

Wireless Security on Public Wi-Fi Networks

Alex Andrews, Keith Cornell, Zach Linderman, Adam Meyer,

Nathan Snyder, Thomas Wilson

CIT 46000 - Wireless Security, IUPUI

Indianapolis, Indiana, USA

Abstract -- The report focuses on designing and creating a Raspberry Pi v2 Model B - called a Rasp-Pi-Scanner - for researching and testing vulnerabilities and weaknesses on open Wi-Fi networks at publicly-accessible establishments. This includes installing the necessary tools on the Raspberry Pi, as well as developing an “automation of wireless information gathering” script. The purpose of the script is to collect data inconspicuously from establishments like Starbucks and Panera Bread. The goal of the project is to uncover any vulnerabilities and provide recommendations for improving them. Analysis of the data from our prototype Raspberry Pi shows unsatisfactory results. Our findings show that a better implementation is possible and could potentially provide rich data for improving open Wi-Fi at local businesses and organizations.

Keywords -- open Wi-Fi, wireless security, Raspberry Pi, Kali, data gathering

I. INTRODUCTION

Open Wi-Fi networks are relied upon by millions of people every day across the country in popular establishments like Starbucks and McDonalds. They are also an important part of many other governmental and religious organizations, like churches, schools, and libraries. Security and privacy are major concerns while connected on an open Wi-Fi network. According to a CNBC cybersecurity article on public WiFi networks, “more than 60 percent of consumers think their information is safe when using public internet [and]...only half of consumers think they are responsible for securing their information” [1]. Our group wanted to know if public WiFi can be made more secure and what people can do to better protect themselves.

Specifically, it is the policy of many business establishments to restrict access to the open

Wi-Fi to a limited amount of time during peak hours [2]. It was assumed that many of the users, once kicked off the public Wi-Fi service, will immediately attempt to re-establish a connection at some point during this time. A new handshake would be initiated, which presented an opportunity for an attacker to exploit. It was within this trajectory that the group tried to emulate.

Examining the traffic generated as users try reestablishing a connection provides invaluable research to developing a more secure open Wi-Fi network. And it is possible to help educate the public as well as teach good practices with public Wi-Fi.

II. BACKGROUND

The main inspiration for the Rasp-Pi-Scanner originated from a fascinating article by Scott Hogg called *Raspberry Pi as a Network Monitoring Node* [3]. The purpose of the node was for troubleshooting remote networks with the Raspberry Pi. The group took this idea and developed and molded it into a tool for performing data-gathering research and tests on open Wi-Fi networks. We coined this idea “war walking”.

The group’s CIT undergraduate education and lab work in wireless capture and security concepts and theories laid the foundation for this project. Specifically, the group’s work with wireless scanning tools (e.g., Wireshark, Kismet, Vistumbler) and the understanding of wireless architectures aided in the troubleshooting on this project.

Wireless security plays a significant role in the transmission process of data over a network medium. It can ensure the reliability, integrity, and confidentiality that the information being sent or received has been delivered to the anticipated device without alteration. Advancements in wireless security through the years has made a drastic change in the communication methodology over the internet. Each network can be configured to its own set of rules to allow or deny specific forms of traffic, which will

correspond with how well defended a network has been constructed. Best known practices are used by most organizations, but they aren't implemented in all cases.

For ease of access, there are businesses that still use unauthenticated open Wi-Fi networks that allow users to connect to an insecure access point. The idea behind this approach is to permit users access to free internet at their establishment. Unfortunately, this is done without really considering the potential threats that could affect the end-user. Data on an open network isn't sent in a secure format, thus putting users at risk to have their information monitored, dropped, or even altered by a malicious user. For the purpose of this project, the group wanted to determine what kind of traffic could be seen on an open network by utilizing tools that will capture data packets to all devices within relative proximity of the network adapter.

III. DATA COLLECTION PROCESS

The data collection process started with building the Raspberry Pi and creating an automated script to gather the wireless packets. Details about both the Pi setup and the script are explained in detail in the next two sections.

The Rasp-Pi-Scanner was taken to different locations by each group member. The startup of the device was simple and efficient. The scanner was booted up, and the script was easily edited with the new file name. After these quick steps, the device was placed out of sight. Our setup is small enough to fit into a backpack, so it allows for easy transport into each establishment. While the member sat at the establishment for a period of 30 - 60 minutes, the scanner was collecting data from the open Wi-Fi network.

