AN ALTERNATIVE SOLUTION TO HANDLE DDOS ATTACKS

[1]AHMAD SANMORINO, [2]RENDRA GUSTRIANSYAH

Faculty of Computer Science, Universitas Indo Global Mandiri, Palembang, Indonesia

E-mail: [1]sanmorino@uigm.ac.id, [2]rendra@uigm.ac.id

ABSTRACT

Through this article, we try to offer alternative solutions to handle DDoS attacks. The discussion begins with a study of literature on DDoS attacks that include the mechanism of DDoS attack detection proposed by several researchers. The discussion then proceeded to propose alternative solutions to deal with DDoS attacks. The proposed alternative solution continues the research we have done before, using the SOM algorithm with added packet per flow feature for classifying incoming packet data. By doing the classification, can distinguish a normal data packet and abnormal one. Through the discussion in this article, we expected to contribute in the world of research, especially those related to system security issues.

Keywords: *DDoS, Packet, Flow, Classification.*

1. INTRODUCTION

Discussions about DDoS attacks are still a hot topic of research until we write this article. Evidenced by the many researchers who conduct research on DDoS attacks. Starting from the mechanism, detection, to handling DDoS attacks with various methods and point of view. Among them are the use of matching pursuit algorithm [1], Software Defined Network (SDN) architecture [2], Machine Learning [3], and detection method in cloud computing environment [4]-[6]. This article begins with a literature study of DDoS attacks From several studies. Then do with an alternative solution to deal with DDoS attacks. The proposed alternative solution is continuing our earlier research [7], using the SOM algorithm with more packet per flow feature for classifying incoming packet data. Our alternative solutions to contribute in the world of research, especially related to the problem of computer network security.

2. LITERATURE REVIEW

Before we offer alternative solutions to handle DDoS attacks, we will discuss several related methods to this study. First, discuss active DDoS attack filtering methods based on IP Flow. This method of DDoS attack detection was proposed by Feng et al. [8] using five IP Flow statistics features. These five features consist of four features of Micro-Flow and one feature of Macro-Flow.

Micro-Flow is a package that is part of a group of packets that have the same characteristics and intervals. The statistical features used by researchers derived from a micro-flow feature, the first feature, the average number of packets in per flow. The continuous and random packet legitimate delivery process used to attack, so the process of sending micro-flow done quickly. Normally, a flow consists of 1 ~ 3 packets [9]. The average number of packets in per flow on traffic when DDoS attacks approach 1 (110i ~ 180i), much different from the average number of packets in per flow in normal traffic. The second feature, the percentage of correlative flow. During the attack, the target host still has the ability to reply to requests from packets sent by zombie computers. But the reply packet sent will never reach the zombie computer. This happens because the IP address used is a fake zombie computer. If flow A comes from SourceIP1 = X to DestIP1 = Y, and flow B is from SourceIP2 = Y to DestIP2 = X, then flow A and flow B considered as a correlative flow pair. The third feature, one direction generating speed. Flow generating speed becomes faster when an attack occurs or in other words, the server becomes very busy.

The fourth feature, ports generating speed. Some researchers [10] chose the port size as a feature that used to detect DDoS attacks, but there are many services and applications (such as the popular P2P app BT, which uses port numbers over 1024, so the proposed method can not use again. Ports generating speed under normal traffic conditions is not more than 200, but when DDoS attacks happen ports generating speed value is more than 6000.

Next, macro-flow feature. As known macro-flow is all packets sent at the same time interval. The macro-flow feature that used is the percentage of abnormal packets. In order for DDoS attacks more effective, hackers usually insert tools on packets sent. This causes the packet size abnormal. For example, there are some TCP packets whose normal size is only about 40 bytes, and UDP packets have a normal size about 28 bytes. The percentage of abnormal packets shows the character of this DDoS attack by calculating the percentage of packets that have a size exceeding normal. There is a significant change of percentage of abnormal packets from 0 (normal condition) to 0.9 when DDoS attacks occur.

According to the author's opinion, the five features of DDoS attack detection proposed by the researcher is good enough. Inspection of every incoming packet done thoroughly and detail. But the question is how about the time needed to detect DDoS attacks? Whether the fixed time efficiency maintained so that handling finish immediately, before the server down.

DDoS attack detection research is also performed by Braga et al. [11], using the traffic flow feature added to the openflow controller ie NOX. This is probably done because NOX platform provides a programmatic interface. The proposed method is named Lightweight DDoS Flooding Attack Detection. The method is divided into three modules that are inserted into the detection loop on the NOX controller. There are three modules that are used to perform loop detection. First, the Flow Collection Module is periodically responsible for all flow entries requested from the flow-table of the connected openflow switches. When receiving a packet ask, Flow Collection will offer a response in the form of an answer that is transmitted via the secure channel. Second, Feature Extractor Module performs feature extraction from received flow. Includes a field header from flow-entries in the flow collection.

Furthermore, the extraction results of this feature used to detect DDoS attacks. Third, Module Classifier, a module that serves to analyze the extraction features of flow-entries obtained from the Feature Extractor. Classify whether packets originating from legitimate traffic or derived from DDoS attacks.

