Logitech Unifying: Angriffe auf Verschlüsselung, Abhören, Receiver-Backdoor.

Marcus Mengs (@MaMe82)

<redacted>

<redacted>

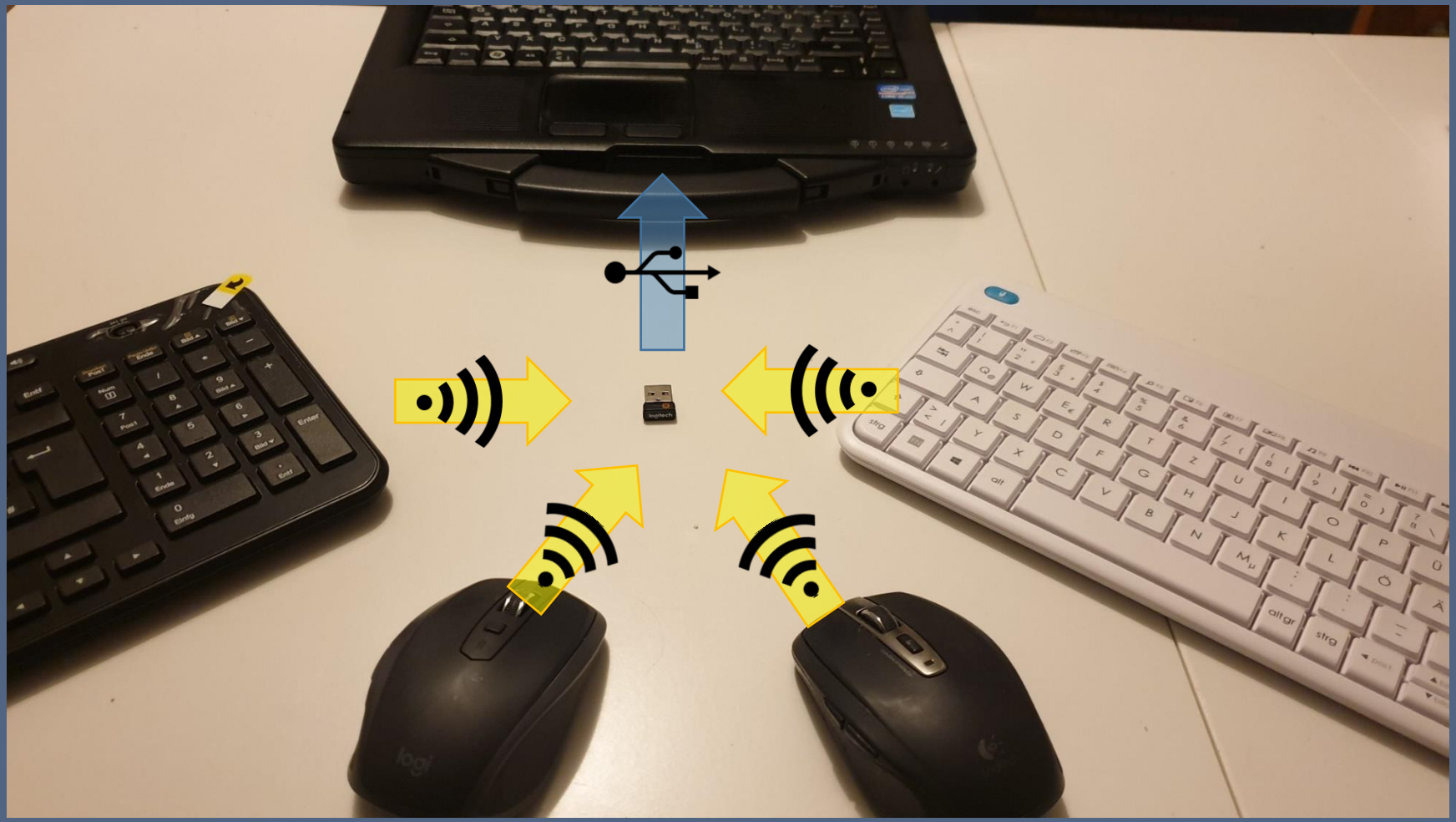
- <redacted>

```
mame82@home:~# whoami
```

Marcus Mengs

Interest:	Family, Vulns, OffSec, new attacks, coding
Languages:	C, Java, C#, Python, Go, PS, ASM
Reports:	Microsoft, Oracle, IBM, Invision, Logitech, <redacted>
Certificates:	Yes, one
CTF:	… no time ☹
Projects:	P4wnP1, P4wnP1 A.L.O.A., Nexmon Mods Pi0W, USB HID covert channel, covert channels (USB, WiFi, Unifying)
Currently:	LOGITacker

Hintergrund: Logitech Unifying



Hintergrund: Logitech Unifying

- Kleiner USB Funk-Empfänger (Dongle)
- Bis zu 6 Geräte pro Dongle (Maus, Tastatur, Fernbedienung, Gamepads, Joystick …)
- Abwärtskompatibel 1st Gen (ca. 2009)
- AES128 für (viele) Tastatureingaben
- Proprietärer Funkstandard im 2.4 GHz Band - „Enhanced Shockburst“ (ESB)
- Robust: bis zu 24 Kanäle

Motivation

- Schwachstellen / Angriffe seit 2010
- Häufig präsentiert, in Fachkreisen bekannt (und genutzt) – bei Nutzern und Endkunden weitestgehend unbekannt oder Unterschätzt
- <redacted> ist Nutzung i.d.R. verboten

… „so why bother?“

Prior Research

- KeyKeriki V2 (Thorsten Schroeder, Max Moser – Dreamlab Technologies), 2010
- ESB Pseudo Promiscuous Mode mit nRF24 (Travis Goodspeed), 2011
- KeySweeper (Samy Kamkar), 2015
- MouseJack (Marc Newlin – Bastille), 2016
- „Of Mice and Keyboards“ (Gerhard Klostermeier, Matthias Deeg – SySS GmbH), 2016
- Presentation Clickers (Marc Newlin – Bastille), 2019

Unifying: no decryption vulnerability, known injection vulnerabilities patched (in theory)