Also, the scanner remained off the network. The monitor mode allowed the device to capture the radio transmissions of the network's access point.

IV. BUILDING THE PI

The original intent of the design was to build it with two important physical characteristics in mind: self-contained and extreme portability--something that could easily fit in a backpack or another innocuous container. This portability would allow any team member to keep it on their person without attracting unwanted attention.

The 7" touchscreen had already been purchased with the Raspberry Pi 2B by one member, and the Alfa antenna had previously been acquired for the Wireless Security class. Inspired by the Hogg article, the group created a shopping list of applications that the article had recommended. Over time and as the need arose, the group installed more applications like tshark and airodump-ng. Figure 1 shows the final results of the Rasp-Pi-Scanner.



Figure 1: Raspberry Pi's final build

Figure 2 shows the Alfa antenna that was used during the project. For convenience and ease, it was taped to the back of the Rasp-Pi-Scanner during the scans.



Figure 2: Alfa network adapter

Figure 3 shows the installation of the tools on the scanner, specifically aircrack-ng.

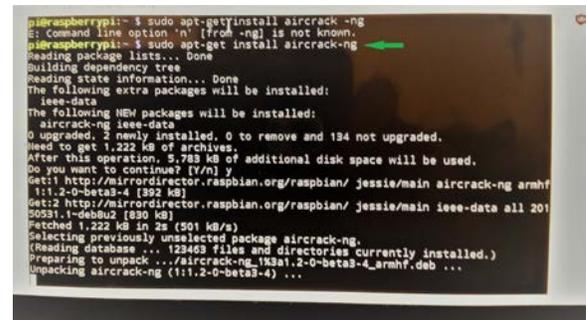


Figure 3: Installation of aircrack

Also, the keyboard, seen in Figure 1, is a Bluetooth-connected, compact touch-pad. The keyboard allowed any group member to easily navigate the Debian-based Raspbian OS.

As stated earlier, a variety of applications were installed. However, not all of these applications were utilized, even though they were part of the original shopping list. For instance, Nmap was not used in the execution of the final project, as shown in Figure 4.

Figure 4: Installation of Nmap on the scanner

V. THE SCRIPT

The purpose of the script was to automate the capturing of data on the open Wi-Fi network. The idea was to capture data while the Rasp-Pi-Scanner was hidden in a backpack or bag. The Rasp-Pi-Scanner can draw attention because of its unusual appearance. So, this script allows for the discrete capture of data. The script can be started and then put into a bag or other discrete location.

The script consists of a few main components and was built from scratch based on what we wanted to accomplish in this project. Figure 5 shows the original script prior to any alterations.

```
#!/bin/sh

#Bash Script for automation of wireless information gathering v1.3
#IUPUI CIT 460 Semester project
#Nathan Snyder, Alex Andrews
#Assumes you have aircrack-ng suite, macchanger, and wireshark installed

echo starting wireless packet capture...
sleep 3
#change mac address to hide real mac
echo changing mac address to hide real mac
sleep 3
macchanger --mac 00:11:22:33:44:55 wlan0
sleep 3
echo hid that mac!
sleep 5
#start airmon on default wlan0, edit if different interface
airmon-ng start wlan0
#log all traffic from nearby APs -I for monitor mode -k to start capture
immediately -w for outfile
#edit after the -w for your specific capture, your name, location, and
number if more than one
sudo touch cit460_name_location_num
sudo chmod o=rw cit460_name_location_num
sudo tshark -i wlan0 -w cit460_name_location_num
```

Figure 5: Original script

This version of the script consists of hiding the device’s MAC address, starting monitor mode on the main wlan interface, and saving the captured data to a file using tshark.

This script assumes that the following packages be installed prior to running it: macchanger, airmon-ng suite, and Wireshark with tshark. Sleep commands are placed between each command and comment so the user can see what is happening and all components of the script are working correctly. Each command logs its progress to the bash shell standard out.

VI. CHALLENGES

During this project, we experienced many challenges. These include the obvious technical challenges as well as some legal challenges. The scope of this project was limited due to the fact that we didn’t want to make this an active attack scenario. An active attack is both legally and ethically wrong. Rather, we decided that making this a passive scanning project would better serve our purpose.

