A Configuration, Exploitation, and Investigation of a Network.

Brian Thomas Carr

Utica College

Author Note

Brian T. Carr, Department of Economic Crime & Justice Studies, School of Business & Justice Studies, Utica College. Correspondence concerning this article should be addressed to Brian T. Carr, Department of Economic Crime & Justice Studies, Utica College, 1600 Burrstone Road, Utica, New York 13502. E-mail: btcarr@utica.edu

Abstract

This article contains information on network configuration, ethical hacking, and digital forensics incident response. The Author's intention for this article is to provide others interested in cybersecurity with a valuable educational and technical resource. This article discusses the configuration of a network of Raspberry Pi computers running Linux Operating Systems, the configuration of a malicious File Transfer Protocol server with the purpose of capturing user credentials, and a forensic investigation of the incident. While there are countless ways to configure, exploit, and forensically investigate networks, this project will give insight into one single configuration with one single procedure.

Keywords: network, Local Area Network (LAN), File Transfer Protocol (FTP), ethical hacking, digital forensics, network forensics, firewall, logging.

Table of Contents

List of Illustrative Material

Figures

A Configuration, Exploitation, and Investigation, of a Network.

The purpose of this article is to provide those interested in cybersecurity with a resource that is both educational and technically informative. This project will begin with a network configuration and the configuration of each device on the network. Then, the Author will continue on to preform malicious activity on the network. And finally, the Author will perform a forensic investigation on the network and its devices. This article will show a single implementation of a variety of networking and security technologies. Additionally, this article is primarily focused on the Linux operating system, and Linux compatible devices with ARM processors.

Network Configuration

The network architecture of this project consists of two Raspberry Pi 3B+ computers, one Raspberry Pi B computer, and a Cisco 8-port ethernet Cat-6 network switch. Each Raspberry Pi is connected to the switch with a one-foot Cat-6 ethernet cable. Each Raspberry Pi was configured to connect their eth0 network interface card to the 10.0.0.X subnet with a 255.255.255.0 netmask. Each Raspberry Pi on this network was configured with Iptables to log all input traffic and its IPv4 information.

UbuntuMate Configuration

The UbuntuMate Raspberry Pi configuration started with downloading the ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img,xz on a MSi GS63 laptop from https://ubuntu-mate.org/download/. After the download, the Author then extracted the disc image file. Once extracted, the Author then loaded the IMG file ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img to a new 16GB microSD card using an MSi GS63 laptop. Upon first power-on, the Author was presented with multiple prompts of the UbuntuMate configuration. The first prompt

presented to the Author asked for the language configuration, the Author selected English and continued on to the next prompt. Next, the Author was presented with a prompt to select the keyboard settings, the Author selected “en-us” which is an abbreviation for English United States. Next, the Author was presented with a prompt to configure the network settings. The Author chose to postpone network configurations until a later time. The next prompt asked for the username, system name, and domain name. The Author chose to name the username and the system name “ubuntu” and chose not to enter a domain name as the system was not running on a valid domain. The last prompt asked for the password configuration, the Author chose to use the password: P@$$W0rd. After entering the initial configuration settings, the Author then rebooted the system to implement those changes.

Upon first power-on, the Author logged into the system using the username and password selected in the initial configuration. Once logged in, the Author then started a terminal session. The Author then proceeded to perform a system update and upgrade using the command : sudo apt-get update && sudo apt full-upgrade. The Author’s execution of the system update and upgrade can be seen in *Figure 1.* After preforming the update and upgrade, the Author then installed the text editor Vim with the command: sudo apt-get install vim. This installation can be seen in *Figure 2*. The armhf build of UbuntuMate on Raspberry Pi does not have the network interface card named eth0 by default. To remediate this issue, the Author renamed the ethernet network adapter. To rename the ethernet adapter eth0, the Author implemented a technique posted by Luis Godinez on raspberrypi.stackexchange.com. (Godinez, 2016).

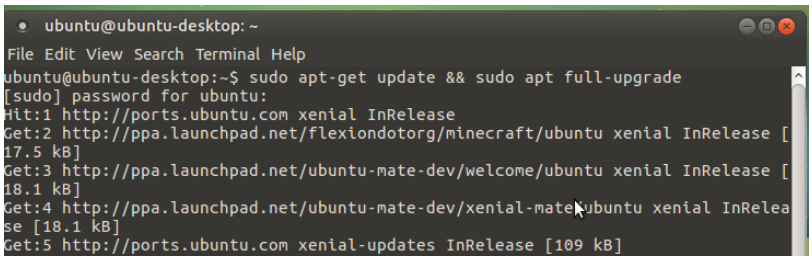

Figure 1: UbuntuMate Update and Upgrade

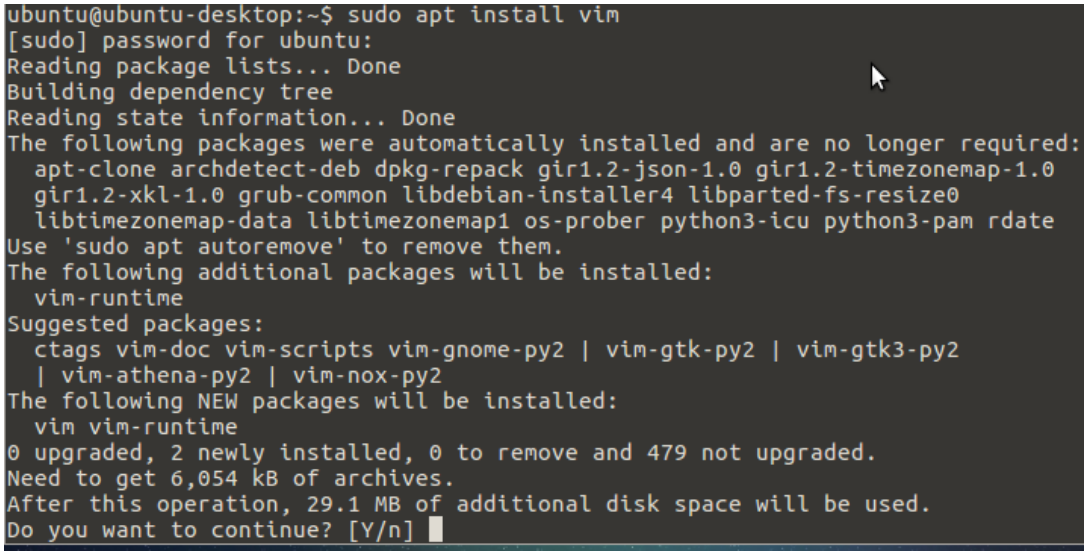

Figure 2: Vim Installation on UbuntuMate Raspberry Pi

To rename the ethernet network adapter, the Author first opened /lib/udev/rules.d/73-usb-net-by-mac.rules in the Vim text editor, this can be seen in *Figure 3*. The Author then changed the “NAME” variable to “eth0” as seen in *Figure 4.* After saving the changes and rebooting the system the newly renamed ethernet network adapter can be seen in *Figure 5*. The name of the ethernet adapter is arbitrary, but for sake of usability the Author prefers to have system ethernet adapters named “eth0”.

