International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-12 Issue-1, October 2022

Alert Clustering using Self-Organizing Maps and K-Means Algorithm

Dayanand D. Ambawade, Jagdish W. Bakal

Abstract: ***Alert correlation is a system that receives alerts from heterogeneous Intrusion Detection Systems and reduces false alerts, detects high-level patterns of attacks, increases the meaning of occurred incidents, predicts the future states of attacks, and detects root cause of attacks. This paper presents self-organizing maps and the k-means machine learning algorithms to reduce the number of alerts by clustering them.***

Keywords: Alert, Clustering, Intrusion Detection System, K-means, SOM

I. INTRODUCTION

A. Related Work

With the rise in cyber attacks and massive security vulnerabilities continue to pressure the security teams while maintaining the organization's security. Security operations centers (SOCs) are already overburden, and a constant stream of alerts makes their jobs more challenging. For security engineers to effectively detect and respond to threats has become of paramount importance. There is a need to enable more reliable alerts that lead to better response strategies. The existence of an Intrusion Detection System (IDS) is a cornerstone in any modern security architecture. A typical IDS analyzes network traces and generates security alerts when a malicious network packet is detected [1]. It tracks target sources of activities, such as audit and network traffic data in computer or network systems, which deploys various techniques to provide security services. The principal objective of IDS is to detect all intrusions effectively [2]. According to [3], an alert is an alarm generated by an intrusion detection system to notify interested parties of an exciting event. An event is a low-level entity analyzed by IDS. A single event can cause multiple alerts, which is in mathematical expression as below: $Event = alert_1\ alert_2, alert_3...alert_n$

Intrusion correlation refers to the interpretation, combination, and analysis of information from all available sources about the target system activity for intrusion detection and response. There are two intrusion correlation types: event correlation and alert correlation [4].

Manuscript received on 29 September 2022 | Revised Manuscript received on 02 September 2022 | Manuscript Accepted on 15 October 2022 | Manuscript published on 30 October 2022.

***** Correspondence Author (s)

Dayanand Ambawade*, Associate Professor, Department of Electronics and Telecommunication Engineering, Sardar Patel Institute of Technology, Mumbai (Maharashtra), India. E-mail: dd_ambawade@spit.ac.in, ddambawade1@gmail.com

Dr. Jagdish W. Bakal, Professor and Principal, Pillai HOC College of Engineering and Technology, Mumbai (Maharashtra), India.

The major difference between these two types of intrusion correlations is that intrusion event correlation analyses raw or neutral events while intrusion alert correlation analyses identify misuse or anomalies. The analyzer in IDS detects an event that matches the rule; it sends an alert to its manager(s) [3].

Regarding data processing types, IDS are divided into two categories: Signature-based and Anomaly-based IDSs. Anomaly-based IDSs detect abnormal behaviors by checking statistical information about system execution and maintaining normal behavioral patterns. Misuse-based IDSs maintain suspiciously or attack pattern categories. An alert is generated whenever the received information corresponds with the IDS signature or contradicts normal behavioral patterns [5].

The common problems in the generation of alerts are a large number of alerts, false alerts, and the inability to identify the multistep attack. To solve these problems, an alert correlation system is used. The number of alerts sometimes generated is so huge that it is impossible to do manual analysis [6]. The first step in the alert correlation system is to reduce those alerts so that it becomes easy for a human analyst to read; thus, alert clustering is essential. This paper addresses the alert clustering step of correlation systems.

B. Snort IDS

Snort is an open-source network intrusion prevention and detection system. It uses a rule-based language combining signature, protocol, and anomaly inspection methods [10][11]. It works in the following modes:

a. A packet sniffer: captures and displays packets from the network with different levels of detail on the console
b. Packet logger: log data in text file
c. Honeypot monitor: deceiving opposing parties
d. NIDS: network intrusion detection system

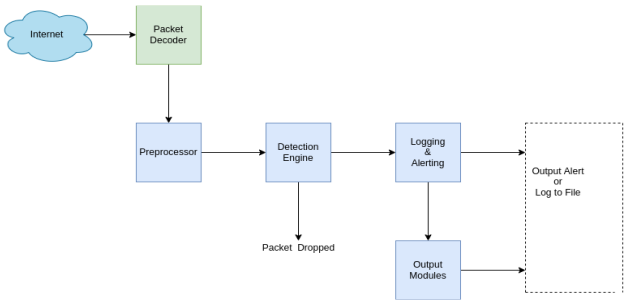

Figure 1: SNORT IDS Components.

