



Sudan University of Science and Technology
College of Graduate Studies



An Intrusion Detection System Using Artificial Neural Network Based on Intrusion Behavior

**نظام كشف التسلل باستخدام الشبكة العصبية الاصطناعية
على أساس سلوك التسلل**

**A Thesis Submitted as Partial Fulfillment of Requirement for M.Sc.
Electronic Engineering (Computer Engineering and Network)**

Prepared by:

ZAHRAA MOHAMMED ADAM OSMAN

Supervised by:

Dr. FATH ELRAHMAN ISMEAL KHALIFA

February, 2018

الآية

قال تعالى:

﴿وَاتَّقُوا اللَّهَ وَيَعْلَمَ اللَّهُ عَلَيْهِ كُلَّ شَيْءٍ عَلِيمٌ﴾

صدق الله العظيم

سورة البقره الآية: ﴿٢٨٢﴾

DEDICATION

TO

My Family who always Support and guided me to the right path.

My Doctors who give me the keys to lead successful life.

My Friends who encouraged me during my life.

My Great Supervisor.

THANKS,,

ACKNOWLEDGEMENT

I would like to thank my supervisor, Dr. FATH EIRAHMAN ISMEAL KHALIFA for the patient guidance, encouragement and advice he has provided throughout me. I have been extremely lucky to have a supervisor who cared so much about my work. I would also like to thank Dr. ELSADIG SAEID GEBREEL who helped me in my research. Finally, I would like to thank all the members of staff at Sudan University of Science and Technology, And everyone helped me to complete this work.

Abstract

Over the past several years, the Internet environment has become more complex and untrusted. Enterprise networked systems are inevitably exposed to the increasing threats posed by hackers as well as malicious users internal to a network. IDS technology is one of the important tools used now-a-days, to counter such threats. The goal of intrusion detection is to identify unauthorized use, misuse and abuse of computer system insiders and outsiders penetrators. Various IDS techniques has been proposed, which identifies and alarms for such threats or attacks.

The thesis proposes design and implement an intrusion detection system based on Artificial neural network to provide the potential and classify network activity based on KDD dataset. The performance of the classification algorithms was evaluated by computing the percentages of Sensitivity(SE), Specificity(SP), Accuracy(AC) and Mathews Correlation Coefficient(MCC). It was found that the system is capable of detecting with a sensitivity of 83.1% and the accuracy is about 75%. Results show system that can detect new types of attacks with fairly accurate results.

المستخلص

على مدى السنوات الماضية اصبحت بيئة الانترنت اكثر تعقيدا وغير موثوق بها وبالتالي تتعرض انظمة شبكات المؤسسات للتهديدات المتزايدة التي تتشكل في المخترقين والمستخدمين الخبيثين. نظام الكشف عن التسلل هو التقنية الوحيدة المستخدمة في هذه الايام لمواجهة هذه التهديدات. تم اقتراح تقنيات الكشف عن التسلل التي تعمل على التحديد والتنبيهات لهذه الهجمات او التهديدات. الهدف من نظام الكشف عن التسلل هو تحديد الاستخدام غير المصرح به وسوء الاستخدام وسوء استخدام المتواجدين داخل و خارج النظام. تقترح الاطروحة تصميم وتنفيذ نظام كشف التسلل اعتمادا على الشبكات العصبية الاصطناعية لتوفير إمكانيات تصنيف نشاط الشبكة اعتمادا ع قاعدة بيانات. تم تقييم خوارزمية التصنيف بحساب النسب المئوية للحساسية والخصوصية ومعامل ارتباط ماثيو. وجد ان النظام قادر على الكشف مع حساسية 83% ودقة حوالي 75%. من النتائج المتحصل عليها يتضح ان النظام لدية امكانية الكشف عن انواع جديدة من الهجمات مع نتائج دقيقة حد ما.

Table of Contents Table of Contents

| | |
|---|------|
| الاية..... | I |
| DEDICATION..... | II |
| ACKNOWLEDGEMENT..... | III |
| Abstract..... | IV |
| المستخلص..... | V |
| Table of Contents..... | VI |
| List of Table..... | X |
| List of Figure..... | XI |
| List of Sample..... | XII |
| Abbreviation..... | XIII |
| CHAPTER ONE..... | 1 |
| 1.1 Preface..... | 2 |
| 1.2 Problem Statement..... | 3 |
| 1.3 Proposed Solution..... | 4 |
| 1.4 Aim and Objectives..... | 4 |
| 1.5 Methodology..... | 4 |
| 1.6 Thesis Outlines..... | 4 |
| CHAPTER TWO..... | 5 |
| 2.1 Intrusion Detection System..... | 6 |
| 2.1.1 Comparison Between IDS and Firewalls..... | 6 |
| 2.2 Classification of Intrusion Detection System..... | 7 |
| 2.2.1 Network Intrusion Detection System..... | 7 |
| 2.2.2 Host Intrusion Detection Systems..... | 8 |
| 2.3 Intrusion Detection Method..... | 8 |
| 2.3.1 Signature-based Method..... | 8 |

| | |
|--|----|
| 2.3.2 Anomaly-based Method..... | 8 |
| 2.4 Intrusion Prevention Techniques..... | 9 |
| 2.4.1 Classification of Intrusion Prevention..... | 10 |
| 2.5 Intrusion Prevention Methods..... | 10 |
| 2.5.1 Statistical Anomaly- based Detection..... | 11 |
| 2.5.2 Tasteful Protocol Analysis Detection..... | 11 |
| 2.5.3 Limitation of Intrusion Detection..... | 11 |
| 2.6 Evasion Techniques of Intrusion Detection..... | 12 |
| 2.7 Intrusion Detection and Behavior..... | 13 |
| 2.8 Publicized Threats to Security..... | 13 |
| 2.9 Intrusion Behavior Examples..... | 14 |
| 2.10 Intrusion Behavior Patterns..... | 15 |
| 2.11 Insider Attacks..... | 16 |
| 2.11.1 Internal Threat..... | 17 |
| 2.11.2 Cybercrimes..... | 18 |
| 2.11.3 Internet Criminals Enterprise..... | 19 |
| 2.12 Related work..... | 19 |
| CHAPTER THREE SYSTEM DESINE..... | 20 |
| 3.1 Proposed System Block Diagram..... | 21 |
| 3.2 Proposed Features for Detection..... | 22 |
| 3.2.1 Measures used for Intrusion Detection..... | 22 |
| 3.2.2 Command or Program Execution Activity Program resource utilization Mean and Standard deviation..... | 22 |
| 3.2.3 File Access Activity..... | 23 |
| 3.3 Artificial neural network..... | 23 |
| 3.3.1 Characteristics of Artificial Neural Networks..... | 24 |

| | |
|---|----|
| 3.4 Application of Artificial neural network..... | 27 |
| 3.4.1 Neural Networks in Practice..... | 29 |
| 3.5 System Model..... | 30 |
| 3.5.1 The features that is covered in this system Sample..... | 31 |
| 3.5.1.1 Elapsed Time..... | 32 |
| 3.5.1.2 Incorrect Password..... | 33 |
| 3.5.1.3 Higher Privileges Detection..... | 34 |
| 3.5.1.4 Abnormal Read/Write Activities..... | 35 |
| 3.5.1.5 Abnormal Processor Usage..... | 36 |
| 3.5.1.6 Unauthorized File Access..... | 37 |
| 3.5.1.7 Intrusion Detection System Flow Chart..... | 38 |
| CHAPTER FOUR..... | 39 |
| 4.1 Simulation Scenario..... | 40 |
| 4.2 Sample Dataset..... | 40 |
| 4.3 Main Program Screen..... | 41 |
| 4.4 Running the Program..... | 41 |
| 4.5 Measures of Performance..... | 41 |
| 4.6 Accuracy of Performance..... | 43 |
| 4.7 Summery..... | 45 |
| CHAPTER FIVE..... | 46 |
| 5.1 Conclusion..... | 47 |
| 5.2 Recommendation..... | 47 |
| References..... | 48 |
| Appendix..... | 51 |
| The Intrusion Detection System Code..... | 51 |

List of Tables

| | |
|---|----|
| 4.1 Sample Dataset File Anomaly Detection..... | 39 |
| 4.2 Examining the Sensitivity and Accuracy of the Software..... | 42 |
| 4.3 Dataset Testing Results among four QOS..... | 42 |
| 4.4 Compared the System Accuracy with other Developed System..... | 43 |

List of Figures

| | |
|--|----|
| 3.1 Proposed Intrusion Detection System Block Diagram..... | 21 |
| 3.2 Artificial Neural Network..... | 24 |
| 3.3 Structure Artificial Neural Networks..... | 25 |
| 3.4 Artificial Neural Networks Processing Unit..... | 25 |
| 3.5 General Activation Function Neural Networks..... | 26 |
| 3.6 Neural Network Activation Function..... | 27 |
| 3.7 Elapsed Time Flowchart..... | 29 |
| 3.8 Incorrect Password Attempts Flowchart..... | 30 |
| 3.9 Higher Privileges Detection Flowchart..... | 31 |
| 3.10 Abnormal Read/Write Activities Flowchart..... | 32 |
| 3.11 Unauthorized File Access Flowchart..... | 33 |
| 3.12 Abnormal Processor Usage Flowchart..... | 34 |
| 3.13 Intrusion Detection System Flowchart..... | 35 |
| 4.1 Sample Dataset File..... | 39 |
| 4.2 Borland Delphi Main Program screen..... | 40 |
| 4.3 Running the Tool on Real Dataset..... | 41 |
| 4.4 Chart that Represent the Detection while Running the Date Set..... | 44 |

List of Sample

| | |
|--|----|
| (W_{ji}) synaptic weights..... | 27 |
| (I_i) Neural Network input..... | 27 |
| (O_i) Neural Network output..... | 27 |
| (Θ_j) external threshold, offset or bias..... | 27 |
| (\sum) N-Ary Summation..... | 42 |
| (T_P) Counts of all Samples which are Correctly called by the algorithm as being Cancer..... | 42 |
| (F_P) Counts of all Samples which are Incorrectly called by the algorithm as being Cancer while they are normal..... | 42 |
| (T_N) Counts of all Samples which are Correctly called by the algorithm as being normal..... | 42 |
| (F_N) Count of all Samples which are Incorrectly called by the algorithm as being normal while they are cancer..... | 42 |

Abbreviation

| | |
|--------------|---|
| ANN | Artificial Neural Network |
| AC | Accuracy |
| CRC | Cyclic Redundancy Check |
| CERTs | Computer Emergency Response Time |
| DDOS | Distributed Denial of Service |
| DIDS | Distributed Intrusion Detection System |
| FP | False Positive |
| FN | False Positive |
| HIDS | Host Intrusion Detection System |
| HIPS | Host-based Intrusion Prevention |
| IDS | Intrusion Detection System |
| IPS | Intrusion Prevention System |
| IDPS | Intrusion Detection and Prevention System |
| ISOA | Information Security Officers Assistants |
| KDD | Knowledge Discovery and Dataset |
| MCC | Mathews Correlation Coefficient |
| NIDS | Network Intrusion Detection System |
| NAT | Network Address Translation |
| NBM | Network Behavior Analysis |
| NSM | Network Security Monitor |
| SIEM | Security Information and Event Management |
| SE | Sensitivity |
| TCP | Transmission Control Protocol |
| TIM | Time-based Inductive |
| TP | True Positive |
| TN | True Negative |
| UEBA | User and Entity Behavior Analytics |
| WIPS | Wireless Intrusion Prevention Sys |

Chapter One

Introduction

Chapter One

Introduction

1.1 Preface

Because of the increasing dependence which companies and government agencies have on their computer networks, the importance of protecting these systems from attack is critical. There are numerous methods of responding to a network intrusion, but they all require the accurate and timely identification of the attack [1].

The timely and accurate detection of computer and network system intrusions has always been an elusive goal for system administrators and information security researchers. The individual creativity of attackers, the wide range of computer hardware and operating systems, and the ever changing nature of the overall threat to target systems have contributed to the difficulty in effectively identifying intrusions[1].

There are two general categories of attacks which intrusion detection technologies attempt to identify - anomaly detection and misuse detection.

Anomaly detection identifies activities that vary from established patterns for users, or groups of user[2]. Misuse detection involves the comparison of a user's activities with the known behaviors of attackers attempting to penetrate a system.

Nowadays, cloud computing is known by more and more people due to its advantages such as high scalability, high flexibility and low operational cost[1]. Cloud service users usually do not need to know how the cloud based software or platform runs; instead, they only need to send the requests to the cloud provider and then wait for the results, which is a much easier and more efficient way to access the needed computing

resources [1]. However, there are several issues for the current cloud platforms.

According to [2], security issues such as information leakage, unreliable data and unauthorized access are the most concerned problems by the majority of cloud users. Other issues such as stable operations, support systems and user friendliness have received less attention. To address the security problem with the cloud, it is a natural choice to deploy a distributed IDS system on the cloud to protect the virtual machines (VMs) and virtual networks against potential attacks. The major issue with such a choice is that the IDS could overload some busy nodes in the cloud and slow down the detection efficiency if no special arrangements are made. On the one hand, the IDS should not use too many resources to affect the performance of the major computing tasks and detect attacks efficiently. Therefore, it is desirable to equip the distributed IDS with the flexibility feature in that it can dynamically adjust its architecture based on the real time resource usage information across the cloud. Moreover, it is important for the IDS system to be capable of detecting unknown (new) attacks in the cloud. Thus, a balance needs to be achieved to satisfy cloud customers as well as provide the reasonable performance of intrusion detection simultaneously.

