Intrusion Detection System (IDS) for Alerting and Classifying Network Attacks

La Wunn Yee[1], Thanda Win [2], Htway Htway Hlaing[3]
[1,2,3] *Department of Information Technology Engineering,*
Yangon Technological University
[1]layaungsue.mlm@gmail.com
[2]thanda80@gmail.com
[3]htwayhlyne@gmail.com

Abstract- Intrusion Detection System (IDS) is a useful defense technique against network attacks as well host attacks because they can help network/host administrator to detect any security violations by showing alerts. Although IDSs can produce thousands of alerts per day for network security, most of them are false positives. The abundance of false positive alerts can be weak for administrator to find successful attacks and give action on them. The system is implemented to be accurate in IDS by classifying network attacks with the created dataset. And also, the attack classification is tested by using both off-line and on-line alerts from the IDS. Then, the calculation of false alarm rate, the accuracy of the system and the Percentage of Successful Prediction (PSP) are presented to be a good IDS system by reducing the workload of human analyst while classifying network attacks.

Keywords- Intrusion Detection System (IDS), attack classification, false alarm rate, accuracy, Percentage of Successful Prediction (PSP)

I. INTRODUCTION

Today, security is a big concern for all network environments. Intrusion Detection System is also a popular method to secure the network infrastructure and communication over the Internet. Intrusion detection methods started appearing in the last few years. Intrusion Detection System (IDS) is a system to detect intruder activity and can collect and analyze audit data or network logs from a variety of computer systems and network sources for signs of intrusions, i.e., actions that attempt to compromise the confidentiality, integrity and availability of a computer resource. There are two basic types of Intrusion Detection System (IDS): Host based IDS (HIDS) and Network based IDS (NIDS). Each has a distinct approach for monitoring, securing data and systems. Host-based IDS examine data on individual computers that serve as hosts, while Network-based IDS capture data packets traveling on the network [1].

And then, IDS has two basic approaches. They are signature or pattern-based IDS and anomaly-based IDS. Signature or pattern-based IDS perform pattern matching techniques to detect known intrusion based on the attacks that are stored in the database. Therefore it can reduce false alarm rates. The disadvantage is that it cannot detect unknown attacks. The second approach, anomaly-based IDS creates a profile from normal behaviors and automatically detect anomalous behaviors. It has the ability to detect novel attacks or unknown attacks. So it increases in false alarm rates and is undetected well-known attacks [2]. The system is implemented with signature based IDS approach on both two types of IDS. As the signature based IDS became more popular, its limitations and problems (false alarms and irrelevant alarms) have become apparent. A perfect IDS does not generate false or irrelevant alarms. In practice, signature based IDS

found to produce more false alarms than expected. This is because of the overly general signatures and lack of built in verification tool to validate the success of the attack. The huge amount of false positives in the alert log makes the process of taking remedial action for the true positives, i.e. successful attacks, delayed and labor intensive [1].

Moreover, the creation of network and host attacks using attacking tools on the Network Lab Environment can be studied by using BackTrack OS. And also alerts' log can be captured on the IDS sensor by using snort. After capturing alerts, all are uncategorized and mixed with false positive alerts and unknown attacks that are not stored in the database of signature-based IDS. Attack classification can be satisfied above the problems. The system can give clearly the types of attack, the false alarm rate, the accuracy of the system and the Percentage of Successful Prediction (PSP) without time consuming for human analyst after attack classification.

The rest sections of the paper are organized by the followings: related works to the system are described in section II. Methods used in the system are presented in section III. The detailed procedures for IDS classification system are explained in section IV. The accuracy of experimental results is presented in section V and the last section is the conclusion of the system.

II. RELATED WORKS

The authors defined false alarm rates for IDS by using Data Mining onto 7000 of the KDD CUP 99 dataset. They used data mining software tools known as IDA analyzer, a capable tool of classifying large amount of data within seconds depending on the speed and condition of computer processors. They classified all the data into six classes of DoS attacks that are Back, Land, Neptune, Pod, Smurf and Teardrop and one class of normal data. They got the highest accuracy (99.99%) in the 4900 training data and 2100 testing data [3]. Moreover, an anomaly based IDS that showed in [4] gave the performance of Random Forest Classifier for classifying attacks on DARPA dataset is better than other classifiers, k-NN and Naive Bayes.

