Indian Journal of Science and Technology, Vol 8(35), DOI: 10.17485/ijst/2015/v8i35/80167, December 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Hybrid Intrusion Detection Algorithm for Private Cloud

Praveen Kumar Rajendran*, M. Rajesh and R. Abhilash

Cognizant Technology Solutions, Chennai, Tamil Nadu, India; praveenkumar558@gmail.com
M.A.M School of Engineering, Trichy, India; rajesh.manoharan89@gmail.com
Device Associate, Amazon Development Centre, Chennai; mailabhi46@gmail.com

Abstract

Detection and Prevention of Intrusion in the cloud environment is a tedious process. The main objective of this research work is to propose a Hybrid Intrusion Detection algorithm for private cloud environment which is efficient interms of security and performance. A detailed study has been carried out on the existing Intrusion detection systems, which gives a clear view that existing system cannot detect intrusion effectively. Artificial Intelligence has been incorporated in the research work to detect any type of intrusion in private cloud environment. Incorporating Artificial Intelligence technique resulted in self adapative Intrusion Detection System, which has been tested using the real time data collected using the network speed. Implementation of the research work gives a clear view about the efficiency of the algorithm in terms of security and performance, which has not been considered in the earlier research work. Detailed statistical analysis has been carried out as a part of the research work to prove the correctness of the proposed algorithm. The result of the implementation and analaysis gives a clear scope for further research work on Intrusion Detection System for cloud based environment. The proposed algorithm can be implemented for highly secured private cloud which are built for Military purpose and Banking sector to monitor the activities of the network efficiently.

Keywords: Artifical Intelligence, Intrusion Detection System, Performance, Private Cloud,Security

1. Introduction

The Term cloud computing can be defined as the process of deploying and hosting the computational requirements of the end user via the Internet. Hence this type of computing can also be defined as Internet Computing. Hardware Components, Software and Platform are the three major requirements for basic computing. All the three are provided as services of cloud computing are Infrastructure As A Service (IAAS), Software As A Service (SAAS) and Platform As A Service (PAAS)[1]. Since cloud computing offers various other services also, it can be defined anything as a service (XAAS). The three major components of cloud computing are Data Centre, Service Provider, End User. Data Centre is the place which can be located in multiple locations where all the data is being stored. Service Provider deploys the computational resources based upon the requirements of the End User. All the services of Cloud Computing can be easily provided and released with minimal effort and interaction[2].

Cloud Computing services are deployed to the end user by the service provider in 4 different ways, such as Private Cloud, Community Cloud, Public Cloud and Hybrid Cloud. Private Cloud is built and deployed for a small group or for a particular organization. Hybrid Cloud is the combination of any two models. Performance and Security level of Cloud deployment is narrated in the below Table 1. The values of Hybrid Cloud are mentioned as depends on the table because, the performance and security of Hybrid Cloud depends upon the clouds which are joined together. The main contribution and research work focused on this paper is to propose an algorithm to build a high performance intrusion detection system for private cloud.

*Author for correspondence

1.1 Research Motivation

In Table 1, the various cloud deployment models are compared across performance and security aspect, as these two are the major issues of Cloud Computing. Although the values of private cloud are mentioned higher than the other deployment models, in general the performance is less and security is very poor in a cloud environment[3]. Enhancement of performance and security, cloud are the two major goals of the research work in the domain of Cloud Computing.

Although there are many cloud service providers, security aspect will be the significant business differentiator. In general the customers no need to worry about the software up gradation and worry about the various issues, as the service provider will take care[4]. Of all the issues Hacking, Intrusion, Data loss, Data theft, Data destruction is the major security issues in cloud computing[5]. These attacks can be classified as Identifiable attacks and Non-Identifiable attacks. Hacking, Data loss, Data theft and Data destruction can be easily identified by the user and administrator. Intrusion is Non-Identifiable attack, which cannot be easily identified by the user and administrator of the network. Hence it is very essential that an effective Intrusion Detection System has to bet that would detect any type of Intrusion by scanning the incoming request. In this paper Hybrid Intrusion Detection Algorithm has been proposed.

2. Hybrid Intrusion Detection System

2.1 Intrusion Detection and Intrusion Detection System

An Intrusion Detection System can be either a software or hardware which is built based upon the nature of the network. An Intrusion Detection System is one which checks all the incoming and outgoing requests of a network, and detects abnormal or unusual activity in a network. The intrusion can be generally classified into two major categories such as Anomaly intrusion and Misuse Intrusion[6]. Identification of unusual activity in a network is said to be Anomaly intrusion. This type of intrusion can be detected, based upon the behavioral analysis upon the user activities. If any user activity violates the predefined set of rules, it is said to be Misuse Intrusion.

