FortifIoT

by

Michael Schroeder, Christopher Lyons, Stephan Manahan

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Michael Schroeder 4/30/2019
Michael Schroeder Date

Stephan Manahan 4/30/2019
Stephan Manahan Date

Christoper Lyons 4/30/2019
Christopher Lyons Date

Technical Advisor

Tyler Hopperton 4/30/2019
Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2019

FortifIoT

Prepared by
Michael Schroeder, Christopher Lyons, and Stephan Manahan

Students of
University of Cincinnati
College of Education, Criminal Justice, and Human Services
School of Information Technology

April 15, 2019

Table of Contents

List of Illustrations

ACRONYMS AND ABBREVIATIONS

IoT	Internet of Things
SSH	Secure Shell
DDoS	Distributed Denial of Service
IDS	Intrusion Detection System
API	Application Programming Interface
JSON	JavaScript Object Notation
NIC	Network Interface Card
LAMP	Linux, Apache, MySQL, PHP

ABSTRACT

Gartner reports that there will be around 20.8 billion [IoT devices] by 2020. Among these devices are countless vulnerabilities and security concerns. As of now, there is no single device that remediates those vulnerabilities or even detects them. The Mirai botnet incident took countless websites offline and affected millions of people. What if there was a device that could have detected this sort of activity? Our product would have thwarted an attack of this type via network monitoring of each IoT device on a user's home network. Our product would function as a plug and play device and would be able to detect abnormal IoT network activity. The objective is to ensure the devices are only operating based on necessity. This gives the owners peace of mind that their IoT devices are functioning normally. The number of IoT devices is going to surpass the number of mobile phones and computers in the coming years. This will require the need for additional security measures similar to that of antivirus. Our product is a step in the right direction for securing these devices.

1.0 PROBLEM STATEMENT

1.1 Introduction

Ever since the first Internet of Things device came into fruition, they've been riddled with security vulnerabilities and have become a popular channel for attackers to utilize for launching attacks. This became very apparent after DYN, an Internet performance management and web application security company, fell victim to a DDoS attack whose traffic belonged mostly to insecure IoT devices. This incident took countless websites offline and affected millions of people. What our product accomplishes is network monitoring of IoT devices to determine a baseline for normal network activity of the device, detect any anomaly and send an alert to the web application that would allow the user to blacklist, whitelist, or ignore the alert. Fortifiot is a plug and play device that continuously scans your network for new IoT devices and pulls their baseline operations from a database and treat any deviation from that baseline as an anomaly which will kick off a notification to the user.

1.2 Problem

A new survey by Clutch finds that 67% people who own a connected device own a smart home appliance such as a smart fridge, oven, or TV. About one-third (35%) own a wearable device, and 27% own a digital assistant such as a Google Home or Amazon Echo (Paul, 4). This highlights the fact that more and more people have these devices in their home, which means more data going in and out of user's home networks. The more data there is, the higher the security risk of it going someplace it should not. According to a report conducted by the

Economist Intelligence Unit, 92 percent [of people] say they want to control what personal information is automatically collected" from IoT devices. Furthermore, 74 percent are concerned that small privacy invasions may eventually lead to a loss of civil rights (Paul, 6). The data shows that people are clearly concerned with what data IoT devices are collecting, and where it is going.

1.3 Solution

The Fortifiot device continuously scans your home network, looking for IoT devices. It then makes an analysis on all traffic passing through its' analysis port and generates an alert if the destination IP that your device is visiting it not in the predetermined whitelist. The user can then open the web application and see the new alerts. They can select one or multiple alerts and choose either whitelist, blacklist, or ignore to tell the Fortifiot device what to do with future traffic to and from that destination IP address. If the user chooses to whitelist an alert, this tells Snort to allow any traffic to and from that IP address(s) and don't alert on it in the future. If the user chooses blacklist, this tells Snort to block all traffic to and from the selected IP address(s) and not alert on it in the future. Lastly, if the user chooses the ignore option, this does not tell snort to do anything and if your device visits that IP address in the future the alert will generate again.

1.4 Project Goals

The goal of this project was to provide an easy management resource to individuals with multiple IoT devices on their home network. The application provides another layer of security within the network in order to detect if an IoT device has been compromised. This provides the owner the ability to control where data goes after it leaves the device. By establishing a baseline

of security throughout all IoT devices, we can determine anomalous network activity by flagging any traffic that deviates from the baseline. As IoT devices become more common, they may become more susceptible to such attacks and will require thorough monitoring. This device should be scalable to an enterprise application. With the anticipated rise in IoT devices, this concept is intended to carry the same weight as having antivirus on your computer.

