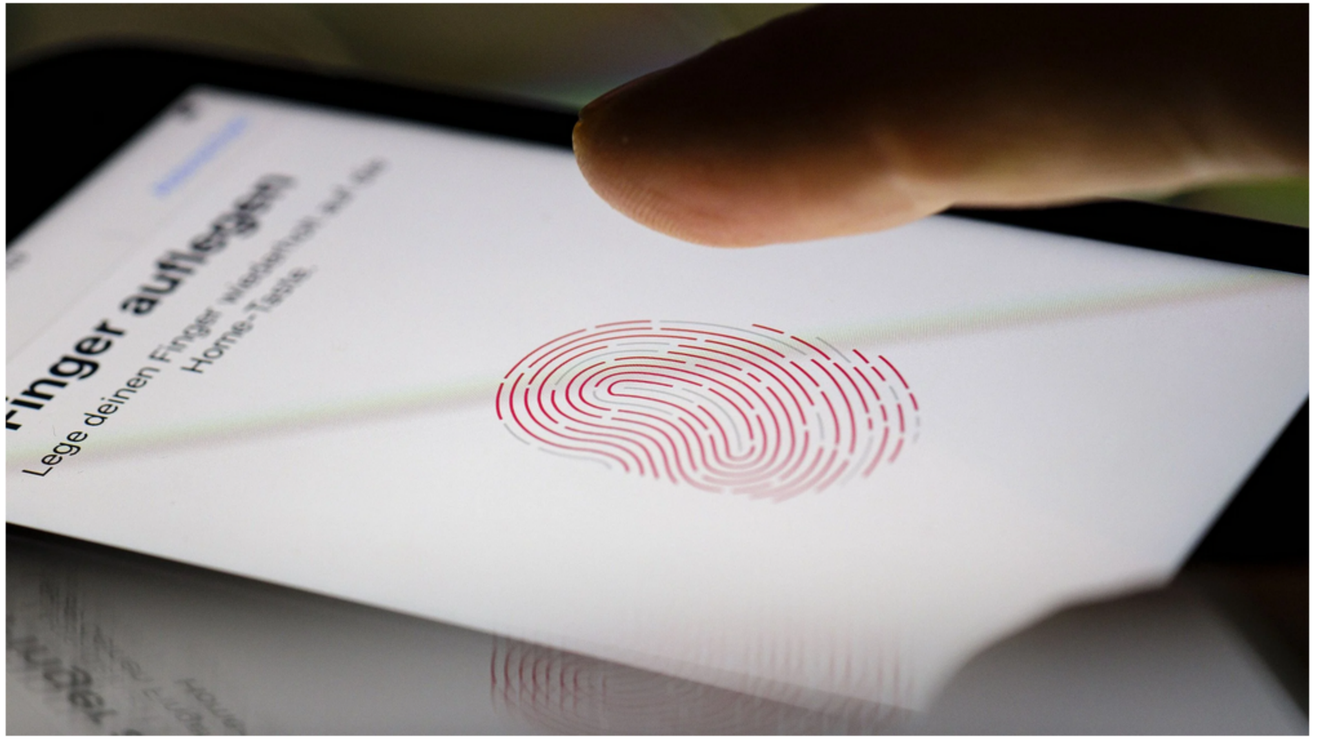
Die Polizeibeamten wollten ihn überraschen, damit Heinle Dateien und IT-Geräte nicht mehr verschlüsseln kann. Die Polizei beschlagnahmt das gesamte Equipment. Laptops, Smartphone, Festplatten und mehrere USB-Sticks nehmen die Beamten mit. Denn Hendrik Heinle ist Beschuldigter in einem Strafverfahren. Der Vorwurf: Ausspähen von Daten.

BY LAW: FINGERPRINT UNLOCK

BGH entscheidet erstmals

Zwangsweises Fingerauflegen zur Smartphone-Entsperrung rechtmäßig

von Joschka Buchholz 20.05.2025



Mittlerweile werden viele Smartphones nicht mehr durch Fingerabdrücke, sondern durch Gesichtserkennung entsperrt. Foto: picture alliance / photothek | Thomas Trutschel/photothek.de

Der BGH hat eine lange umstrittene Frage entschieden: die Polizei darf Finger unter Zwang aufs Handy legen – jedenfalls unter bestimmten Voraussetzungen.

BY CAMERA FOOTAGE

BY COPS BEING COPS

- putting people under pressure
- telling lies
- using violence

Frage 4:
Wann und auf welcher Rechtsgrundlage wurden Geräte bereits wieder an ihre Besitzer*innen herausgegeben? (bitte nach Datum, Art und Anzahl aufschlüsseln)

Zusammenfassende Antwort auf die Fragen 3 und 4:

Insgesamt stellten 72 Personen die Zugänge zu ihren elektronischen Geräten zur Verfügung. Zur zweiten Teilfrage der Frage 3 liegen keine Angaben vor; aus dem erfragten Umstand erwachsen im Übrigen keine rechtlichen Konsequenzen. Es erfolgte bislang keine Herausgabe fragegegenständlicher Geräte. (Stand: 14. August 2023)

HOW I DO FDE?

Q&A

sources, fruther reads & slides can be found:
https://debugging.works/blog/the-current-state-of-full-disk-encryption-is-still-not-good