**PACKET SNIFFER PROJECT**

**Project Description Document**

1. **Project Statement**

The purpose of our project is to identify a user’s network activities such as Video applications, e-commerce applications and Browsing html pages, by sniffing

packets on port 80 and port 443.

We used Ubuntu, eclipse IDE, and java programming to code our project.

We used jnetpcap java wrapper around libpcap library, which is an application programming interface for packet capturing. So our project is related to network protocol stack, as we are capturing packets from application interface.

1. **Project Design**

Start

No

Error in getting list of Devices

Find all Devices on  
System

User Input to Listen on which Device

Open Live Connection

No

Error while opening  
Device

Yes

Capture Packets for  
30 Seconds

Build Modal using Training  
Data

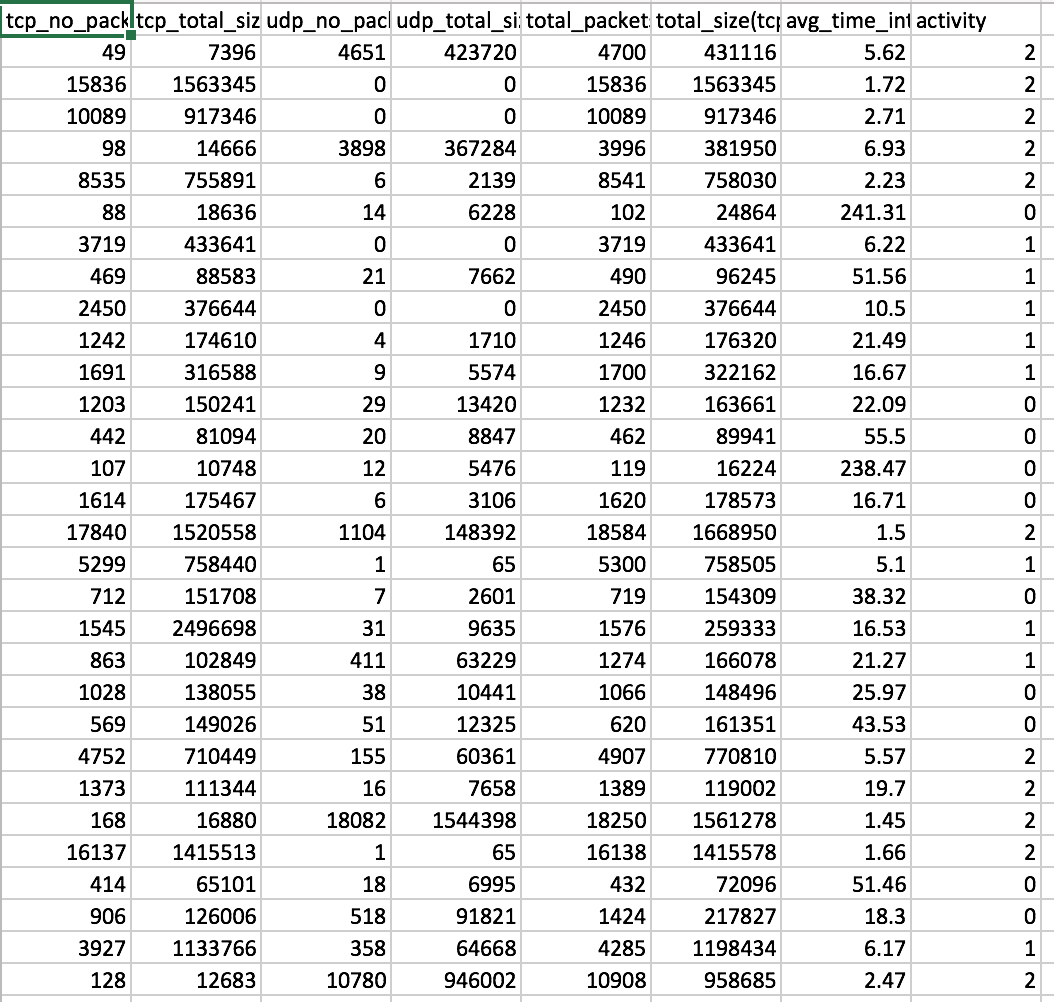
Training Set for Modal

Compare Data with Testbed and Analyze

Display the Activity

1. **Project Implementation and Evaluation**

Project is implemented in java using jnetpcap library to open a connection on one of the interface and listen to the live packets.

The testbed contains data related to number of **TCP** packets and packets size, number of **UDP** packets and packets size, **Total** number of packets and packets size and **Time** **interval** between the consecutive packets and the User activity the captured data is related to. 

The above mentioned testbed forms the ground truth for our project and we evaluate the live packets data with this ground truth using **Knn** classifier provided by **Weka API** and displays the application user is trying to access. Where **0, 1, 2 are referred to Browsing Html Pages, e-commerce and Video applications** respectively.

The performance metrics are, out of 10 we are getting around 8 to 9 correct predictions.

1. **Experiences and Thoughts**

Analyzing the user’s activity and predicting what activity the user is performing is a challenging task. Different systems have different noises like messenger will be active and will be browsing which makes the prediction tough. Other issues we faced are during videos, huge amount of packets are arriving at the beginning of the video, whereas after some time, when we start analyzing the user activity once the video is already playing, it is very similar to heavy html content pages (like e-commerce). If we can decrypt the header of the HTTPS we can use more information to train the modal and add more data in the training set, we can make better predictions.

1. **Instruction / README**

**OS Requirements:**

Used Ubuntu provided to us for Homework-1. Following steps need to be performed on the Ubuntu machine:

1.) Create a root user if it doesn’t exist.

2.) Give eclipse a root access because it needs to monitor the network activity which requires root access.

3.) I open my eclipse using command- gksu eclipse.

4.) Once this is done open eclipse you can run code in the below mentioned way.

**Method to create test environment:**

1.) Create new java project in eclipse.

2.) Import the provided jar files (in library folder) and Add jnetpcap's jar file and native library directory path to project's build path.

3.) Import PacketSniffer.java in the project.

4.) Import the provided trained data **cn\_new.csv.**

5.) Run the program PacketSniffer.java and follow the steps for testing it.

**Testing the code:**

1. Please have a look at the provided demo testing video.
2. For every activity, perform following steps :

**Activity Video :**

a.) Run the code and give input to listen on to the interface which is connected to network on your system.

b.) Open the Browser and paste the following URL :

<https://www.youtube.com/watch?v=GuRoxqkgEIU>

c.) Close the Browser.

**Activity e-commerce :**

a.) Run the code and give input to listen on to the interface which is connected to network on your system.

b.) Open the Browser and paste the following URL :

<https://www.amazon.com/> and go to category and click on particular product and browse through multiple products.

c.) Close the Browser.

**Activity Html pages :**

a.) Run the code and give input to listen on to the interface which is connected to network on your system.

b.) Open the Browser and paste the following URL :

<http://www.cs.umb.edu/~shengbo/teaching/cs446.html>

c.) Close the Browser.

**Activity Nothing :**

1. Run the code and give input to listen on to the interface which is connected to network on your system.
2. Do nothing and stay idle, it will display No Activity is being performed.
3. **References**

We have referenced jnetpcap wrapper (libpcap for linux) to open live connections on an interface. <http://jnetpcap.com/docs/javadocs/jnetpcap-1.3/index.html>

We have referenced WEKA API to compare the captured packets data against test bed and analyze. <http://weka.sourceforge.net/doc.stable/>

**TEAM MEMBERS – Adit Shah, Bhavana Duddugunta**