Software Requirements Specification

for

IoT Sensor Blinding

Version 1.0

Prepared by Cole Clements, Jeremy Gluck, Xuchao 'Steven' Jiang, Todd St. Onge, and Alex Winstead

September 2019

Table of Contents

1. Introduction

1.1 Purpose

The Internet of Things Sensor Blinding project is an attempt to do penetration testing on a variety of network connected home security devices including Amazon echo devices, Arlo cameras, August locks, Blink home security systems, Geeni cameras and doorbells, Google home and nest products, Kwikset locks, Night Owl cameras, Philips light bulbs, Ring motion sensors, cameras, and alarms, Ring motion detectors, cameras, and alarms, Samsung lights, water leak sensors, and cameras, Schlage door locks, Simplisafe security systems, Swann security cameras, TP-Link security cameras, and Yale smart locks. The extent of the penetration testing will involve collecting wireless activity from all listed IoT devices in order to build a robust labeled dataset for machine learning model building for identifying traffic generated by IoT devices in a generalized setting.Previous research by Dr. TJ O'Connor has shown that many IoT devices communicate using the following schema: two channels of communication are used. The first is the always online heartbeat signal that communicates with the master server to inform it that the device is online and functioning. The second channel is the on-demand channel which is utilized when the user (or intruder, etc) interacts with the device. For many devices blocking the on-demand communication channel prevents the device from reporting events (including intrusion) and events are not cached.

1.2 Intended Audience and Reading Suggestions

This document is intended for graders and reviewers of this senior project as well as its team member for reference purposes.

1.3 Product Scope

The following products will be tested:

Amazon - All-New Echo Dot Kids Edition Smart Speaker with Alexa - Rainbow
Amazon - Echo Dot (3rd Gen) - Smart Speaker with Alexa - Charcoal
Amazon - Echo Show 5 Smart Display with Alexa - Charcoal
Arlo - Pro 2-Camera Indoor/Outdoor Wireless 720p Security Camera System - White
August - Smart Lock Pro + Connect - Dark gray
Blink - Wireless Home Security System - White
Geeni - Indoor Wi-Fi Wireless Network Surveillance Cameras (2-Pack) - Black

Geeni - Pan and Tilt Indoor Wi-Fi Wireless Network Surveillance Camera - White
Geeni - Smart Wi-Fi Video Doorbell - Wired - Black
Google - Home Mini - Smart Speaker with Google Assistant - Charcoal
Google - Nest Cam IQ Indoor Full HD Wi-Fi Home Security Camera - White
Google - Nest Hello Smart Wi-Fi Video Doorbell
Google - Nest Secure Alarm System - White
Kwikset - SmartCode Z-Wave Deadbolt Lock - Polished Brass
Night Owl - Smart Wi-Fi Video Doorbell - Wired - Black
Philips - Hue Color 3pk Starter Kit with Lightswitch - Multicolor
Philips - Hue Play White & Color Ambiance Smart LED Bar Light (2-Pack) - Multicolor
Ring - Alarm Motion Detector - White
Ring - Alarm Starter Home Security Kit - White
Ring - Motion Sensor - White
Ring - Spotlight Indoor/Outdoor 1080p Wi-Fi Wireless Security Camera - Black
Ring - Video Doorbell Pro and Chime Pro Bundle - Satin Nickel
Ring - Wi-Fi Smart Video Doorbell - Multi
Samsung - Button - White
Samsung - Motion Sensor - White
Samsung - Multipurpose Sensor - White
Samsung - SmartThings Arrival Sensor - White
Samsung - SmartThings Hub - White
Samsung - SmartThings Indoor 1080p Wi-Fi Wireless Security Camera - White
Samsung - SmartThings Item Tracker - White
Samsung - SmartThings Smart Outlet - White
Samsung - SmartThings White A19 Smart LED Bulb - White
Samsung - Water Leak Sensor
Schlage - Encode Wi-Fi Touch Screen Deadbolt - Matte Black
SimpliSafe - Entry Sensor - White
SimpliSafe - Glassbreak Sensor - White

