

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

Marcus Mengs (@MaMe82)

<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

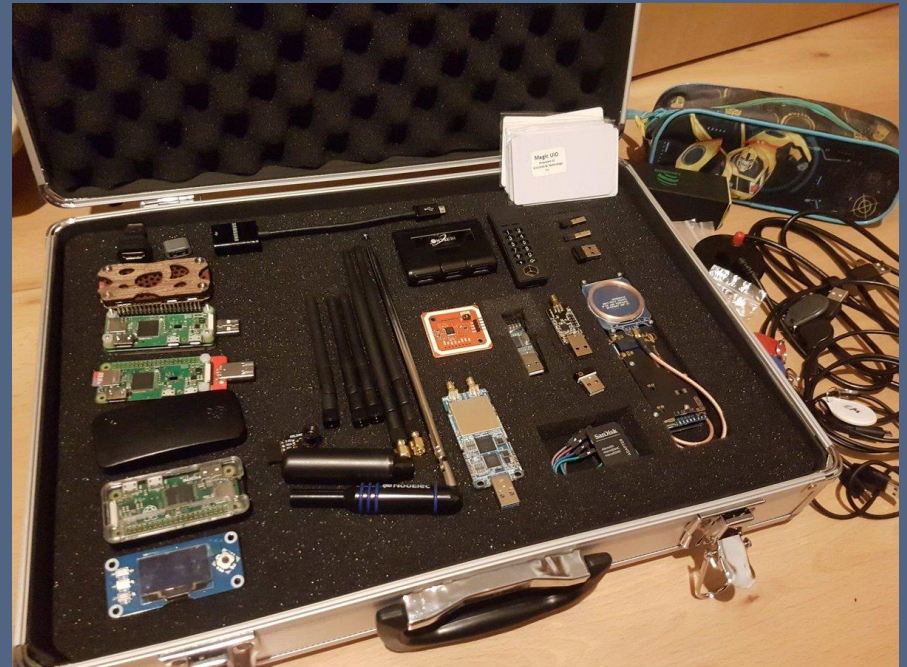


... <redacted>, Arzt der Familie



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



Marcus Mengs

@mame82

Who could guess what I'm working on !!!

Taking a break from ~~the project~~ to
investigate what ~~could~~ could with ~~blatant~~
misinterpreting ~~the~~ in the context. ~~maybe~~
maybe it will be integrated into ~~the~~ ~~future~~



Tweet übersetzen

```
}  
buf := make([]byte, 64)  
err = d.Read(buf, NRF24_DEFAULT_TIMEOUT)  
if err != nil {  
    return err  
}  
return err: nil  
}  
/*  
Enable Amplifier for CrazyRadio PA  
*/  
func (d *NRF24) EnableLNA() (err error) {  
    err = d.SendCommand(ENABLE_LNA_PA, []byte{}, NRF24_DEFAULT_TIMEOUT)  
    if err != nil {  
        return  
    }  
    buf := make([]byte, 64)  
    err = d.Read(buf, NRF24_DEFAULT_TIMEOUT)  
    if err != nil {  
        return err  
    }  
    return err: nil  
}  
func (d *NRF24) NextChannel() (err error) {  
    d.channel_idx++  
    if d.channel_idx >= len(d.channels) {  
        d.channel_idx = 0  
    }  
    return d.SetChannel(d.channels[d.channel_idx])  
}
```

20:35 - 10. Dez, 2018



Luca Bongiorno

@LucaBongiorno

Folge ich

Antwort an @mame82 @jerminalaforce und 2 weitere

Sent the url in another tweet.

P.S> overall, what us the main goal?

Mousjacking? OR using the NRF as C2
channel?

Tweet übersetzen

22:46 - 10. Dez, 2018

2 „Gefällt mir“-Angaben

1

1

1

Twitter deine Antw

Marcus Mengs @mame82

Antwort an @LucaBongiorno

Currently I'm just playing with
@bastillenet state the
payloads along with

Unlikely that there's

Tweet übersetzen

1

1

Marcus Mengs @mame82

... HID output reports
very very very slow.

