UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUTE OF SCIENCE AND TECHNOLOGY

MANAGEMENT AND ASSESSMENT SYSTEM FOR NETWORK ATTACKS BASED ON DATA MINING TECHNIQUES

MASTER THESIS

Ahmed Sami ABDULLAH

INSTITUTE OF SCIENCE AND TECHNOLOGY

INFORMATION TECHNOLOGY DEPARTMENT

DECEMBER 2017

UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUTE OF SCIENCE AND TECHNOLOGY

MANAGEMENT AND ASSESSMENT SYSTEM FOR NETWORK ATTACKS BASED ON DATA MINING TECHNIQUES

MASTER THESIS

Ahmed Sami ABDULLAH

1406050026

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

Thesis Supervisor: Assist. Prof. Dr. Shadi ALSHEHABI

Ahmed Sami Abdullah, having student number 1406050026 and enrolled in the Master Program at the Institute of Social Science at the University of Turkish Aeronautical Association, after meeting all of the required conditions contained in the related regulations, has successfully accomplished, in front of the jury, the presentation of the thesis prepared with the title of: "MANAGEMENT AND ASSESSMENT SYSTEM FOR NETWORK ATTACKS BASED ON DATA MINING TECHNIQUES".

Supervisor : **Assist. Prof. Dr. Shadi ALSHEHABI**
University of Turkish Aeronautical Association .....

Jury Members : **Prof. Dr. Abdül Kadir GÖRÜR**
Çankaya University .....

: **Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV**
University of Turkish Aeronautical Association .....

: **Assist. Prof. Dr. Shadi ALSHEHABI**
University of Turkish Aeronautical Association .....

Thesis Defense Date: 29.12.2017

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that the academic rules and ethical conduct have been adopted in obtaining and presenting the information and in accordance with the requirements of these rules and behavior I have referred to all the non-original results in this thesis.

29.12.2017

Ahmed Sami ABDULLAH

ACKNOWLEDGEMENTS

I am grateful to The Almighty GOD for helping me to complete this thesis.

I would like to thank my family for supporting me throughout my academic career. Without their moral support, interest and encouragement for my academic work, the completion of this effort would not have been possible.

I would like to express my sincere gratitude to Assist. Prof. Dr. SHADI ALSHEHABI for his sound advice and constructive criticisms in the evaluation of the results. He has helped me to the final stage of my thesis and willingly sharing his knowledge and ideas with me.

December 2017 Ahmed Sami ABDULLAH

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IDS : Intrusion Detection System
NIDS : Network-Based Intrusion Detection System
ID3 : Iterative Dichotomize 3
FAE : The feature alert entropy value
No of F : The number of features
NPV : New Priority Value
Info(D) : Entropy for each feature
SNMP : Simple Network Management Protocol
SIEM : Security information systems
CGI : Common Gateway Interface
SMB : Server Message Block

ABSTRACT

MANAGEMENT AND ASSESSMENT SYSTEM FOR NETWORK ATTACKS BASED ON DATA MINING TECHNIQUES

Abdullah Ahmed

MSc. Department of Information Technology

Supervisor: Assist. Prof. Dr. SHADI ALSHEHABI

December 2017, 53 pages

Most of the alerts generated by the Intrusion Detection System are false positive. The security analyst suffers from the difficulty of identifying attacks and taking action to address them because of a large number of false positive alerts. These alerts were not categorized depending on the degree of threat. These alerts must be addressed in order to determine their degree of threat and response time. Therefore, an urgent need to use data mining techniques to classify the degree of alert.

The reasons behind using this system are to classify IDS alerts by assessing them to examine the threat degree of IDS alert. This system contains five phases: Feature Selection Phase, Feature Entropy Phase, Alert Entropy Phase, Assessment Alert Phase, and Classification Alert Phase.

The Feature Selection Phase stores alerts of intrusion detection system, extracts the standard features and saves them in the Database file Microsoft Access. The Feature Entropy Phase determines the value of entropy for each feature of the alert. Alert Entropy Phase uses a new equation to compute the Alert Entropy which will help to be the input data for the next phase. Assessment Alert Phase assess alerts based on a new equation which calculates the assessment degree for the alert. The classification of alerts based on assessment degree of the threats (True or False). The Classification

Alert Phase computes the ratio of classification. The result of the system reduced the amount of false positive alerts by 96.70% using DARPA 1999 data set.

Keywords: The Intrusion Detection System (IDS), DARPA 1999 data set, ID3 Algorithm.

ÖZET

VERİ MADENCİLİĞİ TEKNİKLERİNE DAYANILARAK AĞ SALDIRILARI İÇİN YÖNETİM VE DEĞERLENDİRME SİSTEMİ

Abdullah, Ahmed
Yüksek Lisans, Bilgi Teknolojileri Bölümü
Tez Danışmanı: Yrd. Doç. Dr. SHADI ALSHEHABI
Aralık 2017, 53 sayfa

Saldırı Tespit Sistemi tarafından üretilen uyarıların çoğu yanlış pozitif özellik göstermektedir. Güvenlik analizcileri çok sayıda yanlış pozitif uyarı nedeniyle saldırıları tespit etme ve bunlar ile başa çıkmak için harekete geçme konusunda zorluklarla yüz yüze kalmaktadır. Bu uyarılar tehdit derecesine göre kategorize edilmemektedir. Ancak, saldırının ciddiyeti ve tepki sürelerini belirlemek için bu uyarıların ele alınması gerekmektedir. Bu nedenle, uyarı derecesini sınıflandırmak için veri madenciliği tekniklerini kullanmak için acil bir ihtiyaç ortaya çıkmıştır.

Bu sistemi kullanmanın ardındaki nedenler, Saldırı Tespit Sistemi (IDS) uyarılarını sınıflandırmak ve IDS uyarılarının tehdit derecesini incelemek amacıyla uyarıları değerlendirmektir. Bu sistem beş aşamadan oluşmaktadır: Özellik Seçme Aşaması, Özellik Entropisi Aşamasının, Uyarı Entropisi (Bilgi Yitimi) Aşaması, Değerlendirme Uyarısı Aşaması ve Sınıflandırma Uyarısı Aşaması.

Özellik Seçimi Aşaması, saldırı tespit sisteminin uyarılarının saklanması, standart özelliklerin çıkarılması ve bunların Microsoft Access Veri Tabanı dosyasına kaydedilmesi için bir teknik geliştirmektedir. Özellik Entropisi Aşamasının başlıca görevi, uyarının her bir özelliği için bilgi yitimi değerinin belirlenmesidir. Uyarı Entropisi Aşaması, Uyarı Entropisini hesaplamak amacıyla bir sonraki aşama için girdi verileri olmasına yardımcı olacak yeni bir denklem kullanarak değerlendirilmektedir. Değerlendirme Uyarısı Aşaması, uyarının değerlendirme derecesini hesaplayan yeni

bir eşitliğe dayanan değerlendirme uyarılarından sorumlu olacaktır. Değerlendirme derecesine göre, uyarıların tehdit derecelerini baz alarak (Doğru veya Yanlış) uyarıların sınıflandırılmasını otomatik hale gelecektir. Sınıflandırma Uyarı Aşaması, sınıflandırmanın oranını hesaplayacak. Elde edilen sistem sonuçlarına göre, DARPA 1999 veri seti kullanılarak yanlış pozitif uyarı miktarı % 96.70 oranında azaltılmıştır.

Anahtar Kelimeler: Saldırı Tespit Sistemi (IDS), DARPA 1999 veri seti, ID3 Algoritması

CHAPTER ONE

INTRODUCTION

1.1 Overview

In recent years, networks have heavily been used, and also the associated growth has brought on threats to networks by distributive several sorts of malicious programs that have an effect on the potency of networks, In particular data accessible through the network. Researchers have developed new techniques to explore and address these threats.

Computer intrusions became a more and more significant issue within the past few years. The Intrusion Detection System (IDS) is terribly interesting attentions; this attention is increasing day by day. As a result of the extensive information used in computer networks, maintaining the security of this information is very important, especially its use in many important areas such as; medical, military, scientific, space, and so on. The Intrusion Detection System is an integrated system designed to ensure the security of a computer network. It monitors packets to identify threats and gives warning when it detects any intrusive event and gives the analyst the opportunity to respond quickly to such behavior. The intrusion detection system generates an uncontrollable amount of Alerts, 99% of them are false positives [1].

The security analyst suffers from the difficulty of identifying attacks and taking action to address them because of a large number of false positive alerts. These alerts were not categorized depending on the degree of threat. These alerts must be addressed in order to determine their degree of threat and response time.

The Main objective of this research is to propose data mining algorithms to assessment alerts and give each alert a degree based on seriousness.

1.2 Intrusion Detection System (IDS)

The process of monitoring the system and activities across the network and detecting if any harmful processes exist, happens through the application software called the Intrusion Detection System (IDS). Given the great evolution and use of the Internet these days, there are concerns about how to preserve and transfer digital information safely .At present, specialists are using different techniques, algorithms and strategies to detect attacks aimed at obtaining important information.

Quite simply, Intrusion Detection system (IDS) is a software application whose primary function is to monitor the network and protect it from hackers. The revolution in the world of internet and the great development of Internet-based technologies have led to the growth of many fields of computer networks such as health, industry, education and security. It is worth mentioning that all LAN and WAN have advanced on top of these applications. These applications have made the Internet a fertile environment for infiltration and a major weakness of society. Hackers try to access the internal systems of the organization and collect information and try to create vulnerabilities such as software errors or malfunction in management or return the systems to their default configuration, Hackers imported viruses and worms to create system vulnerabilities such as trying to break the password and detect unencrypted text. Given what we mentioned we need an application to protect your network security from hackers and intruders. The process of protecting the private network from the public network is done using known firewall techniques, but we need to use an intrusion detection system that works in network-related activities such as preventing theft of the security card and others. [3]. Moreover, Firewalls do an awfully smart job of filtering the incoming traffic from the web to bypass the firewall [3]. Some types of intrusion cannot be detected by firewall programs, for example, attempting to connect to the Internet via a modem installed in a local area network of the organization [3].

An IDS is referred as thief alarm. As an example, the lock system within the house protects the house from stealing. However, if somebody breaks the lock system and tries to enter the house, it's the thief alarm that detects that the lock has been broken Associate in alerts the owner by raising the alarm.

1.3 Intrusion Detection System Types

1. HIDS “Host-based Intrusion Detection System”
2. NIDS “Network-based Intrusion Detection System”

HIDS looks for any intrusion signal into the system and uses host system logging and other information for analysis. HIDS processor is referred to as a sensor. Data can be obtained from any host-dependent sensor including object contents that are not reflected in the standard operating system auditing mechanisms, registration mechanisms, operating system logs, system logs. HIDS system is very confident in its audit trail. Intrusion Detection System takes the permissibility from information to detect any malicious type of interference that will not be visible in the progress of the abstraction stages. The main element of the intrusion detection system is the NIDS, which originated from the abnormal research conducted by Denning's on host-based systems. HIDS outperform the NIDS system by providing more relevant information. Host-based intrusion detection system has the ability to analyze attacks instead of ambiguous charges because it has the ability to know exactly what the hacker is doing, what he is using and what files he opens. He can tell if there is a serious hacking attempt or an attempt that is not serious [3].

1.4 Data Mining

The information industry contains a large number of data that cannot be used unless they are created again in helpful data. Extracting useful information from these massive data after doing their analysis is very necessary. In addition to extracting information from large data, we also have to do other operations like (Data Presentation, the Pattern Evaluation, the Data Mining, the Data Transformation, the Data Integration, Data Cleaning,). After all these processes have been completed we get data that can be used in scientific exploration, fraud detection and other fields [4].

the process of extract knowledge from data or extract information from a large set of data is known as data mining [4].

1.5 Data Mining Applications

There are many applications that use data mining technology such as:

1. Detection of Fraud

2. company Risk Management & Analysis
3. market Management and research

There are a lot of fields that benefit from data mining technique like Internet Web Surf-Aid, astrology. sports, science exploration, Client retention, and production control [4].

1.6 Problem Statement

Too many alerts are generated by the intrusion detection system, most are false positive and several are excessive or not real. One of the most difficult issues for a security analyst is the inability of the intrusion detection system to classify alerts to the degree of threat. Hence the need to use data mining technology to classify alerts based on the degree of threats.

1.7 Objectives

The main objective of this research is to propose a data mining algorithm (Enhancement ID3 algorithm) with new equations to assessment alerts and give each alert the threat degree.

1.8 Proposed System

The main object of the proposed system to assessment IDS alerts and classify it into true and false alerts. It is based on data mining techniques to compute the threat degree for each alert.

The proposed system can be divided into five phases; Feature Selection Phase, Feature Entropy Phase, Alert Entropy Phase, Assessment Alert Phase, Classification Alert Phase.

The **Feature Selection Phase** is to extract standard features and save them in a database file Microsoft Access.

Feature Entropy Phase is to compute the Entropy values for each feature of alert.

An Alert Entropy Phase is characterized by using a new equation to compute the Alert Entropy which will help to be the input data for the next phase.

An Assessment Alert Phase will be responsible for assessment alerts based on a new equation which calculates the assessment degree for the alert. According to the assessment, the degree will automate alerts classification according to their threats degree (True or False).

The **Classification Alert Phase** is the final phase that will compute the ratio of classification.

1.9 Related Work

They Support the formal procedures registry by proposing a multi-alert system to detect the misuse of the link element. When observing dynamic systems, these records are used as they provide a high level of declaration language and admission system used in substitutional areas. Formality was a direct reason for North American countries to reduce the overall number of alerts in the operating system as well as to improve overall diagnostic standards. Use logs to link the alert in the intrusion detection. High-level ad language has been provided which it is distinguished by its non-characterization of key input events. they suggested applying logs to link the intrusion alarm reveal. logs strong with the study of theory background. They offer a High-level ad language does not assume the nature of the foundation Input events. Practical and stable implementation of the recognition System exists. logs are already successfully used in many distinct Areas for monitoring dynamic systems where time information is pertinent. they have shown how logs can record some present intrusion Detection issues such as Weak alerts, false positives, and Increase in alerts, currently proposed logs use alarms as domain attributes only. They Plan to integrate a fact recognition system with an alert link Infrastructure, M2D2, in order to expand the attributes to other related Concepts, such as topology, which are more and more dynamic [5].

