Deep Packet Filtering Mechanism for Secure Internetworks

Hyun-Woo Kim and Eun-Ha Song*

Department of Information Security, Baewha Women's University, Seoul, 03039, Republic of Korea
Country Collage of Convergence & Liberal Arts, Wonkwang University, Iksan, Jeonbuk, 54538, Republic of Korea.
*Corresponding author. Email address: ehsong@wku.ac.kr

Article History: Received:11 november 2020; Accepted: 27 December 2020; Published online: 05 April 2021

Abstract: In this paper, we propose a Deep Packet Filtering Mechanism (DPFM) to analyze and filter malicious data packets moving between network environments. DPFM analyzes the behavior of malicious packets on the network and extracts information about the network as a sequence. After performing the word embedding process on the extracted sequence data using the word2vec technique, it detects malicious packets on the network by learning the LSTM model. In the past, research on filters to prevent malicious packets from entering the network by converting packets into data at the sending and receiving destinations and analyzing their purpose and maliciousness is insufficient. Since DPFM proceeds at the network boundary to analyze and extract malicious packets, primary detection is possible. In this paper, more accurate identification is possible by deep learning of network packets as well as OPcode and system calls, which are static analysis data.

Keywords: Packet Filtering, Deep Learning, Long Short Term Memory, Malicious Packet Detection, Secure Internetworks.

1. Introduction

Network communication technology with features of high speed, high reliability, and high density has been commercialized in order to realize a society that connects the entire information and communication technology industry such as artificial intelligence, cloud computing, and big data. This network communication technology provides a high-speed mobile communication environment to users regardless of data capacity and transmission speed, and it is possible to guarantee stable service for low-spec internet of things (IoT) devices. Due to this, various IoT industries such as smart city, smart home, smart farm, and smart factory are rapidly spreading [1-3].

However, with the advan

0m cement of network communication technology, it guarantees fast data transmission, but there is a high possibility of cyber threat as tens of billions of IoT devices with weak security are likely to be connected to each other. Unlike smartphones, tablets, and desktops, IoT devices are equipped with various applications, and there are no standardized architecture and security guidelines because the types of devices used are different for each service field. In addition, it is difficult to mount a high-level security function because the capacity of the battery and CPU performance are limited in order to perform a specific purpose [4]. When IoT devices are connected on a large scale on a network, they are very vulnerable to Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. In order to protect users' desktops, smartphones, IoT devices, etc. from cyber threats, security companies use machine learning-based AI algorithms to map and unsupervise the characteristics of malicious codes and abnormal behaviors. We are developing and providing security solutions to detect malicious behavior and detect malicious behavior.

Security companies have provided security solutions using machine learning techniques to protect various users' devices, but malicious code makers also use malicious code automatic generation tools and adversarial machine learning to respond to solutions from security companies. You are performing an attack that bypasses techniques. Various studies have been conducted to analyze and detect malicious codes in order to protect various devices from cyber threats by analyzing and detecting new and variant malicious codes with intelligent attack techniques [5-8].

Zhou[6] analyzed the Bayesian decision technique with machine learning that analyzes the detection of anomalies for data security and authentication according to network packet movement as well as internetwork, but it is applied to DTE (Data Terminal Equipment), so it is highly dependent on the host. .

Huda [7] proposed a malicious code detection model using a deep belief network technique to protect the industrial control system network connected to IoT devices from cyber threats. In order to analyze and grasp the characteristics of the malicious code, API calls were extracted as features from the data set. In addition, a malicious code detection model that learns the extracted API calls and determines whether there is a malicious code using a deep belief network technique was constructed, but the accuracy is not high due to a limited data set.

HaddaPajouh [8] proposed a model that detects malicious codes by extracting OPcodes and learning them in LSTM model to protect IoT devices used in various IoT industries from malicious codes. To build a malicious code detection model, 281 malicious codes and 270 non-malicious files were used as data sets. In order to learn

***Corresponding author:** Hyun-Woo Kim
Department of Information Security, Baewha Women's University, Seoul, 03039, Republic of Korea
Email address: ehsong@wku.ac.kr

the OPcode extracted as a characteristic of the malicious code, a model was built by setting the number of layers in the LSTM model in various ways. A new malicious code was detected.

This paper proposes a Deep Packet Filtering Machanism that analyzes network packets in a router to determine malicious behavior in order to prevent malicious code from spreading from infected devices to other devices through the network or attackers attacking devices. In order to determine malicious behavior, not only system calls and OPcodes, but also information on packets on the network are features to determine whether or not they are malicious codes, so that malicious code analysis is performed.

