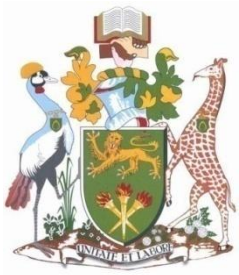


UNIVERSITY OF NAIROBI

College of Biological and Physical Science

School of Computing and Informatics

COMPARATIVE ANALYSIS OF ANOMALLY DETECTION ALGORITHMS

By

PATRICK KABUE

P53/65356/2013

SUPERVISOR

DR. ELISHA O. ABADE

Submitted in partial fulfillment of the requirements for the award of a Master of Science Degree in Distributed Computing Technology of the University of Nairobi.

March 2015

STUDENT DECLARATION

This research project is my original work and has not been presented for award of any degree in any University

……. ……..…….....……

Signature Date

Patrick Kabue

This research project has been submitted for examination with my approval as University Supervisor

…… ……..…….....……

Signature Date

Dr Elisha O. Abade

ACKNOWLEDGEMENT

The Almighty God for strength, safety and guidance throughout the duration of this course,

My Supervisor Dr Elisha O. Abade for his critic, time, patience, direction and guidance during the duration of the project.

My colleagues and friends at my place of work for their understanding during my absence from work, and, my family.

My family for their understanding and patience and support.

Finally I wish to acknowledge my fellow students at School of Computing and Informatics and presentation panelists for their criticisms, correction and suggestions on this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

DEFINITION OF TERMS

Antivirus: This a software that is installed into your with the aim of protecting it against malicious software intended to spoil the computer.

ARP Reply Request: Address resolution protocol is used to convert the IP address of a computer to the physical of the computer. A host requiring the physical address of a computer sends a request to the TCPIP network.

ICMP destination unreachable. This is a type of attack where the attacker keeps sending a destination unreachable port message to either client or server. The attacker keeps brute forcing the client computers with messages and eventually when the client port pair is found, the connection is eventually dropped.

ICMP Flood:. This a type of attacks that where the attacker sends several ICMP messages, to the victim computer leaving it with degraded performance. The attacker spoofs the addresses of the victim host and sends a great number of ICMP echo request packets to the broadcast address of the network. All hosts on the network will respond to that ICMP echo request with a corresponding reply to the spoofed IP address of the victim.

SYN Flooding attacks.: This involves attacking the three way handshake. the handshake begins by machine A initiating a connection with machine B by sending a SYN request. Machine B responds by sending an ACK+SYN signal to A. Machine A responds with an ACK signal. In the case of SYN flood attack. During a SYN attack, the attacker continuously sends continuous SYN signals making the victim computer to continually respond to the SYN requests

Tespok iCRIS: Telecommunications Service Providers Association of Kenya (TESPOK). TESPOK is a professional, non-profit organization representing the interests of service providers in Kenya.

Ping: This is a windows command that is used to check for interconnectivity with another device that is assigned an internet protocol address in a network

Trojan horse: this is a program that is designed to carry out a genuine task but has a hidden code that carries out a task that is not legitimate like having a key logger or retrieving information from your compute and sending the information to another party.

Tracert: Is a windows command that finds the number of hops that are in between the source and destination computer

Tcpdump: this is a Linux based network traffic packet capture software

Virus: Is a program that gets installed into your computers against your wish. These programs replicate themselves and are designed to damage your computer software.

Wire shark: This is a network protocol analyzer that was used to collect data on information flowing through the network

Zero day attack: these are attacks whose nature is not previously known and have not been experienced before and exploit a security hole that is previously not known to the manufacturers of the system.

Zombie: This is a computer that is infected or whose security has been compromised and is used to launch attacks against other computers within a network.

ABBREVIATIONS AND ACRONYMS

Botnet	-	Ro*bot Net*work.
DoS	-	Denial of Service
DNS	-	Domain Name System.
HTTP	-	Hyper Text Transfer Protocol
IRC	-	Internet Relay Chat This is multi channel protocol used to send
ICMP	-	Internet Control Management Protocol
IP	-	Internet Protocol
Malware	-	*mal*icious soft*ware*
SQL	-	Structured Query Language
SMTP	-	Simple Mail Transfer Protocol
TCP	-	Transmission Control Protocol

ABSTRACT

Attacks with no previously known signatures present a challenge on how to detect them. These attacks (commonly are referred to as zero day attacks) have not been experienced before and exploit vulnerability previously no known. However these attacks have characteristics that differ from those of normal of attack free packets. These changes in network traffic packets can be detected by comparing anomalous packets with those that do not have attacks.

we selected algorithms that detect anomalies based on packet header and evaluated them by measuring three metrics (False positive ratio, accuracy and detection rate). This entailed use of two sets of tcpdump data .The first set of data was attack free training data that was used to train the algorithms so as to set a basis for the comparison with the data to be tested. The second data set contained labeled which had been previously identified. These attack have been carefully identified and their location in the dataset was known and documented The algorithms were trained using the training dataset and later attempted to detect the attacks in the test dataset. Once an anomaly was identified, the algorithms the produced a outputs containing IP address of the victim, date of the attack, score and field contributing the most to the anomaly. The dataset used also has an evaluation truth table that contains a list of all the attacks in the dataset. This table contains the date the attack occurred, time, and IP address of the victim computer.

Analysis of the data entailed cleaning of the results of the algorithms to remove unnecessary fields using Ms Excel. These data was uploaded into MS SQL server and a column labeled status was added to the table containing the algorithm results. We compared results of the algorithms with the evaluation truth table detection list. This was done using a program created using Visual Basic and any matching record was updated with an entry of true positive in the status column while records not matching were marked with a false positive entry.

The results indicated the algorithms have a high false positive ratio and a very low accuracy with Packet header anomaly detection algorithm being the performing algorithm among the algorithms evaluated.

CHAPTER ONE: INTRODUCTION

1.0 Background

The need for exchange of information between people and organizations has lead to increased interconnectivity of devices used at both the source and destination computers. This information is transmitted in the form of packets across several devices such as switches, routers, gateways, modems and along various routes to get to the destination device.

Contents of the packet are governed by the Internet Protocol and has two sections, the header and payload sections. The payload section consists of data that is being sent from the source to the destination. The header comprises necessary information to deliver the packet to the destination. Security of this two sections is crucial for delivery of the packet to the end.

These packets are subject to malicious activities such as viruses, worms, network attacks and other non malicious changes such malfunctioning equipment. Malicious attacks are intended to alter accuracy, integrity and confidentiality of the data in transit. These attacks cause the data to be illegible altering the contents of the payload. Viruses, Trojan horses, Worms can be carried on the payload data and may cause harm to computer software's and affect normal operations. These malicious software may also be used to gather information from computers and send the information to third party computers.

The packet headers carry information such as sources address of the source device, destination address ,TOS, header length, checksum among others. These information is necessary for end to end connectivity and changes to the information may affect delivery of the packets. Malicious attacks use these information in launching attacks search as Denial of Service attacks that can bring huge computer networks to a halt affecting many organizations using the networks.

To protect themselves from these malicious activities organizations have invested heavily in software and hardware that can detect and stop these activities. These software such as antivirus software's work by checking traffic in transit for known signatures of the malicious attacks. These attacks (payload or packet header based) are known as cyber attacks and cause huge losses to organization in terms of man hours used to stop them and loss services. According to Cyber

report, 2014 Kenya loses 2 billion shillings annually with denial of service attack being among the top attacks.

These signatures are created based on what is known about the attacks (signatures) and need regularly updated so as to detect the malicious activities. Other software's such as firewalls work by filtering network traffic packets based on rules that are created by the security personnel.

Signature based detection works well with known attacks, however in case of attacks that have not been encountered before, there are no known signatures for use. Such attacks are difficult to stop and may go undetected using signatures. Attempts have been made to detect previously unknown attacks based on changes that occur in the packets. These are known as anomaly based detection systems. These systems can work by trying to detect anomalies on the packet header or payload based or on both sections of the packet.

Packet header based anomaly detection systems work by comparing the normal characteristics of the packet with those of the packets under review for any characteristics that may vary from the norm. Attacks exhibit characteristics that are different from those of normal traffic (Denning, 1987). Such characteristics could be a checksum that is not adding up or TTL field that remains constant in every hop.

It is on this basis that anomaly detection algorithms are designed. These detection algorithms detect changes in characteristics such as IP header flags, checksum, source and destination address packet header information among others, as well as actual data being sent payload and ports in use.

1.1 Problem statement

Network packets in transit in a network occasionally face attacks that exploit previously unknown vulnerability in software or in the protocol in use. These attacks do not have known signatures that can be used to detect them as they are unknown or have not been experienced before.

1.2 Research Objectives

This research aims at analyzing anomaly detection algorithms designed to detect zero day attacks in network traffic in the absence of signatures of the attacks. To analyze the algorithms we aimed at;

1. Determining the false positive ratio. These are the anomalies that are detected by the algorithms as containing network attack however they do not contain any attacks.
2. Determining the accuracy levels of the algorithms in detecting attacks. These are detections that correctly identified by the algorithms as containing network attacks
3. Determining the detection rate of the algorithms. This is the ability of the algorithm to detect anomalies in network traffic.

1.3 Research Questions

The following questions was help to support the research problem ;

1.How many false positives detections can be detected by the anomaly detection algorithms?

2.What are the accuracy levels of the algorithms being analyzed?

3.How many true positive detections exist in the anomalies detected by the algorithms.

1.4 Scope

This study focuses on packet header based anomaly detection algorithms. The aim is to determine the accuracy, detection rate and the false positive ratio of the algorithms in detecting attacks in network traffic based on various characteristics of the packet header values.

1.5 Significance of study

New attacks are generated every minute, these attacks do not prior known signatures that can be used to detect these attacks using signature based detection systems. Anomaly based detection algorithms have been developed that rely on characteristics of the network traffic packets to detect anomalies.

Output from these algorithms are used by network security personnel in order to take corrective actions and stop these unknown attacks. These personnel need accurate and reliable results on detections made by the algorithms.

1.6 Assumptions and Limitations

Assumptions:

- The anomalies in network traffic are as a result attacks. Analysis of data packets will result in detection of patterns that lead to detection of network attack.
- Data used in training anomaly detection algorithms will not contain anomalies as this will have a negative effect on the ability of the algorithms in detecting anomalies

Limitations

- Anomaly detection algorithms do not identify the type of attach that has been detected ,hence there is no way of determining the detected anomalies are a a result of changes in network configuration or malfunctioning equipment.

CHAPTER TWO: LITERATURE REVIEW

2.0 Introduction

The objective is to review previous research done on anomaly detection algorithms in order to compare the false positive ratio, accuracy and detection rate of the selected algorithms against selected network attacks and data that was collected during the duration of this research.

2.1 Network Traffic Monitoring

Is a continuous process of collecting and analysis of network traffic so as to gain an understanding of network behavior and characteristics, performance and reliability in order to effectively and promptly troubleshoot and resolve various issues (Wang. et al., n.d)

2.2 What is an anomaly?

An *anomaly* is defined an *irregularity.* The word *anomalous is* being *different* from what is normal (Oxford Advanced Learners Dictionary).