To monitor the repeated patterns of log files, they provided an algorithm rule set of the log file information set, for the purpose of abnormal log file lines, and log file profiles. For testing, the following specifications have been adopted for the: digital computer with operating system and the hardware (Redhat 8.0 UNIX system as software, 256MB of memory, with 5GHz Pentium4, for all group information tasks, vectors were used the outline of the size of five, 000 meters.). In order to identify outbound points, they intensified training for SLCT. The total number of passes during the trials was four. When the results are extracted it shows them that several sets of

huge log files were found in a short time and the base of their algorithm has modest memory needs [6].

The alarm data has been analyzed in the real world in this research. There are a lot of background noise types for an operational information system that falls into the combination of alerts. There may be major problems or types of attacks within this noise, despite the high alert frequency we have to monitor this noise using many methods. An important structure for the flow of alerts has been revealed through analysis and using both the mathematical and visual tools .In this research, three main points were highlighted: the causes of the noise, according to their opinion why the noise should be monitored despite a large number of alerts, using a reasonable load they suggested one way to monitor the noise. There are a lot of background noise types for an operational information system that falls into the combination of alerts. There may be major problems or types of attacks within this noise, despite the high alert frequency we have to monitor this noise using many methods. An important structure for the flow of alerts has been revealed through analysis and using both the mathematical and visual tools . They are currently implementing this process at the grade of the Intrusion Detection Manager. Certain alarm flows, especially where no pattern matching is needed The assembly can be done using the package example Head information, it can be possible to pay this kind of Processing towards sensors [7].

A new technique for extracting data into cluster production and group alarms has been developed, each extractor cluster generalized an alarm. In order to reduce future alarms, general alarms related to root causes are converted to filters. To deal with cluster alarm, the proposed algorithm uses the concepts of generalization and nearer neighboring. To calculate the distances between alert attribute values, this clustering algorithm uses a new measure. The significance and strength of this measure come from its reliance on background knowledge of network monitoring. The efficiency of this technique was proven to extract data by testing it with many databases and the reduction rate was 82% of total alerts. The security analyst uses a new technique to record alarms to identify root causes and reduce risks in the future. There are many directions for their future work, the most important of which is applying this method to different networks and using this proposed method to solve problems in other areas [8].

How to use data mining technology in security information systems (SIEM) and event management is discussed in this paper. Since the detection of abnormal patterns

needs to employ different types of correlation rules. One of the most important issues to be discussed in future work is how to improve the capacity of the system using other techniques of data mining techniques such as classification technology and clustering technology. Also, they are improving the techniques to minimize false positive alerts and to minimize a load of Central Processing Unit on the system while calculating data mining rules. In addition to all of the above, they are studying new units for an open source security information systems (SIEM) project [9].

For the purpose of monitoring all network traffic passing through the sector, they have developed a data mining technique based on the intrusion detection model in the network intrusion detection. To alert the administrator to expectations of suspicious or abnormal activity and detect the host to infiltrate by observing incoming and outgoing packets, a detector is installed. Based on the experiments, the proposed data extraction model is effective in detecting infiltration to a large degree. For their future work, they recommend the use of a comprehensive defense to eliminate network security concerns. In order to enhance operational effectiveness, the proposed model addresses the negative effects of its weaknesses [10].

This research presents a survey of current research related to false positive alarms. The concentrate of this research was to reduce alerts using data mining technology. In order to be a reference to relevant researches in this field, 30 studies related to this field were presented during the last decade. In this research many open issues were addressed. What came out of this study is due to the possibility of losing the real attack alerts and the low accuracy caused by some false positive alarm algorithms. A rigorous assessment approach should be provided to detect the accuracy of the proposed algorithms. This could be a field for future work [11].

They projected Intrusion detection technique exploitation 3 completely different strategies. These strategies area unit K-means bunch, fuzzy neuro models and C4.5 algorithms. they've projected a three-level framework for Intrusion Detection. In opening k-means bunch area unit wont to produce a variety of coaching subsets. In the second step, the completely different neuro-fuzzy model's area unit trained betting on the coaching subsets. Eventually, they perform classification exploitation C4.5 call tree algorithmic program and notice whether or not knowledge is abnormal or not. They took the assistance of k-means bunch technique to create massive, heterogeneous coaching knowledge set into a variety of homogenized subsets. As a result,

completeness of every set is reduced and consequently the detection performance is raised. when initial bunch within the projected technique, coaching is going to be given to fuzzy neural network and latter classification exploitation C4.5 call tree is performed. they need to be received the most effective results for C4.5 as compare to SVM [12].

They have Surveyed the current data mining techniques employed in detecting intrusions in a computer network. The main purpose of this research is to highlight the technique of data mining. Technique being employed in Intrusion Detection System. This paper concentrate on the usage of varied methodologies of data mining technique like clustering, classification and different data mining rules. The results recommend that classification methodology is being widely used for finding intruder-based issues and Support Vector Machine (SVM) remains widespread inside this arena, for the researchers. Similarly, in clump technique, statistical-based, chance, i.e. theorem clump, and its native square measure used for categorizing attack from a non-attack. Even though these methodologies score well in intrusion detection, the hybrid models introduced generate smart performances in lowering false alarms. The outcomes of this analysis paper recommend that there are an intensive utility and necessity of victimization data mining in Intrusion Detection System. The results conjointly recommend that classification is wide in use for finding intruder-based issues and Support Vector Machine (SVM) remains popular the researchers. Similarly, in clump technique, statistical-based, chance, i.e. theorem clump, and its native square measure used for categorizing attack from a non-attack. Lastly, it's conjointly rife that numerous hybrid models square measure introduced in police investigation intrusion. Hybrid models supported neural, agent or genetic algorithms square measure ideally used beside data mining techniques so as to find intrusion within the system [13].

This research presented different techniques for data mining and discussed the quality of each technology in the field of intrusion detection system (IDS) and in the security of the network. These technologies are equipped with an intrusion detection system that employs data mining technology to detect intrusion into the network. Many network systems perform poorly against various security threats such as network worms, large-scale network attacks, and so on. Raising consciousness of network security scenarios is an effective way to solve such problems. The final method was employed in the following manner, where network security events were recorded

based on time period and net circumference. Safety information was manipulated, the system's suffering and behavior were analyzed against attacks, universal network security was provided, the entire security scenario was assessed and long-term Network security trends were forecasted [14].

They performed an analytical comparison to perform the most important detection techniques in the intrusion detection system and wireless sensor networks, the research provided summary reports for all analyzes and comparisons of approaches that were technically represented. Attacks on the wireless sensor network were also discussed and classified according to several criteria. they prepare their dataset based on KDD Cup 1999 Data (KDD'99) data set for the aim of application and measure of the detection techniques, it summarized in five steps, first normalize their database, second by using the most relevant attributes of the classification process they identified the normal class with four types of attacks. In order to eliminate excessive attributes, they proposed to employ CfsSubsetEval with the BestFirst method as an attribute selection algorithm. Laboratory experiments have shown that the highest detection rates are using the random forest method, accompanied by a significant reduction in the rate of false alarms. In the end, a set of useful recommendations were developed in the future work, as well as to facilitate the researchers in the field of intrusion detection techniques used in the wireless sensor network. Based on the results extracted and the recommendations given, this research strongly recommends the use of data extraction technology in intrusion detection and attacks in the wireless sensor network. In addition to the advantages mentioned in the system, there are many problems facing it which need an analytical effort such as the use of machine learning in the problem of resource management of wireless sensor networks, the patterns of hierarchical clustering, choose and reprocessing a suitable data set. A lot of progress can be made to the intrusion detection system to meet the requirements of the wireless sensor network safely and responsibly by relying on smart algorithms such as compression of input data, narrowing of attributes and simplification of analysis and resolution procedures [15].

Analyzed multiple rules to seek out the best algorithm. Then they scale back the time quality of best rule by eliminating some options while not neutering the potency. It's been mentioned concerning completely different classification and agglomeration algorithms to style IDS. Eventually, the best algorithms Random Forest and sound unit

Scan are found. Then, the time quality of each the best algorithms, reduced by eliminating a number of options that have no impact on sight the attack. The potency of IDS is improved by applying some hybrid algorithms that is that the future work [16].

1.10 Thesis Organization

Chapter one displays general overviews about intrusion detection system (ids) and data mining.

Chapter two, includes introduction to snort and data mining algorithms and techniques, snort, The Snort Architecture, False Alerts, Data Mining Algorithms and Techniques, and Classification.

Chapter three, this chapter introduces the proposed system for this thesis namely Management and assessment system for network attacks based on data mining techniques, Overview for proposed system, Feature Selection Phase, Enhancement ID3 Algorithm, Classification Ratio Component.

Chapter four, this chapter discusses the evaluation of each phase of this system, and the implementation of each phase of proposed and the results that were obtained, Datasets, Evaluation proposed a system, Result in the analysis, and comparing with other approaches.

Chapter Five, in this chapter, section 5.2, it will give a conclusion about the whole system by discussing the objectives and the goals achieved, section 5.3, it will be discussed the future works.

CHAPTER TWO

SNORT AND DATA MINING ALGORITHMS AND TECHNIQUES

2.1 Introduction

Snort is a network intrusion detection system characterized as open source with the ability to perform real-time traffic analysis with packet logging over Internet Protocol (IP) networks. Using snort, we can perform the following operations: Detect and check multiple intrusions such as “overflow happened in the buffer, attempting to infiltrate the fingerprint operating system, (CGI), (SMB)”, performing protocol analysis, searching and matching the content. Snort developed very quickly and became one of the most important options for detecting intrusion. Snort can be configured with three major modes: “the sniffer, the packet logger, and the network intrusion detection”. The Sniffer Mode function is to read the network packets and then display them in a constant stream on the console. Packet logger mode function can be inferred from its name; it logs the packets on the disk. The Network Intrusion Detection Mode (NIDS) is the most complicated and configurable, it relies on Snort to analyze traffic over the network and it matches according to a set of rules previously defined by the user and then perform one of many known actions relative to what he saw. In addition to the set of signatures provided by Snort and enabling the registered user to download signatures of the VDB fire source, the network needs can be achieved by writing your signatures via Snort. This feature adds flexibility to Snort to meet the security needs of the network Additionally, there are many leading online gatherings in the field of intrusion analysis and incident response to share the latest snoring rules to detect the latest viruses. The behavior of the network pattern correspond for Snort has many immediate work able applications. The intrusion detection system is not responsible for cleaning machines infected with viruses, but identifies infected machines. We can reduce the incidence of modern worms spread through the scanning

of the Internet and attack the vulnerable hosts by writing the signature to address this scanning behavior. This identification capability is a very important and useful feature in large viral infections. same manner observation after assumed virus cleanup can aid to emphasize that the cleanup was effective [17].

2.2 Snort

The reason for preferring snort is because it is an intrusion detection system that relies on an open source network, in addition to its other rich features such as log useful packet and sniffing packet. Snort is used as an intrusion prevention system as it sends real-time alerts when it feels an intrusion attempt as it allows real-time alerts to be received by many means instead of sitting in your surveillance desk 24 hours a day.

One of the most important functions of Snort is to log packets and sniff packets, the advantages of intrusion detection come from its ability to match the contents of the packet against the set of intrusion rules. Snort is an intrusion detection system based on a lightweight network, and it works with a variety of operating systems. Snort usage has increased due to the increasing demand for free operating systems such as Linux, FreeBSD, OpenBSD, BSD-based OS NetBSD, and Snort can work with other commercial operating systems. It is possible to find his ports in Windows, IRIX, HP-UX, Mac OS X, Solaris, Snort can detect anomalies packets using the rules of a signature-based intrusion detection system. A rule is a group of requirements that generate an alert. The alert is issued when the intrusion detection rules match the contents of the packet and can send this alert to many places such as Simple Network Management Protocol (SNMP) trap, a database, a log file [17].

The goal behind Snort's design is basically to be a packet sniffer. In the late of 1998, (APE) is designed by Marty Roesch which it is Linux-only packet sniffer characterized by many Values features. However, Roesch greed in increasing the tasks performed by the sniffer These tasks are summarized as:

1. Employ the sniffer to works on several operating systems
2. Focus on The employ of a hexdump payload dump
3. Want to Display various network packets in the same manner

Roesch tried to develop the sniffer for personal use. and he also gave Snort the ability to filter the network and sniff the standpoint by writing snort as a libpcap application. Then, the only tcpdump was also compiled with libcap, so this gave the

system administrator another sniffer with which to work. In the last days of 1998, Snort has become possible for everyone to access it at Packet Storm (www.packetstormsecurity.com), Snort includes about 1,600 lines of code and two files. This was a couple of months when Snort's firstly beginning, and at this stage, the snort was employed only for the purpose of sniff the packet. The primary use of Snort by Roesch's for comprehensive observances the connection of the cable modem and for handle network applications which was coded.

Rules -based Analysis during the Snort society (as well-known as Snort's first signatures -based Analysis) It became a known feature at the beginning of 1999. This was only the beginning of Snort great revolution in the world of intrusion detection [17].

2.3 The Snort Architecture

There are four main components involved in the synthesis of the Snort architecture:

a. "The Sniffer "
b. "The preprocessor"
c. "The detection engine "
d. "The output "

Snort could be a "packet sniffer". First step, the snort selects the packets; the second step, is processed packets by the processor, the third step, the detection engine checks the packets against a series of rules. the architecture of snort illustrates in detail in the "Figure 2.1". the architecture of snort is almost similar to a large degree of the "mechanical sorter for coins". Where it works as follows: in the first step, it gathers whole coins, i.e., in other words, gather whole packets from the backbone of the network., in the second step, sends it in the shape of a waterfall to determine whether it is coins and how it will be rolled to the processor., in the third step, the detection engine will classify coins into four types: (the quarters, the nickels, the dimes, and the pennies). The final step is the most important step where the decision must be made towards the coins, and the decision will most often be rolled and stored. This is a logging and database storage [17].

