How do we detect malware?

A step-by-step guide

Marcus Botacin

[1]botacin@tamu.edu
marcusbotacin.github.io

Who Am I?

- Assistant Professor (2022) - Texas A&M University (TAMU), USA
  - ACES Program Fellowship
- PhD. in Computer Science (2021) - Federal University of Paraná (UFPR), Brazil
  - Thesis: *"On the Malware Detection Problem: Challenges and new Approaches"*
- MSc. in Computer Science (2017) - University of Campinas (UNICAMP), Brazil
  - Dissertation: *"Hardware-Assisted Malware Analysis"*
- Computer Engineer (2015) - University of Campinas (UNICAMP), Brazil
  - Final Project: *"Malware detection via syscall patterns identification"*

Topics

The Malware Problem

How have we been doing? (Overall)

The good side

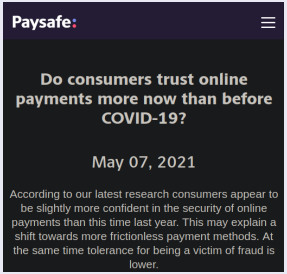


Figure: https://www.paysafe.com/en/blog/do-consumers-trust-online-payments-more-now-than-before-covid-19/

The bad side

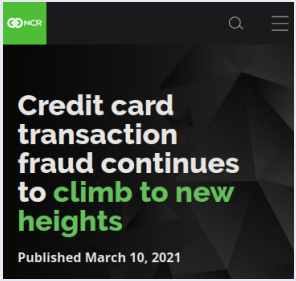


Figure: https://www.ncr.com/blogs/payments/credit-card-fraud-detection

How have we been doing? (Malware Specifics)

The good side

Figure: https://apnews.com/article/europe-malware-netherlands-coronavirus-pandemic-7de5f74120a968bd0a5bee3c57899fed

The bad side

Figure: https://thehackernews.com/2021/06/droidmorph-shows-popular-android.html

Topics

How Do We Detect Malware?

The State-of-the-art in Malware Detection & Prevention

Steps

1. Collection
2. Triage
3. Sandbox Analysis
4. Threat Intelligence
5. Endpoint Protection

Distributed Processing

- Collection

Cloud Processing

- Analysis and Intelligence steps

Limited Processing

- Endpoint

Collection

How to find new malware samples?

- Searching “dark web” forums.
- Crawling software repositories.
- Leveraging honeypots.
- Checking spam traps.
- Downloading Malware repositories.
- Scrapping blocklists.

The result

Figure: https://www.forbes.com/sites/thomasbrewster/2021/09/29/google-play-warning-200-android-apps-stole-millions-from-10-million-phones/

Triage

Why how many new malware samples?

- Variations from the same source code.

Implications

- Increase processing costs and response time.

How to solve this problem?

- Identify and cluster similar samples.

The Statistics

Figure:
https://www.kaspersky.com/about/press-releases/2020_the-number-of-new-malicious-files-detected-every-day-increases-by-52-to-360000-in-2020

Sandbox Analysis

Goals

- Uncover hidden behaviors.

Method

- Trace sample execution.

Challenge

- Handle evasion attempts.

Solution 1

Figure: `https://blog.virustotal.com/2019/05/virustotal-multisandbox-yoroi-yomi.html`

Solution 2

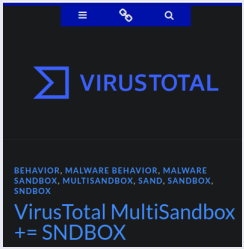


Figure: `https://blog.virustotal.com/2019/07/virustotal-multisandbox-sndbox.html`

Threat Intelligence

Goal

- Identify trends and predict attacks.

How?

- Data analytics over analyzed samples.

Challenges

- Look to a representative dataset.

We should look to:

Figure: `https://www.computerweekly.com/news/252504676/Ransomware-attacks-increase-dramatically-during-2021`

Endpoint Protection

Goal

- Protect customers in their machines.

How?

- Moving the **viable** analyses to the endpoint.

