NetLock

by

Austin Conn and Andrew Hooper

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

*Austin Conn*_____	4/22/2020
Austin Conn	Date
*Andrew Hooper*_____	4/22/2020
Andrew Hooper	Date
*Ryan Moore*_____	4/22/2020
Ryan Moore - Faculty Advisor	Date

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

November 2019

NetLock

Prepared by:

Austin Conn and Andrew Hooper

Students of
University of Cincinnati

College of Education, Criminal Justice and Human Services

School of Information Technology

Table of Contents

List of Figures and Tables

Abstract

Five out of every six home router devices are inadequately updated, leaving sensitive data and connected devices unprotected from known vulnerabilities. The American Consumer Institute Center for Citizen Research released this information after the FBI sent out a warning in the 2018, stating hackers from Russia had successfully compromised hundreds of thousands of routers and were collecting user information. Users treat routers as plug and play devices where, once installed, they are left sitting with no further configurations made. However, blame does not fall solely on the user. ISPs and manufacturers fail to educate their customers of the proper steps to take after installation. NetLock was created with the user in mind, to act as a guide and walkthrough after router installation. Once connected to the router and opened via desktop application, NetLock will assist the user with proper setting configuration while providing network security best practices throughout the setup

1 Problem Statement

1.1 Introduction

With the IoT and smart home technologies on the rise, a secure router is more critical than ever, and the typical user is often daunted by how technically challenging it is to properly configure their home router. With router attacks growing it is becoming increasingly important to securely configure the home network, and the extent of their router configuration is creating a Wi-Fi password. Most users will never visit the overly complicated admin dashboard let alone verify they have proper security settings in place.

1.2 Problem

According to Broadband Genie, 51 percent of people never update their router's firmware, change the default admin account password, or even change their Wi-Fi password. Routers running factory credentials and configurations out of the box make easy targets for attackers. Yet, the user is not solely at fault for leaving their routers vulnerable; device manufacturers do a poor job of designing their software to be intuitive and user friendly. Without an easy to use user interface and an extensive amount of knowledge, many users are open to attacks that can compromise their networks.

1.3 Solution

Our solution is a simple application that is downloaded onto a computer in the network to connect to a router that has factory default settings. Our application will be able to identify the manufacturer and model, check for existing firmware updates from the manufacturer, as well as configure some basic network security settings on the router. Users can connect to the application using their computer to select specific configuration parameters in an intuitive

interface, all without having to have a degree in networking to understand what the various options do. The interface will offer intuitive, step by step instructions that guide the user through their router configuration. Throughout the configuration, best practice recommendations and security tips will be offered to the user so that they can make an informed decision when it comes to setting up their device

1.4 Project Goals

Netlock is developed with the average user in mind. It strives to be an application which provides consumers the ability to make informed decisions about how to secure their home network. NetLock will allow consumers to secure their router based on industry best-practices, add additional functionality to their router, and keep their router up to date.

1.5 Overview

The remainder of this report outlines how the project was developed. The following report will include: project concept, design objectives, methodology/technical approach, user profile, use case diagram, technical architecture, testing, budget, timeline, problems encountered, future recommendations and lessons learned.

2 Discussion

2.1 Project Concept

NetLock was inspired by Austin Conn and Andrew Hooper. Austin proposed the idea for a device that a consumer could purchase and plug in to their newly purchased router, and through an easy to understand interface, would allow non-technical consumers the ability to configure their routers without needing to have a degree in networking. Austin proposed the idea to Andrew, and we agreed that this device could be even more than just a configuration tool. Instead, it could also provide added functionality not found in most consumer routers.

2.2 Design Objectives

To develop a cross-platform web application, accessible by desktop computer, that will provide consumers the ability to make informed decisions to secure their home network, all without needing to have a background in networking. NetLock will allow consumers to secure their router based on industry best-practices, add additional functionality to their router, and keep the consumer informed. NetLock will be a desktop application that has a visually appealing appearance and can be administered by an intuitive user interface. The software will be based on a Electron, while the application will be supported using Python, HTML, JavaScript, and MySQL.

The features NetLock will include are as follows:

- NetLock will be easy to install, requiring the user to download the application. Once the application is booted up, the user will connect to the device's network by plugging into the router. Once connected, the user can begin using the NetLock interface to begin configuring their router.

- NetLock's user interface will offer advice on security best practices for each setting, letting the user make an informed decision
- NetLock will allow the user to:
    - Configure the router's WiFi settings using industry best practice and instructing the user to set a secure password
        - Change default username and password
        - Change SSID or hide network
        - Strengthen Encryption (WPA2)
        - Turn down broadcast power
        - Disable WPS
    - Learn about router and network security during each phase of configuration
        - Each page gives the user industry best practice advice and there is a router FAQ feature that allows users to learn a little more

2.2.1 Potential Ideas

Our team is small and our initial vision for this project was quite large. We started the development thinking we would use a Raspberry Pi, but have since decided to use a desktop application. We have had to scale back our ambition to make sure we met the core ideals of our project, which is configuring the router itself. We wanted to include but have since put on hold the following:

- Configure the router to allow NetLock to act as a Virtual Private Network (VPN) server for the user to use when outside of the network
- Configure the router and NetLock to work as an Intrusion Detection System (IDS)

        - Should this option be chosen, the user can opt to receive a push notification when an intrusion is detected
    - Allow the user to connect a hard drive to the NetLock device and share it across the network
    - Open or close ports on the router to increase security
    - Configure the router to allow NetLock to act as the Domain Name System (DNS) server for the network
        - Should this option be chosen, the user can opt for additional features such as network wide adblocking, domain blocking, or parental controls

2.2.2 Problems Encountered

As we switched from a hardware solution to a software solution our original frontend had to be scrapped. While this was frustrating at first, we quickly realized that with the things we learned from the previous semester and a growing understanding of our own product that we could improve upon the previous build. The client side of our app now is much more fluid and easily scales to whatever screen size the user chooses to use. Our previous build also had no loading screen and we soon found that the backend scripts took a little time to connect to the router. Some features require the router to validate the input before returning control back to the interface. Some calls to the router were able to be waited on until a specific element was editable or valid again, but some needed to wait a preset amount of time before moving on. This was especially true with setting the WPS settings. After a few failed attempts we created a simple loading screen that is pleasant to look at and follows the same color scheme as the landing page which the use would see next. While the backend is developed in Python, it now

must communicate with Electron instead of the Raspberry Pi OS. To accomplish this implementation, we had to utilize zerorpc. The client object then communicates back and forth between Electron and Python.

