Optimal remote access Trojans detection based on network behavior

Khin Swe Yin, May Aye Khine
Faculty of Computing, University of Computer Studies, Myanmar

Article Info

Article history:

Received Jan 15, 2018
Revised Dec 18, 2018
Accepted Jan 11, 2019

Keywords:

Network behavior detection
Random forests algorithm
Remote access Trojans

ABSTRACT

RAT is one of the most infected malware in the hyper-connected world. Data is being leaked or disclosed every day because new remote access Trojans are emerging and they are used to steal confidential data from target hosts. Network behavior-based detection has been used to provide an effective detection model for Remote Access Trojans. However, there is still short comings: to detect as early as possible, some False Negative Rate and accuracy that may vary depending on ratio of normal and malicious RAT sessions. As typical network contains large amount of normal traffic and small amount of malicious traffic, the detection model was built based on the different ratio of normal and malicious sessions in previous works. At that time false negative rate is less than 2%, and it varies depending on different ratio of normal and malicious instances. An unbalanced dataset will bias the prediction model towards the more common class. In this paper, each RAT is run many times in order to capture variant behavior of a Remote Access Trojan in the early stage, and balanced instances of normal applications and Remote Access Trojans are used for detection model. Our approach achieves 99 % accuracy and 0.3% False Negative Rate by Random Forest Algorithm.

Corresponding Author:

Khin Swe Yin,
Faculty of Computing,
University of Computer Studies, Yangon,
No.4, Main Road, ShwePyiThar Township, Yangon, Myanmar.
Email: khinsweyin@ucsy.edu.mm

1. INTRODUCTION

An organization or a person suffers financial loss, reputation loss and business disruption because of data breach. As the threat of data breaches is increasing every year, security of confidential information is more important than before. One of the main reasons of occurring data breach is targeted malware attacks. Remote Access Trojans are installed on endpoints using drive-by-download, email and USB tactics. If a computer is infected with Remote Access Trojan, its' command and control traffic stealthily control the victim and steals confidential information. Advanced persistent threats gather confidential data from target hosts by planting Trojans. Firewall, intrusion detection/prevention systems and antivirus scanners are used to secure network from malicious activities. There are two detection techniques: host-based and network-based. As host-based detection system has to be installed on each host [1], [2], it has some complexity and overhead. Network based detection technique applies Deep Packet Inspection (DPI) techniques that gain a very high accuracy but they cannot detect unknown Trojans. Deep packet inspection uses regular expression (RE) matching [3].

It examines packet payload whether it is matched with any predefined regular expressions. The attack patterns or signatures of antivirus scanner and intrusion detection systems are defined from known malware. A signature is a sequence of bytes or sequence of events or sequence of system calls, etc [4], [5]. If the attack signature is slightly changed by hackers using simple obfuscation technique such as inserting noops and code re-ordering or a novel attack appears, the unseen attack will be considered as acceptable pattern and this attack will be missed. Moreover, maintenance of the signature database is an extremely tedious and time-consuming.

Although various techniques have been introduced to detect remote Access Trojans, there remains two challenges: (1) there is weakness for extracting correct information for features in the early stage and (2) false negative rate that should be taken care of, (3) overhead. When to stop and cut the traffic is very important to extract effective features. Since some features are extracted from a session that starts from SYN packet in TCP three -way handshake to FIN/ACK packet and some are obtained from a session that starts a connection to the end of the traffic, time takes long and confidential information will be leaked before detection. Network behavioral analysis has been used to classify network traffic applications and to detect malware [6], [7]. As features are extracted from the early stage that depends on packet interval time, correct information for features cannot be obtained when error-recovery feature like TCP retremission occurs. Antivirus scanners needs to be installed on each host and it needs to be updated daily. Moreover, just a simple ratio of malicious and normal applications is applied for building a model. As typical network contains approximately 99.99% of normal instances and small number of malicious instances, just a simple ratio of normal and malicious traffic instances is not enough to approach a best detection model with effective features and no overhead.

