

Security Assessment Report (SAR)



Akshay Aravind Nayak

iSmartAlarm

12/6/2017



TABLE OF CONTENTS

- 1.0 Introduction
 - 1.1 Applicable Laws and Regulations
 - 1.2 Scope
 - 1.3 Assumptions/Limitations
- 2.0 System Overview
 - 2.1 System Description
 - 2.2 Overview of the System Evolution
 - 2.3 Relevant System Documentation
 - 2.3.1 Use Case
 - 2.3.2 Interaction Diagram
 - 2.3.2.1 Intruder triggers a sensor
 - 2.3.2.2 Arm system command
 - 2.4 State Diagram
 - 2.5 Block Diagram
- 3.0 Assessment Methodology
 - 3.1 Authorization
 - 3.2 Team Composition
 - 3.3 Assessment tools and resources
- 4.0 Security Assessment Procedure
 - 4.1 Process Overview
 - 4.2 Modelling Techniques
 - 4.3 Component Identification
 - 4.3.1 Hardware Specification
 - 4.3.2 Deductions through testing
 - 4.3.3 Software Specification
 - 4.4 Experiments/Practical Overview
 - 4.4.1 Experiment 1
 - 4.4.2 Experiment 2
 - 4.4.3 Experiment 3
 - 4.5 Recommended Penetration Test
- 5.0 Security Assessment
 - 5.1 Threats and Countermeasures
 - 5.2 Attack Tree
 - 5.3 Vulnerability Assessment Summary

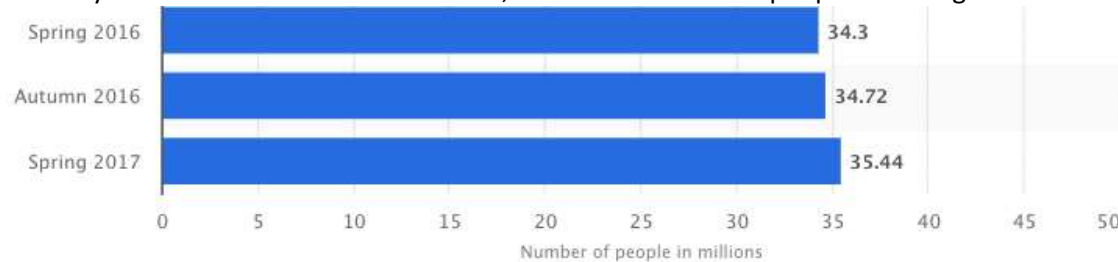
6.0 Remediation Summary
 6.1 Requirements
7.0 Conclusion

ACKNOWLEDGEMENT

I would like to offer my special thanks to Professor Sharon Oneal for the valuable and constructive suggestion. The topics covered in the class helped us in our project and helped us look the cyber world in a different perspective. I would also like to thank my group members, Michael and Mohammad for helping in planning the project and finding the vulnerabilities by reverse engineering in the iSmartAlarm.

1. INTRODUCTION

This document consists of a *Security Assessment Report (SAR)* for the iSmartAlarm as required by FedRAMP. This SAR contains the results of the comprehensive security test and evaluation of the system. It is estimated that in 2017, around 35.44 million people are using home security systems



According to a study done by the University of North Carolina 60% of the thieves who got interviewed admitted that having a security system would decrease the chance of them breaking into your home. Another study also shows homes without security systems are 300% more vulnerable to burglary.

iSmartAlarm is one of the products that ensures a solid home security system. This project will be focusing on ensuring the security of the iSmartAlarm system, identifying security flaws, verifying the effectiveness of security measure and to make sure all security measures will continue to be effective after implementation.

1.1. APPLICABLE LAWS AND REGULATIONS

For this project we are limited in our testing and analysis by law. We are acting as a private entity who is testing for the systems vulnerabilities without approval of the manufacturer. Therefore, we will not be suggesting any testing around the Cloud storage or iOS app due to our Terms of Use agreement with the company upon downloading the app. Our focus will be around the iSmartAlarm devices we purpose since each set is independent of another, so testing our device will have no impact on other iSmartAlarm users.

1.2. SCOPE

The assessment of this system is coming from a blue team point of view. The understanding of the system and its operations will come from information provided by the iSmartAlarm Producers as well as our own actions in reverse engineering. The reverse engineering process will follow the standard list by FireEye. For the sake of simplicity, we will only do our assessment on the four default commands of the iSmartAlarm. There is a level of customization available with the commands, but more complexity would not add any much more information about the vulnerabilities.

After the component interactions and data flow across the system is understood the assessment will take place. The boundary of the assessment will extend beyond the purely the iSmartAlarm due to its large dependency on networking for its functions as an IoT device. The assessment will include

- STRIDE Model

To understand what types of attacks are coming and possible ways we see attackers approaching our model to take advantage of components and their interactions

- DREAD Model

To understand the current severity of the threat

- Attack Tree

To break down the actions an attacker would take in taking advantage of vulnerabilities in our system

Beyond the listed assessment we will also add how to test their own equipment to try and identify issues they might have in their supply chain.

1.3. ASSUMPTIONS / LIMITATIONS

The security assessment limitations are mainly around testing the physical boundary extremes the system can handle. Included in this category are

- The operating temperature the system can handle
- The distances that components can communicate
- The forces the system can handle

We also must make assumptions are certain signals the system is using to communicate and the current procedures the company is uses for testing possible attacks in the supply chain.

2. SYSTEM OVERVIEW

2.1. SYSTEM DESCRIPTION

This system is an IoT device that lets a user take control of their home security independent of a service provider. For this assessment plan the focus will remain on the preferred package because of its popularity. The preferred package consists of 8 components upon purchase:

- 1) 2x Remote Tag (Bottom Left) + Batteries
- 2) 2x Contact Sensor (Bottom Right) +Batteries
- 3) 1x Motion Sensor (Upper Right) + Batteries
- 4) 1x CubeOne(Middle)
- 5) EthernetCable
- 6) Power Source for CubeOne



The iPhone is included in the picture because it is to be considered a main communication method with the CubeOne. Currently, the iSmartAlarm is only compatible with the app on iOS devices. Once the iSmartAlarm system is synced with the app on a device, the device can then add new proprietary iSmartAlarm components to the system and request state changes of the system.

For this analysis of the system I want to cover the basic functions. There are four commands that the remote tag and iPhone can command of the system

- Home Command: Function is for when a user is in the house, but they want to be notified if a door or window is opened that has a contact sensor on it. After selecting the home command, the user has 90 seconds before the motion sensors are waiting to detect an “intruder”.
- Arm Command: Function is for when a user leaves the home, and they want to be notified if any activity in their home is picked up by the sensors. After selecting the arm command, the user has 90 seconds before all sensors are waiting to detect an “intruder”.

- **Disarm Command:** After the “Home Command” or “Arm Command” has been set the Disarm will allow the user to safely trigger the sensors without the system detecting them as an intruder

- **Panic Command:** Will immediately set off an alarm at an audio frequency of 110dB from the CubeOne

For more Information on these commands and the interactions and state changes of the system reference Section 2.3 where many diagrams are included with details.

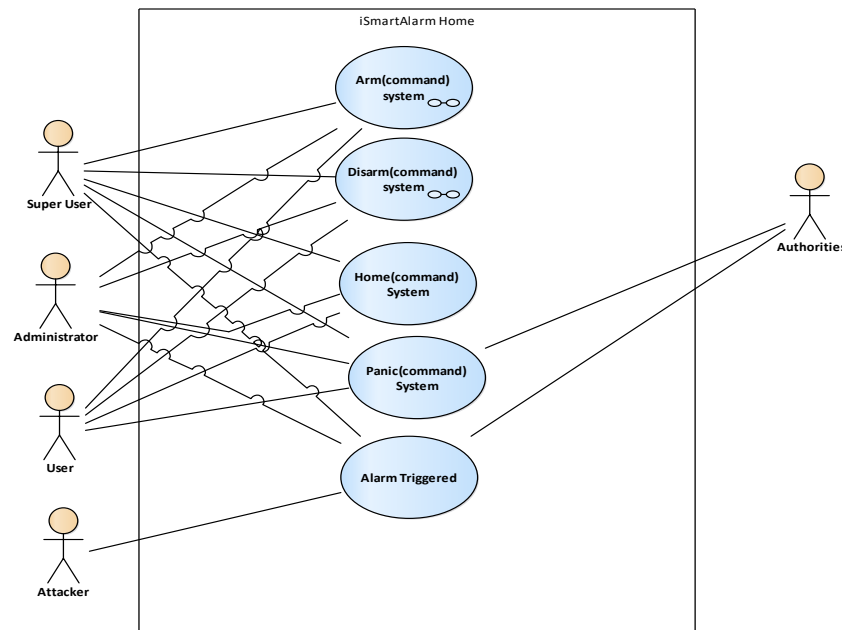
2.2. OVERVIEW OF SYSTEM EVALUATION

The purpose of developing this system is to compete with pricier options for home security by providers such as AT&T and COX that cost a monthly fee instead of one-time cost. At a price of \$149 for the preferred package and \$99 for an apartment set this device could help users take home security into their own hands economically.

2.3. RELEVANT SYSTEM DOCUMENTATION

2.3.1. USE CASE

The Use Cases are added to understand what actors interact with the different commands. One thing to note here is there are three types of Users on an account: super user, administrator, user. The Super User runs the account and can see all state changes of the system through the cloud, can change the state of the system and can add new administrators and Users. The administrator can see the state changes of the system and can the information on the cloud, but they cannot add more users. Lastly the User can only change the state of the system but has no further privileges like retrieving state change information through the cloud.



2.3.2. INTERACTION DIAGRAM

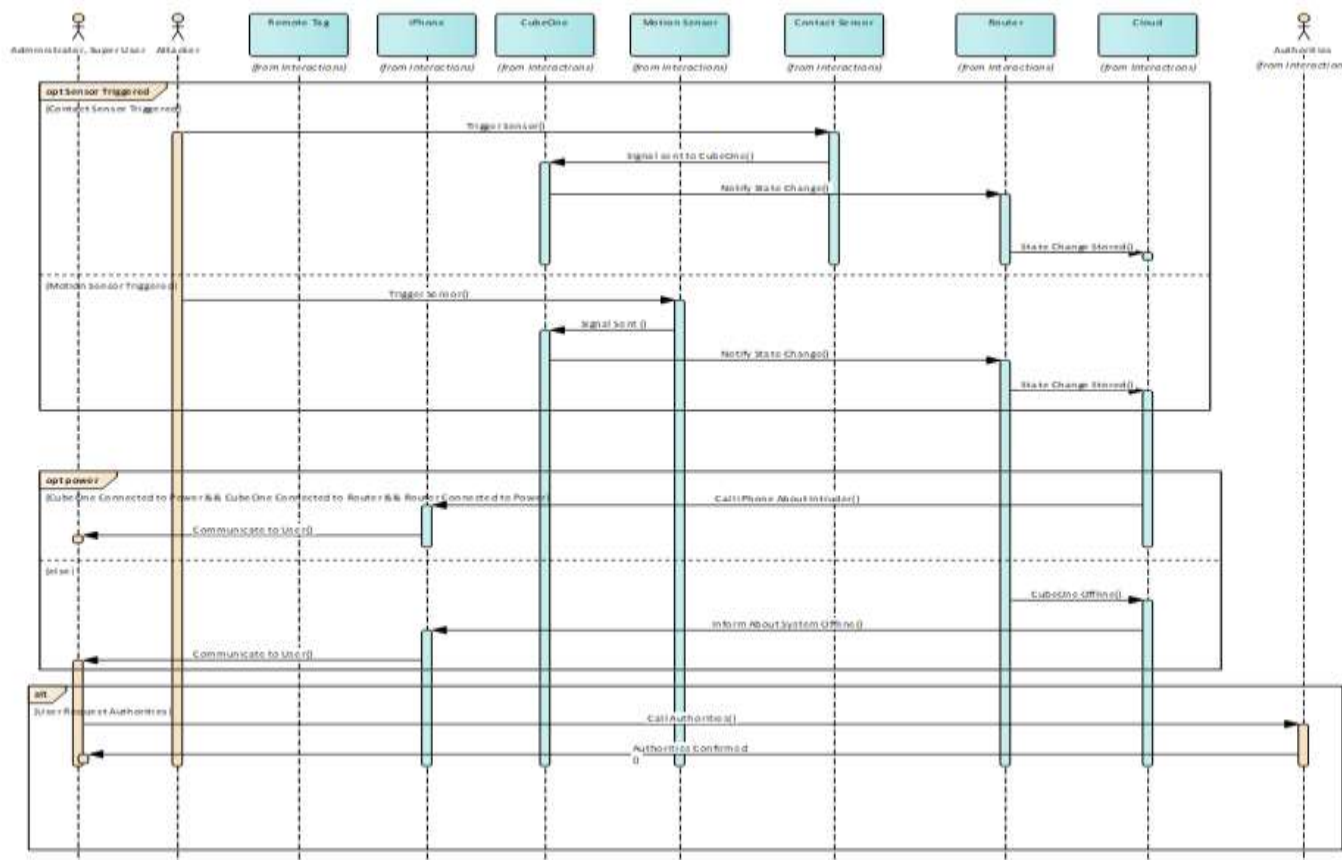
The purpose of the interaction diagram is to start understanding the different components that need to interact in the system for different Use Cases. The conditions for each of the interaction is included as well. This report covers have interaction diagram tying to the Use Cases above.

2.3.2.1. INTRUDER TRIGGERS AN SENSOR INTERACTION DIAGRAM

There is one key point I want to cover in this diagram and it is the condition for the power fragment. Inside one condition is that the CubeOne and the Router must be in communication and both of those devices must have power supplied. The CubeOne attached to the router must be done for the CubeOne to be initialized and run. If the condition is met, then any state change will send information to the cloud and make its way to the user's iPhone, so they can be notified. However, the system can operate without the attachment of the router and the CubeOne after initialization. The CubeOne cannot receive or return information with an iPhone though. If this happens the router will send a message to the cloud saying the CubeOne has become offline.

