Anomaly based Network Intrusion Detection System using Neural Network

Palash Chaturvedi[1,] Prof. Amit Saxena[2,] Dr. Anurag Rai

[1] PG Scholar, CSE, TIEIT Bhopal (M.P),
[2] HOD, CSE, TIEIT Bhopal (M.P),
[3] Prof. COER Roorkee.
Corresponding Author: Palash Chaturvedi

ABSTRACT

As a measurement and significance of the system has expands step by step. At that point odds of a system assaults as likewise increments. So to improve organize security distinctive advances has been taken. System is for the most part assaulted by a few interruptions which can be recognized by organize interruption recognition framework. Many sorts of system interruption identification framework which uses the character and mark of the interruption. These interruptions are chiefly contained in information bundles and every parcel needs to check for its location. This paper attempts to build up an interruption location framework in the comparable form of recognizing mark or examples of various sorts of interruptions. As abnormality recognition framework needs to confront distinctive issue of false caution age which implies recognizing as an interruption all things considered it isn't an interruption. Result got subsequent to examining this framework is very sufficient that about 85% of genuine cautions are produced.

Keywords: Computer Networks, Network Security, Anomaly Detection, Intrusion Detection, KDD, Intrusion Detection System, Artificial Neural Network.

I. INTRODUCTION

As the measure of system clients and machine are expanding day by day to offer distinctive sort of administrations and ease for the smoothness of the whole world. Be that as it may, some unapproved clients or exercises from various sorts of assailants which may inward aggressors or outside aggressors keeping in mind the end goal to hurt the running framework, which are known as programmers or gatecrashers, appear. The primary intention of such sort of programmer and interlopers is to cut down massive systems and web administrations. Because of increment in enthusiasm of system security of various sorts of assaults, numerous scientists has included their enthusiasm for their field and wide assortment of conventions and in addition Algorithm has been created by them, with a specific end goal to give secure administrations to the end clients. Among various kind of assault interruptions is a sort of assault that build up a business intrigue. Interruption recognition framework is presented for the security from interruption assaults.

Giving system security to various web benefits on the web, diverse system frameworks, correspondences arrange many advances has been taken like encryption, firewall, and virtual private system and so on organize Intrusion recognition framework is a noteworthy advance among those. Interruption discovery field rises up out of most recent couple of years and built up a considerable measure which uses the gathered data from various sort of interruption assaults and on the premise of those diverse business and open source programming items appear to solidify your system to enhance organize security of the distinctive correspondence, benefit giving systems. From the previous talk we can close the fundamental point of the system. The intrusion recognition framework is to identify all imaginable interruptions that perform malicious actions, PC assaults, spreading infections, PC abuse, etc. so that an interruption discovery system investigates various information plots as well as sifting them through the web for that kind of vengeful movement. So the smooth running of general system distinctive server needs to settle all in all system which go about as system interruption location framework that screen every one of the bundles developments and recognize their conduct with the pernicious exercises. An extra sort of system Intrusion location framework is produced that can be introduced in a brought together server which additionally work in the comparable form of examining and observing distinctive bundle information units for his or her system interruption conduct. System Intrusion identification framework can be produced by two distinctive methodologies which can be named as signature based and irregularity based. In the event of mark based Network Intrusion recognition framework it builds up an accumulation of security risk signature. So as per the profile of every risk the information stream of various parcels in the system are recognized and the most coordinating profile is doled out to that specific bundles. On the off chance that the profile is pernicious then that information parcel goes under interruption and it needs to expel from the system keeping in mind the end goal to stop his out of line exercises.

II. RELATED WORK

The KDD'99 has been likely the most fiercely utilized informational collection for the assessment of peculiarity discovery techniques is set up by Stolfo et al, in view of the information caught in DARPA'98 IDS assessment program [11]. Agarwal and Joshi [12] proposed a Two phase general to particular structure for taking in a rule based model (PNrule) to learn classifier models on an informational collection that has broadly unique class appropriations in the preparation information. The proposed PN manage assessed on KDD dataset reports high recognition rate. Yeung and Chow [13] proposed a uniqueness identification approach utilizing no parametric thickness estimation predicated on Parzen window estimators with Gaussian bits to develop an interruption discovery framework utilizing typical information. This oddity discovery approach was utilized to recognize assault classifications in the KDD dataset. In 2006, Xin Xu et al. [14] introduced a development for versatile interruption recognition predicated on machine learning.

