International Journal of Recent Engineering Research and Development (IJRERD)
ISSN: 2455-8761
www.ijrerd.com || Volume 03 – Issue 06 || June 2018 || PP. 33-39

Enhancing Security System for distributed database using an Improved Anomaly Detection Technique

Adewumi, Moradeke Grace[1], Ahigbe, Benedict[2], Afolabi, Babajide Samuel[3]
[1]Department of Computer Science,
College of Education, Ikere- Ekiti, Nigeria
[2,3] Department of Computer Science and Engineering,
Obafemi Awolowo University, Ile-Ife, Nigeria

Abstract: This study aimed at developing an enhance security system for distributed database by simulating an existing anomaly detection algorithm and proposed an improved one for the system. KDD99 data set was used as a standard data set. The data set consisted of nine weeks of raw transmission control protocol (TCP) dump data from the network. Forty-one unique attributes were compiled from each raw TCP packet sequence. MATLAB was used as a simulation tool to simulate the existing anomaly detection algorithm. Semi-supervised clustering was used to perform feature subset selection to reduce the data set and output was classified using genetic algorithm. The proposed system was evaluated using detection rate and false alarm rate as performance parameters. The responsiveness of each simulation was measured at the end to check the percentage or the nearness to accuracy by getting the True Positive, True Negative, False Positive, False Negative, Specificity, Sensitivity, False Positive Rate, False Negative Rate, and Accuracy. The result findings of the fitness function of the 50% trainings carried out for the proposed system showed that the system performed better, because the sensitivity rate is higher at that point and grows faster. Furthermore, the result of comparison between the existing models and the proposed model showed that the existing system detected 1782 anomalies, while the proposed model detected 13,661 anomalies. As a result, the false positive rate for the proposed model demonstrated more detective ability, since fewer alarms of 1374.2 times was raised as against 33,036 times for the existing model. It can therefore be concluded that the proposed model is more efficient in detecting anomalies than the exiting model.

Keywords: Sensitivity, Specificity, Anomaly, Detection.

Introduction

Every creature on this planet be it living things or non-living things need security for one reason or the other. The rapid expansion of the Internet in recent years has exposed computer systems to increased number of security threats. Despite numerous technological innovations for information assurance, it is still very difficult to protect computer systems (Lee *et al.*, 2001). Different but complementary techniques have been developed and deployed to protect organisations' computer systems against malicious attacks. Some of these techniques are: firewall, message encryption, secured network protocols, password protection, etc. There are utility software, such as antivirus and malware that are also useful for combating any attacks on computer systems. Despite the use of all these mechanisms, it is still nearly impossible to have a completely secured system (Gong *et al.*, 2005). Therefore intrusion detection is turning into an undeniably critical innovation that monitors networks activities becoming an increasingly important technology that monitors network traffic and identify network intrusions such as anomalous network behaviours, unauthorized network access, and malicious attacks on computer systems (Ilgun, 1992).

Networks are complex interacting systems and are comprised of several individual entities such as routers and switches. The behaviour of individual entities contributes to the ensemble behaviour of the network. The way Internet protocols were evolved makes it difficult to fully understand the dynamics of the system (Thottan and Ji, 2003). However, Network Technologies span data storage systems, encryption and authentication techniques, voice and video over IP, remote and wireless access as well as Web services. As technologies increase daily users are equally facing numerous threats and challenges such as: confidentiality of the system and its data, integrity of the system and its data, non-repudiation, authentication, and also threat to availability of the system and its data. In view of these threats, network security controls are the major concern to network users and owners. The security control may include: vulnerability avoidance, attack detection and neutralization, exposure and recovery. In this paper an enhanced anomaly technique would be developed based on detection rate and false alarm rate to detect intrusion on distributed systems.

Objectives of the Research

The aim of this research is to develop an enhanced anomaly detection technique for detecting security breach in distributed network particularly at the point of intrusion

.

Overview of Research Methodology

The existing anomaly detection model was simulated to ascertain its level of detection. As a consequence, an enhanced anomaly detection model was developed using the principal component analysis technique based on data mining principle. The network data were collected and processed to reduce its high-dimensionality to manageable size. The resulting data was classified using Semi-Supervised Genetic Algorithm with an embedded objective function formulated as an optimisation function to detect abnormal data. The improved algorithm was simulated using MATLAB with KDD99 dataset and evaluated using detection rate and false alarm rate as performance parameters.

