UGANDA TECHNOLOGY & MANAGEMENT

UNIVERSITY

For an Open Mind

A signature-based detector model for Probe and Denial of service attacks using datamining techniques.

By

Babirye Claire

SEP15/COMP/0645U

Supervisor

Ernest Mwebaze (PhD)

UTAMU

A Proposal Submitted To the Graduate School for a Dissertation in Partial Fulfilment of the Requirement for the Award of Msc. Computing at Uganda Technology And Management University.

June, 2017

Chapter 1

Introduction

1.1 Background

In this era, we are in the midst of a computer revolution; data communication networks such as the Internet are now being used to connect millions of computers and personal networks at various organizations, like e-commerce and banking organizations. This has simplified their work in real life. However, in parallel with the ever increasing network sizes has been a concomitant increase in the network traffic data which contains highly confidential and valuable information communicated over the network [1]. To the network administrators and analysts this traffic is resourceful to understand network behavior, provide quality of service and set proper information security policies through monitoring network misuse and ensuring network security. Nowadays it is very important to maintain a high level security to ensure safe and trusted communication of information between various organizations. However, secured data communication over internet and any other network is always under threat of intrusions and misuses. To control these threats, recognition of attacks is a critical matter. These attacks can be recognized through network monitoring; a continuous process that involves inspecting any or all kind of traffic that traverses a particular system or network of interest; to quickly detect anomalies with in traffic behaviour, such as attacks initiated by perpetrators looking to bring down a system or destroy or steal sensitive information [2].

Network monitoring has been facilitated by the use of Intrusion Detection Systems; IDS. An IDS can be classified as an anomaly IDS or a misuse IDS. The misuse detection system performs the monitoring for intrusion detection through matching audit data with known patterns of intrusive network behavior; and anomaly detection systems identify abnormalities from the normal network behavior. To recognize traffic as an attack, IDS must be taught to recognize normal activity.

However, there are no known models for normal network behavior, making it hard to develop an anomaly detector in the strictest sense [3]. Based on the inherent complexity in characterizing the normal network behavior, the problem of anomaly detection can be categorized as model-based and non-model based. In model-based anomaly detectors, it is assumed that a known model is available for the normal behavior of certain aspects of the network and any deviation from the norm is deemed as an anomaly [4].

Furthermore, Intrusion detection can be either host-based or network-based depending on where they look for intrusions. A host based IDS monitors activities associated with a particular host, and a network based IDS considers the whole network traffic [4] [5].

In this study we propose a network-based misuse detector model based on datamining techniques that can be used to predict legitimate network behavior and behavior that deviates from the normal state referred to as anomalies. We focus on mainly 2 common attacks: Denial of Service attacks; DoS and Probe attacks. In common, both of these attacks have a common characteristic of utilizing many packets as seen by the network interface, which are different from other attacks which may use as low as one packet to complete an attack [4] [6]. Probes and DoS attacks are some of the attacks which affect most networks globally on a daily basis. Detection of these attacks is still a major research topic for researchers throughout the world.

A signature-based detector model must have access to large volumes of data that can provide the required samples, from which accurate estimates of a legitimate network behavior, and anomaly like network behaviour is made [5]. Incoming patterns that match an element of the malicious library are labelled as attacks, however, unknown attacks that do not deviate much from the attacks listed in the library can be detected and labelled as neighbouring attacks. Thus we carry out the study using the Knowledge Discovery and Datamining (KDD) data set which is a widely used data set to evaluate intrusion detection systems.

1.2 Problem Statement

Organizations experience large volumes of network traffic that exhibit numerous different characteristics, identifying which traffic is malicious and which is not; can be difficult. Most of the current methods of detecting malicious traffic provide a complex rule based model which cannot identify unique patterns that don't match any rules in the rules database. In order to discover malicious patterns in network traffic effectively, we propose an accurate intrusion detection model that uses data mining techniques to detect and visualize attacks from both known and unknown patterns previously.

1.3 Objectives

1.3.1 General Objectives

To develop a model that can be used to detect DoS and Probe attacks using data mining techniques.

1.3.2 Specific objectives

- To collect and preprocess sufficient training data (both malicious and non-malicious) in order to model the system.
- To import the collected traffic to datamining platforms; Pandas, for analysis.
- To design a model that does analytics on collected data to identify malicious.
- To test and evaluate the model.

1.4 Justification

- As a result of the increase in the network traffic, accurate network monitoring is a necessity for networks to ensure security measures (for confidentiality, integrity, availability triad) are met.
- Model facilitates discovery of meaningful patterns, insights and associations in traffic data through visualization that can be used to infer to the network policies to be configured on the network.

