IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

An Evasion-Resilient IoT Malware Detection Scheme with Invalidating Adversarial Byte Sequences and 1D Convolutional Filters

KOSUKE IGARASHI[1], HIROYA KATO[1], (Graduate Student Member, IEEE), and IWAO SASASE.[1], (Senior Member, IEEE)

[1]Department of Information and Computer Science, Faculty of Science and Technology, Keio University 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa 223-8522, Japan

Corresponding author: Kosuke Igarashi (e-mail: igarashi@sasase.ics.keio.ac.jp).

ABSTRACT Detecting Internet of Things (IoT) malware robustly against evasive techniques is imperative. As an IoT malware detection scheme, we focus on the scheme leveraging binary-derived grayscale images for Convolutional Neural Network (CNN) in IoT device detection because it can cope with the accelerating IoT malware surge due to the automatic and light-weight analysis which CNN can realize. However, that scheme can be evaded by the adversary with manipulated malware grayscale image or Binary Code (BC). In this paper, we propose an evasion-resilient IoT malware detection scheme with invalidating adversarial byte sequences and 1D convolutional filters. The valuable regions still remain which contribute to classification in both manipulated targets, grayscale image and BC, after the attacks. Thus, I improve the detection accuracy by statically extracting/enhancing valuable region of each manipulated target to the CNN model. By evaluation with each manipulated dataset, we show our scheme can improve detection performance against both evasive techniques. Exploring the valuable regions remaining in manipulated target and improvement of the detection accuracy utilizing the regions in my detection are the contributions of my research.

INDEX TERMS IoT malware detection, Convolutional Neural Network, Obfuscation, Adversarial Examples

I. INTRODUCTION

In these days, the Internet of Things (IoT) is interconnetcting a large number of electronic devices with a variety of applications in our lives, such as smart appliances, smart houses, smart grid, energy management systems at a tremendous speed. The number of the IoT devices is continuously increased. It is estimated to be about 50 billion all over the world by 2030 [1], [2]. They become targets of malware attacks due to the rapid increase and the development into devices which have interconnectivity in these days. Unfortunately, vulnerable IoT devices are spreading since the countermeasures cannot keep up with the evolution of IoT malware. Fig. 1 represents an actual case of attack damage by malware called Mirai. In fact, as shown in Fig. 1, attacks against companies using many IoT devices infected with malware have become a problem. Thus, this circumstance results in the urgency of detecting IoT malware.

Existing solutions for detecting malware are mainly classified into network-based schemes [3]–[5] and host-based schemes [6]–[12].

Network-based schemes pay attention to the fact that there exist strong evidences of malware in network traffic. This is because IoT devices infected with malware send and receive anomalous packets which are different from the original purpose of the device, such as flooding traffic, scanning of random IP addresses, or infection of other targets. As a result, these phases cause abnormal traffic which cannot be seen in benign cases. Based on the fact, those schemes utilize the features of packets passing through a gateway of the network. Although network-based schemes are useful, these schemes are subject to internal attacks of the local network itself because the packets do not go through the router.

In order to cope with the limitations of network-based schemes, host-based schemes are proposed. Host-based schemes are conducted by utilizing the features extracted from malware itself at each IoT devices. Most of the host-

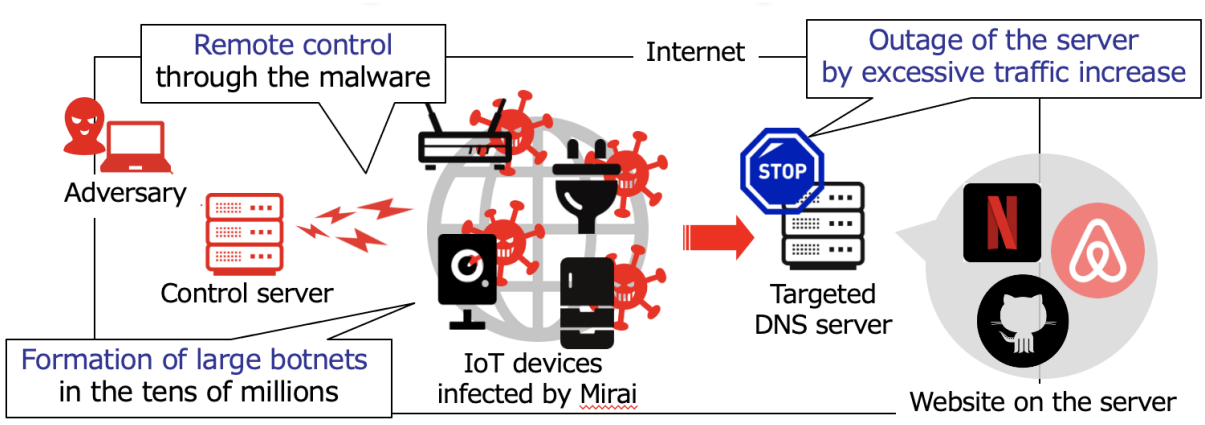


FIGURE 1. The case of attack damage by Mirai malware.

based schemes utilze various features which can be extracted statically because IoT devices are not suitable for dynamic detections due to their limited computing and storage capabilities. Kumar et al. utilezes the features regarding Control Flow Graphs (CFGs) acquired by complicated analysis and definitions of the features [8]. Although that scheme can extract represantative features of the application, it is a time-consuming and laborious method due to the analysis for an enormous number of IoT malware.

In order to cope with the spreading IoT malware with host-based detections, some researchers utilize Convolutional Neural Network (CNN) in their scheme. Su et al. utilezes GrayScale Image (GSI) converted from raw bytes of each malware Binary Code (BC) for its CNN [13]. That scheme is suitable for IoT malware detection because CNN can extract useful features from GSI automatically without feature engineering. However, it can be avoided with two representative evasive techniques, namely adversarial example and obfuscation by adversaries. As a result, Adversarial Malware (AM) and Obfuscated Malware (OM) are created by applying these techniques to GSI or BC of malware, respectively.