1.5 Project Overview

The following details the progression of Fortifiot, including a table defining the user profile, a diagram displaying the use case, a detailed look into the objectives of the project, and the deliverables. A work breakdown structure and subsequent Gantt chart are provided to detail progress and deliverables. Finally, a written report of Fortifiot's technical elements and user interface round out the discussion. The conclusion will detail our final product, lessons learned and potential future plans.

1.6 User Profile

Figure 1: User Profile, illustrates the user profile for the Fortifiot application. It denotes potential users of the application, related experience and similar applications that the user may have encountered, the tasks the user is expected complete when operating Fortifiot, the expected use of Fortifiot per person, and key interface design requirements Fortifiot will need to optimize.

PROJECT:

- IoT Device Management / Intrusion Detection

POTENTIAL USERS:

- Homeowners
- System administrators

SOFTWARE, INTERFACE, AND RELATED EXPERIENCE:

This solution is targeted for homeowners who may not have in-depth knowledge of technology, or experience with network security. Fortifiot aims to create a last defense of security for IoT devices for a home network, in which, the user simply plugs in the device and allows itself to analyze traffic, devices and activity. While the project is targeted for users with little knowledge, there will be options that allow sophisticated users to create network rules and control their network remotely.

The objective is to establish peace of mind to users, despite of tech-savviness, by constructing a secure network which detects anomalies across a IoT related traffic.

EXPERIENCE WITH SIMILAR APPLICATIONS:

There are multiple applications that are similar to Fortifiot, but none that do exactly what ours does. Other applications functions include scanning the network for IoT devices and presenting the user with a dashboard of sorts to monitor each devices traffic and see exactly what they are doing. Fortifiot does this as well, but it also has the capability to automatically block certain domains, thus making it a more versatile device than others on the market.

TASK EXPERIENCE:

Users will have to complete a first time setup for their Fortifiot device. This will include logging into the device via the web application and providing notification details such as preferred email and phone number. The device will automatically detect IoT network devices and determine a baseline for their behavior. Upon setup completion, users will then be presented a dashboard with network statistics. Users can also expect to receive notifications from the device when abnormal activity is detected. This is to be a plug and play device for users with limited technical knowledge.

FREQUENCY OF USE:

This device will be constantly on and watching traffic of all IoT devices on the network. This will require no user input of any kind as it will all be automated. From scanning for new devices to comparing those devices to their respected baseline of normal network activity. This device will truly be plug and play and forget about, as it will automatically scan and add new devices to the dashboard. The only interaction the user should have with the device after first time setup is responding to alerts via the web-app or SMS message.

KEY UI/UX REQ'S THAT PROFILE SUGGESTS:

- Quick glance of security posture of each IoT device via dashboard and alerts
- Easy block, allow traffic interface via a drop down menu for each alert our device creates

Figure 1. User Profile

1.7 Use Case Diagram

The following diagram, Figure 2: Use Case Diagram, displays the use case for Fortifiot. The Diagram depicts all users of Fortifiot along with the corresponding tasks each user will have access to when interacting with the application.

