Network Attack Traffic Detection using Seed based Sequential Grouping Model

Jee-Tae Park, Sung-Ho Lee, Young-Hoon Goo, Myung-Sup Kim
Dept. of Computer and Information Science
Korea University
Korea
{pjj5846, gaek5, gyh0808, tmskim}@korea.ac.kr

***Abstract*— Along with the development of high-speed Internet and smart devices, various attack methods were emerged, and attack traffic has also changed into various and complex forms. In order to provide reliable services and efficient management of network resources, it is essential to detect and analyze the attack traffic. While various application and attack traffic detection or classification methods have been studied, but signature-based methods are still mainstream of the most. In this paper, we propose the seed based sequential grouping model for attack traffic detection. Model is consist of two main indices, which are similarity and connectivity index. In addition to model, we define the set of optimal thresholds of each index by using our balancing algorithm and define it as Guideline. By applying the proposed model to the actual attack traffic, we demonstrate that the model has high detection accuracy and completeness.**

***Index Terms*—traffic grouping model, attack traffic detection, similarity and connectivity between network flows**

I. Introduction

Today's emergence of high-speed Internet and ubiquitous environment has led to a rapid increase of developed applications on the Internet and network traffic complexity. Along with this situation, various attack methods were emerged, and also attack traffic has also changed into various and complex forms.

In order to provide reliable services and efficient management of network resources, it is essential to detect and analyze the attack traffic. In the perspective of Deep Packet Inspection (DPI) based traffic classification [7-10] and detection, many researches has already proposed their method to provide reliable services to users and maximize the utilization of network resources. The payload signature-based classification method has been shown to exhibit the highest levels of performance in terms of accuracy, completeness, and practicality [1-2]. However, these are depend heavily on pre-defined traffic patterns, signatures, and the processing speed of this system is difficult to meet the requirement for real-time handling of the large volume of traffic data passing through high-speed networks [3-4]. Therefore, the previous Signature-based methods cannot response dynamically to newly traffic and the signature generation process is too complicate or heuristic to apply.

To overcome these limits, we propose the seed based sequential traffic grouping model. Proposed grouping model can detect precisely the target traffic related with seed by grouping the flows. The concept of seed is a starting point of the detection. The traffic grouping process performs continuously from the seed flow until there are no more grouped flows.

The detection model consists of two indices, which are Similarity and Connectivity. Similarity index is for verifying the statistical similarity between source and target flow. Connectivity index is for verifying the correlation between similarity group flows and target flow. For the accurate detection, we define the optimal thresholds of each index as the detection Guideline.

The remainder of this paper is organized as follows. In Section Ⅱ, we explain the existing traffic classification and detection methods. We propose seed based sequential grouping model and describe the detail methods and algorithm of attack traffic detection, in Section Ⅲ. In Section Ⅳ, we perform the two evaluation experiment for verifying the effectiveness of the proposed method by testing it on five attack traffic. Finally, Section Ⅴ concludes this paper.

II. Realted work

In this section, we introduce the existing traffic classification and detection methods. Most existing traffic classification and detection methods are based on signatures.

Signature - based detection methods use automatic signature generation system to automatically extract signatures and detect traffic based on them [9-10].

Among the many kinds of signatures, the payload signature has the high accuracy and coverage. However, the extracting signature is very difficult and time consuming. Therefore, studies of automatic payload signature generation are in the limelight in the field of network management. The existing

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2015R1D1A3A01018057) and by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00513)

978-1-5386-3416-5/18/$31.00 ©2018 IEEE

methods are LASER (LCS-based Application Signature ExtRaction), Autosig, and SigBox.

LASER automatically generates an application signature, in the form of a sequence of substrings, in the payload of packet by using a modified version of the LCS (longest common subsequence) algorithm. The inputs of this algorithm are two distinct byte streams of packet payloads that belong to two flows. In order to improve the system's performance in terms of execution time and accuracy, this method only considers the first N packets of a flow and groups these packets by their size, since large packets are not likely to carry the same kind of information as the small ones. Finally, the method compares two inputs to get the longest common subsequence between them, and then compare it with another subsequence iteratively to refine it.

