On the Malware Detection Problem: Challenges & Novel Approaches

Marcus Botacin[1] **, Paulo de Geus**[2] **, André Grégio**[1]

[1] Federal University of Paraná (UFPR)

{mfbotacin, gregio}@inf.ufpr.br

[2]Co-Advisor - University of Campinas (UNICAMP)

paulo@lasca.ic.unicamp.br

Abstract. *Many solutions to detect malware have been proposed over time, but effective and efficient malware detection still remains an open problem. In this work, I take a look at some malware detection challenges and pitfalls to contribute towards increasing system's malware detection capabilities. I propose a new approach to tackle malware research in a practical but still scientific manner and leverage this approach to investigate four issues: (i) the need for understanding context to allow proper detection of localized threats; (ii) the need for developing better metrics for AntiVirus (AV) evaluation; (iii) the feasibility of leveraging hardware-software collaboration for efficient AV implementation, and (iv) the need for predicting future threats to allow faster incident responses.*

1. On The Malware Detection Problem

Malware has been a major threat to most current computer systems, causing from image damages to financial losses to individuals, such that the development of detection solutions became essential to allow for the facing of the current scenario of widespread threats and safe computer usage. Many solutions have been proposed over time to prevent, detect and remedy malware infections – e.g., Anti-Virus solutions (AVs). However, despite all developments so-far, security solutions still suffer from a plethora of drawbacks that significantly limit their operation, such as high-performance impacts for real-time monitoring and/or not informative detection labels. Thus, there is an urgent need to understand the drawbacks of current solutions to allow the development of mitigation procedures that may increase malware detection rates and AVs performance.

The current status of security solutions immediately leads to the following question: *Why the security problem has not yet been solved?* The first and most obvious answer for this question is that *"because security is hard"*—Fred Cohen has proven theorems decades ago to show that there is no algorithm that can perfectly detect all possible viruses [Cohen 1984], such that any attempt towards this direction is just an imperfect approximation of *"security"*. This fact, albeit, does not allow anyone to give up on protecting users, since other protection mechanisms, such as enhancing trust relations, might still provide *"reasonably safe"* computer usage. Therefore, the whole idea of this thesis is to discuss where to place the bar for this reasonably safe approximation of security.

This thesis' hypothesis[1] is that attempts to solve the malware detection problem lack stronger methodologies and that bridging the gap of a more robust methodology is part of

[1] Thesis defense recording: https://www.youtube.com/watch?v=_ZnEm1mtzSw

the answer towards enhancing security. Thus, I propose to delve into the main malware detection challenges and implications to contribute towards increasing malware detection capabilities in systems by relying on methodologically stronger procedures. To do so, I reviewed the body of work of more than 400 papers published under the malware umbrella in the major security conferences and identified common pitfalls that potentially limit the research advances on the malware countermeasures topic.

Based on the aforementioned literature review, I propose a new approach to tackle malware research experiments in a practical, but scientific manner and leverage this approach to investigate four derived issues in depth: (i) the need for understanding context to allow proper detection of localized threats; (ii) the need for developing better metrics for AV evaluation; (iii) the feasibility of leveraging hardware-software collaboration for efficient AV implementation, and (iv) the need for predicting future threats to allow faster incident responses. I propose and implement new security solutions to solve these issues according to the newly proposed methodology. The remainder of this text is dedicated to describe this new methodological approach (Section 2) and the derived issues (Section 3). To conclude, the impact of this thesis is discussed in Section 4.

2. Field Analysis

2.1. Real AV Operation

A major goal of this thesis was to present the limits and drawbacks of current malware research works, which largely includes anti-malware solutions, of which AntiViruses (AVs) are a notable example. A derived goal was to propose new detection approaches to bridge some AV development gaps in a more practical way while targeting actual operational scenarios. For such, it was key to understand what is the actual operational scenario of a real AV solution. Despite AVs popularity, little is known about their internals, since they are mostly closed-source solutions, which often leads to inaccurate claims. To avoid committing pitfalls, I conducted an analysis of real AV's operations to develop the foundations for future developments. The findings were published in a paper [Botacin et al. 2021c] and the results published on it highlight: (i) the performance overhead imposed by monitoring solutions, which motivates the research about more efficient AVs; and (ii) the still significant use of signatures by AV solutions, which motivates my choice for their use in some of the published papers, as following presented.