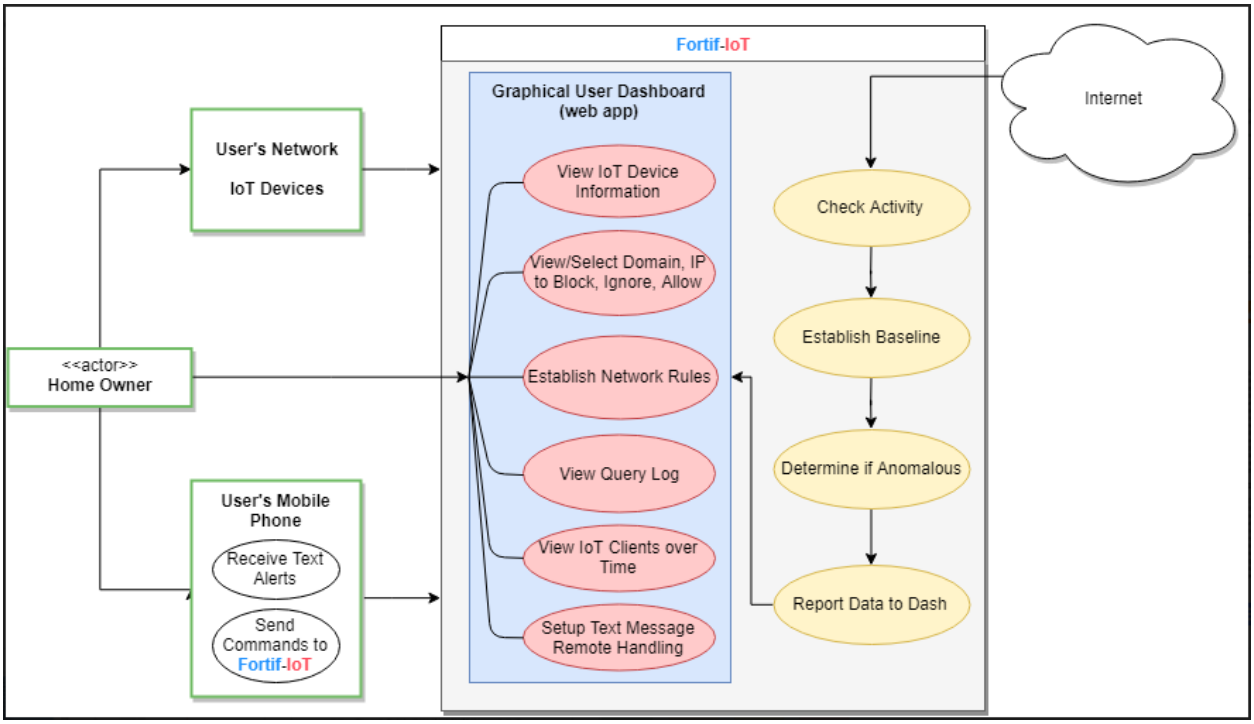


Figure 2. Use Case Diagram

2.0 PROJECT MANAGEMENT

2.1 Budget

Figure 6 displays the budget for the production of the Fortifiot device. This includes costs such as labor, hardware and research. Using real world estimates, the approximate cost of Fortifiot comes out to be **$14,600.00.** The only cost to our team was the hardware component needed to support the device's purposes.

Category	Item	Description	Unit Price ($)	Unit Hours ($/hour)	Total ($)
Labor	Front End Development	Development of the user interface	70	45	$3,150.00
Labor	Back End Development	User interaction and functionality development	70	40	$2,800.00
Labor	Infrastructure	Setting up systems (LAMP, Ubuntu, Snort, etc)	70	45	$3,150.00
Hardware	Minnowboard Quad Core	Cost of the hardware device chosen	250	1	$250.00
Research	Hardware Research	Determining the effectiveness/applicability of hardware devices/infrastructures	70	40	$2,800.00
Research	Software Research	Determining the technology stack which would provide proper functionality	70	35	$2,450.00
Total				**206**	**$14,600.00**

Figure 3. Spring 2019 Project Budget

2.2 Project Schedule

Figure 3: Fall Semester 2018 Gantt Chart, Figure 4: Work Breakdown Structure outlines the projected schedule for completion of this project.

Task	Start Date	Days to Complete
1.0 Initiating	1-Sep	40
1.1 Select PM	1-Sep	1
1.2 Form Team	1-Sep	1
1.3 Team Contract	1-Sep	14
1.3.1 Assign Roles	10-Sep	2
1.3.2 Create Problem Statement	10-Sep	4
1.3.3 Determine Objectives/Goals	14-Sep	4
1.3.4 Develop Project Timeline	18-Sep	4
1.3.5 Meeting With Hopper	10-Oct	1
1.4 Revise Team Contract	10-Oct	1
1.5 Project Abstract	10-Oct	1
2.0 Planning	15-Sep	40
2.1 Gather Requirements	15-Sep	4
2.1.1 Determine Project Scope	15-Sep	1
2.1.2 Refine Initial Req'ts/Goals/Objs	15-Sep	3
2.2 Configure Web Server	10-Oct	4
2.2.1 Set up Raspberry Pi	15-Oct	2
2.2.2 Develop Web App Shell	15-Oct	2
2.3 Research Snort Functionality	11-Oct	2
2.4 Wireshark/TCPDump/Snort Testing Run	20-Oct	3
2.5 Rough Web App UI Design	20-Oct	3
2.5.1 Develop UML Class Diagrams from Design	20-Oct	1
2.5.2 Develop Business Structure Layers	21-Oct	2
3.0 Executing	29-Oct	28
3.1 Business Logic	29-Oct	20
3.1.1 Determine Network Baseline System	29-Oct	5
3.1.2 Develop Notification System	3-Nov	5
3.1.3 Construct JSON Model	8-Nov	5
3.1.4 Web App Shell Development	29-Oct	2
3.2 User Interface	31-Oct	5
3.3 Develop Business Structure Layers	1-Nov	16
3.3.1 Develop Data Transfer Object	1-Nov	4
3.3.2 Develop Data Access Object	1-Nov	4
3.3.3 Develop Service Object	1-Nov	8
4.0 Demonstration	22-Oct	107
4.1 Elevator Speech	22-Oct	1
4.2 Fall Presentation	3-Dec	1
4.3 Testing	18-Nov	5
4.3.1 Bug Testing	3-Jan	5
4.3.2 User Interface Testing	15-Jan	5
4.3.3 IoT Device Testing	18-Jan	5
4.4 Documentation	3-Feb	2
5.0 Closing	15-Feb	65
5.1 IT Expo Preparation	15-Feb	14
5.1.1 Final Paper	15-Apr	4
5.1.2 Create Presentation	1-Mar	6
5.2 Expo Presentation	9-Apr	1

