DDOS ATTACK DETECTION SIMULATION AND HANDLING MECHANISM

Ahmad Sanmorino[1,2] and Setiadi Yazid[1]

[1]Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia
[2]Faculty of Computer Science, Universitas Sriwijaya, Indralaya, Indonesia

Email: ahmad.sanmorino@ui.ac.id

Abstract

In this study we discuss how to handle DDoS attack that coming from the attacker by using detection method and handling mechanism. Detection perform by comparing number of packets and number of flow. Whereas handling mechanism perform by limiting or drop the packets that detected as a DDoS attack. The study begins with simulation on real network, which aims to get the real traffic data. Then, dump traffic data obtained from the simulation used for detection method on our prototype system called DASHM (DDoS Attack Simulation and Handling Mechanism). From the result of experiment that has been conducted, the proposed method successfully detect DDoS attack and handle the incoming packet sent by attacker.

Keywords: *DDoS, simulation, packet, flow, handling mechanism*

Abstrak

Dalam paper ini dibahas bagaimana menangani serangan DDoS yang datang dari penyerang dengan menggunakan metode deteksi dan mekanisme penanganan tertentu. Deteksi dilakukan dengan membandingkan jumlah paket dan jumlah aliran. Sedangkan mekanisme penanganan dilakukan dengan membatasi atau menjatuhkan paket yang terdeteksi sebagai serangan DDoS. Penelitian diawali dengan simulasi pada jaringan yang nyata, yang bertujuan untuk mendapatkan data lalu lintas real. Kemudian, *dump* data lalu lintas yang diperoleh dari simulasi yang digunakan untuk metode deteksi pada sistem prototipe kami disebut DASHM (*DDoS Attack Simulation and Handling Mechanism*). Dari hasil percobaan yang telah dilakukan, metode yang diusulkan berhasil mendeteksi serangan DDoS dan menangani paket masuk yang dikirim oleh penyerang.

Kata kunci: *DDoS, simulasi, paket, flow, mekanisme penanganan*

1. Introduction

DDoS attack is widely used because it is considered the most effective way to cripple a server or network. Until now still very difficult to detect at an early stage of attack. DDoS attacks are usually only discovered when a server or network exhaustion and down for a while. DDoS attack detection due to difficulties in distinguishing between legitimate packets on normal traffic and useless packets originating from DDoS agents. Widely use of DDoS because of the ease of doing attacks. There are many tools that can be used for an attack, such as stacheldraht [1]. In this study, we discuss the mechanism of DDoS attacks and how to handle it. The proposed method of treatment is to perform a DDoS attack detection of incoming packets. Whereas handling mechanism perform by limiting or drop the packets that detected as a DDoS attack. From the result of experiment that has been conducted, the proposed method successfully detect DDoS attack and handle the incoming packet from attacker.

2. Related Works

Research related to DDoS detection and mitigation have long done. Among of DDoS attack detection method was proposed by Feng et al. [2] using statistical features of IP Flow. These features are composed by four features of Micro-Flow and one feature of Macro-Flow. Micro-Flow is a package that is part of a group of packages that have the same characteristics and intervals. While Macro-Flow is the whole package is sent at the same time interval. The use of statistical features

of IP Flow is considered good, but it should be signed from the calculation of the five IP Flow features that some kind of flow pattern that comes in, so that when the next attack happens, do not bother to do the calculations again. Other studies conducted by Wang et al. [3][4], by using multi-core CPU. In this method, pre-processing method and neural model of model as analyzer will be done by different processor cores. In other words, both of these models will run in parallel with each other, with exchange information. So that the time required to detect DDoS attacks can be more quickly. The challenge for researchers is how this method divide detection process into parts without losing sight of data dependence. Another problem is how to maintain balance of performance of each core that are used.