2.2.3 Future Recommendations

Given more time we would have liked to add support for both Windows and Linux machines. As it is now NetLock is exclusively a Mac application. We had to drop some features throughout development and would have liked to be able to include everything we originally thought possible, but as a two-man group we found that we needed to focus more on completing a polished product.

2.3 User Profile

2.3.1 Potential Users

- Technical and non-technical users alike.
    - Fortune 500 network administrators
    - Grandparents
    - Parents
    - Students

2.3.2 Software, Interface, and Related Experience

- NetLock users need to only have experience with desktop or laptop computers, nothing more

2.3.3 Experience with Similar Applications

- Trend Micro™ Home Network Security is a solution offered by Trend Micro that aims to control the network after it has been set up. While it does offer some of the same functionality of NetLock, it lacks the router configuration portion. Much of the functionality that is implemented through Trend Micro's solution is accomplished through monitoring the network and making suggestions to the user that the user is expected to correct.
- NetLock aims to make the configuration changes on the router for the user rather than expecting the user to make the change in the manufacturer's web interface.

2.3.4 Task Experience

- The user will connect the router to the network per the manufacturer's or internet service provider's instructions
- The user then installs the NetLock application
- The user uses their desktop or laptop to connect to the router then the NetLock application using the network name and password listed on the network device
- The user opens NetLock application and will be directed to the web application interface
- Depending on various options selected, the user will be walked through the entire configuration process which NetLock will configure the necessary settings on the router to secure it
- Once the initial configuration is complete, the user can either leave the NetLock active to review information or disconnect

2.3.5 Frequency of Use

- This device has two frequencies of use.
    - First, the device can be used once and then disconnected from the router. During the time the NetLock device is connected, it will walk the user through the interface to configure settings on the router that will make the router more secure
    - Second, the application can be left active to allow the user to view additional router information or make changes on the fly

2.3.6 Key Interface Design Requirements

- Simplification of technical terms for the average person
    - The application should be able to provide the definition of technical concepts and terms in a manner that an average user with no experience in networking could understand
- Visually appealing and easy to navigate interface
- Robust features and capabilities to ensure the user can configure a secure network

2.4 Use Case Diagram

Figure 1: Use Case Diagram displays the use case diagram, which depicts how the user interacts with our system and how our product reacts.