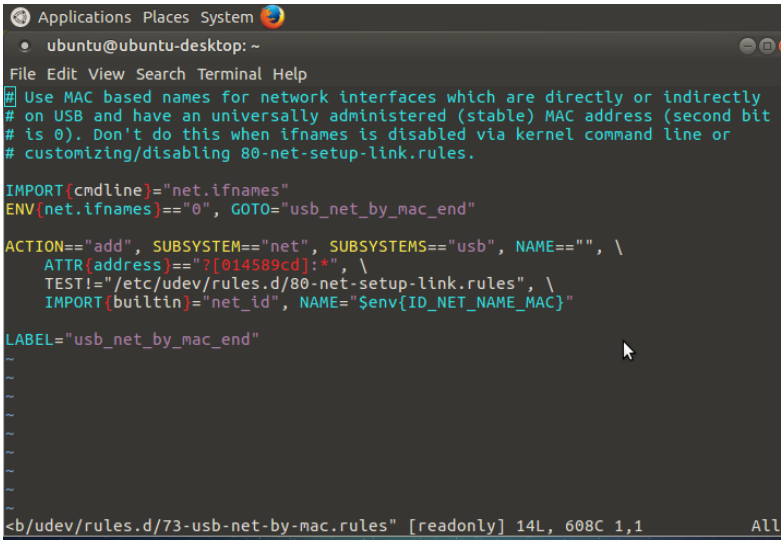

Figure 3: UbuntuMate /lib/udev/rules.d/73-usb-net-by-mac.rules Before Reconfiguration

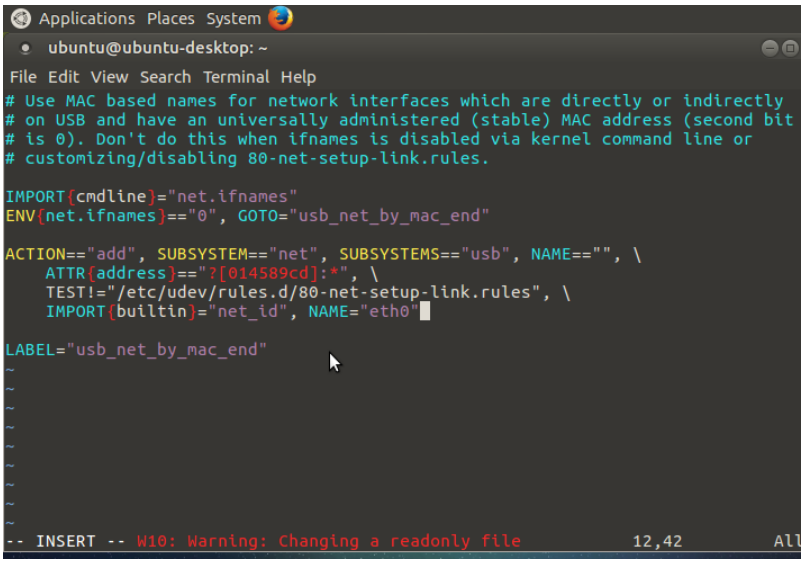

Figure 4: UbuntuMate /lib/udev/rules.d/73-usb-net-by-mac.rules After Reconfiguration

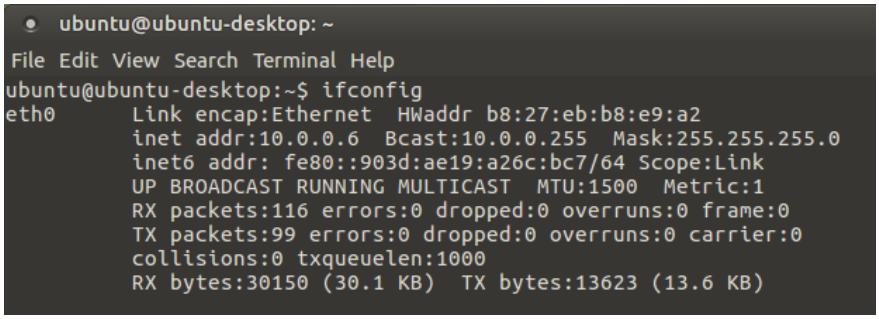

Figure 5: UbuntuMate Ethernet Network Interface Name After Reconfiguration

After the configuration of the ethernet network adapter, the Author continued on to implement logging functionality with Iptables. The logs generated contain both TCP and IP

address data. The Author implemented this with the command: sudo iptables -A INPUT -j LOG --log-ip-options. The Author found this specific utilization of Iptables in Michael Rash's book titled Linux Firewalls. (Rash, 2007, p. 37) The Author can be seen implementing this syntax in *Figure 6.* The LOG switch utilized in the previous command directs Iptables to log all input to syslog, and --log-ip-options switch directs Iptables to log the IP addresses of the connections.

Once the UbuntuMate Raspberry Pi was fully configured and logging functionality was implemented, the Author configured the system on to the Raspberry Pi LAN. Since there was no Dynamic Host Configuration Protocol server running on the LAN, the Author had to manually set the IPv4 address and netmask to communicate with the other Raspberry Pi's. The Author utilized the command: sudo ifconfig eth0 10.0.0.2 netmask 255.255.255.0. The Author's implementation of the previously stated command can be seen in *Figure 7.* At this point, the ethernet network adapter has been configured to communicate on the newly created LAN populated with Raspberry Pi's.

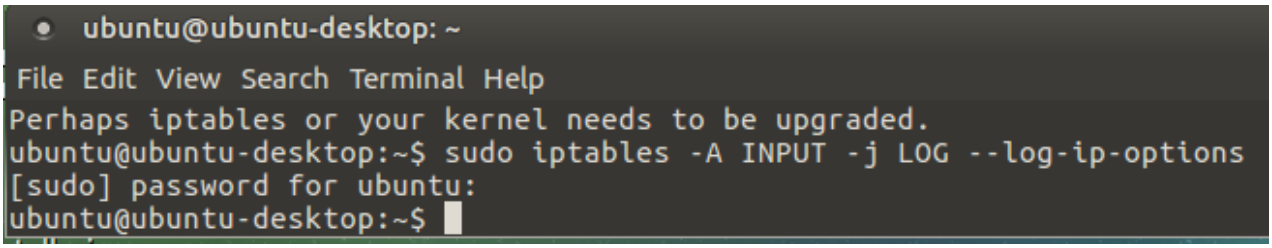

Figure 6: Iptables Command for Logging TCP and IP connections

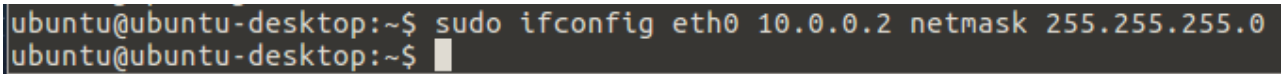

Figure 7: Ethernet Configuration for New Raspberry Pi Network

CentOS 7 Configuration

The configuration of the CentOS 7 Raspberry Pi began with downloading CentOS-Userland-7-armv7hl-RaspberryPI-GNOME-1810-sda.xz from the Princeton download mirror located at http://mirror.math.princeton.edu/pub/centos-altarch/7.6.1810/isos/armhfp/CentOS-

Userland-7-armv7hl-RaspberryPI-GNOME-1810-sda.raw.xz. Once downloaded, the Author continued on to extract the file out of its compressed state. Once extracted, the Author then used Win32 Disk Imager to write the disk image file CentOS-Userland-7-armv7hl-RaspberryPI-GNOME-1810-sda on to a new 32GB microSD card. Once the write was successfully completed, the microSD card was then taken out of the MSi GS63 laptop and inserted into a Raspberry Pi 3B+. Next, the Author connected the display via HDMI, and power via a 2.5A power supply.

Upon first power-on, the Author was presented with system configuration prompts. First, the Author selected English for the language setting and English-US for the keyboard layout setting. Then, the Author skipped the network configuration, and continued on to the privacy settings, where the Author chose to leave the location settings on. Next, the Author configured the Time Zone information for Syracuse, New York. Finally, the Author configured the username and password of the system. The Author chose to name the system: centos, and set the password: 1337P@$$w0rd.

