TELKOMNIKA, Vol.17, No.4, August 2019, pp.1706~1714
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v17i4.12065

Pre-filters in-transit malware packets detection in the network

Ban Mohammed Khammas*[1], Ismahani Ismail[2], M. N. Marsono[3]
[1]Department of Networks Engineering, Collage of Information Engineering,
AL-Nahrain University, Baghdad, Iraq
[2,3]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor Malaysia
*Corresponding author, e-mail: bankhammas@coie-nahrain.edu.iq

Abstract

Conventional malware detection systems cannot detect most of the new malware in the network without the availability of their signatures. In order to solve this problem, this paper proposes a technique to detect both metamorphic (mutated malware) and general (non-mutated) malware in the network using a combination of known malware sub-signature and machine learning classification. This network-based malware detection is achieved through a middle path for efficient processing of non-malware packets. The proposed technique has been tested and verified using multiple data sets (metamorphic malware, non-mutated malware, and UTM real traffic), this technique can detect most of malware packets in the network-based before they reached the host better than the previous works which detect malware in host-based. Experimental results showed that the proposed technique can speed up the transmission of more than 98% normal packets without sending them to the slow path, and more than 97% of malware packets are detected and dropped in the middle path. Furthermore, more than 75% of metamorphic malware packets in the test dataset could be detected. The proposed technique is 37 times faster than existing technique.

Keywords*: malware detection, middle path, network security, SVM*

1. Introduction

Cyber security appeared as great concern for the operations of institutions including governments, banks, business, as well as private users, where massive amount of sensitive information in the form of data are continually mined. Many challenges have appeared, one of these challenges is to prevent the spread of malware through the Internet, which is frequently enhanced over the years. Thus, accurate and efficient detection systems became an absolute need to recognize the malware assisted attacks that occur in the network and computer system. According to Symantec report in 2016, malware that spread currently in the networks is highly mutated and continuously updating them in order to avoid conventional detection systems [1].

Existing techniques [2, 3] try to detect the attacks or malware in the network excluding metamorphic malware. Metamorphic malware often changes their structure or body of codes in each infection, making the detection difficult or ineffective [4-6]. Signature-based technique cannot detect them due to their sophisticated strategy that avoids signature identification [7-9]. Detection of these new network-based malware before it reaches the host that posed non-trivial challenges [10, 11]. Most of the previous researches [4-6] have been dedicated to detect metamorphic malware at the host-level, which required code disassembly. Thus, their implementation cannot be made viable in the network since it is impossible to detect all payload contents.

The detection of malware in the network-level is difficult due to the need for flows process [12-19]. Varghese et al. [12] proposed a fast-slow path technique to detect the attack in the network-level. However, high percentage of normal packets need to be sent to the slow path that delay the transformation of the network traffic which need to be addressed this problem. Also, it is essential to develop a new detection technique to prevent the spreading of malware in the network before it reaching to the host without sending a lot of normal packets to the slow path. The rest of this paper is organized as follows: section 2 provides a comprehensive and critical literature review. Section 3 describes proposed network-based malware detection

Received December 15, 2018; Revised March 26, 2019; Accepted April 23, 2019

technique. In addition, the details of data collection that used in this work are presented in section 4. Section 5 analyzes the results and measures the malware detection ability of the proposed technique in the network to hold the spread of malware in the real network traffic. Also in this section, the speedup of the proposed method is measured and highlighted. Section 6 concludes the paper.

2. Related Works

Over the years, network attacks are conducted to disrupt, degrade, deny, or destroy information resident in computers and computer networks [20]. Detecting network attacks using network intrusion detection system (NIDS) depends on the attack types and the user's goal, which is divided into two approaches. First is the flow inspection and second is deep packet inspection (DPI). The flow inspection deals with the attacks that deviate from the normal network traffic behavior. Usually, this type of attack affects the whole network states and performance such as Denial-of-Service attack (DoS). DPI monitors the packet content and search for the presence of any unauthorized code, malware code, and malware fingerprints [21].