<redacted>

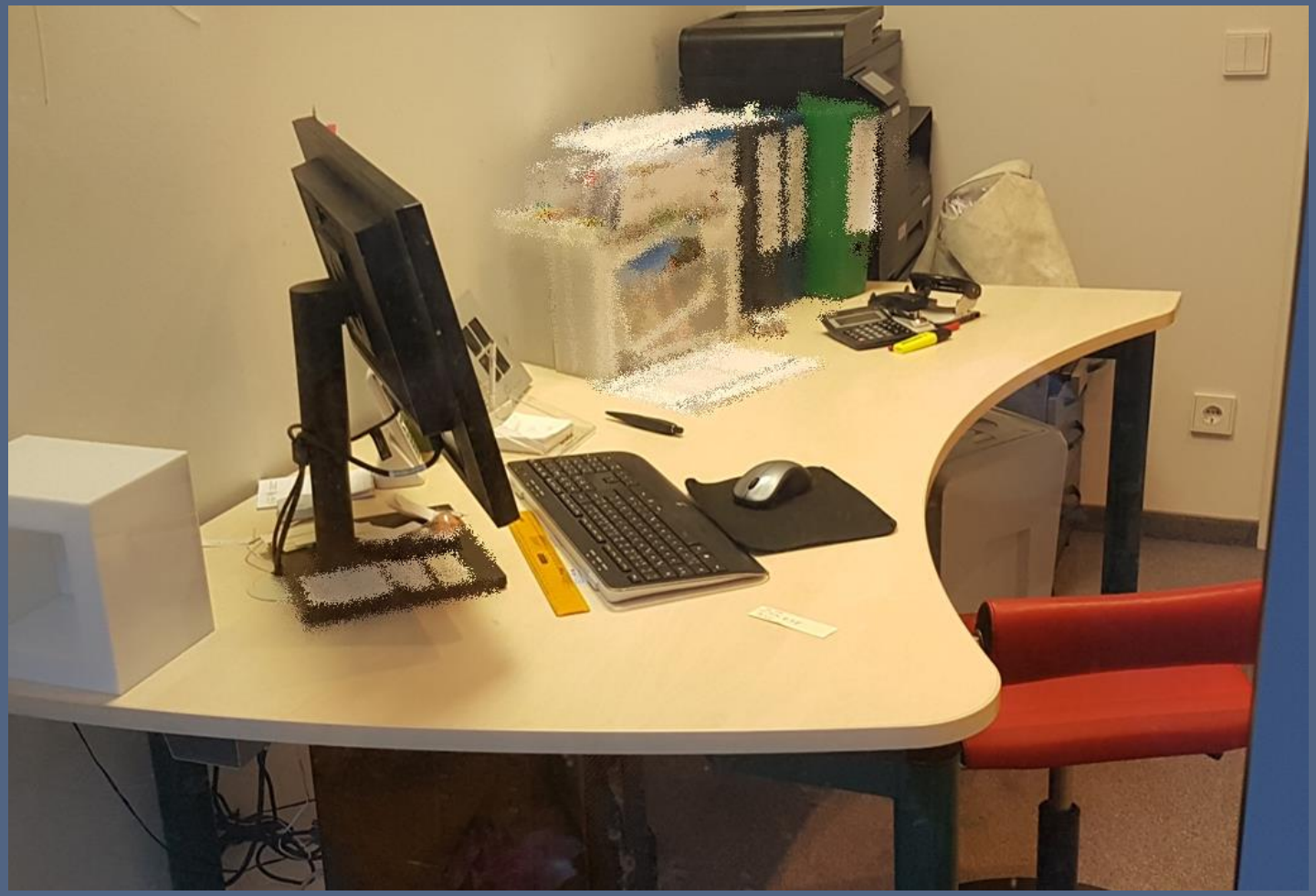
Betriebsarzt

Logitech Receiver (Wireless Mouse)

Ausstattung für Vortragende

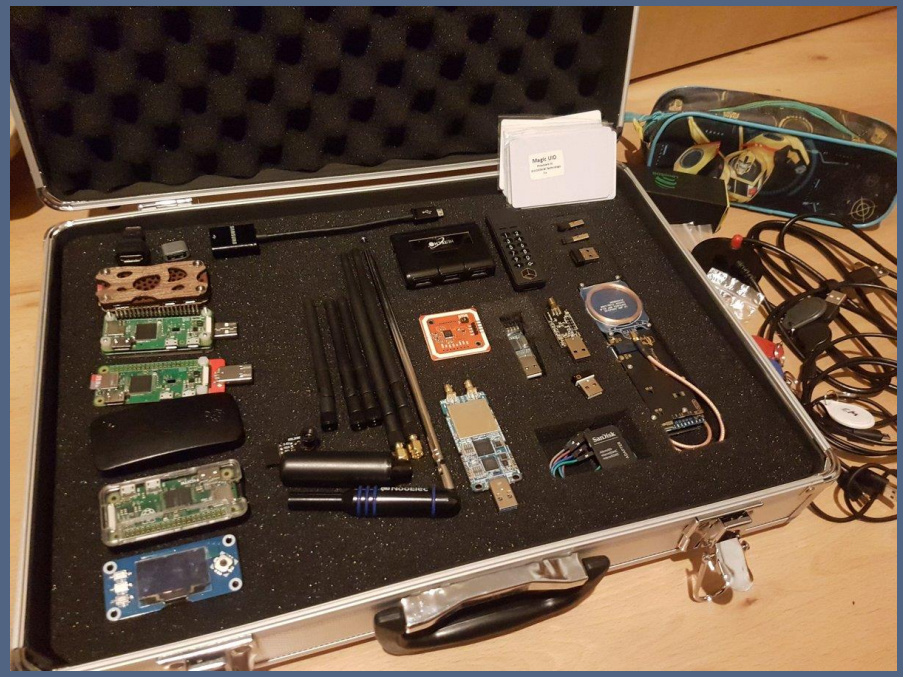
Logitech Presenter

… <redacted>, Arzt der Familie



Verlagerung R&D in den privaten Bereich

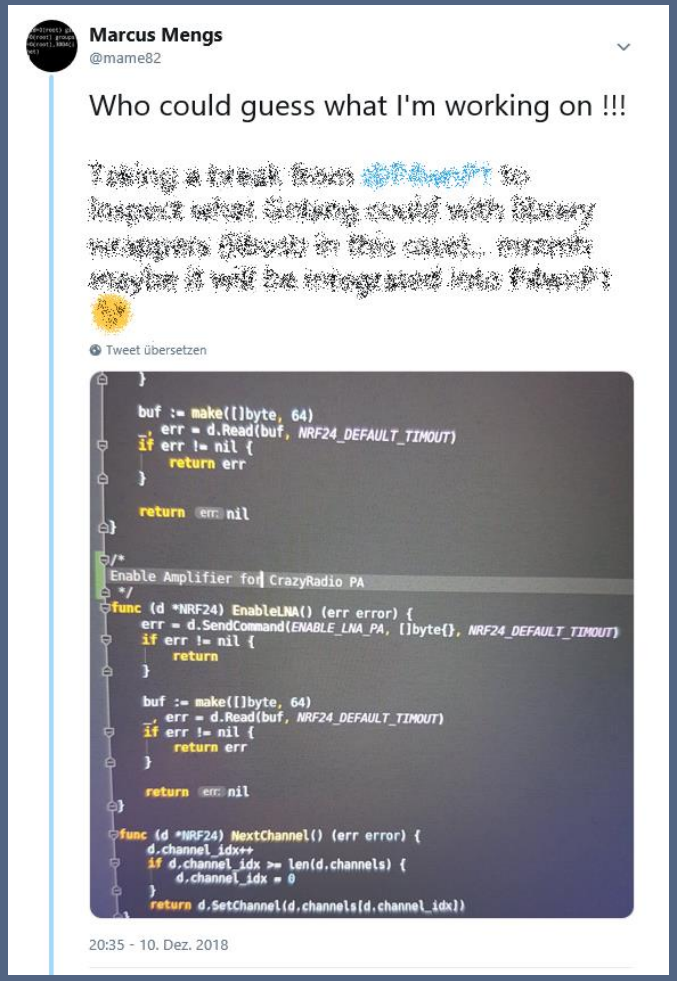
- Kollege testet „keystroke injection“ an Presenter
- Erfolg nach 10 Minuten, aber nicht zuverlässig
- Weitere Analyse & Toolentwicklung nötig
- Mangel-Ressourcen: Spezialhardware, Spezialsoftware, Entwicklungszeit
- Lösung: Erstellung FOSS Projekt in Freizeit (Erlaubnis dienstl. Nutzung)

