ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF NATURAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

INTRUSION DETECTION SYSTEM USING HYBRID DETECTION APPROACH

Meheret Zewdu Wondimu

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UINVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

April 2015

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF NATURAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

INTRUSION DETECTION SYSTEM USING HYBRID DETECTION APPROACH

Meheret Zewdu Wondimu

Advisor: Dejene Ejigu (PhD)

Signature of the Board of Examiners for Approval:

Name	**Signature**
1. Dejene Ejigu (PhD) , Advisor	_____
2. Mulugeta Libise (Assi.Prof)	_____
3. Fekade Getahun (PhD)	_____

Acknowledgements

First and foremost extraordinary thanks go for my Almighty God and His Mother Saint-Merry.

It is with immense gratitude that I acknowledge Dr.Dejene Ejigu, my mentor and thesis supervisor for providing me with many valuable ideas and insights. He has constantly supported me through the thesis work. The successful completion of my thesis work is all due to his guidance and motivation. It was an enormous pleasure for me to work under his supervision.

I am highly grateful to my parents; they taught me the right, encouraged me and gave me hope and unconditional love. I wish to both of them happiness and well health. Especially My mom you mean a lot to me, without you nothing will be completed successfully in my life.

My friends, especially Bez and Selam words are few to express my deepest thanks to you. You were always supporting me and encouraging me with your best wishes.

The last but not the least thanks goes to my husband, you are the motivating force behind me at all times through both ups and downs of my educational life. Thank you for everything you give me.

Abstract

Due to the rapid growth of computer system and Internet, network security became crucial issue for most organizations. Mostly organizations increase usage of different tools and methods to secure their network due to the increase of security threats. Many methods have been developed to secure computer networks and communication over the Internet.

However, none of the existing methods developed by different researches have an accuracy of detecting attacks with high detection rate and low false alarm rate. The other thing is most deal with single detection approach with high number of features which is challenging and time consuming to implement. Also it will examine only either previously known attacks or unknown attacks.

This thesis work is devoted to solve those problems using intrusion detection system architecture that is based on neural network, signatures and dimension reduction that can promptly detect and classify attacks, whether they are known or never seen before.

The proposed hybrid intrusion detection system combines signature based and anomaly based techniques. Signature based open source which uses pattern search for attack detection and the anomaly based system is developed using machine learning technique. We implemented dimension reduction using dataset NSL-KDD and train the system using the well known artificial neural network algorithm in the area of intrusion detection.

The evaluation of performance and implementation of the proposed hybrid intrusion detection system are made with Java programming language using NetBeans. The results obtained by the implementation and evaluation are measured in comparison with other works done using single detection approach. The result shows that the output is encouraging and further refinement of the work can produce more robust and reliable intrusion detection system.

Keywords: Hybrid, Intrusion Detection, Anomaly Detection, SNORT, Artificial Neural Network, Principal Component Analysis

Table of Contents

List of Figures

List of Tables

List of Pseudo codes

Acronyms

AD	Anomaly Detection
ALAD	Application Layer Anomaly Detector
ANN	Artificial Neural Network
AV	Anti Virus
BASE	Basic Analysis and Security Engine
CERT	Computer Emergency Response Team
DoS	Denial Of Service
DR	Detection Rate
FER	Frequent Episode Rule
FN	False Negative
FP	False Positive
HIDS	Host Based Intrusion Detection System
ICMP	Internet Control Message Protocol
ID	Intrusion Detection
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
KDD	Knowledge Discovery Data
LERAD	Learning Rule for Anomaly Detector
MLP	Multilayer Perceptron
NIDS	Network Based Intrusion Detection System
NSL-KDD	Network Security Learning Knowledge Discovery Dataset
PCA	Principal Component Analysis

PHAD	Packet Header Anomaly Detector
Prob.	Probing
R2U	Remote to User
RADIUS	Remote Authentication Dial In User Service
SIEM	Security Information and Event Management
SNMP	Simple Network Management System
SNORT	Signature Based Open Source
SPADE	Statistical Packet Anomaly Detection Engine
TACACS	Terminal Access Controller Access System
TCP	Protocol Transfer Control
TN	True Negative
TP	True Positive
U2R	User to Root
UDP	User Datagram Protocol
VPN	Virtual Private Network
WEKA	Wakiato Environment for Knowledge Analysis
WINPCAP	Windows Packet Capturing Library

Chapter 1: Introduction

1.1 Background

In the early time of information systems management, network security was not thought of as necessity because only little threats existed. At that time the only computers were mainframe computers, so users only had remote terminal access to run programs. But nowadays, with the advent of the Internet, personal computers and computer networks vulnerability increases to various kinds of attacks. Internet has completely changed the way things were done in the past; the sensitive information sent over it has become vulnerable to various types of threats. Because of this reason information has become like an asset that needs to be protected from attacks.

Intruders make use of the security breaches present in the system or network to attack it. lntrusion is a purposefully illicit endeavor to access information, manipulate information or render a system untrustworthy or inoperative [1]. Due to security breaches, individual users and organizations get affected. Because of attack, privacy can be violated and important data can be lost. The attacks are usually caused by a failure to implement security policies and failure of using of security tools that are readily available. There are many real world examples such as: A case to point was the Citibank security breach in which by the time the heist was reported in 1994, 10 million dollars was already lost. Only 400,000 dollars was eventually recovered [2].

According to Joseph and Rod [3] there were a huge number of unauthorized security events, and in the year 2000, 70 percent of organizations at least reported a security incident. This represented 42 percent increase from the year 1996 report. Joseph and Rod [3] continue to write that the Computer Emergency Response Team (CERT) reported 3734 incidents in 1998, 9859 in 1999 and within only the first six months of 2000, 8836 incidences where already reported.

Organizations are striving to maintain confidentiality, integrity and availability of their networked resources and detection of attacks in the network traffic is one of the major goals of security. A number of techniques have been employed to guard against network intrusion. Even if these measures provide a level of security, they have been found to be lacking in a number of ways. For example:-

1. The use of firewall: A firewall is a hardware or software solution used to enforce security policy on a private network. It is mostly used to control traffic to or from a private network. However, these are just a list of permit and deny rules; therefore they may not always have the ability to detect intrusions. Firewall, user authentication, data encryption and Virtual Private Networks (VPN) provide a level of security but they are limited by the fact that they cannot give protection against malicious codes, inside attacks or unsecured modems [4]. Therefore would only be effective as one of the available lines of defense.
2. Cryptography: hides information from unauthorized users, however this method makes it hard to know whether any attack has taken place. Generally key management is not an easy task. Crypto systems may require special key management systems such as the use of a Terminal Access Controller Access System (TACACS) or Remote Authentication Dial. In User Service (RADIUS) server, this could mean specialized hardware or configuration. Otherwise hackers could gain access to these keys and break into the system.
3. The provision of physical security to the network site or to servers: However these are limited by the fact that physical security may not provide a practical solution to attackers who employ telnet sessions to gain access to a network.
4. Authentication: A technique used to verify users of a network resource. The effectiveness of this is weakened by the fact that many still ”use easy to crack passwords” while some users are either untrustworthy or are just careless with their passwords such that many times can easily be got by unauthorized users.
5. Many organizations have also employed anti viruses, however these may not provide information as to whether there has been an intrusion or not. Anti-viruses also require frequent updates.

Unfortunately, we don’t live in an ideal world where there would be no fence or gate locks or guards and where all the humans live peacefully with each other, so that is why we need to protect ourselves and privacy by using guards, alarms and other methods. The same concept applies in the computer world, We need devices and technologies to secure our information and assets and one of those famous technologies is the IDS.

1.2 Motivation

The motivations of this research work are application of intrusion detection system for different organization, availability of open sources and weakness of currently available network security tools with regard to detecting intrusion.

Despite the fact that intrusion detection system applicable in different organization and used more than decade, there still exists many issues around IDS. The shortcomings of the current IDS which handicap its effectiveness is most uses signature based approach which examine only known attacks and needs continuous updating of the database to get all attacks in a network. So it motivates us to combine signature based open source with anomaly based system in order to detect both known and unknown attacks.

1.3 Statement of the Problem

At present, with the rapid growth of the Internet and the ever-increasing security problems associated with its popularity, the need for protection against unwanted intruders has become very important [5]. So intrusion detection system widely applied in different organization.

Currently, most of the intrusion detection tools available in the market as well as freely available in Internet are based on the signature based approach. The main problem of this approach is it only concentrates on signature database and it needs prior knowledge; it does not have any idea about new attacks and use rule based analysis. The rule based analysis depends on sets of predefined rules that are created already. Frequent update of the template is a must to avoid becoming outdated. Due to, this it is an inflexible system that is unable to detect an attack if the sequence of events is slightly different from the predefined profile. If the intruder is intelligent, then the rule based system may fail at one point of time. Zero day attacks are also common threats to such system.

Generally, numbers of previous works have been done on intrusion detection system in which less attention has been given for detection rate, false alarm rate, examining both known and unknown attacks and complexity.

This research work is aimed to fill the gap of the aforementioned problems. In this work to do that it will combine signature based open source and anomaly based system. To have better

detection rate and minimize false alarm rate in anomaly based system we have to have detection algorithm and in this work we also select this algorithm.

This work will be essential to ensuring information security such as data confidentiality, data integrity, and data availability in the organization's network. By emphasizing that having intrusions on organization network have a negative effect on the smooth running of their activities.

There are a number of problems associated to IDS. In this research we will address the following questions:

- How can we minimize intrusion ?
- Which detection technique is the best to use as a candidate for enhancement?
- How can we increase the detection rate of signature based systems, make them detect unknown attacks and minimize false alarm rate?

1.4 Objectives

General Objective

The general objective of this thesis is to design and develop hybrid intrusion detection system for effective defense against network attacks.

Specific Objectives

The specific objectives of the research are to:

- Conduct a detail literature review to understand the application domain for machine learning in intrusion detection and its performance issue.
- Identify which open source is useful and flexible in order to do some modification and to improve detection performance.
- Select relevant features to visualize patterns in data and for its processing.
- Select detection algorithm that can examine unknown attacks.
- Design anomaly based intrusion detection using the selected detection algorithm and integrate it with the selected signature based open source.
- Test and evaluate the proposed intrusion detection system.

1.5 Scope and Limitations of the Study

In this thesis work, we design hybrid intrusion detection system. It focuses on identifying possible incidents, logging information about them, and reporting them to the security administrators. This system is designed to increase detection rate and examine both previously known and unknown attacks, and applicable in any organization's network.

However, this work doesnot focus on intrusion prevention system that can take an action for the coming intrusion but it only gives alert to the administrator. Moreover, the anomaly based system uses only single detection algorithm even if combining a number of detection algorithms may have an impact on increasing detection rate.

1.6 Methodology

Different methodologies will be employed in this research in order to accomplish the general and specific objectives of this study. The first thing to do is to conduct a comprehensive review of literatures to acquire a deeper understanding of the research area and its problem domains. Through this literature we identify the importance of the previous works done in the area of intrusion detection. Existing works related to this research work assessed to identify and point direction in providing solution to identified problems. The second thing to do is to look for data and different datasets to study patterns that can identify attacks from normal.

Based on the proposed solution from the identified problem on the literature survey we have to select appropriate tools, techniques and algorithms used. After identifying those requirements we will design the architecture of a new hybrid intrusion detection system and identify its components to implement. Then finally we will evaluate the system using some metrics like precision, recall, accuracy, and so on.

The overall methodology is based on producing artifacts including architecture, algorithm and libratory experiments.

1.7 Application of Results

Network security is applying different techniques and technologies to keep organization's network from attacks. Intrusion detection system is the main technique to overcome the problem

of network security and minimize the impact of intruders on the proper flow of organization's work. It is smarter security technique which will proactively and intelligently keeps an eagle's eye on the network, monitor and report incidents quickly. In this thesis, we focus on hybrid intrusion detection system that can examine both known and unknown attacks. This helps to reduce problems associated to network attacks caused by intruders in different organizations. Moreover, it would contribute to grant the confidentiality, integrity and availability of networked resources and it simplify network management.

1.8 Thesis Organization

The rest of the thesis is organized as follows:

Chapter Two: focuses on the related work in the field of intrusion detection using different detection techniques. It also discusses how intrusion detection systems are classified, what are the key functions of an IDS, what are key the components of this system , deployment scenario of an IDS in different architecture and finally it discusses about well known signature based open source IDS.

Chapter Three: here it discusses about related works that have significant relation with this thesis. Even if there are a number of works doe on this area, this Chapter selects the most related works to our thesis and presents them based on the attack detection approach.

Chapter Four: concentrates on providing a design for the proposed work. Issues raised here are signature detection module, anomaly detection module and signature generation module. It rises and answers the question "What are the requirements of hybrid intrusion detection system?".

Chapter Five: the proposed system is implemented in this Chapter by applying the proposed algorithms. Evaluation of the work using experimental analysis and comparing to others works is presented here.

Chapter Six: summarizes the contributions made in the thesis, and concludes based on the results obtained from the thesis work. Furthermore, new issues that have been surfacing while working on the thesis are suggested as future work.

Chapter 2: Literature Review

This Chapter is the output of literature survey of intrusion detection systems. Mainly there are two approaches to Intrusion Detection that is Misuse detection and Anomaly detection. Both the approaches have advantages and disadvantages. So recently a new approach has come up which combines both the approaches and gives better results. This Chapter discusses about overview, use, key function, components, classification, deployment scenario of IDS technology and signature based open source IDS.

2.1 Overview

Intrusion detection systems (IDS) are network security appliances that monitor network and/or system activities for malicious activity. It can be any device/software which exercises access control to protect computers from exploitation. "Intrusion prevention" technology is considered by some to be an extension of intrusion detection (ID) technology, but it is actually another form of access control, like an application layer firewall [6].

2.2 Intrusion Detection Systems (IDSs) Technologies

As discussed in [7] IDSs focus on identifying possible incidents. For example, an IDS could detect when an attacker has successfully compromised a system by exploiting vulnerability in the system. The IDS could then report the incident to security administrators, who could quickly initiate incident response actions to minimize the damage caused by the incident. The IDS could also log information that could be used by the incident handlers. As discussed in [8] many IDSs can also be configured to recognize violations of security policies. For example, some IDSs can be configured with firewall rule like settings, allowing them to identify network traffic that violates the organization’s security or acceptable use policies. Also, some IDSs can monitor file transfers and identify ones that might be suspicious, such as copying a large database onto a user’s laptop.

Many IDSs can also identify reconnaissance activity, which may indicate that an attack is imminent [9]. For example, some attack tools and forms of malware, particularly worms, perform reconnaissance activities such as host and port scans to identify targets for subsequent attacks. An IDS might be able to block reconnaissance and notify security administrators, who

can take actions if needed to alter other security controls to prevent related incidents. Because reconnaissance activity is so frequent on the Internet, reconnaissance detection is often performed primarily on protected internal networks. In addition to identifying incidents and supporting incident response efforts, organizations have found other uses for IDSs, including the following [8]:

- **Identifying security policy problems.** An IDS can provide some degree of quality control for security policy implementation, such as duplicating firewall rule sets and alerting when it sees network traffic that should have been blocked by the firewall but was not because of a firewall configuration error.
- **Documenting the existing threat to an organization**. IDSs log information about the threats that they detect. Understanding the frequency and characteristics of attacks against an organization's computing resources is helpful in identifying the appropriate security measures for protecting the resources. The information can also be used to educate management about the threats that the organization faces.
- **Deterring individuals from violating security policies**. If individuals are aware that their actions are being monitored by IDS technologies for security policy violations, they may be less likely to commit such violations because of the risk of detection.

Because of the increasing dependence on information systems and the prevalence and potential impact of intrusions against those systems, IDSs have become a necessary addition to the security infrastructure of nearly every organization.

2.3 Key Functions of IDS Technologies

There are many types of IDS technologies, which are differentiated primarily by the types of events that they can recognize and the methodologies that they use to identify incidents. In addition to monitoring and analyzing events to identify undesirable activity, all types of IDS technologies typically perform the following functions [10]:

- Recording information related to observed events. Information is usually recorded locally, and might also be sent to separate systems such as centralized logging servers,

security information and event management (SIEM) solutions, and enterprise management systems.

- Notifying security administrators of important observed events. This notification, known as an alert, occurs through any of several methods, including the following: e-mails, pages, messages on the IDS user interface, Simple Network Management Protocol (SNMP) traps, and user-defined programs and scripts. A notification message typically includes only basic information regarding an event; administrators need to access the IDS for additional information.
- Producing reports. Reports summarize the monitored events or provide details on particular events of interest. Some IDSs are also able to change their security profile when a new threat is detected. For example, an IDS might be able to collect more detailed information for a particular session after a malicious activity is detected within that session. An IDS might also alter the settings for when certain alerts are triggered or what priority should be assigned to subsequent alerts after a particular threat is detected.