A Hybrid Multi Level Intrusion Detection System which used different Machine Learning Techniques on each level was presented in [5]. The authors also used KDD CUP 99 to get the experimental results and presented 93.2% of classification rate and 9.4% of false alarm rates in their system. IDS with k-Nearest Neighbor classifier was used in [6]. In this paper, the authors experimented with 1998 DARPA. They calculated the similarity between the new process and each training process instance to categorize a new process into either normal or intrusive class with the k-NN classifier. They researched on the various k's value from 5 to 25. They know that k=10 is a better choice than other values in that the attack detection rate reaches 100% faster. Finally, they presented that attack detection rate for DARPA testing data is 91.7% when anomaly detection is combined with signature verification.

Machine learning approaches, Principal Component Analysis (PCA) and Naïve Bayes classifier for IDS classification with KDD CUP 99 are expressed in [7]. In this paper, the author proved that the simply Naïve Bayes classifier can give more correctly classified than the combination of the Naïve Bayes classifier and PCA in testing. And also, there are less memory requirement and high execution speed than the combination.

According to the concepts pointed out from the previous works, the IDS classification system with the created dataset is presented in this paper to reduce false

alarm rate and to get the good performance in classification. And also, attacks' creation in the network environment can be studied to make the created dataset by using BackTrack. Therefore, administrators can reduce time consuming and false alarm rate by using IDS classification.

III. RESEARCH METHODOLOGY

The main intention is to demonstrate Intrusion Detection System (IDS) with alerting and classifying network attacks.

A. BackTrack

BackTrack is used to create attacks on the Network Lab Environment. BackTrack is free Operating System and named after a search algorithm called backtracking. It aids security professionals in the ability to perform assessments in a purely native environment dedicated to hacking. BackTrack is a distribution based on the Debian GNU/Linux distribution aimed at digital forensics and penetration testing use [8].

B. Security Onion

Security onion is a well-known free Open Source Linux Distribution for IDS, Network Security Monitoring (NSM) and log management. It is a live Xubutnu based distribution and contains many security tools such as Snort, Squil, Snorby, Squert, Suricata, OSSEC and so on required to perform the detection and prevention of the exploits [9].

Among many security tools in security onion, snort is used for IDS. Snort, one of the most widely used tool created by Martin Roesch in 1998, is a free and open source network intrusion detection system capable of performing real-time traffic analysis and packet logging on IP networks. Snort can sniffs and examines network data packets for content that matches known attacks and uses rules to check for errant packets in network. If a packet matches that rule, that packet creates an alert

In Security Onion, the important configuration files can be found in the locations "/etc/nsm/rules/" where involves the IDS engine rules utilized for discovering of events. All rules downloaded with pulledpork will be stored into downloaded.rules and rules created by all users are saved into local.rules. Security Onion will detect and alert the attacks using the Emerging Threats ruleset located in the downloaded.rules file [10].

In snort for IDS, rules can be divided into two options: rule header and rule option. In rule header, there are action, protocol, source address, source port, direction, destination address, and destination port. Message is only included in rule option. Snort's rules are simply defined by the following example:

- alert icmp any any -> any 21 (msg: "FTP root login"; content: "USER root")

In this example, action is alert type, protocol is icmp, source address and port are also any and direction is source to destination. From the other side, destination address is allowed any but destination port must be 21 for FTP only. Message is "FTP root login" and content is "USER root" [3].

IV. IDS CLASSIFICATION SYSTEM

Overview of the system is shown in Fig.1.The detailed procedures for the system are presented by the followings:

(1) Initiate and launch the network and host attacks

(2) Define and modify IDS rules

(3) Collect the attacks by using existing rules and modifying rules from IDS sensor
(4) Make the created dataset by collecting alert' logs
(5) Classify types of attacks by using the created dataset

