The Method of Seed Based Grouping Malicious Traffic by Deep-Learning

Ui-Jun Baek
Korea Univ.
Network Management Lab
Sejong, Korea
pb1069@korea.ac.kr

Jee-Tae Park
Korea Univ.
Network Management Lab
Sejong, Korea
pjj5846@korea.ac.kr

Huru Hasanova
Korea Univ.
Network Management Lab
Sejong, Korea
hhuru@korea.ac.kr

Myung –Sup Kim
Korea Univ.
Network Management Lab
Sejong, Korea
tmskim@korea.ac.kr

***Abstract*—Today's networks are growing rapidly and the network sector is an important factor in developing a variety of applications and services. At the same time, a variety of malicious traffic is occurring that threatens the network environment, and it is causing massive damage to the network environment. Thus research to analyze and detect malicious traffic is essential in the field of network management. In this paper, we propose a method based deep-learning to distinguish only malicious traffic from a set of traffic mixed with normal traffic. We intend to test a variety of different groups of experiments using the proposed malicious traffic detection model. We will improve and enhance our detection model through analysis of the result.**

Keywords—Network Security Attack, Malicious Network Traffic Detection, Deep-Learning

I. INTRODUCTION

Today's networks are growing rapidly and the network sector is an important factor in developing a variety of applications and services. At the same time, a variety of malicious traffic is occurring that threatens the network environment, and it is causing massive damage to the network environment. Thus research to analyze and detect malicious traffic is essential in the field of network management. Previously, malicious traffic detection has been studied, but it has limits for accurate detection and extracting the unique characteristics of malicious traffic. Therefore, In this paper, we proposed the method of seed based grouping malicious traffic by Deep-Learning for improvement of high detection accuracy and detection coverage.

The organization of this paper is as follows. The related work is described in Section 2. Section 3 describes our proposed malicious traffic detection model. The detection result of our proposed model is presented in Section 4. Finally, concluding remarks are given and possible future work is mentioned in Section 5.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2018R1D1A1B07045742) and Institute for Information & communications Techonology Promotion(IITP) grant funded by the Korea Government(MSIT) (No.2018-0-00539-001,Development of Blockchain Transaction Monitoring and Analysis Technology).

II. RELATED WORK

In this section, we describe an important concept in this paper, seed-based malicious network traffic detection and existing detection methods.

The seed-based malicious traffic detection assumes that one of the traffic flows of a mixed traffic group of normal and malicious network traffic is certainly a malicious traffic network traffic flow. 6 features are extracted from this flow and this information is defined as seed. Then, calculate the connectivity between the seed and other different flows using the heuristic functions. If the calculated value is greater than certain threshold, the corresponding target flow is detected. And above process is repeated until there is no traffic flow to be detected. However, with existing models, the process of calculating connectivity is not sophisticated, making it difficult to detect malicious network traffic which have diverse characteristics [1].

III. PROPOSED MALICIOUS DETECTION METHOD

In this section we propose a method to detect malicious network traffic. We explain method to collect network traffic for experiments, method to pre-process the collected data, method to merge the pre-processed data, method to learn the data and detect malicious traffic from mixed with normal traffic data. The overall process of proposed method is shown in Fig. 1.

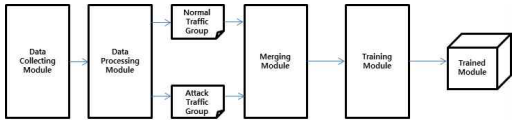

Fig. 1. Overall process of proposed method

A. Collecting of network traffic data

The process of collecting network traffic data consists of collecting normal network traffic and malicious network traffic. First, to collect normal network traffic, run applications and services that use network. Then, Collect normal network traffic from the applications and services you run through Wireshark. Next, collects malicious network traffic by downloading them from malicious network traffic

analysis sites[2]. The collected normal and malicious network traffic is a pcap(packet capture) file.

The collected file is converted into form of flow to facilitate analysis. First, extract only extract only the packet information from pcap file and create a pkt(packet) file. Second, collect the packets which have a same packet information(Source IP Address, Destination IP Address Source Port, Destination Port, Protocol) among the packets that are contained in pkt file. And define the collected packets as flow and generate a fwp(flow with packet) file. Third, from the generated file, collect the 6 information which include 5 packet information and start-time that is the time of occurrence of the first packet comprising flow. Finally, generate csv file which include 6 information of the flow. The overall process of collecting network traffic is shown in Fig. 2.

