

WiP: Towards a Dynamic, Hardware-Agnostic Ransomware Detection System Using Hardware Performance Counters

Joshua Byun^[0009-0004-4470-3960], Aidan Guiney^[0009-0003-8821-5224], Ryan Binder^[0009-0004-8983-522X], Will Thrall^[0009-0003-4235-9871], Ian Roessle^[0009-0004-1746-7245], Dane Brown^[0000-0002-7235-548X], T. Owens Walker, III^[0000-0001-9709-5090], and Jennie E. Hill^[0000-0003-0674-154X]

United States Naval Academy, Annapolis MD 21412, USA
{m270870,m272508,m260534,m266366,roessle,dabrown,owalker,jehill}@usna.edu

Abstract. Many researchers have leveraged Hardware Performance Counters (HPCs) to dynamically detect ransomware execution. However, the vast majority of this work has only been trained and tested on HPC data from virtual machines. Recent studies suggest that virtualized HPC data may be less effective at detecting ransomware on bare-metal systems, and the efficacy of bare-metal HPC data for virtualized systems has not been widely studied. This work investigates the generalizability of HPC-based ransomware classifiers across bare-metal and virtual machines. We formalize a research hypothesis contending that an environment-agnostic classifier can be achieved by aggregating multi-target HPC data to decouple malicious signals from platform-specific noise. Our ongoing work implements an evaluation strategy utilizing a symmetric cross-environment testing matrix to address three core research questions regarding environment generalizability, transfer learning, and signal decoupling. The potential impact of this work is a more architecture-portable, platform-agnostic HPC-based ransomware classifier capable of maintaining high detection fidelity across disparate real-world environments.

Keywords: Hardware trace mechanisms for security · Ransomware · Hardware Performance Counters · Virtual Machine Introspection · Dynamic Analysis · Machine Learning

1 Introduction

Ransomware remains a premier global cybersecurity threat, with 2026 costs estimated to exceed \$57 billion and average breach damages surpassing \$5 million [7]. While traditional signature-based detection often fails against evolving encryption tactics, researchers have turned to Hardware Performance Counters (HPCs) to identify low-level architectural signatures of malicious execution. However, a significant gap exists in the current research landscape: the vast majority of existing HPC-based classifiers are trained and evaluated exclusively within virtual

machines (VMs). This reliance on virtualized data is problematic, as hypervisor-induced noise, skews, and resource sharing can fundamentally alter the HPC signals compared to bare-metal (BM) execution [18]. Consequently, models optimized for VMs may suffer drastic performance degradation when deployed on physical hardware, and the efficacy of bare-metal-trained models on virtualized targets remains largely unexamined.

This work proposes the development of a broadly target-generalizable ransomware classifier based on HPCs, complemented by a refinement methodology designed for unforeseen deployment environments. We define our research hypothesis as the contention that the primary barrier to model portability is the diversity of target architectures. Specifically, we posit that if the training set is sufficiently large to achieve model convergence, the resulting reduction in target generalization error will significantly enhance the model’s ability to remain effective across disparate systems. Our ongoing work utilizes a systematic evaluation strategy: a symmetric, instance-level testing matrix, to address three critical sets of Research Questions (RQs) necessary to develop an environment-agnostic defense:

- RQ1: Environment Generalizability. How well does a classifier perform when tested against a target environment it was not trained on? Does incorporating more targets into the baseline classifier lead to a continual reduction in measured generalization error when tested on unforeseen systems?
- RQ2: Transfer Learning. When incorporating a model for a particular target, is less data needed to train the model to the same level of accuracy, given a pre-trained model on a previous but different target? Will accuracy be improved given the pre-trained model vice without?
- RQ3: Environmental Signal Decoupling. Can a multi-target training approach effectively isolate the “malware signature” from “environmental noise” (e.g., hypervisor interrupts or scheduling jitter)?

By establishing these baselines, we aim to demonstrate the potential impact of this framework: that with sufficient model convergence and transfer learning, a broadly target-generalizable HPC-based classifier is not only valid but essential for securing disparate real-world systems.

The remainder of this work-in-progress paper is organized as follows: Section 2 provides foundational background on microarchitectural telemetry and hardware virtualization mechanics, while Section 3 examines existing literature in HPC-based ransomware detection. Section 4 details our target-generalizable classifier development framework, outlining our core research hypothesis and symmetric evaluation strategy. Finally, Section 5 concludes the paper with a summary of potential cross-platform deployment impacts.

