# Work-in-Progress: MicroArchitectural Events and Image Processing-based Hybrid Approach for Robust Malware Detection

Sanket Shukla, Gaurav Kolhe, Sai Manoj P D and Setareh Rafatirad
George Mason University, Fairfax, VA, USA
{sshukla4,gkolhe,spudukot,srafatir}@gmu.edu

## ABSTRACT

To thwart the detection of malware through traditional and emerging approaches, malware development has seen a paradigm of embedding the malware into benign applications. This calls for a localized feature extraction scheme for detecting stealthy malware with more robustness. To address this challenge, we introduce a hybrid approach which utilizes the microarchitectural traces obtained through on-chip embedded hardware performance counters (HPCs) and the application binary for malware detection. The obtained HPCs are fed to multi-stage machine learning (ML) classifier for detecting and classifying the malware. To overcome the challenge of detecting the stealthy malware, image processing based approach is applied in parallel. In this approach, the malware binaries are converted into images, which is further converted into sequences and fed to recurrent neural networks to recognize patterns of stealthy malware. Based on the localized patterns, sequence classification is further applied to perform binary classification and further discover the variation of the identified malware family. Our proposed framework exhibits high resilience to popular obfuscation techniques such as code relocation.

**ACM Reference Format:**
Sanket Shukla, Gaurav Kolhe, Sai Manoj P D and Setareh Rafatirad. 2019. Work-in-Progress: MicroArchitectural Events and Image Processing-based Hybrid Approach for Robust Malware Detection. In *2019 International Conference on Compliers, Architectures and Synthesis for Embedded Systems Companion (CASES'19 Companion), October 13–18, 2019, New York, NY, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3349569.3351538

## 1 INTRODUCTION

Security challenges and countermeasures in IoT and other embedded computing devices are becoming non-trivial. Among multiple security threats, malware is a critical threat due to relatively less complexity to design and implant and propagate into device(s) [4]. Malicious Software, generally known as 'malware' is a software program or application developed by an attacker to gain unintended access to the computing device in order to perform unauthorized accesses as well as malicious activities such as stealing data, sensitive information such as credentials, contaminating, spying and manipulating the stored information without users consent. To alleviate the threats and meet the IoT device constraints, a comprehensive, lightweight and robust malware detection technique is of dire need.

Traditional and primitive software-based malware detection techniques such as signature-based and semantics-based anomaly detection techniques exist for more than two decades [1], though effective, induces remarkable computational and processing overheads and is inefficient to detect unseen threats [2]. To overcome the limitations of the software-based malware detection approaches, the work in [3, 6] proposed using the microarchitectural event traces captured through on-chip hardware performance counters (HPCs)[1] fed to machine learning (ML) classifiers for classifying benign and malware applications. Despite the better performance compared to the software-based approaches, hardware-assisted malware detection (HMD) fails to detect stealthy malware[2][8] efficiently.

Additionally, with the advancements in computer vision and image processing hardware accelerators [9], researchers exploited utilizing computer vision strategies for malware detection. To perform this, works such as [5, 9] converts malware binaries into images. Based on such image analysis, an accuracy of 98% is reported in [5]. In [9] authors utilize a single-channel lightweight convolutional neural network (CNN) to efficiently detect IoT malware. However, the above malware detection methods fails when a malware application is embedded into benign i.e. stealthy malware. To address the above discussed challenges, this work presents a novel approach of efficiently detecting stealthy malware during runtime.

The proposed detection technique uses a fork-and-integrate approach. In the first branch of the fork, the HPC traces of a given application are collected during runtime and is validated through a two-stage ML classifier for malware classification. In parallel, the images of application binaries are converted into a sequence of symbols, and is further processed by RNN model. The rationale behind utilizing RNN is to exploit the temporal as well as spatial dependencies that attackers utilize to craft stealthy malware.