2.2. Academic Publications

A key research question of this thesis is how malware detection research has been performed so far aiming at understanding its limitations, as well as how to overcome them with the application of a distinct research approach. Overall, malware research integrates science and engineering aspects. Therefore, to evaluate them, the *"malware research method"* was proposed, according to the following steps: (i) Common Core from the Scientific Method; (ii) Research Objective Definition; (ii) Background Research; (iii) Hypothesis/Research Requirements; (iv) Experiment Design; (v) Test of Hypothesis/Evaluation of Solution; and (vi) Analysis of Results.

Based on this framework, we reviewed the body of work published in the most reputable venues of computer security research (Table 1). Therefore, I conducted a critical literature review to identify common challenges and pitfalls in malware research. The findings were published in a paper [Botacin et al. 2021b] which constitutes the core of all criticism

Table 1. Selected Papers. Distribution per year (2000 – 2018) and per venue.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
USENIX (Security, LEET & WOOT)	1	0	0	0	0	1	1	6	2	3	7	8	10	12	9	7	9	13	6	95
CCS	0	0	0	0	0	0	0	2	4	6	6	7	11	9	11	14	2	11	6	89
ACSAC	0	0	0	0	2	3	2	4	4	1	3	8	10	7	10	6	3	7	8	78
IEEE S&P	0	1	0	0	0	1	3	2	1	0	0	10	17	12	3	6	4	5	3	68
DIMVA	0	0	0	0	0	4	4	3	8	2	3	0	8	4	8	7	7	5	4	67
NDSS	0	0	0	0	1	0	2	0	3	3	3	3	2	4	5	4	9	7	3	49
RAID	0	0	1	0	0	1	3	0	0	0	0	0	3	5	5	3	4	3	3	31
ESORICS	0	0	0	0	0	1	0	0	2	1	0	0	2	3	3	0	1	1	0	14
Total	1	1	1	0	3	11	15	17	24	16	22	36	63	56	54	47	39	52	33	491

presented in this thesis. Among all findings, I highlight: (i) the scarce number of longitudinal malware analysis studies in the literature, which motivates my investigation about the Brazilian scenario; and (ii) the uncertainty about the application of AV results in fair comparisons, which motivates my investigation on the development of new AV evaluation metrics. These two investigations are detailed in Section 3.

3. Contributions to the Field

3.1. The need for context

Security solutions, such as AVs, are often expected to protect **all users** against **all types** of threats, thus they adopt a policy of considering hypothetically-defined generic samples as representative of all operational contexts. This approach also implies generic assumptions about systems capabilities (e.g., they have similar configurations), users behaviors (e.g., they are equally vulnerable to a type of threat), and malware families distribution (e.g., all contexts are targeted by the same threats). This approach and assumptions clearly do not hold for all cases, but the implications of this choice are unknown, as the academic literature often overlooks these cases. Therefore, I proposed investigating the impact of addressing localized issues using a generalized approach to understand which factors can be really generalized and which ones require localized handling. The presented hypothesis is that AVs cannot operate in an *"one-size-fits-all"* manner and thus that they should consider particularities of each operational scenario. To evaluate that hypothesis, I delved into two cases of regional threats: First, I analyzed the differences between mobile banker malware and banking applications observed in the Brazilian scenario in comparison with other academic reports [Botacin et al. 2019b]. Second, I investigated the differences between desktop malware samples collected from Brazilian user's machines and the literature reports for *"global"* samples [Botacin et al. 2021a]. As the outcome of these evaluations, I exemplify characteristics unique to the Brazilian malware (e.g., Whatsapp-based banking threats) and establish some guidelines for threat scenario characterizations.

3.2. The need for better evaluations

AVs have become the main defense line against malware for most corporations and end-users, therefore it is natural that these users look for information about which AVs perform better. From a commercial perspective, one can find multiple AV evaluations considering aspects such as detection rate and memory consumption, but, from an academic perspective, these evaluations are very limited, neglecting important factors, such as detection regression, i.e., when a sample stops being detected after some time. Whereas it was clear that these evaluations are limited, it was not clear which metrics should be considered when selecting an AV solution for a given scenario or user. Therefore, I proposed evaluating AVs for a long period of time and to identify distinct metrics for their evaluation, to understand their impact, and thus to provide clearer guidelines for AV selection.