Lee et al. [15], presented information digging approaches for recognizing interruptions. Information digging approaches for interruption location incorporate affiliation decides that focused on finding pertinent examples of program and client conduct. Affiliation rules [16], are utilized to take in the record designs that portray client conduct. These techniques can adapt to emblematic information and the highlights can be characterized as parcel and association record subtle elements. Be that as it may, mining of highlights is constrained by passage level of the parcel and requires the quantity of records to be extensive and low assorted variety in information; else they have a tendency to produce a lot of guidelines which heightens the many-sided quality of the machine [17]. Information bunching strategies including the k means and the fluffy c means have just been connected broadly for interruption recognition. One of the fundamental downsides of grouping procedure is that it depends on figuring numeric separation including the perceptions and thus the perceptions should certainly be numeric.

Perceptions with emblematic highlights can't be effortlessly valuable for grouping, causing error. Moreover, the grouping techniques consider the highlights autonomously and can't catch the organization between various highlights of a solitary record which additionally corrupts assault discovery exactness. Gullible Bayes classifiers have been helpful for interruption location [18]. In any case, they make stark autonomy presumption including the highlights in a statement causing lower assault identification exactness to identify interruptions once the highlights are corresponded, which will be the situation for interruption recognition.

Choice trees have just been helpful for interruption identification [18]. Your choice trees select the best highlights for each and every choice hub all through the development of the tree fixated on some all around characterized criteria. One specific measure is by utilizing the data pick up proportion that is utilized as a part of C4.5. Choice trees for the most part have exceptionally top speed of operation and high assault DR. The examination ers in talked about the use of ANNs for NID. However, the neural systems could work viably with loud information, they may require enormous sum information for preparing and it's regularly difficult to pick the ideal design for a neural system. Bolster vector machines have just been valuable for recognizing interruptions. Bolster vector machines outline esteemed info highlight vector to a higher decent variety in include space through nonlinear mapping and can give realtime discovery ability, manage extensive assorted variety of information. Sen. [19] composed of a circulated IDS is suggested that comprises of a little gathering of self-ruling and collaborating specialists. The machine is equipped for distinguishing and disengaging traded off hubs in the system consequently presenting.

III. BACKGROUND

A). TYPE OF ATTACK: The easy and common criterion to describe all attacks and intrusions in the computer network in the respective literature is always for the types of attack [1]. In this chapter, we categorize all computer attacks in the following classes:

DENIAL OF SERVICE (DOS) ATTACKS:

Denial of Service (DoS) attacks mainly attempt to “shutdown an entire network, computer system, any process or restrict the services to authorized users” [2]. Mainly two types of Denial of Service (DoS) attacks:

- operating system attacks
- networking attacks

In denial of service attack, operating system attacks targets bugs in specific operating system and then may be fixed with patch by patch, on the other hand networking attacks exploits internal limitation of particular networking protocols and specific infrastructure.

Probing (surveillance, scanning):

Probing (surveillance, scanning) attacks scan the networks to identify valid IP addresses and to get information about them (e.g. what services they offer, operating system used). Often, these records supplies a tacker with the list of potential vulnerabilities that will later be used to execute an attack against selected machines and services. These attacks use known vulnerabilities such as for example buffer overflows [8] and weak security points for breaking into the system and gaining privileged access to hosts. Dependant on the origin of the attack (outside attack vs. inside attack), the compromises could be further split into the next two categories:

R2l (remote to local):

Attacks, where an attacker who has the capability to send packets to a device over a network (but does not need an account on that machine), gains access (either as an individual or while the root) to the machine. Generally in most R2L attacks, the attacker breaks into the computer system via the Internet. Typical samples of R2L attacks include guessing passwords (e.g. guest and dictionary attacks) and gaining access to computers by exploiting software vulnerability (e.g. phf attack, which exploits the vulnerability of the phf program which allows remote users to operate arbitrary commands on the server).

U2r (user to root):

Attacks, where an attacker who has an account on some type of computer system can misuse/elevate her or his privileges by exploiting vulnerability in computer mechanisms, an insect in the os or in an application that is installed on the system. Unlike R2L attacks, where the hacker breaks into the machine from the surface, in U2R compromise, the area user/attacker has already been in the machine and typically becomes a root or a consumer with higher privileges. The most frequent U2R attack is buffer overflow, in that your attacker exploits the programming error and attempts to store more data into a buffer that is situated on an execution stack.