THEY TRY TO

PLANT IDEAS, AGAIN

imgflip.com

Neues Ziel



Neues Ziel

Client Agent

- Shell
- Nutzt USB

„Rogue device“

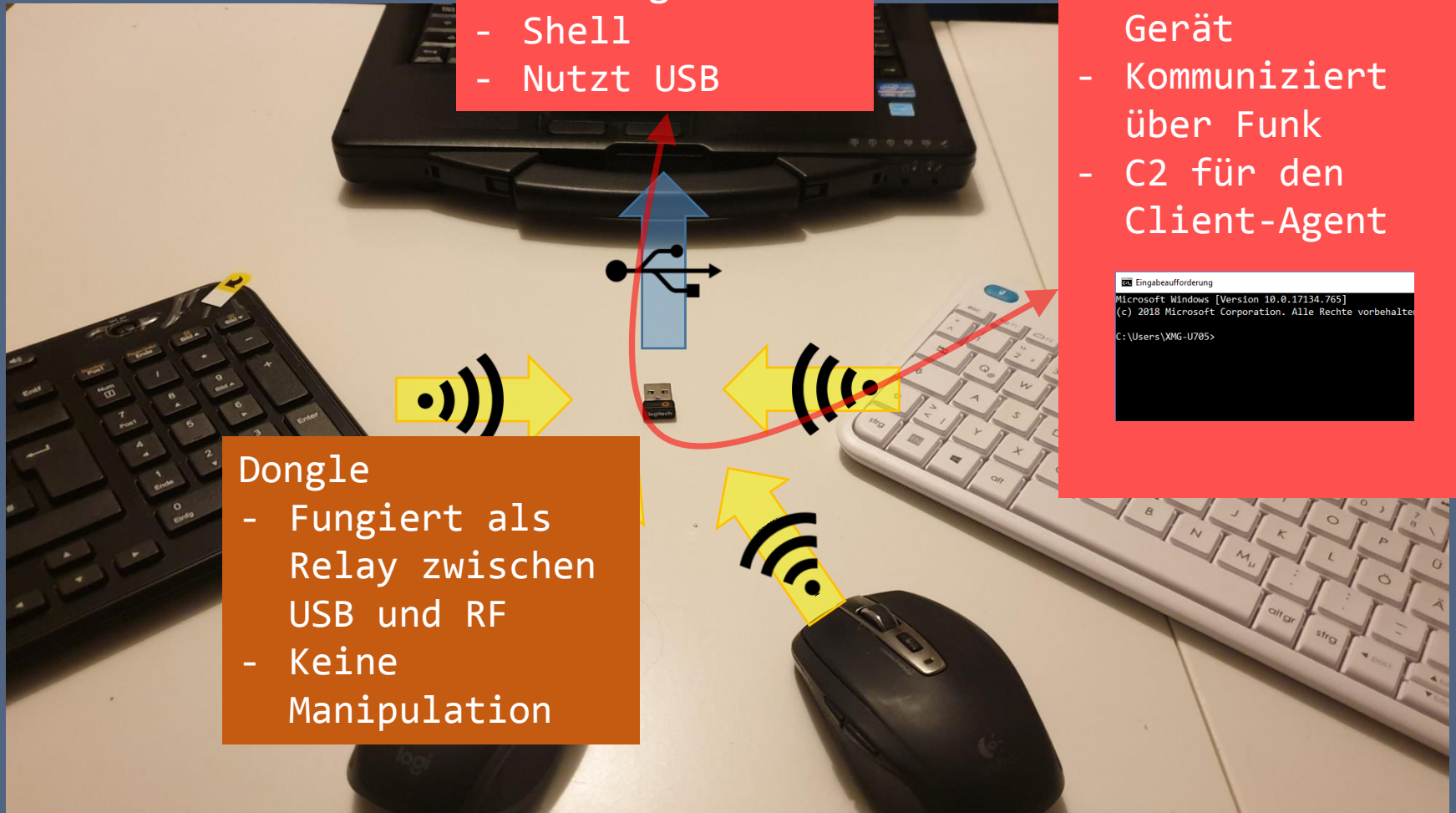
- Simuliert reales Gerät
- Kommuniziert über Funk
- C2 für den Client-Agent

```
Eingabeaufforderung
Microsoft Windows [Version 10.0.17134.765]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten

C:\Users\XMG-U705>
```

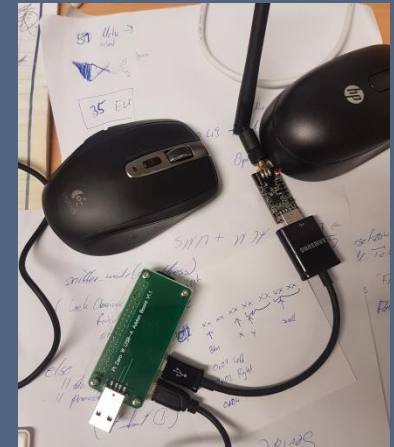
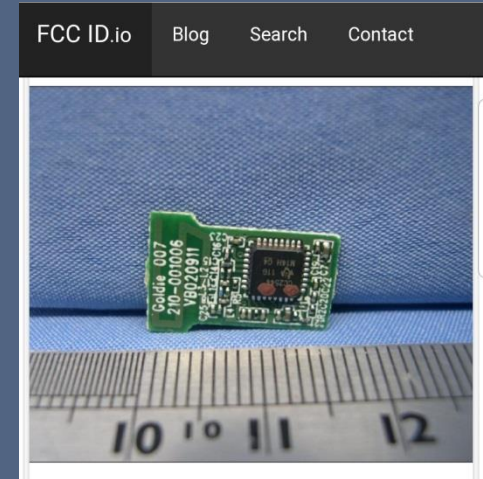
Dongle

- 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 ...)



