A security assessment of Z-Wave devices and replay attack vulnerability.

GIAC (GSEC) Gold Certification

Author: Mark Devito, nobel.gas@gmail.com
Advisor: Dr. Johannes Ullrich

Accepted: August 31st, 2016

Abstract

Within many modern homes, there exists a compelling array of vulnerable wireless devices. These devices present the potential for unauthorized access to networks, personal data and even the physical home itself. The threat originates from the Internet-connected devices, a ubiquitous collection of devices the consumer market dubbed the Internet of Things (IoT). IoT devices utilize a variety of communication protocols; a replay attack against the Z-Wave protocol was accomplished and demonstrated at ShmooCon 2016. The attack was carried out using two HackRF radios. This paper attempts to conduct a similar attack but employing a $35 US SDR, a $130 US sub-1Ghz dongle, and readily available Open Source applications, instead of the more expensive HackRF hardware.

Mark Devito: nobel.gas@gmail.com

1. Introduction

Almost any technology periodical or website published within the last two years will likely include multiple articles discussing Internet-connected or networked devices with embedded operating systems. These devices, now referred to as the Internet of Things (IoT), are expected to reach as many as 100 billion devices by 2020 (Klubnikin, 2015). Although the predicted number of devices varies widely among published works, one element is consistent across all articles utilized in this research; there exists a critical susceptibility of most devices, to both direct access and remote cyber-attack.

The spectrum of IoT device types deployed is broad. Spread across several industry sectors, and at a variety of sophistication levels, the IoT includes but is not limited to, wearable technologies, toys, physical security devices such as locks, HVAC systems, automotive systems, video surveillance, externally worn and implanted medical devices, and home entertainment. The breadth of devices is further widened within the device categories by 1000's of individual device types. As stated by U.S. Federal Trade Commission Chairwoman Edith Ramirez, the threats to IoT include "ubiquitous data collection, potential for unexpected uses of consumer data, [and] heightened security risks." (Hajdarbegovic, 2015) The risk is further compounded by two factors, 1) "Consumers do not perceive value in security and privacy" (Porup, 2016) and 2) device manufacturers poor implementation or even exclusion of security features within products (Klubnikin, 2015).

2. IoT Assessment Research

2.1. State of the IoT Security

Regardless of an IoT device's function, all devices maintain one commonality, communication. IoT devices communicate via two main vectors, radio frequency (RF) signals and Internet gateway devices. Both avenues of communication offer a variety of attack vectors against IoT.

IoT RF communication protocols include 802.11 ("WiFi"), Bluetooth, Zigbee, Z-Wave, Near Field Communication, and 433MHz; all are susceptible to RF signal intercept by software defined radio (SDR) or other similar devices. Each standard does, however, carry a different level of over-the-air security capability.

At present, it is the poor implementation of security within this RF communication that generates a great deal of concern regarding IoT security. The poor implementation or total absence of security and encryption is well documented across many device types. Although there is evidence that points toward poor implementation of security by manufacturers, the securing of these devices is made more difficult by a lack of consensus on "how best to implement security in IoT at the device, network, and system levels." (Wind River, 2015, p. 3)

At the consumer level, user interaction with or control of IoT devices is most often accomplished via mobile device applications. Control interface applications offer many additional attack vectors. Because most consumer mobile device control applications do not speak directly to the IoT device, they must communicate through an intermediary server or service connected via the Internet, thus creating additional attack surfaces for attackers.

In addition to RF vulnerabilities, many home automation IoT devices use a centralized controller, sometimes referred to as the smart hub. This is a bridge between IoT devices and the Internet. It provides a point of egress for sensor and device data information to external endpoints (Figure 1). A Symantec study reported a vulnerability in the Lightwave RF brand Smart Hub, a popular brand of IoT smart home gateway device in the UK. To check for firmware updates, this smart home gateway sends an unencrypted polling signal to a Trivial File Transfer Protocol (TFTP) server every 15 minutes. This scenario created an opportunity for an Address Resolution Protocol (ARP) poisoning attack causing the smart hub to send its traffic to a bogus TFTP server, thus creating a man-in-the–middle (MITM) attack platform. (Symantec, 2015) This demonstrates that IoT vulnerabilities are not limited to RF signal interception but traditional network intrusion methods as well. Traditional network penetration techniques, such as cloud polling, direct connection, cloud infrastructure, and malware attacks, are also useful against IoT devices. (Symantec, 2015)

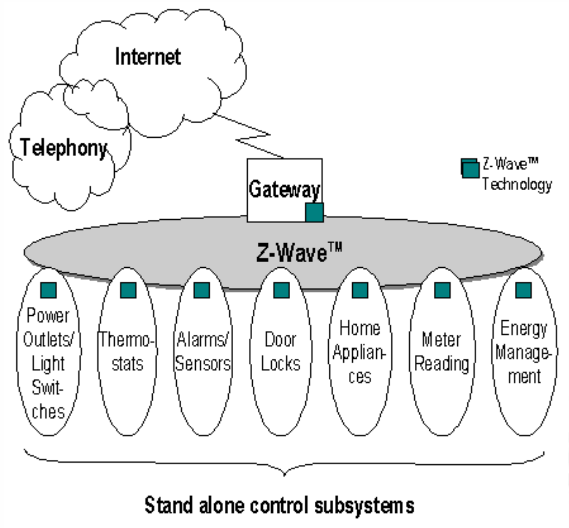


Figure 1: Z-Wave to Internet connectivity. (Jorgensen, 2005)

Although not within the scope of this paper, it is important to note that threats to IoT devices also exist from interruption via Jamming. Beyond knowing the devices operating frequency, there is no need to sniff, demodulate, or abuse the device's signals. Whether vehicles, home security systems or sophisticated industrial systems, if radio frequencies are used to communicate either between devices or to a central control system, they are susceptible to jamming to varying degrees. (Tripwire, 2015, pp. 3-4)

2.2. The Future of IoT Security

As with the evolution of computer network security, the evolution of IoT security continues to develop. According to Vision Mobile's IoT Megatrends 2016 report, there were 4.5 million IoT developers in 2015 (Asay, 2016). This number of developers is driving a staggering number of IoT innovations deployed within the consumer and corporate sectors. In comparison, computer security evolution had 25 years to reach its present level. It is unrealistic to expect the IoT industry to find "…some entirely new, revolutionary security solution…uniquely tailored to IoT…" (Wind River, 2015, p. 3).

3. Z-Wave Communication Standard and Protocol

3.1. Network Communication

Unlike some other IoT communication protocols, Z-Wave is a connection-oriented protocol. This means the nodes on the network acknowledge receipt of messages from the network controller. To accomplish this, Z-Wave uses the concept of mesh networking. A Z-Wave mesh network node can forward commands to and receive responses from neighboring nodes (Figure 2). By overlapping radio zones, the network is capable of overcoming a node failure by routing around the failed point. Each node can route to a maximum of four devices giving the system a maximum range of 400 feet.

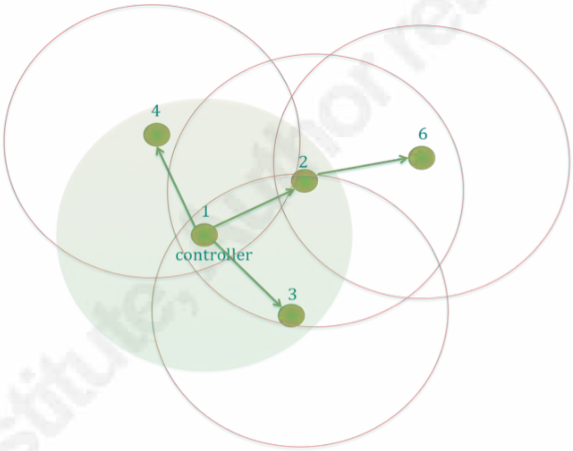


Figure 2: Mesh networking. (DomotiGa.nl, 2011, p. 35)

Illustrated in Figure 3 is a representation of the potential routes between Z-Wave nodes. The Z-Wave controller maintains a routing table of each node and their respective neighbors. The routing table is created when a device is added to a Z-Wave network; this is referred to as the inclusion process. When nodes are included or removed from the network, the network administrator can trigger a request for device identification to construct an updated routing table.

