Budget Network Analysis Suite: Challenges and Solutions for the Deployment of a Raspberry Pi as a Network Traffic Analysis Appliance

By

Ryan Patrick

B.S. Information Security & Intelligence

Ferris State University, 2016

Advisor:

Dr. Greg Gogolin

Full Professor

Information Security & Intelligence Department

Winter 2020

Ferris State University

Big Rapids, MI

Dedication

This paper is dedicated to those of us who are collectively known as “non-traditional students.” Whether it be due to familial obligations or financial limitations, it is this group of students who perhaps have the most difficult times circumnavigating the waters of academia. While University life certainly favors the young, it is important for you to not feel discouraged. In fact, you will find that the wisdom and work ethic which you have accumulated in your advanced years provide benefits that far outstrip anything you feel may have been lost in delaying your academic pursuits.

My advice to you is twofold: do not stop networking and support your peers as much as possible. In my two years on campus at Ferris State University, I made strong connections with everyone from faculty members to incoming freshmen. Not only did I strive to absorb any guidance bestowed upon me by anyone willing to offer support, but I was able to draw upon my previous life experience in both mentoring other students and assisting various student organizations.

While your time spent “in exile” may seem like a burden at the time, I can almost guarantee that you will look fondly upon your time at Ferris if you put forth a concerted effort to leave the University better than when you found it.

Acknowledgements

I would to thank anyone and everyone that assisted me in executing what has become the greatest decision of my life. While there is not enough room to thank every person who provided assistance on my journey, there are a few people whom I feel the need to name:

- To my mom and dad, thank you for wholly supporting my decision to quit my job, leave town, and pursue my calling at Ferris State. Thank you for always providing a warm bed during those weekends spent home and always picking up the phone when I needed to vent.
- To my grandparents, thank you for opening up your home and allowing me a place to stay during my time on campus. I am sure it was not easy having your life uprooted, but you took things in stride. More importantly, you somehow allowed my best friend to tag along, which gave him the break he needed to also pursue his passion.
- To my sister, thank you for "paving the way" for me at Ferris. I know it has difficult living in the shadow of your big brother as you made your way through your K-12 education. It is only fitting that the roles were reversed. If it were not the connections I had made during your time at Ferris, my social life would have been much less exciting.
- To my Uncle John and Uncle Curt, thank you for doing what an uncle does best: providing me an outlet to complain about things I wouldn't share with my parents and enabling my terrible behavior. No matter what the situation, you both were always quick to pick up the phone when I needed to vent or slip me a few dollars for a case of beer.
- To Jeff, thanks for being an excellent friend despite being stuck with me for the better part of two years. There is no doubt that transitioning to "college life" was difficult, but I like to think that we made the best of it. While there is no doubt that our pre-college days

were full of countless exploits, we both can agree that the decision to move to Ferris worked out for the better.

- To the rest of my friends, thank you for putting up with me over the past few years as I went through what could only be described as a quarter life crisis. From the summer of 2016 where I was in full-on denial, to the winter of 2016 where I shut myself off from the world to study, you guys took it in stride. While I joke about my parents never giving me the brother(s) I wanted, you all did a pretty good job picking up the slack.
- To the faculty of the Information Security & Intelligence Department, thank you for putting up with me over these four years. Despite me being what I could only describe as a complete pain, you took my questions in stride and went far above and beyond in assisting me in any way possible.
- To Sandy Gholston, Ferris State News Services Manager, thank you for always leaving your door open. The advice you gave me over the past few years has been invaluable.
- To Dr. Patrick Bishop, thank you for taking me under your wing and, along with Sandy, fully supporting my inclusion in the Public Relations Student Society of America (PRSSA). As one who tends to bury his head in work, joining PRSSA afforded me the opportunity to not only take a “break” from my coursework, but share both my leadership and writing skills with a broader audience.

List of Contents

List of Tables

List of Figures

Abstract

In recent years, we have witnessed what could be considered the advent of commodity single-board computers. No one machine has dominated this segment of the market like the Raspberry Pi. In its latest iteration, the Raspberry Pi 4, we have been gifted with a piece of hardware that includes two network interfaces: a wired Ethernet port, and built-in 802.11 ac wireless connectivity. With this latest iteration, the Raspberry Pi 4 has shed its reputation for being relegated only to the “makers” for low-overhead Internet of Things deployments and solidified itself as an effective micro computing platform. Boasting not only the aforementioned wireless connectivity, but up to 4GB of ram, the Pi 4’s appears as though it could prove to be a cost-effective tool in any information security professional’s arsenal. Therefore, this project explores the deployment of a Pi 4 as a traffic capture device as a means to perform network auditing. Over the course of two weeks, the Pi 4 performed a continuous packet capture of a local network consisting of a singular Google Home Mini. While this task would be trivial with a more traditional hardware configuration, such as a commodity notebook computer, there were several issues in just the configuration of the Raspberry Pi to even collect network packets. In truth, this configuration stage took more man hours than even the packet analysis stage, meaning that once one accounted for labor costs, it would be difficult to make a business case for purchasing a Pi 4 strictly for the purpose of packet capturing if more suitable hardware is already on hand.

Chapter 1

Introduction

On February 29, 2012, the Raspberry Pi Foundation released what is now known as the "Raspberry Pi Model B", a single-board computer brought to fruition by British engineer Eben Upton (Fromaget, 2020). The goal for the Raspberry Pi project was simple: Upton aimed to create an inexpensive computing platform that would allow for the widespread education of the next generation of computer programmers. In the eight years since the Pi's commercial release, Upton's $35 computer has taken the "maker" world by storm, creating entire computer subculture centered upon the credit card-sized machine.

The benefits of the Raspberry Pi extend beyond "maker" spaces, however. While Upton may have intended for his machine to inspire young kids to enter computer science, the machine has become ubiquitous in academia and is starting to become more prevalent in the world of private industry. The machines low coast and flexibility has allowed people to use the device for everything from monitoring plant growth to operating as an automobile stereo system.

Unfortunately, the world of Information Security has seemingly been unable to embrace the Raspberry Pi. While the profession certainly has roots in "hacker" culture, the truth is that to date there have been few applications within the field for low-power hardware. In fact, most security tools carry a hefty price tag. For every open-source vulnerability scanner, there exists countless paid options that come bundled with professional services contracts that can run hundreds of thousands of dollars per year.

Background

It was the goal of this project, then, to identify an area in which a Raspberry Pi could be useful to the modern-day Information Security professional. While there is no debating that the machine could be used for the purpose of penetration testing, the fact remains that not only is the machine relatively underpowered for such a task, it is inferior to a notebook computer in that one is required to connect the Pi to a monitor, keyboard, and mouse in order to perform basic functions. That is, unless one wishes to connect to the Pi via SSH, which would also require a separate machine. Unfortunately, the Pi does not lend itself well to jobs requiring heavy user interaction. The question, then, becomes what tasks does a security professional perform that can be largely automated?

