International Journal of Modern Trends in Engineering and Research
www.ijmter.com
e-ISSN No.:2349-9745, Date: 2-4 July, 2015

A DDoS Detection and Mitigation System For Cloud Computing

Vaibhav G. Kasar [1],Vijay S. Patil[2], Sagar A. More [3]
[1]*Department Electronics and Telecommunication, RCPIT, vaibhavkasar.kasar581@gmail.com*
[2]*Department Electronics and Telecommunication, RCPIT College, vijayshri12@gmail.com*
[3]*Department Electronics and Telecommunication,RCPIT College, sagar.more27@gmail.com*

Abstract- A fundamental inclination in the computer science is towards Cloud Computing and we can see that many cloud services are projected and developed in the Internet services which are built on MPLS (Multi-Protocol Labale Switching). Cloud services can help many organization to build their data centers with high recital computing resources and decrease the cost of maintaining the computing hardware and other devices which are support to cloud computing technology A data center which are included with various types of communication devices i.e. Router, Switches, Communication medium likes wire , connector and boosting devices. It has provides various services like data, Audio and Video Communication Services. The data center provides the internet services may suffer from many security risks including Distributed Denial of Service (DDOS) attack. A large amount cloud services such as Google and Storage devices document are based on HTTP (Hyper Text Transfer Protocol It's Port No is 80) connection . In our system we planned at HTTP-based connection and A system which consist of Client, Server and Admin. In our system we are checking source of attacks, counting that attacks, detecting attack, Turing test, and Question generation modules. We are providing a multistage detection to extra correctly detect the feasible muggers and a text-based Turing test with question generation module to trial the suspected supplicants who are detected by the discovery module .We implemented the system and evaluated the presentation to show that our system works efficiently to mitigate the DDoS traffic from the Internet. Cutting-edge our system when patron attacks on server scheme our system detects that attack and blocks that client and that pattern of attack is stored at admin side. If another client attacks with same pattern then that client is detected and blocked. Admin performs Turing test for client by generating questions.

Keywords-DDoS, Multi-Stage Detection, Turing Testing, CAPTCHA, Cloud Computing, Text-based Question, Fuzzy Clustering , Attack Detection Module.

I. INTRODUCTION

Cloud services provide a very convenient and efficient way for people to deal with their tasks ubiquitously in the Internet. We can use just a browser with Internet to do many works by using cloud services. Most of the cloud services, like Gmail, Drop box, Google Document, Facebook, etc., are all based on HTTP connections such that a user can access these services with a browser via a cell phone, tablet, or laptop which can be carried everywhere. We are relying on the cloud computing insensibly when we store and process most of our data in the cloud .However, more dependence on our works to the cloud service, more reliability the cloud services must be concerned. Many denial of service (DoS) and distributed denial of service (DDoS) attacks use flooding attacks to exhaust a server are computing capability or network bandwidth such that the legal users could not access the

services provided from the victim server. This will become a big problem when people are relying on cloud services for their tasks, business, and even life.

Zombie network, Botnet, [1], [2] is usually being used to conduct DDoS attacks. Many researches are proposed to detect and trace the zombie network [3], [4], [5]. We are concerned that before we detect a zombie network and trace out the attack source to stop and clean zombie virus, the cloud services are still unavailable and this may take for a long time. Hence, a DDoS mitigation system is necessary to defense DDoS attacks at once. A distributed denial of service defense system is aimed at detecting and mitigating the possible attacks from the Internet. To reduce the possible damage due to such attacks in the cloud, the DDoS mitigation system finds the pattern of attack and saves that pattern in case any other client attacks with same pattern and then blocks the suspected client. Admin provides the Turing test for finding attackers and generates the questions to clients for authentication. In Turing test admin finds if client is human or a suspicious program. Turing test is used to test a machine's ability to exhibit intelligent behavior where it can determine the incoming request is initiated by a human or a program on a zombie host. CAPTCHA is a well-known technology which is used in many web sites to challenge the users and determine if the users are programs. However, most of the CAPTCHA system is based on image challenge that it needs a quite amount of bandwidth to transmit the CAPTCHA image to the users.