## 2 PROPOSED MALWARE DETECTION

Our proposed hybrid malware classification comprises of two paths: a) multi-stage HPC-based ML classifier and b) computer vision based classification as shown in Figure 1. The HPC-based ML classifier utilizes the obtained HPC values when executing an application and fed to the ML classifiers. During the training phase, once the HPCs are extracted, principal component analysis is performed for feature reduction and to address the limited number of HPCs

[1] HPCs are a set of special purpose registers which are assembled in new generation microprocessors to track the hardware related events for any running piece of software.

[2] Stealthy malware is a malware which is embedded inside a benign application, thereby making it complex to detect with traditional approaches as well as HMD and image processing approaches.

*CASES'19 Companion, October 13–18, 2019, New York, NY, USA*
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6925-1/19/10…$15.00
https://doi.org/10.1145/3349569.3351538

available on the chip. These reduced features collected at runtime are provided as input to ML classifiers to classify a given application as benign or malware. If binary classifier labels the input as malware, then the HPCs are fed to a multinomial logistic regression classifier to determine the malware class (backdoor, rootkit, trojan, virus and worm) with higher confidence. This HPC based stage approach is fast, robust and accurate in detecting and classifying the malware. Despite the benefits, this approach did not yield similar performance on stealthy malware due to contamination of HPC when malware is embedded into the benign application. To address this, a computer vision based approach is adopted in parallel.

In the computer vision based approach, the application binary is converted into a gray scale image for localized feature extraction. A raster scanning is performed on the converted binary images to find the image patterns (Each pattern is of 32 × 32 block size). We utilize a Cosine similarity to distinguish between multiple patterns i.e., if the cosine similarity of two patterns is higher than threshold (0.75 in this work based on conducted experiments), they are considered to be same. When more than one matched patterns are found, the one with the highest cosine similarity is considered. It needs to be noted that the pattern matching for an incoming binary is performed with the patterns in the database (created through similar process during training phase, offline). Once the image patterns are recognized for a given binary file, the whole image binary is converted into a sequence of patterns (Each pattern is provided with a unique ID). This sequence is fed to a recurrent neural network (RNN). RNN requires the input sequences of equal length but in our case size of sequences can vary according to size of gray scale images created from binary. Therefore, to zovercome this issue we pad 0's to sequence to generate sequences of equal length. As the pattern or sub-pattern of a malware cannot alter despite embedding the malware, this technique can detect the stealthy malware with higher efficacy (i.e. with 94% accuracy). This RNN model is finally used to classify and detect the incoming stealthy malware binary and predict the corresponding class label.

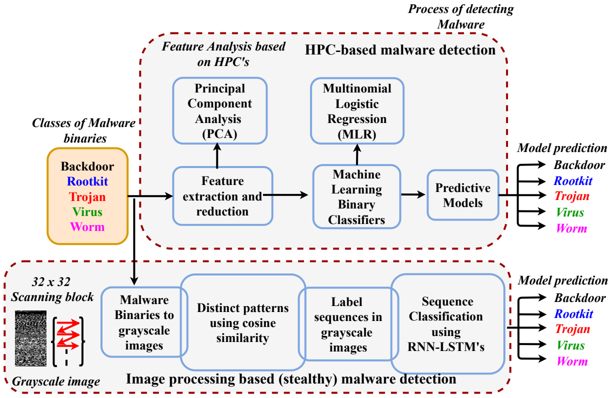
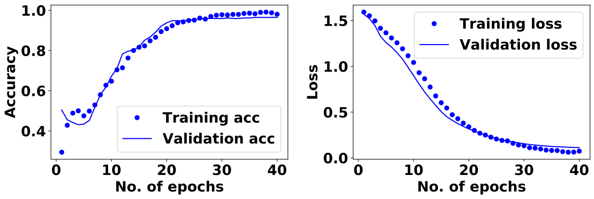

