I.J. Wireless and Microwave Technologies, **2018, 3, 1-13**
Published Online May 2018 in MECS(http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2018.03.01

***Available online at** http://www.mecs-press.net/ijwmt*

Investigation of Application Layer DDoS Attacks Using Clustering Techniques

T. Raja Sree[a,*], S. Mary Saira Bhanu[b]

[a] *Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli - 629501, India.*
[b] *Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli - 629501, India.*

Received: 01 March 2017; Accepted: 04 September 2017; Published: 08 May 2018

Abstract

The exponential usage of internet attracts cyber criminals to commit crimes and attacks in the network. The forensic investigator investigates the crimes by determining the series of actions performed by an attacker. Digital forensic investigation can be performed by isolating the hard disk, RAM images, log files etc. It is hard to identify the trace of an attack by collecting the evidences from network since the attacker deletes all possible traces. Therefore, the possible way to identify the attack is from the access log traces located in the server. Clustering plays a vital role in identifying attack patterns from the network traffic. In this paper, the performance of clustering techniques such as k-means, GA k-means and Self Organizing Map (SOM) are compared to identify the source of an application layer DDoS attack. These methods are evaluated using web server log files of an apache server and the results demonstrate that the SOM based method achieves high detection rate than k-means and GA k-means with less false positives.

Index Terms: Self Organizing Map, k-means, Genetic Algorithm k-means, DDoS attack.

1. Introduction

Today, the users are extremely dependent on the internet services to perform their day to day activities. The advancement in internet technologies and increasing reliance on the network become the cause of new threats and malicious activities which compromises the confidentiality, integrity and availability of the network

* Corresponding author.
E-mail address:

services [1]. The attacker uses various browsing activity to commit crime in network and left no evidence is a crucial component for digital forensic investigation. Security is the major concern in internet based applications and also investigation of crimes like security attacks is very difficult. When investigator analyses the victim, the evidence are retrieved from various components such as hard disk, images, log files, cache, cookies, the time and frequency of user visiting the page etc.

The increase in usage of network tools and scripts enable the attacker to conduct various attacks in network. According to the survey report of Kaspersky, the companies lost an revenue of $444,000 by single Distributed Denial of Service (DDoS) attack in 2014 [2]. This leads to high resource consumption and also increases the economic loss by generating heavy bills to the targeted companies. For example, online gaming networks, telecoms etc are vulnerable to DDoS attacks [3]. To investigate such crimes the investigators have to carry out the necessary steps involved in forensic investigations.

DDoS attacks occur in various layers of the network viz., network layer, transport layer and application layer. The network and transport layer attacks send voluminous requests to the victim to saturate the network bandwidth. Transmission Control Protocol (TCP) flood, User Datagram Protocol (UDP) flood, Synchronous (SYN) flood, Internet Control Message Protocol (ICMP) flood etc., are some of the network and transport layer DDoS attacks. The attacker sends malicious HTTP packets to the web server to exhaust all the server resources in case of application layer DDoS attacks [4]. The DDoS attacks occur in application layer are Hypertext Transfer Protocol (HTTP) flood, Simple Network Management Protocol (SNMP) flood, File Transfer Protocol (FTP) flood etc. To protect the network against these application layer threats and malicious activities, several mechanisms are in use. Intrusion Detection System (IDS) is one such mechanism that aims towards stopping the access of the network by unauthorized entities. The various methods used in the existing literature for IDS are Statistical Methods [5], Machine Learning [6]-[7], etc.

Forensic investigation is the process of identification, collection of evidence, examination and analysis of evidence while preserving the integrity of the data [8]. The forensic examiner collects the evidence by finding the series of action performed by an attacker. Forensic examination isolates the attacked system after identifying it, retrieve the data and detects the attack from virtual hard disk, RAM images of VM, log files, etc., through live or dead analysis. Dead forensic analysis identifies the evidence when the data is at rest [9]-[10]. Live forensic analysis identifies the evidence through continuous monitoring of the devices in the network since the data is evolving over time [9]. Evidence collection plays a major role in identifying the attack sources for forensic examination. The evidence is collected and analyzed from the attacked system by using several validating measures and through the log analysis [10].

The forensic investigator relies on finding the details such as where, why, when, who, what and how the attack has happened. Machine learning techniques are used to classify DDoS attacks from the log file traces located in the server. The new attack patterns cannot be determined using supervised learning techniques due to the temporal distortion in network patterns and its characteristics. This is responsible for the ineffectiveness of the supervised learning techniques [11, 12, 13]. The unsupervised learning techniques viz., k-means, SOM, Art2 etc., are more suitable for the identification of new attacks. It is hard to distinguish the legitimate or attack trace since the request patterns and characteristics of attacks are similar as the benign traces [14, 15, 16, 17]. In addition, the new data patterns cannot be identified by using Intrusion Detection System (IDS) because of the tremendous amount of data generated and it suffers from large processing overheads [17, 18, 19]. Clustering plays a major role in the identification of new attacks which have not been encountered previously for forensic analysis. Clustering algorithms are used to group the similar data patterns which help to enhance the performance of the system.

