Internet Traffic Classification

Network development project

Haider Ali
14L-4163
Bachelor of Computer Science
Lahore, Pakistan
Haiderali9624@gmail.com

Haseeb Ahmed
14L-4315
Bachelor of Computer Science
Lahore, Pakistan
Haseeb.ahmed21@gmail.com

I. ABSTRACT

The internet is continuously evolving these days in terms of complexity and scope. This process is much faster than our ability to understand, control or predict it. This is due to the fact that the number of users and organizations using the internet are increasing day by day. The exponential increase in number of users has increased the demand of internet traffic classification tools. These tools aid the network administrator in network management which includes identifying anomalies in traffic such as spikes in usage and bandwidth monitoring etc. This project is about an internet traffic classification tool which classifies the internet traffic belonging to different application layer protocols. The internet traffic is collected through some sort of packet analyzer and capturing tool. This tool obtains packets from live packet captures obtained through tcpdump and identifies as well as quantizes different types of traffic such as Domain Name Server (DNS), Hyper Text Transfer Protocol (HTTP), Real Time Transfer Protocol (RTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) etc.

II. INRODUCTION

Millions of users are using the internet across the globe. This has elevated the importance of an efficient traffic classification tool as internet service providers or network administrators use these tool to classify packets which are then processed differently by a network scheduler. So, a good network scheduler is dependent on an efficient traffic classifier. Internet Service Providers use such tools to monitor the performance and working of a network and to treat each resulting traffic class differently in order to differentiate the service implied for the data generator or consumer. A predetermined policy can be applied to each class in order to guarantee a certain quality or to provide best-effort delivery service etc. These tools are aiding the network administrators and internet service providers to maintain network security. The overall management of these huge networks needs different methods to monitor their working and performances. Therefore, different traffic classification techniques are used for internet traffic classification belonging to different web applications which have their own benefits and drawbacks. One of these techniques includes classification via port numbers. This mechanism is fast and has low resource consumption but it is only used for application and services which use fixed port numbers. Another technique is deep packet inspection which inspects the actual payload of the packet and detects service regardless of the port number. However, it is slow and requires a lot of processing power. Other mechanisms include various machine learning algorithms such as K-Means or Random Forest which comes in the domain of “Statistical Classification.” So with the help of these techniques we can monitor which packets are requested frequently and which type of traffic is present during a specific time period. The core purpose of all these techniques was to help classifying the traffic on internet so that the network can be made more stable and efficient. Our tool is based on simple traffic classification which captures packets during a given time frame and identifies it on the basis of application layer protocols i.e. port based identification. The output is a pictorial representation in terms of application layer protocol type and quantity of packets belonging to that particular layer.

III. BACKGROUND

For management and monitoring of complex and huge networks, different traffic classification algorithms are employed in the past. The techniques which are so far being employed include port based identification, deep packet inspection (DPI) and various supervised and unsupervised machine learning algorithms are used. Port based identification is a bit vulnerable because it’s easy to change the port number in the system and DPI lacks support for many applications such as Skype. These techniques help to distinguish between three broad types of network traffic: Sensitive, Best-Effort and Undesired. Our project is based on traffic classification on the basis of application layer protocols. The packet analyzer and traffic capturing tool used for the tool is tcpdump.

The 'tcpdump' is a Linux Command based packet analyzer used for monitoring traffic on the network. It allows the user to display TCP/IP and other packets being transmitted and received over a network. TCPdump prints the contents of network packets by reading packets from a network interface

card or a previously created packet file. It is a low level packet sniffer and requires special privileges to sniff packets from a network interface card. It is an open source network utility that is freely available under the BSD license. It provides description of packet contents in several formats; depending upon the command used.

The main objective of this utility is to capture and record packets on run time. It is designed to provide statistics about packets received and captured at operating node for network performance analysis, debugging and to diagnose network bottlenecks. The packets captured by tcpdump have bi-directional flow. It also tells us the total number of packets captured. We can also apply sort of filter expression to get the total number of packets matching that filter expression. It also tells us which packets have been dropped by kernel due to lack of buffer space or the packet capture mechanism in the OS on which tcpdump is running. The basic features which tcpdump provides us is the timestamp telling us when the packet was transmitted/received by the system, IP address and port number of source and destination and length of the captured packet as well.

IV. METHODOLOGY

Our tool uses port numbers of the packets to classify them on the basis of their application layer protocol. Famous application layer protocols uses fixed port numbers to exchange data, so the port number is the perfect metric to classify packets on the basis of their application layer protocol.

Our tool provides a user friendly Graphical User Interface to the user. User can capture incoming and outgoing packets over the network within a time frame. The packets details are shown to the user in the form of a table. The table contains 5 columns, source IP address of packet, destination IP address of packet, source port number of the packet, destination port number of the packet and the last column contains the name of the application layer protocol of the packet.

User can also view the stats of the packet capture in the form of a pie chart or a bar chart.

A. *Tool for capturing traffic over the network:*

Our tool uses the linux command of tcpdump to capture live incoming and outgoing packets over the network. The tcpdump command captures packets which can later be used to analyze the contents of the packets

The tcpdump command which our tool uses is stated below:

tcpdump –i any –w test.pcap

The –i option defines the interface over which we want to capture the packets; any parameter indicates that we want to capture packets over all the interfaces.

The ‘-w’ option indicates that we want to write the details of the capture in some file. tcpdump generates a pcap file which is a packet capture file, it contains the details of all the packets captured by tcpdump.