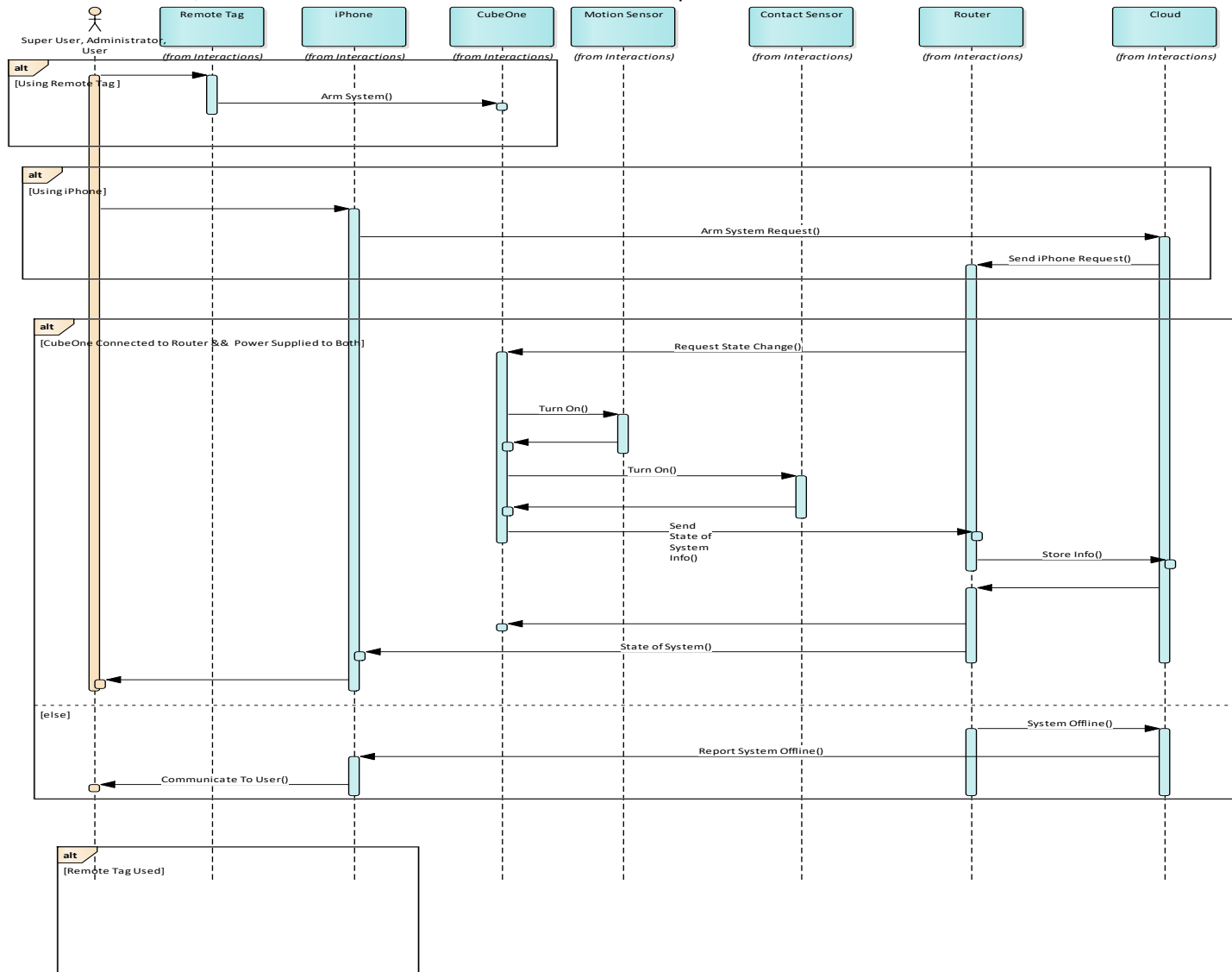
Lastly one other important thing to note is if any of the sensors are triggered, there is a 30 seconds window where someone can disarm the system before the system goes into panic mode and sends information to the administrators and super user of the system.

Assumption: The initial state of the system is Arm

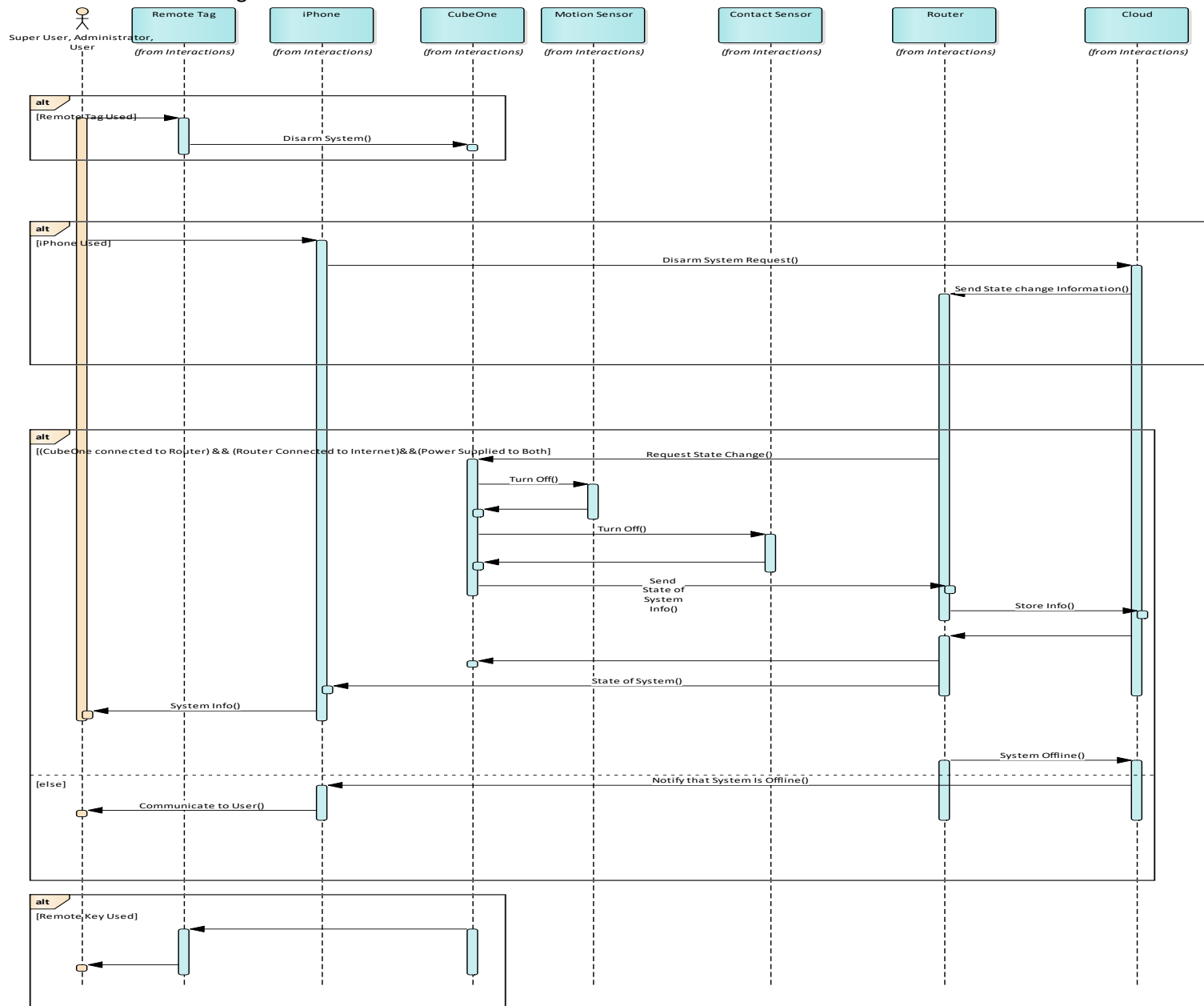


2.3.2.2. ARM SYSTEM COMMAND INTERACTION DIAGRAM

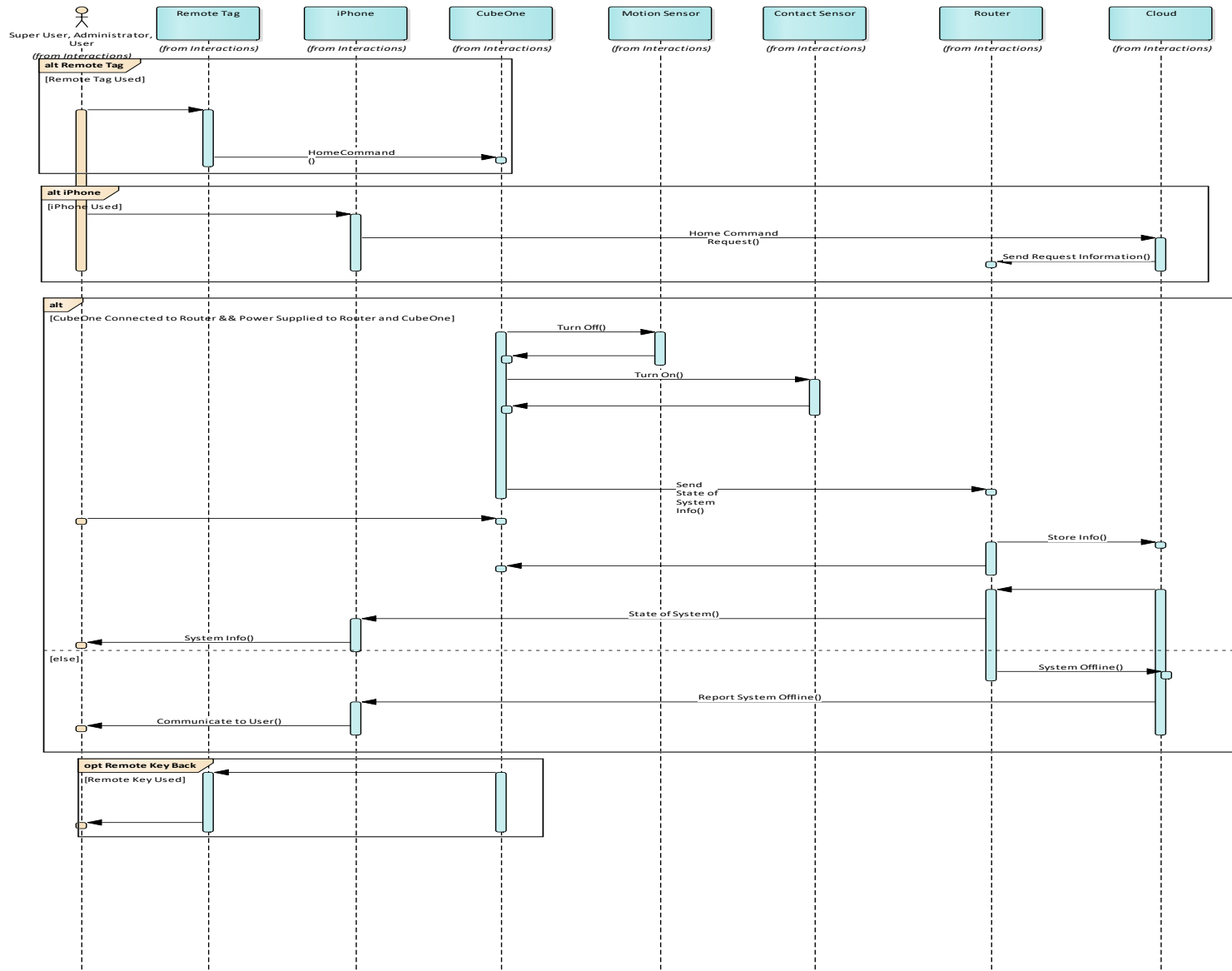
The information described in the previous rate diagram appears here as well. A different but key point that exists in this diagram is the remote tag. This remote tag has 4 buttons each with default commands. This remote tag communicates with the CubeOne directly and does not rely on the Router. The power still needs to be given to the CubeOne and router, but the connection between the two is not important.



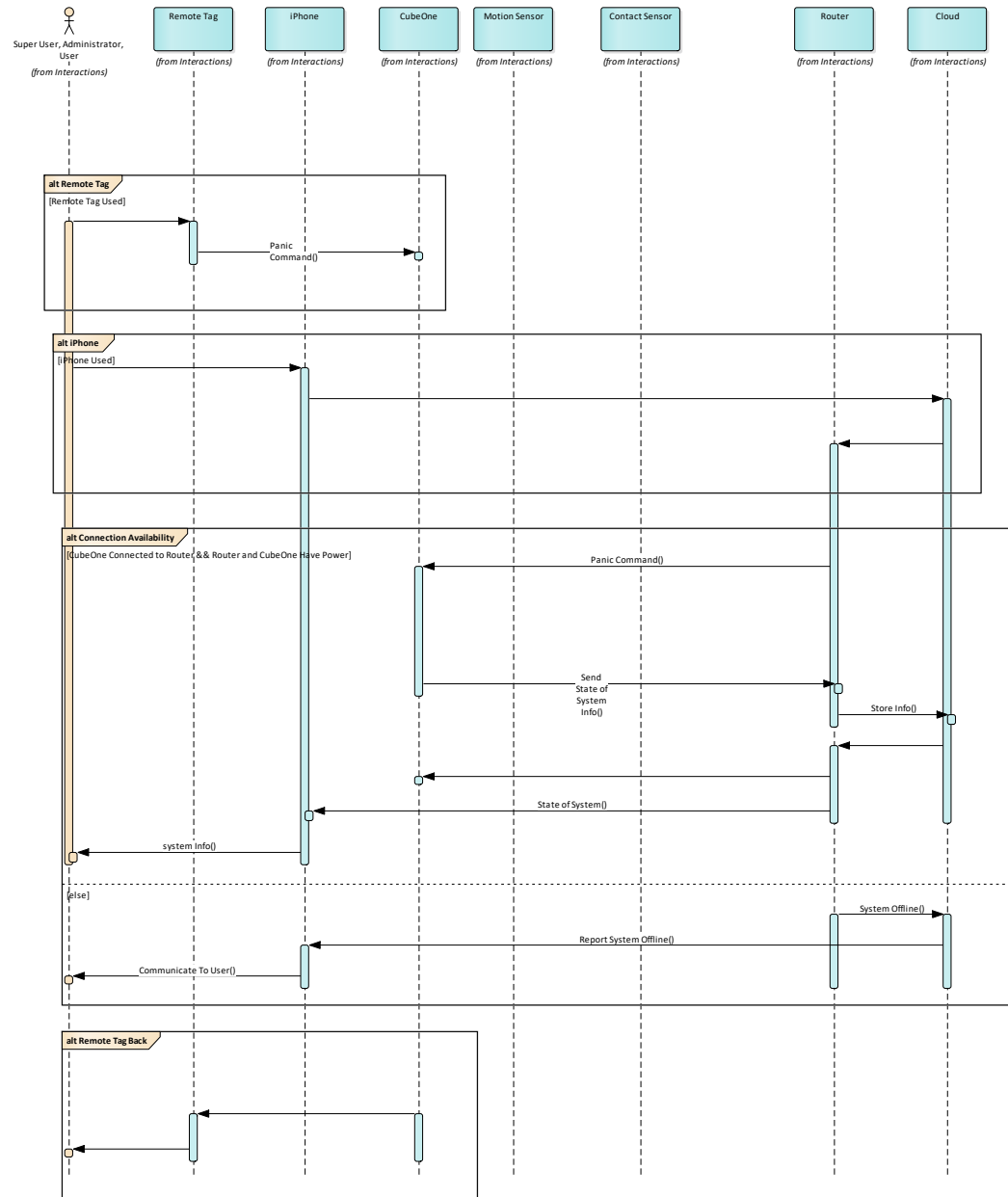
Disarm System Command Interaction Diagram



Home Command Interaction Diagram



Panic Command Interactions Diagram



2.4. STATE DIAGRAMS

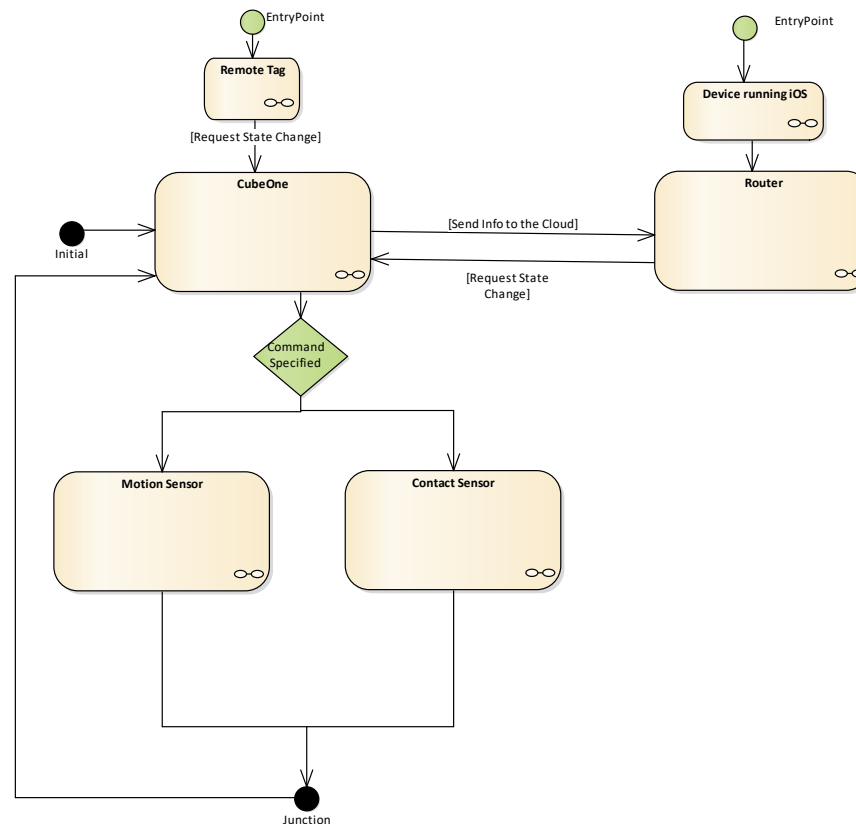
We took the information from the interactions diagram and brought it into the state diagrams

State Diagrams will show how each of the different components as well as an overview of information exchange between each component and how the states would change. The technical information regarding the exchange information such as the WiFi signal frequency will be discussed with the architecture diagram. The first one is the overall iSmartAlarm system.

