**Sudan University of Science and Technology**

**Collage of Engineering**

**School of Electronic Engineering**

# Flow based intrusion detection system

*A Research Submitted in Partial fulfillment for the Requirements of the Degree of B.Sc. (Honors) in Electronics Engineering*

## Prepared by:

1 .Lubna Hassan Fadul

2. Shaima Idriss Abu Algasim

3.Hind Abubakr Abd-alrahman

## Supervisor by:

**Dr. Ahmed Abdalla**

October 2016

# الإستهلال

بسم الله الرحمن الرحيم

**قال تعالى:**

﴿ وَلَقَدۡ كَرَّمۡنَا بَنِيٓ ءَادَمَ وَحَمَلۡنَٰهُمۡ فِي ٱلۡبَرِّ وَٱلۡبَحۡرِ وَرَزَقۡنَٰهُم مِّنَ ٱلطَّيِّبَٰتِ
وَفَضَّلۡنَٰهُمۡ عَلَىٰ كَثِيرٖ مِّمَّنۡ خَلَقۡنَا تَفۡضِيلٗا ﴾

صدق الله العظيم

سورة الإسراء الآية ﴿70﴾

# DEDICATION

***__To our parents__, who directed my first step in the right way, taught us to trust in Allah, believe in our self and that so much could be done with little...*

***__To our brothers and sisters__...*

*** __To our friends__ who are always ready to help...*

***__To our colleges__...*

***We dedicate this work....***

# ACKNOWLEDGMENTS

Our grateful thanks firstly to our God who guided us to the strait way in our life. Then many thanks and appreciations are extended to our Supervisor **Dr.Ahmed AbdAlla**for his valuable advices and endless efforts to make this work come into reality. A lot of thanks is given to the School of Electronic Engineering.

# ABSTRACT

The use of packet based NIDS is expensive because each packet must be inspected deeply. This research provides solution for discovering network attacks in efficient manner using flow based network intrusion detection system. The designed system closely monitors the internet traffic based on some time-based aggregated traffic (TAT) features to determine existence of brute-force attack .These TAT features are extracted from a previously dataset of NetFlow records using a C code program. The designed system provides a property of discovering attacks with undefined signature (unknown attacks).The obtained result shows reduction in false alarm and high level security provided by this system.

# المستخلص

استخدام نظام اكتشاف الاختراقات استناداً علي محتويات الحزمه مكلف لأنه يتطلب فحص دقيق لكل محتويات الحزمه. هذا البحث يوفر حل لاكتشاف الاختراقات التي تحدث في الشبكات بطريقه ذات كفاءة وذلك باستخدام نظام اكتشاف الاختراقات علي الشبكة استناداً على بعض المعاملات. هذا النظام يقوم بصوره محكمة بمراقبة حركة البيانات على الانترنت بناءً على استخلاص خصائص لبيانات مجمعة في فترات زمنية معينة عن طريق برنامج سي بإستخدام مجموعة بيانات من سجلات بروتوكول محدد مسبقا لتحديد ما اذاكان هنالك اختترق ام لا . النظام بتوفير خاصية اكتشاف اللإختراقات التي لا يكون لديها نمط محدد و معروف ، النتائج التي تم الحصول عليها تشير الى انه تم تقليل نسبة الانذار الخاطئ وتوفير درجة عالية من الأمان بإستخدام هذا النظام.

# TABLE OF CONTENTS

## Chapter three Research Methodology



## Chapter Four result



## Chapter Five Conclusion and Recommendations

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATION

| | |
|---|---|
| IDS | Intrusion Detection System |
| NIDS | Network Intrusion Detection System |
| HIDS | Host Intrusion Detection System |
| IPFIX | Internet Protocol Flow Information Export |
| DOS | Denial Of Service |
| DDOS | Distributed Denial Of Service |
| FPR | Fouls Positive Rate |
| TPR | True Positive Rate |
| AVG | Average |
| STDEV | Standard Deviation |

# Chapter one

# INTRODUCTION

## 1.1 Back ground

Intrusion Detection can be defined as " security system that monitors computer systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization"[1]. It detects unwanted exploitation to computer system, both through the Internet and Intranet.

In general, we can divide IDSs into two basic classes based on their position in the network: host-based IDSs (HIDSs) and network-based (NIDSs). IDSs also can be classified based on its detection model into two categories: signature-based and anomaly-based. Signature-based IDS maintains a database of known intrusion technique (attack signature) and detects intrusion by comparing behavior against the database. Disadvantage of this technique is ineffective against previously unseen attacks and hence it cannot detect new and unknown intrusion methods as no signatures are available for such attacks. Anomaly based NIDS monitors network traffic and compares it against an established baseline of normal traffic profile. The baseline characterizes what is "normal" for the network - such as the normal bandwidth usage, the common protocols used, correct combinations of ports numbers and devices to detect malicious traffic, parameters like number of connection request, number of rejected connection ,average packet size, flag values present in the packet headers are used. But due to the dynamic nature of network traffic and application access, this type of detection can generate false alarms.

Required challenge in anomaly detection is the volume of data for analysis. Collection and analyses of network traffic information at packet level for a high-speed network to provide accurate result in real-time is a difficult task. In case of high speed network, packet-level traffic monitoring is expensive because of deep packet inspection requirement and those intrusion detection systems can detect only known attacks based on signatures. Due to these reasons, flow based traffic monitoring and Anomaly detection is important.

## 1.2 Problem statement

Collection and analyses of network traffic information at packet level for a high-speed network to provide accurate result in real-time is a difficult task. In case of high speed network, packet-level traffic monitoring is expensive and it is very time consuming because of deep packet inspection requirement.

Signature-based IDSs cannot detect unknown attacks, either because the database is out of date or because no signature is available yet, and they cannot detect some type of attacks such as scan, flood, DoS and DDoS because they have no signature pattern and they appear as abnormal behavior in the network.

## 1.3 Proposed solution

The proposed solution for the problems is by using a flow based anomaly network intrusion detection system. The system use the "flow-level NIDSs", in which rather than looking at all packets going through a network link, it looks at aggregated information of related packets of network traffic in the form of flow, so the amount of data to be analyzed is reduced.

Also the system use the "Anomaly-based or behavior-based method" which works by building a model of normal traffic data pattern during a training phase, then it compares new inputs to the model. A significant deviation (change) is marked as an anomaly (abnormal or intrusion), so this method is able to detect unknown attacks.

## 1.4 Research aim

The aim of this project is to design and implement a flow based anomaly network intrusion detection system that can detect Denial of Service, flooding, host and port scan attacks.

## 1.5 Research Outline

This thesis is organized as follow: Chapter two introduces the network intrusion detection system and its types. In addition, the chapter reviews types of attacks. In chapter three, the design of "flow based network intrusion detection system" was demonstrated. In chapter four the results were showed and discussed. In chapter five, the conclusion and recommendations were presented.

# Chapter two

# Literature Review

## 2.1 Introduction

The rapid proliferation of computer networks has changed the prospect of network security. An easy accessibility condition cause computer network's vulnerable against several threats from hackers. Threats to networks are numerous and potentially devastating. Recent reports on Internet security breaches indicate that the frequency and he damage costs are continuously rising. Up to the moment, researchers have developed Intrusion Detection Systems (IDS) capable of detecting attacks in several available environments.

## 2.2 Intrusion detection system

An IDS is best defined as software or hardware used to detect unauthorized traffic or activities that are against the allowed policy of a given network [2]. IDS do it by collecting data from network and analysis of transmitted packets inside the network. But generally IDSs do not act operative reaction against occurred attacks. IDS have many classification based on several aspects.

### 2.2.1 Based on detection model

If a system bases the detection on a definition of *normal* behavior of the target system, it is called *behavior-based*. If it matches the input data against a definition of an attack, it is known as *knowledge-based*. In literature, the community usually refers to these classes with the names of *anomaly-based* and *misuse-based* [3].

#### 2.2.1.1 Signature based

The signature-based IDSs, also named "misused-based", works similar to anti-virus software. It employs a signature (pattern that correspond to a known threat) database of known attacks, and if a successful match with current input, an alert is raised. A well-know example of this type is Snort [4] which is an open source IDS that monitors network by matching each packet it observes against a set of rules.

Signature based method is very useful for known attacks, Although Signature-based IDSs cannot detect unknown attacks, either because the database is out of date or because no signature is available yet, it has low false alarms (high accuracy), Signature based systems are reactive, in that they combat against known attacks, that have already affected and damaged a number of systems before being identified.

#### 2.2.1.2 Anomaly based

Anomaly detection is an active area in network intrusion detection research which was originally proposed by Denning [5]. It can detect various types of intrusion based on the deviation in the normal usage of network and this has an advantage over signature based technique.

Anomaly-based or behavior-based IDS works by building a model of normal traffic data pattern, then it compares new inputs to the model. A significant deviation (change) is marked as an anomaly (abnormal or intrusion). Anomaly-based is able to detect unknown attacks but it suffers from producing false alarms [4]. It refers to finding out the abnormal pattern of traffic or abnormal behavior from network or system.

### 2.2.2 Based on their position in the network

Most traditional intrusion detection systems (IDS) take either a network- or a host-based approach to recognizing and deflecting attacks. Each approach has its strengths and weaknesses.

#### 2.2.2.1 Host based

Host-based intrusion detection systems are aimed at collecting information about activity on a particular single system, or host[6] . The term "host" refers to an individual computer.

#### 2.2.2.2 Network based

Network-based intrusion detection systems offer a different approach; "These systems collect information from the network itself," [6] rather than from each separate host. information is collected from the network traffic stream, as data travels on the network segment [6].

