Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

WAMOS 2018
Common Attack Vectors of IoT Devices

09.08.2018

Alexios Karagiozidis

Motivation

- Botnetworks and Internet attacks increased rapidly

Examples of security issues:

- Mirai-Bot-Network
- CVE-2018-10967, bufferoverflow via malicous HTTP-request, D-Link DIR-816
- CVE-2015-2887, Backdoor Credentials, iBaby M3S
- CVE-2015-2888, Authentication-Bypass, Internet-Viewing-System
- CVE-2016-5054, Replay-Attack, Osram Lightify Home

There are permanently security issues found with IoT Devices

Agenda

1. **Arbitrary Code Execution/Return-Oriented-Programming**
2. **Reverse Engineering**
3. **Fault Injections**
4. **Analyzing Signals with SDRs**
5. **Conclusion**

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

01
Arbitrary Code Execution/ROP

Harvard-Architecture

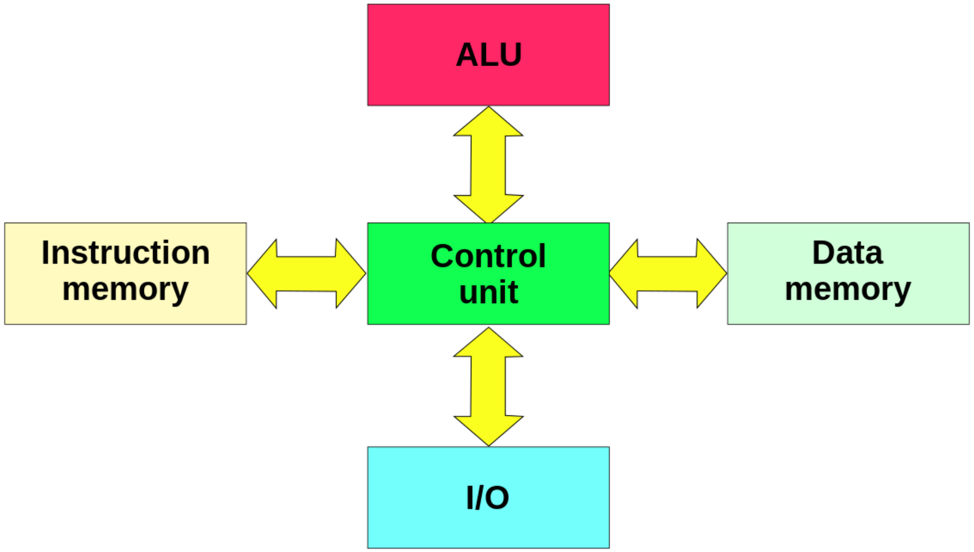


Fig. 1: Shematic of Harvard-Architecture.[1]

Attacks differ from Neumann as x86

- Code and Data are seperated
- Stack is unexecutable
- Most IoT devices use a modified Harvard-Architecture

=> Traditional attack doesn‘t work

For arbitrary code execution only code from Instruction Memory can be used

1: http://www.cs.jhu.edu/~jorgev/cs333/usbkey/uC_3.JPG

Return-Oriented-Programming

- Bufferoverflow-vulnerability required

ROP gadget

- Sequence of instructions terminated by a free return or branch instruction

ROP chain

- Sequence of adresses of ROP gadgets

Return-to-libc

- Simplest form of a ROP
- Adress of System() is placed onto the stack instead of code together with argument

ROP can be used to bypass a non-executable-stack

Why ret2libc doesn‘t work on ARM

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- returns on ARM are performed manually (Load-and-Store-Arch.)

Load and Store-Architecture

- Values must be loaded into registers to operate on them
- No instruction directly operates on values in memory

register	description
R0 to R10	Used for arguments
R13	Stack-Pointer
R14	Link-Register
R15	Program-Counter

Tab. 1: ARM-Registers.