Autosig also generates an application signature automatically, which extracts multiple common substring sequences from input flows as application signature. First, it divides the payload of a set of flows into short substrings called shingles. After extracting all of the relevant, common shingles, Autosig merges them if they are neighbors or overlap. Next, a substring tree is constructed to create all possible combinations of substrings. These combinations are considered as signatures.

SigBox uses the Apriori algorithm [5-6] to solve the above disadvantage. The above methods are necessary preprocessing and post processing in order to compare two strings. The preprocessing is setting the order of traffic and grouping the traffic. The post processing integrates the generated substring into one rule. However, SigBox extracts substring likely to become signatures by increasing the length-1 all the substrings candidates [5]. Therefore, this method does not take much time to the extraction process and does not required preprocessing and post processing.

However, as described earlier, the automatic signature generation process is a very time-consuming task and it is very difficult to find meaningful signatures in encrypted payload content. In addition, there are several disadvantages that the signatures must be updated periodically and be regenerated depending on the type of various traffic.

III. SEED BASED SEQUENTIAL GROUPING MODEL

In this section, describes in detail about the Seed-based Sequential Grouping Model (SGM).

When seed information from attack traffic comes, SGM detects seed flows by using the seed information and these flows are defined as Seed Group. After detecting the seed group, SGM sequentially groups attack traffic based on Flow Correlation Index (FCI). FCI is a value that reflects the characteristics of each flow of traffic. It consists of two indices, Similarity and Connectivity. By using these indices, SGM groups the flows. As shown in Fig. 1, the sequence of grouping is cross-processed with similarity and connectivity.

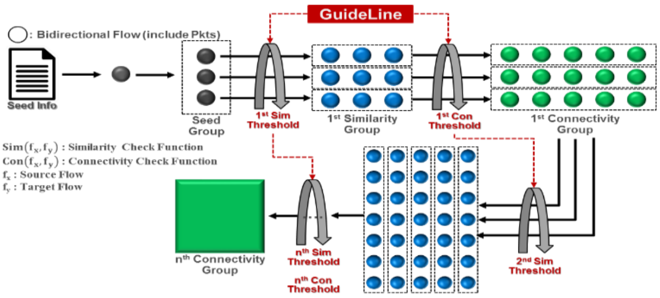


Fig 1. Cross Grouping Concept of SGM

A. Similarity Index (SI)

The similarity index checks the similarity between the source flow and the target flow. In each packets of the flow, the statistical information defined in Table 1 is extracted. The PIT, PSD and PSS features indicate the tendency of the flow packets.

To use these features, the similarity index is calculated by using Euclidean Distance, as shown in Eq. (1).

Table 1. Similarity Index Features

Feature	Explanation	Range
PIT_all_mean	Mean Inter-arrival time of all packets	0 ~ 120 (sec)
PIT_5_mean	Mean Inter-arrival time of first 5-packets	0 ~ 120 (sec)
PSD_all_mean	Mean Packet Size Distribution of all packets	0 ~ 1460 (byte)
PSD_5_mean	Mean Packet Size Distribution of first 5-packets	0 ~ 1460 (byte)
PSS_5_vec	Packet Size Sequence Vector of first 5-packets	-1460 ~ 1460 (byte)

$$\mathrm{SI}(f_x, f_y) = \sqrt{\sum_{i=1}^{5}\left(feature_i(f_x) - feature_i(f_y)\right)^2} \quad (1)$$

The reason for using Euclidean Distance is that it is possible to measure the distance explicitly compared to other similarity measure methods.

However, since flows with various characteristics occur within the same kind of traffic, very precise threshold value setting is necessary to group the target flows accurately. In addition, similarity-based detection can detect flows with similar statistical information, but cannot detect the all associated sessions flows. Therefore, it is necessary to use another detection method to detect these undetected flows. We use connectivity-based detection method for more accurate and diverse detection.