Table 1. Comparison of various deployment models in terms of performance and security

Criteria	Deployment Model			
	Private Cloud	Community Cloud	Public Cloud	Hybrid Cloud
Performance	High	Medium	Poor	Depends
Security	High	Medium	Poor	Depends

In general Intrusion Detection System can be generally classified into two major categories, such as Network Based Intrusion Detection System and Host Based Intrusion Detection System[7]. A Network Based Intrusion Detection System is one which would lay in a common position on a network (Server), and monitor all the activities of the network. Host Based Intrusion Detection System is one which will reside upon all the nodes that are connected to the network and monitor the incoming and outgoing request of that particular host. In some cases, Network Intrusion Detection System is hardware and Host Based Intrusion Detection System is software. Paraget al.[8] has proposed a Network Based Intrusion Detection which would monitor the incoming request to the cloud.

2.2 Hybrid Intrusion Detection System

Both the type of Intrusion Detection technique has got plenty of drawbacks. The major drawback with anomaly intrusion is that it produces a false intrusion alert[9]. Anomaly Intrusion Detection works on the basis of user action history. At many occasions, the authorized user is predicted as an intruder. The major drawback with Misuse Intrusion Detection is, it cannot predict anything other than which has been stored in its history. The history of intrusion has to be continuously updated as the network size and vulnerabilities of the network grow.

Since Network Based Intrusion Detection System lay outside the host and in a common place, it does not give importance for the each and every host in the network. Though Host Based Intrusion Detection System givesimportance to each and every host in the system, it does not get updated itself automatically if there is any change in the network or if there is any new kind of detection has been updated in another host. These are some of the drawbacks of the Intrusion Detection method and Intrusion Detection Systems.

Hybrid Intrusion Detection is an ultimate solution for the above narrated issues. A Hybrid Intrusion Detection is one which lay both in the network as well as in the host, which can detect both the type of Intrusion i. e. Anomaly Intrusion and Misuse Intrusion. A Hybrid Intrusion

Detection System for cloud environment should be Dynamic, Scalable, Self Adaptive and Efficient[10]. In this paper, the earlier works made on Hybrid Intrusion Detection System has been narrated and an effective Hybrid Intrusion Detection Algorithm has been proposed that would meet the requirements for the cloud based environment.

3. Hybrid Intrusion Detection Algorithm

3.1 Related Works

In order to design a Hybrid Intrusion Detection System for Cloud Computing environment, following the earlier works are studied and analyzed. From the best of our knowledge, the following analysis has been written.

The proposal and study made by Patel A, et al.[11] has been taken as the base for creating an Hybrid Intrusion Detection system model[10]. Since the steps for implementation and algorithm has not been given, we have taken the initial step of proposing an algorithm by enhancing the model proposed[4], which is not easy to implement and difficult to understand by the cloud service providers.

Kleber Vieira et al.[12] have proposed a Hybrid Intrusion Detection System for Cloud and Grid environment. This system can detect only kind of attack; hence the efficiency of detection is very less. This system cannot be deployed into a real time distributed environment, as the system cannot synchronize well with the other Intrusion Detection Systems in the network. The authors have proposed the Intrusion Detection System commonly for grid and cloud environment is another major issue that has been addressed, because the architecture and working of the both the model is completely different.

Tupakulaet al.[13] have proposed a Hybrid Intrusion Detection System for Infrastructure as a Service Cloud. This system cannot handle large scale, dynamic, multithread and data processing environment. Since the system has been proposed for Infrastructure as a Service Cloud, the synchronization character is not applicable to the system. Software as a Service and Platform as a Service are the other two services of cloud, which has not been considered by the authors.

Kholidyet al.[14] have proposed a framework for Intrusion Detection in Cloud Systems. This framework does not detect the intrusion in a faster manner; hence the efficiency of detection is very less. This system can partially only handle large scale, dynamic data which is another drawback of the system. The authors have not narrated the scope for implementing the algorithm for the private cloud environment.

Xinwang, et al.[15] has proposed and developed an Intrusion Detection System for cloud with the central management approach. The developed model is not scalable. The efficiency of the system gets reduced when the system is scaled. The implementation of the developed system is quite complicated and difficult to manage.

As per our analysis, in general the above narrated Hybrid Intrusion Detection Systems and Normal Intrusion Detection is not Dynamic, Self adaptive, Efficient and Scalable in nature[4]. The following Hybrid Intrusion Detection Algorithm will overcome the drawbacks of the existing systems and also satisfy the characteristics of Hybrid Intrusion Detection System[10].