Chapter 2

Literature Review

Our goal is to distinguish lethal traffic from the legitimate traffic. Thus we look at lethal traffic, legitimate traffic, current methods of distinguishing them, and how these methods are evaluated.

2.1 Network Traffic

Networks are mainly known to facilitate communication and information sharing, this makes them indispensable since information and communication are two of the most important strategic issues for the success of every enterprise [7]. Nearly today every organization uses a substantial number of computers and communication tools that are facilitated by networks such as the Internet to run day to day activities. Internet is a network of networks that facilitates various services such as online communication, information sharing while overcoming geographic separation problem.

These various activities that take place on the network form network traffic or data traffic and that is the amount of data moving across the network at a given point in time. This traffic is mainly categorized into two; Legitimate traffic and lethal traffic. Legitimate traffic includes the legit packets that are sent by a legitimate user on the network without any bad motive. Lethal traffic those are the malicious packets sent on the network by an attacker who lies somewhere on the network. Such packets are sent by attackers who have different bad motives for example; with an intent of exploiting a vulnerability on the network and thus launching some form of attack.

From thousands of known exploits, [8] [9] [10] describe a taxonomy of attacks, grouping them into four categories: Probes, Denial of service attacks, Remote to Local attacks and User to Root attacks. These are explained below:

- Probe attacks - These are launched when an attacker is testing a potential target to gather information [11]. They are operated with an essence of identifying a weakness in a machine that can be exploited so as to compromise the system [8]. They are usually harmless (and common) unless a vulnerability is discovered and later exploited. According to [12], it is known that before launching the attack, the attacker selects a target and gathers information.
- Denial of service attacks- also known as DoS attacks, such aim at preventing normal operation of the network, such as causing the target host or server to crash, or blocking the network traffic [8]. This happens through overwhelming the target with high volumes of traffic making it unavailable to legitimate users [13]. Such attacks degrade the performance of a network.
- User to Root- these are attacks in which an authenticated user bypasses normal authentication gaining the privileges of another user, usually root [14].
- Remote to Local- unlike user to root attacks, for this case [14]; the root privileges are gained by an unauthorized user who is able to bypass normal authentication through exploiting the vulnerabilities in the system [8].

2.1.1 Probes

Probes gather information to search for vulnerable systems [10] This happens through the attacker performing a scan on a machine or a networking device in order to determine weaknesses or vulnerability that may be later exploited so as to compromise the system [8]. Probes can be launched through a couple of activities such as: inside sniffing; port scans; ip sweep, vulnerability testing, among others.

- Inside sniffing, where an attacker with physical access to a broadcast medium such as Ethernet medium, or wireless medium, sniffs traffic addressed to others on the local network. This is easily done since some protocols such as telnet, File Transfer Protocol (FTP), Post Office Protocol3 (POP3), Internet Message Access Protocol

(IMAP) and Simple Network Management Protocol (SNMP) transmit unencrypted passwords.

- IP sweep which involves testing a range of IP addresses with ping to see which ones are alive [9]. The attacker can also gather a list of potential targets through spoofing a zone transfer request to a DNS server.

- Port Scans- this action involves testing for ports with listening servers that is to say active ports on a machine [9]. Tools such as Network Mapper (NMAP) use sophisticated techniques to make scans hard to detect, for example scanning with Reset (RST) of Finished (FIN) packets which are less likely to be logged. With such tools a probe attack can be launched from a compromised host. For example; performing an idle scan through an intermediate host. Idle scanning is a feature of NMAP that allows an attacker to conceal its address by exploiting any intermediate host that is lightly loaded yielding predictable IP fragment ID values, for example, the ID may be incremented after each packet is sent [15].

  This can happen through the attacker probing the intermediate host on an open port such as a web server on port 80 to get the current ID. The attacker then sends a TCP SYN packet to the target port to be probed, with the spoofed source address of the intermediate host. The source port is set to 80 and the target responds to the intermediate host on port 80 either with a SYN-ACK packet of the port is open or s RST if closed. The intermediate host then replies (with a RST) to the target in case of a SYN-ACK (since no TCP connection was open), but does not respond to a RST from the target. The attacker probes the intermediate host again to see whether the IP ID is incremented by one or two thus learning if the port is open or closed [10] [15].

- Vulnerability testing- This is done by use of various tools such as Motorola Scalable Controller Area Network (MSCAN), NESSUS; which can be used to test for a wide range of vulnerabilities. NESSUS is open source and it has an extensive library of tests, which is updated on discovery of new vulnerabilities. However much as these systems allow network administrators to quickly test their own systems for vulnerabilities, the same provide a platform for attackers to test someone else's system.

