Software Design Specification

for

IoT Sensor Blinding

Version 1.0 not approved yet

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

September 2019

TABLE OF CONTENTS

1 Introduction

This document will hack into IoT (Internet of Things) Devices such as smart locks and sensors. This will be done by blinding IoT sensors using a machine learning model that sniffs out data packets that then intercepts them using an external device. This will show the security vulnerabilities that these smart security devices have and how these issues need to be looked into.

1.1 Purpose

1.1.a This document is meant to define the design of the IoT Sensor blinding project for the readership which is primarily advisors and graders of the project.

1.2 Scope

1.2.1 The project should produce a digital model that can learn to identify which IoT packets in a wireless network are on-demand traffic. This model will integrate with existing hardware and software to blind said packets. This project will be used to demonstrate to researchers as well as companies in the IoT space that vulnerabilities exist. Obvious security risks with this product is the need to keep this from being openly available as people can use it to manipulate IoT devices. It is our hope that by publishing an academic paper on this work companies will create patches to stop this exploit from being used.

1.3 Acronyms and Abbreviations

Acronym/Definition	**Meaning**
IEEE	Institute of Electrical and Electronics Engineers
RF	Radio Frequency
WiFi, Wi-Fi	Wireless Fidelity
IoT	Internet of Things

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 System Overview

The System will be built by the following components.

2.1 Packet Sniffer

A hardware software pair will be chosen based on preliminary tests of equipment to gather data frames from the packets it sniffs. This Pairing can only be decided after data has been collected from each test device. This is to ensure we have quality data on a large amount of IoT devices.

2.2 Machine learning model

The model will be chosen based on the type of data we collect. It could take the form of a neural net or even a regression model. Features will be generated from the data collected by the Packet Sniffer. Features could include size, frequency, ip, and other protocol dependent variables. This data will be used to train and produce a model.

2.3 Blinding device

A device will be assembled to utilize the model generated to blind on-demand traffic, this will follow the example Dr.O'Connor has already built. This will simply use said model to interpret the packet before deciding to blind or not.

2.1 System Characteristics

The model will receive network frames in real time and respond by classifying the frames with high accuracy and outputting to the command line. These classifications can be used as input to a script that affects IoT devices by means that will be determined.

2.2 Infrastructure Services

2.2.1

Due to the nature of the project's goal being, to expose vulnerabilities and encourage companies to fix them, many common Infrastructure Services will not be used. A list below will show what is provided with a brief description of how and why.

2.2.1.a Performance monitoring and reporting

We will have Performance monitoring tools integrated with the model to help train and evaluate the model.

2.2.1.b Error Handling

Basic Error handling will be included, this will include dumping data that is broken or useless and logging reasons a program may fail.

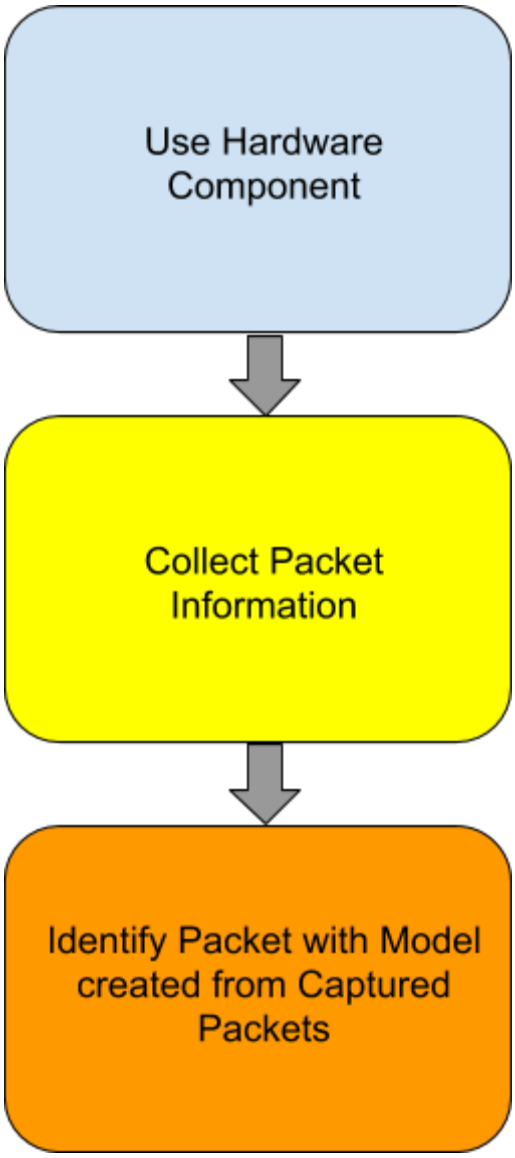
2.2.1.c Logging

As aforementioned the Errors will be logged. In some cases during debugging results will be logged to maintain a complete understanding of the data being used and the process that are happening.