Statement of Problem

As the Information Security space has grown, so too have the number of firms willing to offer their services. From “Big Four” firms such as Deloitte or KPMG, to niche companies such as Whitehat Security or Darktrace, there is no shortage of suitors for organizations with even a meager IT budget.

Unfortunately, with Information Security now being a requirement for even small businesses, such as those beholden to the Payment Card Industry Data Security Standard (PCI DSS), there is a notable shortage of quality tools which could be leveraged by smaller firms and the security personnel that service them.

Purpose of the Study

The goal of this project was to explore a proposed solution which, in theory, would blend the latest in commodity hardware with industry-standard open source software as a means of constructing a portable and versatile network analysis device with a street price of less than $100 USD. For the purpose of this exercise, the proposed solution is centered upon the use of a Raspberry Pi Model 3 B, a single-board computer that is all but ubiquitous in the Science Technology Engineering and Mathematics (STEM) fields.

Rationale

The project centers upon two aspects: hardware and software. From a hardware perspective, this portion will be executed by combining a combination of commercial off-the-shelf (COTS) hardware products to create a Minimum Viable Product (MVP) that is capable of performing any tasks as dictated by the installed software. The software installed on this machine should not only be capable of leveraging the hardware's networking capabilities to intercept network traffic but be capable of producing actionable data for analysis by an Information Security practitioner.

Questions to be Addressed

Bearing in mind the aforementioned project goals and rationale, the author of this project set to answer the following question(s):

- Is it possible to construct a network analysis appliance for less than $100 that could be reliability deployed in a real-world environment?
- Is the quality of the data captured by the device high enough to serve as a viable alternative to more mainstream security appliances?

Nature of the Study

The questions outlined above will be addressed in a systematic order. First, the various parts required to assemble the hardware component of this project, thus resulting in what could be considered to be, for all intents and purposes, a "mini computer." Upon this machine, we will install a standard operating system and configure it in a way which will allow for the capture of network traffic. After 7 days of data collection, the data will be reviewed, and a determination would be made regarding the machine's ability to meet the requirements set forth by the questions.

Project Importance

This project is important for a multitude of reasons. By 2026, today's $173 Billion cybersecurity market is expected to grow to over $270 Billion (Columbus, 2020). Despite the sizable number, these increases will not be seen at the "top" of the industry. In fact, the industry as a whole is seeing significant growth in the wake of automation, particularly involving automobiles. As the industry grows to service this new mobile platform, there will not only be an influx of new vendors in the security space, but an increased need for flexibility in security operations. Therefore, the demand for inexpensive and portable security appliances is likely to only increase as these industries mature. If the proposed devices proves to be an adequate network capture device, it is likely that the hardware could also be tasked to carry out many other security functions, such as acting as a mobile penetration testing lab.

From a personal standpoint, such a device would allow for the streamlining of various security operations tasks within our own organization. In fact, the idea of using a Raspberry Pi as a mobile penetration testing lab originated in this space. When one takes into consideration the computing power of a typical company-issue notebook computer, the specifications are not entirely dissimilar. More importantly, the inexpensive nature of the Raspberry Pi allows for a security professional to acquire a machine that is not tied to any typical corporate restrictions and thus allow for one to perform tasks such as malware analysis without the fear of introducing malicious programs into the company's network.

Definition of Terms

Bridge (Networking) – A type of networking configuration that allows one to "interconnect two Local Area Networks (LANs) together" (Beasley & Piyasat, 2016)

Broadband – "of, relating to, or being a high-speed communications network and especially one in which a frequency range is divided into multiple independent channels for simultaneous transmission of signals (such as voice, data, or video)" (Merriam-Webster, Incorporated, 2020)

Command – "A command is an instruction given by a user telling a computer to do something such a run a single program or a group of linked programs" (The Linux Information Project , n.d.)

Command Line – "A text interface for a computer that takes in commands, which it passes on to the computer's operating system to run" (Codeacademy, n.d.)

Commercial off-the-shelf (COTS) – " Software and hardware that already exists and is available from commercial sources" (National Institute of Standards and Technology, n.d.)

Debian – "A particular distribution of the Linux operating system, and numerous packages that run on it" (Software in the Public Interest, Inc., n.d.)

Dnsmasq – "A simple DHCP/DNS server utility which can be used in a local network" (Gentoo Foundation, Inc., 2020)

DHCP Lease – "The time period for which a DHCP server allocates a network address to a client" (Palo Alto Networks, Inc., 2020)

Ethernet – " A computer network architecture consisting of various specified local-area network protocols, devices, and connection methods" (Merriam-Webster, Incorporated, 2020)

File Allocation Table (FAT) – “A table that the operating system uses to locate files on a disk. Due to fragmentation, a file may be divided into many sections that are scattered around the disk” (Beal, FAT - file allocation table, 2020)

FAT32 – “A version of the file allocation table (FAT) available in Windows 95 OSR 2 and Windows 98” (Beal, FAT32, 2020)

Graphical User Interface (GUI) – “A computer program that enables a person to communicate with a computer through the use of symbols, visual metaphors, and pointing devices” (Levy, 2020)

Flash Memory – “A type of electronically erasable programmable read-only memory (EEPROM), memory chips that retain information without requiring power” (Kay, 2010)

Hardcode – “To put information into a software program so that it cannot be easily changed by a user” (Cambridge University Press, 2020)

Headless – “A computer that is operated without the traditional monitor, mouse and keyboard peripherals” (Techopedia Inc., 2013)

Host Access Point Daemon (Hostapd) – “A user space software access point capable of turning normal network interface cards into access points and authentication servers.” (Gentoo Foundation, Inc., 2020)

Internet of Things (IoT) – “A system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction” (Rouse, internet of things (IoT), 2020)

Internet Protocol (IP) Address – “A unique address that identifies a device on the Internet or a local network” (SharpenedProductions, 2020)

Internet Service Provider (ISP) – "A company that provides Internet connections and services to individuals and organizations" (The Editors of Encyclopaedia Britannica, 2020)

Job – "A job refers to a unit of work or set of instructions given to an operating system to execute" (Techopedia, Inc., 2011)

Linux – "Linux is a Unix-like, open source and community-developed operating system for computers, servers, mainframes, mobile devices and embedded devices" (Rouse, Linux operating system, 2016)

Local Area Network (LAN) – "A local area network (LAN) is a collection of devices connected together in one physical location, such as a building, office, or home" (Cisco Systems, Inc., 2020)

Man in the Middle (MitM) Attack – "A man-in-the-middle (MitM) attack is when an attacker intercepts communications between two parties either to secretly eavesdrop or modify traffic traveling between the two" (Swinhoe, 2019)

Micro Secure Digital (SD) – "A type of memory card used to store data in portable electronic devices" (Sharpened Productions, 2020) Micro denotes a specific size class of SD card.

