International Journal of Advanced Smart Convergence Vol.12 No.1 64-69 (2023)
http://dx.doi.org/10.7236/IJASC.2023.12.1.64

IJASC 23-1-8

SPRT-based Collaboration Construction for Malware Detection in IoT

Jun-Won Ho

Professor, Department of Information Security, Seoul Women's University, South Korea
jwho@swu.ac.kr

Abstract

We devise a collaboration construction method based on the SPRT (Sequential Probability Ratio Test) for malware detection in IoT. In our method, high-end IoT nodes having capable of detecting malware and generating malware signatures harness the SPRT to give a reward of malware signatures to low-end IoT nodes providing useful data for malware detection in IoT. We evaluate our proposed method through simulation. Our simulation results indicate that the number of malware signatures provided for collaboration is varied in accordance with the threshold for fraction of useful data.

Keywords: *Sequential Probability Ratio Test (SPRT), Malware, Collaboration, IoT*

1. Introduction

We consider IoT consisting of high-end nodes and low-end nodes in terms of detection capability against malware. High-end IoT nodes can perform malware detection task and malware signature generation task. On the other hand, low-end IoT nodes cannot do these tasks of high-end IoT nodes, but collect data and send it to high-end devices in order to aid malware signature generation. High-end IoT nodes provide a reward of malware signatures to low-end IoT nodes making a contribution of useful data to them, where useful (resp. useless) data is defined as data that can (resp. cannot) be used for malware signature generation. In this IoT comprising of high-end and low-end nodes, it is imperative to build up efficient and effective collaboration between low-end and high-end IoT nodes.

To meet this necessity, we propose a method utilizing the Sequential Probability Ratio Test (SPRT) [1] for collaboration construction between low-end and high-end IoT nodes. The key idea of our proposed collaboration method is as follows: High-end IoT nodes leverage the SPRT to provide the more malware signatures to low-end IoT nodes contributing the more useful data for malware signature generation to them.

We evaluate our proposed method through simulation. Our simulation results represent that the higher threshold for fraction of useful data leads to the lower number of malware signatures provided to low-end nodes by high-end nodes.

Manuscript Received: December. 28, 2022 / Revised: January. 2, 2023 / Accepted: January. 5, 2023
Corresponding Author: jwho@swu.ac.kr
Tel: +82-2-970-5607, Fax: +82-2-970-5981
Professor, Department of Information Security, Seoul Women's University, South Korea

2. Related Work

As far as the relevant work is concerned, we consider the research work related to zero-day malware [2-4] and the one related to evasive malware [5]. Cooperation construction method based on Shapley value and probabilistic approach for zero-day malware detection in IoT is devised in [2]. In [3], support vector machine is applied for zero-day malware classification. In [4], generative adversarial networks is utilized to detect analogous zero-day malware. The relevant work to evasive malware is explored in [5-10].

3. SPRT-based Collaboration Setup for Malware Detection among IoT nodes

For the SPRT-based cooperation construction for malware detection among IoT nodes, we harness the intuition that the more useful data is sent to high-end IoT nodes by low-end IoT nodes, the more rewards of malware signatures are provided to low-end IoT nodes by high-end IoT nodes. This intuition is reasonable in the sense that useful data is highly likely utilized for generation of worthy malware signatures.

We assume that the entire time consists of a series of time slots such that the SPRT is run in the end of each time slot. Additionally, we define the fraction of useful data as the number of useful data over the total number of data, where the total number of data is summation of the number of useless data and the number of useful data. The fraction of useful data is calculated in the end of each time slot.