In order to detect such malware with exisiting evasive techniques, that is manipulated GSIs and BC, in this paper, we propose an evasion-resilient IoT malware detection scheme with invalidating adversarial byte sequences and 1D convolutional filters. The main idea of our scheme is that the some regions which contribute to decision making in each manipulated target still remain after the evasive techniques by adverasaries. This research aims to improve the detection accuracy by statically by statically extracting/enhancing each beneficial region of the manipulated target for the CNN model.

The contributions of this paper are as follows:

- We found the valuable regions which remain in adversarial malware and improve the detection accuracy of them by denoising procedure.
- We found the valuable features which obfuscated malware have and improve the detection accuracy of them by emphasizeing their valuable regions to CNN model.
- To the best of my knowledge, there is currenlty no reference to research defense schemes against the evasive techniques, namely AM generation and obfuscation by adversaries.

The rest of this paper is constructed as follows. After this introduction, related work are introduced in Section II. The previous scheme and the issues of that scheme are explained in Section III. Proposed scheme is described in Section IV. Various evaluation results are shown in Section V. Finally, the conclusions of this paper are presented in Section VII.

II. RELATED WORK

In order to detect IoT malware, several schemes have been proposed. These schemes are divided into network-based detection and host-based detection. Network-based detection is performed at a gateway on the basis of the features regarding propagation behavior for building a botnet. Meanwhile, host-based detection is conducted on each IoT device by utilizing static features extracted from malware itself. The representative schemes are explained in the following subsections.

A. NETWORK-BASED DETECTION

Network-based detctions [3]–[5] leverage the fact that there exists strong evidence of malware in network traffic. This is because IoT devices infected with malware send and receive anomalous packets which are different from the original purpose of the device, such as flooding traffic, scanning of random IP addresses, or infection of other targets.

Doshi et al. [3] propose the scheme focusing on specific communication patterns, so-called attack signatures, in packets associated with known attacks, such as packet size and destination IP address. While that scheme is effective against exisiting attacks, it is difficult to detect unknown attacks.

In order to cope with unknown attacks, Nguyen et al. [4] propose anomaly detection profiling the normal behaviour of devices. They build normal traffic profiles classifying typically used IoT devices into various device types, such as smart camera and smart speaker. Deviation from those profiles made from thousands of types of IoT devices is flagged

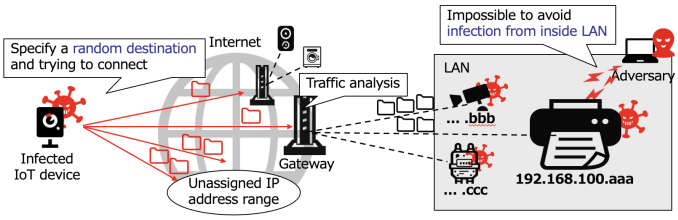


FIGURE 2. The network-based detection in the scanning/propagation flow and its problem.

as anomalous traffic. That scheme is able to discriminate normal traffic and anomalous traffic caused by botnet-attack. However, it cannot detect IoT malware activity much before the actual attack during the scanning/propagation phase, because it does not take into account the malware-induced scanning packet traffic generated by infected IoT devices. Thus, the detection schemes that are able to avert new malware infection or formulation of botnets are needed to reduce the risk of damage from attacks as much as possible.

In order to avoid new malware infection, Kumar et al. [5] propose the scheme focusing on network traffic while the scanning and propagation phase. Most of the existing malware behaviour in the phase are carried out in order to build a botnet as shown in Fig. 2. Therefore, that scheme aims at detecting the bots scanning and infecting vulnerable devices before they participate in a botnet for an actual attack such as DDoS. It utilizes the number of unique destination IP addresses and packets per destination IP addresses as features observing incoming packets which have spcific target destination port number that the existing malware tends to send. The number of unique destination IP addresses in case of malware-induced scanning traffic will be far more than benign traffic since the malware generates random IP addresses and sends malicious requests to them. That scheme can detect and isolate the bots before attacks because the scanning and propagation phase of the botnet life-cycle stretches over many months, which results in such abnormal traffic that cannot be seen in benign cases. Although that scheme can reduce the risk of malware infection itself, it is subject to local network attacks where IoT devices are targeted directly by adversaries in the same local network. Thus, exploring the malware detection schemes which can be realized by each IoT device is needed due to the vulnerability to local network attacks.

B. HOST-BASED DETECTION

Host-based detection schemes are conducted by utilizing the features extracted from malware file itself on each device. They tend to have similality because adversaries create new malware based on the existing ones [6]–[12] in many cases. Most of them utilize various features which can be extracted statically since IoT devices are not suitable for dynamic approach due to their limited computing and storage capabilities.

Researchers [6], [7] propose the schemes which utilize meaningful strings from the BC as metrics of benign software or malware. In particular, Torabi et al. [6] propose the scheme which utilizes strings related to the malicious behaviours, such as IP addresses of malicious hosts controlled by attackers(e.g., C&C servers) and/or embedded commands/payloads. That scheme focuses on the fact that IoT malware needs to communicate with adversarial hosts to obtain malicious command/payload and upload gathered information. This is typically achieved by embedding a series of commands and IP addresses to ensure successful post-infection communication for conducting further malicious activities. Although that scheme realizes lightweight static detection with simple reverse-engineering techniques, its detection can be easily avoided by malware where the meaningful strings are obfuscated in order to prevent them from being extracted. Such obfuscation techniques can be applied even on IoT devices where the computing and storage capabilities are limited.

In order to easily avoid detection rejection by such metric manipulating, researchers [8], [9] propose the schemes which utilize the relationship among the features, in addition to the features themselves, acquired from a graph. Each graph consists of nodes and edges. The nodes in the graph usually represent the actual features themselves, such as APIs, Opcodes, and their relationships are expressed by the edges. Several works have been proposed over the schemes which utilize the graphs, Control Flow Graph (CFG) [8], [9], opcode flow graph [10], function flow graph [11], and API-call graph [12].