2 Background

2.1 Computer Software, Hardware, and Virtual Machines

A traditional computer environment contains one OS for a given set of hardware resources [2]. In virtualized environments, a Virtual Machine (VM) hosts one or

multiple OSes in a host OS, thus enabling multiple environments to share the same hardware resources.

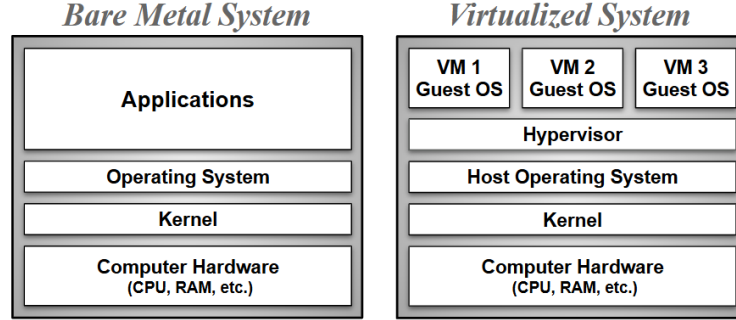


Fig. 1: Traditional "Bare Metal" Architecture vs. Typical VM Architecture.

Virtual machines are widely used over bare-metal systems in ransomware research for two primary reasons: the research overhead of bare-metal and the risk reduction of accidental ransomware spillage. VMs enable relatively easy recovery of a system after data encryption following a ransomware attack compared to the full reimaging required on bare-metal [5]. Furthermore, virtualized OSes are isolated from their hosts, and thus mitigate the risk of ransomware destroying data on the host machine's system or network. Consequently, most of the existing literature on ransomware classification using HPCs has been conducted inside virtual machines [14].

2.2 Hardware Performance Counters (HPCs)

HPCs are CPU registers that capture low-level microarchitectural events, such as cache misses and branch mispredictions. Traditionally used for performance tuning [8], they are now leveraged as side-channels to monitor CPU execution [6]. Managed by the Performance Monitoring Unit (PMU) [9], HPCs can be collected via software like Intel® VTune™ Profiler[5], which we utilize in our work.

Virtualized HPCs Several methods exist for virtualizing hardware performance counters.

1. **Trapping and Emulation:** Guest PMU access is intercepted and emulated by the hypervisor, often resulting in high overhead that negates the low-impact benefits of HPCs [15].
2. **Paravirtualization:** The guest OS uses hypercalls to communicate HPC values in batches, reducing the overhead of trapping every individual instruction [10]
3. **Hardware-assisted virtualization:** Uses modern extensions (Intel VT/AMD-V) to forward guest PMU access directly to the hardware [10].

Frameworks such as Perfctr-Xen [10] extend the compatibility of standard HPC low-level monitoring tools such as Perfctr [12] for virtual machines through virtualized HPCs. Perfctr-Xen showed very small deviations in both bare-metal and virtual collections of HPCs. However, only nine HPCs were used for this analysis [10]. Our ongoing work aims to establish efficacy across and between virtualized and non-virtualized HPCs, with a focus on ransomware classification.

Challenges with Virtualized HPCs Despite recent progress in supporting virtualized HPCs, the literature raises concerns about VM-based HPC systems [18]. Zhou et al. argued that HPC values collected in virtualized environments differ substantially from those obtained on bare-metal systems [18].

Due to the inherent difficulties in virtualizing the PMU, early virtual machines disallowed HPC use altogether [10]. The PMU, like any other hardware resource on a virtualized system, is limited and shared between all guest OSes and the host system [18]. The primary difficulty arises from resource management, as typically the guest OS and the hypervisor are mutually unaware of each other’s scheduling decisions [15]. These timing and scheduling challenges demonstrably affect the HPCs.

Virtualized HPCs must be managed both by the hypervisor and the guest. Consequently, there has been a significant overhead due to the required coordination between the guest OS and the hypervisor to track and update HPCs. Furthermore, some counter registers are impossible to safely update, as they are shared across multiple cores [10].

3 Prior Work

Prior work has primarily focused on virtualized HPCs, with only a few examples of bare-metal HPCs for ransomware detection [5,1,13,11]. No existing work has incorporated both virtualized and bare-metal HPCs for ransomware classification.