The newly-proposed metrics accounts for the effect of time due to AV updates and were published in a paper [Botacin et al. 2020b].

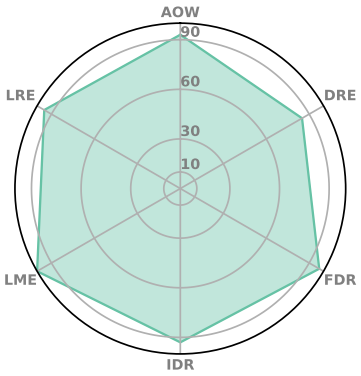

Figure 1. AV1.

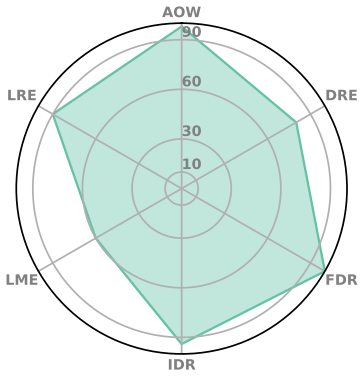

Figure 2. AV2.

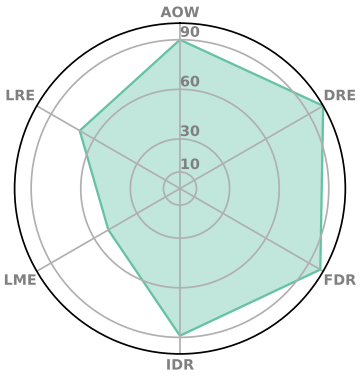

Figure 3. AV3.

Finding: Three AVs that present the same detection rates cannot be simply considered as equivalent. The developed multi-dimensional evaluation metrics allow the AV comparison regarding multiple aspects simultaneously.

The obtained results highlight that AV evaluations should be multi-dimensional to fully characterize AV's operations. Figures 1, 2, and 3 illustrate the case in which the AVs are internally different even though they present the same detection rate. Considering this case, I claim that the selection of an AV should consider the AV's characteristics in conjunction with the operational scenario's characteristics.

3.3. The need for performance-efficient AVs

A major drawback of most current malware detection solutions is that they are completely implemented in software, thus causing their user's machines to slowdown due to the need of executing monitoring instructions instead of the user's application code. A strategy to speed up these solutions is to move them from pure software solutions to hardware-assisted solutions, thus eliminating the whole performance overhead of running additional code. This paradigm shift, however, introduces two new challenges: (i) identifying new features for malware classification, as the previously leveraged software features will not be available in hardware; (ii) allowing for malware definition updates, since hardware storage is much more limited and less flexible in comparison to software

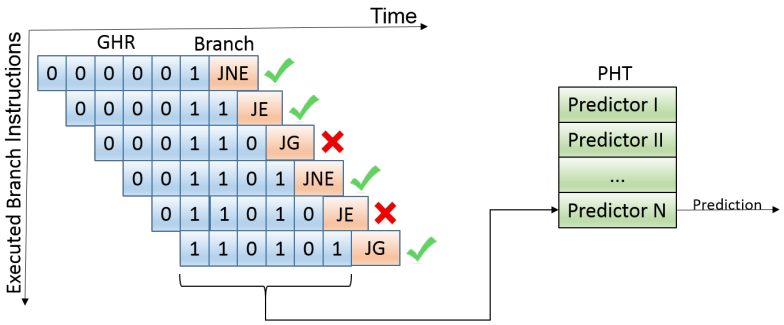

Figure 4. 2-level branch predictor. A sequence window of taken (1) and not-taken (0) branches is stored in the Global History Register (GHR).

Finding: The branch patterns can be used as signatures for real-time malware detection.