3 System Context

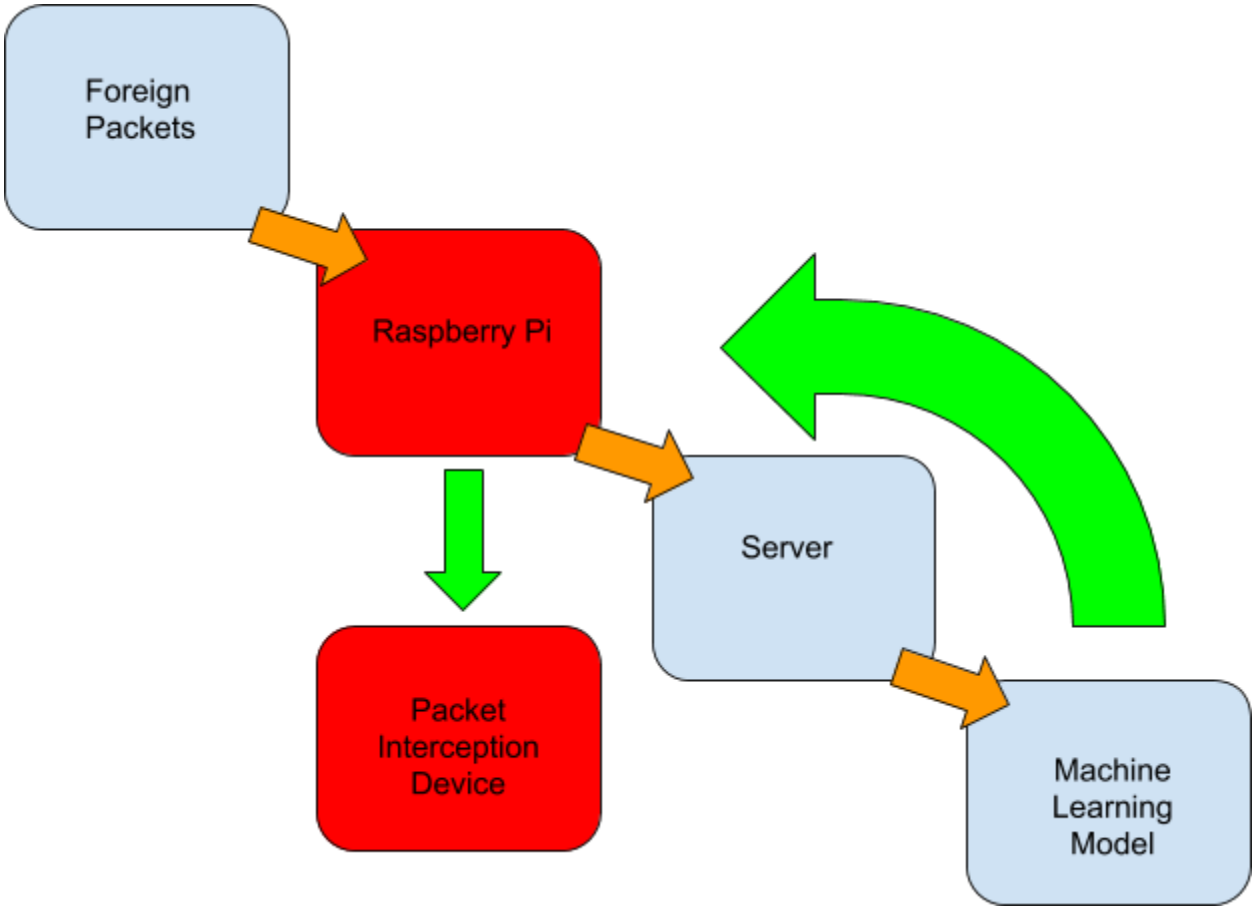
3.1 Hardware Testing Progress:

All hardware components in this project will be used so the packets information they produce can be collected and a machine learning model can then be created from the said packets.



3.2 Raspberry Jam Model:

Forgein Packets are placed into the raspberry pi, which then the data gets transferred into a server, where it will then go through the machine learning model. And be brought back to the raspberry Pi, in which it will either be collected by the Packet interception device. Or go through the machine learning model process again.



4 Documentation

4.1 Documentation Standards

4.2.1 All code shall include a heading including the goal of the code, name and contributions of those who have worked on it and a date.