Justification of the Study

As distributed network system increases daily, intruders' activities also increase and intrusion detection system are not strong enough to detect some of these intrusions. On this platform, this study developed an improved anomaly detection technique using the semi-supervised learning approach in order to detect intrusion at the point of entry.

Literature Review

As advances in networking technology help to connect the distant corners of the globe, internet also continues to expand its influence as a medium for communications and commerce. The threat from spammers, attackers and criminal enterprises has also grown accordingly. It is as a result of such threats that have made intrusion detection systems in the cyberspace equivalent to the burglar alarm. As a result, its joins rank with firewalls as one of the fundamental technologies for network security. However, today's commercially available intrusion detection systems are predominantly signature-based intrusion detection systems that are designed to detect known attacks by utilizing the signatures of those attacks (Cannady, 1998). Such systems require frequent rule-base updates and signature updates, and are not capable of detecting unknown attacks. In contrast, anomaly detection systems, a subset of intrusion detection systems, model the normal system/network behaviour which enables them to be extremely effective in finding and foiling both known as well as unknown or "zero day" attacks. While anomaly detection systems are attractive conceptually, a host of technological problems need to be overcome before they can be widely adopted. These problems include high false alarm rate and failure to scale to gigabit speeds and so on (McHugh, 2001).

Intrusion detection system gathers and analyses information from various areas within a computer or a network to identify possible security breaches. In other words, they detect actions that attempt to compromise the confidentiality, integrity or availability of a system/network (Adetunmbi *et al.*, 2006). In general, these systems automate the process of extracting intelligence about past or present actions that attempt to compromise the confidentiality, authentication, authorization, integrity, non-repudiation, availability (and so on) of a resource. The definition of an intrusion in this context is not fixed, but rather it is a concept that changes depending on the administration or objective of the system. More specifically, the intelligence and information provided by IDS is contingent upon how the system is being used. Therefore the system is as important as the chosen IDS itself. Indeed, there are many ways to use IDSs. If and when an IDS discovers an intrusion, regardless of how it has been defined, it is common for a system to make a record or report of the intrusion. Typically, this is done by way of logging (or generating) an alert that is sent off to an appropriate party (Bishop, 2002). More and more, these systems are built to act not only as a judge of intrusion, but also to react to them.

Anomaly Detection

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behaviour. These non-conforming patterns are often referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants in different application domains (Patcha and Park, 2007). Of these, anomalies and outliers are two terms used most commonly in the context of anomaly detection; sometimes interchangeably. Anomaly detection can be used in a wide variety of applications such as fraud detection for credit cards, military surveillance for enemy activities and many others.

The importance of anomaly detection is due to the fact that anomalies in data translate to significant (and often critical) actionable information in a wide variety of application domains. For example, an anomalous traffic pattern in a computer network could mean that a hacked computer is sending out sensitive data to an unauthorized destination (Kumar and spafford 1994, as cited in Patcha and Park, 2007).

Anomaly detection techniques

This is the technique used to detect novel attacks as well as zero day attacks on the network. It has a wide range for fraud detection for credit cards, insurance or health care, intrusion detection for cyber-security, fault detection in safety critical system and others. An anomaly detection approach usually consists of two phases: A training phase and a testing phase. In the former, the normal traffic profile is defined; in the latter, the learned profile is applied to new data (Patcha and Park, 2007).

Semi-Supervised Genetic Algorithm Clustering Technique

In order to perform feature subset selection on the high-dimensional data and optimization of the anomaly detection system, the Semi-Supervised Genetic Algorithm (SSGA) clustering approach is adopted. Demiriz *et al.,* (1999) proposed a semi-supervised clustering using Genetic Algorithm. This approach is meant to combine the benefit of both supervised and unsupervised clustering to solve classification problems.

Semi-supervised clustering algorithm combines the benefits of supervised and unsupervised learning methods. Data are segmented/clustered using an unsupervised learning technique that is biased toward producing segments or clusters as pure as possible in terms of class distribution. These clusters can then be used to predict the class of future points. For example in database marketing, the technique can be used to identify and characterize segments of the customer population likely to respond to a promotion. One benefit of the approach is that it allows unlabeled data with unknown class to be used to improve classification accuracy.