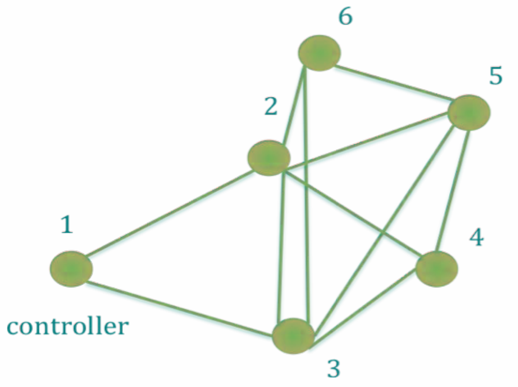


Figure 3: Example of a mesh network. (DomotiGa.nl, 2011, p. 37)

3.2 Protocol Stack

The Z-Wave protocol stack layers are similar to those of the OSI Model; however, because of Z-Wave's RF component, there are some variances (Figure 4). The stack consists of the Physical layer, MAC layer, Transport layer, Network layer, and Application layer. If encryption is used, the encryption layer will exist between the network and application layers (Behrang Fouladi, 2013, p. 20).

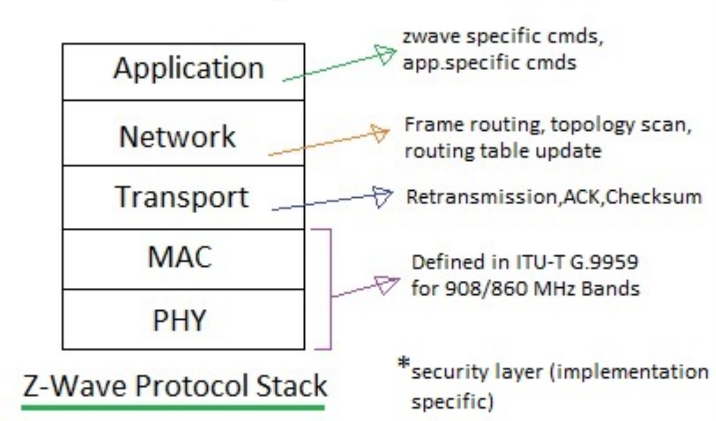

Application
Network
Transport
MAC
PHY
Z-Wave Protocol Stack
zwave specific cmds, app.specific cmds
Frame routing, topology scan, routing table update
Retransmission,ACK,Checksum
Defined in ITU-T G.9959 for 908/860 MHz Bands
*security layer (implementation specific)

Figure 4: Z-Wave Protocol Stack. (Jorgensen, 2005)

3.2. Physical Layer/MAC Layer

The physical layer handles the task of modulation, demodulation, and coding of frame data, radio activation/deactivation, radio frequency selection, transmission and reception of the MAC data frame as well as link quality assessment.

Z-Wave devices operate at 908.42Mhz in the USA and 868.42Mhz in Europe (RF Wireless World, 2012, p. 1). The US frequency falls within the FCC designated Industrial-Scientific-Medical (ISM) band. Z-Wave uses either Frequency Shift Keying (FSK) or Gaussian Frequency Shift Keying (GFSK) for signal modulation allowing for data transmission rates of 9.6Kbps and 40Kbps for FSK and 100Kbs for GFSK. One advantage to frequency shift keying is a better immunity to RF noise, a problem that can impact amplitude-modulated (AM) signals. Basic FSK modulation is accomplished by oscillating two distinct frequencies, one frequency representing a binary zero and the other representing a binary one. These two frequencies are referred to as a mark and space, respectively (Watson-Johnson Company, 1980) (Figure 5).

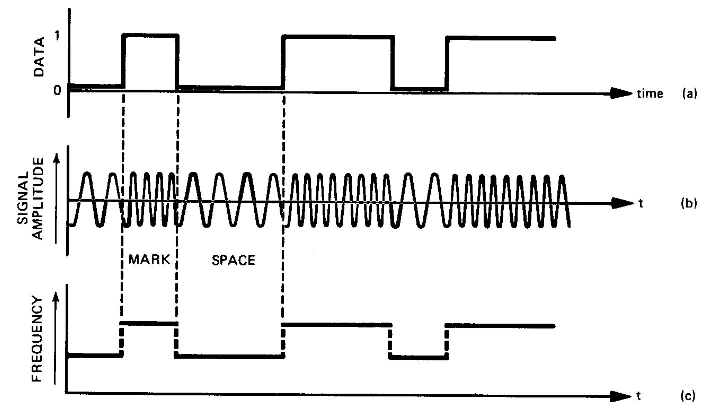


Figure 1. FSK modulation. Binary data (a) frequency modulates the carrier to produce the FSK signal (b) which has the frequency characteristic (c).

Figure 5: FSK representation (Watson-Johnson Company, 1980)

To encode data within the modulated RF signal, Z-Wave uses Manchester or Non-Return-to-Zero (NRZ) encoding. The specific encoding scheme used is dependent on the data transmission rate with Manchester and NRZ utilized in 9.6Kbps FSK and 40Kbps FSK, respectively. GFSK uses NRZ at 100Kbps (Figure 6).

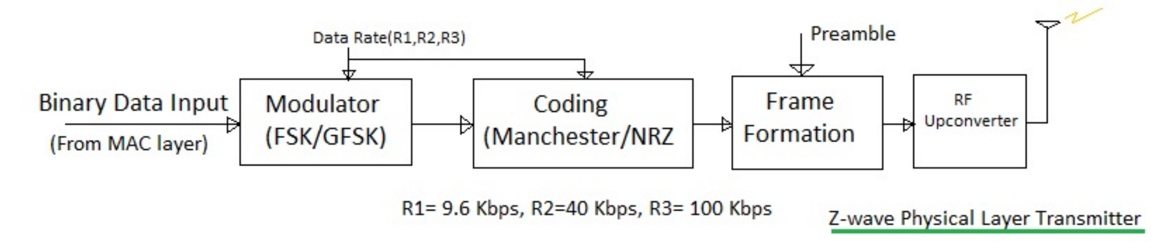


Figure 6: Z-Wave Physical Layer constructs flow (RF Wireless World, 2012)

Mark Devito: nobel.gas@gmail.com

The MAC layer works in concert with the physical layer and is often depicted as a single layer in some protocol stack diagrams. Within the physical layer is the MAC Data frame. With the exception of the Home and Node IDs, which are managed by the MAC layer, the MAC Data frame contains information from the transport layer; this includes frame control information, the destination ID, data length, payload, and checksum (Figure 8).

The Home ID is four-bytes (32-bit) in length and unique to each Z-Wave network. The controller randomly generates a Home ID during every factory default reset and is not modifiable by the user. The uniqueness of the Home ID allows multiple Z-Wave networks to operate within proximity of each other and avoid network crosstalk (RF Wireless World, 2012, pp. 2-3 of MAC layer sub-section document).

The Node ID is one byte (8 bits) long and is unique for each device on a network. The controller device assigns the Node ID during the inclusion process. The Node ID is unique among devices on the logical network.