In a DDoS mitigation system, we need a mechanism to challenge the suspected users with less network bandwidth consumption. Thus, the attackers would not attack the DDoS mitigation system directly by exhausting its bandwidth. Due to this concern, the reflection ratio between the size of incoming packet and out-going challenge packet must be as low as possible. We are using KDDCUPSET for loading types of attacks. The client packets go through the comparing of packets with defined packets and if new pattern is detected it is stored in KDDCUPSET for elimination further attacks by different clients. The client who attacked with new pattern is blocked after detecting new pattern. In KDDCUPSET we are storing predefined attacks for out testing. From that KDDCUPSET we are taking patterns for attacks. We can store new patterns in that KDDCUPSET.

II. OBJECTIVE

To provide maximum security to cloud computing by using fuzzy Architecture. we are going to implement module it will blocked the client once the such types of attacks will happen over the cloud computing services and we also going to implement we provide the less time for that and provide high bandwidth once the detection proceed.

III. METHEDOLOGY

In our system we are dividing our system in different modules which are listed below.

1) Checking source.
2) Counting.
3) Attack detection.
  a) FC Module (Fuzzy Clustering Module) **b)** Ann module **c)** Fuzzy Aggregation Module.
4) Turing Test Module.
5) Question Generation Module.

A. Architecture

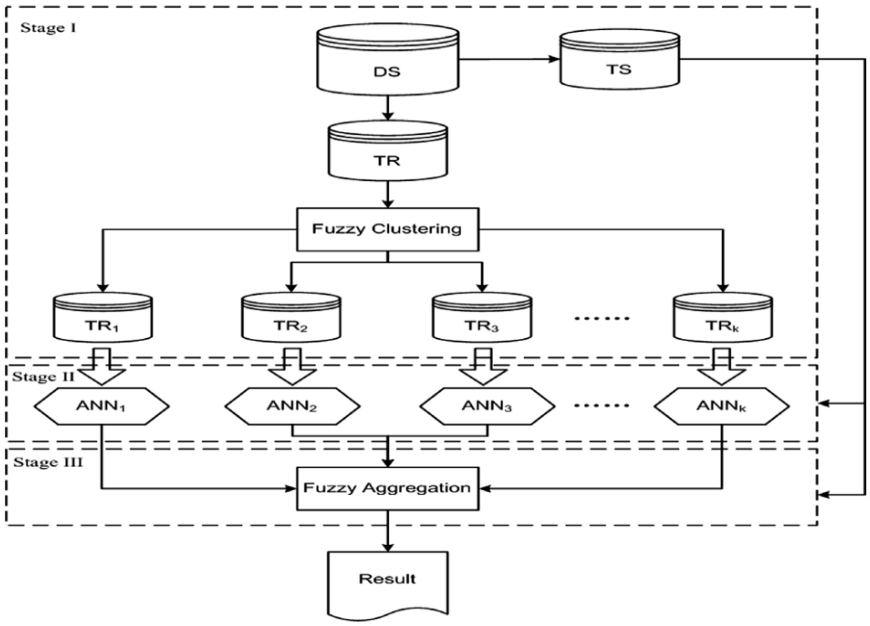

Figure 1. Architecture of FC-ANN and DDoS For Cloud Computing

B. Brief Description of Each Module

Here we are checking the source of attack. We are providing authentication for client for login. If client attacks with some pattern then by identifying that clients IP address we finding its source.

2.1 Checking Source

In this module we are checking the source of attack. We are providing authentication for client for login. If client attacks with some pattern then by identifying that clients IP address we finding its source.

2.2 Counting

In this module we are recording the source address destination address and the time at which client performs login test. After login successful the counting module is reset. It will be enable by the Attack Detection module when there are some suspected traffic been detected.