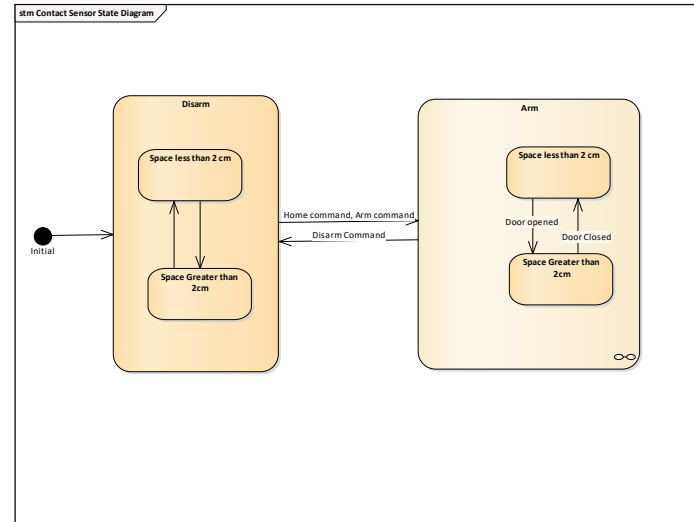
The router can receive information directly from the remote tag or receive information from device running the ismartalarm ios app by taking information from the router. The Cubeone then send's signals to each of the motion sensor and contact sensor based on the command with a change in state.

The next state diagram is that of the CubeOne. The Cubeone initilizes when you turn it on in the diarm state. From their it can change states depending on commands or sensor signals. The system was noted to have to go from the arm command to disarm before it can go to home.

Motion Sensor diagram will send a signal either way based on whether it detects motion but only if it is armed will the signal trigger the cubeone. Same goes for the contact sensor. Signal will be sent but will only send a signal affecting the cubeone if it is in the arm mode. The contact sensor can be triggered if the separation is greater than 2 cm.

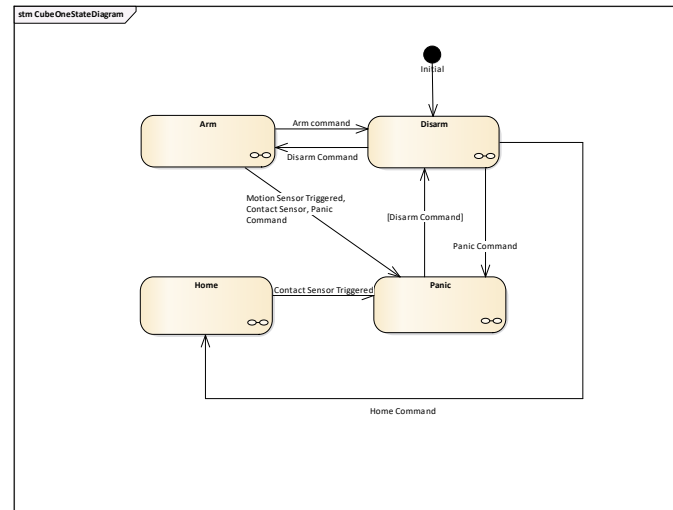


Contact Sensor State diagram

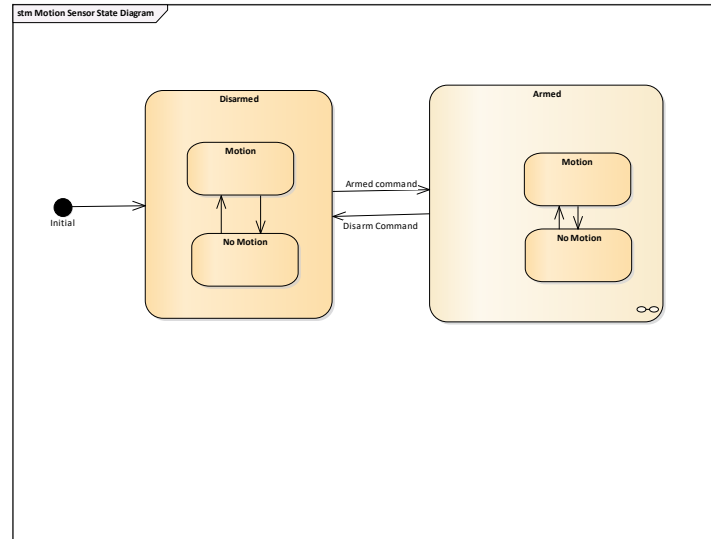


CubeOne State Diagram

In this diagram there is no link between the arm and home command. Each of those commands allows for a sensor or sensors the possibility of being triggered and communicating to CubeOne that there is an intruder in the system. Therefore, it does not make sense to go directly from arm to home and vice versa. The system must be disarmed before entering either of these states from the previous one. Without this setup, someone could continually switch between the two states and this could prevent detecting the intruder.



Motion Sensor Diagram

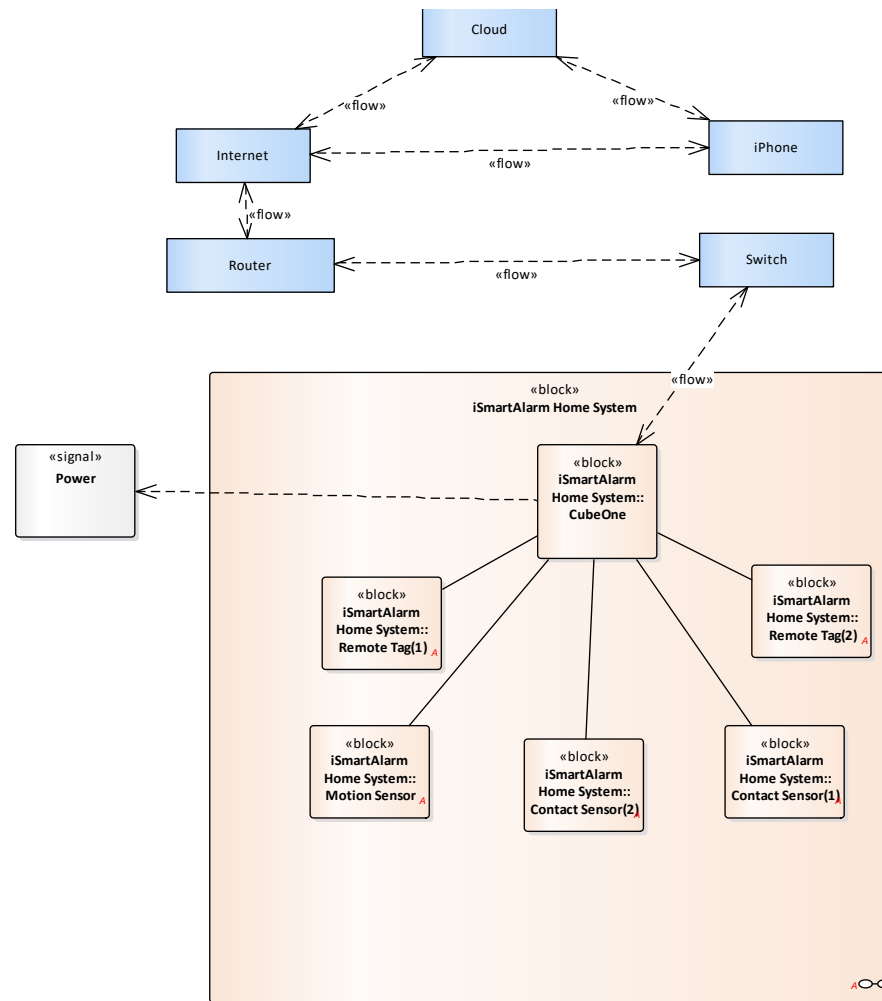


The working of contact sensor and motion sensor is similar. Initially the sensors are in disarm state. When a motion is detected it changes to the motuion state and control is passed on to the armed state. At this point the sensor notifies the cube one about the armed state. The Cube one sends this information to the cloud using the routers. The cloud processes the information and sends the instantaneous messages to the user via message.

2.5. BLOCK DIAGRAM

This block diagram is simplified, but the intention is to show the boundary of the iSmartAlarm and understand where external devices/components meet the iSmartAlarm. As previously mentioned, the CubeOne is the brain of the system and this diagram shows it more explicitly. Every device must connect to the CubeOne whether it's external devices/components or devices within the boundary of the iSmartAlarm.

The difference in this model from previous diagram is we are no presenting the switch. The switch is part of the router most of the time and is the port that transmits the data packets to the CubeOne. By breaking it down even more, we start to understand what communication vulnerabilities we might have in the system.



3. ASSESSMENT METHODOLOGY

The assessment methodology used to conduct the security assessment for the iSmartAlarm system is summarized in the following steps:

3.1. AUTHORIZATION

As covered in the laws and regulations section to start the assessment. This testing and analysis is done independent of the iSmartAlarm manufacturers. We have no authorization by the company to test this system, so our testing and analysis we largely focus around our own device and attached router. There will be a mention of how the app could be better secured but not penetration testing can be done in this area.

3.2. TEAM COMPOSITION

Akshay A Nayak	System Engineering Networking Vulnerabilities Attack Scenarios System Decomposition Experiments
Michael Hailwood	Systems Engineering Reverse Engineering Hardware Vulnerabilities
Mohammad	Project Management Attack Scenarios

3.3. ASSESSMENT TOOLS AND RESOURCES

The Reverse Engineering process followed goes from FireEye.com:

1. Research the device
2. Identify the components
3. Identify the debugging ports
4. Dump the flash
5. Extract/Analyze

The research has already been done and the diagrams were included in the first section. The identify of components and steps after will be covered in section 4.

The tools needed for other testing scenarios are:

- Computer
- WireShark
- Bus Pirate and SOIC8/SOP8 test clip
- Flashrom
- Binwalk

4. SECURITY ASSESSMENT PROCEDURES

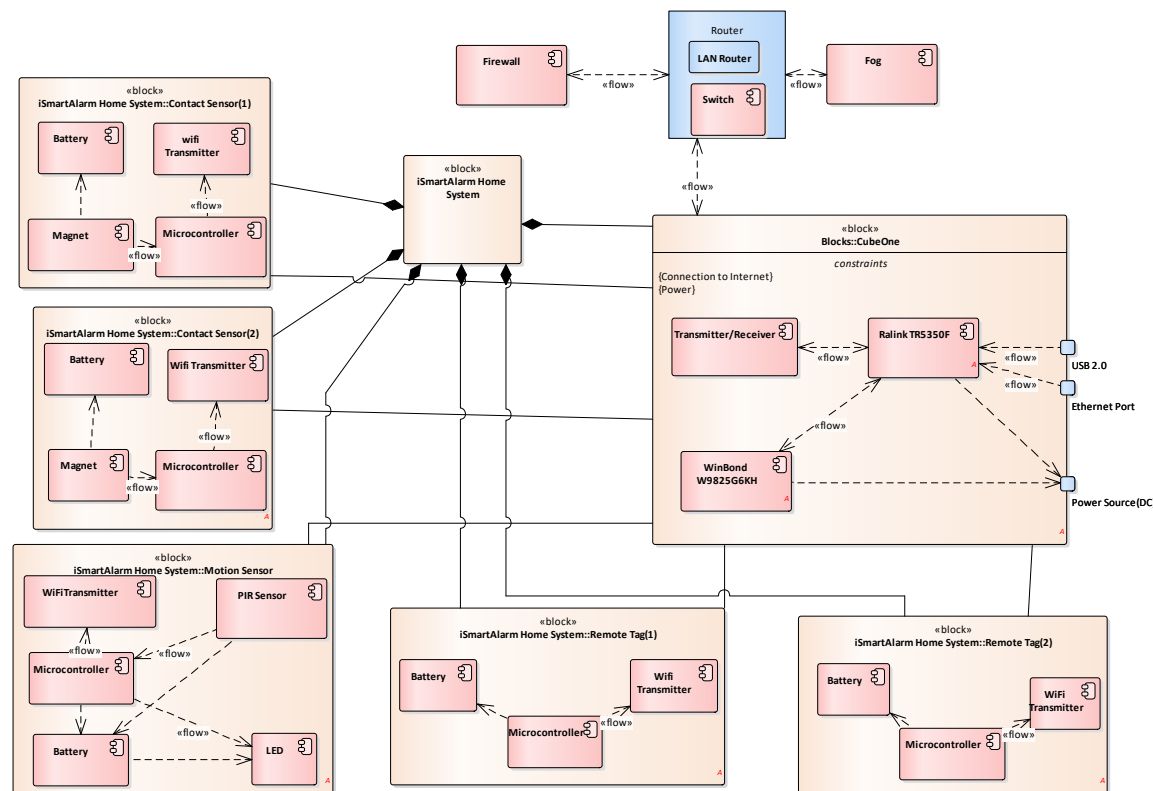
4.1. PROCESS OVERVIEW

The Security assessment process will move forward from the system description into an approach that will yield more detailed results pertaining to how information travels throughout the iSmartAlarm. We will use the resources covered in the assessment methodology section to track the signals by different devices of the system, and continue with the last steps of the reverse engineering process.

