A Cost-Model Argument for the Adoption of Hardware-Assisted Malware Detection

Marcus Botacin
botacin@tamu.edu
Texas A&M University
Texas, USA

Uriel Kosayev
uriel@cymdall.com
CYMDALL
Israel

Amichai Yifrach
amichal@cymdall.com
CYMDALL
Israel

ABSTRACT

In this work, we answer the question: *Is it worth adopting a hardware-assisted zero trust monitoring solution?* To answer it, we revisit a cost model for incident response and complement it with performance costs related to malware infections and security monitoring. We show that having an efficient monitoring solution not only (i) decreases the security costs by preventing attacks and lowering the incident responder's burdens, but also (ii) decreases the performance costs by requiring lower CPU loads to continuously monitor the full system. Based on it, we advocate for the immediate adoption of hardware-assisted, full-system security monitoring solutions.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**; **Network security**; **Systems security**;

KEYWORDS

Malware, Antivirus, Cloud Computing, Hardware Design

ACM Reference Format:
Marcus Botacin, Uriel Kosayev, and Amichai Yifrach. 2023. A Cost-Model Argument for the Adoption of Hardware-Assisted Malware Detection. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (ACSAC '23)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Malicious Software (Malware) infections cause financial losses to organizations and disrupt their operations, such that it is currently more important than ever to defend against system threats. Over time, malware has been evolving not only in number but also in complexity, such that all parts of a system might be now either the source or the target of attacks. The current state-of-the-art in defense against advanced attacks is zero trust, where a system operation is verified in all its operational steps.

The key challenge for deploying zero trust is keeping system performance, since continuously monitoring the entire system operation imposes significant performance overhead to the regular application's execution. A strategy to enable zero trust is to move monitoring from software to hardware [1, 2], thus alleviating the main CPU load. The major drawback of this class of solutions is the need for hardware redesign, which is costly due to the required design paradigm shift. This drawback has been delaying a more widespread adoption of hardware-assisted security monitors.

In this work, we present a cost argument for the immediate adoption of hardware-assisted security monitoring solutions and the streamlining of zero-trust models. Our contribution is to include system performance requirements in addition to security requirements into an established cost model. We model the security monitoring problem mathematically to show that the costs of not adopting hardware-assisted zero trust are larger not only in incident response but also in performance costs.

2 THE ESTABLISHED COST MODEL

The cost of security is typically measured in terms of the cost of detection warnings and incident response. In this sense, we started our investigation by shedding light on the assumptions and hypotheses behind the traditional security cost models. More specifically, we took a consolidated cost model [4] and evaluate its application to a real detection scenario [3].

Equation 1 shows the modeling of the security costs of operating an endpoint solution over the years. The cost is given by an initial solution setup cost (C_i(, plus yearly (Y_i) costs composed of fixed subscription costs (C_b) and the dynamic costs of raising detection warnings for goodware (C_g) and malware (C_m).

$$Cost = C_i + Y_i * [C_b + C_g + C_m] \tag{1}$$

Whereas the setup and subscription costs of a hardware-assisted solution might be a little bit different from software-based solutions, they still do not account for the major part of the costs, that are given by the detection warnings. Thus, we did not focus on these fixed costs ins our analyses, but on the detection ones. Similarly, we did not focus on the total cost over the years, but on analyzing the cost components within the same year. Thus, we concluded that the costs are proportional to the costs of raising warnings for malware and goodware, as in the expression shown in Equation 2.

$$Cost(year) \propto C_g + C_m \tag{2}$$

Since detectors are not perfect, the cost of raising warnings for malware and goodware can be broken down into the costs of labeling (i) goodware as goodware (C_{gg}); (ii) goodware as malware (C_{gm}); (iii) malware as goodware (C_{mg}); and (iv) malware as malware (C_{mm}), as shown in Equation 3. The rationale behind that is that each type of warning implies distinct costs. Correctly labeling goodware implies zero additional costs. Mistakenly flagging

ACSAC '23, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

goodware as malware requires an analyst to flag the false positive. Labeling malware as goodware has the cost of a security breach.

$$Cost(year) \propto C_{gg} + C_{gm} + C_{mg} + C_{mm} \quad (3)$$

Modeling the cost of a triggered warning is hard. Traditionally, the cost is modeled as shown in Equation 4. A set of infections (M) have varied costs over time (t). The rationale of this modeling is that a malware infection has increased but limited impact over time, i.e., the more the malware runs, the more impact it causes (e.g., the more data is exfiltrated). Therefore, detecting early maximizes the return of the security solution. If the sample is not detected, the malware executes all its malicious actions, causing maximum damage/cost. This cost achieves a plateau after some time because there is no infinite damage (e.g., the damage ceases after the malware has exfiltrated all valuable information for a company).

$$DetectionCost(t) = M * e^{-(\frac{\alpha}{t})^2 * \ln 2} \quad (4)$$

To better understand how this model represents reality, we applied it to a concrete scenario. Table 1 shows the parameters used in a validated application of the model to real endpoint security solutions. We used these same parameters in our simulation.