In this paper features are extracted within the first twenty packets that starts SYN of TCP three-way handshake to twentieth packets without depending on how long packet interval time takes. First twenty packets are enough to detect malicious traffic of remote access trojans in the early stage, and it can avoid error-recovery features too. In addition, RATs are run many times and their different behaviors are captured. Different ratios of normal and malicious instances are applied for analyzing detection model. Our approach reduces FNR to 0% while maintaining early stage detection. Moreover, it is easy to manage through network and we do not need to install and update application to each terminal as it is network-based approach. As a behavior-based detection, both unknown RATs and variants of known RATs can be detected without time consuming. The paper is organized as follows: literature review is summarized in Section 2, research method is presented in Section 3, Section 4 describes results and discussion, and the paper is concluded in Section 5.

2. LITREATURE REVIEW

Several techniques have been used to detect variants of malware. Features are categorized on the basis of static and dynamic analysis of program files [8]. In static analysis, the behavior of program is observed by analyzing its binary code or internal structure of files without actually executing it [9]. It is vulnerable to code obfuscation techniques. Dynamic analysis is performed by running a program. In dynamic analysis behavior of malware is monitored in emulated environment. It can deal with code evasion techniques [10].

Network behavioral analysis has been done in recent years for detecting malware. But behavioral features are different depending on when the traffic is cut or stopped to extract features. [11] uses flow level-based features and IP level-based features in order to describe Trojan network behavior accurately. At the flow level, two features – (1) duration, and (2) packet time interval are extracted. At the IP-level, 4 features- (1) number of inbound/outbound packets, (2) volume of inbound/outbound traffic, (3) duration of the communication session, and (4) number of transport layer connections. These features are extracted from the session from a SYN packet in the TCP three-way handshake and ends with a FIN/RST packet. So, it takes time, and confidential information may be leaked before detection. The accuracy is over 91% and FPR is less than 3.2%.

Five typical characteristics are used to describe malicious behavior of RATs in [12]. They are (1) ratio of send and received traffic size, (2) number of connections, (3) proportion of upload connection, (4) proportion of concurrent connection, (5) number of distinct IPs. Its' accurate detection is 97.05% and FPR is 2.94%. As its features are extracted from the start of an application's connection to the end, it may be impossible to detect RAT as fast as possible. The exact number of normal and malicious instances is not mentioned in these works. The malicious traffic of RATs can be detected in the early stage of TCP communication [13]. The early stage of a session is a packet list that starts from the SYN packet of TCP three-way handshake and ends until each packet interval time is less than the threshold t seconds. It does not take into consideration of TCP's error recovery features like TCP Retransmission that occurs in the stage of TCP handshaking. Total 175 sessions are used for classifying. 165 are normal sessions and 10 sessions are RATs'. The detection accuracy is over 96% and FNR is 10% by Random Forest algorithm.

The behavior features are mainly distributed in the network layer, transport layer and application layer [14]. A closed-loop feedback model is designed to remove some false alarms from the detected results.

The false positive results are fed back as inputs for the training sets of machine learning model. So, it takes time and much overhead. 224 data flow of Trojan traffic and 276 data flow of normal applications are used in [14]. Its detection rate is 97.7%. The unbalanced ratio of normal and malicious sessions is used in [13]-[15]. RATs are run once and captured the traffic to extract features.

A hybrid approach that combines anomaly, behavior and signature-based detection techniques is designed to detect zero-day attacks including worm, virus, and so on [16]. So, it is computationally expensive. Since it uses features like same_srv_rate that is percentage of connections to the same service in count feature, Pkt_count_legitimate_ports that is among the past 100 connections whose destination port is same to the port in the legitimate ports list, it needs time and it is not possible to detect the attacks as early as possible. The first few packets are used for early traffic classification, but it needs to use new appropriate features [17]. The summary of related works is shown in Table 1.