3.2.1 Transport Layer

The Frame Control is two bytes in length. It, along with the Header type sub-field, defines the frame-type: singlecast, ACK frame, multicast, or broadcast. Singlecast frames are intended for one destination node and are acknowledged by the node to ensure reception. An ACK frame is the same as a singlecast but absent a payload and is the destination nodes acknowledgment of receipt of the transmission. Multicast frames are sent to more than one node none of which acknowledge the receipt. Broadcast frames are sent to all nodes on the network and no acknowledgment is provided (RF Wireless World, 2012).

The length field is a one-byte field describing the length of the entire MAC Service Data Unit (MPDU), labeled as Transport frame in Figure 7.

The Destination Node ID is the two-byte Node ID of the device for which the transmission is intended.

The payload or data configuration is defined by frame type. The frame type data will also contain a payload comprised of the information passed by the application layer.

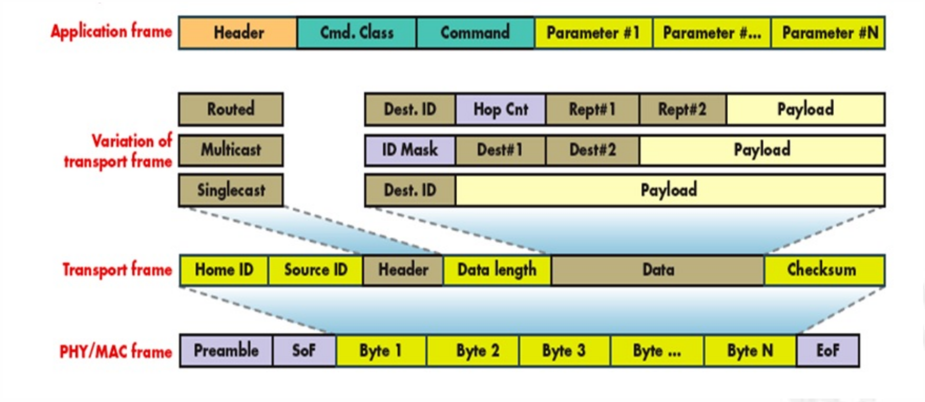


Figure 7: Z-Wave protocol stack layers. (RF Wireless World, 2012, p. 5)

3.2.2 Network Layer

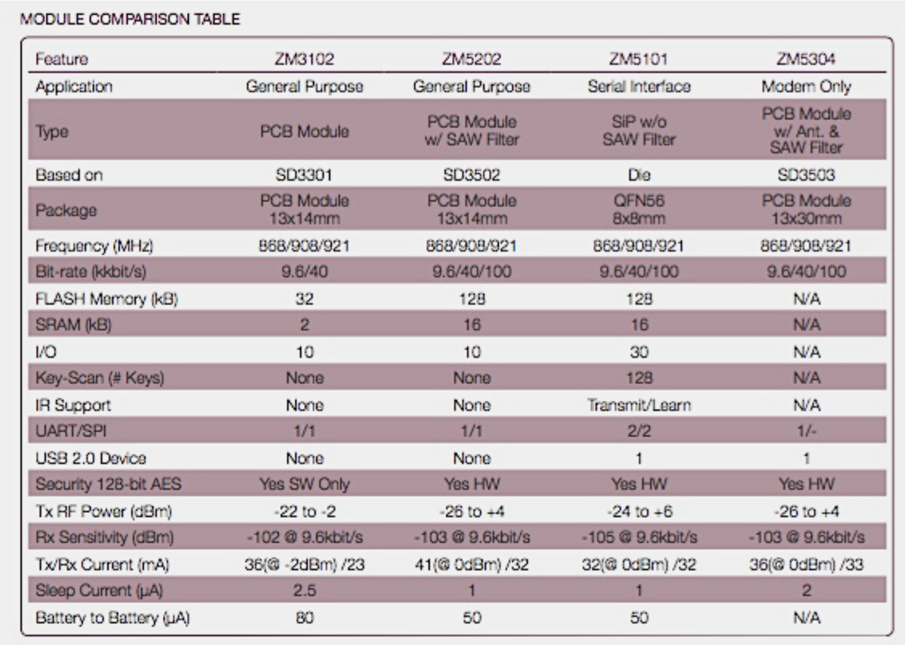
The network layer is responsible for calculating packet routes based on the network routing table (See Section 3.1) This layer is also responsible for topology scans and updating the routing table. The network layer consists of two frame types, Routed Singlecast and Routed Acknowledge. These serve the same function as the frame types of the same name in the Transport layer (RF Wireless World, 2012).

3.2.3 Application Layer

This layer consists of instructions intended for the destination node. The instructions consist of a command class, commands and command parameters. There are 74 command classes, based the device's functionality. The command class structure is analogous to an object-oriented programming (OOP) structure. Command classes are analogs to OOP object classes and commands are analogs to OOP methods.

3.2.4 Encryption Layer

All Z-Wave devices using the 400 and newer 500 series chips are capable of hardware-based 128-bit AES encryption (Sigma Designs, 2016) (Figure 8). ZM0301 and ZM 3102N chips do not have native, on-chip, encryption capabilities.



MODULE COMPARISON TABLE

Feature	ZM3102	ZM5202	ZM5101	ZM5304
Application	General Purpose	General Purpose	Serial Interface	Modem Only
Type	PCB Module	PCB Module w/ SAW Filter	SiP w/o SAW Filter	PCB Module w/ Ant. & SAW Filter
Based on	SD3301	SD3502	Die	SD3503
Package	PCB Module 13x14mm	PCB Module 13x14mm	QFN56 8x8mm	PCB Module 13x30mm
Frequency (MHz)	868/908/921	868/908/921	868/908/921	868/908/921
Bit-rate (kkbit/s)	9.6/40	9.6/40/100	9.6/40/100	9.6/40/100
FLASH Memory (kB)	32	128	128	N/A
SRAM (kB)	2	16	16	N/A
I/O	10	10	30	N/A
Key-Scan (# Keys)	None	None	128	N/A
IR Support	None	None	Transmit/Learn	N/A
UART/SPI	1/1	1/1	2/2	1/-
USB 2.0 Device	None	None	1	1
Security 128-bit AES	Yes SW Only	Yes HW	Yes HW	Yes HW
Tx RF Power (dBm)	-22 to -2	-26 to +4	-24 to +6	-26 to +4
Rx Sensitivity (dBm)	-102 @ 9.6kbit/s	-103 @ 9.6kbit/s	-105 @ 9.6kbit/s	-103 @ 9.6kbit/s
Tx/Rx Current (mA)	36(@ -2dBm) /23	41(@ 0dBm) /32	32(@ 0dBm) /32	36(@ 0dBm) /33
Sleep Current (µA)	2.5	1	1	2
Battery to Battery (µA)	80	50	50	N/A

Figure 8: Z-Wave module comparison (Sigma Designs, 2016)

Systems with hardware-based encryption capability follow the key exchange routine shown in Figure 9. Encryption keys are established during the inclusion phase when a new node is added to the network. The initial key exchange is not conducted in the clear but uses a pseudo-random number generator on the Z-Wave module to generate the key. The key is subsequently "encrypted by using a hardcoded temporary default key in the chip's firmware before being sent to the" {node} (Behrang Fouladi, 2013).