4.2. MODELLING TECHNIQUES

The modeling of the security assessment will start with a component diagram to label more precisely the ports in each device that are used for communication with ports of other devices. Then we will show our current ideas of possible vulnerabilities through a STRIDE and DREAD model, so we can propose how each vulnerability should be tested

4.3. COMPONENT IDENTIFICATION COMPONENT DIAGRAM



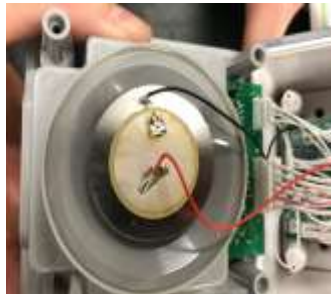
4.3.1. HARDWARE SPECIFICATION

CubeOne-Central Device that controls the rest of the IoT devices

Specifications

1. Model: iPU3
2. Classification: External power
3. Device dimensions: 100mm*100mm*105mm
4. Weight: 370g
5. Power: 5V 1A adapter 100-240V~50/60Hz 0.2A
6. Frequency: 908MHZ
7. Distance: <100m (Outdoor, open space)
8. Connection and Expansion:
USB port 2.0;
10/100 BASE-T Ethernet (RJ-45 connector)
9. Operating temperatures: 14°F~122°F
10. Operating humidity: 85%±5
11. Storage and transport temperature: -40°F~131°F
12. Storage and transport humidity: 90%
13. Alarm Sound Level: =100dB

110 dB siren



Ralink TR5350F**CPU**

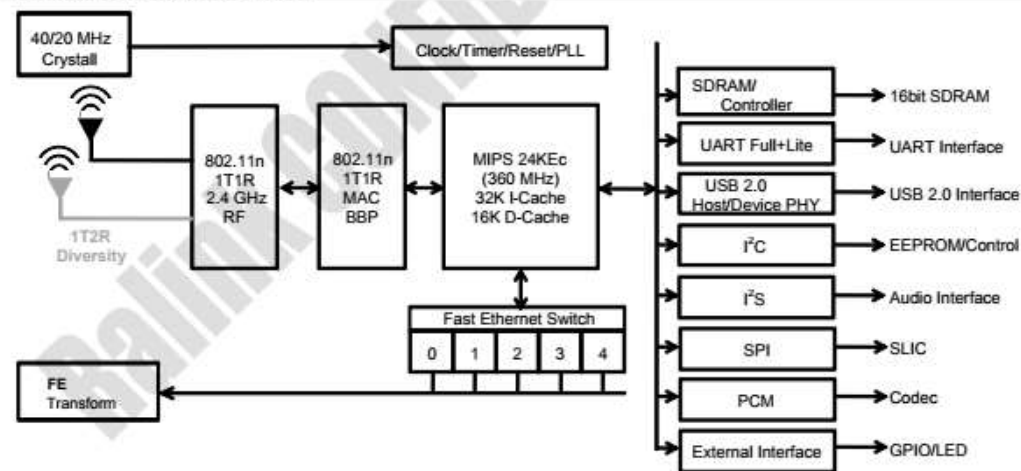
- Processing Speed=360MHz
- 5-port 10/100 Mbps Ethernet switch w/ 5 10/100 PHYs
- USB 2.0 host/client

Wifi System on a chip

- 802.11n 1T/1R (1x1:1) 2.4 GHz 150Mbps MAC/BB/PA/RF

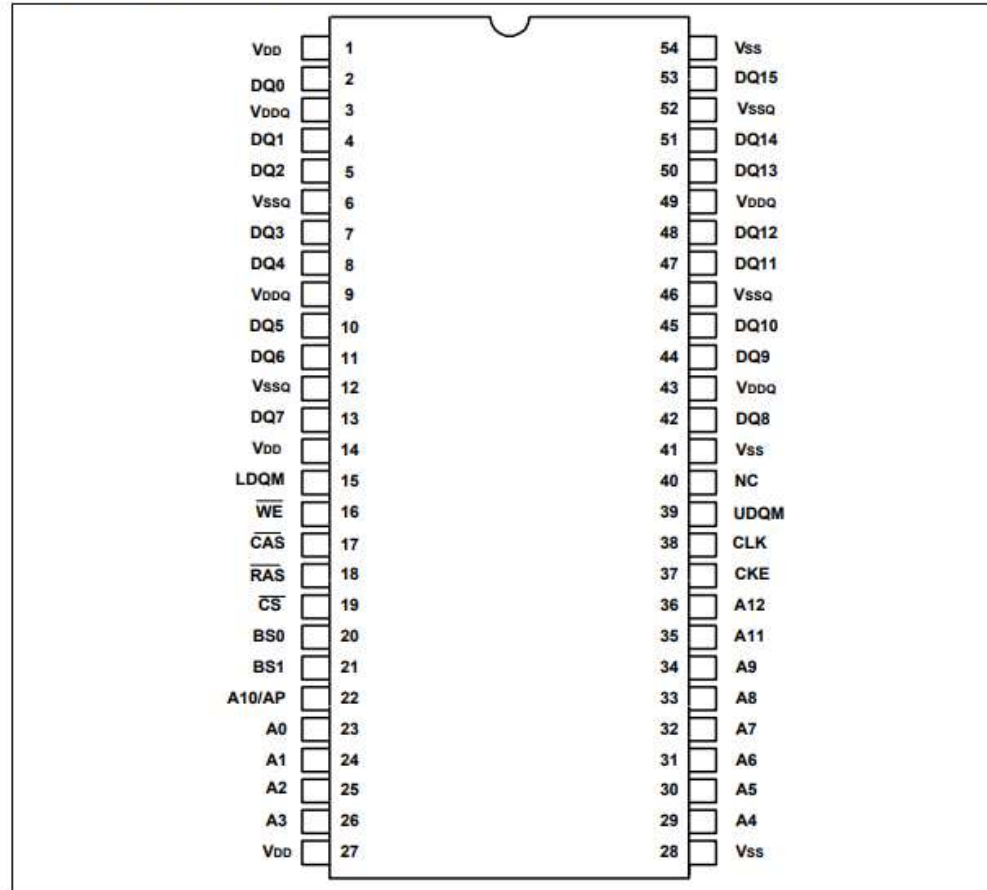
All documentation posted online

- This chip is supported by OpenWRT distribution
- RT5350 SDK on GitHub, includes Linux + U-Boot GPL source code

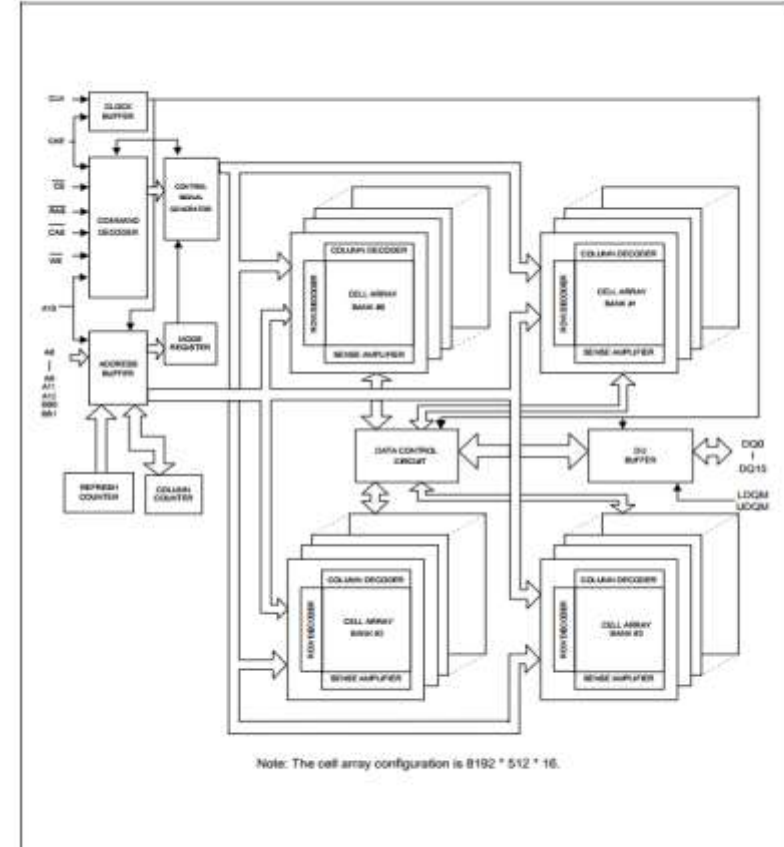
Functional Block Diagram**WinBond W9825G6KH****SDRAM**

- high-speed synchronous dynamic random-access memory
- Total Memory=32MB
- 16-bit communication with Ralink5350

4. PIN CONFIGURATION



6. BLOCK DIAGRAM



Contact Sensor -Motion Sensor that can be mounted anywhere specified

Specifications

1. Model: DWS3
2. Classification: Internally-powered
3. Device dimensions: 50mm*50mm*13mm
4. Weight: 27.4g
5. Power: 1*3V (CR2032)
6. Frequency: 908MHZ

7. Distance: <100m (Outdoor, open space)
8. Operating temperature: -14°F~122°F
9. Operating humidity: 85%±5
10. Storage and transport temperature: -40°F~131°F
11. Storage and transport humidity: 90%
12. Magnetic gap (Open to Close): <20mm
13. Magnetic gap (Close to Open): <20mm

Motion Sensor

Specifications

1. Model: PIR3
2. Classification: Internally-powered
3. Device dimensions: 100mm*69mm*50mm
4. Weight: 80.1g
5. Power: 3*1.5V(AA)
6. Frequency: 908MHZ
7. Distance: <100m (Outdoor, open space)
8. Operating temperature: 14°F~122°F
9. Operating humidity: 30 %±2 =85%±5
10. Storage and transport temperature: -40°F~131°F
11. Storage and transport humidity: 90%
12. PIR detection angle: 90° 10M mounted @ Approximately 6 1/2' from the floor

Remote Tag

Specifications

1. Model: RC3
2. Classification: Internally-powered
3. Device dimensions: 60mm*30mm*10mm
4. Weight: 12.8g
5. Power: 1*3V (CR2032)
6. Frequency: 908MHZ
7. Distance: <100m (Outdoor)
8. Operating temperature: 14°F~122°F
9. Operating humidity: 85%±5
10. Storage and transport temperature: -40°F~131°F
11. Storage and transport humidity: 90%

Fog

Like cloud

Present at the end devices (near sensors)

Faster operation

Store less information but vital information

4.3.2. DEDUCTIONS THROUGH TESTING

Connection Between iSmartAlarm and Router:

iSmartAlarm can function without the Router (Disconnection)

Given:

- When it was initially turned on the Router must be connected to it or it will not work
Therefore, it both go off it will not work again until
- The iSmartAlarm has power
- iSmartAlarm devices already initialized with CubeOne
- Router still supplied with power however connection with CubeOne through ethernet cord is not present

Effect:

- User notified through iOS app that they are disconnected
- No communication available via iOS App
- All Communication with commands to CubeOne must be done via Remote Tag

Assumptions after Effect:

- There are wifi signals sent between the devices and CubeOne
- iSmart Alarm can function without Router(Unplugged)

Given:

- When it was initially turned on the Router has to be connected to it or it will not work
Therefore, it both go off it will not work again until
- The iSmartAlarm has power
- iSmartAlarm devices already initialized with CubeOne
- Router has no power

Effect:

- User NOT notified through iOS app they are disconnected
- No communication available via iOS App
- All Communication with commands to CubeOne must be done via Remote Tag

Assumptions after Effects:

There is communication between router and iSmartAlarm and if that communication is broken the Router is responsible for sending that information in the cloud

iSmartAlarm Cloud StoringCondition1) (router Maintains Power)Given:

- Everything connected with power
- No connection between iSmartAlarm and Router via ethernet

Effect

- iOS will not be able directly receive data about the state of the iSmartAlarm. When the CubeOne and Router are connected again the the information will be sent to the cloud about the changes in system

Assumptions

- NONE

Condition2) (CubeOne Disconnection with Router and power Removal)Order of power removal

- 1)disconnect CubeOne and Router ethernet communication
- 2) change state of CubeOne with Remote Tag
- 3) remove power from the CubeOne
- 4) give power to the CubeOne
- 5)reconnect CubeOne to Router

Effect:

- The user is notified of the disconnection of the CubeOne and Router

- After the CubeOne and Router are re-connected in step 5, the state changes will not be recorded in the cloud

Assumption after Effect:

- There is no hard Drive present (only RAM) because if power is disconnected the memory of iSmartAlarm States there is no repudiation upon reconnecting the devices. This means there is only RAM that keeps the information before sending it to the cloud with the Router since the memory is erased when power is removed.

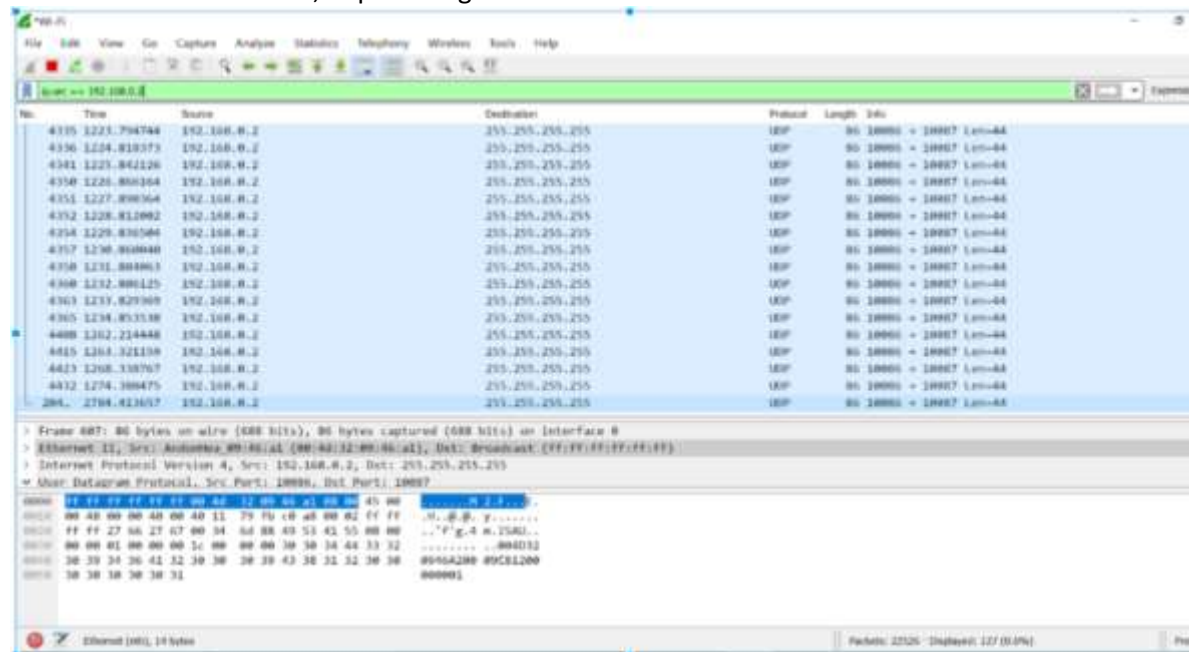
WireShark the Router Switch

Given:

- No Mirroring of the router
- Used a computer to Wireshark the Switch of the router
Only looks at that specific Wifi Switch
- Already connected to the network

Effect

- The switch receives a signal from the Devices attached to the iSmartAlarm
Specifically, the motions sensor
- Any change in the distance between the sensor, implies a signal is sent



The UDP protocol is not a reliable protocol, as there is no ACK of the data received. There is no required connection between two devices that use UDP. A signal

is simply accepted then picked up. The major advantage of this protocol is that it is faster, and more data can be sent in every packet.

- Any one could track in the area whether a signal has been sent regardless of connection with the router.
- The Wireshark capture shows 86 bytes of data being sent by the sensor to the CubeOne.
- No matter the state of the entire system if there is motion found by the sensor a signal is out using the UDP protocol

4.3.3. SOFTWARE/OPERATING SYSTEM

The operating system of the CubeOne was identified as Linux and the source code is from U-Boot GPL. For any further questions about the source code please refer the GitHub since it is open source. On top of this we plan to check if there are any non standard header files or other proprietary information by the iSmartAlarm manufacturer by dumping the flash memory. This process will be highlighted in the Recommended Penetration Test Approach.

