Implementing a Demilitarized Zone Using Holistic Open Source Solution

Dr. Chafic BouSaba, Guilford College

* Joined Guilford College in January 2008 * Serves as Assistant Professor in the Computing Technology and information Systems. * Cybersecurity major coordinator

Implementing a DeMilitarized Zone Using Holistic Open Source Solution

Abstract

Cybersecurity continues to be a growing priority for organizations of all sizes and industries. The threat landscape continues to rapidly evolve producing disastrous cyber attacks that are crippling their targets and debilitating the economy. These attacks continue to increase in frequency, scale, sophistication, and severity of impact. New attack vectors are persistently emerging and new exploit techniques are constantly gaining widespread adoption. Small businesses continue to experience lack of adequate solutions and resources to defend against and repel these attacks. We present a holistic open source software and hardware solution, that successfully implements securing the network architecture by using the "defense-in-depth" approach that ensures the elimination of single or dual point of potential vulnerability within a network.

The Protectli FW108120 firewall, referred to as "The Vault" is used as first line of defense. It is a low power, fanless, durable, customizable small form factor PC. It utilizes a Celeron J1900 processor, 8 GB of DDR3 memory, a 120 GB mSATA SSD, and 4 Gigabit Ethernet ports. The community version of pfSense firewall will be utilized to run on the firewall. pfSense is a free and open source firewall and router, based on FreeBSD, that also features unified threat management (UTM), load balancing, and multi zone setup. pfSense implements, maintains, and polices our multi-zone topology, which is formed of a DeMilitarized Zone (DMZ), a trusted zone, and an untrusted zone. The DMZ achieves defense in depth by adding an extra layer of security beyond that of a single perimeter, separating an external network from directly referencing an internal network, and isolating a particular machine within a network.

Thought the DMZ concept is not new, implementing it using stacked Single-Board Computers (SBC), offers an affordable and flexible, yet secure, network architecture specifically for startups, small businesses, an expansion office, or even home office. The selected single board computers are a combination of third generation Raspberry Pis and Rock64s. These boards provide a remarkable computational power, low energy consumption, light weight, a compact size secure solution, and resourceful community support. These boards will form the DMZ network servers and host services such as a Network Intrusion Detection System (NIDS) using Snort, a selection of honeypots for Secure Shell (SSH), web applications, packet sniffing, private web browsing capabilities via TOR (The Onion Router), LAMP (Linux, Apache, MySQL, PHP or Python or Perl) server, Virtual Private Network (VPN) server, and protected browsing via proxy service. The main goal of this educational project is to leverage the total holistic integration of open source hardware and software to provide an affordable and portable solution that could be promptly deployed in case of an emergency, as a part of an incident response plan (IRP), or in case it is needed for testing purposes. Implementing this project provides valuable hands-on security experience and best practices in network architecture and configuration. Additional security features, both in hardware and software, were added to the single-board computers to add additional hardened security layers.

Introduction

In today's business environment, companies of all sizes, need to be connected to the internet for many reasons. Internet connection allows access to the tools and web apps used to collaborate effectively within teams and with clients, improves productivity, and lowers budget expenses.

Internet connection enables telecommuting and remote work, which are becoming increasingly common via virtual office environment, video conferencing, and data storage in the cloud. A fast and reliable connectivity is a must to improve responsiveness for customers or clients. A fast internet connection enables "heavyweight" multimedia emails, and visually rich websites. Video streaming and Voice over Internet Protocol (VoIP) are also essentials for business success. The constantly growing amount of data requires a reliable internet connection that is vital for offsite data backup which simplifies business continuity and disaster recovery plan.