IPS technologies are differentiated from IDS technologies by one characteristic: IPS technologies can respond to a detected threat by attempting to prevent it from succeeding. The IPS stops the attack itself. Examples of how this could be done are as follows [11]:

- Terminate the network connection or user session that is being used for the attack.
- Block access to the target (or possibly other likely targets) from the offending user account, IP address, or other attacker attribute.
- Block all access to the targeted host, service, application, or other resource.
- The IPS changes the security environment. The IPS could change the configuration of other security controls to disrupt an attack. Common examples are reconfiguring a network device (e.g., firewall, router, switch) to block access from the attacker or to the target, and altering a host-based firewall on a target to block incoming attacks. Some IPSs can even cause patches to be applied to a host if the IPS detects that the host has vulnerabilities.
- The IPS changes the attack's content. Some IPS technologies can remove or replace malicious portions of an attack to make it benign. A simple example is an IPS removing an infected file attachment from an e-mail and then permitting the cleaned email to reach

its recipient. A more complex example is an IPS that acts as a proxy and normalizes incoming requests, which means that the proxy repackages the payloads of the requests, discarding header information. This might cause certain attacks to be discarded as part of the normalization process.

As discussed in [8] another common attribute of IDS technologies is that they cannot provide completely accurate detection. When an IDS incorrectly identifies normal activity as being malicious, a false positive has occurred. When an IDS fails to identify malicious activity, a false negative has occurred. It is not possible to eliminate all false positives and negatives; in most cases, reducing the occurrences of one increases the occurrences of the other. Many organizations choose to decrease false negatives at the cost of increasing false positives, which means that more malicious events are detected but more analysis resources are needed to differentiate false positives from true malicious events. Altering the configuration of an IDS to improve its detection accuracy is known as tuning.

Most IDS technologies also offer features that compensate for the use of common evasion techniques [10]. Evasion is modifying the format or timing of malicious activity so that its appearance changes but its effect is the same. Attackers use evasion techniques to try to prevent IDS technologies from detecting their attacks. For example, an attacker could encode text characters in a particular way, knowing that the target understands the encoding and hoping that any monitoring IDSs do not. Most IDS technologies can overcome common evasion techniques by duplicating special processing performed by the targets. If the IDS can "see" the activity in the same way that the target would, then evasion techniques will generally be unsuccessful at hiding attacks.

2.4 Components of Intrusion Detection/Prevention System

Intrusion detection system consistes of different componentes in which each component will work collaborating with each other. This section disscusses what are those components and how each component work.

2.4.1 Sensors

These are installed on devices (they are devices) that locate inside the network in order to collect data. They gather input from different sources like network activities such as packets, log files and system call traces. Then the collected data is organized and then forwarded to one more analyzers. There are two types of sensors

- Network-based sensors
- Host-based sensors

The network-based sensors are more commonly used instead of host-based network packets only. All the in and out traffic can be captured and controlled by a single sensor. Network sensors do not burden the network with extra traffic in case two interfaces are used, one for monitoring and one for management. The most used programs as sensors by IDSs are TCP dump and libpcap. The host-based sensors can also capture data from network interfaces and then send that to other IDSs components; mainly they provide information about inside attacks [12].

2.4.2 Response Module/Central Engine

It controls the sensors and monitors the alerts and events. It will get data from sensors. This data is compared against IDS behavior models and generates alerts. These alerts may be forwarded to higher level monitors. Then hostile reports are generated and response is determined. An Response Module has three components [12].

- Communication interface
- Listener
- Sender

A communication interface provides communication with other components of IDS. A listener waits for information and data from sensors and other agents to receive it. A sender then transmits the data to the other agents and manger devices. Some additional functions are also performed by agents as they can perform correlation analysis on received data and generate alarms as well.

2.4.3 Manager

The manager is basically designed to provide an ability of master control for the ID. The manger component can provide the following functions [12] .

- Data management
- Alerting
- Event correlation
- High level analysis
- Monitoring of other components

2.4.4 User Interface/Event Generator

In IDS the user interface provides a view to the end user in a way to interact with the system. Only through the interface a user can control or configure the system. There is also an important characteristic of user interface that it can generate reports as well [13].

2.5 Classification of Intrusion Detection/Prevention System

There are several ways to classify IDSs depending on some criteria, such as information source, analysis type, type of response and detection time. The most common criteria are shown in Figure 2.1 and it will be explained in more detail in the following pages [14].

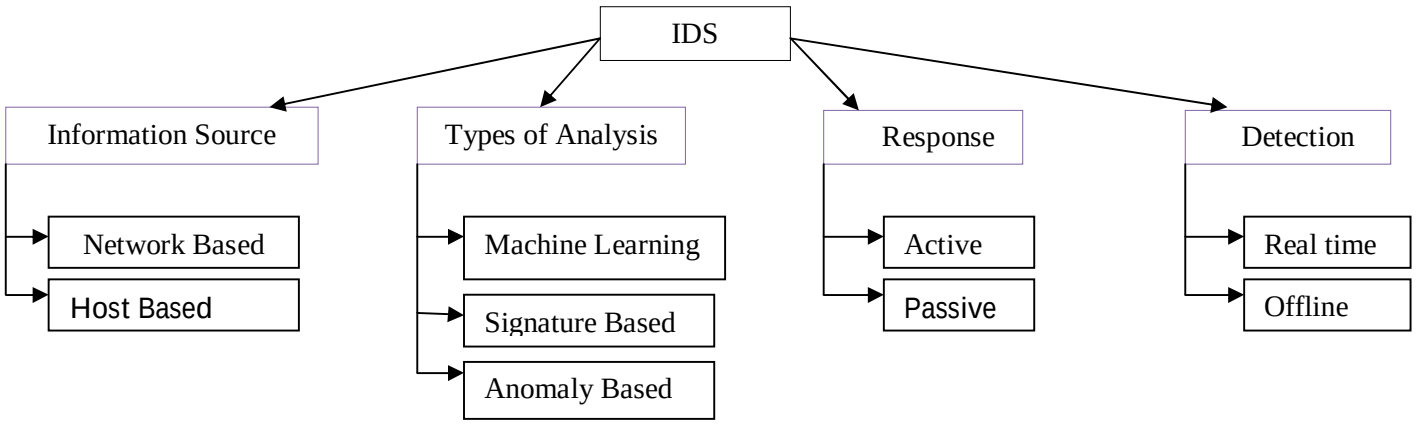

Figure 2.1: IDS Classification

2.5.1 Information Source

Information sources are one of the first issues to focus on when designing an intrusion detection system. These sources can be classified in many ways. With regard to the detection of intrusions, they are classified according to the location due to some IDSs analyzing network packages, captured from the network backbone or LAN segments while other IDSs analyze events generated by the operating systems or application software for signs of intrusion.

2.5.1.1 Network-based Intrusion Detection System (NIDS)

Most of the intrusion detection systems are network-based. These IDSs detect attacks by capturing and analyzing network packets. Listening in a segment, NIDS can monitor traffic affecting multiple hosts connected to that network segment, thus protecting these hosts. They are installed more frequently than those based on host because their configuration is general for the entire segment in which they operate.

The network-based IDSs are often formed by a set of sensors located at various points of the network. These sensors monitor traffic doing local analysis and reporting attacks carried out to the management console.

As sensors are limited to run the detection software, they can be more easily secured against attacks. Many of these sensors are designed to run in hidden mode, so it is more difficult for an attacker to determine their presence and location. There is the possibility of using a one way cable for reception (sniffing cable), so the sensor can only receive data, preventing physically any outgoing signals. One equivalent option to the one-way cable is the use of a network tap (listening network device) a device similar to a network hub that enables to listen to communications without being detected [14].

NIDS has the following advantages and disadvantages [14]:

<u>Advantages</u>

- A well located IDS can monitor a large network as long as it has enough capacity to analyze the traffic in its totality.

• NIDSs can be configured to be invisible to the network in order to increase the security against attacks.

Disadvantages

• The sensors not only analyze the headers of the packages, they also analyze their content, so they may have difficulties processing all packages in a large network or with much traffic and may fail to recognize attacks during periods of high traffic. Some vendors are trying to solve this problem by implementing IDSs completely in hardware, which makes them much faster.

• The network-based IDSs do not analyze the encrypted information. In environments where communication is encrypted it is infeasible to examine the package contents and therefore unable to assess whether this is a package with malicious contents or not. This problem is increased when the organization uses encryption in the network layer (IPSec: Internet Protocol Security) between hosts, but can be solved with a more relaxed security policy (e.g., IPSec in tunnel mode).

• The network-based IDSs do not know whether the attack was successful or not, the only thing known is that it was launched. This means that after a NIDS detects an attack, administrators must manually investigate every host attacked to determine if the attempt was successful or not.

• Some NIDSs have problems dealing with network-based attacks travelling in fragmented packages. These packages make the IDS not notice the attack or be volatile and may even get to fail.

• Due to their general configuration, NIDSs may have a high false acceptance or false positive rate. They may report a lot of normal activities identified as attacks. The problem comes when the number of such alarms is unacceptably high.

• Perhaps the biggest drawback of NIDSs is their implementation of the stack for network protocols that may differ from the stack of the systems they protect. Many servers and desktop systems do not follow in some aspects the current TCP/IP standards, thus it is possible to have them discard packages the NIDS has accepted. An example of an open source NIDS and one of the most used nowadays is SNORT on which we will focus more in-depth later.

2.5.1.2 Host-Based Intrusion Detection System (HIDS)

HIDS were the first type of IDSs developed and implemented. They run on the information acquired from inside a computer, such as audit files of the operating system. This allows the IDS to analyze actual activities with great precision, determining exactly which processes and users are involved in a particular attack within the operating system.

Like any intrusion detection system, HIDSs also report multiple false positives. Once the system is adjusted, the reduction of false positives is remarkable and then also these types of IDSs ignore very few attacks against the system.

In contrast to NIDSs, HIDSs can see the result of an attempted attack, as well as directly access and monitor data files and processes of the attacked system [15].

Although NIDSs have greater development and these days are more accepted, HIDS have certain advantages over them. The advantage and disadvantage of HIDS are [15].

Advantages

- The host-based IDSs, having the ability to monitor local events of a host, can detect attacks that cannot be seen by a network based IDS.
- They can often operate in an environment in which network traffic travels encrypted, since the source of information is analyzed before the data is encrypted on the origin host and/or after the data is decrypted on the destination host.

Disadvantages

- Host-based IDSs are more costly (in time and money) to administer as they must be managed and configured at each monitored host. While the NIDSs have IDS for multiple monitored systems, HIDSs have an IDS for each of them.
- If the analysis station is within the monitored host, the IDS can be disabled if an attack attains success on the machine
- They are not adequate for detecting attacks on an entire network (for example, port scans) since the IDS only analyses those network packets sent to it.

• They can be disabled by certain Denial of Service attacks.

• HIDSs use resources of the host that they are monitoring, influencing its performance.

As a subclass of HIDS, we should quote the multi-host-based IDS. They use the information collected from two or more hosts analyzing it and trying to catch any threat. Its approach is very similar to the classic HIDS with the additional difficulty of having to coordinate the data from several sources.

As we have mentioned SNORT as an open source project in NIDS, we should mention Osiris as an example of HIDS.

2.5.2 Type of Analysis

There are mainly two approaches to the analysis of events for detecting attacks: detection of signatures and detection of anomalies but some machine learning approaches are included here. The signature detection is the technique used by most commercial systems. The anomalies detection, in which the analysis looks for unusual patterns of activity, has been and remains under investigation. The detection of anomalies is used by a small number of IDSs [16, 17].

2.5.2.1 Signature-Based Detection

Signature-based detectors analyze system activities looking for events matching a predefined pattern or signature that describes a well-known attack. They collect network traffic and then proceed to analyze it [16].

The analysis is based on a comparison of patterns (pattern matching). The system contains a database of attack patterns and will be looking for similarities with them and when a match is detected the warning will be sent.

These systems are truly effective in detecting attacks but they generate a large number of false positives. Therefore it is necessary that the period in which they get regulated (tuning period) is as short as possible.

The proper operation of such a system depends not only on a good installation and configuration, but also on the fact that the database where the attack patterns are stored is updated. The advantage and disadvantage of signature based detection are [16]:

Advantages

- Signature detectors are very effective in detecting attacks without generating a large number of false alarms.
- They can quickly and accurately diagnose the use of a specific attack technique. This can help those responsible for security to easily follow security problems and to prioritize corrective actions.

Disadvantages

- Signature detectors only detect the attacks they previously know, so they must be constantly updated with signatures of new attacks.
- Many signature detectors are designed to use very tight patterns that prevent them from detecting variants of common attacks.

2.5.2.2 Anomaly-Based Detection

The anomaly detection focuses on identifying unusual behavior in a host or a network. They operate assuming that the attacks are different from the normal activity. Anomaly detectors construct profiles representing the normal behavior of users, hosts or network connections [17].

These profiles are constructed from historical data collected during normal operation. The detectors collect data from the events and use a variety of measures to determine when the monitored activity deviates from normal activity. The measures and techniques used in the detection of anomalies include [17]:

- Detecting a threshold on certain attributes of user behavior. Such behavior attributes may include the number of files accessed by a user in a given period of time, the number of unsuccessful attempts to enter the system, the amount of CPU used by a process, and so on. This level can be static or heuristic.

• Statistic measures, which can be parametric, where it is assumed that the distribution of the profiled attributes fits a certain pattern, or non parametric, where the distribution of the profiled attributes is learnt from historical values observed over time.

The advantage and disadvantage of anomaly based detection are [17]:

Advantages

• The IDSs based on anomaly recognition detect unusual behavior. Thus they have the ability to detect attacks for which they have no specific knowledge.

• Anomaly detectors produce information that is very useful to define new patterns for signature detection.

Disadvantages

• The detection of anomalies produces a high number of false alarms due to the unpredictable behavior of users and networks.

• They require very hard training to characterize patterns of normal behavior.

2.5.2.3 Machine Learning -Based Detection

The third categorization of type of analysis, machine learning techniques are popular for so many real time problems. Machine learning techniques are based on explicit or implicit model that enables the patterns analyzed and categorized. It can be categorized into Genetic Algorithms, Fuzzy Logic, Neural Networks, Bayesian Networks and Outlier Detection [18, 19].

Fuzzy Logic

Fuzzy logic is derived from fuzzy set theory under which reasoning is approximate rather than precisely derived from classical predicate logic. Fuzzy techniques are thus used in the field of anomaly detection mainly because the features to be considered can be seen as fuzzy variables. Although fuzzy logic has proved to be effective, especially against port scans and probes, its main disadvantage is the high resource consumption involved [20].

Genetic Algorithms

Genetic Algorithms are biologically inspired search heuristics that employ evolutionary algorithm techniques like crossover, inheritance, mutation, selection, etc. So, genetic algorithms

are capable of deriving classification rules and selecting optimal parameters for the detection process. The application of Genetic Algorithms to the network data consist primarily of the following steps [21]:

i. The Intrusion Detection System collects the information about the traffic passing through a particular network.
ii. The Intrusion Detection System then applies Genetic Algorithms which is trained with the classification rules learned from the information collected from the network analysis done by the Intrusion Detection System.
iii. The Intrusion Detection System then uses the set of rules to classify the incoming traffic as anomalous or normal based on their pattern**.**

Neural Networks

A neural network is the ability to generalize from limited and noisy data that is not complete. This generalization capability provides the potential to recognize unseen patterns, i.e., not exactly matched patterns that are different from the predefined structures of the previous input patterns. The neural network has been recognized as a promising technique for anomaly detection because the intrusion detector should ideally recognize unknown attacks which are not previously observed [22].

Bayesian Networks

A Bayesian network is a model that encodes probabilistic relationships among the variables of interest. This technique is generally used for intrusion detection in combination with statistical schemes, a procedure that yields several advantages, including the capability of encoding interdependencies between variables and of predicting events, as well as the ability to incorporate both prior knowledge and data [20].

The advantages and disadvantages of the above mentioned machine learning techniques are summarized in Table 2.1.

Table 2.1: Various Machine Learning Based Intrusion Detection Techniques

Techniques	**Advantages**	**Disadvantages**
Fuzzy Logic	-Reasoning is Approximate rather than precise. -Effective, especially against port scans and probes.	-High resource consumption. -Hard to develop a model from fuzzy system.
Genetic Algorithm	-Biologically inspired and employs evolutionary algorithm. -Uses the properties like Selection, Crossover, and Mutation. -Capable of deriving classification rules and selecting optimal parameters.	-No constant optimization response time.
Neural Network	- Ability to generalize from limited, noisy and incomplete data. - Has potential to recognize future unseen patterns.	- Need long training time. -Greater computational burden.
Bayesian Network	-Encodes probabilistic relationships among the variables of interest. -Ability to incorporate both prior knowledge and data.	- Lack of available probability data

2.5.3 Response

Once the events have been analyzed and an attack has been detected, an IDS reacts. Responses can always be grouped into two categories: passive and active. The passive IDSs send reports to some others who will then take action on the matter, if it is appropriate. The active IDSs automatically launch replies to such attacks [14].