Network-based systems are extremely portable. They only monitor traffic over a specific network segment, and are independent of the operating systems that they are installed on. There are two methods basis on the source of data to be analyzed in NIDSs:

- **Packet based**

In packet-based, also named "Deep Packet Inspection" (DPI), the combination of header and payload scan determines whether a packet is an intrusion or not. Incoming packets are scanned and every single rule of the database is checked against it as shown in figure 2-1. The database rules include thousands of signatures and patterns of attacks.

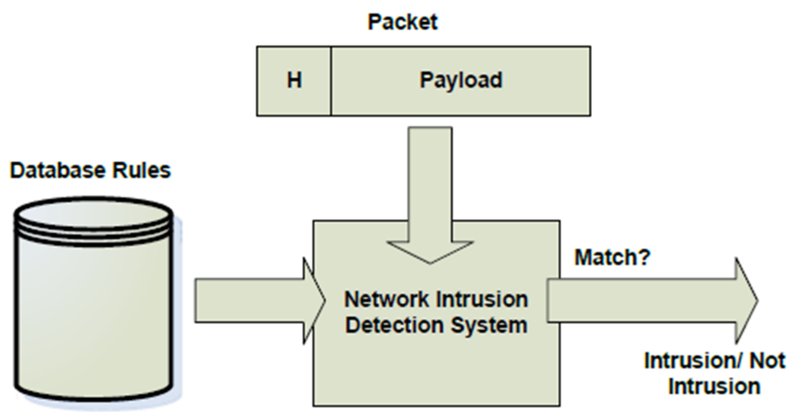


**Figure 2.1:** packet based IDS

The main advantage of packet-based approach is that all common kinds of known attacks and intrusions practically can be detected if the data source deliver entire network packet for analysis. However, it cannot detect unknown attack since it compare with predefined and known malicious signatures [4].

However, systems that are capable of monitoring every packet on a high-speed network are very expensive and high resource consumption. Moreover, a drop of packets will occur if the NIDSs speed is not high enough to let the analysis process be done.

In signature-base detection, as the number of attacks increase, the number of malicious (intrusion) signatures increase in the database in NIDSs. Usually, these databases contain hundreds or thousands of signatures. NIDS has to add these new signatures into its signature list quickly without disturbing its main function of detecting intrusion. NIDSs then search for these signatures in network traffic to detect intrusions. To detect signatures, all network traffic must be compared against every signature to identify if a match exists or not. Therefore, the efficiency in accessing such database for analyses is also critical. Another issue is that signature matching is impossible for most cases of encrypted payload, degrading the detection performance of NIDSs. A comprehensive evaluation of packet-based performance with high volume NIDSs is presented in [4].

- **Flow based**

For high speed networks, it is important to explore alternative to packet-based inspection for efficient NIDSs. One option that currently attracts the attention of researchers is flow-based intrusion detection. Flow-based technique is widely deployed as data source in applications like network monitoring, traffic analysis and security [4]. This method is characterized by flow data or network flow. Flows don't provide any packet payload unlike packet-based approach, as shown in figure 2.2.

It rather relies on information and statistics of network flows. A flow can be defined as a unidirectional data stream between two computer systems where all transmitted packets of this stream share the following characteristics: IP source and destination address, source and destination port number and protocol value [4]. Nowadays special measurement systems are able to provide other characteristics in addition to the above, for instance:

- The number of packets and amount of bytes transferred in a flow.
- The start and end time of a flow (in milli-second).

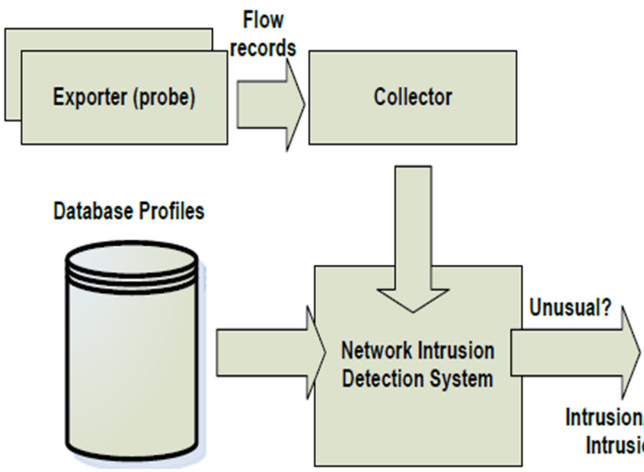


**Figure 2.2:** flow based IDS

## 2.3 network attacks

Network and computer attacks have become pervasive in today's world. Any computer connected to the Internet is under threat from viruses, worms and attacks from hackers. Home users, as well as business users, are attacked on a regular basis. Thus the need to combat computer and network attacks is becoming increasingly important.

Flow-based intrusion detection, since it relies only on header information, can address only a subset of the attacks presented above. In particular, the research community currently provides approaches to detect the following classes of attacks:

- Denial of Service.
- Scans.
- flooding.

#### 2.3.1.1 Denial of Service attack

A Denial of Service attack attempts to slow down or completely shut down a target so as to disrupt the service and deny the legitimate and authorized users can access [7]. Such attacks are very common in the Internet where a collection of hosts are often used to bombard web servers with dummy requests illustrated in Figure 2.3. Such attacks can cause significant economic damage to ecommerce businesses by denying the customers an access to the business. There are a number of different kinds of DoS attacks, some of which are mentioned below. It is shows a denial of service attack (DDoS in this case), wherein an attacker uses a number of compromised hosts to attack a given victim.

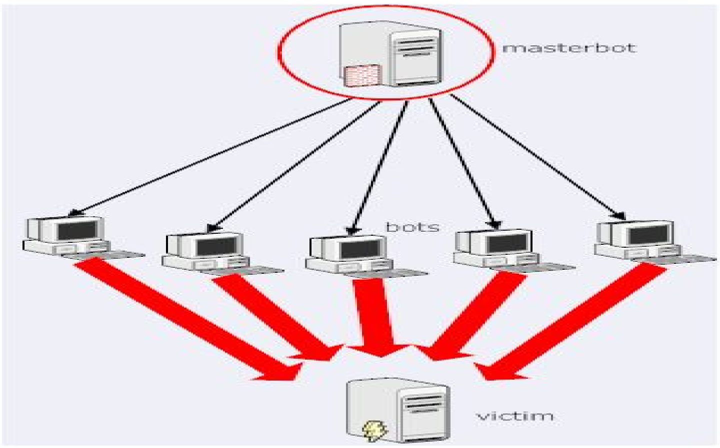


**Figure 2.3:** Denial of Service attack

- **Flaw Exploitation DoS Attacks**

In such attacks, an attacker exploits a flaw in the server software to either slow it down or exhaust it of certain resources. Ping of death attack is one such well known attack. A ping of death (POD) is a type of attack on a computer that involves sending a malformed or otherwise malicious ping to a computer. A ping is normally 64 bytes in size (or 84 bytes when IP Header is considered); many computer systems cannot handle a ping larger than the maximum IP packet size, which is 65,535 bytes. Sending a ping of this size can crash the target computer. Some limitations of the protocol implementation also lead to vulnerability which can be exploited to implement DoS attacks such as DNS amplification attack, which uses ICMP echo messages to bombard a target. For these attacks, a signature can be devised easily, such as to determine a ping of death attack a NIDS needs to check the ping flag and packet size.

- **Flooding DoS Attacks**

In a flooding attack, an attacker simply sends more requests to a target that it can handle. Such attacks can either exhaust the processing capability of the target or exhaust the network bandwidth of the target, either way leading to a denial of service to other users. DoS attacks are extremely difficult to combat, as these do not exploit any vulnerability in the system, and even an otherwise secure system can be targeted. A more dangerous version of DoS attack is called Distributed Denial of Service attack (DDoS), which uses a large pool of hosts to target a given victim host. A hacker (called botmaster) can initiate a DDoS attack by exploiting vulnerability in some computer system, thereby taking control of it and making this the DDoS master (Figure 2-3). Afterwards the intruder uses this master to communicate with the other systems (called bots) that can be compromised. Once a significant number of hosts are compromised, with a single command, the intruder can instruct them to launch a variety of flood attacks against a specified target [7].

#### 2.3.1.2 Scanning Attack

In such attacks, an attacker sends various kinds of packets to probe a system or network for vulnerability that can be exploited. When probe packets are sent the target system responds; the responses are analyzed to determine the characteristics of the target system and if there are vulnerabilities (illustrated in Figure 2-4) where a single attack host scans a number of victims. Thus scanning attack essentially identifies a potential fused which yields these information:

- The network topology.
- The type of firewall used by the system.
- The identification of hosts that are responding.
- The software, operating systems and server applications that are currently running.
- Vulnerabilities in the system.

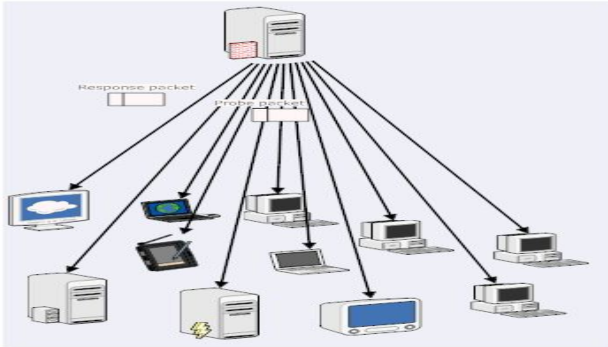


**Figure 2.4:** scan attack

Once the victim is identified, the attacker can penetrate them in a specific way. Scanning is typically considered a legal activity and there are a number of examples and applications that employ scanning. The most well known scanning applications are Web search engines. On the other hand independent individual ay scan a network or the entire Internet looking for certain information, such as a music or video file. Some well-known malicious scanning include Vertical and Horizontal port scanning, ICMP (ping) scanning, very slow scan, scanning from multiple ports and scanning of multiple IP addresses and ports. NIDS signatures can be devised to identify such malicious scanning activity from a legitimate scanning activity with fairly high degree of accuracy [7].