Table 1. Summary of Related Works

Related Works	Approach	Limitation
[8]-[10]	Reviewed malware detection Signature based and behavior based detection Static and dynamic analysis	Signature based approach cannot detect unknown malware Much false positive rate in behavior based detection
[11]	Network behavior based technique. It uses flow-level and IP-level features, e.g. duration, transport layer connection	It takes 5 minutes to terminate sessions in this work RATs will be detected after they stay long in the victim machine
[12]	Application network behavior based approach Examples of features- number of connection, proportion upload of connection,	Complex to obtain features and to manage It may detect RATs after their long stay in the victim host
[13]	It detects RATs in the early stage of network traffic Early stage is defined depending on packet interval time A simple ratio of RATs and normal applications is used (10 instances of RATs and 175 instances of normal applications)	It does not consider error recovery features- TCP retransmission- and RATs will be missed Just a simple ratio of instances is not enough to obtain best detection model
[14]	Features are distributed in three layers: network layer, transport layer and application layer e.g. many sub-connections during primary connection, communications time False Positive Rate is fed back as input to adjust the training model 224 instances of Trojans and 276 instances of normal applications are applied	Complex to extract and obtain features Overhead Just a simple ratio of instances is used for detection model
[16]	A hybrid approach for zero days attacks	It may not be early stage detection
[17]	Early traffic clcassification	It needs effective features

The ability of a host to retransmit packets is one of TCP’s most fundamental error-recovery features in Wireshark. When a packet is sent, but the recipient has not sent a TCP ACK packet back, the transmitting host assumes that the original packet was lost and retransmits the original packet [18]. It is called TCP Retransmission. If TCP retransmission occurs, packet interval time takes more time than usual. Moreover, this kind of situation often occurs in the steps of TCP three-way handshake before establishing connection for transferring data. It is often found in the traffic of remote access Trojans, and if the stopping way for extracting features is packet interval time, correct information for features cannot be obtained in this state. The previous works do not take into consideration this situation and some use packet interval time for defining the early stage and extracting features. In this paper, each RAT is run many times and different behavior of RATs during the first twenty packets are captured and, both balanced and unbalanced sessions are used for building a detection model in order to detect RATs in the early stage and to avoid the bias problem of unbalanced dataset.

3. RESEARCH METHOD

Our method consists of three main phases: Feature Extraction, Training and Detection. After collecting network traffic, features are extracted for each session and labelled, and then these sessions are trained with three supervised machine learning algorithms. Then, the detection model is obtained and used to test with a real session. Our work mainly focuses on Feature Extraction.

3.1. Pre processing

Wireshark is used to capture network traffic traces. The network traces are filtered by TCP protocol. Two different IP addresses that there is interaction between them are chosen and the traces are cut for the first twenty packets that starts from SYN of TCP three-way handshake to the twentieth packet. Then the traces are divided into sessions, and then sessions are labelled.

3.2. Feature extraction

How to extract features for the first twenty packets is shown in Figure 1. Firstly, basic 10 features are initialized, and then the values of basic features are collected until the number of packets is equal to 20. Next, 4 features are calculated depending on the value of the basic features. Finally, 7 features are chosen to generate a feature vector for a session. The selected 7 features are described in detail in Table 2. Comparison of features is shown in Table 3.