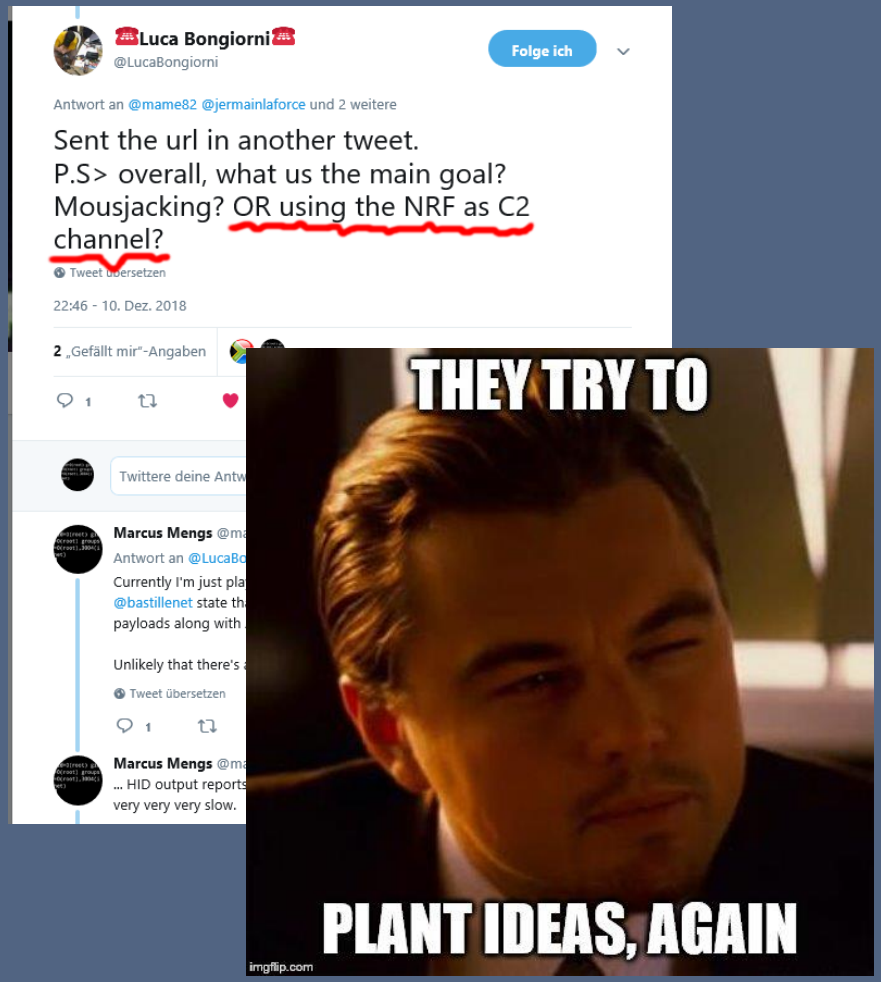


Anfangsziel

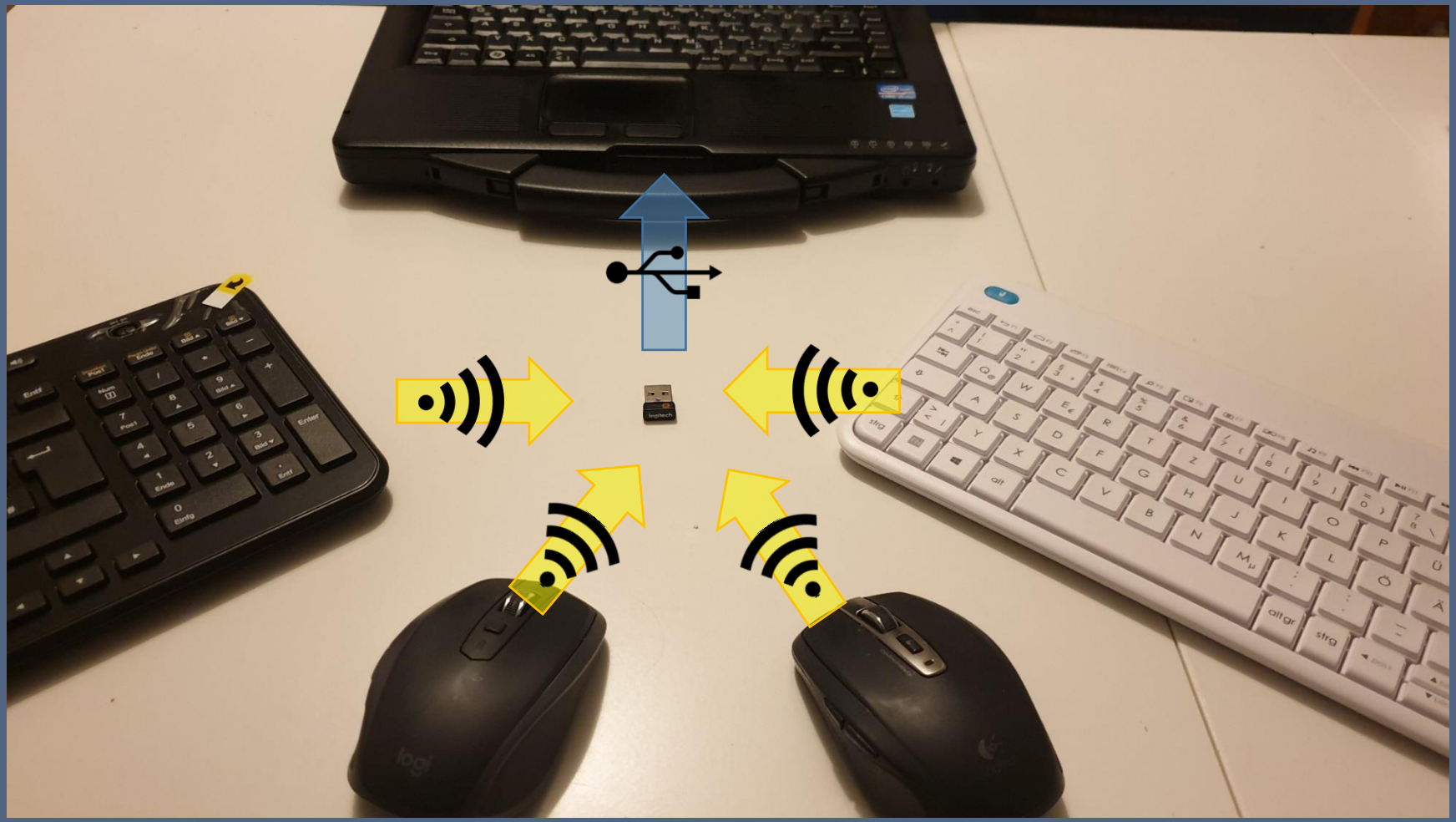
- Verbesserung vorhandener Tools (Verlässlichkeit, Funktionalität)
- Bereitstellung als FOSS
- Nutzbarkeit für Live-Demos in Sensibilisierungsvorträgen

Zwischenergebnis





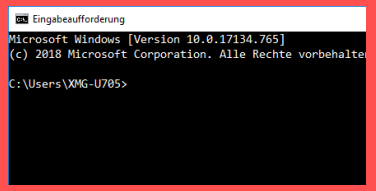
Neues Ziel



Neues Ziel

Client Agent
- Shell
- Nutzt USB

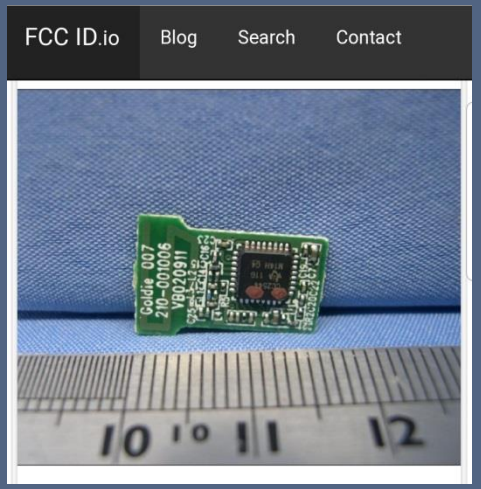
„Rogue device“
- Simuliert reales Gerät
- Kommuniziert über Funk
- C2 für den Client-Agent

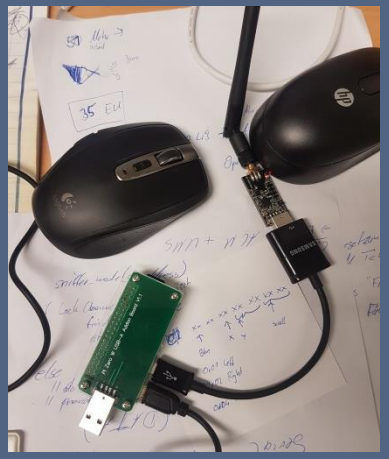


Dongle
- Fungiert als Relay zwischen USB und RF
- Keine Manipulation

Methodik

- Information Gathering
  - Prior Research (ESB, Modulation, CRC Algorithmus …)
  - FCC Datenbanken (Modulation, genutzte Kanäle)
  - Logitech Drafts
  - OSS Projekte z.B. `fwupd`
- Funk
  - SDR (Channel-Hopping, ESB ACK Payloads)
  - Custom Tool `mjackit` (Golang, Device Emulation, Dongle Emulation, Fuzzing …)
  - Angepasste „nrf-research-firmware“ für nRF24LU1+
- USB
  - USBPCap (Kommunikation zw. Host und Dongle)
  - Custom Tool `munifying` (Golang, Device Enumeration, Pairing …)