### 2.3.1.3 Flooding attack

- **Ping flood and ping of death:**

Ping flood is similar to Smurf where in the victim is bombarded with thousands of ping packets. In Ping of death, the victim is sent corrupt packets that could crash the system [8]. Smurf and ping floods are very easy to craft and any novice attacker could do it with ease. These attacks could cause considerable damage in small Local Area Networks.

- **UDP flood:**

UDP flooding is similar to ping flood. Here instead of ping packets, UDP packets are bombarded against the server. UDP could be a lot more effective than ICMP in smaller networks as the size of the UDP packets are enormous. The packet size could be set up to 65000 bytes which could easily flood a given Ethernet network when multiple zombies are set up. This project has analyzed all the above described attacks and has brought down some interesting observations.

## 2.4 Related work

Recently, instead of packet based analysis, flow based security analysis and anomaly detection are getting attention from many researchers. Mayung et al [4] suggests that by aggregating packets of the identical flow, one can identify the abnormal traffic pattern that appears during attack. They formalize detection function for attack detection, which are composed of several traffic parameters and constant value.

Another work based on flow monitoring is explained in[9] which work on monitoring the four predefined metrics that capture the flow statistic of the network. This method is capable to detect UDP flood, ICMP flood and scanning, by using Holt-Winters Forecasting technique. This technique makes projection about future performance based on historical and current data of the network. The prediction which comes out by this technique may arise false alarms because network behavior is not static.

Anomaly-based detection stated by [10] analyses user behavior and the statistics of a process in normal situation, and it checks whether the system is being used in a different manner. In addition [10] has described that this technique can overcome misuse detection problem by focusing on normal system behavior rather than attack behavior. However [10] assume that attacks will result in behavior different from that normally observed in a system and an attack can be detected by comparing the current behavior with pre-established normal behavior. This detection approach is characterized by two phases which is the training phase and detection phase. In training phase, the behavior of the system is observed in the absence of attack, and machine learning technique is used to create a profile of such normal behavior. In detection phase, this profile is compared against the current behavior, and deviations are flagged as potential attacks. The effectiveness of this technique is affected by what aspect or a feature of system behavior is learnt and the hardest challenge is to be able to select the appropriate set of features. The advantage of this detection technique is that it can detect new intrusion method and capable to detect novel attacks. However, the disadvantage is that it needs to update the data (profiles) describing the user's behavior and the statistics in normal usage and therefore it tend to be large and therefore need more resources, like CPU time, memory and disk space. Moreover, the malware detector system often exhibit legitimate but previously unseen behavior, which leads to high rate of false alarm.

In the current trend, few researches such as [10]and[7] have been found to manipulate this detection technique by combining either Signature-based with Anomaly-based detection technique(Hybrid-SA) in order to develop an effective malware detector's tool.

A hybrid-based IDS stated by [4] make a tradeoff between availability of limited data of flow-based techniques, which have negative effect on accuracy of NIDSs, and full data of packet-based which lead to a higher resources consumption. Therefore flow-based detection should not substitute the packet inspection one. However, combination approach to combine both approaches to power their advantages and overcome their drawbacks is proposed. Flow-based technique takes advantage of packet-based technique to reduce false alarms. We therefore expect a potential in mixing both approaches to detect at least the same quantity of attacks while consuming less resources.

## 2.5 Summary

There are problem that some attacks have traffic patterns that cannot be characterized by only one flow. To detect this type of attack, we need traffic information that can identify traffic patterns. Aggregating related flows can generate this information, which is called traffic pattern data. By examining parameters of traffic pattern data we can discover traffic used in attacks, such as flooding and scanning.

# Chapter Three

# Research Methodology

## 3.1 Introduction

This chapter describes the implementation details of developing and implementation of the proposed flow based anomaly detection system.

## 3.2Research Activities

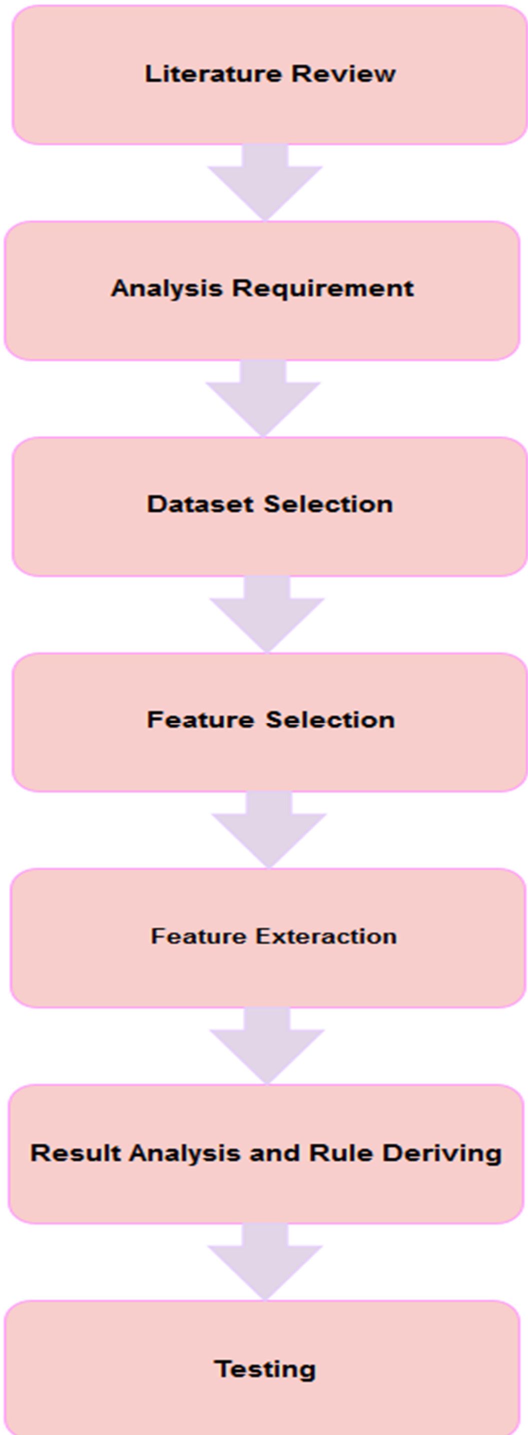

**Figure 3.1:** Research activities sequence

- **Literature Review**

We reviewed a number of scientific papers related with our proposed system and follow major related works, Theories and hypotheses mentioned in the literature.

- **Analyze requirements**

Specification of Dataset was been determined.

- **Data selection**

The selection of dataset depends on [11].

- **Feature selection**

We select some feature from Dataset.

- **Feature extraction**

We extract the selected feature in previous step for TAT instead of flow record.

- **Result analysis and rule deriving**

Analyze results for labeling TAT to obtain good result.

- **Testing**

Training rules with labeling TAT record to improve accuracy of rules.

## 3.3 Flow based anomaly detection system

This section is divided into two parts .The first part describes the characteristics of our selected dataset as prerequisites for system ,The second part describes the implementation details of designing an effective NIDS that is capable to detect brute-force attacks. In the following subsections we will describe and discuss all processes and implementation details.

### 3.3.1Dataset

Our project depend on pre-prepared dataset by[11], [11] explained clearly and in details all the processes and procedures taken to create flow-level dataset.This dataset has ten attributes, but we select eight of them, as illustrated in Table 3.1.

**Table 3.1:** Final Eight Attributes selected for datasets

| | attribute | attribute Description |
|---|---|---|
| 1 | Stime | Unix time of the first packet in the flow |
| 2 | etime | Unix time of the last packet in the flow |
| 3 | SrcIP | source IP |
| 4 | DstIP | destination IP |
| 5 | SrcPT | TCP/UDP source port number or equivalent |
| 6 | DstPT | TCP/UDP destination port number or equivalent |
| 7 | pkt | Number of Packets in the flow |
| 8 | Pyt | Number of Layer 3 bytes in the packets of the flow |

Labeling means determines if the flow record is malicious or free attack. As [11] mention, each malicious flow is assigned three digits to identify the specific attack type. Table 3.2 shows this labeling method. A label of "223" means that the attack is a flood because the first digit is "2", DDoS because the second digit is "2" and the used protocol is UDP because the third digit is "3".

**Table 3.2:** Three Digit labeling method of malicious flow

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **FIRST DIGIT:** main class attack | | | "1" ↓ scan | | | "2" ↓ flood | |
| **SECOND DIGIT** specific type into main class | | | "1" ↓ network scan | "2" ↓ port scan | "3" ↓ protocol scan | "1" ↓ DoS flood | "2" ↓ DDoS flood |
| **THIRD DIGIT** used protocol | 1→ | ICMP | ☑ 111 | | | ☑ 211 | ☑ 221 |
| | 2→ | TCP | ☑ 112 | ☑ 122 | | ☑ 212 | ☑ 222 |
| | 3→ | UDP | ☑ 113 | ☑ 123 | | ☑ 213 | ☑ 223 |
| | 0→ | OTHER | ☑ 110 | ☑ 120 | ☑ 130 | ☑ 210 | ☑ 220 |

### 3.3.2 Flow-Level Brute-Force Attack Detector

The objective of this part is to aggregate previously prepared NetFlow records in fixed time windows and extract traffic features that give optimum detection result according to the selected detection method and finally to come out with rules for detection and identification of brute-Force Attack in near real time. The proposed system is presented in Figure 3.2. As illustrated, the system has the following main sequential processes Time-based Aggregation of Traffic (TAT), labeling TAT records, developing of a detection model based on the TAT records and testing.

