International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 20 (2017) pp. 9467-9474
© Research India Publications. http://www.ripublication.com

A Bifold Software Defined Networking based Defence Mechanism for DDOS Attacks in the Cloud Environment

T. V. Sindia[1] and Dr. Julia Punitha Malar Dhas[2]

[1]Research Scholar, Noorul Islam University, Kumaracoil, India.
[2]Department of Computer Science and Engineering, Noorul Islam University, Kumaracoil, India.

[1]Orcid: 0000-0003-0798-7402

Abstract

Cloud computing is a popular network paradigm, in which security is the major concern. Several security threats are lined up in front of cloud computing technology and is necessary to address the security issues. This work focusses on Distributed Denial of Service (DDoS) attack alone as a partial effort to achieve better security. The main objective of DDoS attack is to halt the server from providing service to the cloud nodes, by triggering much more traffic from different cloud nodes. At this juncture, the server cannot withstand the huge traffic and temporarily stopped. Though these attacks cannot be stopped, it can be detected. For this sake, this paper presents a Bifold SDN based Solution using Genetic algorithm and Covariance matrix (BSSGC). The real time traffic data is collected from the Tshark network analyser tool and the abnormal traffic is distinguished by employing a bifold approach. The initial decision about normal and abnormal attacks is made by genetic algorithm and this decision is refined by forming covariance matrix. The experimental results prove the efficacy of the proposed approach with satisfactory accuracy, sensitivity and specificity rates.

Keywords – Cloud computing, DDoS attack, SDN.

INTRODUCTION

Cloud computing is a service oriented platform that offers a wide range of services to the cloud users. The services provided by a cloud computing model can be platform, infrastructure, software and storage. The cloud users can avail these services on demand for a nominal fee. This feature has caught the attraction of several small and mid-scale industries, as the worries regarding licensed versions and maintenance are no more with cloud model. However, the only concern about the cloud model is security. The term security is generic and it may refer to network or data security. Though all the cloud models offer security, there are several security issues to be addressed.

Cloud model is easily susceptible to several security threats. One of the most common security attacks is the Distributed Denial of Service (DDoS). The main goal of the DDoS attack is to down the server by creating unnecessary traffic. Numerous nodes attack a single server such that the server goes down automatically. The DDoS attack can be achieved by attackers or business competitors. Currently, most of the e-commerce sites utilize cloud computing technology to enjoy hassle-free management. In such scenarios, the competitor may intend to halt the server to catch the market and this can be achieved by DDoS attacks.

The DDoS attack targets the server with overwhelming traffic, so as to halt the server [1]. Though it may look simpler, it is very difficult to detect DDoS attack. The DDoS attack can happen at any instant of time; hence the defence system must be vigilant at all times. As the cloud model relies on Internet, the DDoS attack may be the result of the circulation of worms. These worms replicate themselves without human intervention and can easily halt the server. Hence, DDoS is the most dangerous attack that can shatter the network by stopping the service temporarily. Once the DDoS attack is succeeded, then it takes more time to restore the server back. Instead of recovering the server from attack, it is better to detect DDoS attack before its execution.

Though there are several works that deal with DDoS attack, Software Defined Networking (SDN) based solutions to defend against DDoS attack in cloud computing environment is still scarce. SDN reaps so many useful functionalities by segregating the control and the data plane. The logic is embedded to the switch, which makes the entire process of attack detection easier. The message transmission is much easier, as the transmission is carried out by taking the predefined rules being embedded. As the software controls the entire data transmission, it can easily sniff the abnormal scenarios and take necessary action immediately. This immediate action saves the server from being attacked. Additionally, the SDN based solutions require minimal cost and maintenance.

Understanding the necessity of DDoS attack detection scheme in cloud computing environment, this article presents a Bifold SDN based Solution using Genetic algorithm and Covariance matrix (BSSGC). To distinguish between the normal and abnormal traffic, the SDN based defence system is trained initially. Roughly, the entire system is categorized into training and testing. The training phase imparts knowledge to the defence system, whereas the testing phase classifies between the normal and attack scenarios based on the knowledge obtained from the training phase.