An attacker has to setup arguments and registers manually

Return-to-Zero-Protection

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Presented by Itzhak Avraham in 2009
- Applies ret2libc to ARM

```
ldm sp, r0 , r1
add sp, sp, #12
pop lr
bx lr
```

```
sub sp, fp, #4
pop{fp, pc}
```

- First ROP gadget can be used for loading arguments
- Adresses of gadgets and used arguments have to be placed at the right place on stack

ROPs mustn‘t change adresses and depend from compiler-options

ROP chains on AVR

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

First published worm for Wireless-Sensor-Network (ATmega128s) by Franc Aurellion (2010):

- IP packets with malicious code send to node
- Last packet causes overflow and places ROP-chain on stack
- ROP-chain consists of *SPM* instruction and copies bytes from data to program memory
- Compromised node sends same packets to next node

Trough a ROP chain also a code-injection can be performed on AVRs with a bootloader

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

02
Reverse-Engineering
Software- and Hardware

Reverse-Engineering

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Competitor can copy functionalities
- Attacker can create a malicious firmware (and resell the device)

Software:

- can be searched for vulnerabilities
- Functionalities or security-related routines can be analyzed

Hardware:

- Sniffing on Bus to get (more) information
- Dump memory directly from the device

Reverse-Engineering is essential for finding security issues or creating exploits

Firmware Analysis

- Firmware contains all software-components of an embedded-device (Bootloader, Kernel, Filesystem…)

- Signatures for headers or components can be identified

- filesystem can be searched for passwords, API keys, private certificates or be backdoored

- Individual binaries or fimware itself can be emulated with Qemu and GDB

=> Firmware-Modification-Kit and Firmware-Analysis-Tool can automate process

Through Firmware Analysis software components can be identified and analyzed

Firmware Analysis

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

```
> binwalk Dlink_firmware.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
48            0x30            Unix path: /dev/mtdblock/2
96            0x60            uImage header, header size: 64 bytes, header CRC: 0x7FE9E826, created: 2010-1
1-23 11:58:41, image size: 878029 bytes, Data Address: 0x80000000, Entry Point: 0x802B5000, data CRC: 0x7C
CAE85, OS: Linux, CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image name: "Linux Kerne
 Image"
160           0xA0            LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes, unco
mpressed size: 2956312 bytes
917600        0xE0060         PackImg section delimiter tag, little endian size: 7348736 bytes; big endian
size: 2256896 bytes
917632        0xE0080         Squashfs filesystem, little endian, non-standard signature, version 3.0, siz
: 2256151 bytes, 1119 inodes, blocksize: 65536 bytes, created: 2010-11-23 11:58:47
```

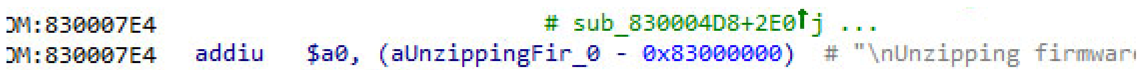
```
alex@alex-virtual-machine:~/Schreibtisch$ dd if=Dlink_firmware.bin skip=917632 bs=1 of=fs_dlink
bin  dev  etc  home  htdocs  lib  mnt  proc  sbin  sys  tmp  usr  var  www
alex@alex-virtual-machine:~/Schreibtisch/squashfs-root$ cd www && ls
__adv_app.php                  comm                       permission_deny.php        tools_
adv_app.php                    conninfo.php               pic                        tools_
adv_apx.php                    DevInfo.php                post_login.xml             tools_
__adv_firewall_httpallow.php   DevInfo.txt                router_info.xml            tools_
__adv_firewall.php             down_limit_info.php        __schedule_combobox.php    tsyslo
adv_firewall.php               down_threshold_info.php    session_full.php           up_lim
__adv_firewall_pingallow.php   do_wps.php                 set_temp_nodes.php         up_thr
```

Fig. 3: Firmware scan and filesystem-extraction

To avoid reverse-engineering firmware is usually obfuscated

Dissasembling

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim



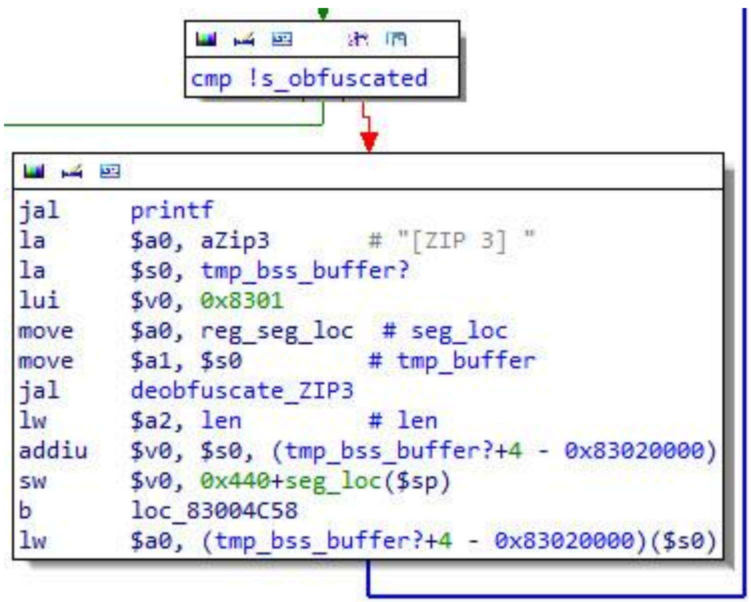


Fig. 4: Dissasembled Deobfuscation-Routine with IDA

- Architecture can be identified
- used instructions can be analyzed
- function calls and program-flow can be traced with known entry-point