SimpliSafe - Motion Sensor - White
SimpliSafe - Pro Smart Wi-Fi Video Doorbell - Wired - White
SimpliSafe - Smart Lock Black + Black PIN Pad - Black/Black
SimpliSafe - Wireless Home Security System - Black
Swann - Indoor/Outdoor 1080p Wi-Fi Wireless Security Camera - White
TP-Link - Kasa Spot Indoor 1080p Wi-Fi Wireless Security Camera - Black/White
Yale - Assure Lock Touch Screen Smart Lock - Satin Nickel
Yale - T1L Z-Wave Touchscreen Deadbolt Replacement Smart Lock - Nickel

1.4 References

1) https://github.com/FloridaTech-IOT-Security-SProject-19-20/Project-Plan?organization=FloridaTech-IOT-Security-SProject-19-20&organization=FloridaTech-IOT-Security-SProject-19-20

2) https://enck.org/pubs/oconnor-wisec19b.pdf

   T. Oconnor, W. Enck, and B. Reaves, “Blinded and confused,” *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks - WiSec 19*, May 2019.

3) https://dl.acm.org/citation.cfm?id=3323409

   T. Oconnor, R. Mohamed, M. Miettinen, W. Enck, B. Reaves, and A.-R. Sadeghi, “HomeSnitch,” *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks - WiSec 19*, May 2019.

2. Overall Description

2.1 Product Perspective

2.1.1 The product is furthers the research of Dr. TJ O'Connor at Florida Tech. The research of Dr. O'Connor includes the use of Groove API on capturing logs generated by IoT devices.

2.1.2 The product uses Z-Wave platform, Zigbee protocol, along with the 802.11 wifi protocol.

2.1.3 Third party open source software is used, which includes the complete library of aircrack-ng and Groove API.

2.2 Product Functions

2.2.1 The product is used to test the security of IoT devices, especially home security devices with Wi-Fi and Z-Wave functions. The list of devices includes popular smart doorbells, smart door locks, home security cameras, home security sensors, Amazon Alexa, Google Home, etc. The detailed list of testing devices is in Section 3.2.

2.2.2 The product captures data frames, decerns the semantic content of frames, identifies the hardware, and manipulates the devices.

2.3 User Classes and Characteristics

2.3.1 IoT manufacturers should use the system during development and testing of their products. .

2.3.2 The user needs to understand basic coding and have basic command line knowledge.

2.4 Operating Environment

2.4.1 List of devices includes Raspberry Pi, Linux computers, Windows computers, Apple computers, and Kali Nethunter certified devices.

2.4.2 The operating systems for capturing devices include ARM based Linux, Debian and related distros, Red Hat Linux and related distros, Windows, Mac OS and Android. The preferred distribution of operating system is Kali Linux.

2.4.3 The devices need to be able to sniff Wi-Fi packets and capture Zigbee packets with third party dongles and compatible third party drivers.

2.4.4 The device need to be able to run aircrack-ng and its dependencies.

2.5 Design and Implementation Constraints

2.5.1 The developers must develop a model that generalizes to a potentially noisy environment. In reality users of smart home devices will often own many devices, so the model will need to be able to classify by device and signal type.

2.5.2 Zigbee protocol, Z-Wave protocol, and IEEE 802.11 protocol are used for data capturing.

2.5.3 The capturing hardware should satisfy the operating environment requirement.

2.6 User Documentation

2.6.1 A less than one minute video of a demo will be available on our website.

2.6.2 An academic paper will be published which will show our results to vendors in the form of security disclosures.

2.6.3 An Academic Paper will be written and submitted to a conference on wireless security.

2.7 Assumptions and Dependencies

2.7.1 All IoT devices are assumed to be powered on and are within range of being captured by the software.

2.7.2 Packet capturing device must meet the basic requirement of operating environment.

3. External Interface Requirements

3.1 User Interfaces

There is no GUI in this project.