Many dedicated efforts are made to detect attack or malware in network by combining the ML with Snort signatures [22-25]. De Lima et al. [22] extracted the feature of attack and benign files from the packet flows. They mentioned that most of the new attack carried and used the strategy similar to the strategy of their predecessors, which implies the attack shared the same features. Based on this fact the neural network is trained to identify new attack, which is implemented on a MLP 256-21-1 network to achieve detection accuracy about 74%. Other research such as [23, 24] used neural network (NN) in attack detection and extracted the feature of attack and benign from the packet content. They combined the Hamming Net NN (HNNN) with 46 Snort signatures for training to classify the illegitimate information in TCP/IP packet payload.

In some network intrusion detection, the researchers used anomaly detection and n-gram features are extracted, they shown that n-gram analysis is not only efficient for the malware detection but also capable of detecting different types of attacks [3, 25-32]. Rieck and Laskov [27] proposed a method using language models that extracted features from payload using variable length of n-grams. They applied unsupervised anomaly detection to detect attack on the TCP connection in application layer. Using clustering algorithms and statistical analysis, a detection accuracy over 80% is achieved with no FPR. Ismail et al. [25] proposed content-based detection of new malware at the network infrastructure level.

As aforementioned, detecting attacks in the network requires careful consideration of packet and flow processing [12]. Varghese et al. [12] proposed a technique to detect evasion attacks without reassembly of TCP flows. The known attack signatures are divided into fixed sub-signature to detecting them in the incoming packet using string matching. A fast path and slow path framework is proposed to improve the detection speed and minimize the memory. In the fast path, the incoming packet is inspected for containing any piece of sub-signature (after divided the signature to fixed size 4 byte) using any matching algorithms. Then this packet is sent to slow path to reassemble the packets for getting all flow and making deeper analysis at the host to decide if it is an attack or not. They described that the packet is inspected through many chips in normal IDS/IPS and the TCP and IP flow state is stored in large state table with huge memory for C connections and W bits per connection. The minimum state for flow connection is 5-tuple and the sequence number is at least 128 bits. Thus, the minimum memory required is 128 Mbits for one million connections, which is sufficiently large. The reassembly of the TCP and IP packets in network intrusion prevention systems (NIPS) is expensive because it needs to keep track for millions of connections. However, using a separate fast path and slow path, the speed of the packet passing in the router can be increased around 10 times than the time needed for detection in layer 7. Although it is possible to extract n-gram features and classification on individual packet payload, efficient architecture is needed to address non-malware packets.

Boukhtouta et al. [2] compared the packet header features with DPI for detecting malware in the network level. Total 22 flow header features are extracted and then different ML technique is used to classify the packet header using WEKA. Experimental results revealed that J48 and Boosted J48 produced the best outcome. Models for different malware families are created using HMM. For DPI, complete captures (pcaps) is used to inject it to the model called

MARFPCAT (Modular Audio Recognition Framework-based PCAP Analysis tool). Each pcap is loaded to that tool, where a signal is interpreted as a waveform. The signal enclosed the flows with both the header and payload. Results showed that 771 out of 1,063 malware classes are classified with 100% accuracy and the rest produced the accuracy lower than 75%.

3. Proposed Network-Based Malware Detection Technique

Accurate detection of malware at the network level requires improved in the technique. Recent researches revealed that host-based ML classification can detect malware with high detection accuracy. The process must be accelerated [12] to overcome the limitations of detecting malware at the packet level. From this view, this paper proposes a strategy to detect the malware contents at the network level as the first line of defense to protect systems connected to network from being infected, this is the main contribution of current study. This technique can be implemented in middle-boxes placed near edge router to protect system connected to network from malware spread.

Earlier methods [33] are enhanced by masking Snort sub-signatures and using term frequency of malware features. The proposed method uses extracted n-gram features from the content of the binaries and filtered the number of n-gram features by masking these features with Snort sub-signature. Snort rules are used for network intrusion detection system. Others rules such as Bro or Suricata can be used as well. IG feature selection method [34] is employed to select only the important features as a second-level filter. This allowed the SVM classifier to focus only on the significant malware features. The processing of network traffic can be done on a packet based, where individual packets are subjected to ML classification of n-gram features to detect the presence of malware.