Challenges

- Performance and usability constraints.

Is there a “best”?

Figure: https://www.av-test.org/en/antivirus/home-windows/

Topics

Enhancing Malware Triage

The good side: Separating Code and Data

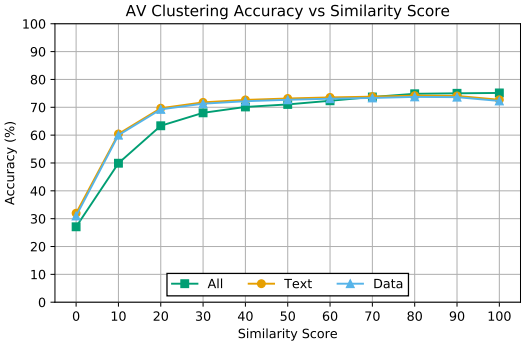


Figure: **Binary Sections Accuracy**

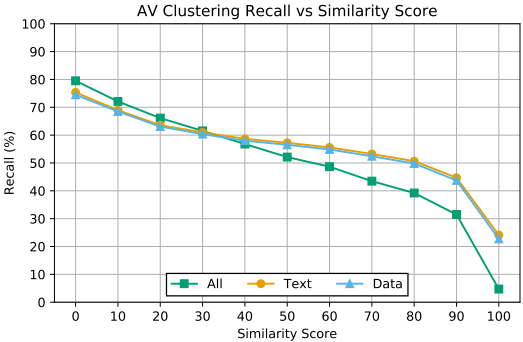


Figure: **Binary Sections Recall**

Source: `https://www.sciencedirect.com/science/article/abs/pii/S2666281721001281`

The bad side: Packed Samples

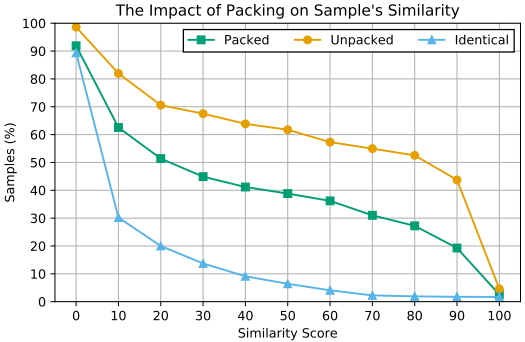


Figure: **The impact of UPX packing.** Packing reduces sample's similarity scores.

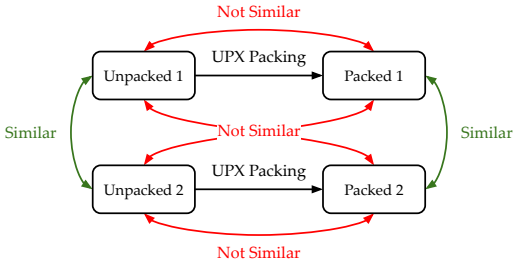


Figure: **Average Packed Sample's Similarity Scheme.** Cross-comparisons should be avoided.

Enhancing Malware Tracing

Software-based Sandbox

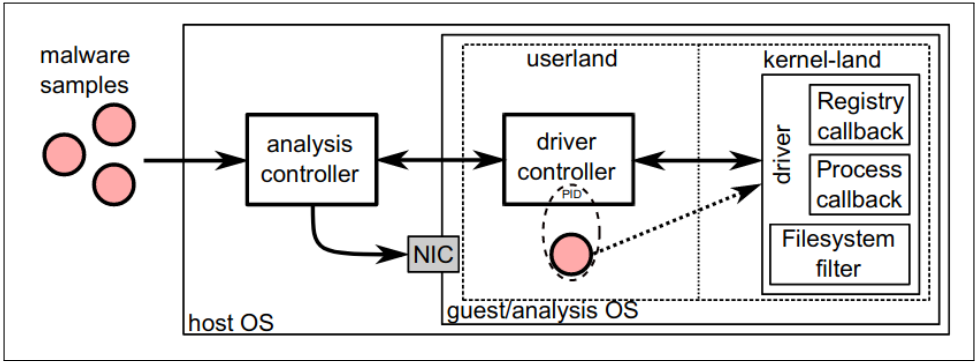


Figure: System Architecture.