1.2 Problem Statement

Misuse detection is the process of attempting to identify instances of network attacks by comparing current activity against the expected actions of an intruder. Most current approaches to misuse detection involve the use of rule-based expert systems to identify indications of known attacks. However, these techniques are less successful in identifying attacks which vary from expected patterns.

1.3 Proposed Solution

This Thesis proposed to use artificial neural networks to provide the potential to identify and classify network activity based on limited, incomplete, and nonlinear data sources.

1.4 Aim and Objectives

The main aim of this research is to make use of the Artificial Neural Networks as a method for intrusion detection. Specific objectives are:

- To design and implement an Intrusion detection system based on Artificial Neural Networks concepts.
- To use Data set to train the ANN.
- To detect and calculate sensitivity and accuracy of the system.
- To specify a network security system that can accurately identify intruders

1.5 Methodology

After study and analysis to the detection methods of intrusion detection systems, We collect data about intrusion detection and select some of intrusion behaviors. Design intrusion detection tool using neural network and training of the neural network will be done to classify sample dataset to detect positive false, negative false, positive true and negative true, after the classification a software was developed based on Borland Delphi has the capability to identify the type, process a behavior. Running the system and calculate the accuracy of the detection.

1.6 Thesis Outlines

The thesis contains five chapters, chapter two includes a literature review of the intrusion detection and the grows of this technology, chapter three explaining the methodology of the research including the tools and requirements used, chapter four the simulation and the analysis of the results chapter five includes the conclusion and recommendations are written also the future work.

Chapter Two

Literature Review

Chapter two

Literature Review

2.1 Intrusion Detection System

An intrusion detection system (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any detected activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system[6]. A SIEM system combines outputs from multiple sources, and uses alarm filtering techniques to distinguish malicious activity from false alarms. There is a wide spectrum of IDS, varying from antivirus software to hierarchical systems that monitor the traffic of an entire backbone network. The most common classifications are network intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS). A system that monitors important operating system files is an example of a HIDS, while a system that analyzes incoming network traffic is an of a NIDS. It is also possible to classify IDS by detection approach: the most ted intrusions. Systems with response capabilities are typically referred to as an intrusion prevention system[6].

2.1.1 Comparison Between IDS and Firewalls

Though they both relate to network security, an IDS differs from a firewall in that a firewall looks outwardly for intrusions in order to stop them from happening. Firewalls limit access between networks to prevent intrusion and do not signal an attack from inside the network. An IDS evaluates a suspected intrusion once it has taken place and signals an

alarm. An IDS also watches for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks, and taking action to alert operators. A system that terminates connections is called an intrusion prevention system, and is another form of an application layer firewall[6].

2.2 Classification of Intrusion Detection System

IDS can be classified by where detection takes place (network or host) and the detection method that is employed[7].

2.2.1 Network Intrusion Detection Systems

Network intrusion detection systems (NIDS) are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. It performs an analysis of passing traffic on the entire subnet, and matches the traffic that is passed on the subnets to the library of known attacks. Once an attack is identified, or abnormal behavior is sensed, the alert can be sent to the administrator. An example of an NIDS would be installing it on the subnet where firewalls are located in order to see if someone is trying to break into the firewall. Ideally one would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall speed of the network. OPNET and NETSIM are commonly used tools for simulation network intrusion detection systems. NID Systems are also capable of comparing signatures for similar packets to link and drop harmful detected packets which have a signature matching the records in the NIDS. When classifying the design of the NIDS according to the system interactivity property, there are two types: on-line and off-line NIDS, often referred to as inline and tap mode, respectively. On-line NIDS deals with the network in real time[8]. It analyses the Ethernet packets and applies some rules, to decide if it is an attack or not. Off-line NIDS deals

with stored data and passes it through some processes to decide if it is an attack or not [9].

2.2.2 Host Intrusion Detection Systems

Host intrusion detection systems (HIDS) run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator if suspicious activity is detected. It takes a snapshot of existing system files and matches it to the previous snapshot. If the critical system files were modified or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission critical machines, which are not expected to change their configurations [9]. Intrusion detection systems can also be system-specific using custom tools and honeypots.

2.3 Intrusion Detection Method

2.3.1 Signature-based

Signature-based IDS refers to the detection of attacks by looking for specific patterns, such as byte sequences in network traffic, or known malicious instruction sequences used by malware [10]. This terminology originates from anti-virus software, which refers to these detected patterns as signatures. Although signature-based IDS can easily detect known attacks, it is impossible to detect new attacks, for which no pattern is available.

2.3.2 Anomaly-based

Anomaly-based intrusion detection systems were primarily introduced to detect unknown attacks, in part due to the rapid development of malware. The basic approach is to use machine learning to create a model of trustworthy activity, and then compare new behavior against this model. Although this approach enables the detection of previously unknown attacks, it may suffer from false positives: previously unknown legitimate activity may also be classified as

malicious [11]. New types of what could be called anomaly-based intrusion detection systems are being viewed by Gartner as User and Entity Behavior Analytics (UEBA)[5]. (an evolution of the user behavior analytics category) and network traffic analysis (NTA)[4]. In particular, NTA deals with malicious insiders as well as targeted external attacks that have compromised a user machine or account. Gartner has noted that some organizations have opted for NTA over more traditional IDS [11].

2.4 Intrusion Prevention

Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSES for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPSES have become a necessary addition to the security infrastructure of nearly every organization [12]. IDPSES typically record information related to observed events, notify security administrators of important observed events and produce reports. Many IDPSES can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g. reconfiguring a firewall) or changing the attack's content [11]. Intrusion prevention systems (IPS), also known as intrusion detection and prevention systems (IDPS), are network security appliances that monitor network or system activities for malicious activity. The main functions of intrusion prevention systems are to identify malicious activity, log information about this activity, report it and attempt to block or stop it [11][12].

Intrusion prevention systems are considered extensions of intrusion detection systems because they both monitor network traffic and/or system activities for malicious activity. The main differences are, unlike intrusion detection systems, intrusion prevention systems are placed in-line and are able to actively prevent or block intrusions that are detected [13]. IPS can take such actions as sending an alarm, dropping detected malicious packets, resetting a connection or blocking traffic from the offending IP address. An IPS also can correct cyclic redundancy check (CRC) errors, defragment packet streams, mitigate TCP sequencing issues, and clean up unwanted transport and network layer options [13].

2.4.1 Classification of Intrusion Prevention

Intrusion prevention systems can be classified into four different types:[7][12]. Network-based intrusion prevention system (NIPS): monitors the entire network for suspicious traffic by analyzing protocol activity. Wireless intrusion prevention systems (WIPS): monitor a wireless network for suspicious traffic by analyzing wireless networking protocols. Network behavior analysis (NBA): examines network traffic to identify threats that generate unusual traffic flows, such as distributed denial of service (DDOS) attacks, certain forms of malware and policy violations. Host-based intrusion prevention system (HIPS): an installed software package which monitors a single host for suspicious activity by analyzing events occurring within that host [14].

2.5 Intrusion Prevention Methods

The majority of intrusion prevention systems utilize one of three detection methods: signature-based, statistical anomaly-based, and Tasteful protocol analysis [14]. Signature-Based Detection: Signature based IDS monitor's packets in the Network and compares with pre-configured and pre-determined attack patterns known as signatures.

2.5.1 Statistical anomaly-based detection

An IDS which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline will identify what is "normal" for that network – what sort of bandwidth is generally used and what protocols are used. It may however, raise a False Positive alarm for legitimate use of bandwidth if the baselines are not intelligently configured[14].

2.5.2 Tasteful Protocol Analysis Detection

This method identifies deviations of protocol states by comparing observed events with "predetermined profiles of generally accepted definitions of benign activity"[9].

2.5.3 Limitations of Intrusion Detection

Noise can severely limit an intrusion detection system's effectiveness. Bad packets generated from software bugs, corrupt DNS data, and local packets that escaped can create a significantly high false-alarm rate[15].

It is not uncommon for the number of real attacks to be far below the number of false-alarms. Number of real attacks is often so far below the number of false-alarms that the real attacks are often missed and ignored[15].

Many attacks are geared for specific versions of software that are usually outdated. A constantly changing library of signatures is needed to mitigate threats. Outdated signature databases can leave the IDS vulnerable to newer strategies [15]. For signature-based IDSES there will be lag between a new threat discovery and its signature being applied to the IDS. During this lag time the IDS will be unable to identify the threat. It cannot compensate for a weak identification and authentication mechanisms or for weaknesses in network protocols. When an attacker gains access due to weak authentication mechanism then IDS cannot prevent the adversary from any malpractice. Encrypted packets are not

processed by most intrusion detection devices. Therefore, the encrypted packet can allow an intrusion to the network that is undiscovered until more significant network intrusions have occurred. Intrusion detection software provides information based on the network address that is associated with the IP packet that is sent into the network. This is beneficial if the network address contained in the IP packet is accurate. However, the address that is contained in the IP packet could be faked or scrambled. Due to the nature of NIDS systems, and the need for them to analyze protocols as they are captured, NIDS systems can be susceptible to same protocol based attacks that network hosts may be vulnerable. Invalid data and TCP/IP stack attacks may cause an NIDS to crash [16].

2.6 Evasion Techniques of Intrusion Detection

There are a number of techniques which attackers are using, the following are considered 'simple' measures which can be taken to evade IDS

- **Fragmentation:** by sending fragmented packets, the attacker will be under the radar and can easily bypass the detection system's ability to detect the attack signature.
- **Avoiding defaults:** The TCP port utilized by a protocol does not always provide an indication to the protocol which is being transported. For example, an IDS may expect to detect a Trojan on port 12345. If an attacker had reconfigured it to use a different port the IDS may not be able to detect the presence of the Trojan.
- **Coordinated, low-bandwidth attacks:** coordinating a scan among numerous attackers (or agents) and allocating different ports or hosts to different attackers makes it difficult for the IDS to correlate the captured packets and deduce that a network scan is in progress.

- **Address Spoofing/Proxy:** attackers can increase the difficulty of the ability of Security Administrators to determine the source of the attack by using poorly secured or incorrectly configured proxy servers to bounce an attack. If the source is spoofed and bounced by a server then it makes it very difficult for IDS to detect the origin of the attack.
- **Pattern change evasion:** IDSs generally rely on 'pattern matching' to detect an attack. By changing the data used in the attack slightly, it may be possible to evade detection. For example, an Internet Message Access Protocol (IMAP) server may be vulnerable to a buffer overflow, and an IDS is able to detect the attack signature of 10 common attack tools. By modifying the payload sent by the tool, so that it does not resemble the data that the IDS expects, it may be possible to evade detection[17].

2.7 Intrusion Detection and Behavior

Unauthorized intrusion into a computer system or network is one of the most serious threats to computer security. Intrusion detection systems have been developed to provide early warning of an intrusion so that defensive action can be taken to prevent or minimize damage. Intrusion detection involves detecting unusual patterns of activity or patterns of activity that are known to correlate with intrusions. One important element of intrusion prevention is password management, with the goal of preventing unauthorized users from having access to the passwords of others in this chapter the methodology was included.

2.8 Publicized Threats to Security

One of the two most publicized threats to security is the intruder (the other is viruses), often referred to as a hacker or cracker. In an important early study of intrusion, Anderson identified three classes of intruders:

- **Masquerader:** An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account
- **Misfeasor:** A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges.
- **Clandestine user:** An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection the masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider. Intruder attacks range from the benign to the serious. At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there. At the serious end are individuals who are attempting to privileged data, perform unauthorized modifications to data, or disrupt the system.

2.9 Intrusion Behavior Examples

Performing a remote root compromise of an e-mail server, Defacing a Web server. Guessing and cracking passwords, Copying a database containing credit card numbers, Viewing sensitive data, including payroll records and medical information, without authorization, Running a packet sniffer on a workstation to capture usernames and passwords, Using a permission error on an anonymous. FTP server to distribute pirated software and music files, Dialing into an unsecured modem and gaining internal network access.

2.10 Intruder Behavior Patterns

The techniques and behavior patterns of intruders are constantly shifting, to exploit newly discovered weaknesses and to evade detection and countermeasures. Even so, intruders typically follow one of a number of recognizable behavior patterns, and these patterns typically differ from those of ordinary users. In the following, looking at three broad examples of intruder behavior patterns, to give the reader some feel for the challenge facing the security administrator[16].

