Malware galéria: a kód vizuális arca

Attila Mester
attila.mester@ubbcluj.ro

Malware gallery: visualizing the code

Babeș–Bolyai University, Romania
Faculty of Mathematics and Computer Science

29th November 2025

The need for automated malware analysis

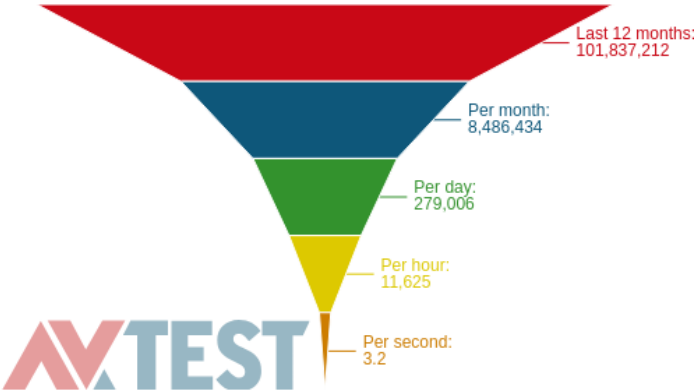


Figure 1: New malicious samples: $300k$/day, worldwide (AV-TEST[1], March 2025).

[1] https://www.av-test.org

Malware analysis – the textbook approach

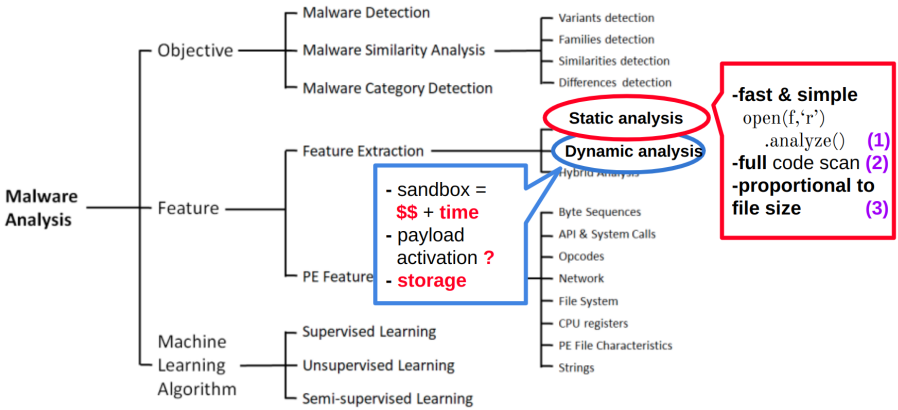


Figure 2: Malware analysis – taxonomy according to (Ucci, Aniello, and Baldoni 2019).

Survey of previous research on malware

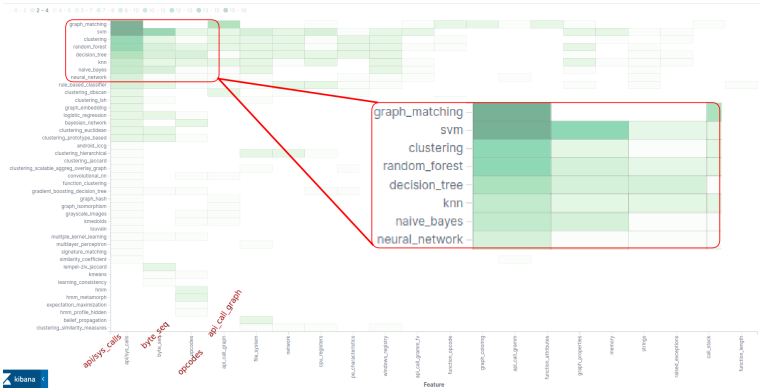


Figure 3: ≈ 100 research papers categorized according to *extracted features* and *algorithms*. Most frequent: *API/sys calls*.