In particular, Fig. 3 shows the overview of host-based detection using CFG. The theoretic metrics of CFGs are the number of edges, density of a graph, shortest paths between each node and so on. They can be multiple features of the model effctively derived from malware constructions under the limitation of static approach. Alasmary et al. [8] focus on the fact that some defferences in those metrics can be observed between benign software and IoT malware. In that research, they reveal that IoT malware, and also each malware family, have various algorithmic and structural properties through complex analysis of CFGs acquired from about 6,000 malware and benign software as shown in Fig. 3. Their scheme reveals that the CFG with the analyzed properties can significantly reveal less complex evasion techniques, such as packing and obfuscation, especially used in IoT malware. Thus, that scheme realizes highly accurate detection. However, that scheme is a time-consuming and laborious method due to the feature engineering for exploring effective metrics with analysing an enormous number of IoT malware samples.

Thus, we consider that the scheme utilizing CNN is the most suitable scheme to detect rapid-expanded IoT malware because it does not require the feature engineering. We focus on the detection proposed by Su et al. [13] as the previous scheme and explain in the following section.

III. PREVIOUS SCHEME

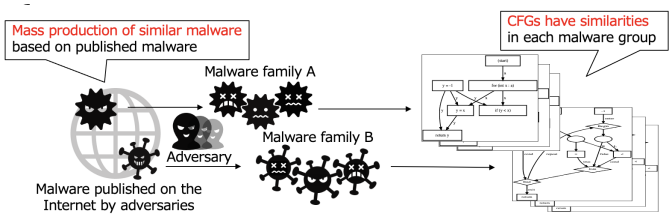


FIGURE 3. The overview of host-based detection using CFG.

A. OVERVIEW

In order to deal with spreading IoT malware with host-based detections in more practical means, CNN is utilized in the previous scheme [13]. The main idea of the previous scheme is that there exists the difference of GSIs converted from each malware BC which is always acquired in IoT devices due to the exectable format of them. In the previous scheme [13], a malware BC is reformatted as an 8-bit sequence and then is converted to a GSI which has one channel and pixel values from 0 to 255. To confirm the differences of malware and benign software, we prepared some samples of malware and benign software GSIs and compared them. Fig. 4 and Fig. 5 show the GSIs of malware and the ones of benign software, respectively. By comparison, some novel differences between them can be observed even by human eyes. For example, it can be seen that malware GSIs always are more dense. On the other hand, the GSI of benign softwares tends to have larger header parts than malware.

These GSIs can represents each features, which means useful features. Fortunately, CNN can extract effective features from these GSI automatically by learning deep non-linear features that can be hardly descovered and understood even by human eyes. Furthermore, once trained, the network model itself can be run with tiny computational resources, because only the trained parameters and information of network structure are shared, which results in lightweight detection on IoT devices. Thus, CNN is a suitable scheme for IoT malware detections since it does not demand feature engineering and also computational costs while the detection.

B. SHORTCOMINGS OF THE PREVIOUS SCHEME

Although the previous scheme can cope with spreading IoT malware with host-based detection in more practical means, it is susceptible to evasive techniques against the inputs of the CNN, GSIs and BC [14], [15]. Adversaries can evade the CNN based detection with two evasive techniques against each input. Fig. 6 shows the overview of evasive techniques. As shown in Fig. 6, malware samples whose GSIs are manipulated to evade the detection are called Adversarial Malware (AM), and ones whose BC is manipulated are called Obfuscated Malware (OM) in this research. The representative vulnerabilities to each attack are explained in the following subsections.

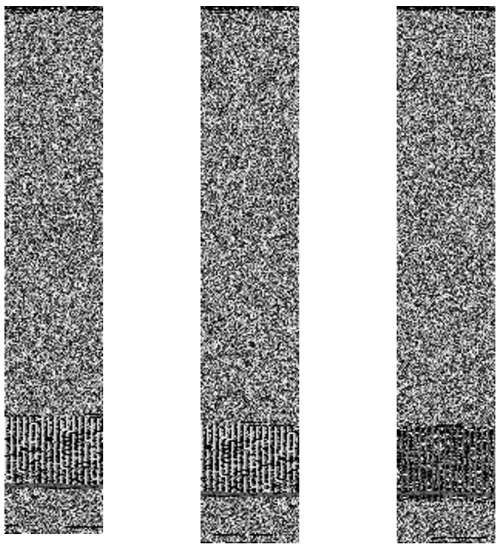


FIGURE 4. The GSIs of malware.

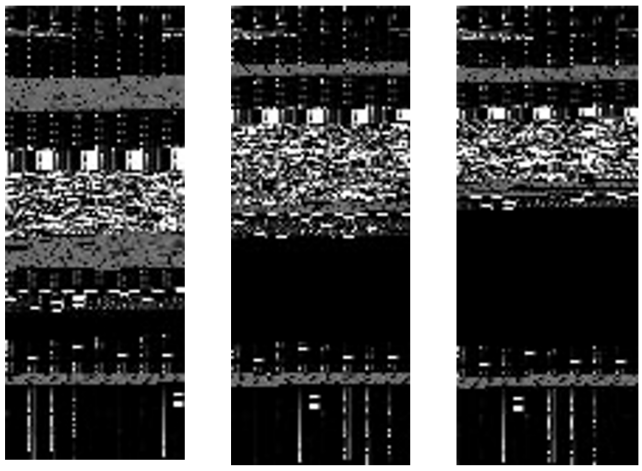


FIGURE 5. The GSIs of benign softwares.

1) Vulnerability to GSI manipulation

The GSIs fed into the network as inputs in the previous scheme also contain pixels that match meaningless BC, which is inserted into the source code by the compiler and has nothing to do with runtime behaviour [16]. Adversaries can make the CNN model cause false judge as benign by intentionally applying some noises to these pixel values [16]. Fig. 7 shows the GSIs of AM. The two are the actual GSIs of a malware before and after adding noise. The noise is added to the GSI of AM by manipulating the pixel values that match metadata or debugging information which is irrelevant to the runtime behaviour. By adding the noise in this way, the AM can mislead CNN in the previous scheme. The AM attack is a serious threat to CNN detection since it allows the classifier to intentionally judge the malware as benign while maintaining the original behaviour of malware.