4.2.2 All written documents will use IEEE conventions.

4.2 Naming conventions

4.3.1 Camel case shall be used in all code for all naming conventions.

4.3.2 Names should be easy to understand and should when possible not include acronyms.

4.3 Software development tools

The list of software tools includes:

- Notepad++
- Sublime Text
- Lucidchart
- Postman
- Python3.7
- gcc
- Google Chrome

4.4 Outstanding Issues

Due to a large portion of this project hinging on viewing and analyzing the data to determine the optimal machine learning algorithm to use the exact type of model and by extension the way it will be trained has been omitted until we have a working lab that is collecting data to be interpreted. This will remain an outstanding issue until the first data has been collected from the lab

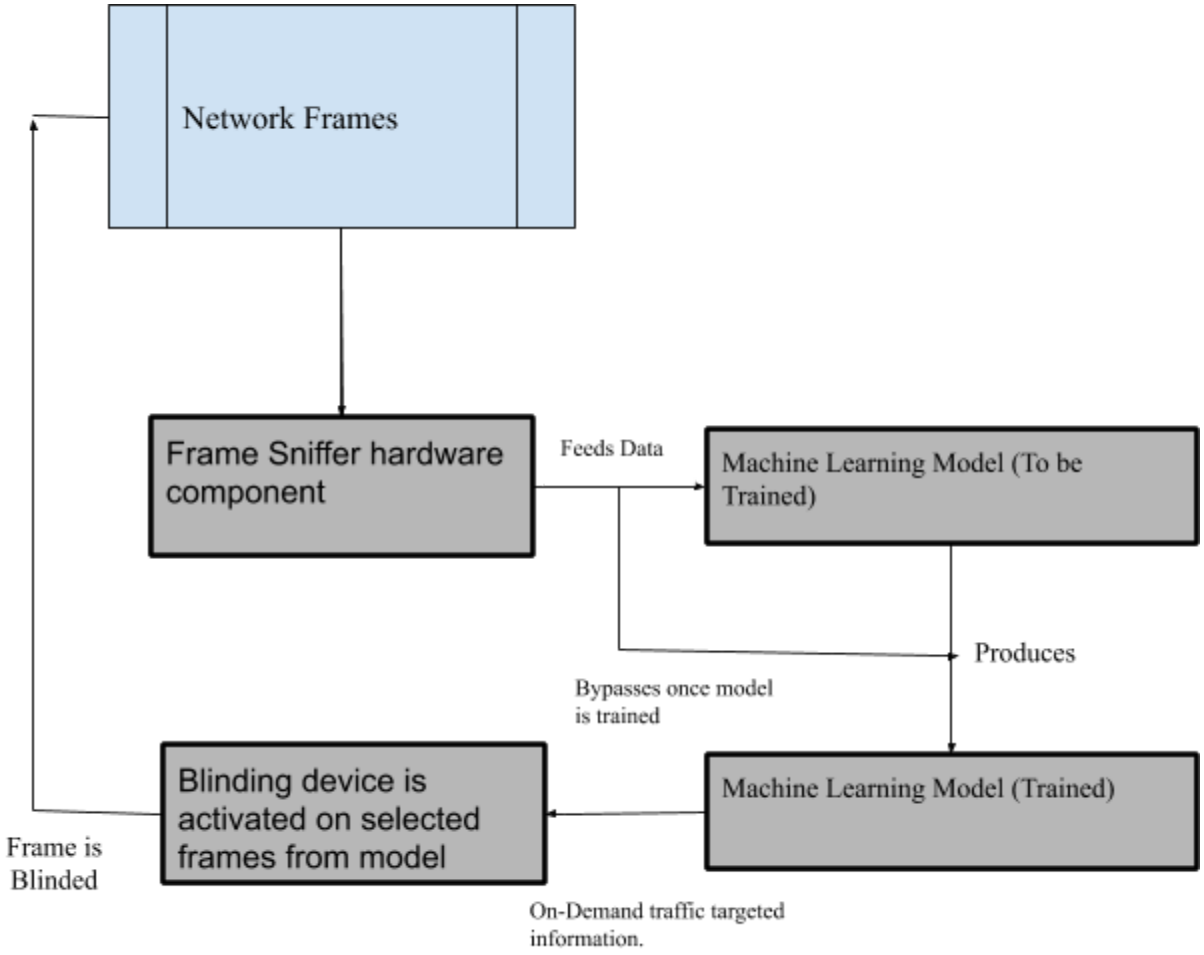
4.5 Decomposition Description

4.6.1 Packet Sniffer: The device on the Raspberry Pi that will access packets and gain information on them. This information varies based on the protocol being used.

4.6.2 Machine Learning Model (to be trained): the untrained model which will be fed all types of data and be assisted by us to train the model to recognize the packets we are looking to blind.

4.6.3 Machine Learning Model (trained): The trained model which will be fed the packet and will decide to blind it or not.

4.6.4 Blinding Device: Is the component that will be integrated with the raspberry pi and allow different types of packets to be blinded from other devices.


Network Frames
Frame Sniffer hardware component
Feeds Data
Machine Learning Model (To be Trained)
Produces
Bypasses once model is trained
Blinding device is activated on selected frames from model
Machine Learning Model (Trained)
Frame is Blinded
On-Demand traffic targeted information.

Document Signoff

Nature of Signoff	Person	Signature	Date	Role
Authors	Team	*Team Signature*	9/30/19	Project Member
Reviewer	Dr. Chan			Grader
Sponsor	Dr. O'Connor			Sponsor

Document Change Record

Date	Version	Author	Change Details
September 29th 2019	Issue 1 Draft 1	Team	First complete draft