In this increasingly connected global economy, businesses need a way to not only separate the good internet traffic from the malicious, but also to keep their internal network securely isolated from the public. Nonetheless, the internal network also needs to have access to the internet in general for web browsing, email, and remote work, as well as other potential uses but yet prevent unwanted traffic from outside. Furthermore, a company needs to protect itself from criminals and hackers who are continually trying to infiltrate the network for whatever their motivation is. Large corporations can afford robust protection with dedicated and trained Information Technology (IT) staff, in addition to a next-generation security platform of firewalls and other security appliances. Smaller businesses, however, may not be able to afford the same protection, or could afford it but might not consider it as a good Return on Investment (ROI) and would rather take the chance that they would not get hacked. Also, if a catastrophic event happens and security is down while IT works to restore the network, there should be a backup plan or emergency security system that could quickly be deployed. This project looks to provide answers to both situations by providing a low-cost solution based upon inexpensive Small Business Computer (SBC) which have come into popularity thanks to the Raspberry Pi foundation.

Network Planning and DMZs

In order to protect the business' network, security must be a priority and a planned outcome. For home networks a simple firewall, along with anti-virus and anti-malware software is usually considered adequate security as it is simply a few users who browse the web, play online games, send and receive email, and stream videos. This solution is inexpensive, relatively easy to configure, and requires little maintenance on the part of the user. However, this solution is considered a very low security solution and has no potential for growth and expansion. Small and large businesses must consider securing their private networks from inside and outside security risks, how to control internet access to and from their internal networks, and how to allow public access to resources the company provides while preventing the public from accessing the internal networks [1]. To accomplish this, network architects usually divide the network into zones. The untrusted zone is for all incoming internet traffic which is untrusted by default, just before it goes through the network's firewall. The trusted zone is for any traffic that is on the internal network. Any resources that the business provides for public consumption, like HTTP requests to the company's webserver, does not need to be on the internal network for security purposes. In order to accommodate that traffic a demilitarized zone is set up with computers that host the web servers as well as machines that provide network security functions like an Intrusion Detection System (IDS), honeypots to draw unwanted attention away from the internal network, and also proxy servers which act as a gateway between the trusted zone and untrusted zone.

Hardware

The initial portion of the project is based off a tower consisting of four Raspberry Pi 3B SBCs, see Figure 1. Each Pi has a 32 GB microSD card for both OS and storage of logfiles and other necessary configuration files, see Figure 2. There were many factors that influenced our decision. First factor was its cost. A single Raspberry Pi at the time of the build was $35 US dollars so building a cluster would be relatively inexpensive as compared to an inexpensive computer. It is also inexpensive enough that keeping a spare board or two in case of board failure would not be a financial burden. Another factor is that it is also easily available through various sellers. The board also has an excellent amount of support from a foundation that maintains its Linux distribution and a large community base that can provide answers if a problem arises.

Figure 1 - Raspberry Pi 3B

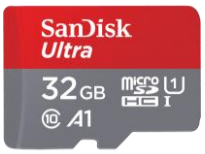

Figure 2 - 32 GB microSDHC Card from Sandisk

There are some concerns with utilizing Raspberry Pi 3B SBCs. The main concerns is its limited amount of memory. One gigabyte seems to be adequate; however, the graphics memory is shared with the rest of the operating system (Raspbian defaults to 64 MB of memory used). For this project, the amount of graphic memory was reduced to 16 MB of RAM so that more memory is allocated to system operation. Further memory savings was achieved by running Raspbian Lite which is a console only version of the Raspbian Linux since they would be ran headless and only accessible via a SSH session. Another concern is the speed of the Pi's onboard Ethernet adapter. While it is rated for 10/100, actual speeds can vary from as low as 64 Mbps to 92.3 Mbps. By utilizing a USB to gigabit Ethernet adapter, see Figure 3, transfer rates reached a better throughput of 220 Mbps [2], which facilitates the deployment of memory dependent security software such as an IDS/IPS on the system.