Retrieval Number: 100.1/ijeat.A38521012122
DOI: 10.35940/ijeat.A3852.1012122
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

We are concerned with the intrusion detection capabilities of snort in this paper. The packet decoder takes packets from different network interfaces and prepares the packets to be preprocessed or sent to the detection engine. "Preprocessors are plug-ins that can be used with Snort. It is to arrange or modify data packets before the detection engine does some operation to find out if an intruder is using the packet. Some preprocessors also detect anomalies in packet headers and generate alerts. Preprocessors are important for any IDS to prepare data packets to be analyzed against the rules in the detection engine. The detection engine is the most key part of Snort. Its responsibility is to detect if any intrusion activity exists in a packet. The detection engine employs Snort rules for this purpose. The rules are read into internal data structures or chains where they are matched against all packets. If a packet matches any rule, appropriate action is taken; otherwise, the packet is dropped. Depending on what the detection engine finds inside a packet, the packet may log the activity or generate an alert [10][11].

C. Dataset

The 1999 DARPA/MIT Lincoln Laboratory off-line intrusion detection evaluation data set is used for analysis. In this dataset, for each TCP/IP connection, 41 various quantitative and qualitative features were extracted [12][13]. Attack types were divided into four main categories:

a. Probing is reconnaissance class of attacks where an adversary scans a network to gather information to identify known vulnerabilities. An attacker with a map of machines and services available on a network can manipulate the information to look for exploits.

b. Denial of Service (DOS) is a class of attacks where an adversary makes some computing resources too much engage or too full to handle legitimate requests, denying legitimate users access to a machine.

c. User to Root (U2R) - In this attack, an attacker starts with access to a normal user account on the system by gaining root access.

d. Remote to User (R2L) - This attack happens when an attacker sends packets to a machine over a network that exploits the machine's vulnerability to gain local access as a user illegally.

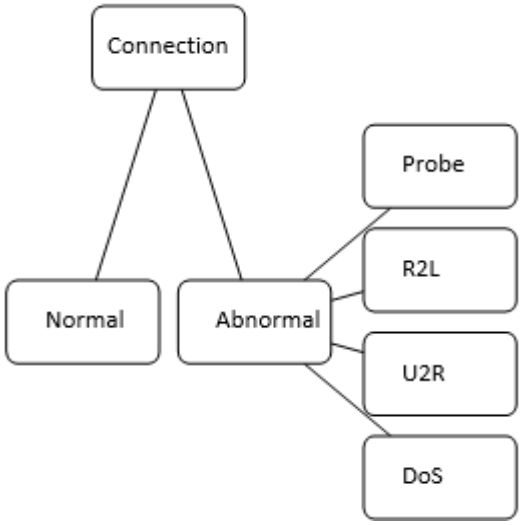

Figure 2: Attack classification.

II. ALGORITHMS

A. Self Organizing Maps

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN). It is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. SOMs are useful for visualizing low-dimensional views of high-dimensional data. In this paper, we use the Kohonen SOM model [7][8]. The principal goal of a SOM is to transform an incoming signal pattern of arbitrary dimension into a one or two-dimensional discrete map and to perform this transformation adaptively in a topologically ordered fashion. Learning in the self-organizing map aims to cause different parts of the network to respond similarly to certain input patterns. Each node has a specific topological position (an x, y coordinate in the lattice) and contains a vector of weights of the same dimension as the input vectors. If the training data consists of vectors, X, of n dimensions

X_1, X_2, X_3, X_4....

Then the weight vector W, corresponding to each node, will be

W_1, W_2, W_3, W_4...

The algorithm involves two stages-first is competitive learning, and the second is cooperative stage.

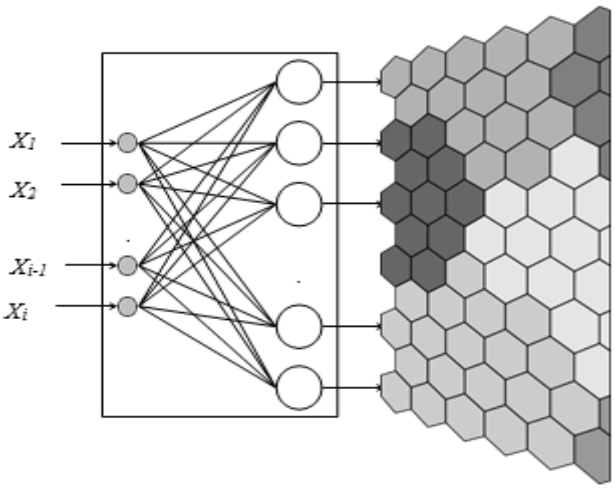

Figure 3: Kohonen Network / SOM Feature Map Steps

- Initially, node weight vectors are randomly initialized. The first phase, competitive learning, begins. A training example is presented to the network of nodes, and its Euclidean distance is calculated from all the weight vectors. The neuron whose weight vector is most similar to the input (least Euclidean distance) is called the best matching unit (BMU). Distance calculation for V1