Hackers traditionally, those who hack into computers do so for the thrill of it or for status. The hacking community is a strong meritocracy in which status is determined by level of competence. Thus, attackers often look for targets of opportunity and then share the information with others. A typical example is a break-in at a large financial institution. The intruder took advantage of the fact that the corporate network was running unprotected services, some of which were not even needed. In this case, the key to the break-in was the Pc Anywhere application. The manufacturer, Symantec, advertises this program as a remote control solution that enables secure connection to remote devices. But the attacker had an easy time gaining access to pc “Anywhere”; the administrator used the same three-letter username and password for the program. In this case, there was no intrusion detection system on the 700-node corporate network. The intruder was only discovered when a vice president walked into her office and saw the cursor moving files around on her Windows workstation[16].

- Select the target using IP lookup tools such as NS Look up, Dig, and others.
- Map network for accessible services using tools such as NMAP.
- Identify potentially vulnerable services (in this case, pc anywhere).

- Brute force (guess) pc Anywhere password.
- Install remote administration tool called Dame Ware.
- Wait for administrator to log on and capture his password.

Use that password to access remainder of network.

2.11 Insider Attacks

Insider attacks are among the most difficult to detect and prevent. Employees already have access and knowledge about the structure and content of corporate databases. Insider attacks can be motivated by revenge or simply a feeling of entitlement. An example of the former is the case of Kenneth Patterson, fired from his position as data communications manager for American Eagle Outfitters. Patterson disabled the company's ability to process credit card purchases during five days of the holiday season of 2002. As for a sense of entitlement, there have always been many employees who felt entitled to take extra office supplies for home use, but this now extends to corporate data. An example is that of a vice president of sales for a stock analysis firm who quit to go to a competitor. Before she left, she copied the customer database to take with her. The offender reported feeling no animus toward her former employee; she simply wanted the data because it would be useful to her. Although IDS and IPS facilities can be useful in countering insider attacks, other more direct approaches are of higher priority. Examples include the following: Enforce least privilege, only allowing access to the resources employees need to do their job. Set logs to see what users access and what commands they are entering.

Protect sensitive resources with strong authentication. Upon termination, delete employee's computer and network access. Upon termination, make a mirror image of employee's hard drive before reissuing it. That evidence might be needed if your company information

turns up at a competitor. In this section, we look at the techniques used for intrusion. Then examining ways to detect intrusion.

2.11.1 Internal Threat

Create network accounts for themselves and their friends, Access accounts and applications they wouldn't normally use for their daily jobs, E-mail former and prospective employers, Conduct furtive instant-messaging chats, Visit Web sites that cater to disgruntled employees, Access the network during off hours.

Benign intruders might be tolerable, although they do consume resources and may slow performance for legitimate users. However, there is no way in advance to know whether an intruder will be benign or malign. Consequently, even for systems with no particularly sensitive resources, there is a motivation to control this problem. Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter this type of hacker threat. In addition to using such systems, organizations can consider restricting remote logons to specific IP addresses and/or use virtual private network technology[18]. One of the results of the growing awareness of the intruder problem has been the establishment of a number of computer emergency response teams (CERTs). These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers. Hackers also routinely read CERT reports. Thus, it is important for system administrators to quickly insert all software patches to discovered vulnerabilities. Unfortunately, given the complexity of many IT systems, and the rate at which patches are released, this is increasingly difficult to achieve without automated updating. Even then, there are problems caused by In compatibilities resulting from the updated software. Hence the need for multiple layers of defense in managing security threats to IT systems[18].

2.11.2 Cybercrimes

Organized groups of hackers have become a widespread and common threat to Internet-based systems. These groups can be in the employ of a corporation or government but often are loosely affiliated gangs of hackers. Typically, these gangs are young, often Eastern European, Russian, or southeast Asian hackers who do business on the Web. They meet in underground forums with names like DarkMarket.org and theftservices.com to trade tips and data and coordinate attacks. A common target is a credit card file at an e-commerce server[19]. Attackers attempt to gain root access. The card numbers are used by organized crime gangs to purchase expensive items and are then posted to carder sites, where others can access and use the account numbers; this obscures usage patterns and complicates investigation. Whereas traditional hackers look for targets of opportunity, criminal hackers usually have specific targets, or at least classes of targets in mind. Once a site is penetrated, the attacker acts quickly, scooping up as much valuable information as possible and exiting. IDSs and IPSs can also be used for these types of attackers, but may be less effective because of the quick in-and-out nature of the attack. For e-commerce sites, database encryption should be used for sensitive customer information, especially credit cards. For hosted e-commerce sites (provided by an outsider service), the e-commerce organization should make use of a dedicated server (not used to support multiple customers) and closely monitor the provider's security services[19].

2.11.3 Cybercrimes Enterprise

Act quickly and precisely to make their activities harder to detect, Exploit perimeter through vulnerable ports. Use Trojan horses (hidden software) to leave back doors for reentry. Use sniffers to capture passwords. Do not stick around until noticed and Make few or no mistakes.

2.12 Related Works

The related works covers papers starting for 1980 in the IEEE European Symposium on Security and Privacy, the IEEE Symposium on Security and Privacy (S&P) has been the premier forum for the presentation of developments in computer security and electronic privacy, and for bringing together researchers and practitioners in the field.

In [3], the authors presented an immune system in both anomaly and misuse detection methods and compared the two methods. The immune system is based on the combination of positive and negative characterizations which come from several features defined as normal or abnormal states. A trusted agent based approach was proposed in[3]. which determines whether a machine in a network is malicious based on the experiences and its previous operations.

In [4]. Vieira and Schulte proposed an ANN based function to realize an IDS on the cloud, and a feed-back structure ANN is used to create a behavior-based system and an expert system to build a knowledge-based system.

In [5]. The authors concentrated on alleviating the network traffic when realizing an IDS based on a Map-Reduce framework. Here a distributed IDS architecture is proposed which consists of nodes running back propagation (BP) based ANNs on the cloud platform.

Chapter Three

System Design

Chapter Three

System Design

In this chapter the methodology was included along with the algorithms used in the program that used to evaluate the algorithms.

3.1 Proposed System Block Diagram

In Figure (3.1) shows block diagram of the proposed system structure that includes the ANN (Artificial Neural Network) which has an input traffic monitoring block and the rules based block, the activity of the detection is stored into database represented by the storage block, the neural network detects the intrusion.

Moreover the neural network output is a classification of the attack and an alarm with a threshold value is configured to notify the administrator whenever an attack is detected.

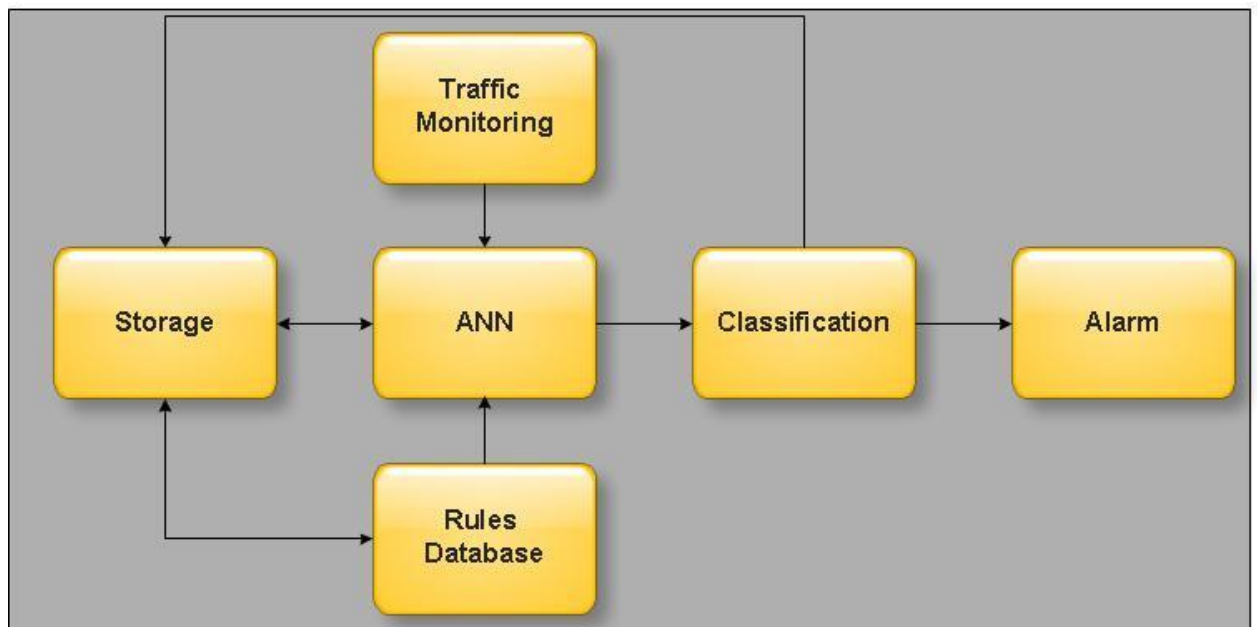


Figure 3.1: Proposed Block Diagram

3.2 Proposed Features for Detection

1. Statistical anomaly detection: Involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

- **Threshold detection:** This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.
- **Profile based:** A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

2. Rule-Based Detection: Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.

- **Anomaly detection:** Rules are developed to detect deviation from previous usage patterns.
- **Penetration identification:** An expert system approach that searches for suspicious behavior.

3.2.1 Measures That May Be Used for Intrusion Detection

Elapsed time per session Mean and standard deviation significant deviations might indicate masquerader, Password failures at login Operational Attempted break-in by password guessing.

3.2.2 Command or Program Execution Activity Program Resource Utilization Mean and Standard Deviation

An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization, Execution denials Operational model May detect penetration attempt by individual user who seeks higher privileges.

3.2.3 File Access Activity

Read, write, create, delete frequency Mean and standard deviation
Abnormalities for read and write access for individual users may signify masquerading or browsing and Failure count for read, write, create, delete Operational May detect users who persistently attempt to access unauthorized files.

3.3 Artificial Neural Network

Artificial neural networks (ANNs), a form of connectionism, are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve performance) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the analytic results to identify cats in other images.

They have found most use in applications difficult to express in a traditional computer algorithm using rule-based programming. An ANN is based on a collection of connected units called artificial neurons, (analogous to axons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream. Further, they may have a threshold such that only if the aggregate signal is below (or above) that level is the downstream signal sent. Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel

from the first (input), to the last (output) layer, possibly after traversing the layers multiple times. The original goal of the neural network approach was to solve problems in the same way that a human brain would. Over time, attention focused on matching specific mental abilities, leading to deviations from biology such as back propagation, or passing information in the reverse direction and adjusting the network to reflect that information. Neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games, medical diagnosis and in many other domains.

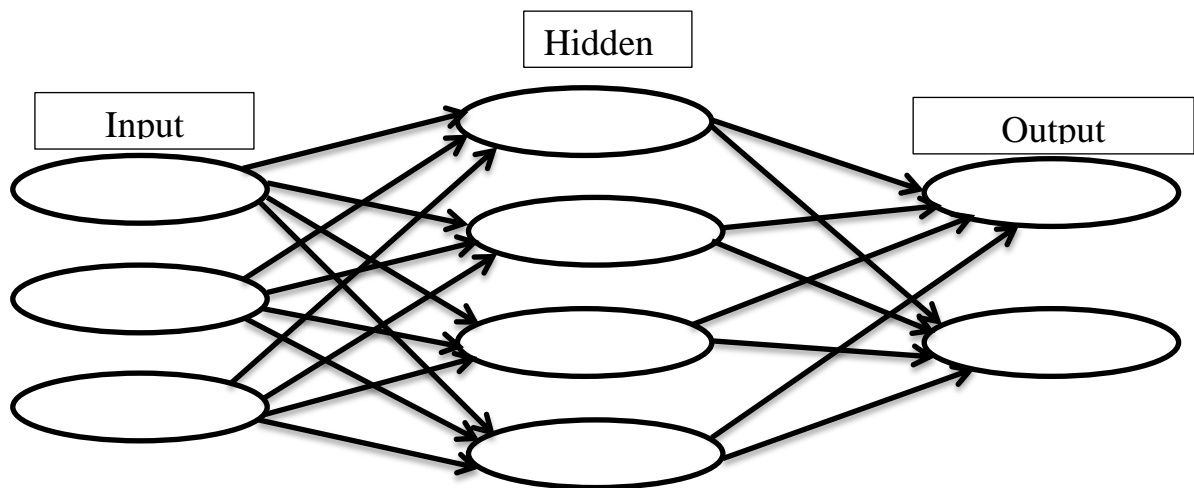


Figure 3.2: Artificial Neural Networks

3.3.1 Characteristics of Artificial Neural Networks

It is a computational system inspired by the Structure Processing Method Learning Ability of a biological brain. A large number of very simple processing neuron-like processing elements, A large number of weighted connections between the elements, Distributed representation of knowledge over the connections and Knowledge is acquired by network through a learning process. Need Artificial Neural Networks to: Massive Parallelism, Distributed representation, Learning ability, Generalization ability, Fault tolerance.

