Available online at http://proceeding.rsfpress.com/index.php/ess/index
LPPM UPN "Veteran" Yogyakarta Conference Series
Proceeding on Engineering and Science Series (ESS)
Volume 1 Number 1 (2020): 160-170

Clustering K-Means Using SNORT Application For Denial Of Service Attacks

Rifki Indra Perwira, Bagus Muhammad Akbar, Hari Prapcoyo

Universitas Pembangunan Nasional Veteran Yogyakarta
[1]E-mail address rifki@upnyk.ac.id; [2]E-mail address bagusmuhammadakbar@upnyk.ac.id; [3] E-mail address hari.prapcoyo@upnyk.ac.id

Abstract

Data quality and transparency are of the utmost importance for organizations. Collecting original data from the source without any indication of interruption or interception is an indicator of an attack on the server. The most common attack is Denial of Service (DoS), which is a type of pattern that will crash, shutdown, reboot, or not respond to services of a host on the network. One technique for this attack is the use of the k-means clustering method with a snort. This study aims to design a SNORT-Intrusion Detection System (IDS) application with a k-means algorithm that can categorize attacks into high, medium, and low attacks and is accurate on DoS attacks. Snort accuracy testing functions to measure the packet size detected by snort using an attack application, then the number of packets caught can be categorized using clustering techniques. From the measurement results, the increase was 73.18%. The contribution of this research is a survey and analysis of anomalous packets contained in a network. It can identify the level of types of attacks and take preventive measures from these attacks.

Keywords: Clustering, K-Means, Snort, DOS

I. INTRODUCTION

Good data service quality is an asset 4 in an organization (Maliki, Irfan, 2016). The data services referred to in it are in the form of data availability and transparency. Data transparency is a condition when the original data is collected from the source (server) without any indication that previous interruption or interception has occurred. The common thing that often happens is that an attack on a running system can damage or paralysis of the system infrastructure. An attack, if it is not known and handled quickly, can disrupt the work of the organization. One of the most common attacks is Denial of Service (DoS). The attack, which is identical to the designation DoS or DDoS, is a type of pattern that will crash, shutdown, reboot, or not respond to a host's service on the network (Juwita, 2013). One system that can detect attacks from DoS is the Intrusion Detection System (IDS). IDS is a software application that can see suspicious activity in a system or network. IDS will monitor data traffic on a network or retrieve data from log files. Clustering techniques are needed to create a grouping scheme for a DoS attack so that an attack can be categorized as an emergency attack or how to handle it. Clustering has an impact on the availability and speed of services of an organization restoring a paralyzed infrastructure—data obtained from UPT. Information Technology and

DOI: https://doi.org/10.31098/ess.v1i1.107

Communication of UPN "Veteran" Yogyakarta, monitoring of attacks is still done manually in monitoring thousands of incoming data packets by relying on server logs. From these problems, it can be concluded that the current method is still ineffective in dealing with attacks that should require fast handling to avoid downtime. Therefore we need a system that can group and provide DoS attack information. In this study, the K-Means clustering algorithm process uses log data by looking at the hit count and duration attack parameters from ICMP Normal, ICMP Flooding, UDP Flooding, TCP SYN Flooding, and TCP PSH-ACK Flooding. The challenge of this research is that log data is stored using the horizontal partitioning technique in the database on the NFAT engine to simplify and speed up the process of finding information related to attacks for forensic purposes.

II. LITERATURE REVIEW

Previous research has included:

Traditional IDS has low detection capability and a high false alarm rate (Xiaofeng & Xiaohong, 2018). IDS will analyze and, with a specific algorithm, decide to give a warning (Gondohanindijo, 2011). To solve the Intrusion Detection System problem, it still requires the right algorithm according to its designation. The K-Means clustering algorithm is an algorithm that can improve invasion detection, reduce false detection rates (Yang, 2017) as well as cluster DoS attacks.

Previous similar research was conducted by Ananta (2017), which discusses the intrusion detection system based attack notification using the K-Means Method. This research requires protocol data and destination port when normalizing training data. The challenge of this research is to send an early warning of attack via short message service (SMS) in real-time labeled with malicious attack data. The weakness of this research is that the attack notification does not indicate the type of attack that occurs in the SMS message content, and all episodes are considered dangerous.