Related work

Akbar *et al.,* (2011) also used the genetic algorithm approach to intrusion detection. In the work, genetic algorithm was used to identify different categories of attacks. The algorithm takes into consideration three different features in network connection. The researchers generated 8 rules with which they achieved their classification. Interestingly, each of the rules set was used to identify specific types of attacks. Genetic Algorithm was implemented and trained on KDD 99 dataset to generate a set of rules that was applied to intrusion detection in order to identify and classify different types of attacks. The system was able to detect some of the attacks accurately because data were reprocessed. The drawback was that it was mainly a misuse detection system while the proposed model is anomaly detection explicitly.

Data Reduction

The data reduction carried out in this study was achieved by carrying out feature subset selection. This was performed by clustering objects with the nearest neighbour(s), as a result of this the centres of the respective clusters were determined. This process was used to facilitate the speed up of the GA during classification process. The algorithm employed for the subset selection is presented as follows.

Algorithm 1

(i) For every object O_i, find the distance to its nearest neighbour

$$d_{NN_j}(O_i) = \| O_i - O_j \| \quad (1)$$

where O_j is the nearest neighbour to O_i and $i \neq j$.

(ii) Compute the average distance of all objects to their nearest neighbour,

$$d_{AVE} = \frac{1}{N}\sum_{i=1}^{N} d_{NN_j}(O_i) \quad (2)$$

(iii) Let $d = scale.d_{AVE,}$

where *d* is computed based on the scale's value;
(assuming initial value to be 0.5);
Now, view the *n* objects as nodes of a graph; &
Connect all nodes that has distance $< | \leq d$;
Then increment scale by 0.1.

(iv) Repeat step (iii) as far as there is no overlap of connected nodes.
<<This is to ensure that all the connected objects are close enough to one another without overlapping the cluster>>

(v) Find all connected nodes and let the data sets represented by these connected nodes be denoted by:

$$B_1, B_2, B_3, \ldots B_{m-1}, B_m \quad (3)$$

where m = the number of connected nodes, and
$m < N$, since B_m consists of 1 or more connected nodes, $i \leq m$.

(vi) Compute Q cluster centres $z_1, z_2, \ldots, z_m$, from all connected components $B_1, B_2, B_3, \ldots B_{m-1}, B_m$ from step (v)

$$\frac{1}{N_i} \sum_{x_{j \in} B_i} x_j, i = 1,2,\ldots,m \quad (4)$$

where N_i is the number of nodes connected in B_i.

This work is consistent with the practice of using the GA, which is found in (Faroun and Rabbi, 2007). Thus, with the aforementioned steps, our data were grouped into nodes, and subsequently, node's centres were determined. This allowed the arrival of more compact and accurate data output. With the foregoing, the further processing of the output data using the GA, made the whole process faster and more accurate.

The Fitness Function

After generating the initial population, the fitness function was used as a metric to select the fit individuals who would undergo crossover and mutation to create the next generation population. Our fitness function is given by the formula:

$$f = \frac{a}{A} - \frac{b}{B} \quad (i)$$

where:

a = the number of attacks connections, the individual correctly classifies out of a total number of attacks;
b = the number of normal connections a network correctly classifies out of normal connections in the population;
A = the total number of attacks; and
B = the normal connections in the population.

Hence, the fitness function value would lie in the region [-1, 1]. A positive value denoted that the individual classifies more number of attacks correctly than it does for the normal ones. To select the fit individuals, a threshold value of 0.95 was set. Thus, all individuals that have a fitness score > 0.95 were selected to produce subsequent generations and were deemed fit otherwise they are not. The design of the fitness function was conceived as presented to make it biased towards individuals that correctly classify only the attack connections, since this was the objective.

Some numbers of fittest individuals were also selected to undergo crossover operations to produce the next generation. This was done cognizance of the assumption that a situation may arise wherein the numbers of fit individuals will be less than the selected number. Therefore, many fit individual will be reproduced to create a parent population of the actual number.