3.2 Hybrid Intrusion Detection Algorithm

Hybrid Intrusion Detection Algorithm has three phases, the algorithm for detecting any kind of intrusion in a cloud environment is stated as Praveen-Muthu Hybrid Intrusion Detection Algorithm.

3.2.1 Algorithm 1: HIDS Algorithm

Input: Incoming Request
Output: Display all the details of the request and highlight the abnormal request

1. Start
2. {
3. Initialize the number of users of the cloud;
4. for all the "n" user of the cloud
5. {
6. Initialization and execution of Registration phase();
7. Initialization and execution of Misuse_check_phase();
8. Initialization and execution of Anamoly_check_phase();
9. }
10. End

The Registration phase of Hybrid Intrusion Detection algorithm is written as follows.

3.2.2 Algorithm 2: Registration phase

Input: Username, Password, Physical Address
Output: Registration of authenticated user

1. Start
2. for all the n users of the cloud;

3. {
4. Get valid email-id and password 'eid', 'pid';
5. Auto fetch physical address of the system padd();
6. if the value of email id eid is valid && password pid is valid and
7. if physical address padd is valid then
8. The registration is done Successfully;
9. else
10. {
11. Stop the registration process;
12. Alert the admin of the network for abnormal activity;
13. }
14. }
15. End

The misuse check phase of Hybrid Intrusion Detection algorithm is written as follows.

3.2.3 Algorithm 3: Misuse_Check_Phase

Input: Email-id, password and physical address
Output: Scan, Display the request details and highlight the abnormal request

1. Start
2. {
3. Scan all the incoming requests;
4. Read email id eid, password pid, and physical address padd;
5. Read Protocol pt and port number pb of the incoming request;
6. If the email id and password is not valid,
7. and
8. If the physical address is not valid, then
9. The request may be an intrusion;
10. Highlight the request and alert the admin;
11. If the protocol pt and port number pb is not well known, then
12. The Request may be an intrusion;
13. Highlight the request and alert the admin;
14. If the misuse_check_phase is successful,then
15. Move to next phase, Anomaly_check_phase();
16. Else
17. Alert the admin to monitor the activities of the network;
18. }
19. End

The anomaly check phase of the Hybrid Intrusion Detection algorithm is written as follows

3.2.4 Algorithm 4:Anomaly_Check_Phase

Input: Incoming Request
Output: Display the Speed and Bandwidth of the request

1. Start
2. {
3. Set threshold value of request speed rst
4. Set threshold value of bandwidth bwt;
5. for all the incoming request "i"
6. until the value of "i" is 10
7. Scan the speed of the request rs& display rs;
8. Scan the bandwidth of the request and display bw;
9. if the request speed is less than or equal to the threshold value of request speed,then
10. The request may be an intrusion;
11. Highlight the request in display;
12. else
13. Check the bandwidth of the request;
14. if the bandwidth value bw is greater than the threshold of the bandwidth bwt, then
15. Request may be an intrusion;
16. Highlight the request in display;
17. Find average value of all the request speed rsg;
18. Find the average value of all the bandwidth bwg;
19. if the average request speed rsg is greater than request threshold value rst, then
20. Update threshold value rst with average value rsg;
21. else
22. Move to bandwidth criteria;
23. if the average bandwidth bwg is less than bandwidth threshold bwt, then
24. Update threshold value bwt with average value bwg;
25. Repeat for all the coming requests;
26. }
27. End

3.3 Design

The above proposed Algorithm 1 is designed considering the nature of the cloud, the characteristics of the cloud and the characteristics of an Intrusion detection for the cloud based environment[10]. Figure 1 will give an overall view about the implementation of Hybrid Intrusion Detection in a cloud based environment and the flow of the algorithm. The cloud admin will store all the user information in the data base. After the registration process, if the user login the incoming login will be checked across various criteria which are predefined by the admin. If the request varies from the predefined information and deviates, an