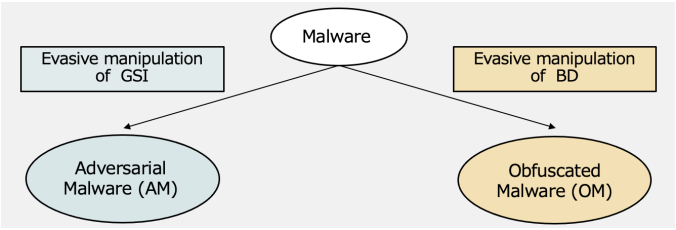


FIGURE 6. The overview of evasive techniques.

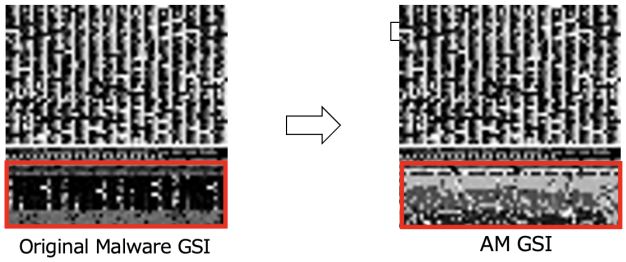


FIGURE 7. The AM GSIs.

2) Vulnerability to BC manipulation

The BC which are converted to a GSI can also be manipulated to evade the CNN detection by being obfuscated [6]. Due to the small computational resources of IoT devices, obfuscation techniques with packing tools, which also can compress the BC while encrypting, are tend to be used in IoT malware obfuscation field. Unfortunately, this packing tool is also tend to be used for data-nosakugenn in benign software in general /bunnkennsounyuu, which makes classification between malware and benign software difficult. Fig. 8 shows the GSIs of OM. As shown in Fig. 8, the original BC is compressed and encrypted at the same time, and new headers and tool-dependent data which do not contribute to the decision for detection are inserted. In addition to the fact that malware obfuscated with the packing tools can be recovered and analyzed only at runtime, the encryption method differs depending on the packing tool. Thus, BC manipulation attack is also severe threat especially for IoT devices which are not suitable for dynamic analysis due to the limited computing resources since there is no way to statically analyze them in advance.

While the previous scheme based on CNN can achieve high accuracy in detection, it is vulnerable to the above two attacks which aim to evade detection, resulting in lower detection accuracy. In fact, TABLE 1 shows the original accuracy and the degradation of them by each attack. The accuracy, which achieves 99.8%, drops to 66.1% under the AM attack with GSI manipulation and 86.2% under the OM attack with BC manipulation. The results indicate that the previous scheme is vulnerable to these attacks. vulnerable to these attacks. These attacks against CNN can be serious problems for other

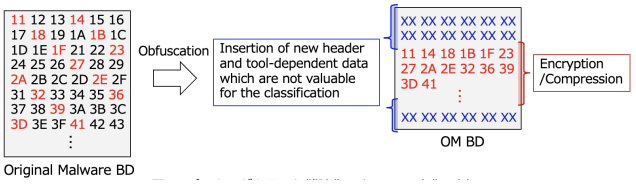


FIGURE 8. The OM GSIs.

CNN based detection schemes because CNN is widely used in the field of IoT malware detection. In spite of these serious threats, the effective coutermeasures have not considered until now.

Hence, in this reserch, I aim to propose defense methods against the evasive techniques on different targets, GSI and BC, for the practical use of static detection based on CNN.

TABLE 1. Detection accuracy of original/manipulated malware.

	Manipulation Target	Accuracy (%)
Original Malware	no manipulation	99.8
AM	GSI	66.1
OM	BC	86.2

IV. PROPOSED SCHEME

A. IDEA

Our goal is to design the defense scheme against each evasive technique on different targets, GSI and BC, to achive the practical CNN based IoT malware detection on IoT devices. In order to accomplish our goal, we propose an evasion-resilient IoT malware detection scheme with invalidating adversarial byte sequences and 1D convolutional filters. The main idea of our scheme is that there still remain regions which contribute to decision making in both manipulated targets after the attacks.

This is because, in the case of AM, adveresaries can add noise only to the pixel regions which do not affect the behaviour of malware. The noising to such area can cause malfunction because the noising of pixel values means rewriting the original executable BC. Therefore, the pixel regions which match the BC for its behavior still have valuable features even after the adversarial manipulation. Fig. 9(a) shows the actual malware GSI devided into two pixel regions, namely the manipulated area by adversaries and the area which have high degree of contribution for CNN to make decision. The red area is the pixel area which noise is added to. On the other hand, the yellow area is the area which contributes to decision making since noise cannot be added to the pixels in the area. We focus on the fact that the adversary can manipulate only the pixels in the red area which match the BC whose information is unnecessary at runtime in order to maintain the original functionality of the malware even after the noise is added.

Also, in the case of OM, the valuable BC is still remain after the obfuscation because packing tools tend to compress the original BC in the process. In the packing process, there

are mainly two procedures conducted, the compression of the original BC and the insertion of the new codes for decompression by each packing tool. Thus, it brings to two parts of BC area created by respective procedure in the malicious code after the obfuscation. We consider that valuable features still remain in the compression areas of the original code even under the discrete sequences. While the new code inserted affect such valuable features for classification because it only brings tool-dependency instead of valuable differences between both obfuscated samples. Fig. 9(b) represents the obfuscated BC. Also in the Fig. 9(b), the red area represents the manipulated area by the obfuscation and the yellow area does remaining valuable area for classification. In the OM, whole BC tends to be manipulated thorough the packing procedures. On the other hand, the yello area is the discrete sequeces which is compressed existing partly in the red area.