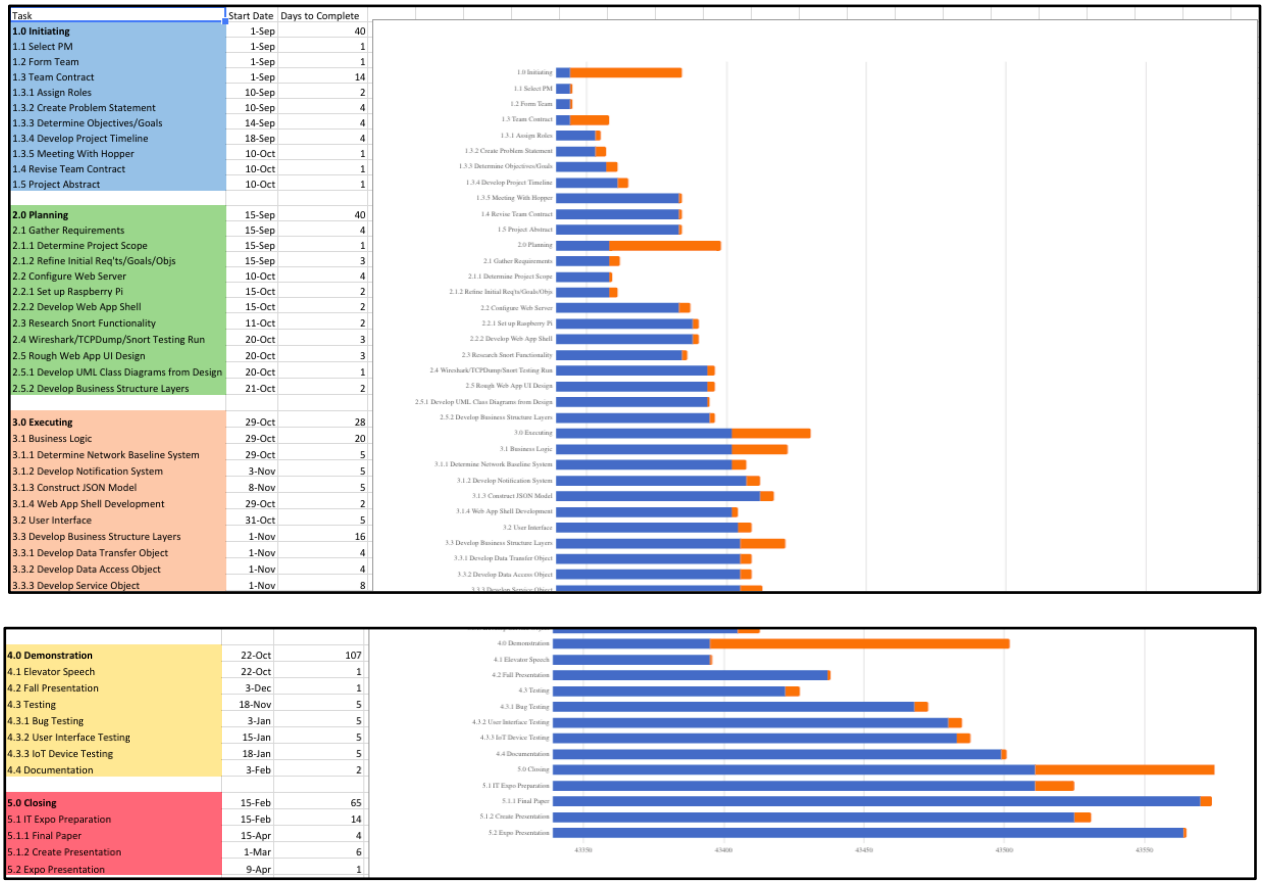


Figure 4. Spring Semester 2019 Gantt Chart

2.3 Work Breakdown Structure

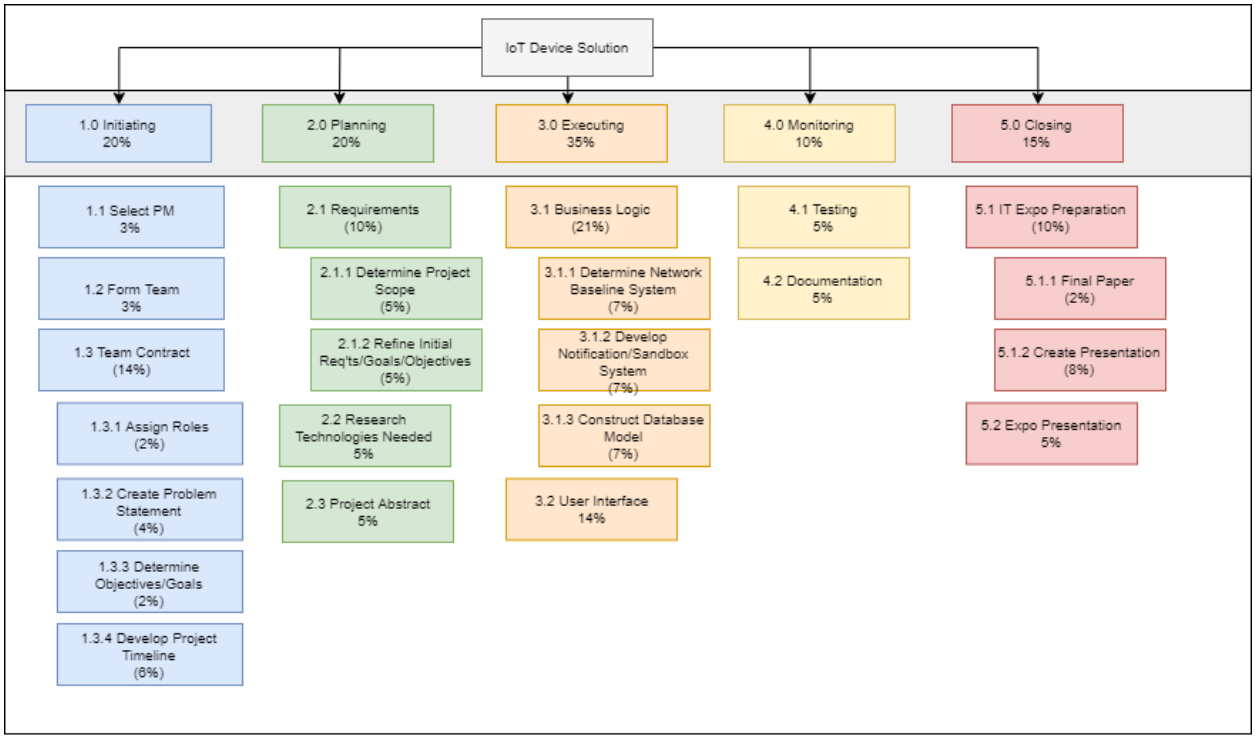


Figure 5. Fall Semester 2018 WBS

2.4 Technical Architecture

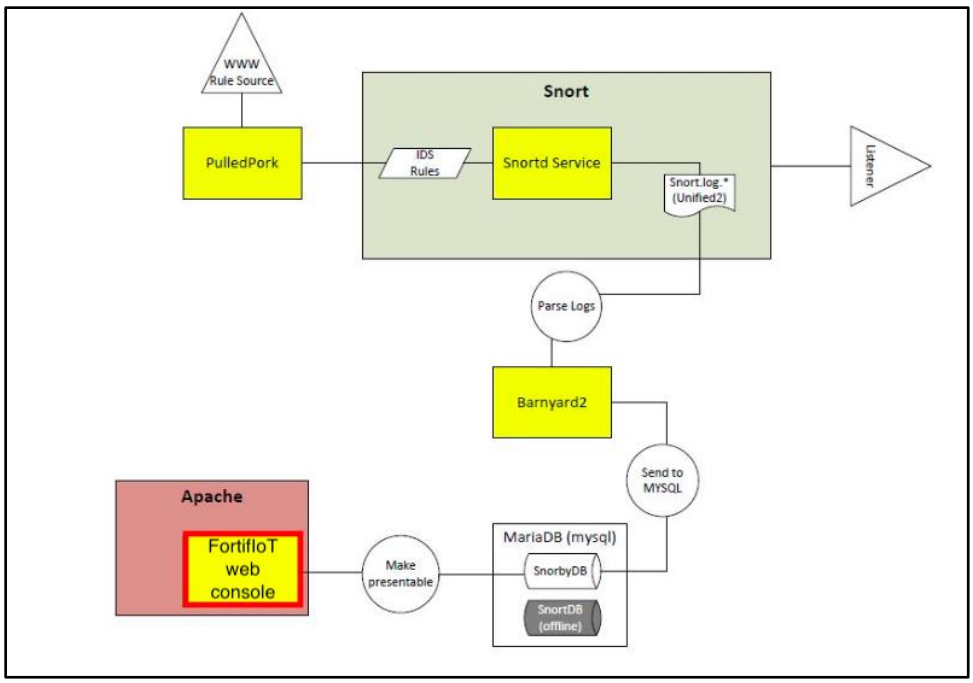


Figure 6. Spring Semester 2019 Technical Architecture

2.5 MinnowBoard

The MinnowBoard was the device we used to run everything on. It includes a quad core intel Atom processor, a 32GB MicroSD card, 2GB of RAM, and 2 GB NIC's.