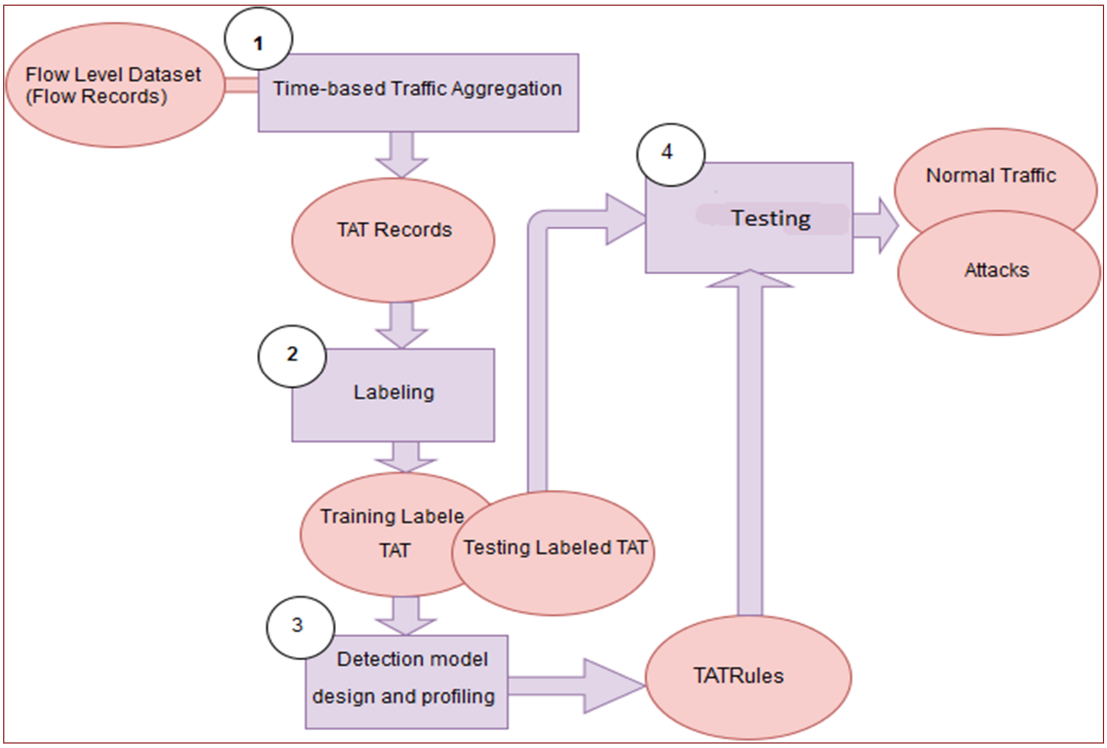


**Figure 3.2:** Designing and Testing of the Flow-Level Brute-Force Detector.

#### 3.3.2.1Time-based Aggregation of Traffic

The *flow-level dataset* entity which appears in Figure 3.1as input of the *Time-based Aggregation of Traffic* process is the same brute-force dataset which had been described in previous section. In this process, we group NetFlow records of the brute-force dataset into window of t second then we extract various statistical metrics which we call Time-based Aggregated Traffic (TAT) features. Figure 3.3represents a conceptual diagram showing this process. We chose time window for our experiments 60 sec, we wrote a C program to implement the described process (**see Appendix A**). Figure 3.4 illustrates the flow chart of that program.

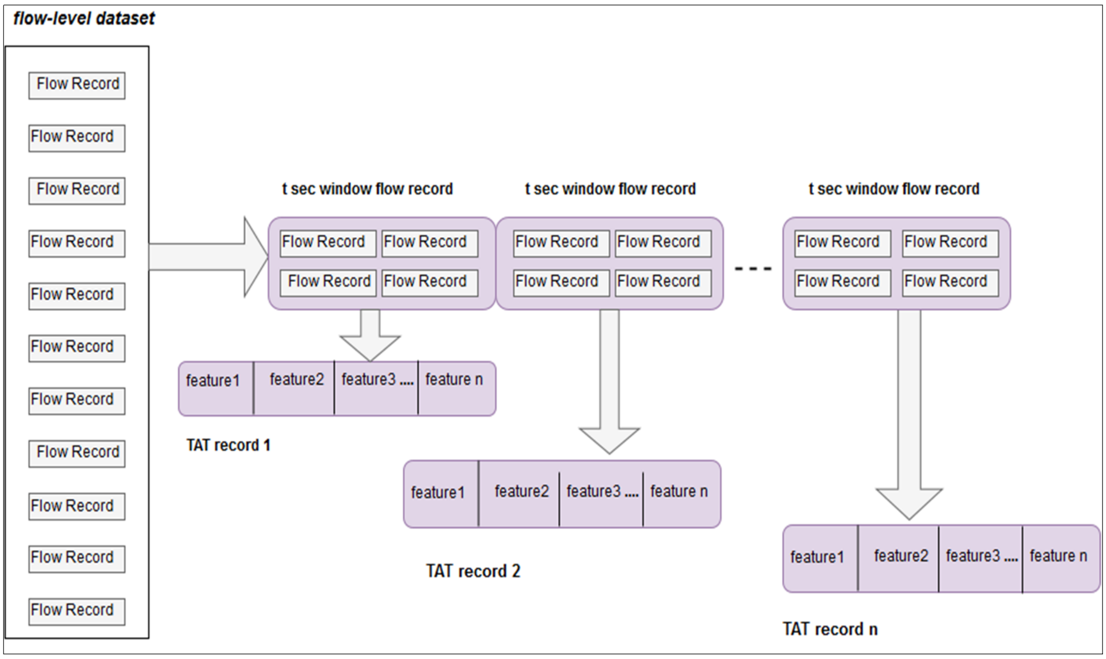


**Figure 3.3**Time-based Aggregation of Traffic

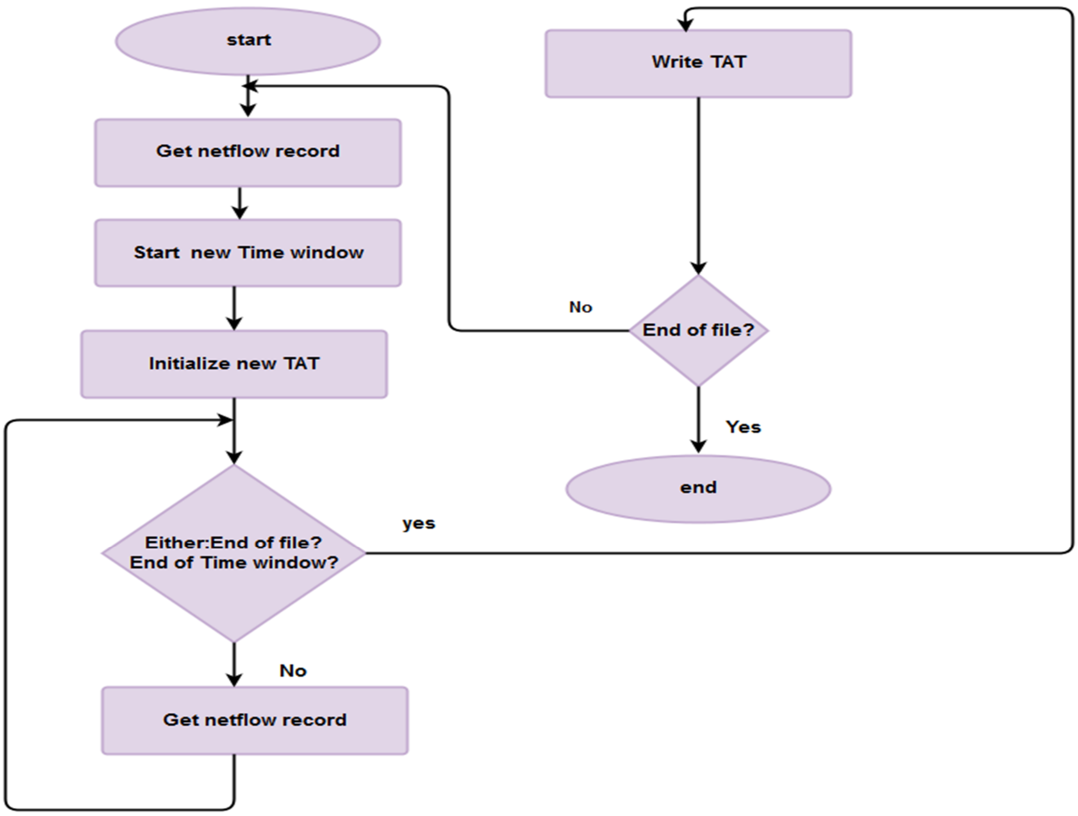

**Figure 3.4:** flow chart of Time Aggregation of Traffic

#### 3.3.2.2 Labeling TAT Records

In labeling TAT there is a problem of how to classify a TAT record when it contains both classes of benign and malicious flows. To deal with this problem, we chose to label a whole TAT record as malicious if the record satisfies that the number of malicious flows is 5o flows or greater. We modified the previous written program that implements Traffic Aggregation labeling for each TAT (**see Appendix A**). The flow chart of the modified program is illustrated in Figure 3.5