Figure 3 - USB 3.0 Gigabit Ethernet From Trendnet

Heat is another factor to be concerned with Raspberry Pi 3 [3]. Its CPU runs hot so the use of a heat sink is a must to keep the units from overheating, see Figure 4. Also, the use of an open platform case allows the heat to dissipate quicker as well. The case selected was a dog bone style open platform case that has many advantages. With the open sides of the case, air can flow over the SBCs and let the heatsinks dissipate heat better. This style of case allows easier access to the GPIO pins, so cables can be directed more easily, and mounted devices would have more room. This style of case also allows for modularity and for quick removal of any of the planes to give access to the SBC.

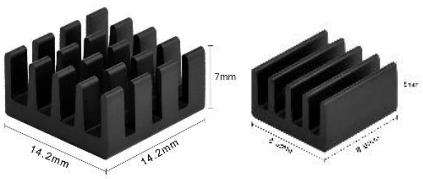

Figure 4 - Black Aluminum Heatsink Coolers for Raspberry Pi from Mudder

Power is supplied by a 60-watt 6 port USB charger, see Figure 5. This allows for a single power cable to supply power to all the boards. All can be powered down as one or each board can be powered down individually by pulling the USB cable from the charger (or Pi).

Figure 5 - RAVPower 60W 6-Port USB Quick Charger

Part of the scope of this project includes utilizing encryption for security purposes. While generating keys via software can be done, a hardware encryption module would be more secure. A Zymkey from Zymbit, see Figure 6, would provide that security, being able to generate and store up to 3 keys and be able to encrypt the file systems of the Raspberry Pi that it is installed on for further security.

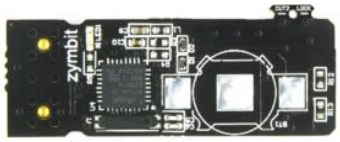

Figure 6 - Zymkey from Zymit

With the limited resources available to the Raspberry Pi 3 in terms of memory and computational resources, a secondary SBC tower was constructed. While utilizing the same case, storage, and miscellaneous cables, the SBC chosen to populate this tower was the Rock64 by Pine64, see Figure 7. The concern of memory was alleviated due to the unit having 4 GB of Low Power Double Data Rate Synchronous Dynamic Random Access Memory. (LPDDR3) versus the Raspberry Pi 3B's 1 GB of LPDDR2 memory. Its processor is also faster, running at 1.5 GHz compared to 1.2 GHz of the Pi. The combination of the two factors will allow for the running of more demanding software like a LAMP server (Linux Apache MySQL PHP), or more than one security application on a device. For a comparison of the two SBC units, see table 1.

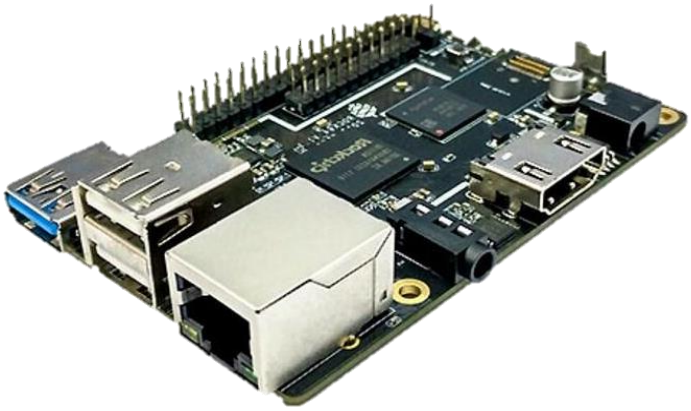

Figure 7 - Rock64 by Pine64

The final component our proposed, open-source, security system is a firewall. The firewall needs to have at least a 4 Ethernet ports. The first port is for the WAN connection, the second port is for untrusted network traffic, the third port is for trusted network traffic, and the fourth port is for the DMZ zone to which the Pi tower would be connected to it. The Protectli FW108120 4 port firewall was chosen, see Figure 8. It is an affordable firewall at $379. It is a fanless customized PC utilizing a Celeron J1900 processor, 8 GB of DDR3 memory, a 120 GB mSATA SSD, and 4 Gigabit Ethernet ports. The community version of pfSense firewall which is free and open source has been utilized.