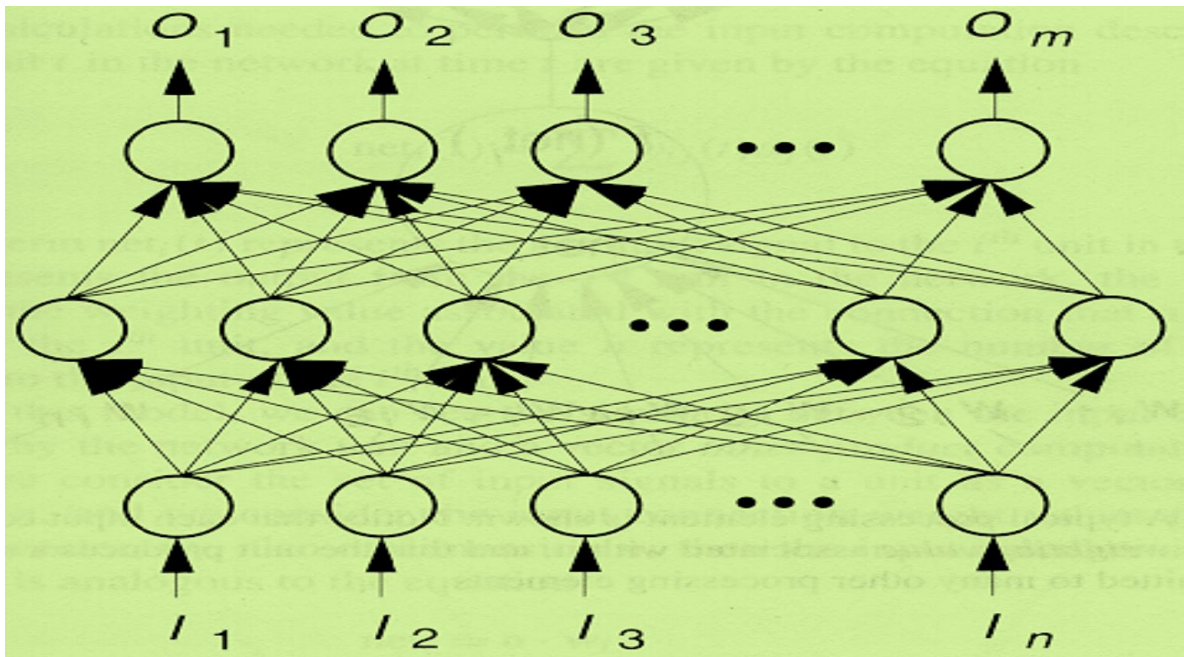


Figure 3.3: Structure Artificial Neural Networks

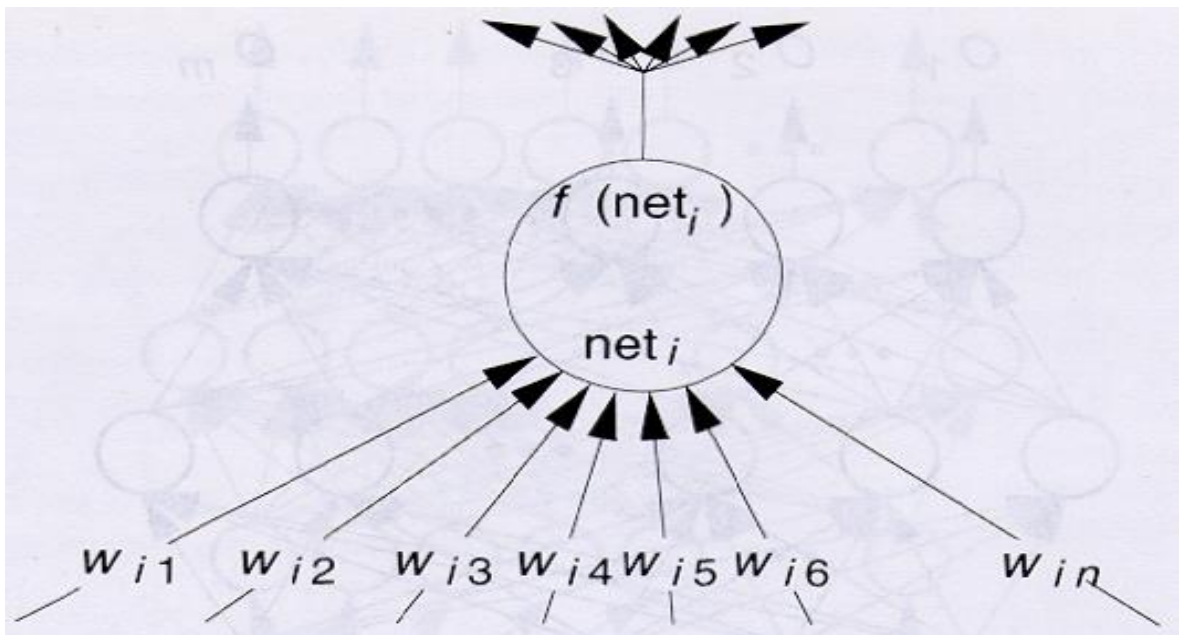


Figure 3.4: Artificial Neural Network Processing Unit

The following diagram represents the general model of ANN followed by its processing.

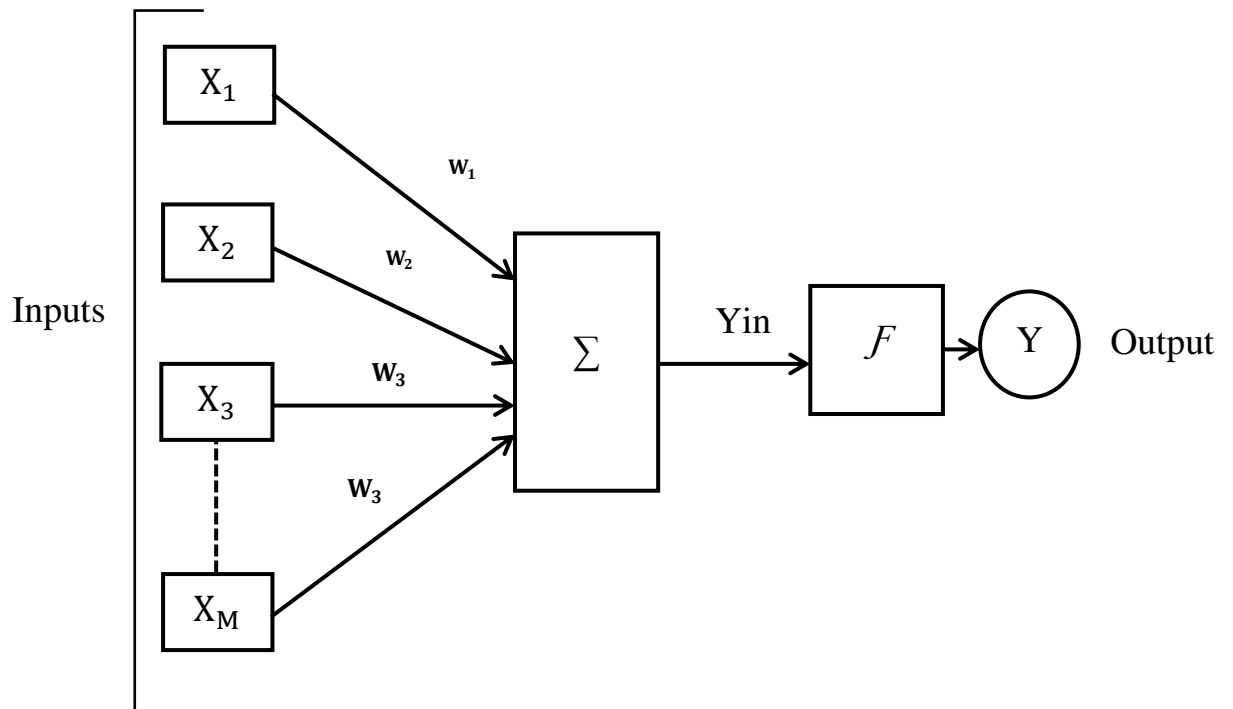


Figure 3.5 General Activation Function

For the above general model of artificial neural network, the net input can be calculated as follows:

$$y_{in} = X_1.W_1 + X_2.W_2 + X_3.W_3 + \dots + X_M.W_M$$

The output can be calculated by applying the activation function over the net input.

$$Y = F(y_{in})$$

Output = function (net input calculated).

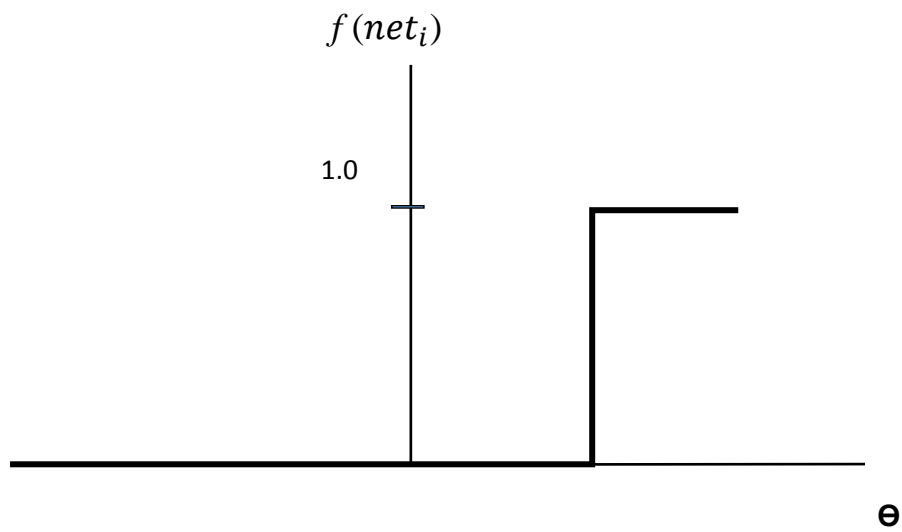


Figure 3.6:Neural Network Activation Function

Node Input: $net_i = j \sum W_{ij} I_i$
(3.1)

Node Out put $O_i = f(net_i)t$
(3.2)

3.4 Applications of Neural Networks

- Pattern Classification
- Clustering / Categorization
- Function approximation
- Prediction / Forecasting
- Optimization
- Content-addressable memory control

3.4.1 Neural Networks in Practice

Given this description of neural networks and how they work, what real world applications are they suited for neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries. Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including: Sales forecasting , Industrial process control, Customer research, Data validation , Risk management and Target marketing. But to give you some more specific examples; ANN are also used in the following specific paradigms: recognition of speakers in communications; diagnosis of hepatitis; recovery of telecommunications from faulty software; interpretation of multi-meaning Chinese words; undersea mine detection; texture analysis; three-dimensional object recognition; hand-written word recognition; and facial recognition.

3.5 System Model

The following computer model illustrate the flow of the program including the decision making among a conditional statements. The program starts by examining the intrusion and its activities, the program examine the rules and execute subroutine in the program to give an output. Six features were included in order to detect a wide activity and increase the accuracy of detection. The classification is neural network based detection was used for classifying the intrusion and to classify it to the system in order to take decision.

- **The features that is covered are**

File access features, Login features, and Command line.

3.5.1 Features Construction System Model

3.5.1.1 Elapsed Time

In Figure (3.7) illustrate is the user elapsed time detection and how the system takes action while exceeding the reference value of the elapsed time. the system start counting the user elapsed time and increase the counter till the threshold value comes the system will block user or Indicate Administrator.