B. Connectivity Index (CI)

Connectivity Index checks the sequential connectivity between the source flow and the target flow. Connectivity index is checked through comparing the 5-tuple header information in the flow.

These four features and weights are defined in Table 2. As shown in Eq. (1), weights are given to the each four features and the connectivity index is calculated by using Euclidean Distance.

$$\mathrm{CI}(f_x, f_y) = \left(w_{ST} \times f_{ST}(f_x, f_y)\right) + \left(w_{IP} \times f_{IP}(f_x, f_y)\right) + \left(w_{Pt} \times f_{Pt}(f_x, f_y)\right) + \left(w_{Pr} \times f_{Pr}(f_x, f_y)\right) \quad (\text{where}, \textstyle\sum_{i=1}^{4} w_i \ = 1) \quad (2)$$

Table 2. Connectivity Index Features

Feature	Explanation	Function	Weight	Value Range
ST	Start Time	$f_{ST}(f_x, f_y)$	w_{ST}	0~1
IP	Source & Destination IP Address	$f_{IP}(f_x, f_y)$	w_{IP}	0~1
PT	Source & Destination Port Number	$f_{PT}(f_x, f_y)$	w_{PT}	0~1
PR	L4 Protocol	$f_{PR}(f_x, f_y)$	w_{PR}	1 or mean

Feature ST indicate the Start-Time of flow. As shown in Eq. (3), the similarity of flow occurrence time between source flow and target flow is compared through Euclidean similarity. As shown in Eq. (3), the flow time difference between the source flow and the target flow is calculated and converged to a value between 0 and 1 through standardization.

$$f_{ST}(f_x, f_y) = 1 - \frac{|(ST_x - ST_y)|}{MAX\_INTERVAL\_TIME} \tag{3}$$

$MAX\_INTERVAL\_TIME$: Maximum value of Interval Time

Feature IP indicates similarity of IP address between source and target flow. As in Eq. (4), the IP feature is defined by the same prefix calculation between the source and destination IP addresses. The reason for taking the square of the prefix is to increase the deviation to get more contrasted results.

$$f_{IP}(f_x, f_y) = \left(\left(\frac{PF\left(src_{P_x}, src_{IP_y}\right)}{32} \right)^2 + \left(\frac{PF\left(Dst_{IP_x}, Dst_{IP_y}\right)}{32} \right)^2 \right) \Big/ 2 \tag{4}$$

Feature PT indicate similarity of Port number between source and target flow. Since the flow of a session that occurs together usually uses port numbers of similar bands, the similarity between port numbers is checked through prefix comparison as well.

$$f_{PT}(f_x, f_y) = \left(\left(\frac{PF\left(src_{PT_x}, src_{PT_y}\right)}{16} \right)^2 + \left(\frac{PF\left(Dst_{PT_x}, Dst_{PT_y}\right)}{16} \right)^2 \right) \Big/ 2 \tag{5}$$

Feature PR checks whether the protocol between source and target flow is the same or not. If the protocol of the two flows is same, the PR is set to 1. However, if the protocol of the two flows is different, the PR is set to the average of the other three features ST, IP, and PT. Because if the PR is set to 0, the deviation value of the CI is calculated too large to detect the flows.

$$f_{PR}(f_x, f_y) = \begin{cases} mean(ST, IP, PT) : PR_x \neq PR_y \\ 1 : PR_x = PR_y \end{cases} \tag{6}$$

C. Detection Guideline (GL)

The Guideline (GL) defines the Optimal Threshold of SI, CI, and weight values described above. Table 3 shows the example of the GL. Applying GL to SGM enables more accurate and sophisticate detection. In order to group the flows associated with a Seed Group via SI and CI, there must be a threshold for the calculated index.