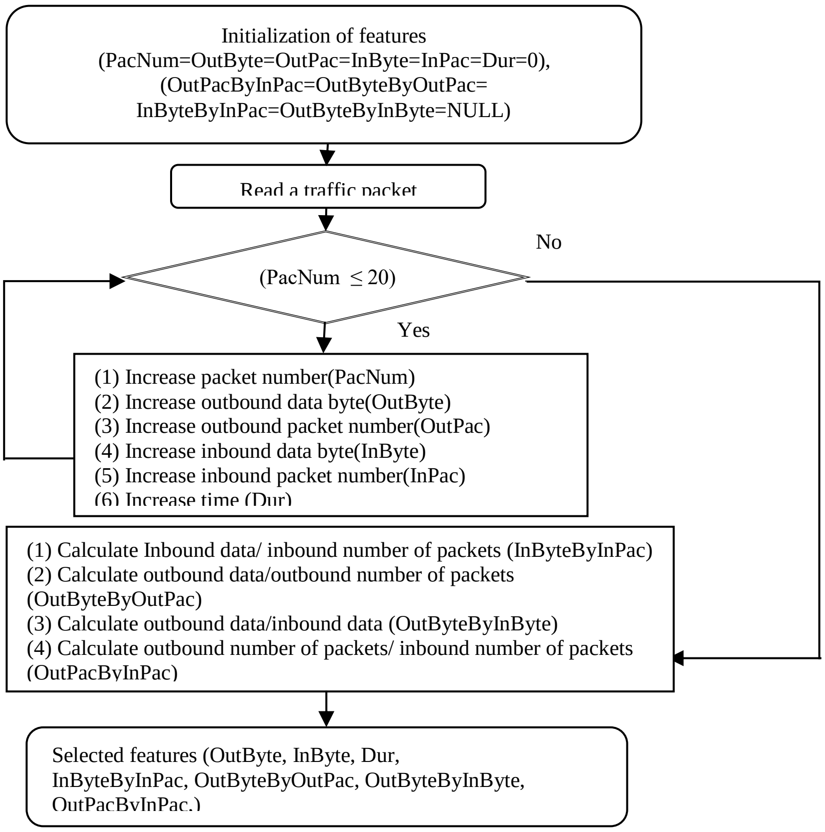
Initialization of features
(PacNum=OutByte=OutPac=InByte=InPac=Dur=0),
(OutPacByInPac=OutByteByOutPac=
InByteByInPac=OutByteByInByte=NULL)
Read a traffic packet
(PacNum ≤ 20)
No
Yes
(1) Increase packet number(PacNum)
(2) Increase outbound data byte(OutByte)
(3) Increase outbound packet number(OutPac)
(4) Increase inbound data byte(InByte)
(5) Increase inbound packet number(InPac)
(6) Increase time (Dur)
(1) Calculate Inbound data/ inbound number of packets (InByteByInPac)
(2) Calculate outbound data/outbound number of packets (OutByteByOutPac)
(3) Calculate outbound data/inbound data (OutByteByInByte)
(4) Calculate outbound number of packets/ inbound number of packets (OutPacByInPac)
Selected features (OutByte, InByte, Dur, InByteByInPac, OutByteByOutPac, OutByteByInByte, OutPacByInPac.)

Figure 1. Process of feature extraction

Table 2. Selected Features

No	Feature	Description
1	Outbyte	Outbound data byte
2	Inbyte	Inbound data byte
3	InByteByInPac	rate of Inbound data Byte/ Inbound number of packets
4	OutByteByOutPac	rate of Outbound data byte/Outbound number of packets
5	Duration	duration from the first packet to twentieth packets
6	OutByteByInByte	ratio of Outbound data byte/Inbound data byte
7	OutPacByInPac	ratio of outbound number of packets/ inbound number of packets

Table 3. Comparison of the Selected Features in Early 20 Packets

Features	Type	Trend
OutByte	N	91.33% are more than 500 bytes
	R	67% are more than 500 bytes
InByte	N	99.667 % are more than 200 bytes
	R	11% are more than 200 bytes
InByteByInPac	N	99.667 % are more than 20 bytes
	R	10.333% are more than 20 bytes
OutByteByOutPac	N	87.333% are more than 50 bytes
	R	67.333% are more than 50 bytes
Duration	N	33.33% take more than 10 seconds
	R	73.667% take more than 10 seconds
OutByteByInByte	N	28% are more than 1
	R	100 % are more than 1
OutPacByInPac	N	33.333% are more than 1
	R	28.667 % are more than 1

N: Normal Application, R: RAT

3.3. Learning with machine learning algorithms

Weka, datamining tool is used to load datasets and three machine learning algorithms are used for building detection model. Three machine learning algorithms used in the experiment are Decision Trees (DT), Random Forests (RF) and Naïve Bayes (NB).