In order to achieve the goal, the proposed approach is subdivided into several modules and each module is responsible to achieve partial outcome. The proposed defence system is subdivided into data collection, data normalization, blacklist generation and decision phase. The data for processing is collected from the Tshark analyser [2] to acquire real time traffic data. As soon as the data collection is over, the data are normalized to study the nature of data. The blacklist generation module aims to detect suspicious traffic by employing genetic algorithm and the final decision about whether or not DDoS attack happens is done by covariance matrix. Some of the noteworthy points of this work are listed below.

- BSSGC employs bifold DDoS attack detection mechanism, which ensures accurate classification between the normal and the abnormal scenarios.
- Utilization of real time traffic empowers the proposed approach, as it notifies that BSSGC is suitable for real time environment.
- BSSGC ensures better classification rates with minimal overhead.

The remaining section of this paper is organized as follows. Section 2 presents the related review of literature with respect to SDN and DDoS attacks. The proposed BSSGC is elaborated along with the overall work flow is presented in section 3. The performance of the proposed work is analysed and evaluated in section 4. The concluding points along with the future direction of the article are presented in section 5.

REVIEW OF LITERATURE

This section presents the state-of-the-art related literature with respect to SDN and DDoS attack in cloud computing environment.

A taxonomy based approach is presented to enforce security in SDN in [3]. This work implements fine grained security policies formed by first-order logic description of the networking environment. Though the article claimed that the performance overhead of this approach is tolerable, it is not up to the mark. In [4, 5], the recent trends and architecture of SDN are presented in the aspect of security. Similarly, the positive and negative sides of SDN are discussed in [6]. Additionally, the security threats being faced by SDN are also discussed.

Software defined security architecture is presented for 5G mobile networks in [7]. A centralized security controller is designed, which communicates with the network controller. However, this work considers the mobile networks alone. In [8], a network security education platform which is based on SDN is presented. This education platform considers the dynamic courseware, training cases and service decomposition scheme are provided as well. A service construction strategy for SDN is presented in [9]. This strategy constructs a security service orchestration centre in the SDN's control plane. The designed security centre is separated from the controller to deal with the security threats. This is achieved by designing a security meta-function base, which composes the service in a rule engine. The rule engine is formed by optimized rule composition algorithms.

An experimental software defined security controller is proposed for software defined network in [10]. This work presents a security controller by utilizing Open vSwitch controller. This work deals with the IP and MAC spoofing attacks on the network. This work claims that its precision rates are high. However, the attacks being focussed are IP and MAC spoofing. A policy based security framework that is based on OpenFlow is presented in [11]. This framework employs a network security operator which is able to create and execute security policies in human understandable language. Initially, the OpenSec framework converts all the security policies to OpenFlow messages and the framework acts according to the framed policies. This work blocks the malicious nodes automatically.

A security policy transition framework is presented in [12], which is meant for revoking security measures. The term transition framework justifies itself by the fact that most of the security solutions fire the security measure, when misbehaviour is observed. After this point, the spotted node cannot access the service effortlessly and to achieve this, the controller has to be reset manually. This work eliminates the need of manual reset of controller. In [13], a work to mitigate DDoS attack in Internet of Things (IoT) based on SDN is presented. This work argues that the SDN based approach is better than packet sampling techniques. This work collects the traffic flow statistics from the SDN enabled switch for attack mitigation.

In [14], a scheduling method for network controller is proposed for mitigating DDoS attacks. This work proposes a multi-queue SDN controller scheduling algorithm by utilizing time slice allocation. The time slices are allocated by taking the severity of DDoS attacks into account. A hybrid flow based handler for DDoS attack is presented in [15]. This work employs two classification algorithms to distinguish between the normal and abnormal traffic. Support Vector Machine (SVM) and Self Organizing Map (SOM). This work utilizes multiple linear SVM classifiers for making initial decision and the final decision is made by SOM.