3.1 Related HPC Ransomware Classification Work

Existing HPC-based ransomware detection methods largely rely on virtualized environments for data collection. Pundir et al. introduced RANSTOP, an LSTM RNN model developed using a Windows VM on a Linux host [13]. Similarly, Olani et al. developed Deepware, which transforms HPC data into 2D images for CNN training within a comparable VM environment [11]. Other virtualized approaches include Thummapudi et al.’s host-level system, which captures HPCs directly from the hypervisor [16], and the RAPPER framework by Alam et al. [1].

More complex hybrid models like HyperMD combine HPCs with system memory dumps for classification [17]. While effective, HyperMD is fundamentally tied to virtualized platforms due to its reliance on PMU virtualization and

VM memory introspection. Similarly, Anand et al.’s HiPeR model was trained and evaluated exclusively using data collected within a Windows VM [3].

In contrast, bare-metal research remains limited. Hill et al. utilized Intel VTune on a bare-metal Windows system to collect over 200 HPC events across thousands of execution trials, systematically evaluating and narrowing down which hardware counters were most effective for classifying ransomware and benign workloads - which consisted primarily of SPEC benchmarks [5]. However, this work identified a critical need for more diverse benign test processes and was limited by testing exclusively on the same underlying bare-metal architecture across both machines during training and evaluation.

Several authors have questioned the ability of HPCs to detect malware. Zhou et al. reviewed existing HPC detection methods and identified major methodological flaws across all existing works. Zhou tested existing classification methods on a larger set of malware and benignware samples in a bare-metal environment. This resulted in drastic degradation of the performance of existing classifiers. Zhou thus concluded that Hardware Performance Counters were fundamentally unable to reliably detect malware [18]. However, Zhou’s evaluation method itself was limited by using principal component analysis (PCA), which can lead to information loss and low detection performance [4].

There are two primary limitations in these prior works: the lack of a target-portability analysis and the absence of robust multi-target dataset evaluations. Despite potential shortcomings of virtualized vs. bare-metal HPCs for ransomware detection, most studies focused exclusively on either virtual or bare-metal platforms for both training data generation and detection model evaluation. There is limited research on the portability of a detection model trained on data from one environment and deployed in another (generalizability). Given the increasing diversity between bare-metal and virtualized targets in both civilian and military contexts, an effective ransomware detection system should accurately detect ransomware in both environments across a wide range of microarchitectures. This limitation underscores a broader limitation in the generalizability of existing work. Most models in prior work were tailored to the specific experimental setups from which they were trained. RAPPER used testing data from the same sandboxed testing environment [1], RanStop used unseen programs for testing but still relied on testing data from the same exact VM/Linux host setup [13]. Deepware was similarly tested on ransomware samples from the same virtualized training environment [11], and Thumapudi et al. used testing data from the same training VM setup [16]. While Hill et al. evaluated their classifier on independent test datasets, their experiments remained restricted to data harvested from identical bare-metal hardware configurations [5]; HyperMD also used a test dataset constructed from the same training environment [17]. We hypothesize that the broad reliance on testing data from the original training setup, without variation in microarchitecture, OS, or VM or bare-metal environments, limits the generalizability of classifiers across varying real-world environments. Our ongoing work aims to assess the validity of this claim.

4 Ongoing Work

To address cross-environment detection gaps, we present a framework for constructing a target-generalizable ransomware classifier leveraging HPC telemetry, accompanied by an ongoing bare-metal and virtualized data collection pipeline.

4.1 Target-generalizable HPC-based Ransomware Classifier Development

Our development framework addresses the structural limitations of cross-environment portability in microarchitectural malware detection by isolating target-specific environmental anomalies from invariant malicious behaviors. As a work-in-progress framework, our methodology outlines a scalable path toward isolating target-specific environment anomalies from invariant malicious behaviors.

Research Hypothesis. We contend that the primary barrier to classifier portability is the systemic divergence of target architectures and hypervisor-induced execution noise. We hypothesize that an environment-agnostic classifier can be achieved by aggregating multi-target HPC data sampled across distinct underlying probability distributions: $P_{BM}(X, Y)$ for physical systems and $P_{VM}(X, Y)$ for virtualized environments. Incorporating diverse architectural data into a unified training set ($D_{BM} \cup D_{VM}$) will continually reduce measured target generalization error, outperforming single-source classifiers (H_{BM} or H_{VM}) when deployed in unforeseen environments.