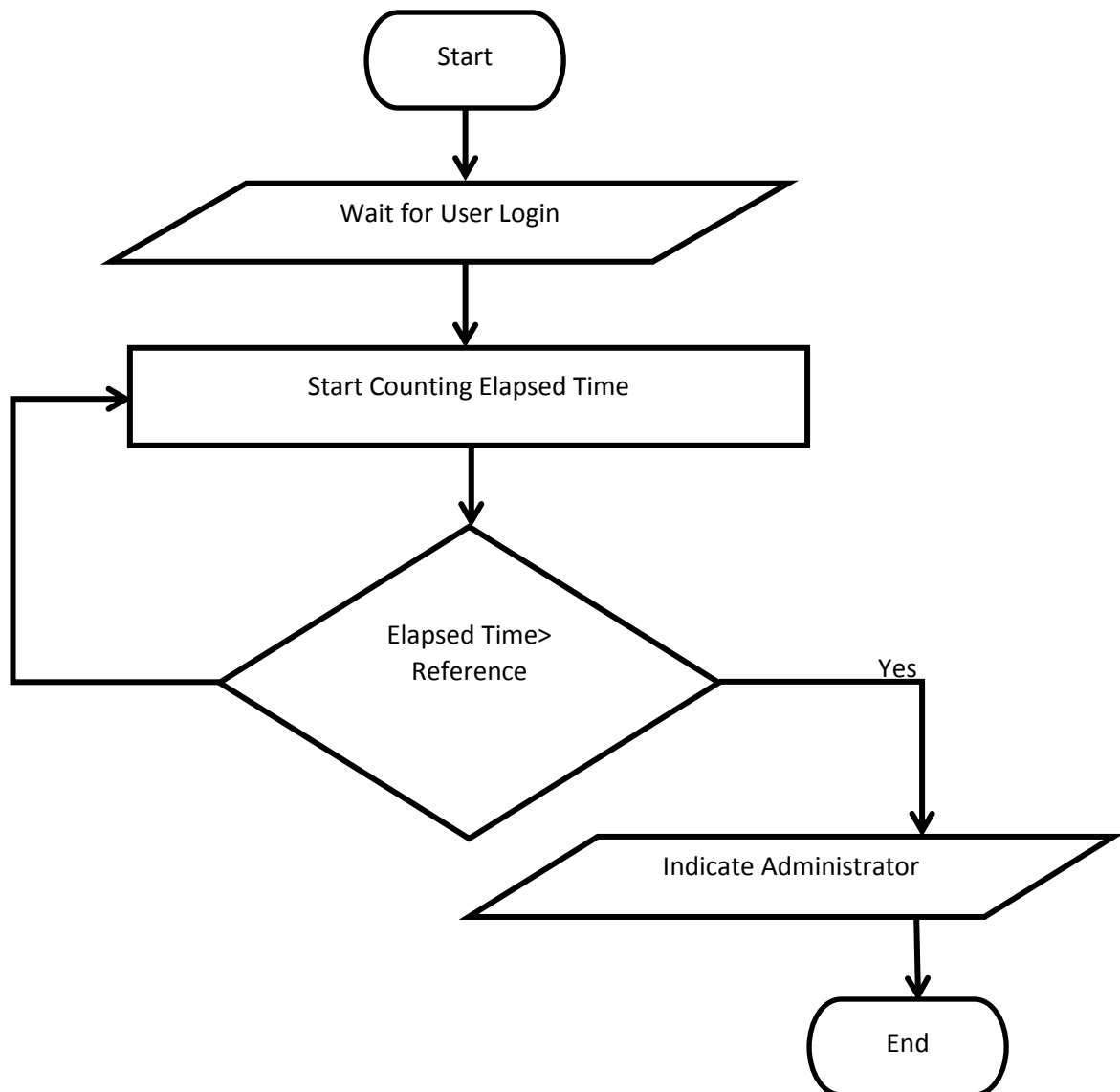


Figure3.7: Elapsed Time

3.5.1.2 Incorrect Password Attempts

In Figure(3.8)shows the user Incorrect Password Attempts detection, the system start counting the number of incorrect password attempts and increase the counter till the threshold value comes the system will block user or contact the

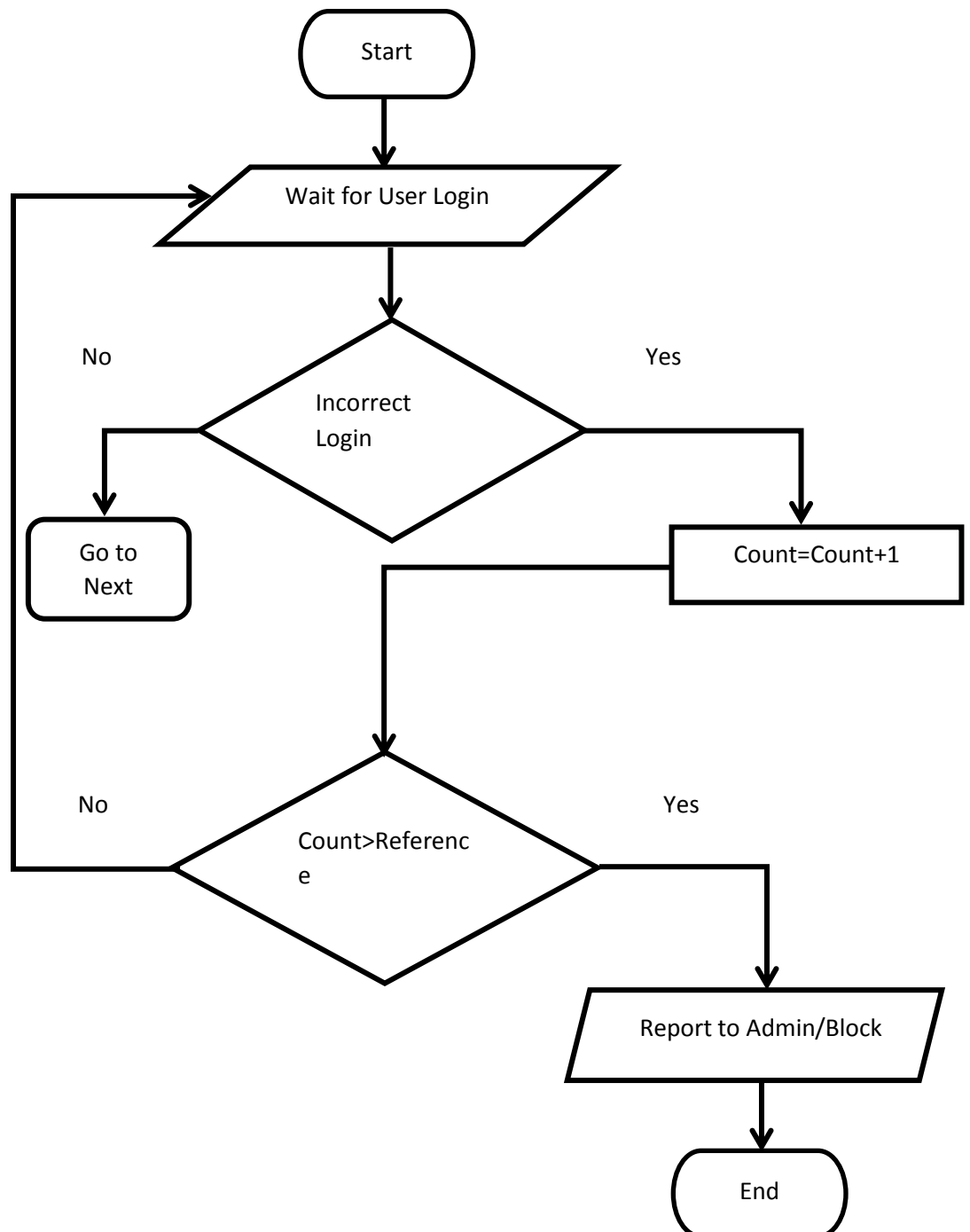


Figure 3.8: Incorrect Password Attempts

3.5.1.3 Higher Privileges Detection

In Figure (3.9) illustrate is Higher Privileges Detection, the system detect he users that seeking for higher privileges this is an intrusion, also the system increase the counter of attempts that requesting to higher privileges. The counter when reaching the threshold value it evaluate the counter and block user or attempt to display a message for the administrator.

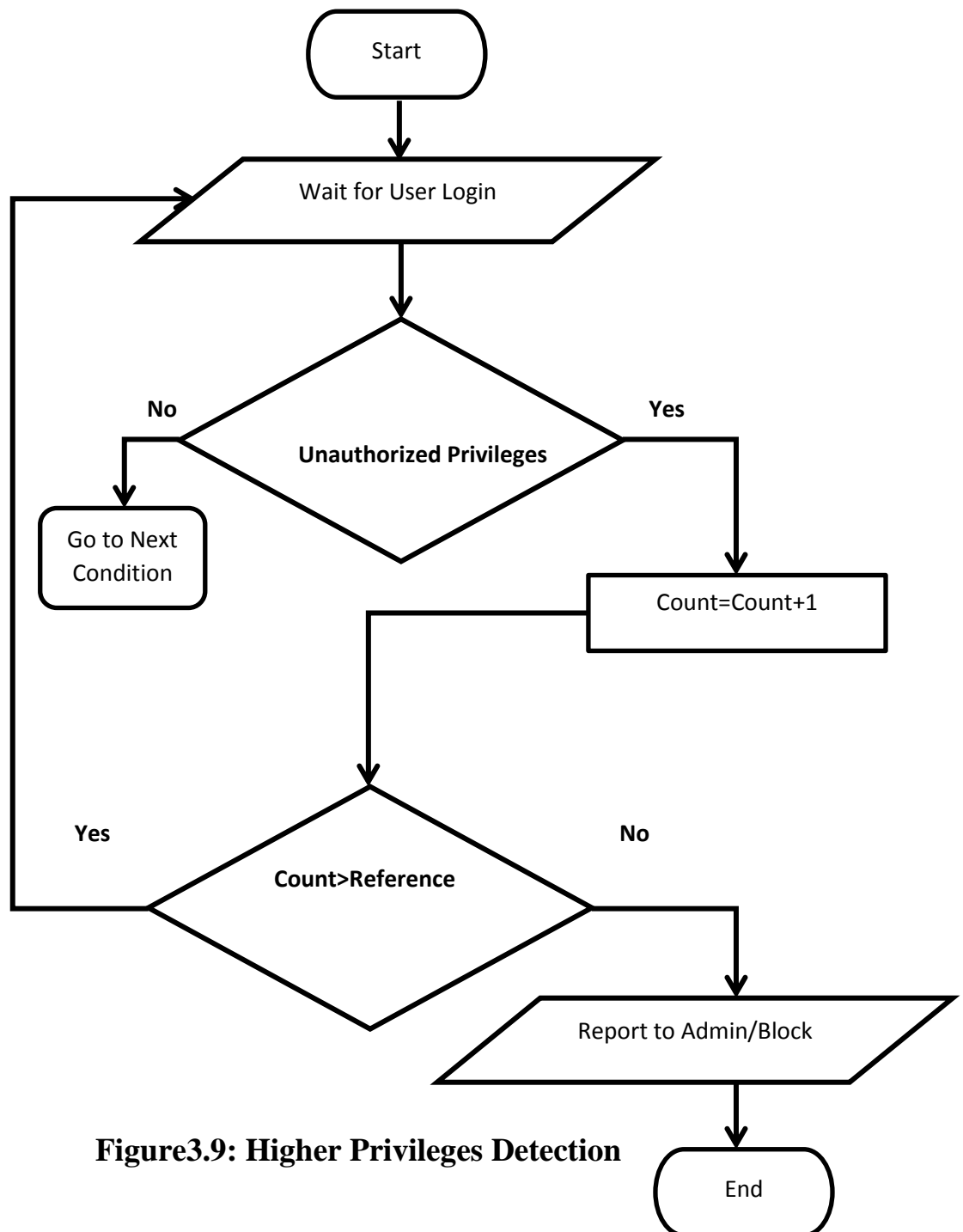


Figure3.9: Higher Privileges Detection

3.5.1.4 Abnormal Read/Write Activities

Reaching the threshold value it evaluate the counter and block user or attempt to display a message for the administrator In Figure (3.10) explain Abnormal Read/Write Activates , the system detect he abnormal read and write of files into the system this is an intrusion, also the system increase the counter of attempts reading and writing.

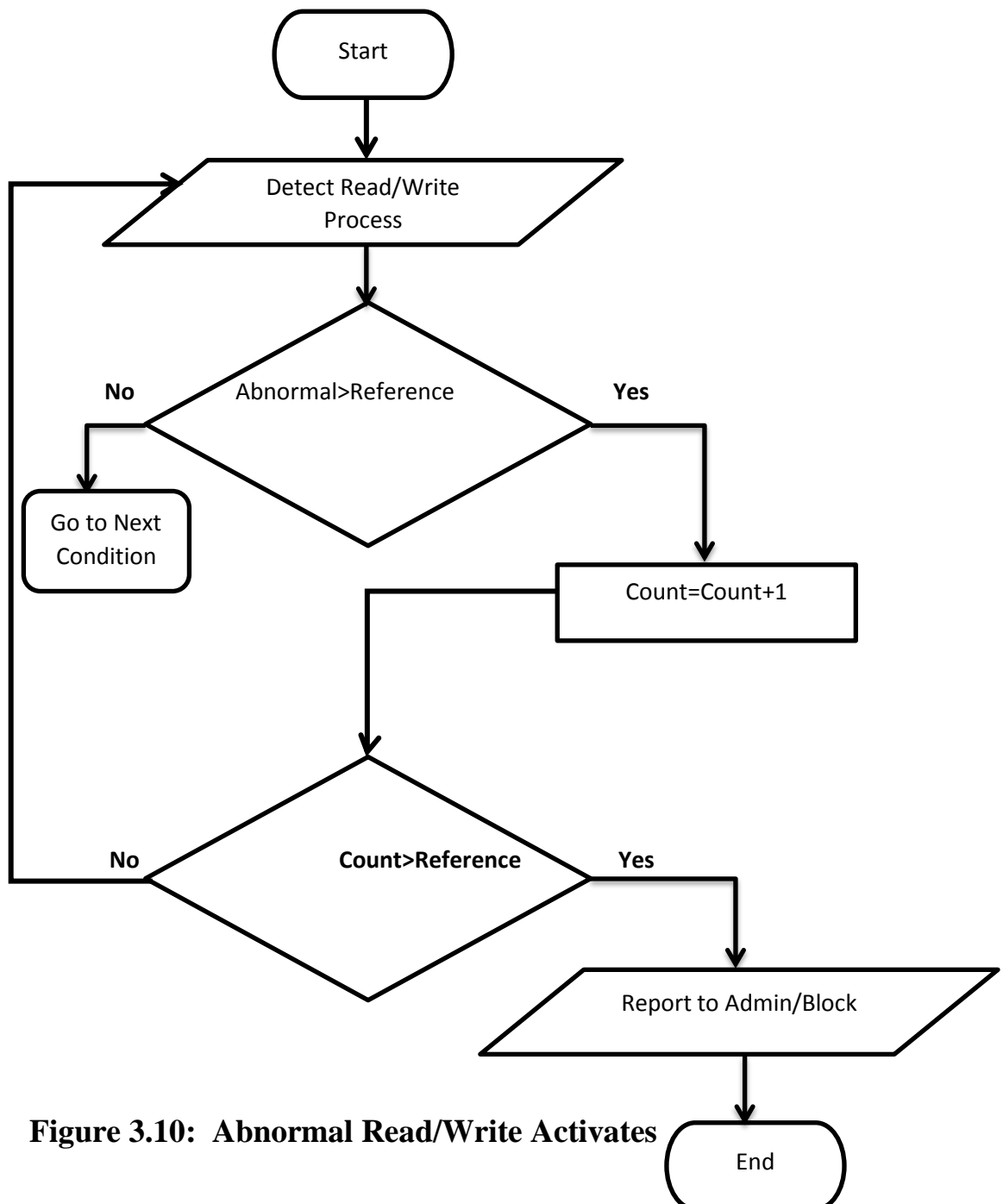


Figure 3.10: Abnormal Read/Write Activates

3.5.1.5 Abnormal Processor Usage

In Figure (3.11) illustrates the Abnormal Processor Usage, the system detect he users that CPU usage this is an intrusion, also the system increase the counter while unexpected usage of the processor is attempt. The counter when reaching the threshold value it evaluate the counter and block user or attempt to display a message for the administrator.