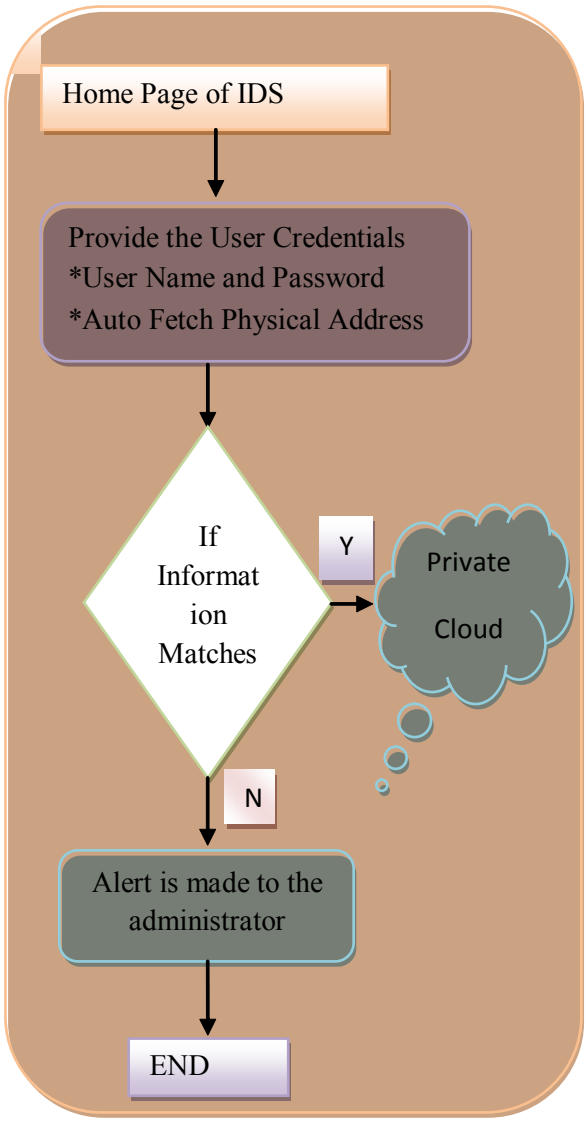

Figure 1. Flow chart of hybrid intrusion detection algorithm.

intrusion alert will be sent to the admin else the admin will grant permission to the user to use the cloud.

3.4 Implementation

The proposed algorithm is implemented using the . Net framework as front end and SQL Server as back end. The algorithm is implemented considering the login phase, as the nature of the cloud could vary. Totally an N number of users are created and details of the user are stored in the database. These users are labeled as registered users or legal users. During the registration process, the application will auto fetch the physical address of the machine through which the registration is done. For security purpose, the legal or the registered user has access the cloud only via that particular machine.

If any request is received apart from the legal user or the registered user physical address, an intrusion alert will be sent to the admin. Intrusion alert will be sent, even if the predefined criteria are not met by the user.

3.5 Performance Evaluation

As described in the design section of this paper the algorithm has been implemented. The performance of the implemented Intrusion Detection System has been checked using Open source Performance testing tool JMeter. The testing of the intrusion detection system has been performed using the following algorithm.

3.5.1 Algorithm 5: Performance Evaluation Algorithm

Input: Test cases of 3 phases

Output: Response time, throughput, amount of data transfer

1. Start
2. {
3. Initialize user thread
4. Specify the number of users, who is going to utilize the system;
5. Initialize the test cases (Samplers) for home page, login page and for admin page;
6. Initialize the output for the test cases(Listeners) for home page, login page, and for admin page;
7. Execute the test cases ;
8. Check whether all the test cases are executed correctly;
9. else
10. Perform the action again;
11. End
12. }
13. }

The above proposed performance evaluation algorithm is implemented and the testing phase executed as the following steps

Step 1: Home Page of the intrusion detection system is executed

Step 2: Login is performed using the login credentials.

Step 3: Admin page is executed, to check whether the system detects the intrusion (If abnormal user get inside the cloud.)

3. 6 Result of Performance Evaluation

Using JMeter, intrusion detection system has been fetched with 5 users, 10 users, 50 users and 100 users. In the general performance of a system is calculated using the criteria such as Average Response time, throughput data transfer. Hence certain parameters such as Median, 90%, Min, etc. produced by JMeter are negotiated. Response time generated by JMeter is in Milliseconds, which are converted into seconds in all following Tables (2–5).

In the Tables (Table 2–Table 5), "Label", indicates the phases of the intrusion detection system. "Index page", is the home page of the intrusion detection system. "Login page", is the page through which both the users and admin can log in. The millisecond values are converted into seconds and given in the table.

Figures 2–5 gives the graphical representation of the Response time. Since the graph generated from the JMeter have readability issues, the values obtained in the tables from 2–5 are interpreted in the form of graph using the statistical software Minitab 17(Trail version). X axis values in the graph (Figure 2–Figure 5) indicate the Sample value and the unit is count. Y axis values indicate the response time value and the unit is seconds (sec). Performance analysis with 5, 10, 50 and 100 users is narrated in the following section of the paper, which is followed by the statistical analysis of the result obtained is being discussed.