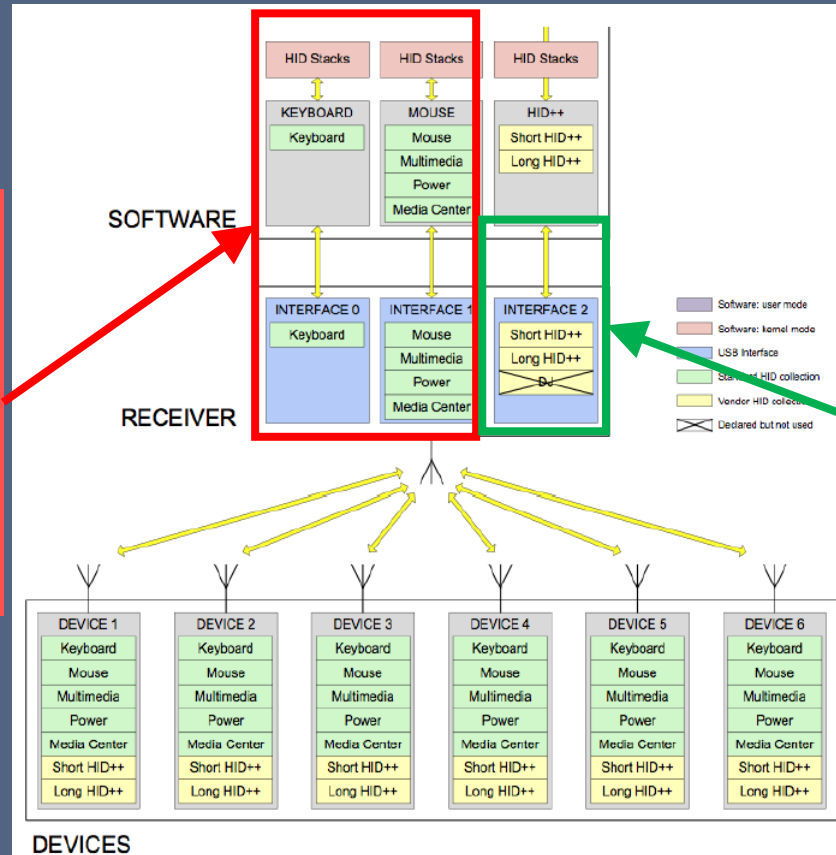
Problemstellungen

- 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 Pairing von Devices an ein Dongle werden maximal 8 Byte zufällige Daten ausgetauscht
- Jedes „gepaarte“ 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 code
235 0x5505  024a2f  ljmp 0x4a2f      <-- continue where we left
236 0x55xx  c3      clr c      <-- special code starts here
237 0x55xx  9082d2  mov dptr, #0x82d2 <-- src memtype for copy
238 0x55xx  ae2a    mov r6, 0x2a  <-- param 2 (src addr high)
