# Opcode Sequence Amplifier using Sequence Generative Adversarial Networks

1st Changhee Choi
*Agency for Defense Development*
Daejeon, Republic of Korea
changhee84@add.re.kr

2nd Sunguk Shin
*Agency for Defense Development*
Daejeon, Republic of Korea
ssw1419@add.re.kr

3rd Inseop Lee
*Agency for Defense Development*
Daejeon, Republic of Korea
dlstjq0711@add.re.kr

***Abstract*—Weapon systems and military networks are threatened by increased cyber attacks. In case of usual cyber attack, commercial grade defence systems are available. Because of speciality of military system, cyber attacks for them are also special and the number is extremely small. For machine learning based defence system, the shortage of malware samples is critical problem. To solve this problem, we proposed the new method for amplifying opcode sequence which is part of the malware and used for malware detection. We first extract opcode sequences from malwares and benign portable files. To make them meaningful and easy to learn, whole opcode sequence is split into several blocks, called OPSEN(OPcode SENtence), using special delimiters. Considering that opcode sequence is not a numerical data but a sequence of instruction, we used SeqGAN with stochastic policy in reinforcement learning and policy gradient. The experimental results shows that the proposed amplified opcode sequence help to improve the detection rate.**

***Index Terms*—Opcode, malware, GAN, SeqGAN, malware family**

## I. Introduction

In recent years, Advanced Persistent Threat(APT) attacks that threaten weapon systems are becoming more threatening and destructive every day [1]. Unlike the general system, if weapons systems are threatened, not only property but also security and people's lives cab be damaged. Attackers who focus on weapon system, create special malware to penetrate a special system [2]. In this situation, defense team can only obtain rare material such as artifact, malware, e-mail, compromised document, and so one. Commonly, an APT group set the period for one or multiple targets with similar attack method, called campaign. According to the report in [3], median value of the period of campaign in Asia-pacific area is 204 days at 2018. In same campaign, attackers use same or similar method to compromised the target. In case of same method, it can be defended by signature based intrusion detection system. However, it is difficult to defense similar attack with only few evidence of malwares even though machine learning equipped intrusion detection system.

To solve this problem, many studies have been conducted. The simple method is to make the signature of malware to solve this problem [4]. However, it is only work for exactly same malware. Another way is to make white list or model user profile which do not use attack data [5], [6]. These methods are fundamentally suffer from high false alarm.

Salem *et al.* propose the new method for aumenting ADFA-LD(system call sequence data) [7] to detect the intrusion [8]. They convert the ADFA-LD to 2-D image, and apply cycle-GAN [9]. Their results shows potential for effectiveness of Generative Adversarial Network(GAN) in cyber security field. However, the converting and training process with image of malware seems to be loose intrinsic properties.

Hu *et al.* design MalGAN to generating malware for black-box attack [10]. They extract 160-dimensional API features which is widely used in malware detection. This method is good for suppress of malware detector, however, it also loose the intrinsic properties in process of extracting 160-dimensional API features. Choi *et al.* tried to re-ordering the portable executables [11]. But only few samples are seems to be successful.

Common problem of previous researches is the loss of information during preprocessing. Since there are numerous good deep learning model for images, it is good strategy to use or tune existing models. In an environment where sampling malware is very difficult, the loss of information can be critical problem. In this paper, we proposed the new method of amplifying opcode sequence that can preserve intrinsic characteristics of malware. In manual analysis with cyber security specialist, they found similarity in many malware of APT by comparing sub-routine with opcode sequence [14]. Opcode sequence of malware is the least damaging method of the APT group's own attack methods. We focus on lossless algorithm for amplification of malware. To amplify sequential data, we use SeqGAN using policy gradient [12], [13].

The experimental results prove that the proposed method increase the defence model even if we have extremely rare samples. In test, Area Under Curve(AUC) rise from 0.79 to 0.89 and accuracy rise from 0.96 to 0.99 with Multi Layer Perception(MLP) classifier.

## II. Malware Preprocessing

There are many attention parts of malwares for detecting them, for example, whole malware, export function list, sub-routine, Portable Executable(PE) header, and so one. For amplification, attention part should be carefully choose due to sensitive properties of GAN. In [10], they use 160-dimensional Application Program Interface(API) features. We focus on the preservation of intrinsic properties of malware.

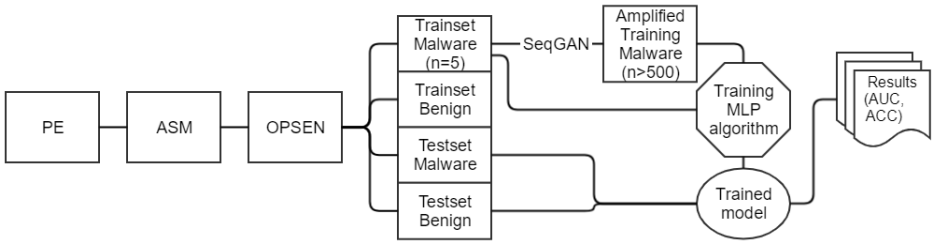


Fig. 1. Overall process of the proposed method

Since similar opcode sequence is good evidence for same APT group or malware family, we choose it. Most of domain of GAN is image or audio and some methods for sequence data is designed as short length. Figure 1 shows the overall process of the proposed malware preprocessing considering limitation of SeqGAN.