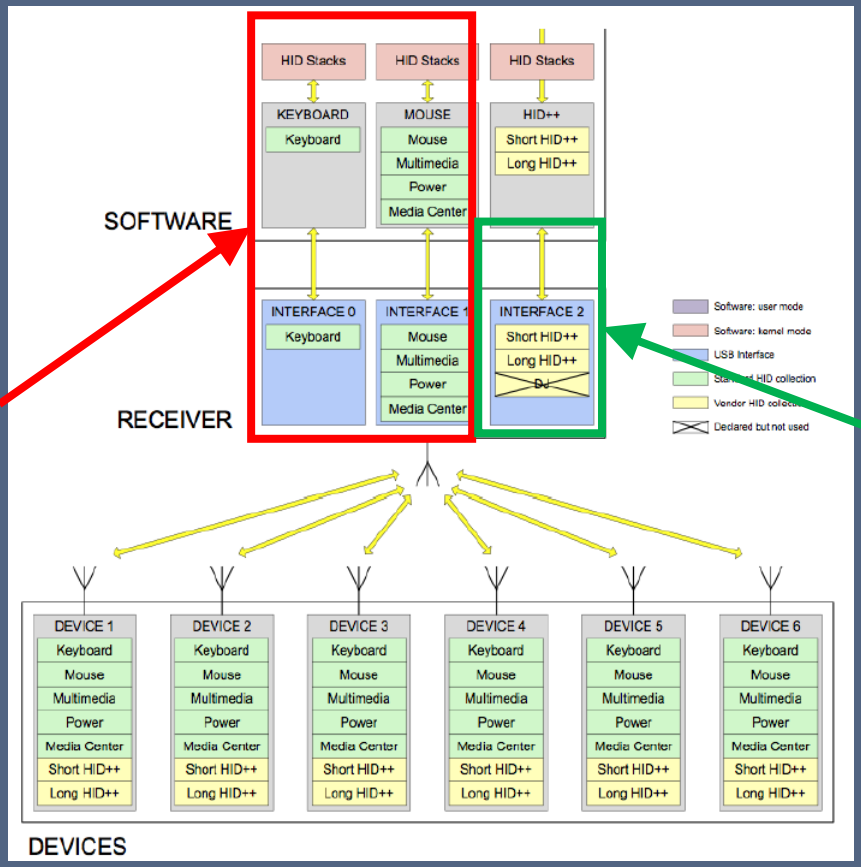


Problemstellungen

- Beliebige Daten vom Client Agent zum Dongle (USB) müssen über Funk „weitergeleitet“ werden (USB to RF Relay)
- Beliebige Daten vom „Rogue Device“ zum Dongle (Funk) müssen über USB weitergegeben werden (RF to USB Relay)
- Keine Manipulation des Dongles (z.B. Firmware)
- Keine Beeinträchtigung der Unifying Geräte

Lösung HID++

USB HID
Tastatur und
Maus
- PnP Treiber
- Low priv
- … aber, OS-exklusiv



Quelle: Unifying Receiver DJ collection specification draft

USB HID generic
- USB HID descriptor „vendor specific“
- PnP Treiber (Generic HID)
- Konkurrentes Lesen/Schreiben für mehrere Prozesse (shared)
- Low priv

Lösung HID++

- Logitech proprietär, aber basiert auf Standard-Konformen USB HID
- U.a. genutzt um Batteriestatus von Geräten abzufragen, d.h.
  - Downstream Channel: Host (USB) → Dongle → Gerät (Funk) - für Anfrage
  - Upstream Channel: Gerät (Funk) → Dongle → Host (USB) – für Antwort
- Ungültige Pakete werden mit einem Fehler beantwortet (keine Beeinträchtigung der Kommunikation)
- Das Protokoll ist für „noch unbekannte Erweiterungen gerüstet“

Details zur Implementierung

Im persönlichen Gespräch … nicht machbar in der Vortragszeit.

USB output report (results in RF frame from dongle to device)

```
byte num: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 len: 20
-----------------------------------------------------------------------------
example:  11 03 bb 4d 69 63 72 6f 73 6f 66 74 20 57 69 6e 64 6f 77 73


00:     USB report ID, 0x11 (HID++ long)
```

RF frame (ESB, needs valid device address, device has to support HID++)

```
byte num: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 len: 22
-----------------------------------------------------------------------------------
example:  02 11 4c bb 4d 69 63 72 6f 73 6f 66 74 20 57 69 6e 64 6f 77 73 25


00:     reserved - TX: unused, RX: destination Device index (0x00..0x05)
01:     reserved - RF report ID, dongle->device 0x11 (HID++ long), device->dongle 0x51 (HID++ long, with
```

Neue Problemstellung - Deployment

Naheliegend: Keystroke Injection, aber Lücken sind gepatcht

Auffälligkeiten in der Verschlüsselung

- Beim Paring von Devices an ein Dongle werden maximal 8 Byte zufällige Daten ausgetauscht
- Jedes „gepairte“ Device verwendet einen 128 Bit AES Schlüssel (16 Byte, wird vermutlich beim Pairing erzeugt)
- Übertragener Cipher-Text besteht nur aus 8 Byte Daten und einem 32 Bit Counter (vermutlich AES CTR)

Erweiterung der Methodik

- Ergänzung von Funktionalität um das Pairing von Devices mit verschiedenen Parametern zu emulieren (mjackit)
- Erweiterung bestehender Tools um Firmware eines Dongles zu extrahieren
- Statische Firmware Analyse
- Firmware-Patching (key-extraction)

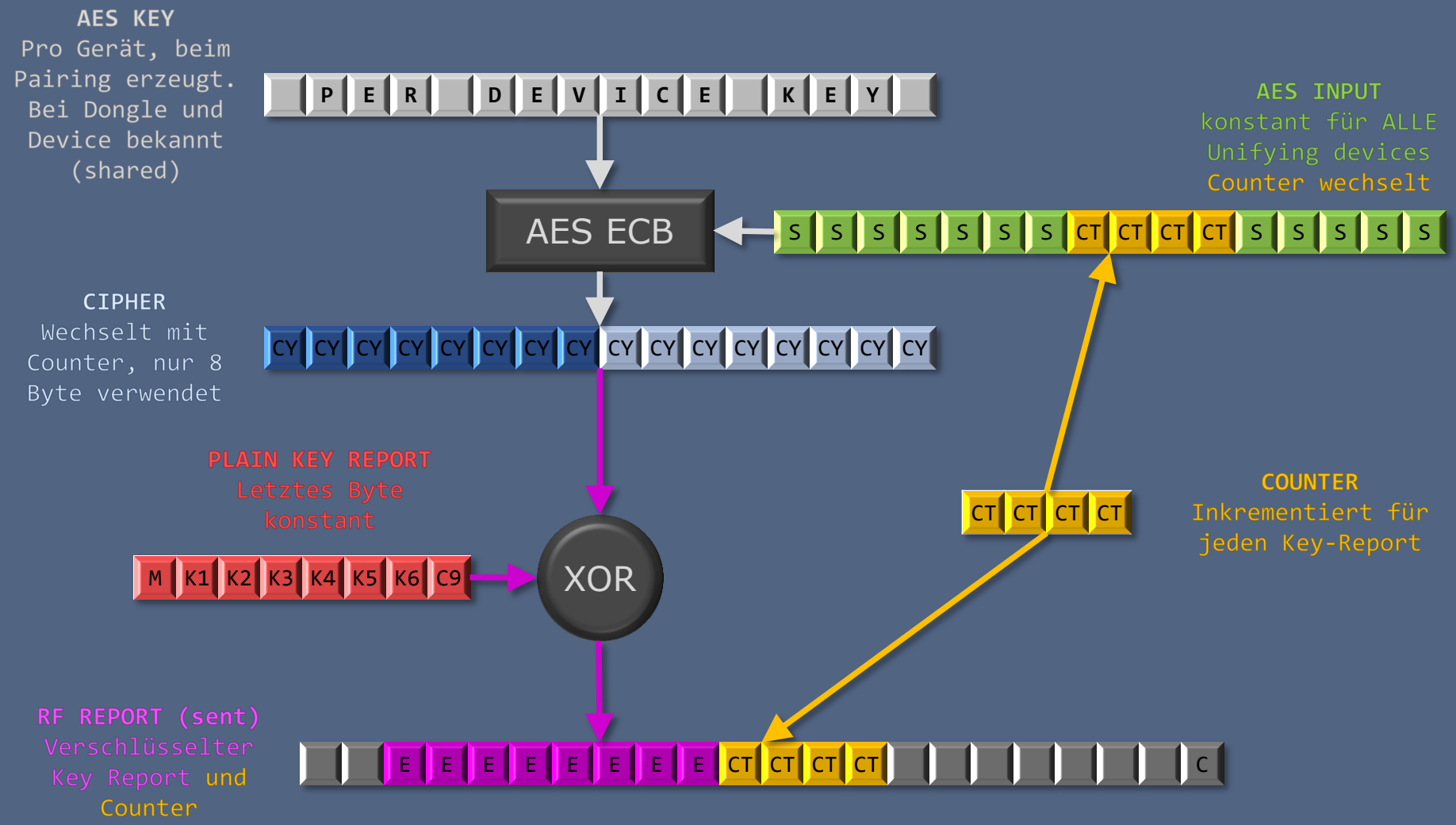
```
227  |      ::   0x00004a2c      025500          ljmp 0x5500
228
229
230  At 0x5500 insert
231  --------------
232  0x5500    d3              setb c
233  0x5501    947f            subb a, #0x7f           <-- doesn't enable carry
234  0x5503    5003            jnc 0x550b              <-- jump behind return co
235  0x5505    024a2f          ljmp 0x4a2f             <-- continue where we lef
236  0x55xx    c3              clr c                   <-- special code starts h
237  0x55xx    9082d2          mov dptr, #0x82d2       <-- src memtype for copy
238  0x55xx    ae2a            mov r6, 0x2a            <-- param 2 (src addr hig
239  0x55xx    af2b            mov r7, 0x2b            <-- param 3 (src addr low
240  0x55xx    1217ad          lcall 0x17ad            <-- store beginning at DP
```