B. *Language used for development:*

Our tool is developed in python 3.6. Python has a library named PyShark which can extract packet contents from the pcap file. The Integrated Development Environment (IDE) used for development is PyCharm.

C. *Library for analyzing Packets:*

Our tool uses PyShark library to analyze the contents of the packets from the pcap file henerated by tcpdump. Pyshark is a Python wrapper for tshark, allowing python packet parsing using wireshark dissectors.

TShark is a network protocol analyzer. It allows capturing packet data from a live network, or reading packets from a previously saved file. TShark's native capture file format is pcap format, which is also the format used by tcpdump.

D. *Packet Classification:*

Our tool uses uses port numbers fields of the packets to classify packets on the basis of their application layer protocols. When the packet contents are read from the pcap file, the contents contain many fields; the fields which are of our interest are the source port number and destination port number of the packet.

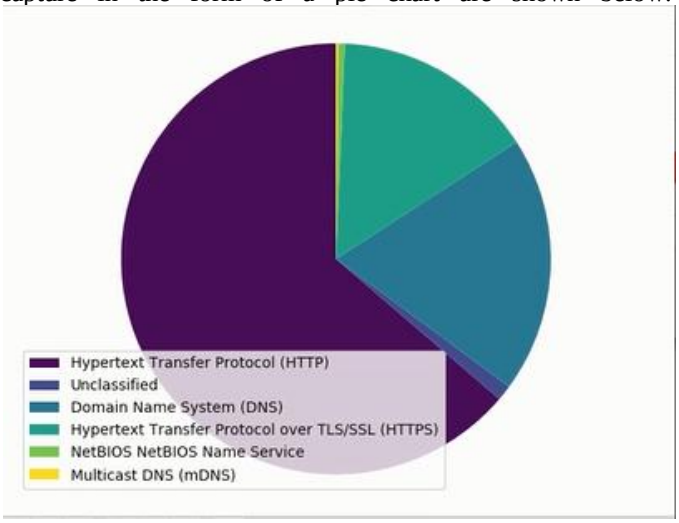
The port number is the parameter over which the application layer protocol of the packets is identified. Famous application layer protocol uses fixed port numbers to exchange data for example Http uses port 80, DNS uses port 53 and FTP uses port number 20 and 21. Therefore port number is the most suitable metric for identifying the application layer protocol of the packet.

The port number used by different application layer protocols and name of the protocols is stored in a json file. The data from this file is read and port numbers of the packet is compared with the data from the json file to identify the application layer protocol of the packet. Both the source port number and destination port number of the packet is compared with the json file data and the application layer protocol is identified by checking if either of the source port number and destination port number of the packet matches the port number used by a specific application layer protocol.

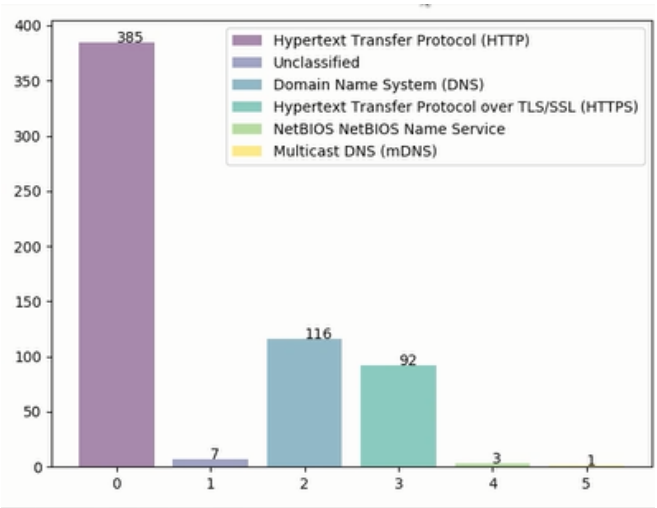
V. TOPOLOGY/TESTING/EVALUATION

For testing the tool, a packet capture was held for approximately 20 seconds. The detailed stats of the packet

capture in the form of a pie chart are shown below:



The detailed stats of the packet capture in the form of a pie chart are shown below:



As the results depict, majority of the packets i.e. 385 packets from the packet capture used the Hypertext Transfer Protocol (HTTP), 116 packets from the packet capture used the Domain Naming System (DNS), 92 packets used Hypertext Transfer Protocol over TLS/SSL (HTTPS) and 7 of the captured packets were unclassified i.e. they did not use any port number used by famous application layer protocols.

VI. CONCLUSION

To conclude, the world of internet is growing day by day and in order to understand the complexity and structure of the network, internet traffic classification is a necessity. An efficient classifier helps us to identify and quantize problems in order to manage and diagnose network related hazards. For this purpose, our internet traffic classification tool comes into play which can help network administrator and internet service provider to classify packets in order to apply some sort of predetermined policy for efficient network scheduling.

REFERENCES

[1] https://pypi.python.org/pypi/pyshark (htt1)J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] https://www.wireshark.org/docs/man-pages/tshark.html

[3] http://www.caida.org/research/traffic-analysis/classification-overview/

[4] https://en.wikipedia.org/wiki/Traffic_classification

[5] https://www.techopedia.com/definition/16162/tcpdump

[6] https://linux.die.net/man/8/tcpdump

[7] Video Demonstraion of Tool: https://www.youtube.com/watch?v=1MGsNZl3A4I&feature=youtu.be