Process of detecting Malware
Feature Analysis based on HPC's
HPC-based malware detection
Principal Component Analysis (PCA)
Multinomial Logistic Regression (MLR)
Classes of Malware binaries
Backdoor
Rootkit
Trojan
Virus
Worm
Feature extraction and reduction
Machine Learning Binary Classifiers
Predictive Models
Model prediction
Backdoor
Rootkit
Trojan
Virus
Worm
32 x 32 Scanning block
Malware Binaries to grayscale images
Distinct patterns using cosine similarity
Label sequences in grayscale images
Sequence Classification using RNN-LSTM's
Model prediction
Backdoor
Rootkit
Trojan
Virus
Worm
Grayscale image
Image processing based (stealthy) malware detection

**Figure 1: Proposed hybrid approach for detecting stealthy malware**

## 3 EXPERIMENTAL RESULTS

The proposed methodology is implemented on a Intel core i7-8750H CPU with 16GB RAM. We have obtained malware applications from VirusTotal [7] with 2300 malware samples that encompasses of 5 malware classes: backdoor, rootkit, trojan, virus and worm. Further, we utilizes benign applications such as documents (.pdf, .txt, .docx) inside which the binaries of above mentioned malware classes are integrated through code obfuscation (code relocation) process to create 2300 stealthy malware samples.

Figure 2 depicts the loss and accuracy for the overall malware dataset (malware and stealthy malware). As seen from Figure 2 (a) and 2 (b) with the increase in epochs the accuracy increases and saturates at 40 epochs, hence we considered the model trained with 40 epochs for final evaluation. For the normal malware i.e., malware spawned as separate thread, an accuracy of nearly 90% is achieved with HPC-based malware detection. However, for stealthy malware, the HPC-based malware detection accuracy plummets to 54% on an average (These individual results are not plotted for the purpose of brevity). However, with the proposed hybrid approach, an accuracy of 94% is achieved, with a loss of 0.14 as shown in Figure 2a and 2b, respectively despite code obfuscation.



**(a) Validation accuracy** **(b) Validation loss**

**Figure 2: Performance of proposed hybrid approach for malware detection**

## 4 CONCLUSION AND FUTURE WORK

In this work, we propose a hybrid approach of utilizing architectural (trace) as well as code properties obtained through HPCs and image conversion, respectively for malware detection. The proposed approach can efficiently detect the malware (spawned as separate thread and stealthy) with an average accuracy of 94%. Despite the achieved performance benefits, one of the major caveats of the proposed approach is the involved latency when performing image-based stealthy malware detection, regardless of leveraging multiple-threads. To overcome this limitation, we are exploring the FPGA implementation and GPU based processing to further accelerate the detection.

## REFERENCES

[1] Brasser and et al. 2018. Advances and Throwbacks in Hardware-assisted Security: Special Session. In *Int. Conf. on Compilers, Architecture and Synthesis for Embedded Systems (CASES '18)*.
[2] Q. Chen and R. A. Bridges. 2017. Automated Behavioral Analysis of Malware: A Case Study of WannaCry Ransomware. In *IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*.
[3] Dinakarrao and et al. 2019. Adversarial Attack on Microarchitectural Events Based Malware Detectors. In *Design Automation Conf.*
[4] Zhen Ling and et al. 2018. IoT Security: An End-to-End View and Case Study. *CoRR* (2018).
[5] L. Nataraj and et al. 2011. Malware Images: Visualization and Automatic Classification. In *Int. Symposium on Visualization for Cyber Security*.
[6] Hossein Sayadi and et al. 2018. Ensemble Learning for Effective Run-Time Hardware-Based Malware Detection: A Comprehensive Analysis and Classification. *Design Automation Conf.(DAC)* (2018).
[7] Gaurav Sood. 2017. virustotal: R Client for the virustotal API. (2017).
[8] Salvatore J. Stolfo and et al. 2007. Towards Stealthy Malware Detection. In *Malware Detection*.
[9] J. Su and et al. 2018. Lightweight Classification of IoT Malware Based on Image Recognition. In *IEEE Annual Computer Software and Applications Conf.*