Table 2. Performance of IDS for 5 users

Label	Samples (Count)	Response Time(Sec)	Throughput (Sample/Sec)	Data Transfer (KB/Sec)
Index Page	5	0. 781	4. 48833034	7. 617889
Login	5	1. 671	2. 80269058	6. 954723
Admin	5	1. 218	4. 244482	24. 671
Total	15	0. 382	7. 653061	25. 487

Table 3. Performance of IDS for 10 users

Label	Samples (Count)	Response Time (Sec)	Throughput (Sample/Sec)	Data Transfer (KB/Sec)
Index Page	10	0. 452	6. 157635	10. 451
Login	10	1. 341	2. 990431	7. 4205
Admin	10	0. 819	3. 49284	20. 302
Total	30	0. 871	7. 911392	26. 348

Table 4. Performance of IDS for 50 users

Label	Samples (Count)	Response Time (Sec)	Throughput (Sample/Sec)	Data Transfer (KB/Sec)
Index Page	50	25. 58	9. 117433	15. 474
Login	50	80. 80	2. 817536	6. 9915
Admin	50	69. 57	2. 563708	14. 901
Total	150	58. 65	7. 245677	24. 131

Table 5. Performance of IDS for 100 users

Label	Samples (Count)	Response Time (Sec)	Throughput (Sample/Sec)	Data Transfer (KB/Sec)
Index Page	100	60. 54	9. 18273	15. 5855
Login	100	168. 46	2. 81936	6. 99609
Admin	100	139. 05	3. 32380	19. 3196
Total	300	122. 68	7. 16400	23. 8590

3.6.1 Performance for 5 Users

When JMeter is fetched with 5 virtual users against the Index page, Login page and Admin page, the following Table 2 is obtained as a result along with the graph (Figure 2.).

3.6.2 Performance for 10 Users

When JMeter is fetched with 10 virtual users against the Index page, Login page and Admin page, the following Table 3 is obtained as a result along with the graph (Figure 3).

3.6.3 Performance for 50 Users

When JMeter is fetched with 50 virtual users against the Index page, Login page and Admin page, the following Table 4 is obtained as a result along with the graph (Figure 4).

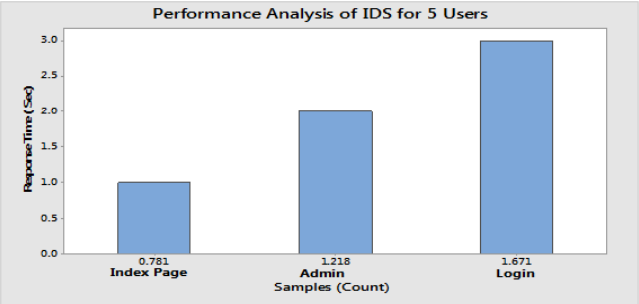

Figure 2. Performance of IDS for 5 user.

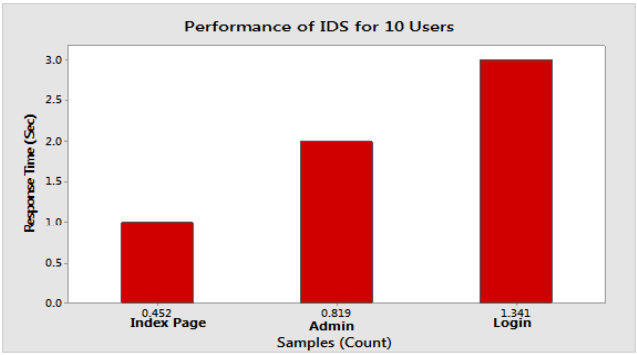

Figure 3. Performance of IDS for 10 user.

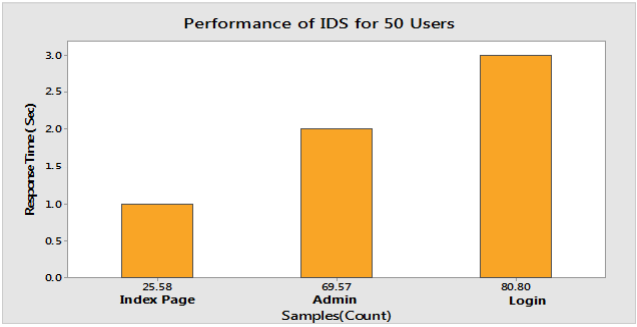

Figure 4. Performance of IDS for 50 users.