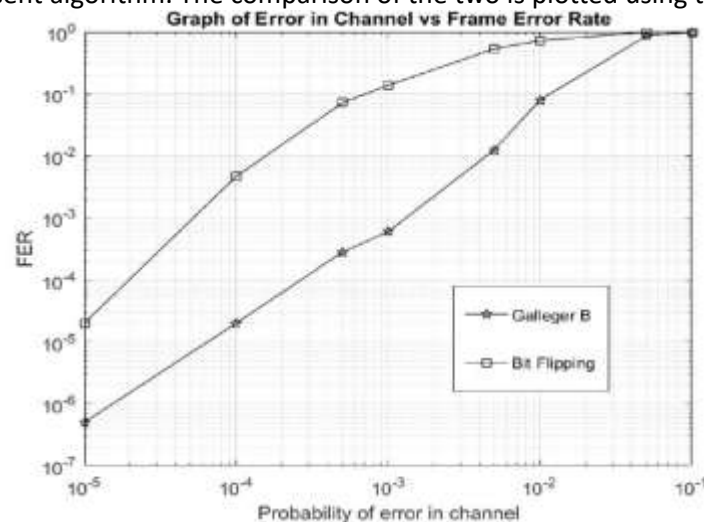
4.4. EXPERIMENTS / PRACTICAL OVERVIEW

4.4.1. EXPERIMENT 1

The data that is sent from the transmitter to the receiver. During the transmission there is a possibility that noise gets added and the data is corrupted. To solve this problem, we use the process of encoding and decoding. The encoder and decoder are of various types. Here we are providing the comparative analysis between the various decoders. The decoders we are comparing are bit flipping, Gallager B and Viterbi decoders.

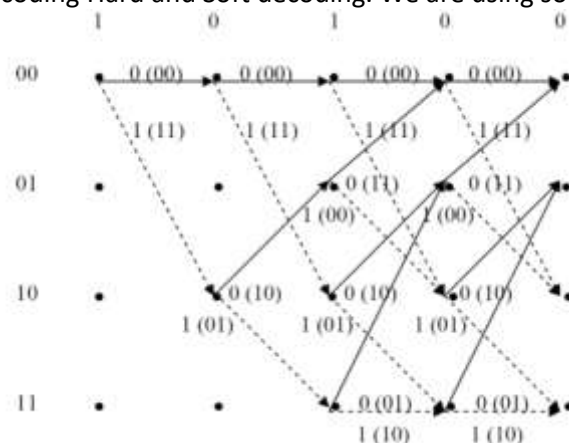
The Bit Flipping Decoder is a decoding algorithm which uses the Low-density parity check matrix. The decoder makes the decision based on the majority votes of the bit position. The efficiency of this decoder is low as it is stateless in nature and the current stream of bits does not consider the information given by corrected bits.

The performance increases when we use a Gallager B decoder. The Gallager B unlike the bit flipping is state full in nature. This means that the previous decoding information is considered while decoding the present algorithm. The comparison of the two is plotted using the MATLAB. The comparison is as shown

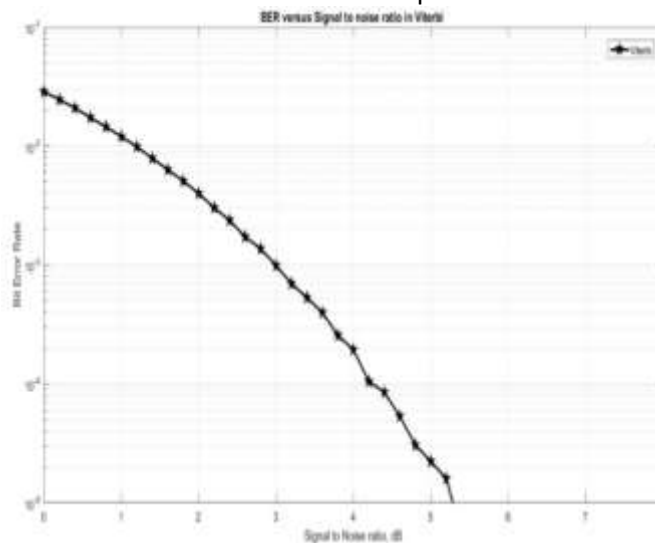


From the above graph the Frame error rate increases as the error in the channel increase. The Gallager B decoder performs better than the Bit flipping decoder for the different values of the decoder as shown. If the probability of error in channel is less than 0.1 then Gallager B decoder performs 100 times better and it keeps decreasing till it reaches to 10 times better. However, both Bitflipping and Gallager B fails to correct error if high amount of noise is introduced.

The most efficient algorithm that we use is the Viterbi algorithm. Here I used maximum likelihood decoding often referred as Viterbi decoding. This decoder considers the signal strength. The decoder checks the sequence that is received, and it computes a metric for each path that it takes, decides using the metric and goes forward in the trellis. This process is continued till 2 paths converge to the same node. Then, as the metric will be the same, we can choose any one path out of them and continue with the trellis. This path stays in the trellis and the remaining unnecessary paths are removed from the Trellis. The paths that are remaining are called the survivor paths. There are two types of decoding Hard and Soft decoding. We are using soft decoding in this experiment.



The trellis diagram for the Viterbi decoder is as shown above. The decoder follows the path of least distance. Here in the above figure it is 00 - 10 - 11 - 00 - 00

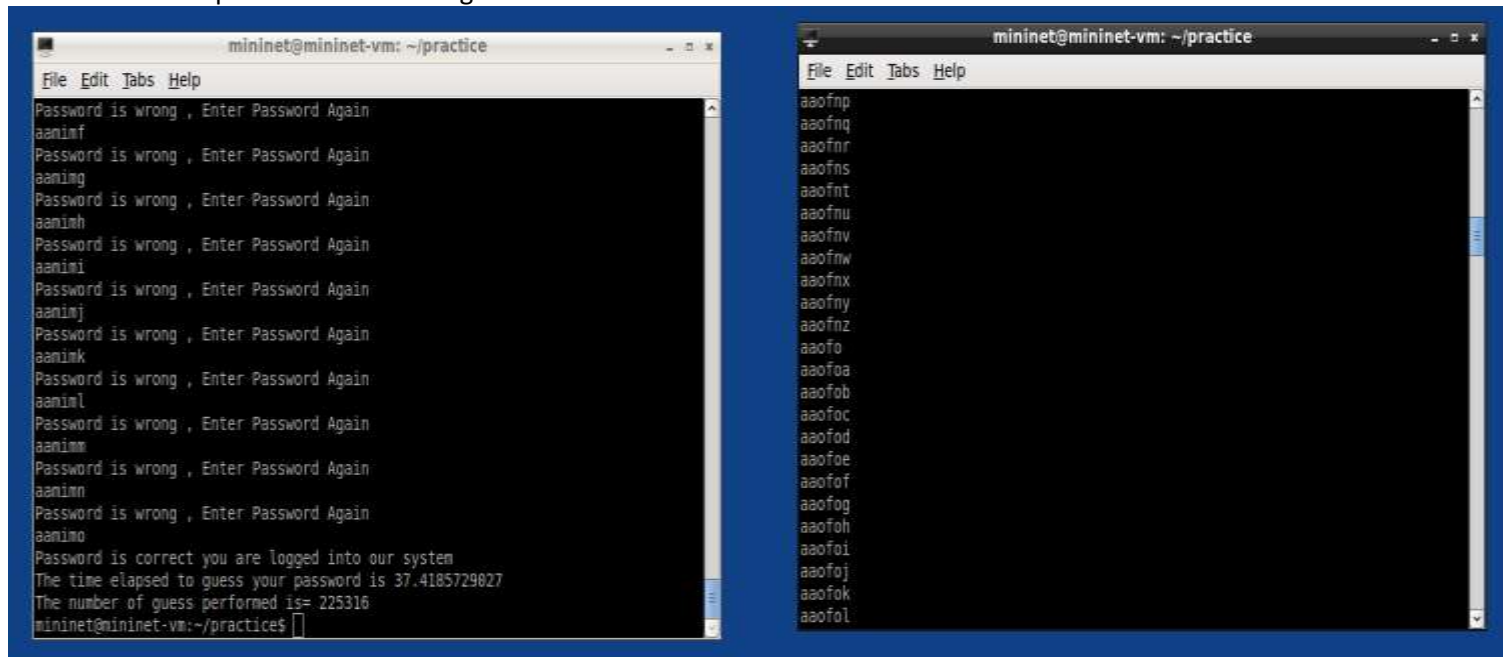


From the graph the Bit error rate decreases as the Signal to noise ratio increases. So, if the SNR in DB is greater than 5.5 then all the errors in the bits are corrected. Thus, this proves to be the best decoding algorithm.

4.4.2. EXPERIMENT 2

Guess the password by Brute Force.

Here I wrote a python code and created a local server and two clients. One client I considered as a normal authentic client and the other as a attacker. I used a password as 'akmimo' and this was stored in the server. The client can access the server using this password. Now coming to the attacker I wrote a code to possibly send request to the server with all passwords and then gain access to the server.



We can see that in matter of 38 seconds the attacker sent 225316 possible passwords before breaking the actual password. So it's always important to use long passwords and combination of letters, numbers and special character.

4.4.3. EXPERIMENT 3

Here I created an encryption scheme that is a combination of shift and substitution cipher. The plain text is first converted using shift cipher and then the cipher obtained is again encrypted to obtain the cipher text. This cipher text is sent to the server and the server decrypts it using the already available key. Here we can see that the plain text is **"ilovethisclass"** is encrypted to **"!%q! hx\$lwfcww"** which is again encrypted as **"p?hb5copxt?1xx"** this is sent to the server. The server decrypts it using the already available key. Then based on the text sent it decides the command the client should perform and sends the control signals.

```

mininet@mininet-vm: ~/MIS543_socket
File Edit Tabs Help
2017-09-24-000709_958x940_scrot.png mininet MIS543_socket pox
Desktop MIS543_Lab3 of-dissector practice
Downloads MIS543_Lab4 oflops Templates
github.com MIS543_Proj2 oftest
mininet@mininet-vm:~$ cd MIS543_socket
mininet@mininet-vm:~/MIS543_socket$ ls
checking.py      file.txt      simple_client http1.0.py  url_parse.py
echo_client.py   one          simple_client http1.1.py
echo_server_once.py one.py       testproj.py
echo_server.py   output.txt   test.py
mininet@mininet-vm:~/MIS543_socket$ python echo_server.py
AKSHAY , MICHAEL AND MOHAMMAD *****SERVER*****
*****SERVER*****
##### Server is ready to receive messages #####
Waiting for Cubeone to send data
RECEIVED SIGNAL FROM CUBE ONE
The data is received
and decipher using 1st decryption method
The result obtained from first deciphered output
l%q!hx$lfw%cww
Decrypting using second decryption method
The plain text sent from client
ilovethisclass
mininet@mininet-vm:~/MIS543_socket$

mininet@mininet-vm: ~/MIS543_socket
File Edit Tabs Help
2017-09-24-000709_958x940_scrot.png mininet MIS543_socket pox
Desktop MIS543_Lab3 of-dissector practice
Downloads MIS543_Lab4 oflops Templates
github.com MIS543_Proj2 oftest
mininet@mininet-vm:~$ cd MIS543_socket
mininet@mininet-vm:~/MIS543_socket$ ls
checking.py      file.txt      simple_client http1.0.py  url_parse.py
echo_client.py   one          simple_client http1.1.py
echo_server_once.py one.py       testproj.py
echo_server.py   output.txt   test.py
mininet@mininet-vm:~/MIS543_socket$ python echo_client.py
enter the mesage
'ilovethisclass'
*****THE OUTPUT AFTER ENCRYPTING FOR THE FIRST TIME IS*****
*****
l%q!hx$lfw%cww
key size is39
*****THE OUTPUT OBTAINED AFTER SECOND TIME ENCRYPTION IS*****
*****
p?hb5copxt?lxx
*****SENDING DATA TO SERVER*****
From Server: Server received the data= ilovethisclass
*****CONTINUE NORMAL OPERATION*****
mininet@mininet-vm:~/MIS543_socket$

```

This encryption event though not the best can prevent the attacker from getting hold of the plain text. The attacker gets the control signals in encrypted form and thus is of no use. So, I recommend that every signal need to be encrypted and hashed so that the integrity and confidentiality of data is protected.

LINKS OF ACTUAL EXPERIMENT

https://drive.google.com/a/email.arizona.edu/file/d/13aZC9IbRcAi7KKNhP_vz_E7YZ8nDErNc/view?usp=sharing
<https://drive.google.com/a/email.arizona.edu/file/d/1agH80J9IYVb2FdTm8JXJa56s8h-STkwN/view?usp=sharing>
<https://drive.google.com/a/email.arizona.edu/file/d/1lcr8m4DAle70VnUt756tSEZLI61yWhma/view?usp=sharing>

4.5. RECOMMENDED PENETRATION TEST

Reverse Engineering for Hardware Testing

We also recommend going further in the reverse engineering process by following these steps:

1. Identify the debugging ports
2. Dump the Flash Memory
3. Extract/Analyze the Flash Memory

Identifying the Debugging ports

In this device the debugging ports will be present on the UART line. To separate the UART_RX(Receiver) vs. the UART_TX(Transmitter) use a multimeter. The UART will exist on the perimeter of the device where there are several nodes next to each other. After identifying what looks like the UART line take the multimeter pins, connect one to the ground and one to one of the nodes that is either the Receiver or transmitter. The receiver will show an oscillating voltage while the transmitter will show a peak. The benefit of having access to the debugging port with a UART line is the ability to bridge the interface to USB, which would allow a computer to communicate with the device.

Dump the Flash

This step will require the resources above: bus pirate, test clip, software. The bus pirate needs to have flashrom on it and be connected to the test clip. Then connect the test clip to the flash memory chip with proper pin alignment, and run the program on the bus pirate. This will extract all the memory in whole.

Extract/Analyze

Now that the information has been extracted use the binwalk tool to pick up the header files and more information in the filesystem. Having all this information will help better define the authentication keys and other proprietary information the iSmartAlarm manufacturer adds to their devices. This information will be important for some of the penetration tests below

Hardware Penetration Test

Our proposed penetration approach includes running the previous process to identify how accessible the information is and then rewriting the bootloader using the information available for U-Boot on Github before uploading it onto the device. Uploading the information to the device needs to be done by running a specialized software script with string compares and sending that data over the UART-USB bridge onto the UART_RX node and into the flash memory chip.

Signal Tracking

The other test that needs to be performed before most of the vulnerabilities are addressed is signal tracking. Wireshark has the capability of mirroring routers. This needs to be done to identify signal information from the iSmartAlarm devices. The protocol used should be identified as well as the kind of information sent by the devices.

5. SECURITY ASSESSMENT

5.1. THREATS/COUNTERMEASURES ASSESSMENT SUMMARY

STRIDE and DREAD

STRIDE is a type of threat classification model developed by Microsoft that focus on the various threats. The threats are classified into 6 categories, they are

Type of Threat	Risk	DREAD Rating
Spoofing	D: Damage Potential R: Reproducibility E: Exploitability A: Affected Users D: Discoverability	H: High (>18)
Tampering		M: Medium (13 – 18)
Repudiation		L: Low (<13)
Information Disclosure		
Denial of Service		
Elevation of Privilege		

While DREAD is a type of classification scheme which is used for quantifying, comparing and prioritizing the amount of risk due to each threat.