Result and discussion

The existing model used for this study was Genetic Algorithm approach to intrusion detection system by Akbar *et al*., (2011). This was simulated by selecting some specific features as features of interest. These were extracted from all other features to form a new dataset which was like the reduced dataset used in the proposed model. The abnormal features and the selected features were also passed to the GA to search, learn and to get the similar and closely related features. Fifty percent (50%) of the total populations were used as a training data for the genetic algorithm optimization. This means maximizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. The whole normalized vector and the training set was optimized using genetic algorithm. The responsiveness of each simulation was measured at the end to check the percentage or the nearness to accuracy by getting the True Positive, True Negative, False Positive, False Negative, Specificity, Sensitivity, False Positive Rate, False Negative Rate, and Accuracy as shown in table 1 and Table 2.

Table 1: Result analysis generated for the existing model

S/N.	Responsiveness of simulation	Objective function
1.	Anomaly	1782
2.	Normal	63753
3.	True positive	30717
4.	True negative	34818
5.	False positive	33036
6.	False Negative	28935
7.	Specificity	0.51313
8.	Sensitivity	0.51494
9.	False positive rate	0.48687
10.	False negative rate	0.48506
11.	Accuracy	0.51398
12.	Precision	0.48181

Evaluation parameters

Both the existing model and the proposed model were evaluated based on the following performance parameters: Detection rate (or Sensitivity) and False alarm rate (Debar *et al.,* 1999).

Detection rate or sensitivity

This is otherwise called True Positive Rate (TPR)**.** It is given mathematically thus:

$$TPR = \frac{TP}{TP + FN} \quad \text{(i)}$$

False alarm rate

This is called False Positive Rate (FPR). It is given mathematically as:

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

Table 2: Result analysis generated for the proposed model

S/No.	Responsiveness of simulation	Objective function
1.	Anomaly	13661
2.	Normal	12553
3.	True positive	12286.8
4.	True negative	13927.2
5.	False positive	1374.2
6.	False Negative	266.2
7.	Specificity	0.91019
8.	Sensitivity	1.02210
9.	False positive rate	0.089809
10.	False negative rate	0.022145
11.	Accuracy	0.95945
12.	Precision	0.89941

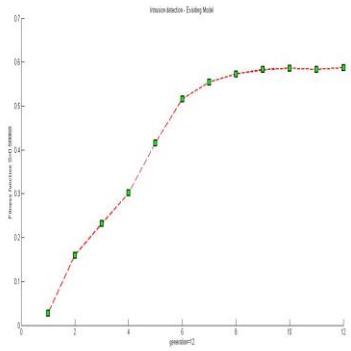

Figure 1: The graph generate for the existing model.

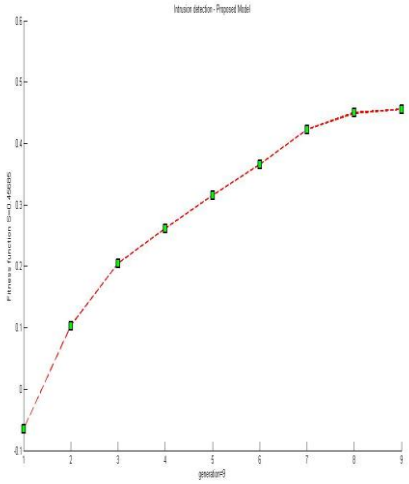

Figure 2: The Graph generated for the Proposed model

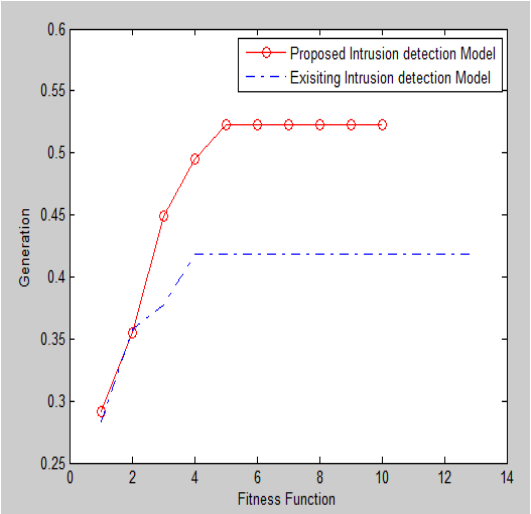

Figure 3: The proposed model versus existing model

Conclusion

The findings of this study indicated that the proposed system improved the detection rate and accuracy over the existing system. Hence the proposed system can be used successfully in the distributed networking environment to capture unauthorized users from accessing the database on the network. Also, it can be used to reduce the false positive rate and false negative rate on the network.