Feststellung 1 - Verschlüsselung

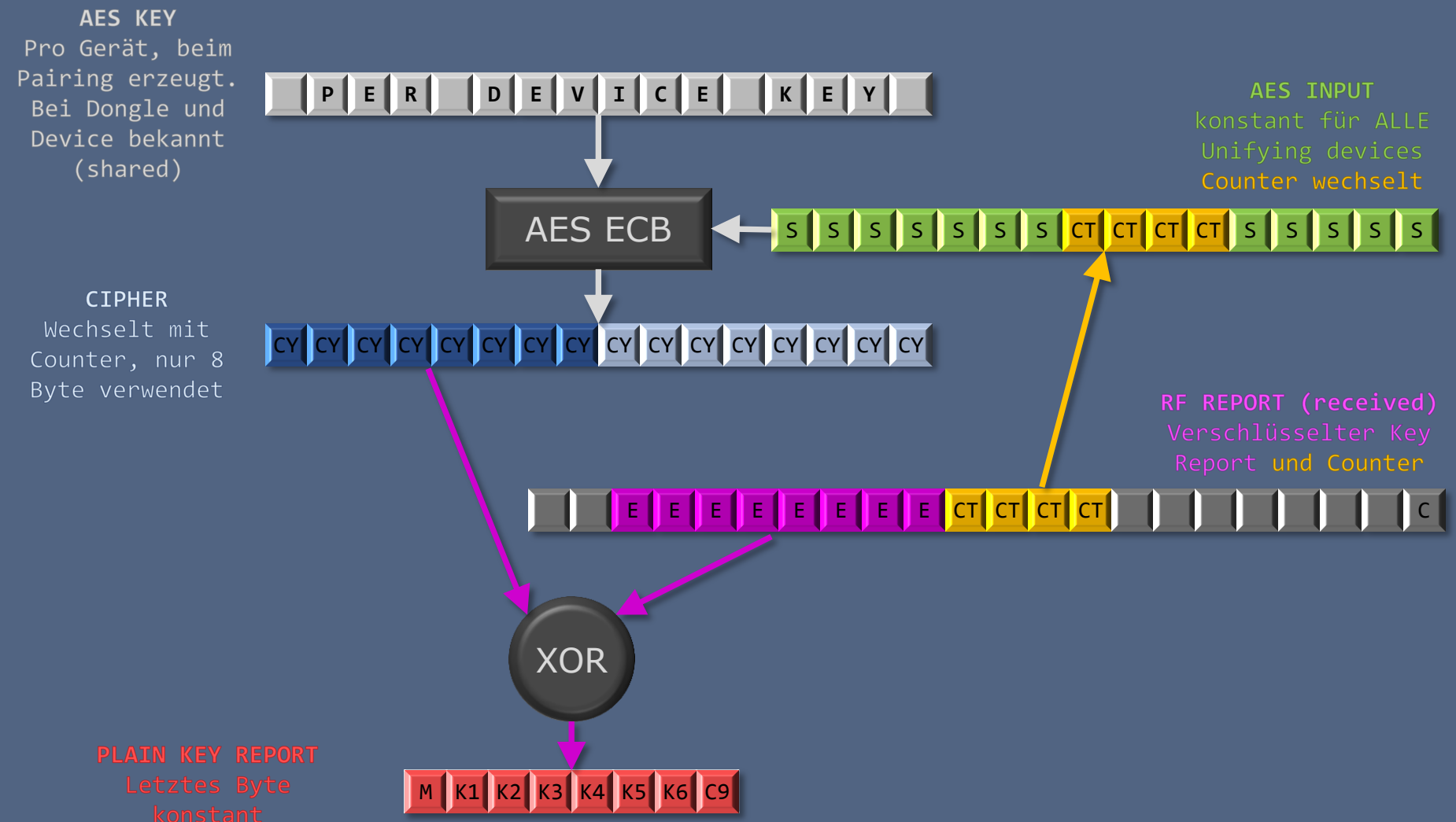
Schwacher Verschlüsselungsalgorithmus

- Keine AES Verschlüsselung des Payloads (Tasten), sondern XOR-Keying mit AES CTR Cipher
- Keine Nutzung von Hardware AES CTR, sondern ECB (CTR in Software)
- Eingangsdaten für Verschlüsselung für alle existierenden Geräte gleich (global AES indata)
- Verfahren ist anfällig für known Plaintext Angriffe
- Benötigt eine Counter-Reuse Schwachstelle (Replay), welche nach den Reports von Bastille durch Logitech geschlossen wurde
- Unverschlüsselt: Maus, Multimedia, Power, System Keys (DoS)