Previous work by Varghese et al. [12, 13] suggested a fast/slow path technique. The current paper extends the previous work [12, 13] by adding a middle path to handle non-malware packets classification. In current paper, middle path technique composed of ML and Snort sub-signature as the main features to be searched in the packet payload for accurate detection of new malware and metamorphic malware. The location of the middle-box is placed near the edge router. The present approach follows the earlier methods [12, 25] where the detection system is based on packets. The present work is the extension to [9, 25] work, by introducing the middle path technique using SVM classifier. Also the problem of the large size of n-gram is solved using feature selection method to choose only 1000 features.

The base of the proposed technique is [12] technique. The propsed technique include the fast path which contains any pattern matching algorithm such as KnuthMorris-Pratt, Aho-Corasick, and Wu-Manber [35] that scans the content of incoming packet payload to detect the presence of any sub-signature. Secondly, upon detection, the packet is sent to the middle path. The middle path is the main focus of attention (called middle box) of the present paper. It classifies the packet either as malware or as normal without reassembly. Packet classified as normal is allowed to pass in the network. However, the packet classified as malware is named critical packet and will be sent to the slow path for further analysis if it is transmission control protocol (TCP) or dropped in the middle path without being sent to the slow path if it is user datagram protocol (UDP). The slow path represents the system that can reassemble the packets which belong to the same flow.

To exam the proposed technique, for example, assumed that one of the metamorphic file is transferred over the Internet after packetizing the file to 5 packets of less than 1500 bytes. Assuming the worst case scenario, where four of these packets are classified as normal because they do not contain enough malware features. Only one packet is classified as malware. If the detected packet is not the first packet in that flow then it is necessary to rest the connection in order to allow the sender source to resend the packets of that flow. Then it can catch all these packets and send them to the slow path for more analysis. Finally, this flow is detected as malware and dropped by the slow path. Figure 1 shows the mechanism when only one packet that belongs to the flow is detected as malware packet. Simultaneously, the connection is reset, and the control bit is set in the lookup table which is included in the IPS Chip in order to keep the 5-taple and the sequence number of the detected packet then can be used to capture the resent packets which have the same 5-taple in order to be sent to the slow path for reassemble the flow of these packets.

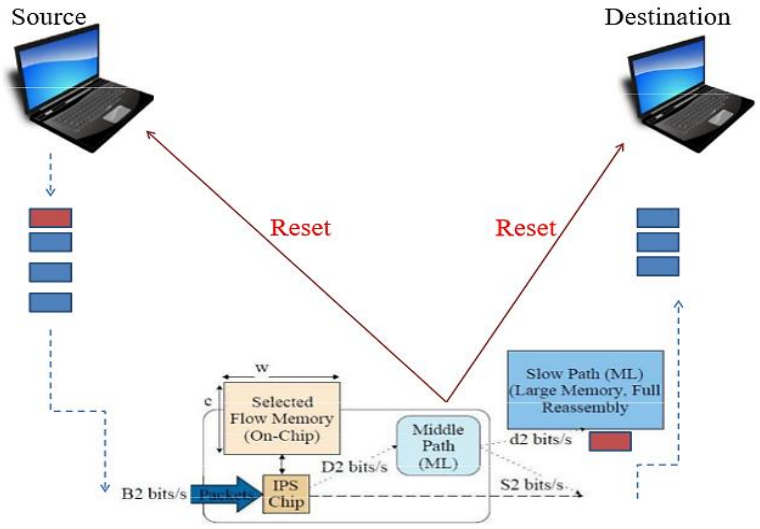


Figure 1. Mechanism of malware packet detected in the network

4. Data Collection

Two types of datasets are used in this article. The first type is a metamorphic executable files, second is real traffic traces, which are captured from UTM campus. In this study Window 10 and also Linux Ubuntu 14.04 operating system has been used. 1) Metamorphic Malware: The metamorphic files are collected from two different sources. The first group of files contained 109 files from [4]. Out of these 109 files, 50 files are from Next Generation Virus Construction Kit (NGVCK), 50 files from Second Generation (G2) virus, and 9 files from Mass Produced Code Generation Kit (MPCGEN) virus. The second group enclosed to 911 files which are generated using NGVCK kit [36] and VX Heavens website and used the same configuration setting where the total number of metamorphic files is 1020.