⇒ **IDA or Radare can automate and visualize part of this process**

Through dissasembling an attacker can search for backdoors or identifying and bypassing security-related functions

Using hardware interfaces

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Logic-Analyzer:

- Can be used to identify protocols and connectors
- Can be used to sniff on Bus lines as SPI connection between CPU and external Memories

Dumping flash:

- Through JTAG or SPI (connectors)
- Desoldering the Chip and read-out with programming device

Fig. 5: Captured transmission of UART-interface with a Logic-Analyzer (example).

External interfaces or components are additional target surfaces

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

03
Fault-Injections
Overclock- and Powerglitch

Overclock-Glitch

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Frequency of clock is increased for a short period of time
- Frequency has to be a factor of max. specified by manufacturer

⇒ Used by Chris Gerlinsky in 2010 to skip Copy-Read-Protection on LPC-series

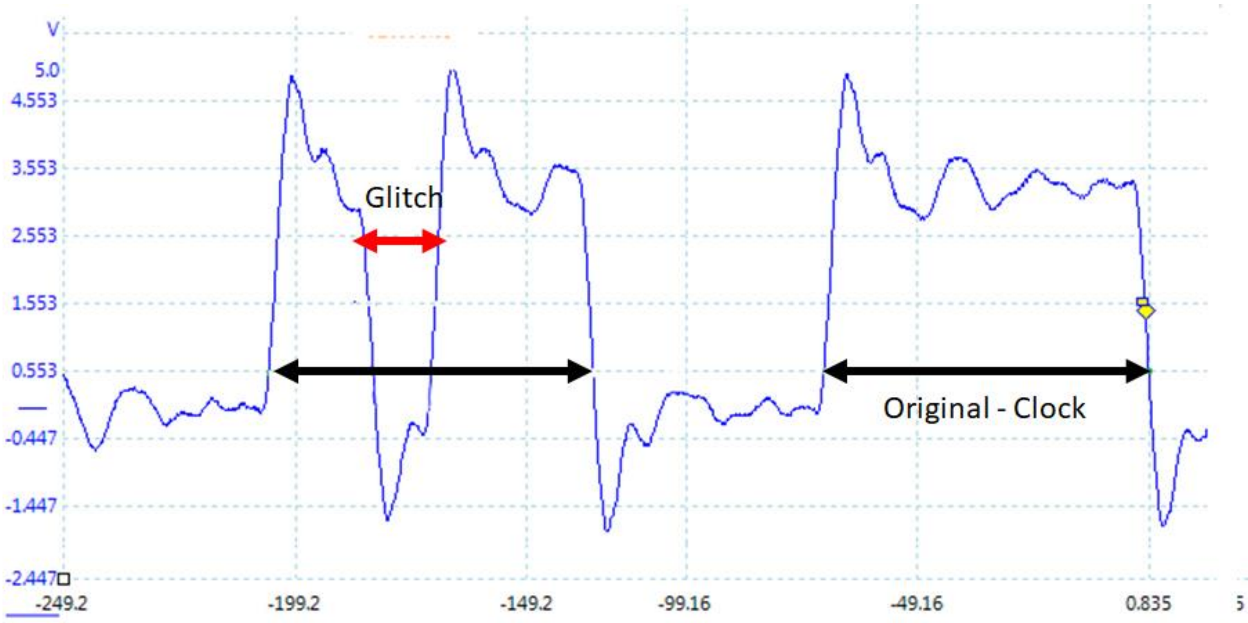


Fig. 6: One glitched and normal clock-pulse for ATmega128P.