3.3.1. Decision trees

In decision trees, the process is broken down into individual tests which begin at the root node and traverse the tree, depending on the result of the test in that particular node. The tree begins at the root node. From the root node the tree branches or forks out to internal nodes. The decision to split is made by impurity measures [19].

3.3.2. Random forest

Random forest is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees. The method combines Breiman's "bagging" idea and the random selection of features and it improves prediction accuracy [20].

3.3.3. Naïve bayes

Naive Bayes is a widely used classification method based on Bayes theory. Based on class conditional density estimation and class prior probability, the posterior class probability of a test data point can be derived and the test data will be assigned to the class with the maximum posterior class probability [21]. Calculating the conditional probability as follows:

$$P(h/D) = \frac{P(D/h)P(h)}{P(D)} \tag{1}$$

3.4. Evaluation

k-fold Cross Validation is used to validate the result of classification in the experiment. Accuracy, False Negative Rate (FNR) and False Positive Rate (FPR) are used for evaluation. Accuracy gives the correctly classified number of both normal and malicious instances on total instances. FPR expresses that the incorrectly classified number of normal instances on the total normal instances. FNR shows that the incorrectly classified number of malicious RAT instances on the total RAT instances. The less FNR while maintaining high accuracy, the better the detection system is for not missing malicious sessions. How to calculate Accuracy, FPR and FNR is shown below:

$$Accuracy = \frac{Correctly\ Classified\ Number\ of\ normal\ and\ RAT\ Instances}{Total\ Number\ of\ normal\ and\ RAT\ Instances} \tag{2}$$

$$FNR = \frac{Incorrectly\ Classified\ Number\ of\ RAT\ Instances}{Total\ Numberof\ RAT\ Instances} \tag{3}$$

$$FPR = \frac{Incorrectly\ Classified\ Number\ of\ Normal\ Instances}{Total\ Number\ of\ Normal\ Instances} \tag{4}$$

4. RESULTS AND DISCUSSION

A virtual environment that attackers and victims are running is set up. The attacker is a place where RAT is executed, and the victim is a place where the attacker's server.exe is executed. 10 types of Remote Access Trojans and 10 normal applications are used in the experiment. Wireshark is run on the victim to capture and collect network traffic. The most widely used RATs are applied in the experiment. Normal applications include cloud services, p2p download tools, browsers and social services that most of people use in Internet today. RATs and normal applications used in the experiment are shown in Table 4.

Table 4. RATs and Normal Applications used in the Experiment

No	RATs	Normal applications
1	ImminentMonitor	Dropbox
2	KilerRat	Pcloud
3	NjRat	Skype
4	Cerberus	YahooMessenger
5	Xtreme	Facebook
6	Pandora	Bittorrent
7	CyberGate	BitComet
8	SpyGate	Google
9	Xena	Firefox
10	Babylon	Chrome

Network behavior features for a session are extracted from the trace that starts from a SYN packet in the TCP three-way handshake and ends at twentieth packets. It is the very first time traffic that collects after the victim is infected by RAT. If this RAT is not detected and removed from victim's computer, it always connects back to the attacker. It is a considerable situation because the attacker may stay as long as possible to control the victim. When a RAT is run next time after system reboots, it always sends connection back to the attacker. In our experiment, each RAT is run many times in order to capture the variant behavior of RAT. In this way the number of malicious sessions is also increased without using sampling method and the balanced normal and malicious sessions are obtained for building a detection model. Different ratios of normal and malicious instances, and their results are shown in Table 5.