239 0x55xx  af2b    mov r7, 0x2b  <-- param 3 (src addr low)
240 0x55xx  1217ad  lcall 0x17ad      <-- store beginning at DPTR
```

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

AES KEY

Pro Gerät, beim
Pairing erzeugt.
Bei Dongle und
Device bekannt
(shared)

P E R D E V I C E K E Y

AES ECB

CIPHER

Wechselt mit
Counter, nur 8
Byte verwendet

CY CY CY CY CY CY CY CY CY CY CY CY CY CY CY CY

PLAIN KEY REPORT

Letztes Byte
konstant

M K1 K2 K3 K4 K5 K6 C9

XOR

RF REPORT (sent)
Verschlüsselter
Key Report und
Counter

E E E E E E E E E CT CT CT CT C

AES INPUT

konstant für ALLE
Unifying devices
Counter wechselt

S S S S S S S S CT CT CT CT S S S S S

COUNTER

Inkrementiert für
jeden Key-Report

CT CT CT CT

Feststellung 1 - Entschlüsselung

AES KEY

Pro Gerät, beim
Pairing erzeugt.
Bei Dongle und
Device bekannt
(shared)

P E R D E V I C E K E Y

AES ECB

CIPHER

Wechselt mit
Counter, nur 8
Byte verwendet

CY CY CY CY CY CY CY CY CY CY CY CY CY CY CY CY

AES INPUT

konstant für ALLE