With an overclock glitch instructions can be skipped

Fault-Injections

Power-Glitch

- Supply voltage is changed rapidly
- Can affect Amplitude change in variable time

⇒ On Atmega128P can be performed by turning suppy on- and off at 12Mhz

Fault-Injections

- Timing- and sidechannelanalysis are required
- Can be done randomly while monitoring interfaces

⇒ FPGAs are cheap tools for glitches against uCs as they can reach higher frequencies

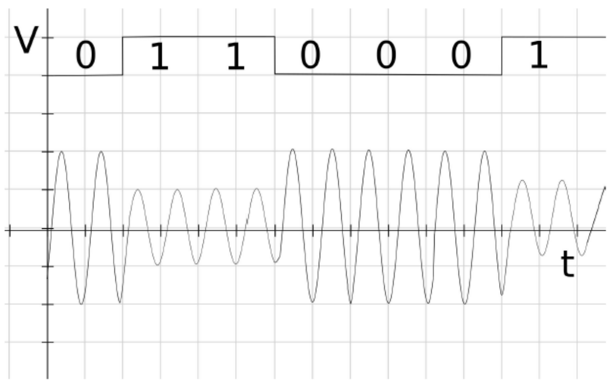
Glitches affect a wide range of uCs and are cheap to perform

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

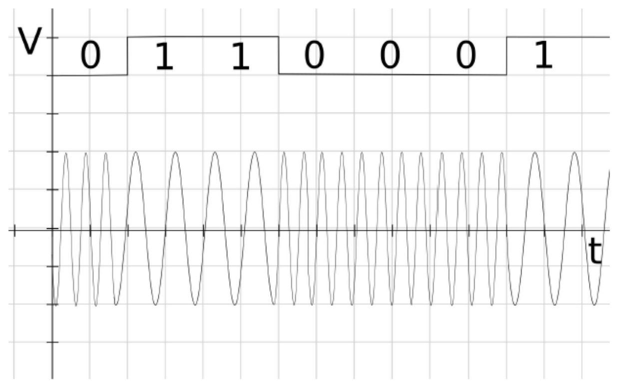
04
Analyzing Signals with SDR

Software-Defined-Radios and wireless transmission

- Hardware takes only care of receiving and transmitting signals
- Signal processing itself is done by soft- or firmware
- Many Open-Source available



Amplitude-Shift-Keying



Frequency-Shift-Keying

Fig. 6: Common modulations for wireless-transmission.

SDRs allow flexible and fast analysis of different wireless signals with the same device

Capturing and Replaying a Signal

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Requirements for capturing a signal

- Frequency
- Bandwith and Sample-Rate
- Frequency or Channel-hopping

Requirements for blind replaying a signal

- Captured or recorded signal
- SDR with transmit capability
- Proper Software (ex.: GNURadio)

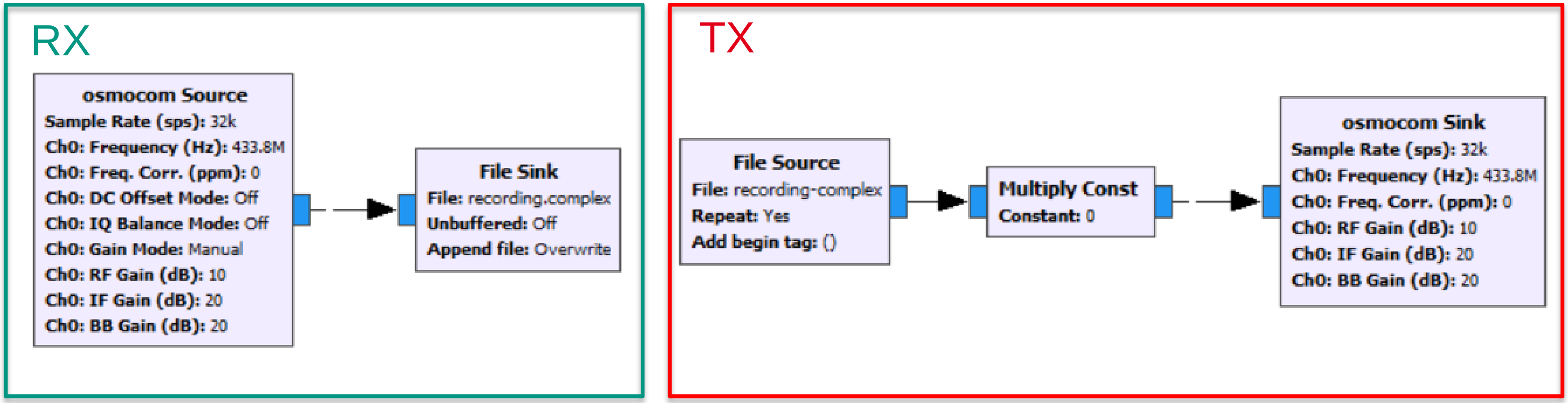


Fig. 7: GNURadio.Flowgraph for recording and replaying a Signal.