Feature 1: byte histogram

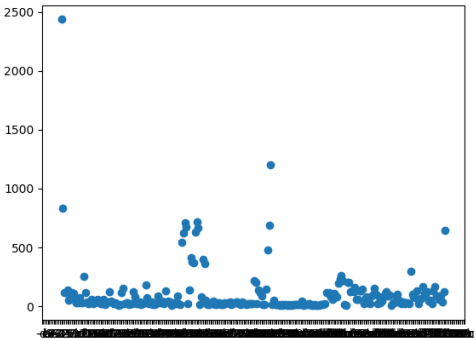


Figure 4: Byte distribution of infected Notepad exe – 1. infection

Feature 1: byte histogram

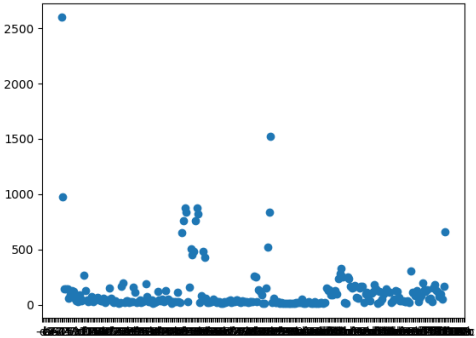


Figure 5: Byte distribution of infected Notepad exe – 2. infection

Feature 1: byte histogram

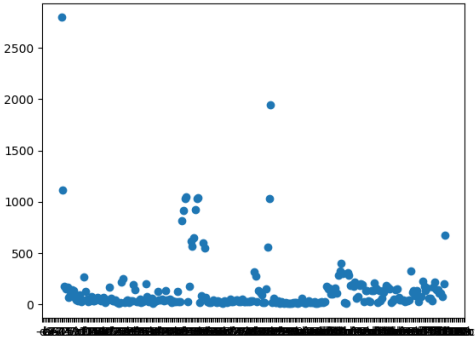


Figure 6: Byte distribution of infected Notepad exe – 3. infection

Feature 2: instruction counts

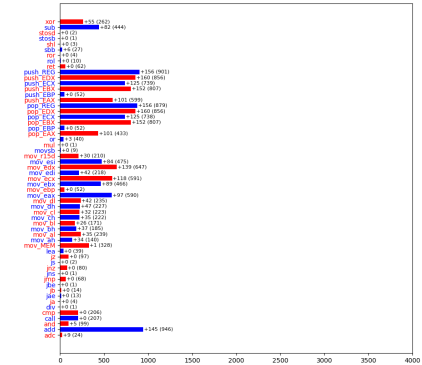


Figure 7: Instructions of infected Notepad exe – 1. infection

Feature 2: instruction counts

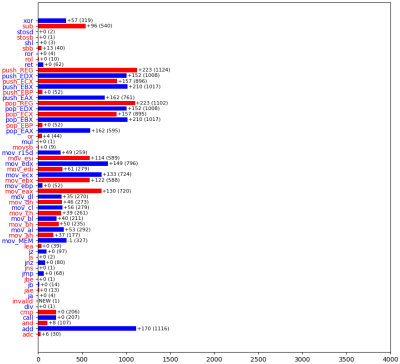


Figure 8: Instructions of infected Notepad exe – 2. infection

Feature 2: instruction counts

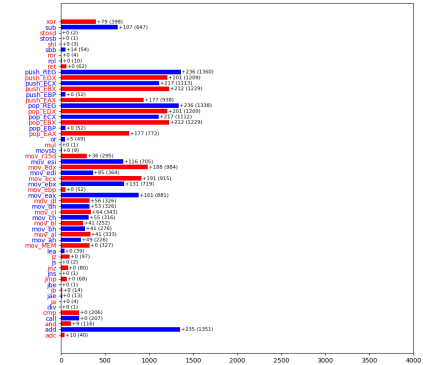


Figure 9: Instructions of infected Notepad exe – 3. infection

Feature 3: call graph

- static = obtained by disassembly – e.g. **IDA**[2], **Radare2**[3]
- node = function (DLL / subroutine)
- link = function call
- usage:
  - similarity measure? NP-hard
  - signature database? metamorphic evasion

[2] https://hex-rays.com/ida-pro

[3] https://rada.re

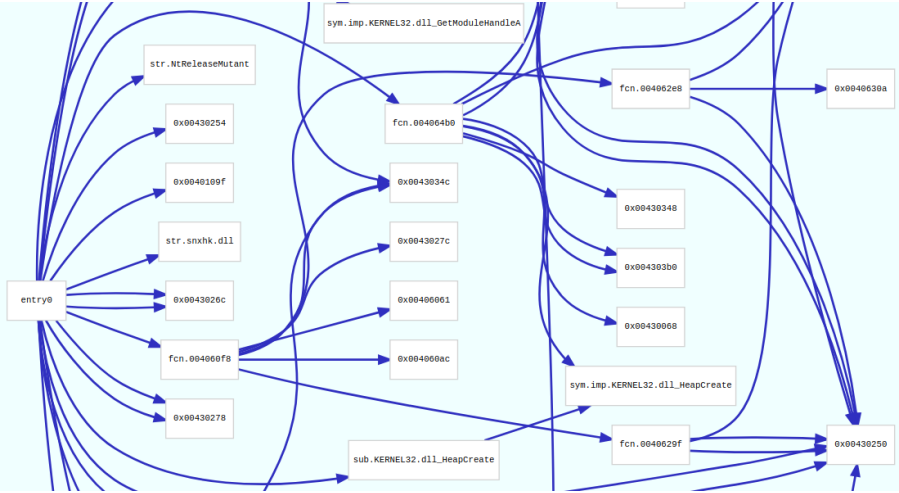


Figure 10: Sample call graph obtained by Radare2.