$$\text{Distance}(dist) = \sqrt{\sum_{i=1}^{i=n} (X1 - Wi)^2}$$

Retrieval Number: 100.1/ijeat.A38521012122
DOI: 10.35940/ijeat.A3852.1012122
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

- The radius of the neighborhood of the BMU is now calculated. This value starts large, typically set to the 'radius' of the lattice, but diminishes after each iteration. The width of the BMU at a given iteration is given by

$$\sigma(t) = \sigma 0 e^{(-\frac{t}{\lambda})}$$

σ_0 – Initial BMU Radius
λ – Time Constant

- Each neighboring node's weights are adjusted to make them more like the input vector. This is the second phase, called cooperation learning. The closer a node is to the BMU, the more its weights get altered. Any nodes within this radius are deemed inside the BMU's neighborhood, and their weights are updated.

$$W(t+1) = W(t) + \theta(t)L(t)(V(t) - W(t))$$

L(t) is the learning rate which is given by

$$L(t) = Loe^{(-\frac{t}{\lambda})}$$

θ is used to represent the amount of influence a node's distance from the BMU has on its learning.

$$\theta(t) = \exp(-dist^2/2\sigma 2(t)$$

$$\theta(t) = \exp(-dist^2/2\sigma 2(t))$$

Where *dist* and σ are values calculated from the steps above. [7]

A. K-Means Clustering

- K-Means clustering a method to partition n objects into k clusters. In which each object be in the cluster with the nearest (centroid) mean. This produces exactly k different clusters of the greatest possible dissimilarity. The most significant separation (distance) is not known as a priori and must be computed from the data to have best number of clusters k.
- The important objective of K-Means clustering is to minimize total intra-cluster variance or the squared error function.
- The objective of K-Means clustering is to minimize total intra-cluster variance or the squared error function.

Objective function f is given by

$$f = \sum_{j=1}^{j=k} \sum_{i=1}^{i=n} (xi - cj)^2$$

k- Number of clusters
n- Number of cases
c_j - Centroid for cluster j

- The first step in K-means is selecting the number of k(number of clusters). Then randomly assign k points as cluster centroids.
- Calculate the distance of each training example from the cluster centroids according to the Euclidean distance. Assign cluster centroids to the data points closest to them.
- Calculate the mean of the data points inside each cluster, which should be assigned as the new cluster centroid.
- Repeat the steps above for several iterations.

III. PROCEDURE

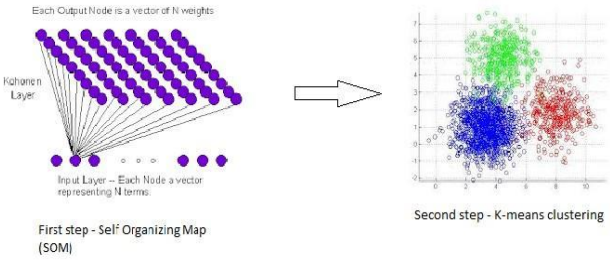

Figure 4: The two-step procedure.

We pass the 1999 DARPA data set through Snort IDS. Snort generates alerts based on the abnormality of the packet. We collect those alerts and analyze the features of the packets.

> The Snort is configured (\etc\snort\snort.conf) and functioned correctly. The following field names are available [14]:
>
> *timestamp, sig generator, sig id, sig rev, msg, proto, src, srcport, dst, dstport, ethsrc, ethdst, ethlen, tcpflags, tcpseq, tcpack, tcpln, tcpwindow, ttl, tos, id, dgmlen, iplen,icmptype, icmpcode, icmpid, icmpseq,and default*
>
> By specifying "default" as a field name, a default set of field names is used.

After much analysis following features are selected for clustering. We pass the features in Table II to Self Organizing Maps (SOM) model. We essentially convert feature space of 11 dimensions into two dimensions which would be easy to visualize. The second advantage of using SOM is that it would gather alerts of similar traits and map them into a single node.