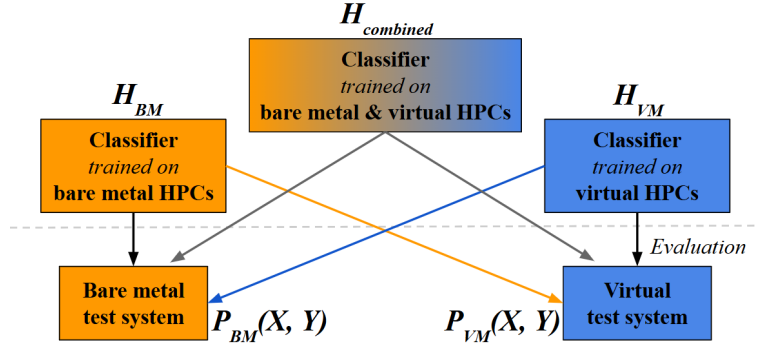


Fig. 2: Evaluation of bare-metal and Virtual Classifiers and Systems.

Evaluation Strategy. To evaluate this hypothesis and resolve our core research questions (RQ1–RQ3), we implement a symmetric cross-environment testing matrix, as illustrated in Figure 2. This dual evaluation directly assesses the efficacy of classifiers trained on bare-metal and virtual HPCs for use across both bare-metal and virtual deployment systems. Through our evaluation of

the combined model $H_{combined}$, we establish the operational efficacy of merging bare-metal and virtualized HPC datasets and quantify their collective impact on classification performance. This methodology lays the foundation for addressing **RQ3** by working toward increasingly generalizable classifiers agnostic of system architectures. To project generalizability trends without immediately incorporating additional physical target datasets, we construct a subset proxy classifier ($H_{90\%combined}$) trained on a random 90% of the entire dataset across both targets and test it against the remaining 10%. Crucially, to eliminate microarchitectural data leakage and guarantee strict experimental independence, our partitioning logic isolates entire, independent data collection instances rather than dividing fractional sub-trial traces or temporal execution windows. This allows us to infer a reliable trend line for generalization to new environments, which can be further refined as additional target datasets are introduced.

Potential Impact. A successful validation of this framework directly mitigates the severe performance degradation caused by a platform mismatch between a classifier’s training domain and its physical deployment target. In practical security operations, a significant gap exists when models trained in controlled, virtualized sandboxes are deployed onto bare-metal enterprise workstations (or vice versa), as hypervisor-induced anomalies distort the underlying microarchitectural telemetry. By successfully decoupling invariant ransomware execution signatures from target-specific scheduling jitter and hypervisor interrupts, this work provides a blueprint for a resilient defense vector that maintains high detection fidelity across disparate environments, neutralizing the operational vulnerabilities of cross-platform deployment mismatches.

4.2 Bare-Metal HPC Collection

For bare-metal collection of dynamic ransomware runs, we employ a dedicated, air-gapped collection system: a Dell Precision 7920 Tower with an Intel Xeon Silver 4208 (8-core, 2.10 GHz, Cascade Lake microarchitecture), 32 GB of DDR4 memory, and a Samsung 970 EVO Plus M.2 NVMe SSD running Windows 10 Pro (Build 10.0.19044.1586, Version 21H1) with Windows Defender disabled. This configuration represents a typical off-the-shelf server/high-end desktop built around a widely deployed recent Intel microarchitecture.

The collection pipeline, shown in Figure 3, is driven by a batch file and Python test script that together orchestrate 43 operations. Rather than introducing sample redundancy, we deliberately selected 15 ransomware samples spanning eight prevalent, structurally diverse families: Babuk, DarkSide, HiddenTear, Phobos, REvil/Sodinokibi, Snatch, WannaCry, and Gibberish. This targeted curation ensures high real-world threat coverage with a relatively small sample footprint by capturing distinct microarchitectural behaviors, ranging from hyper-threaded cache eviction loops to safe-mode boot manipulation and network-propagating payloads. The remaining 28 benign operations consist of SPECworkstation 3.1 benchmarks, standard benign encryption, and custom benign executables explicitly designed to mimic ransomware execution patterns.

This diverse baseline ensures the dataset captures realistic background system activity while challenging the model to differentiate malicious encryption from legitimate, high-I/O administrative and cryptographic workloads.