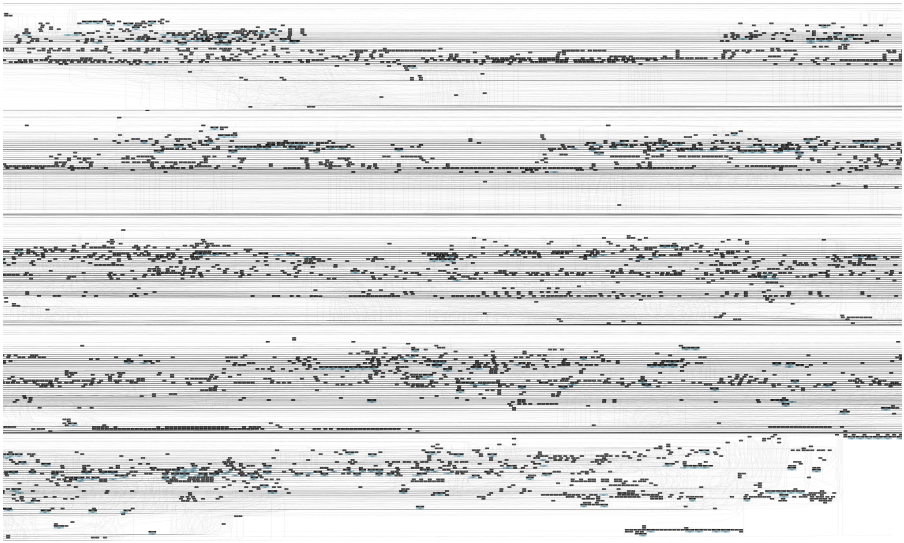


Figure 11: 8 MB malware file: 7k functions, 300k instructions

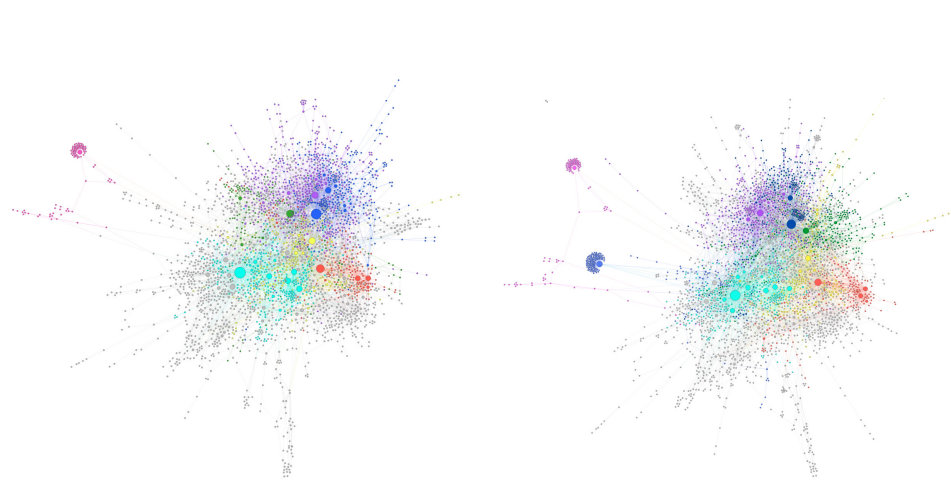
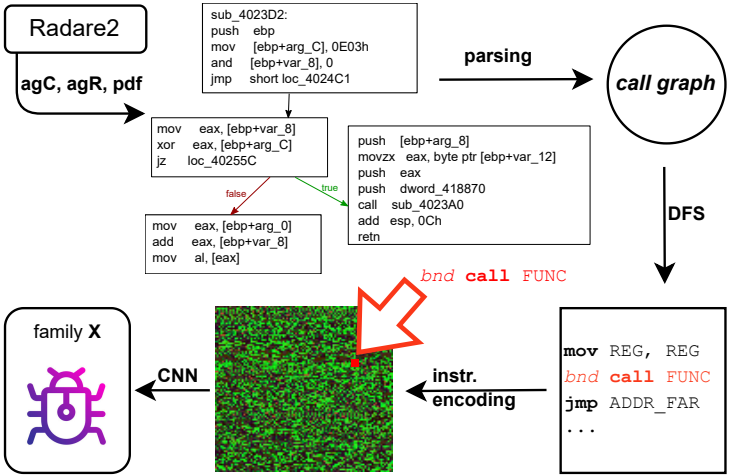


Figure 12: Static call graph of metamorphic virus generations – Gephi[4], Force Atlas layout.