First, we split malware into subsections such as header, text, data, resource, and so on. We assume that the portable executable is based on x64 instruction in our experiments. We extract only text including opcode sequence by using IDA [15] which is the most famous disassembler tool. Then we split very long opcode sequence into small blocks called OPSEN(OPcode SENtence). Due to limitation of SeqGAN, we drop useless OPSENs. Since we deal with the APT attack which take aim at the weapon system, the number of training sample is extremely rare. We feed them to SeqGAN for seed data of amplification.

### A. Dataset

Best stuff for demonstration of the proposed method is use real malwares of APT. However, it is very dangerous to deal with activated top-tier APT and an experimental reproducibility is very difficult. For trustworthiness of an our experiments, however, we choose Microsoft Malware Classification Challenge(BIG2015) of 0.5 terabyte with 20K malware samples which is widely used in cyber security field [16]. This dataset has 9 malware families with various quantity. Fortunately, the 'Simba' family has only 42 samples of 10,868. It is nearly suited for the scenario we planed. We use only 5 samples of them for amplification and 37 samples for test. For benign side, we collect 1,481 portable executable files in my computer equipped with Windows 10. We drop the the sample including intermediate language from '.Net framework' of Microsoft to control the variables.

### B. OPSEN(OPcode SENtence)

Portable executable is consist of opcode and operand [17]. Opcode is machine language instruction such as 'add', 'subtract', 'multiply', and so on. Operand is target data or address of data of opcode. Opcode sequence is has the inherent of malware, some researches use them to detect malware [18], [19]. Particular small block contributes to the identification of malware detection rather than full sequence. In order to find proper unit size, we split whole sequence into various sizes. As a results, the length of sequence about 25 showed good results. We also need terminal punctuation for SeqGAN. We split full sequence with 'JMP', 'RETN'. 'JMP' means the jump unconditionally to label and 'RETN' is the return from procedure. Unconditionally changing the flow is common to both like terminal punctuation.

TABLE I
BENIGN OPCODE SAMPLES

| |
|---|
| push mov test jz mov mov mov mov add mov cmp jnb xor jmp |
| mov mov mov mov pop pop pop mov pop retn |
| mov test jbe cmp jb mov jmp |
| mov mov mov pop pop pop mov pop retn |

Samples of OPSEN of benign is shown in following box. 'RETN' or 'JMP' instruction finalize the OPSEN.

Table II shows the average number opcodes of both of malware and benign.

TABLE II
NUMBER OF OPCODES PER FILE AND SENTENCE

| avg. | length | opcode/file | opcode/sentence |
|---|---|---|---|
| Malware | 60,613 | 2,596 | 38 |
| Benign | 231,889 | 58,148 | 22 |

## III. SEQGAN

Opcode sequence is categorical type as shown in Table I Since there is no intermediate between instructions, gradient from loss can not be applied. To solve this problem, domain translation methods such as word-to-vec, doc-to-vec have been studied. These method designed for a lot of vocabulary such as English word and Korean word. The number of opcode types depends on the type of CPU, there are about 300 kinds of opcodes and most commonly used commands are limited. In case of benign, for example, there are 280 instructions and frequency of top 10 instruction('mov', 'call', 'lea', 'push', 'test'...) is over 81.5%. It is appropriate to use reinforcement learning(RL) using state & action. In [13], Yu *et al.* proposed the SeqGAN which use Reinforcement Learning(RL) as data generator. They nicely solve the problems that discrete values are difficult to apply directly to the original GAN, and they can only calculate losses for the entire sequence. We tuned the code in [20], and applied it to the opcode sequence.

## IV. EXPERIMENTAL RESULT

To test efficiency of the proposed method, we first conduct the amplification process of OPSEN. We set the number of

TABLE III
AMPLIFIED OPSEN OF SIMBA FAMILY

| |
|---|
| xor mov add cmp jb call inc mov pop retn |
| mov sub add mov push lea push lea adc push mov call add jmp |
| push push mov mov cmp jnz push call jmp |
| mov mov push push push mov call mov mov mov push jmp |

TABLE IV
COMPARISON BETWEEN BIASED AND AMPLIFIED DATA

| Data Type | TP | TN | FP | FN | AUC | ACC |
|---|---|---|---|---|---|---|
| Biased data | 0 | 969 | 0 | 37 | 0.792 | 0.963 |
| Amplified data | 28 | 968 | 1 | 9 | 0.980 | 0.990 |
| Gap | 28 | -1 | 1 | -28 | 0.188 | 0.027 |

epoch of discriminator pre-training as 100, maximum length of OPSEN as 64, the number of Monte Carlo search as 128. Table III shows the amplified OPSEN of Simba family(n=5). Note that the end of OPSEN is 'JMP' or 'RETN'. It is an effect reflecting the action&state policy gradient.