Based on the idea above, we aim to improve the detection accuracy by statically extracting/emphansizing each valuable region of the target to the CNN model. In each defense scheme, the detection accuracy can be improved by feeding those yellow region effectively to the CNN model. The representative schemes are explained in the following sections.

B. DEFENSE SCHEME AGAINST AM

A defense scheme against AM, which is malware whose GSI is manipulated, is explained in this section. In our proposal, we call manipulatable pixels by adversaries Junk Pixels (JPs), and we think it would be possible to remove the noise by converting their pixel values into 0, which means turning them black pixels.

The two GSIs as shown in Fig. 10 are the GSI of AM with noise added by adversary and the that with denoising for JPs in our proposal. The red box area is the area where the pixel values are converted into 0 as JPs. This pixel region corresponds to the byte sequences which contain unnecessary information at runtime, such as metadata and debugging information used while linking. Thus, they can be regarded as candidate pixels where adversarial noise can be added when AMs are created.

On the other hand, yellow box area is the area with no manipulation in our proposal. This pixel area corresponds to the byte sequences that are related to malicious behavior and data held in each file. Thus, it is judged to hold valuable information for the classification becausse it is difficult for adversary to manipulate these byte sequences reelated to the malware function without interfering with it.

By doing this denoising process, the detection without being misled by AMs would be possible since only the pixels which tend not to be manipulated can be used for our classification.

In order to identify the JPs, we focus on the fact that BC regions that are necessary or unnecessary at runtime can be classified according to informataion of each section. Based on the information of each section, the JP regions existing in a GSI can be searched by analysing the header in BC. An example of a header obtained by analysing a BC is shown in Fig. 11. From the header, we pay particular attention to the ALLOC flags, which are enclosed in a yellow frame in Fig. 11, indicating memory allocation at runtime. Sections with these ALLOC flags can be considered to be difficult for adversaries to manipulate since they contain information which affects the function of software. The manipulation of those sections can lead to destruction of malware functionality. On the other hand, sections without the ALLOC flags can be denoised since they are considered to be JPs which do not affect its function. The locations of JPs can be identified and denoised on the basis of the ALLOC flags and their addresses in the BC from the analysis as described above.

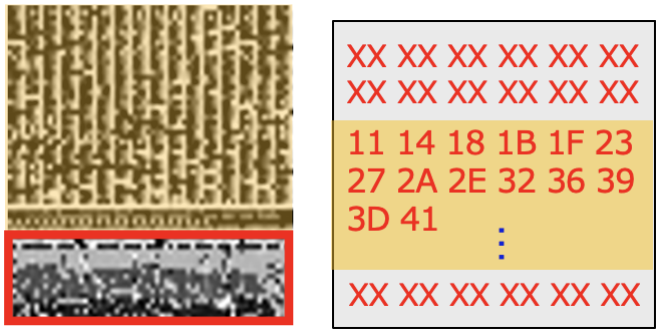
C. DEFENSE SCHEME AGAINST OM

A defense scheme against OM, which is a malware whose BC is obfuscated with packing tools, is explained in this section.

In this defense scheme, we focus on the fact that the appearing order of discrete byte sequences remaining after obfuscation is invariant before and after packing. Thus, showing the honrizontal connection of the byte sequences derived from the original BC can result in emphancizing valuable region remaining in OM for the CNN model. We realize it by convoluting the GSI pixels in the horizontal direction with 1D filters.

I explain the method of convolution in the horizontal direction of pixels with a 1D Convolutional filter (1D Conv.), comparing with that in two dimentions with a 2D Convolutional filter (2D Conv.) filter in the previous scheme. Fig. 12 shows the flow of the previous method using 2D convolutional filters, such as this blue flame, which are commonly used in image analysis due to correlation in both horizontal and vertical directions. The figure with 6×6 squares in Fig. 12 represents the GSI, and each of them is a representaion of a pixel. In particular, the yellow pixels represent pixels valuable information and the gray pixels represent pixels which are not necessary for classification due to tool-dependency. Unlike common images, the correlation in the vertical direction is weak in GSIs since they are converted from byte sequences. Furthermore, by convoluting with such 2D filters, there exist cases where both pixels, namely yellow and gray one coexist in the same filter especially in the convolution of the boundary of those pixel regions. It can result in the false classification because the weight of valuable pixels in the decision becomes small in consequence of the useless pixels coexisting in the same filter.

On the other hand, in our proposal, the GSI is convoluted using 1D filters with a horizontal length as shown in the red frame in Fig. 13, which represents the convolution in the proposal method. The proposed scheme can emphasize the byte regions which are valuable for classification with 1D Conv. As a result, it helps CNN to learn the correlation of discrete byte sequences remaining after obfuscation. Furthermore, it is possible to avoid the degradation of valuable pixel weight caused by the pixel which has coexisted in the vertical direction at the boundary of both pixels.



(a) The valuable/manipulated regions in AM.

(b) The valuable/manipulated regions in OM.

FIGURE 9. The valuable/manipulated regions in each target.

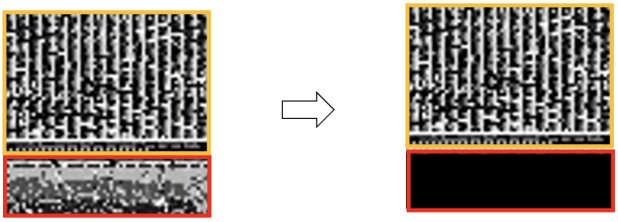


FIGURE 10. The difference with before and after denoising AM GSI.

As described above, by using 1D filters in order to show the horizontal correlations between encrypted byte sequences, we can highlight features retained in the OM, which results in ameliorating detection accuracy OM even under the static analysis.

V. EVALUATION

In order to demonstrate the effectiveness of our scheme, we evaluate Accuracy of each defense method compared to the previous scheme. Accuracy is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{1}$$

where TP, TN, FP, and FN denote the number of True Positive (malware is regarded as malware), True Negative (benign software is regarded as benign one), False Positive (benign software is regarded as malware), and False Negative (malware is regarded as benign software), respectively.