Table 3. An Example of Guideline

Group #	SI Threshold	CI Threshold	Weights			
			W_{ST}	W_{IP}	W_{PT}	W_{PR}
1	0.8651	0.7421	0.12	0.45	0.28	0.15
2	0.9024	0.8453	0.10	0.47	0.19	0.24
3	0.9233	0.9011	0.57	0.1	0.13	0.3

Previously, we had to perform a brute-force approach to find these thresholds to verify all traffic. However, this is a very inefficient and time-consuming task. In addition, we cannot do all the calculations for all of the cases to find the thresholds. Therefore, it is important to make an efficient threshold setting method to get a well-designed model.

D. Threshold-Balancing Method

To find out optimal GL thresholds, we propse the Threshold-Balancing algorithm (TB). This balancing algorithm finds a proper threshold that can generate a similarity and connectivity group adequately for each seed. As shown in Fig. 2, the first initial threshold value is 0.6 to classify the attack Ground-Truth (GT) traffic and Noise traffic (normal traffic).

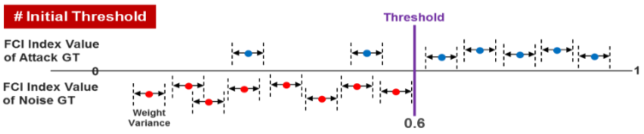


Fig 2. Initial Threshold

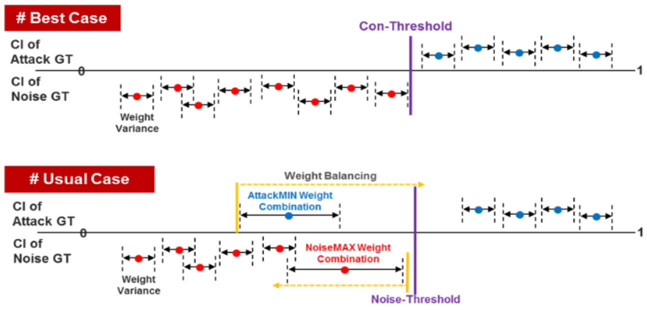


Fig 3. Two cases of Connectivity Threshold

The process of setting the initial threshold can roughly divided into two cases. The first is the best case in Fig. 3, if the Attack GT and Noise flow are completely independent, the optimal threshold value can be set without performing the threshold balancing process. The next is the usual case in Fig. 3, most of cases belong to this case.

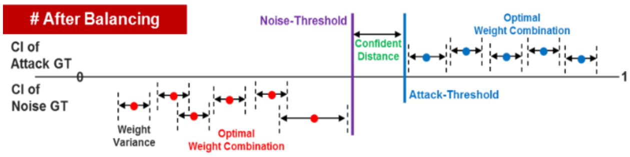


Fig 4. Connectivity Threshold after Balancing

In usual case of the connectivity grouping detection process, the threshold can adjusted by raising CI value of Attack GT flow and lowering CI value of Noise flow. In the Usual Case of similarity grouping detection process, the threshold is set a minimum SI value of the Attack GT flow as a threshold value to focus on the detection accuracy. When the threshold balancing is applied, the CI threshold value can be set as shown in Fig 4. Threshold balancing enables to detect undetectable GT flows and ensures higher detection accuracy than before.

```
A : Attack GT Flows / N : Noise GT Flows / F : Feature
Input : Flows, Initial THs / Output : balanced $TH_{con}$ and $TH_{sim}$
Initial_$TH_{sim}$ = 1.0
for i=1 to Numbers of Flow // $TH_{sim}$ balancing
    if Initial_$TH_{sim}$ > $Flow_i$.SI and N.$SI_{max}$ < $Flow_i$.SI
        if $Flow_i$ is not Grouped
            Initial_$TH_{sim}$ = $Flow_i$.SI
        $TH_{sim}$ = Initial_$TH_{sim}$
if A.$CI_{min}$ < N.$CI_{max}$ // Bset case
    $TH_{con}$ = {Any | in Confident Distance}
else if A.$CI_{min}$ > N.$CI_{max}$ // Usual case
    $F_A$ = find the MAX Feature in A.$CI_{min}$
    $F_N$ = find the MAX Feature in N.$CI_{max}$
    if $F_A$ == $F_N$ // Select the Feature
        $F_N$ = find the Second largest Feature in N.$CI_{max}$
    while true // $TH_{con}$ balancing
        Increase the $F_A$.weight and Decrease the $F_N$.weight
        Figure out the balanced A.$CI_{min}$ and N.$CI_{max}$
        if A.$CI_{min}$ < N.$CI_{max}$
            $TH_{con}$ = {Any | in Confident Distance}
        if $F_A$.weight == 1.0 // balancing fail
            $TH_{con}$ = A.$CI_{min}$
return $TH_{con}$ and $TH_{sim}$
```