Table 5. Results of Naïve Bayes (NB), Decision Trees (DT), Random Forests (RF)

Ratio of normal and		NB			DT			RF	
malicious instances	Acc	FNR	FPR	Acc	FNR	FPR	Acc	FNR	FPR
N150-RAT10	0.963	0.6	0	0.981	0.2	0.007	0.988	0.1	0.007
N300-RAT10	0.971	0.7	0.007	0.99	0.2	0.003	0.99	0.2	0.003
N300-RAT300	0.878	0.217	0.027	0.992	0.003	0.013	0.993	0.003	0.01

Acc: Accuracy, FNR: False Negative Rate, FPR: False Positive Rate

The classification methods are Naïve Bayes, Decision Trees and Random Forests. 15 sessions from each normal application and 1 session from each RAT are collected and, 150 normal instances and 10 RAT instances are used for building a model. Next, 300 normal sessions and 10 RATs sessions are applied for detection model. Then, 300 normal sessions from 10 normal applications and 300 RAT sessions from 10 RATs are applied to build a model.

In Naïve Bayes, accuracy is high but FNR is 0.6 and 0.7 while using unbalanced ratios. Although it's FNR reduces to 0.217 with balanced instances, its accuracy decreases to 0.878. DT has a slight difference in accuracy although different ratios of instances are used for classification. DT has FNR -0.2 when unbalanced ratios of instances are used. But its FNR is reduced to 0.003 when balanced instances are classified. The accuracy of RF does not change much with both balanced and unbalanced instances. The False Negative Rate of RF is fluctuared. It is 0.1 when 150 normal and 10 RAT instances are classified. It increases to 0.2 with 300 normal and 10 malicious instances. But it decreases to 0.003 when the balanced instances are used.

Among three algorithms- Naïve bayes, Decision Trees and Random Forests- DT and RF maintain high accuracy for different ratios of instances. FNR and FPR of DT and RF are the least among these algorithms. RF is slightly better than DT. They are suitable algorithms for detecting RATs. They can reduce FNR to 0.003 while maintaining high accuracy 0.99 when they use balanced ratio of normal instances and malicious instances. So optimal detection model with best accuracy, least FNR and least FPR is obtained by Random Forest algorithm.

A performance comparison of two approaches is shown in Table 6. The first approach depends on packet interval time [13] and the second one is first twenty packets which is our proposed approach. 300 normal instances and 10 RATs instances are classified by random forest. The detection that depends on packet interval time gets 97% accuracy and 0.6 FNR. The detection during first twenty packets obtains 99% accuracy and 0.2 FNR.

Table 6. A Performance Comparison of Two Detection Approaches

Detection Approaches	Random Forest		
	Accurcy	FNR	FPR
Early stage detection that depends on packet interval time	0.977	0.6	0.003
Detection during first twenty packets	0.99	0.2	0.003

5. CONCLUSION

In this paper, the idea of extracting network behavioral features in the early twenty packets for detecting Remote Access Trojans is proposed and implemented. The behavior of Remote Access Trojans is different from normal applications in the early twenty packets of network traffic. Different ratios of normal and malicious instances are used for building detection model. One session for each RAT is not enough to build a best model, and running RAT many times is the best for describing variant behaviors of RATs and increasing malicious instances in order to build optimal detection model. Random Forest algorithm is the best detection model since its accuracy is 99.3%, its FNR is 0.003 and its FPR is 0.01. Thus, it helps to reduce data leakage and increase the security of confidential information. This approach relies on the network behavior features that uses TCP protocol. Future work will be to increase the number of RAT samples and normal applications in order to achieve comparable classification accuracy, FPR and FNR for detection system of RATs in production environments with effective feature set and no overhead.

REFERENCES

[1] B. Cogswell and M. Russinovich, “Rootkitrevealer,” Rootkit detection tool by Microsoft, vo1. 71, 2006.

[2] Y. Wang, *et al.*, “Detecting stealth software with strider ghostbuster,” *Dependable Systems and Networks, DSN 2005. Proceedings. International Conference on. IEEE*, pp. 368-377, 2005.

[3] S. Prithi1, *et al*., “A Survey on Intrusion Detection System using Deep Packet Inspection for Regular Expression Matching,” *International Journal of Electronics, Electrical and Computational System*, vol/issue: 6(1), 2017.

[4] T. S. Chou, “Ensemble Fuzzy Belief Intrusion Detection Design,” Thesis, Florida International University, 2007.