Table 1: Simulation Parameters. Values retrieved from [3].

C_i	2000$	Y_i	1	C_b	8000$		
M	2000	T_{avg}	900s	T_{max}	3000s	$C_{analysis}$	70$
TPR	90%	TNR	95%	FPR	5%	FNR	10%

Figure 1 shows the total cost of detection for multiple detectors over time–i.e. when detection is triggered at different points in time. The more time spent until the detection, the more harm the malware causes. Ideally, the detection should occur as soon as possible, thus mitigating the caused damage.

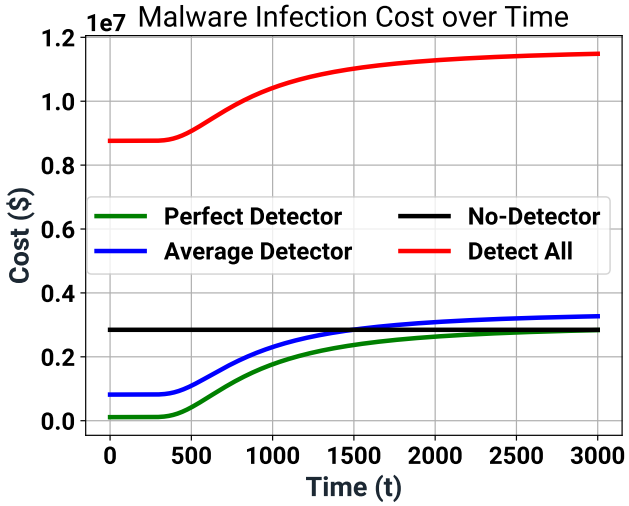

Figure 1: Security Cost Simulation. Best, Worst, and Average scenarios.

The perfect detector presents the lowest cost as it would detect all malware samples and have no False Positives (FPs). Its growth is exponential over time because even though the detector is perfect regarding the number of triggers, the damage caused by the malware by being detected late is still present. The average detector costs a bit more than the perfect one since it misses a few malware samples and also presents a few FPs. The scenario with no detector, when malware run freely, was initially revealed to be most costly than the two previous ones, because every malware infection would cause total harm.

Whereas the no-detection scenario does not change over time, the previous detectors lose their mitigation abilities if detection happens in the long tail. The break-even point is 1500 for the average detector and 2500 for the perfect detector. After this point, having a perfect detector or a no detector makes no difference. Also, having an average detector is worse than having no detector since it will impose the costs of handling FPs.

The extreme case of FPs is shown in the detect-all curve. This scenario presents the highest cost because although all malware infections are detected, all goodware executions are flagged, requiring an analyst to disambiguate the FPs. Since there is more goodware than malware running, the FP cost majors the total monitoring cost. Based on these results we conclude that (i) malware should be detected as soon as possible and that (ii) errors on flagging goodware execution imposes the largest part of the security monitoring costs.

3 PERFORMANCE COST-MODEL

Whereas the previously-presented security modeling is reasonable, it does not take into account the costs of performance monitoring. We here complement the modeling to claim that detection should not only be fast but also efficient. Once again, we opted to skip minor details and focus on the larger effects. Thus, we assumed that (i) Typical CPU load is constant in most server machines; (ii) The performance overhead of software-based monitors is mostly constant; and (iii) hardware-assisted monitors cause CPU bursts [1].

Modeling the performance of malware samples is as hard as modeling the security cost. Thus, we modeled the malware performance following the same parameters accepted for its security cost. However, to make it compatible with the assumption that the malware causes each time less damage over time, we assume that its performance curve has the opposite format of the security curve, i.e., it starts with high CPU usage and presents each time less impact. The final performance cost over time is shown in Equation 5.

$$PerformanceCost(t) = M * (1 - e^{-(\frac{\alpha}{t})^2 * \ln 2}) \quad (5)$$

Table 2 shows the parameters used in the simulations to measure the performance impact of multiple types of detectors.

Table 2: Performance Simulation Parameters. Parameter estimation in [1]

CPU_{avg}	50,00%	CPU_{mw}	50,00%
AvgOverhead_{sw}	20,00%	PeakOverhead_{hw}	50,00%
Cost_{cpu}	0,034	(per second at 50% load)	

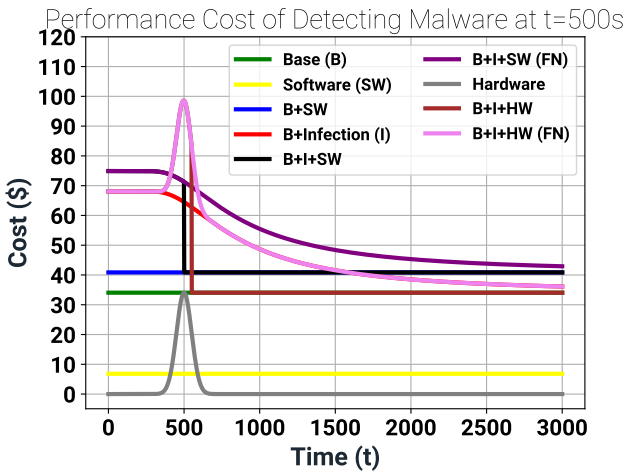

Figure 2: Performance cost of detecting malware at t=500s. Cost for different detection methods.