2.5.3.1 Passive Response

In this type of IDS, the security manager or the system users are notified of what happened. It is also useful to alert the administrator of the site from which the attack was launched, but it is

possible that the attacker can monitor the email of the organization or that he has used a false IP for the attack. In that case it would be useless to alert the administrator.

2.5.3.2 Active Response

The active responses are automatic actions that are taken when certain types of intrusions are detected. Two different categories can be set:

- Collection of additional information: It consists of incrementing the sensor's sensitivity level in order to obtain more clues of the possible attack (e.g., catching all packages from the source that launched the attack, during a certain period of time).
- Changing the environment: Another active response could be to stop the attack; For example, in the case of a TCP connection, the session can be closed by injecting TCP RST segments to the attacker and the victim, or filter the IP address of the intruder or the attacked port, to the access router or to the firewall in order to avoid future attacks.

2.5.4 Detection Time

Two main groups can be identified, those which detect intrusions in real time (in-line) and those which process audit data with some delay (off-line), that means not real time.

Some systems that have in-line detection can also carry out offline detection over historic audit data. These types of systems combining both types of detection time are called hybrids [16].

2.6 Deployment Scenario for IDSs

There are a number of ways to deploy sensor either as network based or host based. In the sections below it shows two ways of deploying IDSs in network based architecture and single way to host based.

2.6.1 Before the Firewall

In this point, the NIDS can keep track of all network events of interest, even those attacks which subsequently may fail. As it has to handle large traffic, NIDS ought to be installed on a faster machine so that analysis is done in real time. Also it has to be configured correctly and number of false alarms can be reduced. Figure 2.2 shows how to deploy IDS before firewall [23].

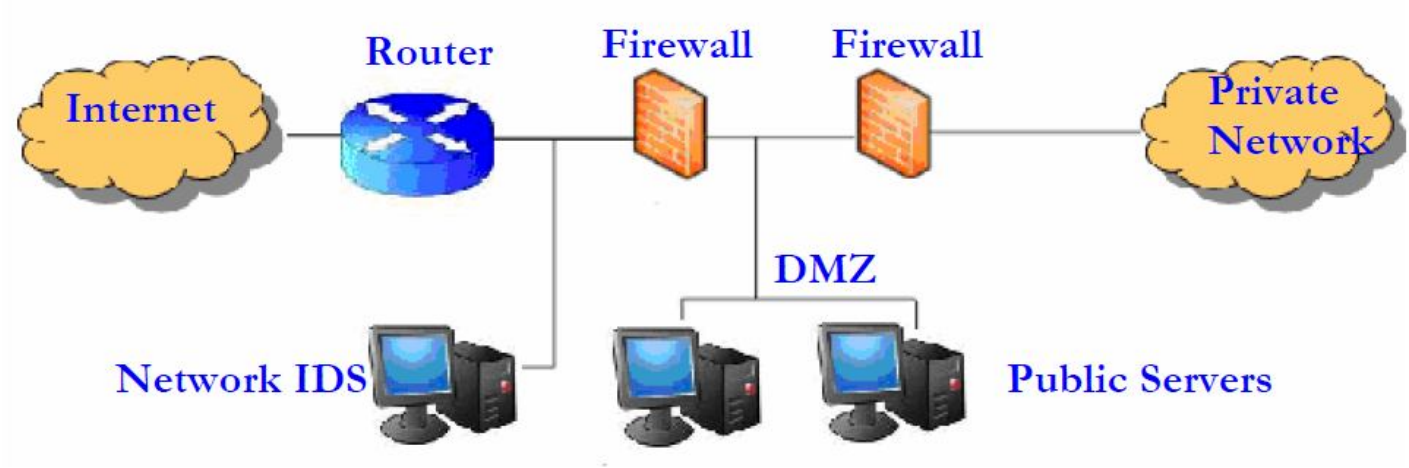

Figure 2.2: Network IDS Placed Before the Gateway Firewall

2.6.2 Inside the Private Network

The other possibility where NIDS can be stationed is within the corporate network as shown in Figure 2.3 [23]. Such a location aims at monitoring the attacks emerging from the local networks and also those which are transmitted via firewall. As the number of attacks possible in this place is lesser than the preceding cases, this makes the application demands smaller. In this case IDS generates few false alarms. The scope of visibility is limited to within the corporate network, thus will not be able to detect the failed attacks as in the previous cases.

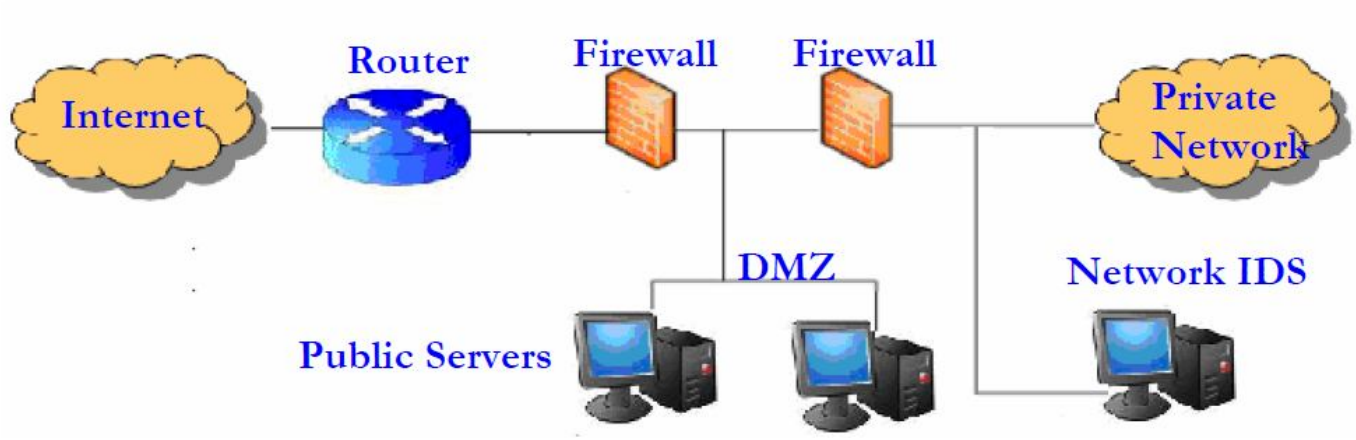

Figure 2.3: Network IDS within the Private Network

2.6.3 Deployment on Individual Hosts

In this case host based IDS will be installed to hosts and gathers information either the operating system audit trials or system logs of host which it has been installed. It does not only monitor the communication traffic in and out of a single computer but also checks the integrity of the system file. Figure 2.4 shows sample host based IDS [24].

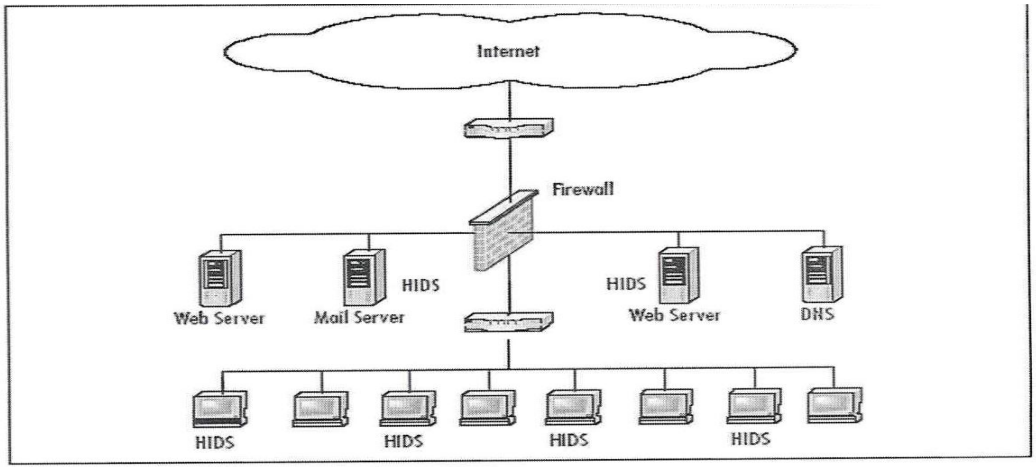

Figure 2.4: Sample Host Based Intrusion Detection System

2.7 SNORT, An Open Source Signature Based IDS

SNORT [25] is a small, lightweight Intrusion Detection System written by Martin Roesch. SNORT is an open source Intrusion Detection System, which may also be configured as an intrusion prevention system for monitoring and prevention of security attacks on networks. It is a cross platform network intrusion detection tool that can be deployed to monitor small TCP/IP networks and detect a wide variety of suspicious network traffic as well as report an attack. SNORT performs protocol analysis and content matching to detect intrusions. It is commonly used to actively block or passively detect a variety of attacks and probes, such as buffer overflows, stealth port scans, web application attacks, SMB probes, and OS fingerprinting attempts.

2.7.1 SNORT Components

SNORT is logically divided into multiple components. These components work together to detect various attacks and to generate output in a required format. These components ride on top of the Libpcap or WinPcap promiscuous packet capturing library, which provides a portable

packet sniffing and filtering capability. The SNORT components are shown in Figure 2.5 and are explained below [26].

i) Packet Capturing: This module is used for capturing the packets. In order to capture real world data there are a number of freely available tools. The packets are captured from network interfaces and passed to the packet decoder module for further processing.

ii) Packet Decoder: It is the module responsible to perform the syntax analysis at MAC, IP and TCP/UDP layers of the IP packet.

iii) Preprocessor: It is the block where multiple preprocessors can be loaded at boot time to analyze protocols of layers above the TCP/UDP with custom made C/C++ programs and use for further processing of the packet coming from packet decoder. Example: - anomaly preprocessor plug in which used to identify unknown attack and can use different detection approach.

iv) Detection Engine: Detection engine finds out if intrusion activity is present in packet or not. SNORT rules are used for this purpose. If any rule is matched, appropriate action is taken. Action may include generating the alerts.

v) Logging & Alerting: It is the module managing the log output. The output log is configurable depending on user needs. Alerting based on the rules is also generated by the administrator.

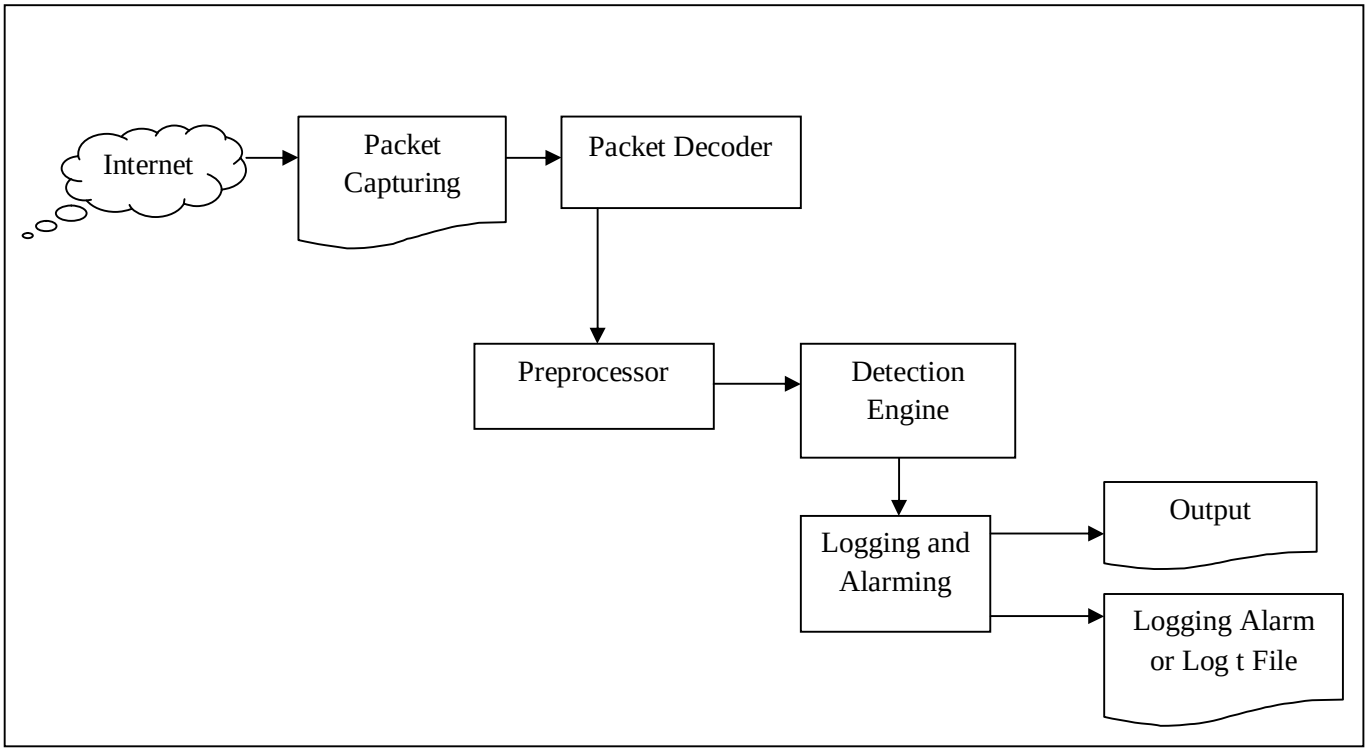

Figure 2.5: Components of SNORT

2.7.2 Basic Feature of SNORT

The basic features of SNORT include packet capturing, logging, performing real time analysis, content searching and matching and detecting attacks which are explained in detail as follows [26]:

i) Packet capturing and logging: SNORT captures the packets using Libpcap or Winpcap to capture the packets coming on an interface.

ii) Real time traffic analysis: SNORT is capable of performing analysis of packets coming on the network in real time. It is also capable of performing analysis on captured packets which can be captured using various tools like Wireshark.

iii) Content searching/matching: SNORT performs the content searching of the packet's content against the rules in the rules file.

iv) Block or passively detect a variety of attacks and probes: SNORT can block the packets over the network when it is operating in inline mode. It is able to detect malicious activities over the network. Figure 4.3 shows how SNORT components work.

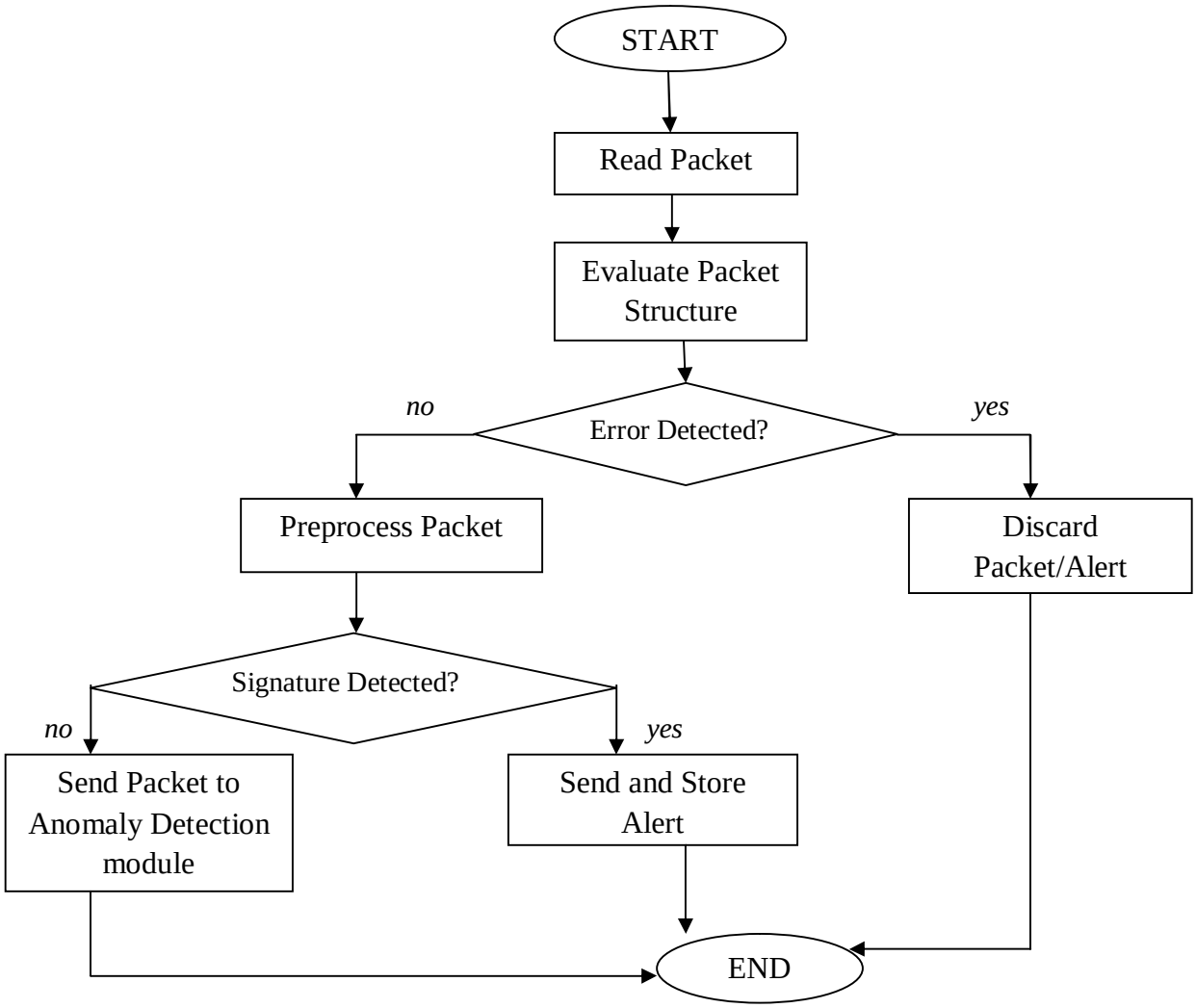

Figure 2.6: How SNORT Components Work

2.7.3 Writing SNORT Rules

SNORT uses a simple, lightweight rules description language that is flexible and quite powerful. This section explains the instructions followed while writing SNORT rules [27]. SNORT rules have two parts *i.e.* Rule header and Rule options.