George Jones (2013) defines an anomaly as a deviation from the norm; strange condition, situation or quality, an incongruity or inconsistency.

In network traffic an anomaly is anything that causes unusual and significant changes in traffic behavior, these anomalies can be malicious (caused by attacks, abusive network usage, worms or virus propagation) or unintentional (device failures, or router/ switch misconfigurations) (Houerbi et al., 2010).

Network anomaly is an event that deviates from the normal network behavior (Thorttan et al., 2013).

Anomalies in network traffic arise from various activities in networks such as malwares and cyber crime activities. Malfunctioning of equipment such as routers and switches could produce anomalous data. Network traffic Anomaly detection is a methodology that works by establishing a baseline of normal network activity over a period of time and then detecting deviations from the baseline (Nortcutt et al., 2005).

These are unusual and significant changes in the traffic of a network (Huang et al.,2014). According to Huang, these changes can be brought about by legitimate activities such as changes in customer demand, or illegitimate changes such as virus and worms, denial of service attacks or even port scans. Other causes could be malfunctioning equipment such switches and routers.

Anomalies are patterns in data that do not conform to a well defined notion of normal behavior (chandola et al, 2009)

This research is focuses on anomalies that are caused by malicious actives such as port scans and denial of service attacks.

For the purpose of this research the word anomalies is used interchangeably with the word attacks. However in this research they are used to mean the same thing.

2.3 Anomaly detection

Anomaly detection is the process of scanning for abnormal activity that is encountered in the network (Eric Cole et al., 2005). It refers to the problem of finding patterns in data that do not conform to expected behavior (Chandola 2008).Anomaly detection, Is a mechanism for detecting computer security violations (Tan., et al 2004).

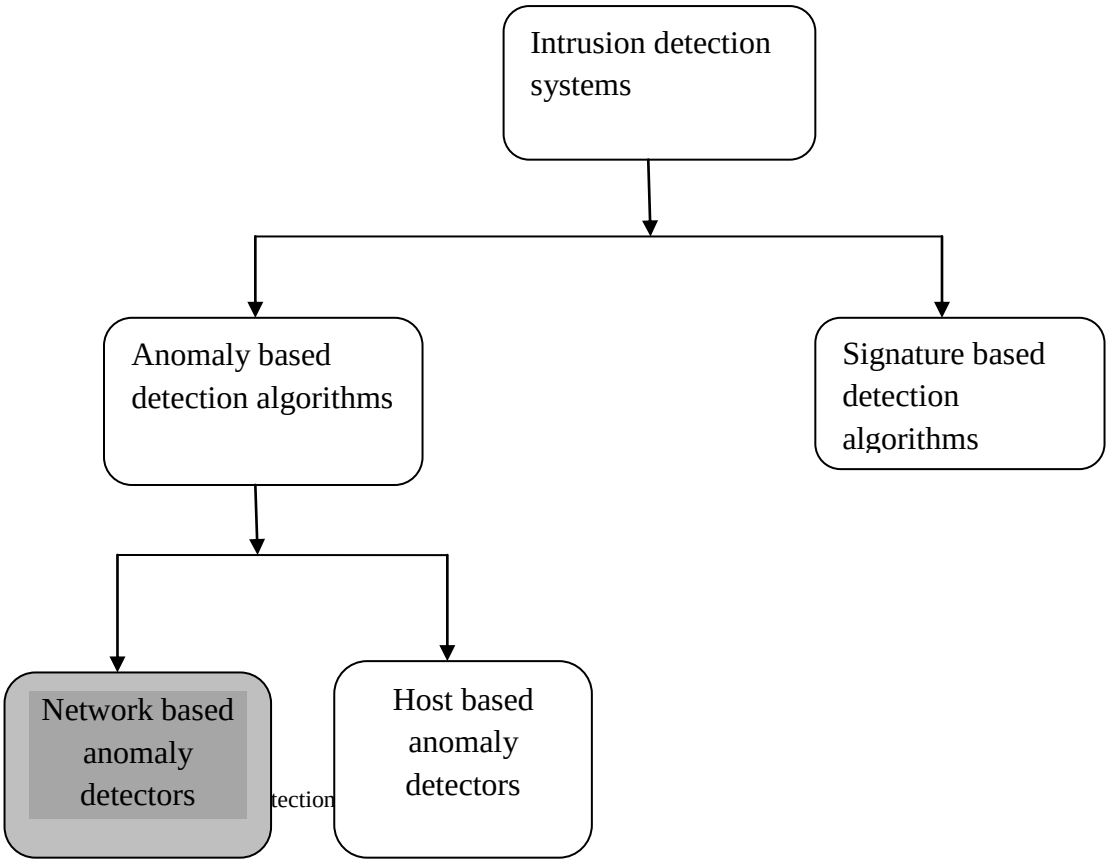
For there to be an anomaly, there has to be a way of identifying normal behavior. This behavior was the baseline for identifying anomalous behavior irrespective of the magnitude.

2.4 Algorithms

This is a technique or methodology used to undertake a certain task. In network anomaly detection, an algorithm is the methodology used to identity network attacks.

According to (Huang et al.,2014) there are two major approaches to network anomaly detection:

1) Signature-based detection algorithms
2) Non-signature-based anomaly detection algorithms.



2.4.1 Signature based detection algorithms

Signatures are normally a set of characteristic features that represent a specific attack or pattern of attacks. Signatures are generated in most cases following an actual attack (Eric Cole et al., 2005). This is a type of detection relies on a database of previously known attack characteristics /patterns. This database is created by a security expert and has to be updated regularly so as to be up to date. It works by comparing patterns in network traffic for known signatures in the signature database.

The signatures need to be manually updated by the user for example SNORT has more than 1800 rules that have been manually created and inserted (Mahoney et al, n.d).

A major disadvantage of Signature based detection is its inability to detect zero day attacks (Bilge et al.,2012). A zero day attack is a cyber attack that exploits vulnerability that has not been disclosed publicly (Bilge et al.,2012),These attacks are not in the signature database since they have not been experienced before (Huang et al., 2014).

2.4.2 Non signature based anomaly detection algorithms

There are several non signatures based detection algorithms currently in use. These algorithms are classified according to the methodology they use in attack detection. This detection

mechanism does not need to have prior knowledge of anomalies so as to detect anomalies (Huang et al., 2014). Attacks that have not been experienced before are known as zero day attacks (Jyothsna, 2014).

These algorithms work by determining normal network behavior and using this normal behavior to compare with captured traffic to determine an abnormal behavior. However there is no known model for determining normal network behavior (Thottan et al,. n.d).

Several algorithms have been used for anomaly detection including. These algorithms are classified according to the methodology used to detect anomalies:

2.4.2.1 Packet Header Anomaly Detection algorithm (PHAD)

This algorithm was developed to learn ranges of values for each packet header field at the data link layer, network and transport layers (IP,UDP,ICMP), (Mahoney et al.,2004).

PHAD assumption is that events occur with a probability *p* hence scoring $\frac{1}{p}$. It uses the rate of anomalies during training to estimate the probability of an anomaly during training mode. If a packet is observed *n* times with r distinct values, then there were r anomalies during training , hence if this rate continues then the probability of the next observation being anomalous is approximately $\frac{r}{n}$. A non stationary model is used is used to deal with the dynamic behavior of real-time traffic. For the non stationary model, if an event occurred t seconds ago, then a probability of then the probability that it will occur in the next one second is $\frac{1}{t}$. (Mahoney, 2004)
Another assumption in PHAD is that anomalies occur in bursts. In training the first bust occurrence is recorded as a single anomaly however in detection mode, subsequent events are discounted by a factor t, the time since the last anomaly occurred in the current field. Hence a score each packet containing an anomalous value is assigned a score inversely proportional to the probability; which is $\frac{tn}{r}$
The scores are added up to score each packet. The fields are treated as occurring sequentially and. Hence if all the $\frac{tn}{r}$ are equal then the probability of observing k consecutive anomalies in a non stationary model is *(r/tn)(1/2)(2/3)(3/4)...((k-1)/k) = (1/k)r/tn* which is consistent with the score ktn/r that would be obtained by summation. The score of a packet is assigned score packet of $\Sigma_{i \in}$ anomalous fields $t_i n_i / r_i$ (Mahoney et al., n.d.)

PHAD works by examining 33 packet header fields. Fields that are smaller than 8 bits are grouped into a single byte. Each field is between one to four bytes divided as nearly as possible to the byte boundary specified in the request for comments (RFC), (Mahoney et al., n.d.). The aim of PHAD was to build as little as possible protocol-specific knowledge as possible into the algorithm. Like other anomaly based algorithms it can only detect unusual events. It works with assumption that the rarer the even the most likely it is hostile, hence it has a ranking mechanism for ranking anomalies. To test data this algorithm requires data first captured using tcpdump however before detecting anomalies in the dump file, the algorithm requires to be trained with training data that is attack free.

2.4.2.2 Learning Rules For Anomaly Detector (LERAD)

This algorithm works by creating rules that identify unexpected events in a time series of tuples of unordered attributes this could be packet field valued or words in a TCP session. The number of matching attribute values decreases as the time interval between tuples increases as described by paxson (1995) in busty models.

Conditional rules for LERAD are of the form "if $A_1 = x_1$ and $A_2 = x_2$ and ...

Am = xm then *Am+1 Î X = {xm+1, xm+2, ..., xm+r}",* where the A_i are nominal attributes, *x*i are values, and *m>=0 X* consists of all values of *A*m+1 observed at least once among the *n* training. At the end of training, we fix *X* and *n*. During testing, if an instance satisfies the antecedent but *A*m+1 is not in the set *X* of allowed values, it generates an anomaly score of *tn/r*, where *t* is the time since the rule was last violated, *n* is the support, and *r* = |*X*|, the number of allowed values. Otherwise the score is 0. The anomaly score for the test instance is S *tn/r* where the summation is over all rules. The score is used to rank alarms, with higher values indicating a greater probability of hostility.

LERAD has a huge number of possible rules, to cope with complexity it uses a randomized sampling approach. It works by sampling pairs of tuples with one or more matching attribute to suggest rules that satisfy both tuples, working with a small sample, *S*, of the training data, it removes "redundant" rules, keeping just enough rules to cover the values in *S* without duplication (and favoring rules with higher $\frac{r}{n}$). Next, it trains the rules on the full training set, fixing *n* and *r*.