Another research was conducted by (Heryanto, A., Stiawan, D., 2016), which discusses the description of Denial of Service attacks by clustering using the K-Means Algorithm. K-Means in this study compared the attack data and standard data using the ISCX (Information Security Center of eXcellence) dataset. The percentage of accuracy using the ISCX dataset using the K-Means algorithm is 97.83%, the detection rate is 98.63%, and the results of the confusion matrix calculation for false alarms from the program are 0.02%. The drawback of this study is that it does not indicate the type of attack detected.

Research by Yang (2017) took the topic of the Efficient K-means Algorithm in Intrusion Detection. This study assesses how efficient attack prevention and detection is based on the maximum density and distance. K-Means uses the word Markov distance of the sample on the sample data to perform clustering. Each model obtained attacks according to the category and the grouping results used in intrusion detection. Experiments prove that the proposed method has a detection rate of 88.23% and a lower error rate of 1.92.

Based on some of the studies above, no one has discussed implementing a decision-making system using SNORT replicas as an Intrusion Detection System (IDS) and the K-Means clustering algorithm to detect and group Denial of Service attacks. Snort functions as a packet sniffer to see network activity used as an alert and classify attacks that occur. The signals generated by the snort are then carried out by clustering techniques using the K-Means algorithm by looking at the number of hits and duration parameters needed to categorize attacks as dangerous (high risk), rather dangerous (medium risk), and not harmful (low risk).

III. RESEARCH METHODOLOGY

The method of research and system development will be built and used as a reference in designing a Flooding attack monitoring application using the K-means algorithm with the clustering technique with the stages of data collection and system design. The data collection process is carried out by observing the snort IDS device that has been built to obtain alerts for attacks that have occurred on the network system being built. The data obtained were then analyzed and produced a concept as a reference for research in solving the problem formulation. The hypothesis is supported by conducting a literature study of several works of literature related to existing problems, while literature is used as reference material to complete research.

III. 1 Data collection.

In the detection of attacks on this network, there are four stages of data collection, namely: literature study, observation, interviews, and data collection. The data collection process is carried out by analyzing the problems associated with the flooding attacks that occur in the IDS system to be designed. Interviews were conducted with staff related to this research.

III.1.1. Observation

Observations are made by identifying the packets sent by the host, which acts as an attacker to the host that acts as a victim of IDS snort in real-time. Each package sent or received will be selected according to the rules designed to detect ICMP Normal, ICMP Flooding, UDP Flooding, TCP SYN Flooding, and TCP PSH-ACK Flooding. The attributes that will be captured are as follows.

Table 1. Observation Stage

No	Process	Description
1	*Capture traffic*	Recording of data traffic in the network. The output from this process will be saved into the snort log database.
2	*Extract data*	The process of extracting log files stored in the snort log database is by taking some information that will be used as parameters, namely: o Ip_src contains the IP address information from which the packet was recorded by a snort and stored in the snort log database. o Ip_dst contains IP address information for packet destination recorded by Snort and stored in the snort log database. o Signature contains attack information detected by snort based on the rules created. o layer4_sport includes the port from which snort detected the packet. o layer4_dport consists of the destination port of the package that snort detects. o Ip_proto contains information in the protocol layer that the package passes through. o Timestamp contains information on the length of time the packet was detected. o Total_alert contains information on the number of packets detected. o Duration includes the length of time the package was detected per Signature seen by the snort

		o
3	*Clustering*	Grouping data that has been stored in the snort database into three groups, namely: o High o Medium o Low

III.2 Development System.

The system development methodology used in this study is the waterfall. The waterfall method has the main stages of the waterfall model that reflects the necessary development activities (Sommerville, 2011). This model includes corrections of various errors that were not found in the previous steps, improvements to the implementation of the system unit, and system development for adjusting user needs as a reference for the output of this study.