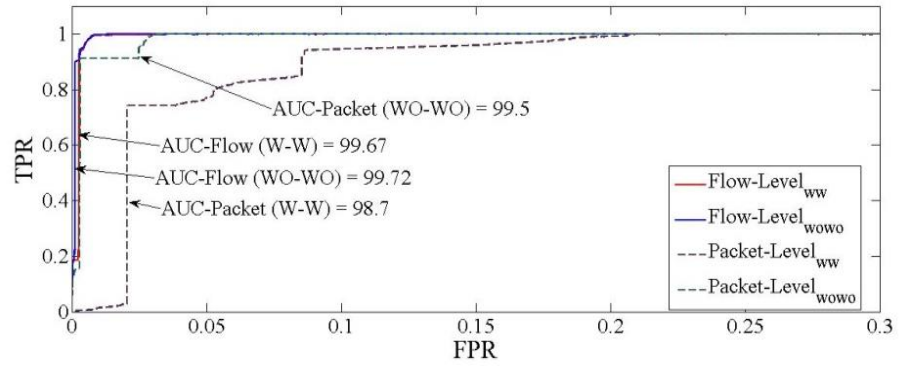
2) UTM traffic traces: Since the proposed method need as much as possible malware packets therefore, the captured traffic traces are obtained from the academic network and student network and it captured for one week. The captured traffic contained both the incoming and out-going information from the network gateway. The captured traces are processed by mirroring the traffic to servers in UTM Centre for Information and Communication Technology (CICT) and Faculty of Electrical Engineering (FKE). The traffic traces are captured using TCPdump on a Linux server and logged in packet capture (pcap). Current article focused on the TCP and UDP traffic only because most malware used these protocols especially worm [37]. For extracting the malware packets, Snort was used to trigger the captured of Snort alert and all the bad traffic.

5. Experimental Results and Analysis

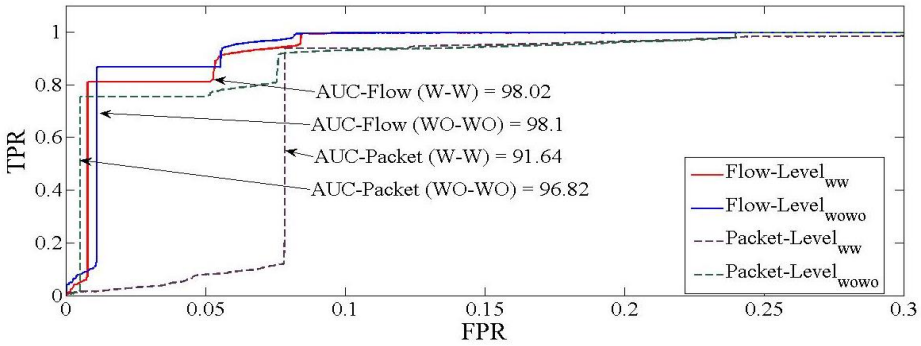
The results of the experements contain two part. First part, a comparison between flow-level and packet-level malware detection is made. Second part, the results of the proposed packet-level malware detection are compared with that of [12, 13] to measure the speedup of the proposed technique as compared to the case when all packets are subjected to ML classification. In this research, several tools are used. These include TCPDUMP, WEKA, Wireshark, IDA-Pro, Snort, and MATLAB.

5.1. The Comparison Between Fow-Level and Packet-Level Malware Detection

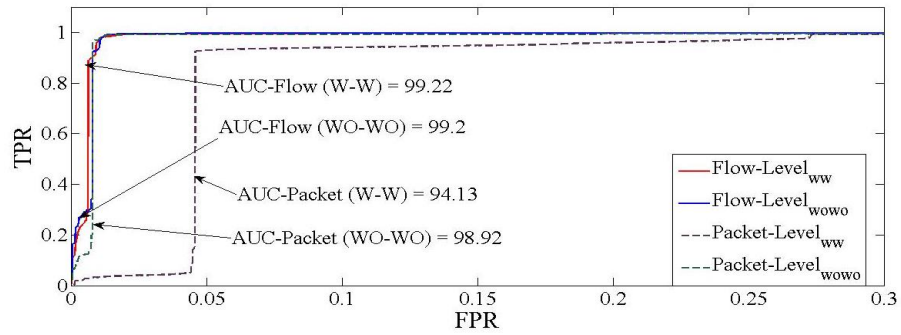
This section contains experiments of the real traffic traces captured from UTM campus in order to test the ability of the proposed technique (using one-week real traffic, one day used as training and the rest days used as testing) when metamorphic are injected or without injected them in these traffic traces for flow and packet level. Metamorphic malware are injected in the attack dataset for further validate the results as shown in Figures 2 and 3, where (W-W) means with metamorphic packets injected in training and testing dataset, (WO-WO) means without metamorphic packets injected in training and testing dataset.