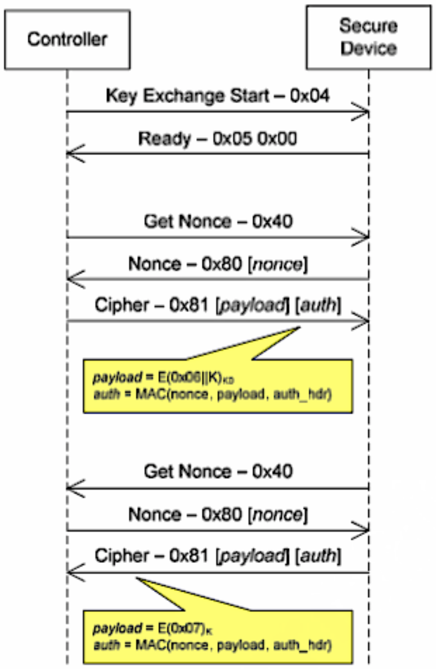

Controller
Secure Device
Key Exchange Start – 0x04
Ready – 0x05 0x00
Get Nonce – 0x40
Nonce – 0x80 [nonce]
Cipher – 0x81 [payload] [auth]
payload = E(0x06||K)KD
auth = MAC(nonce, payload, auth_hdr)
Get Nonce – 0x40
Nonce – 0x80 [nonce]
Cipher – 0x81 [payload] [auth]
payload = E(0x07)K
auth = MAC(nonce, payload, auth_hdr)

Figure 9: Z-Wave Authentication Exchange Process (Behrang Fouladi, 2013)

4. Assessment of Z-Wave Light Switch

4.1. Equipment

For this research, the target device was an Aeon Labs DSC06106ZWUS-Z-Wave Smart Energy Switch (Image 1). This device is marketed as a home automation interface for lights and appliances. In addition to simple on/off commands, the DSC06106ZWUS-Z-Wave Smart Energy Switch can report wattage energy usage to capable Z-Wave gateways or controllers (Aeotec Labs, 2016).

Image 1: Aeon Labs DSC06106ZWUS-Z-Wave Smart Energy Switch

The choice to target a home automation switch was inspired by research conducted by Joseph Hall and Ben Ramsey (Szczys, 2016). Hall and Ramsey hypothesized that due to the absence of encryption on most IoT devices, sufficient havoc could be caused by simple replay attacks. Their research demonstrated how rapid on-off switching of fluorescent lights via Z-Wave device attack could reduce the bulbs life from 30K hours to less than one night (Hall, 2016).

The capture platform system consisted of a Sony Vaio laptop running Windows 8 and VM Ware Player. Kali Linux 2.0 ran within a virtual machine (VM) instance using eight processor cores and 4Gb of RAM. The analysis platform was a mid-2012 Apple Mac Book Pro. Table 1 outlines the hardware and software used in the capture and analysis portions of this research.

	Capture	Analysis	Attack
Software	• Kali Linux 2.0 • GQRX 2.5.3 • GNURadio • Indigo 6 • RF Analyzer	• Audacity 2.1.2 • Baudline	• RfCat
Hardware	• Sony Vaio • RTL-SDR with 900Mhz Antenna (Figure 9) • Samsung Tab S 8.4”	• MacBook Pro (10.11.5, 2.9Ghz i7)	• YardStick One (Figure 10)

Table 1: Software and Hardware used in this research

- RTL-SDR radio dongle (Image 2).

Image 2: RTL-SDR

- Kali Linux (Defensive Security) is a variant of Linux with pre-installed hacking applications.
- GQRX (GQRX Developers) is an Open Source software defined radio (SDR) application designed for use with many common SDR radios.
- "GNU Radio is a free & open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in hobbyist, academic and commercial environments to support both wireless communications research and real-world radio systems." (GNURadio Foundation) GNURadio Companion (GRC) is included in the GNURadio installation but optional for use. It provides a graphical interface to simplify the user experience. GNURadio translates user-developed flow graphs (visual block diagram representations) into a Python-based program.
- Indigo6 (Indigo Domo) is a commercial product which functions as a user interface for direct communication with the Aeotec Z-Stick. (Image 3) This application treats the Z-Stick as a USB-modem and provides communication across the proprietary Z-Wave protocol. Since this software is commercial and was written using the Z-Wave

SDK, it is capable of interacting with both encrypted and non-encrypted devices.

- The Aeotec Z-Stick is a USB based Z-Wave network controller (Image 3).

Image 3: Aeon Labs Aeotec Z-Wave Z-Stick, Gen5 (ZW090)

- Baudline (SigBlips DSP Engineering) is a commercial application with a free for personal use license. It is a sophisticated signal visualization and analysis package for time-frequency display through Fourier transformation.
- RfCat (Atlas0fdoom) is an Open Source application, which communicates with the YARDStick One.
- YARDStick One (Great Scott Gadgets) is a radio dongle to receive sub-1Ghz signals and transmit custom packets within the same frequency range (Image 4).

Image 4: YardStick One

4.2. Experiment Procedure

Experiment procedure	Success
1. Obtain a Z-Wave device controller and a node device.	✔
2. Look up both devices by FCC ID on www.fcc.io or www.fcc.gov.	✔
3. Configure all hardware and software.	✔
a) Build Kali Linux 2.0 VM on Win8 system.	✔
b) Install GQRX, GNURadio, Baudline, and RFCat on and all required dependencies on Linux VM; install Audacity and Indigo 6 on MacBook.	✔
c) Test RTL-SDR with GQRX against a known signal.	✔
d) Configure Z-Wave USB Controller within Indigo 6.	✔
e) Add Z-Wave Smart Energy Switch to Controller's network.	✔
f) Test Z-Wave Smart Energy Switch response to on/off commands with Indigo 6 software.	✔
4. Using MacBook Pro and Indigo 6 send on/off commands to Smart Energy Switch.	✔
5. Using GQRX and RTL-SDR and Kali 2.0, identify the Z-Wave command signals on frequency 908.42Mhz.	✔
6. Attempt to capture raw I/Q data in GQRX.	✔
7. Construct a custom GNURadio flow-graph to capture the Z-Wave signal, demodulate it, and output the date in raw I/Q format to a file.	✔
8. Analyze the captured signals within Baudline.	✔
9. Convert the binary packets hex with the Linux command 'xxd.'	✔
10. Construct a custom packet to replicate the on and off Z-Wave commands.	✖
11. Use RFCat to transmit the custom packets.	✖
a. Re-evaluated capture process.	✔
b. Capture packets using RTL-SDR, Samsung tablet, and RF Analyzer.	✔