The work proposed in [16] analyses the traffic flow streaming of Vietnamese ISP server during both normal and traffic periods. On analysis, attack prevention architecture is developed, which can observe and examine the traffic flow on the go. This attack prevention architecture is based on thresholds and fuzzy inference system. In [17], an SDN based solution for DDoS attack in cloud computing environment is proposed. This work collects the traffic data from a dataset and distinguishes between the normal and abnormal traffic by means of the entropy variance.

Motivated by the above cited related works, this paper intends to propose a Bifold SDN based Solution using Genetic algorithm and Covariance matrix (BSSGC) for cloud computing environment. Though abundant literature is found with respect to DDoS attack, SDN and cloud computing, all those consider them separately. This work considers all these three dimensions to provide a solution and the related literature is observed to be very minimal. The proposed work utilizes bifold solution for detecting attacks. The initial decision about the happening of attack is made by Genetic algorithm and the final decision is taken by framing covariance matrix. The following section explains the proposed approach in detail.

PROPOSED BSSGC

This section elaborates the working principle of BSSGC along with the overview of the proposed approach.

Overview of BSSGC

BSSGC is a Bifold SDN based Solution using Genetic algorithm and Covariance matrix. As this work involves two different steps to decide the strike of attack, the misbehaviour can easily be sniffed. Moreover, there is always an alarming statement that insists all the security based techniques to have reduced false positive rates. In case of attack detection systems, false positive rates mean to conclude that an attack has approached and an alarm is triggered to the controller, while actually that is not the case. This activity consumes more energy and resources. Most of the attack detection systems tend to provide better attack detection rates and the false positive rates are meagrely taken into account. For instance, a cent percent accurate attack detection system can show maximum FPR. Though the attack detection system detects all the attack whensoever it may happen, false positives must also be considered. The more the false positives, the lesser is the quality of the attack detection system.

Taking this valid point into account, this work presents a DDoS attack detection solution for cloud computing environment. The main goal of this work is set to reduce the False Positive Rate (FPR), as much as possible. When the FPR is minimized, the attack detection system can work without any disturbance caused by wrong alarms. This indirectly conserves energy and let the controller to work peacefully, which in turn provides energy efficiency. Besides this, the unnecessary computational overhead is also overthrown by this work. The overall flow of this work is presented in figure 1.

To achieve minimal FPR, this work employs bifold security solution, in which the initial decision is made by genetic algorithm and the final decision is taken by forming the covariance matrix. In order to test the performance of the proposed approach, the real time traffic data is collected by making use of Tshark analysis tool. On analysis, the proposed approach shows minimal FP rates.

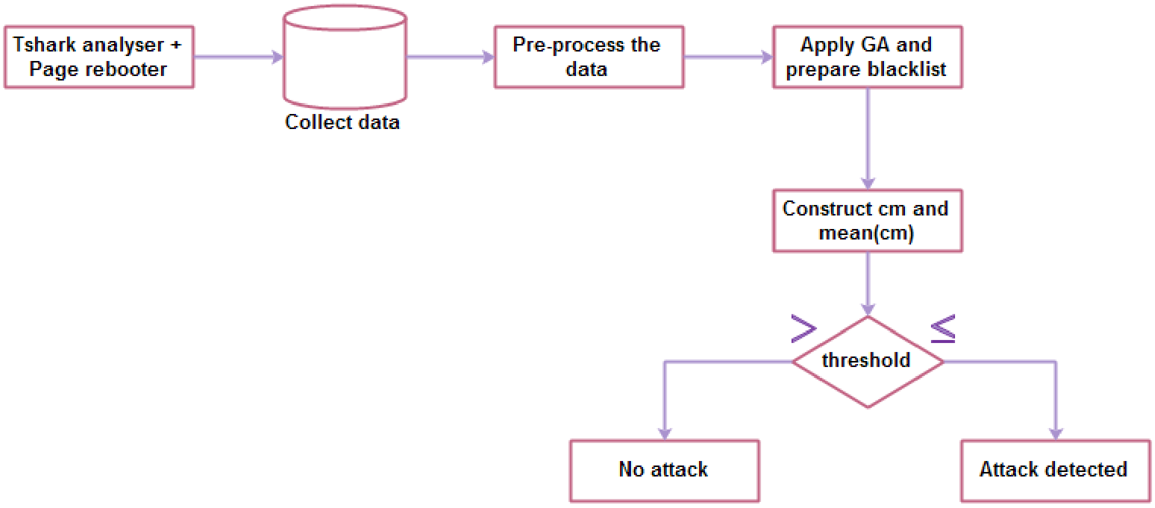
Tshark analyser + Page rebooter
Collect data
Pre-process the data
Apply GA and prepare blacklist
Construct cm and mean(cm)
>
threshold
≤
No attack
Attack detected