3.2 Hardware Interfaces

Communication Protocols: Z-Wave, Zigbee, 802.11

3.2.1 Z-Wave:

Zwave is a wireless communications platform mostly used in home automation to communicate from appliance to appliance. This method of mesh network uses low energy radio waves to achieve the appliance to appliance communication to control home systems such as locks, fire alarms, security systems, windows, swimming pools, and other various home appliances or components using electronic devices such as computers, tablets, laptops, and phones.

3.2.2 Zigbee:

Zigbee is a wireless communication protocol based off of the IEEE 802.15.4 based specification that uses low powered radio waves to create personal networks on a Wi-fi network that communicated on an application to application basis. This communication then results in the ability to control home systems such as locks, fire alarms, security systems, windows, swimming pools, and other various home appliances or components using electronic devices such as computers, tablets, laptops, and phones.

3.2.3 802.11:

The IEEE 802.11 is a series of LAN communication protocols that creates a LAN Wi-fi network that allows electronics to communicate without the need of a hard wired connection. Some of these electronic devices include: Smartphones, printers, and laptops.

Compatible Hardware Component(s):

Amazon - All-New Echo Dot Kids Edition Smart Speaker with Alexa - Rainbow
Amazon - Echo Dot (3rd Gen) - Smart Speaker with Alexa - Charcoal
Amazon - Echo Show 5 Smart Display with Alexa - Charcoal
Arlo - Pro 2-Camera Indoor/Outdoor Wireless 720p Security Camera System - White

August - Smart Lock Pro + Connect - Dark gray
Blink - Wireless Home Security System - White
Geeni - Indoor Wi-Fi Wireless Network Surveillance Cameras (2-Pack) - Black
Geeni - Pan and Tilt Indoor Wi-Fi Wireless Network Surveillance Camera - White
Geeni - Smart Wi-Fi Video Doorbell - Wired - Black
Google - Home Mini - Smart Speaker with Google Assistant - Charcoal
Google - Nest Cam IQ Indoor Full HD Wi-Fi Home Security Camera - White
Google - Nest Hello Smart Wi-Fi Video Doorbell
Google - Nest Secure Alarm System - White
Kwikset - SmartCode Z-Wave Deadbolt Lock - Polished Brass
Night Owl - Smart Wi-Fi Video Doorbell - Wired - Black
Philips - Hue Color 3pk Starter Kit with Lightswitch - Multicolor
Philips - Hue Play White & Color Ambiance Smart LED Bar Light (2-Pack) - Multicolor
Ring - Alarm Motion Detector - White
Ring - Alarm Starter Home Security Kit - White
Ring - Motion Sensor - White
Ring - Spotlight Indoor/Outdoor 1080p Wi-Fi Wireless Security Camera - Black
Ring - Video Doorbell Pro and Chime Pro Bundle - Satin Nickel
Ring - Wi-Fi Smart Video Doorbell - Multi
Samsung - Button - White
Samsung - Motion Sensor - White
Samsung - Multipurpose Sensor - White
Samsung - SmartThings Arrival Sensor - White
Samsung - SmartThings Hub - White
Samsung - SmartThings Indoor 1080p Wi-Fi Wireless Security Camera - White
Samsung - SmartThings Item Tracker - White
Samsung - SmartThings Smart Outlet - White
Samsung - SmartThings White A19 Smart LED Bulb - White
Samsung - Water Leak Sensor

Schlage - Encode Wi-Fi Touch Screen Deadbolt - Matte Black
SimpliSafe - Entry Sensor - White
SimpliSafe - Glassbreak Sensor - White
SimpliSafe - Motion Sensor - White
SimpliSafe - Pro Smart Wi-Fi Video Doorbell - Wired - White
SimpliSafe - Smart Lock Black + Black PIN Pad - Black/Black
SimpliSafe - Wireless Home Security System - Black
Swann - Indoor/Outdoor 1080p Wi-Fi Wireless Security Camera - White
TP-Link - Kasa Spot Indoor 1080p Wi-Fi Wireless Security Camera - Black/White
Yale - Assure Lock Touch Screen Smart Lock - Satin Nickel
Yale - T1L Z-Wave Touchscreen Deadbolt Replacement Smart Lock - Nickel