General Countermeasures

Threat	Countermeasures
Spoofing user identity	Use strong authentication. Do not store secrets (for example, passwords) in plaintext. Do not pass credentials in plaintext over the wire. Protect authentication cookies with Secure Sockets Layer (SSL).
Tampering with data	Use data hashing and signing. Use digital signatures. Use strong authorization. Use tamper-resistant protocols across communication links. Secure communication links with protocols that provide message integrity.
Repudiation	Create secure audit trails. Use digital signatures.
Information disclosure	Use strong authorization. Use strong encryption. Secure communication links with protocols that provide message confidentiality. Do not store secrets (for example, passwords) in plaintext.
Denial of service	Use resource and bandwidth throttling techniques. Validate and filter input.
Elevation of privilege	Follow the principle of least privilege and use least privileged service accounts to run processes and access resources.

SPOOFING

Type	Threat Type	Rank	Threat Identifier	Threat Description	Threat Target	How can we carry out this attack	Countermeasures	Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	Total
1.1	Spoofing	7	Take down routers on the path from source to destination(tamper with the bandwidth of the path)	Attackers can modify the Bandwidth of the path the data is flowing making it to flow through region of malicious network so that the attacker can get hold of data. The attacker can take down the routers on the path and make the data to flow in the intended malicious router	Router	Take down routers that are in left path from the source to destination	Use static routing mechanism to carry out the routing process	4	1	2	3	4	14
1.2	Spoofing	8	Authentication Bypass	remote execution of commands	CubeOne	Run malicious scripts on the authentic user machine and get a bypass code to get hold of the CubeOne	Use signing and hashing	4	3	2	4	2	15
1.3	Spoofing	5	MITM	SSL - Capture messages which are travelling from the web browser to the device	CubeOne	The attacker makes independent connections with the victim and relays messages between them to make them believe they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker	1-Encryption of messages that travel between the user and the cube. 2-Using both symmetric and asymmetric encryption methods for security purposes. 3-Using stronger encryption mechanisms like AES 256. 4- Use of Virtual Private Network	4	2	2	4	4	15
1.4	Spoofing	5	IR Sensor Prevention(Hide Heat)	Wearing styrofoam in front of body to prevent heat from body being exposed	Motion Sensor	Prevents User from knowing someone is in their house	Using of motion sensor along with IR sensor	4	4	4	2	2	15
1.5	Spoofing	9	Spoof IP address of the sensor	Spoof the IP address of the sensor and send false information stating it to be the actual sensor	Sensors		Consider the MAC address even for the process of communication and use private key mechanism prior to data exchange	3	2	3	2	3	12

We identified 6 attacks that fall in the threat type of spoofing. They are

- Take down the routers and tamper with the bandwidth and make sensitive information flow from the router the attacker have control over. The countermeasure for this threat is use of static routers so that the attacker cannot force the traffic to flow from one router to another.
- Authentication Bypass, the attacker can perform authentication bypass on the cube one, one way to protect from this attack is by signing and hashing.
- Man-in-the-Middle attack, capture messages that are sent through SSL. The countermeasure against this type of attack is using encryption method to encrypt the data between the sensor and the cube one and use of a virtual private network.
- IR sensor Prevention (Hide the heat), wearing of Styrofoam to prevent heat from escaping from the body. Even though there is no counter measure for this, use of IR sensor along with the motion sensor (sonar based) can overcome their drawbacks.
- Spoof IP address of the sensor and send information from compromised sensors. To overcome this drawback the Cube one should be made to consider the MAC address along with the IP address and use the private key exchange mechanism before the actual communication.

TAMPERING

2.1	Tampering	5	Attack on a particular Port	Getting inside the network to disrupt the iSmartAlarm	Router, Cube One	Use the open ports to get access to the network	Enable port security feature on the router. Only allow MAC address of the authentic user to access and shut down the devices of other MAC address that is trying to access the port of our device. Use hashing and signed function to prevent attackers from accessing the part of network they are not allowed to access.	3	3	3	4	3	16
2.2	Tampering	4	Magnet Tampering	Using a magnet to act like the contact sensor is still connected	Contact sensor	Physical	None (Synchronization of carious sensors and passing the information through a smart device)	4	4	5	2	2	17
2.3	Tampering	6	Injecting false messages	Inject a malicious malware(example miral virus) turn them into bots or get them into the control of the attacker. Processor(Ralink 5350) has linux documentation online and direct access to it is allowed via USB 2.0 Interface on CubeOne.	CubeOne	prevents user from Using System	So firewall can be enabled with IDS AND IPS enabled to drop requests that are a threat to the system.The techniques of ingress and egress filtering can also be carried out to filter out certain malicious requests. ACL can be enabled to prevent certain user from accessing the cloud service.	3	2	3	4	3	15
2.4	Tampering	8	Sensor Hacking	Hack the sensor	Sensors	Hack the sensor by getting into the network. Then perform malicious activities like supply the sensor with high value of current.	Use end to end encryption. Prevent unauthorized MAC address from accessing the sensors.	3	2	3	2	3	13

We identified 4 attacks that fall in the threat type of Tampering. They are

- Attack on a port, when a port is left open then it becomes easy for the attacker to get inside the port and then access the network. To counter this threat, we can enable port security feature where only the authenticated MAC address scan accesses the port and if a unauthenticated MAC address try to access the port the switch turns off the port.
- Magnet tampering, use of a magnet to act like the door is still in contact. There is no countermeasure for this threat however using all the sensor used together can overcome this threat.
- Injection of false messages can turn our Cube one into a botnet. To counter this threat we can use a Firewall that is enabled with IDS and IPS . Prevent use of USB devices as they can be malicious.
- Sensor Hacking process of hacking the sensor and make the sensor perform what the attacker wants to perform To counter this threat, we can use end to end encryption and prevent a user with unauthorized MAC address from accessing the sensor.

REPUDIATION

3.1	Repudiation	6	Enable False Alarm	Use the tag to enable false alarm and false alert messages	Cube one	A person might send false alert (panic) messages to the cube one to enable false alarms	All messages need to be hashed and signed so that the attacker can't deny that he is the reason for the false alarm.	3	3	4	4	1	15
3.2	Repudiation	9	Flushing RAM	If you remove the connection between the router and the CubeOne and afterwards remove power from CubeOne then all system state changes are erased and cannot be stored in the Cloud	CubeOne and Router	Remove ethernet connection between router and CubeOne then remove CubeOne power	Add a harddrive	3	2	2	3	2	12

We identified 2 attacks that fall in the threat type of Repudiation. They are

- Enabling the alarm using the tag and send false alert messages

To counter from this threat we can hash and sign the data so that the attacker can't deny that he did not sent the data.

- Flushing the RAM, removal of power from the Cube One erases the data that is to be sent

The countermeasure to this attack is to include a harddrive.

INFORMATION DISCLOSURE

4.1	Information Disclosure	7	Access Control (New Key)	"On iSmartAlarm cube devices, there is incorrect Access Control because a "new key" is transmitted in cleartext.	CubeOne	Insider threat	Have privileges set for the people who are authenticated to access CubeOne	3	2	2	4	3	14
4.2	Information Disclosure	3	Password Guessing	Attacker can guess the password of router or the cube one using brute force approach	Router, Cube One	Brute Force approach	Use a strong password, ie a long password that is a combination of special characters and numbers. Use of IPS can also prevent this problem.	5	3	4	3	3	10

We identified 2 attacks that fall in the threat type Information Disclosure. They are

- Access Control, the data is sent in a clear text so that the attacker can easily get hold of the data

To counter this threat, use encrypted data, and do much of the processing in the sensors and send the data directly to the Cloud rather than send to the Cube one process it in the CubeOne and then send it to the cloud.

- Password guessing, common threat that can be performed using social engineering or dictionary attacks.

To prevent from this attack, use a salt table that is included at the end of the password and use a stronger password (Meaning a larger password and a combination of letter, number and special character)

DENIAL OF SERVICE ATTACK

5.1	DDoS, DDoS	1	SYN flood	Sending a SYN flood on port 12345 of CubeOne	CubeOneRouter	Attacker can carry out this attack by running a script to prevent the alarm of the cube one from functioning	Firewall Proxy, Filtering, Active monitor, Intrusion Detection and Intrusion Prevention System.	5	4	3	4	4	20
5.2	DDoS, DDoS	4	IR Sensor Prevention(Constant Light)	Holding constant wavelength of IR device to prevent the change being detected.	Motion sensor	Physical	Noise Synchronization of various sensors and passing the information through a smart device.	4	4	5	2	2	17
5.3	DOS, DDoS	6	Cloud DDoS Attack	The attacker can try to send false requests to the cloud. If the number of requests increases then the cloud capability then the system breaks down.	SmartAlarm Cloud	Syn flood, UDP flood and HTTP flood attack on the cloud.	Use of IDS and IPS system	3	1	2	5	4	15
5.4	DOS	5	IR Sensor Prevention(Rapid State Change)	Flicking lights on and off within 1 second to prevent detection	Motion Sensor	Prevents User from knowing someone is in their house	Use motion sensor along with the IR sensor	4	4	4	2	2	16
5.5	DDoS, DDoS	2	Ping of Death	Flood the router with many packets to deny internet service to the cube one	Router	Using a code in command prompt	Use of Access control, IDS and IPS	4	4	4	4	3	19
5.6	Reflected DDoS attack	8	UDP protocol vulnerability	UDP is connectionless in nature and does not validate the source IP address, so it is very easy to forge IP address	Router, Cube One	Running malicious scripts, Flooding the device a number of times, using bots to carry out the attack.	The application-layer protocol uses countermeasures such as session initiation	3	1	1	4	4	13
5.7	DOS/DDoS	6	Hello Flood attack	In a sensor network, the routing protocol broadcast hello message to announce its presence to its neighbors. A node which receives the hello message may assume that the source node is within its communication range and add this source node to its neighbor list.	Cube one and sensor	Additional sensors and cube one can be placed in the vicinity so that a number of hello messages are generated and this floods the sensors and the cube one	IP and MAC address should be considered before accepting any data. Only important information need to be passed and not the entire data.	4	2	2	4	3	15

We identified 7 attacks that fall in the threat type Denial of Service. They are

- SYN flood attack to the port 12345 of the CubeOne prevents the CubeOne from performing its normal task of receiving data from the sensor.

To counter this threat, we need to use Firewalls enabled with IDS and IPS.

- IR sensor (Constant Light), this attack is carried out by flashing a constant wavelength of light at the IR sensor and prevent it from detecting the motion.
- Cloud DDoS attack is carried out on the cloud, by sending a number of requests to the cloud so that it prevents the cloud from providing the service to the authentic user(CubeOne)

To prevent from this attack the best solution is to perform ingress and egress filtering and use of IDS and IPS.

- IR sensor Prevention (Rapid State Change), flicking the lights on and off stops the IR sensor from working.

There is no particular counter measure for this, but we can use the motion sensor that operates by SONAR.

- Ping of Death is an attack that is carried out on any networking devices by preventing it from providing services to the authentic users.

Use of access control, IDS and IPS can solve this problem.

- UDP protocol does not consider the IP address of the device it is receiving data from. So, it is very easy to forge the IP address and also hijack the sessions. The countermeasure for this attack is to perform session initiation and use of sequence number and Acknowledgement number (TCP) connection.
- Hello Flood Attack, the networking devices initially send a hello messages to state their presence to the neighboring devices, an attacker can get malicious sensors and send number of hello messages so that they fill up the buffer space of the authentic sensors.
IP address and MAC address need to be considered before accepting the data and only processed signals need to be sent from the sensors to the CubeOne.

ELEVATION OF PRIVILEGE

6.1	Elevation of Privileges	3	Run Malicious Script	Elevate yourself to administrator or super user	Cloud, Fog and CubeOne	Use malicious java script into the USB 2.0 and get the administrator privilege.	Disable the port that is not required.	4	3	3	4	4	18
6.2	Elevation OF Privileges	4	Phishing, whalejacking	Get the super user password and username	Cloud	Send email stating to be the CEO of the company to the CubeOne user and get hold of his password and username	check authenticity of the email. Another option is just don't respond to suspicious emails with private information.	4	3	3	5	2	17

We identified 2 attacks that fall in the threat type Denial of Service. They are

- Running Malicious Script to elevate the attacker as the super user.

The countermeasure against this attack is to disable the ports that are not required.

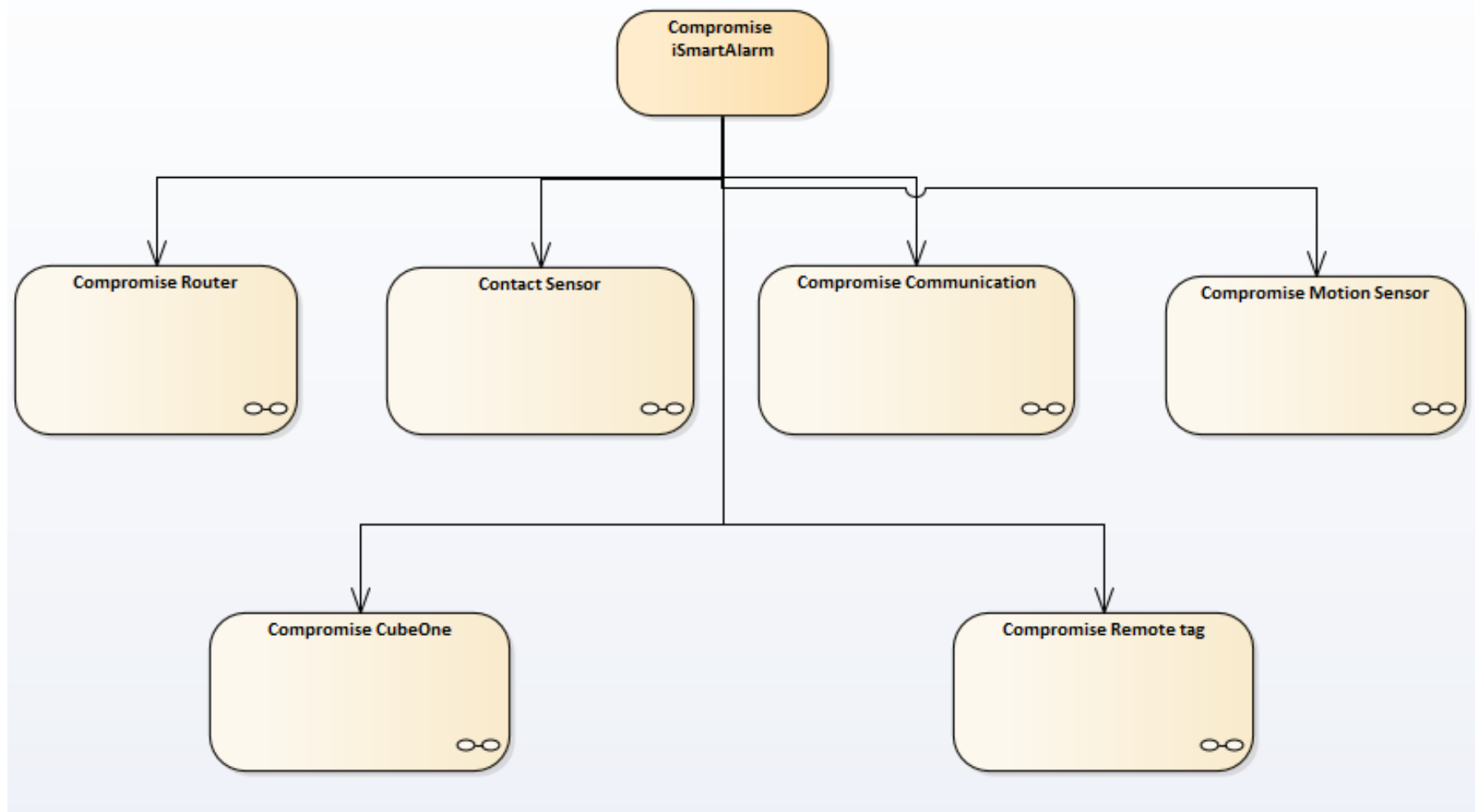
- Phishing and Whale jacking attacks to get the privilege of the super user using social engineering techniques.