Figure 8 - Protectli FW108120 4 port firewall

Table 1 – Raspberry Pi 3B vs Rock64 Hardware Specifications

	Raspberry Pi 3B	Rock64
CPU	1.2 GHz Quad Core A53 Broadcom	1.5 GHz Quad Core A53 Rockchip
Memory	1 MB LPDDR2 (shared with video)	1, 2, or 4 MB LPDDR3 (shared with video)
Ethernet Ports	10/100 fast ethernet	10/100/100 gigabit ethernet
Wireless	2.4 GHz 802.11n, Bluetooth 4.0	n/a
USB Ports	4 USB 2.0, 1 USB micro B (for power)	2 USB 2.0 A, 1 USB 3.0
Storage Medium	MicroSD (though can boot from USB flash drive or external hard drive with a change in boot configuration)	MicroSD, eMMC
Power Supply	5V, 2.5 A max (micro USB B)	5V, 3A max (3.5 mm Barrel Connector)

Software and Configuration

Before any security software or applications can be installed, the first major software decision must be made is which operating system should be installed on the SBCs. For the Raspberry Pi3 the choice was simple. Raspbian Lite [6] was downloaded and installed. The distribution is well supported through the Raspberry Pi foundation and the community at large. The current version of Raspbian Lite is based on Debian Stretch, the current stable version of Debian. It loads a minimal version of the operating system and forgoes graphical desktops which not only uses more system memory, but also allows more storage space. Raspbian is not available for the Rock64 as it is based upon the Rockchip RK3328 A53 CPU, so an alternative distribution needed to be selected. As the Rock64 is a newer SBC, finding a distribution that works well with the board was more involved. After examining multiple distributions like Ubuntu's Xenial and Artful minimal install, Debian Jessie minimal install, Armbian (based on Ubuntu Xenial but with a legacy kernel), and DietPi (based on Debian Stretch), DietPi was settled on [7]. Not only is it a lightweight distribution like Raspbian Lite, but it also has a configuration utility to easily install prepackaged utilities like LAMP server, VPN client, and others.

Well protected networks include a variety of security tools. Traffic both in and outbound is analyzed for malicious traffic by intrusion detection systems (IDS). Intrusion protection systems (IPS) does a slightly similar job to the IDS but also will send alerts when detecting questionable traffic. For these jobs, Snort (along with PulledPork to download signature updates and Barnyard2 to parse Snort's binary data) is installed on Raspberry Pi #1 [8]. An alternative IDS package was investigated. Suricata is similar to Snort in the way it uses rules to sort traffic but differs from Snort by being multithreaded where Snort is single-threaded. This gives it the potential for a balanced load across all the processors. However, initial tests show that Snort has a better throughput than Suricata. With rules loaded, Suricata ran up to 200 MB per second while Snort achieved 894 MB per second [9].

Even with IDS/IPS, intruders can get through. To diverge their attention away from the network, honeypots are used to capture their attention and make them believe they are in the network. Raspberry Pi #2 and #3 contain honeypots for different types of traffic. RPI #2 contains Cowrie which is a honeypot for SSH and telnet (though telnet traffic is blocked from our trusted network) [10]. For HTTP, FTP, and SMTP traffic, InetSim is installed [11]. In case someone does gain SSH access in Cowrie, outbound traffic could be directed to InetSim so that outbound requests would not alert the attacker as they are normally not allowed and will time out [12].

While the towers would mostly run unattended, there will be situations where access to the DMZ would need to be provided remotely as the units are being ran headless. To do so, OpenSSH is being used on the Raspberry Pi #4. To provide an extra layer of security, openSSL key encryption is being handled by a Zymkey where the private key can be stored instead of on the SD card [13]. To prevent legitimate traffic from being routed to the Cowrie honeypot, the SSH ports on all other devices are changed to a non-standard port. This way, once logged into RPI 4, the other devices will be accessible.