Figure 1: Overall flow of the work

Traffic data collection and pre-processing

In order to proceed with, this work collects real time traffic data by exploiting Tshark [2]. Tshark is a network analyser and is capable of grabbing real time traffic. Basically, Tshark is executed on command line, for which the traffic data is generated and displayed in the console. The java code takes the traffic data as input for further processing. The TShark analyser provides several attributes, out of which certain attributes are manually extracted. The attributes of interest are source IP, destination IP, action, packet size and time. These attributes alone are to be processed and are extracted. The extracted data are then pre-processed by arranging the packets with respect to action pair. For instance, the request and response pairs of every source and destination IP are sorted in order. This is to make the data in an organized and intelligible fashion. As soon as this process is done, the blacklist is prepared by the genetic algorithm.

Blacklist Preparation by Genetic Algorithm

Genetic Algorithm (GA) imitates the behaviour of genes and follows evolutionary principle. The GA is based on population initialization, mutation and crossover. Initially, the GA is initialized with a set of chromosomes and the crossover operation can be done, upon the satisfaction of some rules. The performance of the GA relies on the choice of parameters and most significant parameters associated with GA are population size, crossover and mutation rate. The fitness function decides the performance of the genetic algorithm and is given more importance.

Basically, the GA surveys the search space with greater steps and when the optimal solutions are detected the algorithm starts to converge with tinier steps. The GA strictly follows the Darwin’s theory, which is the ‘survival of the fittest’. In this case, the chromosomes which prove maximum fitness value are suitable for the forthcoming stages. This work generates the blacklist of nodes, which are observed to be abnormal based on a fitness function. The fitness function of this work relies on the Time of Existence (ToE). Each service request is associated with an attribute called time. This work fixes the maximum time for a service request, which is 150 seconds. This maximum time is decreased by the value one and the request remains active until its time becomes zero. The value of time of existence, which is the fitness function of this work, is computed as follows.

$$ToE[x] = \frac{(1-r)\times Max_{time}}{|GF-SF|} \times SF + r \times Max_{time} \qquad (1)$$

In the above equation, x is the chromosome, r is the random number ranging from 0 to 1, Max_{time} is fixed as 150. GF and SF are the greatest and smallest fitness values respectively. Initially, both the greatest and smallest fitness values are the same. These fitness values reflect the ToE and it acts as the fitness value of the genetic algorithm. When the ToE is more, it makes sense that it is fitter to be in the next iteration. The more the ToE, the fitter it is to participate in the future processes. In this work, the value of ToE ranges from 22 to 98. The overall algorithm of this work is presented below.