The use of a controller as a detection device can lead to overhead. In the worst case scenario, this method can cause a very high overhead. This happens because of DDoS flooding attacks that can send packets with a very large volume.

The next discussion, Lee et al. [12] using the clustering extraction feature of multiple traffic variables. Clustering becomes to stages group in packet delivery. From this grouping will be known when DDoS attacks occur. Then traced back which traffic variable used in that stage. But first, must know the characteristics of DDoS attacks. The steps of the proposed approach are as follows:

2.1 Select the Detection Parameters

Select the detection parameters to be used. The parameters used by the researcher are IP and port number of packet delivery source, IP and destination packet delivery port number, packet type received, and the number of packets received.

2.2 Cluster Analysis

The purpose of cluster analysis is to separate the normal phase traffic and phase when a DDoS attack occurs. In clustering, the researcher uses Euclidean distance [13] to calculate the distance between x and y. Where x and y are the values of the two variables to be calculated. In the next iteration, the value of x serves as the value of the center of the cluster (centroid), while the value of y is the value of the variable to be calculated the distance to the center of the cluster.

Cluster analysis method is considered less precise in conducting DDoS detection process. The obstacle, in this case, is the matter of time complexity. Because as a clustering method in general, it requires iteration several times to create a stable cluster. So the time required to perform DDoS attack detection becomes twice as long.

Furthermore, a multi-core DDoS attack detection method was proposed by Wang et al. [14]. This is the result of modification of DDoS-based IP Flow attack detection methods previously proposed [15]. The illustrate DDoS Detection method based on IP Flow or better known as IP Flow based DDoS Detection System seen in the picture below.

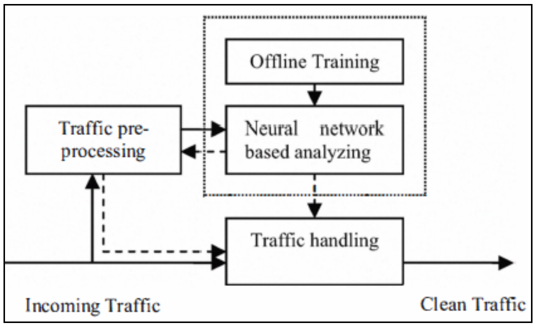


Figure 1: IP Flow based DDoS Detection System Method

The pre-processing Traffic model produces two types of data, they are features and characteristics of the ongoing attack. The attack feature data is calculated at certain times (depending on the network environment and unit processing capabilities used). If an attack is detected, the neural network analyzer will notify the pre-processing model by sending an interrupt. The pre-processing model will then send the ongoing attack characteristics to the traffic handling model. Traffic handling models are not part of IP Flow based DDoS Detection System but can be firewalls that will be enabled by the model analyzer when a DDoS attack occurs. Just like a firewall, the function of traffic handling model is to do filtering. So the resulting output is in the form of clean traffic. The modifications made by researchers is to add the function of the division of labor of each model on the IP Flow based DDoS Detection System method. Each part is done by each core contained on the CPU. The purpose of this modification is to speed up the process of attack detection, but still, maintain the accuracy of the results of detection.

In the Multi IP Flow based DDoS Detection System Method, the preprocessing model and the neural analyzer model performed by different processor cores. In other words, these two models will run in parallel with each other and can exchange information if needed. So the time needed to detect DDoS attacks can be faster, given the number of packets received when the attacks are made can reach hundreds or even thousands of packets per second.

For traffic handling the model performs the same function as the IP Flow based DDoS Detection System method. The challenge for researchers here is how to divide IP Flow based DDoS Detection System into sections without ignoring the dependency of each data. Another problem is how to keep the work balance between each core used.

3. MATERIAL AND METHOD

After studying the various methods that have been presented in the previous section, we try to offer alternative solutions to detect DDoS attacks. Research starts from collecting data. The process of recording data using the nfdump toolkit [16].

Nfdump can display netflow data such as IP address, number of packets, number of statistics flow. While nfcapd works by reading the netflow data and storing it into a file, which is then read by nfdump. For each netflow stream one nfcapd process is required.

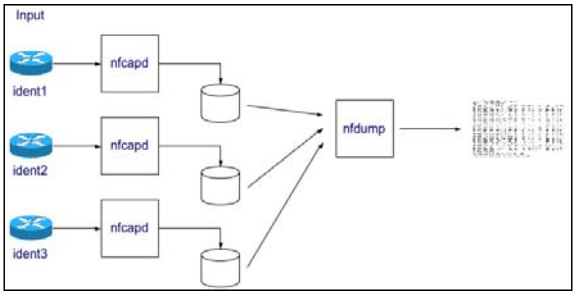


Figure 2: Nfdump and Nfcapd

The installation process and configuration of network traffic using nfdump is described in the following sections.

3.1 Nfdump Installation

Here is the steps nfdump installation on linux operating system. To be able to install nfdump requires additional packages, by typing the following commands:

```
sudo apt-get install flex

sudo apt-get install rrdtool librrd2 librrd2-dev

sudo apt-get install perl-byacc
```

To facilitate the installation move to the home directory, using the command cd~/. Then download the nfdump tool using the command:

```
wget
http://garr.dl.sourceforge.net/sourceforge/nfdump/nfdump-1.5.7.tar.gz
```

Use the following command to extract the downloaded file.

```
tar -xvf ./nfdump-1.5.7.tar.gz
```

Then followed by the command,

```
cd  nfdump-1.5.7
```

Prepare all settings before installing by typing command:

```
./configure --enable-nfprofile --enable-sflow
```

Next do the nfdump installation using the command:

```
make

make install
```

Nfdump is also available in the Ubuntu repository, so it can be instantly installed.