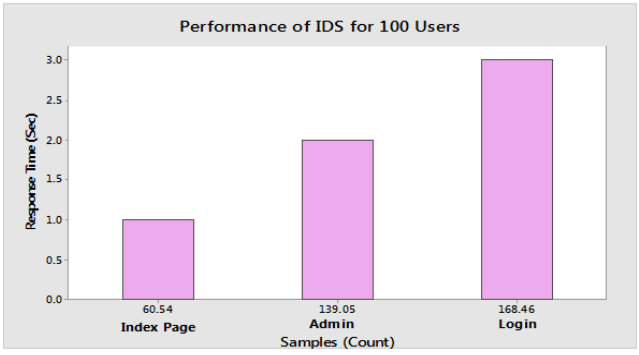

Figure 5. Performance of IDS for 100 users.

3.6.4 Performance for 100 Users

When JMeter is fetched with 100 virtual users against the Index page, Login page and Admin page, the following Table 5 is obtained as a result along with the graph (Figure 5).

3.7 Metrics Calculation and Unit

3.7.1 Response Time

Response time can defined as the time taken for the application to respond to the request[16]. A response time of an application can be calculated using the following Equation

Response time = Time of last request served – Time of first request sent

Rt = T (last) -T (first); Rt = N (Where N is a constant)

The efficiency of the application is good. Response time is calculated in terms of Seconds.

3.7.2 Throughput

Throughput is the number of samples that are executed at a particular instance. Throughput can be calculated using the formula

Throughput = Number of Samples / Response Time of the samples

Let Number of Samples be "N" and Throughput be "T"

T = N/Rt

(If T is constant, the application is efficient). The unit of Throughput can be given as Samples/Second.

3.7.3 Data Transfer

The term data transfer indicates the amount of data that has been transmitted for a single transaction.

Data Transfer = Amount of Data sent+ Amount of Data received

The unit for data transfer is KB/Sec

3.8 Inference - Performance Analysis

Fromthe results obtained, the total average value of Response Time, Throughput and the amount of data transfer is summarized. Summarized values are tabulated in the Table 6, from which certain results can be inferred.

Table 6. Summarized Performance Evaluation

S. No	Total Number of Samples (Count)	Average Response Time (Sec)	Average Throughput (Samples/Sec)	Average Data Transfer (KB/Sec)
1.	15	0. 38	7. 65	25. 48778
2.	30	0. 871	7. 91	26. 34813
3.	150	5. 86	7. 24	24. 13103
4.	300	12. 2	7. 16	23. 85904

The results which are inferred from the Table 6 are as follows.

- The average value of throughput is constant, even the number of users is increased. From this, it can be inferred that algorithm is efficient in terms of time complexity.
- The average value of the amount of data transfer is same, even the number of users is increased. From this, it can be inferred that algorithm is efficient in terms of the amount of data consumed.

3.8.1 Statistical Analysis

Response time is one the key factor of our research; hence statistical analysis on the response time has been carried out. Response time of the application depends upon the number of samples that is being tested. From the graph (Figure 6.), it can be inferred that the variance are equally shared with two values above the mean and two values below the mean value. Among the two values obtained above the mean value, 1 value is very near to the mean value and the other value is little away from the mean value. From the analysis made, we can infer that the 75% of the response time of the application is near to the mean value. Variation in the response time will be minimized in our future enhancement. The following graph (Figure 6) is obtained using Minitab 17 (Trail Version).

3.8.2 One-Way ANOVA

The main goal for conducting One-way ANOVA test is to find the variation among group and between the two groups. One-way Analysis of Variance (ANOVA) for the average response time versus samples is carried out using the values obtained in the Table 6. The output of One-way ANOVA obtained from Minitab 17(trial version) is narrated in the Table 7. For better readability, the values are inserted in the table and formatted.

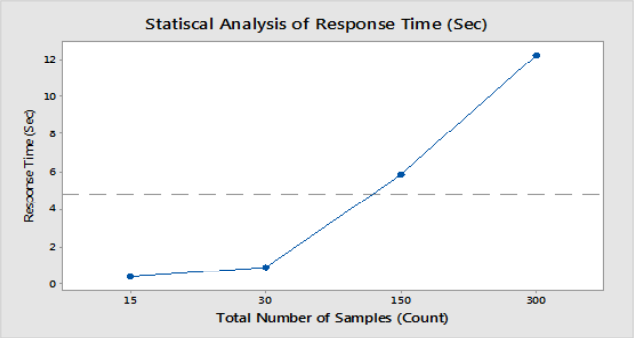

Figure 6. Statistical analysis of response time.