BSSGC Algorithm

```
Input : Traffic data
Output : Service denial
Begin
Collect input data;
Pre-process the data;
Prepare a blacklist by GA;
Construct covariance matrix and refine the decision;
Deny the service for abnormal nodes;
End;
```

Crossover operation is performed, when the number of chromosomes is diminishing. The crossover operation generates a set of chromosomes. For this sake, it would be better to choose the optimal chromosomes and is done by finding the average lifespan of the chromosomes, which is divided by the Max_{time}. The resultant value ranges between 0 and 1, where 0 indicates the chromosomes are not optimal to proceed further. When the resultant value is nearer to 1, then the chromosome is considered as optimal. This process is repeated to generate the blacklisted nodes, based on the rule set. When a source IP is added to the black list, then the service is denied to that node for a particular period of time. In our case, the misbehaving node is denied with the service for about 200 seconds. This time is fixed on the basis of trial and error method. The rules are easily formed as the collected data are arranged with respect to request and response pair. With this operation, the GA produces the blacklist of the suspicious nodes, which seem to be abnormal. The final decision of whether to provide service or not is made by the covariance matrix formation. The covariance matrix formation is provided in the forthcoming section.

Covariance Matrix Formation

Covariance matrix aims to measure the relationship between the blacklisted records. The degree of correlation between the records is measured, so as to distinguish between the normal and the abnormal traffic. Initially, the blacklisted records are gone through and the decisions are made accordingly. A simple scanning and analysis of records can make this process easier, as the normal traffic abruptly differ from the abnormal traffic. Hence, the correlation coefficients are framed between the records. In this work, the covariance matrix is formed by taking the ToE into account.

Let the feature vector is formed by considering the feature value at a particular instant of time. For instance, $ToE(i)$ is the value of ToE at the i^{th} interval of time and is represented as follows.

$$cm = \begin{pmatrix} ToE^{1,i1} & ToE^{1,i2} & \cdots ToE^{1,ii} \\ ToE^{2,i1} & ToE^{2,i2} & \cdots ToE^{2,ii} \\ \vdots\ddots & \vdots\ddots & \cdots\ \vdots\ddots \\ ToE^{n,i1} & ToE^{n,i2} & \cdots\ ToE^{n,ii} \end{pmatrix} \quad (2)$$

Now, the mean value of the cm is computed as follows.

$$\mu_{cm} = \begin{pmatrix} \mu_{ToE^{1,ii}} \\ \mu_{ToE^{2,ii}} \\ \vdots\ddots \\ \mu_{ToE^{n,ii}} \end{pmatrix} \quad (3)$$

As soon as the mean value is computed, the distance between the original cm and μ_{cm} is computed and is represented as

$$dis_{a,b} = ||cm - \mu_{cm}|| \quad (4)$$

This distance measure can find the difference between the normal and the abnormal traffic. Here, a threshold is fixed by the trial and error method, which is 62. When the $dis_{a,b}$ is greater than the threshold, then the traffic is normal and the $dis_{a,b}$ is smaller than the threshold then the traffic is considered as abnormal and the service is denied. By this way, the service denial is applied to all the suspicious nodes. The performance of the proposed approach is analysed in the following section.

PERFORMANCE ANALYSIS

The performance of the proposed approach is analysed by collecting the traffic data by means of Tshark network analyser. To collect the traffic data of a particular site, this work utilizes a webpage, which can be found in [18]. The proposed approach is implemented in Java. The performance of BSSGC is tested against the work proposed in [17] with respect to accuracy, sensitivity and specificity. DDoS attack detection rate is the most important metric, which measures the accurate attack detection. The value of attack detection accuracy must be preferably greater. Sensitivity measures the rate of correct differentiation between the normal and the abnormal traffic. On the other hand, specificity is the rate of accurate classification of abnormal traffic to the total classified results, which includes both the false positives and true negative rates. The formulae for computing the accuracy, sensitivity and specificity are presented as follows.

$$detection_{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (5)$$

$$detection_{sensitivity} = \frac{TP}{TP+FN} \times 100 \quad (6)$$

$$detection_{specificity} = \frac{TN}{FP+TN} \times 100 \quad (7)$$

In the above equations, TP, TN, FP and FN are the True Positive, True Negative, False Positive and False Negative rates respectively. The experimental results of the performance analysis are presented below.