2.1.2 Denial of Service Attacks

Denial of Service attacks are launched with a basic purpose of over hauling the network so as to deny authentic user services of the network [2]. DoS attacks are launched at different layers of the OSI or TCP/IP model [16]. At the physical layer through the signal jamming attack disabling normal communication; at the link layer, malicious nodes can capture the channel or medium and prevent other nodes from channel access; at the network layer DOS attacks are mounted on routing protocols and disrupt the network performance through flooding various types of routing packets; at the transport and application layers, DOS attacks happen through [9] SYN flooding, session hijacking and malicious programs [8].

Denial of service attacks target a server, a host or a network. They either flood the target with data to exhaust resources, or use malformed data to exploit a bug. Swati [8] and Kendall [9] give the following examples of DoS attacks,

- SYN flood (Neptune)- A server is flooded with TCP SYN packets with forged source addresses [16]. Because each pending connection requires saving some state information, the target TCP/IP stack can exhaust memory and refuse legitimate connections, since all system resources have been consumed.

- Smurf [16]- An attacker floods the target network by sending ICMP ECHO REQUEST (ping) packets to a broadcast address (x.x.x.255) with the spoofed source address of the target. The target is then flooded with ECHO REPLY packets from multiple sources.

- Apache2- some versions of the apache web server will run out of memory and crash when sent a very long HTTP request. Kendall describes one version in which the line 'User-Agent: Sioux' is repeated 10,000 times.

- Mail bomb-a user is flooded with mail messages.

- Ping of death - many operating systems could be crashed by sending a fragmented IP packet that resembles to 65,536 bytes, one byte larger than the maximum legal size. It is called the 'ping of death' because it could be launched from Windows 95 or NT with the command "ping- 65510 target".

- Process table- An attacker opens a large number of connections to a service such as finger, POP3 or IMAP until the number of processes exceeds the limit. At this point no new processes can be created until the target is rebooted.
- UDP storm- this attack sets up a network flood between two targets by sending a spoofed UDP packet to the echo server of one target with the spoofed source address of one target and the port number of the chargen server of the other target.

2.2 Current Detection Methods

2.2.1 Detection of DoS and Probe attacks using the Genetic Algorithm

The algorithm is based on the Darwin's theory of evolution; with a basic rule of Survival for the fittest, the algorithm handles a population of possible solutions where each solution is represented through a chromosome [8]. A chromosome is a threadlike structure of nucleic acids and protein found in the nucleus of most living cells, carrying genetic information in form of genes. The algorithm uses evolution and natural selection evolving chromosomes using selection, combination and mutation operators [17]. When the Genetic Algorithm is used for solving various problems three factors are considered to have a vital impact on the effectiveness of the algorithm and also of the applications. These factors include: fitness function, representation of the individuals and the parameters for the Genetic Algorithm. The determination of these factors often depends on applications and/or implementation [8].

Working Principle of the algorithm

The process undertaken in the algorithm begins from an initial population of randomly generated individuals which stand for a possible solution of a problem that are considered as candidate solutions [5]. Then the population is evolved [8] for a number of generations while progressively improving the qualities of individuals by increasing the fitness value as the measure of quality [17]. During each generation; selection, cross over, and mutation are one after the other applied to each individual as shown in Figure 1, with certain probabilities. Selection is the phase where population individuals with better fitness are selected otherwise it gets damaged. However, the selection phase of chromosomes is biased towards the fittest chromosomes for survival and combination [8]. Taking basis on a user-defined fitness function [5], the numbers of the best-fit individuals are selected first and then the remaining individuals are selected later on and paired with each other. Each individual pair produces one offspring by partially exchanging their genes around one or more randomly selected crossing points. At the end a certain number of individuals are selected, and the mutation operations are applied [17]. The fitness function is used as the evaluation function and is used to calculate the goodness of each chromosome according to the desired solution [8]. At that point crossover is done to simulate natural reproduction as mutation mutates species. To get a better string, a cross over operator is used which recombines two strings to get a better one [5]. The two strings participating in the crossover operation are normally known as parent strings and the resulting strings are known as child strings [8]. The cross over operator recombines good sub-strings from good strings together, hopefully to create a better substring. Mutation in a way is a process of randomly disturbing genetic information. It adds new information in a random way to the genetic search processes by introducing diversity in the population whenever the population tends to become homogeneous due to repeated use of reproduction and cross over operators.

How the algorithm functions in relation to detection of attacks in network traffic.