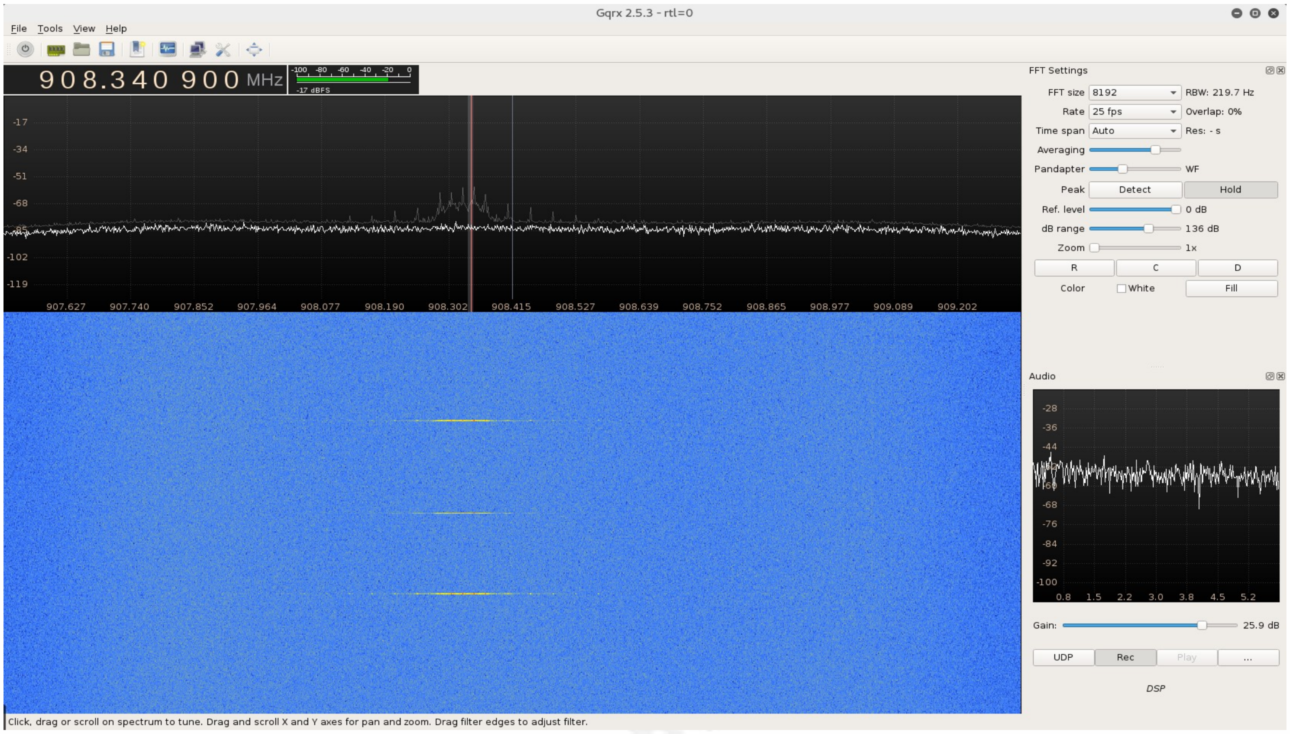
c. Evaluated capture in Baudline and Audacity	✓
d. Reverse Manchester encoding of signal	✓
e. Convert bytes to hex and construct packet	✓
12. Successfully conduct replay attack against the Smart Energy Switch	✖

4.3. Results

The Smart Switch’s FCC ID was used to conduct an FCC database query. Little useful information was available due to the manufacturer’s request for limited public disclosure of proprietary product specifications. The FCC database did include the manufacturer's internal photos of device components, but the component details were obfuscated. A visual inspection of the smart switch circuit board identified the Z-Wave chip as a ZW0301. The product datasheet for a ZW0301 chip was downloaded from the DigiKey website (Zensys, 2007). As described in 3.2.5, the ZW0301 chip only offers encryption at the software level. Based on this, it was assumed the switch did not use encryption.

4.3.1. Initial Capture Method

Several attempts were made to capture I/Q data using GQRX (Image 5). Each capture was evaluated using Audacity. The raw data was imported for conversion into a graphical display (Image 6). Although signals appeared in the GQRX waterfall display, which correlated to the on-off keying of the smart switch, the waveforms did not contain properly formatted FSK modulated signals or Manchester encoded data.



Gqrx 2.5.3 - rtl=0

908.340 900 MHz

FFT Settings

FFT size 8192 RBW: 219.7 Hz

Rate 25 fps Overlap: 0%

Time span Auto Res: - s

Ref. level 0 dB

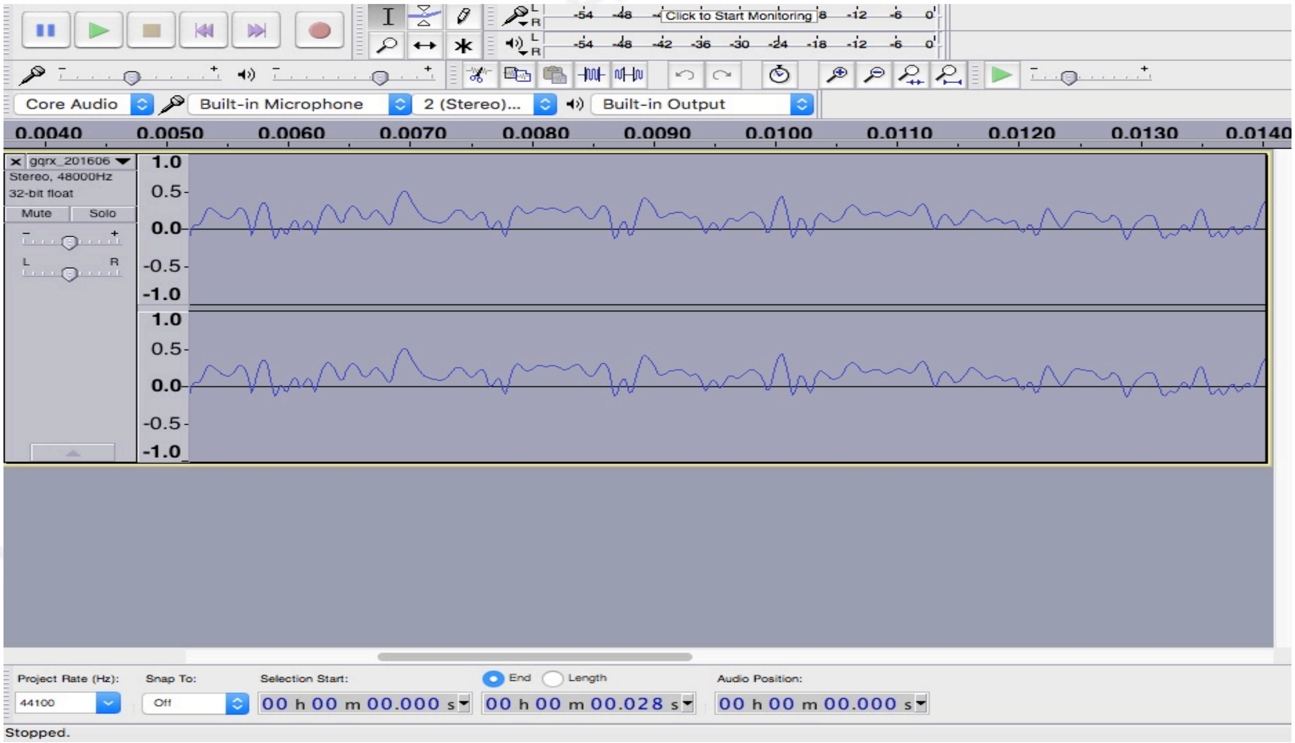
dB range 136 dB

Zoom 1x

Gain: 25.9 dB

Click, drag or scroll on spectrum to tune. Drag and scroll X and Y axes for pan and zoom. Drag filter edges to adjust filter.

Image 5: Z-Wave signal capture



Core Audio | Built-in Microphone | 2 (Stereo)... | Built-in Output

gqrx_201606
Stereo, 48000Hz
32-bit float

Project Rate (Hz): 44100

Snap To: Off

Selection Start: 00 h 00 m 00.000 s

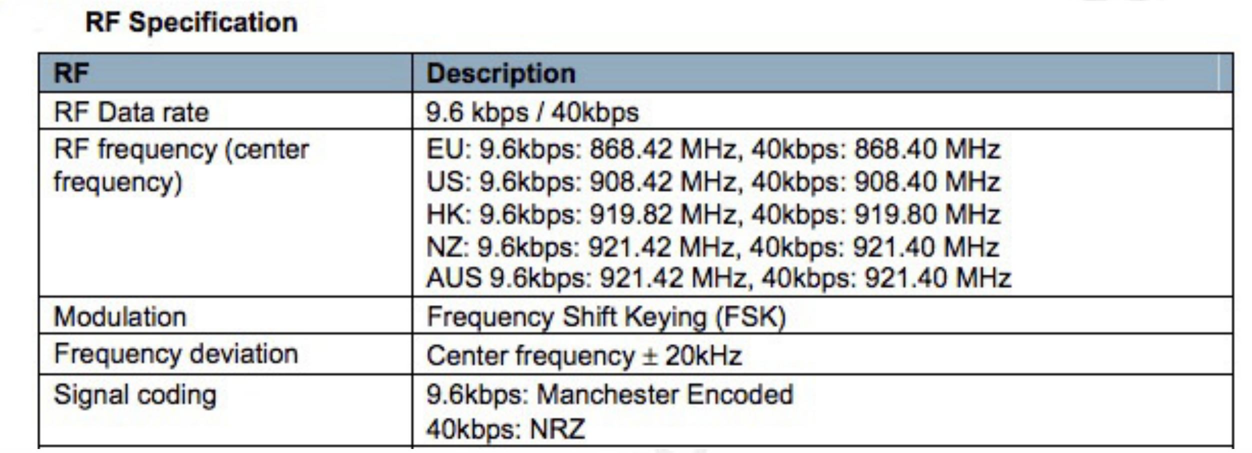
● End ○ Length: 00 h 00 m 00.028 s

Audio Position: 00 h 00 m 00.000 s

Stopped.