(a)



(b)



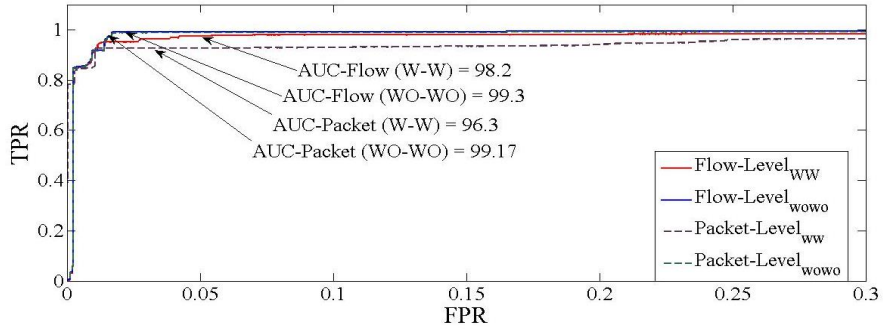
(c)

Figure 2. Comparing the flow-level and packet-level detection capacity with and without metamorphic packets for day 1, day 2, and day 3 of UTM dataset.
(a) Day 1 training and day 2 testing, (b) Day 2 training and day 3 testing,
(c) Day 3 training and day 4 testing

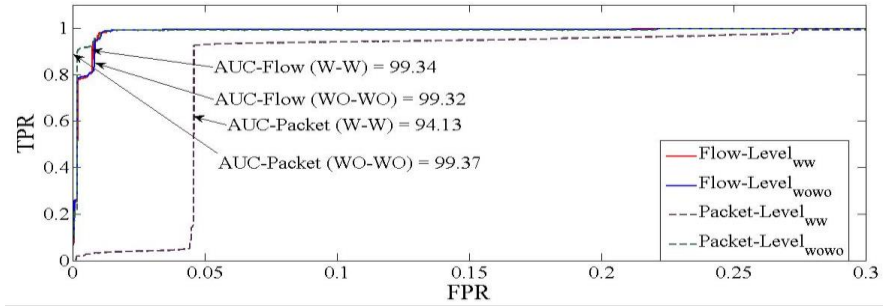
5.2. Compare with other Works

The results obtained using the proposed malware detection technique for UTM dataset is compared with [12, 13] after repeating their method. This validates the speedup of the proposed technique as well as the percentage of the malware and normal packets that are sent to the network or to the slow path. Figure 4 clearly demonstrates that the average malware packets which are detected in the middle path correctly without sending them to the slow path are more than 97%. However, when [12, 13] technique is used around 98% of malware packets need to be sent to the slow path to make the reassembly process in the packets for attaining the flow of these packets. In union, using the proposed technique more than 98% of the normal

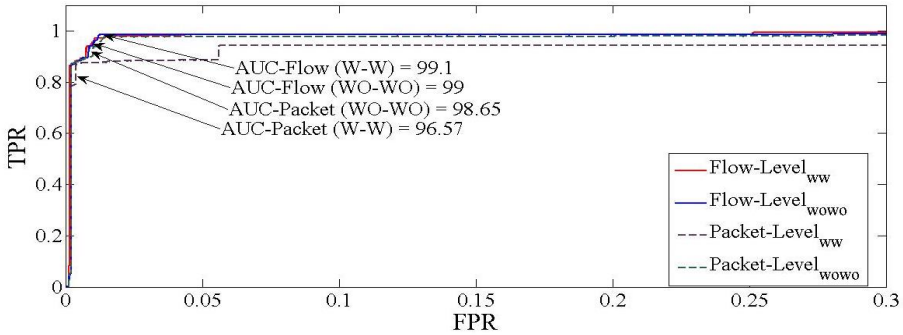
packets are sent to the network as compared to only 30% using [12, 13] technique. Figure 4 shows the comparison between the proposed technique and [12, 13] technique for the UTM dataset. More details about the number of packets that are sent to slow path according to the proposed technique is depicted in Table 1. From this table it is clear that only around 2% packets on an average are required to send to the slow path. Figure 5 shows that the proposed technique sent only 2% of the total malware and normal packets to the slow path as compared to 70% of normal and 98% of malware using [12,13] technique. This means that the proposed technique is around 37 times faster than the existing technique because of the no need to send several normal packets to the slow path. It is because the middle path can detect most of the malware packets and simultaneously can filter several normal packets.