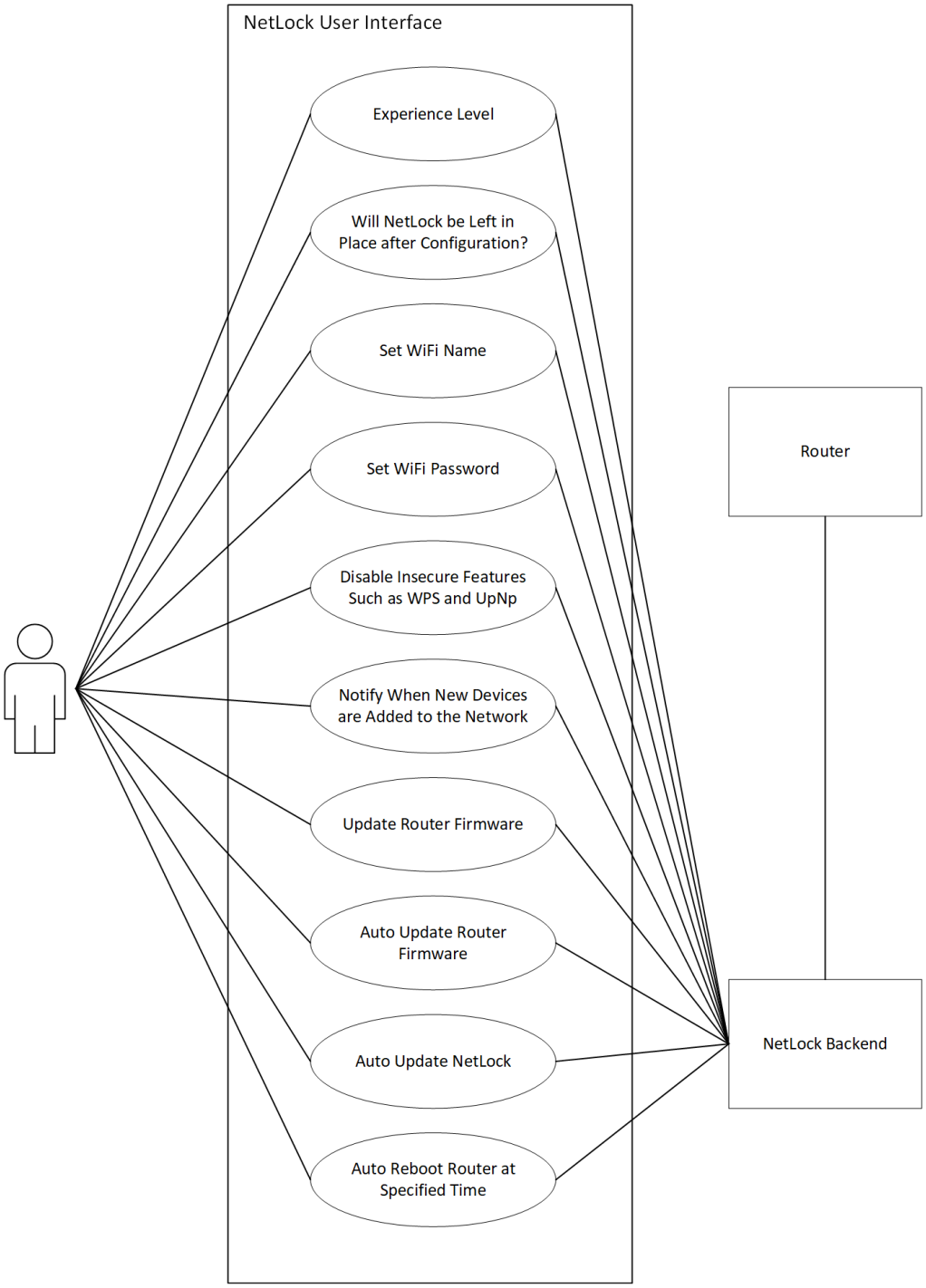


Figure 1: Use Case Diagram

3 Project Management

3.1 Budget

Table 1: Project Budget The table depicts the real-world cost of creating a product like NetLock.

Project: NetLock Title: Budget				
Ref	**Project Items**	**Unit/Hours**	**Unit Price**	**Total**
1	**Hardware**			
1.1	Routers	4	$ 75.00	$ 300.00
1.1	PC's (new or upgrades)	2	$ 1,300.00	$ 2,600.00
1.2	Cables	3	$ 15.00	$ 45.00
2	**Software**			
2.1	Operating systems	2	$ 100.00	$ 200.00
2.2	Software modifications	1	$ 100.00	$ 100.00
2.3	Additional security applications	1	$ 75.00	$ 75.00
3	**Network**			
3.1	Internet Access	1	$ 250.00	$ 250.00
4	**Project Team**			
4.1	Project Manager/Backend Dev	180	$ 60.00	$ 10,800.00
4.2	Front end Dev	180	$ 50.00	$ 9,000.00
4.3	Testers	20	$ 20.00	$ 400.00
4.4	Travel and Expenses	1	$ 200.00	$ 200.00
	Subtotal			$ 23,970.00

Table 1: Project Budget

3.2 Objectives/Deliverables

Table 2: Project Objectives/Deliverables. This table displays all due dates associated with the capstone course, as well as project deliverables assigned by the project manager.

Task # and Task	Duration	Start Date	End Date
1 Other	167 days	Mon 8/26/19	Tue 4/14/20
1.1 Team Building	167 days	Mon 8/26/19	Tue 4/14/20
1.2 Idea Planning	6 days	Mon 8/26/19	Mon 9/2/19
1.3 Tech Expo	1 day	Tue 4/14/20	Tue 4/14/20
1.3.1 Prepare for Tech Expo	1 day	Mon 4/13/20	Mon 4/13/20

2 Deliverables (Assignments)	178 days	Mon 8/26/19	Wed 4/29/20
2.1 IT5001 Assignment 0: Team Members & Project Name	6 days	Mon 8/26/19	Mon 9/2/19
2.1.1 Project Name	6 days	Mon 8/26/19	Mon 9/2/19
2.1.2 Problem Statement and Solution	6 days	Mon 8/26/19	Mon 9/2/19
2.2 IT5001 Assignment 1: Contract	21 days	Mon 8/26/19	Mon 9/23/19
2.2.1 Project Approval	21 days	Mon 8/26/19	Mon 9/23/19
2.2.1.1 Gantt Chart	21 days	Mon 8/26/19	Mon 9/23/19
2.2.1.2 Work Breakdown Structure	21 days	Mon 8/26/19	Mon 9/23/19
2.2.1.3 Logo	21 days	Mon 8/26/19	Mon 9/23/19
2.2.1.4 Contract Drafting	21 days	Mon 8/26/19	Mon 9/23/19
2.3 IT5001 Assignment 2: Project Abstract for Tech Expo	36 days	Mon 8/26/19	Mon 10/14/19
2.4 IT5001 Assignment 3: Team Contract Resubmission	15 days	Tue 9/24/19	Mon 10/14/19
2.4.1 Integrate Feedback from Prof. Scott	5 days	Tue 9/24/19	Mon 9/30/19
2.4.2 Resubmit Contract	10 days	Tue 10/1/19	Mon 10/14/19
2.5 IT5001 In Class Assignment: 3-Minute Elevator Speech	41 days	Mon 8/26/19	Mon 10/21/19
2.6 IT5001 Assignment 4: User Profile	1 day	Mon 10/21/19	Mon 10/21/19
2.7 IT5001 Assignment 5: Use Case Diagram	1 day	Mon 10/21/19	Mon 10/21/19
2.8 IT5001 Assignment 6: Draft Report	51 days	Mon 8/26/19	Mon 11/4/19
2.9 IT5001 Assignment 7: Final Fall Semester Report	15 days	Tue 11/5/19	Mon 11/25/19
2.10 IT5001 In Class Assignment: Final Fall Presentation	66 days	Mon 8/26/19	Mon 11/25/19
2.10.1 Create Presentation	66 days	Mon 8/26/19	Mon 11/25/19
2.10.2 Practice Presentation	66 days	Mon 8/26/19	Mon 11/25/19
2.11 IT5002 Assignment 1: Testing Plan and Report	21 days	Mon 1/13/20	Mon 2/10/20
2.12 IT5002 Assignment 2: Final Abstract	91 days	Mon 10/14/19	Mon 2/17/20
2.13 IT5002 Assignment 3: Tech Expo Poster Draft	36 days	Mon 1/13/20	Mon 3/2/20
2.14 IT5002 Assignment 4: Tech Expo Poster Final	6 days	Mon 3/2/20	Mon 3/9/20
2.15 IT5002 Assignment 5: Final Report	41 days	Mon 2/10/20	Mon 4/6/20
2.16 IT5002 Assignment 6: Final Report Safe Assign	41 days	Mon 2/10/20	Mon 4/6/20
2.17 IT5002 Assignment 7: Final Library Copy Of Project	11 days	Mon 4/6/20	Mon 4/20/20
3 Project Analysis	171 days	Mon 8/26/19	Mon 4/20/20
3.1 Research	41 days	Mon 8/26/19	Mon 10/21/19
3.1.1 Hardware Requirements	5 days	Mon 8/26/19	Fri 8/30/19
3.1.1.1 Verify Raspberry Pi Version	1 day	Mon 8/26/19	Mon 8/26/19
3.1.1.2 Purchase Raspberry Pi	4 days	Tue 8/27/19	Fri 8/30/19
3.1.1.3 Router Research	5 days	Mon 8/26/19	Fri 8/30/19
3.1.1.3.1 Purchase Routers	4 days	Tue 8/27/19	Fri 8/30/19
3.1.1.3.2 Create List of Routers to Support	1 day	Mon 8/26/19	Mon 8/26/19
3.1.2 Software Requirements	8 days	Mon 9/2/19	Wed 9/11/19
3.1.2.1 Operating System	2 days	Mon 9/2/19	Tue 9/3/19
3.1.2.2 Security Programs to Include	3 days	Wed 9/4/19	Fri 9/6/19
3.1.2.3 Establish Programming Language and Framework	3 days	Wed 9/4/19	Fri 9/6/19