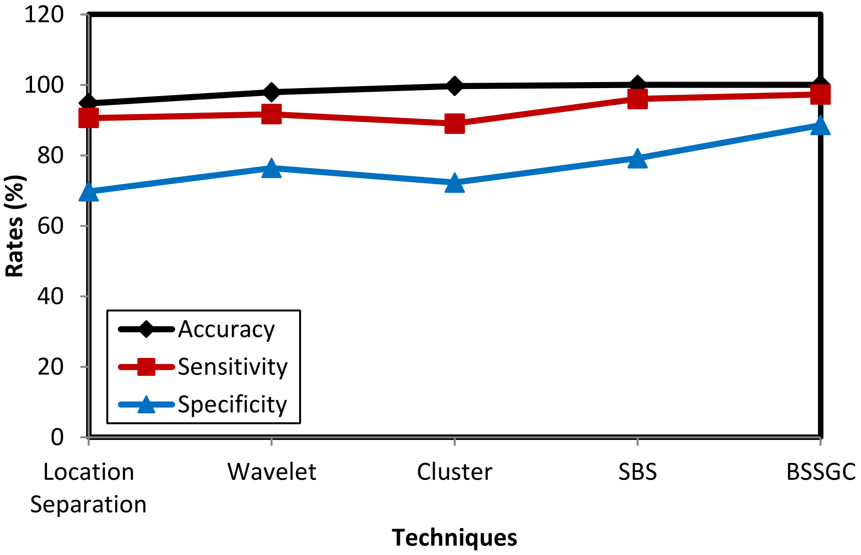

Figure 2: Comparative analysis on accuracy, sensitivity and specificity

Figure 2 shows the experimental results of the proposed approach, which are compared against the existing approaches. The performance of BSSGC is compared against the existing techniques which detect DDoS attacks by means of location separation, wavelets and cluster analysis [19-21]. Though the accuracy rates of all the techniques show satisfactory results, the accuracy rate of BSSGC and SBS is cent percent respectively. Experimentally, it is a bit complex to achieve cent percent accuracy, however achieving better sensitivity and specificity are the greatest challenge. Accuracy rates are measured on the whole, but the sensitivity and specificity rates are specific by nature. This is the reason for why most of the techniques cannot achieve better sensitivity and specificity rates.

All the performance measures such as accuracy, sensitivity and specificity rates depend on TP, TN, FP and FN values, however sensitivity and specificity measures exploit TP, FN and TN, FP respectively. Hence, the sensitivity values are dominated by the FN value and the specificity values are determined by the FP. The accuracy rate considers TP and TN, which can be greater for any work that performs relatively low. The measure which considers the false positive and negative rates, determines the actual performance of the approach. Whenever the FN value is greater, the sensitivity value of the approach is lowered. The greater FP rates affect the specificity value.

Hence, the potential of the proposed approach is measured by accuracy, sensitivity and specificity. The maximum accuracy rate is cent percent and is shown by SBS, BSSGC. The second maximum rate is shown by the clustered approach, which is 99.7. Though the accuracy rates of all the comparative approaches are better, they tend to differ in sensitivity and specificity rates. The greater the sensitivity rate, the lesser is the FN value. Sensitivity rates rely on TP and FN values. Similarly, the specificity rate is indirectly proportional to FP value and this measure considers TN and FP values respectively. Hence, FP and FN values must be minimal, such that the energy and the network resources are preserved.

The greatest sensitivity value is shown by the proposed BSSGC, which is 97.3. The SBS stands next to BSSGC by proving 96.1 percent sensitivity. The least sensitivity rate is shown by cluster based approach. Moreover, the proposed BSSGC proves the highest specificity value too and is 88.6 percent. SBS shows 79.2 as the specificity value and is followed by the wavelet approach with the value of 76.4 percent. The least specificity value is 69.8 percent, which is shown by the location separation technique. Hence, the greatest sensitivity and specificity values indicate that the FP and FN rates are minimal. When it comes to attack detection systems, it is preferably better to have the least FP and FN rates, as increased rates may alarm the network controller frequently. This may result in the increased energy consumption with computational overhead.

