# Poster: Beyond a Single Benign Class: What Functional Categories of Benign Software Reveal About Static Malware Detection

Viktor Medvedev✉
viktor.medvedev@mif.vu.lt
Vilnius University
Institute of Data Science and Digital Technologies
Vilnius, Lithuania

Juozas Dautartas
juozas.dautartas@mif.stud.vu.lt
Vilnius University
Institute of Data Science and Digital Technologies
Vilnius, Lithuania

Juozapas Rokas Čypas
juozapas.cypas@mif.vu.lt
Vilnius University
Institute of Data Science and Digital Technologies
Vilnius, Lithuania

Olga Kurasova
olga.kurasova@mif.vu.lt
Vilnius University
Institute of Data Science and Digital Technologies
Vilnius, Lithuania

## Abstract

Static malware detectors treat benign software as a single class despite differences in program purpose and API use. Binary malware evaluation reports false negatives but does not reveal which benign patterns are most associated with missed malware or which benign-category features most readily change correct detections. Using our previously introduced WinAPI-AdvMal dataset with five benign categories, we perform two evaluations. Among malware samples missed at a calibrated threshold by six multiclass models, Administration is the highest-scoring benign output in 34.3–46.8% of cases, versus 4.1–8.1% for Multimedia. Across four binary models, Multimedia-associated API additions produce the most decision changes and Development-associated additions the fewest, with differences of 25.2–69.8 percentage points. In controls on two tree-based models, category-derived lists produce 5.9–9.6 percentage points more changes than matched random benign groups. Thus, existing false negatives and misses after API additions are associated with different benign categories, revealing weaknesses hidden by binary evaluation of these API-import models.

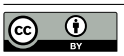
## CCS Concepts

• **Security and privacy → Malware and its mitigation**; • **Computing methodologies → Supervised learning by classification**.

## Keywords

static malware detection, PE files, Win32 API imports, benign software categories, model evaluation, adversarial machine learning

**ACM Reference Format:**
Viktor Medvedev, Juozas Dautartas, Juozapas Rokas Čypas, and Olga Kurasova. 2026. Poster: Beyond a Single Benign Class: What Functional Categories of Benign Software Reveal About Static Malware Detection. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3830454.3846425



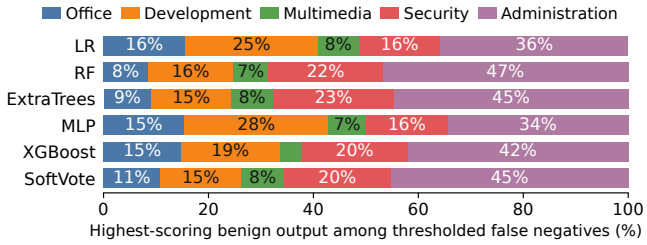
This work is licensed under a Creative Commons Attribution 4.0 International License.
*CCS '26, The Hague, Netherlands*
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3846425

## 1 Introduction

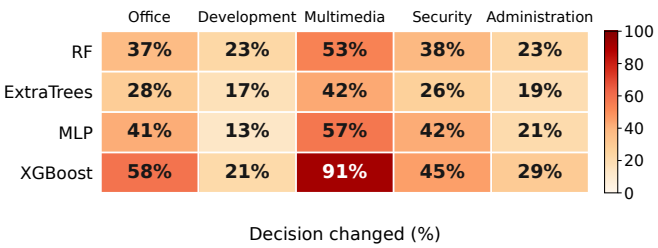
Static malware datasets commonly divide software into benign and malicious classes. Widely used resources such as EMBER and BODMAS follow this binary distinction [1, 6]. This representation supports standard detection evaluation but hides differences within legitimate software. Office, development, multimedia, security, and administration programs are all benign, although they use different Windows APIs. A missed malware sample is therefore recorded only as a false negative. A binary model does not show which benign software pattern is most associated with the error or whether APIs from different benign categories affect its decisions equally.

We previously introduced WinAPI-AdvMal, which contains five functional categories of benign software and one malware class [4]. Previous work used these labels for category-specific feature addition [2] and targeted generative evasion [3]. Related work such as MPass studies functionality-preserving evasion [5]. This study does not propose another attack. Instead, we ask what the category structure adds to model evaluation. We use the five labels in two complementary evaluations. For each malware sample whose score falls below the calibrated detection threshold, a six-class model records the highest-scoring benign output. Binary models are then tested by adding APIs from each benign category to malware that is initially detected correctly. The first evaluation describes errors already present in unmodified data, while the second shows which category-associated APIs most easily change a correct decision.

