Enhancing Hardware Malware Detectors' Security through Voltage Over-scaling

Md Shohidul Islam*§, Ihsen Alouani‡, Khaled N. Khasawneh*
*ECE Dept., George Mason University, Fairfax, VA, USA
§CSE Dept., Dhaka University of Engineering & Technology, Gazipur, Bangladesh
‡IEMN CNRS-UMR8520, Universite Polytechnique Hauts-De-France, Valenciennes, France
Email: {mislam20, kkhasawn}@gmu.edu, ihsen.alouani@uphf.fr

I. Introduction

Computing systems are under continuous attacks by increasingly motivated and sophisticated adversaries. These attackers exploit vulnerabilities to compromise systems and deploy malware. Although significant effort continues to be directed at making systems more resilient to attacks, the number of exploitable vulnerabilities is overwhelming. While preventing compromise is difficult, signature based static analysis techniques can be easily bypassed using metamorphic/polymorphic malware or zero-day exploits since their signatures have not yet been encountered. On the other hand, dynamic detection techniques can detect unseen signatures since they monitor the behavior of the program. However, the complexity and difficulty of continuous dynamic monitoring have traditionally limited its use due to constrained resources.

Against this backdrop, several research studies proposed using Hardware Malware Detectors (HMDs) to make the continuous dynamic monitoring resource-efficient through hardware support. Specifically, HMDs are machine learning classifiers that use low-level hardware features such as instructions traces, memory access patterns, etc. and classify malware as a computational anomaly. HMDs can offer a significant advantage to defend against malware attacks because they can be 'always on' with small-to-no impact on performance. It appears that the industry started to show interest in using HMDs too; SnapDragon processor from Qualcomm appears to be using hardware features to detect malware, but the technical details are not published [4].

As HMDs showed potential defense effectiveness, it is natural to expect that attackers attempt to find adaptive ways to evade detection. As a consequence, it was shown that attackers can adapt malware to continue to operate while avoiding detection by HMDs [3]. We address the challenge of defending HMDs against evasive malware by utilizing approximate computing (AC). In particular, we propose V-HMDs, which are HMDs that uses voltage over-scaling (VOS) for evasion resilience purpose; it induces stochastic computations in HMD's model during inference, resulting in V-HMDs that are resilient to adversarial evasion attack.

II. The Threat Model & Dataset

This section explains how the attackers generate adversarial evasive malware that can bypass HMDs' detection. In particular, these attacks assume black-box access to the HMDs; the attacker can query the victim HMD with input and observe the output without knowing the victim HMD model. Therefore, the attack consists of two steps: (1) Reverse-engineering the victim HMD to create a proxy model and (2) Developing evasive malware based on the proxy model to bypass detection while preserving the malware's intended functionality.

Dataset. Our dataset consist of 600 benign programs and 3000 malware (downloaded from the Zoo malware database). The collected features are based on the frequency of executed instruction categories during run-time (similar to [3]).

Non-secure HMD. The non-secure HMD is a multi-layer perception (MLP) neural network, which consist of 1 hidden layer with 50 neurons (similar to the number of features), and Rectified Linear Unit (ReLU) as an activation function.

Reverse engineering. In black-box attacks, the attacker have access only to the input/output of the HMD and has no information about its internal architecture, e.g., structure, weights, hyperparameters, or training data. However, the HMD internal model is necessary for the attacker to be able to develop evasive malware methodically; otherwise, the problem becomes NP-Hard. Therefore, in such a setting, the attacker utilizes the observed inputs/outputs of the victim HMD to reverse-engineer it and train a proxy model [3]. After training a proxy model, we then evaluated the effectiveness of reveres-engineering. Our result shows that we were able to effectively reverse-engineer the *non-secure HMD*, with less than **1%** error. This result demonstrates that current HMDs can be reverse-engineered effectively.

Developing evasive malware. After reverse-engineering the HMD, the attacker's goal is to utilize the knowledge of the proxy model, i.e., reverse-engineered HMD, to systematically create evasive malware. In particular, the attacker have to start by identifying the features that can be used to create evasive malware and then embed the identified features in the malware without changing their intended functionality.

Firstly, to identify the features that can be used to create evasive malware, we employ a slightly modified version of the Fast-Gradient Sign Method (FGSM), which is widely employed in image processing [2]. After identifying the features, e.g., instructions, we need to systematically embed them in the malware binaries. Since we did not have access the malware source code and the malware were obfuscated, we follow

a similar approach to [3]. In particular, for each malware program, we constructed a Dynamic Control Flow Graph (DCFG) using Intel's Pin tool. Then we added instructions, identified from the previous step, to each basic block of the DCFG. Furthermore, when we add each instruction, we ensure that they do not affect the state of the program to preserve the malware's intended functionality. For example, if we are introducing an `add` instruction, we add *zero* to any register.