B). KDD' 99 DATASET

The KDD'99 dataset includes a couple of 41 features produced from each connection and a brand which specifies the status of connection records as either normal or specific attack type. The list of these features can be found in [21]. These features had all types of continuous, discrete with significantly varying ranges falling in four categories:

1. Basic Features: Basic features could be produced from packet headers without inspecting the payload.
2. Content Features: Domain knowledge is used to gauge the payload of the initial TCP packets. Including features such as for instance how many failed login attempts.
3. Time based Traffic Features: These features are designed to capture properties that mature over a 2 second temporal window. An example of this kind of feature will be the number of connections to exactly the same host over the 2 second interval.
4. Host based Traffic Features: Start using a historical window estimated over how many connections. Time based and Host based traffic referred to as a Traffic features in KDD'99. Likewise, attacks fall under four main categories: DoS, R2L, U2R, Probe.

Type	Quantity of Samples
Normal	97227
DoS	39145
Probe	4107
R2L	1126
U2R	52

Table 1: KDD dataset was employed here and this sample distributed

C). PRE-PROCESSING

In order to increase the efficiency of the work data set, it really should be a pre-process because the preprocessing of the raw data set is compared to the direct input of the raw data set to the selected classifiers; the raw data set is preprocessed in different ways to overcome different problems such as training overload, classifier confusion, false alarms and detection frequency rates. Separating feature space from each other is quite necessary and arrange in vector. Let's consider single vector of the dataset {0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00,normal,20}

In above vector presence of comma ‘,'and d**iscarding symbolic characters which can be of** three kind s of symbolic features (tcp, ftp_data and SF etc.) in feature space of 41 features. As symbolic values aren't of interest to the research, these three feature vectors are discarded to obtain the feature space. So after the preprocessing the obtain vector is {491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00,normal,20} where all element are require for dataset analysis.

D). FEATURES SELETION

Feature selection is an important element in NID. Since, the large numbers of features which can be monitored considering the large variety of possible values particularly for continuous feature even for a small network. For ID purpose, which will be truly useful and reliable, which are significant features or less significant features and which might be useless? . The questions are relevant as the elimination of insignificant and useless features from audit data will boost the accuracy of detection while speeding up the computation, thus will improve the entire performance of our proposed benefit detecting intrusions. So, the main concentration is on selecting significant features.

Now the vector of obtaining contains two important characteristics to select the characteristics, first it is the pattern of the different class type in numerical form like {491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00, 0.05,0.00} and another is the class name as {normal}. In the same way, different patterns of the same class are collected in the single vector and used to decide the type of attack or the normal network.

E). TRAINING ALGORITHM

In order to efficiently detect anomalies in the network for intrusion detection, the following algorithm is implemented:

Algorithm begins with the following inputs DataSet (Ds) number of vector space (n), number of iteration for neural network (N).

```
Training(Ds, N, n)
Vs←Load_dataset(Ds, n)
// For Creating the feature vector
Pv ←Pre-Process (Vs)
Loop I = 1: Pv
Loop J = 1:Ci
If Isequal( Pv(I), Ci(J))
Fv{j} ← Pv(I)
End If
End Loop
End Loop

Tn←Feedforward_neural_network(Fv, N)
```

In above algorithm

Vs: Raw feature Vector

Pv: Pre-Processed Vector

Fv: Feature Vector

Ci :Class index Vector for different attack class

Tn: Trained Neural Network

For training, the appropriate data set function of the neural network is required since the different class has a different set of patterns containing 36 different values. On the basis of this, the neurons of the network will adjust their weight. Fv the feature vector is grouped during the characteristics collection steps of the different class types that match, in the network. Finally, Tn (trained neuronal network) is obtained.

Testing Algorithm

For testing following are the parameter to be pass: Dataset size Ds, number of vector to be use for testing (n) and Trained neural network Tn.

```
Testing(Ds, Tn, n)
Vs←Load_dataset(Ds)
Pv ←Pre-Process (Vs)
Loop I = 1: Pv
Fv(I) ← Pv(I)  // Collect numeric feature
End Loop
Rc←Tn(Fv)  // Pass feature in Trained network
```

```
Loop I = 1: Pv
If Isequal( Pv(I), Rc(I))
TP = TP + 1;
Otherwise
TN = TN + 1
End If
End Loop
```

In above Testing Algorithm
Rc : Resulting Class
TP : True Positive
TN : True Negative

As for the test, the data set of the trained network is again required with different vector, of different or it can be from the same class pattern. Here it is also necessary to make the vector of characteristics of the whole vector to test from the neural network, but only the numerical characteristic is collected in the Fv and then, according to the training, the values of the network are obtained that the input vector belongs what class. Such as {491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00}feature is give as input which will specify the corresponding class. At the conclusion to be able to evaluate the results it is necessary to check on that the specified class is correct or not too each Rc resulting class is match up against the attach class of the numeric feature like normal.