Simple replay-attacks affect garage-openers, wireless-bells or simple sensor-nodes

Analyzing Signals

Requirements spoofing commands

- Modulation of signal
- Data/Symbol-rate
- protocol-analysis

With demodulated signal further protocol-analysis can be performed and data as ASCII or HEX extracted

Selected security-related SDR Open-Source (for standarized) wireless-protocols[2]:

- ble_dump
- SecBee (based on killerbee)
- EZ-Wave
- GPS-SDR-SIM
- OpenBTS, OpenLTE

For non-standarized protocols manual analysis has to be performed

2: on GitHub available

Demodulating a signal

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

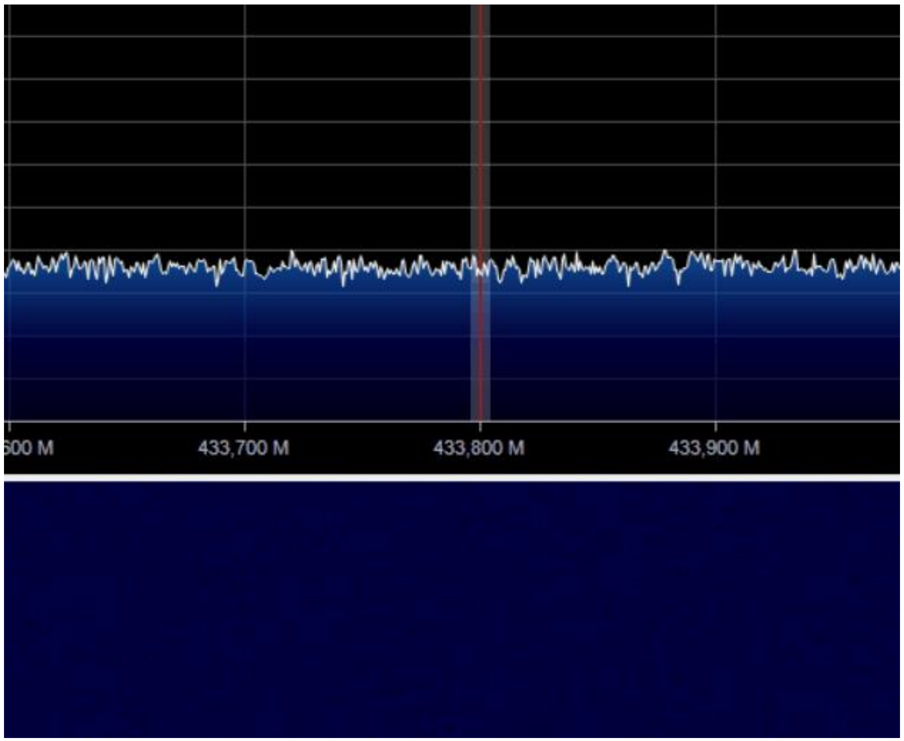


Fig. 9: Spectrum while up-Button of presenter is pressed.