Image 6: Audacity display of signal capture

As an alternative to GQRX, a GNURadio flow graph was created to capture the packets in raw I/Q format. Elements of the GNURadio flow graph were configured from the RF specifications shown in Figure 10 (Zensys, 2007). Again, several attempts were made to capture the Z-Wave signal.



RF Specification

RF	Description
RF Data rate	9.6 kbps / 40kbps
RF frequency (center frequency)	EU: 9.6kbps: 868.42 MHz, 40kbps: 868.40 MHz US: 9.6kbps: 908.42 MHz, 40kbps: 908.40 MHz HK: 9.6kbps: 919.82 MHz, 40kbps: 919.80 MHz NZ: 9.6kbps: 921.42 MHz, 40kbps: 921.40 MHz AUS 9.6kbps: 921.42 MHz, 40kbps: 921.40 MHz
Modulation	Frequency Shift Keying (FSK)
Frequency deviation	Center frequency ± 20kHz
Signal coding	9.6kbps: Manchester Encoded 40kbps: NRZ

Figure 10: ZN0301 Chip Specifications (Zensys, 2007)

Despite observing signals within the Fast Fourier Transformation (FFT) plot of GNURadio, which correlated to on-off switching of the smart switch, none of the captured signals contained data, only noise. To confirm this, the capture files were analyzed in Baudline but no identifiable signal was observed. Several variations of GNURadio flow graphs were attempted. They include:

- Use of throttles to downgrade the sample rate.
- Use of Xlating filters to offset the frequency to minimize the effect of DC spike interference.
- Use of a quadrature demodulation element for FSK demodulation.

In an effort to manually identify a preamble, sync word or other identifiable packet sequences all captured signals were converted to hex with the Linux xxd command. No useable information was found.

4.3.2. Alternative Capture Method

One final capture attempt was made using an alternative tool. The RTL-SDR was connected to a Samsung Galaxy Tab S 8.4. Using the application RF Analyzer, an SDR application for Android devices, a signal was observed within the FFT plot and captured in raw I/Q format. This file was further evaluated in Baudline (Image 7) and determined to contain an FSK modulated signal correlating to the Z-Wave on/off command sequence issued from the Indigo 6 Z-Wave controller application.

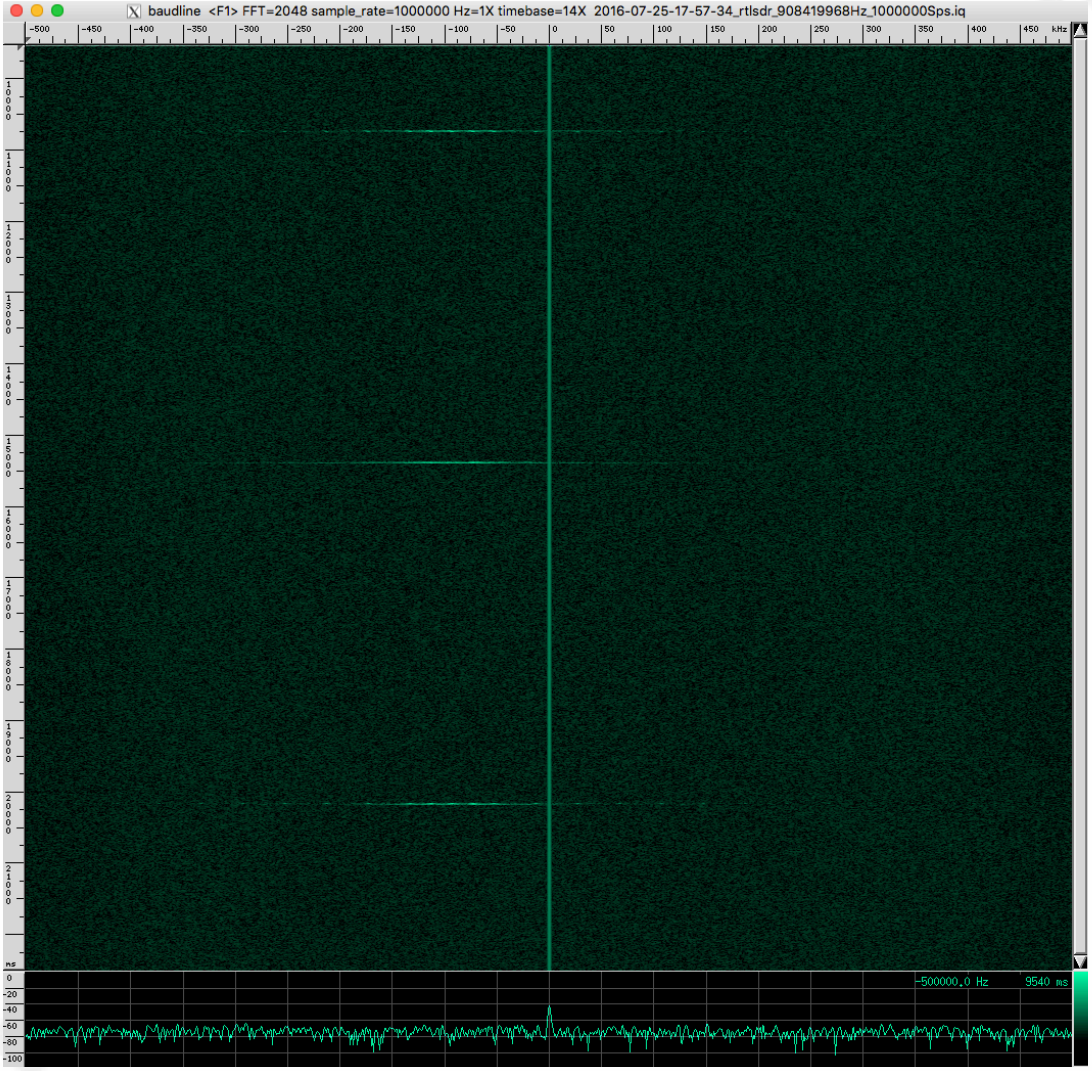
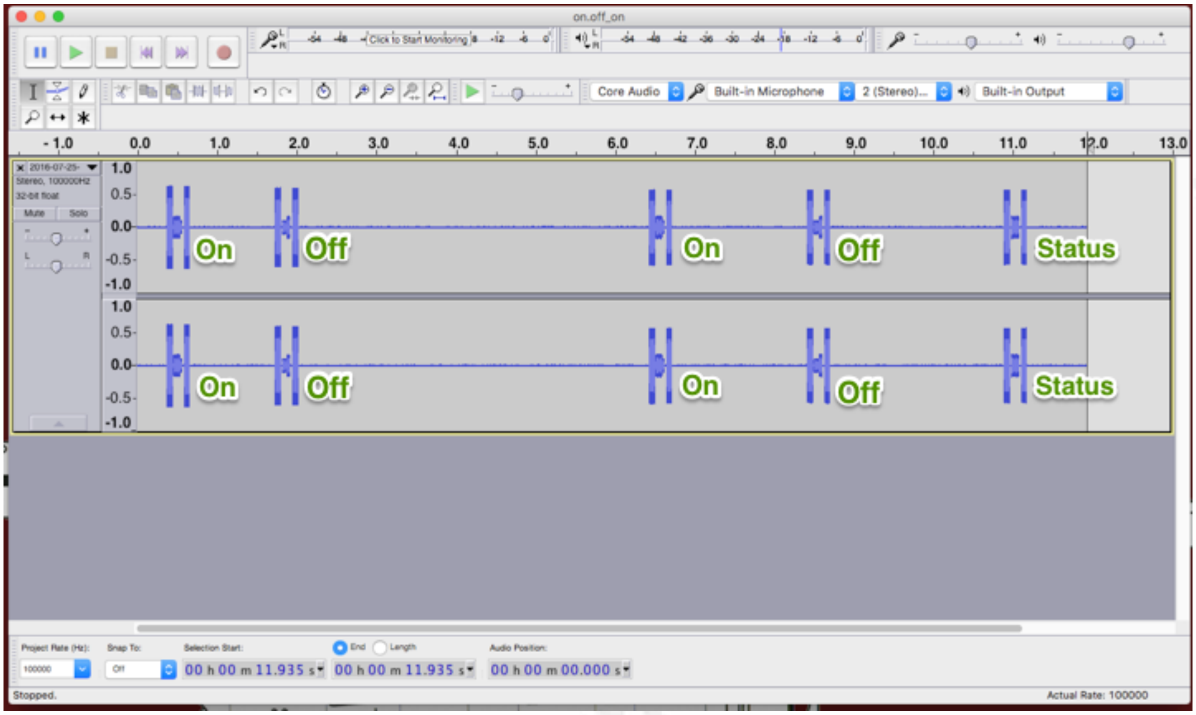