(a)



(b)



(c)

Figure 3. Comparing the flow-level and packet-level detection capacity with and without metamorphic packets for day 4, day 5, and day 6 of UTM dataset.
(a) Day 4 training and day 5 testing, (b) Day 5 training and day 6 testing,
(c) Day 6 training and day7 testing

Table 1. The details of TCP Packets That Classified In Middle Path for UTM Traffic Traces

Day	Total malware packets	No. of malware TCP packets	No. of malware UDP packets	TCP normal packets	% Malware detected by middle path	%TCP packets sent to slow path
2	8,883	0	8,883	13,560	96.78%	0.56%
3	9,308	38	9,270	17,380	93.58%	0.88%
4	5,872	9	5,863	15,076	97.83%	2.07%
5	10,206	6	10,200	20,932	98.32%	2.20%
6	8,449	22	8,427	17,032	98.53%	2.38%
7	5,973	11	5,962	7,767	98.59%	5.41%

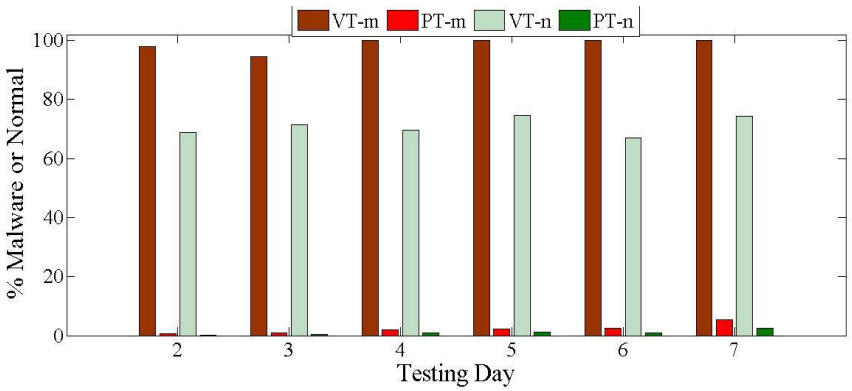
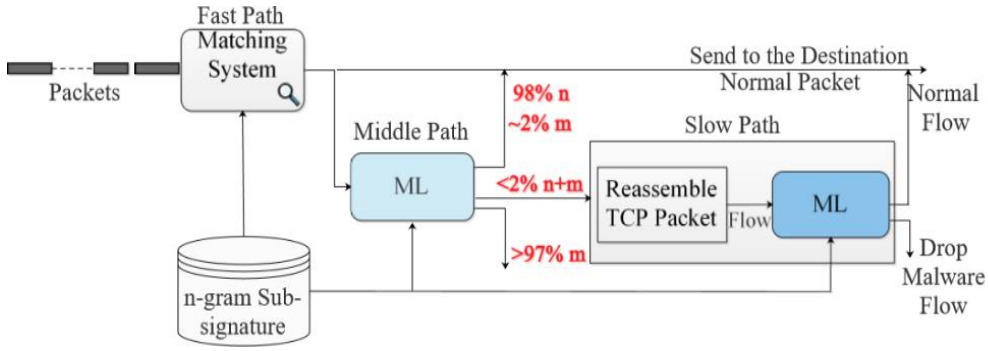
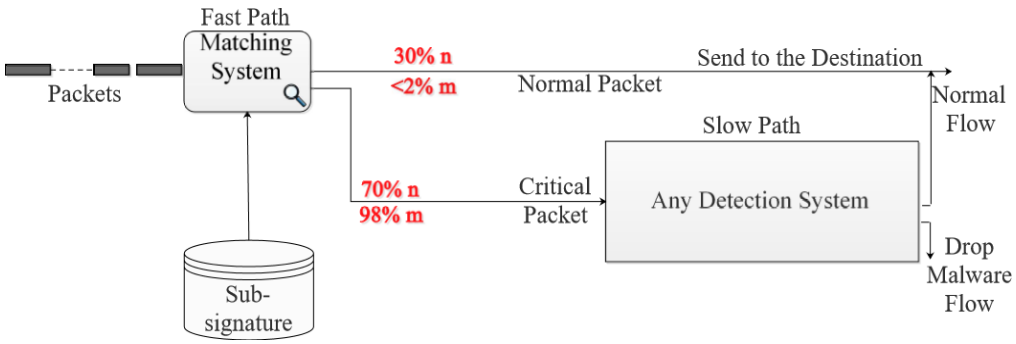