Additionally, we configure threshold for the fraction of useful data, which is a threshold value applied for the SPRT. Note that the entities performing the SPRT are high-end IoT nodes. In the SPRT, we define a null hypothesis as a hypothesis that it is not time for high-end IoT node to send a certain number of malware signatures to low-end IoT node. We also define an alternate hypothesis as a hypothesis that it is time for high-end IoT node to send a certain number of malware signatures to low-end IoT node. Moreover, we assume that each sample in the SPRT is deemed as independent and identically distributed Bernoulli random variable, where BD is denoted as a success probability in Bernoulli distribution. Note that H (resp. G) is a user-configured false-negative rate (resp. user-configured false-positive rate). Both BD_0 and BD_1 are pre-planned parameters such that $BD_0 < BD_1$. The configuration of BD >= BD_1 (resp. BD <= BD_0) will increase the chance with which the SPRT chooses an alternate (resp. a null) hypothesis.

The specific procedure of the SPRT is as follows:

(1) UW (resp. UZ) is a variable utilized to compute the number of samples (resp. the number of samples with type of alternate hypothesis) in the SPRT. Initialize the variables UW and UZ as follows.
UW=UZ=0;

(2) Whenever the end of each time slot is reached, the fraction of useful data is calculated. The fraction of useful data is regarded as a sample in the SPRT. If the fraction of useful data is smaller than threshold for the fraction of useful data, perform runSPRT(0) procedure

(3) Otherwise, perform runSPRT(1) procedure

(4) The specific procedure of runSPRT(type) is as follows:
Increment UW by 1;
If type == 1, then Increment UZ by 1;
Variable LV (resp. HV) is a threshold used for accepting null (resp. alternate) hypothesis. LV and HV are calculated as follows:

$$LV = \frac{\ln \frac{H}{1-G} + UW \ln \frac{1-BD_0}{1-BD_1}}{\ln \frac{BD_1}{BD_0} - \ln \frac{1-BD_1}{1-BD_0}} \quad , \quad HV = \frac{\ln \frac{1-H}{G} + UW \ln \frac{1-BD_0}{1-BD_1}}{\ln \frac{BD_1}{BD_0} - \ln \frac{1-BD_1}{1-BD_0}}$$

If UZ <= LV, then UW=UZ=0; Take null hypothesis;
If UZ >= HV, then UW=UZ=0; Take alternate hypothesis; return 1;
return 0;

4. Performance Evaluation

In order to evaluate our developed method, we write a simple simulation program considering a situation where it is assumed that a low-end IoT node sends both useful and useless data to a high-end IoT node which adapts the SPRT with the fraction of useful data sent by a low-end IoT node in order to decide when to transmit malware signatures to a low-end IoT node.

We set G=H=0.01. We consider two configurations of (BD_0, BD_1)= (0.3, 0.7), (0.1, 0.9). We also set the number of time slots to 100, time slot size to 10. Thus, our proposed method is simulated for 100 time slots such that a time slot size is 10. We configure the number of signatures assumed to be sent to low-end IoT node by high-end IoT node once the SPRT reaches a decision of accepting an alternate hypothesis to 100. Moreover, we consider two configuration sets of threshold for fraction of useful data, (0.86, 0.87, 0.88, 0.89, 0.9), (0.972, 0.974, 0.976, 0.978, 0.98). We consider the occurrence of error that useful (resp. useless) data is misjudged as useless (resp. useful) data and set such error probability to 0.01.

Additionally, we adopt Poisson traffic model for both useless and useful data traffic assumed to be sent to high-end IoT nodes by low-end IoT nodes. Hence, inter-arrival time of useful (resp. useless) data follows exponential distribution with rate parameter of χ_f (resp. χ_l). We set $\chi_l = 1$ and consider two configurations of $\chi_f = 3, \chi_f = 5$.

The simulation is repeated 100 times and we present average results of simulation results. As displayed in Figures 1,2,3,4, we discern that the number of malware signatures decreases as threshold for fraction of useful data increases. These observations imply that a rise in threshold for fraction of useful data leads to a reduction in the times that the SPRT accepts an alternate hypothesis, contributing to the smaller number of malware signatures sent to low-end IoT node by high-end IoT node. We also recognize that the number of malware signatures in case of (BD_0, BD_1)=(0.1, 0.9) is higher than the one in case of (BD_0, BD_1)=(0.3, 0.7) under the same configuration of χ_f.