The algorithm finally applies a validation step, removing rules that generate anomalies on a separate validation set, *V* (for example, this could be the last 10% of the training data). Validation favors "well behaved" rules, where the set of allowed values is learned quickly and then does not change, over "poorly behaved" rules, where *r* grows steadily over time, indicating that future anomalies are likely . If we did not remove this rule, then we would continue to observe new values in testing and generate (probably false) alarms . This is depicted in Figure 1 below is derived from a paper by (Mahoney,2003)

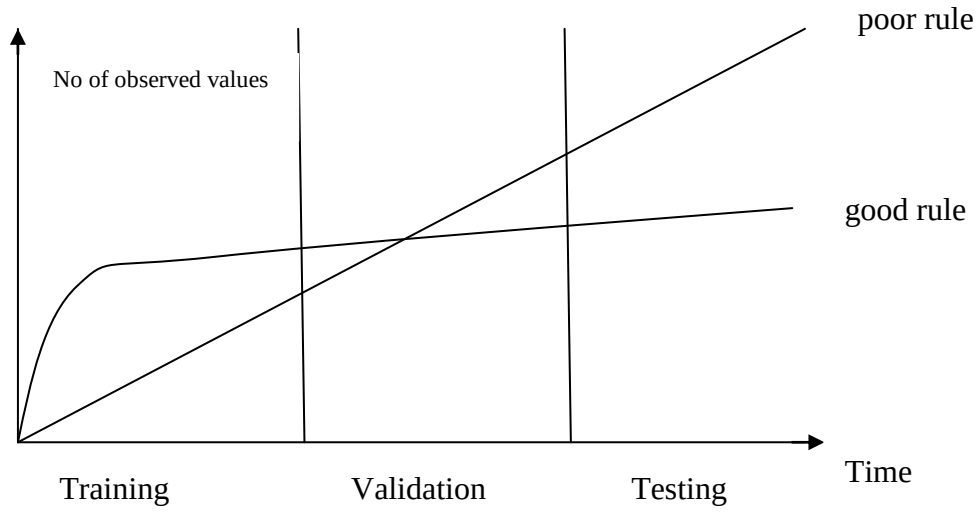


Figure 2 Growth r of good and poor rule, poor rules will be removed

The LERAD algorithm is basically a rule generating algorithm that uses the rules to fix a score to each TCP connection. Any deviation from the assigned score signifies an anomaly (Cheema at el, 2009).

2.4.2.3 Application Layer Anomaly Detection Algorithm

It works by checking for anomalies only in the traffic directed to the server with the assumption that attacks only originate from the client and are directed to the server. It distinguishes ports 0-1023 as the server ports and 1024 to 65535 as client ports.

In training it models the normal set of clients to for each host hence the number of clients allowed to access each service

It models the, source, and destination IP address, destination port, TCP flags and application layer keyword (Mahoney et al.,2004). These five attributes are modeled because they give better performance in detecting novel attacks. It works by assigning anomaly scores to each packet and assigns a score to each incoming server TCP connection (Mahoney et al., n.d.). TCP Connections are reassembled from packets. ALAD is configured knowing the IP addresses it is supposed to protect and it distinguishes server ports from client ports (Mahoney et al., n.d.). It maps:

- source address and destination address, where the source is the client computer making a request while the destination is the computer receiving the request and learns the normal clients for each host.
- source address, destination address and destination port , it creates a model for each server on each host and learns the number of clients for each server.
- destination IP and destination port this learns the normal set of server and the requests they receive, this would help if a client attempts to access a nonexistent port
- TCP Flags and destination port this learns the normal TCP sequences for TCP flag sequences for the first, next to last and last packet of connection FIN-ACK (request to close and acknowledge the previous packet) and ACK (to acknowledge the FIN).
- Keyword Destination port, this shows the key words to find the allowable keywords for each application layer protocol.

  The ALAD algorithm examines the first 1000 bytes for normal requests and for SMTP and HTTP it examines only the header address. Use of a rarely used word could signal an anomaly

2.4.2.4 Network Traffic Anomaly Detection Based on Bytes

It detects anomalies based on packet bytes. Traffic that is not of interest if filtered out .The basis of this is the assumption that attacks are normally targeted to the server hence traffic from the server to the host does not contain anomalies hence ignore and filtered out.

Only In coming traffic to the server is considered and only the first 48 bytes of the traffic are checked. Each of the first 48 bytes of the packet is treated like an attribute.

Table 1. Network Traffic Anomaly Detection Based on Bytes

Filtered traffic	**Reason**
Non IP packets (ARP, hub test, etc.)	Currently is working with IP packet header only
All outgoing traffic	Attacks emanate from clients directed to the server and not vice versa
All TCP connections starting with a SYN-ACK packet	indicating the connection was initiated by a local client. Normally, attacks are initiated remotely against a server
UDP to high numbered ports (>1023)	Normally this is to a client (e.g. a DNS resolver).
TCP starting after the first 100 bytes	A 4K hash table is used to store the starting TCP SYN sequence number for each source/destination address/port combination. There is a small amount of packet loss due to hash collisions.
Packets to any address/port/protocol combination (TCP, UDP, or ICMP) if more than 16 have been received in the last 60 seconds.	Again, a 4K hash table (without collision detection) is used to index a queue of the last 16 packet times. This limits packet floods.

This algorithm models 48 attributes which consists if the first 48 bytes of a packet staring with IP Header with each byte treated a nominal attribute with 256 possible values. attributes of the TCP packet are the 20 bytes of the IP header and 20 bytes of the TCP header and the first 8bytes of the application payload. If the packet is less than 48 bytes, the extra attributes are set to 0.

NETAD models, all IP Packets, all TCP Packets, all TCP SYN packets, TCP ACK packets, all TCP ACK packets to port 0-255, TCP ACK packets to port 21(FTP), TCP ACK to port 23 (TELNET), TCP ACK to port 25 (SMTP), TCP ACK to port 80 (HTTP).
Any anomaly occurring during training lead to the n(the number of training examples) being reset to back to 0. This is based on the assumption that the training data contains no attacks.

If a value occurs even once in training, its anomaly score is 0, to correct this a model, $t_i/(f_i + r/256)$, where t_i is the time (packet count in the modeled subset) since the value i (0-255) was last observed (in either training or testing), and f_i is the frequency in training, the number of times i was observed among training packets. Meaning ,the score is highest for values not seen for a long time (large t_i), and that occur rarely (small f_i). The term $r/256$ prevents division by 0 for novel values. It is preferred over a simple count offset (e.g. $t_i/(f_i + 1)$) because for novel events it reduces to a value that is large for small r. Thus, the NETAD anomaly score for a packet is
$\Sigma\ tna(1 - r/256)/r + t_i/(f_i + r/256)$ where the summation is over where the summation is over the $9 \times 48 = 432$ subset/attribute combinations.

The filtering aspect of the NETAD algorithm increases speed of processing since many of the attributes have been removed. Further it minimizes possibility of the false positives.

2.4.2.5 Payload Based Anomaly Detection Algorithm

These algorithms work in two phases; the training phase that involves profiling of normal behavior and storage of these profiles. The testing phase involves comparison of current network activity profile to that of the stored profiles and reports anything other than the normal profile, (Thorat et al.,n.d). The network payload does not have a fixed length or format,(Wang et al n.d). Due to the large size of the payload, a methodology of diving the size of the payload into smaller clusters needs to be devised.

An example of Payload based anomaly detection algorithm is PAYL algorithm that extracts 256 features from the payload (Perdisci et al. 2008). Payload in PAYL is considered anomalous if the payload under test and model of traffic exceeds pre determined threshold (Perdisci et al. 2008). PAYL works with unlabeled data and may suffer from high suffer false positive rate.

There are other anomaly detection algorithms that work based on different characteristics such as Pattern matching approach, implemented by (Maxion et al) and Histogram based anomaly detectors that use information gathered from traffic feature distributions e.g. distribution summary statistics such as entropy -quantitative measure of disorder in a system Daniela Brauckhoff, (2010).

Based on the assertion by Denning (1987,) that anomalies exhibit a behavior that is characteristically different from normal behavior. Anomaly detection algorithms, work by identifying normal behavior. This normal behavior identified is used as a threshold to compare the current identified behavior in the network and detect anomalies as those activities that do not match the identified normal behavior.

2.4.2.6 Ourmon algorithm

This is a near real time anomaly detection algorithm that detects attacks on the payload and packet header. It is both statistically flow collection oriented system. It is based on promiscuous packet collection on Ethernet interfaces . It works by using an probe for collecting packets deemed important and ignoring those that deemed to be noise. It works based on a notion of tuples focused on IP host addresses and layer 7 Internet relay Chat (IRC) channel. Hence ourmon can deal with both packet header and payload based anomalous traffic,(Binkley et al.,2005).

It defines its own internal format for flows allowing flows to be more efficient focusing on only gathering information of interest to the flow tuple of interest. It has IP and IRC flow tuples. Flow tuples based on IP address give a clue on what the host is doing and are useful in detecting anomalies. IRC based flow tuples help in detecting payload based anomalies

Data analysis uses Berkerly Packet filter (BKF) and various hashed top talker lists and displays this data using various graphic tools ad histograms producing data every 30 seconds and summarized report daily. The BKF filter is a row protocol independent socket interface to the data link layer that allows filtering of packets in a very granular fashion (Jeff Stebelton, n.d),

This algorithm works by collecting various features of the Internet Protocol (IP) packets using various item based top talker filters(top-N). Top talker filters are an algorithms of comparing

similarities and analyzing item by item to identify a set of items to be recommended,(Mukund Deshpande., n.d) and multiple instances of Berkerly packet Filter (BKF). These data is displayed using network visualization tools to display the resulting measurements to detect the anomalies. Ourmon algorithm consists of several filters . The BKF is used to collect unfiltered packets from an Ethernet device. The probe reads all the packets and subjects each packet to a set of configuration filters.

2.4.2.7 Dialogue Correlation Algorithm

This algorithm works by monitoring the two way communication between two internal networks for a sign of a bot or other malware. It tracks tow way communication flows and develops an audit trail of data exchanges that matches a state based infection sequence model. In the dialogue correlation Model, infections are modeled as a set of loosely ordered communication flows that are exchanged between internal hosts and external entities. Bots are modeled as sharing common underlying that occur during the infection life cycle which is; target scanning, infection exploit and finally binary egg download and execution and establishment of command and control channel. (Gu G et al,n.d).

These events are not mandatory for every botnet attack, however the dialogue correlation algorithms collects evidence trail of relevant infection events per host and looks for threshold combination of sequences that satisfies requirements for bot declaration.

This algorithm is implemented in the Bothunter Intrusion Detection system. That comprises of three intrusion detection systems, the snort, SCADE and SLADE . the snort is a signature based system however the SLADE and SCADE are anomaly based algorithms. These three IDS's produce alerts that are referred to as dialogue warning in this methodology. This is because the alarms are not processed as individual alerts but are used to drive a bot dialogue correlation analysis and the results used to capture and reports the actors and provide evidence trail of the complete infection. (Gu et al, n.d)

SLADE is an anomaly based bothunter plug-in that conducts byte distribution payload anomaly detection on inbound packet. It examines the payload of every request packet sent to the

monitored services and outputs an alert if its lossy n-gram frequency deviates from an established normal profile (Gu G et al, n.d)

SCADE is an anomaly detection engine that attempts to detect port scans used by malwares . SCADE tracks scans directed to targeted to internal hosts. it also scans ports for failed connection attempts (Gu G et al, n.d)

2.5 Hybrid Anomaly Detection Algorithms

There are two major categories of anomaly detection algorithms, the signature based and the anomaly based detection algorithms. However there has emerged a third category that uses both of these methodologies to improve on detection capabilities.