We have taken a grid of 10 x 10 lattices of nodes. Each node is connected with a weight vector to the input. The weight vector with maximum weightage (least euclidean distance) corresponding to a particular input vector closely resembles that input vector. SOM allows us to visualize a single variable's distribution across the nodes' lattice. In the next step, we apply the k-means algorithm to the 10 x 10 grid lattice nodes. However, first, we need to calculate the number of clusters required to perform k- means. We use the elbow method to calculate the number of clusters [9]. We plot the objective function f discussed in the Algorithms section against the k. We select k=13 as the number of clusters.

Table I: Features of Packets in Generated Alerts

timestamp	sig_ generator	sig_ id
sig_ rev	msg	protocol
srcip	dstip	srcport
dstport	ethsrc	ethdst
ethlen	tcpflags	tcpseq
tcpack	tcpwindow	ttl
tos	id	iplen

Table II: Features Used for Clustering

sig_ id	msg	protocol
srcport	dstport	iplen
dstport	ethsrc	ethdst
ethlen	ttl	

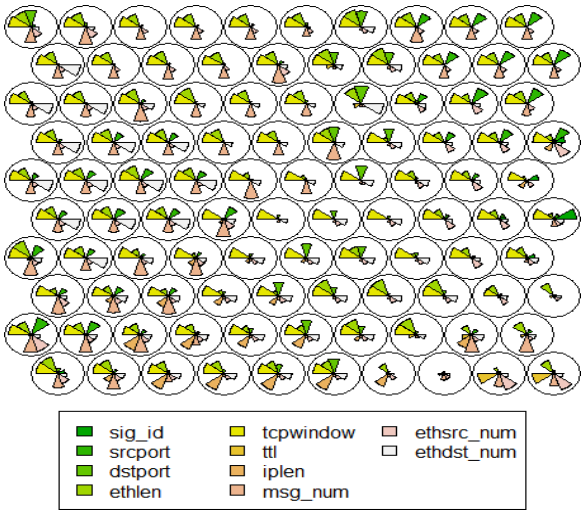

Figure 5: Grid of 10 x 10 nodes showing weightage of each of the input features. Greater the area of the fan, the greater the weightage of the input vector for that corresponding node.

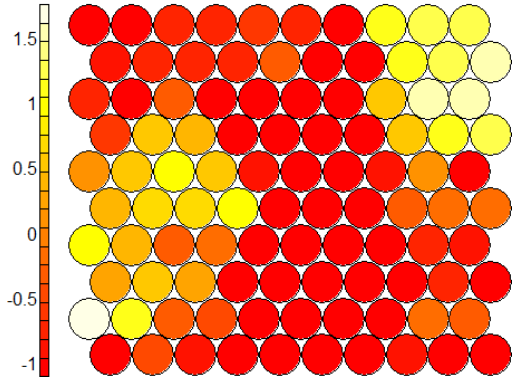

Figure 6: The distribution of the srcport variable across the nodes. The scale on the left gives us an estimate of weights associated with a node regarding the input.

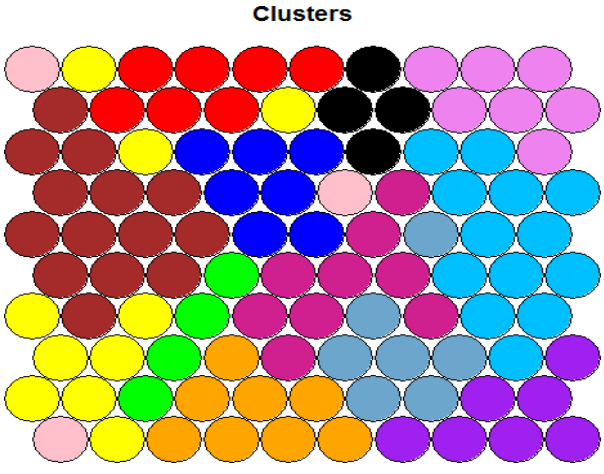

Figure 7: Partition of weight vectors into 13 clusters. Each cluster contains similar nodes.