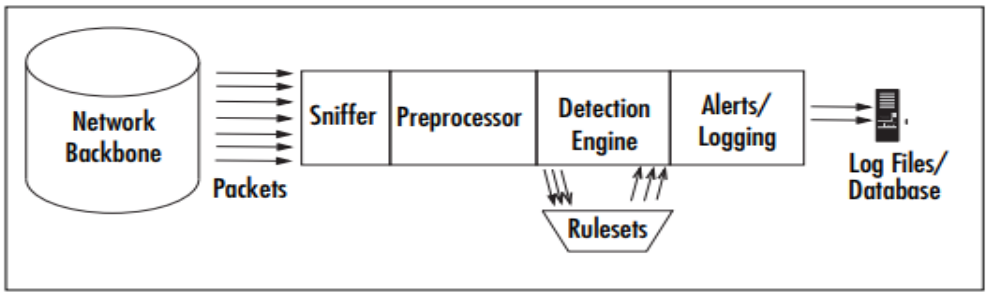

Figure 2.1: The architecture of the snort [17].

(the "preprocessor"," the detection engine", and "the alert") programs which written to adjust to Snort plugin API and need to be the section of the core Snort code and for the purpose of making reliable and easy modifications to the source code of the source, they were separated [17].

2.3.1 Packet Sniffer

We can express “packet sniffer” as a software or hardware tool for clicking on networks. It is similar in its work with wiretap telephone but packet sniffer is utilized for data networks. The application or device takes the command of the network sniffer to spy on the network traffic of data. on the internet, it is consisting of Internet Protocol Traffic, but there are other protocols that can be used with Local Area Network and old networks likeAppleTalk traffic and Internetwork Packet Exchange (IPX). Because Internet Protocol traffic consists of many important protocols (such as Transmission Control Protocol, User Schema Protocol, Web Message Control Protocol, Routing Protocols, Internet Protocol Security) ,Many domain specialists analyze various network protocols to interpret packets into something that can be read by humans. There are many uses for "Packet sniffers:”

a. troubleshooting after analyzing the network
b. the analysis of performance and benchmarking
c. Prevent hackers from sniffing your packages and convert it to something readable by encrypting network traffic.

We can use packet sniffers for permanently and evil, he named the appliance as a result of it will quit sniffing—it snorts. Snort is known for its ability to save packages for later review and processing as a” packet logger”. It is possible to observe Snort's ability to “packet-sniffing " by looking at Figure 2.2 [17].

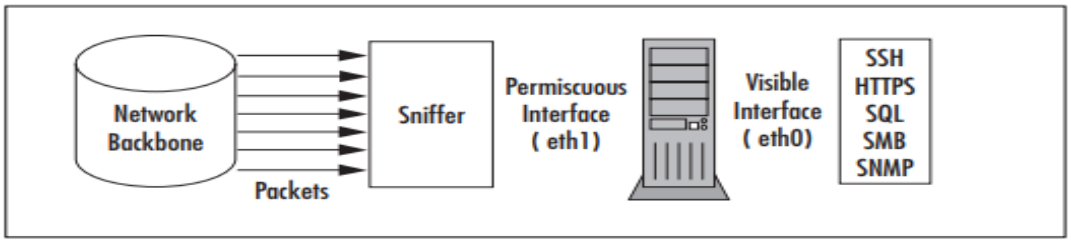

Figure 2.2: Ability to "packet-sniffing of snort.

2.3.2 Preprocessor

Our arrival at this point means that the coin's sorter obtained all the packets that can be obtained from the network and is now ready to send packets in the shape of a waterfall. then The processor checks the coin's sorter to see if they are actually coins or not and If they are coins, what shall they be sorts. Finally, "the detection engine" rolls for the coins. The processor will take the "raw packets "and check them versus plugins set such as port scanner plugins, remote procedure calls plugins, and Hypertext Transfer Protocol plugins. The plugins are realized by certain behaviors that packets may take. When this behavior is identified, it is sent to the detection engine, and for more detail, you can view the Figure 2.3. Some of the snort functions are "supports many sorts of preprocessors and their attendant plug-ins, covering many ordinarily used protocols similarly larger-view protocol problems like IP fragmentation handling, port scanning and flow control, and intensive inspection of richly featured protocols (like the HTTP inspect preprocessor handles)". The intrusion detection system can disable or enabled plugins as needed by the processor, generating alerts and specify resources for computational [17].

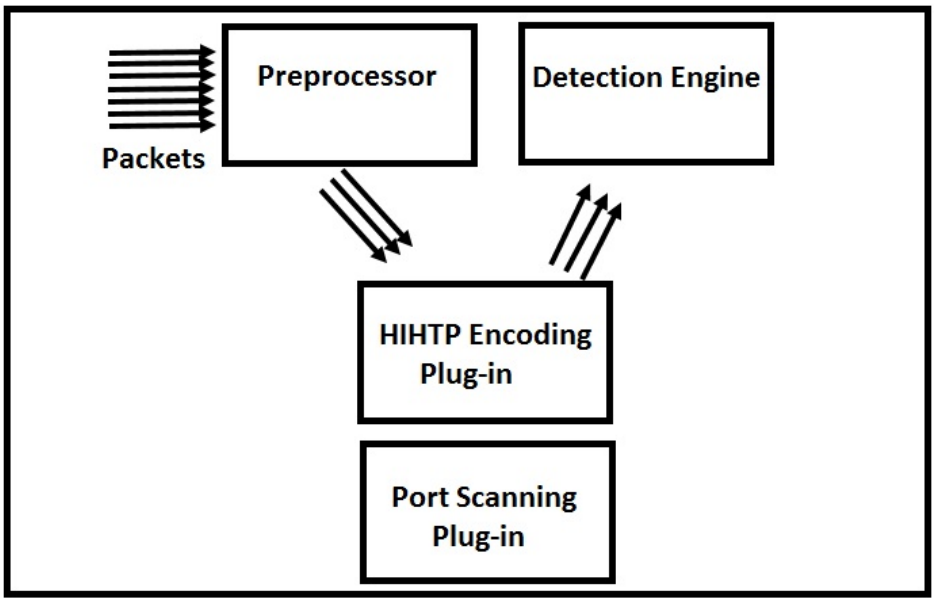

Figure 2.3: The processor of snort [17].

2.3.3 Detection Engine

After the handling of the packages by the processor turned to the detection engine, which in turn capture the data and plugins from the processor, then examine the data depending on a set of rules. If the data in this package matches the set of rules, it will have sent directly to the alert processor

The classification of the set of rules depends on three types which can be summarized as (access to various applications, buffer overflows, Trojan horses) and The update is organized. The set of rules consists of two parts

a. The “rule header” is essentially Determines what action to take (alert or log), network packet type (Transmission Control Protocol, User Datagram Protocol, Internet Control Message Protocol), root and destination for the internet protocol addresses, Plus ports.
b. The rule option whose main work is to match the packets with a set of rules is because it is the content of the packets.

The largest part of the new information for learning and understanding with Snort is given to the “detection engine” and its rules. Snort uses his own syntax with his set of rules which contains the header, the length, the content, type of protocol and another different element such as trash letter for knowing the rules overflow. You can fine-tune and customize the Snort function and manipulate rules in a way that serves your environment. “Figure 2.4” show the Detection Engine of Snort

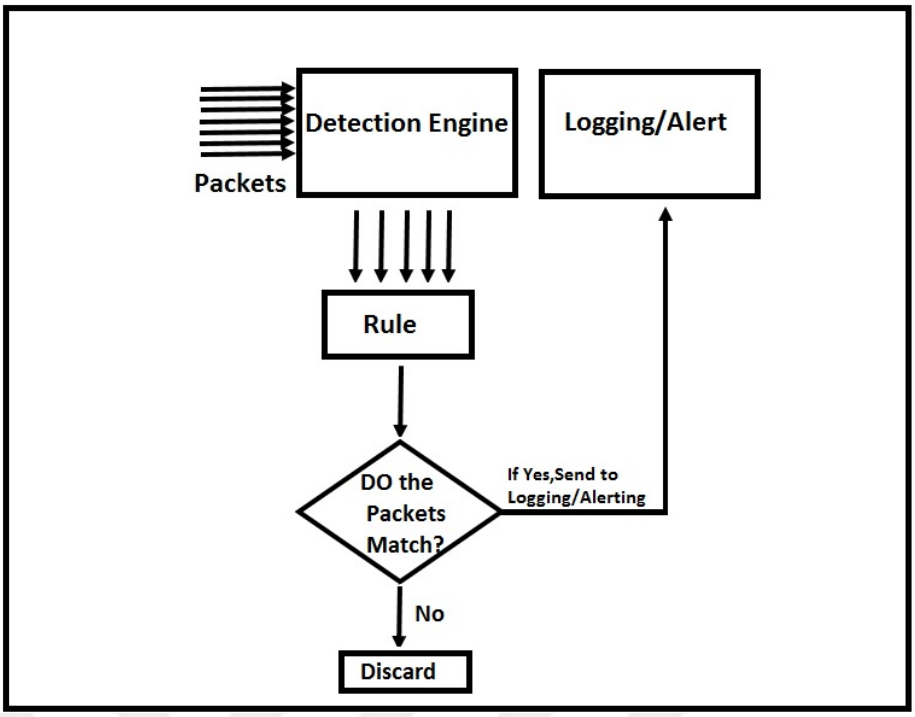

Figure 2.4: The detection engine of snort [17]

2.3.4 Logging /Alerting Component

The alert fires when the data matches the set of rules in the detection engine. the log file, in turn, receives the alert sent from the network connection, over Pop-up Windows (Server Message Block), or Simple Networks Management Protocol traps or UNIX sockets. It is possible to use databases to store alerts such as Structured Query Language (SQL) database like Postgres and MySQL. There are many additional tools that can be used with Snort as Web Servers, Personal Home Page Tools (PHP), and Perl's plugins to view logs over the Web interface. Logs storage is only two forms either in database or text files.

Alerts are sent to databases via network protocols such as Simple Network Management Protocol. Figure 2.5 shows that in details. It is possible to inform the system administrator about Snort alerts in real-time via e-mail based on Syslog tools such as Swatch.

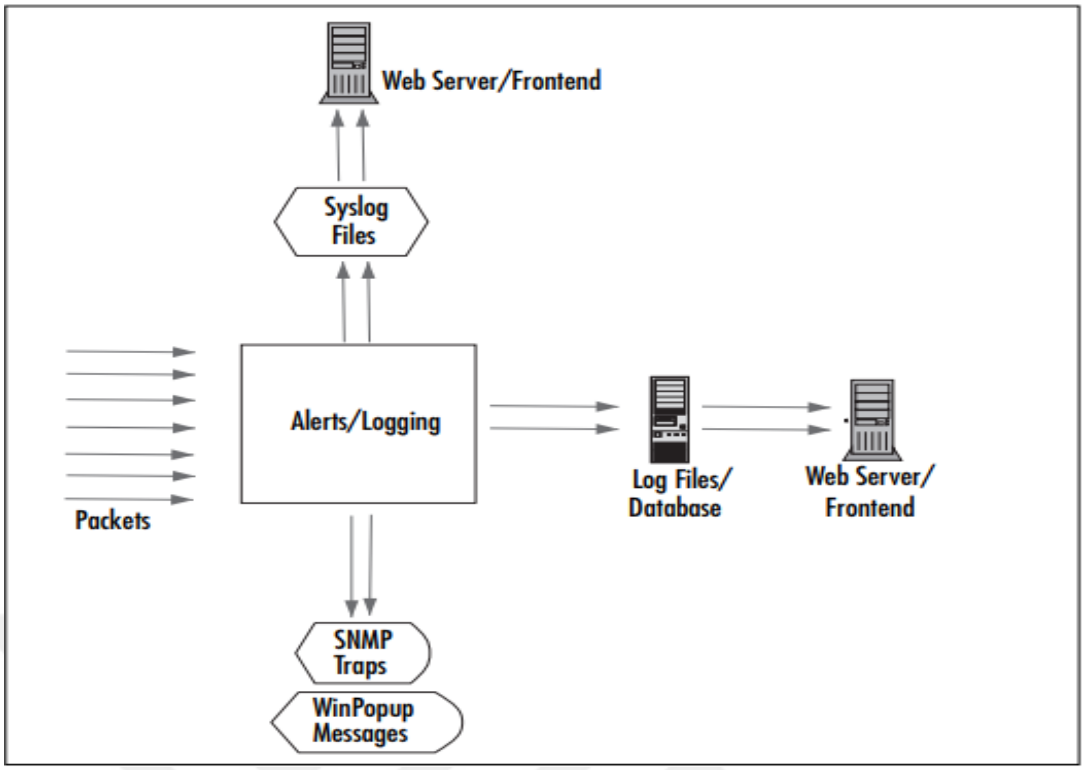

Figure 2.5: Alerting component of snort

2.4 False Alerts

The false positives are when snort gives an alarm that it must not. The default rules for Snort generate a lot of false alerts. Getting false alerts and acquittals is far better than missing out on data that may be a critical attack. The new installation of Snort would generate a lot of false alerts until you take your time to decide which ones are closely related to your network. The increase in the number of alerts to be monitored depends on the degree of network openness. At the same time, you can get false negatives. When someone tries to sneak into your Snort system and the Snort system unable to notice it. When another system administrator sends you a message via email that describes a suspicious system, this scenario usually happens with outdated laws or with new attacks that have not yet been written rules so you should make sure that your set of rules is the latest.

2.5 Data Mining Algorithms and Techniques

The process of knowledge discovery from databases requires the use of many techniques such as" Nearest Neighbor method, Decision Trees, Genetic Algorithm,

Association Rules, Neural Networks, Artificial Intelligence, Clustering, Regression, Classification, etc.". Figure 2-6 shows step data mining in the operation of knowledge detection.