Figure 7. MinnowBoard Turbot Quad Core Dual NIC

3.0 TECHNICAL DISCUSSION

3.1 Hardware

The hardware we used was a MinnowBoard Turbot Quad Core Dual Network Interface Card (NIC) single board computer. We chose the MinnowBoard because running the Linux, Apache, MySQL, PHP (LAMP) stack and Snort requires more resources than the default Raspberry Pi can provide. In addition to this, the dual NIC capability allows us to run a management and monitoring interface through Snort.

3.2 Software

The software we used was Ubuntu Linux, Snort, Apache Web Server, MySQL, and Php. We used Ubuntu Linux specifically because it was more compatible and reliable with Snort. Apache Web Server, MySQL and Php tied together the remainder of the LAMP stack which allowed us to update our website accordingly with Snort. Lastly, Snort was our first choice for an Intrusion Detection System because it was open source, flexible and has a great user community.

3.3 Application

The application gives users access to a dashboard which displays the alerts on each IoT device on their network with interactive capabilities. The dashboard displays information diagrams based on your device alerts and other useful tabs such as a whitelist, blacklist, and settings feature. The programming languages used were PHP, HTML5, CSS, and Javascript.

The dashboard contains multiple elements that are designed for easy use and provide a clean, modern view. The two core elements are the navigation bar, and the display section. The navigation bar contains all the standard options that come packaged with most applications, (i.e.:

Settings, Activity, Query Logs, Logout, etc.). Each option has corresponding information or UX inputs that make sense per option. For settings, there are options for the user to customize their Fortifiot experience. Within activity, users can review all Fortifiot device activity, and choose what they want to do with the activity displayed. Query Logs delivers more details on the activity, and why some activity may have been deemed anomalous.

4.0 TESTING

4.1 Overview

This section explains the methods we went through to test the Fortifiot device. The following individuals should use this section:

- Developers
- Project Managers
- Team Members
- Network Administrators

4.2 Scope

The Scope of testing is to test the operation of the Fortifiot device. The tests were organized based upon the requirements of the device.

4.3 Objective

The objective of testing is to ensure the reliability, functionality, and usability of the device. These tests include testing each component of the device in isolation, as well as testing the all components in unison.

4.4 Entry and Exit Criteria

4.4.1 Entry Criteria

- Build complete
- Configuration tests complete
- Test environment is setup

4.4.2 Exit Criteria

- All tests have been run
- Bugs are documented and fixed

4.5 Logging Test and Reporting

If a feature is not working properly, a meeting will be held with all developers to determine how that feature should work. The feature will then be documented, investigated, and resolved after the developers meeting. All features required for device's proper operation will be tested for reliability.

4.6 System Testing

Fortifiot was tested as one whole application running in unison with all components. This helped to identify any bugs that only show when the device is running as a whole. It is intended to test the overall functionality and performance of the device.

4.7 Testing Procedure

The following steps must be taken prior to testing:

- Fortifiot device must be powered on and functioning normally
- Link back end and front-end interfaces (Web GUI with backend)
- IoT test device must be powered on and functioning normally
- Produce anomalous traffic for the IoT device

Below are tests that will be performed on the device:

1. **Stability Test -** This test will ensure stability of the device over a period of time. This device is intended to act autonomously so it will need to be available for days, even weeks at a time without user input. The device will be left on for at least one week, running all necessary components to ensure availability and stability. Power outages may affect this test.
2. **Communication Test -** Communication must be made with all components of the Snort Stack. This is will be completed through back end configurations, specifically ensuring that data is being logged to the database and sent to the user interface and vice versa.
3. **User Interface Test -** Basic UI/UX functionality will be examined during this test. Desktop and mobile views will be taken into consideration when testing UI.

4. **Functionality Test -** Ensure backend components are functioning properly with the UI. Specifically, make sure scripts are running to update the whitelist and blacklist components of snort based on the user's input from the web application.

4.8 Pass/Fail Conditions

It will be expected that the Fortifiot device must pass all tests in every category to be successful. If it does not pass, then the tester will document the bug and report it to the developers.