Image 7: Baudline display of captured signal activity.

The captured signal was imported into Audacity for further analysis. Image 8 shows an annotated screen capture of the signals. Image 8 was modified for clarity by removing unnecessary dead space between signals.



Project Rate (Hz): 100000
Snap To: Off
Selection Start: 00 h 00 m 11.935 s
● End ○ Length: 00 h 00 m 11.935 s
Audio Position: 00 h 00 m 00.000 s
Stopped.
Actual Rate: 100000

Image 8: Audacity Display of Capture

Within Audacity the 'on' command was isolated and magnified. The 'H' shaped signals shown in Image 8 were enlarged. Each 'H' shaped sequence is composed of four individual signals.   The morphology of the individual signals was determined to be identical when separated and overlaid within Audacity; each signal appeared to only vary in amplitude. It is hypothesized this is a transmission power change to ensure delivery of the signal. In as such, only one of the four signals within the 'on' signal was decoded.

Image 9 shows an annotated enlargement of the FSK signal and example transition points where the signal changes between high and low frequency. The change between high and low frequency indicates the change in bits from 1 to 0 or 0 to 1.

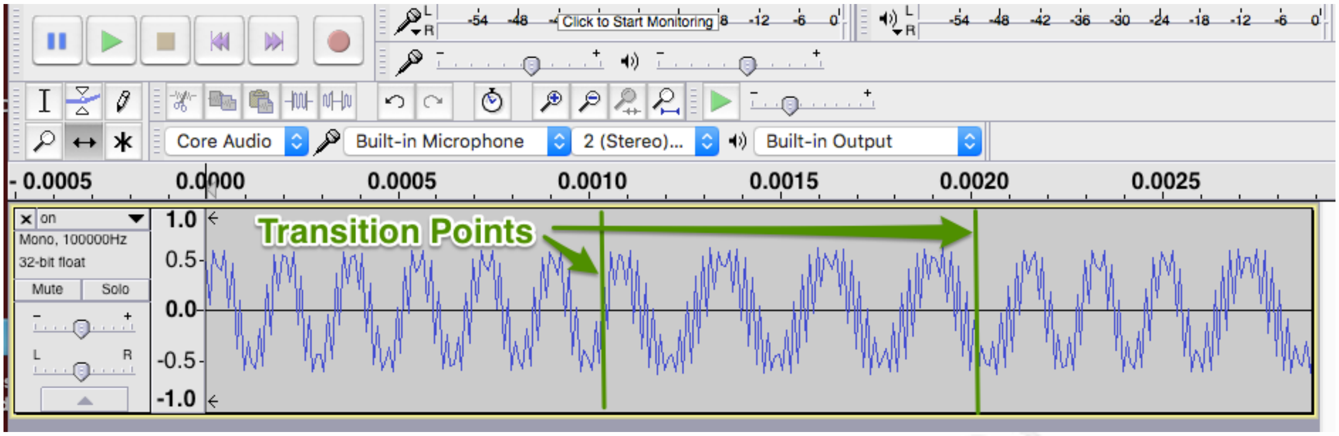


Image 9: FSK Transition Point

Using the transition points as markers, the duration between transitions was measured as 0.5ms. Measuring every 0.5ms, the number of high and low-frequency waves is counted to determine the construct of a symbol. This determines the symbol rate, which is necessary to convert the signal to 0's and 1's. Using the known Z-Wave preamble of 010101, which is 10 bytes long, the signal was decoded into its 32 bytes. (Joseph Hall, 2016). Although Z-Wave frames do not have a fixed length, as the application layer packet can vary in length, known offsets include:

- Preamble: 10 Bytes
- Start of Frame: 1 Byte
- Home ID: 4 Bytes
- Source ID: 1 Byte
- Frame Control: 2 Bytes
- Length: 1 Byte
- Destination ID: 1 Byte

The entire signal was printed and assembled as a single long document. This permitted manual evaluation of the symbols into bytes. Once complete, the known packet offsets were compared to the decoded signal, the known offsets did not align as expected. A much longer preamble length was observed which caused other bytes to misalign. Despite the misaligned offsets, the encoding was converted to hex and a replay packet composed.

4.3.3. Transmission

For transmission of the replay packet, RFcat was configured with the following settings:

- d.setFreq(908420000)
- d.setMdmModulation(MOD_2FSK)
- d.setMdmDRate(9600)
- d.RFxmit("<Hex Packet>")

The first packet transmission did not generate a change in the smart switch state; however, on visualization of the signal in RF Analyzer, the signal's morphology was similar to the original signal capture. The same packet was retransmitted using data-rates of 40kBaud, 100kBaud, and 1/.0005 Baud; none of the variations changed the state of the Z-Wave appliance switch.

To ensure an error was not made assigning 0 or 1 to the high and low-frequency sequences, the original signal bits were reversed. As before, this was converted to hex and re-transmitted. It too did not result in a state change of the smart switch and in this case, the signal morphology was very different from the original captured signal.

5. Conclusion

Despite visualization of signal activity in the FFT plots for both GQRX and GNURadio, no useable data was captured using these methods. It is unclear why GQRX and GNURadio failed to provide usable captures. This requires further analysis.

As outlined in section 4.3.2, the alternative capture process did result in a useful signal capture. The analysis determined the signal used an FSK modulation with a Manchester encoding. The signal decoding, however, did not appear as expected when bytes were compared to known offsets. The hexadecimal conversion and resultant packet transmitted via the YardStick One and RFcat did not cause a state change in the smart switch. Multiple attempts to cause a change in the switch were unsuccessful. Assuming an encryption scheme was not overlooked, the author feels success is possible with additional trial and error attempts to identify the proper decoding.