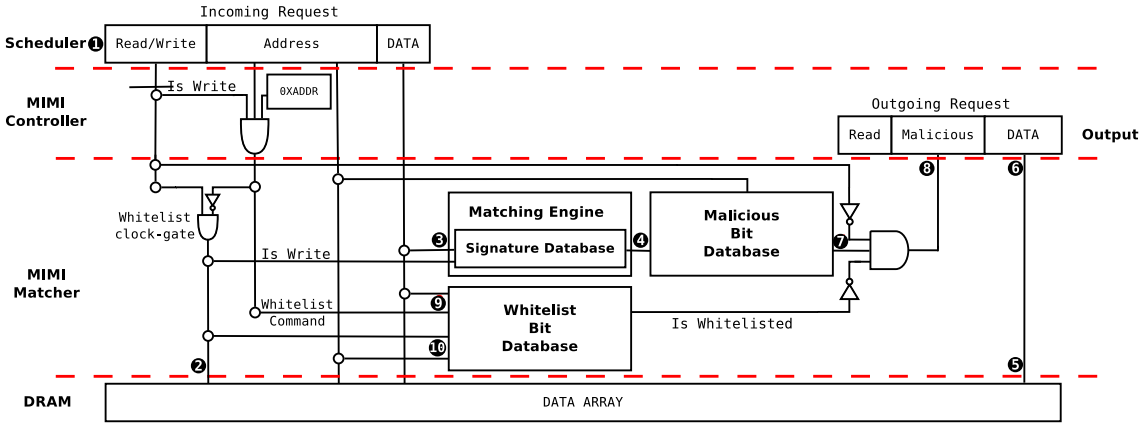

Figure 5. Memory controller instrumented to check for malware signatures.

Finding: A secure-by-design system can scan memory contents within the memory controller to deliver only safe to execute data.

solutions. Therefore, aiming to mitigate these issues, I proposed investigating: (i) how malicious software execution impacts existing architectural structures at low-level (e.g., CPU pipeline, cache, memory, and so on) and how these existing low-level entities could be leveraged to support detecting malicious behaviors (e.g., self-modifying code) at higher abstraction levels (AV detection triggering) [Botacin et al. 2020e]; (ii) how reconfigurable hardware (FPGA) could be used to implement an updatable AV solution for matching low-level features (Hardware Performance Counters data) [Botacin et al. 2019]; (iii) finally, I proposed an innovative use for a low-level feature (branch patterns) that can be obtained and matched via low-level component extensions (branch predictors), and leveraged for fingerprinting malicious behaviors at higher levels (AV detection triggering). Among all findings, I would like to empashize: the possibilities brought by interpreting low-level events in conjunction with their associated high-level constructions, and the feasibility of relying on hardware support for performance overhead mitigation in real-time AVs.

Figure 4 illustrates the proposal for an instrumented CPU's Branch Predictor Unit (BPU) able to also predict malware execution. The investigation about the impact of hardware on software execution revealed that the branch patterns stored in the BPU might be used to fingerprint malware execution in runtime.

3.4. The need for predicting attacker's movements

Security solutions have always been operating reactively. For instance, AV solution's operations consist of capturing samples in-the-wild, identifying the exploited breaches, and then deploying signatures or heuristics for the known threats. This approach opens a huge attack opportunity window, as the company takes a long time to respond to a newly created sample or exploited vulnerability. I strongly believe that security solutions should shift their operation scheme to a more proactive mode, trying to understand attack opportunities before they are exploited by actual samples and thus hypothetically reducing the response time to detect them. To test this hypothesis, I propose investigating, in an exploratory fashion, two scenarios of hypothetical future threats: (i) First, I investigated how current defensive solutions operating in a serial manner can be evaded by distributed (e.g., multi-core) malware samples and how security solutions can be adapted to handle these samples [Botacin et al. 2019]; (ii) Second, I investigated the threat of in-memory

malware samples which do not exhibit a disk counterpart for AV scanning and how scans could be triggered directly within the memory chip in future (smart memory-powered) architectures [Botacin et al. 2020d]. Among all findings, I highlight: (i) the need for efficiently scanning memory for effective malware detection; and (ii) the possibility of performance overhead reduction brought by memory-granular checks.

Figure 5 illustrates the proposal of a next-generation, secure-by-design platform based on the instrumentation of memory controllers with an AV engine. In this architecture, every data written in RAM is scanned by an integrated AV and only scanned data is delivered back to the CPU. Memory scans are delivered via special page faults (security faults), allowing the CPU to be aware that it is about to execute suspicious code without the need for a software AV to scan all memory pages, which eliminates performance penalties.