Figure 4. Percentage of all 6 days UTM dataset that are sent to slow path, where PT: proposed technique, SP: slow path, VT: [12, 13] technique, n: normal, and m: malware



(a)



(b)

Figure 5. Result comparison between the proposed technique and [12,13] technique (a). Proposed technique result, (b). Varghese et al. [12] technique result

6. Conclusion

The proposed middle path pre-filters in-transit packets to minimize the number of packets sent to the slow path, where packets are reassembled and are subjected to full malware detection. This proposed technique can speed up the transmission of several normal packets without sending them to the slow path like other method. It is verified that the proposed technique can process the normal packets much faster than Varghese et al. technique due to the non-requirement of sending many normal packets to slow path that need time for reassembling and checking these packets. Only very few number of the total malware and normal packets are sent to the slow path for further analysis. While stile can detect several new malware and some of metamorphic malware more than half of them can be detected correctly. It is affirmed that the developed method is suitable for detecting fixed and some of mutated malware in the network level before it reach the host. The higher flexibility of the present method allows building the detection in network layer after training the model on the malware packets in the middle-boxes or in flow-level at the host level on the whole flow of malware after reassembling the packets flow. It is established that the proposed method can protect networks from the spread of some types of malware from their payload. For the future works, the proposed technique can be implemented in hardware devices for real time traffic classification in FPGA.

References

[1] Symantec, *Symantec Report.* https://www.symantec.com/products/threat-protection. 2016.
[2] Boukhtouta A, et al., Network malware classification comparison using DPI and flow packet headers. *Journal of Computer Virology and Hacking Techniques*. 2016; 12(2): 69-100.
[3] Oza A, et al. HTTP attack detection using n-gram analysis. *Computers & Security*. 2014; 45: 242-254.
[4] Deshpande S, Y Park, M Stamp. Eigenvalue analysis for metamorphic detection. *Journal of Computer Virology and Hacking Techniques*. 2014; 10(1): 53-65.
[5] Attaluri S, S McGhee, M Stamp. Profile hidden Markov models and metamorphic virus detection. *Journal in computer virology.* 2009; 5(2): 151-169.
[6] Canfora G, AN Iannaccone, CA Visaggio. Static analysis for the detection of metamorphic computer viruses using repeated-instructions counting heuristics. *Journal of Computer Virology and Hacking Techniques*. 2014; 10(1): 11-27.
[7] Song F, T Touili. *Efficient malware detection using model-checking*. in International Symposium on Formal Methods. Springer. 2012.
[8] Mohammed M, A Lakhotia. A method to detect metamorphic computer viruses. *The IEEE Computer Society's Student Magazine*. 2003; 10(1): 24-36.
[9] Khammas BM, et al. Metamorphic Malware Detection Based on Support Vector Machine Classification of Malware Sub-Signatures. *TELKOMNIKA Telecommunication Computing Electronics and Control.* 2016; 14(3): 1157-1165.
[10] Yen T-F, MK Reiter. *Traffic aggregation for malware detection.* in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. 2008.
[11] Misra A, M Verma, A Sharma. *Capturing the interplay between malware and anti-malware in a computer network.* Applied Mathematics and Computation. 2014; 229: 340-349.
[12] Varghese G, JA Fingerhut, F Bonomi. Detecting evasion attacks at high speeds without reassembly. in *ACM SIGCOMM Computer Communication Review*. 2006; 36(4):327-338.
[13] Varghese G, FG Bonomi, JA Fingerhut. *Methods and systems to detect an evasion attack*. *Google Patents*. 2013.
[14] Chen Z, et al. Machine learning based mobile malware detection using highly imbalanced network traffic. *Information Sciences*. 2018; 433: 346-364.
[15] Taylor TP, et al. Methods, systems, and computer readable media for detecting malicious network traffic. *Google Patents*. 2018.
[16] Wang S, et al. A mobile malware detection method using behavior features in network traffic. *Journal of Network and Computer Applications*. 2019; 133: 15-25.
[17] Aijaz UN, et al. Malware Detection on Server using Distributed Machine Learning. Perspectives in Communication. *Embedded-systems and Signal-processing-PiCES*, 2018; 2(7): 172-175.