Two more security issues remain to be addressed. The first issue is how to allow remote users in the untrusted zone to be able to access the trusted network while preventing unwanted traffic back to the trusted network. Options for VPN placement are either placing the VPN outside the firewall which opens the machine to untrusted traffic, or behind the firewall in the DMZ. The firewall must be configured to allow inbound and outbound VPN traffic to flow between the server and client [14]. For this project, the VPN is located on the Rock64 #3 which is configured with OpenVPN via the PiVPN package [15].

The second security issue is how to allow requests for web pages to leave the trusted network and be able to receive responses from the untrusted network. To protect the trusted network, Rock64 #4 is configured as a proxy server. Squid is used direct web traffic between the trusted and untrusted zones while SquidGuard is used for blacklisting web pages the business deems inappropriate [16] [17]. The Rock64's Gigabit Ethernet controller provides enough bandwidth for quick access.

The other two Rock64 systems are being put to non-security use. Rock64 #1 and #2 are put to more mundane use as a LAMP server (#1) and as a mail server (#2). The LAMP configuration was Apache2, MySQL, and PHP which DietPi has a ready-to-run package. Email service on #2 has to be manually downloaded and installed as there is not a ready-to-run package in DietPi.

Table 2 summarizes the operating systems and the software packages used, their functionality, source, and the device installed on.

Table 2 – Software Used

Software	Function	Source	Device Installed On
Operating System			
Raspbian Lite (Stretch)	Raspberry Pi OS	www.raspberrypi.org/downloads/raspbian/	Raspberry Pi 1-4
DietPi (Stretch)	Rock64 Linux OS	www.dietpi.com	Rock64 1-4
pfSense	Protectli Firewall	https://protectli.com/ https://www.pfsense.org/	Protectli
Utilities			
Snort	IDS	www.snort.org	Raspberry Pi 1
Cowrie	SSH/Telnet Honeypot	www.micheloosterhof.com/cowrie/	Raspberry Pi 2
InetSim	HTTP/FTP/SMTP Honepot	www.inetsim.org	Raspberry Pi 3
OpenSSH	SSH Server	https://www.openssh.com	Raspberry Pi 4
LAMP server	Web server	Included with DietPi setuip	Rock64 1
Postfix, Squirrel Mail, SpamAsassin, Seive	Mail Server and spam filter	https://samhobbs.co.uk/raspberry-pi-email-server	Rock64 2
PiVPN (Open VPN & tools)	VPN	www.pivpn.io	Rock64 3
Squid and Squidguard	Proxy Server	www.squid-cache.org www.squidguard.org	Rock64 4

Network Configuration

The final part of the project is the firewall and configuration. The firewall hardware is a Protectli 4 port Firewall model FW10812. The four ports were configured as WAN, untrusted LAN (192.168.1.0/24), DMZ (192.168.100.0/24), and trusted network (10.0.0.0/32 – though the subnet could be adjusted for the number of network devices on the network). Each SMB is configured for a static IP address to allow ease of port forwarding the required traffic to the requisite SBC. As both towers can be run independent of each other, each tower is equipped with a five port Gigabit Ethernet switch. If more than one tower is used, then either a third switch, or a larger single switch must be used in order to have enough Ethernet ports available. Figure 9 presents the network architecture and configuration.