Table 7. Result of one-way ANOVA

Method				
Null hypothesis	All means are equal			
Alternative hypothesis	At least one mean is different			
Significance level	α = 0. 05			
Equal variances were assumed for the analysis.				
Factor Information				
FactorLevelsValues				
Factor 2	Response Time, Samples			
Analysis of Variance				
Source DF Adj SS Adj MSF-ValueP-Value				
Factor 1	28285	28285	3. 24	0. 122
Error 6	52460	8743		
Total 7	80745			
Model Summary				
S	**R-sq**	**R-sq(adj)**	**R-sq(pred)**	
93. 5054	35. 03%	24. 20%	0. 00%	
Means				
Factor	**N**	**Mean**	**StDev**	**95% CI**
C1	4	123. 8	132. 1	(9. 4, 238. 1)
C2	4	4. 83	5. 50 (-109. 57, 119. 23)	
Pooled StDev = 93. 5054				

3.8.3 Inference from One-Way ANOVA

In the Table 7 only the P value, F value, R value and CI value has been considered in our research. In the following table CI indicates Confidence interval, F indicates the factor and R indicates the response due to the factor and P value indicates the hypothesis.

In our analysis, P value has been compared to the significance value (α = 0. 05). Since the P value is greater than the significance value, the mean values are not statistically significant. Hence the null hypothesis can be rejected and concluded that some samples have different mean response time, which is experimentally true (Response time will vary with the number of samples).

Confidence Interval (CI) value of C1 (Response time) obtained is (9. 4, 238. 1) which does not include Zero. From the result it can be inferred that the response time is statistically significant without considering the number of samples, which is experimentally true.

3.9 Discussion

The proposed algorithm scans all incoming request and detect an Intrusion using the Registered Email id, password, physical address and also by scanning the port number, protocol through which the request comes inside the cloud. The speed of the request and the bandwidth of the request are also calculated, to detect the abnormal request which may be an intruder. The proposed algorithm is compared with the earlier work in the table 8 in terms of Dynamic, Scalable, Self Adaptive and Efficiency[10]. Using the proposed algorithm, highly efficient, basic intrusion detection system can be built.

The framework of proposed algorithm can be expanded as per the nature of the network in which the system is deployed. Working of the system will vary based upon the nature of the network. Hence the proof for protocol correctness can be arrived for the proposed algorithm, as it will lead to a false research work in the future.

Statistical Analysis was carried out for response time of the application, as it was the key factor for performance analysis. In our future research work, statistical analysis will be performed on other parameters such as throughput and the data transfer.

Table 8. Comparison of proposed algorithm with earlier work

S. No	Earlier Work	Dynamic	Scalable	Self Adaptive	Efficiency
1	Kleber et al[12]	NO	NO	NO	Partial
2	Tupakula et al[13]	NO	NO	NO	Partial
3	Kholidy et al[14]	Partial	Partial	Partial	NO
4	Our Work	YES	YES	YES	YES

Table 8 will compare the proposed algorithm with the earlier research work which has been carried in the domain of Intrusion detection system.

From Table 8, it is very clear that the proposed algorithm is efficient, since the algorithm can change the nature of its work and thereby satisfies the dynamic nature of Hybrid Intrusion Detection system. The algorithm will hold well, if the number of users is increased. As the algorithm has the capacity to change its nature, it can also meet the scalability and the adaptive characteristics of Hybrid Intrusion Detection system.

4. Future Scope

The proposed algorithm can be implemented in other deployment models of cloud and also in any kind of network. The efficiency of the algorithm can be improved by adding much more criteria to detect an Intrusion in a network. The performance of the algorithm can be improved by updating the values based upon the time limit. The performance of the proposed algorithm will hold well, even if the number of users gets increased.

Including the fuzzy concept in proposed algorithm can improve the efficiency of the algorithm, which can be considered as the future work with reference to Vikrant G. Deshmukh et al[17]. Study on the proposed algorithm along with the proposal made by Nader SohrabiSafa et al[18] on identification of customers using artificial intelligence would give a different dimension for research in Intrusion detection system for cloud based environment.

The anomaly intrusion detection of the proposed algorithm can be further improved by using the outlier approach proposed by Jabezet al.[19], which will enhance the performance of the proposed algorithm and also the efficiency of the algorithm would improve. The proposed algorithm can also be a part of "Fog computing" approach proposed by Salvatore J. Stolfoet al.[20] and also to protect the private cloud which has been located within the public cloud[21].