Network Interface Card (NIC) – A piece of "Network Adapter Hardware" that "provides the hardware interface between a computer and a network, regardless of whether the network is wired or wireless" (Mitchell, 2019)

Open Source – "Something people can modify and share because its design is publicly accessible" (Red Hat, Inc., 2019)

Operating System – "Software that controls the operation of a computer and directs the processing of programs" (Merriam-Webster, Incorporated, 2020)

Packet – "A container used to convey data or information via TCP/IP or similar protocol across a network" (Beckert, 2020)

Packet Sniffer – "A piece of hardware or software used to monitor network traffic" (AO Kaspersky Lab, 2020)

Public Key Encryption – "An asymmetric encryption technique which uses different keys for encryption and decryption, allowing computers over the Internet to securely communicate with each other" (Fox, 2020)

Random Access Memory (RAM) – "A computer's short-term data storage; it stores the information a computer is actively using so that it can be accessed quickly" (Micron Technology, Inc., 2020)

Raspberry Pi – "A low cost, credit-card sized computer" (Raspberry Pi Foundation, n.d.)

Raspberry Pi Foundation – "A United Kingdom (UK) charity that aims to educate people in computing and create easier access to computing education" (Red Hat, Inc., 2019)

Router – "A device that communicates between the internet and the devices that connect to the internet" (NortonLifeLock Inc., 2020)

Secure Shell (SSH) – "A remote administration protocol that allows users to control and modify their remote servers over the Internet" (L., 2017)

Single-Board Computer – "Any complete computer that is built on a single circuit board and contains functional computer components including the microprocessor, input/output (I/O) and memory" (Beal, SBC - single-board computer, 2020)

Smart Home – "A home setup where appliances and devices can be automatically controlled remotely from anywhere with an internet connection using a mobile or other networked device" (Chen, 2020)

Tcpdump - “A command-line packet sniffer/analyzer” (The Tcpdump Group, 2020)

Wireless Access Point (WAP) – “A device that creates a wireless local area network (WLAN)” (Belkin International, Inc., 2020)

Wireshark – “A network protocol analyzer” (OffSec Services Limited, 2020). *See Packet Sniffer*

Assumptions and Limitations

In preparation for this project, a list of notable assumptions and limitations have been developed. These are recorded as a means to not only lay the groundwork for the project itself, but expeditated the planning stages by eliminating some variables. For example, it is assumed that the hardware purchased will arrive in a functional state and be free of defects. Likewise, it is assumed that once the device is constructed and configured that it will be able to perform at least some level of traffic capture. Regarding limitations, it would be nearly impossible to test for all combinations of hardware, therefore the project was constructed around using parts either created or recommended by the Raspberry Pi Foundation.

In addition, the nature of the project dictates that any testing be performed within a controlled network. Therefore, it is assumed that a controlled network could be created for the purpose of testing, bearing in mind the limitation that in order to account for sound conclusions to be made, one must limit the engagement to a single network with a single configuration in order to ensure that project outcomes could be duplicated.

Chapter 2

In preparation for this project, significant research was conducted in order to identify studies where a Raspberry Pi was used in a similar capacity. The purpose of this research was not only an attempt to identify "best practices" from a configuration standpoint, but to determine whether the premise of the project was flawed in its entirety.

Being in that the Raspberry Pi was created as a pseudo-educational tool, there has been numerous studies conducted that center entirely upon identifying use cases for the machine. In attempting to research similar use cases, however, it was difficult to identify any peer-reviewed journal articles involving the Pi's use in the Information Security field. Much has been made of the performance and versatility of the machine's photographic capabilities, with articles published covering topics such as "Raspberry Pi–powered imaging for plant phenotyping" (Tovar, et al., 2018), to "Assessing the Raspberry Pi as a low-cost alternative for acquisition of near infrared hemispherical digital imagery" (Kirby, Chapman, & Chapman, 2018).

Perhaps the closest approximation, in a sense, is the later mentioned article. While the use case of Kirby, Chapman, and Chapman's project was focused upon the optical capabilities of the Pi, the impetus for their project was not dissimilar to that of our network analysis suite. By their own admission, the Raspberry Pi "designed to minimize the cost of computing and thus make it, and computer programming more generally, accessible to a wide audience" (Kirby, Chapman, & Chapman, 2018). To put it plainly, this group selected the Raspberry Pi for the basis of their project for the same reasons it was felt the Pi would be able to meet the requirements of this project: it is inexpensive, it is readily available, and it is versatile in how it can be configured and/or programmed.

More importantly, the Pi was versatile enough that it “could be left in the field …. for long periods [of] time” (Kirby, Chapman, & Chapman, 2018). While in this context, “the field” literally referred to the device being left out in a field, this situation is not dissimilar to the ability for a security practitioner to leave the Pi in a remote location for an extended period of time. That, perhaps, is the beauty of the Raspberry Pi platform: it has a set of universal characteristics which can be leveraged in a multitude of ways due to its inherent flexibility.

Despite there being a lack of studies involving the use of a Raspberry Pi in the Information Security space, there is previous precedent for the use of the device as part of a larger “smart home” deployment. This is relevant in that for the purpose of our project, the Pi would be configured to intercept traffic on a Local Area Network (LAN), consisting of a Google Home Mini. In this instance, however, the Pi was used as a “hub” that would control various smart peripherals in the home, ranging from fans to light bulbs.

The goal of this project was to bridge the gap from the home network to the device of a homeowner’s choosing through the “integration of information technology with the home environment, systems and equipment” allowing them to “communicate in an integrated manner” (Saputri & Rofiq, 2018). This is particularly relevant in that not only was the Pi configured as a way to bridge the gap between humans and Internet of Things (IoT) devices but was means to operate in a “headless” state and thus perform its programmed functions without regular administrative oversight.

This operation mirrors how one may utilize the Pi to capture network traffic in that one would theoretically connect the Pi to a specific target network and allow the packet capture to occur for a specific length of time. As mentioned previously, the relative portability and small cost associated with deploying the Raspberry Pi allow for deployments such as in this situation to occur without tying up more resource-intensive hardware such as a notebook computer or similar.

Further research regarding the Pi's use in home automation uncovered several projects centering upon leveraging the Pi's versatility to act as a "hub" for controlling other devices. One In particular, however made specific mention of security. This group in particular went as far as to say that "security is the most important feature of a home assistant solution" (Iftimie & Vințe, 2017).