3.2 Nfdump Configuration

Configuration is performed on the router that will be used to capture the netflow data. Here is one example of nfdump configuration steps.

```
interface fastethernet 0/0

ip route-cache flow
```

To tell the router where netflow data will be sent, use the command:

```
ip flow-export

ip flow-export version 5

ip flow-cache timeout active 5
```

Next use the following command.

```
mls flow ip interface-full

mls flow ipv6 interface-full

mls nde sender version 5
```

Actually more to the version of the router used, usually, each type of router can have different configurations. For routers that serve multiple routes, can use commands,

```
mls aging fast time 4 threshold 2

mls aging normal 32

mls aging long 900
```

Record traffic data contains information about incoming packet delivery. Here is the example of packet data successfully recorded:

Table 1: Record Packet Data

tm(s)	Ip Address	Flow	Packet	Bytes
927	xxx.xxx.xxx.94	81304	204415	9829622
1072	xxx.xxx.xxx.62	82749	187368	8966992
908	xxx.xxx.xxx.118	209413	558894	5368820
926	xxx.xxx.xxx.218	96857	194889	9661869
1272	xxx.xxx.xxx.109	73758	194190	9147116
1149	xxx.xxx.xxx.137	212215	597375	8070826
1243	xxx.xxx.xxx.89	212004	584858	6226588
1240	xxx.xxx.xxx.79	88297	182591	5128862
1223	xxx.xxx.xxx.87	4953	13146	6599621
1285	xxx.xxx.xxx.108	4526	11186	6497480
1388	xxx.xxx.xxx.82	5011	10480	8753579
1075	xxx.xxx.xxx.236	76185	182344	6283037
1011	xxx.xxx.xxx.101	5174	13774	6372816
1273	xxx.xxx.xxx.22	4966	12479	6867204
1469	xxx.xxx.xxx.68	97491	191062	8142371

After getting the record packet data, the next step to classify the recording packets that coming into the network. Package classification is done using various modern algorithms that exist today. The current algorithm with the support of hardware progress ensures efficiency in terms of the time and memory complexity required for classification. Classification algorithm used for various types of problems and has proven its relevance. As we have mentioned in our paper which discusses various classification algorithms for handwritten signature verification [17]. Another alternative is the use of fuzzy algorithms, as some researchers have done to solve various problems [18]. This algorithm is widely used because of its flexibility and accuracy. But for this research, we will only focus on the classification algorithm alone. One of the classification algorithms that used for packet classification is Self Organizing Map [19]. The steps of packet classification using SOM algorithm are:

a. Initialize weight matrix in each record traffic.
b. Repeat the initialization process to get the value of each weight matrix
c. Find the nearest traffic record of the new sample by calculating the distance between the new sample of all traffic records or the data center using the distance formula.
d. Update weight matrix of the data center or record of nearest traffic.
e. Find neighbor or data center traffic record based on the threshold of distance calculation.
f. Update weight matrix of each traffic record or data center identified as the neighbor.
g. Until threshold or weight matrix of each record of traffic or data center is stable.
h. Assign each sample data to the nearest traffic record.

4. RESULTS AND DISCUSSION

The proposed form alternative is implemented in the form of an easy-to-use graphical user interface. Selection of GUI based on the general application used today. Many examples of the use of graphical display or GUI to implement a method, as has been done by some researchers [20][21]. After the packet classification finish, then the mechanism that performed against the packet detected as DDoS package is to drop the packet or limiting packet. By using the drop packet mechanism, all packets detected as DDoS attacks will be dropped without exception to prevent the server from downing, because it can not accommodate the overflow packet number. The quota-limiting mechanism works by limiting the number of quota packets allowed to log in or getting service from the server. However, this handling mechanism only applies to packets that exceed the normal (high-rate DDoS) threshold. To make sure the accuracy level remains relevant, we also classify the packet and flows separately. From the results of the monitoring of packet and flow is compared with the results of the classification of packet per flow. So that the final classification results are relevant to the accuracy level as expected.

In the data record 300, the monitoring results show the traffic packet in normal circumstances. No abnormal traffic activity yet. Monitoring continued by increasing the number of records data in stages. This is done so that can still observe the changes that occur. So if an anomaly occurs can be immediately known. We do the same for data records 450, 600, 900, 1050 and 1209 (*Figure 7 - Figure 12*). Then proceed with the classification of packets for each record data systematically, ranging from the smallest to the largest.