Also, passive scans have limitations, since they are not transmitting packets. The Rasp-Pi-Scanner had to wait for beacons versus actively probing to locate access points. Another limitation is that if the scanner does not wait long enough on a channel, then it might miss an access point beacon [4]. Therefore, using this passive scanning method limits what we collected.

It also limited what could be done with any data collected. For instance, the group wanted to collect information about the activities of users on an open Wi-Fi network without having any identifiable attributes. Ethically, the plan was to strip away any identifiable, sensitive, and personal information, because this doesn’t pertain to the scope of the project nor is it ethical.

VII. ALTERATIONS TO SCRIPT

The specific results of the scans are discussed in detail in the following Results Section. However, the group didn’t get the results they expected. After collecting only beacon frames during the first round of scans, the group made changes to the original script. Specifically, the following changes were made:

1. Removing the “macchanger” line, since no packets were sent out
2. Removing the “sleep” line

- Using airodump-ng tool instead of tshark
- Ensuring the interface name doesn't change when wlan0 was set to monitor mode

Unfortunately, there were no significant changes in the results of the next round of scans following these alterations. Ultimately, the group tried performing a final round of scans using a laptop and without the Rasp-Pi-Scanner.

VIII. RESULTS

The first results the group collected were disappointing. All of the packets were broadcast packets, as shown in Figure 6.

No.	Time	Source	Destination	Protocol	Length	Info
2389	28.509187	ArubaNet_10157:12	Broadcast	802.11	181	Beacon frame, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
2390	28.511698	ArubaNet_10157:12	Broadcast	802.11	181	Beacon frame, SN=2985, FN=0, Flags=....., BI=100, SSID=OGMAGG
2391	28.564347	ArubaNet_10157:12	Broadcast	802.11	206	Beacon frame, SN=2921, FN=0, Flags=....., BI=100, SSID=OGMAGG
2392	28.565224	ArubaNet_10157:12	Broadcast	802.11	206	Beacon frame, SN=3074, FN=0, Flags=....., BI=100, SSID=OGMAGG
2393	28.565500	ArubaNet_10157:12	Broadcast	802.11	181	Beacon frame, SN=3074, FN=0, Flags=....., BI=100, SSID=OGMAGG
2394	28.579457	ArubaNet_10157:12	Broadcast	802.11	200	Beacon frame, SN=4039, FN=0, Flags=....., BI=100, SSID=OGMAGG
2395	28.579457	ArubaNet_10157:12	Broadcast	802.11	200	Beacon frame, SN=2232, FN=0, Flags=....., BI=100, SSID=OGMAGG
2396	28.622151	ArubaNet_10157:12	Broadcast	802.11	181	Beacon frame, SN=4039, FN=0, Flags=....., BI=100, SSID=OGMAGG
2397	28.655121	ArubaNet_10157:12	Broadcast	802.11	144	Probe Request, SN=1795, FN=0, Flags=....., BI=100, SSID=OGMAGG
2398	28.669731	ArubaNet_10157:12	Broadcast	802.11	200	Beacon frame, SN=2921, FN=0, Flags=....., BI=100, SSID=OGMAGG
2399	28.669732	ArubaNet_10157:12	Broadcast	802.11	181	Beacon frame, SN=2921, FN=0, Flags=....., BI=100, SSID=OGMAGG
2399	28.679331	ArubaNet_10157:12	Broadcast	802.11	200	Beacon frame, SN=3074, FN=0, Flags=....., BI=100, SSID=OGMAGG
2399	28.682932	ArubaNet_10157:12	Broadcast	802.11	181	Beacon frame, SN=3074, FN=0, Flags=....., BI=100, SSID=OGMAGG
2399	28.683232	ArubaNet_10157:12	Broadcast	802.11	200	Beacon frame, SN=4039, FN=0, Flags=....., BI=100, SSID=OGMAGG

Figure 6: All broadcast frames collected

Broadcast packets do contain some important basic information pertaining to the open network, such as the SSID, channel, and associated IP addresses. Unfortunately, the large number of broadcast packets collected was not the anticipated result due to the scope of the project.

There were also a number of ARP packets that had been captured during several scans on the Rasp-Pi-Scanner. These packets are useful since they help determine IP conflicts, link bounces, system reconnects, or updating the ARP table. However, the idea behind the data collection process was to gather other forms of data than just ARP and broadcast packets. Figure 7 shows the ARP packet captures.