In this paper, performance of the clustering techniques is compared by extracting the features from the web server log. These features are processed by using clustering techniques such as k-means, Genetic Algorithm (GA)-k-means and Self Organizing Map (SOM) for the identification of application layer attacks that had happened. The k-means clustering is used due to its easiness and simplicity of application, which is not suitable to deal with the overlapping clusters. The k-means clustering depends on the initial seed, hence it stuck to local minimum [16,17]. In order to overcome these drawbacks, GA-k-means clustering algorithm is used. In GA-k-

means clustering, the initial seed is selected from the set of random values which helps in determining the optimal clusters. SOM isolates the unknown patterns from the neighbouring neuron. It is also responsible for mapping 'N' dimensional data into one or two dimensions that groups the similar input patterns.

The remainder of the paper is structured as follows: Section 2 outlines the related research work about forensics. Section 3 outlines the overview of the system model. Section 4 elaborates experimental work and comparison results. Section 5 concludes the paper with future work.

2. Related Work

Digital forensics is the process of identification, collection, validating the digital information by preserving the evidence [8]. The forensic examiner analyzes the attack by collecting the evidence from physical memory, virtual hard disk, log files etc., through online or offline.

Application layer attacks play a major role in attacking the web server and their applications. Krugel et al. proposed web based attack detection by automatically retrieving the profiles such as length and structure of web server logs [20]. These profiles are compared with the incoming user requests to classify the attacks. It results in large false positives. Lee et al. introduced a method for the detection of benign or attack traces using cluster analysis on each attack phase [16]. This method selects only few input features which results in low detection of attacks.

Yatgai et al. introduced DDoS attack detection using the browsing order of the page and finding the correlation to the page information size [15]. The usage of large access log file has not been addressed to detect the new attack and result in high false positives. Oh et al. adopted a method for the identification of DDoS attacks by clustering of traffic patterns using SOM and the labelling is performed using the correlation of features [17]. The detection accuracy is reduced by the labelling of each map units. This method results in large number of false positives.

Konar et al. combines the idea considered in [17] with the fuzzy logic to achieve the high detection rate [18]. SOM algorithm has been used to identify the suspicious nature of unseen patterns and modelling the fuzzy rule from every neighbouring map unit. When a new attack occurs, the new rules correspond to the map units will be updated instead of updating the entire model in the fuzzy rule base. Zolotukhin et al. proposed a method for the identification of benign or malicious requests using n-gram analysis and through statistical methods [21]. This method takes more computational time since the size of the feature is large. Bhuyan et al. proposed a method to distinguish the low rate and high rate malicious traffic from benign traffic using information theory with low computational overhead [21].

Maggi et al. adapted a method to distinguish between the benign or malicious behavior in web based applications. The HTTP traffic response is analyzed to determine the historically modelled parameters [23]. This method needs huge volumes of well labelled data for initial training to determine the malicious behavior. Chwalinski et al. proposed a method for the detection of HTTP-GET flood attack by using clustering of categorical data points and information theoretic measures. This method distinguishes the legitimate and attacking sequences by analyzing the behavior of web request sequences [24]. Prior knowledge is not required to detect the attack behavior. It is difficult to find the number of clusters that had spread across the various entropy ranges because many sequence of requests follow uniform distribution.

The methods discussed in the existing literature have not addressed the problem for identifying the unknown attacks located in the server. The existing methods take high processing time, high resource consumption, and result in large false positives. Clustering plays a major role in the identification of unknown attacks. The performance of the clustering algorithms such as k-means, GA k-means and SOM are compared for the effective identification of attacks.

3. System Model

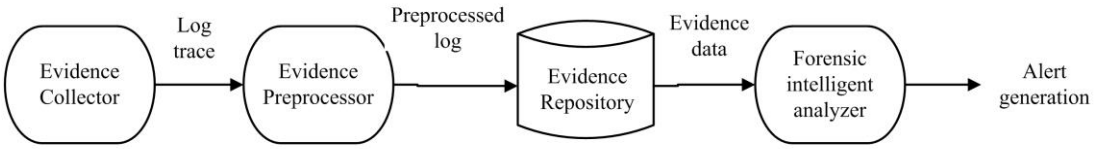

Fig.1. Architecture of HTTP flood attack detection

The architecture of the system model is depicted in Fig. 1. The system model consists of four stages namely Evidence collection, Evidence Preprocessor, Evidence Repository and Evidence Analyzer.