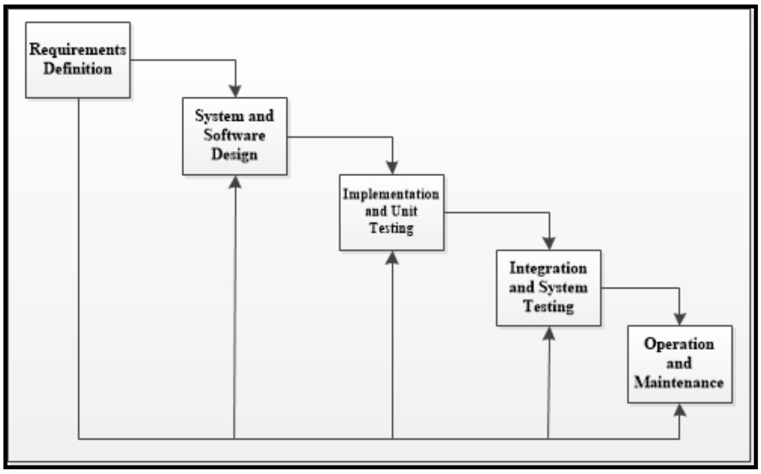
Requirements Definition
System and Software Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Figure 1. Waterfall model.

III.3. Information Gathering.

Information gathering was obtained based on the analysis of the data obtained in the data processing. Information collection is needed to understand an application to be built. The information collected results from observations from related books and the internet. Then field observations are made to look for the needs needed to categorize the flooding attack.

IV. FINDING AND DISCUSSION

The flowchart image above is the process of retrieving alert data captured by Snort, which is used in the k-means clustering calculation process. Alerts collected in the database are grouped into high, medium, and low categories. After that, the generator centroid K is generated and calculates the data distance to the generator centroid cluster. Calculate the data distance to the centroid cluster by looping until the centroid value does not change.

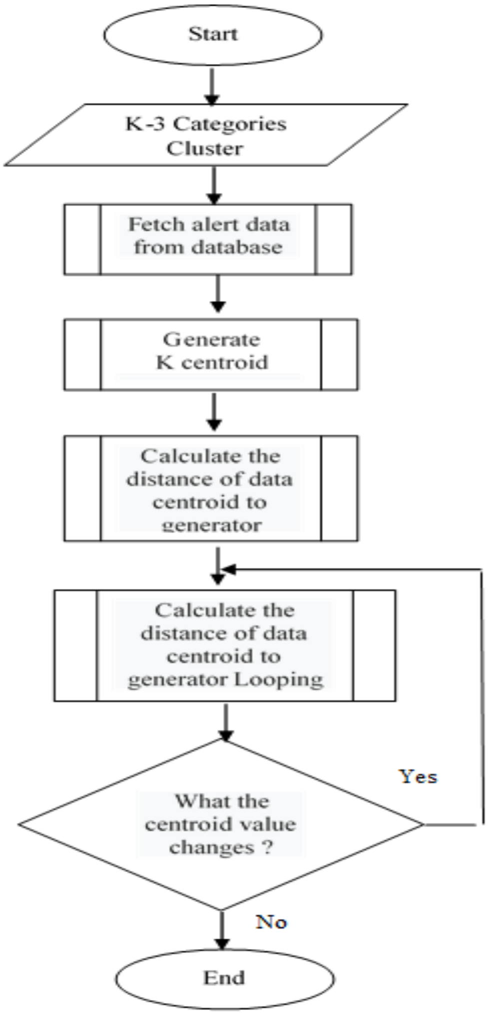
Start
K-3 Categories Cluster
Fetch alert data from database
Generate K centroid
Calculate the distance of data centroid to generator
Calculate the distance of data centroid to generator Looping
What the centroid value changes ?
Yes
No
End

Figure 2. Flowchart clustering k-means.

IV. 1. Sample k-means clustering calculations

A. Stages of the k-means clustering process

In the clustering calculation, there are three steps: determining the number of clusters, generating the centroids, and calculating the distance for each data to the center of the group. The clustering parameters used are the number of items and the duration of each IP address.