3. Methodology

To know the type of incoming packets, classification algorithms needed to distinguish between normal packets or packets that sent by attacker. The algorithm used to perform the classification is self organizing map (SOM) [5]. The steps of packet classification using SOM algorithm is as follows:

1. Initialize the weight matrix in each packet
2. **repeat**
3. Select next sample
4. Find the closest packets or centroid to the new sample by calculating the euclidean distance between the new sample and all packets or centroids
5. Update the weight matrix of the closest packets or centroid
6. Find neighbor packets or centroid based on a predefined distance threshold
7. Update the weight matrix of packets or centroids which are identified as the neighbors
8. **until** The weight matrix of packet or centroid does not change or the threshold is exceed
9. Assign each sample to its closest packet or centroid

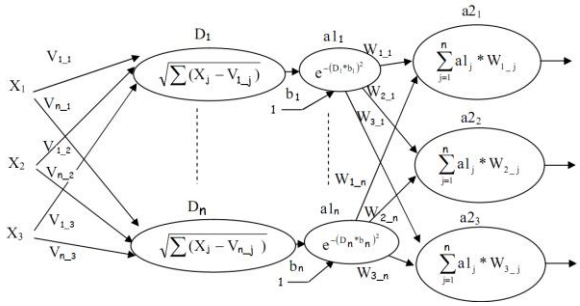
To determine initial value of centroids we used formula (1), whereas average number of packets in per low according to [2] [7] in the range of 1 ~ 3.

$$packet\ per\ flow = \sum_{k=1}^{FlowNum} (PacketsNum_{k}) / FlowNum \quad (1)$$

To calculate the distance between the centroids and each packets, we use distance formula (2) [8]:

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}, \quad (2)$$

Due to the amount of distance that we want to search from the *n* record data, and we expect the output are three classes, the calculation of the distance formulas become a network as follows:



4. Simulation

We divide this simulation become two steps, first step used to gather real traffic data from network on our campus. Second, step we do the simulation using DASHM system that implements detection method and handling mechanism. In the first step, we use 10 computers as DDoS agents and stacheldraht [1] as DDoS tool.

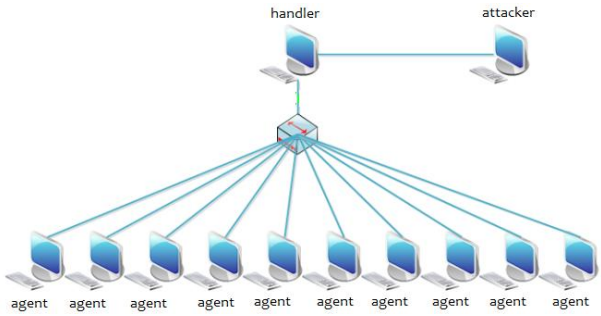


Fig 1. DDoS Network Topology

Attack carried out in three phases. The first phase was done by sending a number of packets in large amount of packets. The second phase was done by sending a number of packets with a smaller intensity than the normal package. The third phase was done by sending a number of very huge amount of packets, more than the first phase, but with a shorter duration of time. We captured network traffic information from the simulation using nfdump [9]. Nfdump record all incoming packets to the network server. The amount of data successfully recorded traffic of approximately 1200 records, which consist of normal packets and packets come from DDoS agents. Furthermore, the data that has been obtained used in the second step of the simulation. The second simulation was done

in eight phases, where each phase consists with a different testing data. DASHM systems perform packet classification using SOM algorithm with calculation number of packet per flow as attribute. The detection results of the simulation that has been done can be seen in the next section.

5. Result and Discuss

In addition to testing using the proposed detection method, we also performed comparison using the IP Flow. The performance of each method is measured by the number of true positives (TP), false positive (FP), true negative (TN), false negative (FN), false positive ratio (FPR), false negative ratio (FNR), response time detection (RTD), and CPU usage. Especial for CPU usage comparison, will be discussed separately in the end of this section. For the amount of TP and TN, the highest the better. As for the number of FP and FN the lower the better. A detection mechanism is successful or accepted if RTD < LTD. If on the contrary, the proposed method is considered failed or refused to make the detection. Testing results using IP Flow can be seen in Table I. While the testing result using method that we proposed, can be seen in Table II.