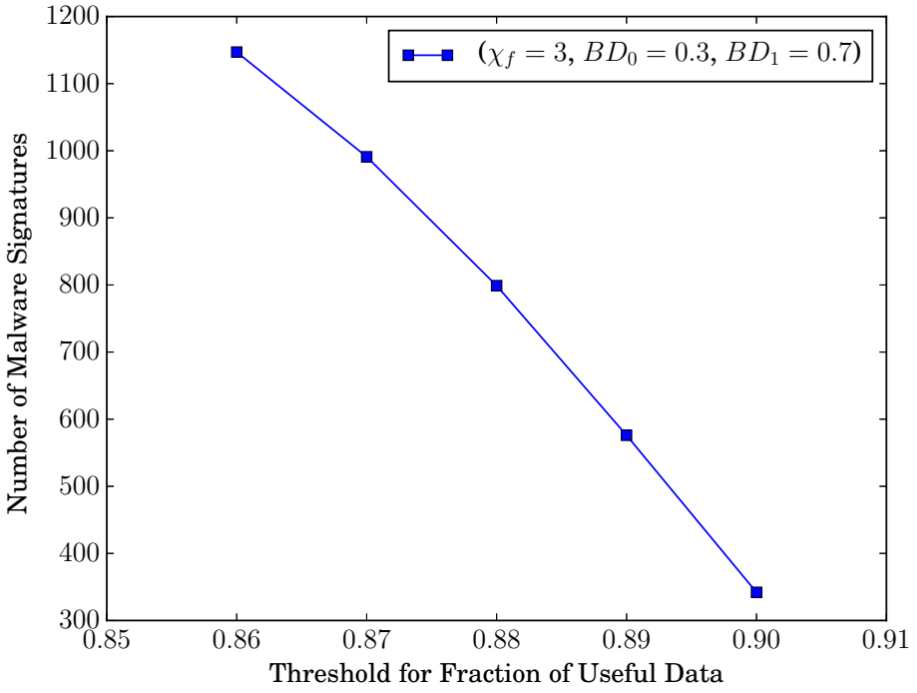

Figure 1. Effect of threshold for fraction of useful data on number of malware signatures when $\chi_f = 3$, BD_0=0.3, BD_1=0.7.

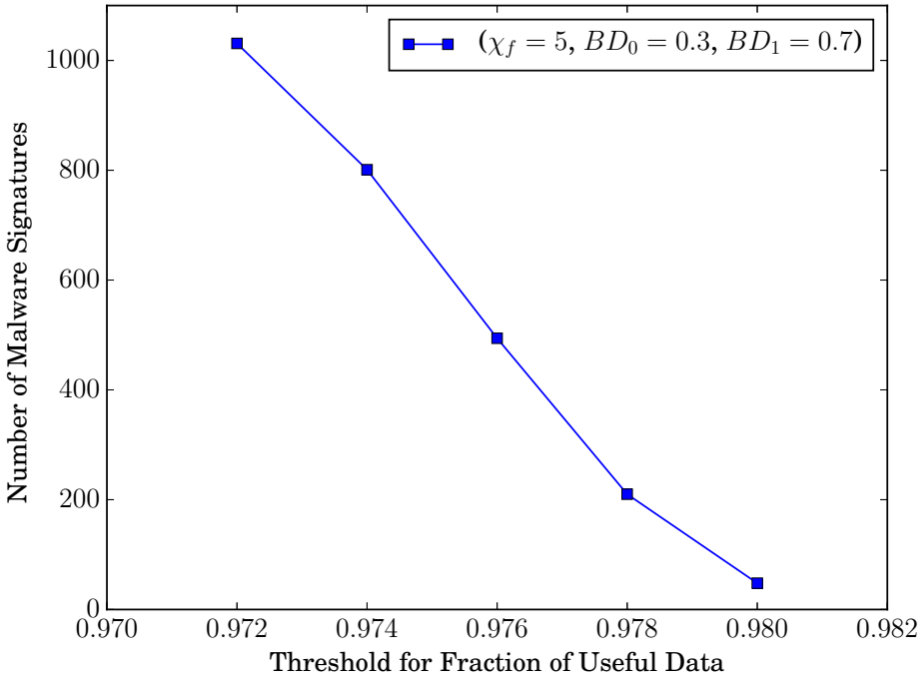

Figure 2. Effect of threshold for fraction of useful data on number of malware signatures when $\chi_f = 5$, BD_0=0.3, BD_1=0.7.