Mark Devito: nobel.gas@gmail.com

6. References

Aeotec Labs. (2016, Jan). Aeon Labs Smart Energy Switch. Santa Clara, CA, US.

Aeotec Labs. (n.d.). *Z-Wave 500 Series.* Retrieved 2016, from Aeotec: http://aeotec.com/z-wave-500-series-module-chip

Asay, M. (2016, Jan 29). *Why 10 million developers are lining up for the Internet of Things.* Retrieved Jun 2016, from TechRepublic: http://www.techrepublic.com/article/why-10-million-developers-are-lining-up-for-the-internet-of-things/

Atlas0fdoom. (n.d.). *rfcat*. Retrieved 2016, from BitBucket: https://bitbucket.org/atlas0fd00m/rfcat

Behrang Fouladi, S. G. (2013, Oct 1). *Security Evaluation of the Z-Wave Wireless Protocol.* Retrieved Apr 2016, from Sensepost UK: https://www.sensepost.com/cms/resources/conferences/2013/bh_zwave/Security%20Evaluation%20of%20Z-Wave_WP.pdf

Behrang Fouladi, S. (2013, Oct 1). *Honey, I 'm Home!! Hacking Z -Wave Home Automation Systems.* Retrieved Jun 19, 2016, from Slide Share: http://www.slideshare.net/sensepost/hacking-zwave-home-automation-systems

Defensive Security. (n.d.). Retrieved 2016, from Kali: https://www.kali.org/

DomotiGa.nl. (2011). *Z-Wave Technical Basics.* https://github.com/DomotiGa/DomotiGa.

Federal Communications Commission (FCC). (n.d.). *FCC ID Search*. Retrieved May 27, 2016, from FCC: https://www.fcc.gov/general/fcc-id-search-page

Federal Trade Commission. (2015, Jan 27). *FTC Report on Internet of Things Urges Companies to Adopt Best Practices to Address Consumer Privacy and Security Risks*. Retrieved Jun 2016, from FTC.gov: https://www.ftc.gov/news-events/press-releases/2015/01/ftc-report-internet-things-urges-companies-adopt-best-practices

Gianmarco Baldini, e. (2012). Security Aspects in Software Defined Radio and Cognitive Radio Networks: A Survey and A Way Ahead. *IEEE Communications Surveys & Tutorials , 14* (2), 355-379.

GNURadio Foundation. (n.d.). *About GNURadio*. Retrieved from GNURadio: http://gnuradio.org/about/

GQRX Developers. (n.d.). Retrieved 2016, from GQRX SDR: http://gqrx.dk/

Great Scott Gadgets. (n.d.). *YARD Stick One*. Retrieved 2016, from Great Scott Gadgets: https://greatscottgadgets.com/yardstickone/

Hajdarbegovic, N. (2015). Retrieved 2016, from https://www.toptal.com/it/are-we-creating-an-insecure-internet-of-things

Hall, J. &. (2016, 01). *Breaking Bulbs Briskly by Bogus Broadcasts*. (M. S, Producer) Retrieved 2016, from YouTube: https://www.youtube.com/watch?v=EDzxMfx1v5Q&ab_channel=MichailS

Indigo Domo. (n.d.). Retrieved from Indigo Domo: https://www.indigodomo.com/

Ionescu, P. (2015, Apr 08). *The 10 Most Common Application Attacks in Action*. Retrieved Jun 19, 2016, from Security Intelligence: https://securityintelligence.com/the-10-most-common-application-attacks-in-action/

Jorgensen, T. (2005, Jun 01). *Z-Wave as Home Control RF Platform*. (Zensys) Retrieved Jun 05, 2016, from Home Toys: http://www.hometoys.com/content.php?url=/htinews/jun05/articles/zensys/homecontrol.htm

Joseph Hall, B. R. (2016, 02 22). Z-Wave Network Reconnaissance and Transceiver Fingerprinting Using Software-Defined Radios. (D. T. Greiman, Ed.) *International Conference on Cyber Warfare and Security* , 164.

Lomas, N. (2015, Jan 08). *The FTC Warns Internet of Things Business To Bake In Privacy and Security*. Retrieved Apr 2016, from techcrunch.com: http://techcrunch.com/2015/01/08/ftc-iot-privacy-warning/

Mark Devito: nobel.gas@gmail.com

Moor Insights & Strategy. (2013, Sep 26). Retrieved 2016, from Forbes,com: http://www.forbes.com/sites/patrickmoorhead/2013/09/26/the-problem-with-home-automations-iot/#ee9d9da6efe0

Porup, J. (2016, Jan 23). *"Internet of Things" security is hilariously broken and getting worse*. Retrieved 2016, from Arstechnica: http://arstechnica.com/security/2016/01/how-to-search-the-internet-of-the-things-for-photos-of-sleeping-babies/

RF Wireless World. (2012). *z-wave Tutorial-frequency, frame, protocol, PHY, MAC, z-wave security basic tutorial*. Retrieved May 02, 2016, from RF Wireless World: http://www.rfwireless-world.com/Tutorials/z-wave-tutorial.html

Security. (2015, Dec 28). *How the Internet of Things Got Hacked*. Retrieved 2016, from www.wired.com: http://www.wired.com/2015/12/2015-the-year-the-internet-of-things-got-hacked

SigBlips DSP Engineering. (n.d.). Retrieved 2016, from Baudline: http://www.baudline.com/

Sigma Designs. (2016). *Products Web Page: 500 series brochure.* Retrieved 2016, from Sigma Designs: http://z-wave.sigmadesigns.com/docs/brochures/ZM5101_br.pdf

Symantec. (2015, Mar 12). (M. B. Wueest, Ed.) Retrieved from Symantec Security Response: https://www.symantec.com/content/dam/symantec/docs/white-papers/insecurity-in-the-internet-of-things-en.pdf

Szczys, M. (2016, 01 16). Retrieved 04 2016, from Hackaday.com: http://hacakday.com/2016/01/16/shmoocon-2016-z-wave-protocol-hacked-with-sdr/

Texas Instruments. (2005). *ISM-Band and Short Range Device Regulatory Compliance Overview.* Dallas: Texas Instruments.

Texas Instruments. (2007). *RF BASICS - Low Power Wireless Texas Instruments.* Dallas: Texas Instruments.

Mark Devito: nobel.gas@gmail.com

Tripwire. (2015, Jun 03). *Radio Killed the Security of Things: RF Jammers & Crime.* Retrieved 2016, from http://www.tripwire.com/state-of-security/security-awareness/radio-killed-the-security-of-things-rf-jammers-crime/

Watson-Johnson Company. (1980, 09). *Multidisciplinary Engineering Program - Edge Project.* Retrieved 2016, from EDGE - Rochester Institute of Technology: http://edge.rit.edu/edge/P09141/public/FSK.pdf

Wind River. (2015). *Security in the Internet of Things - Lessons from the Past for the Connected Future.* Alameda: Wind River Systems.

Zensys. (2007, 10 01). *ZM3102N Z-Wave Module Datasheet.* (N. T. Johansen, Ed.) Retrieved 04 2016, from Digikey: https://media.digikey.com/pdf/data%20sheets/zensys%20pdfs/zm3102n.pdf

Acknowledgments: The author would like to thank Michael Ossmann, Founder of Great Scott Gadgets for the donation of the Yardstick One and Matt Bendiksen founder of Indigo Domotics for the contribution of multiple 30-day trial keys for Indigo 6.

Mark Devito: nobel.gas@gmail.com