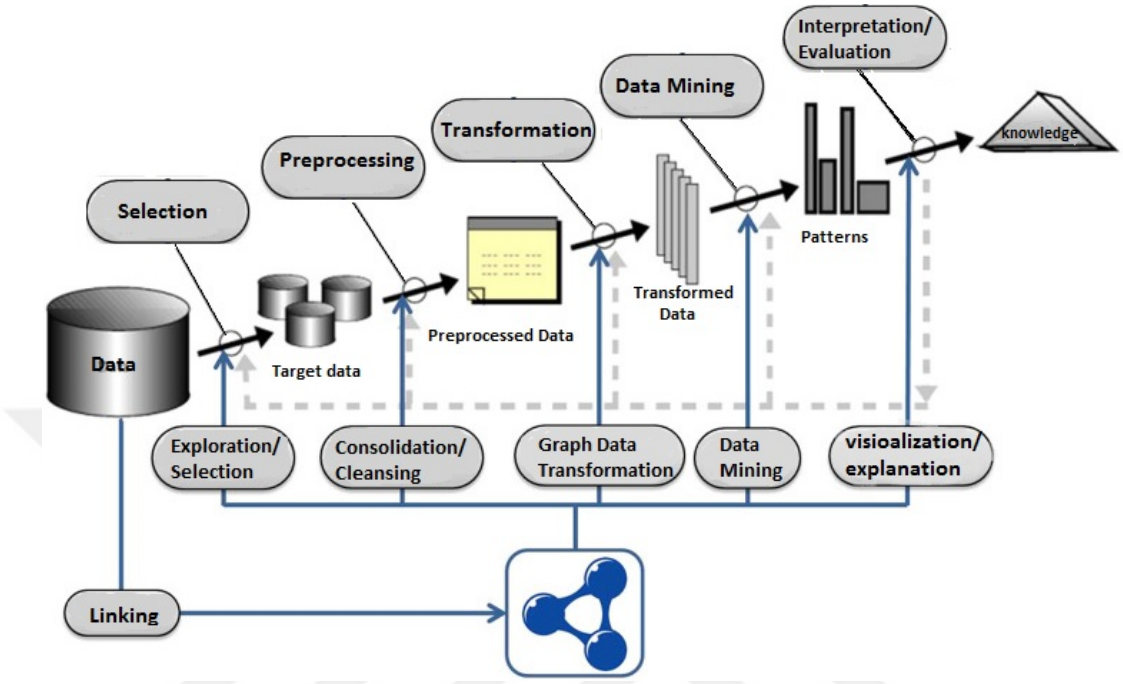

Figure 2.6: Step data mining in the operation of knowledge discovery

2.6 Classification

Classification is based on specific inputs to predict a particular outcome, to predict particular outcome, the algorithm is used a prediction or goal attribute which is a training package that contains a set of attributes and related results that the algorithm processes it to predict the outcome. The algorithm works on predicting the outcome by creating relationships between attributes. The prediction set is then given to the algorithm which it is a set of data not previously displayed and containing the same set of attributes except for the prediction attribute. Production of prediction is made after input analysis by the algorithm. The criterion for predictive reliability depends on the quality of the algorithm. We will take an illustrative example, in a medical database where the predictive trait is if the patient has or has no heart problem either as for the training group will have relevant patient information already registered. To illustrate sets of prediction and training see Tables 2.1 and Table 2.2.

Table 2.1: Training sets of medical database [18]

Age	Heart rate	Blood pressure	Heart problem
65	78	150/70	Yes
37	83	112/76	No
71	67	108/65	No

Table 2.2: Prediction sets for medical database [18]

Age	Heart rate	Blood pressure	Heart problem
43	98	147/89	?
65	58	106/63	?
84	77	150/65	?

There are many kinds of knowledge represented in literature, Expressions are made by employing classification for prediction rules. Usually, we use the form of IF-THEN rules, to express the rules of prediction, Where the part IF expresses the conditions and rules, in order to states the antecedent item, the part THEN predicts a proven value of predictions attribute. The conjunction feature in the classification is summarized as follows: Each case is separated by OR locate a smaller rule for holding relations between attributes. Produces prediction If any one of these small rules is satisfied. These small rules are formed using "AND", which has the advantage of Narrow relationships between the attributes. For example, based on table 2.1,2.2:

IF (Age>60 AND Blood pressure>140/70) OR (Age=65 AND Heart rate>70) THEN The Heart problem=yes

The quality of predictions is measured by the percentage of right predictions versus the total number of predictions. The rule must have a higher infection rate than the occurs of Predictability, For example if you employ the algorithm for rain detection in Seattle Although rainfall will continue to 80% of the time, predicting of rain all the time is the only way available to the algorithm to infect predicting rate 80% of the time. In such cases, all algorithms must achieve 80% rate. When the rule reaches the predicted inflect ratio 100%, so the ideal solution is achieved, but this is something that is hard to reach. approximation algorithms used classification by definition as just a resolve except for some very specific problems [18].

The ID3 algorithm is one of the most important approximation algorithms, it was Discovered by J. Ross Quinlan, and It was first published in 1975 in a book called

“Machine learning”. Multiple examples of different classes are developed to train the ID3 algorithm as a learning algorithm under supervision. The algorithm must be able to do its part in predicting a new item class after this training. First, all attributes must be defined and It must also be selected from a range of known (such as the country of citizenship) or continuous (such as temperature) values, then the ID3 algorithm identifies the salient attributes that can distinguish between classes. In order to determine the most important features, the ID3 algorithm depends on the statistical property of the entropy. The amount of information available in the attribute is measured by the entropy. The test of future cases depends on the way the decision tree is built. An attribute that has a large number of values is considered one of the disadvantages of the ID3 algorithm as in social security numbers. These features are not useful in predicting and they have a low entropy value. This problem can be overcome by using the C4 algorithm, which employs a statistical feature called information gain. The goodness of the attribute offered in the training sets in the output categories is measured using the information gain. The C4 algorithm overcomes the ID3 algorithm in minimizing the problems of low entropy values for attributes using information gain [18].

2.7 The ID3 Algorithm

Iterative DiChotomiser 3 (ID3) Invented by Quinlan at Sydney University and It was first published in 1975 in a book called “Machine learning**”.** The learning system algorithm is the concept behind the ID3 algorithm by using a set of trained examples C. The ID3 algorithm is taught under supervision to adopt the decision tree from a set of examples and then used to classify future samples. Depending on the information gain which resulting from the trained examples, the decision tree of the ID3 algorithm will build and then employ it to classify the data of test. For a classification-free of lost values, the" ID3 algorithm" employs nominal attributes [27].

2.7.1 Summary

A top-down- Greedy approach has been employed in order to build the decision tree of ID3 algorithms. Briefly, algorithm steps are:

1. Start with a training data set (S) which have attributes and classifications.
2. define the better attribute in the data set (S).

3. Split (S) into subsets that correspond to possible values of the best attribute.
4. Make a decision tree node that contains the best attribute.
5. Recursively generate a new decision tree node with the subsets of data that formed in step #3. Attributes can't be reused. If a subset of data agrees on the classification, choose that classification. If no more attributes to split on, choose the most common classification.

2.7.2 Pseudocode

The pseudocode supposes that the attributes are discrete and that the classifications are either yes or no. Pseudocode deals with conflicting training data by choosing the most common classification label at any time a possible conflict arises.

```
def id3(examples, classification attribute, attributes):
    create a root node for the tree
    if all examples are positive/yes:
        return root node with positive/yes label
    else if all examples are negative/no:
        return root node with negative/no label
    else if there are no attributes left:
        return root node with most popular
classification attribute label
    else:
        best_attribute = attribute from attributes that best
                         classifies examples
        assign best_attribute to root node
        for each value in best_attribute:
            add branch below root node for the value
            branch_examples = [examples that have that value
                               for best_attribute]
            if branch_examples is empty:
                add leaf node with most popular
                    classification_attribute label
    else:
        add subtree id3(branch_examples,
            classification_attribute,
            attributes - best_attribute)
```

When there is an attribute for the data to be divided, the algorithm should call itself recursively, with the original set of examples being split into groups depending on the set of available attributes and the value of the best attribute to be divided having the better attribute removed from it. For the reason of the algorithm recursive, the base cases: all examples having the same classification, no attributes being left, or no examples remaining, are tested first [27].

CHAPTER THREE

THE PROPOSED SYSTEM

3.1 Introduction

This chapter introduces our proposed system for this thesis. The reasons behind this system's proposal are to reduce false alerts in the intrusion detection system and to assess them in order to check the threat of intrusion detection alerts. At present, the use of the Internet and networks has become plentiful in many fields of knowledge.

The fact that nothing is perfect is also applicable to internet and networks. That is, though such a technology has added a lot, it yet has some negative effect. As a case in point is that such a technology has the ability to disseminate different types of threats and malicious programs. The latter is found to have a negative impact on the efficiency of data transmission; matters that motivated researchers to think of applicable ideas that help to get rid of such threats.

The intrusion detection system is designed to provide computer systems with additional protection from the thousands of alerts received by the system daily by creating a security analyzer to verify each alert based on the aggregation standard.

Many systems are designed (Fredrik Valeur, 2006 [19]; Maggi, et al. 2008; Elshoush and Osman, 2011 [20]), methods and techniques in this regard in an attempt to reduce the number of threats by compiling or linking to understand the threats mechanism on which are working. The number of threats and extraneous activities continues to expand with a marked increase in intelligence due to the large increase in the use of the Internet and networks.

Intrusion detection system generates a huge amount of alerts that analysts try to analyze to find the cause and find the relationship between them and other intrusion features. Many problems arise during the analysis process due to a large number of alarm data generated by the intrusion detection system. In order to assist analysts in the study of the alert databases, several studies have been raised.

3.2 Overview of Proposed System

Our proposal system contains five phases: Feature Selection Phase, Feature Entropy Phase, Alert Entropy Phase, Classification Alert Phase, and Assessment Alert Phase. Figure 3.1 depicts the Architecture Proposed System.

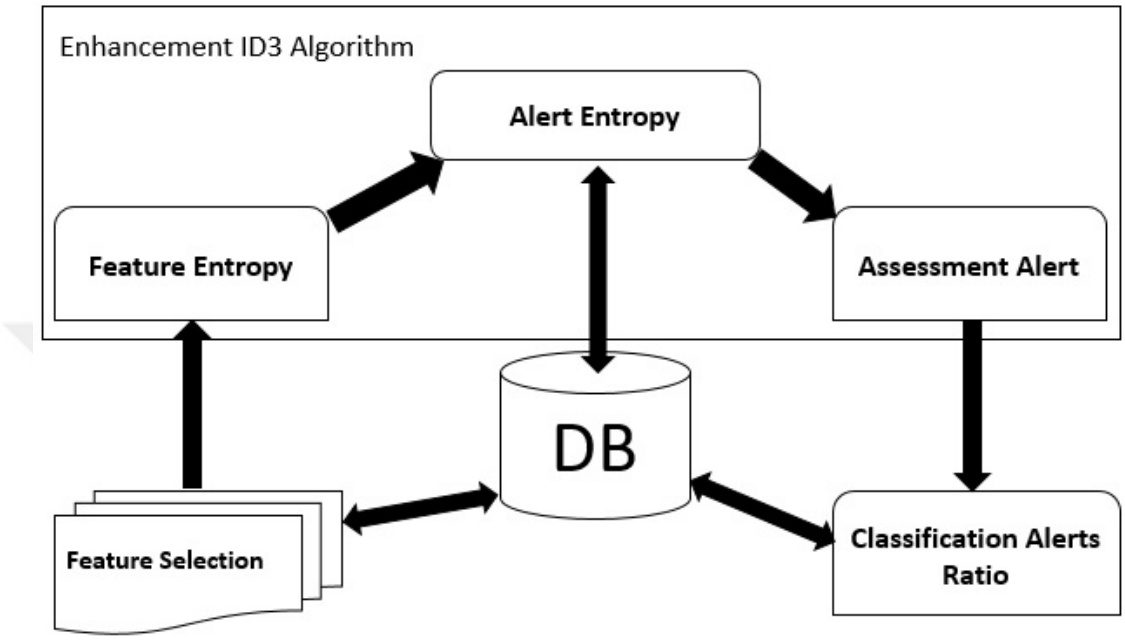

Figure 3.1: Architecture of our proposed system

The Feature Selection Phase is to work as reading Alerts from Database alerts which it is got it from DARPA dataset and converter to Access format file, this phase will make preprocess and will know what are the feature and categories of it.

The main function of **Feature Entropy Phase** is to compute the Entropy values for each feature.

The third Phase, **the Alert Entropy Phase**, is characterized by using the equation to compute the Alert Entropy which will help to be the input data for the next phase.

Assessment Alert Phase is the final phases, in this phase will be assessment alert based on the new equation and calculate the ration of classification.

The Classification Alert Phase will classify each alert (True or False) based on the alert entropy value of each alert in the previous phase (Alert Entropy Phase).

The next sections organized as follow: Section 3.3 will explain the Feature Selection Phase. Section 3.4 is dedicated to illustrating the ID3 Algorithm Phase.

Section 3.5 will illustrate the Alert Entropy Phase. Section 3.6 will explain the Classification Alert Phase. And finally, section 3.7 is about Assessment Alert Phase.

3.3 Feature Selection Phase

The phase is considered the most important in the system because the system depending on it to work. This phase is a preprocessing for data; it will be converted DARPA dataset from Excel file to Access file then selects the features and exclude another features, in first try all features and test by test selected the effective features in work to get the result better.

As well as add the new features named class, the new class features categories as (yes, no), yes for alert who has the value of priority is (1, 2) other value will be no, Figure 3.2 depicted the phase of Features Selection.