No.	Time	Source	Destination	Protocol	Length	Info
296			Broadcast	802.11		Beacon frame, SN=1406, FN=0, Flags=....., BI=100, SSID=BHNTG1
304			Broadcast	802.11		Beacon frame, SN=344, FN=0, Flags=....., BI=100, SSID=OGMAGG
274			Broadcast	802.11		Beacon frame, SN=3065, FN=0, Flags=....., BI=100, SSID=BHNTG1
96			Broadcast	802.11		Gratuitous ARP for 192.168.128.1 (Request)
96			Broadcast	802.11		Gratuitous ARP for 192.168.128.1 (Request)
96			Broadcast	802.11		Gratuitous ARP for 192.168.128.1 (Request)
96			Broadcast	802.11		Gratuitous ARP for 192.168.128.1 (Request)
299			Broadcast	802.11		Beacon frame, SN=3909, FN=0, Flags=....., BI=100, SSID=BHNDG1
314			Broadcast	802.11		Beacon frame, SN=1362, FN=0, Flags=....., BI=100, SSID=NETGEA
302			Broadcast	802.11		Beacon frame, SN=3616, FN=0, Flags=....., BI=100, SSID=Broadc

Figure 7: ARP packets

After a couple more scans, the group started making alterations to the script, as mentioned in the previous section. One of the last ideas was running some scans with a laptop instead of the Raspberry Pi. The group had an improvement in the results. To do this, Kali Linux VM was run on a laptop with the Alfa network adapter in monitor mode. The tools

airodump-ng, tshark, and Wireshark were run at different times over a 60-minute period.

The group collected beacon frames, probe request frames, TCP packets, TLSv1.2 “Encrypted Handshake Message”, acknowledgement (ACK) frames, and data frames. Figure 8 is an example of some of the TLS and TCP protocols.