Next, the Author attached the system to the Author's home network. Then the Author ran a system update and upgrade with the command: sudo yum update && sudo yum upgrade. After updating and upgrading the system, the Author implemented logging functionality with Iptables. The logging functionality would send both input connections and IP address information to syslog. The Author found this specific utilization of Iptables in Michael Rash's book titled Linux Firewalls. (Rash, 2007, p. 37) The Author's implementation of Iptables can be seen in *Figure 8*. Once the logging functionality was implemented on the CentOS Raspberry Pi, the Author then configured it for the new Raspberry Pi network. The Author configured network connectivity with the command: sudo ifconfig eth0 10.0.0.3 netmask 255.255.255.0. The Author can be seen

preforming this action in *Figure 9.* At this point, the CentOS 7 Raspberry Pi was configured on the network with logging functionality implemented.

```
[centos@localhost ~]$ sudo iptables -A INPUT -j LOG --log-ip-options
[centos@localhost ~]$
```

Figure 8: CentOS Iptables Syntax for Logging Input and IPv4

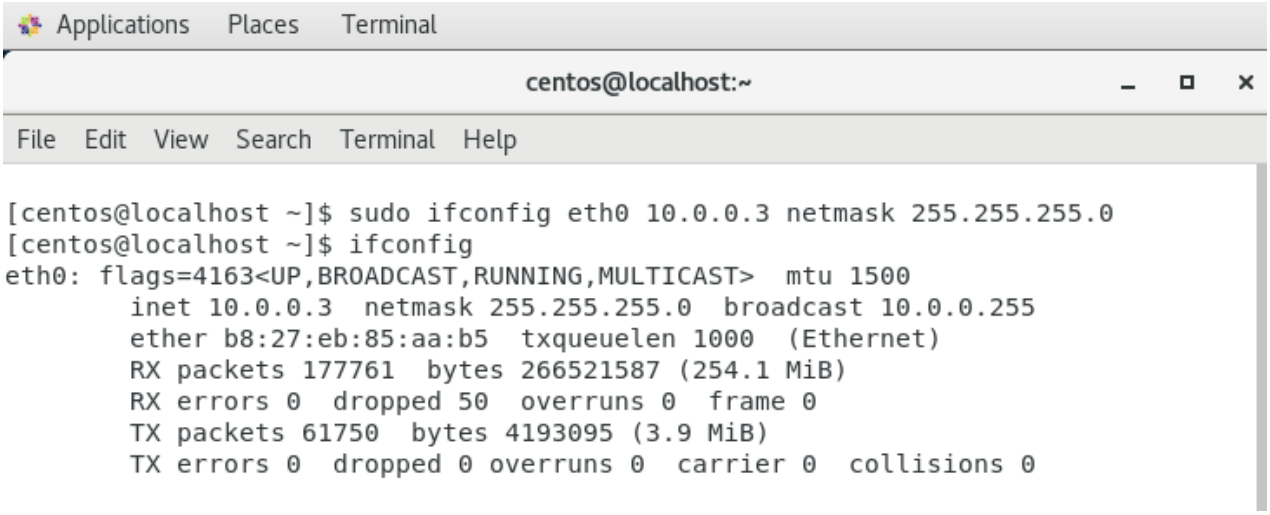

Figure 9: Network Configuration for CentOS Raspberry Pi on Project Network

Raspbian Configuration

The configuration of the Raspbian Raspberry Pi began when the Author downloaded 2018-11-13-raspian-stretch-full.zip from https://www.raspberrypi.org/downloads/raspbian/ to a MSi GS63 laptop. Once downloaded, the Author then extracted 2018-11-13-raspbian-stretch-full from 2018-11-13-raspbian-strech-full.zip. Once the disk image file was fully extracted, the Author then used Win32 Disk Imager to write the disk image file 2018-11-13-raspbian-stretch-full to a new 32GB microSD card. Once successfully written, the Author then inserted the newly imaged microSD card to a Raspberry Pi 3B+. Next, the Author connected a display to the HDMI port, and a 2.5A power adapter to the power port.

On first power-on, the Raspbian operating system started with a default configuration and prompted the Author for time zone and language settings which can be seen in *Figure 10*. The next prompt requested the Author to set a password. The Author chose the password for the

Raspbian Raspberry Pi to be “P@$$W0rd”. The password configuration can be seen in *Figure 11*. The Author was then presented with the prompt seen in *Figure 12*, requesting to reboot the system to finish initializing configuration settings. After rebooting the system, the Author connected the system to the Author’s home network for internet connectivity. Upon connecting the system to the internet, the Author then implemented a system update and upgrade using the command: sudo apt-get update && sudo apt full-upgrade. This command can be seen being executed by the Author in *Figure 13*.

Figure 10: Language and Timezone Configuration for Raspbian Raspberry Pi

Figure 11: Password Configuration for Raspbian Raspberry Pi

Figure 12: Raspbian Reboot After Initial Configuration

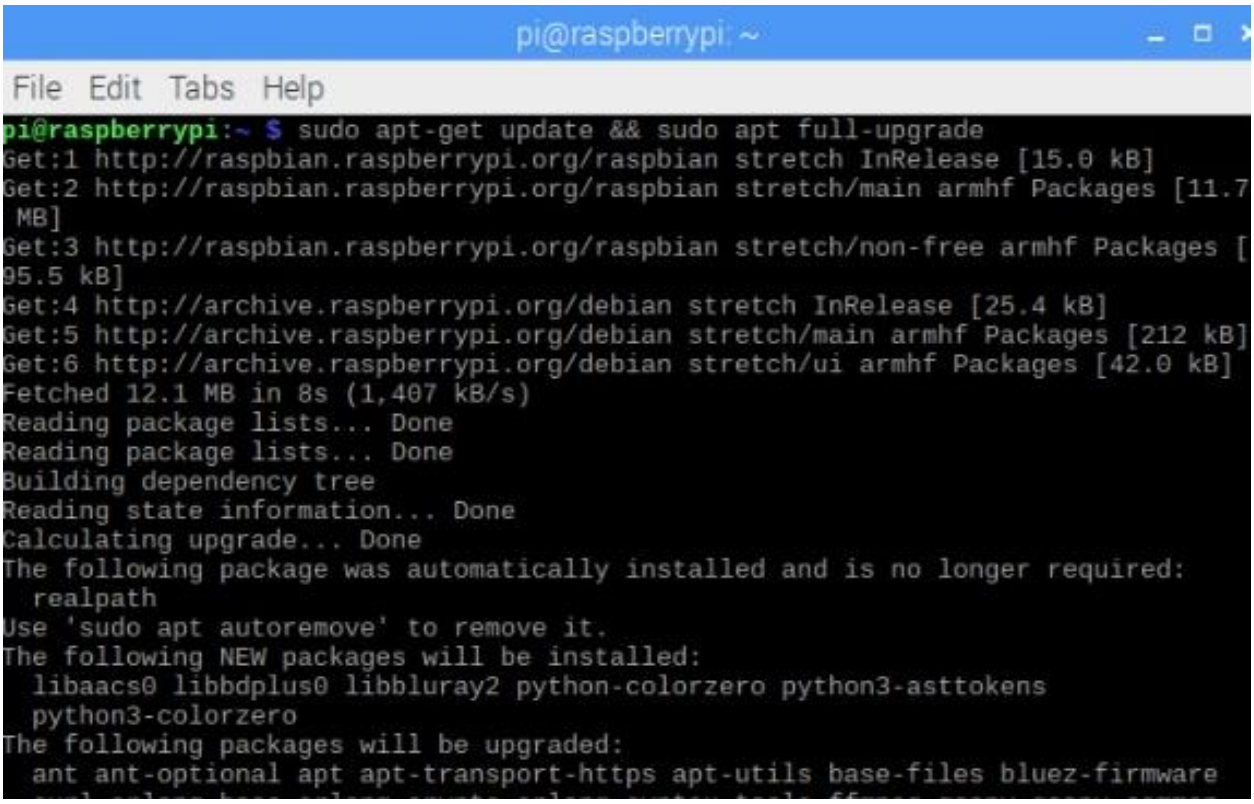

Figure 13: Update and Upgrade Command Executed on Raspbian Raspberry Pi

Next, the Author removed the Raspbian Raspberry Pi from the home network and connected it to the new Raspberry Pi LAN through the Cisco 8-Port switch previously mentioned. The Author implemented logging functionality using the command: sudo iptables -A INPUT -j LOG --log-ip-options. The Author can be seen executing this command in *Figure 14*. With logging functionality now enabled, the Author proceeded to connect the Raspberry Pi to the LAN network using the command: sudo ifconfig eth0 10.0.0.4 netmask 255.255.255.0. The Author can be seen implementing this command in *Figure 14*.