3.3 Software Interfaces

3.3.1 Aircrack-ng and its packages are used for capturing and decoding Wi-Fi data.

3.3.2 Groove API is used to decode captured Z-Wave log information.

3.4 Communications Interfaces

3.4.1 Z-Wave protocol used for Z-Wave data capturing or blinding.

3.4.2 IEEE 802.11 protocol for Wi-Fi data capturing or blinding.

3.4.3 Web browser is required to view Z-Wave details in Groove API.

4. System Features

4.1 Detect Wifi Packets

4.1.1 The System will be able to detect wifi packets that are being transferred. It will be able to look at different protocols such as Z-Wave, Zigbee and 802.11. This is considered high priority because this feature will feed its data to our system for multiple purposes

including its use in a machine learning model as well as to help determine which devices are devices we are looking to blind.

4.1.2 Stimulus/Response Sequences

This behavior will be initiated by the user. This feature will also be reused to gather data which will be used to train the machine learning model. This Feature will continue returning results of the packets it finds and is in contact with. It will not stop until the user requests to halt the program.

4.1.3 Functional Requirements

REQ-1: Discover wifi packets

REQ-2: Interpret packet and send it to Parser.

REQ-3: Continue process until requested to stop.

4.2 Parse Wifi Packets

4.2.1 The System will be able to receive wifi packets that are being transferred from feature 4.1. It will be able to look at different protocols such as Z-Wave, Zigbee and 802.11. This is considered high priority because this feature will parse the received data. Once this data is parsed it will trigger different actions from other systems and send the parsed data to the Machine learning software for it to be used to train the feature.

4.2.2 Stimulus/Response Sequences

This behavior will be initiated by the receipt of data from feature 4.1. This feature will initiate features 4.3 and 4.4 depending on if the data is being used to train a model or the data is being passed to the model to be utilized.

4.2.3 Functional Requirements

REQ-1: Parse Data received

REQ-2: Show basic information from parsed packet

REQ-3: Export the packet data into a usable format for the model.

REQ-4: Unreadable data should be purged and an error message logged.

4.3 Use Data to Train Model

4.3.1 The System will be able to take the data given after it has been parsed and it will use it to train a model to decipher which traffic is on-demand traffic and which traffic is always on traffic. It will also be compatible with different protocols such as Z-Wave, Zigbee, and 802.11. This is considered high priority because this feature will train and improve the machine learning model which will identify target packets for feature 4.4.

4.3.2 Stimulus/Response Sequences

This behavior will be initiated by the user and the receipt of parsed data. This Feature will continue collecting data to train itself on until the user deems the data set large enough to begin the training. There will be no error messages generated from training because it will be up to the user to deem its correctness by displaying test data and actual data in a controlled environment.

4.3.3 Functional Requirements

REQ-1: Store parsed data

REQ-2: Sort parsed data

REQ-3: Extrapolate new data points from existing data.

REQ-4: Train model on data set.

REQ-5: Package Trained Model for use.

4.4 Input Data into Model to decipher packet type.

4.4.1 Upon the users request the system will take data and transfer it to the trained model to output if the data is from a packet that is always on or is on-demand traffic. This will trigger 4.5 in the case of on-demand traffic. This feature is critical to solving the issue as being able to identify the packet type directly correlates with our success in regards to blinding said device in 4.6.

4.4.2 Stimulus/Response Sequences

This behavior will be initiated by the user. This feature will also use the model created in 4.3 to look at the data being passed from 4.2 . This Feature will continue feeding in new results of the packet data it receives and as a result will continue outputting if the packet is on demand or not. It will not stop until the user requests

to halt the program. In cases of finding on-demand traffic the feature 4.5 will be executed and passed the information on the Device the packet game from.