A. SIMULAION PARAMETERS

TABLE 2 shows our simulation parameters.

In accordance with the benign dataset in the previous scheme [13], we also use the benign software from Ubuntu system files [17] as the begin files. On the other hand, malware samples are collected from IoTPOT [18] and VirusTotal [19] which is a repository of malware samples for security researchers. In order to use those samples as inputs for CNN, we convert each sample to the correponding GSI by following the same procedures implemented in [13]. In particular, a BC can be reformatted to a sequence whose elements are 8-bit strings. Then, each string can be converted to a decimal number which can be seen as the value of a one-channel pixel. After that, we rescale the GSIs to the size of 64×64 such that they can be input to the CNN. In particular, manipulated malware for each attack is created with an adversarial technique implemented in [16], [20] and two packing tools [21], [22] shown in TABLE 2.

TABLE 2. Simulation Parameters.

The name of parameters	Value
benign softwares	Ubuntu 16.04.3 [17]
Malwares	IoTPOT [18] VirusTotal [19]
Number of benign software	1,442
Number of malware	7,263
AM attack method	White-box [16], [20]
Packing tools for OM	UPX [21] kiteshield [22]
Number of samples in AM-resiliency evaluation	800
Number of samples in OM-resiliency evaluation	1000

In the case of AM-resiliency evaluation, we apply a white-box method as a adversarial technique for AM, where an the adversary knows the training data and the structure of the CNN model because it is currently the mainstream in this reserch field [16], [20], [23].

In the case of OM-resiliency evaluation, we use two packing tools, called UPX [21] and kiteshield [22], in order for malware to be obfuscated. The UPX is a tool which is often utilized mainly for software compression purposes and, according to [24], the use of it is confirmed in more than 50% of obfuscated malware. On the other hand, the kiteshield is a tool which is often utilized mainly for the purpose of encrypting software, which realizes obfuscation of malware. Especially, in this evaluation, we prepared three datasets on the basis of tools used for their obfuscation in order to show that our scheme is independent of the packing tools. Two datasets out of the three are composed of malware obfuscated by each packing tools, namely UPX and kiteshield, and the other is a mixed dataset that comprises these datasets.

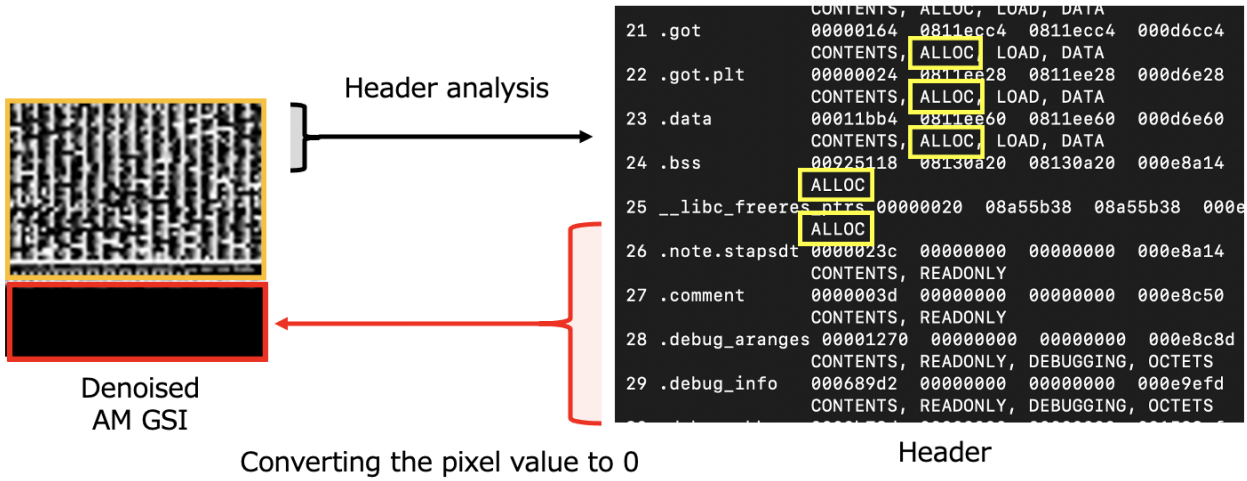


FIGURE 11. Header analysis and denoising process based on ALLOC flags.

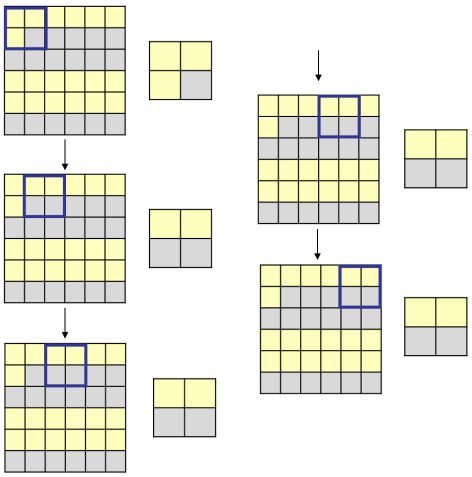


FIGURE 12. The flow of convolution with 2D Conv.

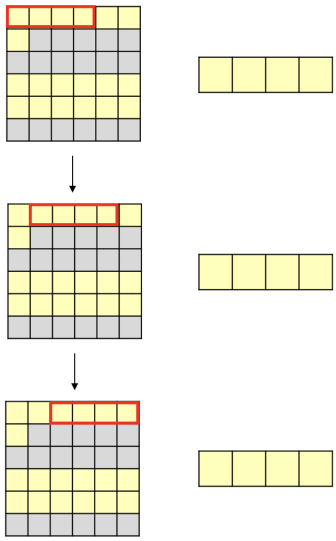


FIGURE 13. The flow of convolution with 1D Conv.