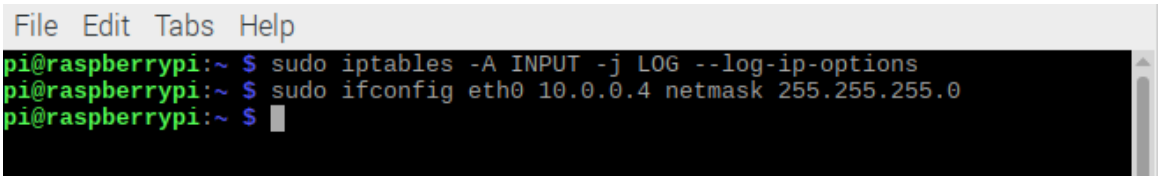

Figure 14: Iptables and Network configuration on Raspbian Raspberry Pi

Malicious Activity

The Author began the network exploitation with the configuration of a Kali Linux Raspberry Pi. Once the Raspberry Pi was fully configured, the Author then proceeded to locate devices on the subnet with Nmap. After locating devices on the subnet, the Author then continued on to configure a malicious file transfer protocol server running on the Kali Linux Raspberry Pi. Then the Author waited for an unsuspecting user on the network to FTP into the machine. For sake of providing an authentic context, the malicious user could pass a note to an unsuspecting employee with the words “Christmas_Bonuses.docx can be found at FTP 10.0.0.19” written on it. The unsuspecting user then went to the FTP server and entered their credentials, providing it directly to the malicious user. Then the Author took the credentials and saved them to a file named credentials.txt.

Kali Linux Configuration

The configuration of the Kali Linux Raspberry Pi began with the Author using a MSi GS63 laptop computer to download kali-linux-2019.1-rpi3-nexmon-64.img from the Kali Linux Arm Image page located at https://www.offensive-security.com/kali-linux-arm-images/. Once downloaded, the Author then used Win32 Disk Imager to write kali-linux-2019.1-rpi3-nexmon-64.img to a new 16GB microSD card with a MSi GS63. Once the microSD card was loaded with the image file, the Author inserted the microSD into a Raspberry Pi 3B+. Next, the Author connected a display via HDMI and power via a 2.5A power supply.

Upon first power-on, the Author was presented with a preconfigured Kali Linux installation which utilized the default credentials username: root, password: toor. Immediately after logging in the Author connected the Raspberry Pi to the Author’s home network. Once connected to the home network, the Author preformed a system update and upgrade using the

command: sudo apt-get update && sudo apt-get upgrade. Once upgraded and updated, the Author continued on to install ImageMagick. ImageMagick is a very versatile program, and in this article the Author will be utilizing its screenshot functionality. ImageMagick can take screenshots using the import command as follows: import <file name>. The Author can be seen installing ImageMagick in *Figure 15*. Next, the Author removed the Kali Linux Raspberry Pi from the Author's home network and inserted it onto the Raspberry Pi LAN. The Author then configured the eth0 network interface of the Kali Linux Raspberry Pi to communicate over the 10.0.0.X subnet, a default internal subnet for a variety of routing devices. The Author configured the Eth0 network interface adapter of the Kali Linux Raspberry Pi with the command: sudo ipconfig eth0 10.0.0.19 netmask 255.255.255.0.

Once the Author configured the ethernet network adapter, the Author proceeded on to preform Nmap scans in hopes of locating any live machines. The first Nmap scan performed by the Author utilized the syntax: nmap 10.0.0.1-10. This scan simply scanned 1000 default ports on every IPv4 address between 10.0.0.1 and 10.0.0.10. The Author can be seen implementing Nmap in this fashion in *Figure 16*. As seen in *Figure 16*, there were three live devices between IPv4 address 10.0.0.1-10.0.0.10. The live IPv4 addresses were 10.0.0.2, 10.0.0.3, and 10.0.0.4. After locating each live machine, the Author moved on to using Nmap to scan each live machine for vulnerabilities. The Author first scanned 10.0.0.2 for vulnerabilities with the command: nmap --script vuln 10.0.0.2. The Author located this specific utilization of Nmap in Patrick Engebretson's The Basics of Hacking and Penetration Testing. (Engebretson, 2013, p. 70) The Author can be seen preforming this scan in *Figure 17*. After scanning 10.0.0.2 for vulnerabilities with Nmap, the Author then proceeded to replicate this procedure for both 10.0.0.3 and 10.0.0.4.

The Author can be seen scanning 10.0.0.3 for vulnerabilities in *Figure 17*, and 10.0.0.4 in *Figure 18*.