Unifying devices
Counter wechselt

S S S S S S S S CT CT CT CT S S S S S

RF REPORT (received)
Verschlüsselter Key
Report und Counter

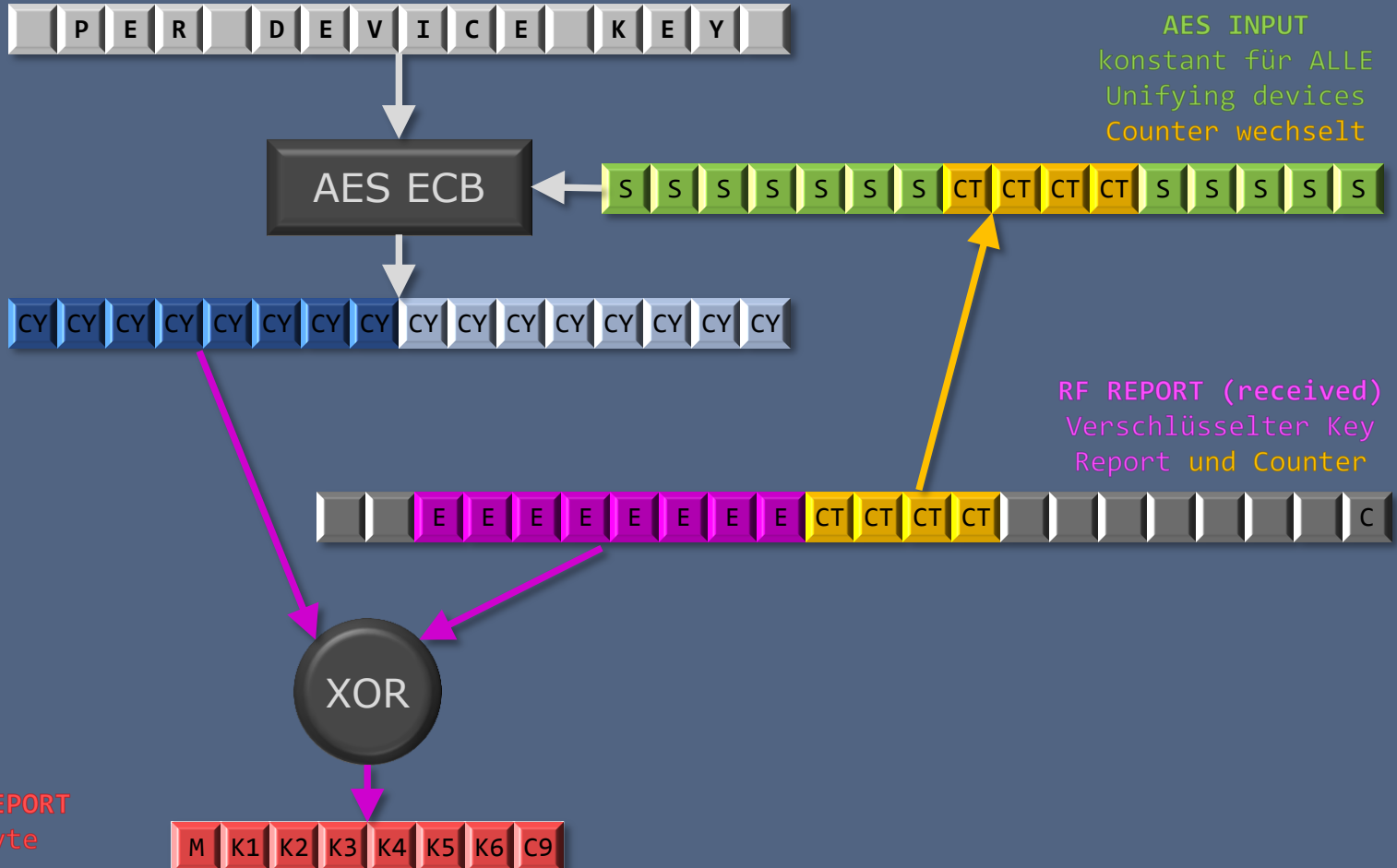
E E E E E E E E CT CT CT CT C

XOR

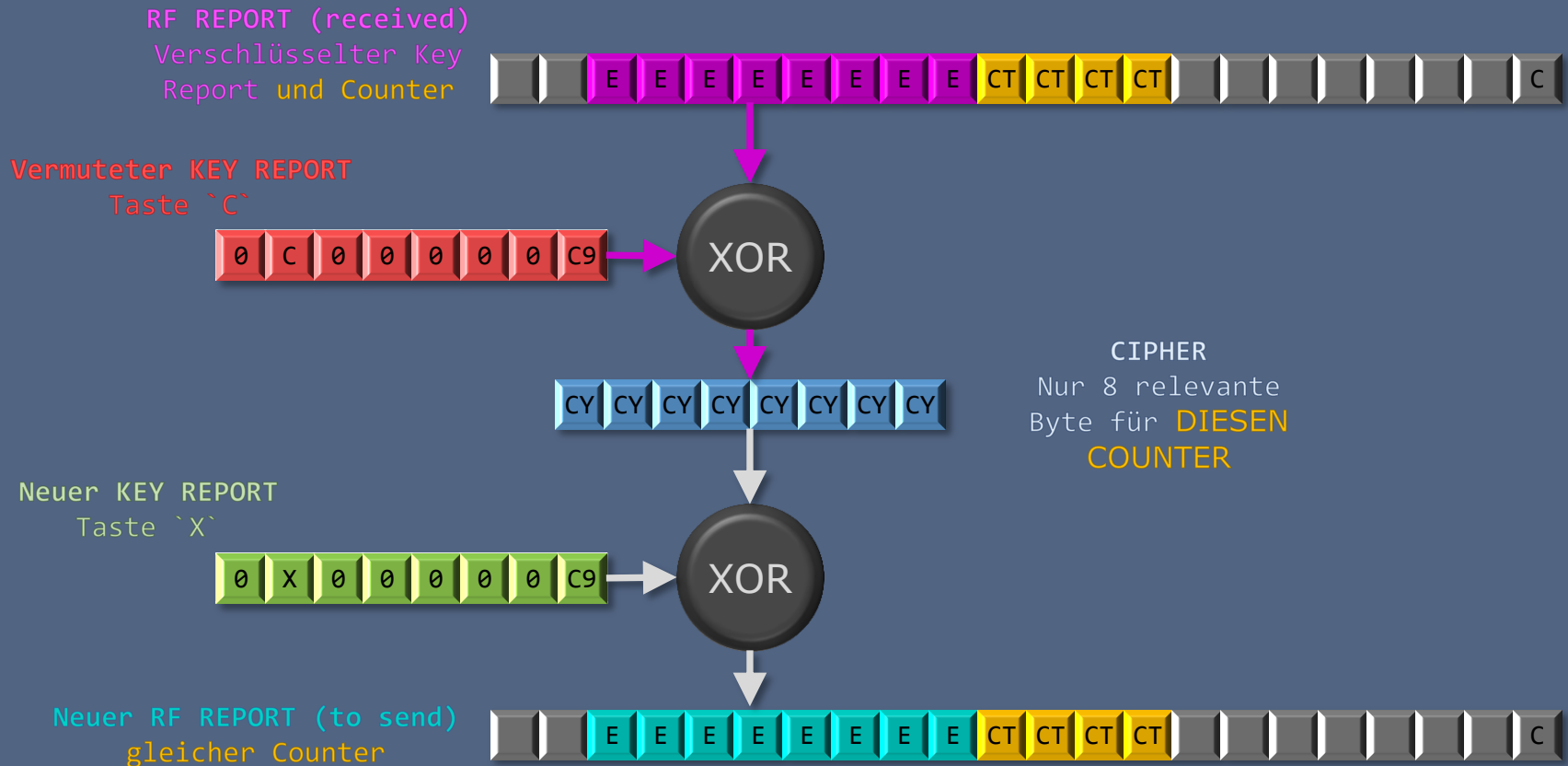
PLAIN KEY REPORT

Letztes Byte
konstant

M K1 K2 K3 K4 K5 K6 C9



Feststellung 1 - Angriff



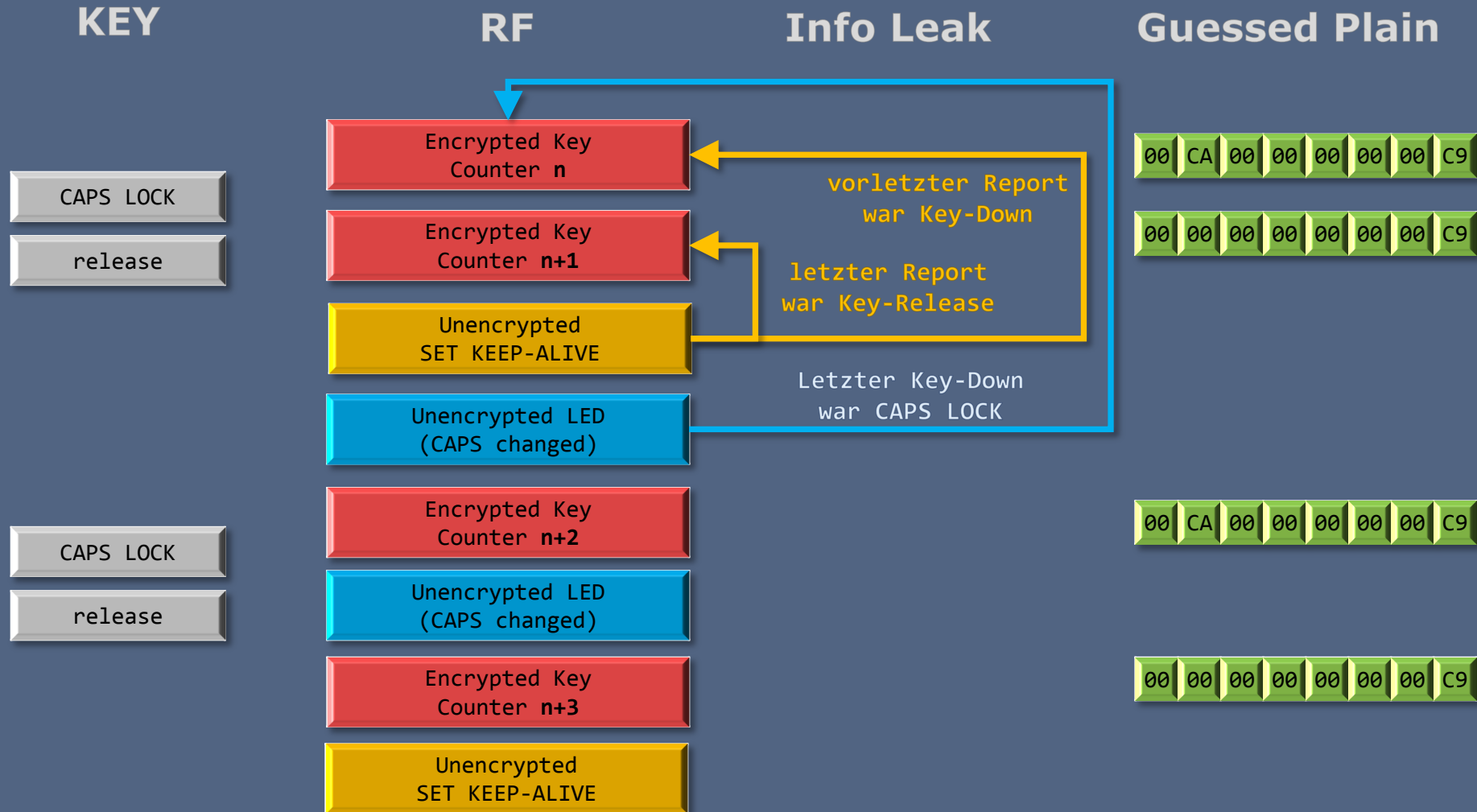
Mehrfachverwendung des Counters nötig (counter reuse). Aber, entsprechende Schwachstelle gepatcht!

Beispiel ... vor einer Stunde