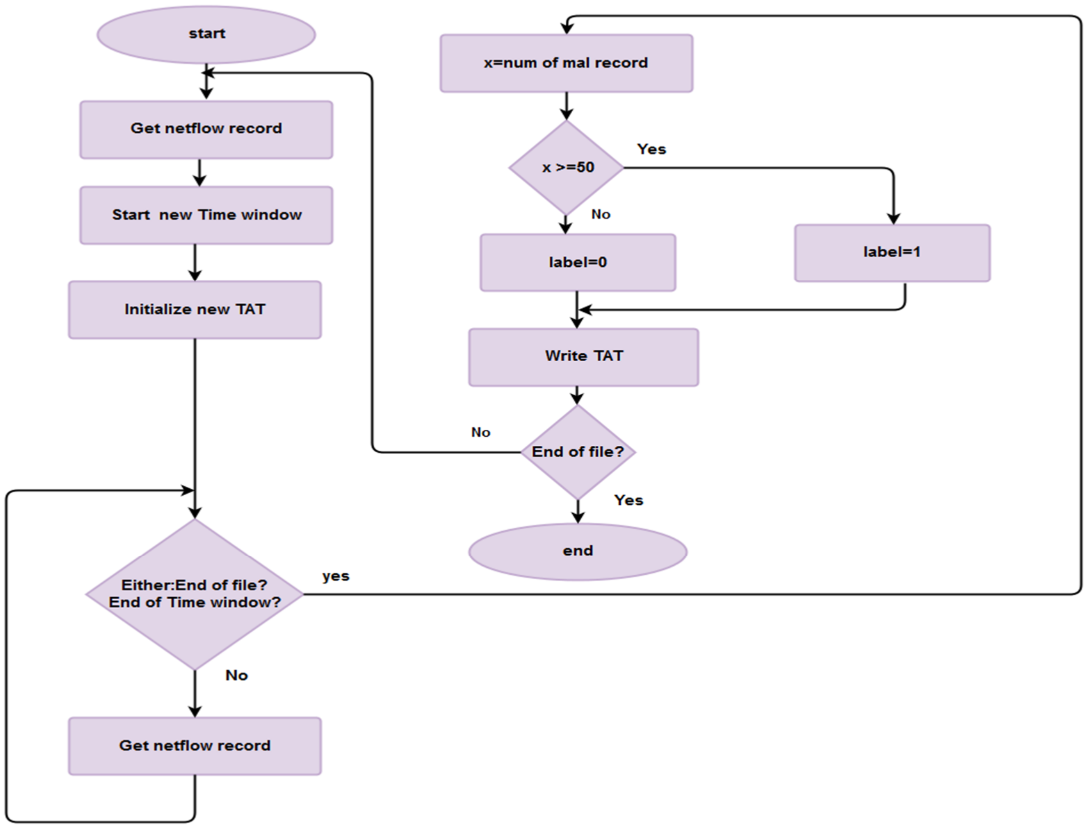


**Figure3.5:** Flow Chart of Traffic Aggregation of Traffic and TAT record Labeling**.**

#### 3.3.2.3 Detection Model Design and Profiling

Design of the detection model is done in three sequential steps. The first phase is the selection of significant TAT attributes that reflect traffic variations of brute-force attacks. The second step is determining threshold values that differentiate normal and malicious TAT records. The final step is to formulize detection rules from combinations of these threshold values.

- **Selection of Significant TAT attributes**
  We may assess the effect of each feature in detecting attacks by analyze the labeled TAT records to study the effect of each attack over TAT attributes.
- **TAT statistics and Profiling**
  For detection process we calculated statistical features (AVG, STDEV) for each TAT attribute to characterize, differentiate and identify normal TAT records and malicious TAT records.
- **Threshold Values and Detection Rules**
  We use the combination of (AVG, STDEV) for TAT attribute as threshold, then detection rule will explain in next chapter.

#### 3.3.2.4 Testing

This testing is done on labeled TAT records to check the detection method and identification accuracy in terms of true positive and false positive for brute force attack. We implement the constructed attack detection rules showed in the previous section. We checks a TAT record whether it is benign or malicious.

# Chapter four

# Results

## 4.1 Introduction:

In this chapter we illustrate result and output of each phase in proposed system.

## 4.2 Result of Time-based Aggregation Traffic

Table 4.1 lists and describes the full attribute set (TAT record fields) extracted out of one time window of NetFlow records.

**Table 4.1:** Extracted Features of Time Window NetFlow Records

| | TAT feature | Description |
|---|---|---|
| 1 | #flows | number of flows |
| 2 | #2pflow | number of flows that contain 1 to 2 packets |
| 3 | #pkts | number of packets |
| 4 | #byts | number of bytes |
| 5 | #srcIP | number of distinct source IP addresses |
| 6 | #dstIP | number of distinct destination IP addresses |
| 7 | #srcPt | number of distinct source Ports |
| 8 | #dstPt | number of distinct destination Ports |

## 4.3 Result of Labeling TAT Records

After labeling phase the TAT records statistics shown in Table 4.2 calculated from result obtained from C program (see Appendix B).

**Table 4.2:** Statistics of 60 seconds window TAT records

| TAT records statistics | 60 sec TAT |
|---|---|
| Malicious TAT records | 49 |
| free attack TAT records | 74 |

## 4.4 Result of Detection Model Design and Profiling

Statistical from Significant TAT attributes and threshold used in rule for detection illustrated in Table 4.3.

**Table 4.3:** TAT attributes threshold

| Attribute | Threshold |
|---|---|
| # flows | 18417 |
| # srcIP | 1408 |
| # srcPt | 7534 |
| # pkt2flow | 41 |

Where:

pkt2flow=2pflow/ flows.

After experiment and observation, we notice that the features in Table 4.3 have more effect as shown in figure 4.1 this features had been used to formed suitable rule.

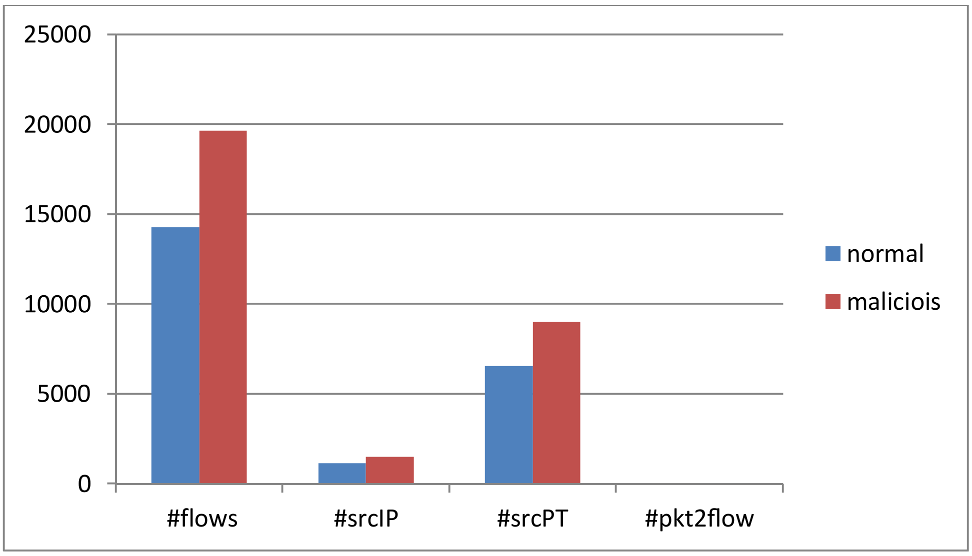

**Figure 4.1:** TAT Attributes for Normal and Brute force attack (Average Value)

**The Rule:**

**IF((flows>=18417 AND srcIP>=1408) OR (**pkt2flow <=41 AND srcPt >=7534**))**

**Label =1**

**else**

**Label=0**

## 4.5 Result after Testing Rule

After applied rules of detection method with labeled TAT record we obtain the result shown in Table 4.4

**Table 4.4:** accuracy of detection rule

| TPR | FPR |
|---|---|
| 80% | 8% |

# Chapter Five

# Conclusion

## 5.1 Conclusion

A flow based detection system is proposed in this research. The proposed system provides a generic solution for detecting network anomalies like scan and flood (Brute force attack) for high speed network. The achieved results were satisfactory with ratio 86.99%.

## 5.2 Recommendations

This research has provided report on the design process of our system; definitely we need more study to improve reliability, capability and accuracy of our system. The following improvements can be recommended for possible future work:

- Another statistical method can be used, like: chi squire .
- More features can be used, or combination of any effective features.
- This system can be implemented online.
- Determining the classification of the attacks types can be added.
- Using more time windows and select the optimum one, will give more accurate result.

# REFERENCE


[1] I. D. Systems, “Interested in learning SANS Institute InfoSec Reading Room Intrusion Detection Systems : Definition , Need and tu , A ll r igh ts.”

[2] M. Alenezi and M. J. Reed, “Methodologies for detecting DoS / DDoS attacks against network servers,” no. c, pp. 92–98, 2012.

[3] S. S. Rajan and V. K. Cherukuri, “An overview of intrusion detection systems,” *Retrieved May*, vol. 12, no. 3, pp. 559–563, 2010.

[4] H. Alaidaros, M. Mahmuddin, and A. Al Mazari, “From Packet-based Towards Hybrid Packet-based and Flow-based Monitoring for Efficient Intrusion Detection: An overview,” *Iccit*, pp. 844–849, 2012.

[5] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection † 2 Evaluation of Intrusion Detection Systems.”

[6] “Interested in learning more ?,” no. Security 401.

[7] R. Srivastava, “Survey of Current Network Intrusion Detection Techniques,” *J. Inf. Eng. Appl.*, vol. 3, no. 6, pp. 27–33, 2013.

[8] Denial of service attacks and migration technique, “Interested in learning SANS Institute InfoSec Reading Room ", 2011.

[9] H. A. Nguyen, T. Van Nguyen, D. Il Kim, and D. Choi, “Network Traffic Anomalies Detection and Identification with Flow Monitoring,” 2008.

[10] Y. Robiah, S. R. S, M. Z. M, S. Shahrin, M. A. Faizal, and R. Marliza, “A New Generic Taxonomy on Hybrid Malware Detection Technique,” vol. 5, no. 1, pp. 56–61, 2009.