Table 2. datalog Snort

No	Ip.src	Ip.dst	Sig.name	Dst.port	Alert	Duration(s)	Protokol
1	192.168.1.4	192.168.1.6	TCP Syn flood Attack Detected	80	2739	301	TCP
2	192.168.1.3	192.168.1.6	TCP Syn flood Attack Detected	80	2181	307	TCP
3	192.168.1.10	192.168.1.6	TCP Syn flood Attack Detected	80	1755	205	TCP
4	192.168.1.12	192.168.1.6	UDP flood Attack Detected	80	2134	292	UDP
5	192.168.1.18	192.168.1.6	UDP flood Attack Detected	80	2129	289	UDP
6	192.168.1.8	192.168.1.6	UDP flood Attack Detected	80	3078	284	UDP
7	192.168.1.25	192.168.1.6	Push Ack flood Attack Detected	80	1328	263	TCP
8	192.168.1.21	192.168.1.6	Push Ack flood Attack Detected	80	2512	284	TCP
9	192.168.1.27	192.168.1.6	Icmp Flooding Detected	-	2562	244	ICMP
10	192.168.1.23	192.168.1.6	Icmp Flooding Detected	-	4825	382	ICMP

IV.1.1. Determine the number of clusters

In the clustering process, the flooding attaches attack has determined 3 clusters, namely high, medium, and low.

IV.1.2. Generating K centroid.

Determine the position of the centroid for each cluster using an interval that has a range of maximum and minimum values then divided by the number of groups.

$$IntervalX1 \leftarrow \frac{(maxx1-minx1)}{(k)}$$

$IntervalX2 \leftarrow \frac{(maxx2 - minx2)}{(k)}$

$IntervalX1 = \frac{(4825-1328)}{(3)} = 1165{,}6666666667$

$IntervalX2 \leftarrow \frac{(maxx2 - minx2)}{(k)}$

$IntervalX2 = \frac{(382-205)}{(3)} = 59$

When:

X1 = Amount of attack

X2 = Duration (s)

K = 3

When the interval values for the number of alerts and duration are obtained, continue by finding the value of the generator centroid for each cluster.

a. $C1(X1C1, X2C1)$

- ○ $X1C1 \leftarrow minx1 + (2 * IntervalX1)$
- ○ $X1C1 \leftarrow minx2 + (0 * IntervalX2)$

b. $C2(X1C2, X2C2)$

- ○ $X1C2 \leftarrow minx1 + (1 * IntervalX1)$
- ○ $X1C2 \leftarrow minx2 + (1 * IntervalX2)$

c. $C3(X1C3, X2C3)$

- ○ $X1C3 \leftarrow minx1 + (0 * IntervalX1)$
- ○ $X1C3 \leftarrow minx2 + (2 * IntervalX2)$

Table 3. Data generator centroid

No	Centroid total attack	Centroid duration	Centroid Cluster	Category

1	Cluster 1
	K2=1328 + (2 x 1165,6666666667)
	K2=3659.3333333333
		Cluster 1
		K2 = 59 + (0 x 205)
		K2 = 205
			(3659.3333333333 , 205)	High

2	Cluster 1
	K1=1328 + (1 x 1165,6666666667)
	K1=2493.6666666667
		Cluster 2
		K1 =59 + (1 x 205)
		K1 = 264
			(2493,6666666667 , 264)	Medium

3	Cluster 3
	K0=1328 + (0 x 1165,6666666667)
	K0=1328
		Cluster 3
		K0 = =59 + (2 x 205)
		K0 = 323
			(1328 , 323)	Low

IV.1.3.Calculate the distance for each data center to the generator cluster.

To find out the high, medium, and low categories for each IP address detected by the snort, the distance calculation for each data is carried out based on the number of alerts and duration with each generator centroid obtained in the previous step. Each data generator cluster centroid has the closest distance to one of the cluster centers; it is included in the cluster category. The process of calculating the most relative distance is repeated using the euclidean distance concept.