Before performing data record classification with packet per flow feature, we first perform the classification of packet data records. Classification is done on the same machine resource. Classification is used to distinguish between legitimate packet and abnormal packet. Classification consists of 3 classes, normal packet per flow, high-rate packet per flow and low-rate packet per flow. We display the results of classification of data records in the form of the line chart in several data record groups:

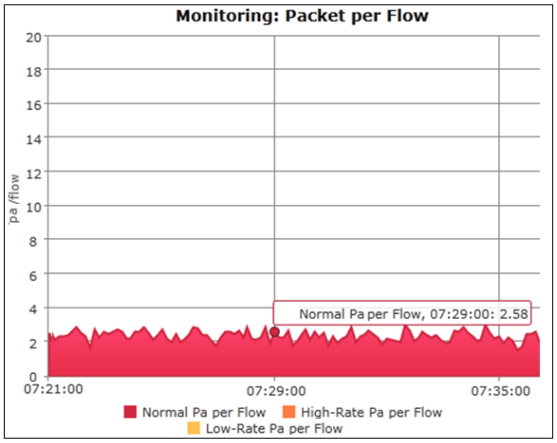


Figure 3 : Classification of 300 records of data

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

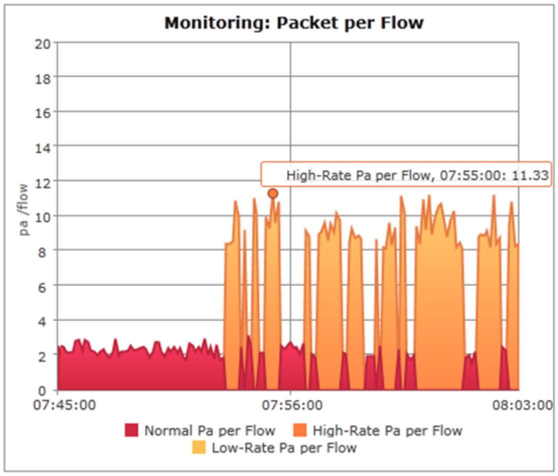


Figure 4 : Classification of 450 records of data

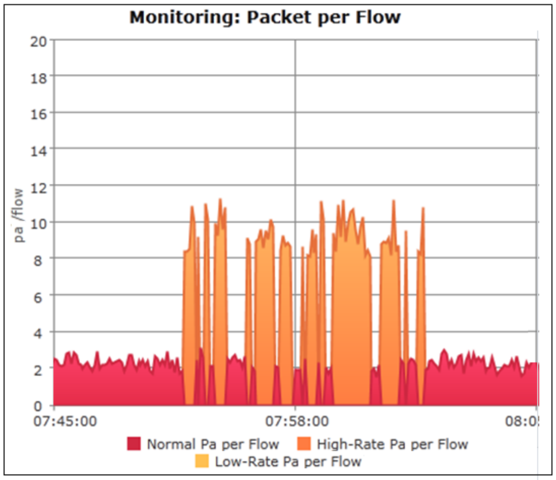


Figure 5 : Classification of 600 records of data

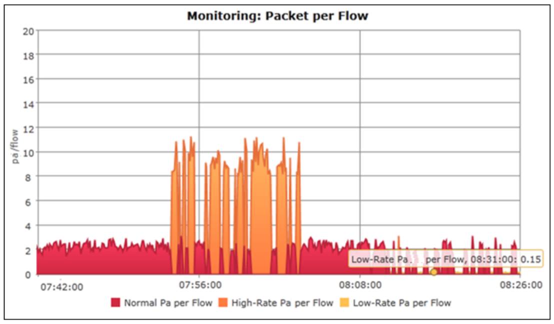


Figure 6 : Classification of 750 records of data

The accuracy rate of packet data classification result:

Table 2: Accuracy(%)

tm (s)	Record Packet Data	Accuracy Rate (%)	with Packet per Flow (%)
190	300	98.21	98.56
234	450	96.34	97.33
410	600	92.20	98.12
490	750	85.81	95.78
550	900	78.45	96.10
720	1050	72.89	95.06
880	1209	69.12	94.90

Accuracy rate obtained from the data packet classification using SOM algorithm. Data packets are classified into 3 classes of normal packets, high-rate packets, and low-rate packets. The accuracy of classification results is greatly influenced by the number of packet data records, the greater the population the lower the accuracy obtained. Not only is the accuracy, time and memory complexity required to perform greater classification. This seen from the classification shown in the table. To maintain accuracy as the number of incoming packets increases, researchers use flow traffic as a population divider. So initialize the weight matrix in each record traffic using the formula:

$$packet\ per\ flow = \sum_{w=1}^{JmlFlow} \left(JmlPacket_{w}^{\square}\right) / JmlFlow \quad (1)$$

Logically with the addition of this flow traffic feature, the classification of incoming packet records is not done at once, but per flow in a certain interval of time. This reduces the workload of SOM algorithms and resources in classification. Another impact of time complexity will decrease and memory usage becomes more efficient. When viewed from the classification results, the accuracy of the incoming data packet is relatively stable, in other words, the increase in the number of record data does not affect the accuracy level of incoming packet classification. In the record packet data (RPD) of 300, the accuracy rate (AR) reached 98% for classification with or without using the packet per flow (PPF) feature. For RPD = 450, there is also no significant difference, accuracy rate in the range of 96-97%. However, when RPD> 450, 600-900 there is a significant difference in AR. As in

RPD = 900, the classification without PPF only reaches 78%, whereas by using PPF, AR reaches 96% with the same duration of time.