References

[1]. Adetunmbi, A.O., Falaki, S.O., Adewale, O. S., and Alese, B.K. (2008). Network Intrusion Detection Based on Rough Set and K-nearest Neighbour, *International Journal of Computing and ICT Research*, *2*(1): pp. 60-66.

[2]. Akbar, S., Rao, K.N., and Chandulal, J.A. (2011). Implementing Rule based Genetic Algorithm as a Solution for Intrusion Detection System. *International Journal of Computer Science and Network Security*, 11(8): pp 138-144.

[3]. Bishop, M. (2002). Computer Security: Art and Science (2nd Ed.). New York, USA: Addison-Wesley.

[4]. Cannady, J. (1998). Artificial Neural Networks for Misuse Detection. *In Proceeding of the National Information Systems Security Conference*: pp.368–381.

[5]. Debar, H., Becke, B., and Siboni, D. (1999). A Neural Network Component for an Intrusion Detection System, *in Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*. pp. 240–250.

[6]. Demiriz, A., Bennett, K. P. and Embrechts, M.J. (1999). Semi-Supervised Clustering using Genetic Algorithms. Technical Report, Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, New York.

[7]. Denning, D. E., and Neumann, P.G. (2002). Requirement and Model for IDES- A Real Time Intrusion Detection Computer Science Laboratory, *SRLI international, Menlo Park, CA94025-3493, Technical Report.*

[8]. Faroun, K. and Rabbi, A. (2007). Data Dimensionality Reduction Based on Genetic Selection of Feature Subsets, INFOCOMP, *Journal of Computer Science*, 6(36): p. 46.

[9]. Gong, R. H., Zulkernine, M., and Abolmaesumi, P. (2005). A Software Implementation of a Genetic Algorithm based approach to Network Intrusion Detection. In IEEE *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks.* (pp. 246-253).

[10]. Ilgun, k. (1992). A Real-Time Intrusion Detection System for UNIX. Master thesis, University of California, Santa Barbara.

[11]. KDD (2013). KDDCup Data Set. Retrieved from http://kdd.ics.uci.edu/databases/ kddcup99 /kddcup99.html on 20/08/2014 @11.30 am.

[12]. Lee, W., Stolfo, S., Chan, P.K., Eskin, E., Fan, W., Miller, M., Hershkop, S., and Zhang, J. (2001). Real Time Data Mining-based Intrusion Detection. *In Proceedings of DISCEX II*, pp. 13-16.

[13]. McHugh, J. (2001). Intrusion and Intrusion Detection. *Technical Report*. CERT Coordinated Centre, Software Engineering Institute, Carnegie Mellon, University.

[14]. Patcha, A., and Park, J.M. (2007). An overview of Anomaly Detection Techniques: Existing Solutions and latest Technological Trends. *Computer Networks*, 51(12): pp. 3448-3470.

[15]. Song, D., Heywood, M.I., and Zincir-Heywood, A.N. (2005). Training Genetic Programming on half a million patterns. An example from anomaly detection in *IEEE Transactions on Evolutionary Computation*, 9(3): pp. 225-239.

[16]. Thottan, M., and Ji, C. (2003). Anomaly Detection in Internet Protocol Network, *IEEE Transaction on Signal Processing*. 51(8): pp. 2191-2204.

Biography of Authors

Moradeke Grace Adewumi is a Lecturer in the Computer Science Department, College of Education, Ikere-Ekiti, Nigeria. She holds a master degree in Computer Science from Obafemi Awolowo University, Ile-Ife, Osun State, Nigeria. Mrs Adewumi is thorough and organised; she has many publications both local and international to her credit.

Bernard Ijesunor Akhigbe is currently Lecturer I in the Department of Computer Science and Engineering of Obafemi Awolowo University, Ile-Ife, Nigeria. He holds a Ph.D. from the same institution.

Babajide Samuel Afolabi is currently a Senior Lecturer in the Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria. He holds a Ph.D. from Nancy II University, France. He is the Vice-Dean, Students' Affairs in OAU, Ile-Ife.