The proposed BSSGC shows greater sensitivity and specificity values, owing to its least FP and FN values. The reason behind the least sensitivity and specificity values is that the BSSGC detects the attack in a bifold process, in which the initial decision is refined again to attain minimal false positive and negative rates. The initial decision is made by GA and the blacklist so formed is refined by the formation of covariance matrix. The following figure presents the time consumption analysis.

Figure 3 presents the time consumption analysis of the proposed approach. From the experimental analysis, it is evident that the proposed BSSGC consumes a tolerable time, which is the third minimal out of five techniques. SBS shows the least average time for detecting the attack, which is 0.57 seconds. SBS is followed by the clustered approach that shows 1.2 seconds, as the average attack detection time. The third minimum detection time is proven by the proposed BSSGC and is 1.3 seconds. The reason for the increased attack detection time is the incorporation of bifold attack detection scheme, which refines the initially formed blacklist. However, the main objective of this work is to prove greater sensitivity and specificity rates and is achieved.

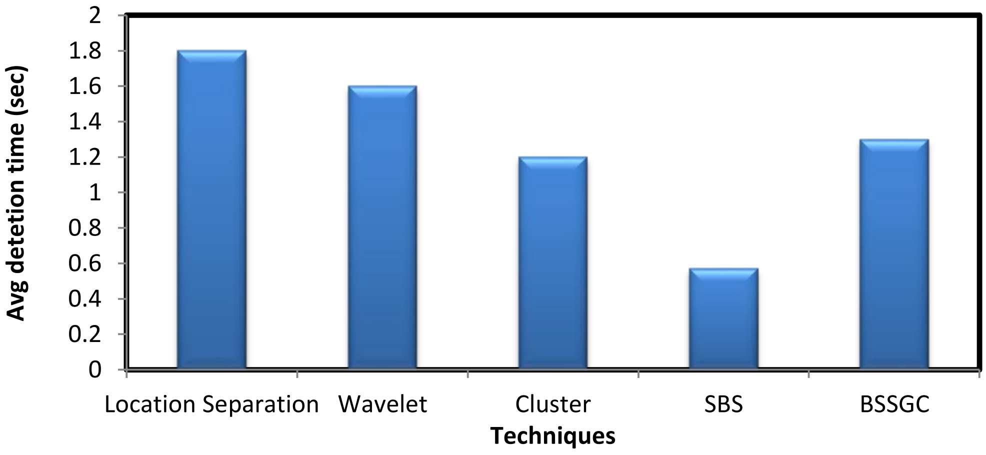

Figure 3: Time consumption analysis

CONCLUSION

This paper presents a solution for DDoS attack detection in cloud computing environment based on SDN. The proposed BSSGC collects the real time traffic data by utilizing the Tshark network analyser tool and a webpage namely ‘page reboot’. Initially, the data is analysed and differentiated as normal and abnormal traffic by GA. The fitness function of GA is computed by Time of Existence (ToE). The greater the ToE, the fitter is the chromosome. When the ToE is greater, then the network traffic is found to be normal. By this way, the blacklist is prepared and is submitted to the next phase in which the covariance matrix is formed. Here, the distance between the records differentiates between the normal and abnormal traffic and the service is denied accordingly. The experimental analysis show greater sensitivity, specificity and accuracy rates. However, the time consumption for attack detection is greater but tolerable. In future, the threshold can be fixed by bio-inspired algorithm and the time consumption can be minimized.

REFERENCES

[1] http://www.cert.org/reports/dsit_workshop-final.html.

[2] https://www.wireshark.org/docs/man-pages/tshark.html

[3] Christian Banse, Julian Schuette, "A taxonomy-based approach for security in software defined networking", IEEE International Conference on Communications, 21-25 May, Paris, 2017.