[4] https://gephi.org

Malware family classification using CNN on the encoded call graph



Converting a malware into an RGB image

- **image**: list of instruction–pixels (900 instructions = 30x30 image)
  instruction: **[bnd?] [prefix?] mnemonic [param1 [param2, ...]]**
  2019 mnemonics, 11 prefixes
- **(FE)** Full Encoding: *mnemonic, prefix, bnd, two parameters*
- (PE1) Partial Encoding: mnemonic, prefix, bnd
- (PE2) Partial Encoding: only mnemonic

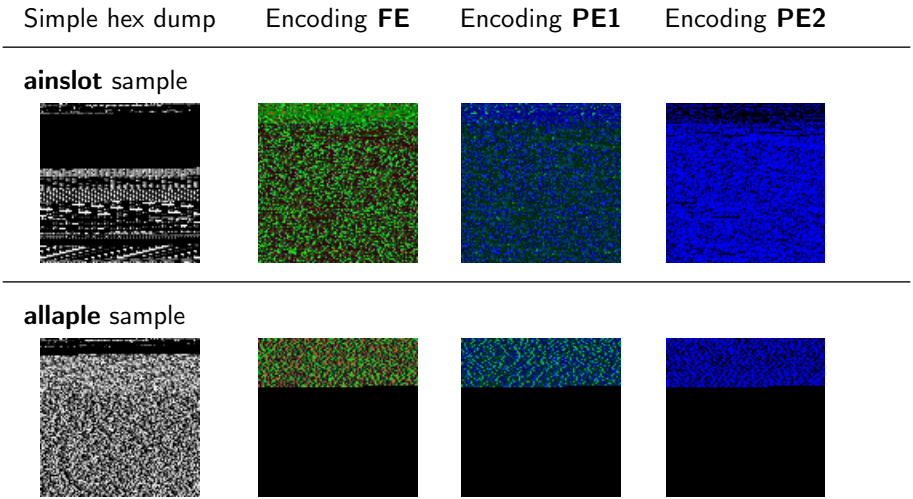


Figure 13: Comparing different instruction encodings and the simple hex dump image on different malware families.

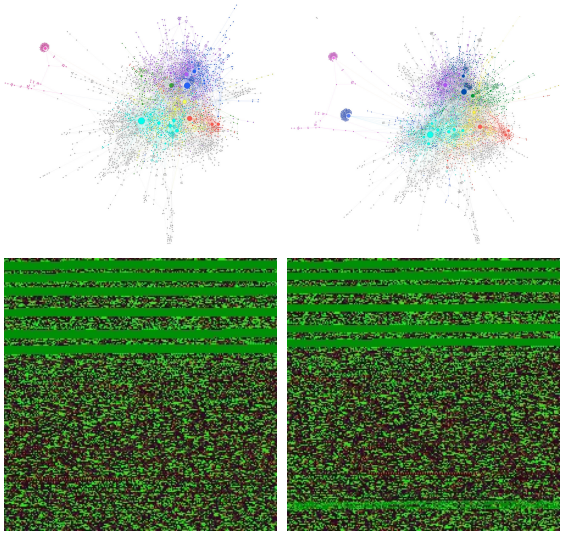


Figure 14: Call graphs and RGB images of metamorphic virus generations.

- **Code:** https://github.com/attilamester/malflow
- **Dataset:** https://www.kaggle.com/datasets/amester/malflow
- **Python package:** https://test.pypi.org/project/malflow/
- **Docker:** https://hub.docker.com/repository/docker/attilamester/radare2

```
$ malflow --help
usage: malflow [-h] [--version] {info,nodes,edges,export,image,dfs} ...

Malflow - Static analysis tool for PE executables using Radare2

positional arguments:
  {info,nodes,edges,export,image,dfs}
                        Available commands
    info                Analyze PE file and display call graph metadata
    nodes               List and query nodes
    edges               Display call graph edges
    export              Export call graph to various formats
    image               Generate image representation of call graph
    dfs                 Perform DFS traversal

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
```

Thank you for your attention! Questions?

attilamester.github.io/call-graph