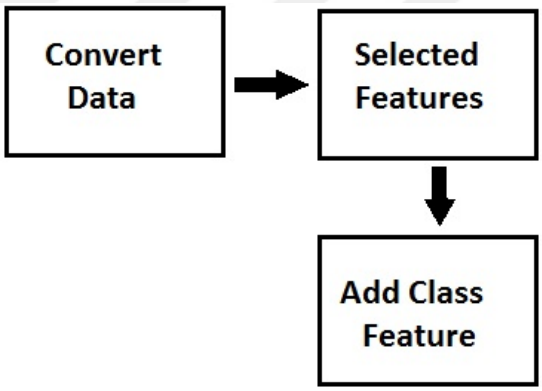

Figure 3.2: Feature selection phase

3.4 Enhancement ID3 Algorithm

The ID3 Algorithm one of the algorithm of data mining classification, it is using in this phase to calculate the Entropy value for each feature of the alert. Implementing the ID3 algorithm done, but selected just the Entropy value, the others exclude because it is not useful in this thesis when tried to use it and it is not given the good result.

The ID3 algorithm works for many iterations to calculate the Entropy value; maybe each feature obtains many values for each iteration. For this reason, will need to calculate the average entropy that will do it in the next phase.

The ID3 algorithm enhanced to can be working on the alert to be ready for assessment. There are many new equations created to the enhanced ID3 algorithm; it is depicted in the algorithm (1) in the steps (7-9) will be explained later.

Enhanced ID3 algorithm

1. Establish Classification Attribute (in Table R)
2. Compute Classification Entropy based on the following equations:

   $\mathrm{Info}(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$,

   $\mathrm{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \mathrm{Info}(D_j)$
3. For each attribute in R, calculate Information Gain using classification Attribute based on the following equation:

   $\mathrm{Gain}(A) = \mathrm{Info}(D) - \mathrm{Info}_A(D)$.
4. Select the maximum gain Attribute to be the next Node in the tree (starting from the Root node).
5. Remove Node Attribute, making reduced table RS.
6. Repeat steps $3-5$ till all attributes have been used, or the same classification value remains for all rows in the reduced table.
7. Compute Average Entropy for each Feature alert

   $\text{Average Feature Entropy} = \sum \text{AllEntropy Feature/No of entropy}$
8. Compute Alert Entropy

   $\text{Alert Entropy} = \sum \text{FAE/No of F}$

   Where: FAE is the feature alert entropy value

   No of F is The number of features
9. Calculate the assessment alert value:

   $\text{If Priority} = 1 \text{ then NPV} = 1$

   $\text{If Priority} = 2 \text{ then NPV} = 0.5$

   $\text{If Priority} = 3 \text{ then NPV} = 0.3$

   $\text{Assessment Alert} = (\text{Alert Entropy} + \text{NPV})/2$

   $\text{If Assessment Alert} \geq 0.55 \text{ then Alert} = \text{True Otherwise Alert} = \text{False}$

Table 3.1 shows the best value of NPV, so we took the first day of the fifth week as an example.

Table 3.1: The best value of NPV

Priority = 1	Priority = 2	Priority = 3	ratio
NPV=1	NPV=0.5	NPV=0.3	95.94%
NPV=2	NPV=1	NPV=0.5	91.08%
NPV=2	NPV=0.5	NPV=0.3	91.58%
NPV=3	NPV=2	NPV=1	74.75%

Algorithm (1): Enhanced ID3 algorithm

This phase consists of three components; Feature Entropy component, Alert Entropy Component, and Assessment Alert Component. In the next paragraphs will explain.

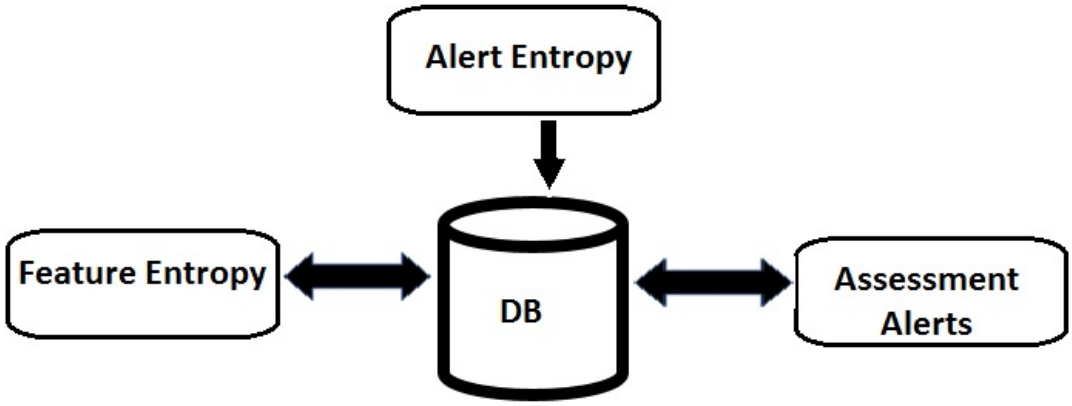

Figure 3.3: Enhanced ID3 algorithm architecture

3.4.1 Feature Entropy Component

This component is responsible for calculating the Entropy value for each feature. The Entropy value will compute based one step 2 from the Algorithm (1). Table 3.2 shows the example of feature entropy.

Table 3.2: Entropy values for features

symbols	entropy1	entropy2	entropy3	entropy4
ICMP Time-To-Live Exceeded in Transit	0			
Misc. activity	0		0	
Attempted Information Leak	0.08284	0.0914	0.81128	
Potential Corporate Privacy Violation	0			
Attempted User Privilege Gain	0			

From the above table noted the value of entropy differenced from feature to feature, because the ID3 calculate entropy value many times if find there is a value so in the next time also calculate entropy to find the best Entropy value.

The second step of this component computes the average of Entropy for each feature. This step based on the step (7) from the Algorithm (1)

$$\text{Average Feature Entropy} = \sum \text{AllEntropy Feature/No of entropy}$$

Table 3.3 shows an example for the average feature entropy after applying the above equation.

Table 3.3: Average feature entropy

Feature	average
Attempted Information Leak	0.3285067
Potential Corporate Privacy Violation	0
Attempted User Privilege Gain	0
Executable code was detected	0
Potentially Bad Traffic	0
3	0
2	0.1892475
1	0
ICMP	0
TCP	0.080045
64	0.320905
127	0.50028

3.4.2 Alert Entropy Component

The Alert Entropy component will be responsible for calculating the value of Alert entropy which is based on the Equation in step (8) from the Algorithm (1).

$$\text{Alert Entropy} = \sum \text{FAE}/\text{No of F}$$

3.4.3 The Assessment Alert Component

This is the last component in the Enhancement ID3 Algorithm; it will be responsible for assessing alerts which based on the step (9) in the Algorithm (1) as follows:

Calculate the assessment alert value:

$\text{If Priority} = 1 \text{ then NPV} = 1$

$\text{If Priority} = 2 \text{ then NPV} = 0.5$

$\text{If Priority} = 3 \text{ then NPV} = 0.3$

$\text{Assessment Alert} = (\text{Alert Entropy} + \text{NPV})/2$

$\text{If Assessment Alert} \geq 0.55 \text{ then Alert} = \text{True Otherwise Alert} = \text{False}$

Where:

NPV= New Priority Value

The Table (3.4) shows the results of executing the above step for assessment alerts.

Table 3.4: Assessment alerts

Alert_ID_<3499> Class-Value= 0.15 , Final-Class= False
Alert_ID_<3500> Class-Value= 0.15 , Final-Class= False
Alert_ID_<3501> Class-Value= 0.15 , Final-Class= False
Alert_ID_<3502> Class-Value= 0.15 , Final-Class= False
Alert_ID_<3503> Class-Value= 0.15 , Final-Class= False
Alert_ID_<3504> Class-Value= 0.15 , Final-Class= False
Alert_ID_<3505> Class-Value= 0.3270567 , Final-Class= False
Alert_ID_<3506> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3507> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3508> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3509> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3510> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3511> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3512> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3513> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3514> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3515> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3516> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3517> Class-Value= 0.1729218 , Final-Class= False
Alert_ID_<3518> Class-Value= 0.5528868 , Final-Class= True
Alert_ID_<3519> Class-Value= 0.5528868 , Final-Class= True
Alert_ID_<3520> Class-Value= 0.5528868 , Final-Class= True

3.5 Classification Ratio Component

The Classification ratio component it is the last component of the proposed system. It is performing to compute the ratio value based on the following equation:

$$\text{Ratio} = \text{False alert}/\text{Totel alert}$$

Example1: If we have the total input numbers of alerts equal (3811), the number of True alerts equal (167), and the number of False alerts equal (3644) then:

$$\text{Ratio} = 3644/3811$$

$$\text{Ratio} = 0.956$$

CHAPTER FOUR

EVALUATION AND RESULT DISSECTION

4.1 Introduction

As it mentioned previously in Chapter three for our proposed system (Management and assessment system for network attacks based on data mining techniques.), it was built mainly on five phases: Feature Selection Phase, Feature Entropy Phase, Alert Entropy Phase, Assessment Alert Phase, and Classification Alert Phase. In this chapter will discuss the evaluation of each phase of this system, and the implementation of each phase of proposed and the results that were obtained.

This chapter will be organized as follow: Section 4.1 Introduction, section 4.2 describes datasets which had been used in the evaluation process, section 4.3 the evaluation metrics of the proposed system and Degree of Threat intrusion detection system Alert Processes will be described, section 4.4 discusses the result analysis of the proposed system with the others systems.

4.2 Datasets

To check the validity of the proposed framework, our checks were evaluated based on the DARPA 1999 data set; a standard data set known by all authors in this field. We used the dataset for the purpose of compared it with that of another author within the same area of the same data set.

The ‘KDDCUP99 Data’ (Irvine 1999) are the data sets, which were issued for use in the KDDCUP ’99 Classifier-Learning Competition. These sets of training and test data were made available by Stolfo and Lee (http:// kdd.ics.uci.edu/ databases/kddcup99/task.htm. 1999) and consisted of a preprocessed version of the 1998 DARPA Evaluation Data. This team’s IDS had performed particularly well in

the Intrusion-Detection Evaluation Program of that year, using data mining even as a 'pre-processing' stage to extract characteristic intrusion features from raw TCP/IP audit data. The original raw training data were about four gigabytes of compressed binary tcpdump data obtained from the first seven weeks of network traffic at MIT. This was preprocessed with the feature-construction framework MADAM ID (Mining Audit data for automated models for Intrusion Detection) to produce about five-million connection records. A connection is defined to be a sequence of TCP packets starting and ending at some well-defined times, between which data flow to and fro from a source IP address to a destination IP address, under some well-defined protocol. Each connection is labelled as either 'normal' or with the name of its specific attack type. A connection record consists of about 100 bytes. Ten percent of the complementary two-weeks of the test data were, likewise, pre-processed to yield a further less than half-a million connection records. For the information of contestants, it was stressed that these test data were not from the same probability distribution as the training data, and that they included specific attack types which are not found in the training data. The full amount of labeled test data with some two million records was not included in this data set.

In the KDDCUP99 Data, the initial features extracted for a connection record (Eskin 2002; Lee 1994-1999) include the basic features of an individual TCP connection, such as: its duration, protocol type, number of bytes transferred and the flag indicating the normal or error status of the connection. These 'intrinsic' features provide information for general network-traffic analysis purposes. Since most DOS and Probe attacks involve sending a lot of connections to the same host(s) at the same time, they can have frequent sequential patterns, which are different to the normal traffic. For these patterns, a "same host" feature examines all other connections in the previous 2 seconds, which had the same destination as the current connection. Similarly, a "same service" feature examines all other connections in the previous 2 seconds, which had the same service as the current connection. These temporal and statistical characteristics are referred to as the "time-based" traffic features. There are several Probe attacks which use a much longer interval than 2 seconds (for example, one minute) when scanning the hosts or ports. For these, a mirror set of "host-based" traffic features were constructed based on a 'connection window' of 100 connections: The R2L and U2R attacks are embedded in the data portions of the TCP packets and

it may involve only a single connection. To detect these, ‘connection’ features individual connections were constructed using domain knowledge. These features suggest whether the data contains suspicious behavior, such as: a number of failed logins successfully logged in or not, whether logged in as root, whether a root shell is obtained, etc. In total, there are 42 features (including the attack type) in each connection record, with most of them taking on continuous values

4.2.1 Aggregation Dataset

The aggregation dataset applied to assess the performance of the Reduction of intrusion detection system Alert Processes System. It is created from DARPA dataset according to IDS Snort file. in the following sections will explain the datasets utilized as follows:

A- Aggregation DARPA dataset

To get alerts that triggered as a result of the detection of threats or suspicious traffic within the network was used the open source IDS Snort, the system depends on the rules prepared in advance for this purpose. And it allows the system user to update or add new rules.

The snort log file saves all alerts caused by intrusion detection system Snort. every alarm in the snort file is symbolized by a 24 feature. The intrusion detection system provides two types of alerts, namely full modes alerts and fast mode alerts. The concentrate is on full mode alerts because they contain all the features.

4.3 Evaluation Proposed System

To evaluate the proposed system, which was presented in chapter three, it was implemented. In the evaluation of the system, and because of this proposed system is based mainly on five phases; Feature Selection Phase, Feature Entropy Phase, Alert Entropy Phase, Classification Alert Phase, and Assessment Alert Phase.

The evaluation of the overall system will depend on their evaluation. In subsequent paragraphs will be addressed to evaluate the system.

To evaluate this system, there are three metrics are used, Multiple feature types, Ease of use, Multiple feature use.

A- Multiple feature types

The alert in the IDS has been many features. These features consist of different types of data such as categorical features, time features, numerical features and soon. The aggregation system proposed is supporting all the several of feature's types with high effectiveness, where the proposed system was not affected by this diversity in the data features of alerts and deals with all the flexibility and accuracy. The Table (4.1) shows the several of feature's data types.

Table 4.1: The several of feature's data types

No.	Feature name	Type data	Example
1	IP addresses	Categorical	171.113.80.124
2	Time stamp	Time	01:55.3
3	Flags	Numerical	TTL:255, Priority: 1
4	Classification	Text	Misc activity

B- Ease of use

Simply interface system design is one importance of the priorities of the system designer. To take into account that the user system is not an expert, but need a simple knowledge to be able to use the system. Based on this basis, the system was designed to be easy and convenient for the user or system analyst. It was adopted a few variables that the user can select through the interface system before implementation with a default value.