- *Evidence Collection:* This process collects the evidence from information sources such as network routers, switches, server and hosts which is under investigation.
- *Evidence Preprocessor:* It takes the log file as input and analyzes the log file to identify the evidence of an attack in terms of features. It pre-processes the feature set and selects a feature subset to describe the attack.
- *Evidence Repository:* This is the process of storing all the pre-processed relevant information for the identification of evidences.
- *Evidence Analyzer:* The feature subset of evidence is given as input to the evidence analyzer, which compares the newly generated log files from incoming traffic with the predetermined rules from knowledge base to generate forensic alert.

3.1. Evidence Collection

The evidences are collected from the network sources such as router, switches, server, hosts and the internal components viz hard disk, RAM images, physical memory etc. which are under forensic investigation. The logs collected from the network play an important role in evidence collection. Application layer attacks are reflected in various logs viz., system log, network log, authentication log etc., stored on Apache server. These logs are used for forensic examination to detect the application layer attacks. The various attack information stored in the log traces are listed as follows.

- *System log* – determines if someone is trying or has executed buffer overflow
- *Debugging log* – determine the nature of application and service based attacks
- *Firewall log* – direct method for auditing firewall
- *Authentication log* – auditing of attacks on credentials and determines the unauthorized access
- *Dmesg log* – this is not a log file, but this is used for determining anomalous activity from recent bots.
- *Access log* – useful for determining web based attacks (XSS, XSRF, SQLI, remote file inclusion, local file inclusion and DDoS attacks.
- *Error log* – useful for determining web based attacks
- *Database log* – useful for determining the database related attacks

Since DDoS attack is reflected in the access log file, this log is taken for forensic analysis. The entries in the access log file of an web server consists of the following attributes as discussed in [25] are: remote host, remote login name, remote user, request time (day/month/year:hr:min:sec +zone), HTTP request (HTTP method, URL

and HTTP version), HTTP status code, length of the data in bytes, referral URL and user agent header field. The access log trace of an web server log file are as follows.

101.38.64.23 - - [20/Mar/2015:08:33:23 +0700] “GET /d/winnt/system32/cmd.exe?/c+dir HTTP/1.0” 200 458 “https://www.nitt.edu/OLCLD/view.php?q= book/” “-“

These logs are used to determine the attacks by the analysis of the essential features.

3.2 Evidence Preprocessor

The logs obtained from the web server are converted to common format by removing the uncleaned or unwanted attributes. The results consist of only an essential attributes viz., remote host, time of request, HTTP request and referral URL for determining the legitimate or suspicious user [25].

The remote host (IP address) is converted to 010214134056 by removing dot and turns to whole digits. The second attribute request time is transferred to digit by removing the symbols and zone times (ie) [15/Feb/2015:06:56:19 +0700] to 15022015065619. The next attribute is HTTP request (HTTP method, URL and HTTP version). The static method values are considered as HTTP/1.1 - 80, HTTP/1.0 - 70, GET - 10, POST - 30, HEAD - 50. The URL's are converted to its corresponding numbers by using hash functions. The URL part varies arbitrarily by applying hash function to convert to its unique numbers. Similarly, the referral URL is also converted to unique numbers by using hash function. Then, the preprocessed features are passed as input to the clustering module for the identification of anomalous behavior.

3.3. Evidence Repository

The evidence repository stores all the preprocessed data for the identification of attacks.

3.4. Evidence Analyzer

The preprocessed data is passed to the Evidence analyzer for the grouping of the similar input patterns using clustering techniques for the effective identification of attacks. Clustering is the process of combining the similar input patterns into groups of clusters. The grouping of data objects into ‘N’ dimensional features to maximize the similarity of data within clusters and minimize the similarity of the set of data objects between different clusters. The preprocessed data is fed as input to the various clustering algorithms viz., k-means, GA k-means and SOM.

The pre-processed log files consist of relevant features such as IP address, timestamp, requested URL of the page and referral page of the user. These relevant features are converted to numerical values because SOM, GA k-means and k- means resolve only the numerical data to perform clustering on the pre-processed data.

3.4.1. k-means clustering algorithm

The k-means algorithm is widely employed in finding the near optimal partition with the given number of clusters. It uses iterative hill climbing algorithm [25]. The steps in k-means clustering are as follows.

(i) The initial seed of each cluster is selected based on given ’k’ (number of clusters), and the partition is made using seed as the centre of initial clusters.
(ii) The record which is nearer to the centre, groups the similar patterns thus forming the cluster.
(iii) Keeping the fixed cardinality of clusters, determine the centre for each cluster.
(iv) Repeat the steps (ii) and (iii) until the clusters converge or it satisfies the stopping criteria.