The countermeasures against this type of attack is to check the authenticity of the email and not responding to any mail with any private information.

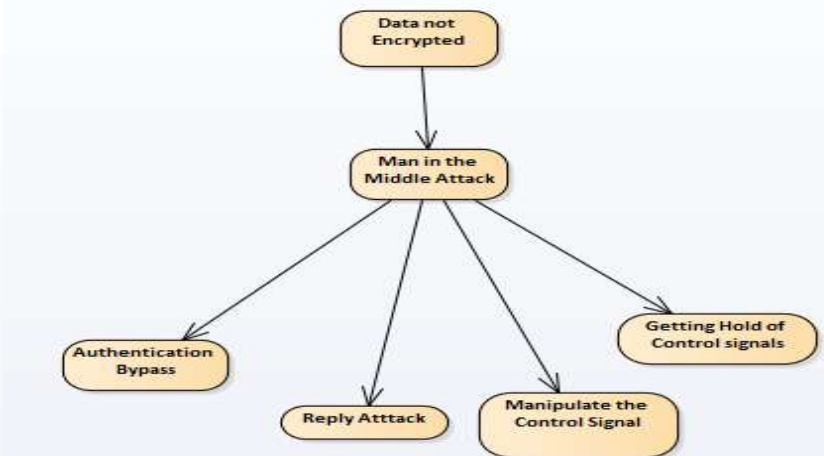
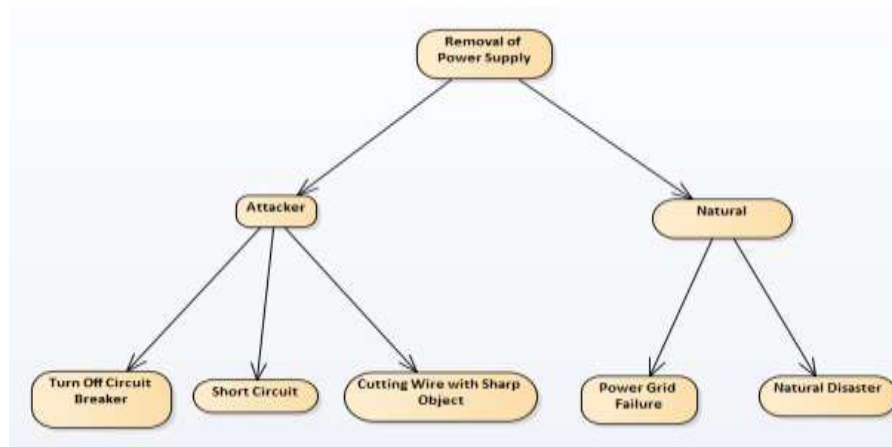
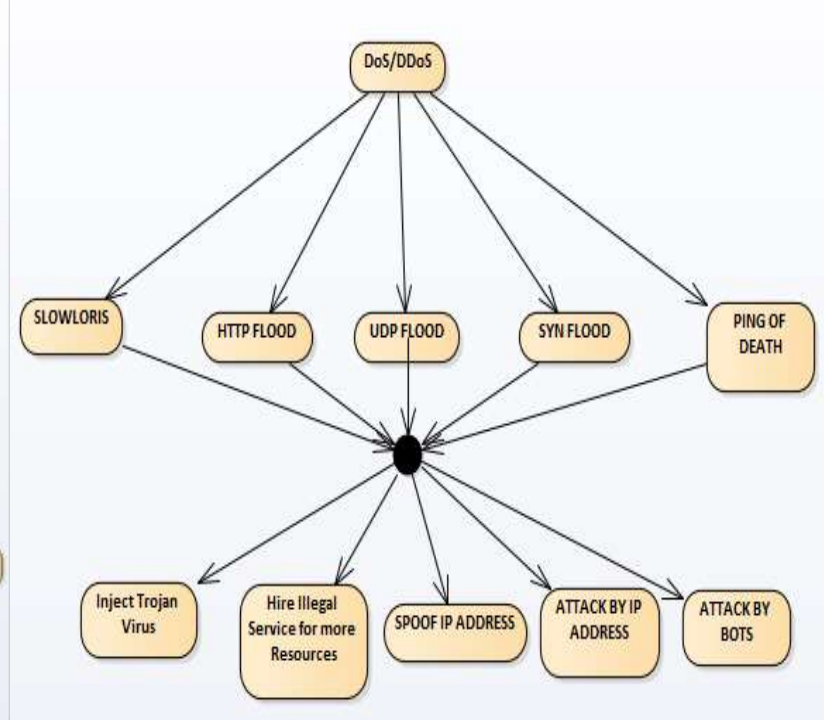
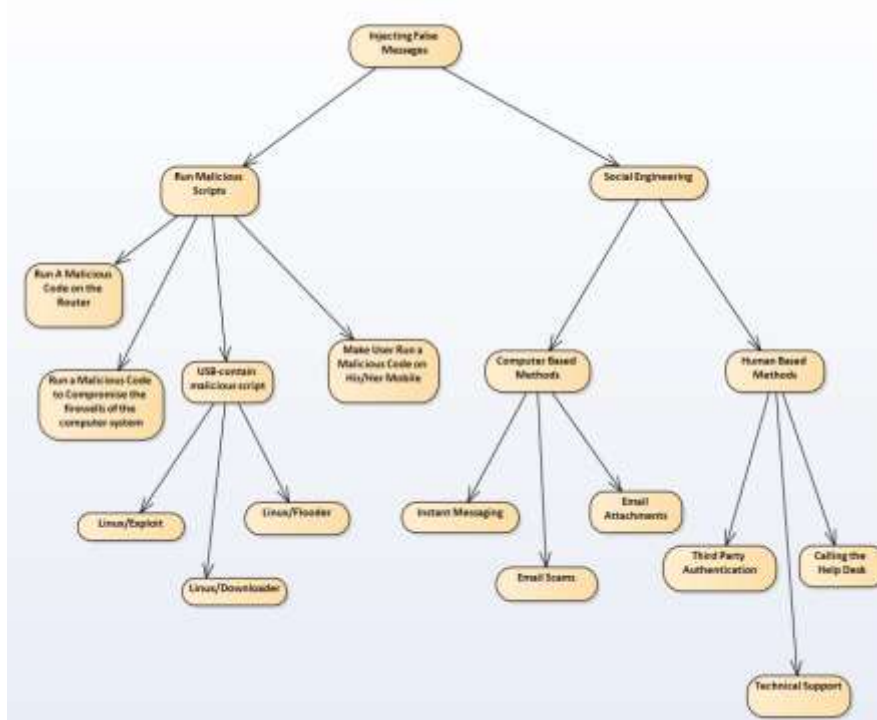
5.2. ATTACK TREE

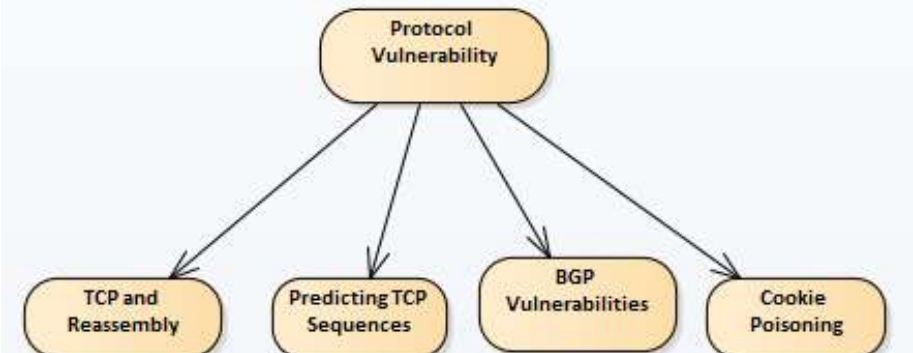
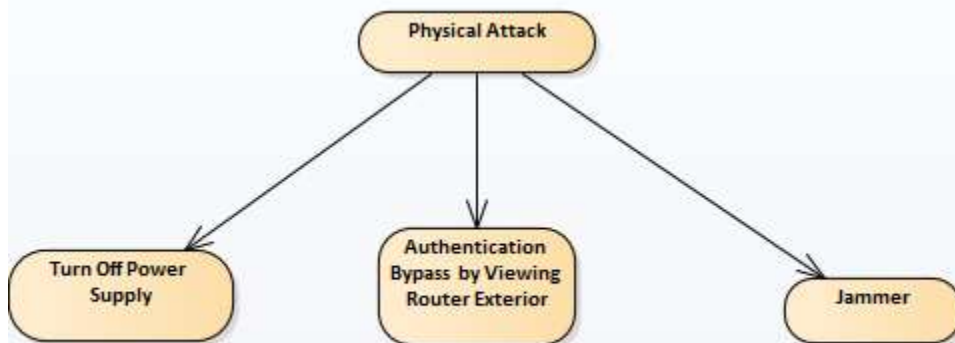
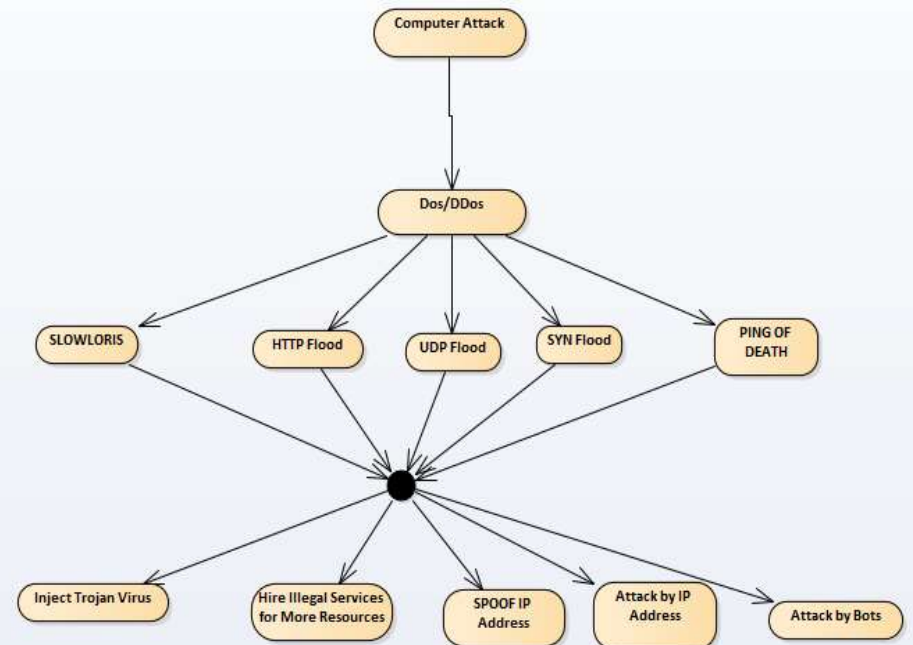
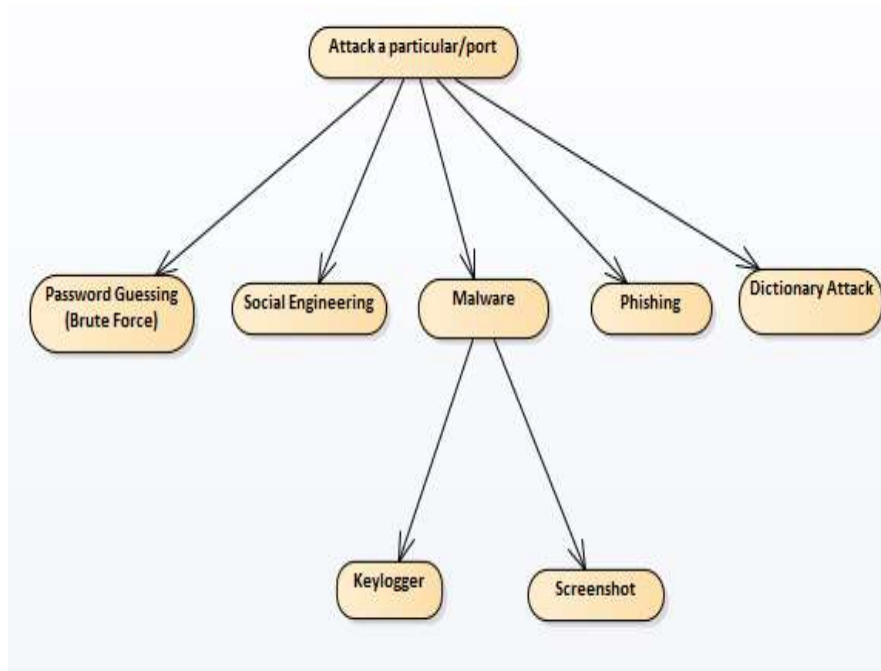
Attack trees are diagrams that shows the different ways in which our asset can be attacked. The below explains the different ways in which the iSmartalarm can be compromised. In the field of cyber security attack trees are used to describe the various attacks that cab be carried out on the system. Attack tree are multi levele diagrams that conatins one root, leaves and children. The root is the one that is on the top. From the bottom up child nodes are the conditions that need to be satisfied to make the parent node that is connected directly to them true. When a root is satisfied the attack is complete. Each node can only be satisfied by its direct child nodes. The attack tree is not just limited to the the conventional information system. Attack tree are used in industries to find the range of attacks that can be carried on their products. The attack tree method is a simplified process that helps to identify all the attacks on the system.

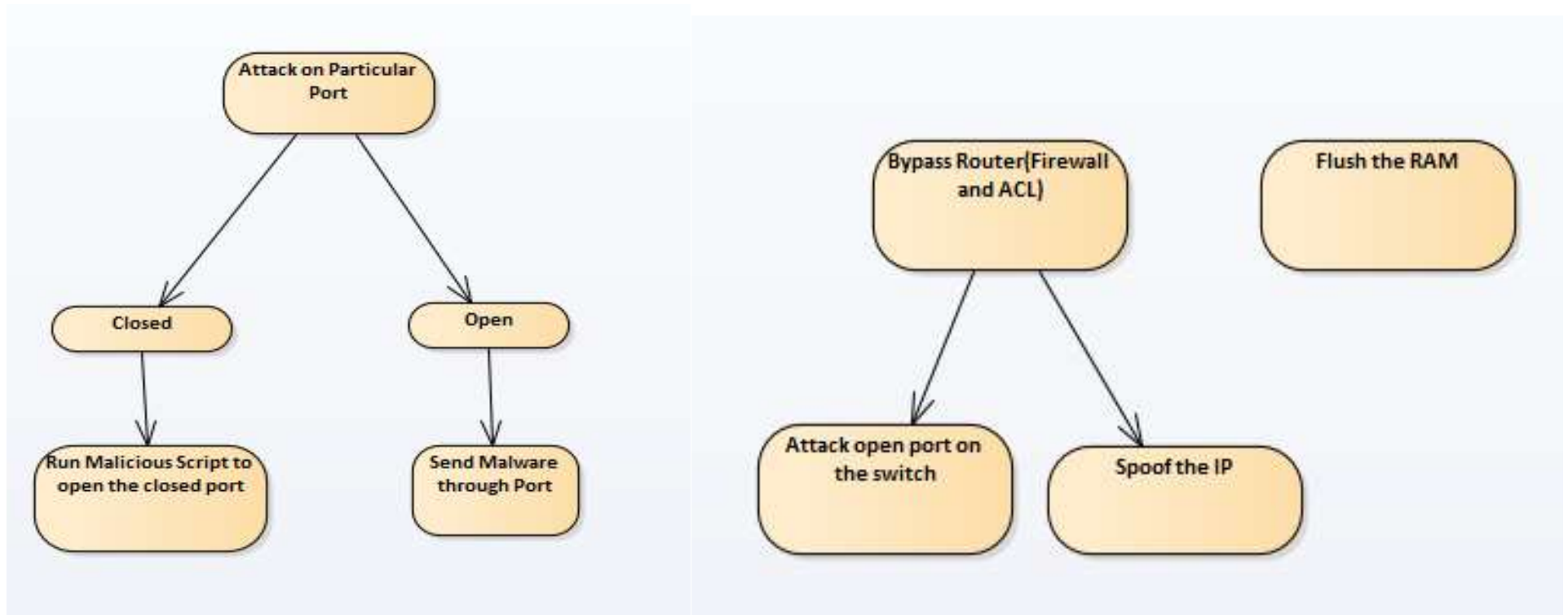
Compromising the iSmartAlarm is the root of the tree and the the Compromise routers, compromise sensors, compromise communication, compromise motion sensors ,Compromise CubeOne and Compromise Remote tag are its branches. Each of the branch is subdivided in depth.