Probably the accuracy is still decreased but not very significant like before added packet per flow feature. A more in-depth study is needed to measure the extent of the effect of adding packet-per-flow features in maintaining the accuracy of packet data record classification. For the duration of classification, time is greatly influenced by a large number of record data and heat generated by the processor.

5. CONCLUSION

An alternative solution for dealing with DDoS attacks that we propose probably a good option for the web server administrator. The use of SOM algorithm with the addition of packet per flow feature can be the answer to various weaknesses owned by other methods. This is clear from the accuracy of the results of the classification of incoming packet records. By looking at the results of classification, the administrator can distinguish between legitimate packets and packets coming from the attacker. Continuous testing and evaluation are needed before this DDoS attack handling method implemented.

REFERENCES:

[1] D. Erhan, E. Anarım, G.K. Kurt, “DDoS attack detection using matching pursuit algorithm,” 2016 24th Signal Processing and Communication Application Conference (SIU), 2016.

[2] L. Barki, A. Shidling, N. Meti, D.G. Narayan, M.M. Mulla, “Detection of distributed denial of service attacks in software defined networks, 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI),”, 2016.

[3] M.S. Hoyos, G.A. Isaza , J.I. Vélez, O.L. Castillo, “Distributed Denial of Service (DDoS) Attacks Detection Using Machine Learning Prototype,” In: Omatu S. et al. (eds) Distributed Computing and Artificial Intelligence, 13th International Conference. Advances in Intelligent Systems and Computing, Springer, vol 474, 2016.

[4] R. Kumar, S.P. Lal, A. Sharma, “Detecting Denial of Service Attacks in the Cloud, Dependable, Autonomic and Secure Computing,” 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, IEEE, 2016.

[5] B.B. Gupta, O.P. Badve, “Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment,” Neural Comput Appl, Springer, pp. 1-28, 2016.

[6] K. Borisenko, A. Smirnov, E. Novikova, A. Shorov, “DDoS Attacks Detection in Cloud Computing Using Data Mining Techniques,” Industrial Conference on Data Mining ICDM 2016: Advances in Data Mining. Applications and Theoretical Aspects pp 197-211, 2016

[7] A. Sanmorino, S. Yazid, "DDoS Attack detection method and mitigation using pattern of the flow,” IEEE International Conference of Information and Communication Technology, 2013

[8] Y. Feng, R. Guo, D.Wang, and B. Zhang, “Research on the Active DDoS Filtering Algorithm Based on IP Flow,” in 2009 Fifth International Conference on Natural Computation, IEEE, pp. 628–632, 2009.

[9] C. Jin, H. Wang, and K. G. Shin, “Hop-count filtering: An effective defense against spoofed DDoS traffic,” Pro-ceedings of the 10th ACM Conference on Computer and Communication Security, ACM Press, pp. 30–41, October, 2003.

[10] V. A. Siris and F. Papagalou , “Application of anomaly detection algorithms for g SYN flooding attacks In: Regency H, ed,” Global Telecommunications Conf. (GLOBECOM’04). Dallas: IEEE, pp. 2050–2054, 2004.

[11] R. Braga, E. Mota, A. Passito, “Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow,” 35th Annual IEEE Conference on Local Computer Networks, pp. 408-415, 2010.

[12] K. Lee, J. Kim, K.H. Kwon, Y. Han, S. Kim, “DDoS attack detection method using cluster analysis,” ELSEVIER Expert Systems with Applications, vol 34, pp. 1659-1665, 2008.

[13] M.M. Deza and E. Deza, “Encyclopedia of Distances,” SPRINGER-VERLAG Berlin Heidelberg, 2009.

[14] D. Wang, Z. Yufu and J. Jie, “A Multi-core Based DDoS Detection Method,” 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), pp. 115-118, 2010.

[15] D. Wang, G. Chang, X. Feng, R. Guo, "Research on the detection of distributed denial of service attacks based on the characteristics of IP flow," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 5245 LNCS, pp. 86-93, 2008.

[16] “NFDUMP version: 1.6.9,” 2013. [Online]. Available: http://nfdump. sourceforge.net/

[17] A. Sanmorino, S. Yazid, "A survey for handwritten signature verification," 2012 2nd International Conference on Uncertainty Reasoning and Knowledge Engineering (URKE), Jakarta, Indonesia, pp. 54-57, 2012.

[18] R. Gustriansyah, D.I. Sensuse, A. Ramadhan, “A sales prediction model adopted the recency-frequency-monetary concept,” Indonesian

Journal of Electrical Engineering and Computer Science, Vol.6, No.3, 2017.

[19] T. Kohonen, “Self-Organizing Maps,” Series in Information Sciences, Vol. 30. Springer, Heidelberg. Second ed. 1997.

[20] A. Sanmorino, “Development of computer assisted instruction (CAI) for compiler model: The simulation of stack on code generation,” International Conference on Green and Ubiquitous Technology (GUT), Jakarta, pp. 121-123, July 2012.

[21] A. Sanmorino, Isabella, “The design a system of retention and control on broiler farms based on the flow of data,” 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, September 2017.