3.1.2.4 Web App Requirements	6 days	Wed 9/4/19	Wed 9/11/19
3.1.2.4.1 Determine Languages and Framework	3 days	Wed 9/4/19	Fri 9/6/19
3.1.2.4.2 Hosting Requirements	3 days	Mon 9/9/19	Wed 9/11/19
3.1.2.4.3 DB Research	3 days	Mon 9/9/19	Wed 9/11/19
3.1.3 Other Research	14 days	Thu 9/12/19	Tue 10/1/19
3.1.3.1 Budgetary	14 days	Thu 9/12/19	Tue 10/1/19
3.1.4 Design	41 days	Mon 8/26/19	Mon 10/21/19
3.1.4.1 Design Use Cases	41 days	Mon 8/26/19	Mon 10/21/19
3.1.4.2 User Profile	41 days	Mon 8/26/19	Mon 10/21/19
3.1.4.3 Create Wireframes	41 days	Mon 8/26/19	Mon 10/21/19
4 Project Build	141 days	Mon 8/26/19	Mon 3/9/20
4.1 Set-up Environment	21 days	Mon 8/26/19	Mon 9/23/19
4.1.1 Install Operating System	2 days	Wed 9/11/19	Thu 9/12/19
4.1.2 Install Appropriate Applications on Raspberry Pi	7 days	Fri 9/13/19	Mon 9/23/19
4.1.2.1 Python	7 days	Fri 9/13/19	Mon 9/23/19
4.1.2.2 Selenium	7 days	Fri 9/13/19	Mon 9/23/19
4.1.3 Setup Git Repo	7 days	Fri 9/13/19	Mon 9/23/19
4.1.4 Physical Hardware Set-up	12 days	Mon 8/26/19	Tue 9/10/19
4.1.4.1 Raspberry Pi	7 days	Mon 9/2/19	Tue 9/10/19
4.2 Back-End Dev	77 days	Mon 8/26/19	Tue 12/10/19
4.2.1 Set-up Structure (Selenium w/ Python)	14 days	Tue 9/24/19	Fri 10/11/19
4.2.2 Create Models for Router Pages	14 days	Mon 10/14/19	Thu 10/31/19
4.2.3 Create Connections to DB	14 days	Fri 11/1/19	Wed 11/20/19
4.2.4 Store Router Data	14 days	Thu 11/21/19	Tue 12/10/19
4.3 Front-End Dev	119 days	Mon 8/26/19	Thu 2/6/20
4.3.1 Create Mock UI	14 days	Wed 12/11/19	Mon 12/30/19
4.3.2 Determine Style Libraries	14 days	Tue 12/31/19	Fri 1/17/20
4.3.3 Create Landing Page	14 days	Mon 1/20/20	Thu 2/6/20
4.4 Testing	140 days	Mon 8/26/19	Fri 3/6/20
4.4.1 Testing of Supported Router	21 days	Fri 2/7/20	Fri 3/6/20
4.4.2 Testing of Front-End Interfaces	21 days	Fri 2/7/20	Fri 3/6/20
4.5 Deployment	1 day	Mon 3/9/20	Mon 3/9/20

Table 2: Project Objectives/Deliverables

3.3 Project Schedule

Figure 2: Gantt Chart shows our project milestones with timeline.