In the respective evaluation, some of the samples from malware dataset, 800 samples and 1,000 ones are picked randomly in AM and OM attack scenario, respectively. Furthermore, the same number of benign samples randomly selected are also utilized as inputs in order to keep balance between the number of malware samples and that of benign ones.

B. EVALUATION OF THE INVALIDATING ADVERSARIAL BYTE SEQUENCES AGAINST AM ATTACK

In this subsection, we show the effectiveness of the proposed method against AM which invalidates adversarial byte sequences with noise removal in the non-valuable pixels. We compare the accuracy of the proposed scheme with that of the previous scheme. Fig. 14 shows the accuracy of the proposal against AM. The left graph represents the accuracy of the previous scheme in detecting original malware, the middle one does that in detecting AM, and the right one does the accuracy of the proposal, where non-valuable pixel values in benign and AM samples are converted to zero. As shown in Fig. 14, the proposed scheme achieves improving the accuracy degraded by AM from 66.1% to 99.4%. Although a slight decrease in detection accuracy in the proposal is observed compared to the previous scheme, the proposal is able to maintain high accuracy. Thus, the loss of valuable information for classification by removing noise can be considered to be small. Hence, we conclude that the proposed method against AM, where the non-valuable pixels at runtime are focused on, achieves a robust defense method which can invalidate adversarial techniques in AM.

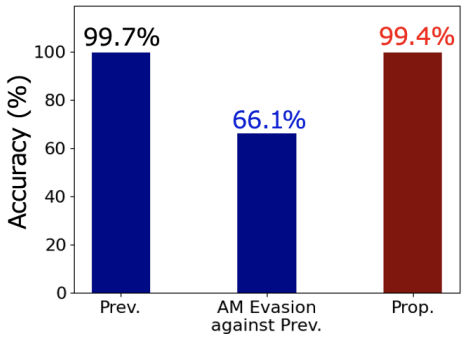


FIGURE 14. The accuracy of the proposal aginst AM (Prev. and prop. mean the previous scheme and the proposed one, respectively).

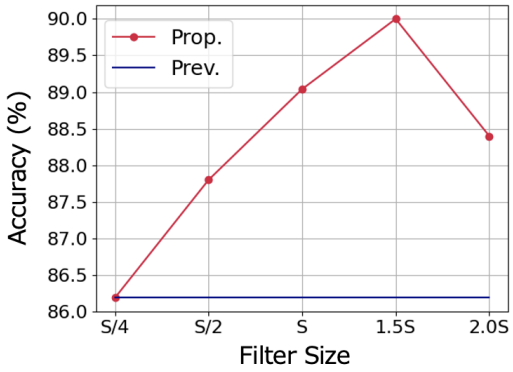


FIGURE 15. The relation between the accuracy and 1D Conv. size of the proposal aginst OM.

C. EVALUATION OF THE METHOD WITH 1D CONVOLUTIONAL FILTERS AGAINST OM ATTACK

The effectiveness of the proposed method with 1D convolutional filters against OM is shown in this subsection. We compare the accuracy of our proposed scheme using 1D Conv. with that of the previous scheme using 2D Conv. in each dataset as shown in TABLE 3. Compared to 2D Conv. on the previous scheme, the detection accuracy is improved by 1D Conv. on the proposal for all cases. In particular, in the datasets where the both-tools-made OM are mixed, the accuracy improves from 86.2% to 90.0%. According to the result above, we conclude that the proposed method with 1D Conv. against OM achieves a robust defense method with less tool-dependency against OM.

TABLE 3. The datasets used for the OM-resiliency evaluation.

	Accuracy	
	Prev. (2D filters)	Prop. (1D filters)
UPX	94.8%	**96.2%**
kiteshield	85.0%	**87.8%**
mix	86.2%	**90.0%**

In order to investigate the relation between the detection accuracy and the filter size of 1D Conv., we evaluate the accuracy varing the filter width. The 1D Conv. size is changed on the basis of the area of the 2D Conv., which is defined as S in the evaluation, in the previous scheme. The result is shown in Fig. 15, where the red line graph indicates the accuracy of the proposed scheme, and the blue one does that of the previous scheme. In the range of $1/4S$ to $1.5S$, the accuracy increases as the filter size also increases, and reaches 90.0% at $1.5S$. The result reveals that a certain width is necessary to emphasize the order of appearance to the CNN model. This is because the remaining original byte sequences is highly disjoint due to resizing during GSI conversion and compression by obfuscation procedures. In other words, it is difficult to extract useful features from discrete byte sequences with conventional short horizontal filters for the CNN model. Thus, the long horizontal filter is suitable for obtaining horizontal connections as features from a scattered byte sequences after obfuscation. On the other hand, the accuracy decreases from $1.5S$ to $2.0S$. We consider that this is caused by the decrease in learning efficiency due to the increase in parameters caused by the expansion of the filter size.

VI. LIMITATION

VII. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an evasion-resilient IoT ,malware detection scheme with invalidating adversarial byte sequences and 1D convolutional filters. We utilize remaining regions which contribute to decision making in both manipulated targets, GSI and BC, even after the attacks. In order to emphasize the valuable regions in GSI, the defense method converting JP values to zero is proposed aginst AM attacks. On the other hand, in order to emphasize the valuable regions in BC, the defense method showing the horizontal connection of appearing order of byte sequences with 1D Conv. is proposed aginst OM attacks. The defense methods of our scheme are effective in both cases. In the evaluation of the method against AM, it improves detection accuracy without the loss of valuable information. In the evaluation of the method against OM, it improves detection accuracy independent of the packing tool. Furthermore, I discover the horizontal connections in encrypted byte regions and the efficiency of 1D Conv.