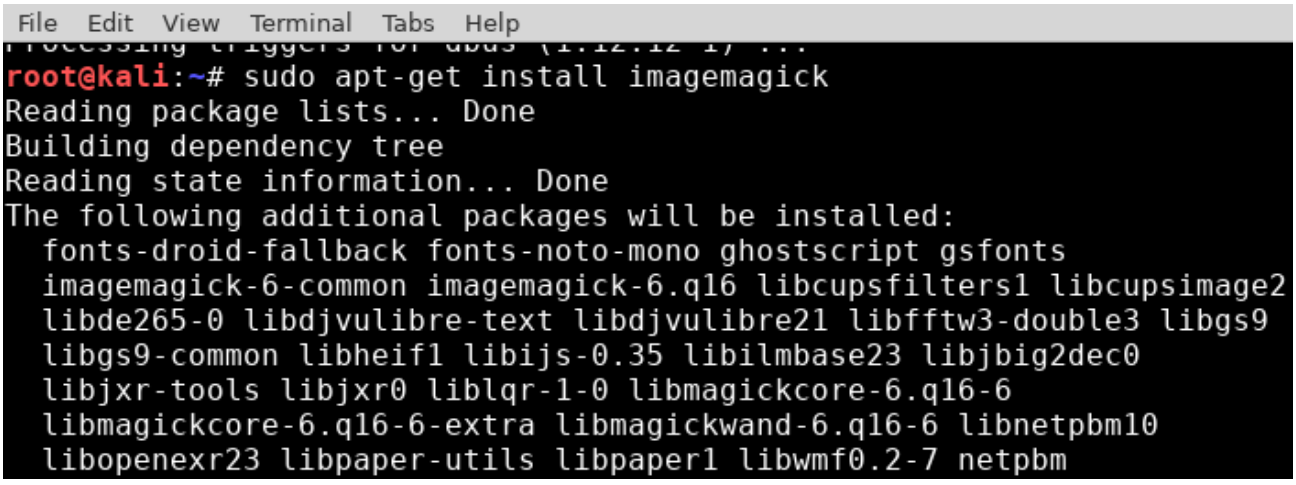

Figure 15: ImageMagick Installation on Kali Linux Raspberry Pi

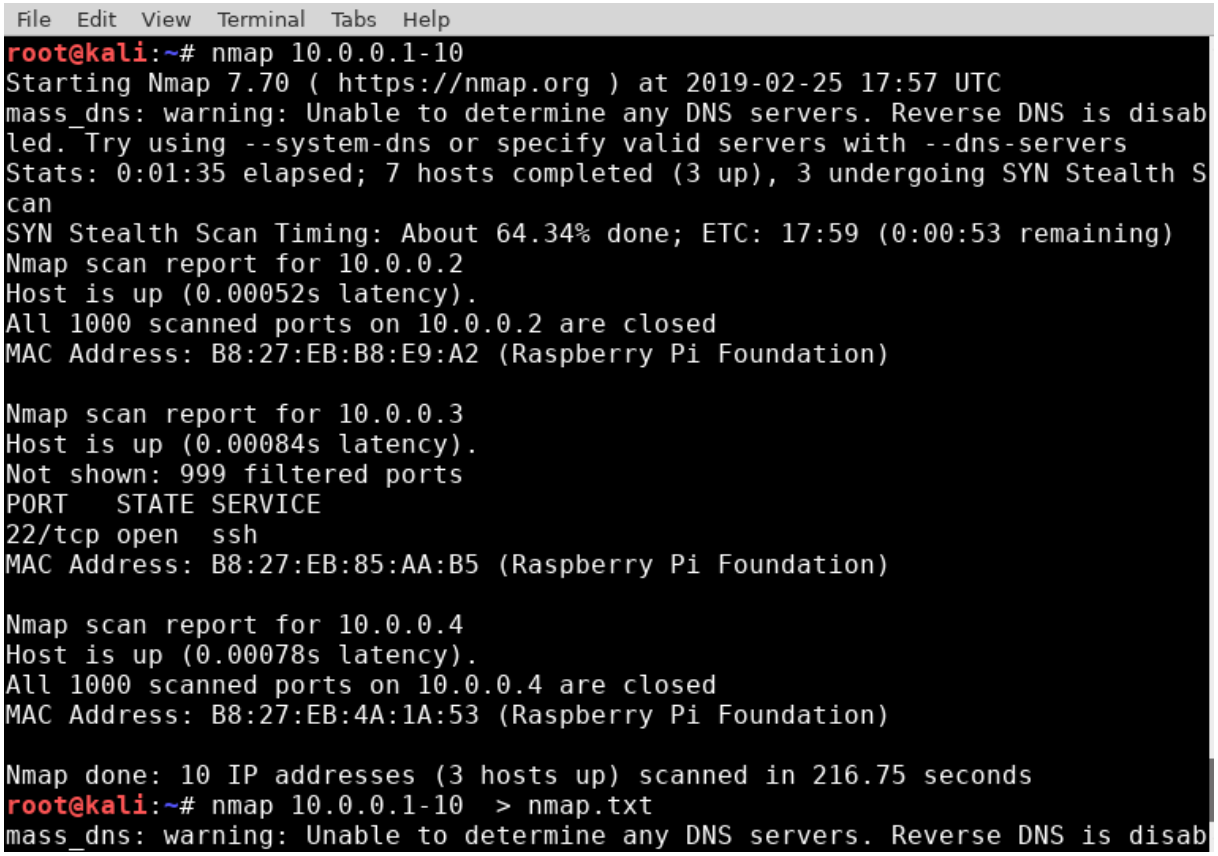

Figure 16: Nmap on Subnet to Locate Live Machines, and Creation of nmap.txt=

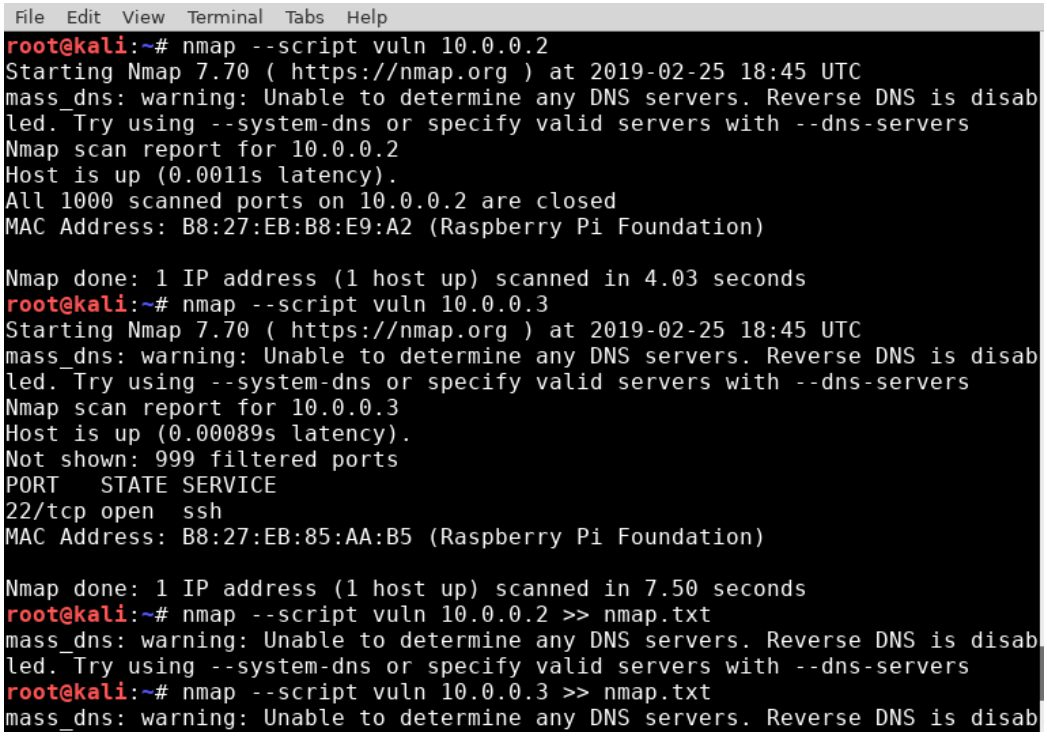

Figure 17: Nmap Vulnerability Script Ran on Live Hosts 10.0.0.2 and 10.0.0.3

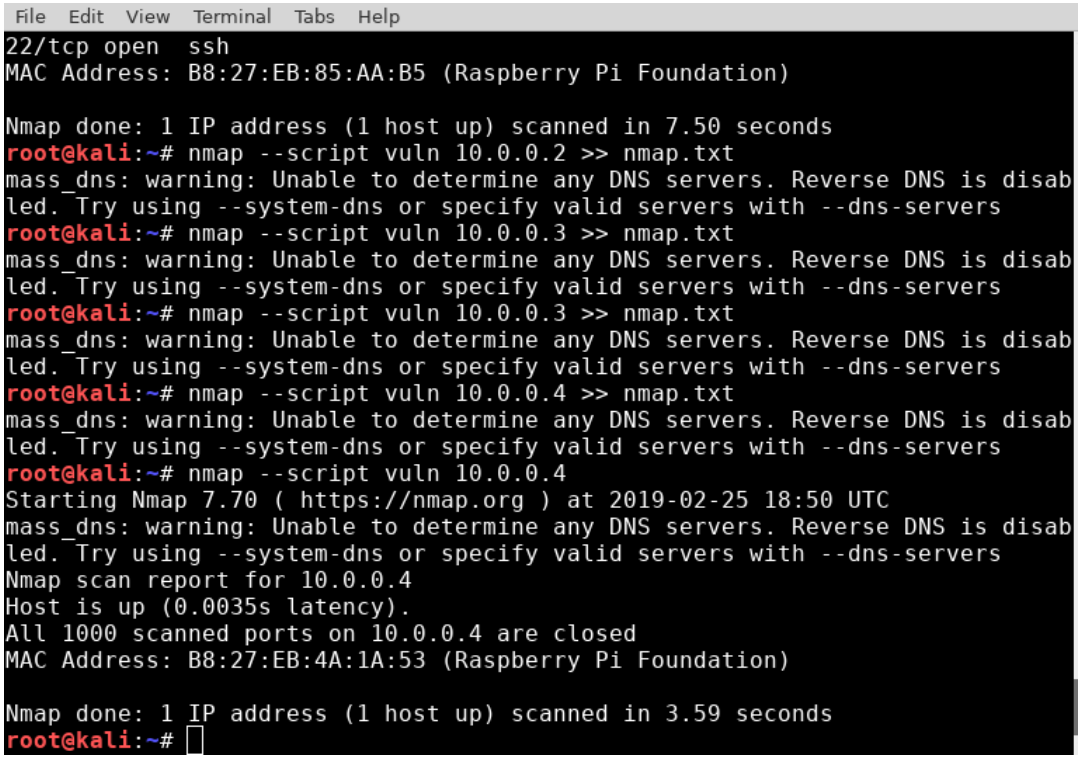

Figure 18: Nmap Vulnerability Script Run on 10.0.0.4

File Transfer Protocol Server Deployment

The Author began configuring the malicious FTP server by starting up an instance of the Metasploit Framework. To start the Metasploit Framework, the Author utilized the command: msfconsole. The Author can be seen executing this command in *Figure 19*. After starting an instance of the Metasploit Framework, the Author then proceeded to start up the FTP server capture auxiliary module. The Author first discover this module on Offensive Security's official website on a webpage entitled 'Server Capture Auxiliary Modules'. (Offensive Security, 2019) The Author navigated to the malicious FTP server within the Metasploit framework by implementing the command: use auxiliary/server/capture/ftp/. The Author can be seen implementing this command in *Figure 20.* Next, the Author started the server with the command: run. Now the Metasploit Framework auxiliary server capture module was configured and running on the Kali Linux Raspberry Pi.

Figure 19: The Author Starting the Metasploit Framework

User Credentials Capture

With the malicious FTP server fully configured and running on the IPv4 address 10.0.0.19, the Author proceeded to capture user credentials. In the context of this project, the Author would need to alert the unsuspecting users to the malicious FTP server. The Author suggests that a social engineering attack vector would be the most effective route of navigating unsuspecting users to provide the Metasploit FTP capture server with their credentials. Each Raspberry Pi on the LAN entered the command: ftp 10.0.0.19. After entering the command, the Author provided login formation for each Raspberry Pi. Each of the three Raspberry Pi's login information can be seen in *Figure 20*. The Author then copy and pasted the relevant information

into a file named credentials.txt. The credential file named credentials.txt can be seen in *Figure 21*.