III. EXPERIMENT AND RESULT

In order to implement above algorithm for intrusion detection system MATLAB is use, where dataset is use of different size. It was found that as the data size increase numbers of different class also increase as during 1000 to 5000 only two classes were found in dataset 'normal' 'u2r'.

While increasing the size will increase the different class, as by working on 25,000 data size we found following attack classes 'normal' 'dos' 'probe' 'r2l' 'u2r'.

To test our results, use following measures the accuracy of the write mining approach, that's to state Precision, Recall and F-score.

Precision = true positives / (true positives+ false positives)
Recall = true positives / (true positives +false negatives)
F-score = 2 * Precision * Recall / (Precision + Recall)

DataSet Size	Precision	Recall	F-score
10,000	0.8870	0.7889	0.7736
15,000	0.9672	0.7545	0.7563
20,000	0.8528	0.8678	0.8083
25,000	0.9387	0.8041	0.8437

Table 2: Different dataset and corresponding values

Evaluation of Algorithm for different Data Size from above table (b) it has observed that F-Score values continuously increase as the data Size for training is increases. It has seen that at smaller data size for training some time results of F-score was above 0.9 but that was not true for all as it not cover all type if intrusion attacks. So testing with small size may produce unexpected result.

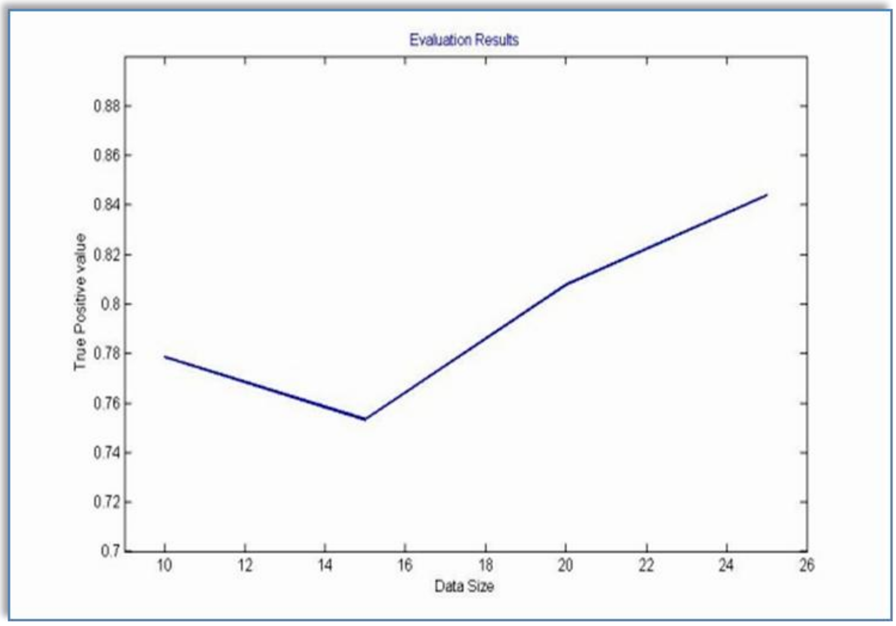

Fig 1: Data size (in thousand scales) Vs True positive values

From above table (b) and graph fig(a) it has found that as the training data size increase the true positive values is also increase so after 15000 training session a continuous growing graph is obtain which tends towards one. As shown in figure 0.844 true positive values are obtain against 25000. So overall detection is good enough as it cover almost each class of different attack.

IV. CONCLUSION

In this paper, IDS tool is develop for effectively identify the different intrusion of any class. Here a neural network is trained by learning the behavior of the different intrusion feature vector, it is obtained after testing that this system can efficiently detect attacks with 85 percent accuracy. One more valuable information is obtain from the system is that network works better for training vector of more than 25000 vector space. In the future, this work only uses the KDD'99 dataset, while there are also other data sets to learn the function and detect different intrusions.

REFERENCES

[1]. K. Kendall, A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, Massachusetts Institute of Technology Master's Thesis, 1998.

[2]. D. Marchette, Computer Intrusion Detection and Network Monitoring, A Statistical Viewpoint. New York, Springer,2001.