The algorithm works in two phases; learning [8] or training phase and the testing phase as shown in figure 2.

Learning/Training phase.

In the learning phase [17], network data which contains both normal network connections (normal network data) and attacks (abnormal data) is collected for audit. Then a network sniffer analyses this data and sends it to the genetic algorithm and the fitness function is applied to generate a set of rules for detecting intrusion. These rules are stored in a rule base.

The records from the learning phase are represented in the form of chromosomes. Each chromosome is a rule within which certain features of a connection are encoded in

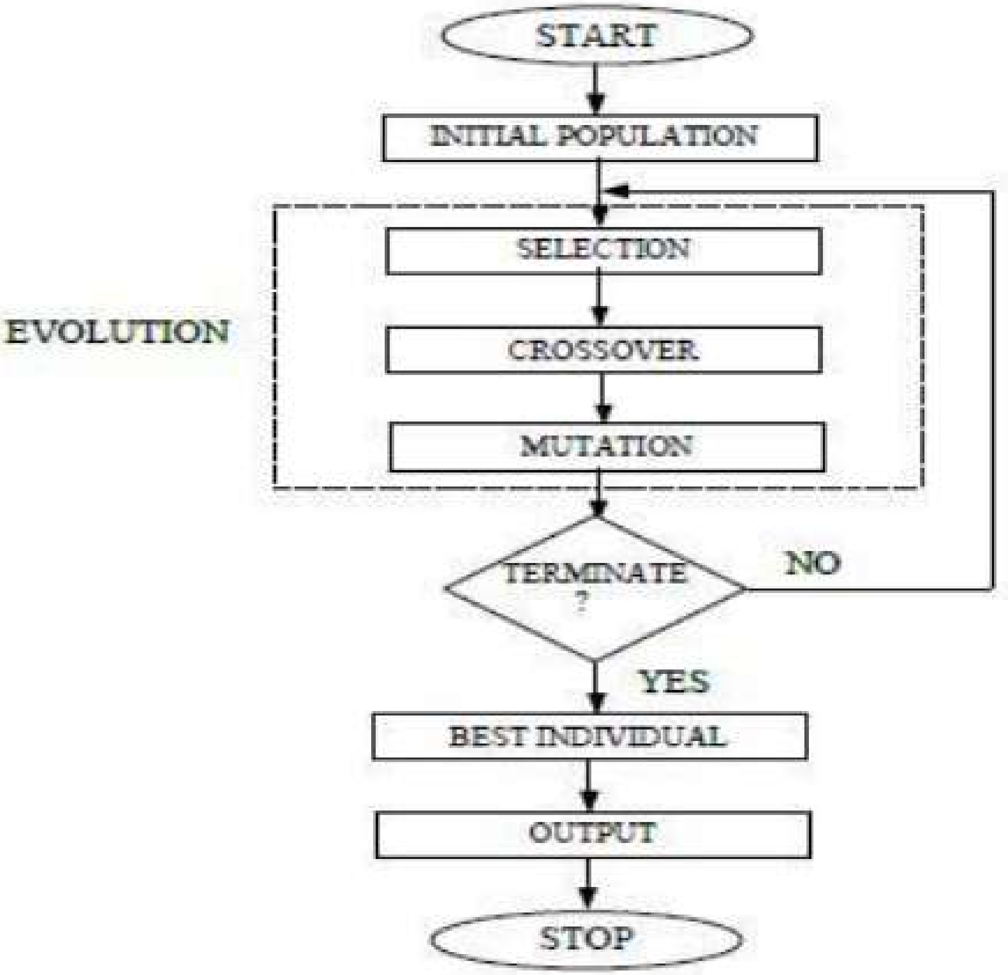


Figure 2.1: Structure of the Genetic Algorithm

the form of fixed length vector. A fitness function is then applied to each chromosome in order to evaluate its goodness. If a chromosome helps to identify an attack correctly, it is considered good or fit else it is considered bad [17]. The algorithm proceeds with mutation and combination operators; combination and mutation operators are applied to the good chromosomes obtained from the fitness function to produce a new generation. The entire process is recurred by using the newly generated population. Thus the evolution process is repetitive until a solution is reached; a set of rules capable of detecting attacks is generated [5]. In the rule base, the rules are stored in the following format [17]:

if condition then act

For example, a rule can be defined as [8]:

if the connection has following information: source IP address 145.33.17.6; destination IP address 160.106.20.55; destination port number: 21; connection time: 10.1 seconds then stop the connection

This implies: if there exists a network connection request with source IP address 145.33.17.6, destination IP address 160.106.20.55, destination port number 21, and connection time 10.1 seconds, then stop the connection establishment - since IP address 145.33.17.6 is recognized by the IDS as a blacklisted IP address.