One example of these algorithms is SNORT (Mahoney et al) which is a signature based ADS however an anomaly based plug-in Statistical Packet Anomaly Detection Engine (SPADE), (Mahoney et al., 2006) has been developed to improve of capabilities of snort .

2.6 Hardware based network anomaly detection

Software based anomaly detection system mechanisms are the most common in detecting anomalies in network traffic, however there are hardware devices that can detect anomalies in traffic. Cisco Traffic Anomaly Detector Module for Cisco Catalyst 6500 switches and Cisco7500 routers is one such device, (Cisco systems, 2013). It uses behavioral analysis and attack recognition technology by compiling profiles of how individual devices behave under normal conditions and sends alerts to the operators if a potential attack is detected

Some attacks exploit vulnerability that has not been disclosed publicly, are known as Zero day attacks, Such attacks do not have known defense (Belge, et al, 2012)

2.7 Conceptual framework

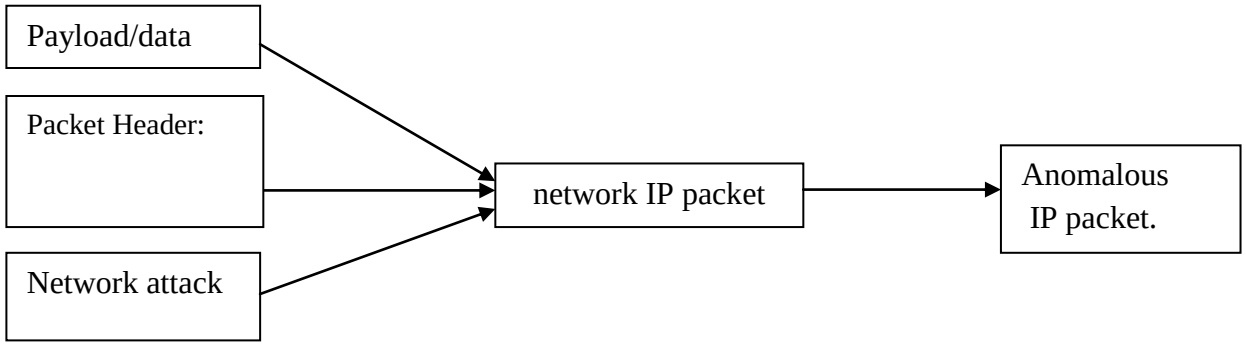


Figure 3 Conceptual Framework

Network attacks cause changes that Internet Protocol packet fields making them differ from normal network traffic packet. Based on this differing characteristics, anomaly detection systems can detect network attacks by comparing the packet header fields (Flags, source address, destination address, checksum) of a normal packet with other packets in network traffic.

CHAPTER THREE: METHODOLOGY

3.0 Introduction

The objective of this research is to evaluate a algorithms used in detecting attacks in a network based on characteristics of the packet header field by determining their accuracy, false positive ratio and their detection rate.

3.1 Research Design

This research involved use of tcpdump files that were used for training anomaly detection algorithms. These tcpdump data were used for training and testing the algorithms. The training data was attack free and the algorithms used it as bench mark data for use in detecting anomalies.

The TCP dump files for testing the algorithms was normal traffic data that had labeled network attacks that had been carefully identified in the dataset. The attacks detected by the algorithms were compared with the documentation of the attacks known to exist in the data set in order to determine whether they were detected correctly.

3.2 Data collection

3.2.1 Real Life Data

Tcpdump was used to capture live real life network traffic data that contained. Data capture was done in a network environment comprising of two computers interconnected to form cabled network running Microsoft Windows 8.0 and Windows 7.0 operating systems. The computer was running Windows 8.0 and virtualized SUSE Linux and Microsoft Windows XP operating system. TcpDump file was captured in the computer running Windows 8.0 operating system.

Third party software's ware used to generate attacks in the network traffic. This attacks were generated from computer address 192.168.1.4. To generate attacks, hynaeFE software was used to generate denial of service attacks and distributed denial of service attacks using TCP,UDP protocol. Port scan attacks were generated using Advanced Port Scanner Version 1.3.

3.2.2 Darpa 99' Data set

This tcpdump was data created by the MIT university's Lincoln Laboratory in a project funded by Department of Defense of the United States of America. It is was created for evaluation of anomaly detection algorithms. The dataset contain one week of attack free data used to train

anomaly detection algorithms and further two weeks of data containing labeled attacks that the algorithms will try and detect.

The DARPA 99 dataset was created in network consisting of four inside host computers running windows NT, SunOS, Solaris, Linux operating systems and a CISCO router. These networks were interconnected via a CISCO router and a gateway leading to almost 100 workstations and another for connecting thousands of websites. Data collection was via sniffers connected on the inside the network. Part of the traffic in this dataset is generated using network traffic generators.

The DARPA 1999 test data consists of over 190 instances of 57 attacks which included 37 Probes, 63 Denial of Service attacks, 53 Remote to Local attacks and 37 User to Remote data attacks generated for a duration of two weeks on weekdays from Monday to Friday (Lippmann, 2000).

This dataset was used because to determine the accuracy of these algorithms, we needed to know the total number of attacks that were contained in the test dataset for comparison with the results generated by the anomaly detection algorithms. This would also help in determining the true positive records count in the whole dataset.

3.3 Documents and archival records

Documentation on operations and configurations of the algorithms was reviewed so as to know how to use the algorithms and the type of anomalies that can be detected by the algorithms.

Details of network attacks being studied was reviewed in order to correctly identify the network attacks the eventual detection process

3.4 Data analysis methods

This research used data from tcpdump files. TCP dump files are streams of network traffic data captured as they are on transit in a network. To analyze this data, the anomaly detection algorithms was used to read through the tcpdump files. These algorithms reported anomalies if any were detected.

Unnecessary records generated by the algorithms and are not required during data analysis was removed. The fields required from the algorithms are;

- time of occurrence
- date of occurrence,

- IP address of the victim/source
- the anomaly score

Data on the attacks present in the test data set was so as to remain only with the relevant fields that was used to compare with the results generated from the algorithms .Any record from the results of the algorithms matching an entry from the labeled attacks list was marked with a TP entry (true positive). Results that do not match were marked as a false positive entry

3.5 Limitation of methodology

- Changes in configuration of the network and introduction of new equipment in the network can introduce anomalies in the network that might be mistaken for network attacks by the algorithms
- Network traffic generators were used to generate part of the traffic hence some of the addresses generated were not complete. The last octet of the network address identifying the host was missing for part of the traffic.

CHAPTER FOUR: DATA ANALYSIS RESULTS AND FINDINGS

4.1 Introduction

Several algorithms have been developed to detect anomalies. These algorithms detect and produce a text file containing a list of all anomalies detected These anomalies could be as a results of new equipment being introduced into the network or due to changes in configuration of both hard ware or software or could be due to an attack in progress such .These anomaly detection algorithm have a high false positive ratio and the need to be evaluated on their ability to detect anomalies .

4.2 Evaluation of the algorithms

Evaluation of this algorithms entailed calculation of false positive ratio these are situations where the algorithms detect an anomaly when actually there is none present. Detection rate, this is the number of true detections that have detected divided by the total number of detections made. Accuracy, this is the number of correct detection of attacks compared to the total number of known attacks known to exist in the dataset under review.

4.2.1 Algorithms

The following algorithms were evaluated :

- Packet Header Anomaly detection algorithm (PHAD)
- Learning rules for Anomaly detection (LERAD)
- Network Traffic Anomaly Detection Based on Packet Bytes (NETAD)
- Application Layer Anomaly Detection (ALAD)

4.2.2 Training data

These algorithms work with static data however they need to be exposed to a training data that is attack free. Training data was collected in a network protected with a firewall. Care was taken to ensure that data for training the algorithms was attack free. Computers to be used in this research project were disconnected from removed from the Local Area Network and the Internet and thoroughly scanned to ensure they were attack free. Traffic from this attack free network was used to create training data. Tcpdump was used to capture network packets.

The dump files comprises of data captured from Monday through to Friday. The training data totals up to 2.7 Gb of tcpdump data. This data was collected from a network comprising of several computers running Windows XP, Linux and Sun Solaries operating systems.

Real life training data consists of 1.2 Gb worth of training data. the data was collected as discussed in section 3.2.1. The data was attack free and was collected from two real computers running three virtualized computer operating systems. Tcpdump was running on a virtualized Suse linux operating system. All computers (virtualized and real) have independent IP addresses for communication as follows.

Table 2. Real life computer network

Computer Operating system	Address	Type
Windows 8	192.168.1.11	Physical computer
Windows XP	192.168.1.4	Virtualized- is source of network attacks
Suse linux	192.168.1.10	Virtualized - Victim Computer
Windows 7	192.168.1.2	Physical computer
Suse Linux	192.168.1.3	Virtualized

4.2.3 Attack data

The DARPA 99' dataset, contains 4.6gb worth of dump data collected over a duration of two weeks. This data contains labeled network attacks (attacks that have been identified and their location and identity is known) however on day of data file is missing from the data dump files available .

The real life data contains, attacks that were generated using a virtualized computer running Windows XP operating system. The computer generated attacks directed to the virtualized linux box(192.168.1.10). The victim computer was running *tcpdump* software to capture live network traffic. *HynaeFE* and *Advanced Port Scanner version 1.3* were used to generate port scan and TCP,ICMP and ARP based attacks. Only one computer was generating network attacks (192.168.1.4). All the computers except the one generating attacks were running *ping command* directed towards the victim computer.

Third party network generating software were used due to the fact that the existing network works in a firewall protected environment. Hence chances of getting attacks were minimized . Further real life data is not labeled. To determine being measured in this research, the correct number of attacks in the whole data set needs to be determined before hand for comparison with the results of the anomaly detection .

4.3 Data Analysis

Analysis of the results from the algorithms involved;

Changing the output of the anomaly detection algorithms into a flat file/text file. This data was imported into a spreadsheet (Microsoft Excel). The data from the algorithms was tab delimited meaning each category occupied its own column. Unnecessary records/columns were deleted so as to remain with the time, date, victim address, anomalous score and the filed contributing to the score.

This data was imported into an SQL server database and a column called *status* was added to the database table. A computer program was created in Visual Basic 6.0 to compare contents of the evaluation truth table (contains pre identified labeled attacks) with the results from the algorithm. In case a record contained in the database matched an entry in the evaluation truth table, an entry of TP (True positive) was updated into the status column of the table containing results of the algorithms. The table columns were victim IP address, date, and time, anomalous field and score.

Records in the database that did not match entries in the truth evaluation table were marked with FP (False Positive). According to the DARPA data set instructions, for an attack to be correctly identified, the victim address and date need to be identified to within a duration sixty (60) seconds .