4. Current and Future Impact

Since this thesis' goal is to contribute towards the enhancement of the malware detection field, I adopted distinct communication strategies to reach out the distinct stakeholders in this community. First, as it is standard to any academic work, research results were published in academic papers, which are summarized in Table 2.

Table 2. Published Papers.

Paper	Venue	Research Type	Research Goal
[Botacin et al. 2021b]	Computers & Security	Landscape	Background
[Botacin et al. 2021c]	Computers & Security	Landscape	background
[Botacin et al. 2021a]	ACM TOPS	Landscape	Context
[Botacin et al. 2019b]	ACM ARES	Landscape	Context
[Botacin et al. 2020b]	Computers & Security	Landscape	Evaluation
[Botacin et al. 2022b]	ACM TOPS	Defensive	Hardware
[Botacin et al. 2022a]	Expert Systems	Defensive	Hardware
[Botacin et al. 2020e]	Springer JCVHT	Defensive	Hardware
[Botacin et al. 2019]	IEEE ReCoSoC	Defensive	Hardware
[Botacin et al. 2020c]	Springer JCVHT	Defensive	Hardware
[Botacin et al. 2020d]	ACM MEMSYS	Defensive	Hardware+Prediction
[Botacin et al. 2019]	Springer JCVHT	Offensive	Predicting
[Botacin et al. 2020a]	DIMVA	Landscape	Application
[Botacin et al. 2021d]	Digital Investigation	Defensive	Application
[Botacin et al. 2019a]	ACM ROOTS	Defensive	Application
[Beppler et al. 2019]	Springer ISC	Defensive	Application
[Sun et al. 2020]	IEEE TDSC	Defensive	Application
[Ceschin et al. 2019]	ACM ROOTS	Offensive	Application
[Ceschin et al. 2020]	ACM ROOTS	Offensive	Application

I tried in each paper to investigate a significant challenge to the enhancement of the security field and present possible paths to overcome the investigated challenges. In some cases, the published papers already accomplished their goals by influencing the methodology of further developed studies by third researchers. For instance, a master dissertation on AV evaluations [Raffa 2021] presents investigation strategies based on the AV evaluation metrics proposed in a paper [Botacin et al. 2020b] by me.

In addition to papers, I made an active effort to reach out the community via other communication channels, which included giving talks in multiple events. Thus, talks derived from this thesis were presented at the USENIX ENIGMA [Botacin 2021] and at the Security Work Group (GTS) from the Brazilian Internet Committee (NIC.br) [Botacin 2019, Grégio and Botacin 2020].

Finally, some impact is also expected in the long-term. More specifically, contributions placed in the computer architecture domain are expected to take a bit more time to be incorporated by the industry due to the need for designing new hardware. However, the first signs of the industry movements towards this direction can be already seem, with Intel patenting branch-based mechanisms for malware detection [Intel 2020], such that I believe my proposals of hardware-software security collaboration platforms might be somehow adopted in the future.

References

Beppler, T., Botacin, M., Ceschin, F. J. O., Oliveira, L. E. S., and Grégio, A. (2019). L(a)ying in (test)bed. In *Information Security*. Springer.

Botacin, M. (2019). Análise do malware ativo na internet brasileira: 4 anos depois. o que mudou? `https://gtergts.nic.br/`.

Botacin, M. (2021). Does your threat model consider country and culture? a case study of brazilian internet banking security to show that it should! In *USENIX Enigma*.

Botacin, M., Aghakhani, H., Ortolani, S., Kruegel, C., Vigna, G., Oliveira, D., Geus, P. L. D., and Grégio, A. (2021a). One size does not fit all: A longitudinal analysis of brazilian financial malware. *ACM TOPS*.

Botacin, M., Alves, M. Z., Oliveira, D., and Grégio, A. (2022a). Heaven: A hardware-enhanced antivirus engine to accelerate real-time, signature-based malware detection. *Elsevier ESWA*.

Botacin, M., Bert ao, G., de Geus, P., Grégio, A., Kruegel, C., and Vigna, G. (2020a). On the security of application installers and online software repositories. In *DIMVA*. Springer.