[18] Wang S, et al. *Deep and Broad Learning based Detection of Android Malware via Network Traffic*. in 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). 2018.
[19] Machlica L, M Sofka. Hierarchical feature extraction for malware classification in network traffic. *Google Patents*. 2019.
[20] Liu J-X, D-M Zhao, F Wang. *Networks Attack-Defense model based on the improved Lanchester equation.* 2013 International Conference on Machine Learning and Cybernetics (ICMLC). 2013.
[21] Day C. Intrusion Prevention and Detection Systems, in *Managing Information Security (Second Edition)*. Elsevier. 2014: 119-142.
[22] de Lima IVM, JA Degaspari, JBM Sobral. *Intrusion detection through artificial neural networks*. in Network Operations and Management Symposium, NOMS. 2008.
[23] de Sá Silva L, et al. Detecting attack signatures in the real network traffic with ANNIDA. *Expert Systems with Applications*. 2008; 34(4): 2326-2333.
[24] de Sa Silva L, et al. *A neural network application for attack detection in computer networks.* in Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. 2004.
[25] Ismail I, et al. Incorporating known malware signatures to classify new malware variants in network traffic*. International Journal of Network Management*. 2015; 25(6): 471-489.
[26] Hijazi AA, Network Traffic Characterization Using (p, n)-grams Packet Representation. *Carleton University*. 2014.
[27] Rieck K, P Laskov. *Detecting unknown network attacks using language models.* in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer. 2006.
[28] Ahmed I, K-s Lhee. Classification of packet contents for malware detection*. Journal in computer Virology*. 2011; 7(4): 279.
[29] Torrano-Gimenez C, et al. *Applying feature selection to payload-based web application firewalls*. in Security and Communication Networks (IWSCN), 2011 Third International Workshop on. 2011.
[30] Perdisci R, et al. McPAD: A multiple classifier system for accurate payload-based anomaly detection. *Computer networks.* 2009; 53(6): 864-881.
[31] Song Y, AD Keromytis, SJ Stolfo. Spectrogram: *A Mixture-of-Markov-Chains Model for Anomaly Detection in Web Traffic.* in NDSS. 2009.
[32] Pektaş A, T Acarman. Malware classification based on API calls and behaviour analysis. *IET Information Security*. 2017; 12(2): 107-117.
[33] Kolter JZ, MA Maloof. Learning to detect and classify malicious executables in the wild. *Journal of Machine Learning Research*. 2006; 7(Dec): 2721-2744.
[34] Khammas BM, et al. Feature selection and machine learning classification for malware detection. J*urnal Teknologi*. 2015; 77(1): 243-250.
[35] AbuHmed T, A Mohaisen, D Nyang. A survey on deep packet inspection for intrusion detection systems. *Magazine of Korea Telecommunication Society*. 2007; 24(11): 25-36.
[36] Alam S, et al. A framework for metamorphic malware analysis and real-time detection*. Computers & Security.* 2015; 48: 212-233.
[37] Li P, M Salour, X Su. *A survey of internet worm detection and containment.* IEEE Communications Surveys & Tutorials. 2008; 10(1): 20-35.