C- Multiple feature use

This system is characterized as working in the form of dynamic, where it was enhanced to have the ability to work on multiple features simultaneously, allowing the user or system analyst to choose more than one feature at a time.

4.4 Result Analysis

The results got by the proposed system have two parts, the aggregation system final results and threat score final results. Before discussing proposed system results, the hardware and software specifications were used to check system are described in the next section.

4.4.1 Hardware Specifications

To test the performance of the proposed system, these hardware specifications were utilized:

1. CPU: Intel ® i3 Core CPU.
2. Memory: 4.00GB.
3. Hard Drive: 500GB.

4.4.2 Software Specifications

The following program specifications were used to test the performance of the proposed system as follows:

1. Operating System: Windows 7 Professional
2. Compiler: Visual Basic
3. Database: Microsoft Office Access 2010

4.4.3 Results for Five Steps

The implantation of the proposed system can be summarized into five steps:

Step1: Feature types

In this step will read the input file of DARPA dataset to create table contains from two columns, Feature column which consists name of each feature. The second column consists types of this feature. This step very important because of each DARPA dataset day differences from one to another. As well as will assess the next step. Table 4.2 depicts the result of this step.

Table 4.2: Features with its types

Feature	Classify
RuleFileComes	ICMP PING,ICMP Echo Reply,ATTACK-RESPONSES 403 Forbidden,ICMP Destination Unreachable Port Unreachable,CHAT IRC nick change,CHAT IRC channel join,CHAT IRC message,SHELLCODE x86 NOOP,ICMP Fragment Reassembly Time Exceeded,X11 xopen,ICMP Time-To-Live Exceeded in Transit,ICMP PING *NIX,ICMP PING BSDtype,SHELLCODE x86 inc ecx NOOP,ATTACK-RESPONSES Invalid URL,EXPLOIT javascript handler in URI XSS attempt,SHELLCODE x86 inc ebx NOOP,WEB-CLIENT Telnet protocol specifier in web page attempt,SHELLCODE sparc NOOP,ATTACK-RESPONSES directory listing
Classification	Misc activity,Attempted Information Leak,Potential Corporate Privacy Violation,Executable code was detected,Unknown Traffic,Attempted User Privilege Gain,Potentially Bad Traffic
Priority	3,2,1
Protocol	ICMP,TCP
TTL	32,255,64,63,254,62,61,60,59,58,57,56,55,54,53,52,51,50,49,48,47,46,45, 44,43,42,41,40,39,38,37,36,35,34,33,31,30,29,28,27,26,25,24,23,22,21,20, 19,18,17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,253,252,251,250,249,248, 247,246,245,244,243,242,241,240,239,238,237,236,235,234,233,232,231, 230,229,228,227,226,225,224,223,222,221,220,219,218,217,216,215,214, 213, 212,211,210,209,208,207,206,205,204,203,202,201,200,199,198,197, 196, 195,194,193,192,191,190,189,188,187,186,185,184,183,182,181,180, 179,178,177,176,175,174,173,172,171,170,169,168,167,166,165,164,163, 162,161,160,159,158,157,156,155,154,153,152,151,150,149,148,147,146, 145,144,143,142,141,140,139,138,137,136,135,134,133,132,131,130,129, 128,127,126,125,124,123,122,121,120,119,118,117,116,115,114,113,112, 111,110,109,108,107,106,105,104,103,102,101,100,99,98,97,96,95,94,93, 92,91,90,89,88,87,86,85,84,83,82,81,80,79,78,77,76,75,74,73,72,71,70,69, 68,67,66,65
Type	8,0,NONE,3,11
Code	0,NONE,3,1

Step 2: Entropy Features

This step based on the Enhanced ID3 algorithm which explained in chapter three section 3.4. in this step will calculate the Entropy for each feature. The equation below used for this reason

$\mathrm{Info}(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$,

The result of this step shows in the following Table

Table 4.3: Results of entropy feature

Feature	entropy1	entropy2
EXPLOIT javascript handler in URI XSS attempt	0	
SHELLCODE x86 inc ebx NOOP	0	
WEB-CLIENT Telnet protocol specifier in web page attempt	0	
SHELLCODE sparc NOOP	0	
ATTACK-RESPONSES directory listing	0	
Misc activity	0	
Attempted Information Leak	0	
Potential Corporate Privacy Violation	0.15791	0.23868
Executable code was detected	0	
Unknown Traffic	0	
Attempted User Privilege Gain	0	
Potentially Bad Traffic	0	
3	0	
2	0	
1	0	0.23868
ICMP	0	
TCP	0.09794	0.23868
32	0	
255	0	
64	0.26944	0
63	0.56724	0.59167

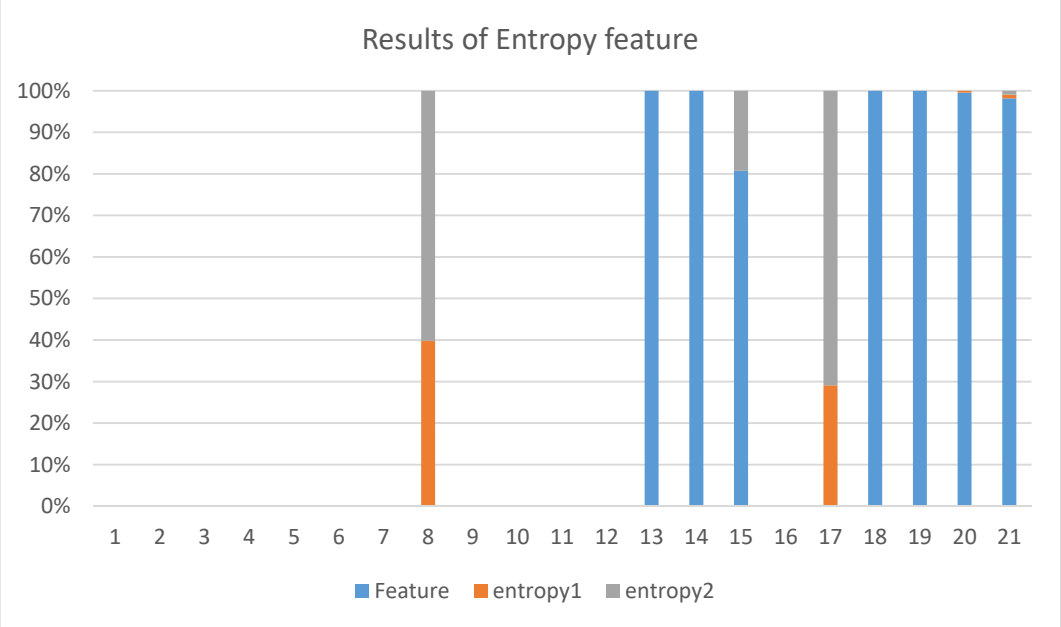

Figure 4.1: Result of entropy feature

From the results of the above table, could notice there are many entropy values for each feature and some features has just one entropy value. This because the ID3 algorithm takes many iterations until stop generating entropy value. For this reason, will need to compute the average of Entropy value will explain in the next step.

Step 3: Average Entropy value

In this step as mentioned in Step 2, will compute average entropy value for each entropy based on the new equation in the Enhancement ID3 algorithm in below:

$$\text{AverageFeatureEntropy} = \sum \text{AllEntropy Feature} / \text{No of entropy}$$

Table 4.4 shows the result of this step. The above equation flexible to calculate the average because vary from one feature to another, some of which have one value and others have two or more values. Thus, the equation can calculate the rate of grade, despite the different numbers.

Table 4.4: Result of average entropy features

F1	F2	F3	F4	F5	F6	F7
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0.16831	0.13472	0.16831	0.16831
0	0	0	0	0.13472	0	0
0	0	0	0	0.13472	0	0
0	0	0	0	0.13472	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0.198295	0.11934	0.16831	0.579455	0.16831	0.16831
0	0.198295	0.11934	0.16831	0.579455	0.16831	0.16831
0.23868	0.198295	0.11934	0.16831	0.579455	0.16831	0.16831
0	0	0	0	0.13472	0	0
0	0	0	0	0.13472	0	0
0	0	0	0	0.13472	0	0
0	0	0	0.16831	0.13472	0.16831	0.16831
0	0	0	0.16831	0.13472	0.16831	0.16831

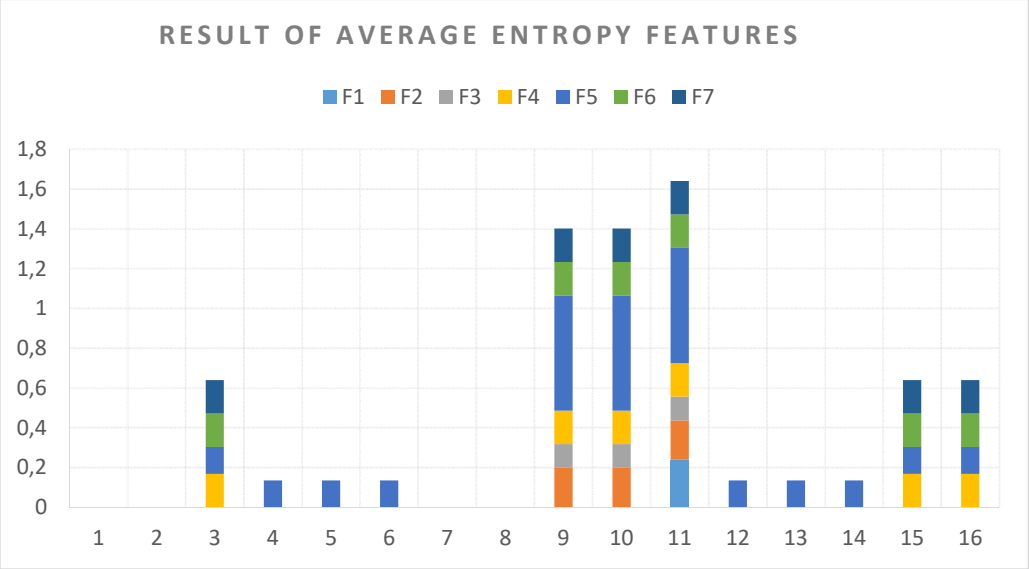

Figure 4.2: Chart of result of average entropy features

Step 4: Entropy value of Alert

According to on the average entropy value for features that computed in step3, the entropy value of alert will calculate. It will have based on the output of step3 to be as input data. Implemented the new equation in the enhancement ID3 algorithm in chapter three section 3.4 will use to calculate the entropy value for each alert. Below the new equation as mentioned above:

$$\text{AlertEntropy} = \sum \text{FAE} / \text{No of F}$$

Where:

FAE: Feature alert entropy value

No of F: numbers of Features

Step 5: Assessment Alert:

The final step; it is calculating the assessment degree for each alert and classify it into two types (True or False). The Table 4.5 depicted result of the implemented step 5. This Table consists of 3 columns, first column name of alert, the second column for assessment value which it is computed based on the following:

$$\text{If Priority} = 1 \text{ then NPV} = 1$$

$$\text{If Priority} = 2 \text{ then NPV} = 0.5$$

$$\text{If Priority} = 3 \text{ then NPV} = 0.3$$

$$\text{AssessmentAlert} = (\text{AlertEntropy} + \text{NPV})/2$$

And the third Colum for the Final class that calculated based on the following:

If AssessmentAlert >= 0.55 then Alert = True Otherwise Alert = False

Table 4.5: Result of assessment alerts

Name Alert	Assessment Value	Final Class
Alert 1	0.5	False
Alert 2	0.5	False
Alert 3	0.59	True
Alert 4	0.5	False
Alert 5	0.5	False
Alert 6	0.195	False
Alert 7	0.15	False
Alert 8	0.15	False
Alert 9	0.6	True
Alert 10	0.4	False

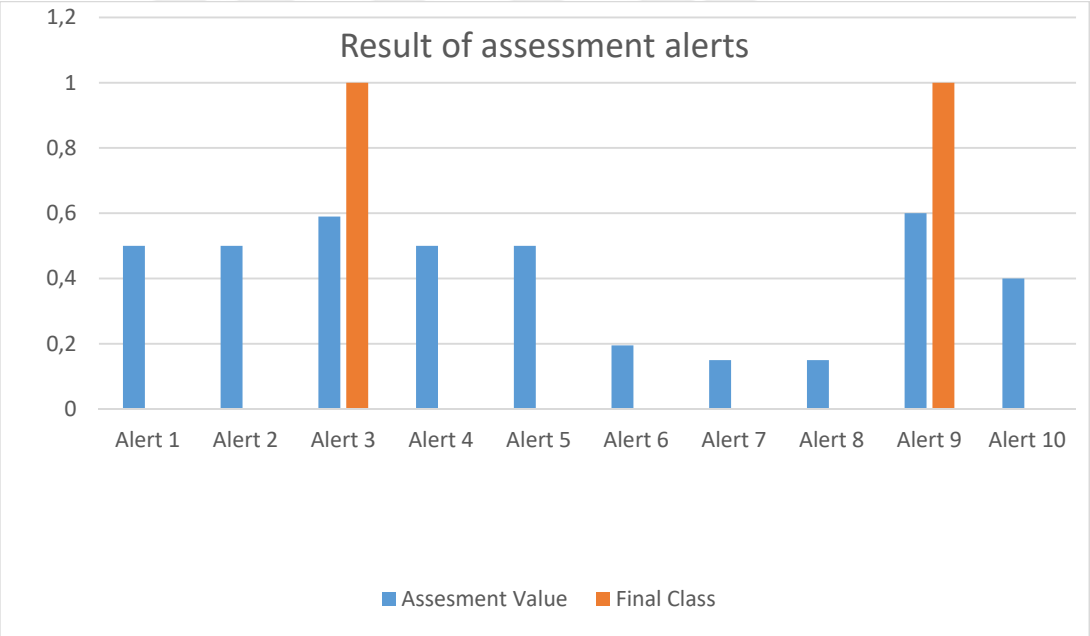

Figure 4.3: Chart of result of assessment alerts

4.4.4 Final Results for Proposed System

For the purpose of generating an acceptable alert group, a database IDS Snort 2.9 [20] was used in our check leverage.