[11] A. Abdlla, S.Nour, "Dataset Generation and Network Intrusion Detection Dataset based on Flow Information", PhD thesis, UTM university, Nov 2015.

# APPENDIX

# A.APPENDIX A

# CODE OF AGGREGATION AND LABELING

```
#include<stdio.h>
#include<stdlib.h>

#define SIZE 100000 //maximum size of array
FILE *rf, *wf,*wr;
unsigned long no_flow
,sum_pkts,sum_bytes,pkt2,s[SIZE],y[SIZE],x[SIZE],
v[SIZE],usrcIP,udstIP,usrcPT,udstPT,
foundIPadrs,foundIPdst,foundptdrs,foundptadst,e,o
o;

int j=0,n,a,b,c,r,label;


struct FlowRecord {
    unsigned long    flowId;
     unsigned long    stime;
     unsigned long    stmsec;
     unsigned long    etime;
     unsigned long    etmsec;
     unsigned char    protocol;
     unsigned long    srcIP;
     unsigned long   srcPT;
     unsigned long    dstIP;
```

```
    unsigned long   dstPT;
    unsigned long    pkts;
    unsigned long    bytes;
    unsigned long    clss;
 };
struct FlowRecord FR;

void R_file();
void W_file();
void parameter();
void writeTAT();
void intializeTAT();

 int main(void)
{
if ((rf=fopen("D:/eflows2014-11-
13_15_43_05clss.csv", "r"))==NULL)//reading file
{
  printf( "Read file could not be opened\n" );
  exit(1);
}

if ((wf=fopen("D:/writingfile.txt",
"w"))==NULL)//writing file
{
  printf( "Write file could not be opened\n" );
  exit(1);
}
```

```
R_file();
while(!feof(rf)){
   intializeTAT();
    do{

        W_file();//calling function
        parameter();//calling function
        R_file();//calling function

    } while( !feof(rf) &&(r +60 >
FR.etime));//feof was test

writeTAT();
}

fclose(rf);
fclose(wf);

 return 0;
}

void R_file()//function read from file and write
on other file
{
fscanf(rf,"%lu,",&FR.flowId);//reading
fscanf(rf,"%lu,",&FR.stime);
fscanf(rf,"%lu,",&FR.stmsec);
fscanf(rf,"%lu,",&FR.etime);
```

```
fscanf(rf,"%lu,",&FR.etmsec);
fscanf(rf,"%u,",&FR.protocol);
fscanf(rf,"%lu,",&FR.srcIP);
fscanf(rf,"%lu,",&FR.srcPT);
fscanf(rf,"%lu,",&FR.dstIP);
fscanf(rf,"%lu,",&FR.dstPT);
fscanf(rf,"%lu,",&FR.pkts);
fscanf(rf,"%lu,",&FR.bytes);
fscanf(rf,"%lu,",&FR.clss);
}

void intializeTAT()
{
 no_flow=0
,sum_pkts=0,sum_bytes=0,pkt2=0,usrcIP=0,udstIP=0,
usrcPT=0,udstPT=0;

     r=FR.etime;
     oo=0;
}

void W_file()//function read from file and write
on other file
{
fprintf(wf,"%lu,",FR.flowId);
fprintf(wf,"%lu,",FR.stime);
fprintf(wf,"%lu,",FR.stmsec);
fprintf(wf,"%lu,",FR.etime);
```

```
fprintf(wf,"%lu,",FR.etmsec);
fprintf(wf,"%u,",FR.protocol);
fprintf(wf,"%lu,",FR.srcIP);
fprintf(wf,"%lu,",FR.srcPT);
fprintf(wf,"%lu,",FR.dstIP);
fprintf(wf,"%lu,",FR.dstPT);
fprintf(wf,"%lu,",FR.pkts);
fprintf(wf,"%lu,",FR.bytes);
fprintf(wf,"%lu\n",FR.clss);
}



void parameter()
{

no_flow++;
sum_pkts+= FR.pkts;
sum_bytes+=FR.bytes;
 if(FR.pkts<=2)
    pkt2++;

foundIPadrs = 0;
for (  n=0;n < usrcIP; n++ )
    {
        if(s[n]==FR.srcIP)
            {
           foundIPadrs = 1;
```

```
            break;
          }

      }
   if (foundIPadrs ==0){
      s[usrcIP]=FR.srcIP;
      usrcIP++;
   }

  foundIPdst = 0;
  for (  a=0;a < udstIP; a++ )
      {
          if(y[a]==FR.dstIP)
              {
            foundIPdst = 1;
             break;
          }

      }
   if (foundIPdst ==0){
      y[udstIP]=FR.dstIP;
      udstIP++;
   }

   foundptdrs = 0;
  for (  b=0;b< usrcPT; b++ )
      {
          if(x[b]==FR.srcPT)
```

```
            {
           foundptdrs = 1;
           break;
        }

    }
 if (foundptdrs ==0){
    x[usrcPT]=FR.srcPT;
    usrcPT++;
 }

 foundptadst = 0;
for (  c=0;c < udstPT; c++ )
    {
        if(v[c]==FR.dstPT)
            {
           foundptadst = 1;
           break;
        }

    }
 if (foundptadst ==0){
    v[udstPT]=FR.dstPT;
   udstPT++;
 }
```

```
if (FR.clss!=0)
 {
     oo++;
     }
}

 void writeTAT()
{



    if ((wr=fopen("D:/result.csv",
"a+"))==NULL)//reading file
{
  printf( "writing file could not be opened\n" );
  exit(1);
}
e=(oo*1.0/no_flow) ;

//if(oo>=1)
//    label=1;
//    else
 if(oo>= 50)
    label=1;
    else
    label=0;

fprintf(wr,"%lu,",no_flow );
fprintf(wr,"%lu,",sum_pkts);
```

```
fprintf(wr,"%lu,",sum_bytes);
fprintf(wr,"%lu,",usrcIP);
fprintf(wr,"%lu,",udstIP);
fprintf(wr,"%lu,",usrcPT);
fprintf(wr,"%lu,",udstPT);
fprintf(wr,"%lu,",pkt2);
fprintf(wr,"%lu,",oo);
fprintf(wr,"%lu,\n",label);
fclose(wr);
}
```

# B.APPENDIX B

# RESULT OBTAINED FROM C PROGRAM

| TAT_ID | flows | pckts | bytes | srcIP | dstIP | srcPT | dstPT | 2pflow | 2pflow/flows | mal | label | RULE | TP | FP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 20677 | 848356 | 540325054 | 1408 | 5519 | 8579 | 6379 | 6973 | 34 | 376 | 1 | 1 | 1 | 0 |
| 1 | 17990 | 579621 | 297161470 | 1547 | 5656 | 7752 | 5693 | 7376 | 41 | 83 | 1 | 1 | 1 | 0 |
| 1 | 15508 | 484412 | 257453804 | 1374 | 4657 | 6715 | 5044 | 5657 | 36 | 80 | 1 | 0 | 0 | 0 |
| 2 | 8584 | 365851 | 269799515 | 963 | 3525 | 4363 | 2811 | 3998 | 47 | 219 | 1 | 0 | 0 | 0 |
| 2 | 17915 | 507467 | 286803947 | 1425 | 5302 | 7927 | 5553 | 6812 | 38 | 165 | 1 | 1 | 1 | 0 |
| 2 | 17567 | 625479 | 380394784 | 1416 | 5590 | 7534 | 5398 | 6899 | 39 | 159 | 1 | 1 | 1 | 0 |
| 2 | 18619 | 644045 | 501219605 | 1734 | 6329 | 8001 | 5848 | 7634 | 41 | 189 | 1 | 1 | 1 | 0 |
| 2 | 20209 | 651262 | 406589647 | 1827 | 6545 | 8892 | 6545 | 8465 | 42 | 82 | 1 | 1 | 1 | 0 |
| 2 | 19778 | 1283797 | 799847061 | 1623 | 5842 | 8753 | 6335 | 7275 | 37 | 198 | 1 | 1 | 1 | 0 |
| 2 | 16841 | 584019 | 362862599 | 1515 | 5773 | 7333 | 5093 | 7263 | 43 | 145 | 1 | 0 | 0 | 0 |
| 2 | 17044 | 520887 | 390218228 | 1496 | 5745 | 7167 | 5232 | 7367 | 43 | 112 | 1 | 0 | 0 | 0 |
| 2 | 18587 | 646203 | 342534678 | 1448 | 6101 | 7886 | 5473 | 7675 | 41 | 108 | 1 | 1 | 1 | 0 |
| 2 | 18461 | 404032 | 272220206 | 1439 | 5822 | 7825 | 5401 | 7284 | 39 | 253 | 1 | 1 | 1 | 0 |
| 2 | 17795 | 1822400 | 748844609 | 1468 | 5980 | 8280 | 4944 | 7196 | 40 | 1128 | 1 | 1 | 1 | 0 |
| 3 | 10963 | 1538206 | 1432941350 | 1032 | 4458 | 6216 | 3053 | 5181 | 47 | 1027 | 1 | 0 | 0 | 0 |
| 3 | 18546 | 492545 | 311375506 | 1413 | 5769 | 7953 | 5645 | 7315 | 39 | 266 | 1 | 1 | 1 | 0 |
| 3 | 19052 | 944574 | 719329701 | 1589 | 6020 | 8654 | 5953 | 7889 | 41 | 694 | 1 | 1 | 1 | 0 |
| 3 | 19816 | 550920 | 361802548 | 1530 | 6260 | 9349 | 5790 | 7967 | 40 | 1090 | 1 | 1 | 1 | 0 |
| 3 | 19430 | 539170 | 447759029 | 1463 | 6005 | 8647 | 5572 | 7416 | 38 | 1092 | 1 | 1 | 1 | 0 |
| 3 | 19714 | 652063 | 513076347 | 1555 | 5780 | 8030 | 5681 | 7387 | 37 | 1102 | 1 | 1 | 1 | 0 |
| 3 | 18850 | 421759 | 288400418 | 1466 | 5734 | 8230 | 5718 | 7260 | 39 | 562 | 1 | 1 | 1 | 0 |
| 3 | 19696 | 431599 | 282028717 | 1484 | 6004 | 8930 | 5797 | 7649 | 39 | 679 | 1 | 1 | 1 | 0 |