Fig 5. Threshold-Balancing Algorithm

Threshold Balancing Algorithm is shown in Fig. 5. Inputs of this algorithm are initial threshold value and flows, and outputs are balanced similarity and connectivity thresholds. First, a Similarity Threshold Balancing is preceded. As we metioned before, the similarity threshold is set to minimum SI value. After the similarity threshold balancing, a Connectivity Threshold Balancing is preceded. It divided into best case and usual case. In best case, threshold is set as its value without balancing. In usual case, a minimum value of Attack GT flow threshold is raising and a maximum value of Noise flow threshold is lowering by comparing its value.

E. Threshold-Optimization Method

After the Threshold Balancing, a GL of seed group flows is extracted. However, GL generated through Threshold Balancing is individually optimized threshold for the each of seed group flows. Thus, in order to cover all of the seed group flows, the Threshold-Optimization (TO) process is needed for the combining each of GL into one Optimal GL. The entire process of the GL generation is shown in Fig. 6.

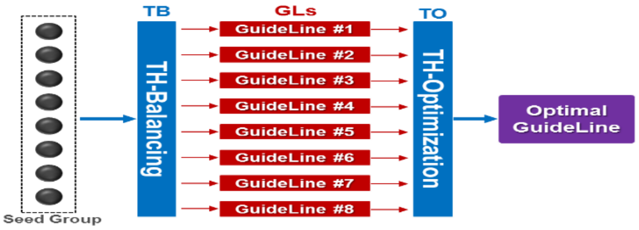


Fig 6. Guideline Generation Process

Threshold Optimization assigns a rating based on the detection rate and detection accuracy of each GL before combining. Eq. (7) shows how to calculate the rating.

When the rating is calculated, a threshold of each GL multiplies the rating as shown in Eq. (8). Finally, a threshold of Optimal GL calculated by the average of all GL's rated thresholds in Eq. (9). In other words, GLs are created for all of seeds from one trace, and each of GL is optimized to one Optimal GL This defined Optimal Threshold that provides an optimized method of detecting specific attack traffic.

$$GL_i\_TH\_Rating = G{:}Flows_A \times Precision$$
$$= G{:}Flows_A \times \frac{G{:}Flows_A}{G{:}Flows_A + G{:}Flows_N}$$
$$G{:}Flows_A = Grouped\ Attack\ GT\ Flows$$
$$G{:}Flows_N = Grouped\ Noise\ GT\ Flows \qquad (7)$$

$$GL_i\_TH_{Rated} = GL_i\_TH\_Rating \times GL_i\_Threshold \qquad (8)$$

$$GL_i\_TH_{Optimal} = \frac{\sum_{i=1}^{Numbers\ of\ GL} GL_i\_TH_{Rated}}{Numbers\ of\ GL} \qquad (9)$$

However, in order to detect various types of traffic, the threshold should cover all types of traffic. Therefore, we generate a single Converged GL by combining Optimal GLs of each traffic type as shown in Fig. 6 and Eq. (10). As shown in Fig. 7, each Optimal GL of traces is converged into one Converged GL by using Eq. (10).

$$GL_{Converged\ TH} = \frac{\sum_{i=1}^{N_t} GL_i\_TH_{Optimal}}{N_t}$$
$$(N_t = Numbers\ of\ Traffic\ Traces) \qquad (10)$$