4.4.3 Functional Requirements

REQ-1: Run data in the model and return a result of on-demand or always-on.

REQ-2: In the case of an on-demand packet, execute 4.5.

REQ-3: Continue process until requested to stop.

REQ-4: Log all results and raw data of this process for review later.

4.5 Blind Wifi Packets

4.5.1 This feature will begin the Blinding process on all packets that are on-demand traffic packets. It will take in the information returned from the model as to weather the packets it is receiving are on-demand or not and will blind those that are. It will do this using hardware supplied by Dr.O'Connor and software such as Groove API. This feature while important is the demonstration phase of our research and as long as we can identify the packets types this step will be of little importance as this step can be replaced by other third party software to accomplish this task.

4.5.2 Stimulus/Response Sequences

This behavior will be initiated by the user and the data received from the model. It will result in an error if no data is received or if the blinding fails.

4.5.3 Functional Requirements

REQ-1: receive list of target packets from 4.4.

REQ-2: Choose the selected device based on the packets.

REQ-3: blind those packets by stopping them from being passed to other devices.

4.6 Send Wifi Packets

4.6.1 The System will be able to send out packets based on the data received from the model and parser to create fake packets to replace those we have blinded. These will be sent to the IOT device awaiting a response.

4.6.2 Stimulus/Response Sequences

This behavior will be initiated by the user after a target and basic information is received from 4.5 and 4.4. The Feature will send out a Fake packet to the IOT device and request directions and input. It can also be told to stop transmitting at anytime.

4.6.3 Functional Requirements

REQ-1: Create a fake packet with data passed from 4.4

REQ-2: Send the created Packet to the targeted device.

REQ-3: Continue Blinding or turn off blinding after packet is sent.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 System should be able to identify the on-demand packet, blind and send back a fake packet before the request is fulfilled by the authenticated device.

5.1.2 Device should be able to handle multiple IoT devices interacting with each other and other devices at one time.

5.1.2.1 Devices will be on a network and not hardwired so lots of traffic will need to be sorted and analyzed to distinguish devices and traffic types.

5.2 Safety Requirements

5.2.1 A kill switch should be built so that if the device begins to cause harm to anything it can be shut off instantly.

5.2.2 Any action that would endanger the schools network must be prohibited.

5.3 Security Requirements

Note: Because this is a research project we are able to participate in testing said project in the name of penetration testing and vulnerability research.

5.3.1 No use of this research is permitted outside of lab conditions as it is illegal.

5.3.2 The sale or sharing of this knowledge not under open source or educational means is prohibited.

5.3.3 No data not from within the lab or from a machine owned by a group member is to be used.

5.3.4 Dr. O'Connor serves as our certifier of security and privacy. He will inform the project of ways to ensure no laws or certifications are broken.

5.4 Software Quality Attributes

5.4.1 All code will be kept open source to ensure licensing and use by all team members.

5.4.2 Data created and stored will be backed up to ensure safety.

5.4.3 The system will not be required to be user friendly and is meant to be used only by team members, Dr.O'Connor and other academics for the purpose of vulnerabilities research.

5.5 Business Rules

5.5.1 All functions will be unlocked to the user and no verification to use said product will be required.

6. Other Requirements

6.1 A research paper will be published in conjunction with Dr.O'Connor.

6.1.1 This paper may be presented at security conferences.

6.2 All work shall be able to be passed to Dr.O'connor at the end of the project.

6.3 All supplies and lab space are property of the Florida Tech IoT lab.

6.4 The Supplies are to be kept in a usable state for further use by the Florida Tech IoT lab.

Appendix A: Analysis Models

After collecting data frames from devices we will explore different supervised models for classifying and clustering new frame traffic to find the models most suitable to satisfy the requirements.

Appendix B: To Be Determined List

Build labeled dataset of semantic behaviors of IoT devices.
Test machine learning methods for supervised learning given dataset.
Observe methods for signee and Z-Wave jamming.