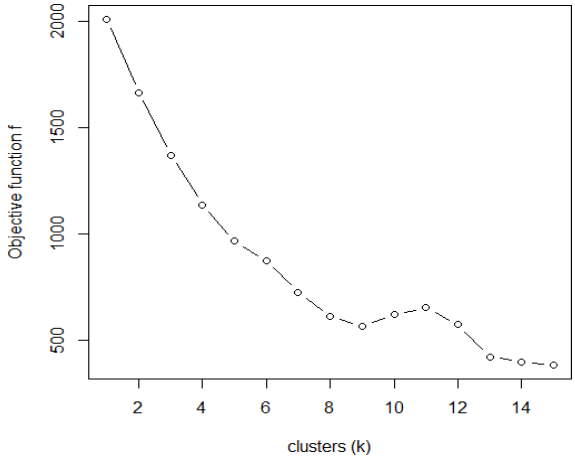

Figure 8: Elbow method, Plot of objective function f versus k.

IV. RESULTS

Looking at Table III, we observe the following:

- For the alert, Protocol mismatch, we have the destination port as 22. The IP datagram length is constant for all packets in this alert cluster, i.e., 60.
- For SENSITIVE-DATA Email Addresses alert, we have destination port as 25 and a variable IP datagram length ranging from 400 to 17000.
- For alert NO CONTENT-LENGTH OR TRANSFER ENCODING IN HTTP RESPONSE, we have source port as constant 80 and destination port varying between 23 to 30000. The IP datagram length varies from 40 to 23000.
- For other alerts under HTTP, such as LONG HEADER, SIMPLE REQUEST, and UNESCAPED SPACE IN HTTP URI, the destination port is constant as 80 and has a constant IP datagram length of 40.
- For alert, Consecutive TCP small segments exceed the threshold, the source port is 23, and the variable IP datagram length is 70-168. The sig id column is directly related to the alert messages. For each message, there is a corresponding sig id number.
- Almost all the alerts have TTL as 63 or 64 ms except for SENSITIVE-DATA Email Addresses and Protocol mismatch, which have 255 ms TTL (time-to-live) in some cases LONG HEADER and NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE, alerts have 128 ms TTL.

V. CONCLUSION

We introduce a method of intrusion alert clustering using Self Organizing Maps (SOM) and the K-means algorithm. Applying SOM + K-means, we have successfully clustered the snort alerts into 13 clusters. Each cluster defines a particular type of attack. This method reduces the significant workload of the security team. This time saving helps the security team to address issues significantly faster than they could before, elevating their security. We could use this knowledge to classify false alerts further, and the model could also be used to classify new alerts.

Retrieval Number: 100.1/ijeat.A38521012122
DOI: 10.35940/ijeat.A3852.1012122
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

REFERENCES

1. Manganiello, F., Marchetti, M., Colajanni, M. (2011). Multistep Attack Detection and Alert Correlation in Intrusion Detection Systems. In: Kim, Th., Adeli, H., Robles, R.J., Balitanas, M. (eds) Information Security and Assurance. ISA 2011. Communications in Computer and Information Science, vol 200. Springer, Berlin, Heidelberg. [CrossRef]
2. Vaibhav Gowadia, Csilla Farkas, Marco Valtorta, "PAID: A Probabilistic Agent-Based Intrusion Detection System," Computers Security, Volume 24, Issue 7,2005, Pages 529-545, ISSN:0167-4048. [CrossRef]
3. Antti Hatala, Camillo Sa¨rs, Ronja Addams-Moring and Teemu-pekka," Event Data Exchange and Intrusion Alert Correlation in Heterogeneous Networks," Proceedings of the 8th Colloquium for Information Systems Security Education West Point, NY, June 2004
4. Gorton, D., "Extending Intrusion Detection with Alert Correlation and Intrusion Tolerance.", MPhil Thesis, Chalmers University of Technology, Department of Computer Engineering, Goteborg, Sweden (2003)
5. Mirheidari, S.A., Arshad, S., Jalili, R.," Alert Correlation Algorithms: A Survey and Taxonomy.", In Wang, G., Ray, I., Feng, D., Rajarajan, M. (eds) Cyberspace Safety and Security. CSS 2013. Lecture Notes in Computer Science, vol 8300. Springer(2013) [CrossRef]
6. Wesley Mullins, Alert fatigue crippling security operation centers-Discover how to improve alert fatigue and catch security threats. April 11, 2022, https://www.securitymagazine.com/articles/97400-alert-fatigue-crippling-security-operation-centers
7. Kohonen Self-Organizing Maps by Shyam M. Guthikonda, Wittenberg University December 2005
8. Kohonen's Self Organizing Feature Maps [Online],Available-http://www.ai-junkie.com/ann/som/som1.html
9. Trupti M. Kodinariya, Dr. Prashant R. Makwana, "Review on determining the number of Cluster in K-Means Clustering", International Journal of Advance Research in Computer Science and Management Studies, November 2013
10. Rafeeq Ur Rehman, Introduction to Intrusion Detection and Snort, Intrusion Detection Systems, Upper Saddle River, New Jersey, Pearson Education, 2003 with Snort
11. Tony Howlett, Open Source Security Tools: A Practical Guide to Security Applications, Pearson Education, Inc (2005)
12. 1999 DARPA Intrusion Detection Evaluation Data Set, Available. https://www.ll.mit.edu/ideval/data/1999data.html
13. Wenke Lee, Sal Stolfo and Kui Mok, "A Data Mining Framework for Building Intrusion Detection Models", Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, 1999.
14. Official Documentation and Snort FAQ https://www.snort.org/faq/readme-csv