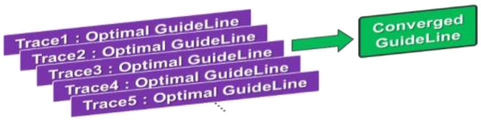

Trace1 : Optimal GuideLine
Trace2 : Optimal GuideLine
Trace3 : Optimal GuideLine
Trace4 : Optimal GuideLine
Trace5 : Optimal GuideLine
Converged GuideLine

Fig 7. Guideline Converging

F. Multple Seed based SGM

Although we use many methods and algorithm to get a better result, it has a limitaion to detect all of Attack GT flows by using only a single seed. Even if a seed information is correct, it is difficult to expect high detection coverage by only statistical and header information of the flow. Because various types of flows occur depending on the attack method, it is difficult to cover all of these various flows with using a signle seed.

In order to overcome this problem, we propose a method of multiple seed method. This method is to utilize several seed information in detection. Each single seed has a limited coverage. However, when we use these seeds together, the coverage of seed grately improve. The method of selecting multiple seeds is simple. First, we select a seed with the highest detection rate. Then we select the next seed that can detect the flow, which is not detected in former process. This selecting process repeats until the entire coverage is maximized.

IV. Evaluation

In this section, perform some experiment to evaluate the detection performance and validate the effectiveness of the SGM described in Section Ⅲ.

In order to perform the experiment, we built a detection system based on SGM. The input of the system is the traffic file, the seed information and GL, and the output is the detection result. The system divided into a Training Part and a Testing Part.

In the Training Part, GL is made with a seed information which is received from outside, and learned through the GL Generation, Threshold Balancing and Threshold Optimization modules described in Section Ⅲ.

However, if we cannot receive any seed information from the outside, the seed information is extracted from the attack traffic collected internally in the Seed Generation module.

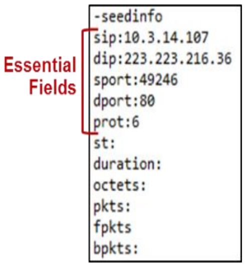
In the Testing Part, the SGM based traffic detection process performed by using the guideline. Then seed information generated in the training process as input. After the detection, flow information and an analysis log of the grouped flow generated. The detection result is measured by the three measurement (Recall, Precision, and F-measure).

A. Generate Seed Information

An important point in performing SGM-based detection is seed information. SGM performs continuous detection with seed information. When the seed information is provided from outside, we cannot know whether the seed information is accurate attack traffic information or not. Thus, the most important assumptions for the detection is whether the seed information is reliable.

As we methion before, if SGM-based system can get precise seed information from the ouside such as an external IDS or a firewall, it also can use those seed information in detection.

However, when it is hard to get an external seed information, the Seed Generation module starts. Seed information is extracted from all of attack flows. Because as we described in multiple seed, we have to evalute the performance of each seed to select mutiple seeds.


Essential
Fields

```
-seedinfo
sip:10.3.14.107
dip:223.223.216.36
sport:49246
dport:80
prot:6
st:
duration:
octets:
pkts:
fpkts
bpkts:
```

Fig 8. Seed Information

Fig. 8 shows an example of seed information. The essential seed information fields are 5-tuple of attack flow packets and other additional fields are not compulsory required. In this experiment, we use only 5-tuple fields as seed information.

B. Generate Guideline

A converged guideline of detection experiment is shown in Table 4. A converging stack of GL indicates the number of converged GL and conducted experiments. In this experiment, converging stack is 50 with five kinds of attack traffic.

Table 4. Converged Guideline

Converging(Learning) Stack : 50						
Group #	SI Threshold	CI Threshold	Weights			
			W_{ST}	W_{IP}	W_{PT}	W_{PR}
1	0.838190	0.841920	0.44	0.1	0.13	0.33
2	0.99	0.886890	0.26	0.15	0.11	0.37
3	0.99	0.865500	0.38	0.1	0.1	0.42
4	0.99	0.804750	0.43	0.1	0.23	0.24

From the second group of similarity groups, a fixed threshold value 0.99 is given. The reason for giving a fixed threshold value is that the ratio of False Postive is very high. Because the similarity is checked for a large number of connectivity group flows which are grouped in the previous step in the cross grouping process.