To validate effectiveness of amplification data of OPSEN, we prepare imbalanced data, and amplification data. We extract n-gram(1 to 3) features from OPSENs, and only use top 500 features. We test all available estimators in 'sklearn' in 'python', and we find out that MLP Classifier is well-matched to amplification data.

Table IV shows the results the proposed method. Our interest is true positive(observed=malware, estimated=malware). In biased data, MLP classifier can not detect any malware due to a very small number of samples(n=5). In amplified data, data volume has increased about 2200 times and it help machine learning model. As a results, 28 of 37(75%) malwares can be detected in our proposed method. To against high-end APT attacks, it is important to detect next second and third attacks even if we have very few samples.

Figure 2 shows the ROC curve between biased and amplified data. AUC of amplified data is outperform the original biased data.

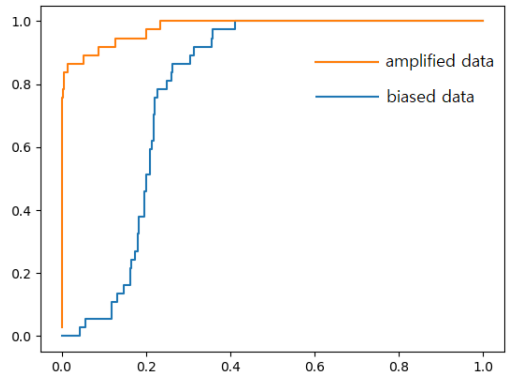


Fig. 2. ROC curve between biased and amplified data

## V. CONCLUSION

We proposed the new method for amplifying the opcode sequence in portable executable. Opcode sequence in portable executable is extracted by IDA. We divide them into small block, called OPSEN, so that they can be trained by sequence GAN model. The OPSEN data is amplified by SeqGAN more than 2200 times and then combined with the seed data for training. Experimental results shows that the proposed method can be help construct more accurate model under extremely biased dataset. AUC is increased from 0.79 to 0.98. MLP classifier can not be detect any malware in biased data, however, 75% malware in amplified data. In future, we will study about evaluating method for amplified cyber data to make them more reasonable.

## REFERENCES


[1] Symantec, "Internet Security Threat Report VOLUME 24, FEBRUARY 2019," *Network Security*, vol. 24, no. February, 2019.

[2] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[3] Fireeye, *M-Trends 2019: Fireeye Mandiant Services Special Report*, 2019.

[4] V. S. Sathyanarayan, P. Kohli, and B. Bruhadeshwar, "Signature generation and detection of malware families," in *Australasian Conference on Information Security and Privacy*. Springer, 2008, pp. 336–349.

[5] Y. Yang, K. McLaughlin, S. Sezer, T. Littler, E. G. Im, B. Pranggono, and H. Wang, "Multiattribute scada-specific intrusion detection system for power networks," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1092–1102, 2014.

[6] J. Peng, K.-K. R. Choo, and H. Ashman, "User profiling in intrusion detection: A review," *Journal of Network and Computer Applications*, vol. 72, pp. 14–27, 2016.

[7] G. Creech and J. Hu, "Generation of a new ids test dataset: Time to retire the kdd collection," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2013, pp. 4487–4492.

[8] M. Salem, S. Taheri, and J. S. Yuan, "Anomaly generation using generative adversarial networks in host based intrusion detection," *arXiv preprint arXiv:1812.04697*, 2018.

[9] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[10] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," *arXiv preprint arXiv:1702.05983*, 2017.

[11] C. Choi, "Seqgan application study for opcode re-ordering," in *KIMST autumn*, 2018, pp. 1159–1160.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[13] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[14] J. Sexton, C. Storlie, and B. Anderson, "Subroutine based detection of apt malware," *Journal of Computer Virology and Hacking Techniques*, vol. 12, no. 4, pp. 225–233, 2016.

[15] C. Eagle, *The IDA pro book*. No Strach Press, 2011.

[16] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft malware classification challenge," *CoRR*, vol. abs/1802.10135, 2018. [Online]. Available: http://arxiv.org/abs/1802.10135

[17] K. R. Irvine *et al.*, *Assembly language for Intel-based computers*. Citeseer, 2003.

[18] C. Feher, N. Tzachar, E. Berger, M. Gitelman, S. Dolev, and Y. Elovici, "Unknown malcode detection using OPCODE representation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5376 LNCS, pp. 204–215, 2008.

[19] "Opcode sequences as representation of executables for data-mining-based unknown malware detection," *Information Sciences*, vol. 231, pp. 64–82, 2013.

[20] "Seqgan implementation with keras," https://github.com/tyo-yo/SeqGAN, accessed: 2019-08-01.