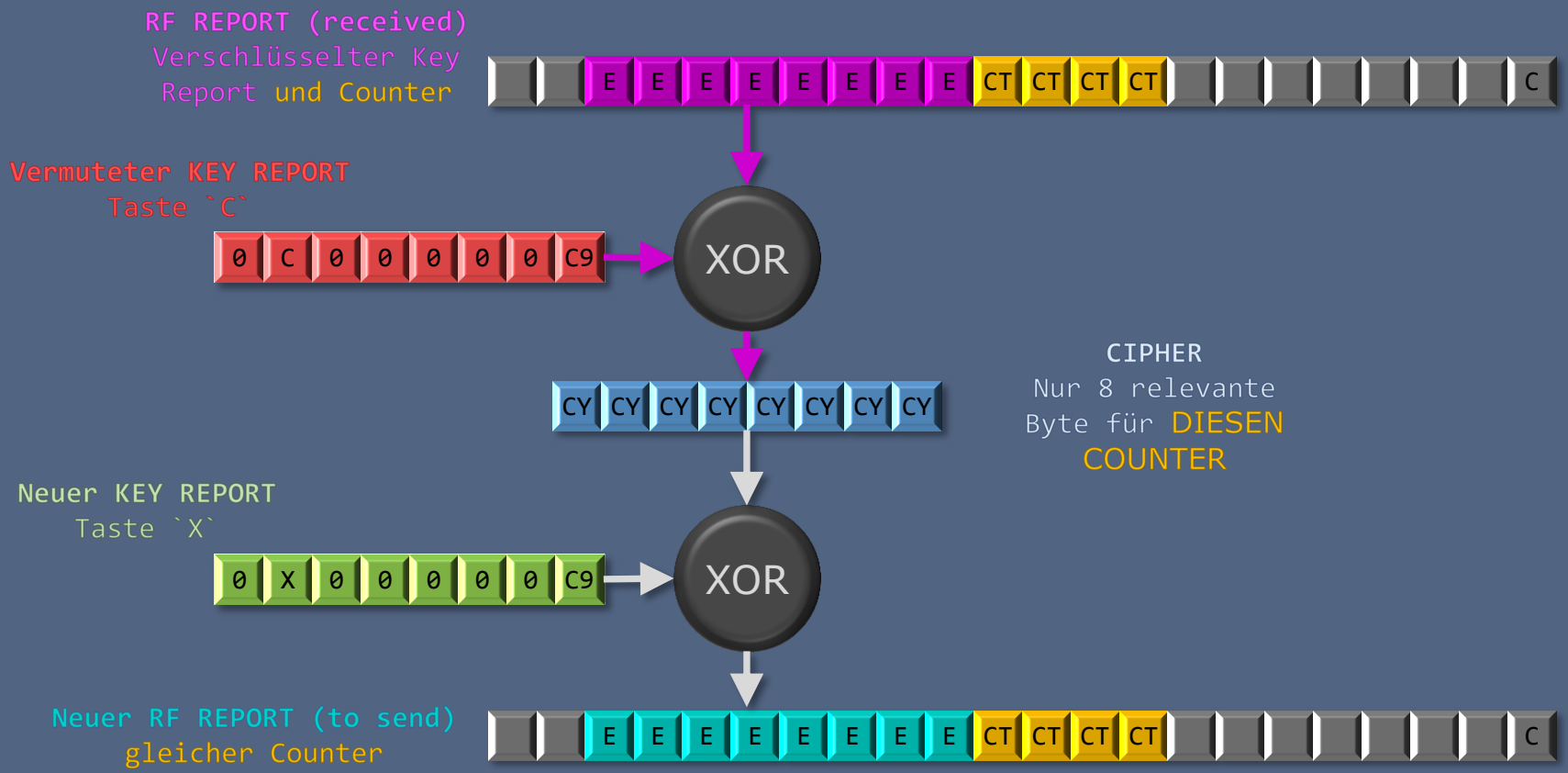
Feststellung 1 - Verschlüsselung



Feststellung 1 - Entschlüsselung

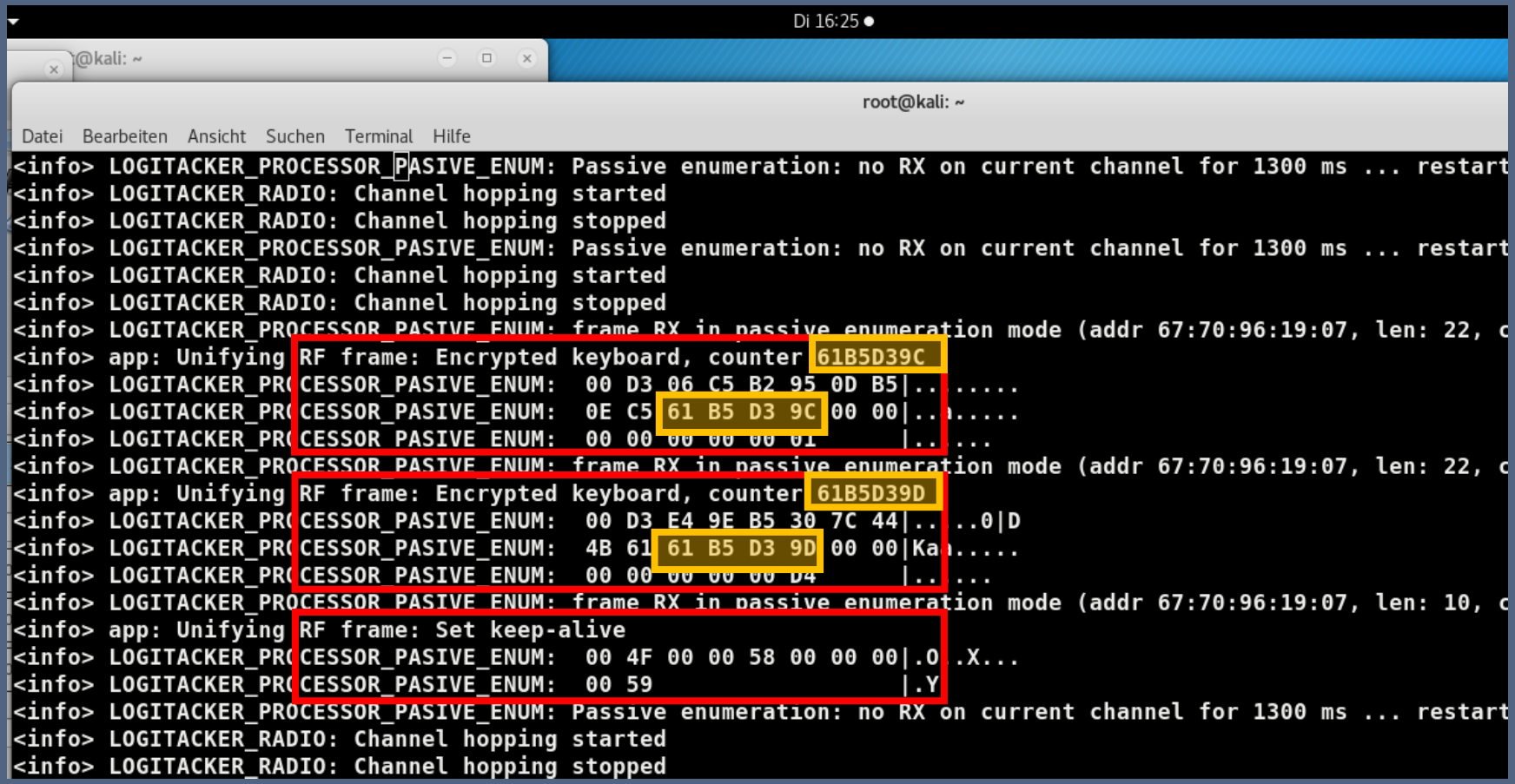


Feststellung 1 - Angriff

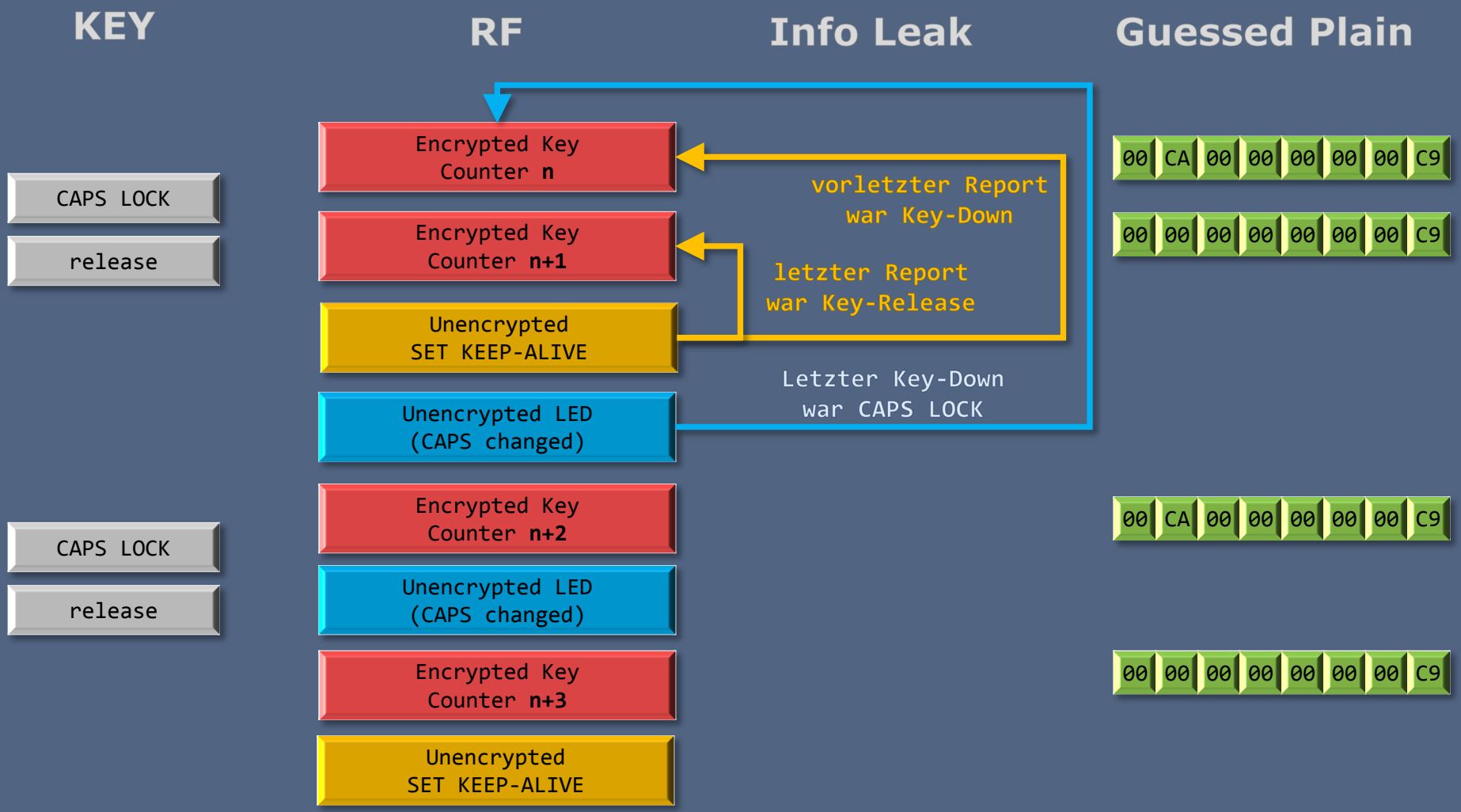


Mehrfachverwendung des Counters nötig (counter reuse). Aber, entsprechende Schwachstelle gepatcht!

Beispiel … vor einer Stunde



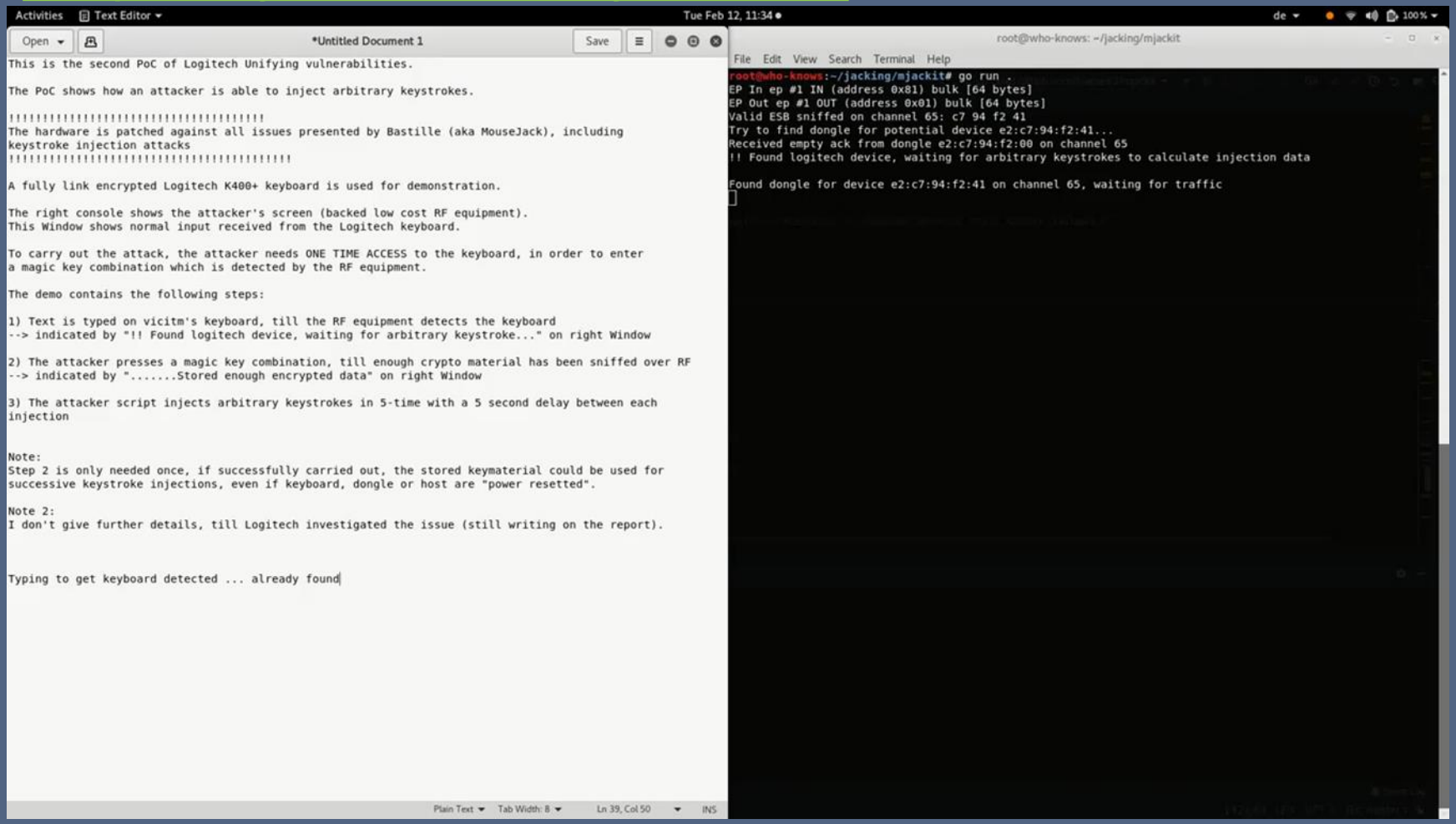
Feststellung 1 – Angriff



Plaintext für vier Counter. Ab 23 Countern reuse möglich!!