It should be noted, however, that the primary goal of Iftimie and Vințe was not to construct a security specific tool. Instead, their primary concern was ensuring the security of the Raspberry Pi from the perspect of managing access to the device itself. This research is relevant to our own project in that it is likely that a security professional may let the network traffic capture device to run for an extended period of time in a remote location, thus requiring some form of remote management not unlike a more traditional server. In this instance, Iftimie and Vințe, attempted to "secure" their device by assigning the administrative interface a "random" port number and employing Secure Shell (SSH) with public key encryption. The pitfalls of such an implementation is that a non-standard port is not inherently much more difficult to identify than a default port. Most importantly, the decision was made to "hardcode" the private encryption key because "it would have been too much overhead for the user to remember an encryption key" (Iftimie & Vințe, 2017). While the intent of the authors was to create a secure system, this particular example not only highlighted the possible security vulnerabilities involved when attempting to deploy a Raspberry Pi in a remote manner, but the misunderstanding of secure deployments by those outside of the Information Security field.

Based upon the resources at this group's disposal, there has yet to be specific research involving the efficacy of the Raspberry Pi as a dedicated security tool. In no-academic circles, however, projects such as the one proposed here are very common. In fact, there is no shortage of tutorials that highlight the different means in which one can configure a Raspberry Pi to operate as some sort of packet "sniffer"/traffic analysis tool.

Based upon the information collected, we have determined that is absolutely possible to capture network traffic with a Raspberry Pi. The question, however, remains what exactly is the best combination of operating system and software that can lead to the desired results. More importanly, will this combination be able to produce data on par with more traditional methods? For the most part, these online "tutorials" involve some sort of non-standard packet capture software. With the Raspberry Pi allowing for any number of Linux distributions, there are seemingly endless software options for capturing traffic. However, in the security world, there are two main standouts: tcpdump and Wireshark. If the Raspberry Pi were to be an invaluable tool for security professionals, it needed to work with those two programs at a minimum.

Chapter 3

Description of Methodology

This study is unique in that it centers entirely upon a real-world use case and therefore no amount of research would be able to answer the questions set forth in Chapter 1. As evidenced by the research outlined in Chapter 2, most scholastic inquiries regarding the use of a Raspberry Pi has centered upon aspects only tangentially related to information security at best. The research was helpful, however, in identifying the steps required in order to configure the Pi to capture network traffic, but this information did not provide any insight into the machine's ability to carry out the required tasks effectively.

In short, the only way to answer these questions is to mirror a typical engagement that may be carried out by a security practitioner. With that being said the question of whether a Raspberry Pi can be configured to capture traffic is almost a given, but the outcome of this project depends almost entirely upon the initial configuration of the device. Thus, it is expected that there be some level of "trial and error" in the construction of a working unit that will allow for the engagement itself to commence.

Design of Study

As previously stated, the execution of this project is dependent upon the successful configuration and deployment of the Raspberry Pi. In this case, this means the assembly of the unit itself, the installation of the operating system, the installation and configuration of the security-related software, and a testing phase to confirm correct operation.

Assembling the machine itself is pretty straightforward. The Raspberry Pi 4 is a single board computer, with nearly all interfaces built into the board itself. In order to operate, the machine requires the following:

- A Class 10 Micro SD flash memory card with a minimum of 16 GB capacity (Raspberry Pi Foundation, 2020)
- An AC power supply capable of supplying at least 3 amps @ 5 volts (Raspberry Pi Foundation, 2020)

The above represents the absolute minimum required to deploy the Raspberry Pi in a "headless" state, meaning without the use of connected input/output devices. For this project, however, the configuration process would be significantly faster if the device were connected to a traditional computer monitor, mouse, and keyboard. In addition, it was decided to employ the use of a protective case for the Raspberry Pi, being in that the use case for this project would involve leaving the device in an unknown environment for some length of time. Therefore, a case would be required to provide a nominal level of protection from accidental damage due to falls, liquid spills, etc.

Aside from the input/output devices, all other parts required had to meet certain criteria in order to be compatible with the Raspberry Pi. Therefore, it was decided to source the official power supply and case offered by the Raspberry Pi Foundation to ensure that all requirements were met as the producers had intended. The Micro SD cars was selected from a list of "preferred" models published by a multitude of vendors, with most settling upon the 64GB SanDisk Extreme as the best selection from a cost/performance perspective (Hildenbrand, 2020).

In order to carry out this project on their own, they must first acquire the materials listed above, paying particular attention to overall cost in order to not exceed the $100 total cost. Once this has been completed, a user must format the micro SD card in Fat 32 format, thus allowing for the writing of the desired operating system to the SD card.

Originally, the project plan called for the use of Kali Linux, a specific Linux distribution aimed at security professionals. Kali is based upon the Debian Linux distribution and is therefore almost universally portable. In this instance, Kali has a specific image for use on Raspberry Pi computers.

After initial setup had been completed, the machine would be configured for performing the packet capture. As stated previously, either Tcpdump or Wireshark could be used for this purpose. For a lot of security professionals, both products are used with frequency. The benefit of Tcpdump is that it is a command line utility and therefore is considered by some to be more versatile. Not only can Tcpdump be run as a background process, but that process can be broken into "jobs" which can reset at a certain interval. For example, one could craft a Tcpdump command in a way which would allow for a new capture (and thus a new capture file) every 24 hours. This, in theory, would allow for a much easier analysis for the security professional performing the analysis.

On the other hand, Wireshark is a more "traditional" tool in that it leverages the Graphical User Interface (GUI) not unlike the majority of user-focused computer programs. Due to this, Wireshark is generally considered to be easier to use for those who do not possess the acumen to use Tcpdump effectively. More importantly, the graphical nature of Wireshark allows for the utility to be significantly more useful than Tcpdump in the analysis phase.

Depending upon how one may wish to capture traffic, the above steps may be all that is needed in order to create a functioning capture device. At a low level, one would simply have to plug an ethernet cord into the Raspberry Pi and launch Wireshark. In moments one could be collecting traffic.

This, however, is a severe underutilization of the Raspberry Pi's capabilities. As stated previously, the Pi 4 contains both a wired and a wireless Network Interface Card (NIC). Therefore, one could use either the Pi to capture traffic on a wired or wireless Local Area Network (LAN).

Perhaps more interesting is the ability to "bridge" these connections, thus allowing for the Pi to operate as a Wireless Access Point (WAP) and operate as a "man in the middle" (MITM). Here, the wired connection of the Pi is plugged into a router or modem and that connection is shared via the wireless connection. This would allow for one to intercept any and all traffic coming through the device. While this type of MITM attack could be used to exploit the Pi's connection with other devices, for the purposes of this project the Pi was connected in this manner in an attempt to isolate specific devices from the rest of the author's home network.

Data Analysis

Being in that this is a project and not strictly a research assignment, data does not play as large of a role in determining whether the proposed solution will answer any or all of the questions laid out in Chapter 1. With regards to whether the device can be configured to capture network traffic for less than $100, this is a question that is binary in nature; either you can configure the device to capture traffic for less than that cost, or you cannot. As previously stated, the premise of this question is at least partly flawed in that it has been proven by others that the Raspberry Pi can be configured to operate in this manner.