Five days of the five weeks were used in the experiments for the data set (DARPA 1999). In subsequent paragraphs will be explained experiments.

The Experiments; 1, 2, and 3 for three weeks (Second, Fourth, and Fifth) which it is contained attack.

Experiment 1: This experiment was conducted in five days (Monday, Tuesday, Wednesday, Thursday, and Friday) in the Second week of the DARPA 1999 data set, which contained traffic prepared for use as the test set of our proposed framework.

Table 4.6: Alert reduction for DARPA 1999 (second week)

Date	Amount Alerts Before Reduction	False Alerts	True Alerts	Ratio
Monday	2731	2596	135	0.950567
Tuesday	1550	1469	81	0.947741
Wenesday	3811	3788	23	0.993964
Thursday	4126	4023	103	0.975036
Friday	3401	3336	65	0.9808873
Total	15619	15212	407	0.97394199

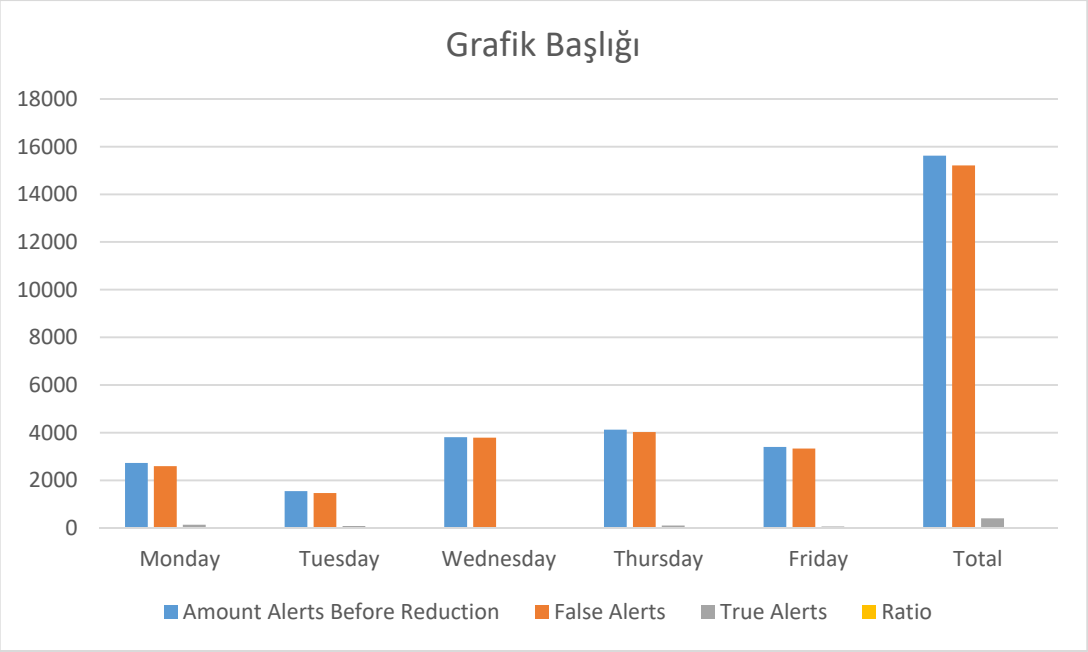

Figure 4.4: Chart of alert reduction for DARPA 1999 (Second Week)

Table 4.6 shows alerts results that obtained before and after implementation of the proposed system. alerts number reduced in our system with an average alert reduction rate of 97.39%.

Experiment 2: This experiment was conducted in five days (Monday, Tuesday, Wednesday, Thursday, and Friday) in the fourth week of the DARPA 1999 data set, that contained traffic prepared for use as the test set of our proposed system.

Table 4.7: Alert reduction for DARPA 1999 (Fourth Week)

Date	Amount Alerts Before Reduction	False Alerts	True Alerts	Ratio
Monday	2598	2347	251	0.903387
Tuesday	1197	1116	81	0.932330
Wednesday	1728	1624	104	0.939814
Thursday	1314	1132	182	0.861491
Friday	1813	1681	132	0.927192
Total	8650	7900	750	0.9132948

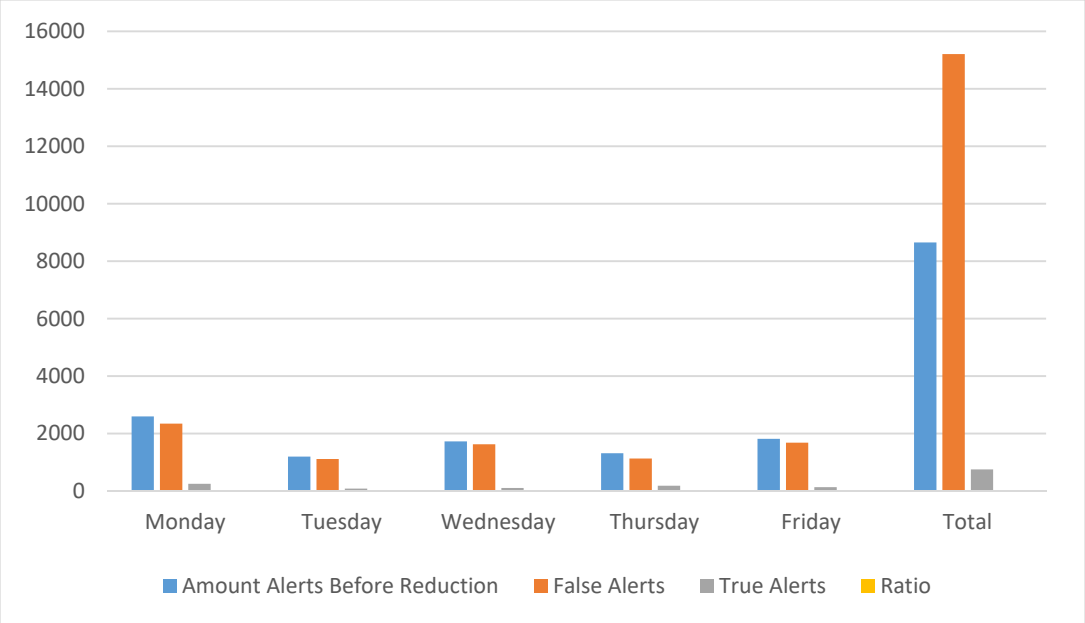

Figure 4.5: Chart of alert reduction for DARPA 1999 (Fourth Week)

Experiment 3: This experiment was conducted in five days (Monday, Tuesday, Wednesday, Thursday, and Friday) in the fifth week of the DARPA 1999 data set, which contained traffic prepared for use as the test set of our proposed system.

Table 4.8: Alert reduction for DARPA 1999 (Fifth Week)

Date	Amount Alerts Before Reduction	False Alerts	True Alerts	Ratio
Monday	1604	1539	65	0.959476
Tuesday	9509	9403	106	0.988853
Wednesday	1408	1357	51	0.96377841
Thursday	1457	1228	229	0.84282773
Friday	4120	4102	18	0.99563107
Total	18098	17629	469	0.97408553

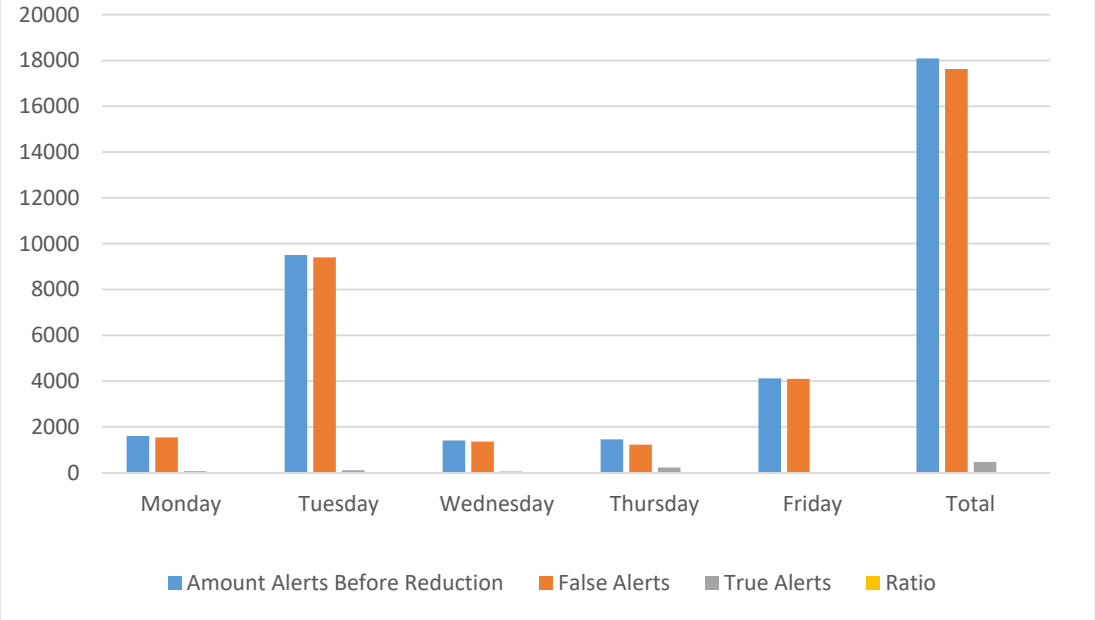

Figure 4.6: Chart of alert reduction for DARPA 1999 (Fifth Week)

Experiment 4: This experiment was conducted in five days (Monday, Tuesday, Wednesday, Thursday, and Friday) in the first week (label-free attack) of the DARPA 1999 data set, which contained traffic prepared for use as the test set of our proposed system.

Table 4.9: Alert reduction for DARPA 1999 (First Week)

Day	Total	FALSE	TRUE	First
Monday	1164	1164	0	100%
Tuesday	1294	1279	15	98.84%
Wednesday	1041	1032	9	99.13%
Thursday	1063	1034	29	97.27%
Friday	2713	2639	74	96.63%
Total	7275	7148	127	98.25%

Experiment 5: This experiment was conducted in five days (Monday, Tuesday, Wednesday, Thursday, and Friday) in the Third week (label-free attack) of the DARPA 1999 data set, which contained traffic prepared for use as the test set of our proposed system.

Table 4.10: Alert reduction for DARPA 1999 (third week)

Day	Total	FALSE	TRUE	First
Monday	1281	1266	15	98.82%
Tuesday	1138	1104	34	97.01%
Wednesday	1307	1304	3	99.77%
Thursday	1224	1189	35	97.14%
Friday	1354	1348	6	99.55%
Total	6304	6211	93	98.52%

The Table 4.11 depicted summarized the result for all five weeks, the first and third weeks (Free attack) and the Second Fourth, and Fifth weeks (attack label).

Table 4.11: Classification the assessment rate based on attack label

Week	Amount Input Alerts	Amount Alerts After Assessment	Assessment
First & Third (free Attack)	13579	220	98.37%
Second&Fourth&Fifth (label Attack)	42367	1626	96.16%
Total Weeks	55946	1846	96.70%

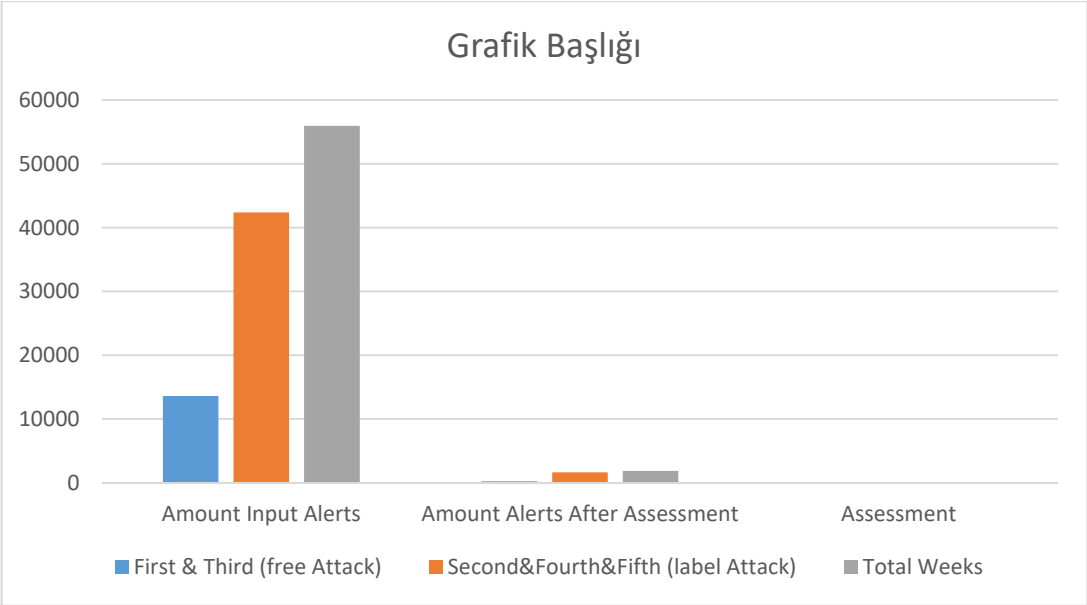

Figure 4.7: Classification the assessment rate based on attack label

Aby was observing and analyzing the Table (4.11) that was obtained through the implementation of the proposed method to reduce and, depending on the information provided by DARPA Website. The first week and third are free from attack (Free attack label), and through the results in the Table (4.11), that the rate of reduction for those two weeks, is 98.37% with an average error rate not to exceed 1.63% and this is something that gives a clear indication of strongly the proposed method and credibility to reduce False alert.

As the weeks (second, fourth and fifth), they contain attacks (attack label) and note when the results of these three weeks, it is noted that the ratio of rate reduction is 96.16% and the ratio is good, as the overall rate of false alerts is 99% (Elshoush and Osman, 2011). Thus, the error rate of the proposed method does not exceed 2.84%. The total amount of the whole alerts before reduction was 55946 alert, and the amount of the whole alert after reduction was 1846 alert, so the rate reduction of the alert was **96.70%**.