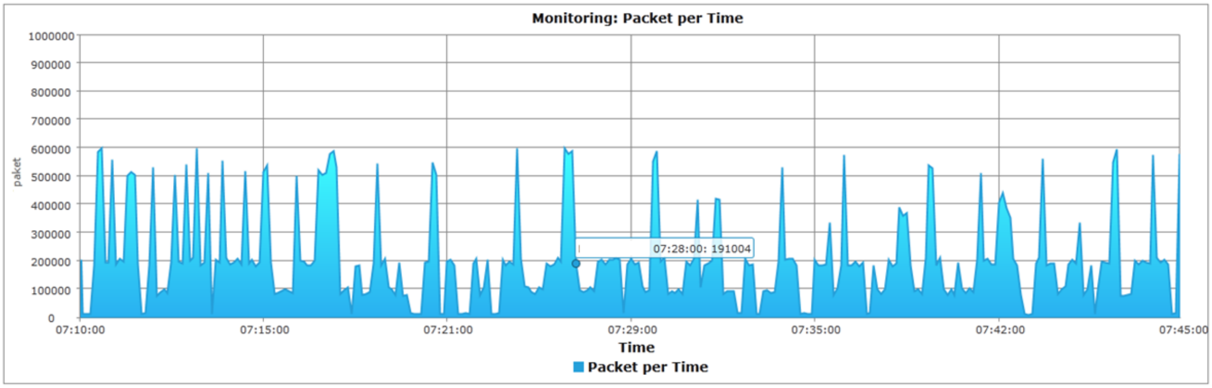


Figure 7 : Monitoring Packet per Time (300 records data)

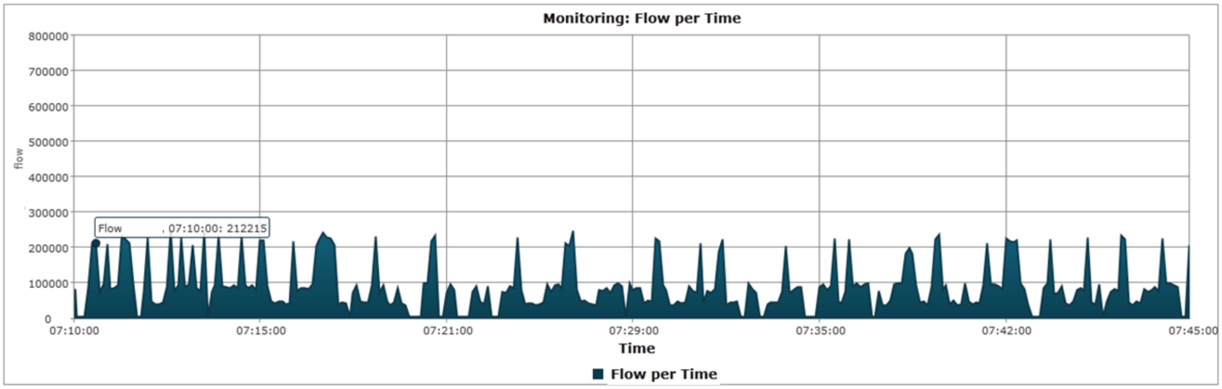


Figure 8 : Monitoring Flow per Time (300 records data)

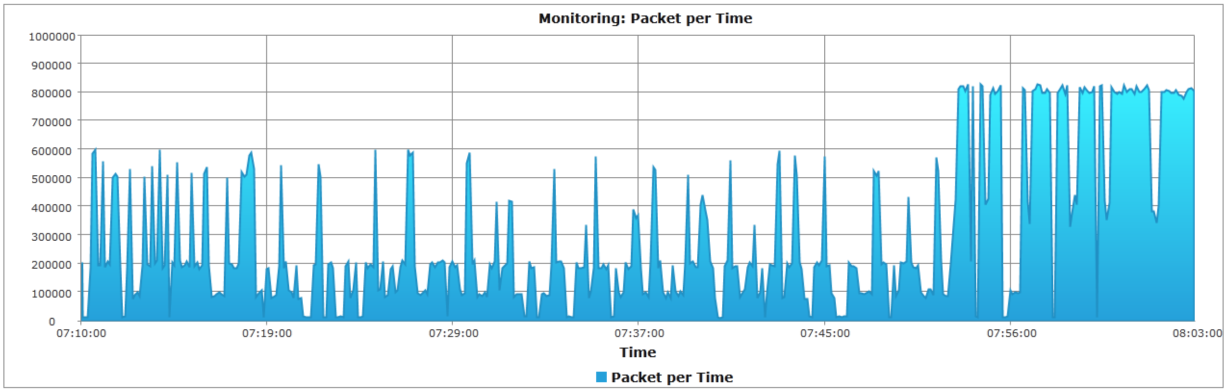


Figure 9 : Monitoring Packet per Time (450 records data)