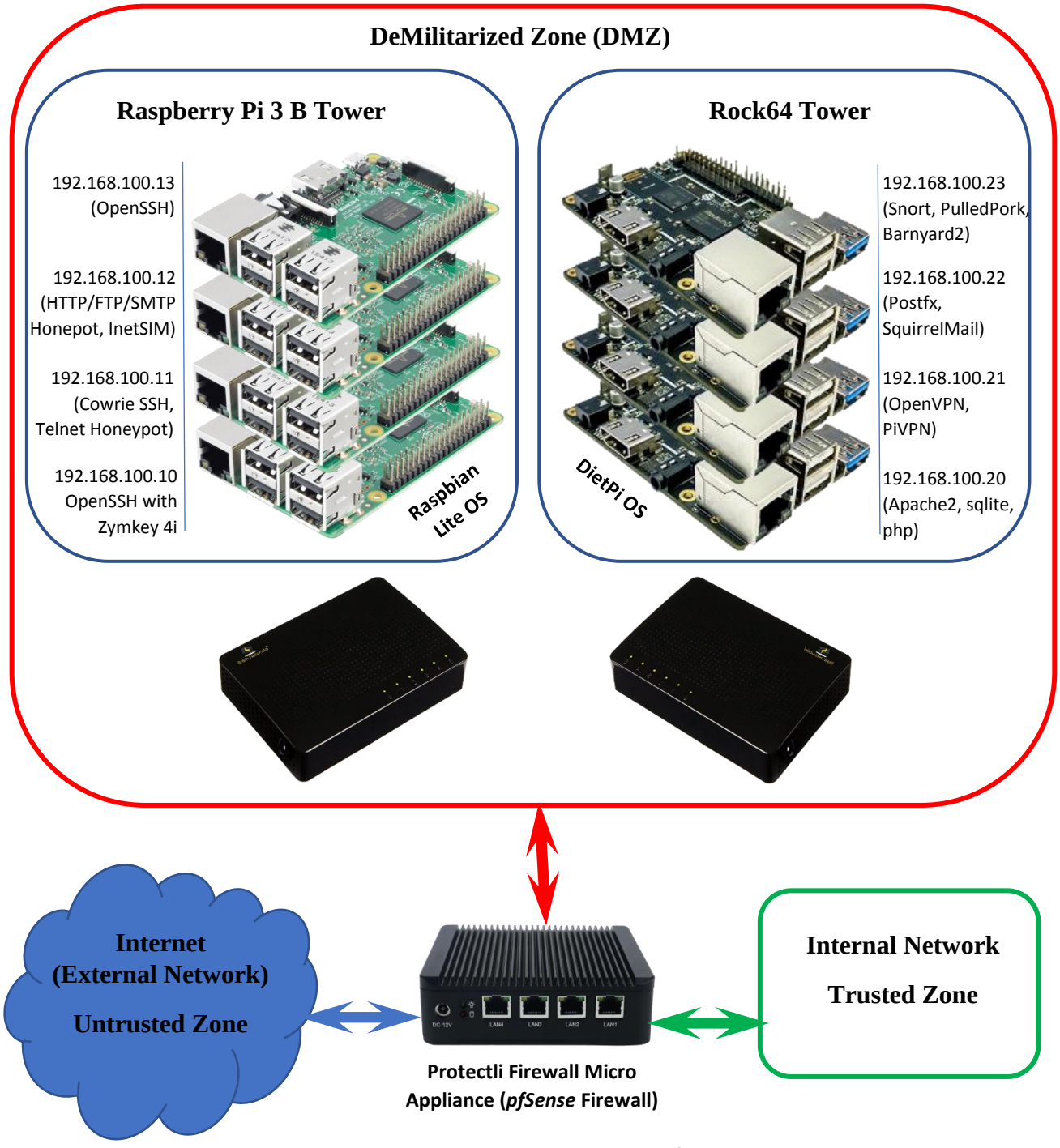

Figure 9 - Network Architecture and Configuration

Conclusion and Future Work

The project was successfully built and tested. The project's total cost was 1,190.00 US dollars distributed over three main areas: 400 for the Pi tower, 340 for the Rock64 tower, and 450 for the Firewall, switches, and Ethernet cables. Three students worked on the implementation,

configuration, and documentation for a total of 100 hours during an academic semester. The students reported that they learned by doing hands-on labs and testing, and enjoyed this project. This project is suitable as a capstone or senior group project. Testing results proved that this project is suitable for securing the network of a startup company or a small business. This project could also be deployed in case of a disaster recovery or as a part of an Incident Response Plan (IRP) after a cyberattack. Future work involve additional security packages for network monitoring, filtering, and encryption.