The real question, then, is the quality of this collection and whether the device could provide an alternative to more expensive security appliances. While answering this question requires the collection of data, the truth is that said data will not allow for a quantitative assessment of the Pi's success or failure. In layman's terms, the data collected in this project has as much bearing on the outcome as the projects listed in Chapter two. At best, each has only a minor influence over the project at any level.

With this in mind, the success of the device will be judged based upon the following criteria:

- Can the device be configured to collect network traffic?
    - If so, was the cost of the device and required peripherals less than $100?
    - If so, was the data collected on par with traffic captured by security-specific appliances?

Chapter 4

In preparation for this project, the following materials were purchased from my local Micro Center in Madison Heights, MI:

Table 1

Initial Hardware Purchase

Item Name	Price	Tax	Total
Raspberry Pi 4 Model B – 4 GB RAM	$49.99	$3.00	$52.00
SanDisk 64GB Extreme Plus microSD	$29.99	$1.80	$31.79
Raspberry Pi 4 Official Power Supply	$7.99	$0.48	$8.47
Raspberry Pi 4 Official Case	$4.99	$0.30	$5.29
TOTAL	**$92.96**	**$5.58**	**$98.54**

It should be noted that the above table represents the absolute minimum which would be required to deploy the Raspberry Pi in a real-world scenario, but does not include the aforementioned input/output devices required for the initial setup, nor does it account for the fact that a second computer is required for the formatting of the MicroSD card and loading of the operating system. Such hardware was assumed to be standard for any computer user and therefore does not constitute a purchase specific to the Pi itself.

While one could debate the veracity of these prices, the fact remains that the world of retail is fluid. Therefore, pricing and availability on these items has likely changed since the time of purchase. It should also be noted that these items were for purchasing from a brick-and-mortar establishment and therefore better deals could possibly be had from online retailers.

Nevertheless, through careful selection of components, the device was able to be constructed under budget. Substitutions could possibly be made for any of the above items which may also have an effect on price. For example, a third-party power supply or less expensive SD card would almost certainly reduce the overall cost of the project. In addition, there are versions of the Raspberry Pi 4 which feature 1 or 2 GB of Random Access Memory (RAM). Such versions of the Pi retail for $29.99 and $35.00, respectively. Considering that this project was carried out exclusively with the 4 GB model, no guarantees can be made regarding the ability of those devices to meet the stated requirements.

While configuration can vary, for the purpose of this project we identified four main tasks required to configure the machine to act as the wireless capture device:

- Install operating system
- Install packet capture/analysis software
- Install wireless access point/router software
- Bridge wired and wireless network interface cards

As previously stated, simply installing the operating system and the packet capture/analysis software would be sufficient if one were to use this device for strictly wired connections. However, the native wireless connectivity opens up the device to being used in wireless environments without the need for an add-on wireless card, which typically run in the $40-50 range (HackersGrid, 2020).

For the completion of the first configuration step, the installation of the operating system, the decision was made to install Kali Linux, a Debian-based Linux distribution that is targeted at the Information Security professional. This distribution contains a multitude of built-in tools that see regular use in the security field, including the two being used in this project: Wireshark and Tcpdump.

This also was where the first compatibility issue started to rear its head. As previously mentioned, the Micro SD card for a Raspberry Pi needs to be formatted in FAT32, a which is a legacy format that had been used by Microsoft Windows' previous versions. Unfortunately, the machine being used for this initial configuration is built around a Macintosh notebook and therefore there exists no built-in utility to properly format this disc. While macOS' Disc Utility claims to format in MacOS, any card formatted this way was not readable by the Raspberry Pi. Fortunately, the organization who maintains the SD standard, the SD Association, offers a free disk formatting tool for a multitude of platforms, including macOS. This utility was able to format the Micro SD card without issue.

With the formatting issue out of the way, there was still the issue of "burning" the disc image to the SD card. Despite the Raspberry Pi foundation offering the "Raspberry Pi Imager' software, this software was unable to operate properly on the Mac. Attempts with this software were also made within a Windows 7 Virtual Machine but were to no avail. The Raspberry Pi user community had made the recommendation to try Balena Etcher, an open-source tool whose sole purpose was to write operating system images to SD cards for use in a Raspberry Pi. This program operated flawlessly and thus allowed for the project to move forward.

The installation of the operating system was not unlike any other installation. What was originally expected to be a lengthy period due to the Raspberry Pi's relatively unimpressive specifications turned out to be a fairly seamless installation. Both Tcpdump and Wireshark were able to install without issue and a test was performed to ensure that these programs were able to capture traffic moving through the Raspberry Pi's network interfaces.

Problems arose when installing two pieces of software, Hostapd and Dnsmasq. These programs allow the Raspberry Pi to act as a wireless access point and a router, respectively. Despite multiple attempts, both programs could not be configured to operate together and therefore capture traffic while acting as the proverbial "man in the middle". Whether it be missing utilities, deprecated package managers, or missing drivers, there were seemingly endless compatibility issues in attempting to use Kali Linux in this manner. 11Rather than spend further time troubleshooting these issues, the decision was made to revert to a known compatible Linux distribution: Raspbian.

The installation of Raspbian was not unlike any other operating system, including Kali Linux. Being in that Raspbian is optimized for the Raspberry Pi, the installation was even less difficult. More importantly, all official Raspberry Pi tutorials for setting up the device as a wireless access point used a machine running Raspbian.

Installation of all software took only minutes, with all tools able to be installed without issue. At most, one will be required to editing a few configuration files for the purpose of assigning a SSID and password to the new network with Hostapd. Dnsmasq not only was able to handle the routing side of things but could also be configured to bridge the wired and wireless connections. This would allow the Raspberry Pi to be plugged into a traditional broadband modem/router via ethernet as a means of providing internet connectivity. From there, the Pi would act as a wireless router/access point and allow a device to connect to the Pi's wireless connection and reach the internet via the broadband connection. This would allow for a layer of network segmentation and, in theory, allow the Pi to only capture traffic passing through it.

With the machine up and running, it was time to perform a test. A Google Home Mini was set up and connected to the wireless network running on the Raspberry Pi. In order to reduce traffic generated by the Google Home to an absolute minimum, the decision was made to leave the device in a secluded closet where it would not pick up normal voice chatter and attempt to reach out to Google servers for erroneous queries. In addition, the following options were disabled:

- Diagnostic Data Telemetry
- Voice Match
- Contacts, Web History, and Calendar Sync
- Personalized Search Results

The reasoning behind these configuration choices were that the reduction in “personalized” service provided by the Google Home would result in less network traffic.

With everything configured and both the Raspberry Pi and the Google Home configured, it was time to perform testing to ensure that all aspects of the project were operational. For the first test, it was decided to run Wireshark for a 24-hour period and review the results. During this period, the Pi would be left “running” and the status of the machine would be checked at random intervals throughout the day.