TABLE I
TESTING RESULTS USING IP FLOW

Data	TP	FP	TN	FN	FPR	FNR	RTD(s)	LTD(s)	RTD < LTD
150	0	0	150	0	0	-	0	0	-
300	0	1	299	0	0,00333333	-	0,020037	1065	accept
450	64	2	384	0	0,00518135	1	1,320578	111045	accept
600	64	3	533	0	0,00559701	1	1,340176	112442	accept
750	127	7	616	0	0,01123596	1	3,068489	184128	accept
900	145	7	748	0	0,00927152	1	3,546029	203697	accept
1050	145	10	895	0	0,01104972	1	3,604576	207334	accept
1209	176	12	1021	0	0,01161665	1	4,264388	264532	accept

With using same testing data, the results obtained by SOM algorithm as follows:

TABLE II
TESTING RESULT USING SOM AND SOM-PATTERN

TP	FP & FN	TN	FNR	SOM			SOM - Pattern		
				RTD(s)	LTD(s)	RTD < LTD	RTD(s)	LTD(s)	RTD < LTD
0	0	150	-	0	0	-	0	0	-
0	0	300	-	0	0	-	0	0	-
64	0	386	1	0,140206	109034	accept	0,138287	109034	accept
64	0	536	1	0,140206	109034	accept	0,138287	109034	accept
127	0	623	1	0,273321	176893	accept	0,254672	176893	accept
145	0	755	1	0,310788	196462	accept	0,285949	196462	accept
145	0	905	1	0,310788	196462	accept	0,285949	196462	accept
176	0	1033	1	0,38497	250827	accept	0,377112	250827	accept

It can be seen in Table I and Table II, our proposed method has a value of TP, FP, TN, FN, FPR, and RTD are better than IP Flow. Performance measurement based on false positives, giving a significant difference. IP Flow method detects two normal packets as DDoS packet in the second testing, three packets on the third testing, and continue increased at each testing up until the eighth testing. Conversely, our method could not false positive value, in each test. In other words, our proposed method successfully detects normal packet as a normal packet of 100%. As we know, false positive value, the lower the better. It can be stated that based on the measurement of false positive (FP), our proposed method better than IP Flow method.

The difference false positive value occurs because the IP Flow method using if-then rules, so that the threshold value for each class tends to be static. Tolerance limits are given for each packet to be very rigid, this causes the normal packet threshold value is inserted into the abnormal class, in other words normal packets detected as packets originating from DDoS agent. Instead, our proposed method made classification not based on if-then rules, but based on the similarity of each packet, by calculating the distance between each packet. So that the threshold value of each class, become more dynamic.

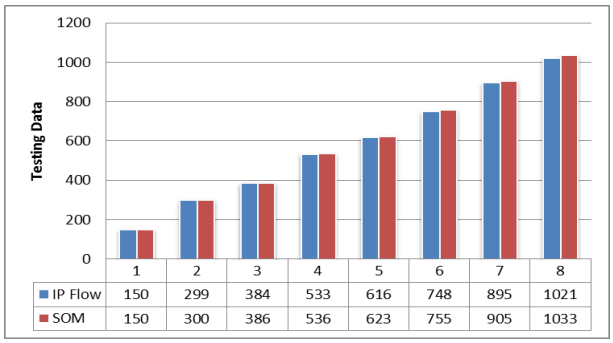


Fig 2. True Negative (TN) Comparison

For the measurement results based on true negative value can be seen in Figure 2. Based on the measurement results, the TP values obtained by the method that we proposed always better than IP Flow method, except in the first phase test. This shows that our proposed method successfully detects normal packet as normal packets more than the IP Flow method. In other words, our proposed method does not perform error on detection process, which is considered normal packets as packets originating from DDoS agent, as is done using the IP Flow.

Figure 3 shows the results of response time detection (RTD) measurement of each method that are tested. RTD was the overall time required to perform the detection of a packet. In other words, RTD is the time required to perform mathematical calculations based on the value of attributes of a

packet, to provide output that can be used as a measure, whether the packet is detected as a normal packet or vice versa. IP Flow RTD requires more time than the method that we proposed. At each testing, time required by IP Flow method are 10 to 12 times longer than our proposed method. Our proposed method especially with the addition of traffic patterns features, made the RTD value become shorter.