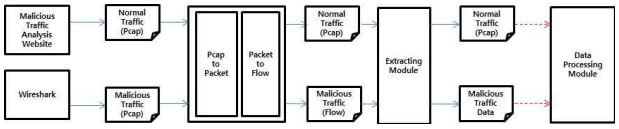

Fig. 2. Overall process of collecting network traffic

B. *Pre-Processing of network traffic data*

The process of pre-processing network traffic data consists of labeling, IP address scaling, time zone shifting, normalizing data. The overall process of pre-processing of network traffic data is shown in Fig. 3.

Fig. 3. Overall process of pre-processing of network traffic data

a) Labeling the data : First, label each of the network traffic flow data in order to learn whether it is normal or malicious. Normal network traffic data is labeled with zero. In contrast, malicious network traffic flow are labeled as one. Network traffic flow data processed in the ip scaling process contains 6 flow information and has a total of 7 features with label values added. The structure of labeled traffic flow data is shown in Fig. 4.

Start Time	Source IP	Destination IP	Source Port	Destination Port	Protocol	Label
(a) Structure of Labeled Traffic Flow Data						
0.1134	163.152.219.202	168.126.63.1	80	42798	6	0
(b) Example of Normal Traffic Flow Data						
0.1134	163.152.219.202	168.126.63.1	80	42798	6	1
(c) Example of Malicious Traffic Flow Data						

Fig. 4. The structure of labeled traffic flow data. (a) shows that the structure of labeled flow data, (b),(c) shows example of normal and malicious network traffic data.

b) IP Scaling : In this process, 4-bytes size IP values are divided into four one-byte values to extract a distinguishable patterns between normal and malicious traffic IP bandwidth in learning process. The structre of the normal and malicious network traffic flow data after this process is shown in Fig.

Start Time	S_IP [1]	S_IP [2]	S_IP [3]	S_IP [4]	D_IP [1]	D_IP [2]	D_IP [3]	D_IP [4]	Source Port	Destination Port	Protocol	Label
(a) Structure of Scaled Traffic Flow Data												
0.1134	163	152	219	202	168	126	63	1	80	42798	6	0
(b) Example of Scaled Normal Traffic Flow Data												
0.1134	163	152	219	202	168	126	63	1	80	42798	6	1
(c) Example of Scaled Malicious Traffic Flow Data												

Fig. 5. The strcutures of scaled network traffic flow data. (a) shows structure of scaled network traffic flow data, (b),(c) shows examples of scaled normal and malicious traffic flow data.

The normal and malicious network traffic flow data that has passed the merging process have 13 features including 1 start-time value, 4 source IP address values, 4 destination IP address values, 1 source port value, 1 destination port value, 1 protocol value and 1 label value.

c) Time Zone Shifting : We collected normal network traffic and malicious network traffic in the above steps. However, if collected normal and malicious network traffic did not occur at the same time zone, the learning and tests would be flawed. During the learning process, weight values and bias values that depend on start-time features will be assigned. The learning result can not be said to have been detected through a cleary distinguished. Therefore, when we collect normal and malicious network traffic, it is important to collect network traffic that was occurred at same time zone for an accurate experiment. But, it is difficult to collect normal and malicious network traffic that occurred at the same time zone. So, in this experiment, malicious network traffic files are donwloaded from malicious traffic analysis site. Because malicious network traffic downloaded from the analysis site has been traffic in the past, there is a difference in time between normal network traffic collected. Time zone shifting is the process of aligning occurring time of the normal and the malicious network traffic for accurate testing and detection. For example, if malicious network traffic that occurred in 2015 was downloaded from the analysis site, and normal network traffic that is currently occurring is collected through the wireshark in 2018, there is a difference of 3 years between malicious and normal network traffic. Thus, through a time zone shifting, the files are modified to assume normal and malicious network traffic occurred at the same time zone. In this process, we modify start-time value of normal and malicious traffic to January 1, 1970. This process allows an experiment to assume that normal and malicious traffic occurred at the same time. The simple explanation is shown in Fig. 6.