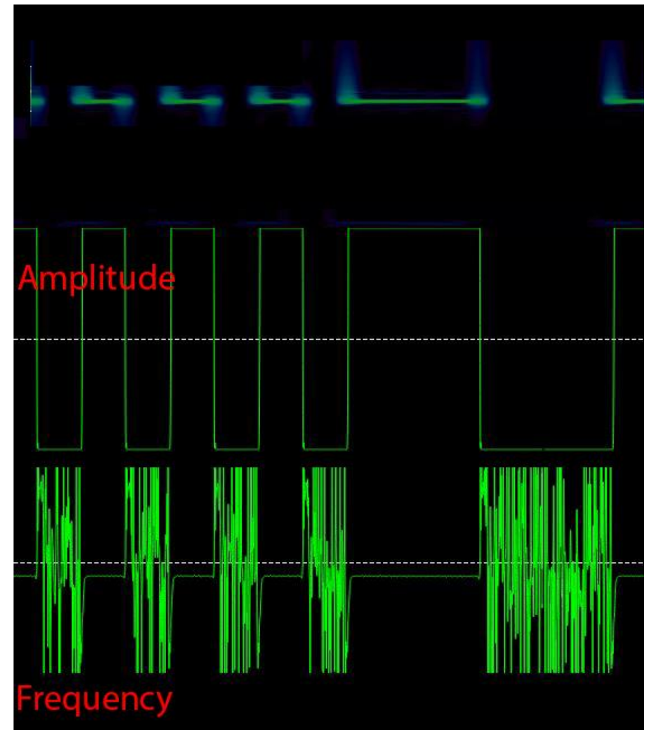


Fig. 10: Demodulation of recorded presenter-control with dspectrum

Unencrypted (simple) wireless-transmission can be broken in a short time

Demodulating a signal

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

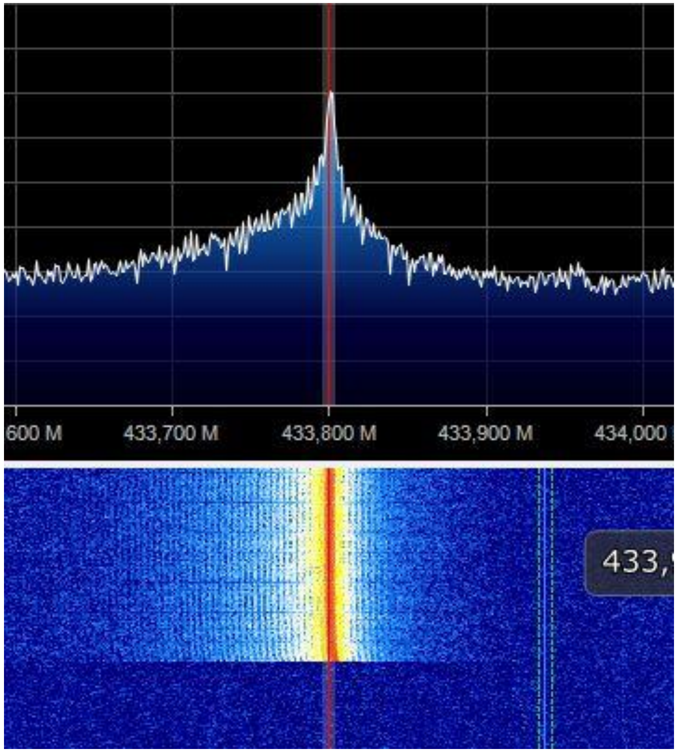


Fig. 9: Spectrum while up-Button of presenter is pressed.

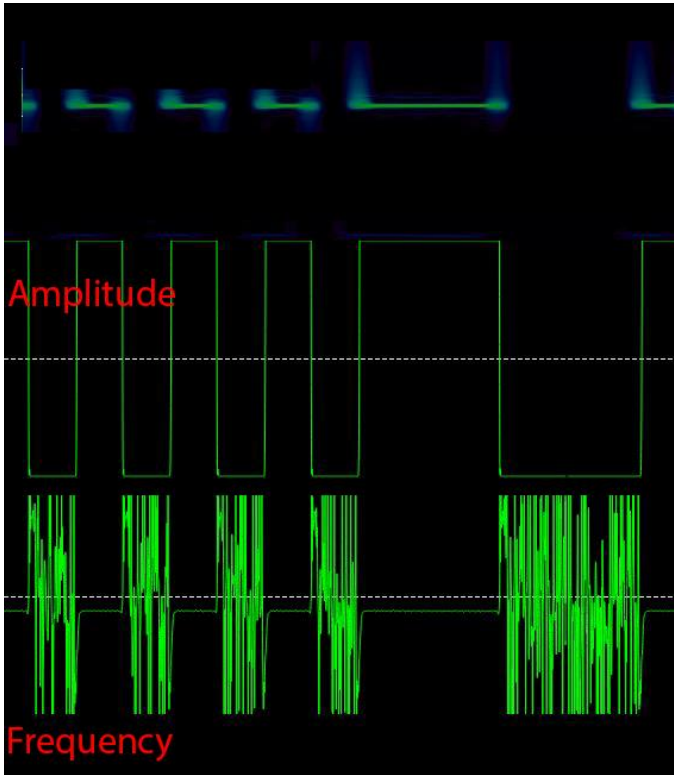


Fig. 10: Demodulation of recorded presenter-control with dspectrum

Unencrypted (simple) wireless-transmission can be broken in a short time

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

06
Conclusion

Conclusion

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Vectors can be used independently or be combined
- Many Open-Source-Software available keeping time expenditure low
- Inexpensible Hardware for hardware or wireless related attacks
- Fully implemented mitigations would make devices too expensive

Security-Analysis of IoT Devices is recommended

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Questions & Answers