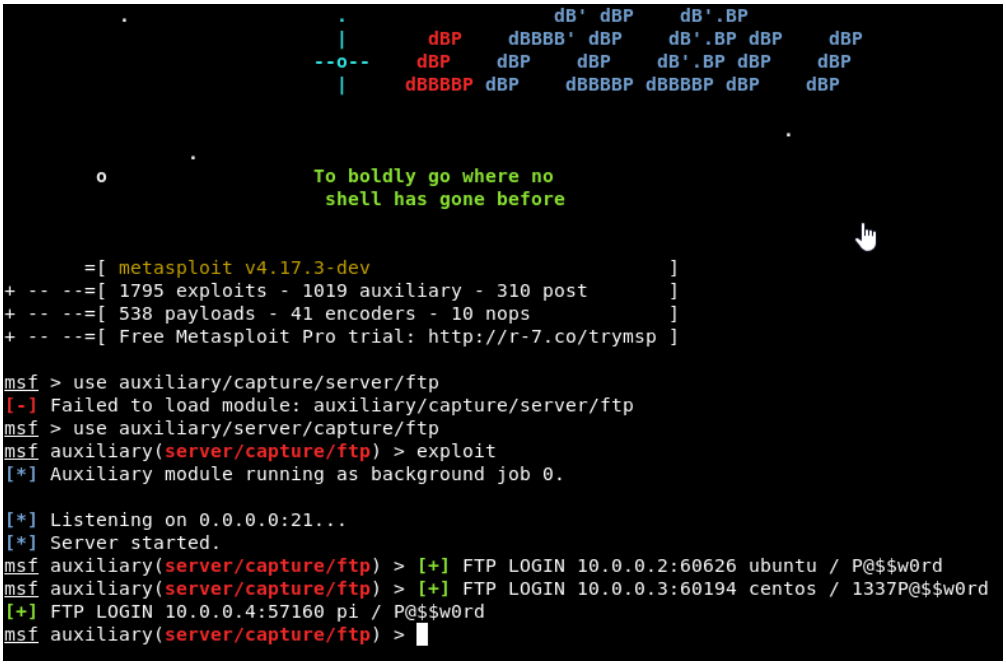

Figure 20: Metasploit FTP Server Auxiliary Module Deployment and Credential Capture

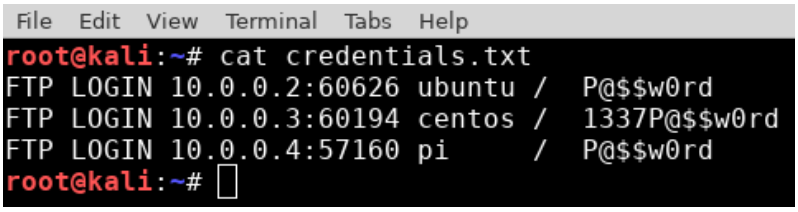

Figure 21: Stored Credential File Named credentials.txt

Forensic Investigation

The Forensic investigation of the malicious activity on the LAN of Raspberry Pi computers was comprised of two primary parts, a network forensics investigation, and a digital forensics investigation of the Kali Linux Raspberry Pi. The network forensics investigations will consist of analyzing the system logs. The digital analysis will include carving for files on the disk image file with scalpel and then locating those files within a hex editor.

Network Forensics Investigation

The Author began the investigation of the network forensic data by logging on to the Raspbian Raspberry Pi and navigating to /var/logs/ where the system logs files are contained. The Author manually examined the file kern.log.1 located within the /var/log/ directory. The Author was able to find evidence within kern.log.1 that an unauthorized IPv4 address attempted to connect to the Raspbian Raspberry Pi numerous times. The Author was able to determine through analysis of the log that the unauthorized system had an IPv4 address of 10.0.0.19, and a MAC address of: B8:27:EB:4A:1A:53:B8:27:EB:E5:42:3B:08:00:45:00:00:2C:18:12:00:00, which can be seen in *Figure 22*. To show how many events in the logs related to the IPv4 address of the malicious Kali Linux Raspberry Pi, the Author preformed a grep search. The syntax implemented by the Author for the grep search was: grep “10.0.0.19” -r /var/log/ -c. As shown in *Figure 23*, the Author located 14,872 instances of “10.0.0.19” within both sys.log.1 and kern.log.1.