Thus, service request initiated from it, is rejected.

Testing phase.

The testing entails detection of whether a real-time network connection is a normal connection or it is an intrusive attack [17]. This is obtained using rules stored in the rule base during the training phase. Since the algorithm is rule-based, if the characteristics of a new connection match with the condition section of some pre-defined rule in the rule base then the connection is considered as an attack else it is considered as a normal connection [5].

The sub attack labels such as smurf, mailbomb, among others are recognized with respect to the fitness criteria by selecting the best-fit chromosomes capable of detecting the attacks from every population [8].

Incase an attack is detected then IDS performs the necessary actions as defined by the security policies of the organization. The algorithm is desirable because of various reasons such as it is easily retrainable, the result gets better with time, provides room for

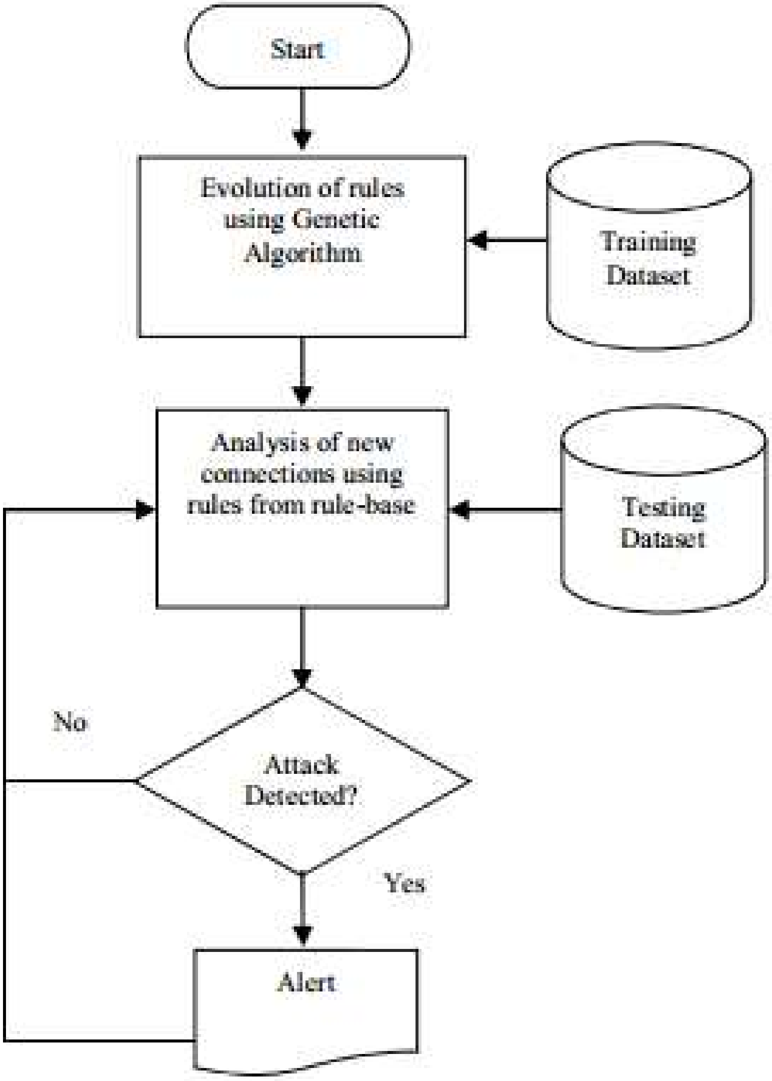

Start
Evolution of rules using Genetic Algorithm
Training Dataset
Analysis of new connections using rules from rule-base
Testing Dataset
No
Attack Detected?
Yes
Alert

Figure 2.2: Flow of the Genetic Algorith based

exploiting previous or alternate solutions [8]. As a result of the multiple offspring, the algorithm is intrinsically parallel; the solution space can be explored in multiple directions at once. this makes it suitable in solving problems where space of potential solution is truly large [17].

However, the algorithm has various limitations such as; it is a difficult task to represent a problem space in the algorithm, find the fitness function as well as choosing parameters for the algorithm and yet such factors determine the performance and effectiveness of the algorithm [5]. Configuration of a genetic algorithm based system is also known to be a hard task.