4.9 Risks

The following will impact the test cycle:

- Availability of the device
- Whitelisted traffic changing throughout the testing process
- Inability to Identify false positives

4.10 Problems Encountered and Solution Analysis

With almost every component that was installed, we encountered problems. Configuring these different components to run in unison proved to be a challenge. We had to manually search for individual packages to be downloaded and copy them to the device and install them to the right directories. We then had to run multiple configuration tests to ensure that all components were running correctly. To solve all the dependency and configuration issues we had to

investigate why certain components were not working, figure out which dependencies were needed, and finally install those dependencies.

4.11 Testing Results

Component	Date	Expected Outcome	Actual Outcome	Pass/Fail	Reason for failure
Snort	2/7/19	Snort runs and analyzes packets on network	Snort ran and analyzed packets on network	Pass	N/A
MySQL	2/7/19	Receiving logs from snort alerts	Received and stored logs	Pass	N/A
Barnyard2	2/7/19	Parsing logs from Snort and logging to database	Parsed logs and logged to database	Pass	N/A
Pulledpork	2/7/19	Grabs rule updates for Snort	Grabbed most recent updates for Snort	Pass	N/A
Apache	2/7/19	Hosting website	Website hosted	Pass	N/A
Minnowboard	2/7/19	Running Ubuntu and properly running all components	Running (Uptime for 4 days)	Pass	N/A

5.0 DISCUSSION

5.1 Future Recommendations

The biggest setback we had starting this project was finding a single device that would have enough resources to run all of our applications. Initially we had all our software running off a Hyper-V virtual machine, which we then ported to the MinnowBoard early in the second semester. During this process we ended up re-doing some things we had done last semester which was not ideal. Had we done this again, we would have bought the device first, then installed the software so that we did not have to port from a virtual machine to a physical device. A couple features we wanted to add but could not due to time constraints were adding a threat score and recommended action to each alert on the dashboard. This would have guided the user towards making the right decision whether that be blacklist, or whitelist and ultimately help make the dashboard more user friendly. We also wanted to add a settings page that would allow the user to add or remove devices, and setup their home network subnet so Snort knows which IP addresses it is protecting. Furthermore, we desperately wanted to push alerts to mobile devices so the user could be more connected and react to alerts in the least amount of time possible, but it was not in scope due to time constraints and our limited knowledge of SMS and push notifications. Some suggestions that we got from others was to have a map with geolocated points via the external IP's your devices are visiting. This would allow the user to visually see where their devices are going. Another suggestion we received was to have our device block all traffic during certain times, like when you are away from home or not using the device. This

would ensure that your devices are not getting out the internet while you are away and not able to respond to the alerts in a timely manner.

5.2 Design Objectives

Fortifiot was designed from the ground up to be user friendly and effective. By placing all of the needed parts on one device, we have made Fortifiot a one stop shop for securing your IoT devices, and the data they transmit. Initially, we wanted to have the device automatically scan the network for new devices, add any new devices into the dashboard, and pull their respected whitelists from a database. To do this we would have needed much more time and manpower to generate the whitelists and create the database. Fortifiot's initial use will serve as a proof of concept that this technology can work effectively at securing IoT devices. Ultimately, this product could serve as an IoT security sensor for both home and enterprise networks.

5.3 Technical Elements

The future of Fortifiot lies within our physical plug and play device rather than the application. The device must be able to determine the baseline behavior of the IoT device and then detect if abnormal behavior is taking place. This is the core of our product. Figure 2: Use Case Diagram shows the relationship between the device and the web application. The process essentially starts outside of the local network, then goes to the physical device, the web application, and finally ending with notifying the user of the anomaly and a recommended course of action.

5.4 Network

Fortifiot is hosted on a web server built on a quad-core MinnowBoard featuring dual NICs. The hardware also serves as the device that will be analyzing network traffic with Snort. Since everything is hosted on the device, it connects to your local network, and is available locally. The web app is accessible via a browser at fortifiot.localhost. This is where the user will go to interact with the device which passes commands from the web application to Snort to configure your IoT devices network activity.

6.0 CONCLUSION

6.1 Spring Semester 2019

The expected obstacles that came with configuring the Fortifiot device to work were very real for the Spring semester. With just over three months between the start of the semester and the IT Tech Expo we worked to get a functioning, demo ready device which accurately represents our ideas for the Fortifiot project.

Fortifiot is a plug and play IoT security management device. Once connected to your network, the Fortifiot device determines baseline behavior of all IoT devices on a network and with this, alerts the user on any deviations from this. The motivation behind the project was to give users insight into their IoT devices and to give them full control of them at the same time, aiming at providing them “piece of mind”.