Upon completion of this test, preliminary results looked promising. There was no doubt that the Raspberry Pi could function as a traffic capture device in this manner. The primary issue, however, was with the amount of traffic the device was capturing. Despite the Wireshark “listener” being pointed towards the wireless connection, it became readily apparent that the device was also capturing any and all traffic as it came across the broadband router. Therefore, all traffic for the entire home network was being captured.

Fortunately, Wireshark offers fairly robust analysis features, and it would be, in theory, relatively easy to parse out data relating to the Google Home. Unfortunately, however, this would prove to be difficult with the current networking hardware being used to the project. The broadband router/modem was a bespoke unit supplied directly by the Internet Service Provider and therefore lacks features common in routers available at retail.

One such feature is the DHCP Lease, which effectively manages what IP address is assigned to a device on a network and the length of that assignment. In order to allow for easy parsing of the collected data, it would be easiest to assign the Google Home a specific IP address as a means to filter out any data not passing through the Raspberry Pi. While most devices allow for an "unlimited" lease, the equipment in question only had a max lease time of 7 days, making for additional work during analysis. It should be noted, however, that this specific problem is not one rooted in a problem with the Raspberry Pi but could have acted as a mitigation strategy of sorts due to the Pi's collection of all network traffic.

With these issues in mind, the decision was made to forward with a 7-day test in order to determine the Pi's ability to handle a prolonged engagement. All variables were kept the same from the single day test, except for conducting one daily status check. For the first few days, the Pi soldered on without issue. On day six, however, it was noted that the machine was no longer operating and refused to boot. Based upon further research, the machine had overheated and suffered a catastrophic failure. The official Raspberry Pi case is said to restrict airflow and thus can cause issued when the machine is running at or near it's resource limit for an extended period of time.

Figure 1. Raspberry Pi Official Case

Not accepting defeat, the decision was made to acquire a replacement Raspberry Pi and a different case. This cost was not figured into the budget due to the fact that the local retailer had agreed to warranty the unit. The new case was an open design, using a basic plastic skeleton and features both CPU and GPU and a cooling fan. This case costs $9.99, and thus the budget has been updated as seen below:

Table 2

Updated Hardware Purchase

Item Name	Price	Tax	Total
Raspberry Pi 4 Model B – 4 GB RAM	$49.99	$3.00	$52.00
SanDisk 64GB Extreme Plus microSD	$29.99	$1.80	$31.79
~~Raspberry Pi 4 Official Power Supply~~	~~$7.99~~	~~$0.48~~	~~$8.47~~
GeeekPi Acrylic Case for Raspberry Pi 4 Model B	$9.99	$0.60	$10.59
Raspberry Pi 4 Official Case	$4.99	$0.30	$5.29
TOTAL	**$94.96**	**$5.70**	**$100.66**

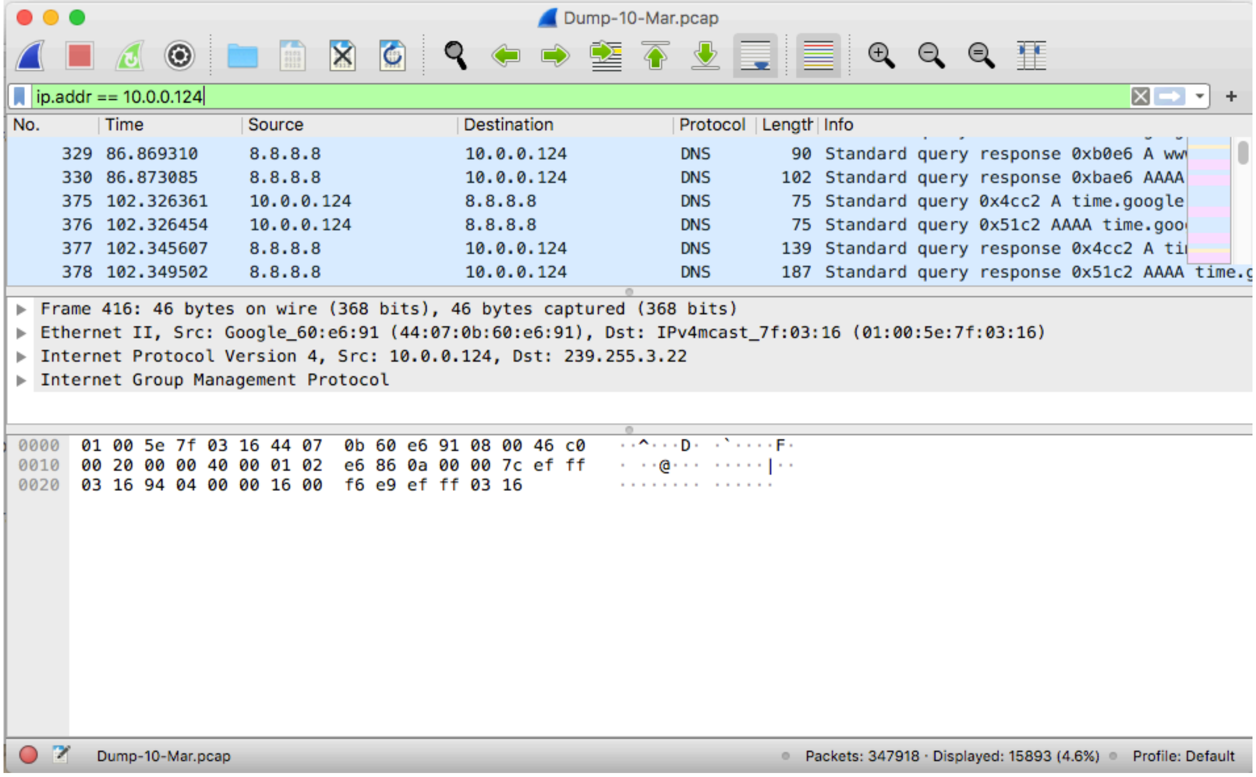
As evidenced above, in current configuration the $100 budget had been exceeded by a $0.66. Despite this overrun, the decision was made to move forward with the project and continue with testing.

Figure 2. GeeekPi Acrylic Case for Raspberry Pi 4 Model B

With the new case installed and functioning, a decision was made to fundamentally change the test; for the next 7 days, the Raspberry Pi would be running Tcpdump instead of Wireshark. The reasoning for this decision was twofold: first, Tcpdump was a command line program and thus would require less computing overhead as it requires little in the way of graphical processing power to operate. This would result in less heat generation. More importantly, Tcpdump can be set to run as a “job” and therefore save traffic capture files based upon a predetermined time period. This would not only allow for easier analysis considering the amount of traffic being captured, but in theory would ensure that not all data would be lost in the event of another thermal shutdown.

As this test commenced, the status of the device was checked daily, with the previous day's capture being reviewed in Wireshark to ensure proper function. After 7 days, the device continued to operate flawlessly, not counting the network segmentation issue outlined above. In an attempt to try and overload the system, these files were uploaded to a cloud storage solution and the decision was made to continue for another 7 days. By day 14, no performance degradation was noticed. From the perspective of thermal management, it appears as though all issues with the device had been addressed through the use of the updated case.