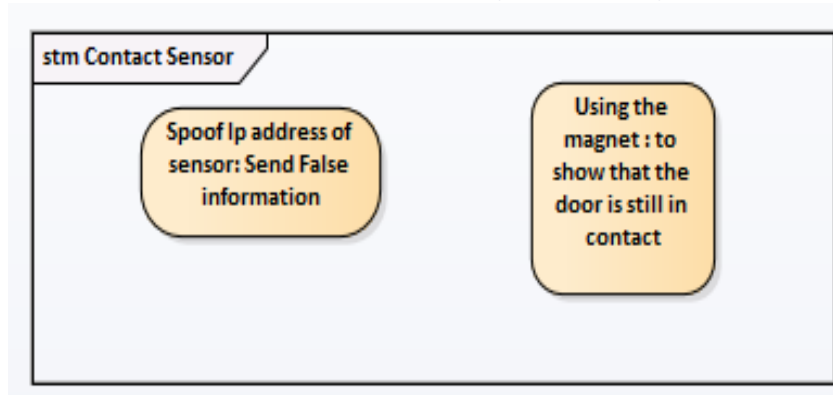
5.2.1. COMPROMISE CUBEONE(4 BRANCHES)



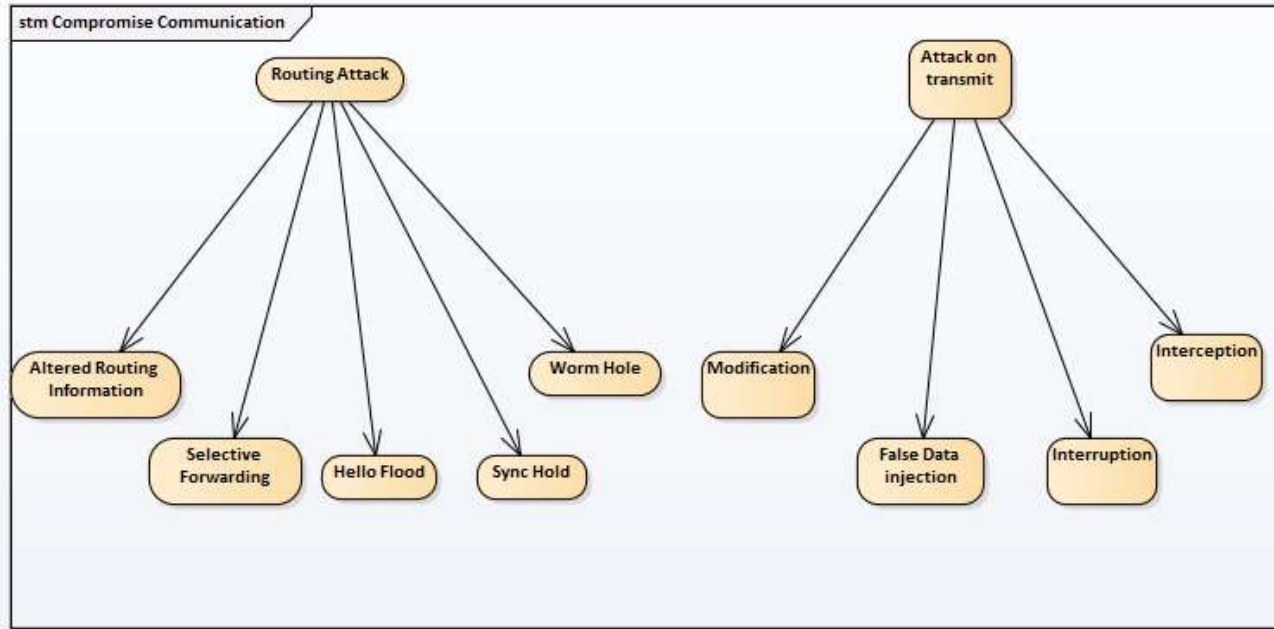
5.2.2. COMPROMISE ROUTER(6 BRANCHES)



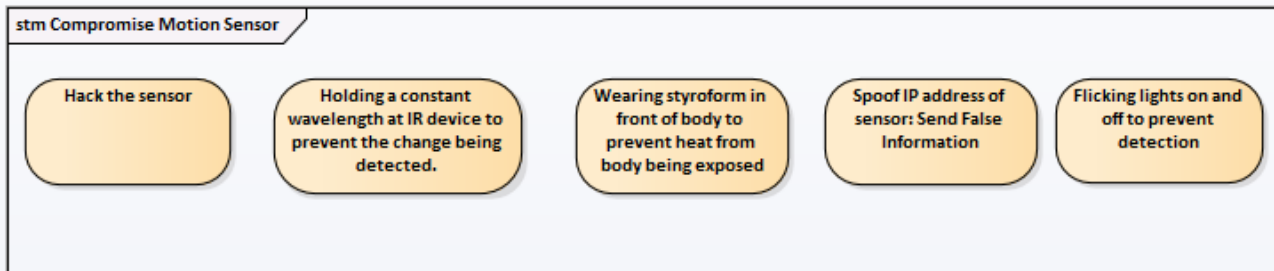
5.2.3. COMPROMISE CONTACT SENSOR (2 BRANCHES)



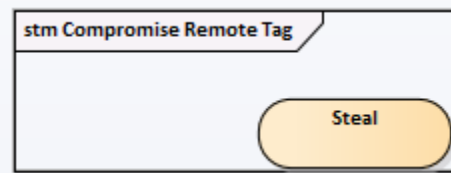
5.2.4. COMPROMISE COMMUNICATION (2 BRANCHES)



5.2.5. COMPROMISE MOTION SENSOR (5 BRANCHES)



5.2.6. COMPROMISE REMOTE TAG



Explanation of the above attack tree

Compromise the CubeOne

1-DOS/DDOS

1. PING OF DEATH – Send IP packets larger than 65,535 bytes making the packet to be broken down into smaller fragments.
2. SYN FLOOD- Attacker sends a succession of SYN packet and does not send the ACK packet making the system to consume enough server's resources preventing it from providing service to the authentic user.
3. UDP FLOOD – Many UDP packets are send to the ports to get the information about the open ports (ie we get reply as destination unreachable)
4. HTTP FLOOD - Large number of HTTP attacks are send to the actual server making it deny the service that it should provide to the actual user.
5. SLOWLORIS – Send partial requests and make the server wait for the complete packet. Make the connections to hold for as long as possible by sending the partial requests and never completing the request.
 1. SPOOF IP ADRESS
 2. ATTACK BY IP ADRESS
 3. ATTACK BY BOTS:
 4. Available for free online
 5. Available by buying it
 6. Inject Trojan virus and make bots

2-Removing Power supply

1. Natural:
 - 1-Power Grid
 - 2-Natural disaster
- 2.Attacker:
 - 1-Physical cutting it
 - 2-Short Circuit
 - 3-Turn off circuit breaker

3-Data not encrypted -MITM:

1. Authentication bypass – The attacker can get hold of the data that is in plain text and use it for malicious purposes later.
2. Reply attack – Catch the data that is being transmitted and use this data at a later point of time to get access to the system.
3. Getting hold of control signals – The control signals are the most important signals and getting hold of them in plain text is like getting hold of the vital function of the system.
4. Manipulate the control signals – Manipulation of control signals by using the control signals of the attacker to perform task as the actual control signals.

4-Injecting false messages:

1. Social engineering:
 1. Human Based Methods
 2. Impersonation
 3. Technical support
 4. Calling the Help Desk
 5. Third Party Authorization
2. Computer Based Methods
 1. Instant Messaging
 2. Email Scams
 3. Email Attachments
3. Run malicious scripts:
 1. USB - contain malicious script
 - Linux/Exploit
 - Linux/Downloader
 - Linux/Flooder
 2. Make the user run a malicious code on his mobile
 3. Run a malicious code on the router
 4. Run a malicious code to compromise the firewall of the computer system

Compromise the communication

1-Routing attacks

- 1-Altered routing information – The routing information can be altered by modifying the bandwidth of the routers and make the data to flow from the compromised routers.
- 2-Selective forwarding - Selecting the route the data need to travel from CubeOne to the cloud.
- 3-SYNC flood – Several SYNC packets are sent to the networking device so that it prevents the device from functioning.
- 4-Wormhole – A attacking node captures the packet from one location and transmit them to other node that is located at a distant location.
- 5-hello flood – The attack is denial of service attack that uses datagram protocol (User Datagram Protocol) a session less/connectionless computer networking protocol.

2-Attack on transmit

- 1-Interruption – The data is interrupted while being transferred from the source to the destination.
- 2-Interception – The data in transmit is intercepted by the attacker. The attacker can get hold of the data in transmit.
- 3-Modification – The data is modified while being transferred from the source to the destination. The attacker can modify the data in transit.

4-False data injection – The attacker can inject false information into the data.

Compromised Router:

1. Physically

- Turn Off power supply
- Authentication bypass
- Jammer

2. Software-DDoS attack:

- 1-Sync flood
- 2-Ping of death
- 3-UDP FLOOD
- 4-HTTP FLOOD
- 5-SLOWLORIS

3.Attack a particular port

1-Open port

- a. Attack port – Use the port as a gateway for the attack to take place. The open ports are more vulnerable to the attack.
- b. Send malware through this port – Use a USB port to inject malware into a port. Always port must be closed so that an attacker cannot get into the system.

2-Closed port

- a. run malicious script to open the closed port

4.Password guessing

- 1-Brute force – Use of all possible combination of passwords so that the access can be obtained.
- 2-social engineering
- 3- Dictionary attack – Usually common words are stored, and this is used to carry out the attack.
- 4-Phishing – It is a attempt to obtain the sensitive information, this attack is usually carried out by email spoofing.

5-Malware: Inject the malware using USB

- 1-Key loggers – Captures the key that are pressed in keyboard.
- 2- Screen shots – Captures screen shot every small interval of time.

5.Protocol vulnerability:

- 1.TCP and Reassembly – The packets need to be reassembled at the destination. This can be used as a attack by not sending the portion of packet, so that the destination keeps waiting for the packet.
- 2.BGP – This vulnerability existed, but now a patch is released, and it is fixed.

- 3-Predicting TCP Sequences – The sequence number of TCP can be predicted and the data in the particular TCP sequence packet can be obtained.
- 4-Cookie Poisoning – This process involves modification of the cookie to get unauthorized information about the user.

6.Bypass routers (firewall and ACL):

- 1-Spoof the IP address of sensor-IP address can be changed; the attacker changes his IP so that his IP address cannot be found, and it is impossible to block the IP.
- 2-Attack open port - When a port is opened it can be used as a gateway of the attack.
- 3-Flush the RAM - The RAM can be flushed so that the memory stored in it is removed. The RAM cannot hold information if the device turns off.

Motion sensor:

- 1-Holding constant wavelength at IR device to prevent the change being detected
- 2-Spoof the IP address of the sensor and send false information stating it to be the actual sensor
- 3-Hack the sensor
- 4-Wearing styrofoam in front of body to prevent heat from body being exposed
- 5-Flicking lights on and off (within 1 second) to prevent detection

Contact Sensor


- 1-Using a magnet to act like the contact sensor is still connected
- 2-Spoof the IP address of the sensor and send false information stating it to be the actual sensor

Compromise remote tag:

- 1-Steal

5.3. VULNERABILITY ASSESSMENT SUMMARY

Likelihood	Consequences				
	1	2	3	4	5
5					
4		1.2	2.1 3.1	5.6 2.2 5.2	5.1
3		4.1 2.3	5.5	6.1	4.2
2	1.5	3.2	5.7	1.3 1.4 5.4	6.2
1			1.5	5.8	5.3

Extreme | 

High

Moderate/Low

5 = almost certain, 4 = Likely, 3 = Moderate, 2 = Unlikely, 1 = Rare

Rank	Description
1	SYN Flood (DOS, DDoS)
2	Ping of Death (DoS, DDoS)
3	Password Guessing (Information Disclosure)
4	Magnet Tampering(Tampering) IR sensor prevention (DoS, DDoS)
6	Enable False Alarm(Repudiation)
9	Spoof IP address of sensor(Spoofing)

6. REMEDIATION SUMMARY

6.1. REQUIREMENTS

- 1-The system shall encrypt of messages that travel between the cloud and the cube. (1.3)
- 2-The system shall use both symmetric and asymmetric encryption methods for security purpose. (1.3)
- 3-The system shall have strong encryption mechanisms like AES 256. (1.3)
- 4-The system shall use a strong password, ie a long password that is a combination of special characters and numbers, Use of IPS can also prevent this problem. (4.2)
- 5-The system shall notify IP and MAC address before accepting any data. Only important information need to be passed and not the entire data. (5.7)
- 6-The system shall disable the port that is not required. (6.1)
- 7-The system shall use static routing mechanism to carry out the routing process. (1.1)
- 8-The system shall use a hard drive. (3.2)
- 9-The users shall have privileges set to access CubeOne. (4.1)
- 10-The system shall use Firewall, Filtering, Active monitor, Intrusion Detection and Intrusion Prevention System. (5.1)

6.2. VULNERABILITY ASSESSMENT SUMMARY AFTER COUNTERMEASURES

Comparing the above diagram in 5.3 with the below diagram we can see that implementing the countermeasures made the risk to decrease the impact and likelihood.

Likelihood	Consequences				
	1	2	3	4	5
5					
4				5.1	
3		2.1	2.2	6.1	
2		2.3 1.2 2.4	5.5	5.2 4.2	
1	3.2 1.5	4.1 3.1	5.4 1.4 1.3 5.7		

7. CONCLUSION

Through our analysis of this iSmartAlarm, we believe there are a lot places this system can improve, and following our suggested requirements would be a good start. IoT devices are currently struggling with vulnerabilities and this system is no different. We personally would not recommend buying this device as we believe home security is an invaluable resource to keep safe from intruders. The price difference between this home security system compared to more traditional systems is significantly different; however, the difficulty of getting around the ladder's security would be much tougher and the response time to intruders would be quicker. The iSmartAlarm creators as well as the IoT industry need to start putting more money into security before these devices can properly serve their intended purpose