2.2.2 Detection of DoS and Probe attacks using the Principal Component Analysis (PCA)

The model uses a multivariate statistical method called Principal Component analysis to detect Denial-of-service and network Probe attacks. Principal Component Analysis is a multivariate statistical technique [6] applied to reduce the dimension of feature vectors and to achieve parsimony by extracting the smallest number components that account for most of the variation in the original multivariate data and to summarize the data with little loss of information to enable better visualization and analysis of the data [18]. The algorithm inputs data and portions of the data sets are processed to create a new database of feature vectors which represent the IP header of the packets [6]. The feature vectors are analyzed using PCA and various statistics are generated during this process including the principal components, their standard deviations, the loading of each feature on the principal components and bi-plots to represent a graphical summary of these statistics [19].

The variance and standard deviation of a random variable are measures of dispersion. The variance is the average value of the squared deviation from the variable's mean, and the standard deviation is the square root of the variance [20].

For instance, in IPsweep attacks, one or more machines (IPs) are sweeping through a list of server machines looking for open ports that can later be utilized in an attack while in port sweep attacks, one machine is sweeping through all ports of a single server machine looking for open ports. In both cases, there is an irregular use of port numbers that causes the variance in the principle components to vary, with an associated irregularity in the loading values [19].

According to [19], the advantage of the PCA algorithm for detection of anomalies in network traffic is its ability to operate on the input feature's vector space directly without the need to transform the data into another output space as in the case with self-learning techniques.

However, [20] discusses it is possible to create an effective attack which is undetectable by the PCA detector due to the unrealistic manner in which items are rated in attack profiles of standard attacks.

2.3 Weaknesses in the existing current detection methods

- It is a difficult task to represent a problem space in the genetic algorithm, find the fitness function as well as choosing parameters for the algorithm and yet such factors determine the performance and effectiveness of the algorithm [11]. Configuration of a genetic algorithm based system is also known to be a hard task.
- The PCA algorithm has scalability issues; the cause of this is twofold; the algorithm reduces on the dimensionality by removing components with large Eigen values; this affects the sample space making some anomalies not detectable or traceable.

2.4 Proposed System

The proposed system will be a misuse detector model that will be used to detect DoS and probe attacks in network traffic using data mining techniques. Data mining techniques will be applied to; help in distinguishing normal activity from alarm data so as to allow analysts focus on real attacks; identify log, ongoing patterns such as different IP address on the same activity; find anomalous activity that uncovers a real attack. To accomplish this; various techniques will be employed such as defining normal activity and discovering the anomalies; and predicting the category to which the record belongs to.

This will be done by applying classification techniques; machine learning algorithms that will used to map data instances into one of the various predefined categories based on some features. The result of the analysis will be a percentage confidence level on how valid the intrusion detection results are.

Chapter 3

Methodology

3.1 Introduction

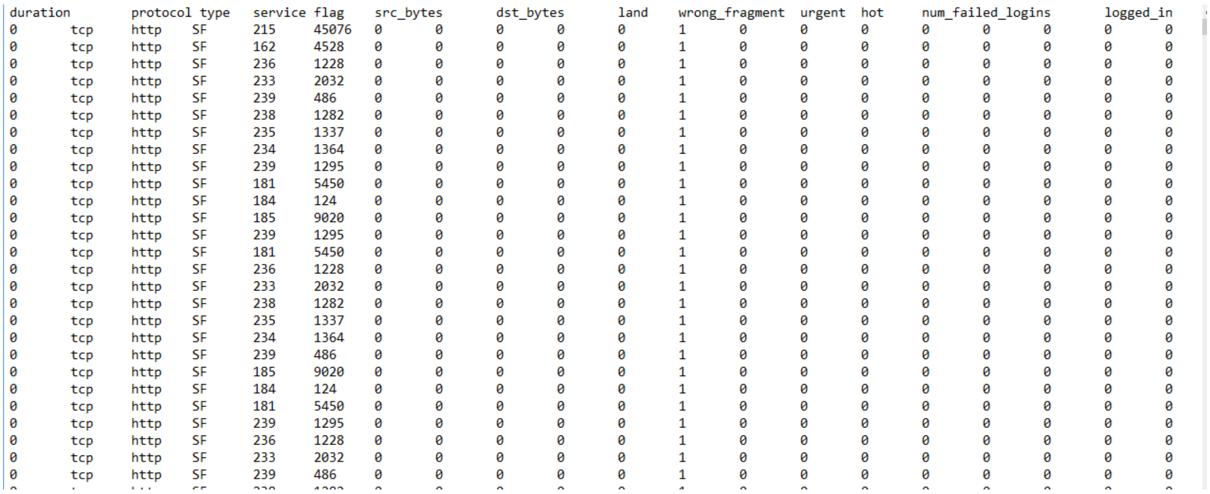
In this section we undertake to elaborate on the main methods, tools and techniques that will be used to develop the proposed approach. The methodology provided will be followed so as to provide a solid realistic working application structure. Methodology includes tools, approaches, processes and techniques that will be used in data collection, analysis and feature extraction, model design, and testing of the model.