Our contribution is a category-aware evaluation of API-import malware models. We report baseline detection rates to show how much malware is missed before any API additions. We also compare real categories with matched random benign groups to test whether category-derived API lists cause more decision changes in initially detected malware. The two evaluations highlight different categories. Administration is most often the highest-scoring benign output among false negatives, whereas Multimedia-associated APIs most easily change correct decisions. The results can guide inspection of missed samples and testing after model updates, but apply only to classifiers using API-import features.




(a) Highest-scoring benign outputs




(b) Category-related decision changes

**Figure 1: Two different measurements enabled by the 5 benign labels. (a) Highest-scoring benign outputs among thresholded malware false negatives. (b) Fraction of initially detected malware that becomes missed after adding category-associated APIs.**

## 2 Using Benign Categories for Model Evaluation

WinAPI-AdvMal contains 3,799 Windows Portable Executable (PE) files represented by 2,713 binary API-import features [4]. We repeated five-fold cross-validation five times. Related or similar samples are kept in the same group to prevent overlap between training and test sets. Each training fold has separate partitions for detector fitting, API ranking, and threshold calibration. The malware-score threshold is calibrated on benign data to a target FPR of 2%. No test data are used for model fitting, API ranking, or threshold selection. We report means over five repetitions with paired 95% bootstrap confidence intervals.

We perform two complementary evaluations. First, six-class logistic regression (LR), random forest (RF), extra trees (ET), multilayer perceptron (MLP), XGBoost, and a soft-voting ensemble of LR, RF, and ET are used to identify the highest-scoring benign output among false negatives. A malware sample is a false negative when its malware score falls below the calibrated threshold. For each sample, we record the highest-scoring benign output among Office, Development, Multimedia, Security, and Administration.

Second, we evaluate binary RF, ET, MLP, and XGBoost models on malware correctly detected before modification. Each architecture uses samples detected by all three variants: one with a single benign class, one with five benign categories, and one with randomly shuffled benign-category labels. The same selected samples are used in all five categories. A separate LR trained only on the five benign classes ranks candidate APIs for the primary API-addition test. For each benign class, up to 100 APIs with the largest positive LR coefficients form a separate, potentially overlapping candidate list. For each malware sample and benign category, we add the 20 highest-ranked candidate APIs not already present in the sample. All selected cases permit 20 additions. The outcome is the fraction of malware samples missed after these additions. Only binary indicators change from 0 to 1. PE files are not modified and API calls are not executed.

As a separate control, we test whether real functional categories provide more useful API candidate lists than random benign groups. Random groups are formed by shuffling benign category labels without changing group sizes. For each malware sample, the real and random lists contain the same number of APIs not already present. For both lists, at each step we add the absent API that most reduces the current malware score. Only the source of the candidate APIs differs. The control uses 20 additions and 15 malware samples per fold (75 per repetition).

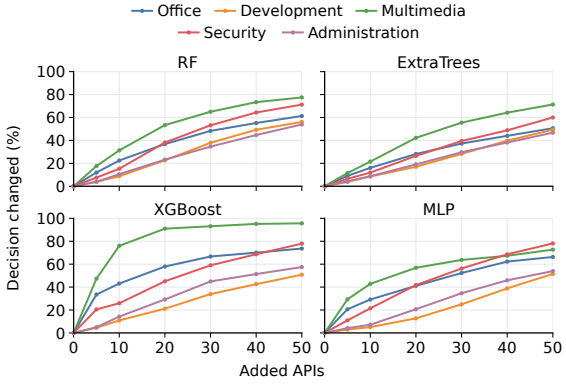



**Figure 2: Decision-change rates for 5–50 category-associated API additions. The primary comparison uses 20 additions.**