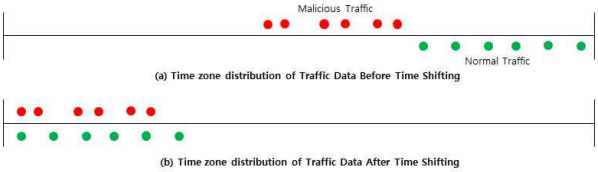

Fig. 6. Simple process of time shifiting. (a) shows the time distribution of normal and malicious traffic before the time shifting process. In this case, it is difficult to assume that malicious and normal network traffic are occurred in the same time zone. (b) shows the time distribution of normal and malicious traffic after the time shifting process. In this case, it is possible to assume that malicious and normal network traffic are occurred in the same time zone.

d) Normalization : The normalization process adjusts the value range of network traffic flow data to suit the

learning. Without normalization, it is difficult to properly calcuate the weigths and the bias values. The normalization process uses (1) so that the values of all data are in the range of 0 to 1.

$$\tilde{d}_i := \frac{d_i - \min\{d\}}{\max\{d\} - \min\{d\}} \quad (1)$$

C. Merging normal and malicious network traffic flow

The merging process merges the pre-processed normal and malicious network traffic flow to create the merged data for the learning process. The overall process is shown in Fig. 7.

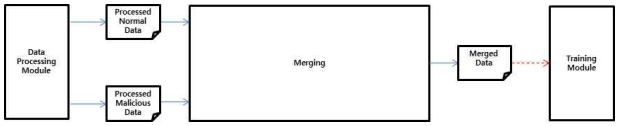

Fig. 7. Overall process of merging normal and malicious network traffic flow

The merging of normal and malicious network traffic flow data is intended to classify the malicious network traffic flow associated with it and not to classify normal network traffic flows unrelated to the malicious network traffic flows. The merged network traffic flow data is a combination of source flow and target flow, and the elements that each flow contains are shown in (2).

$$SourceFlow = \{x | x \in Malicious\ Traffic\}$$
$$TargetFlow = \{x, y | x \in Malicious\ Traffic, y \in Normal\ Traffic\} \quad (2)$$

To calculate the connectivity between the malicious network traffic and the target flow and to distinguish whether the target flow is a malicious or normal network traffic flow, the source flow includes only the malicious traffic flows and the target flow includes both normal and malicious network traffic flows. This implements the concept of seed-based malicious network traffic detection. The structure of merged network traffic flow data is shown in Fig. 8.

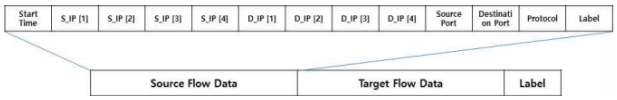

Fig. 8. The structure of merged network traffic flow data

After merging the source flow with the target flow, new label values are added to the merged data depending on the label values of the normal and malicious network traffic flow. Because all source flows are malicious traffic, they are labeled with 1 and consequently new values labeled is added to the merged network traffic flow data depending on whether the target flow is normal or malicious. If the labeled value of the source flow is 1 and the labeled value of the target flow is 0, a new labeled value of 0 is added to the merged network traffic flow data. It means learning the detection model not to detect target flow, which are normal network traffic that is not associated with the malicious network traffic, the source flow. Conversely, if the labeled value of the source flow is 1 and the labeled value of the target flow is 1, a new labeled value of 1 is added to the merged network traffic flow data. It means learning the detection model to detect target flow, which are normal network traffic that is associated with the malicious network traffic, the source flow. This process is shown in Fig. 9.

Source Flow Data	1	Target Flow Data	0	0
Source Flow Data	1	Target Flow Data	1	1

Fig. 9. Example of Labeling in merging process

The merged network traffic flow data that has passed the merging process have 25 features including 12 source flow data feature with deleted label value, 12 target flow data feature with deleted label value, and 1 newly labeled value.

D. Learning the merged network traffic flow data

The learning process includes learning the data that has passed the pre-processing and merging process according to the label value and creating a learned model for use in detection of malicious network traffic flow. The overall process is shown in Fig. 10.

Fig. 10. Overall process of learning process

The deep learning model used in learning process is multi-layer logistic regression [3], which consists of 1 input layer, 3 hidden layers and 1 output layers. It uses the sigmoid function as an active function and the gradient descent algorithm as a cost minimization function. The structure of deep-learning model is shown in Fig 11.

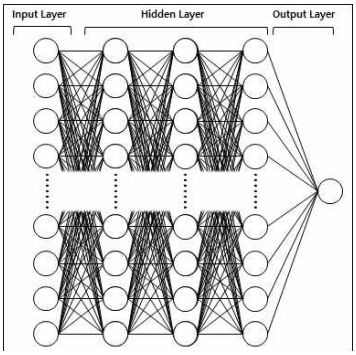

Fig. 11. The structure of deep-learning model

Out of the total of 25 merged network traffic data, array X is assigned 24 values, excluding label value, and a label value is assigned to array Y. X arrays with 24 values are multiplied by the 24 * 24 random weight variables matrix to produce an array of size 24. The generated array is then added with a 24 random bias variable. Each of 24 elements of array is inserted as an input to the sigmoid function, resulting in array of 24 size. The sigmoid function is shown in (3).