3.2 Data Mining Process

Data mining is a process that involves extraction of information to discover useful and previously unknown information. The process includes various steps: data collection, data munging, data analysis and data visualization.

3.2.1 Data Collection

The first step in the datamining process involves collecting or gathering the data on which the analysis is to be done on. Data to be used is categorized into two: training data set and test data set. The training data set is obtained from the KDD intrusion detection data set as the testing data set will be collected using Wireshark, a network sniffer tool that enables viewing and capturing traffic on a specific interface; a Wi-Fi Network Interface. The data files for the captured sessions will be saved as PCAP files, and imported into comma separated value format for munging and analysis. The data



```
duration        protocol type   service flag    src_bytes       dst_bytes       land    wrong_fragment  urgent  hot     num_failed_logins       logged_in
0       tcp     http    SF      215     45076   0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      162     4528    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      236     1228    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      233     2032    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      239     486     0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      238     1282    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      235     1337    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      234     1364    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      239     1295    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      181     5450    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      184     124     0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      185     9020    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      239     1295    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      181     5450    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      236     1228    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      233     2032    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      238     1282    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      235     1337    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      234     1364    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      239     486     0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      185     9020    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      184     124     0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      181     5450    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      239     1295    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      236     1228    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      233     2032    0       0       0       0       0       1       0       0       0       0       0       0       0       0
0       tcp     http    SF      239     486     0       0       0       0       0       1       0       0       0       0       0       0       0       0
```

Figure 3.1: Sample of the training dataset obtained from KDD cup data

will be collected from the school internetwork Wi-Fi interface, and used during the testing phase.

3.2.2 Data Munging

Data Munging is the art of dealing with and/or converting ill formatted data into a format that more easily learns itself for analysis. This is a very crucial step in the process since the data has to be organized prior to the analysis phase. This process will be carried out on the collected dataset.

3.2.3 Feature Selection Phase

This is a pre-processing phase that involves selection of features that will be used in the construction of the algorithm. The effectiveness of an intrusion detection system is dependent on the features selected. The dataset collected by the network sniffer tool contains various attributes or features. Some of these features are of use to the algorithm while performing the learning task, as others are noise since they donot make any contribution whilst training the algorithm.

3.3 Model Design

3.3.1 Learning Phase

Learning phase will involve training an algorithm so as to learn of the features in the data set and be able to do future prediction. Machine Learning can be either supervised learning form or unsupervised learning form. With supervised learning form the algorithm learns or is trained on a known data set to make predictions. This data set is referred to as known since it is labelled. For unsupervised learning; training happens on unlabelled dataset.

Supervised learning will be used to train the algorithm for future prediction. This will be done over the KDD intrusion detection dataset.

3.3.2 Testing Phase

After training the algorithm then it is tested by feeding it with a new unknown data set so as to make predictions basing on the knowledge it has obtained from the feature during the training process. Testing will be done on the data set that will be collected on the school Wi-Fi (UTAMU) network.

3.4 Model Evaluation

Evaluation will be made to measure how well the trained model makes predictions on the test dataset. The defined problem is a classification problem; solved through classifying the network traffic dataset, and thus the model will be evaluated using a confusion matrix. Confusion matrix is a two by two-matrix that compares the actual class or category with the predicted class. This will be used to evaluate the model through the use of various metrics such as accuracy.