The Visual Basic program included queries for counting the number of records whose status was updated to either true positive or false negative.

4.4 Packet Header Anomaly Detection (PHAD) algorithm results.

6351 records were detected from the whole DARPA dataset with a score ranging from 0.000371 to 0.748198. The tables below show the results from the algorithms being tested.

Table 3. The table below indicates the top 10 scoring records detection and their classifications.

key	Date	Time	Victim Address	Score	Most anomalous attribute	Status	
P2469	04/06/99	8:59:16	172.016.112.194	0.748198	TCP Checksum=x6F4A 67%		FP
P421	04/01/99	8:26:16	172.016.114.050	0.697083	IP Frag Ptr=x2000 100%	TP	
P512	04/01/99	11:00:01	172.016.112.100	0.689305	TCP URG Ptr=49 100%	TP	
P92	03/31/99	8:00:28	192.168.001.030	0.664309	IP TOS=x20 100%		FP
P277	03/31/99	11:35:13	000.000.000.000	0.664225	Ether Src Hi=xC66973 68%		FP
P279	03/31/99	11:35:18	172.016.114.050	0.653956	Ether Dest Hi=xE78D76 57%		FP
P2886	04/08/99	8:01:20	172.016.113.050	0.644237	IP Frag Ptr=x2000 35%		FP
P1821	04/05/99	8:39:50	172.016.112.050	0.634027	IP Frag Ptr=x2000 100%		FP
P2454	04/05/99	20:00:27	172.016.113.050	0.628749	UDP Checksum=x90EF 100%		FP
P2034	04/05/99	11:45:27	172.016.112.100	0.626234	TCP URG Ptr=49 100%	TP	

Table 4. The table below indicates the least 10 scoring records detection and their classifications.

Key	Date	Time	Victim Address	Score	Most anomalous attribute	Status	
P4585	04/09/99	0:37:31	207.136.086.223	0.001768	IP Src=172.016.118.010 100%		FP
P4411	04/08/99	22:46:43	207.136.086.223	0.001768	IP Src=172.016.118.010 100%		FP
P3325	04/08/99	10:39:50	172.016.112.050	0.001289	TCP Dest Port=514 50%		FP
P6132	04/09/99	15:02:04	172.016.112.100	0.000589	TCP Window Sz=17099 100%		FP
P2461	04/06/99	5:36:47	192.168.001.020	0.000556	UDP Len=31 100%		FP
P6241	04/09/99	19:56:31	172.016.116.194	0.000531	TCP Dest Port=3544 100%		FP
P6240	04/09/99	19:56:31	207.121.184.081	0.000519	TCP Src Port=3544 100%		FP
P3743	04/08/99	13:59:03	172.016.114.148	0.000419	TCP Seq=18446744072770717897		FP
P468	04/01/99	9:01:33	172.016.114.207	0.000419	TCP Seq=1431242622 100%		FP
P2868	04/07/99	23:39:44	172.016.112.050	0.000371	UDP Dest Port=37933 100%		FP

Table 5. below Top ten (10) highest True positive Scoring records detected and their classification

Key	Date	Time	Victim Address	Score	Most anomalous attribute	Status	
P421	04/01/99	8:26:16	172.016.114.050	0.697083	IP Frag Ptr=x2000 100%	TP	
P512	04/01/99	11:00:01	172.016.112.100	0.689305	TCP URG Ptr=49 100%	TP	
P2034	04/05/99	11:45:27	172.016.112.100	0.626234	TCP URG Ptr=49 100%	TP	
P19	03/29/99	11:15:08	192.168.001.001	0.615613	TCP Flg UAPRSF=x01 100%	TP	
P513	04/01/99	11:00:01	172.016.112.100	0.545068	TCP Flg UAPRSF=x39 100%	TP	
P2035	04/05/99	11:45:27	172.016.112.100	0.392407	TCP Flg UAPRSF=x39 100%	TP	
P239	03/31/99	10:13:13	172.016.113.050	0.375782	IP Src=172.016.118.060 100%	TP	
P2843	04/07/99	13:46:35	172.016.112.050	0.362795	TCP Flg UAPRSF=x01 100%	TP	
P1754	04/02/99	8:45:14	172.016.112.050	0.353835	IP Src=001.012.120.006 71%	TP	
P378	03/31/99	18:29:12	172.016.112.100	0.335414	ICMP Checksum=x0000 51%	TP	

Table 6. Least ten (10) True Positive PHAD true positive detections

Key	Date	Time	Victim Address	Score	Most anomalous attribute	Status	
P2777	04/06/99	20:57:08	172.016.112.100	0.189132	TCP URG Ptr=245 82%	TP	
P516	04/01/99	11:00:22	172.016.112.100	0.189132	TCP URG Ptr=245 82%	TP	
P24	03/29/99	11:18:09	192.168.001.001	0.187491	TCP Flg UAPRSF=x01 96%	TP	
P25	03/29/99	11:19:09	192.168.001.001	0.186667	TCP Flg UAPRSF=x01 96%	TP	
P21	03/29/99	11:16:08	192.168.001.001	0.186667	TCP Flg UAPRSF=x01 96%	TP	
P23	03/29/99	11:17:08	192.168.001.001	0.186667	TCP Flg UAPRSF=x01 96%	TP	
P4652	04/09/99	8:01:26	206.048.044.050	0.179109	TCP Dest Port=50460 100%	TP	
P13	03/29/99	9:36:20	172.016.114.050	0.174434	IP Src=202.077.162.213 100%	TP	
P515	04/01/99	11:00:10	172.016.112.100	0.159029	TCP URG Ptr=245 82%	TP	
P2037	04/05/99	11:45:36	172.016.112.100	0.159029	TCP URG Ptr=245 82%	TP	

In the DARPA 99 dataset, the attacks were directed to the 172.016 network with several attacks such as *portsweep ,IPsweep, UDF storms*, *dosnuke* attacks among others since this was a synthetic dataset created for evaluation of datasets, majority of this attacks were directed to that part of the network.

False positive ratio

these are detections made by the algorithm yet non exist dived by the normal classification. The algorithm detected a total of 6303 records that did not have a matching entries in the truth evaluation table. In this case the false positives ratio is

$$\frac{6303}{6351} = 0.9923$$

According to the tables above, the scores with the highest anomaly score did not have the true positive detection of attacks in the network traffic. Similarly the score of the attribute that contributes most to the anomaly score percentage doe not imply that it is a true positive.

Accuracy

Accuracy of the algorithms is a percentage of the correct detections to the total number of correct known attacks that exit in the truth table. After removing external attacks from the truth evaluation table, a total of 150 attacks were identified classified as inside attacks. The algorithm detected a total of 48 attacks with matching entries in the truth and evaluation table.

the accuracy was

$$\frac{48}{150} = 0.32$$

Detection rate

The detection rate of this algorithm is the ratio of the true positive detection to the total number of detections made. Number of correct detections from the algorithms were 48. The total number of detection were 6351 records

$$\frac{48}{6351} = 7.5578 \times 10^{-3}$$

4.5 Network Anomaly Detection Algorithm results

62457 records were detected from the whole DARPA dataset with a score ranging from 0.004697 to 1.096395

Out of these detections only 23 records matched attacks from in the detection truth table. hence only 23 true positive detections.

Table 7. Top 10 true positive detection by NETAD

Date	Time	Victim Address	Score	Anomalous Attribute	Status
04/05/99	16:46:20	172.016.114.050	0.813702	SA1=76,72% SA0=46,28%	TP
04/08/99	17:01:08	172.016.112.100	0.788382	SA3=CF,59% SA2=88,32%	TP
04/05/99	16:46:27	172.016.114.050	0.705067	SA1=76,24% SA0=46,74%	TP
04/08/99	12:04:37	172.016.112.100	0.645258	C4=3A,76% C5=5C,24%	TP
04/08/99	17:01:12	172.016.112.100	0.640949	SA3=CF,58% SA2=88,32%	TP
04/08/99	17:01:15	172.016.112.100	0.63764	SA3=CF,58% SA2=88,32%	TP
04/08/99	17:01:14	172.016.112.100	0.624071	SA3=CF,58% SA2=88,32%	TP
04/05/99	16:46:26	172.016.114.050	0.618494	IPlen0=CB,15% C0=60,85%	TP
04/08/99	17:01:06	172.016.112.050	0.597179	SA3=CF,45% SA2=88,24% SA1=56,20% SA0=DF,11%	TP
04/08/99	17:01:06	172.016.112.010	0.590459	TCPchk1=20,100%	TP

Table 8. least 10 true positive detections by score NETAD

Date	Time	Victim Address	Score	Anomalous Attribute	Status
04/08/99	17:01:07	172.016.112.100	0.559503	SA3=CF,45% SA2=88,24% SA1=56,20% SA0=DF,11%	TP
04/08/99	12:04:17	172.016.112.100	0.550924	SA3=D0,42% SA2=F0,23% SA1=7C,20% SA0=53,14%	TP
04/08/99	17:01:18	172.016.112.100	0.536471	SA3=CF,57% SA2=88,31%	TP
04/06/99	10:19:01	172.016.112.050	0.525951	Seq2=D8,14% Seq1=B4,29% Ack3=7A,11%	TP
04/08/99	17:01:07	172.016.112.100	0.523293	SA3=CF,35% SA2=88,20% SA1=56,17% SA0=DF,17%	TP
04/08/99	17:01:09	172.016.112.100	0.515084	SA3=CF,56% SA2=88,31%	TP
04/08/99	17:01:08	172.016.112.100	0.51378	SA3=CF,58% SA2=88,32%	TP
04/08/99	17:01:11	172.016.112.100	0.511595	SA3=CF,42% SA2=88,25% SA1=56,20% SA0=DF,13%	TP
04/08/99	17:01:17	172.016.112.100	0.49671	SA3=CF,57% SA2=88,31%	TP
03/29/99	9:15:43	172.016.113.050	0.486136	ID1=34,16% Seq2=3E,25% TCPchk1=BE,26%	TP

Table 9. Top 10 False Negative detections by score NETAD

Date	Time	Victim Address	Score	Anomalous Attribute	Status
04/01/99	3:26:16	172.016.114.050	1.096395	Frag0=03,100%	Fp
04/06/99	4:59:16	172.016.112.194	1.093415	Frag1=20,33% TCPhdr=00,67%	Fp
03/29/99	4:15:01	172.016.113.050	1.08217	IPlen0=28,16% TCPhdr=50,16% C0=00,16% C1=00,16% C2=0	Fp
04/09/99	13:27:16	172.016.114.050	1.077881	C0=4D,12% C1=41,12% C2=49,12% C3=4C,12% C4=20,12% C5	Fp
04/06/99	4:59:16	172.016.112.194	1.075755	Frag0=02,100%	Fp
04/07/99	6:26:08	172.016.114.050	1.073058	IPlen1=05,12% C1=45,12% C2=54,12% C3=20,12% C4=2F,12	Fp
04/01/99	7:51:19	172.016.113.084	1.068059	TOS=D0,100%	Fp
03/31/99	13:29:12	172.016.112.100	1.066355	IPlen1=02,96%	Fp
04/01/99	3:26:16	172.016.114.050	1.06615	Frag1=20,100%	Fp
04/09/99	10:32:17	172.016.113.050	1.054935	IPlen0=28,16% TCPhdr=50,16% C0=00,16% C1=00,16% C2=0	Fp

