

Unsupervised Anomaly-Based Malware Detection Using Hardware Features

Adrian Tang, Simha Sethumadhavan, and Salvatore J. Stolfo

Columbia University, New York, USA
{atang,simha,sal}@cs.columbia.edu

Abstract. Recent works have shown promise in detecting malware programs based on their dynamic microarchitectural execution patterns. Compared to higher-level features like OS and application observables, these microarchitectural features are efficient to audit and harder for adversaries to control directly in evasion attacks. These data can be collected at low overheads using widely available hardware performance counters (HPC) in modern processors. In this work, we advance the use of hardware supported lower-level features to detecting malware exploitation in an anomaly-based detector. This allows us to detect a wider range of malware, even zero days. As we show empirically, the microarchitectural characteristics of benign programs are noisy, and the deviations exhibited by malware exploits are minute. We demonstrate that with careful selection and extraction of the features combined with unsupervised machine learning, we can build baseline models of benign program execution and use these profiles to detect deviations that occur as a result of malware exploitation. We show that detection of real-world exploitation of popular programs such as IE and Adobe PDF Reader on a Windows/x86 platform works well in practice. We also examine the limits and challenges in implementing this approach in face of a sophisticated adversary attempting to evade anomaly-based detection. The proposed detector is complementary to previously proposed signature-based detectors and can be used together to improve security.

Keywords: Hardware Performance Counter, Malware Detection.

1 Introduction

Malware infections have plagued organizations and users for years, and are growing stealthier and increasing in number by the day. In response to this trend, defenders have created commercial antivirus (AV) protections, and are actively researching better ways to detect malware. An emerging and promising approach to detect malware is to build detectors in hardware [3]. The idea is to use information easily available in hardware (typically via HPC) to detect malware. It has been argued that hardware malware schemes are desirable for two reasons: first, unlike software malware solutions that aim to protect vulnerable software with equally vulnerable software¹, hardware systems protect vulnerable software with

¹ Software AV systems roughly have the same bug defect density as regular software.

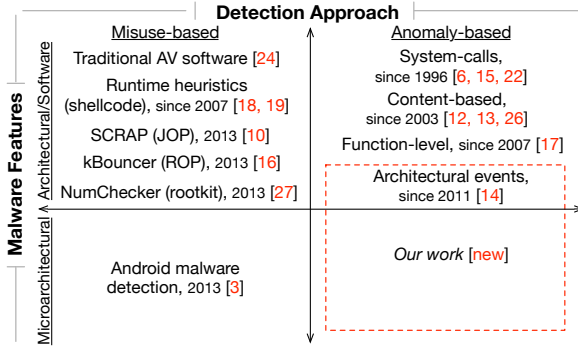


Fig. 1. Taxonomy of malware detection approaches and some example works

robust hardware implementations that have lower bug defect density because of their simplicity. Second, while a motivated adversary can evade either defense, evasion is harder in a system that utilizes hardware features. The intuition is that the attacker does not have the same degree of control over lower-level hardware features as she has with software ones. For instance, it is easier to change system calls or file names than induce cache misses or branch misprediction in a precise way across a range of time scales while exploiting the system.

In this paper we introduce techniques to advance the use of lower-level microarchitectural features in the anomaly-based detection of malware exploits. Existing malware detection techniques can be classified along two dimensions: *detection approach* and the *malware features* they target, as presented in Figure 1. Detection approaches are traditionally categorized into misuse-based and anomaly-based detection. Misuse-based detection flags malware using pre-identified attack signatures or heuristics. It can be highly accurate against known attacks but can be easily evaded with slight modifications that deviate from the signatures. On the other hand, anomaly-based detection characterizes baseline models of normalcy state and identifies attacks based on deviations from these models. Besides known attacks, it can potentially identify novel ones. There are a range of features that can be used for detection: until 2013, they were OS and application-level observables such as system calls and network traffic. Since then, lower-level features closer to hardware such as microarchitectural events have been used for malware detection. Shown in Figure 1, we examine for the first time, the feasibility and limits of anomaly-based malware detection using both architectural and low-level microarchitectural features available from HPCs.

Prior misuse-based research that uses microarchitectural features such as [3] focuses on flagging Android malicious apps by detecting payloads. A key distinction between our work and prior work is *when* the malware is detected. Malware infection typically comprises two stages, exploitation and take-over. In the exploitation stage, an adversary exercises a bug in the victim program to hijack control of the program execution. Exploitation is then followed by more elaborate take-over procedures to run a malicious payload such as a keylogger.