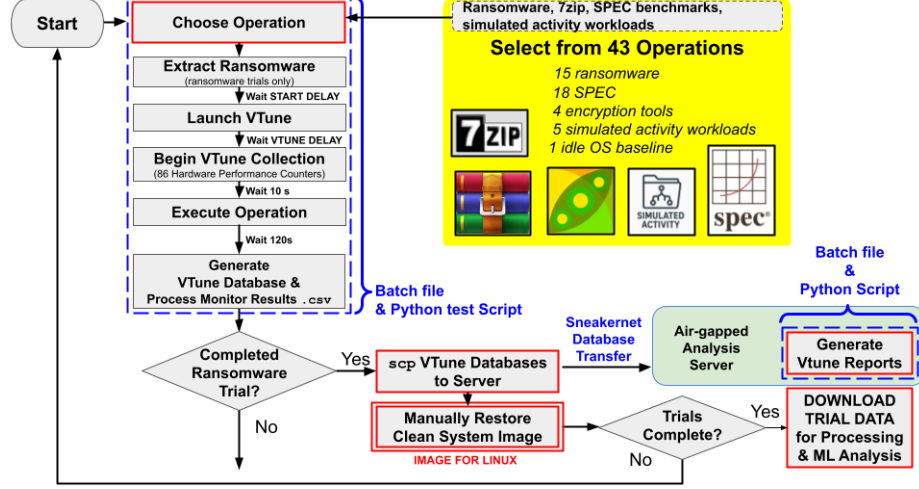


Fig. 3: Bare-metal HPC collection pipeline. A batch file and Python test script select one of 43 operations per trial; Intel VTune samples 86 HPCs over a 120-second window while Process Monitor runs concurrently to log parent-child process metadata. Completed VTune databases and Process Monitor logs are transferred via `scp` to an air-gapped analysis server for offline report generation, and the collection host is manually restored to a clean image before each subsequent ransomware trial.

To prevent sample propagation and eliminate network-induced microarchitectural noise, the collection host is completely air-gapped. While isolation suppresses live C2 network telemetry, our dataset explicitly selects variants capable of offline encryption (via embedded public keys or local key generation), isolating the core host-level I/O and cryptographic execution signatures. For ransomware trials, the collection host is then manually restored to a clean system image before the next trial, guaranteeing that each trial starts from an identical, uncompromised state. The resulting bare-metal dataset, D_{BM} , represents our empirical sampling of the underlying source distribution $P_{BM}(X, Y)$ used to train H_{BM} and, in combination with the virtualized collection D_{VM} , the unified classifier $H_{combined}$.

4.3 Virtualized HPC Collection

To maintain architectural consistency between environments, the virtualized testing platform is configured via a physical-to-virtual (P2V) clone of the original bare-metal system and is deployed on a second, nearly identically specified Dell

Precision 7920 using Microsoft Hyper-V. This second machine has 64GB instead of 32GB of RAM to accommodate the guest and host partitions. The guest VM is allocated 16 vCPUs and 32 GiB of statically assigned RAM to mirror the source machine’s logical topology and minimize memory ballooning artifacts. The guest boots from a fixed size VHDX, and has no virtual network adapter to prevent the spread of ransomware. Symmetrically matching our bare-metal pipeline, this virtualized collection yields the empirical dataset D_{VM} , mapping our observations of the underlying virtualized distribution $P_{VM}(X, Y)$ used to train H_{VM} and construct $H_{combined}$.

5 Conclusion

This work addresses a critical deficiency in current HPC-based ransomware detection: the lack of cross-environment portability. By formalizing a clear **research hypothesis** regarding multi-target microarchitectural signal aggregation, this work moves beyond environment-siloed paradigms. We outline a systematic **evaluation strategy** designed to address our core research questions (**RQ1–RQ3**) surrounding environmental signal decoupling and model convergence by evaluating symmetric, instance-level classifier performance across bare-metal and virtualized domains. Ultimately, the **potential impact** of successfully answering these questions will move the security ecosystem toward a more resilient defense-in-depth posture, ensuring that platform-agnostic microarchitectural detection remains robust regardless of whether the target threat is executed on virtualized cloud infrastructure or high-end bare-metal servers.

Acknowledgments. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Naval Academy, the Department of the Navy, the Department of War, or the U.S. Government. AI Acknowledgment: This work used Google Gemini for linguistic refinement, structural editing, and proof-reading. All research questions, hypotheses, arguments, and conclusions were developed by human authors.