REFERENCES

[1] "Scannetsecurity," Available: https://scan.netsecurity.ne.jp/article/2020/07/13/44308.ht
[2] "Itmedia," Available: https://techfactory.itmedia.co.jp/tf/articles/1704/13/news010.htm
[3] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in 2018 IEEE Security and Privacy Workshops (SPW), 2018, pp. 29–35.
[4] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏot: A federated self-learning anomaly detection system for iot," in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 756–767.
[5] A. Kumar and T. J. Lim, "Edima: early detection of iot malware network activity using machine learning techniques," in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT). IEEE, 2019, pp. 289–294.
[6] S. Torabi, M. Dib, E. Bou-Harb, C. Assi, and M. Debbabi, "A strings-based similarity analysis approach for characterizing iot malware and inferring their underlying relationships," IEEE Networking Letters, 2021.

[7] C. Hwang, J. Hwang, J. Kwak, and T. Lee, "Platform-independent malware analysis applicable to windows and linux environments," Electronics, vol. 9, no. 5, p. 793, 2020.
[8] H. Alasmary, A. Khormali, A. Anwar, J. Park, J. Choi, A. Abusnaina, A. Awad, D. Nyang, and A. Mohaisen, "Analyzing and detecting emerging internet of things malware: A graph-based approach," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8977–8988, 2019.
[9] H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, "A novel graph-based approach for iot botnet detection," International Journal of Information Security, vol. 19, no. 5, pp. 567–577, 2020.
[10] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, "Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning," IEEE transactions on sustainable computing, vol. 4, no. 1, pp. 88–95, 2018.
[11] H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, "A novel graph-based approach for iot botnet detection," International Journal of Information Security, vol. 19, no. 5, pp. 567–577, 2020.
[12] F. Xiao, Z. Lin, Y. Sun, and Y. Ma, "Malware detection based on deep learning of behavior graphs," Mathematical Problems in Engineering, vol. 2019, 2019.
[13] J. Su, D. V. Vasconcellos, S. Prasad, D. Sgandurra, Y. Feng, and K. Sakurai, "Lightweight classification of iot malware based on image recognition," in 2018 IEEE 42Nd annual computer software and applications conference (COMPSAC), vol. 2. IEEE, 2018, pp. 664–669.
[14] Q.-D. Ngo, H.-T. Nguyen, V.-H. Le, and D.-H. Nguyen, "A survey of iot malware and detection methods based on static features," ICT Express, vol. 6, no. 4, pp. 280–286, 2020.
[15] A. D. Raju, I. Y. Abualhaol, R. S. Giagone, Y. Zhou, and S. Huang, "A survey on cross-architectural iot malware threat hunting," IEEE Access, vol. 9, pp. 91 686–91 709, 2021.
[16] A. N. Carey, H. Mai, J. Zhan, and A. Mehmood, "Adversarial attacks against image-based malware detection using autoencoders," in Pattern Recognition and Tracking XXXII, vol. 11735. International Society for Optics and Photonics, 2021, p. 117350A.
[17] "ubuntu.com," Available: https://ubuntu.com/.
[18] "Yokohama national university," Available: https://sec.ynu.codes/iot/.
[19] "Virustotal.com," Available: https://virusshare.com.
[20] B. Chen, Z. Ren, C. Yu, I. Hussain, and J. Liu, "Adversarial examples for cnn-based malware detectors," IEEE Access, vol. 7, pp. 54 360–54 371, 2019.
[21] "Upx," Available: https://upx.github.io/.
[22] "kiteshield," Available: https://github.com/GunshipPenguin/kiteshield.
[23] B. N. Vi, H. N. Nguyen, N. T. Nguyen, and C. T. Tran, "Adversarial examples against image-based malware classification systems," in 2019 11th International Conference on Knowledge and Systems Engineering (KSE). IEEE, 2019, pp. 1–5.
[24] R. Isawa, Y. Tie, K. Yoshioka, T. Ban, and D. Inoue, "A first trend review of runtime packers for iot malware," IEICE Technical Report, vol. 117, no. 79, pp. 19–24, 2017.

KOSUKE IGARASHI was born in Hokkaido, Japan in 1996. He received his B.E. and M.E. degrees from Keio University, in 2020 and 2022, respectively. His research interest is security & privacy for IoT.

HIROYA KATO was born in Gunma, Japan in 1994. He received his B.E., M.E. and Ph.D. degrees from Keio University, in 2017 , 2019 and 2022, respectively. His research interest is security & privacy for IoT. He is a member of IEICE.

IWAO SASASE was born in Osaka, Japan in 1956. He received the B.E., M.E., and D.Eng. degrees in Electrical Engineering from Keio University, Yokohama, Japan, in 1979, 1981 and 1984, respectively. From 1984 to 1986, he was a Post Doctoral Fellow and Lecturer of Electrical Engineering at the University of Ottawa,ON, Canada. He is currently a Professor of Information and Computer Science at Keio University, Yokohama, Japan. His research interests include modulation and coding, broadband mobile and wireless communications, optical communications, communication networks and information theory. He has authored more than 301 journal papers and 446 international conference papers. He granted 48 Ph.D. degrees to his students in the above field. Dr. Sasase received the 1984 IEEE Communications Society (ComSoc) Student Paper Award (Region 10), 1986 Inoue Memorial Young Engineer Award, 1988 Hiroshi Ando Memorial Young EngineerAward, 1988 Shinohara MemorialYoung EngineerAward, 1996 Institute of Electronics, Information, and Communication Engineers (IEICE) of Japan Switching System Technical Group Best Paper Award, and WPMC2008 Best Paper Award. He is now serving as a Vice-President of IEICE. He served as President of the IEICE Communications Society (2012-2014). He was Board of Governors Member-at-Large (2010-2012), Japan Chapter Chair (2011-2012), Director of the Asia Pacific Region (2004-2005), Chair of the Satellite and Space Communications Technical Committee (2000-2002) of IEEE ComSoc., Vice President of the Communications Society (2004-2006), Chair of the Network System Technical Committee (2004-2006), Chair of the Communication System Technical Committee (2002-2004) of the IEICE Communications Society, Director of the Society of Information Theory and Its Applications in Japan (2001-2002). He is Fellow of IEICE, and Senior Member of IEEE, Member of the Information Processing Society of Japan.