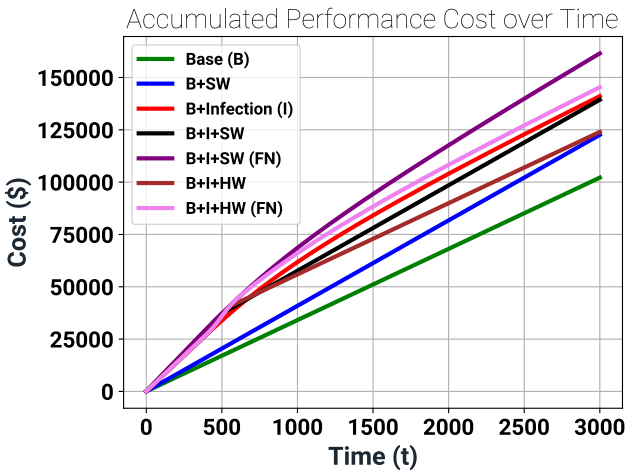

Figure 3: Accumulated Performance Costs. Malware and Monitoring solution execution impose execution coststs.

Figure 2 shows simulation results for scenarios with and without hardware-assisted monitoring solutions. We illustrate the case where the detection occurs at t=500s. We show the cases of true detection and False Negatives (FNs). We observe that the detectors operate differently by design. Software monitoring (SW) imposes a continuous overhead (B+SW). hardware monitoring (HW) imposes a peak overhead during detection, but no continuous overhead. When malware infects the system (I), it costs CPU. thus, fighting malware is key not only to prevent security violations but to cut performance costs. If the malware is not detected (FN), the performance cost (B+I) tends to return to the initial values as the malware finishes executing. If the malware is detected by the software monitor (B+I+SW), the malware operation ceases, but the performance cost does not return to the baseline (B), because the software monitor keeps occupying the CPU with constant monitoring (B+SW). The trigger of the hardware detector (HW) has the highest performance cost, as it operates in bursts. However, after malware is detected (B+I+HW), the cost returns to the base value (B), as the hardware imposes no CPU cost to the monitoring of the remaining goodware files.

The effect of different detectors is highlighted when we consider the accumulated cost of monitoring, as shown in Figure 3. In addition to the base cost of computing the main task (B), software monitoring incurs an additional cost (25K$ in the period) even without detecting anything. Infecting a system imposes additional costs as the malware runs, but the cost is different if detection is performed in hardware or software. If a hardware detector is considered, the cost tends to return to the base level in the long term. If the detection is software-based, the total cost keeps increasing even after the detection, as the monitoring keeps loading the CPU.whereas. In the case of FPs, the performance cost is of additional 50K$, or 50% more than the base cost.

4 CONCLUSION

In this work, we enriched security cost models with performance models to provide an argument for the immediate adoption of hardware-assisted zero-trust monitoring solutions. We concluded that whereas security-wise detection should happen early to avoid the malware impact, performance-wise detectors should be hardware-assisted to avoid the impact of goodware monitoring.

ACKNOWLEDGMENTS

The authors thank CYMDALL for supporting the development of this research project.

REFERENCES

[1] Marcus Botacin, Marco Zanata Alves, Daniela Oliveira, and André Grégio. 2022. HEAVEN: A Hardware-Enhanced AntiVirus ENgine to accelerate real-time, signature-based malware detection. *Expert Systems with Applications* 201 (2022), 117083. https://doi.org/10.1016/j.eswa.2022.117083

[2] Marcus Botacin, Francis B. Moreira, Philippe O. A. Navaux, André Grégio, and Marco A. Z. Alves. 2022. Terminator: A Secure Coprocessor to Accelerate Real-Time AntiViruses Using Inspection Breakpoints. *ACM Trans. Priv. Secur.* 25, 2, Article 9 (mar 2022), 34 pages. https://doi.org/10.1145/3494535

[3] Robert A. Bridges, Sean Oesch, Michael D. Iannacone, Kelly M. T. Huffer, Brian Jewell, Jeff A. Nichols, Brian Weber, Miki E. Verma, Daniel Scofield, Craig Miles, Thomas Plummer, Mark Daniell, Anne M. Tall, Justin M. Beaver, and Jared M. Smith. 2023. Beyond the Hype: An Evaluation of Commercially Available Machine-Learning-Based Malware Detectors. *Digital Threats* (feb 2023). https://doi.org/10.1145/3567432 Just Accepted.

[4] Michael D. Iannacone and Robert A. Bridges. 2020. Quantifiable & comparable evaluations of cyber defensive capabilities: A survey & novel, unified approach. *Computers & Security* 96 (2020), 101907. https://doi.org/10.1016/j.cose.2020.101907

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009