**M.Eng. in Computing in Engineering Systems**

**International College**

***King Mongkut’s Institute of Technology Ladkrabang***

**Thesis Proposal**

Academic Year 2017

|  |  |
| --- | --- |
| **Project Title** | Packet Flow Classification |
| **Advisor** | Dr. Isara Anantavrasilp |
| **Co-Advisors (if applicable)** | - |
| **Students** | 60610023 Mr. Sovan Tep  ---------------------------------------------------  (Sovan Tep)  Date: ………/…………./……………. |
| **Advisor’s Agreement:** |  |
|  | ---------------------------------------------------  (Dr. Isara Anantavrasilp)  Date: ………/…………./……………. |

# Packet Flow Classification

## Abstract

Packet flow classification system is an application used for anomaly detection. This system can be done by using the security concept behavior-based of intrusion detection system(IDS). Many researches have been done in this area focusing on which features or algorithms should be used seeking for the high accuracy while the rapid increasing of the internet traffic as well as the network attack required fast monitoring application. There are several terms can be improved to get the better performance of the system, once is at the point where the feature extraction algorithms should be faster. Meanwhile, parallel computing in which many calculations or the execution of [processes](https://en.wikipedia.org/wiki/Process_(computing)) are carried out concurrently so that the faster algorithm can be obtained. The objective of this work is to proposed parallel computing method for faster feature extraction and applying machine learning algorithms to predict the anomalous activities in term of real-time IDS.

## Background

During the last decade, the rapid growth of the Internet has been observed. The global Internet traffic already reached zettabytes ( by the year of 2016 and has been expected to be twice of that by the year 2019 regarding to the Cisco report [1]. As the Internet traffic is being grown, the problems are also coming along with those innovation such as Internet attack which is main concern in Internet globalization. For example, the increased proportion of the attacks targeting TCP applications are 18%, and 75% over 50-100 Gbps attacks occurred in US and Canada respectively as 99.2% were TCP SYN attacks in Q2 2015 [2]. Hence, it is the serious problem needed to be tackled for the overall companies and organizations to get equipped with the appropriate system for attack detection.

Intrusion detection system (IDS) play an important role in network security protection architecture. Generally, there are two types of approach of Network Intrusion Detection System(NIDS) are signature-based detection and behavior-base detection. Signature-based detection, it compares the new data with intrusion-base knowledge, yet the fact that this approach cannot recognize the new type of attacks, it is mostly used in commercial intrusion detection system. Behavior-based detection, compares the new data with the normal user behavior model, analyze the flow from this model as an anomaly using machine learning. Therefore, this approach can be used machine learning algorithms to analyze the network traffic for identifying malicious attack behaviors in zero-day attack. Recently, many works on IDS have been done only focusing in the point of accuracy and detection rate while the increasing of the traffic volume caused the several challenges of performance to implement real-time IDS.

To overcome the aforementioned problems, parallel computing is one of the technique which can be used to code efficiently. Since parallel programming saves time, allowing the execution of applications in a shorter wall-clock time. As a consequence of executing code efficiently, parallel programming often scales with the problem size, and thus can solve larger problems. In general, parallel programming is a means of providing concurrency, particularly performing simultaneously multiple actions at the same time.

## Literature Review

Recently, many researchers have proposed and developed numerous of approaches to detect intrusion. Some have been concerning on the data preprocessing based on the features selection problems which aimed to reduce dimension of the data and some other researchers have been focusing on the classification problems to obtains the minimal false intrusion detection. By the way, the one main problem that should be also concerned is how we can speed up the performance of the system.

In [1], they proposed a distributed architecture-based intrusion detection system(IDS) that is capable of detecting the anomalies in the network in real-time. They applied the Apache Spark framework and Netmap. In the end, the proposed system result in the better performance when the number of worker added. However, the requirement of the hardware implementation of the system is at high cost since many virtual machines is needed to simulate the distributed system.

In this project, we have provided a solution to solve the mentioned issues by implementing the parallel computing which many calculations or the execution of [processes](https://en.wikipedia.org/wiki/Process_(computing)) are carried out concurrently so that the faster algorithm can be obtained

## Methodology

### 3.1 Architecture

****

Figure 1 architecture work flow

### 3.2 Dataset

The KDD cup is widely used more than 15 years in the academic research and even in data mining competition in the world. NSL-KDD [2], the replicated dataset and has a completely record of KDD, was proposed since it has some advantages over the original KDD data set. There are 148,517 flows in the NSL-KDD dataset, each traffic sample has forty-one feature and available for both binary class and multi-class. The flows are labeled as normal or abnormal in the binary class while the flows are labeled as normal or any different types of attack in the multi-class.

### 3.3 Proposed Methodology

Above figure 1 showed the whole process of IDS architecture base on real-time. There are four steps to complete the whole process are Raw Network packet capturing, Feature extraction, Data preprocessing, and classification step. Network packet capturing is the computer network method to intercept the data packet being send over the specific network. Once a packet is captured, it can be stored as temporary data in a specific place waiting for being analyzed by the system. Feature extraction is the process of transforming the input data into the specific feature set. In our case, we want to extract the raw packet captured in the previous step into NSL-KDD feature set. Next step is data preprocessing used to convert the feature set into smaller size by eliminating the inconsistent or unnecessary data, so that only the high quality or effective data remain in the dataset, this help to improve the performance of the learning model. Last is the classification process, with its algorithm we can take the existing data to generate a predictive model for use in classifying the future data point.

In order to build such system with the high performance and accuracy. There is one specific block of the process where we want to work on to reach the objective of this project. We have focused on speed up the feature extraction by using the parallel computing method. Parallel computing is a type of computing architecture that process or execute the application simultaneously. The primary objective the of the parallel computing is to help accelerate the application processing or task resolution with the available computation power.

## References

[1] A. M. Karimi, Q. Niyaz, S. Weiqing, A. Y. Javaid, and V. K. Devabhaktuni, "Distributed network traffic feature extraction for a real-time IDS," in *2016 IEEE International Conference on Electro Information Technology (EIT)*, 2016, pp. 0522-0526.

[2] N. Paulauskas and J. Auskalnis, "Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset," in *2017 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 2017, pp. 1-5.

[3] Cisco Visual Networking Index, “http://tinyurl.com/mev32z8.”

[4] Arbor Networks Attack Data, “http://tinyurl.com/htpbl95.”

[5] openMP, “http://www.openmp.org/”.