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

The model used consists of 3 hidden layers and repeat the above process 3 times. Then, through the 3 hidden layers, an array of 24 size is generated. The overall calculation process within the hidden layer is shown in Fig. 12.

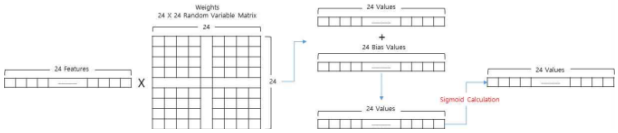

Fig. 12. Hidden-layer calculating process in learning process.

In the calculation process of the output layer, the predicted values are derived from the 24 size array generated in the above steps and calculation of output layer. An array of 24 size is multiplied by a random weight variable of 24 * 1 size, resulting in 1 variable. This result variable is added to random bias variable and is calculated by the sigmoid function. The last calculated value is the output value. The calculation process in output layer is shown in Fig.12.

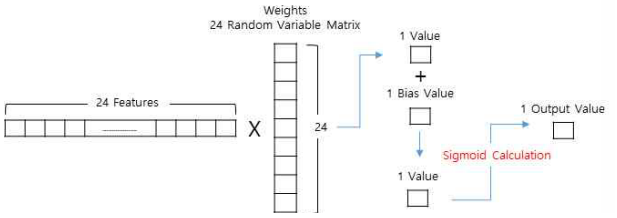

Fig. 13. Output-layer calculating process in learning process

An input array of 24 sizes is entered into the model and the model generates one output value. If this output value is greater than 0.5, increase the predicted value to 1 and vice versa, the predicted value decreases to 0. The predicted values are then compared with those in array Y, which is label values of original data, and the costs of both values are calculated. The function used to calculate cost is shown in (4).

$$cost(W) = \frac{1}{m}\sum \quad c\,(H(x), y)$$

$$c\,(H(x), y) = \begin{cases} -log(H(x)) & : y = 1 \\ -log(1 - H(x)) & : y = 0 \end{cases}$$

$$c\,(H(x), y) = -ylog(H(x)) - (1 - y)log(1 - H(x)) \tag{4}$$

Finally, after ending the above process and reduce the error using gradient descent algorithm. And repeat this process to adjust weights and bias values appropriately for detecting malicious network traffic. The weight and bias values of the model after end of learning are stored on ckpt(check point) file, which is the model learned. Model produced through the above process are used in the test process.

E. Testing Process

The testing process inputs network traffic data, which is different network traffic data other than the data used to learn, into the learned model to conduct malicious traffic detection tests. This results in normal and malicious network traffic group. The overall process of testing is shown in Fig.14.

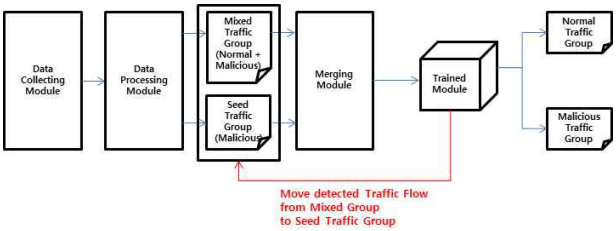

Fig. 14. Overall process of testing

The testing process includes the collecting process, the pre-processing process and the merging process. The collecting and pre-processing process is the same, but there are differences in the merging process. Only one of the flows whose elements make up the source flow consists of malicious network traffic, not the entire malicious traffic flow. This is shown in (5).

$$SourceFlow = One\ of\ Malicious\ Traffic\ Flow$$
$$TargetFlow = \{x, y | x\ \in Malicious\ Traffic, y\ \in Normal\ Traffic\} \tag{5}$$