Link: `https://link.springer.com/article/10.1007/s11416-017-0292-8`

Drawbacks: Anti-VM

Technique	Description	Detection
VM Fingerprint	Check for known strings, such as serial numbers	Check for known strings inside the binary
CPUID Check	Check CPU vendor	Check for known CPU vendor strings
Invalid Opcodes	Launch hypervisor-specific instructions	Check for specific instrutions on the binary
System Table Checks	Compare IDT values	Look for checks involving IDT
HyperCall Detection	Platform specific feature	Look for specific instructions

Hardware-based Sandbox

Monitoring Steps

1. Software executes a branch.
2. Processor stores branch address in memory page.
3. Processor raises an interrupt.
4. Kernel handles interrupt.
5. Kernel sends data to userland.
6. Userland introspects into this data.

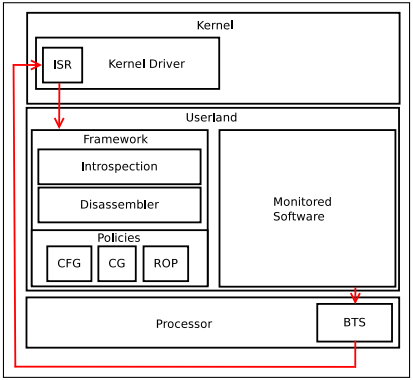


Figure: System Architecture.

Key Insight: Branches define basic blocks

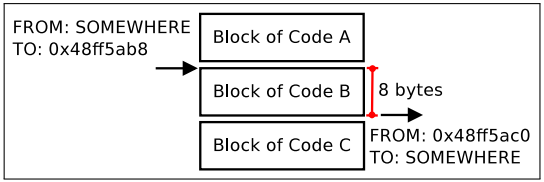


Figure: Identified branches and basic blocks..

Source: `https://dl.acm.org/doi/10.1145/3152162`

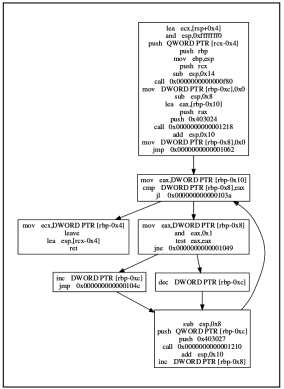


Figure: CFG Reconstruction.

From Tracing to Threat Intelligence

Brazilian Financial Malware on Desktop

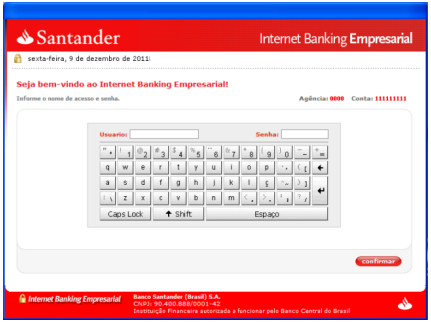


Figure: Passive Banker Malware for Santander bank waiting for user's credential input.

Figure: Passive Banker Malware for Itaú bank waiting for user's credential input.