The limitation of k-means algorithm is that the clustering depends on the initial seed. If the dataset has large outliers and initial seed is not chosen properly, then it generates large differences in clustering results. The selection of initial seed by random may degrade the performance of clustering quality; hence it converges into local minimum. In order to overcome these limitations, Genetic Algorithm (GA) is used as an optimization technique for the selection of initial seeds in k-means algorithm [15].

3.4.2. GA k-means clustering algorithm

Genetic Algorithm (GA) (Goldberg, 1989) is a method for solving constrained or unconstrained optimization problem that mimics the biological evolution [26]- [27]. In GA, the set of intermediate solutions called candidate solutions are further optimized using Darwin's theory with repetitive computations. The fitness is measured to find the solutions of each individual. The commonly used genetic operations are selection, crossover and mutation. The new population is generated by choosing the individuals on the basis of selection strategy. Crossover is the generation of new individual by the mixing of two off-springs from selected parents [23]. Mutation is a background operator which is used by randomly varying the values of an individual at one or multiple positions of the selected chromosome. The steps in GA-k-means process are shown in Fig. 2.

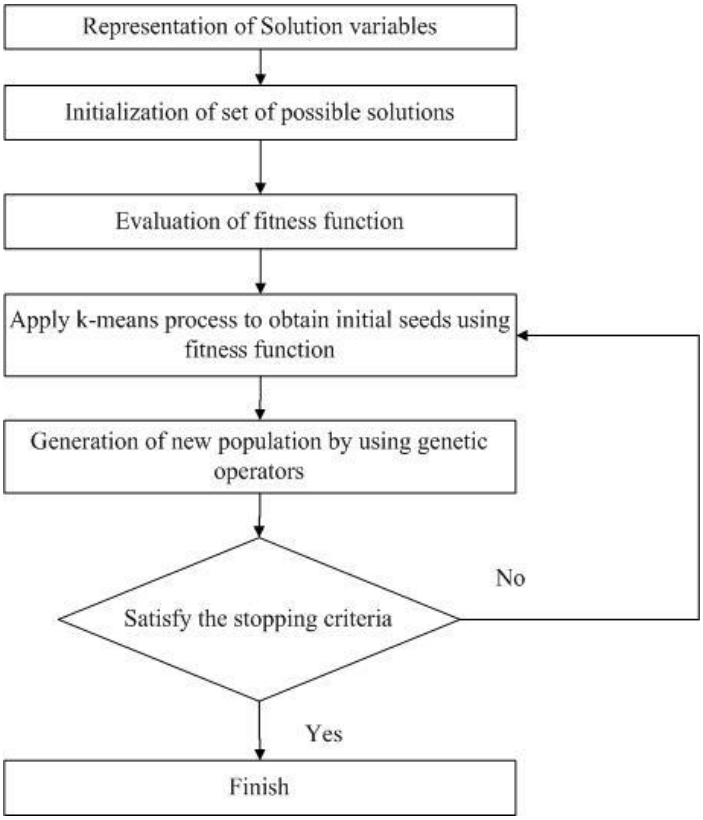

Fig.2.Steps in GA k-means process

GA is used as an optimization technique for the selection of initial seed in k-means algorithm. The global optimal initial seed is selected from the set of the random values which helps to increase the performance of the classifier. The steps in GA k-means algorithm is depicted in Fig. 2 is as follows:

(i) *Representation of solution variables:* This step is used to identify the solution variable from the objective function for denoting the values. Here, the solution variable is represented using the real coded strings. The value of the solution variables should be specified within the range.
(ii) *Population initialization:* The population is generated randomly to determine the global optimal initial seed. The values of the chromosome are initialized with random values for the searching of optimal seed. Real coded strings are used to determine the optimal initial seeds.
(iii) *Fitness function Evaluation*: The next step is the selection of k-means clustering for finding the optimal clusters. The maximum number of adjustment of centroid value in k-means is determined. The k-means process is repeated iteratively and the fitness function value is updated. The intra class inertia is chosen as the fitness function of GA k-means, through which the optimal initial seed is obtained by minimizing the function after the completion of k-means clustering.
(iv) *Genetic operations*: The GA performs various genetic operations viz., selection, crossover and mutation on the current population [26]. A new individual is produced from these operations.

- *Selection:* The selection of new individuals from the population plays a vital role in GA. Tournament selection [25] is used as the selection operation to produce new individual. In tournament selection, the best of 'T' individual is selected from the set of population in the mating pool. This process is repeated for further genetic processing until the mating pool is filled.
- *Crossover:* This uses arithmetic crossover to perform genetic operations. The new strings are produced by the exchange of information among the real coded strings in the mating pool.
- *Mutation:* It introduces new strings into the population and it prevents trapping into local optimum value. This uses uniform mutation operator to perform genetic operations. Here, the selection of variable is by uniform random number and this number is set between the variables lower and upper limit. These steps are repeated until it satisfies the stopping criteria.