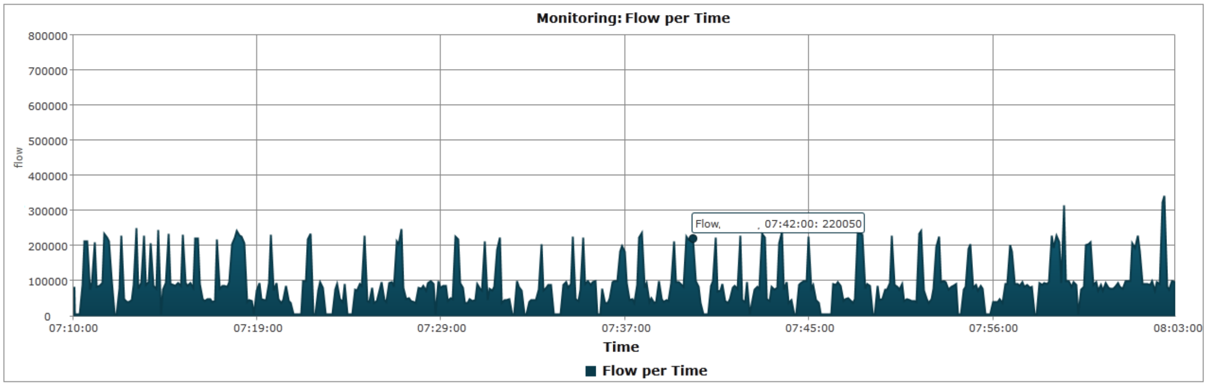


Figure 10 : Monitoring Flow per Time (450 records data)

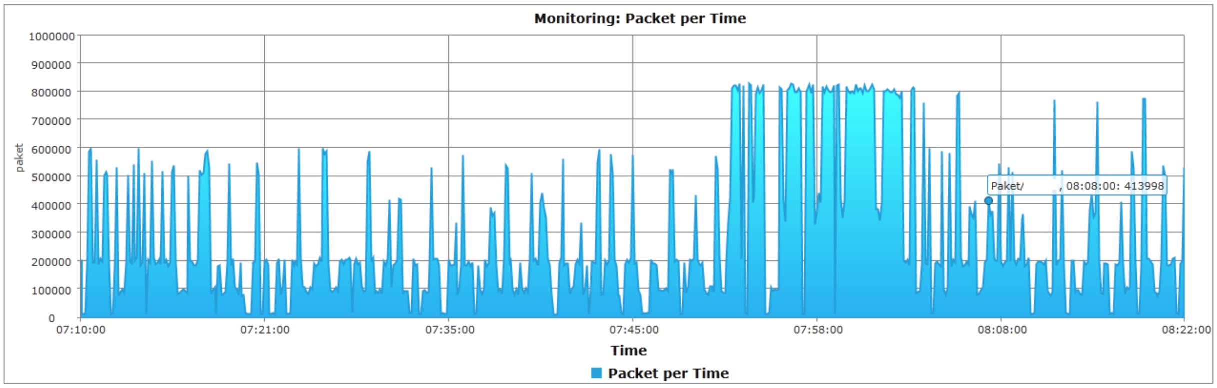


Figure 11 : Monitoring Packet per Time (600 records data)

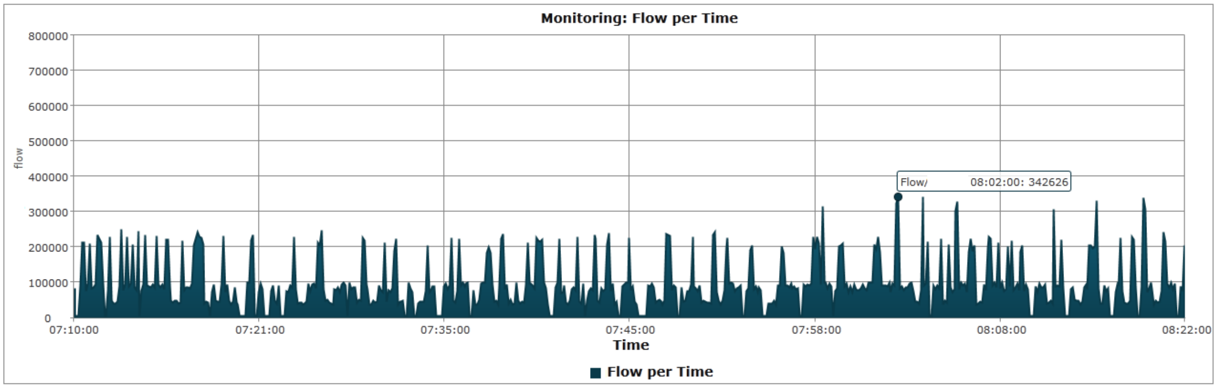
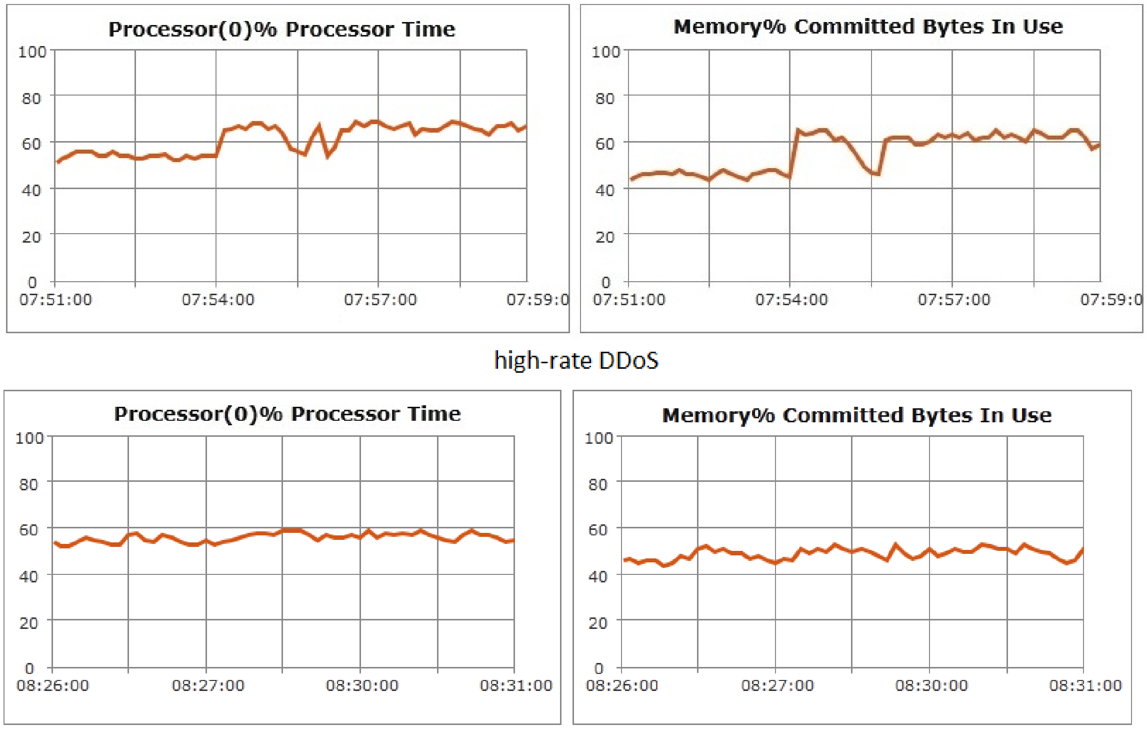


