Summer Project Report: SAFE App Modifications

Abhro Bhuniya - 160050017

Rahul Chunduru - 160050072

July 24, 2018

Contents

1 Introduction **2**

2 Implementation **2**
2.1 Raspberry pi essentials 2
2.2 libpcap library 2
2.3 airmon software 2
2.4 Wireshark 3
2.5 iNotify 3
2.6 Main C program 3
2.7 Displaying user statistics 4

3 Feasibility and Bugs **4**

4 Testing **4**

5 References **5**

1 Introduction

This is the accumulation of all the research and packet analysis and capturing through c libraries done by us for the summer research internship under Prof. Bhaskaran Raman. We had an existing App, SAFE QUIZ, in function which was used for online quizzing. Our job was to run a C main program on A Raspberry Pi 3 which would capture packets from the existing network created by the SAFE server and make certain connection details available to the instructor.

2 Implementation

2.1 Raspberry pi essentials

- Watched video Tutorials on the Raspberry Pi 3 to understand its functionalities.
- Installed Raspbian OS onto the pi, to kickstart the pi which will host the main C program and capture the network packets.

2.2 libpcap library

- read up tutorials on the libpcap library which "provides implementation-independent access to the underlying packet capture facility provided by the operating system".
- Specifically focussed on the pcaploop function. When pcaploop() is called it will grab a specified number packets and pass them to the callback function.
- The callback function will process every packet captured and can be used for dissecting it into LLC packet, IP packet, TCP packet and so on.

2.3 airmon software

- The airmon software is used to capture Wireless packets from the network which is not possible by simple libpcap. It will configure a wireless interface to capture wireless packets.

2.4 Wireshark

- We included the netinet package which provides direct access to fields inside each of the headers inside the packet by holding the struct information of each of the packet headers.
- We used Wireshark for analysis and debugging in every packet capture analysis. Whenever we needed to know the field information location or hidden packet details we would use Wireshark to get the necessary details.

2.5 iNotify

- It is a UNIX API to notify the program whenever a file in specified directory is modified.
- This was used to check of the log file is getting updated by the server and will notify the program about the change. The change can be to show if the student device has downloaded the quiz, submitted the quiz, so on.

2.6 Main C program

- First section of code captured the packets. These packets are received in Byte format which have to be typecasted to access the required fields.
- We created the structure of a user device having fields like MAC address, IP address, State data, TCP bytes and other required fields. The We also have a Student struct holding the array of different devices associated with one roll number.
- The code constantly checks the values inside these struct objects against the Log files supplied by the server. Any change will be notified by iNotify.
- In the call back function, each packet captured is parsed into relevant values which were updated and stored for every device. First we got the details from Wifi header, then the LLC header, then the IP header and atlast the TCP header.
- Implemented a State Diagram which captured the notion of state of every device. Everytime a packet of higher state was received, the state of the corresponding device was updated. Everytime an Association Packet was received it is assumed that the device got disconnected and reconnected and will start from the minimum state again.

- Finally lpthread library was used to make the program multi-threaded so that the checking of log files and packet parsing can take place parallelly.
- The program finally prints the user device data statistics into the JSON file.

2.7 Displaying user statistics

- Displayed statistics about user devices connected into the network to the instructor through JavaScript.
- Statistics like Device State, TCP Bytes Received and IP address were displayed in a dynamic sortable table. Also displayed the number of users in different states inside a Pi Chart.
- Used Canvas element in JavaScript and stored the values to be displayed in appropriate JSON format.

3 Feasibility and Bugs

All these separate modifications put together in association with the SAFE app can hugely aid the instructor to judge the attendance of students and will aid in weeding out any foul play. The main portions of the project are implemented separately and are needed to be properly combined together. The DNS name of the server side needs to be edited into the main program.

4 Testing

- Testing was done by running the code on Raspberry Pi 3 and configuring the wireless interface wlan1 with airmon to intercept the wireless packets.
- The pi was switched into VNC server mode and accessed by the laptop through SSH. A virtual desktop of the pi was created through VNC viewer to check for the debugging of the code.
- Details of the packets captured were computed and used in C code by using Wireshark to gather all the information needed.

5 References

- $http://yuba.stanford.edu/\ casado/pcap/section1.html$
- $https://askubuntu.com/questions/512926/how-to-configure-wifi-adaptor-to-monitor-mode$
- $https://askubuntu.com/questions/543960/how-do-i-monitor-which-file-changes-during-changing-settings$
- $https://code.tutsplus.com/tutorials/how-to-draw-a-pie-chart-and-doughnut-chart-using-javascript-and-html5-canvas--cms-27197$
- $https://www.youtube.com/watch?v=IDqQIDL3LKgt=202s$
  Raspberry pi tutorials by sentdex