3.4.3. Self Organizing Map

The feed forward neural network has proposed by Kohonen (1982) [2, 28, 29]. It consists of neurons with 'n' input patterns that are associated to 'm' output cluster units. These patterns are represented in two dimensional spaces where the input pattern of the weight vector acts as an exemplar. The similar input patterns are grouped together and the comparison of each neuron is made between the input weight vector and the associated pattern. The closest neuron is selected as the winner neuron. The Best Matching Unit (BMU) of neuron is calculated by adjusting the nearest neuron and the weight of winner's neuron.

Steps of SOM algorithm

(i) *Initialization of the network:* For every node *l*, initialize the weight vector w_l to some random value.
(ii) *Assigning of Input:* Assign the input pattern vector X to all the nodes in the network.
(iii) *Estimation of winning node:* Calculate the winning neuron $p(x) = \min\{ d_l(x)\} = \sum_{i=1}^{D} (x_i - w_{li})^2$ ie., minimum distance among the weight vector and the associated input vector. The minimum distant node is declared as the BMU of the node.
(iv) Weight updation: Update the weights of each node by using the following equation $\Delta W_{nm} = \delta(t) T_{n,p(x)}(t)(x_n - w_{mk})$, where $T_{m,n}(t)$ and $\delta(t)$ represents the Gaussian neighborhood and the learning rate respectively.
(v) Repeat the steps (ii) to (iv) until the criteria of minimal distance is met with intact feature map.

4. Experimental results

This section details the experimental evaluation for evidence collection, evidence preprocessor and the experimental results of the clustering algorithms.

4.1. Experimental setup

The normal traffic is obtained by using the different browsing activities carried out on the different machines using valid user agents, HTTP methods and HTTP header parameters, which are reflected in the web server log. The huge volumes of real traffic that are flowing to and from the web server is captured and reflected continuously as log file during one week period. The attack was subsequently executed during this period. The HTTP GET flood attack is launched by using bots or through various attacking tools viz., HULK [30], HTTP DoS [31], HOIC [32]. There is a substantial increase in the flow of traffic during the peak hours and slowly it degrades in the afternoons. The experimental setup is depicted in Fig. 3. The DDoS attacks are reflected in the access log file of an apache server.

4.2. Result Analysis

The transmission traffic collected is pre-processed by formatting the log files. The pre-processed log files are fed as input to the different clustering algorithms for the effective identification of traffic patterns. The initial seed is chosen for the selected number of clusters. The maximum number of adjustment of centroid value in k-means is fixed as 5.

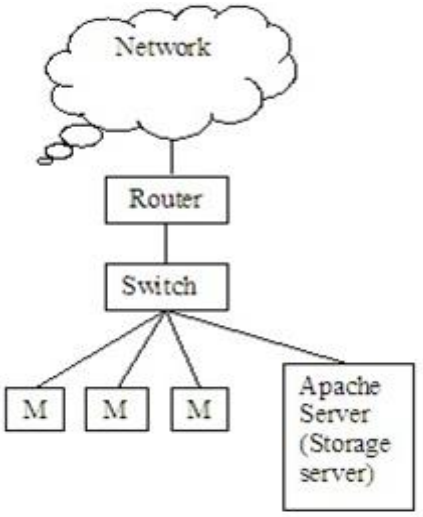

Fig.3. Experimental test bed setup

The GA was run with 100 independent trials with distinct random seed values and the various control parameters of GA. The optimal results are obtained with the settings given as follows: size of population – 200, Crossover rate – 0.9, Mutation rate – 0.01. The arithmetic crossover is used for real-coded strings that generate new individual from the two parents. Uniform mutation operator is performed and the values are selected randomly from an individual for finding the optimal value. If a number obtained is lesser or equal to the

mutation rate, then the mutation is performed at the particular gene. It satisfies the stopping criterion by allowing 100 generations.

SOM is used to preserve the topological property of the input neuron by grouping of nearby neurons as matching unit. The results obtained from these clustering techniques are used to distinguish the suspicious or the normal behaviour of the user. The dataset generated using various attacking tools are considered for the different number of test cases as depicted in Table 1.