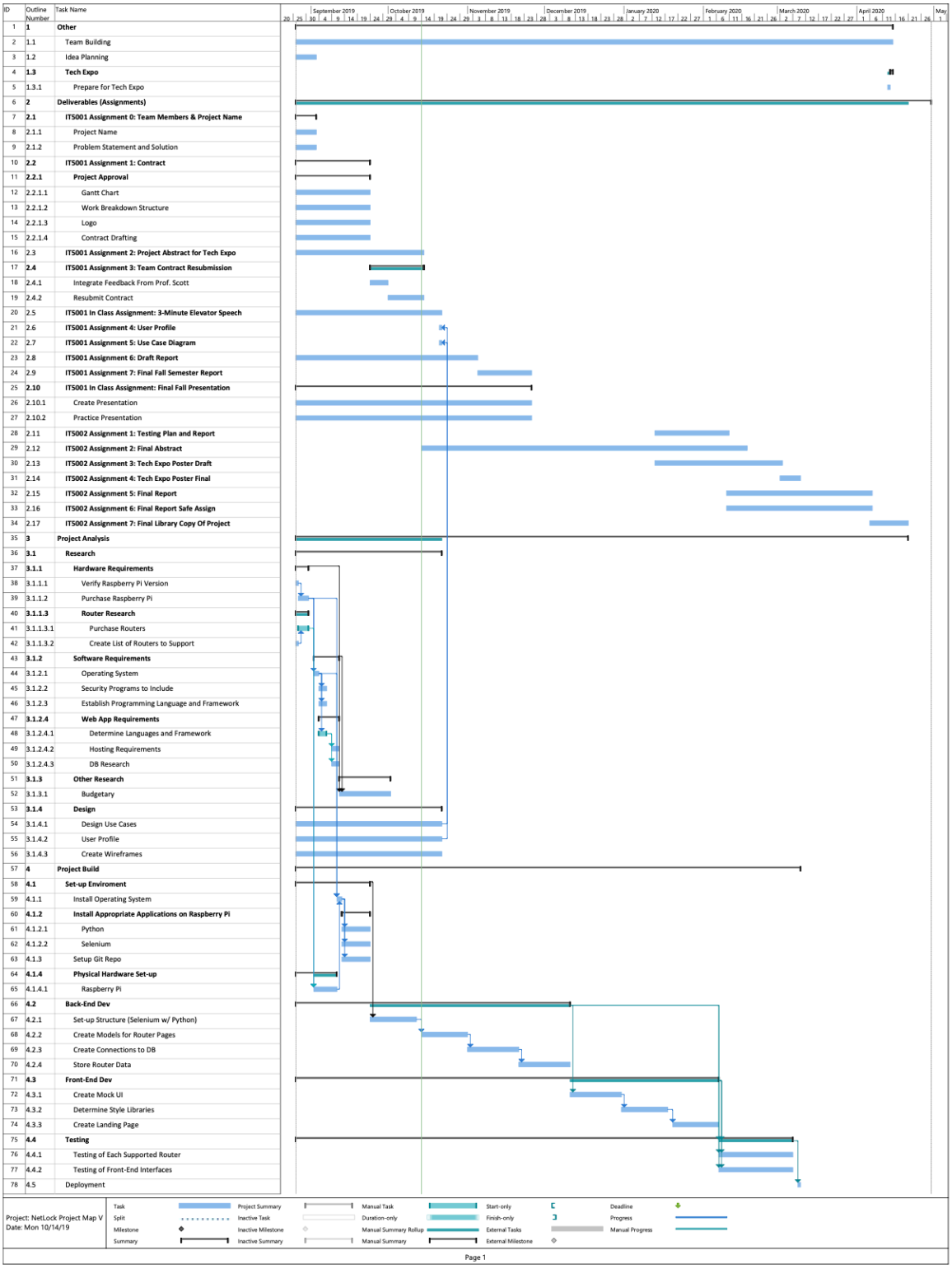


Figure 2: Gnatt Chart

4 Technical Elements

4.1 Network

NetLock is designed to be used in the home, so the user will already have an existing home network that the device will be connected to. Since the user will already have a router, most likely given to them from their internet service provider, NetLock will act as a desktop application to automatically scan and assist the user in set-up. This way our application will be able to communicate with the router and configure its security settings.

4.2 Application Interface

The application itself needs to be user friendly and modular. There are many frameworks available that each have their own strengths and weaknesses and it can be a daunting decision which to choose when starting a project, but its important to choose based on the needs of the project. Bootstrap is the most popular front-end framework available today and because of its great popularity and giant community there is a wide variety of options and assistance. For this reason, we decided to use Bootstrap as our framework for NetLock's front-end. It is important that the user has the option to use any device they wish and because of Bootstrap's grid system it becomes relatively simple to automatically scale based on screen size.

The user interface is being designed to be as simple as possible, acting as a step by step guide that the user can easily follow. Because of this there are many pages involved and a lot of text that include best practice recommendations. It is proving a challenge to spread out information this way when there are many options involved in network configuration. Our team went through several discussions involving wireframes and architectures before we managed to start

building the interface.

The application is launched via Electron and is able to communicate to the python backend directly from the desktop application.

4.3 Security

As a desktop application, NetLock is only communicating with one HTTPS web address. By implementing secure coding practices, we can defend against the most common types of web vulnerabilities.

4.4 Technical Architecture

Figure 3: Technical Architecture illustrates the technology used to power NetLock.

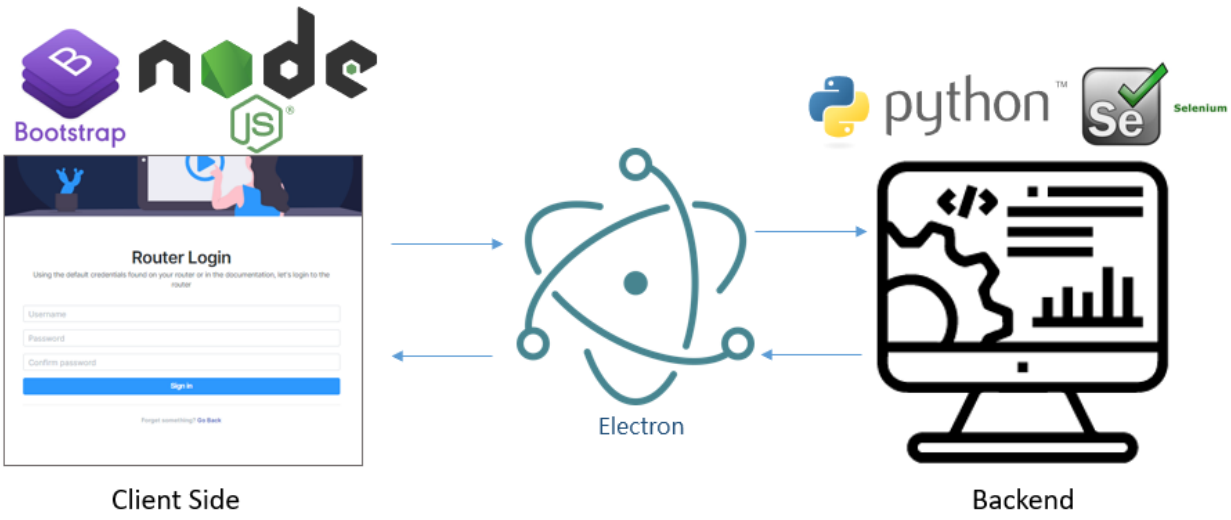


Figure 3: Technical Architecture

4.5 Test Plan

This section lays out the testing that was performed throughout NetLock's development. Reference this section to review the methodology and practices used while testing the application for functionality, usability and aesthetics.