No.	Time	Source	Destination	Protocol	Length	Info
802	1078.509248	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	802 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
803	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	803 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
804	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	804 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
805	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	805 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
806	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	806 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
807	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	807 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
808	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	808 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
809	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	809 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
810	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	810 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
811	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	811 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
812	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	812 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
813	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	813 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
814	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	814 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
815	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	815 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
816	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	816 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
817	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	817 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
818	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	818 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
819	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	819 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
820	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	820 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
821	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	821 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
822	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	822 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
823	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	823 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
824	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	824 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
825	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	825 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
826	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	826 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
827	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	827 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
828	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	828 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
829	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	829 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
830	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	830 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
831	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	831 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
832	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	832 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
833	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	833 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
834	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	834 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
835	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	835 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
836	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	836 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
837	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	837 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
838	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	838 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
839	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	839 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
840	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	840 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
841	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	841 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
842	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	842 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
843	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	843 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
844	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	844 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
845	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	845 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
846	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	846 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
847	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	847 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
848	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	848 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
849	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	849 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
850	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	850 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
851	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	851 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
852	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	852 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
853	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	853 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
854	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	854 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
855	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	855 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
856	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	856 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
857	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	857 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
858	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	858 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
859	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	859 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
860	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	860 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
861	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	861 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
862	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	862 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
863	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	863 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
864	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	864 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
865	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	865 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
866	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	866 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
867	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	867 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
868	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	868 Probe Request, SN=3095, FN=0, Flags=....., BI=100, SSID=OGMAGG
869	1078.510150	Client: 10.10.10.10	Server: 10.10.10.10	TCP	60	

```

Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 58
Protocol: TCP (6)
Header checksum: 0x6055 [correct]
Source: 184.50.239.16 (184.50.239.16)
Destination: 10.243.16.243 (10.243.16.243)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Transmission Control Protocol, Src Port: http (80), Dst Port: 62571 (62571), Seq: 1, Len: 1448
Hypertext Transfer Protocol
0020 08 00 45 00 05 dc 17 9e 40 00 3a 06 60 55 31 25 ..E....8..1..L
0030 18 0a f3 10 f3 05 50 f4 0b 02 81 c6 75 7b a6 5.....P.k...vIF
0040 57 16 80 10 03 40 af c5 00 00 01 01 08 0a 41 7b .....:.....:
0050 00 52 50 b0 b0 26 9e 56 72 af 67 6e 54 37 7b 2b 45 .RP.5.vP dgrTPv+E
0060 57 36 32 6e 72 48 71 31 39 64 42 05 58 34 36 45 WcDhPq11 S8BhAGE
0070 4d 31 05 2f 54 50 64 32 57 03 6e 06 4c 53 7a 33 M1v/TP#2 wchL5C3
0080 42 4a 35 4b 74 73 5a 41 47 64 55 30 52 54 59 B3Kts2A G8J0ZTY
0090 69 67 48 6a 73 58 31 74 71 30 78 6c 41 4c 38 64 i gpxe11E dN1LuLm
00a0 2b 6a 62 73 55 6d 32 47 6f 72 43 4c 58 64 2f 75 +tBuLhG0 ofCLx3/u
00b0 44 66 6f 44 71 05 71 41 7a 68 3a 35 72 41 0f D1Dh+eg AchMzP#A
00c0 73 66 64 33 6f 4c 32 2f 78 31 4a 35 37 68 7a 63 vfd3uL2/ x1N07hcc
00d0 31 6a 54 42 49 61 62 42 51 30 74 6f 26 62 6d 59 INT8abb 00t+bv#
00e0 56 41 6a 66 2f 2b 57 77 33 67 48 31 39 61 65 56 vartfnew 3p1u#vav
00f0 75 71 6c 4e 62 68 74 56 05 48 69 57 48 77 4f 4e uq1N8hTv kH5hWdN
0100 05 42 6c 67 67 48 4f 63 78 30 4b 2b 05 6a 79 64 #llggp4Kc X0#v#y#
0110 01 69 30 76 63 51 37 31 70 42 2f 36 2f 63 59 64 77 10vC071p B6/cv#v
0120 54 52 45 57 41 66 43 64 6e 48 45 51 72 54 6a 72 TR8a#C# RH0Q7j#r
0130 42 67 62 63 64 54 4f 62 72 48 54 67 4c 55 77 Bpct#0# P#C7g#L#
0140 4b 4b 69 54 62 53 62 79 48 5a 2f 78 6a 61 61 67 Kk1T8By HZ/j#A#B
0150 44 69 49 30 64 73 6d 73 4f 6a 6a 74 52 55 53 D1Dh+eg 0j1TRUS
0160 2b 50 05 42 55 05 62 4a 62 68 62 44 32 50 53 31 +P#uL#J b#dCPS1
0170 67 2f 4e 78 73 59 4f 67 36 2b 35 4b 52 42 38 43 g#h#tDg 6#K#R#B#C
0180 41 77 45 41 61 63 4f 43 59 30 77 67 67 67 64 #d#A#A#C #Y#e#p#J
0190 4d 41 73 67 41 31 35 64 44 77 51 45 41 77 49 46 M#G#L#H D#G#E#I#F
01a0 6f 44 41 64 62 67 6a 66 48 53 55 45 46 6a 41 55 c#d#B#W# H#E#F#J#I
01b0 42 67 67 72 42 67 45 46 42 51 63 44 41 51 59 49 B#p#B#F B#C#D#Z#Y
01c0 4b 77 59 42 42 51 55 48 41 77 49 77 46 77 59 44 K#T#B#J#H #d#L#M#Y#D
01d0 56 52 30 66 42 43 77 46 4a 41 6f 43 61 67 67 64 V#T#D#w K#h#e#p#
01e0 4d 49 59 61 61 48 52 30 63 44 6f 76 4c 32 4e 79 J11#H#R# C#D#L#Z#Y#
01f0 62 43 35 62 6a 52 79 64 58 4a 30 4c 6d 35 6c l#C#L#W#Y #0#L#S#L#
0200 64 43 39 73 5a 58 5a 6c 62 44 46 72 4c 6d 4a 79 d#G#Z#Z#L B#F#L#M#Y
0210 62 44 42 4c 42 67 4a 56 48 53 41 45 52 44 42 43 B#L#B#W# H#A#F#R#C#
0220 4d 44 59 47 43 6d 43 67 33 41 47 47 2b 6d 77 4b M#V#C#C#C #d#G#e#k#
0230 41 51 55 77 4b 44 41 6d 42 67 67 72 42 67 45 46 A#L#K#D#M B#p#B#F#
0240 42 51 63 43 41 52 59 61 61 48 52 30 53 44 4f 76 S#C#A#F# #H#E#F#Y#
0250 4c 33 64 33 64 79 35 6c 62 6a 52 79 64 58 4a 30 L#J#J#Y#L b#y#d#N#D
0260 4c 6d 35 6c 64 43 39 79 63 47 45 77 43 41 59 47 L#M#L#C#Y# c#E#C#A#G#

```

Figure 10: Single HTTP packet captured

Figure 11 shows the information found regarding this packet.