Figure 12 : Monitoring Flow per Time (600 records data)



high-rate DDoS

low-rate DDoS

Figure 13 : The usage of CPU and memory

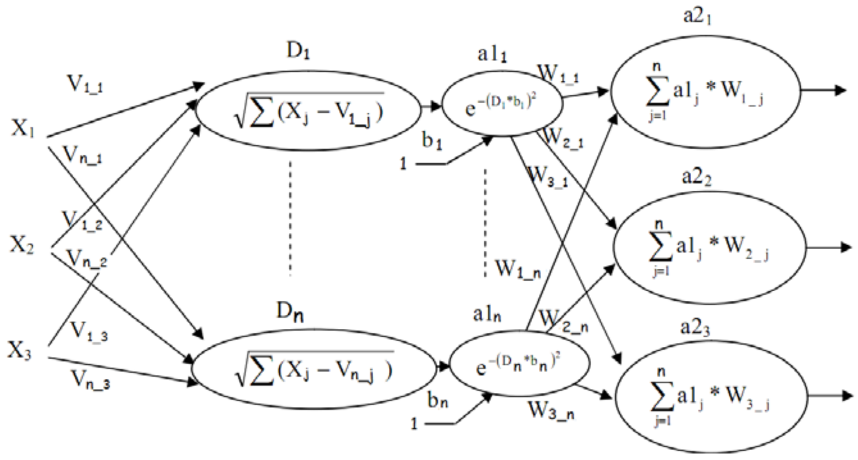
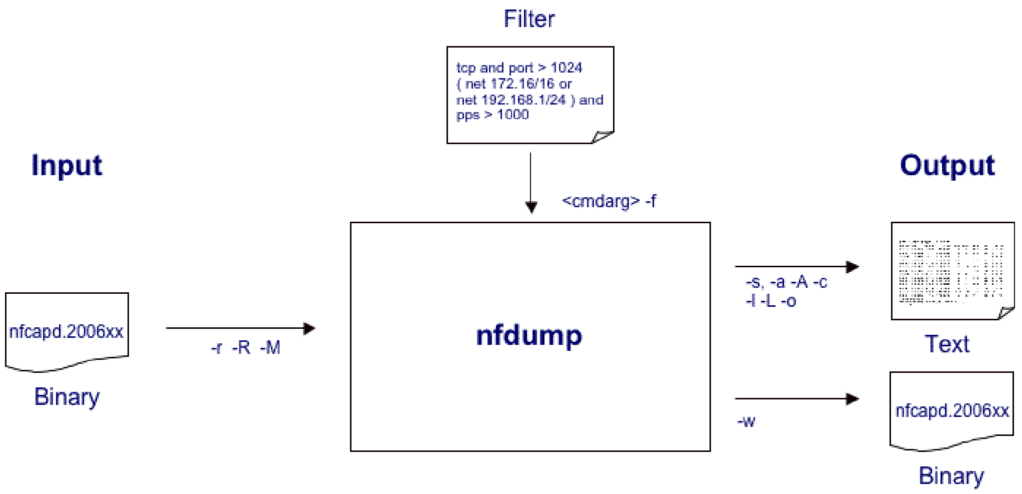


Figure 14 : The distances formula network used on proposed method



Source: http://nfdump.sourceforge.net/

Figure 15 :Nfdump Mechanism (Input, Process, and Output)