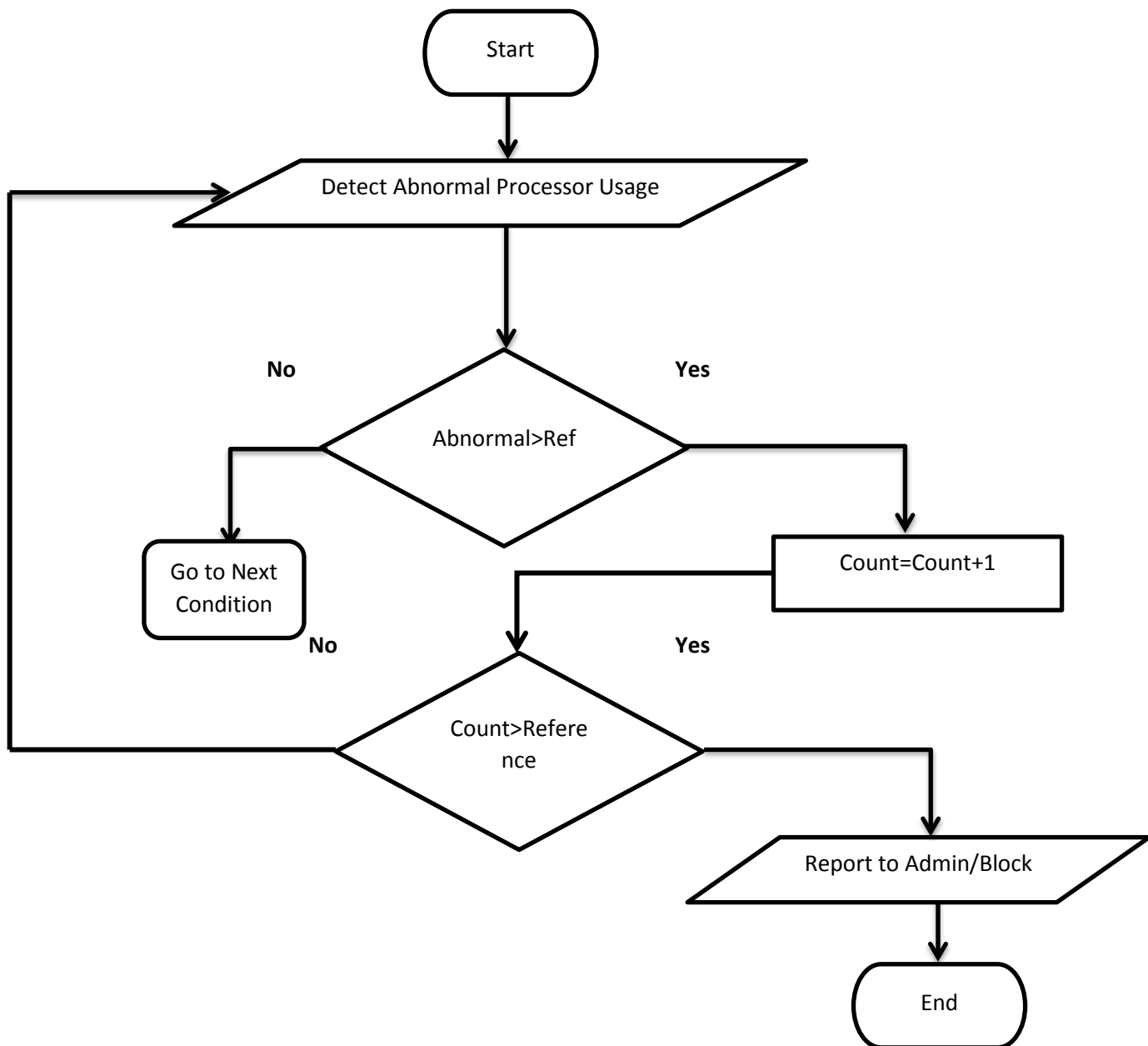


Figure 3.11: Abnormal Processor Usage

3.5.1.6 Unauthorized File Access

In Figure (3.12) shows Unauthorized File Access, the system detect unauthorized file access by increasing the counter, also the system increase the counter of attempts that accessing unauthorized files. The counter when reaching the threshold value it evaluate the counter and block user or attempt to display a message for the administrator.

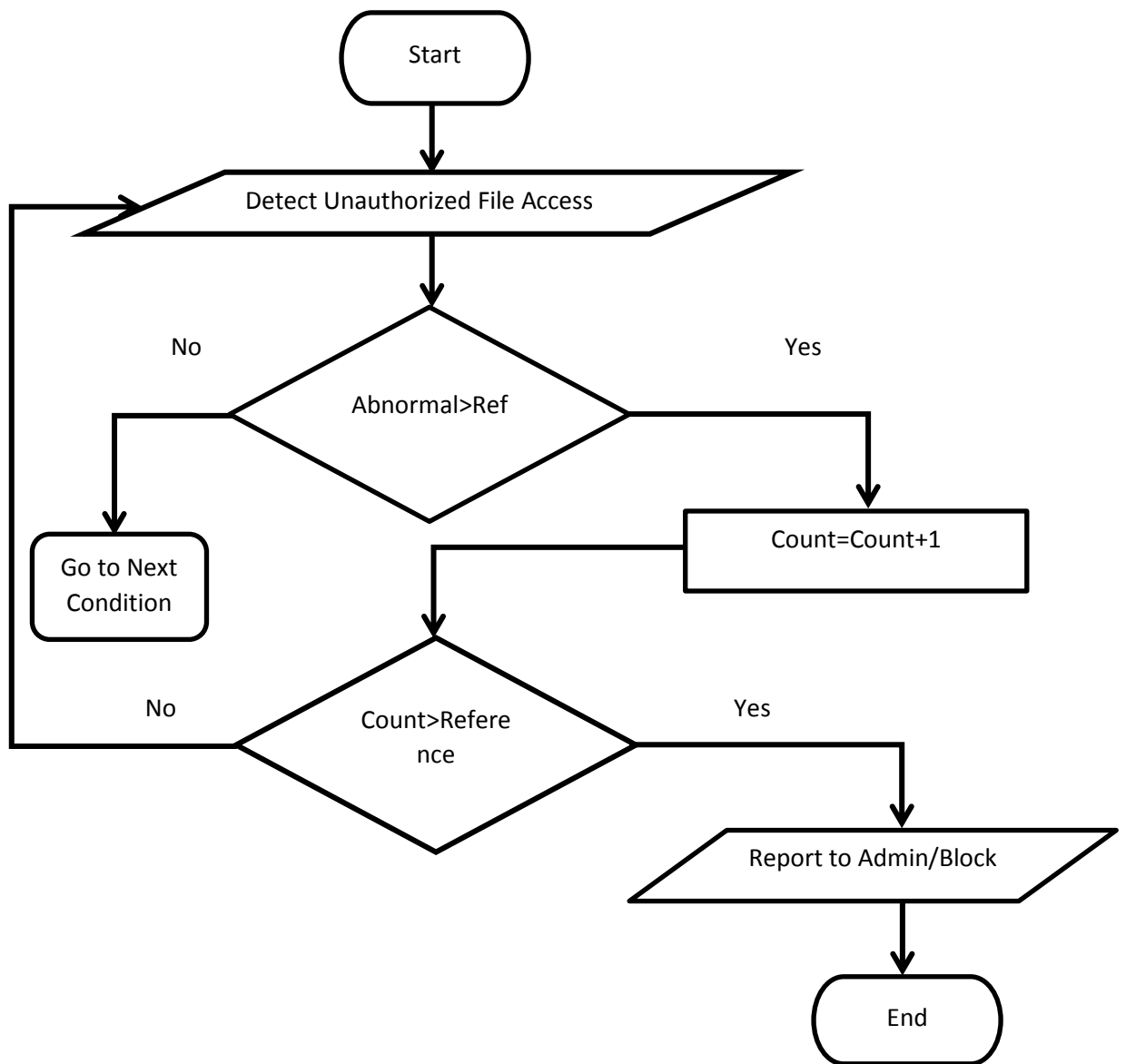
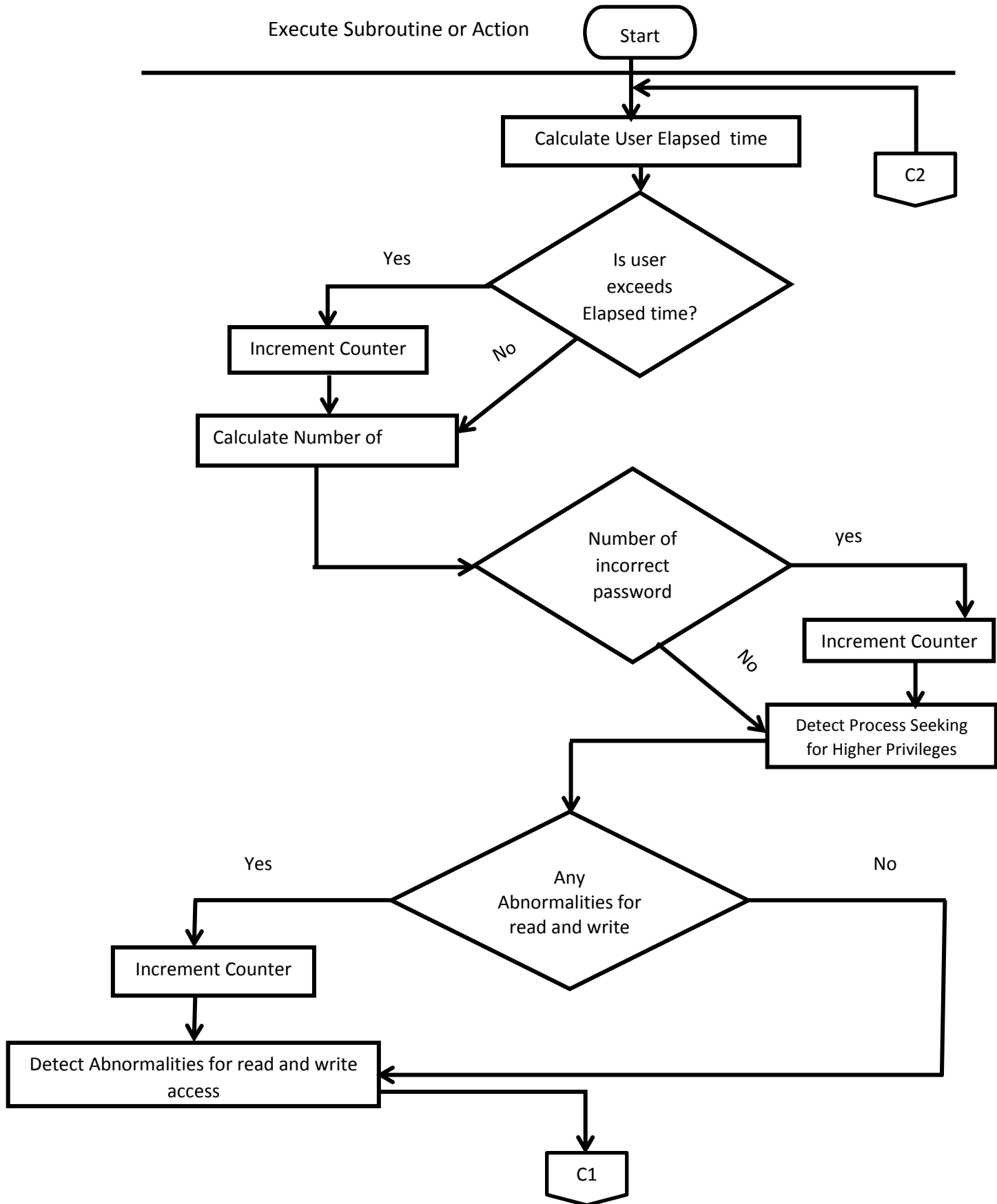