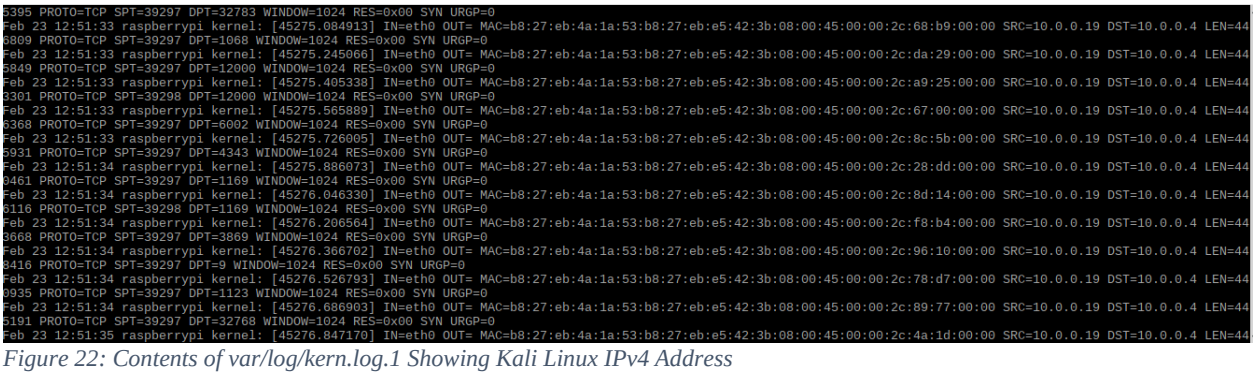

Figure 22: Contents of var/log/kern.log.1 Showing Kali Linux IPv4 Address

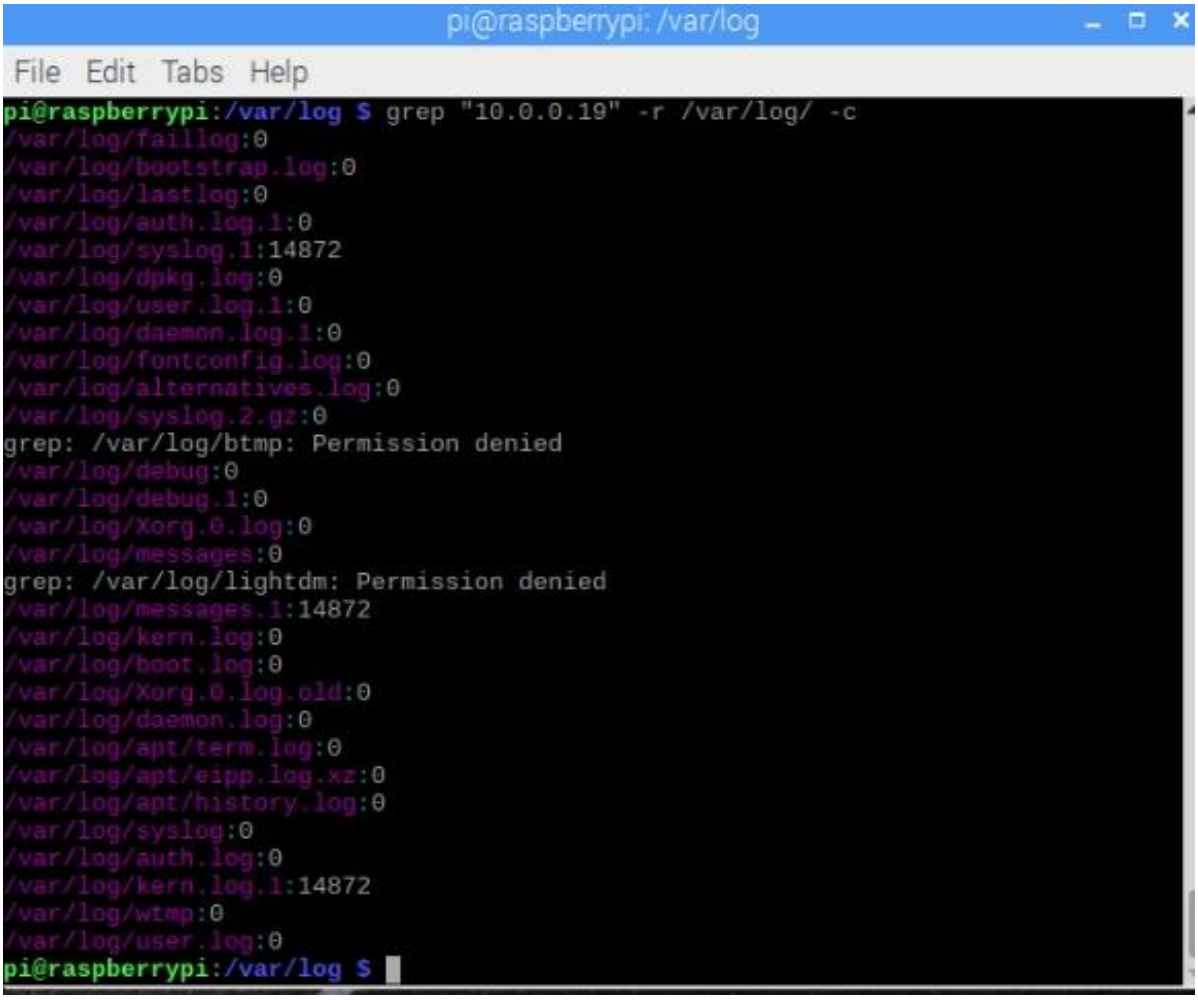

Figure 23: GREP Results for 10.0.0.19 in /var/log/

Digital Forensics Investigation

After locating the IPv4 and MAC address of the Kali Linux Raspberry Pi, the Author moved on to investigating the digital storage of the Kali Linux Raspberry Pi. The Author began this investigation by first powering-down the Kali Linux Raspberry Pi and removing the 16GB microSD card from the system. The Author then inserted the microSD card from the Kali Linux Raspberry Pi into a MSi GT60 forensic workstation running Ubuntu 18.04LTS. The Author then updated and upgraded the MSi GT60 workstation using the command: sudo apt-get update && sudo apt full-upgrade. The Author can be seen running this command in *Figure 24*.

After updating and upgrading the workstation, the Author proceeded to acquisition three images of the 16GB microSD card. The Author utilized the dd program to create kali.dd, WC_kali.dd, and PRES_kali.dd. The Author can be seen creating these images with dd in *Figure 25*. After the creation of each of the DD files containing the disk images, the Author then verified their integrity with md5sum. The Author verified the Integrity of the disk images by first retrieving the MD5 hash of each DD file, and then comparing them. The Author can be seen utilizing md5sum in *Figure 26.* Subsequently , each DD file produced an identical MD5 hash with md5sum which can be seen in *Figure 26*.

The Author then proceeded to open WC_kali.dd with the wxHexEditor hexadecimal editor. WC_kali.dd can be seen open with in wxHexEditor in *Figure 27*. Once WC_kali.dd was opened within the hexadecimal editor, the Author then Proceeded to Analyze the contents of the plaintext. The Author located some suspicious information which the Author suspected was a text file stored from hexadecimal offset 0x00AC06F7B-0x00AC06FFE. This segment of the DD file can be seen within *Figure 27*. The Author proceeded to dump that segment of .dd to a file named: credentials.txt. The Author can be seen printing the contents of credentials.txt with cat in *Figure 28.* At this point, the Author had discovered the extent of the network compromise by locating all user credentials on a file within the Kali Linux Raspberry Pi Hard Drive storage.