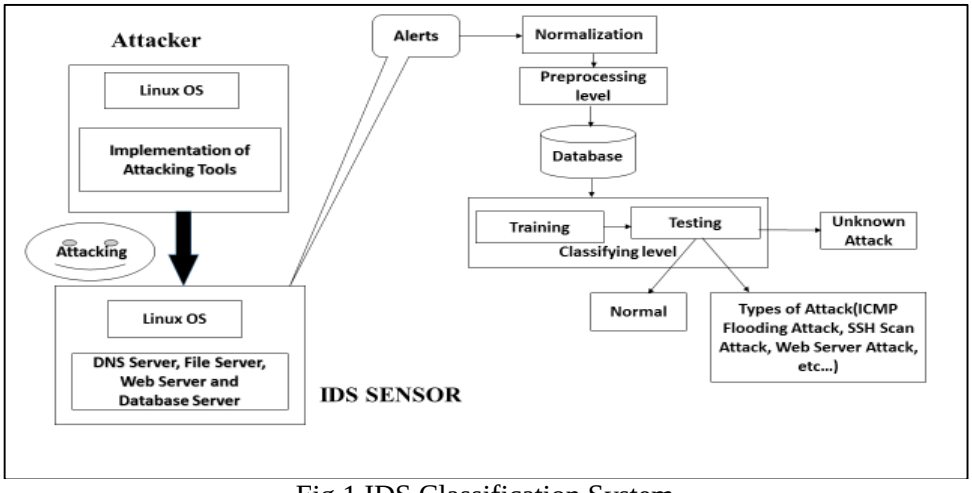

Fig.1 IDS Classification System

Firstly, the network and host attacks are initiated and launched on the Network Lab Environment with BackTrack. Fig.2 shows how to create the example of one of the network attacks, web server attack on the BackTrack.

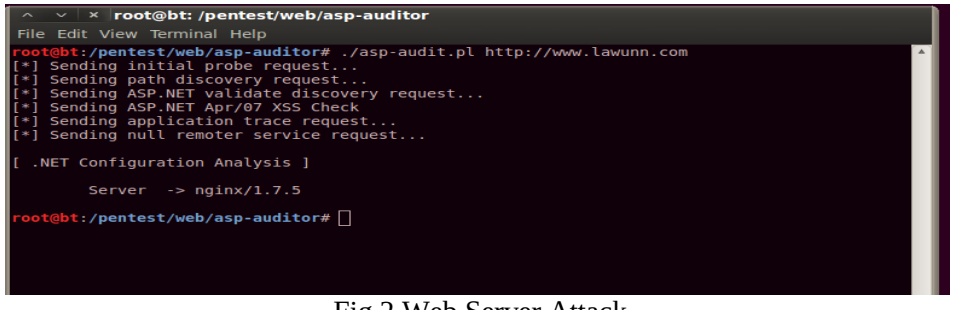

Fig.2 Web Server Attack

And then, IDS rules are defined and modified to capture attacks that come from the attacker to IDS sensor. After defining and modifying rules, attacks' alerts are collected from the IDS sensor by using Snorby, the front end user. This is shown in Fig.3. In this figure, alerts are uncategorized and mixed with false alarms.

After collecting attacks' alerts, the alerts are saved in Comma-Separated Value (CSV) file format that convert native alerts from IDS alert sensors to custom format. And alerts are stored in a central database and performed analysis on the data with custom algorithm. After making the created dataset, alerts are ready to classify with their types.

To classify attacks, there are two divisions in the program: Training and Testing. Training is needed to train attacks' types by using classifying algorithm according to the created dataset.

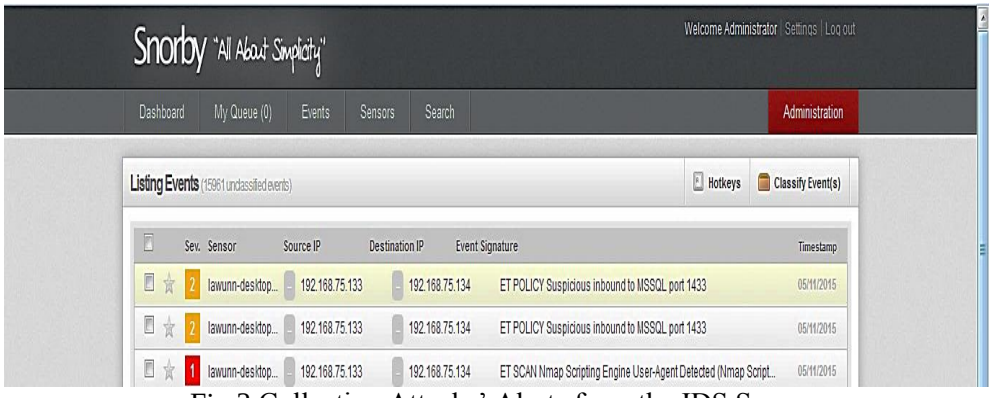

Fig.3 Collecting Attacks' Alerts from the IDS Sensor

Examples of conditions to classify attacks in the classifying algorithm by using the created dataset are implemented by the followings:

- **if** (srcport (A_i) == "21" and tcpwindow (A_i) == "0x5B" and iplen (A_i) == 75776)
  {
  C← "FTP Brute Force Attack"
  }
- **else if** (dstport (A_i) == "21" and tcpwindow (A_i) == "0x391" and iplen (A_i) == 75776)
  {
  C ← "FTP Buffer Overflow Attack"
  }

After training with conditions, other datasets are tested with classifying algorithm to classify the types of attacks. But in the testing case, unknown attacks' count and normal count are included with the types of attacks. Because the training dataset and the testing dataset are different to classify the types of attacks. If there are a lot of unknown attacks after classification of attacks, new conditions for the classifying algorithm are needed to update and reconsider. Training and testing have the same procedures for the program. First of all, training is worked before testing. In Fig.4, it presents the procedures for the program of classifying attacks with the block diagram.

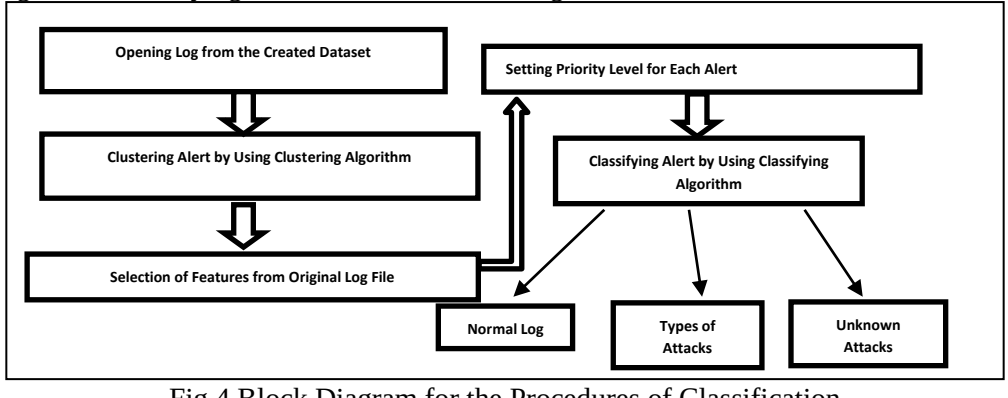

Fig.4 Block Diagram for the Procedures of Classification

Firstly, original log file that has 27 features is opened from the created dataset. After opening the created dataset, alerts are clustered by time based clustering algorithm. And then features from the original log file are selected to classify attacks because of reducing waste of time. Among all features from the original log file, 17 features are selected. Then, priority levels are set according to the alerts' message. After setting priority levels, attacks are classified by classifying algorithm. Fig.5 shows the attack specification of testing on the 5390 data counts with pie chart after classifying attacks. According to the results, the number of the DNS server attacks is the highest and the number of telnet bad login attacks, FTP login successful attacks and SNMP attacks are the lowest. Moreover, the number of unknown attacks is 34 and normal is 35 of all after classification.

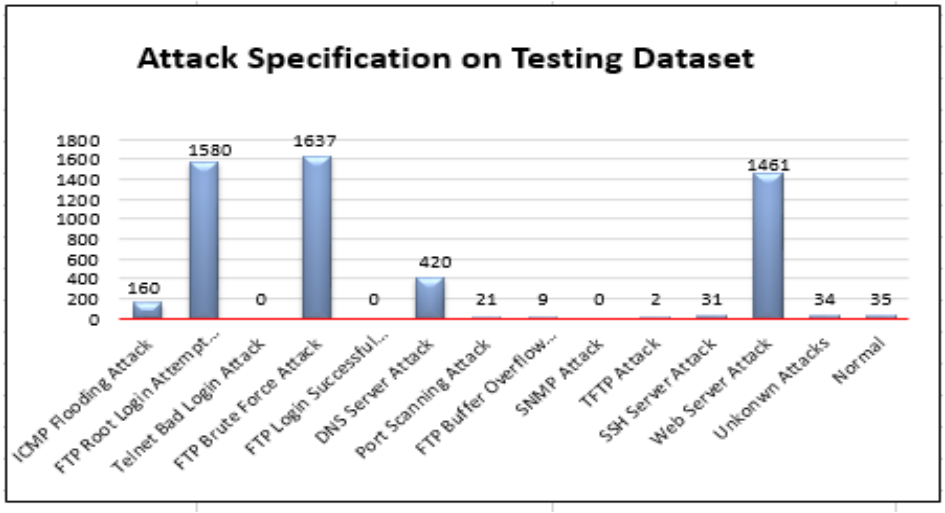

Fig.5 Attack Specification of Testing Dataset

V. EXPERIMENTAL RESULTS

The system is tested on the Virtual Machine that is installed Attacking OS (BackTrack) and IDS Sensor (Security Onion). To test the performance, the system is implemented with C#.Net. The results of training and testing for alerting and classifying network attacks of IDS are described in this section. Moreover, the following equations are used to get the performance of the system. They are:

Accuracy = [TP/ (TP+FP)] * 100% (1)

False Alarm Rate = [FP/TN] * 100% (2)

Percentage of successful prediction
(PSP) = [Number of Attack that have been successfully classified/TA] *100% (3)

In which equations, TA is total number of attack records, TN is total number of true negative attacks, TP is total number of true positive attacks and FP is total number of false positive attacks [5]. And the accuracy between training and testing dataset of the system are also presented in Fig.6. According to the experimental results, the accuracy of the system in testing increased more than the accuracy of training.

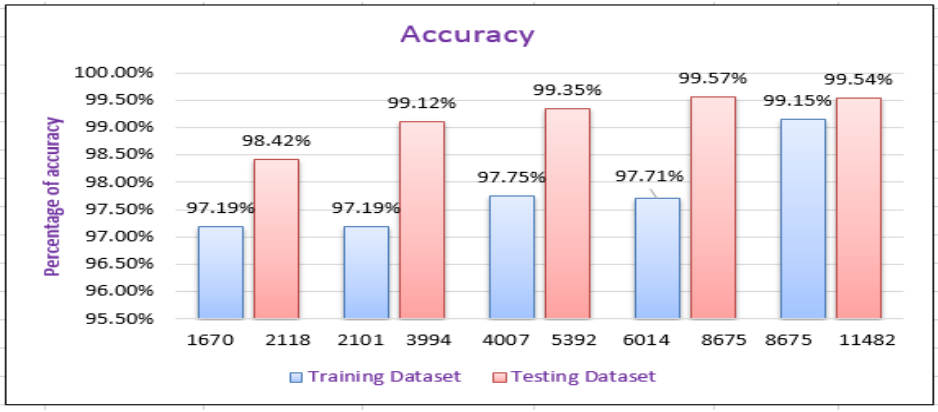

Fig.6 Accuracy between Training and Testing Datasets

Moreover, the false alarm rate moderately decreased in the calculation of the experimental results when there were changes in datasets. This is shown in Fig.7.

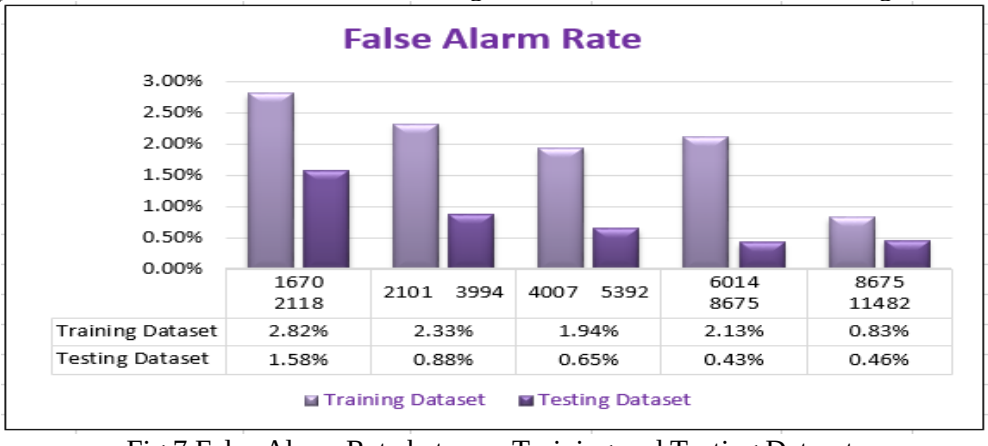

Fig.7 False Alarm Rate between Training and Testing Datasets

And also, the percentage of successful prediction of testing raised noticeably according to the experimental results. Therefore, Fig.8 shows the percentage of successful prediction of testing in different datasets.