Figure 3.12: Unauthorized File Access

3.5.1.7 Intrusion Detection System Flow Chart



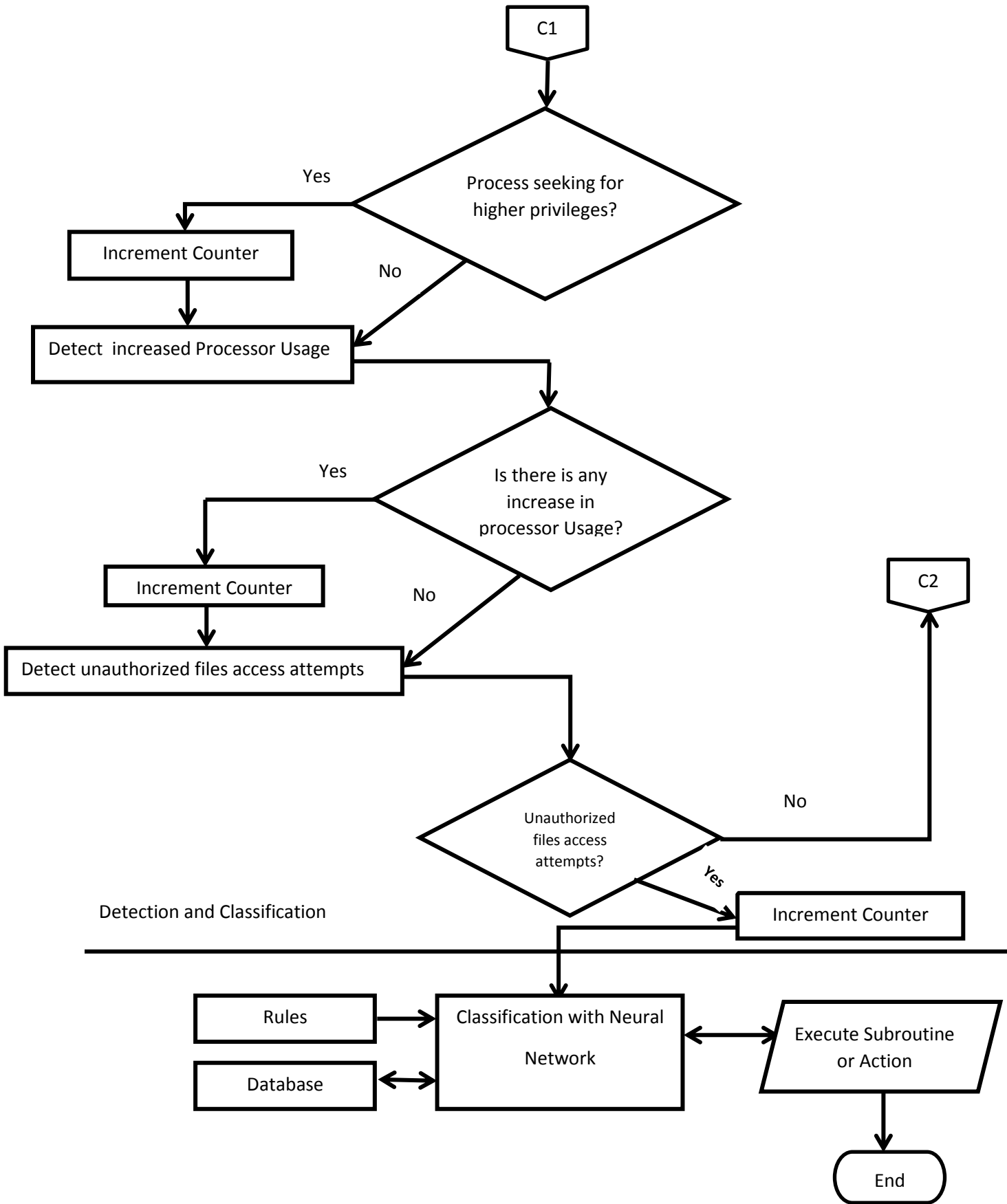


Figure 3.13: System flow Chart

Chapter Four

Results and Discussion

Chapter Four

Results and Discussion

In this chapter the results and discussion was included along with the analysis to the accuracy of detection of the proposed system.

4.1 Scenario of IDS using ANN

Many features were used to generate a power full detection tool these features are:

The Elapsed time per session Mean and standard deviation significant deviations might indicate masquerader. Password failures at login Operational Attempted break-in by password guessing. Abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization and Execution denials Operational model May detect penetration attempt by individual user who seeks higher privileges. File access activity Read, write, create, delete frequency Mean and standard deviation Abnormalities for read and write access for individual users may signify masquerading or browsing. Failure count for read, write, create, delete Operational May detect users who persistently attempt to access unauthorized files.

4.2 Sample Dataset Files

In Figure (4.1) Shows monitoring log files data set was used to test the program, the dataset includes the extracted features which is examined through the ANN to classify the output into Attack or normal process and an administrator action. These results is a dataset used for training of any intrusion system, it allow the system to detect normal from up normal activities.














| | | | |
|--|-------------------|----------------------|-----------|
|  Dataset_Anomaly.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 781 KB |
|  Dataset_Misuse.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 782 KB |
|  Optimized_Back.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 164 KB |
|  Optimized_BufferOverflow.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 6 KB |
|  Optimized_FTPWrite.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 2 KB |
|  Optimized_GuessPassword.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 10 KB |
|  Optimized_Neptune.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 37,157 KB |
|  Optimized_NMap.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 245 KB |
|  Optimized_Normal.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 93,328 KB |
|  Optimized_PortSweep.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 492 KB |
|  Optimized_RootKit.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 2 KB |
|  Optimized_Satan.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 825 KB |
|  Optimized_Smurf.csv | 5/20/2015 1:02 PM | Microsoft Excel C... | 474 KB |

Figure 4.1: Sample Data Set Files

The sample dataset from an internet sources and available for download and the fact is to learn how to measure the intrusion behavior and the process required to be monitored.

The files includes mainly more than six intrusion types, through examining the fields on the log file such as data, time, number of wrong login, privileges problem, spoofing.

Table 4.1: Sample Data Set Files Anomaly Detection Method

| | | | | | | | | | |
|-------|-------|------|-------|-------|-------|-------|-------|-------|---|
| 0.218 | 0.001 | 0 | 0.095 | 0.096 | 0 | 0.096 | 0.1 | 0 | 0 |
| 0.255 | 0.001 | 0 | 0.002 | 0.003 | 0 | 0.002 | 0.1 | 0.001 | 0 |
| 0.031 | 0.255 | 0.1 | 0 | 0.003 | 0.004 | 0 | 0 | 0 | 0 |
| 0.235 | 0.255 | 0.1 | 0 | 0 | 0.001 | 0 | 0 | 0 | 0 |
| 0.021 | 0.156 | 0.1 | 0 | 0.005 | 0.004 | 0 | 0 | 0 | 0 |
| 0.166 | 0.255 | 0.1 | 0 | 0.001 | 0.002 | 0 | 0 | 0 | 0 |
| 0.072 | 0.072 | 0.1 | 0 | 0.001 | 0 | 0 | 0 | 0 | 0 |
| 0.098 | 0.018 | 0.01 | 0.005 | 0.001 | 0.011 | 0 | 0 | 0 | 0 |
| 0.039 | 0.255 | 0.1 | 0 | 0.003 | 0.004 | 0 | 0 | 0 | 0 |
| 0.025 | 0.255 | 0.1 | 0 | 0.004 | 0.004 | 0 | 0.001 | 0 | 0 |
| 0.255 | 0.255 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.049 | 0.255 | 0.1 | 0 | 0.004 | 0.002 | 0 | 0 | 0 | 0 |

4.3 Main Program Screen

The intrusion detection program uses a pre monitored data set includes many features and behaviors of the Intrusion. In Figure (4.2) the form includes a display to all of the features that used for detection, a rules was set to count the number featured that has been attacked, while two features is attacked the system automatically block the process, while three feature attack the administrator notification is displayed, up to six features the system allow, The GUI was build using Borland Delphi and a log file saving to storage is used to store the output.

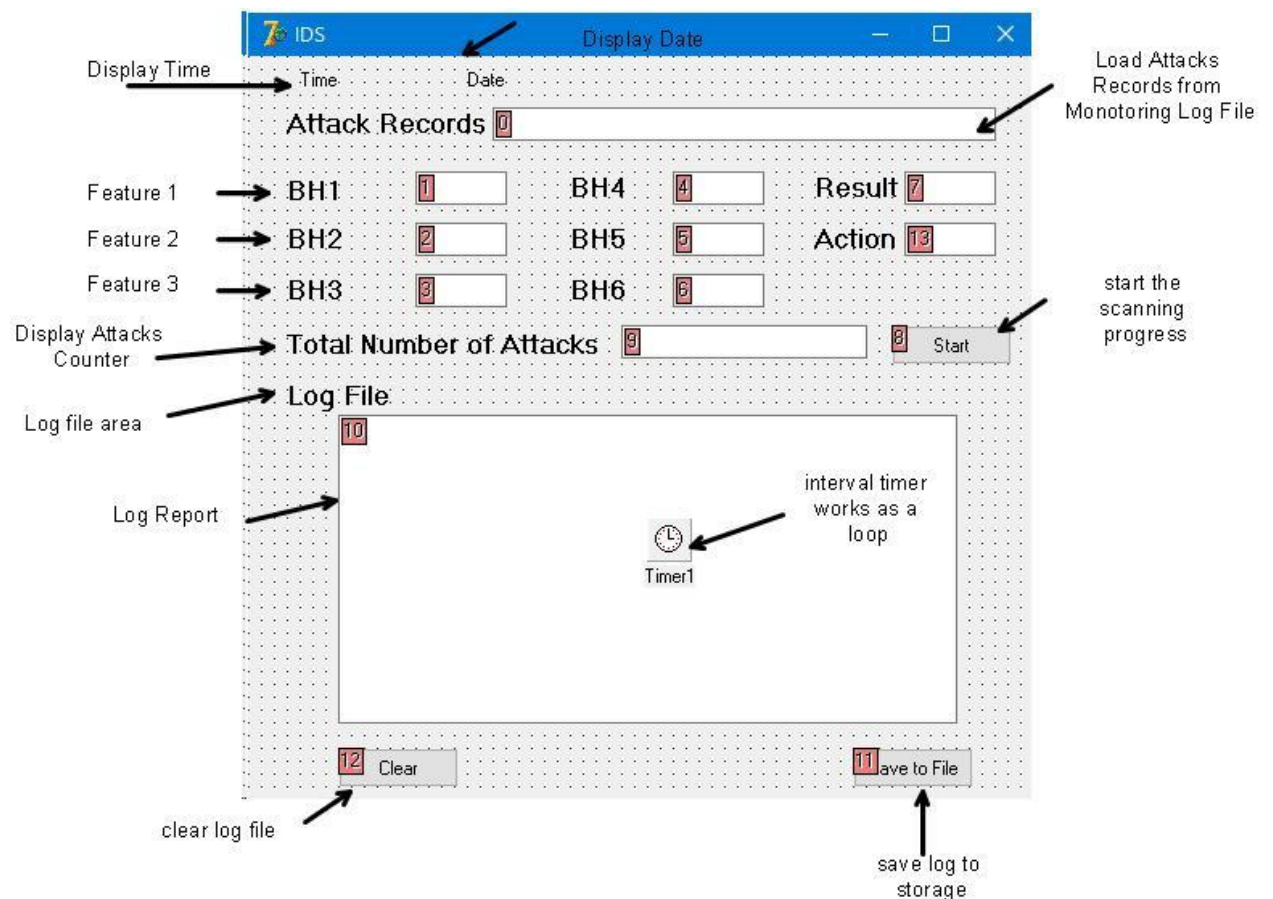


Figure 4.2: Borland Delphi Main Program Screen

True Positive (T_P) – counts of all samples which are correctly called by the algorithm as being cancer.

False Positive (F_P) – counts of all samples which are incorrectly called by the algorithm as being cancer while they are normal.

True Negative (T_N) – counts of all samples which are correctly called by the algorithm as being normal.

False Negative (F_N) – count of all samples which are incorrectly called by the algorithm as being normal while they are cancer.

The performance of the classification algorithms was evaluated by computing the percentages of Sensitivity (SE), Specificity (SP), Accuracy (AC) and Mathews Correlation Coefficient (MCC),

4.5.1 The respective definitions are as follows

$$SE = T_P / (T_P + F_N) * 100 \dots\dots\dots(4.1)$$

$$SP = T_N / (T_N + F_P) * 100 \dots\dots\dots(4.2)$$

$$AC = (T_P + T_N) / (T_P + T_N + F_P + F_N) * 100 \dots\dots\dots(4.3)$$

$$MCC = \frac{(T_P \times T_N - F_P \times F_N)}{\sqrt{(T_P + F_P)(T_P + F_N)(T_N + F_P)(T_N + F_N)}} \dots\dots\dots(4.4)$$

While examining the sensitivity of the software it was found that the system is capable of detecting with a sensitivity of 83.1% and the accuracy is about 75% which concluded that more training samples and new rules is required to be set.