| TAT_ID | flows | pckts | bytes | srcIP | dstIP | srcPT | dstPT | 2pflow | 2pflow/flows | mal | label | RULE | TP | FP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 19589 | 403088 | 289407206 | 1612 | 6512 | 8694 | 5510 | 8521 | 43 | 1212 | 1 | 1 | 1 | 0 |
| 3 | 17405 | 478040 | 366502051 | 1448 | 5712 | 7704 | 4818 | 7236 | 42 | 1190 | 1 | 0 | 0 | 0 |
| 3 | 19009 | 1522670 | 1227728311 | 1464 | 5603 | 8662 | 5634 | 7117 | 37 | 929 | 1 | 1 | 1 | 0 |
| 3 | 17577 | 711579 | 448100428 | 1403 | 5519 | 7820 | 5346 | 7118 | 40 | 154 | 1 | 1 | 1 | 0 |
| 4 | 19317 | 475012 | 356276046 | 1904 | 6385 | 9845 | 6178 | 9168 | 47 | 2155 | 1 | 1 | 1 | 0 |
| 4 | 19762 | 381591 | 270607009 | 1857 | 6361 | 9635 | 6477 | 8574 | 43 | 1564 | 1 | 1 | 1 | 0 |
| 4 | 20089 | 620034 | 429908606 | 1515 | 6167 | 9445 | 6327 | 9235 | 46 | 1848 | 1 | 1 | 1 | 0 |
| 4 | 19620 | 531385 | 352863997 | 1501 | 6474 | 9031 | 5636 | 8308 | 42 | 965 | 1 | 1 | 1 | 0 |
| 4 | 19478 | 1392149 | 539408241 | 1485 | 5896 | 9813 | 6152 | 7786 | 40 | 2645 | 1 | 1 | 1 | 0 |
| 4 | 19159 | 879486 | 449489052 | 1458 | 6355 | 9076 | 5774 | 8043 | 42 | 1557 | 1 | 1 | 1 | 0 |
| 4 | 19528 | 525267 | 335401946 | 1501 | 6205 | 8829 | 6048 | 8318 | 43 | 930 | 1 | 1 | 1 | 0 |
| 4 | 21973 | 551953 | 419939038 | 1795 | 6800 | 11120 | 6966 | 8969 | 41 | 1938 | 1 | 1 | 1 | 0 |
| 4 | 18929 | 378804 | 246874300 | 1623 | 6586 | 9302 | 5968 | 8665 | 46 | 1433 | 1 | 1 | 1 | 0 |
| 4 | 18058 | 2666904 | 1776439584 | 1399 | 6059 | 8482 | 5586 | 7905 | 44 | 1925 | 1 | 0 | 0 | 0 |
| 4 | 18941 | 329687 | 215168228 | 1629 | 6519 | 9933 | 6250 | 8305 | 44 | 2405 | 1 | 1 | 1 | 0 |
| 4 | 18950 | 359759 | 224902421 | 1508 | 6441 | 9604 | 5985 | 8104 | 43 | 1149 | 1 | 1 | 1 | 0 |
| 4 | 12707 | 316107 | 179509984 | 1150 | 4402 | 6422 | 4218 | 5332 | 42 | 215 | 1 | 0 | 0 | 0 |
| 5 | 22104 | 528195 | 353103371 | 1310 | 6082 | 10582 | 7063 | 8567 | 39 | 106 | 1 | 1 | 1 | 0 |
| 5 | 20431 | 1705664 | 409907127 | 1248 | 6102 | 9219 | 6483 | 8264 | 40 | 916 | 1 | 1 | 1 | 0 |
| 5 | 22291 | 1073603 | 627362315 | 1273 | 6098 | 11735 | 7166 | 7999 | 36 | 2030 | 1 | 1 | 1 | 0 |
| 5 | 20809 | 532990 | 320944204 | 1337 | 6461 | 10272 | 6537 | 8349 | 40 | 2030 | 1 | 1 | 1 | 0 |
| 5 | 20307 | 498737 | 236897342 | 1540 | 6613 | 9665 | 6498 | 8643 | 43 | 2153 | 1 | 1 | 1 | 0 |

| TAT_ID | flows | pckts | bytes | srcIP | dstIP | srcPT | dstPT | 2pflow | 2pflow/flows | mal | label | RULE | TP | FP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 18672 | 723199 | 330145238 | 1565 | 6710 | 8946 | 5982 | 8517 | 46 | 1555 | 1 | 1 | 1 | 0 |
| 5 | 19035 | 443066 | 297580540 | 1439 | 6616 | 9059 | 5915 | 8441 | 44 | 1927 | 1 | 1 | 1 | 0 |
| 5 | 18417 | 524346 | 320531844 | 1279 | 6304 | 8782 | 5505 | 7773 | 42 | 2580 | 1 | 0 | 0 | 0 |
| 5 | 18570 | 455112 | 327703669 | 1426 | 6267 | 8868 | 5699 | 8027 | 43 | 2476 | 1 | 1 | 1 | 0 |
| 5 | 1215 | 28399 | 16061103 | 247 | 731 | 841 | 416 | 623 | 51 | 479 | 1 | 0 | 0 | 0 |
| 1 | 2498 | 37323 | 25291825 | 482 | 1481 | 1757 | 803 | 1661 | 66 | 0 | 0 | 0 | 0 | 0 |
| 1 | 15945 | 392211 | 295215205 | 1444 | 5601 | 7442 | 4754 | 7019 | 44 | 0 | 0 | 0 | 0 | 0 |
| 1 | 18257 | 545033 | 352867533 | 1858 | 6210 | 8371 | 5964 | 8382 | 46 | 0 | 0 | 0 | 0 | 0 |
| 1 | 18407 | 385685 | 258533239 | 1574 | 5848 | 8294 | 5758 | 7868 | 43 | 0 | 0 | 0 | 0 | 0 |
| 1 | 19341 | 517467 | 419437455 | 1553 | 6157 | 8575 | 5833 | 8252 | 43 | 0 | 0 | 1 | 0 | 1 |
| 1 | 18367 | 1514438 | 1321584451 | 1652 | 5967 | 8175 | 5722 | 7816 | 43 | 0 | 0 | 0 | 0 | 0 |
| 1 | 16073 | 624246 | 380146654 | 1706 | 5773 | 7453 | 5141 | 7436 | 46 | 0 | 0 | 0 | 0 | 0 |
| 1 | 16394 | 770140 | 646815980 | 1488 | 5514 | 7597 | 5244 | 7061 | 43 | 0 | 0 | 0 | 0 | 0 |
| 1 | 23142 | 1447094 | 796137019 | 1536 | 5686 | 9366 | 7174 | 7329 | 32 | 0 | 0 | 1 | 0 | 1 |
| 1 | 24286 | 1233385 | 328449049 | 1544 | 6072 | 9705 | 7484 | 7977 | 33 | 0 | 0 | 1 | 0 | 1 |
| 1 | 22297 | 1380669 | 877136387 | 1482 | 5455 | 9221 | 7101 | 7130 | 32 | 0 | 0 | 1 | 0 | 1 |
| 1 | 818 | 34720 | 10888481 | 239 | 464 | 535 | 353 | 369 | 45 | 0 | 0 | 0 | 0 | 0 |
| 2 | 16530 | 504356 | 351770353 | 1331 | 5374 | 7157 | 4973 | 6885 | 42 | 19 | 0 | 0 | 0 | 0 |
| 2 | 16453 | 1342483 | 579417990 | 1400 | 5425 | 7083 | 4875 | 6978 | 42 | 2 | 0 | 0 | 0 | 0 |
| 2 | 17423 | 1813337 | 473438841 | 1492 | 5603 | 7382 | 5213 | 7098 | 41 | 0 | 0 | 0 | 0 | 0 |
| 2 | 8194 | 1220715 | 1183308214 | 894 | 1762 | 3957 | 3376 | 1668 | 20 | 4 | 0 | 0 | 0 | 0 |