Since the Fall semester we accomplished many tasks required to get Fortifiot running. The first was finding the right hardware for our device. A simple Raspberry Pi did not contain the resources required to run Snort and the entire LAMP stack. We eventually found a MinnowBoard Turbot Quad Core Dual NIC single board computer which gave us everything we needed. After this, we installed Ubuntu Linux with Snort and began configuring both. Once this

was completed, it became a struggle to determine where to place the Fortifiot device on the network to capture traffic correctly. This became so much of a struggle that we even considered turning the Fortifiot device into a router to grab the correct internal and external IP addresses instead of just the internal IP of a device going to the router's gateway address. Once this was working, we moved to the backend implementing the LAMP stack and interconnecting Snort with our already integrated website which houses the Fortifiot dashboard. This was a complicated process with a main goal of detecting alerts via Snort, sending those updates to the database, website, and finally back to the database based on the user action of the alert.

The abilities and skills we developed was primarily how to set up software on a headless version of Linux. Most of the time as students, we are fortunate to have pre-configured software on a graphical user interface when doing assignments. However, in this case we had to build it all from bare bones and do it on a headless version. Overall, Linux navigation became second nature, networking is now a clearer subject to us all, as is overall software development, specifically when attempting to get multiple pieces of software to communicate.

The Tech Expo was a new experience for all of us for many reasons. Not only was presenting an idea outside of the classroom one of those, but convincing others outside of the classroom the need for that idea. We also learned that when sharing those ideas with others, that they may see outside variables we never thought of because they sometimes interpret the idea totally different than expected. We learned very quickly that once you test a technical component in a new environment, it sometimes doesn't work, which required us to spend the first few hours of the morning configuring the device to the University of Cincinnati's network. Overall, the Tech Expo was an extremely rewarding experience and we were glad to have shared our idea with the community and receive feedback from experts in the industry.

APPENDIX A. REFERENCES

"Cisco Global Cloud Index: Forecast and Methodology, 2016–2021 White Paper." *Cisco*, 20 Nov. 2018, www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html.

"Gartner Says 6.4 Billion Connected." *Gartner IT Glossary*, Gartner, Inc., www.gartner.com/newsroom/id/3165317 https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html.

Paul, Fredric. "People Are Really Worried about IoT Data Privacy and Security." *Network World*, Network World, 28 Mar. 2018, www.networkworld.com/article/3267065/internet-of-things/people-are-really-worried-about-iot-data-privacy-and-securityand-they-should-be.html.

29, 2018 Help Net SecurityOctober. "IoT Users Uncertain If Personal Data Is Shared across Multiple Devices." *Help Net Security*, 2 Nov. 2018, www.helpnetsecurity.com/2018/10/29/iot-personal-data-sharing/.

Final Poster

Michael Schroeder
Stephan Manahan
Chris Lyons

Advisor: Tyler Hopperton

FortifIoT

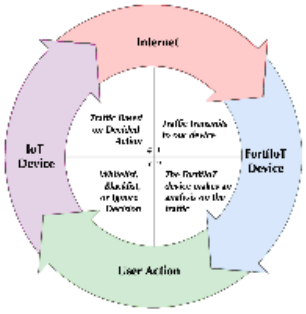
An IoT Security Solution

University of CINCINNATI

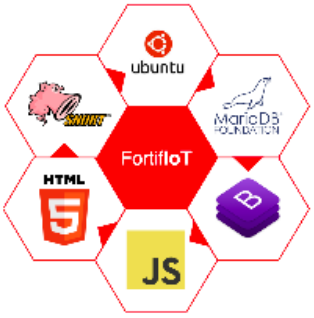
Abstract

IoT devices can be extremely useful for home and enterprise networks. However, the security of these devices has been overlooked, leaving them vulnerable to cyber-attacks. By leveraging an IDS and comparing traffic to and from your IoT devices, users can be alerted and act upon their devices' suspicious activity via the FortifIoT web console.

How It Works

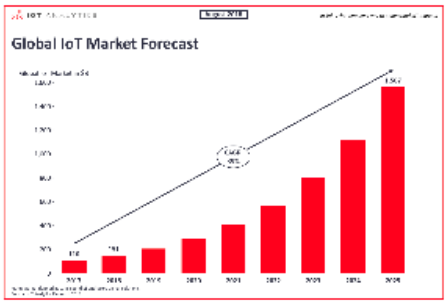


Technologies



Problem

- IoT devices are notoriously insecure
- No simple way to monitor devices
- Number of IoT devices will grow rapidly
- As the number of IoT devices increase, security will become even more important



Solution

- Easy to use management interface
- One Plug and Play device
- Intrusion detection system plus added benefits of user action
- Predetermined whitelist for comparison of traffic

University of Cincinnati | College of Education, Criminal Justice & Human Services | School of Information Technology