4.5 Comparing With Other Approaches

There are many approaches mentioned in chapter two for IDS alerts have been reduced, according to on it, Table 4.12 shows the comparison between our proposed reduction system approach and the previous related approaches [21], [22], [23], [24], [25], [26], all of them using Darpa1999 data set.

Table 4.12: The comparison with others systems

Approaches	Amount Alerts	Duration	Reduction Rate
(Pietraszek, 2006)	59812	5 weeks	63.00%
(Jie Ma et al., 2008)	None	5 weeks	90.00%
(Al-Mamory et al., 2010)	233615	5 weeks	70.00%
(Njou and Jiawei, 2010)	29548	Three Days(Thursday 4th week, Thursday and Friday 5th week)	78.00%
(Karim Al-Saedi 2013)	57785	5 weeks	92.27%
(Dhiya Ibraheem Selman, 2017)	62785	5 weeks	88.00%
Our proposed system	55947	5 weeks	96.70%

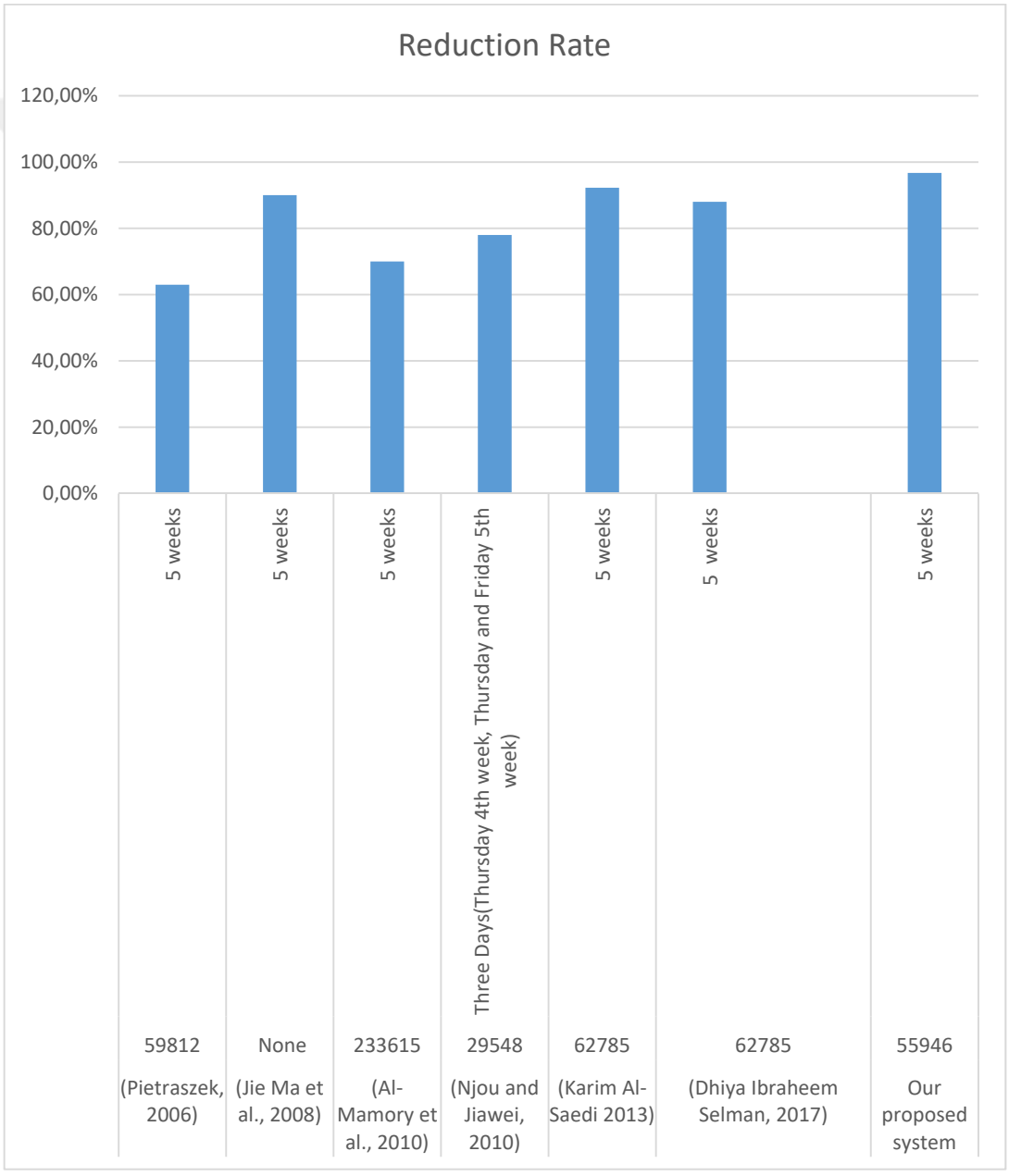

Figure 4.8: Chart of the comparison with others systems

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1 Introduction

In the preceding chapters, Management and Assessment system for network attacks based on data mining techniques and phases of it (Feature Selection Phase, ID3 Algorithm Phase, Alert Entropy Phase, Assessment Alert Phase, and Classification Alert Phase.) it was discussed by details. In this chapter, section 5.2, it will give a conclusion about the whole system by discussing the objectives and the goals achieved, section 5.3, it will be discussed the future works.

5.2 Conclusions

In this thesis, all the objectives were achieved, Studied and analyzed alerts generated from intrusion detection system, to know the features contained in the alert and feature selection which it is given good result during the testing and saved it in DB file type Microsoft Access.

To compute the entropy value for each feature and each alert have been achieved through Feature Entropy Phase, and Alert Entropy Phase based on the new Enhancement ID3 algorithm.

To assess the alerts which arrived, we used Assessment Alert Phase based on new equations which calculate the assessment degree for the alert. Depending on the degree of assessment, alerts are classified according to threat degree (false and true).

To calculate the ratio of reduction alert has been achieved, we used Classification Alert Phase based on ratio equation.

This system was proven the efficiency from the results obtained. For examples, the DARPA 1999 data set that was used, the weeks (second, fourth and fifth), they contain attacks (attack label) and when noting the results of these three weeks, it is noted that the ratio of rate reduction is 96.16% it is considered a good rate compared with past results for researchers and also with the truth rate of false alerts by up to 99% (Elshoush and Osman, 2011). Thus, the error rate of the proposed method does not exceed 2.84%.

the first week and the third week, they contained free real alerts (free attack label). its alerts were reduced using the classification alerts phase and received a high percentage (98.37%) from 100%. Thus proved its effectiveness in achieving good results, and can be considered as a subsystem integral to the process of reducing to get the results of high-resolution.

The total amount of the whole alerts before reduction was 55946 alert, and the amount of the whole alert after reduction was 1846 alert, so the rate reduction of the alert was 96.70%.

5.3 Future Works

The system implemented and carried out with Snort IDS and it is possible to work with any other type of IDS. Despite the good results achieved, to increase efficiency more, it is possible to achieve that by design a new IDS, it will be an integrated and more efficient performance without based on common detection systems.

REFERENCES

1- El Mostapha Chakir*, Chancerel Codjovi, Youness Idrissi Khamlichi, Mohammed Moughit," False Positives Reduction in Intrusion Detection Systems Using Alert Correlation and Data Mining Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 4, 2015, ISSN: 2277 128X.

2- Dheeraj Mishra, Pratik Patil, Akash Naundla, Roshan Shah," Intrusion Detection System", 3rd International Conference On Recent Trends in Engineering Science and Management, Vedant College of Engineering and Technology, Bundi, Rajasthan, 10th April 2016, Isbn:978-81-932074-4-4.

3- Dr. S. Vijayarani1 and Ms. Maria Sylviaa.S," Intrusion Detection System – A Study", International Journal of Security, Privacy and Trust Management, Vol 4, No 1, February 2015, (IJSPTM).

4- Chanchal Yadav, Shuliang Wang, Manoj Kumar," Algorithm and approaches to handling large Data-A Survey", IJCSN International Journal of Computer Science and Network, Vol 2, Issue 3, 2013 ISSN (Online): 2277-5420.

5- Benjamin Morin and Herv_e Debar, "Correlation of Intrusion Symptoms: an Application of Chronicles", International Workshop on Recent Advances in Intrusion Detection: Recent Advances in Intrusion Detection pp 94-112, Springer, Berlin, Heidelberg, ISBN978-3-540-40878-9.

6- Risto Vaarandi, "A Data Clustering Algorithm for Mining Patterns from Event Logs", Reprinted from Proceedings of the 2003 IEEE Workshop on IP Operations and Management, (ISBN: 0-7803-8199-8).

7- Jouni Viinikka, Herv´e Debar, Ludovic M´e, Renaud S´eguier," Time Series Modeling for IDS Alert Management", ACM Symposium on Information, Computer and Communications Security, ASIACCS, Mar 2006, France. ACM, pp.102-113, 2006.

8- Safaa O. Al-Mamory · Hongli Zhang," New data mining technique to enhance IDS alarms quality", J Comput Virol, (2010), 6:43–55.

9- Anita Rajendra Zope, Amarsinh Vidhate, and Naresh Harale, "Data Mining Approach in Security Information and Event Management", International Journal of Future Computer and Communication, Vol. 2, No. 2, April 2013.

10- David Ndumiyana1, Richard Gotora2 and Hilton Chikwiriro," Data Mining Techniques in Intrusion Detection: Tightening Network Security, International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 5, May – 2013, ISSN: 2278-0181.

11- Asieh Mokarian, Ahmad Faraahi, Arash Ghorbannia Delavar," False Positives Reduction Techniques in Intrusion Detection Systems-A Review", IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.10, October 2013

12- Meghana solanki, Vidya Dhamdhere," A Hybrid Approach for Intrusion detection using Data Mining", Vol. 4, Issue 7, July 2015.

13- Inadyuti Dutt, Dr. Samarjeet Borah," Some Studies in Intrusion Detection using Data Mining Techniques", Vol. 4, Issue 7, July 2015, ISSN: 2319-8753.

14- Ankit Naik, S.W. Ahmad,"Data Mining Technology For Efficient Network Security Management ", International Journal Of Computer Science Trends And Technology (Ijcst) – Volume 3 Issue 3, May-June 2015.

15- Yousef El Mourabit, Ahmed Toumanari, Anouar Bouirden, Nadya El Moussaid," Intrusion Detection Techniques in Wireless Sensor Network Using Data Mining Algorithms: Comparative Evaluation Based On Attacks Detection", (Ijacsa) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 9, 2015.

16- Premansu Sekhara Rath, Manisha Mohanty, Silva Acharya, Monica Aich," Optimization of Ids Algorithms Using Data Mining Technique", International Journal of Industrial Electronics and Electrical Engineering, Volume-4, Issue-3, Mar.-2016, Issn: 2347-6982

17- Andrew R, Baker Joel Esler," Snort IDS and IPS Toolkit", Copyright © 2007 by Syngress Publishing, Inc, ISBN-13: 978-1-59749-099-3.

18- Fabricio Voznika Leonardo Viana" Data Mining Classification", (Proceedings of The 5th European Conference, Pkdd 2001) – Lecture Notes in Artificial Intelligence 2168, 314-325, Springer, 2001.

19- Fredrik Valeur," Real-Time Intrusion Detection Alert Correlation", UNIVERSITY OF CALIFORNIA Santa Barbara, A Dissertation Submitted in Partial Satisfaction of the Requirements for The Degree of Doctor of Philosophy in Computer Science, June 2006.

20- Huwaida Tagelsir Elshoush and Izzeldin Mohamed Osman,” Alert Correlation in Collaborative Intelligent Intrusion Detection Systems -ASurvey”, Applied Soft Computing Journal, doi: 10.1016/j.asoc.2010.

21- Tadeusz Pietraszek," Alert Classification To Reduce False Positives In Intrusion Detection", Institut F¨Ur Informatik, Albert-Ludwigs-Universit¨At Freiburg Georges-K¨Ohler-Allee 52, 79110 Freiburg I. Br., Germany, July 2006.

22- Dong Li, Zhitang Li, Jie Ma,” Processing Intrusion Detection Alerts in Large-scale Network”, Electronic Commerce and Security, International Symposium on Guangzhou City, China, IEEE, 2008, **ISBN**: 978-0-7695-3258-5.

23- Safaa O. Al-Mamory · Hongli Zhang," New DATA MINING TECHNIQUE TO ENHANCE Ids ALARMS QUALITY", J Comput Viro, L, 2010 ,6:43–55.

24- Humphrey Njogu,” Using Alert Cluster to reduce IDS alerts”, Conference Paper, DOI: 10.1109/ICCSIT.2010.5563925 · Source: IEEE Xplore, August 2010.

25- Karim Hashim Kraidi Al-Saedi,” A False Alert Reduction and an Alert Score Assessment Framework For Intrusion Alerts “, PhD thesis, Universiti Sains Malaysia, 2013.

26- Dhiya Ibraheem Selman,"false positive alerts reduction system based on data mining techniques", master thesis, Institute of Informatics for Postgraduate Studies, Iraq, 2017.

27- Badr Hssina, Abdelkarim Merbouha, Hanane Ezzikouri, Mohammed Erritali," A Comparative Study of Decision Tree Id3 and C4.5", (Ijacsa) International Journal of Advanced Computer Science and Applications, Special Issue On Advances in Vehicular Ad Hoc Networking and Applications,2014.

CURRICULUM VITAE

PERSONAL INFORMATION

Name, Surname : Ahmed Abdullah
Nationality : Iraqi
Date and Place of Birth : 05 September 1979, Iraq- Baghdad
Marital Status : Married
Phone : +90 539 791 78 94
Email : ahmed_prog_2008@yahoo.com

EDUCATION

Undergraduate : University of Baghdad / Faculty of Science Ibn Al - Haytham / Computer Science Department, 2005.

WORK EXPERIENCE

Place	Year
Iraqi Ministry of Water Resources/ General Commission for Dams & Reservoirs.	2006-2012
Iraqi Ministry of Water Resources/ Rafidain Company for the implementation of dams	2012- present

FOREIGN LANGUAGE

English