2.3 Attack Detection

In this section, we sumptuous our new approach; FC-ANN. FC-ANN firstly divides the training data into numerous subsets by using fuzzy clustering method. Afterward, it trains the different ANN by using different subsets. Then it regulates association grades of these subsets and combines them via a new ANN to get final results. The whole framework of FC-ANN is illustrated in Fig 1. As typical machine learning framework, FC-ANN includes both the training phase and testing phase. The training phase includes the following three major stages:

Stage I: For an random data set DS(Data Set) we have used the KDDKUPSET Algorithm , it is firstly divided into training set TR and testing set TS. Then the different training subsets TR1, TR2, TR3,TR4…, TRk are bent from TR with fuzzy clustering module.
Stage II: For each training subset TRi (i = 1, 2,3,4…, k), the ANN model, ANNi, (i = 1, 2,3,4…, k) is training by the specific education algorithm to formulate k different base ANN models.
Stage III: For decrease the error for every ANNi, we simulate the ANNi by using the whole training set TR and get the results. Then we use the membership marks, which were generated by fuzzy clustering module, to combine the results. Then, we train another new ANN using the combined results.
In the testing phase, we directly input the testing set data into the k different ANNi and get outputs. Based on these outputs, the final results can then be achieved by the last fuzzy aggregation module.

The three stages of FC-ANN framework raise three important issues:

a) From the original training dataset TR how to create k different training subsets.
b) From with different training subsets how to create different base model ANNi.
c) how to aggregate the different results produced by different base model ANNi.

These issues will be addressed by the following sections, respectively.

2.3.1 FC Module (Fuzzy Clustering Module)

The goal of fuzzy cluster module is to partition a given set of data into clusters, and it should have the following properties: similarity within the clusters, concerning data in same cluster, and heterogeneity between bunches, where data belonging to different clusters should be as different as possible. Through fuzzy clustering module, the training set is clustered into several subsets. Due to the fact that the size and complexity of every training subset is reduced, the competence and efficiency of subsequent ANN module can be improved.
The clustering techniques can be alienated into hard clustering techniques and soft clustering techniques (Bezdek, 1973). Beside partition of training set, we also need to aggregate the results for fuzzy aggregation module. Therefore, we choose one of the popular soft clustering techniques, fuzzy c-means clustering, for fuzzy clustering module (Chiu, 1994; Yager&Filev, 1994).

2.3.2 Ann Module

ANN module aims to learn the pattern of every subset. ANN is a biologically inspired form of distributed computation (Anderson, 1995; Haykin, 1999). It is composed of simple processing units, and connections between them. In this study, we will employ classic feed-forward neural networks trained `with the back-propagation algorithm to predict intrusion.
A feed-forward neural networks has an input layer, an output layer, with one or more hidden layers in between the input and output layer. The ANN functions as follows: each node i in the input layer has a signal xi as network's input, multiplied by a weight value between the input layer and the hidden layer.

2.3.3 Fuzzy Aggregation Module

The goal of fuzzy aggregation module is to aggregate different ANN's result and reduce the detection errors as every ANNi in ANN module only learns from the subset TRi. Because the errors are nonlinear, in order to achieve the objective, we use anther new ANN to learn the errors.

IV. RESULT

Following are the result of detected by Fuzzy architecture for cloud computing that will block the client once the such types of attacks will happen over the cloud.