Link: `https://dl.acm.org/doi/10.1145/3429741`

Brazilian Financial Malware on Mobile

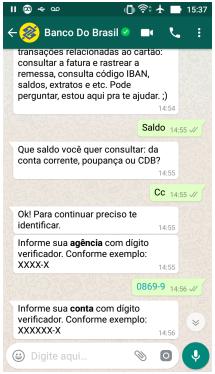


Figure: **BB's Whatsapp chatbot.**

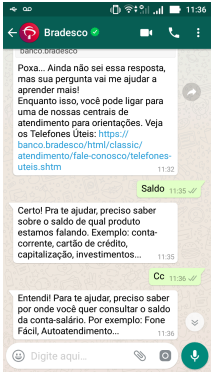
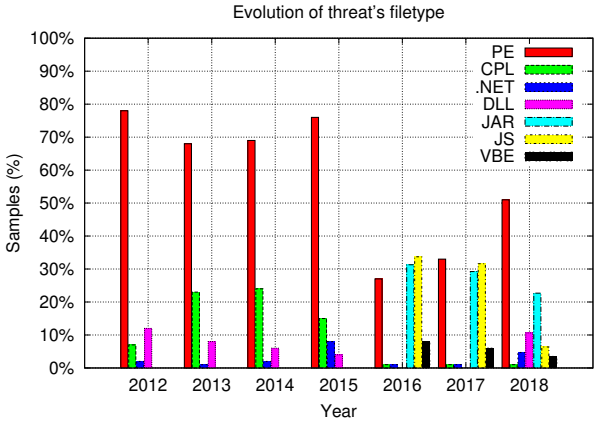


Figure: **Bradesco's Whatsapp chatbot.**

Link: `https://dl.acm.org/doi/10.1145/3339252.3340103`

Brazilian Financial Malware Filetypes.



Brazilian malware filetypes.

Varied file formats are prevalent over the years.

More about Brazilian Malware

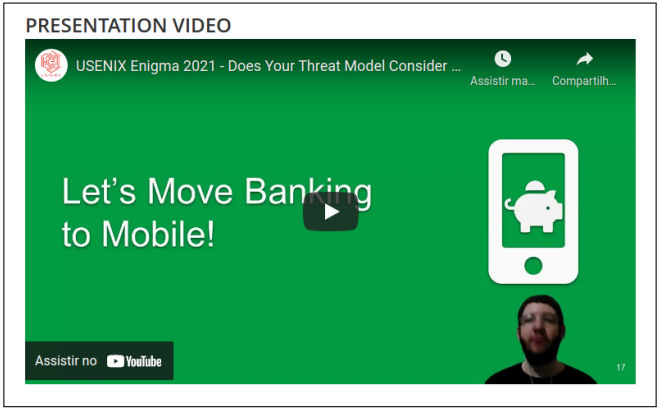


Figure: **Source:** https://www.usenix.org/conference/enigma2021/presentation/botacin

From Threat Intelligence to Endpoint Protection

Drawback: Real-time monitoring performance penalty

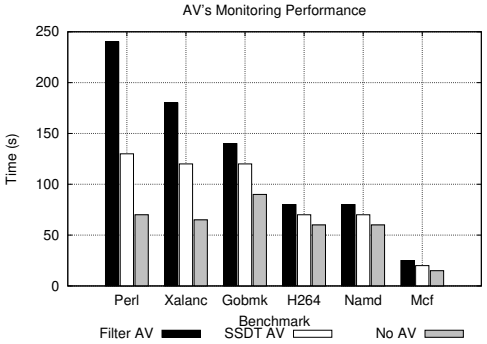


Figure: **AV Monitoring Performance.**

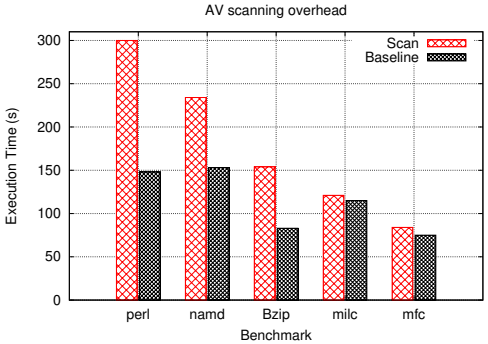
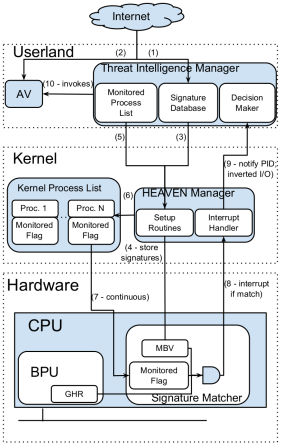