III. Proposed Approach: V-HMD

In this section, we propose a new class of adversarial evasion resilient HMDs (V-HMDs). V-HMDs impact the inference computation of an HMD, making its decision boundaries stochastic over time. Thus, they prevent the adversary from having reliable access to the HMD's output (reverse-engineering attacks) and reduce the transferability of evasive malware built using the victim HMD's exact model. Specifically, V-HMDs exploit AC, which is a computing paradigm that can trade energy consumption and computing time with the accuracy of results. Traditionally, this is a significant advantage for error-resilient applications (such as deep/machine learning, big data analytics, and signal processing) with respect to performance, power efficiency, flexibility, and cost. However, in this work, we reveal an additional advantage of using AC for a machine learning application, which is security. In this paper, we use a circuit level approximation, specifically VOS [1] as a practical source of randomness to harden HMDs against adversarial attacks. Our goal is to intentionally cause random timing violations driven by the supply voltage level.

IV. Experimental Evaluation

Resilience against black-box attacks: We examine V-HMDs resilience against black-box attacks. V-HMD is simply an over-scaled non-secure HMD (Section II). We assume two black-box attack scenarios: (1) attacker knows the V-HMDs training data, and (2) attacker does not know the V-HMDs training data. Note that the first scenario assumes a stronger attacker than the second scenario since the attacker will use the same data distribution that the victim is trained on.

Figure 1 shows the black-box attack (reverse-engineering) effectiveness while increasing the VOS-induced computational faults rate. The results show that using a V-HMDs with a 0.1 fault rate makes the black-box attack substantially more difficult; the reverse-engineering effectiveness using MLP drops from 99.1% to 75.5% (around 24% drop) when using the attacker training set (not the victim's training set) and from 99.2% to 86.0% (around 13.3% drop) when using the victim training set. Furthermore, the results show that the V-HMDs resilience to black-box attacks increases by increasing the computational faults rate, irrespective of the machine learning algorithm used to perform the attack. As seen from the results, reverse-engineering attacks become harder with VOS.

Detection accuracy: Figure 2 shows the V-HMDs detection accuracy, false positive rate, and false negative rate while increasing the computational faults rate (scaling the voltage). We repeated this experiment 50 times, to obtain representative

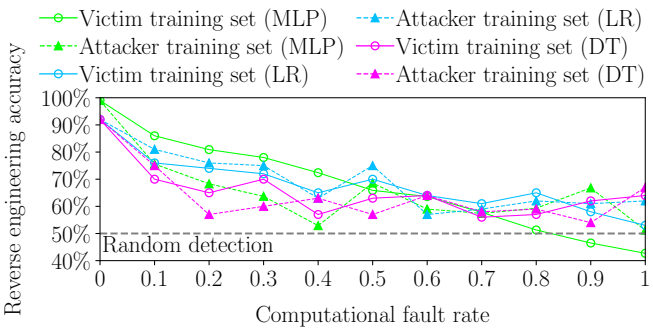

Fig. 1: V-HMDs resilience against black-box attacks (reverse-engineering): reverse-engineering effectiveness while increasing the computational faults rate

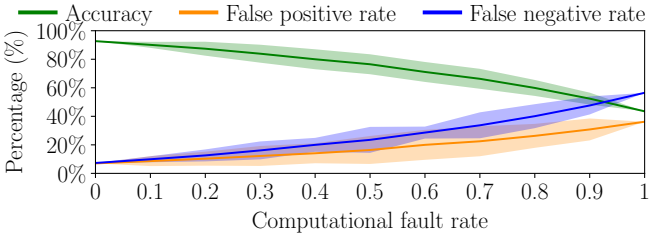

Fig. 2: VOS effect on HMD's detection performance

results, and presented mean (solid line) and standard deviation (shaded region). An interesting observation is that the standard deviation increases while increasing the VOS until a 0.5 computational faults rate, and then it starts decreasing. Notice that the standard deviation represents the stochasticity that VOS adds to the output due to the non-deterministic decision boundaries. Figure 2 also shows that the accuracy degradation diverges logarithmically as the computational faults rate approaches 1; the relationship is not linear. The same observation also applies to the false positive rate (increases logarithmically as the computational faults rate approaches 1). This is a strong advantage from the defender perspective since adding more computational faults (specifically, until 0.5 computational faults rate) would not significantly impact detection accuracy loss. For example, at 10% computational faults rate, the detection accuracy of V-HMDs drops by around 2% only. Furthermore, increasing the computational faults rate from 0.1 to 0.4 (4× increase in computational faults) would result in only 0.1× detection accuracy loss.

References

[1] V. K. Chippa, D. Mohapatra, K. Roy, S. T. Chakradhar, and A. Raghunathan, "Scalable effort hardware design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[3] K. N. Khasawneh, N. Abu-Ghazaleh, D. Ponomarev, and L. Yu, "Rhmd: evasion-resilient hardware malware detectors," in *MICRO*, 2017.

[4] "Qualcomm smart protect technology," 2016, last Accessed July 2016 from https://www.qualcomm.com/products/snapdragon/security/smart-protect.