- **Rule Header**

The rule header contains the information that defines the who, where, and what part of a packet, as well as what to do in case a packet with all the attributes indicated in the rule shows up. Rule header in a SNORT contains all those portions which are the compulsory part. It includes following parts [27].

i) Rule Actions: The rule action tells SNORT what to do when it finds a packet that matches the rule criteria. There are five available default actions in SNORT.

a) Alert: Generate an alert using the selected alert method, and then log the packet

b) Log: Log the packet

c) Pass: Ignore the packet

d) Activate: Alert and then turn on another dynamic rule

e) Dynamic: Remain idle until activated by an activate rule, then act as a log rule

f) Drop: Make IP tables, drop the packet and log the packet

g) Reject: Make IP tables, drop the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.

h) Sdrop: Make IP tables drop the packet but does not log it.

ii) Protocol: The next field in a rule is the protocol. There are four protocols that SNORT analyzes for suspicious behavior that are TCP, UDP, ICMP and IP.

iii) IP Addresses: The keyword any may be used to define any address. The addresses are formed by a straight numeric IP address and a CIDR block. The CIDR block indicates the net mask that should be applied to the rule's address and any incoming packets that are tested against the rule. Source IP and Destination IP are written in the rule header.

iv) Port Numbers: Port numbers may be specified in a number of ways, including any ports, static port definitions, ranges, and by negation. Any ports are a wildcard value, meaning literally any port. Static ports are indicated by a single port number, such as 111 for port mapper, 23 for telnet, or 80 for http, *etc.* Port ranges are indicated with the range operator (:). Port negation is indicated by

using the negation operator (!). Both source and destination ports are required in rule. The keyword any may be required to use any port number.

v) The Direction operator: The direction operator -> indicates the orientation, or direction, of the traffic that the rule applies to. The IP address and port numbers on the left side of the direction operator is considered to be the traffic coming from the source host, and the address and port information on the right side of the operator is the destination host. There is also a bidirectional operator, which is indicated with a <> symbol. This tells SNORT to consider the address/port pairs in either the source or destination orientation.

- **Rule options**

Rule options [27] form the most important part of SNORT's intrusion detection engine, combining ease of use with power and flexibility. All SNORT rule options are separated from each other using the semicolon (;) character. Rule option keywords are separated from their arguments with a colon (:) character. There are four major categories of rule options.

i) General rule options: These options provide information about the rule but do not have any affect during detection**.**

ii) Payload detection rule options: These options all look for data inside the packet payload and can be inter-related.

iii) Non-Payload detection rule options: These options look for non-payload data.

iv) Post-detection rule options: These options are rule specific triggers that happen after a rule has fired.

2.7.4 SNORT Rules Classification

The rules in SNORT are able to perform the analysis on network data on different criteria. The multi-rule search engine is broken into three distinct searches based on unique SNORT rule properties [28].

i) Signature detection rule: SNORT handles its signatures based detection with the rules created for it. The signatures allows a rule to specify a byte string to match against anywhere in the packet including the payload data. Example of the rule is shown in (2.1).

alert tcp any any -> any 80 (msg: "CodeRedII root exe"; flags: A+; content: "root.exe"; depth:624; classtype: attempted-admin;) ...(2.1)

The rule looks for the "CodeRedII" worm in the packet and displays the message "CodeRedIIroot exe" if the worm is detected.

ii) Protocol detection rule: SNORT is provided with rule set for detecting protocols and suspicious behavior of these protocols in use. SNORT rules have the protocol field that allows the detection of the protocol. Example is shown in (2.2).

alert tcp any any -> 192.168.1.0/24 any (flags: SF; msg: "SYNC-FIN packet detected";)...(2.2)

The rule detects any scan attempt using SYN-FIN TCP packets. The flags keyword is used to find out which flag bits are set inside the TCP header of a packet.

iii) Anomaly detection rule: The packet anomaly search allows a rule to specify characteristics of a packet or packet header that is cause for alarm. Many attacks use buffer overflow vulnerabilities by sending large size packets. Using the 'dsize' keyword in the rule, one can find out if a packet contains data of a length larger than, smaller than, or equal to a certain number. Example is shown in (2.3).

alert ip any any -> 192.168.1.0/24 any (dsize: > 6000; msg: "Large size IP packet detected";) ...(2.3)

The rule generates an alert if the data size of an IP packet is larger than 6000 bytes [48].

2.7.5 Operation Modes of SNORT

SNORT runs in different operational modes which are as follows [28]:

- **Sniffer Mode:** This mode reads the packet off the network and displays them in a continuous stream on the console.

Command: ./SNORT –[option]

Options to run in a sniffer mode are:

i) v: The command prints the TCP/IP header on the screen.

ii) vd: The command prints the application data too.

iii) vde: The command prints the data link layer contents as well.

- **Packet Logger Mode:** This mode logs the packets to the disk. Options to run SNORT in sniffer mode are:

  i) ./snort –dev –l ./log The command logs the packets to the specified directory.

  ii) ./snort –l ./log –b : The command performs the binary log, binary file may be read back using –r switch).

- **Network Intrusion Detection Mode (NIDS):** It is most complex and configurable mode. It allows SNORT to analyze network traffic against a user defined rule set and performs actions based on detections.

  Command: ./snort –c snort.conf

  where snort.conf is the name of SNORT configuration file. This will apply the rules configured in the snort.conf file to each packet to decide if an action based upon the rule type in the file should be taken.

- **Inline Mode:** It obtains packets from IP tables instead of the Libpcap or Winpcap and then causes IP tables to drop or pass packets based on snort rules that use inline-specific rule types. This is the mode used when SNORT is to act as an IPS [27].

Chapter 3: Related Work

The Intrusion Detection area of research is about two decades old, beginning with James P. Anderson's technical report [29]. In his report Anderson divided the possible attackers of computer systems into four groups: external penetrators, masqueraders, misfeasors, and clandestine users. Since then, different researchers have used a number of approaches in the development of intrusion detection systems to overcome those attacks. This chapter reviews different researcher works based on the detection approach.

3.1 Signature Based Intrusion Detection System

A variety of tools have been developed for the purpose of network intrusion detection and detect anomalies by matching the traffic pattern or the packets using a set of predefined rules that describe characteristics of the anomalies.

The work in [30] uses the open source network intrusion detection tool which is SNORT and tests it on high speed network that is in the campus. Their work concentrates on ICMP flood attack and they detected 12 signatures among which ICMP PING has maximum alert. But this work is not efficient on detecting novel attack and the SNORT was down during heavy traffic.

According to [31] they mainly concentrated on implementing IDS on wireless LAN. They proposed a new Network Intrusion Detection System (NIDS) that will mainly detect the most prominent attack of Wireless Networks, i.e., DoS attack and Man-in-the-Middle attack. This proposed IDS works as lower layer of firewall which means after checking from the database using IP address and system content and if intruder is detected it sends it to the firewall for blocking. This work assumed that it is part of firewall system. So the main problem of this system is it only checks the database, concentrate only on two kinds of attacks but there are a number of attacks in networking also it sends for firewall for decision making and it is only governed by making decision allow and deny no alerting and signature generation for the new attack.

3.2 Anomaly Based Intrusion Detection System

In anomaly based intrusion detection systems the frequently used method is to set up a statistical model of normal network traffic. A deviation from the model will be marked as suspicious.

Those statistical models are built from different perspectives of the traffic analysis. Current network anomaly systems such as ADAM [32] and SPADE [33] belong to this category.

ADAM is a statistical anomaly-based IDS developed at George Mason University, which uses a rule-based data mining technique for network intrusion detection. ADAM builds a normal profile by mining attack free data and it also has a rule set, which is initialized by user defined abnormal patterns and is constantly updated with new rules. ADAM obtains the good result when applied to DARPA evaluation data set. However, it is highly dependent on the training data even if they use pseudo-Bayesian system to avoid dependency.

SPADE builds the statistical model on the IP addresses and ports. For example, SPADE uses SNORT as the engine and builds the normal traffic model on the number of connections observed from certain IP and port pairs. The less frequent IP port pair is more likely to be flagged as suspicious. The drawback of SPADE is that its false alarm rate is high on the traffic from less frequent IP-port pairs.

In [34] the researchers used three different statistical methods, i.e., chi-square, Gaussian mixture distribution and Principal Component Analysis (PCA). They measured the performance of these three methods by generating performance log from hosts. After the performance log has been generated for each day, the log is divided into 4 groups, and the average values for each column of the table are calculated. After finding the average values, the values are maintained in another table. These values are used as normal data set. They obtained results in terms of detection rate and false alarm for PCA and Gaussian 97.5% and 2.5% respectively but for chi-square 90% and 10%. So this work shows that most statistical methods generate false alarm which degrades the performance of the intrusion detection system.

In [35] the authors did a survey on incremental method for anomaly detection and tried to evaluate the problems of anomaly detection which are high false alarm rate, non-scalability issue and not fit for high speed networks. Most of the works evaluated are based on the benchmark of KDD dataset which is not updated and they said that they observe good result in most of the work but it does not have any idea whether it works good for real network data or not and even they suggested another technique to be combined with the incremental approach to have better performance.

Other work in [36] proposed a hierarchical off-line anomaly network intrusion detection system based on Distributed Time-Delay Artificial Neural Network due to the assumption that this method is flexible than the others, efficient and simple to classify dataset. The author measured the performance and compared it with other works and it performs better but still it has to be combined with other methods to increase performance and it does not deal on real time data.

The work in [37] proposed a multi-level hybrid classifier utilizing a combination of tree classifiers and clustering algorithms. One of the most interesting ideas of this work is that they fuse both supervised learning (tree classifiers) and unsupervised learning (clustering) techniques. Although the supervised learning technique needs a clear label for training, it was claimed in the paper that unsupervised learning might play an essential role on improving the detection rate. Using the KDDCUP 1999 data set, they evaluated their approach and compared it with other popular approaches (i.e., MADAM ID and 3-level tree classifiers). Evaluation results showed that their hybrid approach was very efficient in detecting intrusions with an extremely low false negative rate of 3.37%, while keeping an acceptable level of false alarm rate of 9.1%. The problem of this work, even if they use multiple algorithms for classification and obtained a good result, it only detects unknown attacks in a network and give emphasis for offline dataset only. Also the detecction rate of this work is not clearly described in the document.

3.3 Hybrid Intrusion Detection System

The above mentioned approaches in which using single detection technique for each architecture have both advantages and disadvantages. None of the approaches is better than the other. Each approach has some limitations. No approach is capable of detecting all the attacks. So many authors have proposed IDS based on a hybrid approach which consists of combination of techniques mentioned above. Using a hybrid approach eliminates many more false positives and false negatives. Hybrid based IDS has been discussed by the following authors in their research work.

The work in [38] proposed a hybrid intrusion detection system combing a signature-based method, SNORT, and an anomaly detection system. In contrast to the other hybrid IDSs, their approach only relies on SNORT to generate the alerts, and the anomaly detection is only used to automatically generate SNORT. To this end, normal traffic is passed to the anomaly system to

build a normal profile of frequent episode rules (FER). Having done with the training phase, the real traffic will be fed to the system. FERs generated from the real traffic will be compared to the normal profile and considered as suspicious if it does not match any of the FREs in the normal profile. When the matched rule occurs beyond the threshold, it will be reported as an anomaly and the system will automatically add the rule to the SNORT. Instead of combining signature detection techniques and anomaly detection techniques, some other hybrid systems fuse multiple anomaly detection systems according to some specific criteria considering that the detection capability for each anomaly detection technique is different. The main goal of such a hybrid system is to reduce the large number of false alerts generated by current anomaly detection approaches and at the same time keep an acceptable detection rate. The experimental results show a 60 percent detection rate of the HIDS, compared with 30 percent and 22 percent in using the SNORT and Bro systems, respectively. The main drawback of this work is they did not use any module that describes about the counter measure after the intrusion detected and use limited Internet trace data for training purpose in real time even if the performance of an IDS is dependent on the training data.

The work in [39] tried to analyse the performance of SNORT combined with different statistical based anomaly detection methods, i.e., Application Layer Anomaly Detector (ALAD), Learning Rules for Anomaly Detection (LERAD), and Packet Header Anomaly Detection (PHAD). The data used for evaluation is mixed data which is simulated data and real data collected from SNORT. In this case SNORT is set as sniffer mode and got a performance of 83%. Then they proposed a new semi supervised method to improve the detection performance and they got 98.88% accuracy and false alarm rate of 0.5533% after training on 2500 data instances. The main drawback of this system is during testing the statistical method they used mixed data and got lesser performance but in semi-supervised they used only simulated data and got better performance. So it is not possible to make sure that whether the semi supervised approach will have good performance or not in real time data.

The work in [40] proposed a new intrusion detection system by adding anomaly pre-processor to SNORT and extends its functionality to hybrid scheme. The general scheme of the anomaly detection module uses two different operation modes: training mode and anomaly detection mode. Using the training mode the system records in a database the network traffic considered as

normal and expected. Later, a profile of this network activity is automatically created, and the anomaly detection module stores in the database the abnormal activity. In this paper, they have implemented a basic statistical method that consists of using the moving average corresponding to the network traffic, which is used to generate the profile of the network. The problem in this work is they said that the result obtained denote that long time training of the system will improve the detection rate but they did not give any description and comparison on their result.

Another work in [41] proposed a hybrid intrusion detection system that combines anomaly based and signature based. The proposed method includes ensemble feature extraction and data mining classifier. The former consists of four classifiers using different sets of features and each of them employs a machine learning algorithm named fuzzy belief *k*-NN classification algorithm. The latter applies data mining technique to automatically extract computer users' normal behavior from training network traffic data. The main concern of this work is not only to increase the detection rate but also to reduce the false alarm rate. The problem here is that even if they said that their work is hybrid system it concentrates on the anomaly detection and this work does not give any description of how the signature based system works and how they integrated the two approaches. The other problem of this work is that they described in their paper they achieved their goal and got good result using KDD dataset but their experimental evaluation is missed in the paper.

In the research work of [42] they discussed a methodology to apply artificial intelligence to signature based intrusion detection system SNORT. Their hybrid model is the combination of SNORT system, neural network and involving data mining technique. In this research different steps are involved. The first one takes input from various sources; including network packets, log files, and system call traces. Input is collected, organized and then forwarded to one or more analyzers. The second one is the SNORT system which is used to detect intrusion by first parsing. The third step is using neural network classifier with combination of misuse technique which efficiently and rapidly classifies observed network packets with respect to attack patterns which it has been trained to recognize. Then the result will be declared, report generated after that if it detects intrusions it raises an alarm. This work is like an introduction or it is just to create awareness on how to use SNORT with artificial intelligence. No tangible performance is obtained.

Other work in [43] proposed a two-stage hybrid intrusion detection and visualization system that leverages the advantages of signature-based and anomaly detection methods. It was claimed that their hybrid system could identify both known and unknown attacks on system calls. However, evaluation of results for their system are missing in the paper. The work is more like an introduction on how to apply a multiple stage intrusion detection mechanism for improving the detection capability of an IDS. No indepth account of the work is presented.

3.4 Summary

The works provided in Section 3.1 are benchmarks for signature based systems that are basically dependent on well known open source SNORT. Even if these works have contribution for security then, it has different challenges. The first one is it requires entry for the database also complete database needs hundreds of even thousands of entries, This means it need high memory and CPU capacity. The second one is it does not detect novel attack even if it is on database and a little modification done it cannot detect. So in order to give a solution for these basic issues researchers provide a new work which is indicated in Section 3.2 using anomaly based approach. But the problem here is there are number of false positives.

The other recent works done are discussed in Section 3.3 which try to overcome the above mentioned problems by combining the advantage of the two approaches. Most of the works that follow hybrid approaches use almost similar architecture and the only difference is the algorithm they use for anomaly detection and the steps they follow. Most researchers apply this approach in offline data so the system will not examine real time attack taking over though a network. On the other hand some researchers use only online data in this case in order to say this packet is normal or attack they set some threshold; otherwise they need knowledge engineers; most of the time this might cause high false positive rate.