2. Deep Packet Filtering Mechanism

This paper analyzes information on packets transmitted and received in order to share information with other devices on the network, and learns them in a Long Short Term Memory (LSTM) model to efficiently detect packets with malicious characteristics. We propose Filter Mechanism (DPFM). DPFM analyzes the behavior of malicious packets on the network and extracts information about the network as a sequence. After performing the word embedding process using the extracted sequence data using the Word2Vec technique, it detects malicious packets on the network by learning the LSTM model.

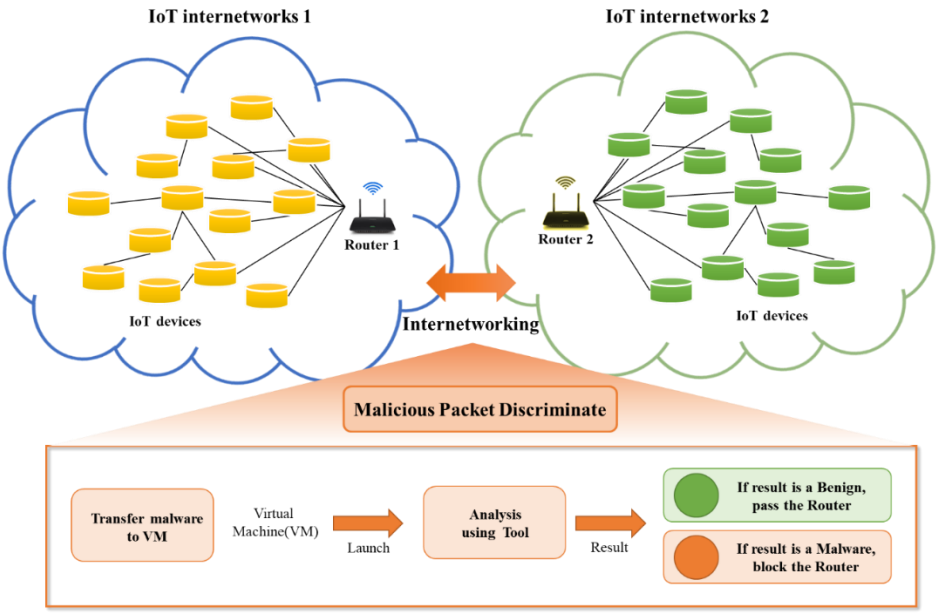

Figure 1 DPFM overview

Figure 1 shows the overview of DPFM. In an Internetwork environment where two or more networks are connected, various devices such as desktops, mobile devices, and IoT devices are all connected through routers. If the malicious code tries to access the router through the network, DPFM first moves the malicious code to the virtual machine and then executes the malicious code in the virtual environment. When malicious code is executed, it analyzes system calls and OPcodes and packets transmitted by malicious codes using IDA Pro, a disassembler tool, and WireShark, a packet analysis tool, and extracts log files for network-related activities. The extracted log files are trained in the LSTM model to determine whether there is a malicious code. If it is determined that the file executed in the virtual machine is malicious code, it is filtered and cannot pass through the router. On the other hand, if the executable file is classified as a non-malware file, the file is transmitted to another router through the router or received by other devices connected to the router.

This DPFM is largely composed of network packet capture, packet data preprocessing, and learning steps to analyze malicious packets and determine whether they are malicious through learning.

2.1 Network Packet Capture

DPFM must accurately capture packets flowing on the internetwork where various devices are connected by multiple routers. Therefore, in the network packet capture step, the first step of this paper, a packet is captured using WireShark, an open source packet analysis program, and a filtering process is performed to select packet data with malicious characteristics.

In DPFM, a file executed in a virtual environment uses WireShark to analyze packets that are intended to be transmitted through the network. WireShark is flexible in detecting malicious packets by compressing the amount of log data related to the network by automatically displaying detailed information within the packet data sent and received by the executable file, as well as capturing only the specific packets needed through the

filtering function. Provides. In the main packets captured through WireShark, features that can be judged as malicious behavior are extracted through the next step, feature filtering.

In the feature filtering step, first, a rule for selecting packets that perform malicious actions from a vast amount of data sets is required in consideration of the sequence of network packets mainly found in malicious codes and the frequency of use. First, features that show major malicious behaviors are selected according to the order and frequency of calls of network packets by various malicious codes discovered through previous studies, and these are generated as signatures. The generated signature is an index that can determine whether there is a malicious code, and it is compared whether a similar signature exists in another executable file in the data set. If there is the same packet call order and frequency as the signature through the comparison operation, it can be determined that the corresponding data performs a network-related malicious activity.

2.2. Packet Data Preprocessing

In order to train the filtered packet data as input data in the LSTM model, a pre-processing process of processing the packet data into data suitable for input into the LSTM model is required. The data preprocessing process largely performs extraction data cleanup and data embedding.