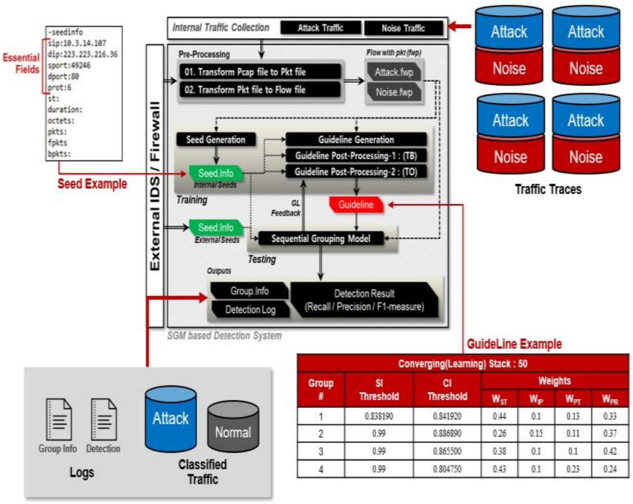


Fig 9. Entire System of SGM

Therefore, by setting the threshold accurately, we prevent false positive goruping and group flows that have almost the same statistical information as the flow detected by the previous step of connectivity group. To summarize entire system of SGM, Fig. 9 shows entire system of SGM with input traffic, seed information and GL.

C. Experiment Traffic Description

We used five types of attack traffic to the verification test: Malspam, GodZilla, Malware, Zbot, and Ransomware. Table 5 shows attack and noise traffic information. We used Microsoft Network Monitor to collect internet web site or other applications traffic for noise traffic and received attack traffic from a security company.

Table 5. Experiment Traffic Information

Attack Traffic Information				
Trace #	Attack Method	Size		
		Flow	Packet	Byte
1	MalSpam	71	18,055	15,167,725
2	GodZilla-Loader	69	1,862	1,358,410
3	MalWare	27	1,448	1,308,543
4	Z-Bot	23	1,232	1,229,269
5	Ransomware	3259	4246	1,169,285
Noise (Normal) Traffic Information				
Trace #	**Application**	**Size**		
		Flow	**Packet**	**Byte**
1	Chrome, IE web	491	60,266	46,852,995
2	Skype	576	107,534	95,635,596
3	Melon (Music Streaming)	746	55,126	46,069,984
4	KaKaoTalk (messanger)	844	53,655	49,471,332
5	Torrent	614	909,737	94,989,389

D. *Experiment Result*

This section describes the results of two detection experiments with five types of attack traffic. We conduct two kinds of experiment by using SGM and SnorGen (Automatic Signature Generation System) to compare the results and verify the validation of our proposed method in this paper. We used same input data in each of experiment. Table 6 shows the result of detection experiments.

Table 6. Experiment Result

Detection Test Result								
Input Traffic		**Measurement**	**Coverage (%)**					
Attack	**Noise**		**SGM**			**SnorGen**		
			Flow	**Pkt**	**Byte**	**Flow**	**Pkt**	**Byte**
MalSpam	all	Recall (%)	90.4	97.1	99.2	100	100	100
		Precision (%)	100	100	100	78.5	89.1	90.5
		F-easure(Flow)	94.9			87.9		
		F-measure Multiple Seed (2)	100			-		
GodZilla Loader	all	Recall (%)	89.2	95.7	99.7	94.2	96.1	95.8
		Precision (%)	100	100	100	90.5	94.5	96.6
		F-easure(Flow)	94.2			94.3		
		F-measure Multiple Seed (4)	100			-		
MalWare	all	Recall (%)	98.5	99.9	99.9	90.2	95.7	98.7
		Precision (%)	100	100	100	98.5	99.4	99.9
		F-easure(Flow)	99.2			94.1		
		F-measure Multiple Seed (3)	100			-		
Z-Bot	all	Recall (%)	45.7	67.2	80.6	85.7	92.2	95.7
		Precision (%)	100	100	100	100	100	100
		F-easure(Flow)	64.7			92.2		
		F-measure Multiple Seed (3)	90.5			-		
Ransom ware	all	Recall (%)	34.1	86.4	90.4	100	100	100
		Precision (%)	100	100	100	100	100	100
		F-easure (Flow)	50.8			100		
		F-measure Multiple Seed (2)	100			-		