```
Di 16:25 ●
@kali: ~
root@kali: ~

Datei Bearbeiten Ansicht Suchen Terminal Hilfe
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: Passive enumeration: no RX on current channel for 1300 ms ... restart
<info> LOGITACKER_RADIO: Channel hopping started
<info> LOGITACKER_RADIO: Channel hopping stopped
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: Passive enumeration: no RX on current channel for 1300 ms ... restart
<info> LOGITACKER_RADIO: Channel hopping started
<info> LOGITACKER_RADIO: Channel hopping stopped
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: frame RX in passive enumeration mode (addr 67:70:96:19:07, len: 22, c
<info> app: Unifying RF frame: Encrypted keyboard, counter 61B5D39C
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: 00 D3 06 C5 B2 95 0D B5|... ..
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: 0E C5 61 B5 D3 9C 00 00|... ..
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: 00 00 00 00 00 01 |... ..
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: frame RX in passive enumeration mode (addr 67:70:96:19:07, len: 22, c
<info> app: Unifying RF frame: Encrypted keyboard, counter 61B5D39D
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: 00 D3 E4 9E B5 30 7C 44|... ..0|D
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: 4B 61 61 B5 D3 9D 00 00|Ka.....
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: 00 00 00 00 00 04 |... ..
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: frame RX in passive enumeration mode (addr 67:70:96:19:07, len: 10, c
<info> app: Unifying RF frame: Set keep-alive
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: 00 4F 00 00 58 00 00 00|.0 .X...
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: 00 59 |.Y
<info> LOGITACKER_PROCESSOR_PASIVE_ENUM: Passive enumeration: no RX on current channel for 1300 ms ... restart
<info> LOGITACKER_RADIO: Channel hopping started
<info> LOGITACKER_RADIO: Channel hopping stopped
```

Feststellung 1 – Angriff



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

Demo – Encrypted Injection (post patch)

<https://youtu.be/EksyC00DzYs>

The screenshot displays a Linux desktop environment with two windows open.

The left window is a text editor titled "*Untitled Document 1". It contains the following text:

This is the second PoC of Logitech Unifying vulnerabilities.

The PoC shows how an attacker is able to inject arbitrary keystrokes.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!

The hardware is patched against all issues presented by Bastille (aka MouseJack), including keystroke injection attacks

!!!!!!!!!!!!!!!!!!!!!!!!!!!!

A fully link encrypted Logitech K400+ keyboard is used for demonstration.

The right console shows the attacker's screen (backed low cost RF equipment).

This Window shows normal input received from the Logitech keyboard.

To carry out the attack, the attacker needs ONE TIME ACCESS to the keyboard, in order to enter a magic key combination which is detected by the RF equipment.

The demo contains the following steps:

- 1) Text is typed on victim's keyboard, till the RF equipment detects the keyboard
--> indicated by "!! Found logitech device, waiting for arbitrary keystroke..." on right Window
- 2) The attacker presses a magic key combination, till enough crypto material has been sniffed over RF
--> indicated by ".....Stored enough encrypted data" on right Window
- 3) The attacker script injects arbitrary keystrokes in 5-time with a 5 second delay between each injection

Note:

Step 2 is only needed once, if successfully carried out, the stored keymaterial could be used for successive keystroke injections, even if keyboard, dongle or host are "power resetted".

Note 2:

I don't give further details, till Logitech investigated the issue (still writing on the report).

Typing to get keyboard detected ... already found|

The right window is a terminal window titled "root@who-knows: ~/jacking/mjackit". It shows the output of a script:

```
root@who-knows:~/jacking/mjackit# go run .
EP In ep #1 IN (address 0x81) bulk [64 bytes]
EP Out ep #1 OUT (address 0x01) bulk [64 bytes]
Valid ESB sniffed on channel 65: c7 94 f2 41
Try to find dongle for potential device e2:c7:94:f2:41...
Received empty ack from dongle e2:c7:94:f2:00 on channel 65
!! Found logitech device, waiting for arbitrary keystrokes to calculate injection data