Table 1. Dataset considered for various tests

Test Cases	Dataset Considered
1	HOIC
2	HTTP DDoS
3	HULK
4	HULK, HTTP DDoS
5	HULK, HOIC, HTTP DDoS

Various tests are carried out using the log files generated by different tools such as HULK, HTTP DDoS and HOIC. The combination of various attack instances are tested for different scenarios. From the experimental results, it is observed that the false positive rate is higher in k-means because the initial seed is taken randomly and it stuck into local minimum. The GA k-means misclassification rate is lesser when compared to k-means and the optimum value is identified for initial seed. The false positive rate is relatively less in SOM, because SOM maintains the topological preserving property. The false positive rate of k-means, GA k-means and SOM are shown in Fig. 4.

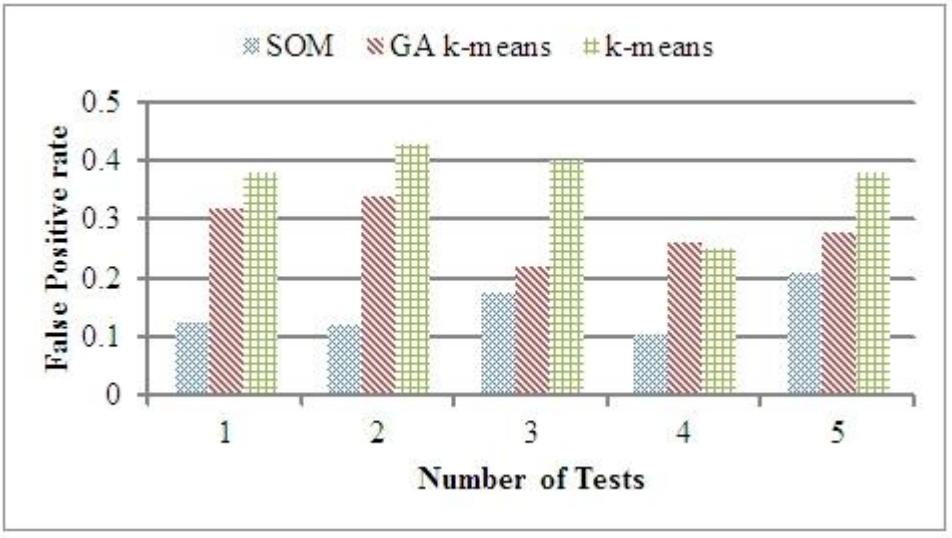

Fig.4. False positive rate for various tests

The accuracy for the detection of attacks has been measured using equation (5)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where,

True Positive (TP) = Number of attack instances correctly classified as attack
False Positive (FP) = Number of normal instances incorrectly classified as attack
True Negative (TN) = Number of attack instances incorrectly classified as normal
True Negative (TN) = Number of normal instances correctly classified as normal

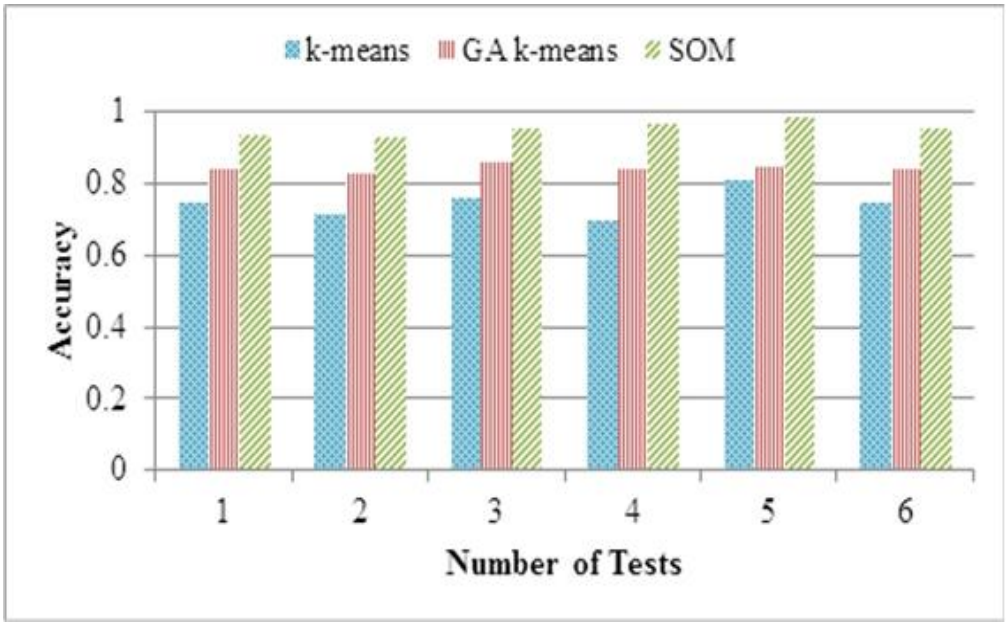

Fig.5. Accuracy of clustering algorithms for different tests

It is inferred from Fig.5, that the accuracy of SOM is higher when compared with k-means and GA k-means for the different number of test cases. The k-means based on GA outperforms well than k-means algorithm because initial seed is selected randomly in k-means and also stucks into local minimum value. However, GA k-means selects the initial seed data based on the iteration and it helps to determine the global optimal value. Hence, GA k-means detects the malicious user efficiently than k-means. The accuracy of the different clustering algorithms for various tests is depicted in Fig. 5.