Comparing with SGM using a single seed and SnorGen, the coverages of SGM were higher than SnorGen in most traces, but not in Z-Bot and Ransomware trace. However, when we apply mupltiple seed method in SGM, the coverage of SGM is higher than SnorGen in all traces.

The results show that average 95% of flow detection rate by using SGM (using a single seed), average 92% of flow detection rate by using SnorGen and average 98% of flow detection rate by using SGM with multiple seeds. In conclusion, we verify that the performance of SGM is higher than Signature based detection method.

V. Conclusion and future work

In this paper, we proposed a seed based sequential grouping model to detect attack traffic and implemented the system to verify the detection performance. Because of the verification, the proposed model showed high detection performance and proved its efficiency comparing with the result of SnorGen.

However, diversity of the applied traffic is relatively low comparing with other researches and the volume of traffic is small. Therefore, for the future work, we will conduct additional experiments by using a variety of attack traffic, and enhance the SGM.

References

[1] J. S. Park, S. H. Yoon, and M. S. Kim, “Software architecture for a lightweight payload signature-based traffic classification system,” in Proceedings of International Conference on Traffic Monitoring and Analysis Workshop, 2011, pp. 136-149.

[2] A. Dainotti, A. Pescape, and K. Claffy, “Issues and future directions in traffic classification,” IEEE Network: The Magazine of Global Internetworking, Vol. 26, 2012, pp. 35-40.

[3] . N. F. Huang, G. Y. Jai, H. C. Chao, Y. J. Tzang, and H. Y. Chang, “Application traffic classification at the early stage by characterizing application rounds,” Information Sciences, Vol. 232, 2013, pp. 130-142.

[4] T. Ban, S. Guo, M. Eto, D. Inoue, and K. Nakao, “Towards cost-effective P2P traffic classification in cloud environment,” IEICE Transactions on Information and Systems, Vol. E95-D, 2012, pp.2888-2897

[5] K. S. Shim, S. H. Yoon, S. K. Lee, M. S. Kim, “SigBox: Automatic Signature Generation Method for Fine-grained Traffic Identification,” Journal of Information Science and Engineering, Vol. 33, No. 2, Feb. 2017, pp. 541-573

[6] R. Agrawal and R. Srikant, "Mining sequential patterns," in Data Engineering, 1995. Proceedings of the Eleventh International Conference on, 1995, pp. 3-1

[7] N. Cascarano, L. Ciminiera and F. Risso, “Optimizing Deep Packet Inspection for High-Speed Traffic Analysis,” Journal of Network and Systems Management, Vol. 19, No. 1, Mar. 2011, pp. 7-31

[8]] N. Cascarano, L. Ciminiera, F. Risso, “Improving Cost and Accuracy of DPI Traffic Classifiers”, 25th ACM Symposium on Applied Computing (SAC 2010), Mar. 2010, pp.643-648.

[9] CA. Catania and CG. Garino, “Automatic network intrusion detection: Current techniques and open issues,” Computers and Electrical Engineering, Vol. 38, Issue. 5, Sep. 2012, pp. 1062-1072

[10] F. Risso, M. Baldi, O. Morandi, A. Baldini and P. Monclus, “Lightweight, payload-based traffic classification: An experimental evaluation”, In Proceedings of IEEE International Conference on Communications ICC, May. 2008, pp.5869-5875