Found dongle for device e2:c7:94:f2:41 on channel 65, waiting for traffic
```

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):

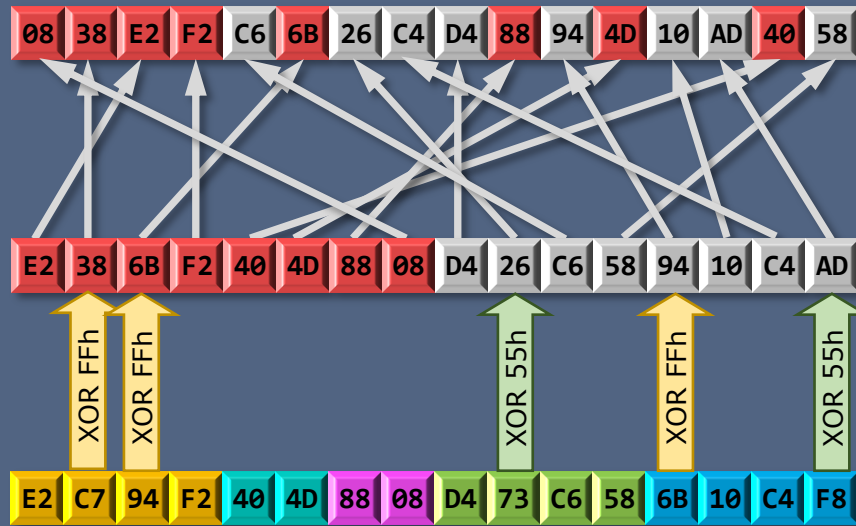
08	38	E2	F2	18	6B	7E	ED	F9	88	B6	4D	A3	8E	40	CE
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

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):

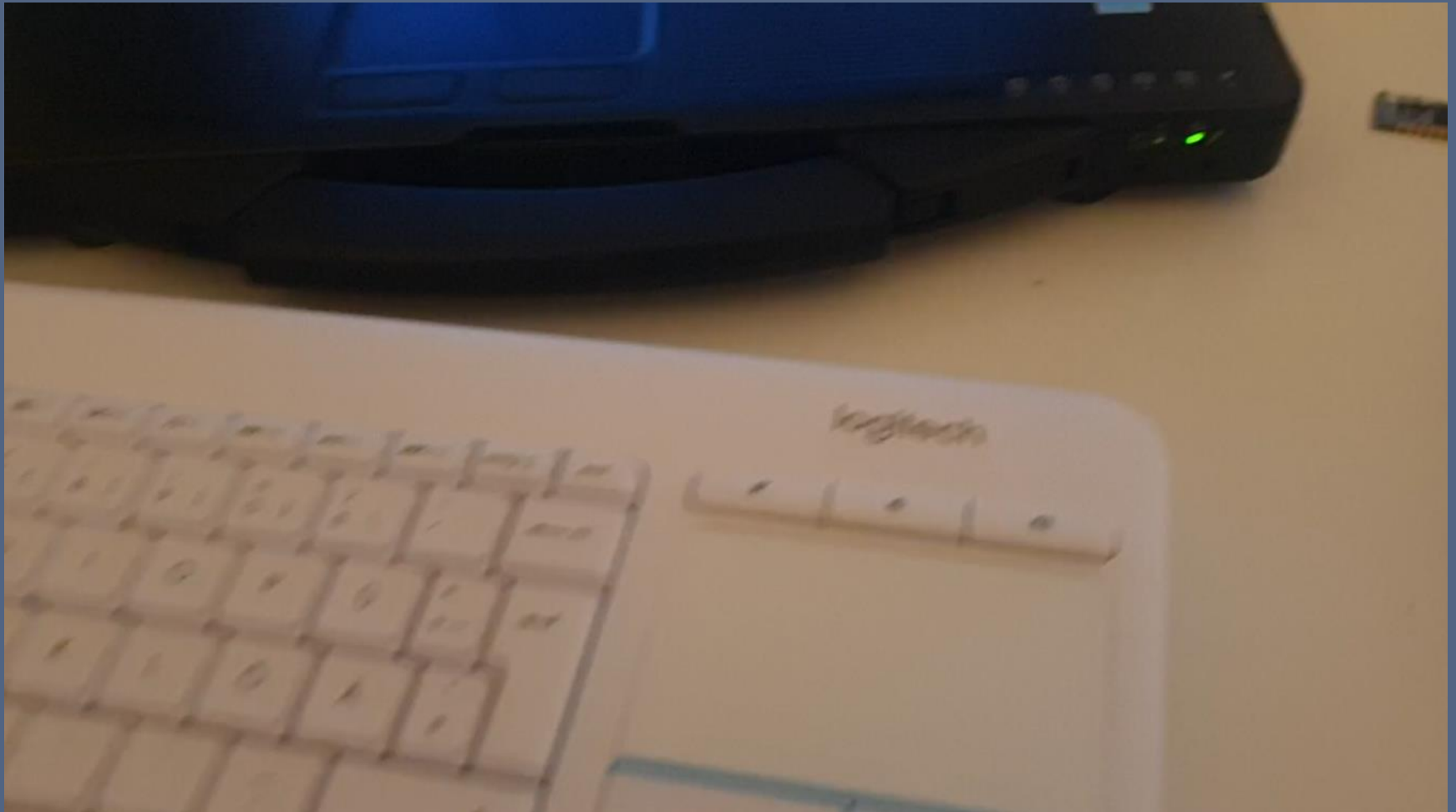
D4 73 C6 58

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>



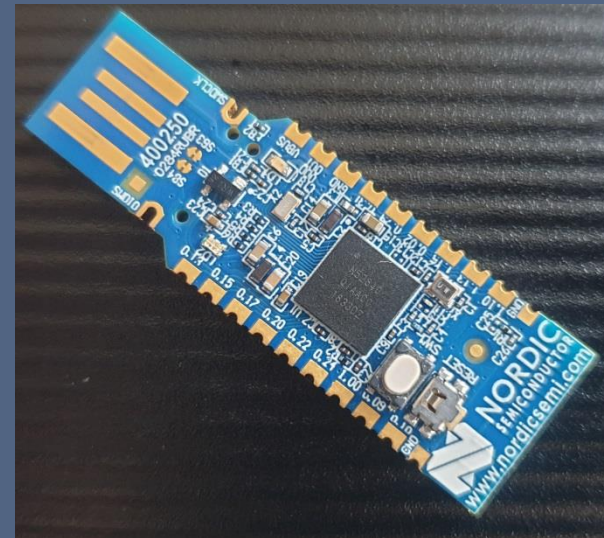
Impact

- 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