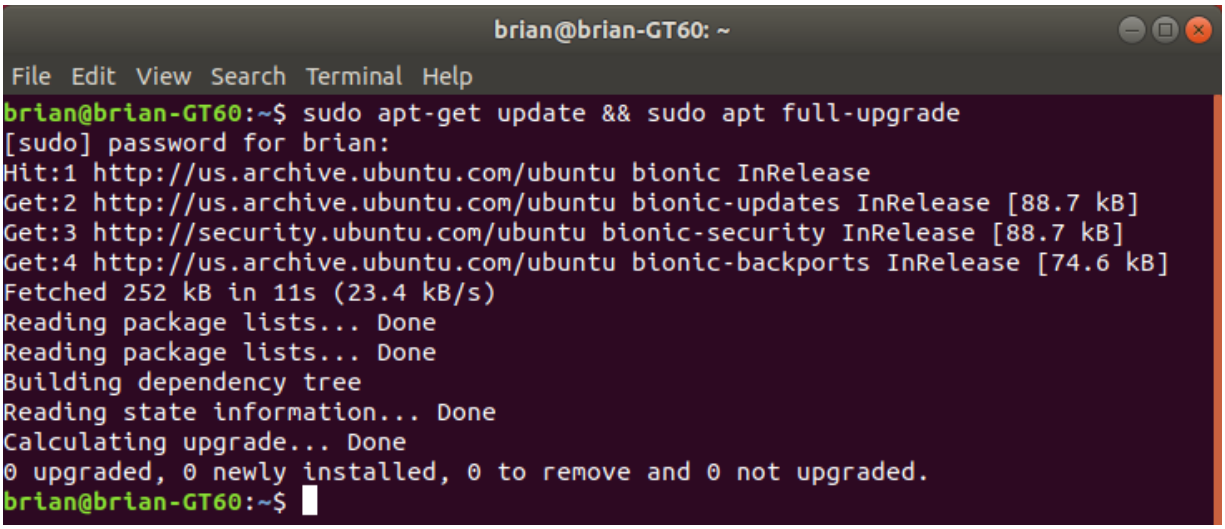

Figure 24: Update and Upgrade on Ubuntu 18.04LTS Forensic Workstation

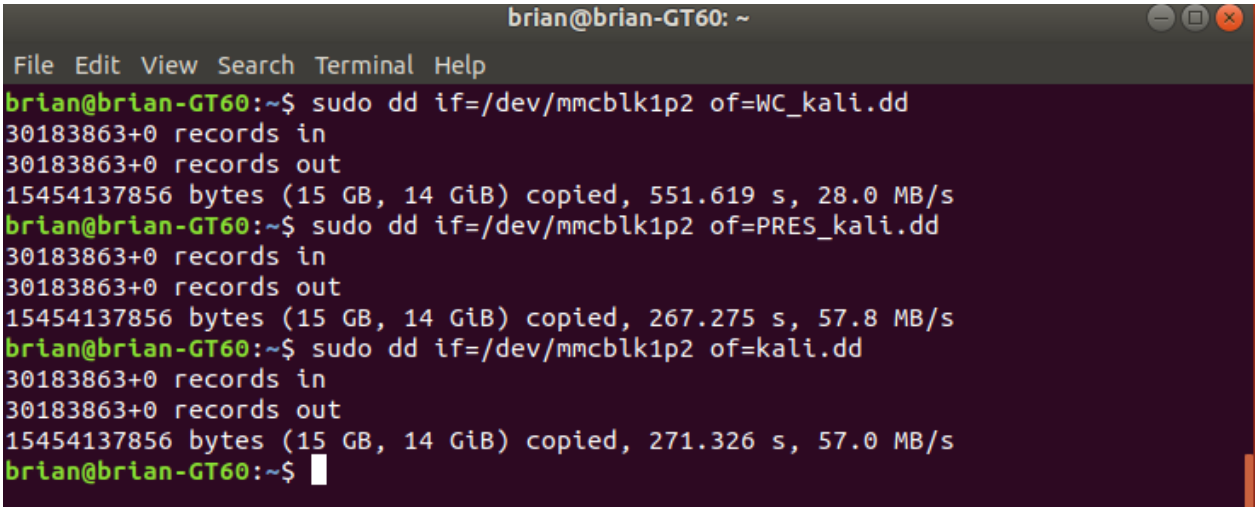

Figure 25: Creation of kali.dd, PRES_kali.dd, and WC_kali.dd

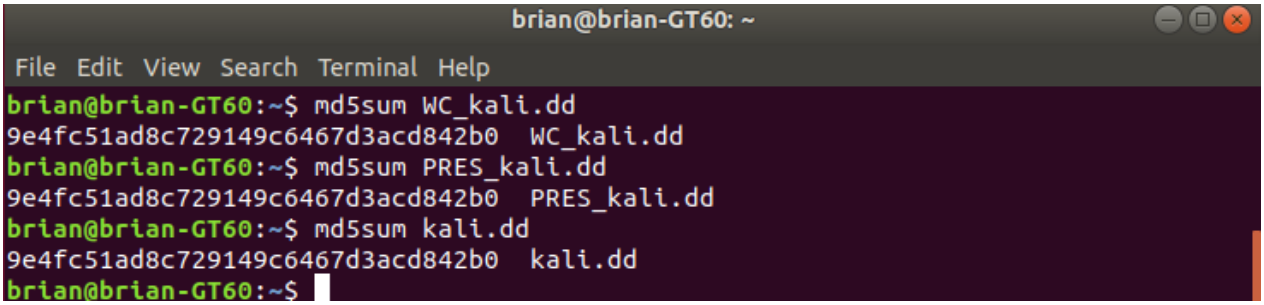

Figure 26: MD5 Hashes of Each DD File to Check File Integrity

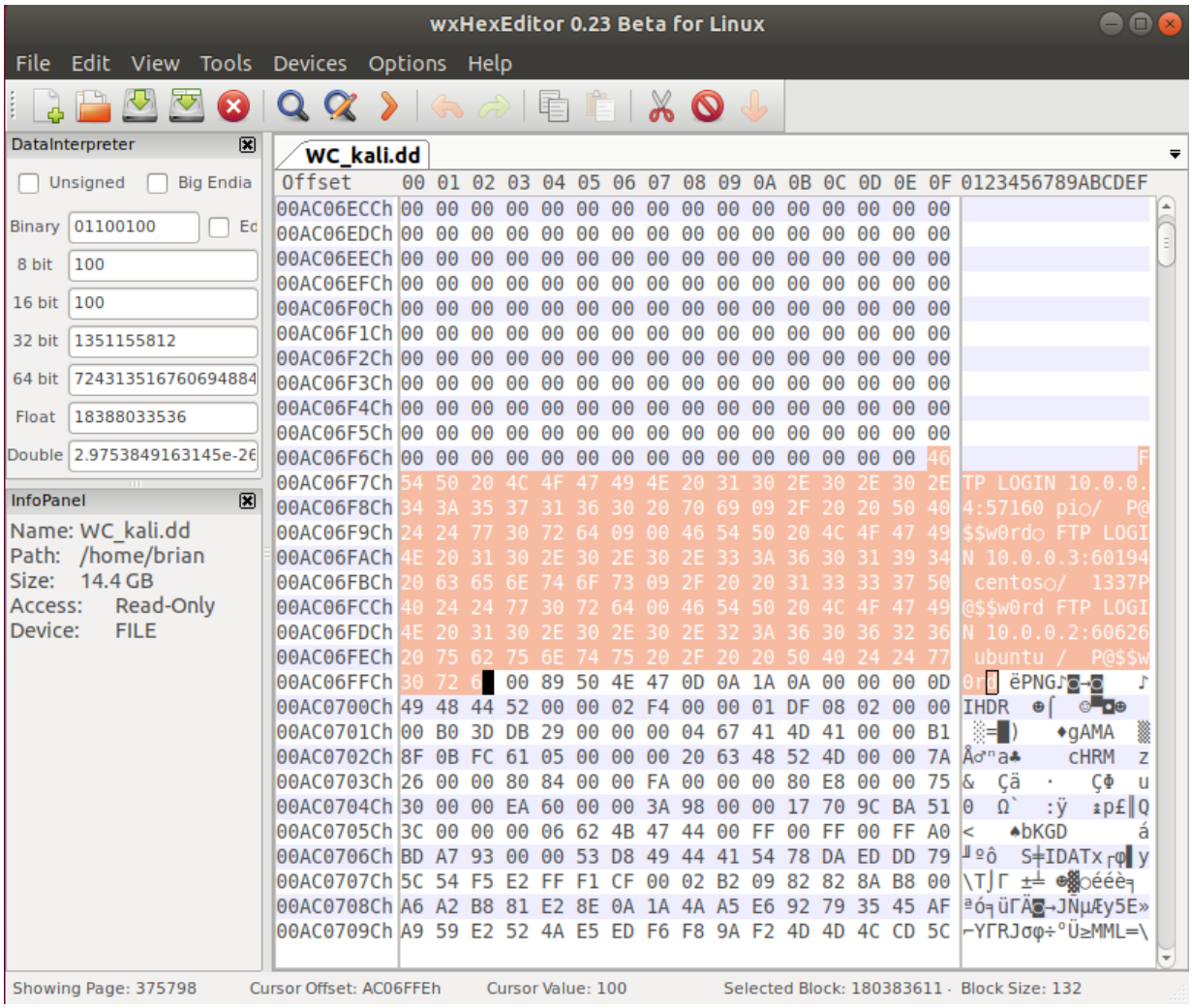

Figure 27: Location of credentials.txt Within the DD file Viewed With wxHexEditor

```
brian@brian-GT60:~$ cat credentials.txt
FTP LOGIN 10.0.0.4:57160 pi      /  P@$$w0rd      FTP LOGIN 10.0.0.3:60194 centos/  1337P@
$$w0rdFTP LOGIN 10.0.0.2:60626 ubuntu /  P@$$w0rdbrian@brian-GT60:~$ cat credentials.txt
```

Figure 28: credentials.txt Extracted from the DD File

Conclusion

Throughout the course of this article, the Author has taken an exploration into the configuration of a network of Raspberry Pis with the implementation of input and IPv4 logging, the configuration of a malicious FTP server on that network, and the forensic investigation of the incident. The Author explored numerous topics to combine various technical elements to produce a valuable educational reference, as well as a reproducible project. The Author went through the installation and configuration of four Linux operating systems on four different Raspberry Pi computers. The Author then preformed multiple network scans before configuring a malicious capture server with the Metasploit framework. After capturing user credentials and saving them to credentials.txt, the Author proceeded to forensically investigate the incident. Through investigation of the logs configured by the Author's utilization of Iptables, the Author was able to determine the IPv4 and MAC address of the unauthorized system on the LAN. The Author then forensically investigated the microSD card which the Raspberry Pi Utilized as a Hard Drive. Finally, during the investigation, the Author located all three user credentials in a text file on the Kali Linux Raspberry Pi hard drive.

The Author's goal was not only to preform the responsibilities of various cybersecurity professionals, but to display how their functions fit together in a broader view. The Author firmly believes that cybersecurity professionals need more educational and resource content. It is the Author's hope that others can utilize or recreate this Article in the future.

References

American Psychological Association. (2010). *Publication Manual* (Sixth ed.). Washington, D.C., United States of America: American Psychological Association . Retrieved 2019

Davidoff, S., & Ham, J. (2012). *Network Forensics* (1st ed.). Uttar Pradesh, India: Prentice Hall. Retrieved 2019

Engebretson, P. (2013). *The Basics of Haching and Penetration Testing.* (D. Kennedy, Ed.) Waltham, Massachusetts, United States of America: Syngress. Retrieved 2019

Godinez, L. (2016, October 27). *Raspberry Pi 3 - eth0 wrongfully named 'enx…'.* Retrieved 2019, from raspberrypi.stackexchange.com: https://raspberrypi.stackexchange.com/questions/43560/raspberry-pi-3-eth0-wrongfully-named-enx

Nelson, B., Phillips, A., & Steuart, C. (2016). *Guide to Computer Forensics and Investigations* (Fifth ed.). Boston, MA, USA: Cengage Learning. Retrieved 2019

Offensive Security. (2019). *Server Capture Auxiliary Modules*. Retrieved 2019, from offesnive-security.com: https://www.offensive-security.com/metasploit-unleashed/server-capture-auxiliary-modules/

Rash, M. (2007). *Linux Firewalls.* San Fransisco, California, United States of America: No Starch Press. Retrieved 2019

Appendix

Tools

- Win32 Disk Imager v1.0
- Iptables v1.6.0
- Metasploit Framework v4.17.3
- Nmap v7.70
- wxHexEditor v0.24 Beta
- MD5SUM v8.28
- dd v8.28
- Grep v3.1