OPcode, system call, and network packet data extracted from the data set using IDA Pro and WireShark are integrated into one data format before vectorization into a vast amount. Log data is recorded in one log file.

After vectorizing the extracted data arranged in the LSTM model, word embedding is performed through a one-hot encoding technique. Through this embedding process, it is possible to learn the data sequence and appearance frequency related to OPcode, system call, and network packet from the LSTM model and determine whether it is malicious.

2.3. Learning

By entering the preprocessed data as input data into LSTM Learner, the characteristics of the malicious code are learned. Through learning, whether or not each data is malicious is detected.

3. Implementation and Performance Evaluation of DPFM

To build the DPFM proposed in this paper, we utilize the published malware data set and Virus Total of Microsoft Malware Classification Challenge (BIG 2015) hosted by Microsp in an environment consisting of AMD FX-8370E, 32.9 GB of memory. The experiment was carried out [9, 10].

As Microsoft recently distributed the Windows 10 version of the embedded operating system for IoT, IoT devices using the Windows operating system have begun to appear in the IoT market, so this paper does not detect embedded Linux-based IoT malware To detect IoT malware, the data set provided by BIG 2015 was used. In addition, the LSTM model was constructed using the Tensorflow library to learn the features of the LSTM model from the extracted data [11].

Figure 2 shows the DPFM's operation flow built to determine whether a network packet contains malicious code. In order to perform router filtering to protect the network from cyber threats by DPFM, the data set is divided into malicious codes and non-malignant files, and labeling is performed. After labeling the data set, WireShark is used to capture the network packets that each data sends and receives. After extracting the captured network packets, the data is processed to train the LSTM model in consideration of the call order and frequency. Among the network packets extracted from the data set, if there is a packet having the same characteristics as a packet of a malicious code that has been identified as a malicious packet performing a malicious action, it is filtered. Since the data related to the filtered packet is the main feature data to be trained in the LSTM model for malicious packet detection, it is essential to perform a preprocessing process that vectorizes these features according to the sequence before being input to the LSTM model.

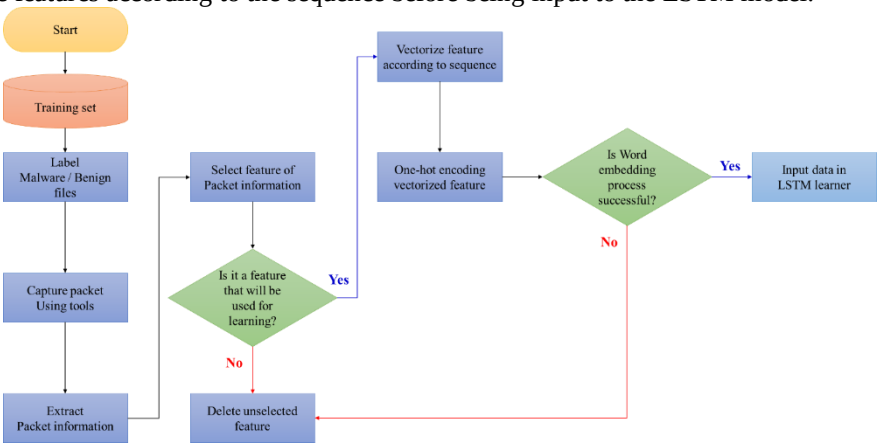

Figure 2 DPFM operation flow for detect and identification to malware

If the network-related data is not used for learning, that is, has unnecessary features, it is regarded as unnecessary when the feature data is trained in the LSTM model and related data is excluded. The network-related feature data vectorized according to the sequence is subjected to a word embedding process using a one-hot encoding technique. When the word embedding process is completed, it is possible to learn not only the sequence of opcodes, system calls, and network packet data, but also the frequency of use. After entering the filtered feature data into the LSTM model using the input data and labeling, training is performed. Table 1 provides network-related metadata extracted from outgoing and incoming network packets within the network[12,13].

Table1 DPFM metadata of detect malicious behavior

Type	Description
PID	It means process id used for network transmission
TTL (Time To Live)	It means packet validity period
Header Length	It means the length of the header, and the minimum length is 20 bytes
Header Checksum	This is an error detection device to check for errors that occur when sending packets
Source IP Address	It means the local IP address to which data is to be transmitted
Destination IP Address	It means the target IP address to receive data
Source Port	It means the port number used for data transmission
Destination Port	It means the port number used to receive data
Protocol	It refers to the protocol name used for data transmission over the network
Data	It means the content to be transmitted
Total Length	It means the total size including header and data

For the performance evaluation of DPFM, malicious code that transmits malicious packets to other devices and routers on the network was artificially generated and analyzed. The LSTM was modeled in DPFM. LSTM consists of 2 hidden layers and 128 cells in the layer. As a result of setting the epoch to 100, the learning accuracy is 95.02%, the loss value is 0.2517, the learning accuracy when learning is performed using test data not included in the training data set is 95.24%, and the test accuracy is 94%. Was measured.