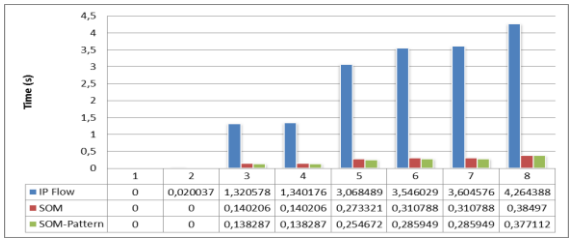


Fig 3. Response Time Detection (RTD) Comparison

IP Flow is always made calculation when to detection of the packet, so it takes much time. While our proposed method, using classification method that was done in once iteration, it can save the time required to perform the detection. More important, with the addition of traffic pattern features, the method that we proposed only need to compare the incoming packets with the packets contained in the pattern. If the pattern are matching, it can immediately be shown the results of the detection, without perform repetitive calculations.

Furthermore, we will discuss the comparison of CPU usage percentage, CPU clock speed, and costs of each method tested. The CPU clock speed and wattage in various conditions based on the percentage of CPU usage is as follows:

TABLE III
CPU CLOCK SPEED AND WATTAGE

System State(%)	Clock Speed (MHz)	Load	Notes
Idle	-	5 Watts	-
CPU at 25% load	525 MHz	13 Watts	Using Intel Processor
CPU at 50% load	1050 MHz	27 Watts	Using Intel Processor
CPU at 75% load	1575 MHz	41 Watts	Using Intel Processor
CPU at 100% load	2100 MHz	55 Watts	Using Intel Processor

In this simulation we use Intel Processor. We measure CPU usage from each method on every stage of testing. In making measurements, we make sure the conditions of CPU usage are 0% at the beginning of each test. In other words, we make sure no other processes are working simultaneously during the test process. CPU clock speed that is used on each method can be obtained using formula (3).

$$\text{Clock Speed} = \text{Percentage of Usage} \times \text{MaxClockSpeed} \quad (3)$$

Based on the amount of the wattage that required by CPU and with using total cost calculation formula, total cost for each detection method can be calculated (4).

$$\text{Total Cost} = \frac{\text{Watts} \times \text{TimeUsed}}{1000} \times \text{CostPerKilowatt} \quad (4)$$

So the comparative results obtained by each detection method can be seen in Table IV.

CPU clock speed is measured by counting the number of instructions that the processor can be completed in a certain time. If there are more calculations that have done, there are also more computation instructions will be given to the processor. IP Flow has calculation in determining the type of each incoming packet, and this is done repeatedly that make the instructions given to the processors continues to grow.

TABLE IV
CPU USAGE AND CLOCK SPEED COMPARISON

Data	CPU Usage(%)		CPU Clock Speed (MHz)		Time Used(s)		Total Cost($)	
	IP Flow	SOM	IP Flow	SOM	IP Flow	SOM	IP Flow	SOM
150	24%	17%	504	357	1,9745	0,3169	8,98E-06	1,11E-06
300	25%	20%	525	420	3,9480	0,6348	1,79E-05	2,44E-06
450	26%	21%	546	441	6,3224	0,9525	2,87E-05	3,66E-06
600	27%	22%	567	462	8,2922	1,2677	4,06E-05	5,32E-06
750	29%	22%	609	462	11,0571	1,5873	5,80E-05	6,66E-06
900	29%	23%	609	483	13,2462	1,9030	6,95E-05	8,65E-06
1050	31%	24%	651	504	15,2360	2,2433	8,53E-05	1,02E-05
1209	34%	25%	714	525	17,5462	2,6919	0,0001	1,22E-05