4.5.1 Methodology

Our team is small and our vision for this project has shifted as the course has gone on. We have gone from Web-Application to Desktop-Application and have had to become flexible to make sure NetLock delivers what it aims to. Because of this we have decided to leverage an agile approach to testing. This methodology allows to us to be more malleable and make critical repairs during development. Our focus with this methodology will be on unit testing and making sure that our code functions as it should. NetLock runs on Electron and this platform will verify whether the application will integrate properly, leaving the bulk of the testing to be done at the lowest level.

4.5.2 Scope

The scope of our testing will confirm connection and manipulation of desired router settings in an easy to use manner. This will verify that our application is able to communicate with the router in question, while making the necessary changes displayed by the user interface.

4.5.3 Objectives/Scenarios

Tests to be completed:

- Compilation
    - NetLock is primarily an Electron application with a python backend. As such, we must ensure that both pieces of the application can be compiled for cross platform compatibility.
    - Ensure that NetLock runs on each operating system without requiring any dependencies to be installed by the user
- Startup

- Ensure that NetLock launches successfully on each operating system
- Ensure that all required binaries (compiled python, selenium library, etc.) are included and properly referenced in the compiled version

- Router Connection
  - Identify the router on the current network
  - Once the router is identified, verify whether a connection to the targeted router can be made
  - Once the connection to the target router has been established, can NetLock properly authenticate with the router to make the necessary configuration changes.
- Setting Manipulation/Functionality
  - Each option offered by the User Interface appropriately makes changes to the router
  - Should the router not have built in functionality to check for firmware updates, NetLock can appropriately download the update from the manufacturers website and apply the update without requiring intervention from the user

4.5.4 Testing Tables

Table 3: Client-Side Testing shows the testing done on what the user sees during NetLock's configuration.

Project:	NetLock
Testing:	Client Side

Tester	Test Title	Test Summary	Expected Result	Actual Result	Status	Notes
Dev	Application Launch	From the command line launch the application	Electron will launch the index page	Index page opens	pass	N/A
Dev	Scalability	Application should scale depending on screen size	Icons/imgs will scale correctly	Icons/imgs scale correctly	pass	N/A
Dev	Button Functionality	Test button response when screen is scaled down	Responsive buttons	Buttons respond	pass	N/A
Dev	Menu Functionality	Test collapsible menu in different sizes	Menu opens and closes	Menu behaves	pass	N/A
Dev	Checkbox Functionality	Checkboxes check correctly	One check box should check at a time	Check boxes work	pass	Failed in the previous builds
Dev	Link Functionality	Go back and number links should take you to correct pages	Links behave as they should	Links do as they should	pass	N/A

Table 3: Client-Side Testing

Table 4: Backend Testing describes how NetLock's backend was tested.

Project:	NetLock
Testing:	Backend

Tester	Test Title	Test Summary	Expected Result	Actual Result	Status	Notes
Dev	Router Detection	Detect and connect to target router	Establish connection to router	Connected to router	Pass	N/A
Dev	Router UI	Login to the router's web interface	Python scripts should log into router's web interface	Scripts were able to login to the web interface	pass	N/A

Dev	Change Settings	Change settings within the router UI	Scripts should change necessary parameters	Scripts changed selected settings	pass	N/A
Dev	Integrate Python	Integrate python into Electron installer	Use rpc process to communicate between python and electron	Python piece and Electron are talking	pass	N/A
Dev	Accept front end calls	Backend scripts accept calls from front end	scripts will accept calls properly and perform necessary actions	Calls are accepted	pass	N/A

Table 4: Backend Testing

Table 5: User Experience Testing displays the information gathered from outside users.

	Project:	**NetLock**				
	Testing:	**User Experience**				
Tester	**Test Title**	**Test Summary**	**Expected Result**	**Actual Result**	**Status**	**Notes**
User	Launch App	User finds and selects the NetLock icon to start	NetLock will open to the landing page	User found icon and NetLock Launched	Pass	N/A
User	Navigate App	User clicks through the app until router configuration is complete	User should be able to follow the steps without issue	User navigated the application easily	pass	N/A

Table 5: User Experience Testing

4.6 User Interface Visuals

Figure 4: Landing Page is what the application looks like once launched.