Table 10. Bottom 10 False positive detections by score NETAD

Date	Time	Victim Address	Score	Anomalous Attribute	Status
03/29/99	3:02:33	172.016.255.255	0.06658	IPchk1=A9,83%	Fp
03/29/99	3:01:58	172.016.255.255	0.06515	ID1=10,74% IPchk1=62,23%	Fp
03/29/99	3:02:03	172.016.255.255	0.059996	ID0=CD,12% IPchk1=AC,76% IPchk0=B7,12%	Fp
03/29/99	3:02:33	172.016.118.255	0.058965	IPchk1=32,96%	Fp
03/29/99	3:01:20	172.016.112.020	0.053122	IPchk0=70,20% Seq1=BD,34% Seq0=2D,25%	Fp
03/29/99	3:02:23	172.016.112.020	0.041178	ID0=19,27% IPchk0=7B,37%	Fp
03/29/99	3:01:47	192.168.001.010	0.038981	IPlen0=44,16% ID0=74,10% IPchk0=5E,14% Seq2=30,16% S	Fp
03/29/99	3:01:18	172.016.112.010	0.037747	ID0=A1,19% IPchk0=C1,15% Seq1=E3,29% Seq0=D9,23% Urg	Fp
03/29/99	3:02:03	192.168.001.010	0.035254	IPlen0=80,15% ID0=CC,21% Seq2=6C,14% Seq1=A2,19% Seq	Fp
03/29/99	3:00:03	172.016.255.255	0.004697	ID0=46,49% IPchk1=B1,31% IPchk0=3E,19%	fp

false positive

62,457 records were found after scan by Network Anomaly Detection Algorithm. Records that were detected by the algorithm but did not have a matching entry in the evaluation truth table were 62434 records.

False positive ration is the ratio of the detections that were classified as anomalous yet they were not divided by the total number of detections in the datasets.

$$\frac{62434}{62457} \; 0.999631 \text{ or } 99.9\ \%$$

Accuracy

this is the ratio of the true positive detections to the number of detection existing attack in the dataset. this is the number of attacks as found in the truth table. Only 23 records had matching entries in the evaluation truth table . The accuracy was.

$$\frac{23}{150} = 0.15 \text{ or } 15\%$$

detection rate

This is the ratio of the total number of correct detections divided by the total number of detections in the whole dataset. A total of 62457 records were detected by the algorithm.

A total of 23 records were detected by the algorithm and had a matching entry in the truth evaluation table

$$\frac{23}{62457} = 3.682533583105176\text{e-}4$$

4.6 Learning Rules for Anomaly Detection algorithm results

24754 detections were made by this algorithm in the DARPA 99' dataset ranging from 0.711 to 4.2753. However 8 of these were true positive. 24746 were false positive.

Table 11. LERAD true positive results

Date	Time	IP Address	Score	Anomaly	Status
04/05/99	16:46:27	172.016.114.050	2.116485	(60.68) DA1=114 DA0=050 DP?=113 F1=.S	TP
04/08/99	17:01:11	172.016.112.100	0.90674	(30.26) W1=.^@GET W3?=.^@^@^@^@^@^@^@^@	TP
04/08/99	17:01:12	172.016.112.100	0.480244	(26.93) W1=.^@GET W3?=.^@^@^@^@^@^@^@^@	TP
04/08/99	17:01:07	172.016.112.100	0.404767	(55.22) DP?=110 F1=.S	TP
04/08/99	17:01:06	172.016.112.050	0.121138	(31.69) SA0?=223 F2=.AP	TP
04/08/99	17:01:07	172.016.112.100	0.044969	(27.3) DP?=143 DUR=0 F1=.S	TP
04/08/99	17:01:07	172.016.112.100	0.042047	(25.46) DP?=110 F1=.S	TP
04/08/99	17:01:12	172.016.112.100	0.005574	(20.09) W1=.^@GET W3?=.^@^@^@^@^@^@^@^@	TP

12. LERAD Top ten false positive detections

Date	Time	IP Address	Score	Anomaly	Status
04/07/99	4:39:43	172.016.114.050	4.945093	(61.04) W1?=.GET W3=.HTTP/1.0^M^	fp
04/06/99	17:32:45	172.016.118.100	4.774303	(99.99) DA1?=118 DA0=100	fp
04/08/99	15:58:59	172.016.114.148	4.752043	(77.94) DA1=114 DA0?=148 DP=80 DUR=0	fp
04/06/99	17:15:42	172.016.114.148	4.429058	(100) DA0?=148 DP=80	fp
04/01/99	6:45:29	172.016.118.040	4.364142	(75.95) DA1?=118	fp
04/07/99	4:58:14	172.016.114.050	4.343149	(40.67) W1=.^@GET W3?=.	fp
04/09/99	13:27:07	172.016.114.050	4.332811	(97.33) DP=25 W1?=.^@MAIL	fp
04/01/99	6:00:01	172.016.112.100	4.324902	(39.95) DUR=0 F1=.S F2?=.UAP	fp
04/08/99	7:42:26	172.016.112.100	4.323141	(50.54) DUR=0 F1=.S F2?=.A	fp

false positive ratio

False positive ratio is the ratio of the detections that were classified as anomalous yet they contained no attacks as per the truth evaluation table, divided by the total number of detections in the dataset. A total of 24,746 records were detected that did not have matching entry in the truth evaluation table.

$$\frac{24746}{24754} = 0.99968$$

Accuracy

this is the ratio of the true positive detections to the number of detection existing attack in the dataset. this is the number of attacks as found in the truth table. Only 8 records had matching entries in the evaluation truth table . The accuracy was.

8 records were found to be true positive

$$\frac{8}{150} = 0.053$$

Detection Rate

This is the ratio of the total number of correct detections divided by the total number of detections in the whole dataset. A total of 24,746 records were detected by the algorithm.

A total of 8 records were detected by the algorithm and had a matching entry in the truth evaluation table

This is the ratio of the total number of detections correct divided by the total number of detections in the whole dataset.

$$\frac{8}{24754} = 3.23218 \times 10^{-4}$$

4.7 Application Layer Anomaly Detection Algorithm

This algorithm generated a total of 886 records, of which only 2 records had a match in the detection truth table.

Table 13. ALAD True Positive results

Date	Time	IP address	Anomaly	Anomalous Attribute	Status
04/05/99	16:46:27	172.016.114.050	0.829879	113=5458,25 To=172.016.114.050:113 172.016.114.050=172.016.118.070	Tp
04/08/99	17:01:11	172.016.112.100	0.637247	172.016.112.100=207.136.086.223 172.016.112.100:80=207.136.086.223	Tp

Table 14. False Positive results

Date	Time	IP address	Anomaly	Anomalous Attribute	Status
04/06/99	8:59:16	172.016.112.194	0.748198	TCP Checksum=x6F4A 67%	Fp
04/01/99	8:26:16	172.016.114.050	0.697083	IP Frag Ptr=x2000 100%	Tp
04/01/99	11:00:01	172.016.112.100	0.689305	TCP URG Ptr=49 100%	Tp
03/31/99	8:00:28	192.168.001.030	0.664309	IP TOS=x20 100%	Fp
03/31/99	11:35:13	000.000.000.000	0.664225	Ether Src Hi=xC66973 68%	Fp
03/31/99	11:35:18	172.016.114.050	0.653956	Ether Dest Hi=xE78D76 57%	Fp
04/08/99	8:01:20	172.016.113.050	0.644237	IP Frag Ptr=x2000 35%	Fp
04/05/99	8:39:50	172.016.112.050	0.634027	IP Frag Ptr=x2000 100%	Fp
04/05/99	20:00:27	172.016.113.050	0.628749	UDP Checksum=x90EF 100%	Fp

False positive ratio

False positive ratio is the ratio of the detections that were classified as anomalous yet they contained no attacks as per the truth evaluation table, divided by the total number of detections in the dataset. A total of 886 records were detected that did not have matching entry in the Detection truth evaluation table.

$\frac{884}{886}$=0.9977 or 99.77%

Accuracy

This is the ratio of the true positive detections to the number of detection existing attack in the dataset. this is the number of attacks as found in the truth table. Only 2 records had matching entries in the evaluation truth table . The accuracy was.

$\frac{2}{150}$=0.013 or 1.3 %

Detection Rate

This is the ratio of the total number of correct detections divided by the total number of detections in the whole dataset. A total of 886 records were detected by the algorithm.

A total of 2 records were detected by the algorithm and had a matching entry in the truth evaluation table

This is the ratio of the total number of detections correct divided by the total number of detections in the whole dataset.

$\frac{2}{886}$ =0.00265

CHAPTER FIVE: SUMMARY AND CONCLUSIONS

5.1 Summary

Algorithm	Anomalies detected	False Positive Ratio	Accuracy	Detection rate
PHAD	6,351	6,303 (99.2%)	48 (32 %)	7.55786×10^{-3}
NETAD	62,457	62,434 (99.9%)	23 (15%)	3.6825×10^{-4}
LERAD	24,754	24,746 (99.9%)	8 (5.3%)	3.23218×10^{-4}
ALAD	886	884 (99.7%)	2(1.3%)	0.00265

Table 15 Comparison of results from evaluation of algorihms

Based on false positive ratio, PHAD performed slightly better than the all the other algorithms. All the algorithms had a very high false positive ratio of 99% . PHAD outperformed the other algorithms by 0.7% it clocked a ratio of 99.2% compared to the NETAD and LERAD both of which scored 99.7%. ALAD false positive ratio was 99.7 % similar to LERAD

Based on accuracy PHAD was the best performing algorithm with an accuracy of 32 % or 48 detections out of a possible 150 followed by NETAD with an accuracy of 15% or 23 detections out of a possible 150 attacks, while LERAD had the worst performance of 5.3 % out of a possible 150 attacks. ALAD detected only 2 attacks out of a possible 150 attacks and had an accuracy of 1.5%