Demo – Encrypted Injection (post patch)

https://youtu.be/EksyC00DzYs



Feststellung 2 - Schlüsselerzeugung

Schwache Schlüssel

- Keine hohe Entropie, weitreichende Übereinstimmungen
- Offensichtlich schwache Schlüsselerzeugung
- Aufgrund der Eingangsannahmen, vermutlich kein sicherer Schlüsselaustausch (nur 8 zufällige Bytes im Pairing)

Keys des selben Dongle

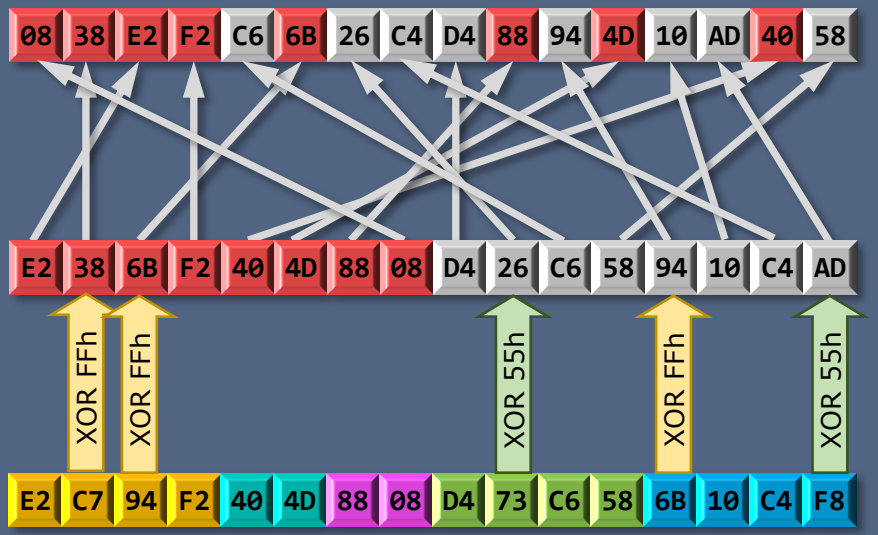
Tastatur: 08 38 E2 F2 C6 6B 26 C4 D4 88 94 4D 10 AD 40 58