General IP Information

IP: 184.50.239.16
 Decimal: 3090345744
 Hostname: a184-50-239-16.deploy.static.akamaitechnologies.com
 ASN: 20940
 ISP: Akamai Technologies
 Organization: Akamai Technologies
 Services: None detected
 Type: [Corporate](#)
 Assignment: [Static IP](#)
 Blacklist: [Blacklist Check](#)

Geolocation Information

Continent: North America
 Country: United States 🇺🇸
 State/Region: Massachusetts
 City: Cambridge
 Latitude: 42.3626 (42° 21' 45.36" N)
 Longitude: -71.0843 (71° 5' 3.48" W)
 Postal Code: 02142

Figure 11: Information discovered on the packet's source

IX. CONCLUSIONS

Overall, there is a lot of room for improvement in this project. The results were not the expected aim when the project was first proposed. The group had hoped to collect meaningful data that could be statistically analyzed and presented in context to offering recommendations for improving open Wi-Fi networks.

With the current design, setup, and implementation of the Raspberry Pi and the automated script, the group only collected beacon

frames. There was a significant difference when doing the same tasks on a laptop. This led the group to consider that there was something overlooked on the initial design and setup of the Raspberry Pi itself.

One idea that was never tested was running the laptop and the Rasp-Pi-Scanner at the same location at the same time to compare the results. At this time, it is inconclusive whether this was strictly a Raspberry Pi hardware/software issue or something related to the locations of the first round of scans with the Rasp-Pi-Scanner.

There is something behind the idea that there are no failures, only learning experiences. Though the group didn't get the expected results, the project was by no means a failure. A lot was learned from this project. A major component to IT-related work is troubleshooting. The group learned how to take an idea, develop it into a working test model, and refine it. Each new alteration and test brought the group closer to a working solution. Due to time constraints, however, this solution was not attained.

X. RECOMMENDATIONS

Though the group didn't get the expected results, it was still a learning experience. The group compiled a number of recommendations for users to remain better secured while surfing on open Wi-Fi networks.

1. Ensure the connection is encrypted -- always use HTTPS protocol when browsing
2. Enable "Secure Browsing" in the security settings
3. Avoid services that are not encrypted (e.g., FTP, HTTP)
4. Avoid submitting payment and other sensitive and confidential information
5. Use a VPN (e.g., Opera VPN service)

Similar recommendations, tips, and advice are echoed on numerous online resources related to best practices on open Wi-Fi. For instance, the Department of Homeland Security gives users a non-technical list of recommendations to protecting and securing information [6]. As mentioned in the Introduction Section, the level of ignorance among users is alarming -- only half of the users think they're responsible for their own security. One of the most important things to remember is that businesses in general aren't actively protecting users' activities

on open Wi-Fi networks. It is solely up to the user to implement safeguards to protect his or her self.

XI. REFERENCES

- [1] Schlesinger, J. (2016, Jun 28). *Most people unaware of the risks of using public Wi-Fi* [Online].
<http://www.cnn.com/2016/06/28/most-people-unaware-of-the-risks-of-using-public-wi-fi.html>
- [2] Starbucks (2017). *Starbucks Wi-Fi policy* [Online]. <https://www.starbucks.com/about-us/company-information/online-policies/sdn-terms-of-use>
- [3] S. Hogg. (2013, Oct 30). *Raspberry Pi as a Network Monitoring Node* [Online].
<http://www.networkworld.com/article/2225683/cisco-subnet/cisco-subnet-raspberry-pi-as-a-network-monitoring-node.html>
- [4] What are passive and active scanning? (2017). Wi-Fi Alliance [Online].
<https://www.wi-fi.org/knowledge-center/faq/what-are-passive-and-active-scanning>
- [5] Continuation or non-HTTP traffic. (2012, Jan 12). Ask Wireshark [Online].
<https://ask.wireshark.org/questions/8305/continuation-or-non-http-traffic>
- [6] Best Practices for Using Public Wi-Fi Tip Card [Online]. Department of Homeland Security.
<https://www.dhs.gov/sites/default/files/publications/Best%20Practices%20for%20Using%20Public%20Wi-Fi.pdf>