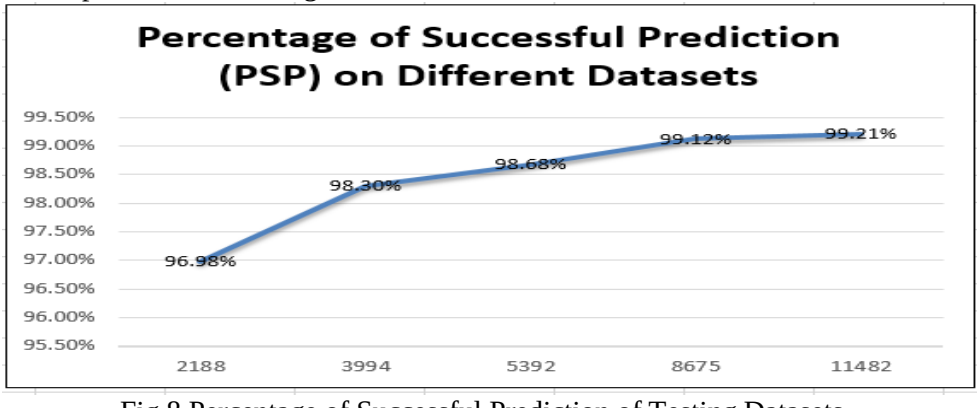

Fig.8 Percentage of Successful Prediction of Testing Datasets

VI. CONCLUSION

Intrusion Detection Technology is an effective approach to the problems of network security. The system can help and assist the security engineers and network administrators to secure their network infrastructure. It can generate the alerts which are useful for the security engineers to take down the attack origin definitely. Moreover, it can decrease the number of false positive attacks by classifying types of attacks from the experimental results. Then the accuracy of the system is also better for testing dataset than training. The more data counts of dataset in testing, the better performance in the PSP. Therefore, the system administrator can reduce audit load and time cost by calculating the accuracy of the classification of attacks and the system will be able to detect by classifying what types of attack are occurred in the network. If unknown attacks are abundant, we need to add the new conditions to the classifying algorithm. This is the disadvantage of the system because we used signature or pattern-based IDS to capture attacks for the system. This study can be extended to get better results of accuracy and increase the detection rate for the IDS system.

ACKNOWLEDGEMENT

The authors would like to express special thanks to all who support their guidance and advice for this paper.

REFERENCES

[1] Rafeeq Ur Rehman. 2003. *Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort*. New Jersey. Apache, MySQL, PHP, and ACID, Pearson Education, Inc., Publishing as Prentice Hall.

[2] Verwoerd, T., and Hunt, R. No Date. *Intrusion Detection Techniques and Approaches*. New Zealand, Department of Computer Science University of Canterbury.

[3] Fatin Norsyafawati Mohd Sabri, Norita Md.Norwawi, and Kamaruzzaman Seman. April 2011. "Identifying False Alarm Rates for Intrusion Detection System with Data Mining". (*IJCSNS) International Journal of Computer Science and Network Security*, Vol. 11: No.4.

[4] Phyu Thi Htun and Kyaw Thet Khaing. 2013. "Important Roles of Data Mining Techniques for Anomaly Intrusion Detection System*". International Journal of Advanced Research in Computer Engineering and Technology (IJARCET),* Vol. 2: No 5.

[5] Sahar Selim, Mohamed Hashem and Taymoor M. Nazmy. 2011. "Hybrid Multi- level Intrusion Detection System*". (IJCSIS) International Journal of Computer Science and Information Security*, Vol. 9: No. 5.

[6] Yihua Liao and V. Rao Vemuri. 2002. "Use of K-Nearest Neighbor Classifier for Intrusion Detection", *Computers & Security*, Vol. 21: No.5, pp (39-448), Department of Computer Science, University of California.

[7] Neethu, B. No Date. "Classification of Intrusion Detection Dataset using Machine Learning Approaches*". International Journal of Electronics and Computer Science and Engineering*, (ISSN: 2277-1956), V1N3- (1044-1051), Department of Computer Science, Amtrita University.

[8] Pritchett, W., and Smet, D.D. 2012. *BackTrack 5 Cookbook*. Published by Packt Publishing Ltd, Livery Place35, and Livery Street Birmingham B32 PB, UK.ISBN978-1-84951-738-6.

[9] Sunil Gupta. 2012. "Logging and Monitoring to Detect Network Intrusions and Compliance Violations in the Environment". M.Eng. Thesis, SANS Institute InfoSec.

[10] Harrykar's Techies Blog. No Date. "*Snort, IDS, IPS, NSM, Hacking and……Beyound*". August 2014 <http://harrykar.blogspot.com/2009/05/snort-ids-ips-nsm-andbeyond.html>.