Tastatur (re-pair): 08 38 E2 F2 18 6B 7E ED F9 88 B6 4D A3 8E 40 CE

Maus: 08 38 E2 F2 6D 6B F8 78 4B 88 B2 6A 39 E6 40 B0

Feststellung 2 – Schlüsselerzeugung

Derived device key



Plain key data (pairing)

Dongle SN (known, Pairing): E2 C7 94 F2

Device WPID (guessable, Pairing): 40 4D

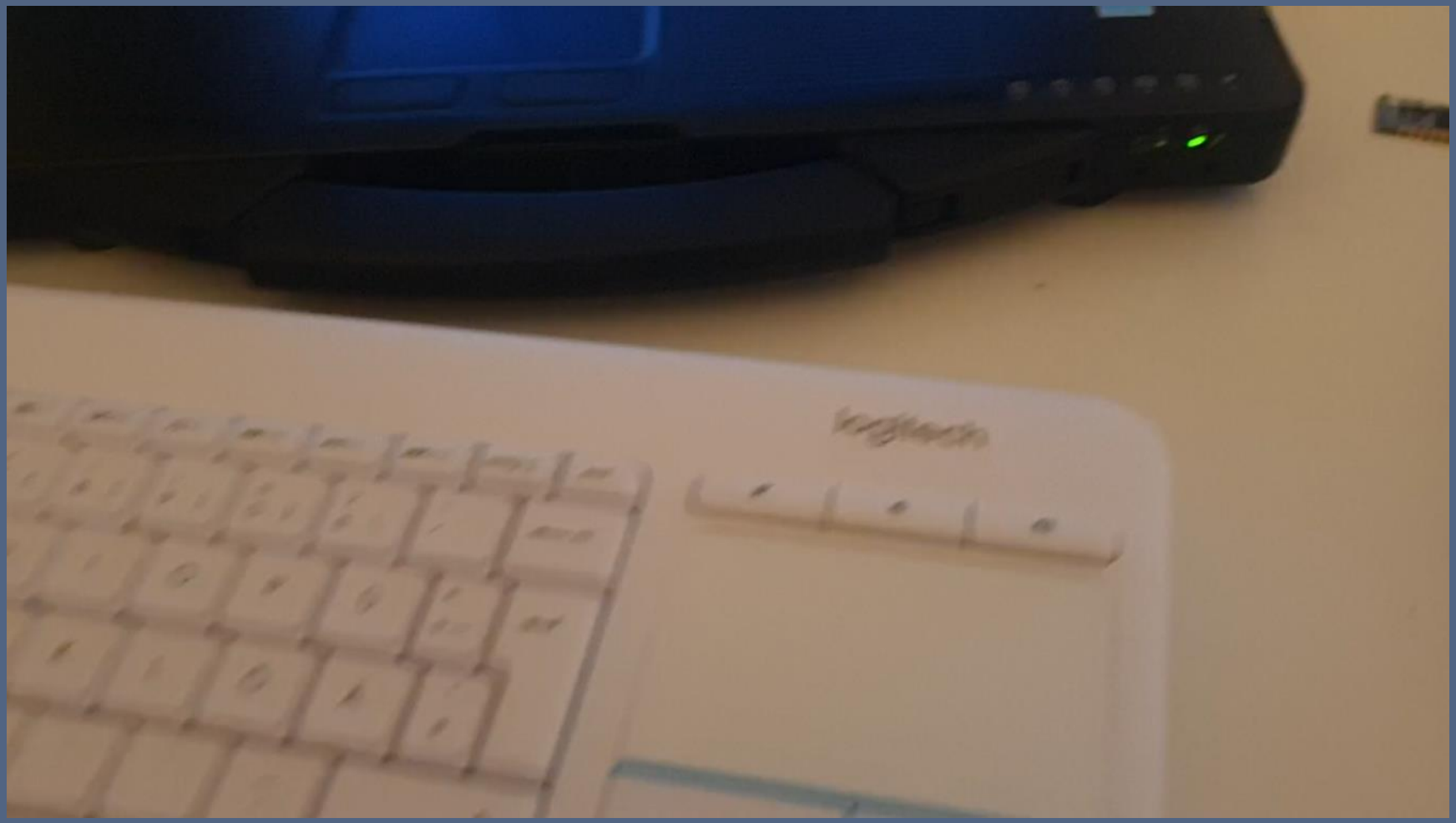
Dongle WPID (known, Pairing): 88 08

Device Nonce (PRNG, Pairing): D4 73 C6 58

Dongle Nonce (PRNG, Pairing): 6B 10 C4 F8

Demo – Sniff Pairing, live decryption

https://youtu.be/GRJ7i2J_Y80



Vendor Response

- 08. Feb 2019, Public Video (PoC 1 - Pair Sniffing, Eavesdropping)
- 11. Feb 2019, Kontaktaufnahme durch Logitech Firmware Engineering Team
- 12. Feb 2019, Public Video (PoC 2 – Injection, Counter reuse)
- 12. Feb 2019, Platzierung von Fragen an Logitech (Rahmenbedingungen Reporting? Möglichkeit zur sicheren Kommunikation?)
- 21. Feb 2019, Logitech verweist auf HackerOne, tatsächlich existiert aber kein öffentliches Programm
- 27. Feb 2019, Übersendung Reports „PoC 1“ und „PoC 2“ an Firmware Team (PGP-Mail, da Reporting-Modalitäten noch immer unklar)
- 03. Mar 2019, Public Video (PoC 3 – Schlüsselextraktion aus Dongle)
- 12. Mar 2019, Logitech stellt Link zu **PRIVATEM** HackerOne Programm bereit
- 15. Mar 2019, Übersendung Report „PoC 3“ (PGP-Mail), da HackerOne Programm fehlerhaft
- 04. Apr 2019, Mitteilung durch Logitech: An „Fix“ für alle Schwachstellen wird gearbeitet
- **14. Mai 2019, Finale Entscheidung:**
  - **KEIN „Fix“ für PoC 1 & 2 (Hardwareänderungen notwendig, Interoperabilität gefährdet), Freigabe zur Veröffentlichung**
  - **An „Fix“ für PoC 3 wird noch gearbeitet, Bitte um Verschiebung der Veröffentlichung (nicht Heise von Gestern)**

Demo - Combine everything and gimme Shellz

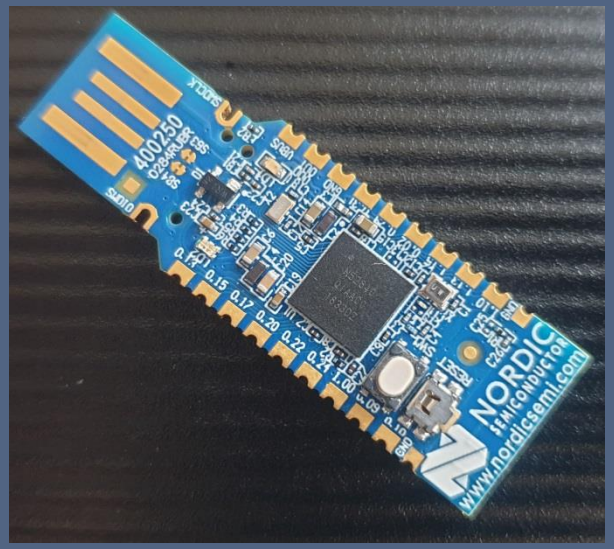
https://youtu.be/RcRF8pj5jpg

Impact

- Alle Unifying Devices betroffen
- Auslieferung ungepatchter Hardware, Updates durch Endkunden erschwert
- Supply-Chain (key re-generation)
- Post Capture key-stealing
- Replay decryption (physical access Dongle)
- Verbreitung unsicherer Dongles ????

Ergebnis / Ausblick

- Rückfluss interne Awareness Ausbildung
- Disclosure (raw-2-researchers)
- LOGITacker
- Schwachstelle 3
- Zukunft:
  - BLE (eher Gegenwart)
  - NB-IoT / LTE-M
  - BT long range



Schwachstelle 3 – key extraction from dongle in less than 1 second

https://youtu.be/5z_PEZ5PyeA

Q&A