Figure: **In-memory AV scans worst-case and best-case performance penalties.**

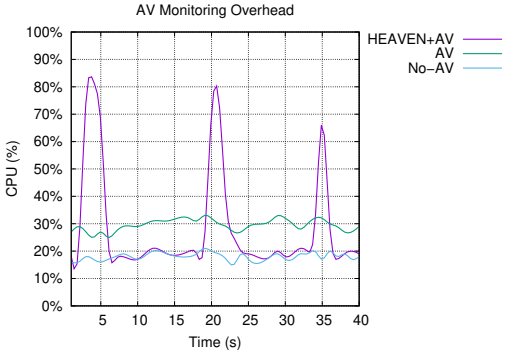
Hardware AV Architecture



2-level Architecture

Do not fully replace AVs, but add efficient matching capabilities to them.

Performance Characterization



2-Phase HEAVEN CPU Performance

The inspection phase causes occasional, and quick bursts of CPU usage. The AV operating alone incurs a continuous 10% performance overhead.

A first idea: Hardware features as signatures

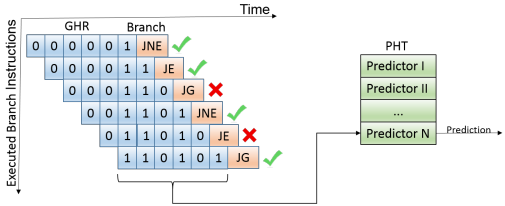


Figure: **Two-level branch predictor.** A sequence window of taken (1) and not-taken (0) branches is stored in the Global History Register (GHR).

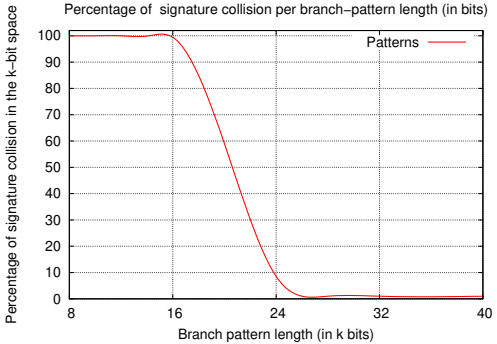


Figure: **Branch patterns coverage.**

Result: Performance penalty reduction

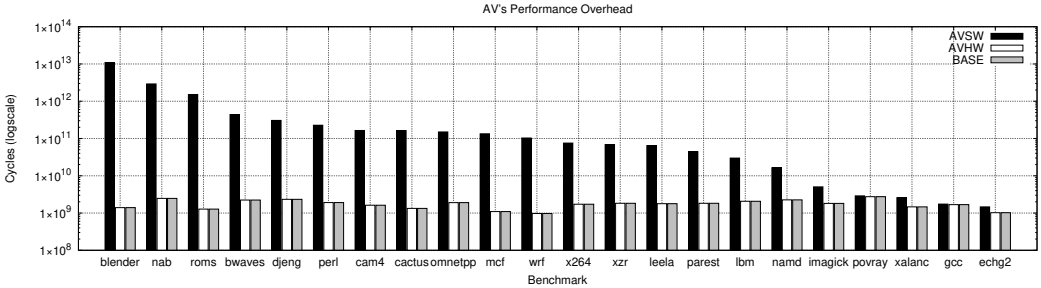


Figure: **Performance evaluation when tracking all function calls.** Comparison between execution without AV (BASE), execution with software AV, and execution with the proposed coprocessor model.

Topics

Deep Learning: From Images to Binaries

Malware Binaries as Textures

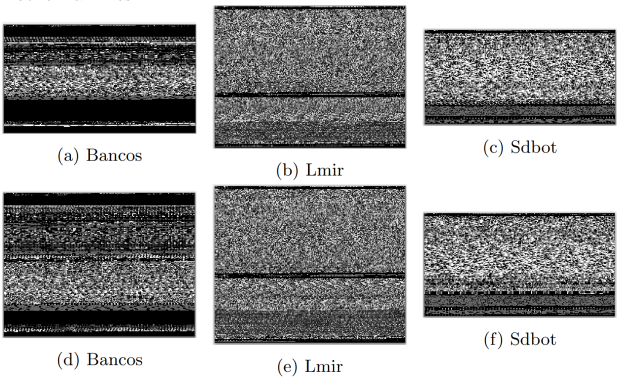


Figure: **Source:** `https://link.springer.com/chapter/10.1007/978-3-030-30215-3_19`