Botacin, M., Ceschin, F., de Geus, P., and Grégio, A. (2020b). We need to talk about antiviruses: Challenges & pitfalls of av evaluations. *Computers & Security*.

Botacin, M., Ceschin, F., Sun, R., Oliveira, D., and Grégio, A. (2021b). Challenges and pitfalls in malware research. *Computers & Security*, page 102287.

Botacin, M., de Geus, P. L., and Grégio, A. (2019). "vanilla" malware: vanishing antiviruses by interleaving layers and layers of attacks. *Comp. Vir. and Hack. Tech.*

Botacin, M., de Geus, P. L., and Grégio, A. (2020c). Leveraging branch traces to understand kernel internals from within. *Comp. Vir. and Hack. Tech.*

Botacin, M., Domingues, F. D., Ceschin, F., Machnicki, R., Zanata Alves, M. A., de Geus, P. L., and Grégio, A. (2021c). Antiviruses under the microscope: A hands-on perspective. *Comp. & Sec.*

Botacin, M., Galante, L., Ceschin, F., Santos, P. C., Carro, L., de Geus, P., Grégio, A., and Alves, M. A. Z. (2019). The av says: Your hardware definitions were updated! In *ReCoSoC*.

Botacin, M., Galante, L., de Geus, P., and Grégio, A. (2019a). Revenge is a dish served cold: Debug-oriented malware decompilation and reassembly. In *ROOTS*. ACM.

Botacin, M., Galhardo Moia, V. H., Ceschin, F., Amaral Henriques, M. A., and Grégio, A. (2021d). Understanding uses and misuses of similarity hashing functions for malware detection and family clustering in actual scenarios. *FSI: Digital Investigation*.

Botacin, M., Grégio, A., and Alves, M. A. Z. (2020d). Near-memory & in-memory detection of fileless malware. In *MEMSYS*. ACM.

Botacin, M., Kalysch, A., and Grégio, A. (2019b). The internet banking [in]security spiral: Past, present, and future of online banking protection mechanisms based on a brazilian case study. In *ARES*. ACM.

Botacin, M., Moreira, F. B., Navaux, P. O. A., Grégio, A., and Alves, M. A. Z. (2022b). Terminator: A secure coprocessor to accelerate real-time antiviruses using inspection breakpoints. *ACM Trans. Priv. Secur.*, 25(2).

Botacin, M., Zanata, M., and Grégio, A. (2020e). The self modifying code (smc)-aware processor (sap): a security look on architectural impact and support. *Journal of Comp. Virology (JCVHT)*.

Ceschin, F., Botacin, M., Gomes, H. M., Oliveira, L. S., and Grégio, A. (2019). Shallow security: On the creation of adversarial variants to evade machine learning-based malware detectors. In *ROOTS*. ACM.

Ceschin, F., Botacin, M., Lüders, G., Gomes, H. M., Oliveira, L., and Gregio, A. (2020). No need to teach new tricks to old malware: Winning an evasion challenge with xor-based adversarial samples. In *ROOTS*. ACM.

Cohen, F. (1984). Computer viruses - theory and experiments. `http://web.eecs.umich.edu/~aprakash/eecs588/handouts/cohen-viruses.html`.

Grégio, A. and Botacin, M. (2020). Integridade, confidencialidade, disponibilidade, ransomware. `https://ftp.registro.br/pub/gts/gts35/03-IntegridadeDisponibilidadeConfidencialidadeRansomware.pdf`.

Intel (2020). Technologies for hardware assisted native malware detection. `https://patentimages.storage.googleapis.com/fb/23/ff/9d11b27884f050/US10540498.pdf`.

Raffa, G. (2021). Testing antivirus in linux: An investigation on the effectiveness of solutions available for desktop computers. `https://www.royalholloway.ac.uk/media/16565/techreport-giusepperaffa.pdf`.

Sun, R., Botacin, M., Sapountzis, N., Yuan, X., Bishop, M., Porter, D. E., Li, X., Gregio, A., and Oliveira, D. (2020). A praise for defensive programming: Leveraging uncertainty for effective malware mitigation. *IEEE TDSC*.