3.5 Architectural Framework of the proposed method

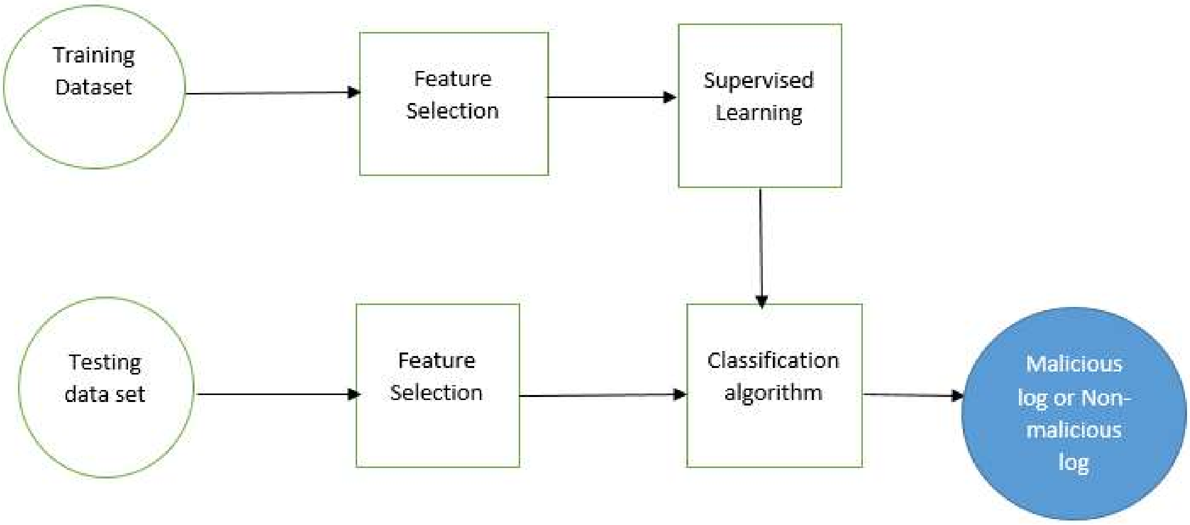


Figure 3.2: Operation of the proposed method

REFERENCES

[1] Becker, R. A., Eick, S. G., & Wilks, A. R. (1995). Visualizing network data. IEEE Transactions on visualization and computer graphics, 1(1), 16-28.

[2] Aggarwal, N., & Dhankhar, K. (2014). Attacks on Mobile Adhoc Networks: A Survey. International Journal of Research in Advent Technology, 2(5), 307-316.

[3] Thottan, M., Liu, G., & Ji, C. (2010). Anomaly detection approaches for communication networks. In Algorithms for Next Generation Networks (pp. 239-261). Springer London.

[4] Jyothsna, V., Prasad, V. R., & Prasad, K. M. (2011). A review of anomaly based intrusion detection systems. International Journal of Computer Applications, 28(7), 26-35.

[5] Kshirsagar, V. K., Tidke, S. M., & Vishnu, S. (2012). Intrusion detection system using genetic algorithm and data mining: An overview. International Journal of Computer Science and Informatics ISSN (PRINT), 2231, 5292.

[6] Labib, K., & Vemuri, V. R. (2004, June). Detecting and visualizing denialof-service and network probe attacks using principal component analysis. In Third Conference on Security and Network Architectures, La Londe,(France).

[7] Patra, M. P. M. R. (2009). Evaluating machine learning algorithms for detecting network intrusions. Int. J. of Recent Trends in Engineering and Technology, 1(1).

[8] Paliwal, S., & Gupta, R. (2012). Denial-of-service, probing & remote to user (R2L) attack detection using genetic algorithm. International Journal of Computer Applications, 60(19), 57-62.

[9] Kendall, K. (1999). A database of computer attacks for the evaluation of intrusion detection systems. MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

[10] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on (pp. 1-6). IEEE

[11] Kayacik, H. G., Zincir-Heywood, A. N., & Heywood, M. I. (2005, October). Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets

[12] Hansman, Simon Luke. "A taxonomy of network and computer attack methodologies." (2003).

[13] Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: classification and state-of-the-art. Computer Networks, 44(5), 643-666.

[14] Sathya, S. S., Ramani, R. G., & Sivaselvi, K. (2011). Discriminant analysis based feature selection in kdd intrusion dataset. International Journal of Computer Applications, 31(11), 1-7

[15] Portnoy, L., Eskin, E., & Stolfo, S. (2001). Intrusion detection with unlabeled data using clustering. In In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001).

[16] Alkasassbeh, M., Al-Naymat, G., Hassanat, A. B., & Almseidin, M. Detecting Distributed Denial of Service Attacks Using Data Mining Techniques. International Journal of Advanced Computer Science & Applications, 1(7), 436-445.

[17] Shaveta, E., Bhandari, A.,& Saluja, K. (2014). Applying Genetic Algorithm in Intrusion 7Electrical Engineers.

[18] Revathi, S., & Malathi, A. Detecting Denial of Service Attack Using Principal Component Analysis with Random Forest Classifier. International Journal of Computer Science & Engineering Technology, 1(5), 248-252.

[19] Labib, K., & Vemuri, V. R. (2004, June). Detecting Denialof-service and network probe attacks using principal component analysis. In Third Conference on Security and Network Architectures, La Londe,(France).

[20] Hurley, N., Cheng, Z., & Zhang, M. (2009, October). Statistical attack detection. In Proceedings of the third ACM conference on Recommender systems (pp. 149-156). ACM