[5] R. R. Patel and C. S. Thaker, “Zero-Day attack signatures detection using honeypot,” *IJCA Proceedings on International Conference on Computer Communication and Networks CSI-COMNET-2011 comnet (1)*, pp. 66-71, 2011.

[6] T. Nguyen and G. Armitage, “A survey of techniques for internet traffic classification using machine learning,” *Communications Surveys & Tutorials, IEEE*, vol/issue: 10(4), pp. 56-76, 2008.

[7] T. Yen and M. Reiter, “Traffic aggregation for malware detection,” *Proc. International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 207-227, 2008.

[8] S. Ranveer and S. Hiray, “Comparative Analysis of Feature Extraction Methods of Malware Detection,” *International Journal of Computer Applications (0975 8887)*, vol/issue: 120(5), 2015.

[9] I. A. Saeed, *et al.*, “A Survey on Malware and Malware Detection Systems,” *International Journal of Computer Applications*, vol/issue: 67(16), 2013.

[10] E. Gandotra, *et al*., “Malware Analysis and Classification: A Survey, Department of Computer Science and Engineering, PEC University of Technology, Chandigarh,” *India Journal of Information Security*, vol. 5, pp. 56-64, 2014.

[11] S. Li, *et al.*, “A General Framework of Trojan Communication Detection Based on Network Traces,” *IEEE Seventh International Conference on Networking, Architecture, and Storage*, pp. 49-58, 2012.

[12] Y. Liang, *et al.*, “An Unknown Trojan Detection Method Based on Software Network Behavior,” *Wuhan University Journal of Natural Sciences*, vol/issue: 18(5), pp. 369-376, 2013.

[13] D. Jiang and K. Omote, “A RAT Detection Method Based on Network Behavior of the Communication’s Early Stage,” *The Institute of Electronic, Information and Communication Engineers (IEICE) Trans.Fundamental*, vol/issue: E99-A(1), 2016.

[14] W. Jinlong, *et al*., “Closed-loop Feedback Trojan Detection TechniqueBased on Hierarchical Model,” *Proceedings of Joint International Mechanical, Electronic and Information Technology Conference, Chongqing, China*, 2015.

[15] K. S. Yin and M. A. Khine, “Network Behavioral Features for Detecting Remote Access Trojans in the Early Stage,” *Proceedings of the 2017 VI International Conference on Network, Communication and Computing*, pp. 92-96, 2017.

[16] R. Kaur and M. Singh, “A Hybrid real-time zero-day attack detection and analysis system,” *I.J. Computer Network and Information Security*, pp. 19-31, 2015.

[17] B. Qu, *et al.*, "On accuracy of early traffic classification," *IEEE Seventh International Conference on Networking, Architecture, and Storage*, pp. 348-354, 2012.

[18] C. Sanders, "Practical Packet Analysis using Wireshark to solve real-world Network Problems," 2nd Edition, pp. 165-168, 2011.

[19] T. M. Mitchell, "Machine Learning, Decision Trees Learning," pp. 52-76, 1997.

[20] L. E. O. Breiman, "Random forests," *Machine Learning*, vol/issue: 45(1), pp. 5-32, 2001.

[21] T. M. Mitchell, "Machine Learning, Bayesian Learning," pp. 154-178, 1997.

BIOGRAPHIES OF AUTHORS

Khin Swe Yin received M.C.Sc in Computer Science from University of Computer Studies, Mandalay(UCSM) in 2009. She is a PhD candidate in University of Computer, Studies, Yangon(UCSY). Her research interest includes Network Security and Machine Learning.

May Aye Khine received M.I.Sc and Ph.D. degrees in information technology from the University of Computer Studies, Yangon, Myanmar in 1999 and 2004, respectively. She is currently a full professor from faculty of computing in the University of Computer Studies, Yangon, Myanmar. Her main research interests include Big Data Analytics, Mathematical Modeling especially in Computational Methods and Data Science.