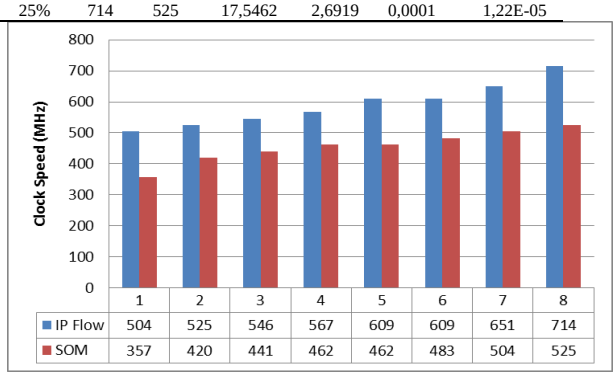


Fig 4. CPU Clock Speed Comparison

Our method do the process of distance calculating between the packet only once, packet classification decision-making based on the results of this calculation. Then, the IP Flow method does not have a centroid values that become reference by each packet, so that each packet has a comparison to all incoming packets. Conceivably, if the packet is coming in hundreds or even

thousands, there also will be more calculation given. Our method makes the centroid as a reference value for each incoming packet, so that each packet only needs to calculate the distance to the centroid value, no need to calculate the distance to each incoming packet. Next we will show the monitoring interface of our detection system. Figure 5 shows the simulation results of packet monitoring based on time of arrival.

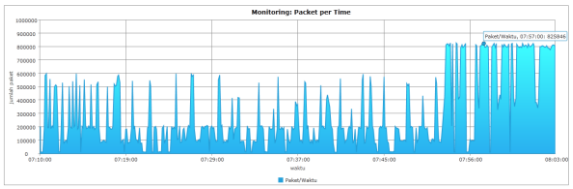


Fig 5. Packet per Time

Figure 6 shows the amount of flow based on time of arrival, number of flow below the normal threshold is detected as an abnormal packet.

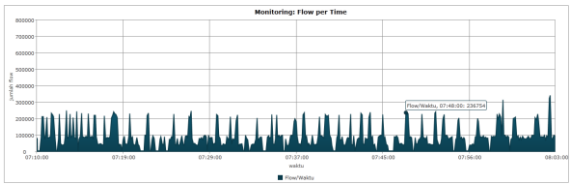


Fig 6. Flow per Time

Figure 7 shows key factors whether a packet is determined as a normal packet or packets coming from the attacker, by looking at the number of packets per flow in interval of time.

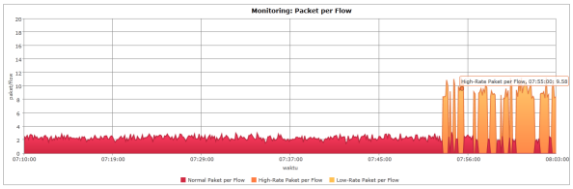


Fig 7. Packet per Flow

Furthermore, we will discuss how handling DDoS attacks. DDoS attack with a very high intensity can cause the server down. It can be seen from the CPU usage when simulating DDoS attacks carried out. In normal condition, the CPU usage on the server that becomes target of the attack in the range of 52% - 56%. When DDoS attacks happen the CPU usage increases to 65% - 69%, but this is not to cause overflow on server. Similarly to CPU usage, memory usage also increased from the range of 44% - 48% to 59% - 65% when DDoS attack happen. Figure 8 shows the CPU and memory usage transition from normal condition to the condition when DDoS attack happen.

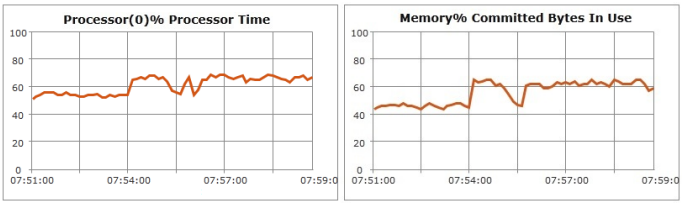


Fig 8. CPU and Memory Usage

The increase of CPU usage because of the number of packets that want to get services suddenly increase when DDoS attack happen. An increasing number of incoming packets is causing queue scheduling allocates CPU instruction more than normal conditions. This causes the CPU clock speed increases, in accordance with the instructions given by scheduling mechanism while an increasing number of memory usage because during packet waiting to get service from the CPU, the packet that has been in the queue is stored in memory. The allocation of the amount of memory is given according to the number of incoming packets. Number of incoming packets is increased when the DDoS attack, beyond the usual capacity of allocated memory in normal conditions.