[3]. J. Mirkovic, G. Prier and P. Reiher, Attacking DDoS at the Source, 10th IEEE International Conference on Network Protocols, November 2002

[4]. C.Cheng, H.T. Kung and K. Tan, Use of Spectral Analysis in Defense Against DoS Attacks, In Proceedings of the IEEE GLOBECOM , Taipei, Taiwan, 2002

[5]. H. Burch and B. Cheswick, Tracing Anonymous Packets to Their Approximate Source, In Proceedings of the USENIX Large Installation Systems Administration Conference, New Orleans, LA, 319-327, December 2000.

[6]. A.D. Keromytis, V. Misra and D. Rubenstein, SoS: Secure Overlay Services, In Proceedings of the ACM SIGCOMMConference, Pittsburgh, PA, 61-72, August 2002

[7]. S.Robertson, E. Siegel, M. Miller and S. Stolfo, Surveillance Detection in High Bandwidth Environments, In Proceedings of the 3rd DARPA Information Survivability Conference and Exposition (DISCEX 2003) , Washington DC, April 2003.

[8]. CERT® Advisory CA-2003-25 Buffer Overflow in Sendmail, http://www.cert.org/ advisories/CA-2003-25.html, September, 2003.

[9]. C. Cowan, C. Pu, D. Maier, H. Hinton, J. Walpole, P. Bakke, S. Beattie, A. Grier and P. Zhang, StackGuard: Automatic Adaptive Detection and Prevention of Buffer Overflow Attacks, In Proceedings of the 7th USENIX Security Symposium,San Antonio, TX, 63-77

[10]. CERT® Advisory CA-2000-14 Microsoft Outlook and Outlook Express Cache Bypass Vulnerability, http://www.cert.org/advisories/CA-2000-14.html, July 2000

[11]. Leonid Portnoy ,Eleazar Eskin and Stolfo, “Intrusion Detection with Unlabeled Data Using Clustering” Department of Computer Science, Columbia University, Newyork, NY 10027

[12]. R. Agarwal, and M. V. Joshi, “PNrule: A New Framework for Learning Classifier Models in Data Mining”, Technical Report TR 00-015, Department of Computer Science, University of Minnesota, 2000.

[13]. Dit-Yan Yeung, Calvin Chow, "Parzen-Window Network Intrusion Detectors," icpr, vol. 4, pp.40385, 16th International Conference on Pattern Recognition (ICPR'02) - Volume 4, 2002

[14]. Xin Xu, Adaptive Intrusion Detection Based on Machine Learning: Feature Extraction, Classifier Construction and Sequential Pattern Prediction, Institute of Automation, College of Mechantronics Engineering and Automation, National University of Defence Technology, Changsha, 410073, P.R.China, International Journal of Web Services Practices, Vol.2, No.1-2 (2006), pp. 49-58

[15]. Lee W., Stolfo S., and Mok K., “A Data Mining Framework for Building Intrusion Detection Model,” in Proceedings of IEEE Symposium on Security and Privacy , Oakland, pp. 120132, 1999.

[16]. Agrawal R., Imielinski T., and Swami A., “Mining Association Rules between Sets of Items in Large Databases,” in Proceedings of the International Conference on Management of Data , USA, vol. 22, pp. 207216, 1993.

[17]. Abraham T., “IDDM: Intrusion Detection using Data Mining Techniques,” available at: http://www.dsto.defence.gov.au/publications/234 5/DSTOGD0286.pdf, last visited 2008.

[18]. Amor N., Benferhat S., and Elouedi Z., “Naive Bayes vs Decision Trees in Intrusion Detection Systems,” in Proceedings of the ACM Symposium on Applied Computing , USA, pp. 420424, 2004.

[19]. Sen J., “An AgentBased Intrusion Detection System for Local Area Networks,” International Journal of Communication Networks and Information Security , vol. 2, no. 2, pp. 128140, 2010.

[20]. Chimphlee W., Abdulla A., Sap M., Chimphlee S., and Srinoy S., “A RoughFuzzy Hybrid Algorithm for Computer Intrusion Detection,” The International Arab Journal of Information Technology , vol. 4, no. 3, pp. 247254, 2007.

[21]. KDDCUP 1999 Data, available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcu p99.html, last visited 2013.

Palash Chaturvedi1. “Anomaly based Network Intrusion Detection System using Neural Network.” International Journal of Computational Engineering Research (IJCER), vol. 08, no. 01, 2018, pp. 58–64.