Table 4.2: examining the sensitivity and accuracy of the software

| Sensitivity (SE) | Specificity (SP) | Accuracy (AC) | Correlation Coefficient (MCC) |
|---------------------|---------------------|------------------|----------------------------------|
| 83.03571 | 49.72973 | 74.59677 | 0.5072442 |

4.6 Accuracy of Results

The dataset testing results among the four QOS factor:

False positive: is the number of detection by the system with a failure in detection, the system say that there is an intrusion and it is not intuition. False negative: represent the number of unsuccessful detection of the system to the intrusion. True positive: is the number of success detection of the program. True negative: is the success detection of the intrusion that is not appears. These values were inserted into a table after examining the dataset files of the training. The file includes a notes about each record including behavior and the classification.

Table 4.3 : Dataset testing results among the four QOS factor:

| Set Number | F_P | F_N | T_P | T_N |
|------------|-------|-------|-------|-------|
| 1 | 5 | 4 | 10 | 12 |
| 2 | 5 | 2 | 12 | 12 |
| 3 | 4 | 2 | 12 | 13 |
| 4 | 4 | 2 | 12 | 13 |
| 5 | 4 | 2 | 12 | 13 |
| 6 | 3 | 2 | 12 | 14 |
| 7 | 3 | 2 | 12 | 14 |
| 8 | 3 | 2 | 12 | 14 |
| 9 | 4 | 3 | 11 | 13 |
| 10 | 5 | 3 | 11 | 12 |
| 11 | 5 | 3 | 11 | 12 |
| 12 | 5 | 3 | 11 | 12 |
| 13 | 17 | 0 | 14 | 0 |
| 14 | 9 | 2 | 12 | 8 |
| 15 | 7 | 3 | 11 | 10 |
| 16 | 5 | 3 | 11 | 12 |

The Figure (4.4) chart that represented as a graph analysis such as the following figure, each iteration was counted for analysis purpose.

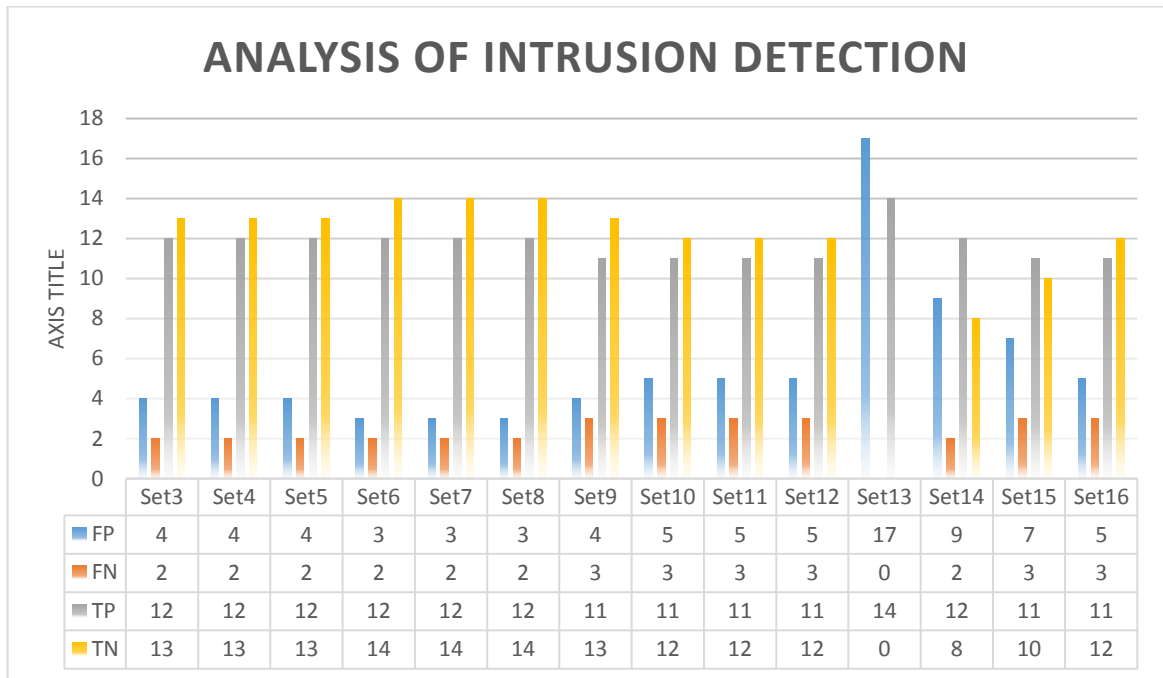


Figure 4.4: Chart that represent the detection while running the data set.

4.7 Summery

The Table (4.4) shows the tool was developed to cover six features and behaviors which give the tool power compared with other developed systems

Table 4.4 : the tool power compared with other developed systems

| No | Number of behavior supported | Accuracy | Classification Method |
|----|------------------------------|----------|-----------------------|
| 1 | 3 | 76% | ANN |
| 2 | 4 | 80% | SVM |
| 3 | 6 | 72% | ANN |

Chapter Five

Conclusion and Recommendation

Chapter Five

Conclusion and Recommendations

5.1 Conclusion

An intrusion detection system (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations.

After study and analysis of intrusion detection techniques a software was developed based on Borland Delphi which has the capability to identify the Type process and behavior based on classification using neural network in order to Provide a simple training program to increase the efficiency of detection and Improve the detection diagnosis and decrease the mistakes, Reduce time consumed for diagnosis and Provide high accuracy of detection and classification based on computerized method. But the computer technique is used to solve the problem, reduce the time, and will give more accuracy for classification and detection.

After running the simulation program on 16 sample normal and abnormal attacks it was found System accuracy 75%, then this System is effective.

5.2 Recommendations

After finishing this research work there are still some open issues can be considered for future research; these include:

- 1- Apply automatic updates to the system for increasing the system accuracy.
- 2- Design an alert system based on SMS.
- 3- Design of remote monitoring tool for remote access.

References

1. Kumar, VIPIN, JAIDEEP SRVASTAVA, and Aleksandra LAZAREVIC, “Managing cyber threats: Issues, approaches, and challenges”. J. ACM, vol. 52, no. 2, pp. 217-244, 2006.
2. Mahesh Kumar SABHNANI and GURSEL SERPEN, “Why Machine Learning Algorithms Fail in Misuse Detection on KDD Intrusion Detection Data Set”. Transactions on Intelligent Data Analysis(ACM), vol. 1.5, no. 6, pp. 403-415, 2004.
3. M. SHYU, S. Chen, K. SARINN APAKORN, and L. Chang, “A novel anomaly detection scheme based on principal component classifier”. Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM03), vol. 1.6, no. 10, pp. 172– 179, 2003.
4. J. McHugh, “Testing intrusion detection systems: a critique of the 1998 and 1999 draper intrusion detection system evaluations as performed by Lincoln laboratory”. ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262– 294, 2000.
5. Mamboed TAVALLAEE, Ibrahim BAGHERI, Wei Lu, and Ali A. Ghorbanifar, “A Detailed analysis of the KDD CUP 99 Data Set”. In the Proc. Of the IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA), vol. 1.8, pp. 1-6, 2009.
6. S. REVATHI, Dr. A. Malachi, “A detailed analysis of KDD cup99 Dataset for IDS”. International Journal of Engineering Research & Technology (IJERT) , Dec. 2013, pp. 319-327.
7. R. P. Lippmann, D. J. Fried, and I. Graf, “Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection

evaluation”. In Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX'00), (2000).

8. “NSL-KDD data set for network-based intrusion detection systems”. Available on: <http://nsl.cs.unb.ca/NSL-KDD/>, November 2014.

9. Lee W., and STOLFOS.J., “A framework for constructing features and models for intrusion detection systems”. ACM Transactions on Information and System Security, 3 (4) (pp. 227-261) (2000).

10. XINDONG Wu, VI pin Kumar, J. Ross Quinlan, Joy deep GHOSH, QIANG YANG, Hiroshi MOTODA, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, ZHI-HUA Zhou, Michael Steinbach, David J. Hand, Dan Steinberg, “ Top Ten Data Mining Algorithms”. Knowledge and Information Systems Journal, Springer-Verilog London, vol. 14, Issue 1, pp. 1-37, 2007.

11. Lei Li, De-Zhang Yang, Fang-Cheng Sheen, “A Novel Rule-based Intrusion detection System Using Data Mining”. In the Proc. Of 3rd IEEE International Conference on Computer Science and Information Technology, pp. 169-172, 2010.

12. J. E. Gaffney and J. W. Uvula, “Evaluation of intrusion detectors: A decision theory approach”. In Proceedings of the 2001 IEEE symposium on Security and Privacy, pages 5061, Oakland, CA, USA, 2001.

13. D. Aldous, “The continuum random tree. I”. The Annals of Probability, pp. 1–28, 1991.

14. BREIMAN, Leo, Friedman, J. H., Olsten, R. A., Stone, C. J., “Classification and regression trees”. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. (ISBN) , vol. 7, no. 4, pp. 162-164, 1984.

15. Data Mining Practical Machine Learning Tools and Techniques by Ian H Witten, Elbe Frank, Mark A Hall. 2000, pp. 199–212.
16. Han, JIAWEI, and Michelin KAMBER. Data Mining, Southeast Asia Edition: Concepts and Techniques. Morgan Kaufmann, vol. 7, no. 4, pp. 162-164, 2006.
17. PREETI AGGARWAL, SUDHIR K Sharma, “An Empirical Comparison of Classifiers to Analyze Intrusion Detection”. International Conference on Advanced Computing & Communication Technologies, IEEE/ACM, vol. 10, no. 6, 2015, pp.721–734.
18. G. GU, P. FOGKA, D. Dagon and W. Lee, “An Information-Theoretic Measure of Intrusion Detection Capability”. In Proceedings of the 2006 ACM Symposium on Information, computer and communications security, vol. 10, no. 6,2006.pp. 21-24

Appendix

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Edit8: TEdit;
    Button1: TButton;
    Edit9: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Memo1: TMemo;
    Label10: TLabel;
    Button2: TButton;
    Button3: TButton;
    Label11: TLabel;
    Label12: TLabel;
    Timer1: TTimer;
    Label13: TLabel;
    Edit10: TEdit;
```



```

Button4: TButton;
procedure Button1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
  my File : Text File;
  text : string;
source,target,target1,target2,target3,target4,target5,target6,target7:string;
  I,ss:Integer;
  x,count:Integer;
  a,b,c,d,e,f:Real;
  var
  buttonSelected : Integer;
begin
  count:=0;
  ss:=0;
  // Try to open the Test.txt file for writing to
  AssignFile(myFile, 'sample.txt');
  // Reopen the file for reading
  Reset(myFile);
  i:=1;
  // Display the file contents
  while not Eof(myFile) do
  begin
    ReadLn(myFile, text);
    Edit1.Text:=text;

```

```
Sleep(100);
Edit1.Refresh;
//1
source:=Edit1.Text;
target1:=Copy(source,1,5);
Edit2.Text:=target1;
Edit2.Refresh;
target2:=Copy(source,7,5);
Edit3.Text:=target2;
Edit3.Refresh;
target3:=Copy(source,13,5);
Edit4.Text:=target3;
Edit4.Refresh;
target4:=Copy(source,19,5);
Edit5.Text:=target4;
Edit5.Refresh;
target5:=Copy(source,25,5);
Edit6.Text:=target5;
Edit6.Refresh;
target6:=Copy(source,31,5);
Edit7.Text:=target6;
Edit7.Refresh;
target7:=Copy(source,38,6);
Edit8.Text:=target7;
Edit8.Refresh;
if (target7='Attack') then
begin
    count:=count+1;
    Edit9.Text:=IntToStr(count);
    Edit9.Refresh;
end;
```



```

    if (ss=3) then
    begin
        Edit10.Text:='Admin';
        Edit10.Refresh;
        // Show a confirmation dialog
        buttonSelected := messagedlg('Confirmation',mtError, mbOKCancel, 0);
        // Show the button type selected
        if buttonSelected = mrOK then ShowMessage('Pass');
        if buttonSelected = mrCancel then ShowMessage('Block');
        end;
        if (ss=4) then
        begin
            Edit10.Text:='Admin';
            Edit10.Refresh;
            // Show a confirmation dialog
            buttonSelected := messagedlg('Confirmation',mtError, mbOKCancel, 0);
            // Show the button type selected
            if buttonSelected = mrOK then ShowMessage('Pass');
            if buttonSelected = mrCancel then ShowMessage('Block');
            end;
            ss:=0;
        end;
        // Close the file for the last time
        CloseFile(myFile);
    end;
    procedure TForm1.Timer1Timer(Sender: TObject);
    var
        t,d:TDateTime;
    begin
        t:=Time;
        d:=Date;
        Label11.Caption:=TimeToStr(t);
        Label11.Refresh;
        Label12.Caption:=DateToStr(d);
        Label12.Refresh;
    end;
    procedure TForm1.Button2Click(Sender: TObject);

```

```
begin
  Memo1.Lines.SaveToFile('log.txt');
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
  sssssMemo1.Lines.Clear;
end;
end.
```