The other issue, the improvement of intrusion detection system comes through ages by applying different techniques. This improvement is measured through maximum detection rate and minimum false alarm rate. So most works done before even if they use combined approach and multiple detection algorithms the result is less detection rate and high false alarm rate. So the proposed hybrid intrusion detection system aims to use singe detection algorithm to minimize complexity but tries to give better detection rate and minimal false alarm rate.

Chapter 4: Design of the Proposed Hybrid Intrusion Detection System

4.1 Introduction

In this Chapter we have presented the design of the proposed hybrid intrusion detection system. Different components of the proposed hybrid IDS are described with their relevance and techniques to use while building those components. The Chapter presents the architecture with implemented algorithms.

A hybrid intrusion detection system is a system which is a combination of both signature based and anomaly based IDS. A signature-based IDS analyzes the network traffic looking for patterns that match a library of known signatures. These signatures are composed by several elements that allow identifying the traffic. On the other hand anomaly-based IDSs try to find suspicious activity on the system. In the initial phase, the IDS must be trained in order to get an idea about what is considered “normal” and “legitimate”. After that, the system will inform about any suspicious activity if there is a deviation from normal.

4.2 Requirements of Hybrid Intrusion Detection System

To build an IDS, a certain requirement must be fulfilled to make the system efficient. Before describing those requirements, it is necessary to introduce four important terms that clearly define the requirements [44]:

- False Positive (FP): represents the number of instances which are classified as anomalous by the IDS but they are normal.
- True Positive (TP): represents the number of instances which are classified as anomalous by the IDS and they are truly anomalous.
- False Negative (FN): represents the number of instances that are classified by the IDS as being legitimate when in fact they are anomalous.
- True Negative (TN): represents the number of instances that are classified by the IDS as being legitimate and that really are truly legitimate.

In IDS the main problem is false alarm rate. For instance, if we consider a 10Gbit/s Ethernet network, the number of packets per second that an IDS should be able to handle vary between

812,740 and 14,880,960 [45]. If the IDS misclassifies one packet every second (or every 14,880,960 packets) as being anomalous when it is not, this means that the network administrator will have to deal with 86,400 false alerts at the end of the day. In this case, the IDS becomes a pain for the administrator who will most probably not use it any more even though the primary objective of the IDS was to help him. So in order to make the IDS helpful it should have to fulfill the following requirements [44].

- Accuracy: this property ensures that the IDS does not classify legitimate instances as anomalous.
- Performance: it must be able to classify the traffic without adding a noticeable overload to the network. The performance of an IDS is measured using confusion matrix including training and testing time.
- Completeness: this property is the core of the IDS. It states that an IDS should be able to detect all intrusion attempts leading to a false alarm rate equal to 0. In practice, this property is very hard to achieve because the IDS must be able to detect known attacks as well as the new ones.
- Fault Tolerance: an IDS must itself be resistant to attacks.
- Scalability: an IDS must be able to process the traffic of the network in real-time without dropping any packets because of a higher bandwidth than what the IDS can handle. The IDS must be designed in order to be robust in the worst case scenario.

4.3 System Architecture

In this Section, we have proposed an architecture for hybrid intrusion detection system that can increase the detection performance of the intrusion detection system. It has different components as shown in Figure 4.1. Each component can be implemented either in network based architecture or in host based architecture.

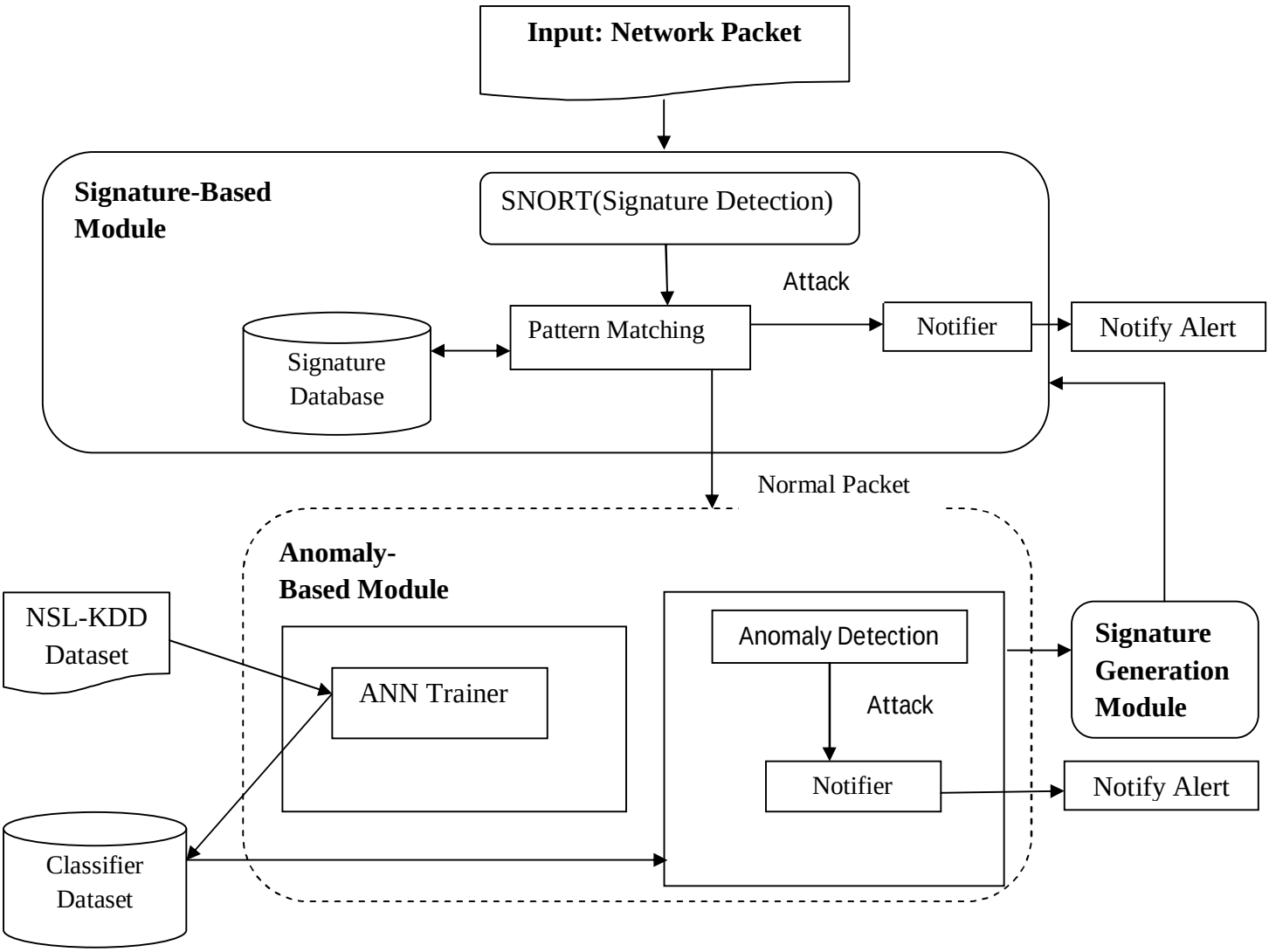

Figure 4.1: Proposed Architecture for Hybrid Intrusion Detection System

Initially data will be captured using data capturing tool from a given network and the signature based module will accept those collected packets and match the pattern from the signature database. If it matches to the database then if it is known it will sent it to the notifier and it will sent an alert otherwise, it will send to the anomaly detection module. Anomaly detection module will have two phases which are training mode and detection mode. In training mode, it will identify what is normal and abnormal using a simulated dataset then store the trained data using training algorithm . Then the detection module will accept packets from signature based module and identify those activates that deviated from the normal, record them as intruder notify alert also send it to signature generate module to create signatures.

4.4 Major Components of the Proposed IDS

4.4.1 Signature Detection Module

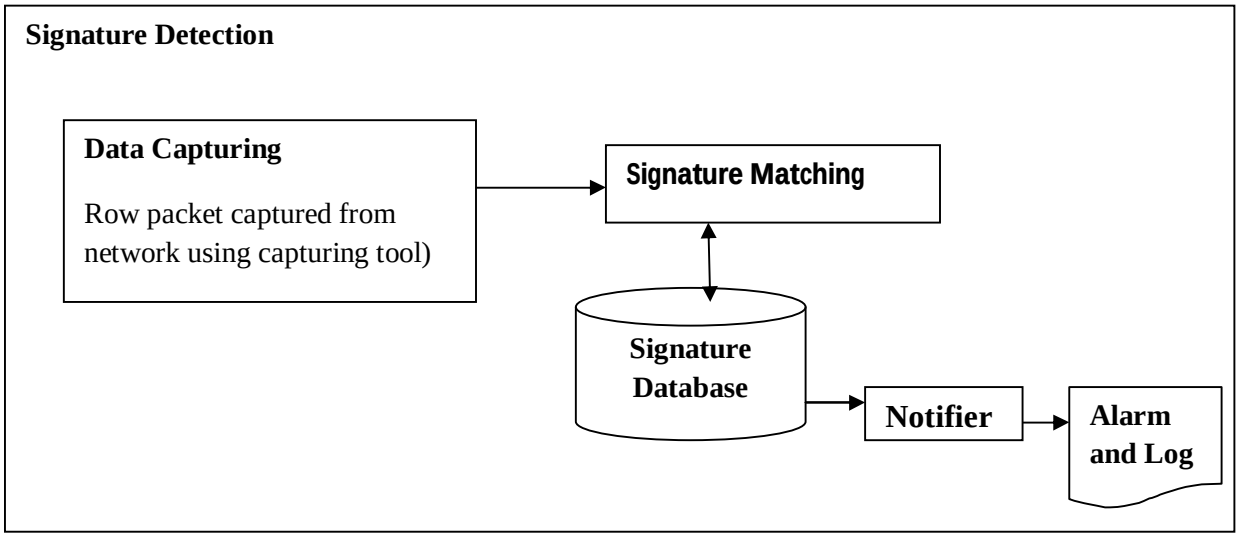

Figure 4.2: Detail Description of Signature Detection Module

As we mentioned above, this signature detection module is responsible to match patterns of the packet to the given signature database. In this module we have used an open source SNORT. There are many reasons to select SNORT in our thesis but the main reason is because can run in different mode means it can be configured as sniffer mode , intrusion detection mode and intrusion prevention mode, freely available, lightweight, can run both in hosts and network based architecture easily. The proposed signature detection module goes from data capturing up to attack detection comparing the coming packet with the database. The detail of this module is presented below.

Data Capturing

In order to capture data we use windows packet capturing library (winpcap) (i.e., a popular freely available capturing tool which is used to capture packet in windows environment). In windows environment we have install winpcap tool and create connection to the network card using IP address. The process is shown in Pseudo code 4.1.

```
Input: Network IP
Process:
            If network card on then
                Capture packets
            End If
Output: Row Packet
```

Pseudo code 4.1: A Pseudo code for Data Capturing

Signature Matching

This is done by using the selected open source SNORT that has its own pattern matching algorithm. First SNORT reads the captured packet and then preprocessing procedure is performed over the set of strings, and then all packet contents are searched at once in the search phase. The algorithm starts in the idle state as the root state of the automata. Then the characters of the first pattern are added to tries one by one. If the entire strings match then the system sends alarm to the administrator else it will log as normal packet. Pseudo code 4.2 shows steps for signature matching.

```
Input: Row Packet
Process:
            If string of each packet found then
                All packets contents searched
                Algorithm starts at ideal state
                Add patterns to tries one by one
                Entire string matching
            End If
Output: Alarm, Log
```

Pseudo code 4.2: A Pseudo code for Signature Matching

Signature Database

The signature database enables the IDS to have a set of signatures, criteria or rules against which they can compare packets as they pass through the sensor. The database of signatures needs to be installed along with the IDS itself. After the signature database is in place, the signature matching will compare the coming packets from data capturing. The signature database contains mainly snort database that have different tables like signature, sensor, tcphdr, iphdr, and so on.

Notifier

This component is found in both signature based and anomaly based module which is responsible to accept the result after decision is given on each packet to be alerted. Then it will notify the administrator by giving this alert.

4.4.2 Anomaly Detection Module

This module works based on the training of normal activities and if there is any operation out of normal it will record it as abnormal activity in the network (intruder). It has different components and works in two different operation modes: training mode and detection mode. Figure 4.3 shows that the components of the anomaly detection module. And the detail of this module is presented below.

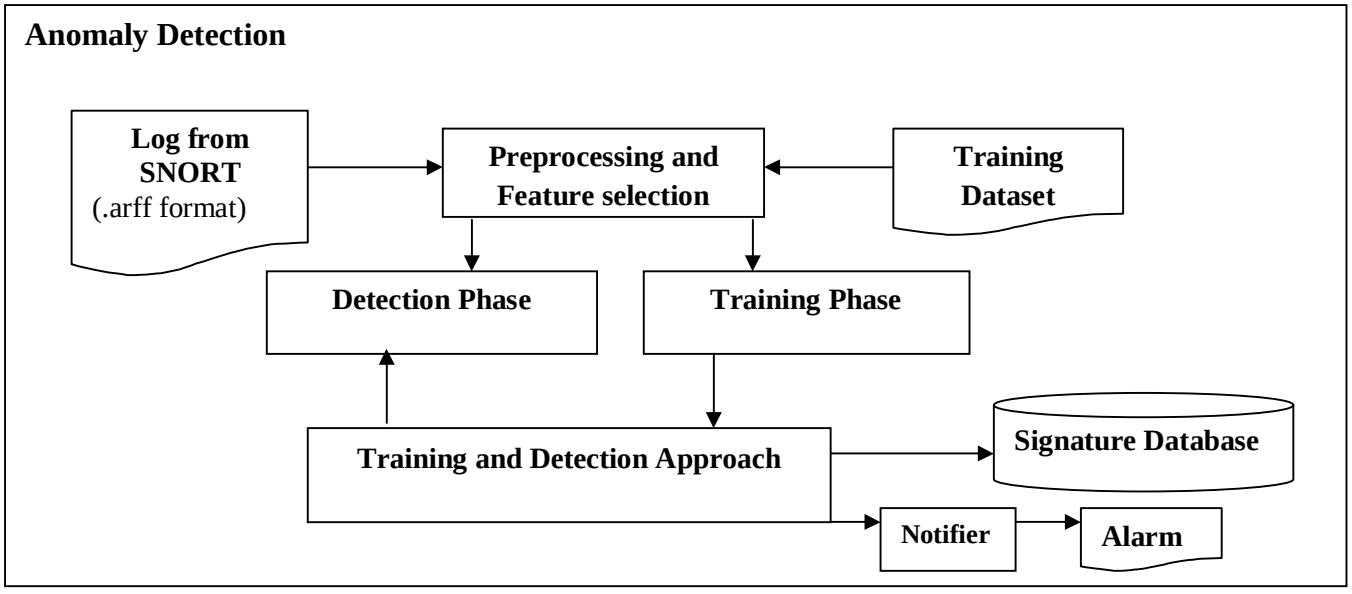

Figure 4.3: Detail Description of Anomaly Detection Module

Log from SNORT

One of the input used for this module is, log from SNORT which is assumed as normal packet by SNORT but it may not be in reality. This data is unlabeled, real time dataset and used after converting it in to weka readable format (.arff in this thesis work) in the detection phase after the system is trained what is normal and abnormal then using this dataset we can classify or detect attacks.

Training Dataset

The other input for this module is, this offline dataset which is used in the training phase of this module. It is labeled dataset that can easily learn the system. For our case, we have used simulated dataset called NSL-KDD for this phase. This dataset is selected because it is the latest version of all simulated dataset in the area of network security; redundant records are eliminated from training set and it is affordable to use for experiment purpose as it consists of reasonable number of instances both in the training and testing set [11]. The detail description of this dataset is provided Appendix 1 and 2.

Preprocessing and Feature Selection

Preprocessing

In this preprocessing phase the main task done is transforming symbolic column of the data to numeric. Each column has its own customization table, depending on the column content. After analyzing the protocol column, it has been shown that it has three protocols values: TCP, UDP, & ICMP. Tables below demonstrate the customize transformation tables for protocol and flag feature columns.

Table 4.1: Protocol Column Feature Transformation

Protocol	No
TCP	1
UDP	2
ICMP	3

Table 4.2: Flag Column Feature Transformation

Flag	No
OTH	1
REJ	2
RSTO	3
RSTO0	4
RSTR	5
S0	6
S1	7
S2	8
S3	9
SF	10
SH	11

Sample row of the dataset before and after transformation are shown below.