References

[1] H. Flynn, "Designing and Building Enterprise DMZs," Syngress, 2006.

[2] Mat, "Rasberry Pi network speed test: RPI2, RPI3, Zero, ZeroW (LAN&WIFI)," NotEnoughTech.com, 5 April 2017. [Online]. Available: http://www.notenoughtech.com/raspberry-pi/raspberry-pi-internet-speed/. [Accessed 1st February 2018].

[3] J. Geerling, "Getting Gigabit Networking on a Raspberry Pi 2, 3, and B+," 16 February 2015. [Online]. Available: https://www.jeffgeerling.com/blogs/jeff-geerling/getting-gigabit-networking. [Accessed 1 February 2018].

[4] Multiple, "Prevent SD-Card Corruption (forum)," RaspberryPi Foundation, 9 March 2013. [Online]. Available: https://www.raspberrypi.org/forums/viewtopic.php?t=36533. [Accessed 14 January 2018].

[5] Multiple, "Consistenct Corruption of SD Card," Raspberry Pi Stackexchange, 9 October 2017. [Online]. Available: https://raspberrypi.stackexchange.com/questions/51884/consistent-corruption-of-sd-card. [Accessed 13 January 2018].

[6] R. P. Foundation, "Raspbian Download Page," Raspberry Pi Foundation, [Online]. Available: https://www.raspberrypi.org/downloads/raspbian/. [Accessed 28 November 2017].

[7] D. Knight, "Diet Pi," Diet Pi, 2017. [Online]. Available: http://dietpi.com. [Accessed 10 January 2018].

[8] Cisco, "Snort," Cisco, 2018. [Online]. Available: https://www.snort.org. [Accessed 16 December 2017].

[9] G. Khalil, "Open Source IDS High Performance Shootout," SANS Institute, 2 February 2015. [Online]. Available: https://www.sans.org/reading-room/whitepapers/intrusion/open-source-ids-high-performance-shootout-35772. [Accessed 28 January 2018].

[10] M. Oosterhof, "Cowrie Honeypot," Security Intelligence, 2017. [Online]. Available: http://www.micheloosterhof.com/cowrie/. [Accessed 10 January 2018].

[11] T. &. E. M. Hungenberg, "Welcome to INetSim HomePage," 2017. [Online]. Available: http://www.inetsim.org/. [Accessed January 2018].

[12] B. Ruberg, "Small Scale Honeynet with Raspberry Pi," Redpill-Linpro, 19 December 2016. [Online]. Available: https://www.redpill-linpro.com/sysadvent/2016/12/19/raspberry-pi-honeynet.html. [Accessed 15 November 2017].

[13] E. Fairchile, "OpenSSL: Apache Setup, Generating CSR," Zymbit, 7 December 2016. [Online]. Available: https://community.zymbit.com/t/openssl-apache-setup-generating-csr/107. [Accessed 17 January 2018].

[14] "VPNs and Firewalls," Microsoft, [Online]. Available: https://technet.microsoft.com/en-us/library/cc958037.aspx. [Accessed 18 January 2018].

[15] "PiVPN," The PiVPN Project, [Online]. Available: https://pivpn.io. [Accessed January 2018].

[16] "Squid-cache.org Optimizing Web Delivery," Squid-cache.org, [Online]. Available: http://www.squid-cache.org/. [Accessed January 2018].

[17] "Squidguard," Squidguard, [Online]. Available: https://www.squidguard.org. [Accessed January 2018].