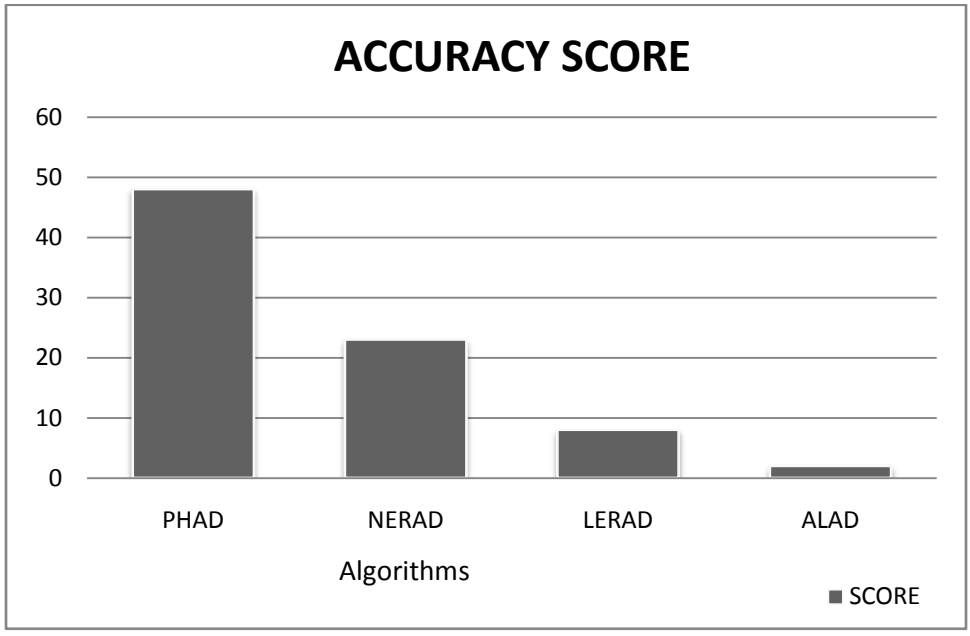


Figure 4. Bar Graph showing accuracy of algorithms

Detection rate is a ratio of the total number of true positive detections to the total number of detections. On this metrics, PHAD again became the best scoring algorithm though with very low score of 7.55786 $x10^{-3}$, LERAD algorithm outperform NERAD algorithm with paltry margin, they score 3.23218 $x10^{-4}$ and 3. 6825 x 10^{-4} respectively. ALAD scored 0.00265 or 0.265%.

In terms of anomalies generated, the NETAD generated the highest amount of anomalies with 62,457 records generated, despite using a network traffic filter mechanism that filters out all traffic coming from the victim computer. LERAD comes second with a score of 24,754 while PHAD has the least score of 6351.

PHAD algorithm models 33 attributes of the IP packet header and scored better that all the algorithms in terms of the metrics measure, out performs the other two algorithms, in all metrics measured in this research,

The DARPA 99' dataset contains labeled attacks, some of these attacks do not have complete address, the 8 bits of the destination address are missing. this affects the final results as these would be increase the false positive ration and reduce the accuracy of the final results.

However in terms of accuracy, according to cheema et.al (2009), LERAD algorithm performs best defeating NERAD which was amongst six (6) other algorithms evaluated. HAD was not amongst the algorithms evaluated in the paper by Cheema et al (2009). However Cheema et al (2009), agrees that the algorithms have a very low accuracy and a very high false positive ratio. An attempt to find some of the algorithms evaluated (Threshold Random Walk, Maximum Entropy Estimation and Virus Throttle) by Cheema proved futile.

5.2 Conclusion

Zero day/novel attacks have no known signatures that can be relied upon to detect them. however these attacks bring about changes to various fields in the packet header fields. Dorothy Dennings, (1987) proposed that 'Network anomalies have characteristics that differ from those of network traffic'. True to these comments network attacks are detectable and can be detected by changes they bring about to normal traffic packet field attributes among other factors that can be used.

To detect anomalies, anomaly detection algorithms need to be trained so that they can learn about the network traffic packet attributes. This creates the basis for comparing the network traffic for anomalies.

However while these attacks can be detected, high false positive ratio and low accuracy, make adoption and implementation of these algorithms unviable both in the commercial and real life scenarios.

Anomalies in network traffic originate from several sources such as ; introduction of new equipment in the network , malfunctioning equipment in the network, activities of users such as backup activities and malicious activities both on the inside and outside of the network.

None of the algorithms was able to work with real life data captured in a real life network scenario that contained attack free training data and simulated network attacks generated through third party attack generating tools captured using tcpdump software.

5.3 Recommendations

1. Data sets available for evaluation of anomaly detection algorithms need to be updated to keep up with the changing face of network attacks. The only publicly available is the DARPA 99' dataset created by the Lincoln Laboratory of Massachusetts Institute of Technology in a project funded by the Defense Advanced Research Project Agency (DARPA).
2. the DARPA 99 dataset contains simulated data part of which was generated using network traffic generators. Part of this data contains is incomplete network address for several devices. The address of the victim computer does not have an address meaning that if this could be an attack it might be classified among the false positives.
3. Poor sharing of information by organizations and developers of algorithms makes it difficult to evaluate the algorithms. Many papers have been published discussing the anomaly detection algorithms, however these algorithms are not public available hence testing them is impossible

5.4 Areas for further research

The algorithms have a high false positive ratio and poor accuracy, to overcome this challenge, we recommend merging of these algorithms , instead of being used in isolation so as to improve their ability to detect network attacks as is done in *snort* and *spade.*

REFERENCES

1. Bilge et al (2012), Before We Knew It, An Empirical Study of Zero-Day Attacks in the Real World , pg1.
2. Binkley.J, Massey B (2005), Ourmon and Network Monitoring Performance, Portland State University, pg 96-98
3. Brauckhoff D (2010)., Network Traffic Anomaly Detection and Evaluation, A dissertation submitted to ETH ZURICH, pg22
4. Chandola.V, Banerjee.A, Kumar.V,(2008) Anomaly detection: A Survey University of Minnesota, Pg2.
5. Cheema F.M, Akram A, Zeshan Iqbal Z (2009), Comparative Evaluation of Header vs. Payload based Network Anomaly Detectors, pg1.
6. Ciza Thomas, Vishwas Sharma, N. Balakrishnan, (2012) Usefulness of DARPA Dataset for Intrusion Detection System Evaluation, Indian Institute of Science, Bangalore, India p,g.2
7. Communications commission of Kenya," annual report 2014"
8. Eric Cole, (2005)et al , Network Security Bible, pg 476
9. D.E Denning,(1987) D.E. Denning: An Intrusion Detection Model. In IEEE Transaction on Software Engineering, .Deshpande M, Karypis G.,(n.d), Item-Based Top-N Recommendation Algorithms, Dept. of Computer Science & Engineering, University of Minnesota,pg3
10. Jeff Stebelton,(n.d), "Berkerly Packet Filters- The Basics.pg1
11. Geer et al. (1997), "System Security: A Management Perspective.". The Usenix Association.
12. Gu.Y, McCallum.A, and Towsley.D.(2005), Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation, Department of Computer Science, University of Massachusetts, n.d., pg1
13. Gu. G, Porras P, Yegneswaran.V, Fong,M, Lee. W,(2007), BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation, pg 1-7.
14. George Jones, (2012) Introduction to Anomaly Detection ,Software Engineering Institute, Carnegie Mellon University

15. Huang.H, Al-Azzawi.H, and Brani.H.,(2014)" Network Traffic Anomaly Detection" ,Las Cruces, arXiv:1402.0856v1 (cs.CR) NM, USA Klipsch School of Electrical and Computer Engineering, New Mexico State University, n.d., pg1
16. Nortcutt.s, et al (2005)., Inside Network Perimeter Security
17. Mahoney.M, Chan .P, (2008), Learning Models of Network Traffic for Detecting Novel Attacks, Florida Institute of Technology Technical Report, pg 4,5
18. Mahoney.M, Chan .P (2008),Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks ,Department of Computer Sciences, Florida Institute of Technology, pg3.
19. Mahoney.M, Chan .P, (2003), Network Traffic Anomaly Detection Based on Packet Bytes,Florida Institute of Technology, Pg1.
20. Ramah.K, Ayari. H, Kamoun.F ,(2010) "Traffic Anomaly Detection and Characterization in the Tunisian National University Network" . CRISTAL laboratory École Nationale des Sciences de l'Informatique , pg1-4,6-7.
21. Richard Lipmann,(2000), MIT Lincoln Library, Summery and Plans for the 1999 DARPA evaluation ,pg.5
22. Thottan.M and Ji.C.(August 2003) Anomaly Detection in IP Networks, IEEE Transactions on Signal Processing, Vol. 51, No. 8, pg2
23. Oxford English Dictionary (1999), low priced, advanced learners edition, pg 41
24. Perdiski R, Fogla D,Giacinto G, Lee W, McPAD (November 2008) : A Multiple Classifier System for Accurate Payload-based Anomaly Detection, pg3
25. Lazarevic.A, Ertoz. L,Kumar.V,Ozgur.A, Srivastava.J, (2003) A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection, SIAM Journals, pg1,
26. Jyothsna, V. V. Rama Prasad, K., Prasad M,(August, 2011) A Review of Anomaly based Intrusion Detection Systems. International Journal of Computer Applications (0975 – 8887) Volume 28– No.7,pg1
27. Tan.K.C.M, Maxion R.A (November, 2004) Performance Evaluation of Anomaly-Based Detection Mechanisms, Technical Report Series CS-TR-870 University of Newcastle upon Tyne, Pg3.

APPENDICES

APPENDIX 1:PROJECT PLANNING AND MANAGEMENT

Activity	June	July	August	September	October	November	December
1.Research Title Submission							
Proposal preparation							
Literature review							
Proposal writing							
Proposal Submission							
Proposal presentation							
2.Data collection							
Data Collection Instruments							
Collect Data							
Study of the network							
3 Data analysis							
Computer system							
Presentation of results and findings							
Writing report							
4.Final presentation							

Table 16 Project Plan

APPENDIX II: BUDGET

Table 17 Budget

	Item Description	**Qty**	**Unit Cost(Shillings)**	**Total Cost(Shillings)**
1	Laptop	1	50,000	50,000
	Software			
	MS Office	1	10,000	10,000
	Ms SQL server(developer edition)	1	0	0
	TCP Dump	1	0	0
	MS Windows 7 professional	1	30,000	30,000
	Suse Linux 11	1	20,000	20,000
	C programming language	1	15,000	20,000
	Visual Studio	1	15,000	15,000
2	Storage device	2	2,000	4,000
3	Internet modem	1	2,500	5,000
4	Data bundles		10,000	10,000
5	Stationery		5,000	7,000
6	Printing, Photocopying & binding		10,000	10,000
7	Transport		4,000	4,000
8	Miscellaneous		5,000	5,000
	Total			**185,000**