AUTHORS PROFILE

Dayanand Ambawade, is currently working as Associate Professor in Electronics and Telecommunication Engineering Department, Sardar Patel Institute of Technology, Mumbai. Working as IEEE SB Counselor and member of Network Infrastructure Team of Institute. Received M.Tech in Communication Engineering from Indian Institute of Technology Bombay (IIT-B). More than 20+ years teaching experience in the areas of Computer Networking, Network Security & Management, Open Source Software (OSS). Member of IEEE, Fellow of IETE, Life Member of ISTE.

Dr. Jagdish W. Bakal, is currently working as Professor and Principal, Pillai HOC College of Engineering and Technology, Mumbai University. Received M.Tech and Ph.D in Computer Engineering. More than 25+ years of teaching experience in the areas of interest includes IoT, Mobile Networking, Information Security. Member of IEEE, President of IETE, Life Member of ISTE, member of CSI,I/C Director at the School of Engineering and Applied Sciences, University of Mumbai.

Table III: Cluster Containing Packets of Following Features

Cluster No	sig_id	srcport	dstport	ethlen (bytes)	tcpwindow	ttl (ms)	iplen (bytes)	msg	ethsrc	ethdst
Cluster 1	4	3908	22	699	32120	63	60	10	3	1
Cluster 2	5	80-33708	25	3-615	20440-32736	63-64	4091-17124	13	1,3,4,6	1,3,4
Cluster 3	4	22	3908	39-980	31744-32736	64	60	10,13	1	3
Cluster 4	3	80	1029-32727	262-750	20440-32120	63-64	8000-23110	4	3	4
Cluster 5	3	80	15463-33103	738-997	32120	63,128	100-3985	4	2,3	3,4,6
Cluster 6	8,19,32,33	23-33092	80	556	32120	63,64	40	2,3,6,7	1,3,4	1,3,4
Cluster 7	5,12	23,25	4186-32665	741-889	32120-32736	63,64	73,168	11,13	1,3,4,5,6	3,4
Cluster 8	3,4,5,8	22-33052	22,4192	699	8760	60,63,128,255	40,60	2,4,10,13	1-6	1-6
Cluster 9	4	25658-29977	22	699	32120	63	60	10,13	3	1
Cluster 10	4	1652-230	22	699	32120	63	60	10	3	6
Cluster 11	3	80	1027-33106	2-485	32120	63,64	242-11748	4	1,3	3,4
Cluster 12	5	25-33044	25	278-985	32120	63,64	73-3904	13	3,4,6	3,4
Cluster 13	3	80	23-29714	417-997	32120	63,128	41-4039	4	2,3	3,4

Table-IV: Messages Corresponding to The Msg Column in Table III

msg_number	msg
1	(ftp telnet) FTP bounce attempt
2	(http inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE
3	(http inspect) LONG HEADE
4	(http inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE
5	(http inspect) NON-RFC DEFINED CHAR
6	(http inspect) SIMPLE REQUEST
7	(http inspect) UNESCAPED SPACE IN HTTP URI
8	(http inspect) UNKNOWN METHOD
9	(spp sdf) SDF Combination Alert
10	(spp ssh) Protocol mismatch
11	Consecutive TCP small segments exceeding threshold
12	Reset outside window
13	SENSITIVE-DATA Email Addresses

Table V: Corresponding to Table Iii Mac Address of Source and Destination

Sno	**ethsrc/dst**
1	00:00:C0:17:79:5A
2	00:10:5A:9C:B2:54
3	00:10:7B:38:46:33
4	00:C0:4F:A3:57:DB
5	08:00:20:09:B9:49
6	08:00:20:89:A5:9F

Retrieval Number: 100.1/ijeat.A38521012122
DOI: 10.35940/ijeat.A3852.1012122
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)