In the step of creating the first group, the data from which the seed flow and target flow are merged are entered to detect network traffic flow related to the malicious network traffic flow. Detected network traffic flows move from the mixed group, which consists of normal and malicious network traffic flow, to seed group. Since then, the seed flow group has more than two malicious network traffic flow, including seed flow, and seed group is referred to as a malicious traffic group. From the second step, set the flow of the malicious traffic group as the source flow, and set the flow of the mixed traffic group as the target flow, merge the both flows and input them into learned model. However, from the second stage, there is one or more traffic flows that make up the source flows. Therefore, if, for example, the flow A of the source flows merges with the flow C of target flows, the flow C is detected as malicious, and the flow B of the source flows merges with the flow C of target flows, the flow C is detected as normal, problem arise that it is not possible to clearly determine whether flow C is normal or malicious. To solve this problem, the voting process between source flows is conducted. When more than half of traffic flows that make up the source flow detect that the destination flow is malicious, consider the target flow to be malicious network traffic flow and move it to the malicious traffic group. Conversely, when less than half of traffic flows that make up the source flow detect that the destination flow is malicious, consider the target flow to be normal network traffic flow and the target flow is left in the mixed traffic group. Repeat until there is no longer a target flows to detect the above process and eventually create a malicious traffic group and normal traffic group that excludes the malicious traffic flows. This process is shown in Fig. 15.

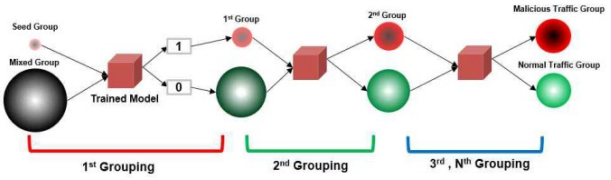

Fig. 15. Overall process of grouping

Detection results are derived through traffic flows traffic flows within each group. The detection results are shown as recall and precision, and the function for calculating recall and precision is shown in (6).

$$Recall = \frac{detected\ TRUE}{total\ number\ of\ existing\ TRUE}$$
$$Precision = \frac{TRUE\ detections}{whole\ detections\ of\ an\ algorithm} \quad (6)$$

IV. RESULT OF EXPERIMENT

In this section, the results of the learning and testing conducted with the data sets generated earlier and with the learned models are compared and tabulated with the detection results of the existing model. The results of the experiment are presented in two tables, result of learning and result of testing. Table 1 shows the accuracy of the learning process of the existing model and the deep learning model. Table 2 shows the accuracy of the testing process of the existing model and the deep learning model. Recall and precision are calculated as the average of the detection results for all seeds in each trace.

TABLE I. COMPASRISON ACCURACY BETWEEN PROPOSED AND EXISTING MODEL IN LEARNING PROCESS

	Existing		**Proposed**	
	recall	***precision***	***recall***	***precision***
T1	0.3535	1.0000	1.00	1.00
T2	0.2193	1.0000	1.00	1.00
T3	0.4866	1.0000	1.00	1.00
T4	0.5380	1.0000	1.00	1.00

TABLE II. COMPASRISON ACCURACY BETWEEN PROPOSED AND EXISTING MODEL IN TESTING

	Existing		**Proposed**	
	recall	***precision***	***recall***	***precision***
T1	0.1202	1.00	0.9728	1.00
T2	0.1097	1.00	0.8732	1.00
T3	0.3406	1.00	1.0000	1.00
T4	0.2152	1.00	0.8333	1.00

The results of the experiment show that the model using deep-learning showed a higher recall than the existing model, and the test results show that the model showed an average 4.6 times higher recall than the existing model.

V. CONCLUSION

In this paper, we proposed and experimented on how to seed-based detect malicious network traffic using deep-learning. The results of the experiment confirmed that the detection range of the model using deep-learning is larger than the existing model. Through this, it proved that the model using deep-learning is more suitable for detecting malicious network traffic than the existing model. In reality, however, malicious traffic has many different types of attack and sometimes it has different attack processes, even if it is the same attack type. The downside is that simple models used for detection are difficult to detect accurate malicious network traffic. Also, due to the complex characteristic of deep-learning method, it is difficult to explain the detection process in detail. accordingly, we intend to improve our detection model applying various deep-learning method and to analyze weight and bias values with in the learning and testing process for an accurate description of the detection process.

REFERENCES

[1] Jee-Tae Park, Sung-Ho Lee, Young-Hoon Goo, Huru Hasanova, Myung-Sup Kim. (2017). Attack Traffic Detection using the Correlation of the Flow and Seed based Sequential Grouping Model. Proceedings of Symposium of the Korean Institute of communications and Information Sciences, , 608-609.

[2] " Malware-traffic-analysis.net.". Malware-Traffic-Analysis.net.last modified Mar 28. 2018, accessed Mar 15. 2018, https://www.malware-traffic-analysis.net/index.html

[3] "GitHub.". hunkim/DeepLearningZeroToAll. Last modified May 13. 2017, accessed Jan 11. 2018 https://github.com/hunkim/DeepLearningZeroToAll/blob/master/lab-09-2-xor-nn.py