In analyzing the traffic captured by the Raspberry Pi, it was readily apparent that the device was capturing any and all traffic. While the number of packets captured varied by day, the amount of traffic captured by the device was almost overwhelming, even when filtering for packets only relating to the Google Home. For example, if one were to look at the capture from 10 March and filtered that file to only display data originating from or being sent to the Google Home's IP address (10.0.0.124), they would be met with 15,893 packets alone!



```
ip.addr == 10.0.0.124
```

```
No.    Time        Source        Destination   Protocol  Length  Info
  329  86.869310   8.8.8.8       10.0.0.124    DNS          90   Standard query response 0xb0e6 A ww
  330  86.873085   8.8.8.8       10.0.0.124    DNS         102   Standard query response 0xbae6 AAAA
  375  102.326361  10.0.0.124    8.8.8.8       DNS          75   Standard query 0x4cc2 A time.google
  376  102.326454  10.0.0.124    8.8.8.8       DNS          75   Standard query 0x51c2 AAAA time.goo
  377  102.345607  8.8.8.8       10.0.0.124    DNS         139   Standard query response 0x4cc2 A ti
  378  102.349502  8.8.8.8       10.0.0.124    DNS         187   Standard query response 0x51c2 AAAA time.g
```

```
▶ Frame 416: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)
▶ Ethernet II, Src: Google_60:e6:91 (44:07:0b:60:e6:91), Dst: IPv4mcast_7f:03:16 (01:00:5e:7f:03:16)
▶ Internet Protocol Version 4, Src: 10.0.0.124, Dst: 239.255.3.22
▶ Internet Group Management Protocol
```

```
0000  01 00 5e 7f 03 16 44 07  0b 60 e6 91 08 00 46 c0   ··^···D· ·`····F·
0010  00 20 00 00 40 00 01 02  e6 86 0a 00 00 7c ef ff   · ··@··· ·····|··
0020  03 16 94 04 00 00 16 00  f6 e9 ef ff 03 16         ········ ······
```

Figure 3. Packet capture for 10 March 2020, filtered by IP address for Google Home (10.0.0.124)

Chapter 5

If one were to take the outcomes of this project strictly at face value, the question as to whether the Raspberry Pi was able to effectively answer the questions laid out in Chapter 1 would be a resounding no. Under the conditions set forth for the project, the device failed to meet either criteria by the strict letter of the requirements.

With respect to budget concerns, the requirements set forth a limit of $100 for the device and all peripherals. While the project was able to come in a few dollars under budget in the initial stages, it was proven that the purchase of the official Raspberry Pi case was a poor one. While the price difference between the official case and the one purchased later was $2.00 (plus tax), that was enough of a difference to mean that the final cost of materials was $100.66, and therefore over budget.

Secondly, it was difficult to determine the device's effectiveness with respect to the packet capture. As stated previously an undetermined configuration issue, perhaps even rooted in operator error, resulted in the device capturing traffic on both the wired and wireless interfaces, therefore resulting in the capture of additional traffic orders of magnitude larger than intended. This could also be determined a failure.

Although these issues may exist, we are fortunate to live in a world that is not so black and white. For all intents and purposes, the budget is a moot point. There is rarely a project that is completed within the initial budget, and an overrun of 0.66% equates to little more than a rounding error. In addition, it was stated in Chapter 3 that more expensive components were chosen in order to ensure device stability. One immediate error where money could be saved is with the Micro SD card; not only was entirely too much storage purchased for even 14 days of capture data, but an ultra-premium card was purchased. It is very likely that a more generic, white-label, card with smaller capacity would suffice.

In addition, there exists the possibility of a less expensive model of Pi 4 working in this type of environment, especially if one were to use the device strictly for captures via Tcpdump and perform the analysis on a more powerful machine. This has the potential to also reduce the initial cost by as much as 20% (Micro Electronics, Inc., 2020).

With respect to the ability of the Raspberry Pi to perform tasks with similar quality to more mainstream security appliances, the conclusion is at least partially inconclusive. Initial testing had shown that the device was able to capture traffic via its built-in wired and wireless interfaces. When performing any type of network capture/analysis, this would be the most likely use case. The decision to execute the “man in the middle” interception was not only a test of the device’s capabilities, but an attempt to determine whether one could also leverage a Raspberry Pi for more offensive security tasks.

A logical follow-up to this project would not only involve refining its capabilities as a capture device, but further exploiting the device’s capabilities in exploring new use cases for the device. It is likely that with more time to troubleshoot the issued highlighted earlier in this paper, the device could function as a proverbial “Swiss Army knife” for Information Security professionals. Considering not only the Raspberry Pi’s proven capabilities, but its potential, one can conclude that the $100 Raspberry Pi certainly punches above its weight class.

References

AO Kaspersky Lab. (2020). *What is a Packet Sniffer?* . Retrieved from Kaspersky: https://usa.kaspersky.com/resource-center/definitions/what-is-a-packet-sniffer

Beal, V. (2020). *FAT - file allocation table*. Retrieved from Webopedia: https://www.webopedia.com/TERM/F/file_allocation_table_FAT.html

Beal, V. (2020). *FAT32*. Retrieved from Webopedia: https://www.webopedia.com/TERM/F/FAT32.html

Beal, V. (2020). *SBC - single-board computer*. Retrieved from Webopedia: https://www.webopedia.com/TERM/S/sbc_single_board_computer.html

Beasley, J. S., & Piyasat, N. (2016). *Networking Essentials: A CompTIA Network+ N10-006 Textbook, 4th Edition.* Indianapolis: Pearson IT Certification. Retrieved from https://www.pearsonitcertification.com/articles/article.aspx?p=2474237&seqNum=2

Beckert, K. (2020, March 3). *Internet Packet: Definition & Explanation* . Retrieved from Study.com: https://study.com/academy/lesson/internet-packet-definition-lesson-quiz.html

Belkin International, Inc. (2020). *What is an Access Point and How is it Different from a Range Extender?* Retrieved from Linksys: https://www.linksys.com/us/r/what-is-a-wifi-range-extender/what-is-a-wifi-access-point/

Cambridge University Press. (2020). *hardcode*. Retrieved from Cambridge Dictionary: https://dictionary.cambridge.org/us/dictionary/english/hardcode

Chen, J. (2020, February 25). *Smart Home*. Retrieved from Investopedia: https://www.investopedia.com/terms/s/smart-home.asp

Cisco Systems, Inc. (2020). *What Is a LAN?* Retrieved from Cisco: https://www.cisco.com/c/en/us/products/switches/what-is-a-lan-local-area-network.html

Codeacademy. (n.d.). *List of Command Line Commands* . Retrieved from Code Academy: 2020