APPENDIX III: SCREEN SHOTS

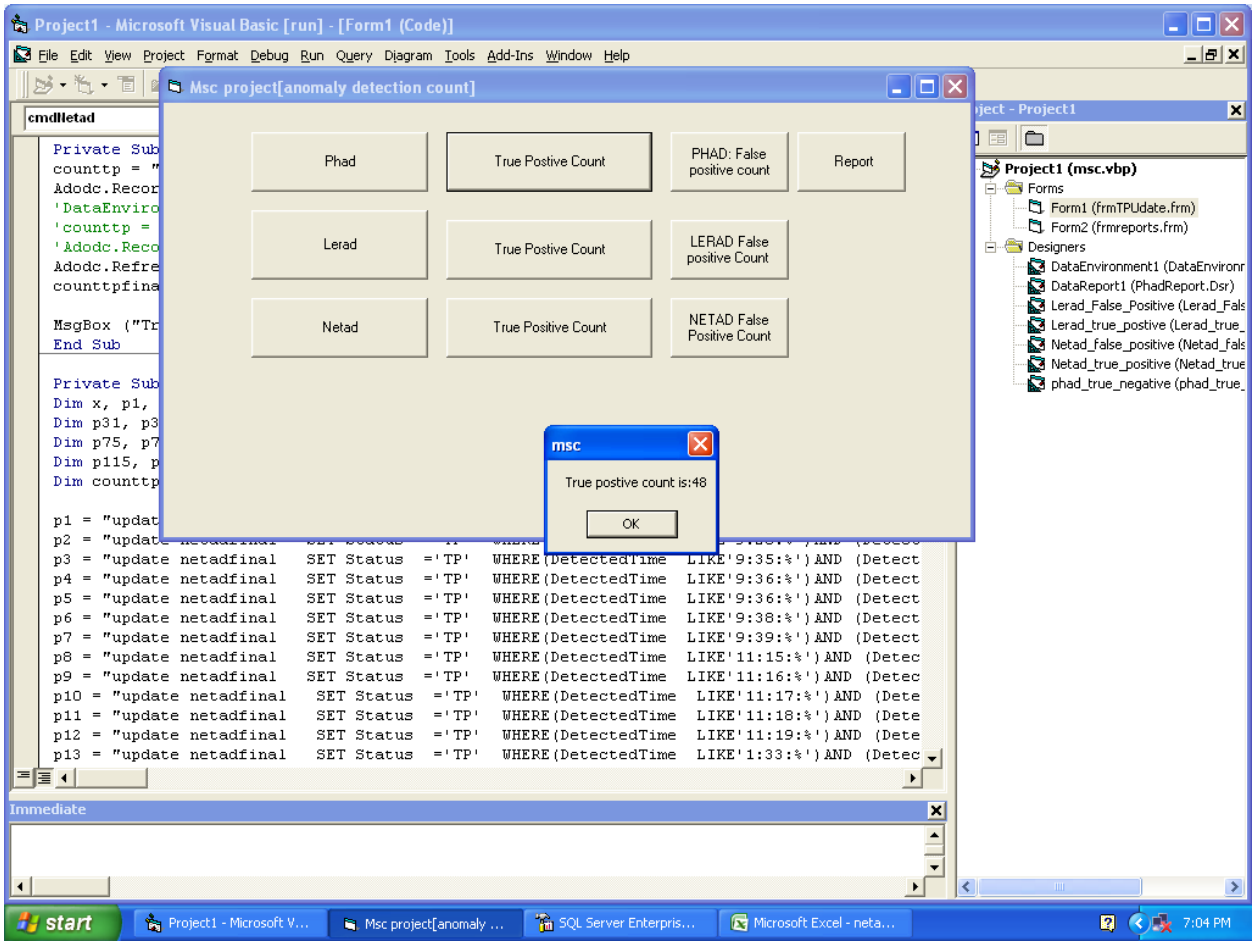


Figure 5 Count matching True Positive entries in SQL Table

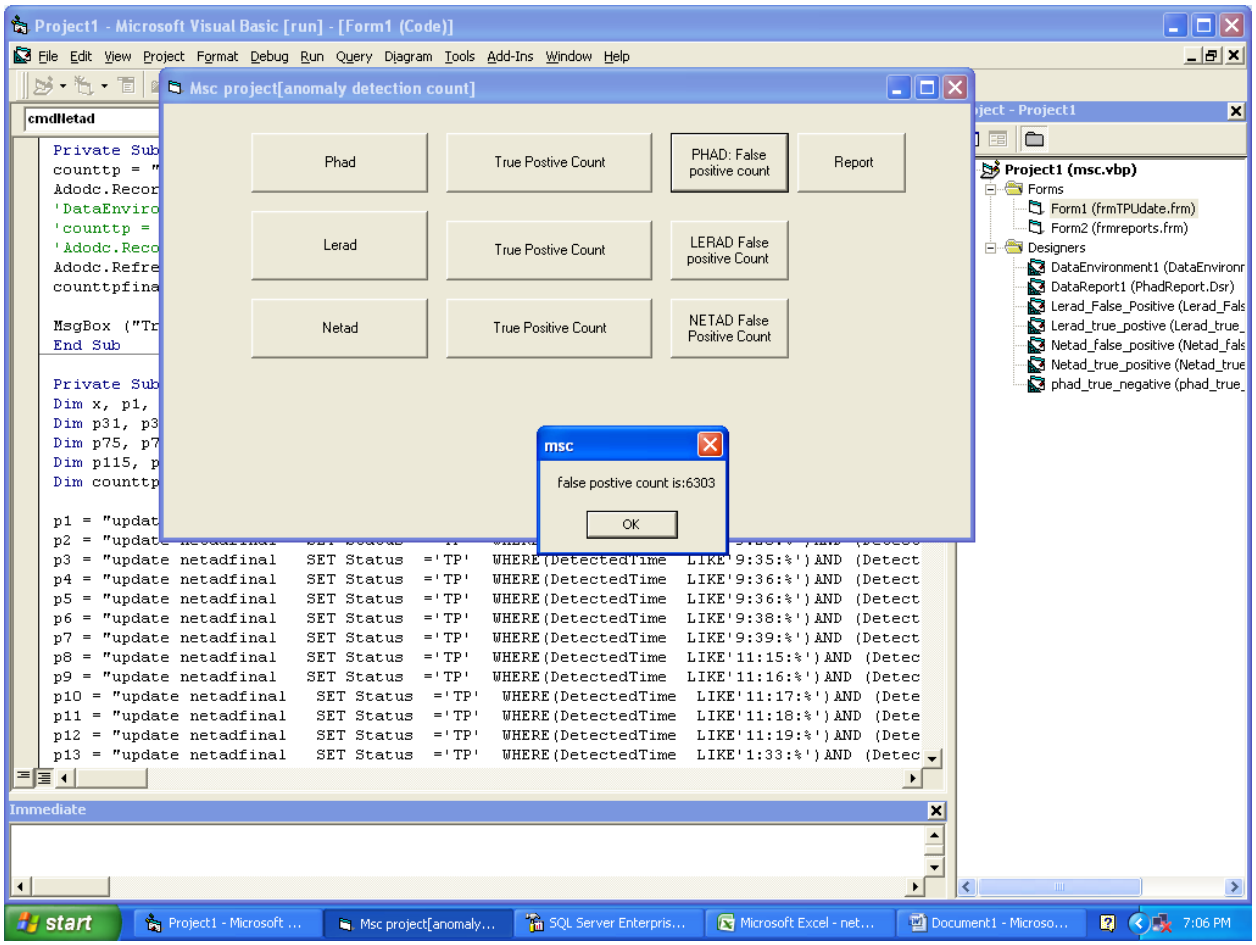


Figure 6 Count False Positive entries in SQL Table

SAMPLE CODE

```
Private Sub Command1_Click()
Dim a, b, c, d, e, f, detection_algorithm, detection_status As String
a = "select * from phadfinal order by score where status = 'TP'desc "
b = "select * from phadfinal order by score where status = "asc "
c = "select * from netadfinal order by score where status = 'TP'desc"
d = "select * from netadfinal order by score where status = "asc "
e = "select * from leradfinal order by score where status = "asc "
f = "select * from aladfinal order by score where status = "asc "

Select Case detection_algorithm = cboalgorithms.Text And detection_status = cboptions.Text
Case cboalgorithms.Text = "Phad" And cboptions.Text = "true positive"
Form1.Adodc.RecordSource = a

DataEnvironment1.Connection1.
DataReport1.Show

Case cboalgorithms.Text = "Phad" And cboptions.Text = "False Negative"
Form1.Adodc.RecordSource = b

Case cboalgorithms.Text = "netad" And cboptions.Text = "true positive"
Form1.Adodc.RecordSource = c

Case cboalgorithms.Text = "netad" And cboptions.Text = "false negative"
Form1.Adodc.RecordSource = d

Case cboalgorithms.Text = "leradfinal" And cboptions.Text = """
Form1.Adodc.RecordSource = e

Case cboalgorithms.Text = "alad" And cboptions.Text = "false negative"
Form1.Adodc.RecordSource = f
End Select

End Sub
Private Sub Command2_Click()

DataReport1.Show
End Sub
```

```
Private Sub Command3_Click()
phad_true_negative.Show
End Sub

Private Sub Command4_Click()
Netad_true_positive.Show
End Sub

  Private Sub Command5_Click()
  Lerad_false_positive.Show
End Sub

Private Sub True_negative_report_Click()
  Netad_false_positive.Show
End Sub

Private Sub True_positive_report_Click()
   Lerad_true_postive.Show
End Sub
```

SAMPLE C CODE

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <stdlib.h>
#include <stdarg.h>

void phad();
void netad();
void lerad();
void alad();

void prt();

int main(int argc,char **argv)
{
char choice,repeat,y,Y;
int selection;
printf("\n ANOMALY DETECTION ALGORITHMS, \n\n");
printf("1:Packet Header Anomaly Detection:-PHAD-Algorithm\n");
printf("2:Network Anomaly Detection-NETAD:- Algorithm\n");
printf("3:Learning Rules for Anomaly Detection:-LERAD-Algorithm \n\n");
printf("4:Application Layer Anomaly Detection:-ALAD-Algorithm \n\n");

printf("Please enter 1: for Phad or 2: for netad or 3: for lerad algorithms and 4:for Alad and
5  to exit: \n");

scanf("%d",&choice);
if (choice==1){
  phad();
 printf("select another algorithm:\n") ;
scanf("%d",&choice);
          }
if (choice==2){
netad();
printf("select another algorithm:\n") ;
scanf("%d",&choice);
          }
/*switch(choice) {
case 1:
phad();
printf("select another algorithm:\n") ;
scanf("%d",&choice);
}
```

```
*/
if (choice==3){
lerad();
printf("select another algorithm:\n") ;
scanf("%d",&choice);
        }

if (choice==4){
alad();
printf("Final algorithm, press 5 to exit:\n") ;
scanf("%d",&choice);

        }
else if (choice == 5){
printf("You have selected to exit:\n\n");
printf("Thank you, good bye:\n");
        }
return 0;
        }

        void phad(){
printf("you have selected to run PHAD Algorithm\n");
char command[50];
strcpy(command,"./phad 1123200 in3* in4* in5*");
system(command);
        }

        void netad(){
printf("you have selected to run NERAD Algorithm\n");
char command[50];
strcpy(command,"./netad in3tf in45tf");
system(command);
        }

void lerad(){
printf("you have selected to run LERAD Algorithm\n");
char command[50];
strcpy(command,"./lerad train.txt test.txt 0");
system(command);
        }

void alad(){
printf("you have selected to run ALAD Algorithm\n");
char command[50];
strcpy(command,"perl alad.pl train test");
system(command);}
```