Table 2 shows the results of evaluating the performance of the constructed DPFM after variously setting the number of cells in the LSTM to 64, 128, and 256. Due to the hardware specifications used to analyze and detect malicious packets, the number of cells is only possible up to 256, and when the number of cells is 128, the accuracy of the MaPS model is most suitable as 97.59%.

Table2 DPFM accuracy measurement with increasing cell counts

Cell	Hidden Layer	Accuracy (%)
64	2	97.54
128	2	97.59
256	2	97.59

Table 3 shows the measurement results after setting the number of hidden layers in the LSTM of DPFM to 2

or 3. Due to the physical hardware specifications, it is possible to configure only up to three hidden layers. When DPFM is built by setting the number of hidden layers to two, the accuracy of the most optimized malware detection model is measured at 97.59%.

Table3 DPFM accuracy measurement with increasing hidden layers

Cell	Hidden Layer	Accuracy (%)
128	2	97.59
128	3	97.19

4. Conclusions

Anyone can easily create malicious codes by using various automatic malicious code generation tools. As a result, many antivirus programs have been developed to defend against various attack techniques, but safety is being crippled due to the rapid spread of new and variant malicious codes. In this paper, a Deep Packet Filtering Mechanism (DPFM) was proposed to prevent malicious code from accessing through the network to infect target devices and systems. DPFM learns information about malicious packets by using the LSTM model, a deep learning technique, to detect and filter a vast amount of malicious packets existing in the network environment. Through DPFM, it is possible to actively and efficiently discriminate whether applications and executable files are malicious by learning not only opcodes and system calls, which are static analysis data, but also network packets through deep learning techniques. In future research, we intend to develop a technique to pattern malicious command combinations and detect malicious packets by considering various commands in heterogeneous operating systems such as Linux and Android. In addition, we intend to develop a network simulation environment for detecting various malicious packets and determining their accuracy.

5. Acknowledgements

This paper was supported by Wonkwang University in 2020.

6. References

1. Jagsir S, Jaswinder S. Detection of malicious software by analyzing the behavioral artifacts using machine learning algorithms. Information and Software Technology. 2020. p. 1-13.
2. Miguel L, Alberto P, Andrés O. An extensive validation of a SIR epidemic model to study the propagation of
3. jamming attacks against IoT wireless networks. 2019. p. 1-15.
4. Xiao CY, Zeng GL, Lewis N, Bruce N. Toward an Applied Cyber Security Solution in IoT-Based Smart Grids: An Intrusion Detection System Approach. Sensors. 2019. p. 1-22.
5. Shamsul H, Jemal A, Baker AR, Lei P, Mohammad MH. Automatic extraction and integration of behavioural indicators of malware for protection of cyber-physical networks. 2019. p. 1247-1258.
6. Alireza S, Rahil H. A state-of-the-art survey of malware detection approaches using data mining techniques. 2018. p. 1-22.
7. Donghao Z, Zheng Y, Yulong F, Zhen Y. A survey on network data collection. Journal of Network and Computer Applications. 2018. p. 9-23.
8. Shamsul H, Suruz M, John Y, Sultan A, Hmood AD, Robin D. A malicious threat detection model for cloud assisted internet of things (CoT) based industrial control system (ICS) networks suing deep belief network. 2018. p. 23-31.
9. Hamed HP, Ali D, Raouf K, Kim KRC. A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting. 2018. p. 88-96.
10. Ignacio M, José AH, Sergio DLS. Machine-Learning based analysis and classification of Android malware signatures. 2019. p. 295-305.
11. Sanjay S, C RK, Sanjay KS. Detection of Advanced Malware by Machine Learning Techniques. Soft Computing: Theories and Applications. 2019. p. 333-342.
12. Liyang H, Siqi L, Jinmian Y, Zenglin X. TensorD: A tensor decomposition library in TensorFlow. Neurocomputing. 2018. p. 196-200.
13. Pradeep Kumar Mallick,Sandeep Kumar Satapathy,Shruti MishraAmiya Ranjan Panda, Debahuti Mishra, “Feature Selection and Classification for Microarray Data Using ACO-FLANN Framework”, In: Mishra D., Buyya R., Mohapatra P., Patnaik S. (eds) Intelligent and Cloud Computing. Smart Innovation, Systems and Technologies, vol 194. Springer, Singapore. https://doi.org/10.1007/978-981-15-5971-6_53

14. Bisoy, S. K., Mallick, P. K., & Mishra, A. Fairness Analysis of TCP Variants in Asymmetric Network. International Journal of Engineering & Technology, 7(2.12), 231-233.