Columbus, L. (2020, April 5). *2020 Roundup Of Cybersecurity Forecasts And Market Estimates*. Retrieved from Forbes: https://www.forbes.com/sites/louiscolumbus/2020/04/05/2020-roundup-of-cybersecurity-forecasts-and-market-estimates/#134a9054381d

Fox, P. (2020). *Public key encryption* . Retrieved from Khan Academy: https://www.khanacademy.org/computing/ap-computer-science-principles/the-internet/tls-secure-data-transport/a/public-key-encryption

Fromaget, P. (2020). *The awesome story of Raspberry Pi*. Retrieved from Raspberry Tips: https://raspberrytips.com/raspberry-pi-history/

Gentoo Foundation, Inc. (2020). *Dnsmasq*. Retrieved from Gentoo Linux: https://wiki.gentoo.org/wiki/Dnsmasq

Gentoo Foundation, Inc. (2020). *Hostapd*. Retrieved from Gentoo Linux: https://wiki.gentoo.org/wiki/Hostapd

HackersGrid. (2020). *Best WiFi Adapter For Kali Linux 2020*. Retrieved from Hackers Grid: https://hackersgrid.com/2020/02/wifi-adapter-for-kali-linux.html

Hildenbrand, J. (2020, February 29). *Best SD cards for Raspberry Pi 4 in 2020* . Retrieved from Android Central: https://www.androidcentral.com/best-sd-cards-raspberry-pi-4

Iftimie, A., & Vințe, C. (2017). Upon a Home Assistant Solution Based on Raspberry Pi Platform. *Informatica Economica*, 5-16.

Kay, R. (2010, June 7). *Flash Memory*. Retrieved from Computer World: https://www.computerworld.com/article/2550624/flash-memory.html

Kirby, J., Chapman, L., & Chapman, V. (2018). Assessing the Raspberry Pi as a low-cost alternative for acquisition of near infrared hemispherical digital imagery. *Agricultural and Forest Meteorology*, 232-239.

L., A. (2017, November 10). *How does SSH Work* . Retrieved from Hostinger Tutorials: https://www.hostinger.com/tutorials/ssh-tutorial-how-does-ssh-work

Levy, S. (2020). *Graphical user interface*. Retrieved from Encyclopedia Britannica: https://www.britannica.com/technology/graphical-user-interface

Merriam-Webster, Incorporated. (2020). *Broadband*. Retrieved from Merriam-Webster: https://www.merriam-webster.com/dictionary/broadband

Merriam-Webster, Incorporated. (2020). *Ethernet*. Retrieved from Merriam-Webster: https://www.merriam-webster.com/dictionary/Ethernet

Merriam-Webster, Incorporated. (2020). *operating system*. Retrieved from Merriam-Webster: https://www.merriam-webster.com/dictionary/operating%20system

Micro Electronics, Inc. (2020). *Boards and Projects*. Retrieved from Micro Center: https://www.microcenter.com/category/4294910344/boards-projects

Micron Technology, Inc. (2020). *What is computer memory (RAM)?* . Retrieved from Cruicial US: https://www.crucial.com/articles/about-memory/support-what-does-computer-memory-do

Mitchell, B. (2019, November 14). *Network Interface Cards Explained* . Retrieved from Lifewire: https://www.lifewire.com/definition-of-nic-817866

National Institute of Standards and Technology. (n.d.). *commercial-off-the-shelf (COTS)* . Retrieved from Computer Security Resource Center: https://csrc.nist.gov/glossary/term/commercial_off_the_shelf

NortonLifeLock Inc. (2020). *What is a router, and how does it work?* . Retrieved from Norton: https://us.norton.com/internetsecurity-iot-smarter-home-what-is-router.html

OffSec Services Limited. (2020). *wireshark Package Description* . Retrieved from Kali Tools: https://tools.kali.org/information-gathering/wireshark

Palo Alto Networks, Inc. (2020). *DHCP Leases*. Retrieved from Palo Alto Networks: https://docs.paloaltonetworks.com/pan-os/8-1/pan-os-admin/networking/dhcp/dhcp-addressing/dhcp-leases

Raspberry Pi Foundation. (2020). *Raspberry Pi 4 Tech Specs*. Retrieved from Raspberry Pi: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

Raspberry Pi Foundation. (2020). *SD cards*. Retrieved from Raspberry Pi: https://www.raspberrypi.org/documentation/installation/sd-cards.md

Raspberry Pi Foundation. (n.d.). *What is a Raspberry Pi?* . Retrieved from Raspberry Pi : https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/

Red Hat, Inc. (2019). *What is a Raspberry Pi?* . Retrieved from opensource.com: https://opensource.com/resources/raspberry-pi

Red Hat, Inc. (2019). *What is open source?* . Retrieved from opensource.com: https://opensource.com/resources/what-open-source

Rouse, M. (2016, December). *Linux operating system*. Retrieved from TechTarget: https://searchdatacenter.techtarget.com/definition/Linux-operating-system

Rouse, M. (2020, February). *internet of things (IoT)*. Retrieved from TechTarget: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

Saputri, T. A., & Rofiq, S. (2018). Designing Smarthome Through Internet Networking Using Raspberry Pi Computer. *International Journal of Information System and Computer Science*, 92-101.

Sharpened Productions. (2020). *SD*. Retrieved from TechTerms: https://techterms.com/definition/sd

SharpenedProductions. (2020). *IP Address*. Retrieved from TechTerms: https://techterms.com/definition/ip_address

Software in the Public Interest, Inc. (n.d.). *The Debian GNU/Linux FAQ* . Retrieved from Debian: 2020

Swinhoe, D. (2019, February 19). *What is a man-in-the-middle attack? How MitM attacks work and how to prevent them* . Retrieved from CSO Magazine: https://www.csoonline.com/article/3340117/what-is-a-man-in-the-middle-attack-how-mitm-attacks-work-and-how-to-prevent-them.html

Techopedia Inc. (2013, January 21). *Headless Computer*. Retrieved from Techopedia: https://www.techopedia.com/definition/27269/headless-computer

Techopedia, Inc. (2011, August 18). *Job*. Retrieved from Techopedia: https://www.techopedia.com/definition/3377/job

The Editors of Encyclopaedia Britannica. (2020). *Internet service provider*. Retrieved from Encyclopedia Britannica: https://www.britannica.com/technology/Internet-service-provider

The Linux Information Project . (n.d.). *Command Definition* . Retrieved from The Linux Information Project : 2004

The Tcpdump Group. (2020). *Welcome*. Retrieved from Tcpdump: https://www.tcpdump.org/

Tovar, J. C., Hoyer, J. S., Lin, A., Tielking, A., Callen, S. T., Elizabeth Castillo, S., . . . Gehan, M. A. (2018). Raspberry Pi–powered imaging for plant phenotyping. *Applications in Plant Sciences*.