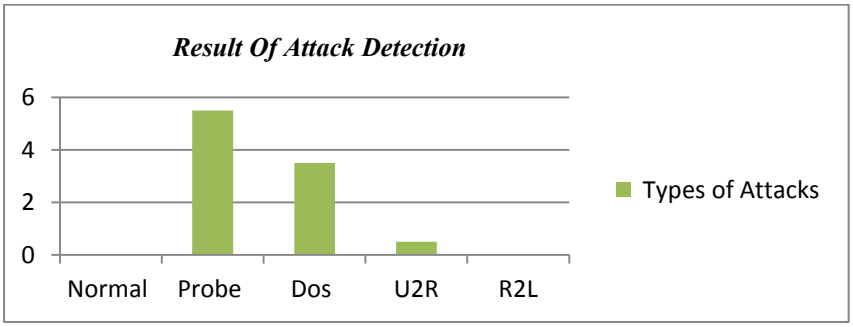

Figure 2. Attack Detected Over Cloud Computing

Table No. 1. Selected Attribute for Individual Attack

Attacks	Selected Attributes In %
Dos	2.6
Probe	5.5
U2R	0.5
RL2	0

Resulting features will be reduced from each of network packets, since may be irrelevant with poor calculation ability to the target patterns, and some of the them may be unneeded due to they are highly inter-correlated with one of more of the other features which decreases not only the detection speed but also detection accuracy possibly in cloud services After reducing KDD features from each record, pre-processing will be done by converting each feature from text or symbolic into numerical form. In this conversion, for each symbol an integer code is assigned. For instance, in the case of protocol type feature, 0 is assigned to TCP, 1 to UDP, and 2 to the ICMP symbol. Attack names were first map to one of the five classes, 0 for Normal, 1 for Probe, 2 for DoS, 3 for U2R, and 4 for R2L Figure 2 Shows that result of attack detection by using intrusion detection system. In our system we are going to implement the client server application over the cloud computing. In that system when detection module will check the coming request from human or Machine. It should be machine (i.e. Attacker) at the time of instant the client will be blocked by server.

V. CONCLUTION

Architecture of the Fuzzy compact rules can be used to implement a network intrusion detection system and DDoS based upon the assumption that an attack can be identified as burst traffic in audit logs that will Happen over the cloud Computing. It will detect the attack and check if the attack pattern exists in system. If the attack pattern not exist it will consider as new pattern this new pattern will store in data base block and block that client from the cloud computing Server . In order to improve the accuracy of such detection system, we have been using 11 features of the TCP/IP Stack header packet. However, with this relatively large number of features we obtain many uninteresting

rules (since the number of possible combinations of the fuzzy variables is quite large) and the running time of the Apriori algorithm increases dramatically.

REFERANCES

[1] Vincent Shi-Ming Huang, Robert Huang, Ming Chiang, "A DDoS Mitigation System with Multi-Stage Detection and Text-Based Turing Testing in Cloud Computing," 2013 27th International Conference on Advanced Information Networking and Applications Workshops.

[2] G. Goth, "Fast-moving zombies: Botnets stay a step ahead of the fixes," IEEE Internet Computing, vol. 11, pp. 7–9, 2007.

[3] P. Salvador, A. Nogueira, U. Franca, and R. Valadas, "Framework for zombie detection using neural networks," in Internet Monitoring and Protection, 2009. ICIMP '09. Fourth International Conference on, may 2009, pp. 14 –20.

[4] F. Giroire, J. Chandrashekar, N. Taft, E. Schooler, and D. Papagiannaki, "Exploiting temporal persistence to detect covert botnet channels," in Recent Advances in Intrusion Detection, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5758, pp. 326–345.

[5] W. Chun-dong, L. Ting, and W. Huai-bin, "Botnet detection based on analysis of mail flow," in Biomedical Engineering and Informatics, 2009. BMEI '09. 2nd International Conference on, oct. 2009, pp. 1 –4.

[6] A. Pinar Saygin, I. Cicekli, and V. Akman, "Turing test: 50 years later," Minds and Machines, vol. 10, pp. 463–518, 2000.

[7] Sebastian Roschke, Feng Cheng, ChristophMeinel," Intrusion Detection in the Cloud", Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009.

[8] Chi-Chun Lo, Chun-Chieh Huang, Joy Ku, "A Cooperative Intrusion Detection System Framework for Cloud Computing Networks", 39th International Conference on Parallel Processing Workshops, 2010.

[9] Andreas Haeberlen," An Efficient Intrusion Detection Model Based on Fast Inductive Learning", Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007.

[10] Richard Chow, Philippe Golle, Markus Jakobsson, "Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control", ACM Computer and Communications Security Workshop, CCSW 09, November 13, 2009.