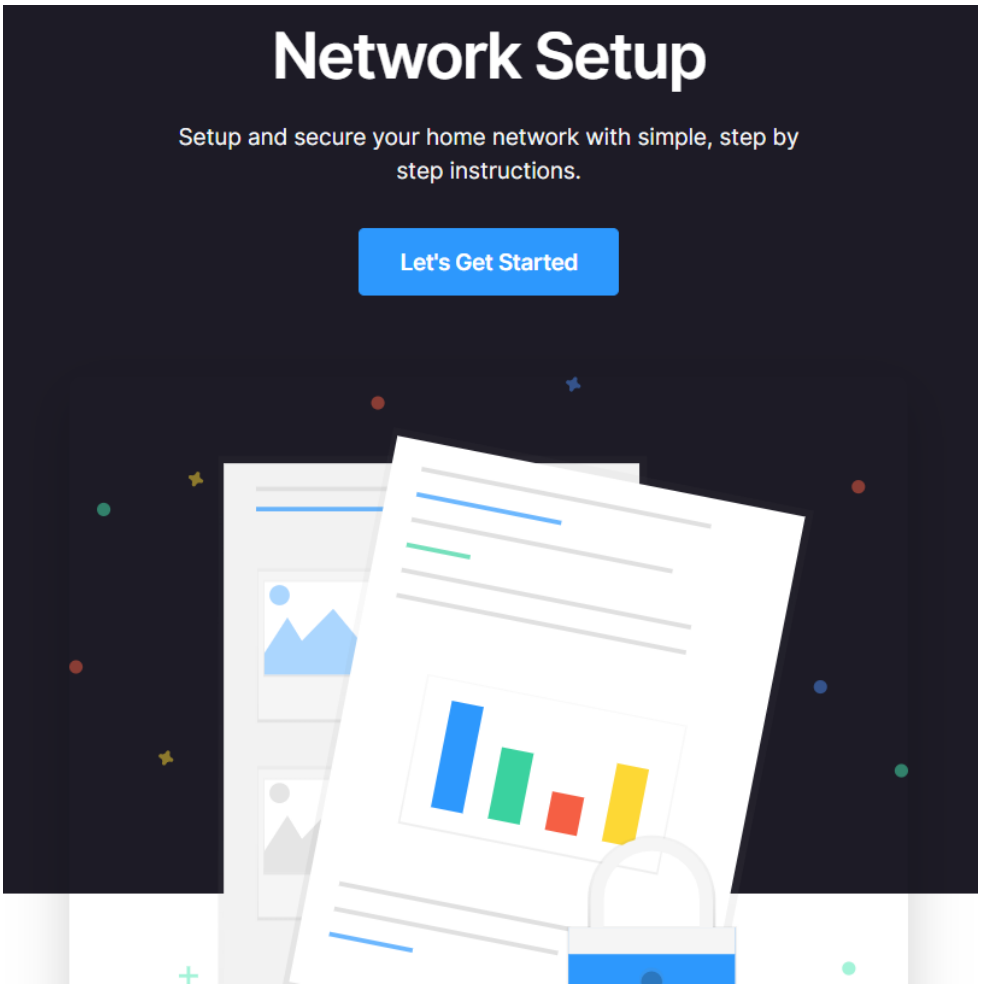
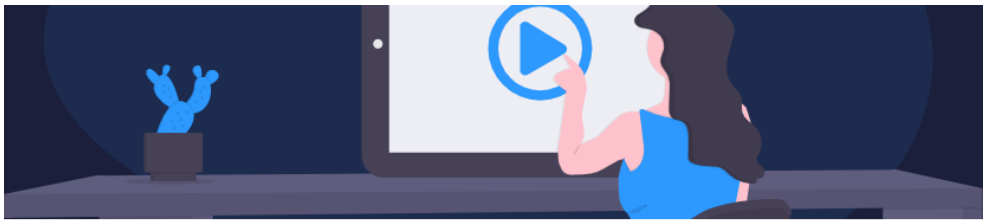


Figure 4: Landing Page

Figure 5: Login shows the next step after getting started, which is logging into the router's admin page.



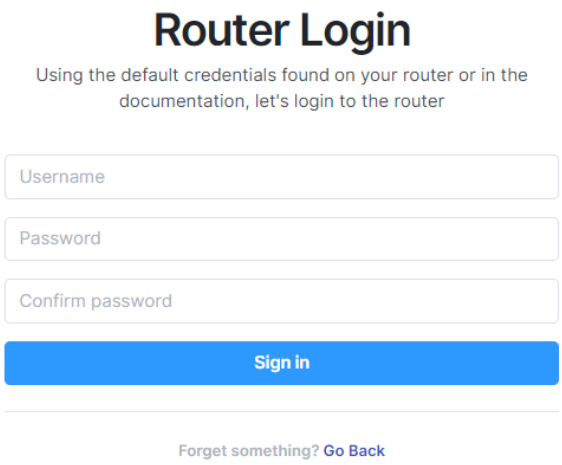


Figure 5: Login

Figure 6: Change Default Login shows the first step of the new router configuration. Also, it gives best practice recommendations.

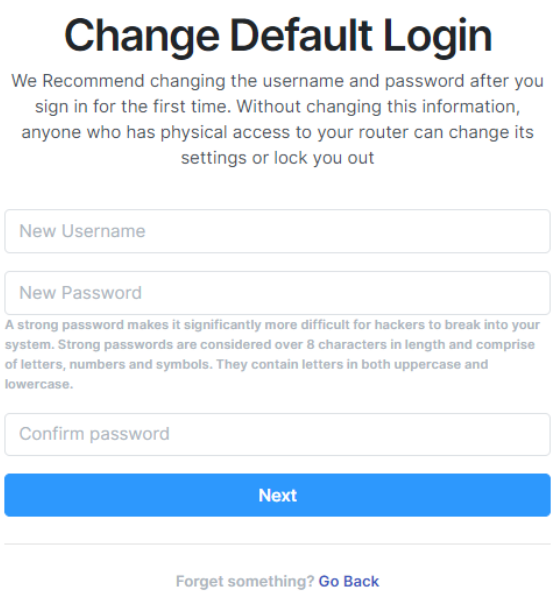


Figure 6: Change Default Login

Figure 7: FAQ shows the final thing the user will see after router configuration.

Frequently Asked Questions

What is a router?

What is the difference between Router and a Modem?

Why should I update my Firmware?

Commonly used abbreviations.

DHCP - Dynamic Host Configuration Protocol
DNS - Domain Name System
HTTP - HyperText Transfer Protocol
HTTPS - HyperText Transfer Protocol Secure
IDS - Intrusion Detection
IP - Internet Protocol
IPS - Intrusion Prevention System
ISP - Internet Service Provider
LAN - Local Area Network
NAT - Network Address Translation
SSID - Service Set Identifier (Wi-Fi)
WAN - Wide Area Network
VPN - Virtual Private Network
WPA - Wi-Fi Protected Access

Why change the default passwords?