5. Conclusion

The proposed algorithm can be implemented for a highly secured private cloud such as cloud which is being built for defense purpose, research purpose, and educational

purpose in order to monitor the activities of the network in an efficient manner. Guidance for further research work to enhance the proposed model has been discussed in the Future scope of this paper would take the research activities much forward in this domain.

The Efficiency of the proposed algorithm in terms of time complexity and data consumption holds good. The proposed algorithm can also be implemented using open source technology such as PHP, Perl or Python and can be deployed in open source cloud such as Open Stack, Eucalyptus etc.

6. Refrences

1. Borylo, Piotr, Lason A, Rzasa J, Szymanski A, Jajszczyk A. Fitting green anycast strategies to cloud services in WDM hybrid power networks. IEEE Global Communications Conference (GLOBECOM); 2014. p. 2592–8.
2. Xiao, Zhen, Chen Q, Luo H. Automatic scaling of internet applications for cloud computing services. IEEE Transactions on Computers. 2014;63(5):1111–23.
3. Vecchiola, Christian, Pandey S, Buyya R. High-performance cloud computing: A view of scientific applications. IEEE 10th International Symposium on Pervasive Systems, Algorithms and Networks (ISPAN); 2009. p. 4–16.
4. Kandukuri BR, Paturi RV,Rakshit A. Cloud Security Issues. 2009 IEEE International Conference on Services Computing; Bangalore, India. 2015. p. 517–20.
5. Muthukumar B, Rajendran PK. Intelligent intrusion detection system for private cloud environment. Security in Computing and Communications. Springer International Publishing; 2015. 536: 54–65.
6. Zarrabi A, Zarrabi A. Internet intrusion detection system service in a cloud. IJCSI. 2012;9 (5): 308–15.
7. Roschke S, Cheng F, Meinel C. Intrusion detection in the cloud. Proceedings of 8th IEEE International Conference on Dependable, Autonomic and Secure Computing. 2009. p.730–4.
8. Parag, Sontakke S, Gawande AD. Intrusion detection system for cloud computing. International Journal of Scientific and Technology Research. 2012;1(4): 67–71.
9. Govindarajan M, Chandrasekaran RM. Intrusion detection using neural based hybrid classification methods. Computer Networks. 2011; 55: 1662–71.
10. Rajendran P, Muthukumar B, Nagarajan G. Hybrid intrusion detection system for private cloud: A systematic approach. procedia computer science. 2015; 48: 325–9.
11. Patel A, Taghavi M, Bakhtiyari K, Celestino J,Junior. An intrusion detection and prevention system in cloud computing: A systematic review. Journal of Network and Computer Applications. 2013; 36(1): 25–41.
12. Vieira K, Schulter A, Westphall CB, Westphall CM. Intrusion detection for grid and cloud computing. IEEE Journal: IT Professional.2010; (4): 38–43.
13. Tupakula U, Varadharaja V, Akku N. Intrusion detection techniques for infrastructure as a service cloud. Proceedings of 9th IEEE International Conference on Dependable, Autonomic and Secure Computing. 2011. p. 744–51.
14. Hisham A. Kholidy, Baiardi F. CIDS: A framework for intrusion detection in cloud systems. Proceedings of 9th IEEE International Conference on Information Technology-New Generations. 2012. p. 379–85.
15. Xin W, Ting-Lei H, Xiao-Yu L. Research on the Intrusion detection mechanism based on cloudcomputing. Proceedings of International Conference on Intelligent Computing and Integrated Systems; Guilin. 2010. p. 125–8.
16. Lakshmi M, Sowmya K. Sensitivity analysis for safe grainstorage using big data. Indian Journal of Science and Technology. 2015; 8(S7): 156.
17. Deshmukh VG, Borkut AG, Agam NA. Intrusion detection system for cloud computing. International Journal of Engineering Research and Technology (IJERT). 2013; 2(4): 2149–53
18. Safa NS, Abdul Ghani N, Ismail MA. An artificial neural network classification approach for improving accuracy of customer identification in E-Commerce. Malaysian Journal of Computer Science. 2014; 27(3): 171–85
19. Jabez J, Muthukumar B. Intrusion Detection System (IDS): Anomaly detection using outlier detection approach. Procedia Computer Science. 2015; 48: 338–46.
20. Stolfo, Salvatore J, Ben Salem M, Keromytis AD. Fog computing: Mitigating insider data theft attacks in the cloud. IEEE Symposium onSecurity and Privacy Workshops (SPW); 2012. p. 125–8.
21. Zhang, Hongli, Ye L, Du X, Guizani M. Protecting private cloud located within public cloud. Global Communications Conference (GLOBECOM). 2013. p. 677-81.