CS371 Final Project Report

**Abstract:**

The goal of this project is to successfully detect bidirectional flow in an internet browsing session, and to then accurately detect what kind of internet browsing is being performed by using three machine learning algorithms. This project requires only the correct detection of four major kinds of web browsing:

1. Web browsing
2. Video Streaming
3. Video Conferencing
4. File downloading

These main types of browsing are very generic but encapsulates all major types of traffic that may be going on a network at any given moment. To do this, a library called scapy will be used to monitor web traffic and extract features that will then be passed through three machine learning algorithms to output what kind of web traffic is most likely being performed. From there, a report is generated on the accuracy of these predictions.

**Introduction and motivation:**

Packet sniffers are a category of software that sits between two hosts on a network and intercepts packets transferred between them. Packet sniffers may be used for malicious intent or not. However, the basis of our project is to assume we are extracting this information for malicious purposes. With a packet sniffer, one can analyze the packets collected and find information passed over the internet such as cookies, passwords, and other sensitive information. Our project will be a little less malicious than that, and instead only collect more general information held within these packets such as destination and source IP, number of packets passed between them, and the time it takes a flow to complete. This will allow us to then pass this information into our machine learning algorithms to detect what a user is doing on the internet.

Machine learning is the process of feeding an algorithm a large, known, set of data that it can learn from and manipulate to find key values in the data set that produce the known result. Then that algorithm is used to process through other sets of data provided to it and compare it to results of training data to predictively interpret the new set of data provided. This means that, given enough relevant, and sometimes even irrelevant, data a machine learning algorithm can, with some confidence, interpret a larger set of data more quickly and accurately than any human possibly could. It is that characteristic that motivates the use of machine learning in this project. Web browsing generates an absurd amount of data, and by training our algorithms to interpret this data, we can have it predict the activity that is being performed by a host automatically. The specific machine learning algorithms used in this project are Decision trees, Neural networks, and support vector machines. Decision trees function by breaking data down into specific identifiers that branch to other identifying junctions, down until the algorithm is able to, with some accuracy determine a likely result based on the information presented to It. Neural networks function on a similar principal, instead of flowing based on junctions and decisions however, each junction assigns a weight to the value passed to it, and from there flows along to the next level of junctions to a junction that corresponds to the resulting value of the previous junction. This newly found value will eventually be spit out by the algorithm as, again, the most likely result based on the information used to train the network. Support vector machines are able to assign an output to a combination of input data by attempting to form groups with as wide margins as possible based on data used to train it. Meaning that given a set of data, the machine will attempt to take all of the data points presented to it and produce bins of results that are based on as wide of variance in the data points given for each result. From there, the SVM can assign data to one of the assigned bins that corresponds to the most likely outcome of that data set.

**Proposed Features:**

The features of the packets that were extracted in this project were the total amount of packets, total amount of source packets, total amount of destination packets, total amount of bytes, total amount of source bytes, total amount of destination bytes, and total time of the flow of information. These features should be unique for each type of web browsing being performed in testing sessions. With this information, the machine learning algorithm is able to learn how each of these features’ changes depending on the kind of web browsing being performed and thus accurately predict said web browsing.

**Implementation:**

This project has been implemented in two separate python scripts. These scripts are scapy-skeleton.py and ML-skeleton.py.   
 Scapy-skeleton.py outputs two CSVs of the data collected utilizing scapy, and the average value of each feature collected for each browsing type. This CSV has 14 columns that contain information in the following order: source IP, destination IP, source port, destination port, protocol, total packets, source packets, destination packets, total bytes, source bytes, destination bytes, current time, duration time, and label. This csv is constructed of flows that have been detected as bidirectional, meaning that information has been passed from source to destination and from destination back to source. This is detected by comparing each sniffed packet to the list of flows that has been detected already. Specifically, scapy-skeleton checks if the new packet source IP or Destination IP is the same as either the source IP or the destination IP in a previous flow. As well, it compares the source and destination ports against the same previous flow as mentioned before, then compares the protocol as another layer of verification for bidirectional flow. If the packet is detected as belonging to an existing flow, the feature values of total packets, destination packets, total bytes, destination bytes, source packets, and destination packets are incremented and updated. Else, the newly found flow is appended to the existing list of flows, which will then be compared against future packets. For training purposes, each browsing session is delimited by prompting the user to input the type of browsing they are about to perform and appends the browsing session ID to each line of the CSV. Also calculated within each delimited session is the average of values of features collected in the browsing session. This is done at the end of each session by iterating through the flows collected and averaging each feature’s value, then appending that to another CSV file to also be imported into the machine learning script, this is not for training purposes, but instead for analysis of the efficacy of each selected feature chosen.

ML-skeleton.py takes input from the two CSV files outputted by scapy-skeleton. ML-skeleton then takes the following selected columns from the data exported and applies them to the machine learning algorithm: protocol, total packets, source packets, destination packets, total bytes, source bytes, destination bytes, current time, and duration time. The specific machine learning algorithms used are Decision trees, Neural networks, and support vector machines. Each of these algorithms is fed two training sets of data 10 times. Through each iteration, a prediction is made, and the test set is compared for accuracy, precision, recall, and F1 score against the prediction test. These values are tracked within a list to later be graphed, which is accomplished by passing the collected data from the machine learning modules into stacked bar graphs pulled from the matplotlib python library. As well, the aforementioned graph of the average of each feature’s values are shown.

**Experiments:**

The experiments performed on our data falls into three major groups, the first experiments done were to verify the efficacy of the features we selected to model. The second experiments were performed to ensure the machine learning algorithms were functioning properly, by feeding them handmade data that should produce wildly different results. The third, and last group of experiments performed is the combination of both scripts to ensure the machine learning algorithms were capable of computing the correct identifier for the web browsing session.

The first round of experiments was not done using the software created, or by using scapy, but instead was done by analyzing traffic flows from splunk, a software a team member is able to utilize. By analyzing traffic flows from splunk, we were able to find that the most difference between each traffic type could be seen by using the features we selected.

The second round of experiments were performed by feeding our machine learning algorithms made up data to ensure that a notable difference in values would correctly lead to the expected outcome. Our experiments successfully returned a 100% prediction rate when inputting the made-up information. This experiment also helped to verify the process of pulling in a CSV file to the machine learning script.

The third round of experiments was performed utilizing the nearly final versions of each of the scripts, scapy-skeleton.py and ML-skeleton.py. These experiments were used to both debug and optimize our scripts. In testing, we found various issues with the calculations of some information. With these issues resolved, it was then seen clearly that the machine learning model that most accurately predicted the correct browsing session was <insert MLA here>.

**Conclusions:**

This project was some complete bullshit. Machine learning is dumb.

TODO

Clarify some of the above information and proofread.