0, tcp, ftp_data, SF, 491, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0,

1, 0, 0, 150, 25, 0.17, 0.03, 0.17, 0, 0, 0, 0.05, 0, normal.....Before Transformation

0, 1, 18, 10, 491, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 1, 0, 0,

150, 25, 0.17, 0.03, 0.17, 0, 0, 0, 0.05, 0, 1,0,0,0,0.....After transformation

Feature Selection

Feature selection is an important issue in intrusion detection system because it answers the question that which feature is truly useful for the classification and elimination of useless features enhances the accuracy of detection while speeding up the computation, thus improve the overall performance of an IDS. In this research work, we have used Principal Component Analysis (PCA) algorithm because it well known and outperforming in other research also it retain features with their originality. Pseudo code 4.3 shows how PCA select features.

```
Input: NSL-KDD dataset
Process:
  BEGIN
    Read features from simulated dataset
    Subtract the mean
    Calculate the covariance matrix
    Calculate the eigenvectors and eigenvalues of the covariance
    matrix
    Choosing components and forming Feature Vector (FV)
   END
Output: New Dataset with reduced dimension
```

Pseudo code 4.3: A Pseudo code for Feature Selection

Training and Detection Approach

To start our proposed hybrid intrusion detection system having this training and detection approach is a must. In order to do this research we have selected artificial neural network as a training and detection algorithm this is because we have got a number of advantages that are stated below [46].

- They have inherent property of learning through training.
- The complex internal structures make them to learn and accommodate large number of patterns.
- They can easily generalize form similar patterns through the knowledge they get it from training.
- They have efficient storage capacity for large patters.

Training Phase

The objective of this training phase is to train the system with artificial neural network for classifying the real time data logged by SNORT with the desire attack and normal. In this phase both the input and output data set is available and supervised learning method can be used. The system gathers knowledge about the normal behavior of the network users from the preprocessed input data, and stores the acquired knowledge. The input to the module is the simulated network data packets and it accepts the TCP, UDP and ICMP connection. Pseudo code 4.4 shows steps in the training phase.

```
Input: New Dataset with reduced dimension
Process:
  BEGIN
      Read features from dataset with reduced dimension
      Define network structure by giving input, hidden and output
      layers
      Initialize the weighted matrix and biases
      Update connection information to ANN
  END
Output: The weight of the neural network attain a stable value
```

Pseudo code 4.4: A Pseudo code for Training Module

Detection Phase

In this phase the system is already trained with the knowledge of which is normal or attack. The input data for the detection phase is retrieved from SNORT which is assumed normal by SNORT. The two phases share the same functionality. When we are talking about the training phase we mean that we are storing knowledge about normal and attack data. But when we mean detection it is retrieving the stored knowledge. So in detection phase deviation from the records assumed normal, it means that something abnormal is happening, and an incidence of abnormality is registered by the system. Pseudo code 4.5 shows how the detection phase works.

```
Input: New Dataset
Process:
  BEGIN
      Read new dataset logged by SNORT
       Call neural network to classify as attack or normal
            If status is attack then
               Sent alarm and generate new signature
            End If
        END
Output: Alarm and New signature
```

Pseudo code 4.5: A Pseudo code for Detection Phase

4.4.3 Signature Generation Module

The signature generation module generates the rules that would be passed on to the Signature-based IDS, Snort. It will receive its data from the anomaly detection module which is an input file. The selected network attributes originating from the previous module serves as the basis for the signature creation. A set of signatures will be generated in each distinct threat coming from a specific source that was identified. Once the administrator pass signature creation new. rules will be generated to C:\Snort\ rules. Samples for rule generation are as follows.

```
alert ip any any -> 192.168.1.0/24 any (dsize: > 6000; msg: "Large size IP
packet detected";)
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"DOS Apache APR
apr_fn match infinite loop denial of service attempt";
```

CHAPTER 5: Implementation and Experiment

In this Chapter the implementation detail for hybrid intrusion detection system will be presented. We have used this implementation to evaluate the performance of the proposed work which is the combined approach of signature and anomaly detection and the evaluation will be presented here. In this Chapter we will cover overview of how the implementation is done, tools used to do this work, how components of the system are implemented and finally measure the performance of the work done.

5.1 Overview

The main goal of this research is to design a hybrid intrusion detection system using combined approach of signature based and anomaly based intrusion detection. This work is proposed in order to detect both known and unknown anomalies in a given network. The flow of this work starts by installing signature based open source SNORT.

On the other hand the basic work in which this thesis concentrates is on anomaly based intrusion detection system. It has training phase to train what is normal for the system then it uses detection phase to detect deviation from the normal activities in a network. It uses different algorithms i.e., PCA for dimension reduction and artificial neural network for training and detection that are selected to maximize the detection rate and minimize the false alarm rate.

5.2 Tools Used

Several tools are used in the development of the proposed intrusion detection system. The following list of programming; different open sources and database management tools have a great contribution for the accomplishment of this work.

- SNORT is signature based intrusion detection system.
- Windows packet capturing library (Winpcap) is an open source which is responsible for real time packet capturing and operates in Windows environment.
- Waikato Environment for Knowledge Analysis (weka) is a toolkit which is composed of machine learning and data mining in which this toolkit is used as it is or it can be called from Java code.

- Basic Analysis and Security Engine (BASE) is an open source code written in the PHP programming language which displays information from a database in a user friendly web front end.
- Java programming language is used to develop the anomaly based module which is a general-purpose, concurrent, strongly typed, class-based object-oriented language. We chose this programming language because it uses strong libraries and fair level of abstraction of unnecessary details.
- NetBeans Integrated Development Environment (IDE) Version 8 is used for developing the prototype with Java. This version of NetBeans supports Java Platform Standard Edition (Java SE) specification Version 8 with Java Development Kit (JDK) 8 and Java SE Runtime Environment (JRE) 8.
- MySql is a database management tool which is used to store signature database.
- Principal Component Analysis is an algorithm which is useful to select relevant features in a given dataset. The detail of this algorithm is presented in Appendix 3.
- Artificial Neural Network is a data mining algorithm which is used in this thesis as classification and detection algorithm. There are many types of ANN in which the detail is shown in Appendix 4.

5.3 Implementation of the Components

The starting point for this research work is the signature based module which is mainly focused on the above mentioned open source IDS. Installation of SNORT has some complexity due to its dependency to other software for capturing packet, storing signature log and alerts and to display alerts.

5.3.1 Signature Based

This research work is done on Windows 7 PC so we have to first install packet capturing tool windows version Winpcap 4.1.3, MySql 5.0.45 then SNORT version 2.9.0.3. SNORT needs some configuration to adjust with the given network and for Windows. Signature based module also needs a number of steps that is stated below.

Step 1: SNORT Configuration

The snort.conf file defines how SNORT will run once the application is started. Sample snort.conf file how it is edited to make SNORT functional is as follows:

- To configure the network variable:

```
# Setup the network addresses you are protecting
var HOME_NET 192.168.0.106/24, 10.4.15.1/24, 10.6.39.1/24
# List of DNS servers on your network
var DNS_SERVERS 10.90.10.28/10.90.10.31
```

- Paths of rule files which is appropriate for window user:

```
var RULE_PATH c:\SNORT\rules
var SO_RULE_PATH c:\SNORT\so_rules
var PREPROC_RULE_PATH c:\SNORT\preproc_rules
```

- Configure output plug in

```
output log_tcpdump: tcpdump.log
output database: log, mysql, dbname=SNORT user=root
password=arsema host=localhost
```

Step 2: MySql Configuration

The first step here is to start MySql service then using command line interface creates a database SNORT. Then check the table content of the database using command Show tables; and it should have to display list of tables shown in snapshoot Figure 5.1.

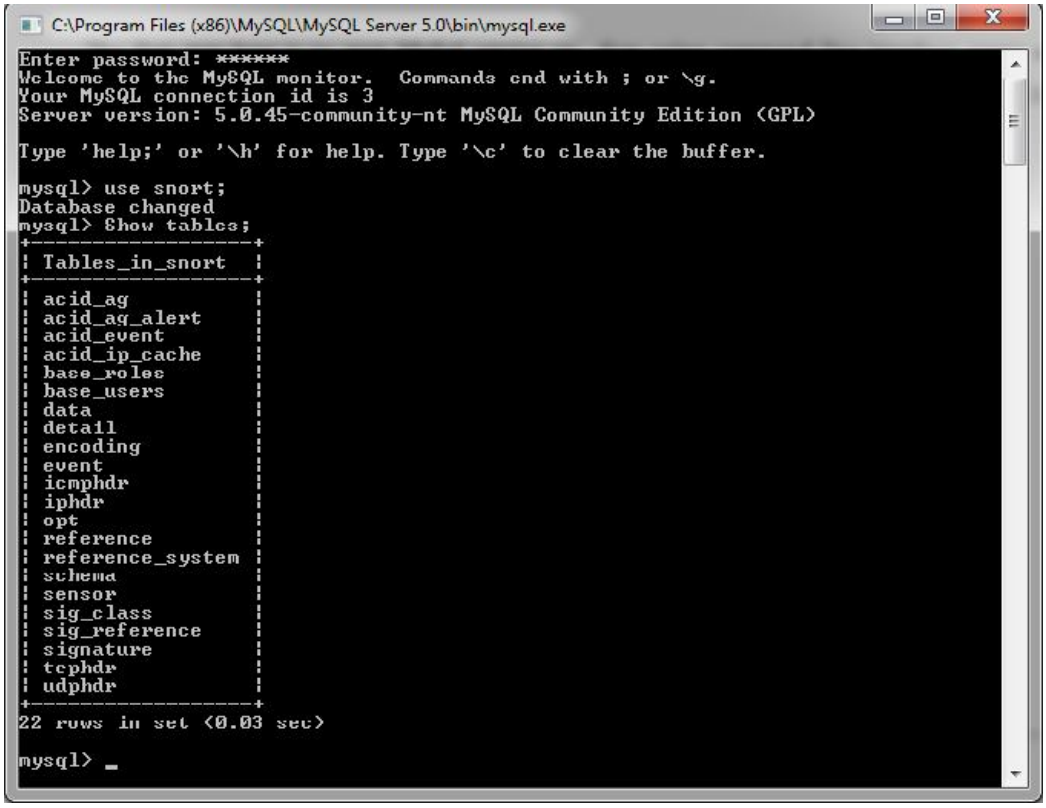

Figure 5.1: Tables in SNORT Database

Step 3: BASE Configuration

This open source tool needs some configuration to have a connection with the database and display the content in the database. The configuration starts by putting the BASE 1.4.5 folder into wwwroot file in order to display it on explore as a web page. Some editing is done on base_conf.php file. Sample configuration on this file is shown below.

- Set the base_urlpath to the url location that is the root of your BASE install.

```
$BASE_urlpath = '/base-1.4.5 ';
$DBlib_path = 'C:\PHP\adodb5';
$alert_dbname = 'SNORT';
$alert_host = 'localhost';
$alert_user = 'SNORT';
$alert_password = 'SNORT';
```

- output plug in configuration.

```
$alert_dbname   = 'SNORT';
$alert_host     = 'localhost';
$alert_port     = '3306';
$alert_user     = 'SNORT';
```

```
$alert_password = 'SNORT';
```

Figure 5.2 shows the snapshoot of how BASE displays the content of the database.

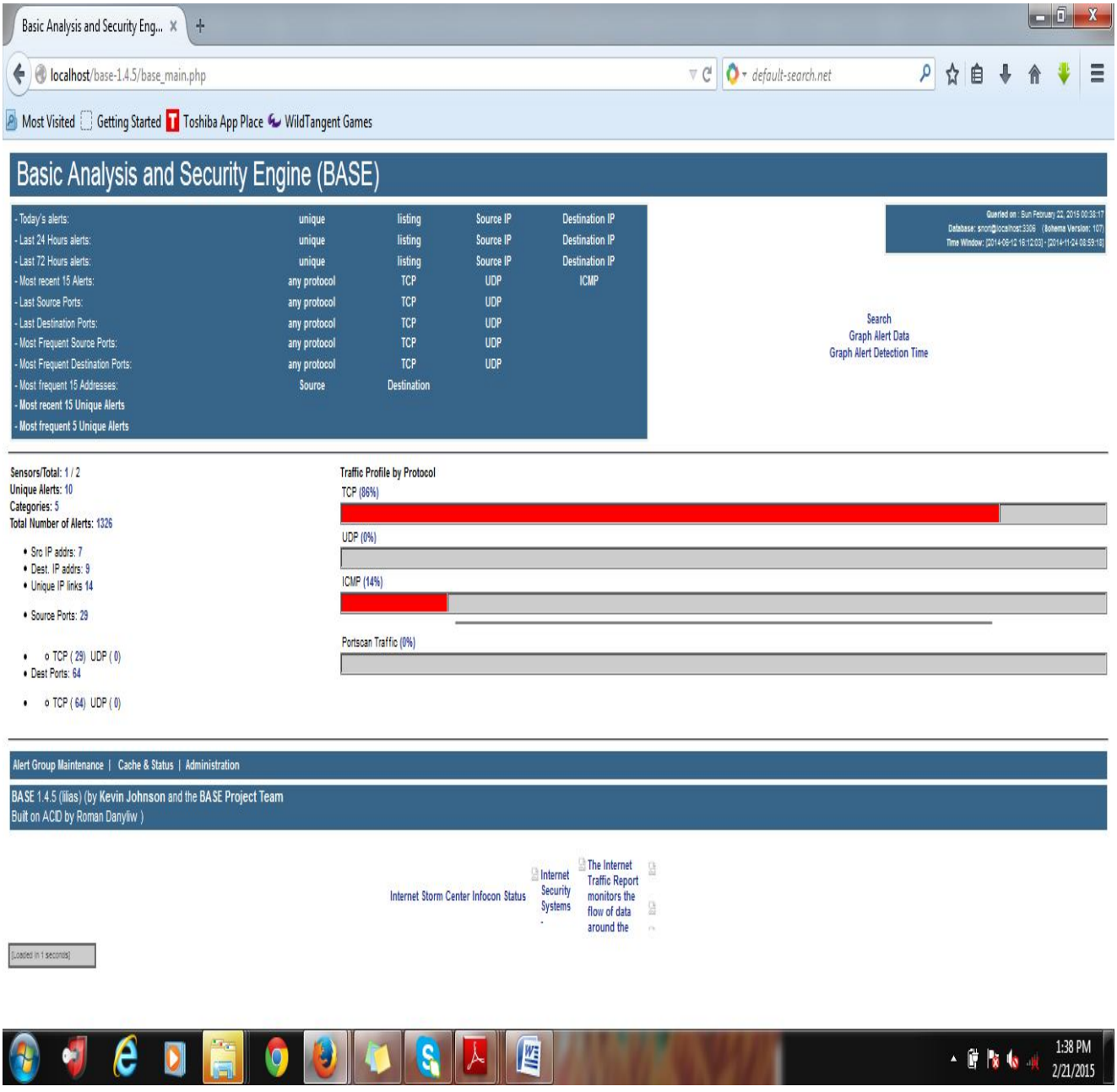

Figure 5.2: Display of SNORT Database using BASE

After finishing the configuration and installation of SNORT, we have deployed it on Addis Ababa University's network to take sample data and to be sure that it is functional for less than a half day in host based architecture. From this experiment, we have obtained 1326 alerts having 5 categories Table 5.1 shows these categories with their count.

Table 5.1: Categories of Alerts Obtained from SNORT

Categories	Count
Misc-activity	226
Attempted-user	8
Shellcode-detect	1024
Policy-violation	64
Attempted-admin	4

5.3.2 Anomaly Detection

Intrusion detection systems have an application in different organizations networks. Anomaly detection is one of the important components which will determine the detection rate of the intrusion detection system. Anomaly detection most of the time uses statistical method or machine learning methods. In this work, we used supervised machine learning method in which the system is trained with labeled simulated data then it classifies unlabeled data which is logged by SNORT.

As we have discussed previously this module is responsible for training normal activities in a network then identify those activities which is far from the normal. This component is the main part of this thesis work which is done using Java programming language, weka as library.

For training the system, we used simulated dataset that have 41 numbers of features. If all features are used as input to the neural network, it results in large size of the network and hence needs larger training time degrade the performance. So we used dimension reduction using PCA algorithm in weka tool. But before doing this dimension reduction preprocessing and feature transformation is an important issue so we transformed nominal values to numeric values. Then PCA returns the data with 21number of features. After looking the order of features which is provided by PCA we selected the first 9 basic features manually then comparing the mean square

errors gained from dataset having 21 and 9 features using weka tool. And the result was 0.1788 and 0.1744 respectively, so we selected the second features combination for training our system.

In this work, we built a classifying model using an algorithm artificial neural network; Multilayer Perceptron (MLP) which is supervised artificial neural network algorithm that have been widely used for data mining , also it has been found to be effective in intrusion detection systems in most researches. Once the training was over, the weight value is stored to be used in recall stage.

The next steps on this move is converting the data obtained from SNORT in Addis Ababa University's network to .arff format and try to classify it using the classifying model generated using the training dataset. Finally it will display the connection recognition between the training and the obtained data. The result of this system is to identify normal or anomaly. Figure 5.3 shows how our hybrid system trained with the new dataset with reduced dimension and then detect intrusive actions from the new dataset.