Figure 7: FAQ

5 Conclusion

5.1 Fall Semester 2019

Throughout this semester we have done a lot of research about frameworks and potential designs. While our team has had some experience with front-end development there was still a learning curve with HTML, CSS and JavaScript. The initial idea was to add additional features to our system such as VPN and intrusion detection, but these have been put on hold while we focus on making sure our backend and front-end are able to communicate without error. Much of the focus has been on design, as our entire idea revolves around 'ease of use', and now that much of the ideas are solid, we can continue to build our product. Our initial project started as a hardware device meant to be purchased by end users and plugged in to their newly purchased router and be able to easily and effectively configure their new router to be more secure and optimized than the manufacturer's default settings would allow. However, after hearing several presentations in class, we felt that from a marketability standpoint, it didn't make sense to have to purchase a secondary piece of hardware after a consumer had just purchased a new router. Given this, we decided to pull back on our idea of a hardware-based device and opted to have a cross-platform desktop application that would enable a user to use our application, regardless of operating system, to configure these settings without needing to have a networking degree. Our initial project was to be built in two parts. The backend was to be built using python and the selenium (https://selenium.dev/) framework to facilitate the "behind the scenes" configuration of the router, and MySQL to house the router configuration and various settings. Our front end was to be built using HTML, Javascript, and PHP. However, with our decision to go to a desktop application has caused us to change some of our

technologies and thus, we have had to abandon some already built pieces of the project. Our new approach will still use python and selenium to facilitate the “behind the scenes” portion of the application. However, instead of using MySQL as our backend database, we will now be using sqlite instead of MySQL due to the smaller anticipated database size, as well as needing a cross-platform technology to house our data. Our existing front end has largely been able to be recycled and built using the electron framework (https://electronjs.org/).

5.2 Spring Semester 2020

Our team completed the project, though we had to drop more features than we initially would have liked. In order to ensure a more polished product we decided to focus on developing the application for Mac devices while dropping support for Linux and Windows. We had to reevaluate once we made the switch from a hardware solution to a software solution. Opting to work with Electron as a desktop application opposed to using a Raspberry Pi seemed prudent. The targeted user of this app wouldn’t also want to go out and buy a raspberry pi along with their router. This change gave us the chance to re design our frontend while the backend was being developed. Where before NetLock was primarily going to be a mobile application, it is now a full screen application. During the redesign the client side became more fluid and scalable. While the backend is developed in Python, it now must communicate with Electron instead of the Raspberry Pi OS. To accomplish this implementation, we had to utilize a zerorpc server in the python backend. The front end then becomes a zerorpc client that connects to the backend server to establish two-way communication.

Using this new architecture allowed us to be extremely flexible with how we chose to

provide information to the user. We were able to contain all of our business logic to the python backend, and only perform input validation in the front end. This should help with portability in the future if we decided to continue development of the project to expand it to the originally planned operating systems.

While we have only managed to support one router at this point in time, our framework is built such that the system is modular. By creating new python files for a specific router and implementing the same function names as our current router, NetLock can easily be extended to support a wide range of routers. Using this method, a potential router manufacturer does not necessarily need to use the selenium framework to emulate a user interacting with the web interface. For example, if the router supports configuration over SSH or telnet, that communication and configuration can happen within the python methods that NetLock calls on.

6 Appendix A. Poster

NetLock

NetLock

Team 14: Austin Conn and Andrew Hooper
CECH – School of Information Technology
Technical Advisor: Ryan Moore

University of CINCINNATI

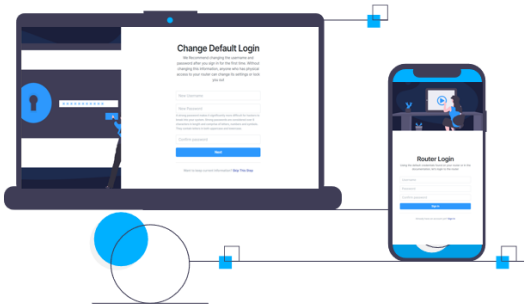
Problem

- Most home routers are inadequately updated or inappropriately configured, leaving sensitive data unprotected
- Users treat routers as Plug-and-Play devices where, once installed, they are left sitting with no further configurations made
- ISPs and manufacturers fail to educate their customers of the proper steps to take after installation

The Solution

NetLock was created with the user in mind, to act as a guide and walkthrough after router installation.

- Once connected to the router and opened via desktop application, NetLock will assist the user with proper setting configuration while providing network security best practices throughout the setup



Benefits

Security

Best Practice Recommendations

Ease of Use

Technical Elements

python

Selenium

Bootstrap

Electron

7 Appendix B. References

Armerding, T. (2018, October 9). Vulnerable routers are still out there-and hackers are noticing: Synopsys. Retrieved October 14, 2019, from https://www.synopsys.com/blogs/software-security/vulnerable-routers-hackers/.

Editor. (2018, September 26). New Study Warns of Inadequate Security Provisions in Home and Office Routers. Retrieved October 14, 2019, from https://www.theamericanconsumer.org/2018/09/new-study-warns-of-inadequate-security-provisions-in-home-and-office-routers/.

The American Consumer Institute Center for Citizen Research. (2018, September). Wi-Fi Router Vulnerabilities. Retrieved February 2020, from https://www.theamericanconsumer.org/wp-content/uploads/2018/09/FINAL-Wi-Fi-Router-Vulnerabilities.pdf

The Cybersecurity and Infrastructure Security Agency. (2018, April). Alert (TA18-106A) Russian State-Sponsored Cyber Actors Targeting Network Infrastructure Devices. Retrieved February 2020 from https://www.us-cert.gov/ncas/alerts/TA18-106A