The performance of the clustering algorithm is compared using intraclass inertia. Intraclass inertia is represented in n-dimensional space of the preprocessed features to check the compactness of each cluster. The intraclass inertia of SOM, GA k-means and k-means are represented in Table 2.

Table 2. Intraclass inertia of k-means, GA k-means and SOM

Clustering technique	k-means	GA k-means	SOM
Intraclass inertia	1.924	1.652	1.843

Table 3. Performance comparison of k-means, GA k-means and SOM

Performance Measure	k-means	GA k-means	SOM
Detection rate	74.4%	86.3%	94.2%
Processing time (sec)	17	35.6	9.5

Table 3. shows the performance comparison of k-means, GA k-means and SOM. The detection rate of SOM is 94.2% due to the co-operative nature for the selection of winning neuron whereas the detection rate of GA k-means and k-means clustering algorithm is 86.3% and 74.4% respectively. SOM reduces the computational complexity by preserving the topological property and it maintains the co-operative neighbourhood for finding the winning neuron. The processing time of GA k-means is higher than k-means and SOM since the optimal value of initial seed is selected iteratively using the genetic algorithm.

5. Conclusions

In this paper, the clustering techniques such as k-means, GA-k-means and SOM are used to detect application layer DDoS attacks and to enhance the performance. The normal traffic is obtained by using the normal browsing activities and the attacks are performed by using different attacking tools, scripts, bots and these attacks are reflected in the access log file of an apache server. The acquired log evidence is pre-processed by extracting the relevant features from the web server log file. These pre-processed features is then passed to the clustering techniques viz., k-means, GA k-means and SOM, which helps to identify the attacks from the analysis of incoming patterns. The experimental results indicate that SOM based clustering technique achieves higher detection rate, reduces false positives and determines unknown attacks than GA-k-means and k-means algorithm.

References

[1] Scarfone, K., Mell, P.: Guide to intrusion detection and prevention systems (IDPS) NIST Special Publications 800-94,1–127 (2007).
[2] Kaspersky Labs, Global it security risks survey 2014 Distributed Denial of Service (DDoS) attacks, 2014, http://media.kaspersky.com/en/B2B-International-2014-survey-DDoS-Summary-report.pdf.
[3] DDoS attack, http://www.digitaltrends.com/computing/ddos-attacks-hit-record-numbers-in-q2-2015/ (Accessed on 25/11/2015).
[4] W. Lee, S. J. Stolfo, "Data mining approaches for intrusion detection," Columbia University, New York dept. of computer science, 2000.
[5] Zhang, Z., Li, J., Manikopoulos, C., Jorgenson, J., Ucles, J.: HIDE: a Hierarchical Network Intrusion Detection System using statistical preprocessing and Neural Network classification, In: Proceedings of IEEE Workshop on Information Assurance and Security, pp. 85–90, (2001).
[6] Govindarajan, M., Chandrasekaran, R.: Intrusion Detection using neural based hybrid classification methods, J. Comput. Netw., vol. 55, 1662–1671, (2011).