IV.2. Stages of the calculation process of generating cluster centroid generators

Calculation of data to the centroid in cluster 1 uses equation (1), cluster 2 uses equation (2), and cluster 3 uses equation (3).

$$L_1 \leftarrow d(e_1, C_1) = \|e_i - C_1\|^2 = \sqrt{\sum_{i=1}^{n}\left(e_{i_i} - C_{1_i}\right)^2} \quad \ldots\ldots (1)$$

$$L_2 \leftarrow d(e_1, C_2) = \|e_i - C_2\|^2 = \sqrt{\sum_{i=1}^{n}\left(e_{i_i} - C_{2_i}\right)^2} \quad \ldots\ldots (2)$$

$$L_3 \leftarrow d(e_1, C_3) = \|e_i - C_3\|^2 = \sqrt{\sum_{i=1}^{n}\left(e_{i_i} - C_{3_i}\right)^2} \quad \ldots\ldots (3)$$

No	**Ip.src**	**Jumlah Alert**	**Duration (s)**	**K2**	**K1**	**K0**	**Category**
1	192.168.1.4	2739	301	925.326669044421	248.10772749845	1411.171499145 3	*medium*
2	192.168.1.3	2181	307	1481.847982906 6	315.60963934019	853.1500454199 1	*medium*
3	192.168.1.10	1755	205	1904.333333333 3	741.01919303379	443.0045146496 8	*low*
4	192.168.1.12	2134	292	1527.812415769	360.75491834639	805.5959335379 8	*medium*
5	192.168.1.18	2129	289	1532.636979558 8	365.52260911984	801.7212732614 7	*medium*
6	192.168.1.8	3078	284	586.6766097642 2	584.67550354401	1750.434517484 2	*medium*
7	192.168.1.25	1328	263	2332.054697281 2	1165.6670956057	60	*low*

8	192.168.1.21	2512	284	1150.0499022989	27.131367660166	1184.6421400575	*medium*
9	192.168.1.27	2562	244	1098.0261583607	71.200031210979	1232.5261824968	*medium*
10	192.168.1.23	4825	382	1179.0283193282	2334.317697125	3497.4976769113	*high*

The results of the sampling calculation are as follows.

Table 4. Result of calculation

Cluster	Centroid Generator	Distance
1	(3659.333333, 205)	1179.0283193282
2	(2493.666667, 264)	2334.317697125
3	(1328, 323)	3497.4976769113

Table 5. Data cluster centroid generator

IV.3. Accuracy Testing Results

Snort accuracy testing functions to determine the packet size detected by snort using an attack application, then the number of packets caught can be categorized using clustering techniques. In this study, they test the accuracy of the attack packet using the Hping3 and Loic tools to attack the destination IP that has been paired with a snort and is connected to the flooding attack monitoring application. The data to be carried out in this test were eight attacks on port 80 on 23 September 2020. The results of the attacks to be tested can be seen in Table 6 as follows.

Table 6. Accuracy Results

No	Signature name	IP Source	Port	Request Issued	Request Received	Detect
1	Push Ack Flood Attack	192.168.1.2	80	1722150	1935000	89%
2	TCP Syn Flood Attack Detected	192.168.1.7	80	4140396	294900	14.04%
3	UDP Flood Attack Detected	192.168.1.13	80	118560	197600	60 %
4	ICMP Flooding Detected	192.168.1.24	-	4464069	2446000	54.79%
5	Ping ICMP Normal	192.168.1.28	-	220	220	100%
6	Push Ack Flood Attack	192.168.1.31	80	102756	107600	95.50%
7	UDP Flood Attack Detected	192.168.1.32	80	119096	124800	95.43%
8	Ping ICMP Normal	192.168.1.29	-	56	56	100 %

$$\frac{89 + 14.04 + 60 + 54.79 + 100 + 95.50 + 95.43 + 100}{8} \text{x } 100\% = 76.095\%$$

V. CONCLUSION AND FURTHER RESEARCH

K-Means clustering method use rule as a string match to make decisions. The weakness of this IDS based on string matching is that the occurrence of strings in a packet must be exactly the same, making it difficult to detect attacks that are similar but have different string patterns. Based on the results of the analysis, design, and discussion that has been done before, it can be produced a flooding attack monitoring application using the k-means algorithm with clustering techniques. Application testing is carried out by simulating attacks using the flooding attack application on the LAN network by attacking the target PC paired with Snort for web-based monitoring to see the types of attacks that occur and categorizing these attacks into the high, medium, or low categories using the K-Means clustering techniques. Testing the average success rate based on the attack scenario using the tools in this study was 76.095% for the success rate of packets captured by a snort. With a monitoring application using snort makes it easier to speed up monitoring and attack handling.