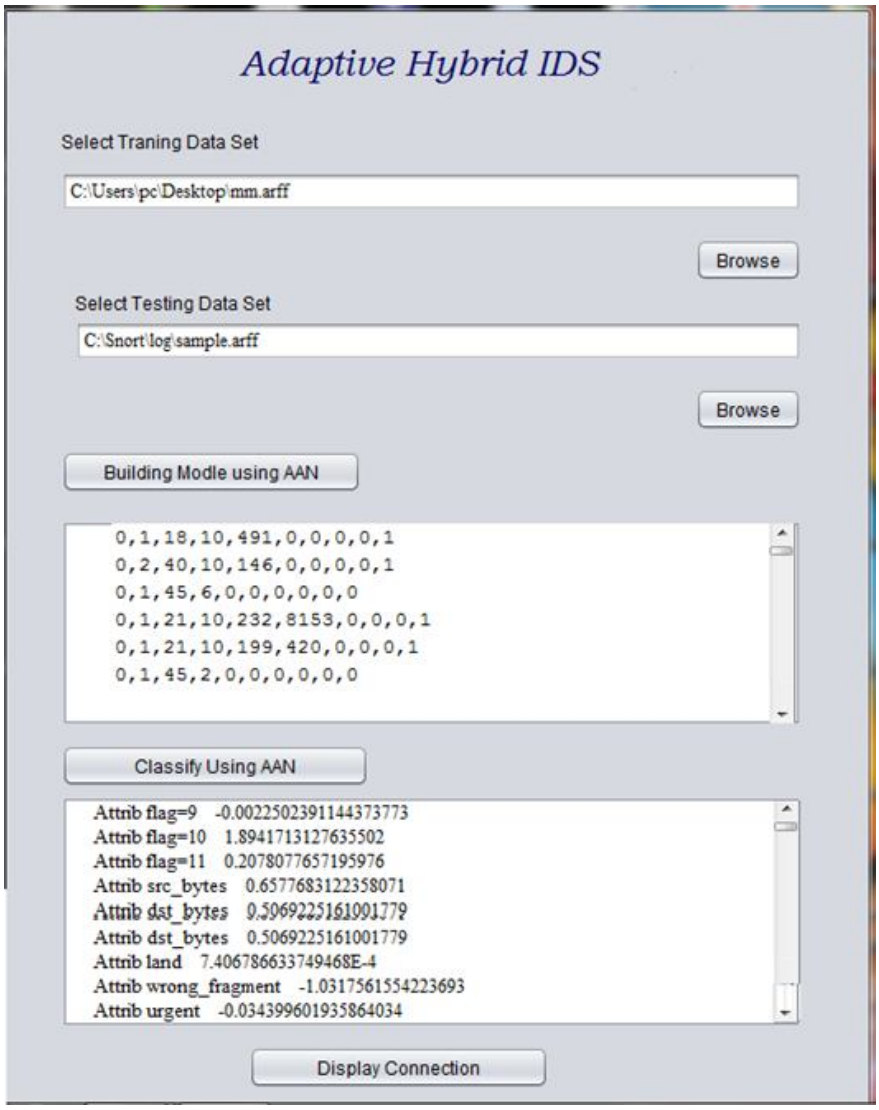

Figure 5.3: How Proposed Hybrid Intrusion Detection System Works

5.4 Experiments and Results

This Section covers the experiments performed during entire thesis. Experiments are broadly divided into two parts. The first part consists of the experiments is done on signature detection. The second part consists of experiments performed on conjuncture of anomaly and signature detection. This experiment, involves machine leaning tool since it is done using data mining techniques. The evaluation becomes easy using this tool.

5.4.1 Experimentation on Signature Detection

Run SNORT as an intrusion detection mode, SNORT stores the alerts/signatures generated by it in the Snort database. In this case we have used mixed dataset that is real time data and NSL-KDD dataset in which the total no of attacks and normal data are clearly defined. For this signature detection experiment we have selected 500 data's of NSL-KDD which is randomly selected and it consists of 320 normal data and 180 attacks. After that we go to alert and look for and count how many anomaly's are there also we look in to C:\Snort\log\ Snort.log file and count the number of data which belongs to the NSL-KDD dataset and assumed normal by SNORT. That is helpful to calculate TP, TN, FP and FP which is classified by SNORT from the given testing dataset.

5.4.2 Experimentation on Signature Detection + Anomaly Detection (Hybrid)

In this hybrid intrusion detection we used data mining technique. So to do the experiments we have use training NSL- KDD dataset which is used in the signature detection and snort.log for testing. For this experiment we have used Weka 3.7.9 latest Windows version is used and the default heap size is changed to 1024 for evaluating the data of NSL-KDD and snort.log file. In this experiment we also use Microsoft Excel for manual labeling the data.

5.5 Performance Evaluation

In this section, we measure the performance of the proposed hybrid intrusion detection sysem and compare the result with the signature based system. To conduct this evaluation, we consider the requirements of hybrid intrusion detection system. Therefore, the evaluation includes those measures like accuracy, performance, completeness and scalability.

Standard metrics for measuring the performance of IDS is evaluated by computing measures from the value in the confusion matrix shown in Table 5.2.

Table 5.2: Confusion Matrix

		Actual class	
		Negative Class (normal)	**Positive Class (Attack)**
Predicted Class	**Negative Class (normal)**	True Negative (TN)	False Negative FN)
	Positive Class (Attack)	False Positive (FP)	True Positive (TP)

5.5.1 Performance Evaluation: Accuracy

As mentioned earlier the effectiveness of hybrid intrusion detection system is measured in terms of accuracy in which it identifies how much do the IDSs classify the coming packet as normal and attack. The accuracy of the proposed system is calculated using Equation 5.1.

- Accuracy :

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (5.1)$$

For our case we first get the accuracy of signature based IDS first and we compared it with the proposed hybrid IDS. In signature based IDS we used 500 data as stated above and we got 126 attacks out of the given which means 54 are classified as falsely as normal. And also from the normal 320 data feed to SNORT 289 is predicted normal and the rest 31 is predicted as attack falsely. From this we got TP =289, FP =31TN =126 and FN= 54 so we can calculate it accuracy and we got 83%.

For anomaly detection we got TP =300, FP =20, TN =137 and FN= 43 and its accuracy is calculated using the given formula 87.4%. So from this we found that the accuracy of the proposed system is better than that of the single detection approach.

5.5.2 Performance Evaluation: Performance

The main objective of this performance evaluation is to identify whether it add noticeable overload or not to the IDS. In particular, the following measures will be used to assess the IDS's performance including accuracy:

- False Positive Rate (FPR) or False Alarm Rate (FAR) :

$$FPR = \frac{FP}{TN + FP} \quad (5.2)$$

- Precision:

$$P = \frac{TP}{TP + FP} \quad (5.3)$$

- Recall or True Positive Rate or Detection Rate (DR):

$$R = \frac{\text{TP}}{\text{TP + FN}} \quad (5.4)$$

- F1score :

$$F1 = 2\frac{R*P}{R+P} \quad (5.5)$$

The other issues which are useful to measure the performance of an IDS are training time and testing time in which training time is the time needed by the algorithm to build the model and testing time is the time needed by the classifier to classify new example using a given model. Even if these issues are dependent on data set size doing it fast enough is important to avoid the slowness or overload on the network. But the most commonly used are detection rate and false alarm rate.

From this we can calculate each for detection approaches. For Signature based FPR = 0.197, P = 0.903, R = 0.843 and F1score = 0.872. For the proposed approach we also calculate FPR = 0.127, P = 0.938, R = 0.875 and F1score = 0.905.

5.5.3 Performance Evaluation: Completeness

The aim of this completeness is to minimize false alarm rate. When the false alarm rate decreases we can assume that the IDS will starts examining both known and unknown attacks in a network. So for our case false alarm rate for the hybrid detection is less than the single detection technique so from this we can conclude that the proposed IDS is better detecting network attack and relatively it is complete.

5.5.4 Performance Evaluation: Scalability

Scalability is the other issue which responsibly identify whether the IDS works in large scale network or not. And we have test it using dataset having 11850 instances and we got accuracy of 92%. The proposed system easily accepts the coming data in .arff format and classify.

5.6 Discussion

Based on the results we have obtained in the previous Chapter we present here our perception of the result in relation to the research objectives stated in Chapter 1. As a reminder, the primary objective of this work was to design a hybrid intrusion detection system and that can increase the detection rate and minimize false alarm rate and this thesis work answers the questions:

- How can we minimize intruders?
- Which detection technique is the best to use as a candidate for enhancement?
- How can we increase the detection rate of signature based systems and make them detect unknown attacks?

After this research work and the experimentation we have obtained encouraging results as we have assumed at the beginning of this work. To begin with, we have to look for an open source intrusion detection system that is flexible and scalable.

In this thesis work, we have tried to integrate the advantage of signature based open source with the anomaly based system using detection algorithm.

On the other hand we have used a dataset for this research work training purpose that is selected because of the advantage over the other simulated network dataset over the Internet. We assume that it will increase the detection performance of our system and our result proves this expectation. We developed and tested a reinforcement learning approach to optimize anomaly detection systems. We propose an optimization scheme that incorporates prediction confidence with precision, recall, and F1 measure and accuracy metrics.

Compared to results using only signature detection and the proposed hybrid detection we obtained a promising result.

CHAPTER 6: Conclusion and Future Work

6.1 Conclusion

Intrusion detection has improved with age, but this improvement seems to be a continuous process as advancement in the technology opens the door with a loop-hole for intruders every time.

In this thesis work, we have tried to develop a hybrid intrusion detection system that tries to solve the shortcomings of IDS. This work mainly concentrates on three modules, i.e., signature based module, anomaly detection module and signature generation module. In which Signature-based module can only detect attacks that are known before whereas anomaly-based module is able to detect unknown attacks. Anomaly-based IDSs make it possible to detect attacks whose signatures are not included in rule files. The major concerns in this method are identifying the appropriate network features to characterize the network and build artificial neural network model and also the rate of false positives may increase sharply if the IDS is not trained sufficiently in the target network.

In the present work, we discussed the design and development of “Anomaly based intrusion Detection system” which is built on top of an existing open source signature based network IDS, called SNORT. This anomaly based component is trained using a simulated dataset NSL-KDD.

The thesis presented two techniques for detecting anomaly based intrusions. The first one is dimension reduction PCA which chooses the best field that can represent the network and we selected 9 out of 41. The second one is MLP neural network which is used to classify attack connection with normal.

The main contribution of this thesis work is to build an intrusion detection system that can examine both known and unknown attacks with high detection rate and low false alarm rate.

6.2 Further Work

While our work has produced some promising results, it is necessary to improve our system further to detect more known and unknown attacks. Our proposed system also needs further testing on a dataset with more variety of attacks.

Some potential future works that could be a continuation of our work is as follows:

- This work is done using SNORT as an intrusion detection system. This means it only gives alerts for the administrator. But on the future SNORT can be set as an intrusion prevention system and give action.
- The other issue that can be improved is adding artificial neural network to SNORT directly as preprocessor of SNORT in order to make every work on real time data rather than using simulated data for the training phase.
- Our future work will also be directed towards developing a more accurate model that can be used in real-time for detecting and classifying anomaly with minimum false alarms and less time. To do this combining number of detection algorithms and applying to anomaly detection module.
- The other issue to increase the detection rate in SNORT is to add other perfect pattern matching algorithm to SNORT first then add anomaly detection system to it.
- In this thesis work, even if we performed evaluation for the performance of its completeness and scalability; we could not observed that it is 100% complete and it needs more time. So it is also an open issue to develop more scalable and complete IDS.

References

[1] Tech-FAQ 2010. "*Network Attacks*", Accessed on March 14 201, <http://www.tech-faq.com/network-attacks.html>.

[2] Damien H. and Mathew W. (2003), "*Security for Internet Banking*: *A Framework.*" Logistics Information Management, 16(1): 64-73.

[3] Joseph S. and Rod A. (2003), "*Intrusion Detection: Methods and Systems*", Part II. Information Management and Computer Security 11(5):222-229.

[4] Ajith A., Crina G., and Yuehui C. (200, *"Cyber Security and the Evolution of Intrusion Detection Systems"*, Information Management and Computer Security, 9(4): 175-182.

[5] Sundaram A. (1996), "*An Introduction to Intrusion Detection, Crossroads*": The ACM Student Magazine, 2(4), acm.org/Crossroads.

[6] Wikipedia, the free encyclopedia, "*Intrusion Prevention System"*, Accessed on March 10 2014, < http://en.wikipedia.org/wiki/Intrusion_prevention_system>.

[7] Carl F., "*Intrusion Detection and Prevention"*, McGraw-Hill, Osborne Media, 2003.

[8] Karen S. and Peter M., "*Guide to Intrusion Detection and Prevention Systems"*, National Institute of Standards and Technology, Department of Commerce, USA, 2007.

[9] Christos D. and Aikaterini M., "*DDoS Attacks and Defense Mechanisms: Classification and State-of-the-art of Computer Networks"*, the International Journal of Computer and Telecommunications Networking, Vol. 44, No.5 , pp. 643 - 666, 2004.

[10] Crothers M., "*Implementing Intrusion Detection Systems"*, a Hands-On Guide for Securing the Network, USA, 2002.

[11] Tigabu D., *"Constructing Predictive Model for Network Intrusion Detection", Unpublished* Masters Thesis, Addis Ababa University, Addis Ababa, 2012.

[12] Endorf C., Eugene S., and Mellander C. (2004), *" Intrusion Detection":* MvGraw Hill.

[13] Tzeyoung M. (2009)., "*Information Assurance Tools Report – Intrusion Detection Systems."* Sixth Edition Information Assurance Tools Report(6ed., pp.93):IATAC

[14] Maiwald E., "*Network Security a Beginners Guide"*. Mc Graw Hill Professional, 2002.

[15] Brenton C., and Hunt C., "*Mastering Network Security (2nd Edition)"*, Sybex, Incorporated 2002.

[16] Jeams D., Scott D., "*Cisco Security Professional´s Guide to Secure Intrusion Detection Systems (IDS)"*, Syngress, 2003.

[17] Stewart J., *"CISSP Professional: Certified Information Systems Security Professional Study Guide"*, Sybex, Incorporated, 2005.

[18] Chris P. (2012), *" LogRhythm"*, <www.logrhythm.com>, last accessed, March 13, 2014.

[19] Jyothsna V., Ramaprasad V.V., and K Munivara P., *"A Review of Anomaly based Intrusion"*, International Journal of Computer Applications, Vol. 28, No. 7, pp. 26-35, August 2011.

[20] P Garcia T., J Diaz V., et.al, G Farnandez M., and Vazquez E., *"Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges"*, Journal of Computers & Security, Vol. 28, No. 1, pp. 18-28, February 2009.

[21] Rajdeep B., *"FuGeIDS: Fuzzy Genetic Paradigms in Intrusion Detection Systems"*, International Journal of Advanced Networking and Applications, Vol. 3, No. 6, pp. 1409-1415, 2012.

[22] Sang Jun H. and Sung Bae C., *"Evolutionary Neural Networks for Anomaly,"* IEEE Transaction on Systems and Cybernetics, Part B: CYBERNETICS, Vol. 36, No.3, pp. 34-46, 2006.

[23] Dinakara K.., *"A Master's Thesis on Anomaly Based Intrusion Detection"*, Computer Science and Engineering Indian Institute of Technology, Kharagpur -721302, India, May 2007.

[24] Beale J., Faster J. C., and Posluns J., *"SNORT 2.0 Intrusion Detection", S*yngress Publishing Inc, 2003.

[25] Roesch M., *"SNORT- Lightweight Intrusion Detection for Networks,"* in Proceedings of 13th USENIX Conference on System Administration, LISA '99, CA, USA, Nov. 1999, pp. 229-238.

[26] Rehman R. U., *"Intrusion Detection Systems with SNORT: Advanced IDS Techniques using SNORT, Apache, MySQL, PHP, and ACID"*, New Jersey: Prentice Hall Professional, 2003.

[27] Snort Team, *"Snort Users Manual"*, http://www.snort.org/assets/166/snort_manual.pdf, May 23, 2012.

[28] Sen M. and Soumya A., *"Performance Characterization & Improvement of Snort as an IDS"*, Technical Report, Princeton University, 2007.

[29] Anderson J., *"Computer Security Threat Monitoring and Surveillance"*, 1980.

[30] Suman R. and Vikram S**.,** *"SNORT: An Open Source Network Security Tool for Intrusion Detection in Campus Network Environment"***,** International Journal of Computer Technology and Electronics Engineering (IJCTEE), Vol. 2, Issue 1, 2010.

[31] Munish S. and Anuradha., *" Network Intrusion Detection System for Denial of Service Attack Based on Misuse Detection",* IJCEM International Journal of Computational Engineering & Management, Vol. 12, April 2011.

[32] Barbar D., Wu N. and Jajodia S., *"ADAM: Dectecting Intrusions by Data Mining"*, Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security. 2001.

[33] Silicon Defense., *"SPADE",* http://www.silicondefense.com/software/spice, Accessed on November 18, 2014.

[34] Hari O. and Tanmoy H., *"Statistical Techniques in Anomaly Intrusion Detection System",* International Journal of Advances in Engineering & Technology, Nov. 2012.