[7] Hu, W., Liao, Y., Vemuri, V. R.: Robust anomaly detection using Support Vector Machines, In: Proceedings of International Conference on Machine Learning, pp. 592–597, (2003).
[8] Adrian T.N. Palmer, Computer Forensics, The six steps, US-CERT, (2008).
[9] Liao, N., Tian, S., Wang, T.: Network forensics based on fuzzy logic and expert system, J. Computer Communications, vol. 32, 1881—1892, (2009).
[10] Carrier, B.: File System Forensic Analysis, Addison-Wesley Professional, (2005).
[11] Liao, H. J., Lin, C.-H.R., Lin Y.C., Tung, K.Y.: Intrusion Detection System: a comprehensive review, J. Netw. Comput. Appl., vol. 36, 16–24, (2013).
[12] A. A. Sebyala, T. Olukemi, L. Sacks, and D. L. Sacks, "Active platform security through intrusion detection using naive bayesian network for anomaly detection," In London Communications Symposium, pp.1-5, 2002.
[13] S. S. Kim, A. L. N. Reddy, M. Vannucci, "Detecting traffic anomalies at the source through aggregate analysis of packet header data," Springer Verilog, pp.1-13, 2004.
[14] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection," Pattern Recognition Letters, vol: 51, pp. 1-7, 2015.
[15] T. Yatagai, T. Isohara and I. Sasase, "Detection of HTTP-GET flood attack based on analysis of page access behavior," In Communications, Computers and Signal Processing, IEEE Pacific Rim Conference, pp. 232-235, 2007.
[16] K. Lee, J. Kim, K. H. Kwon, Y. Han and S. Kim, "DDoS attack detection method using cluster analysis," Expert Systems with Applications, vol. 34, No. 3, pp. 1659-1665, 2008.
[17] H. Oh and K. Chae, "Real-Time Intrusion Detection System Based on Self- Organized Maps and Feature Correlations," In Convergence and Hybrid Information Technology, 3rd IEEE International Conference on ICCIT'08, vol. 2, pp. 1154-1158, 2008.
[18] A. Konar and R. C. Joshi, "An Efficient Intrusion Detection System Using Clustering Combined with Fuzzy Logic," Contemporary Computing, Springer Berlin Heidelberg, pp. 218-228, 2010.
[19] Sree TR, Bhanu SM. Identifying HTTP DDoS Attacks Using Self Organizing Map and Fuzzy Logic in Internet Based Environments. In Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics 2016 (pp. 259-269). Springer, India.
[20] Kruegel, C., Vigna, G.: Anomaly detection of web based attacks. In: Proceedings of the 10th ACM conference on communications security, pp. 251–261, ACM, (2003).
[21] M. Zolotukhin and T.Hamalainen, "Detection of anomalous http requests based on advanced n-gram model and clustering techniques," Internet of Things, Smart Spaces, and Next Generation Networking, Springer Berlin Heidelberg, 371-382, 2013.
[22] Bhuyan MH, Bhattacharyya DK, Kalita JK. An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. Pattern Recognition Letters. 2015 Jan 1;51:1-7.
[23] Maggi, F., Robertson, W., Kruegel, C., Vigna, G.: Protecting a moving target: Addressing web application concept drift. In: Kirda, E., Jha, S., Balzarotti, D., (eds.), Recent Advances in Intrusion Detection 2009. LNCS, vol. 5758, pp. 21–40. Springer, Berlin Heidelberg (2009).
[24] Chwalinski P, Belavkin R, Cheng X. Detection of HTTP-GET attack with clustering and information theoretic measurements. In: Foundations and Practice of Security. Springer; 2013. p. 45-61.
[25] Z. Pabarskaite, "Enhancements of preprocessing, analysis and preparation techniques in web log mining," Vilnius Technikes, 2009.
[26] D. E. Golberg,"Genetic algorithms in search, optimization, and machine learning," Addison Wesley, 1999.
[27] P. G. Kumar and D. Devaraj, "Improved genetic algorithm for optimal design of fuzzy classifier," International Journal of Computer Applications in Technology, vol. 35. No. 2, pp.97- 103, 2009.
[28] T. Kohonen, "Self-organized formation of topologically correct feature maps," Biological cybernetics, vol. 43 No. 1, pp. 59-69, 1982.
[29] SOM Toolbox for Matlab, http://www.cis.hut.fi/projects/somtoolbox/
[30] HULK attack, http://github.com/grafov/hulk

[31] OWASP HTTP DdoS attack, www.exploiterz.blogspot.in/2013/07/owasp-http-getpost-ddos-attacker-tool.html.
[32] HOIC attack tool, www.thehackersnews.com/2012/03/another-ddos-tool-from-anonymous-hoic.html.

Authors' Profiles

T. Raja Sree received her B.Tech. in Information Technology from Anna University, Chennai in 2008 and M.Tech. in Information Technology from Anna University, Coimbatore in 2010. Currently, she is pursuing her Ph.D. degree at the Department of Computer Science and Engineering in National Institute of Technology, Tiruchirappalli, India. Her research interests include Cloud Computing, Network security, and Cloud Forensics.

S. Mary Saira Bhanu received her B.E. in Electronics and communication from Madurai Kamaraj University in 1986, M.E. in Computer Science from Bharathidasan University in 1989 and Ph.D. degree from the Department of Computer Science and Engineering from National Institute of Technology, Tiruchirappalli in 2009. Currently, she is an Associate Professor at the Department of Computer Science and Engineering in National Institute of Technology, Tiruchirappalli, India. Her research interests include OS, Real-Time Systems, Distributed Computing, Grid Computing, Cloud Computing, Big Data and Cloud Forensics.

How to cite this paper: T. Raja Sree, S. Mary Saira Bhanu," Investigation of Application Layer DDoS Attacks Using Clustering Techniques", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.8, No.3, pp. 1-13, 2018.DOI: 10.5815/ijwmt.2018.03.01