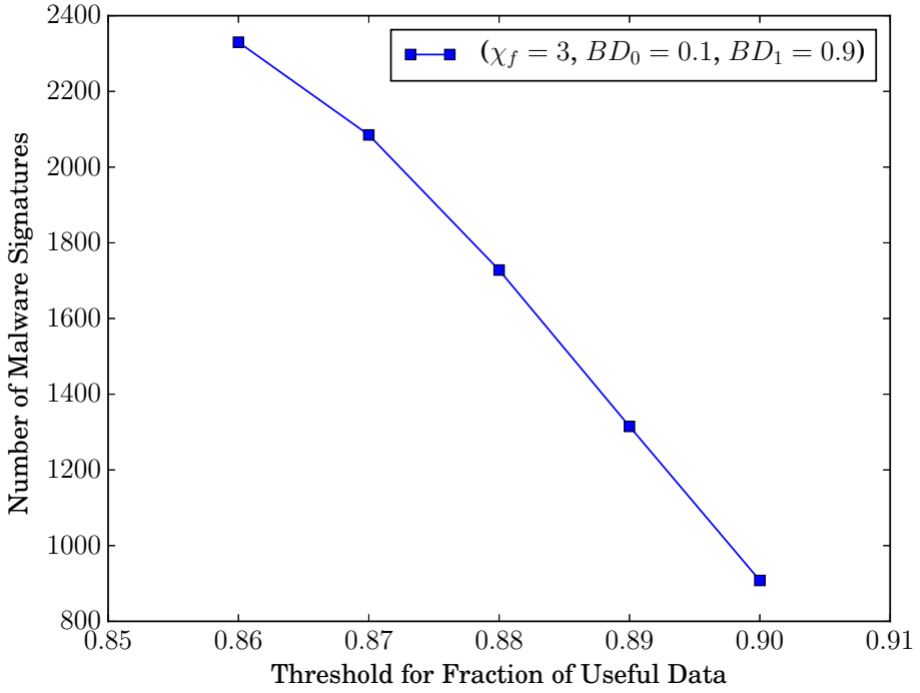

Figure 3. Effect of threshold for fraction of useful data on number of malware signatures when $\chi_f = 3$, BD_0=0.1, BD_1=0.9.