[4] Mudit Saxena ; Rakesh Kumar, "A recent trends in software defined networking (SDN) security", International Conference on Computing for Sustainable Global Development, 16-18 Mar, New Delhi, 2016.

[5] Danda B. Rawat ; Swetha R. Reddy, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey", IEEE Communications Surveys & Tutorials , vol. 19, no.1, pp.325-346, 2017.

[6] Mehiar Dabbagh; Bechir Hamdaoui; Mohsen Guizani ; Ammar Rayes, "Software defined networking security: pros and cons", IEEE Communications Magazine, Vol.53, No.6, pp.73-79, 2015.

[7] Xiaodong Liang ; Xiaofeng Qiu, "A software defined security architecture for SDN-based 5G network", IEEE International Conference on Network Infrastructure and Digital Content, 23-25 Sep, Beijing, China, 2016.

[8] Jun Wu ; Shen Wang ; Jianhua Li ; Yang Wu, "SDSEP: A Network Security Education Platform Based on Software-Defined Networking Technology", International Conference on Information Technology in Medicine and Education, 23-25 Dec, Fuzhou, China, 2016.

[9] Zhenji Wang ; Dan Tao ; Zhaowen Lin, "Dynamic Virtualization Security Service Construction Strategy for Software Defined Networks", International Conference on Mobile Ad-Hoc and Sensor Networks, 16-18 Dec, Hefei, China, 2016.

[10] Malek Al-Zewairi ; Dima Suleiman ; Sufyan Almajali, "An experimental Software Defined Security controller for Software Defined Network", International Conference on Software Defined Systems, 8-11 May, Valencia, Spain, 2017.

[11] Adrian Lara ; Byrav Ramamurthy, "OpenSec: Policy-Based Security Using Software Defined Networking", IEEE Transactions on Network and Service Management, vol.13, no.1, pp. 30-42, 2016.

[12] Jacob H. Cox ; Russell J. Clark ; Henry L. Owen, "Security policy transition framework for Software Defined networks", IEEE Conference on Network Function Virtualization and Software Defined Networks, 7-10 Nov, CA, USA, 2016.

[13] M. Ejaz Ahmed ; Hyoungshick Kim, "DDoS Attack Mitigation in Internet of Things Using Software Defined Networking", IEEE International Conference on Big Data Computing Service and Applications, 6-9 April, CA, USA, 2017.

[14] Q. Yan ; Q. Gong ; F.R. Yu, "Effective software-defined networking controller scheduling method to mitigate DDoS attacks", Electronics Letters, vol.53, no.7, pp.469-471, 2017.

[15] Trung V. Phan ; Nguyen Khac Bao ; Minho Park, "A Novel Hybrid Flow-Based Handler with DDoS Attacks in Software-Defined Networking", IEEE Conference on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, 18-21 July, France, 2016.

[16] Phan Van Trung ; Truong Thu Huong ; Dang Van Tuyen ; Duong Minh Duc ; Nguyen Huu Thanh ; Alan Marshall, "A multi-criteria-based DDoS-attack prevention solution using software defined networking", International Conference on Advanced Technologies for Communications, 14-16 Oct, Vietnam, 2016.

[17] T. V. Sindia, Dr. Julia Punitha Malar Dhas, "SBS - SDN based Solution for Preventing DDoS Attack in

Cloud Computing Environment", vol.12, no.11, pp.3593-3599, 2017.

[18] http://pagereboot.com/

[19] Luo H, Lin Y, Zhang H, Zukerman M. Preventing DDoS attacks by identifier locator separation. IEEE Network 2013; 27(6): 60–65.

[20] Srihari V, Anitha R. DDoS detection system using wavelet features and semi-supervised learning, Springer Second International Symposium on Security in Computing and Communications (SSCC), Delhi, India, 2014; 291–303.

[21] Bhaya W, Manaa ME. A proactive DDoS attack detection approach using data mining cluster analysis. Journal of Next Generation Information Technology 2014; 5(4): 21–36.