References

1. Alam, M., Bhattacharya, S., Dutta, S., Sinha, S., Mukhopadhyay, D., Chattopadhyay, A.: Rapper: Ransomware prevention via performance counters. arXiv preprint arXiv:2004.01712 (2020)
2. Amazon: What is a virtual machine? - virtual machine explained - aws (2025), [Online]. Available: <https://aws.amazon.com/what-is/virtual-machine/> (accessed Dec. 6, 2025)
3. Anand, A., Sharma, R., Peddoju, S.K.: Hiper: Hardware performance counters-based ransomware detection using real-time dynamic analysis. ACM Trans. Manage. Inf. Syst. **14**(3), 1–28 (2023). <https://doi.org/10.1145/3608484>
4. Elnaggar, R., Servadei, L., Mathur, S., Wille, R., Ecker, W., Chakrabarty, K.: Accurate and robust malware detection: Running xgboost on run-time data from performance counters. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems pp. 1–1 (2021). <https://doi.org/10.1109/tcad.2021.3102007>

5. Hill, J.E., Walker, T.O., Blanco, J.A., Ives, R.W., Rakvic, R., Jacob, B.: Ransomware classification using hardware performance counters on a non-virtualized system. *IEEE Access* **12**, 63865–63884 (2024). <https://doi.org/10.1109/ACCESS.2024.3395491>
6. Hill, J.: A side-channel framework and microarchitectural analysis application. In: *Proceedings of the International Symposium on Memory Systems*. pp. 159–161 (2025)
7. House, P.: Ransomware statistics [2026]: Costs, trends attack data (2026), <https://app.stationx.net/articles/ransomware-statistics>
8. Kuruvila, A.P., Mahapatra, A., Karri, R., Basu, K.: Hardware performance counters: Ready-made vs tailor-made. *ACM Trans. Embedded Comput. Syst.* **20**(5s), 1–26 (2021). <https://doi.org/10.1145/3476996>
9. Microsoft: Recording pmu events (2024), [Online]. Available: <https://learn.microsoft.com/en-us/windows-hardware/test/wpt/recording-pmu-events>
10. Nikolaev, R., Back, G.: Perfctr-xen: A framework for performance counter virtualization. In: *Proc. 7th ACM SIGPLAN/SIGOPS Int. Conf. Virtual Execution Environ.* pp. 15–26. Newport Beach, CA, USA (2011). <https://doi.org/10.1145/2007477.1952687>
11. Olani, A.S., Ganfure, M.: Deepware: Imaging performance counters with deep learning to detect ransomware. In: *Proc. IEEE Int. IOT, Electron. Mechatronics Conf. (IEMTRONICS)*. pp. 1–7. Toronto, ON, Canada (2022). <https://doi.org/10.1109/IEMTRONICS55184.2022.9770351>
12. Pettersson, M.: Perfctr library (2011), <http://user.it.uu.se/~mikpe/linux/perfctr/>
13. Pundir, N., Tehranipoor, M., Farahmandi, F.: Ranstop: A hardware-assisted runtime crypto-ransomware detection technique. *arXiv preprint arXiv:2011.12248* (2020)
14. Sayadi, H., He, Z., Makrani, H.M., Homayoun, H.: Intelligent malware detection based on hardware performance counters: A comprehensive survey. In: *2024 25th International Symposium on Quality Electronic Design (ISQED)*. IEEE (2024). <https://doi.org/10.1109/ISQED60706.2024.10528369>
15. Serebrin, B., Hecht, D.: Virtualizing performance counters. In: *Euro-Par 2011: Parallel Processing Workshops*, vol. 7155, pp. 223–233. Springer-Verlag, Berlin, Germany (2012). https://doi.org/10.1007/978-3-642-29737-3_26
16. Thummapudi, K., Lama, P., Boppana, R.V.: Detection of ransomware attacks using processor and disk usage data. *IEEE Access* **11**, 51395–51407 (2023). <https://doi.org/10.1109/ACCESS.2023.3279819>
17. Wang, J., Bao, S., Mu, Y., Guo, S.: Hypermd: A hybrid approach for malware detection using hardware performance counters and memory dumps. In: *Proc. IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*. pp. 1821–1828. Exeter, UK (2024). <https://doi.org/10.1109/TrustCom62996.2024.00235>
18. Zhou, Y., Wang, M., Guo, W., Guo, X., Chen, X., Ren, B.: Hardware performance counters can detect malware: Myth or fact? In: *Proc. ACM Asia Conf. Comput. Commun. Secur.* pp. 457–468. Incheon, Republic of Korea (2018). <https://doi.org/10.1145/3196494.3196515>