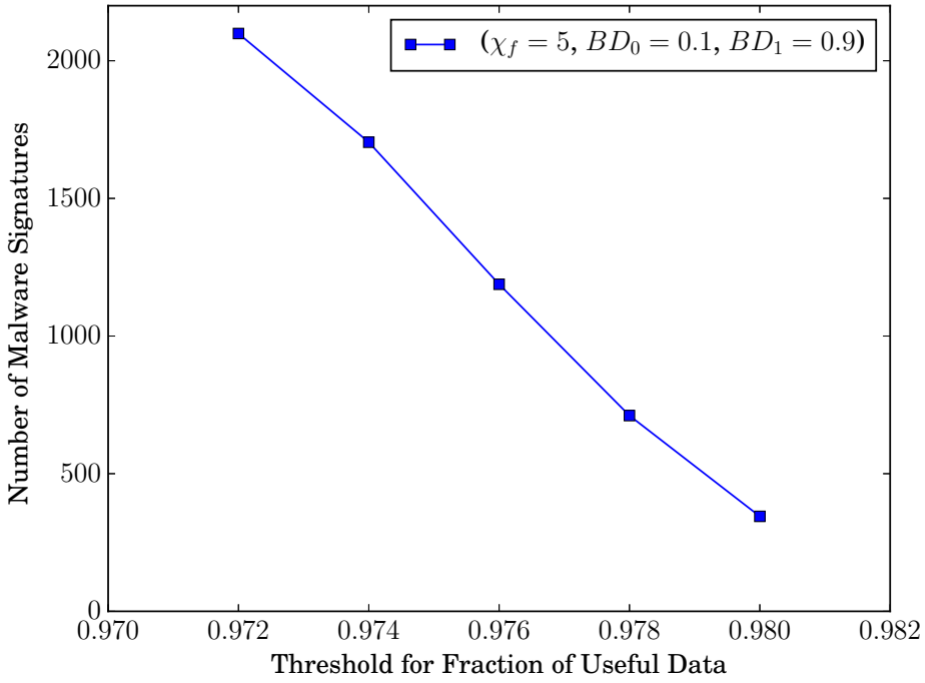

Figure 4. Effect of threshold for fraction of useful data on number of malware signatures when $\chi_f = 5$, BD_0=0.1, BD_1=0.9.

5. Conclusion

In this paper, we develop the SPRT-based collaboration method between high-end and low-end nodes for malware detection in IoT. We perform evaluation of our proposed method through simulation. Our evaluation results reveal that the larger threshold for fraction of useful data contributes to the smaller number of malware signatures sent to low-end IoT nodes by high-end IoT nodes.

Acknowledgement

This work was supported by a research grant from Seoul Women's University(2023-0004).

References

[1] A. Wald. Sequential Analysis, Dover, 2004.

[2] Ho, Jun-Won. Interactive method and apparatus for defending against zero-day malware. Republic of Korea Patent.Registration Number/Date: 10-2022-0157667 (2022.11.22).

[3] R. El-Sayed, A. El-Ghamry, T. Gaber and A. E. Hassanien, "Zero-Day Malware Classification Using Deep Features with Support Vector Machines," *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, Egypt, 2021, pp. 311-317, DOI: https://doi.org/10.1109/ICICIS52592.2021.9694256.

[4] D. -O. Won, Y. -N. Jang and S. -W. Lee, "PlausMal-GAN: Plausible Malware Training Based on Generative Adversarial Networks for Analogous Zero-day Malware Detection," in *IEEE Transactions on Emerging Topics in Computing*, DOI: https://10.1109/TETC.2022.3170544.

[5] D. C. DElia, E. Coppa, F. Palmaro, and L. Cavallaro. "On the Dissection of Evasive Malware," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2750-2765, 2020. DOI: https://doi.org/10.1109/TIFS.2020.2976559.

[6] J. Zhang, Z. Gu, J. Jang, D. Kirat, M. Stoecklin, X. Shu, and
H. Huang. Scarecrow: Deactivating Evasive Malware via Its Own Evasive Logic. In50th Annual IEEE/IFIP Internation al Conference on Dependable Systems and Networks (DSN), 2020, pp. 76-87.

[7] W. Diao, X. Liu, Z. Li, and K. Zhang. Evading Android Runtime Analysis Through Detecting Programmed Interacti ons. In ACM WiSec, 2016. DOI: https://doi.org/10.1145/2939918.2939926.

[8] N. Miramirkhani, M. P. Appini, N. Nikiforakis, and M. Polychronakis. Spotless Sandboxes: Evading Malware Anal ysis Systems using Wear-and-Tear Artifacts. *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 1009-1024, 2017. DOI: https://doi.org/ 10.1109/SP.2017.42.

[9] D. Kirat, G. Vigna, and C. Kruegel. BareCloud: Bare-metal Analysis
based Evasive Malware Detection. In Usenix Security, 2014.

[10] L. Bello and M. Pistoia. Ares: Triggering Payload of Evasive Android Malware. In IEEE/ACM 5th International C onference on Mobile Software Engineering and Systems (MOBILESoft), pp. 2-12, 2018.