| TAT_ID | flows | pckts | bytes | srcIP | dstIP | srcPT | dstPT | 2pflow | 2pflow/flows | mal | label | RULE | TP | FP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 18358 | 396417 | 280491027 | 1420 | 5824 | 7509 | 5471 | 7527 | 41 | 0 | 0 | 0 | 0 | 0 |
| 3 | 11866 | 283959 | 159097466 | 1207 | 3999 | 5244 | 3829 | 4773 | 40 | 0 | 0 | 0 | 0 | 0 |
| 3 | 129 | 1707 | 558736 | 61 | 72 | 67 | 80 | 24 | 19 | 0 | 0 | 0 | 0 | 0 |
| 4 | 3428 | 81102 | 59611811 | 605 | 1827 | 2142 | 1121 | 2019 | 59 | 0 | 0 | 0 | 0 | 0 |
| 4 | 61 | 1563 | 1081848 | 31 | 34 | 37 | 31 | 7 | 11 | 0 | 0 | 0 | 0 | 0 |
| 5 | 13158 | 264790 | 126838072 | 1043 | 4754 | 7935 | 4087 | 5985 | 45 | 3 | 0 | 0 | 0 | 0 |
| 5 | 17297 | 346091 | 194685648 | 1336 | 6341 | 8962 | 5126 | 8086 | 47 | 3 | 0 | 0 | 0 | 0 |
| 5 | 18846 | 1511556 | 1251900453 | 1324 | 6265 | 9942 | 5668 | 8549 | 45 | 1 | 0 | 0 | 0 | 0 |
| 5 | 18848 | 2454892 | 786913931 | 1371 | 6370 | 9535 | 5699 | 8122 | 43 | 4 | 0 | 0 | 0 | 0 |
| 6 | 11088 | 203179 | 118178983 | 933 | 4179 | 5790 | 3541 | 4807 | 43 | 3 | 0 | 0 | 0 | 0 |
| 6 | 19106 | 404486 | 241163465 | 1276 | 6374 | 8392 | 5591 | 8039 | 42 | 3 | 0 | 0 | 0 | 0 |
| 6 | 18893 | 588556 | 438847925 | 1211 | 6266 | 8775 | 5599 | 7746 | 41 | 7 | 0 | 1 | 0 | 1 |
| 6 | 19382 | 471304 | 321743722 | 1351 | 6499 | 8769 | 5798 | 8090 | 42 | 0 | 0 | 0 | 0 | 0 |
| 6 | 17836 | 578234 | 271201400 | 1218 | 6412 | 7759 | 5164 | 7983 | 45 | 4 | 0 | 0 | 0 | 0 |
| 6 | 18619 | 518570 | 347672056 | 1300 | 6523 | 8590 | 5382 | 8083 | 43 | 1 | 0 | 0 | 0 | 0 |
| 6 | 19626 | 893806 | 632052060 | 1620 | 6907 | 9452 | 6336 | 8921 | 45 | 4 | 0 | 1 | 0 | 1 |
| 6 | 18057 | 391309 | 286653967 | 1374 | 6171 | 8383 | 5734 | 7774 | 43 | 0 | 0 | 0 | 0 | 0 |
| 6 | 17914 | 473343 | 290946817 | 1265 | 6063 | 7898 | 5493 | 7545 | 42 | 3 | 0 | 0 | 0 | 0 |
| 6 | 16714 | 787490 | 633110569 | 1203 | 6013 | 7861 | 5051 | 7333 | 44 | 2 | 0 | 0 | 0 | 0 |
| 6 | 16829 | 357956 | 228596400 | 1207 | 6066 | 8104 | 4964 | 7473 | 44 | 0 | 0 | 0 | 0 | 0 |
| 6 | 17851 | 466487 | 262707460 | 1598 | 6758 | 8477 | 5557 | 8327 | 47 | 3 | 0 | 0 | 0 | 0 |
| 6 | 15970 | 613983 | 443316073 | 1299 | 6253 | 7441 | 4718 | 7506 | 47 | 0 | 0 | 0 | 0 | 0 |

| TAT_ID | flows | pckts | bytes | srcIP | dstIP | srcPT | dstPT | 2pflow | 2pflow/flows | mal | label | RULE | TP | FP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 16583 | 392218 | 235493516 | 1326 | 6460 | 7754 | 5135 | 8077 | 49 | 0 | 0 | 0 | 0 | 0 |
| 6 | 5532 | 139935 | 88447672 | 699 | 1845 | 3187 | 2331 | 1807 | 33 | 1 | 0 | 0 | 0 | 0 |
| 7 | 1505 | 81948 | 20416801 | 371 | 879 | 1114 | 501 | 1083 | 72 | 2 | 0 | 0 | 0 | 0 |
| 7 | 13286 | 479297 | 350884329 | 1258 | 6224 | 6615 | 3725 | 7541 | 57 | 0 | 0 | 0 | 0 | 0 |
| 7 | 14986 | 509368 | 324742978 | 1391 | 6284 | 6722 | 4152 | 7890 | 53 | 2 | 0 | 0 | 0 | 0 |
| 7 | 16650 | 1096622 | 578337379 | 1317 | 6619 | 7345 | 4716 | 8182 | 49 | 1 | 0 | 0 | 0 | 0 |
| 7 | 15532 | 572566 | 498489831 | 1159 | 6220 | 7181 | 4331 | 7552 | 49 | 0 | 0 | 0 | 0 | 0 |
| 7 | 15541 | 331208 | 256741710 | 1245 | 6508 | 7198 | 4314 | 8008 | 52 | 4 | 0 | 0 | 0 | 0 |
| 7 | 14854 | 520562 | 402675349 | 1227 | 6782 | 6533 | 3848 | 8139 | 55 | 0 | 0 | 0 | 0 | 0 |
| 7 | 14795 | 458990 | 344741307 | 1154 | 6709 | 6608 | 3801 | 7882 | 53 | 2 | 0 | 0 | 0 | 0 |
| 7 | 16890 | 453502 | 335407114 | 1236 | 6605 | 6997 | 4667 | 8100 | 48 | 1 | 0 | 0 | 0 | 0 |
| 7 | 17429 | 407840 | 309879703 | 1227 | 6836 | 7298 | 4749 | 8528 | 49 | 0 | 0 | 0 | 0 | 0 |
| 7 | 17333 | 816504 | 691355604 | 1324 | 7140 | 7220 | 4596 | 9045 | 52 | 2 | 0 | 0 | 0 | 0 |
| 7 | 15725 | 397191 | 259950558 | 1223 | 7002 | 7301 | 4027 | 8352 | 53 | 0 | 0 | 0 | 0 | 0 |
| 7 | 15864 | 421433 | 312232234 | 1263 | 6799 | 6465 | 4054 | 8335 | 53 | 4 | 0 | 0 | 0 | 0 |
| 7 | 16337 | 404915 | 264546592 | 1181 | 6673 | 6868 | 4348 | 8181 | 50 | 0 | 0 | 0 | 0 | 0 |
| 7 | 16650 | 438777 | 266068118 | 1314 | 6988 | 7583 | 4729 | 8489 | 51 | 1 | 0 | 0 | 0 | 0 |
| 7 | 17161 | 673578 | 366135116 | 1511 | 7475 | 8004 | 4788 | 9663 | 56 | 2 | 0 | 0 | 0 | 0 |
| 7 | 9462 | 263510 | 214345408 | 979 | 3515 | 4758 | 3172 | 3903 | 41 | 0 | 0 | 0 | 0 | 0 |
| 8 | 12922 | 287026 | 220686518 | 1108 | 6459 | 5522 | 3296 | 7568 | 59 | 2 | 0 | 0 | 0 | 0 |
| 8 | 16139 | 520941 | 430371911 | 1177 | 6883 | 6837 | 3996 | 8519 | 53 | 1 | 0 | 0 | 0 | 0 |

| TAT_ID | flows | pckts | bytes | srcIP | dstIP | srcPT | dstPT | 2pflow | 2pflow/flows | mal | label | RULE | TP | FP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 17494 | 787629 | 653047918 | 1254 | 6912 | 7966 | 4680 | 8308 | 47 | 0 | 0 | 0 | 0 | 0 |
| 8 | 16306 | 413223 | 292545094 | 1227 | 7012 | 6951 | 4114 | 8442 | 52 | 0 | 0 | 0 | 0 | 0 |
| 8 | 15934 | 1051463 | 1089584825 | 1281 | 6783 | 6585 | 4209 | 8331 | 52 | 0 | 0 | 0 | 0 | 0 |
| 8 | 16473 | 642484 | 286619802 | 1205 | 6938 | 7258 | 4308 | 8312 | 50 | 0 | 0 | 0 | 0 | 0 |
| 8 | 15775 | 589599 | 300784703 | 1133 | 6621 | 7282 | 4007 | 8239 | 52 | 0 | 0 | 0 | 0 | 0 |
| 8 | 16695 | 602730 | 278897830 | 1162 | 6977 | 7187 | 4309 | 8616 | 52 | 0 | 0 | 0 | 0 | 0 |
| 8 | 17155 | 1075163 | 1074631170 | 1144 | 6883 | 7213 | 4558 | 8521 | 50 | 0 | 0 | 0 | 0 | 0 |
| 8 | 16772 | 1560959 | 643634764 | 1205 | 6932 | 7662 | 4496 | 8448 | 50 | 0 | 0 | 0 | 0 | 0 |
| 8 | 16452 | 505609 | 428453835 | 1151 | 6561 | 7378 | 4251 | 8363 | 51 | 0 | 0 | 0 | 0 | 0 |
| 8 | 16852 | 845286 | 780282119 | 1252 | 6755 | 7261 | 4555 | 8273 | 49 | 0 | 0 | 0 | 0 | 0 |
| 8 | 12871 | 673403 | 521157259 | 1046 | 5192 | 6425 | 3732 | 5771 | 45 | 0 | 0 | 0 | 0 | 0 |
| 8 | 12149 | 763155 | 701345151 | 994 | 5188 | 6490 | 3528 | 5909 | 49 | 0 | 0 | 0 | 0 | 0 |
| 8 | 16650 | 562686 | 384317786 | 1215 | 6972 | 7497 | 4301 | 8577 | 52 | 0 | 0 | 0 | 0 | 0 |
| 8 | 13271 | 601277 | 380246401 | 1073 | 4370 | 6391 | 4274 | 5189 | 39 | 0 | 0 | 0 | 0 | 0 |
| 8 | 90 | 3462 | 1034443 | 39 | 81 | 72 | 36 | 64 | 71 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | 49 | 45 | 39 | 6 |
| avg | 16260.16 | 656712.5 | 421529215 | 1293.74 | 5673.496 | 7465.561 | 4858.024 | 7135.772 | 44.36585366 | | | | 0.8 | 0.08 |
| std | 4929.49 | 461359.6 | 297193341 | 346.3115 | 1599.467 | 2212.683 | 1533.568 | 2111.073 | 8.076750983 | | | | | |