REFERENCES

Alfiansyah, B. (2018). *Pengelompokan Notifikasi Alert Intrusion Detection System Snort Pada Bot Telegram Menggunakan Algoritma K-Means*. University of Muhammadiyah Malang.

Anand Sukumar, J. V., Pranav, I., Neetish, M. M., & Narayanan, J. (2018). Network Intrusion Detection Using Improved Genetic k-means Algorithm. *2018 International Conference on Advances in Computing, Communications, and Informatics, ICACCI 2018*, 2441–2446. https://doi.org/10.1109/ICACCI.2018.8554710

Ananta, A. Y. P. N. M. (2017). Seleksi Notifikasi Serangan Berbasis IDS Snort. *SMARTICS Journal*, *3*(2), 31–38.

Christine, E. J., Hadi, M. Z. S., & Kusumaningtyas, E. M. (2011). Aplikasi hierarchical clustering pada intrusion detection system berbasis snort. *ITS*.

Effendy, D. A., Kusrini, K., & Sudarmawan, S. (2018). Classification of the intrusion detection system (IDS) based on the computer network. *Proceedings - 2017 2nd International Conferences on Information Technology, Information Systems, and Electrical Engineering, ICITISEE 2017*, *2018–January*, 90–94. https://doi.org/10.1109/ICITISEE.2017.8285566

Elsa Kusuma, Jefri, H. A. (2019). Aplikasi Perhitungan Dan Visualisasi Jarak Terpendek Berdasarkan Data Coordinate Dengan Algoritma Dijkstra Dalam Kasus Pengantaran Barang Di Kawasan Jabodetabek. *Jurnal SISFOKOM*, *8*(1).

Gondohanindijo, J. (2011). *Sistem Untuk Mendeteksi Adanya Penyusup (IDS : Intrusion Detection System). 2*, 46–54.

Heryanto, A., Stiawan, D., & N. (2016). Visualisasi Serangan Denial Of Service Dengan Clustering Menggunakan K-Means Algorithm. *ANNUAL RESEARCH SEMINAR 2(1)*, 348–354.

Israelsson, P. (2005). *A quick overview of Snort*.

Juwita, S. (2013). Analysis Explotasi Keamanan Web Denial Of Service Attack. *ComTech Computer Science Department, School of Computer Science, Binus University*, *4*(2), 1199–1205.

Maliki, I. (2016). Penilaian Tingkat Kematangan Manajemen Kualitas Informasi dengan Metode Caldea dan Evamecal,. *Jurnal Imliah UNIKOM*, *8*(2).

Qiao, Y., & Yu, R. (2016). Software-Defined Networking (SDN) and Distributed Denial of Services (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges. *On IEEE Communications Survey & Tutorials, Vol. 18*.

R. I. Perwira, Y. Fauziah, I. P. R. Mahendra, D. B. P. and O. S. S. (2019). Anomaly-based Intrusion Detection and Prevention Using Adaptive Boosting in Software-defined Network. *5th International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia*, 188–192.

Singh, A., Rana, A., & Pradesh, U. (2013). K-means with Three different Distance Metrics, 67(10), 13–17. *International Journal of Computer Applications*, *67*(10).

Stiawan., D. (2009). *Network Development Life Cycle," Fundamental Internetworking Development & Design Life Cycle*.

Suyanto, A. H. (2004). *PENGENALAN JARINGAN KOMPUTER*.

Tanenbaum, A., S., D. J. W. (2013). *Computers Network* (5th ed.). Pearson Education India.

Xiaofeng, Z., & Xiaohong, H. (2018). Research on intrusion detection based on an improved combination of K-means and multi-level SVM. *International Conference on Communication Technology Proceedings, ICCT*, *2017–Octob*er 2042–2045. https://doi.org/10.1109/ICCT.2017.8359987

Yang, W. (2017). *Efficient K-means Algorithm in Intrusion Detection*. *132*(Msam), 193–195. https://doi.org/10.2991/msam-17.2017.43