[35] Bhuyan M., Bhattacharyya D. and Kalita. J., *" Anomaly Based Intrusion Detection Using Incremental Approach: A Survey ",* University of Colorado, Colorado Springs, March, 2012.

[36] Laheeb M., *"Anomaly Network Intrusion Detection System Based on Distributed Time-Delay Neural Network (DTDNN)",* Journal of Engineering Science and Technology Vol. 5, No. 4 (2010), pp. 457 - 471.

[37] Xiang C. and Lim S., *"Design of Multiple-Level Hybrid Classifier for Intrusion Detection System. "*In 2005 IEEE Workshop on Machine Learning for Signal Processing, pp. 117–122, 2005.

[38] Kal H. and Min C., "*Hybrid Intrusion Detection with Weighted Signature Generation Over Anomalous Internet Episodes*", IEEE Transactions on Dependable and Secure Computing, 4(1):41–55, 2007.

[39] Nadiammai G. and Hemalatha M., "*Handling Intrusion Detection System using SNORT Based Statistical Algorithm and Semi-supervised Approach ",* Research Journal of Applied Sciences, Engineering and Technology 6(16): pp.2914-2922, 2013.

[40] Gómez J., Gil C., Padilla N., Baños R., and Jiménez C., *"Design of a SNORT-Based Hybrid Intrusion Detection System",* IWANN 2009, Part II, LNCS 5518, pp. 515–522, 2009.

[41] Roshni D. and Pradeep N., *"KNN based Classifier Systems for Intrusion Detection"*, International Journal of Advanced Computer Technology (IJACT) ISSN: pp.2319-7900, 2007.

[42] Divya, Surender L., *"HSNORT: A Hybrid Intrusion Detection System using Artificial Intelligence with SNORT"*, International Journal Computer Technology & Applications, Vol. 4 (3), pp.466-470, May-June 2013.

[43] Peng J., Feng C., and Rozenblit J., *"A Hybrid Intrusion Detection and Visualization System."* In Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems, page 3, 2006.

[44] Debar H., Dacier M., and Wespi A., *"Towards a Taxonomy of Intrusion Detection Systems"*, Computer Networks, 31:805{822, April 1999.

[45] Gregg S., *"Bandwidth, Packets Per Second, and Other Network Performance Metrics,"* Accessed on January 23, 2015 http://www.cisco.com /web /about /security/ intelligence/network_performance_metrics.html.

[46] Naveen N., "*An Analytical Approach For Real TimeIntrusion Detection Using Machine Learning Paradigm*", Unpublished Doctor of Philosophy Thesis, Department of Computer Science and Engineering Srm University, Kattankulathur- 603 203, January 2013.

Appendix 1

Feature description for NSL-KDD dataset

Feature Name	Description	Type
duration	Length (number of seconds) of the connection	continuous
protocol_type	Type of the protocol, e.g. tcp, udp, etc	discrete
service	Network service on the destination, e.g., http, telnet, etc	discrete
src_bytes	Number of data bytes from source to destination	continuous
dst_bytes	Number of data bytes from destination to source	continuous
flag	Normal or error status of the connection	discrete
land	1 if connection is from /to the same host/port; 0 otherwise	discrete
wrong_fragment	Number of "wrong" fragments	continuous
urgent	Number of urgent packts	continuous
Basic feature of individual TCP connections.		

Feature Name	Description	Type
hot	Number of "hot" indicators	continuous
num_failed_logins	Number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	Number of "compromised" conditions	continuous
root_shell	1 if root shell is obtained; 0otherwise	discrete
su_attempted	1 if "su root" command attempted ; 0 otherwise	discrete
num_root	Number of "root" accesses	continuous
num_file_creations	Number of file creation operations	continuous
num_shells	Number of shell prompts	continuous
num_access_files	Number of operations on access control files	continuous
num_outbound_cmds	Number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise	discrete
is_guest_login	1 if the login is a "guest" login; 0 otherwise	discrete
Content features within a connection suggested by domain knowledge.		

Feature Name	Description	Type
Count	Nuumber of connetions to the same-host as the current connection in the past two seconds	continuous
	Note: The folloing features refer to thse same-host connection.	
Serror_rate	% of connections that have "SYN" errors	continuous
Rerror_rate	% of connections that have "REJ" errors	continuous
Same_srv_rate	% of connections to the same service	continuous
Diff_srv_rate	% of connections to the diff erent services	continuous
Srv_count	Number of connetions to the same services as the current connection in the past two seconds	continuous
	Note: The folloing features refer to thse same service connections.	
Srv_serror_rate	% of connections that have "SYN" errors	continuous
Srv_rerror_rate	% of connections that have "REJ" errors	continuous
Srv_diff_host_rate	% of connections to different hosts	continuous
Traffic features computed using a two-second time window.		

Appendix 2

Attack Types of NSL- KDD Dataset

DOS	Back, Land, Neptune, Pod, Smurf , Teardrop
PROBE	Ipsweep, Nmap, PortSweep, Satan
R2L	Ftp_write, Guess_passwd, Imap, Multihop, Phf, Spy, Warezclient, Warezmaster
U2R	Buffer_overflow, Loadmodule, Perl, Rootkit

Appendix 3

Detail of Principal Component Analysis Algorithm

A mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The primary purpose of principal component analysis is to determine the utility in reducing the variables into linear combinations that explain the majority of variance from the dataset; this is essentially a technique in reducing dimensionality. The reduction works in the following way [51]:
Considering a set of observations $x_1,\ x_2. \ . \ .x_M$ are N x 1 vectors where each observation is represented by a vector of length N and can be represented as below matrix.

$$X_{M\times N} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{M1} & \cdots & x_{MN} \end{bmatrix} = [x_1,\ x_2, \ldots, x_M]$$

The mean value for each column is defined by the expected value as shown in Equation i. Once the mean value is subtracted from the data yields expression Equation ii.

$$\overline{X} = \frac{1}{M} \sum_{i=1}^{M} x'_{i,} \qquad \text{(i)}$$

$$Ø_i = X_i - \bar{X} \qquad \text{(ii)}$$

C is a correlation computed from matrix A=$[Ø_1\ Ø_2\ \ldots Ø_M]$.(N x M) matrix as shown in Equation iii. Sampled N x N covariance matrix characterizes how data is scattered [52].

$$A = [Ø1Ø2....ØM].C=\frac{1}{M} \sum_{n=1}^{M} Ø_n Ø_n^T = AAT \qquad \text{(iii)}$$

The eigen values of C: $\lambda_1 > \lambda_2 \ ... > \lambda_N$ and the eigen vectors of C: $u_1, u_2, ..., u_N$ have to be calculated. Because C is symmetric, $u_1, u_2, ..., u_N$ form a basis (i.e. any vector x or actually (x - $\bar{x}$)) can be written as a linear combination of the eigenvectors as shown in Equation iv.

$$X-\bar{X} = b1.u2+b2.u2+.....+bN.uN = \sum_{i=1}^{N} b_i . u_i \qquad \text{(iv)}$$

During the dimensionality reduction, only the terms corresponding to the K largest eigen values are taken into consideration as shown in Equation v [53]

$$\hat{x} - \bar{x} = \sum_{i=1}^{K} b_i . u_i, where\ K << N \qquad \text{(v)}$$

The representation of $\hat{x}$-$\bar{x}$ into the basis $u_1, u_2, ..., u_K$ is thus, $[b_1 b_2 \ldots b_k]^T$. The linear transformation $R^N => R_K$ by PCA that performs the dimensionality reduction is shown in Equation vi.

$$\begin{bmatrix} b1 \\ b2 \\ \vdots \\ bk \end{bmatrix} \quad \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_K^T \end{bmatrix} \quad (X\text{-}\bar{X}) = U^T . (X\text{-}\bar{X}) \qquad \text{(vi)}$$

If we consider the new variables (i.e. b_i's) to be uncorrelated. The covariance matrix for the b_i's can be represented in Equation vii. The covariance matrix represents only second order statistics among the vector values.

$$U^T.CU = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{bmatrix} \qquad \text{(vii)}$$

Suppose n be the dimensionality of the data. The covariance matrix is used to calculate a diagonal matrix, U^TCU. U^TCU is sorted and rearranged in the form of $\lambda_1 > \lambda_2 ... > \lambda_N$ so that the data exhibits greatest variance in y_1, the next largest variance in y_2 and so on, with minimum variance in y_n [54, 55].

Appendix 4

Detail of Artificial Neural Network Algorithm

This algorithm is inspired by human brain and it is a powerful data modeling tool that is able to capture and represent complex input/output relationships. Neural network acquire knowledge from input through leaning process. It has different layers of neurons i.e input layer, output layer and hidden layer which are the node between the two. Each layer has its own task, and does some calculations. The inputs to the input layer are set by the environment. This layer feeds information into the neural network. The hidden layers have no external connections; they only have connections with other layers in the network. The interaction between the hidden layers continues until some condition is satisfied. The outputs from the output layer are returned to the environment. Figure 1 shows the relationship between different layers of artificial neural network.

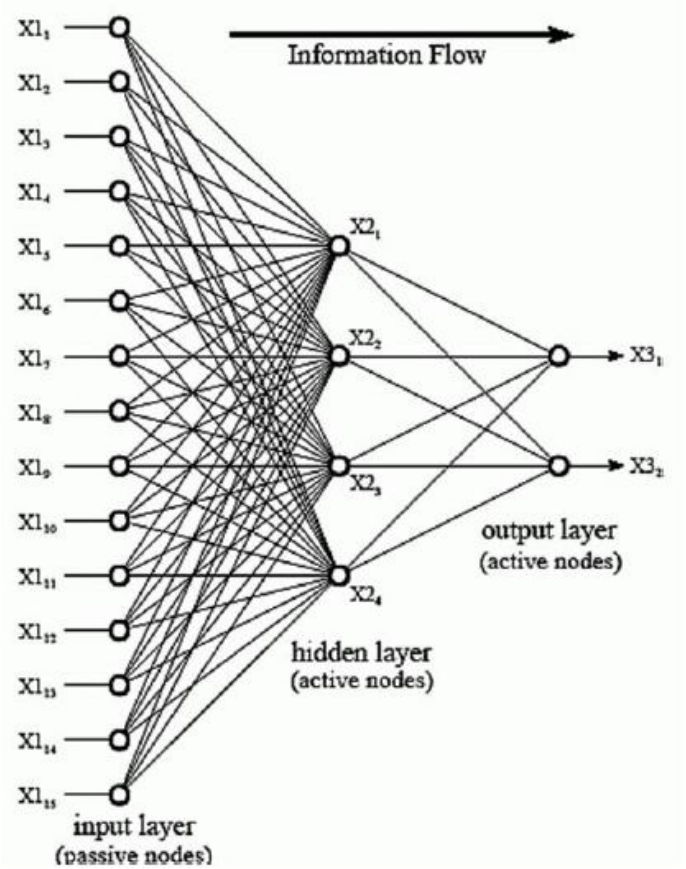

Figure 1: Layers of Neural Network

The hidden node and the output node are active nodes and input node is passive. The variables: $X1_1$,$X1_2$…$X1_{15}$ hold the data to be evaluated. Each value from the input layer is duplicated and sent to *all* of the hidden nodes. This is called a fully interconnected structure. The values entering a hidden node are multiplied by weights, a set of predetermined numbers stored in the program. The weighted inputs are then added to produce a single number. Before leaving the node, this number is passed through a nonlinear mathematical function called a *sigmoid*. The outputs from the hidden layer are represented by the variables: $X2_1$,$X2_2$,$X2_3$and $X2_4$. Just as before, each of these values is duplicated and applied to the next layer. The active nodes of the output layer combine and modify the data to produce the two output values of this network, $X3_1$ and$X3_2$.

Types of Artificial Neural Network

There are several types artificial neural network which can be classified according to:

I. Topology

a. Feed Forward: It produces quick response to the input because the connection between units does not form loop or cycle.

b. Feed Back or Recurrent: there are cycles (loops) connections between units. An input is presented each time. The ANN must iterate for a potentially long time before it produces a response. Training is usually difficult.

II. Kind of Data to Accept

a. Categorical Variable: which is used represents symbolic values like protocol type services and flag for our case and it must be encoded into numeric value.

b. Quantitative Variable: is used to represent numeric values such as duration, src_bytes and dst_bytes.

III. Learning Methods

a. Supervised Learning

It is also called classifier because it aims at building a predictive model (classifier) to classify the incoming patterns. This classifier should be trained with labeled patterns, so it can be able to classify the new unlabeled pattern later [56]. The well known architecture of supervised neural network is the Multi-Level Perceptron (MLP). In this thesis work we use supervised learning of neural network as shown in the Figure 2 [56].

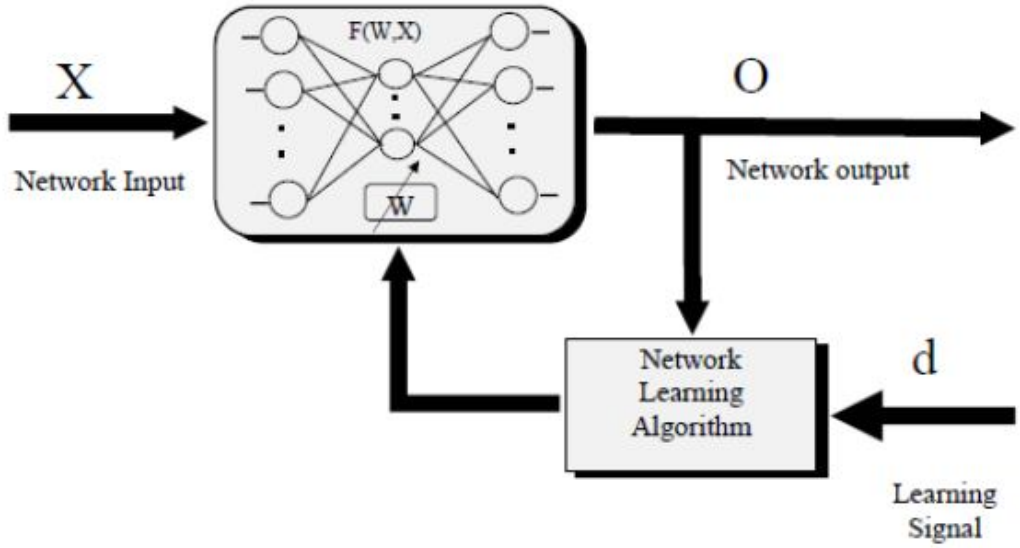

Figure 2: Supervised Learning

b. Unsupervised Learning

In the learning phase, the network learns without specifying desired output. Unsupervised learning is mostly used in applications that fall within the domain of estimation problems such as statistical modeling, compression, filtering, blind source separation and clustering [56]. The main aim this learning method is to find data organization as shown in Figure 3.

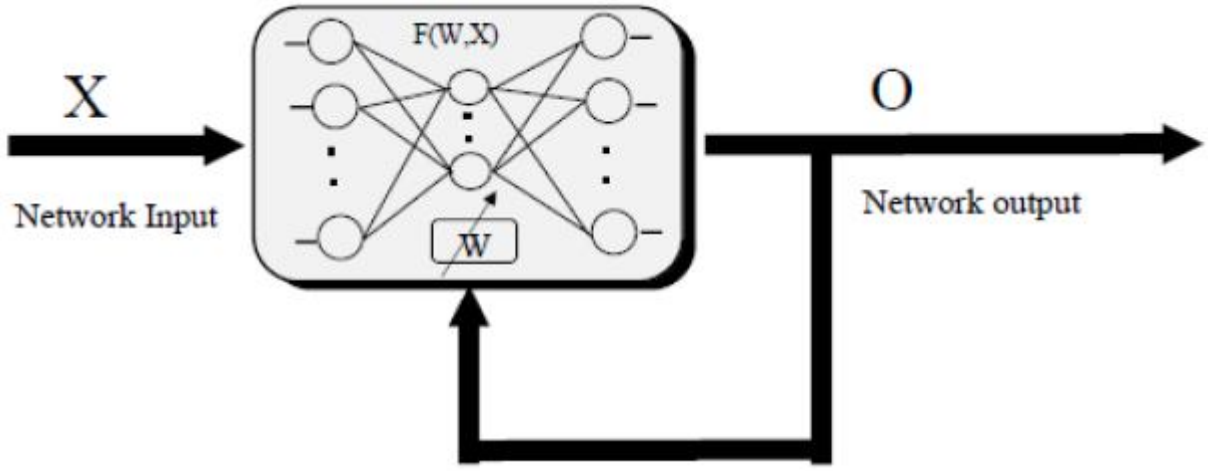

Figure 3: Unsupervised Learning

DECLARATION

I, the undersigned, declare that this research is my original work and has not been presented for degree in any other university, and that all sources of materials used for the research have been acknowledged.

Declared by:

Name: **Meheret Zewdu Wondimu**

Signature: _____

Date: 4/2/2015

Confirmed by Advisor:

Name: **Dr. Dejene Ejigu**

Signature: _____

Date: 4/2/2015

Place and date of submission: Addis Ababa University, April 2015