6. Handling Mechanisms

As a solution to overcome the DDoS attack, we propose two handling mechanisms, which limit the amount of packets or drop all packets that are detected as DDoS attack. When implemented on a real network, the detection method and handling mechanisms that we proposed can be placed on a computer that acts as a controller or routers that connecting server to the internet. So the prevention can be done on this controller, before DDoS packet allowed to the server. The information about packet header that is detected as DDoS can still be allowed to the server. Figure 9 shows network architecture with detection method and handling mechanisms that we proposed.

Back to simulation, Figure 10 and Figure 11 shows the result of the mechanism used to handle packets that are detected as a DDoS attack. Limiting packets works by limiting the size of the throughput that can be passed by a packet sent to the server. But limiting packet only effective to handle high-rate DDoS attack. Another way to handle both types of DDoS attacks is the second mechanism.

The Second handling mechanism drops the packets out that are detected as a DDoS attack, both high-rate and low rate DDoS will be discarded.

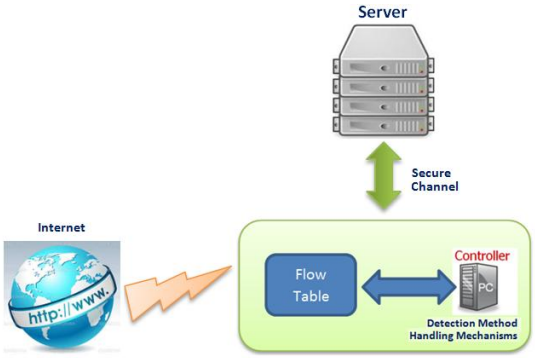


Fig 9. Secure Network Architecture

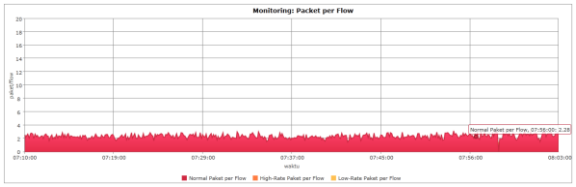


Fig 10. Packet per Flow Monitoring After Limiting Packets

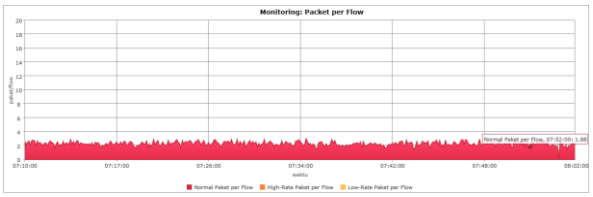


Fig 11. Packet per Flow Monitoring After Drop Packets

7. Conclusion

The conclusion that can be drawn based on the results of the experiment and performance comparison that has been done is as follows:

1. The proposed method can make the detection of DDoS attacks quick and accurate. This is evidence from the results of the testing that has been done.
2. The simulation that has been done successfully proved the proposed detection method can make the detection of distributed denial-of-service attack.
3. The use of SOM algorithm is the right choice to perform network packet classification.
4. The increase of number of packets in each test, does not affect the performance of the proposed method.

The proposed method not only successfully perform a DDoS attack detection with high accuracy, but also perform packet handling mechanism against packets that are detected as DDoS attacks. With handling mechanism the time required to handle DDoS attacks become shorter, in other words before the server down, the system has managed to take precautions against attack sent by attacker.

References

[1] "The stacheldraht ddos attack tool," 2012. [Online]. Available: http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt

[2] 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, 2009, pp. 628–632.

[3] 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.

[4] 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.

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

[6] 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.

[7] M.M. Deza and E. Deza, "Encyclopedia of Distances," Springer-Verlag, Berlin Heidelberg, 2009. DOI 10.1.1007/978-3-642-00234-2.

[8] "NFDUMP version: 1.6.9," 2013. [Online]. Available: http://nfdump.sourceforge.net/