Adversarial Machine Learning Detection Bypasses

Adversarial Machine Learning

Adversarial Machine Learning: trend in recent years, as everybody knows

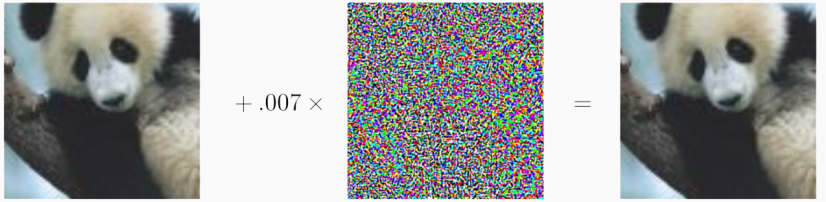


Figure: **Source:** `https://github.com/marcusbotacin/Talks/tree/master/Waikato`

Adversarial Malware

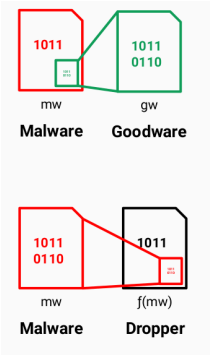


Figure: **Dropper Strategy.**

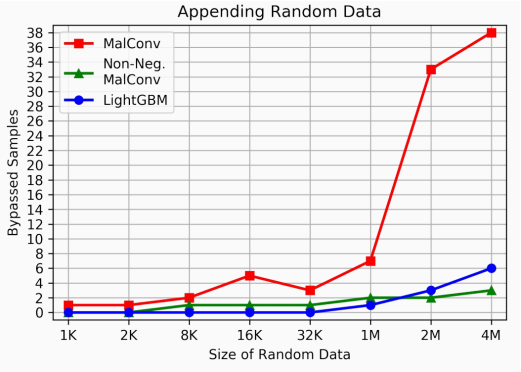


Figure: **Data Appendix Result.**

ML Evasion Contest

Figure: mlsec.io

Figure: https://cujo.com/machine-learning-security-evasion-competition-2020-results-and-behind-the-scenes/

Transition to Practice: Analysis Platforms

A Current Public Malware Analysis Platform

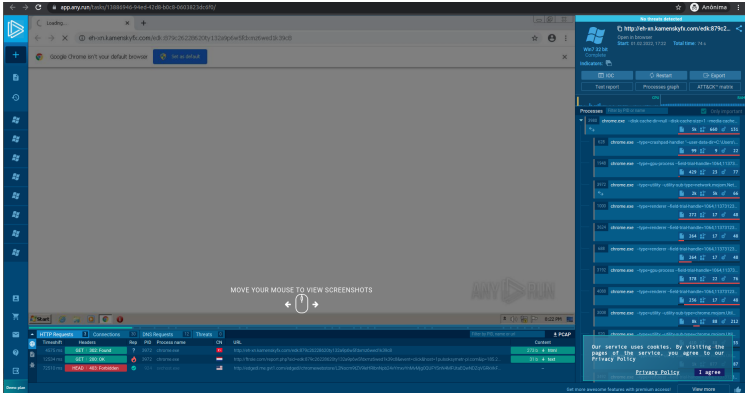


Figure: `https://app.any.run`

Topics

Summary

Malware Detection

- No definitive solution, but a pipeline of attempts.
- World is better with some approximation of security.

Academic Contributions

- Better Triage with Similarity Hashing
- Better Analyses with new Sandboxes
- Better Threat Intelligence for Brazilian Malware.
- Better endpoint protection with Hardware AVs

Moving Forward

- Open research positions. Get in touch!

Thanks!

Questions? Comments?

@MarcusBotacin
botacin@tamu.edu
marcusbotacin.github.io