**Table 1: Baseline test results (means over five repetitions). Rates are percentages; FN and API-test are malware counts.**

| Model/setup | FPR | TPR | FNR | FN | API-test |
|---|---|---|---|---|---|
| LR/6-class | 2.07 | 53.1 | 46.9 | 171.8 | — |
| RF/6-class | 2.13 | 52.9 | 47.1 | 172.4 | — |
| ET/6-class | 1.95 | 53.3 | 46.7 | 170.8 | — |
| MLP/6-class | 2.06 | 49.8 | 50.2 | 183.6 | — |
| XGBoost/6-class | 2.13 | 57.4 | 42.6 | 155.8 | — |
| SoftVote/6-class | 2.15 | 58.0 | 42.0 | 153.6 | — |
| RF/binary | 2.42 | 52.9 | 47.1 | 172.4 | 179.4 |
| ET/binary | 2.53 | 54.5 | 45.5 | 166.6 | 183.8 |
| MLP/binary | 2.14 | 50.2 | 49.8 | 182.4 | 123.0 |
| XGBoost/binary | 2.42 | 59.3 | 40.7 | 149.0 | 192.6 |

## 3 Results

Table 1 shows test FPRs near the calibration target and TPRs of 49.8–59.3%. The second evaluation therefore covers only correctly detected malware, not the full malware set. Benign categories describe model errors, but these results do not show improved detection. Figure 1(a) shows that thresholded malware false negatives do

**Table 2: Category similarity to malware, highest-scoring benign outputs, and misses after 20 API additions.**

| Category | Cosine similarity to malware | Highest benign output | Missed after 20 APIs |
|---|---|---|---|
| Administration | 0.935 | 41.4% | 23.0% |
| Development | 0.926 | 19.8% | 18.4% |
| Security | 0.897 | 19.5% | 37.8% |
| Office | 0.887 | 12.4% | 41.0% |
| Multimedia | 0.882 | 7.0% | 60.8% |

not have one common benign output. Across the six models, Administration is the highest-scoring benign output for 34.3–46.8% of missed malware, whereas Multimedia is highest for only 4.1–8.1%. This output does not imply that the malware belongs to that software category or that it is the model's final prediction. Instead, it identifies the benign pattern most strongly reflected in the model's scores and can be stored with the missed sample for later inspection.

The second evaluation asks which benign-category APIs most easily change a correct decision. After adding 20 absent category-associated APIs, Multimedia produces the highest decision-change rate and Development the lowest for RF, ET, MLP, and XGBoost (see Figure 1(b)). The differences between these two categories are 30.7, 25.2, 44.1, and 69.8 percentage points, respectively. Thus, the same malware sample may remain detected when APIs from one benign category are added but become missed when APIs from another category are used. This result does not imply that APIs associated with Multimedia are malicious. It shows that the models are more sensitive to some benign categories than to others.

Figure 2 shows that category differences remain visible with 5–50 API additions. Development usually produces the fewest decision changes, whereas Multimedia remains among the most effective categories. At larger values, the ordering can change because many samples have already become missed. We therefore use 20 additions as the primary comparison. In the control experiment, real category API lists caused more misses than random-group lists. For RF, the miss rates were 74.1% and 64.5%, respectively, a difference of 9.6 percentage points (95% CI [6.1, 13.3]). For ET, the rates were 61.9% and 56.0%, a difference of 5.9 percentage points (95% CI [2.7, 9.1]).

The two evaluations rank the categories differently (see Table 2). Administration is most often the highest-scoring benign output among false negatives in unmodified data, but is one of the least effective directions when category-related APIs are added. Multimedia shows the reverse pattern. It is least often the highest-scoring benign output but produces the highest decision-change rate across all four binary classifiers. The contrast is consistent with the categories' average API-import profiles. Administration tools have the highest cosine similarity to malware (0.935), whereas Multimedia software has the lowest (0.882) and produces the most decision changes after API additions. API additions therefore generally produce more decision changes for categories less similar to malware, although Development-associated APIs are an exception. Because only five category profiles are compared, similarity supports but does not fully explain the pattern. False-negative analysis alone in unmodified data does not show which category-associated APIs most strongly change an initially correct decision.

## 4 Practical Use and Limitations

The analysis evaluates static malware models after training or updating. For each missed malware sample, the developer records which benign output has the highest model score. Correctly detected malware is tested by adding APIs from each benign category. These cases can be saved and checked after retraining. The analysis could guide training with added APIs from categories that cause the most decision changes, while monitoring TPR and FPR; benefits remain unverified. The dataset and API-addition procedure come from our previous work and are not new contributions. The contribution here is the use of these categories to describe model errors and to compare how APIs from benign categories affect the same malware samples. The experiments modify only API indicators without creating or executing modified PE files. The benign labels describe model outputs and feature directions, not malware families. Practical malware-evasion claims require real-file construction and sandbox testing. Results apply only to this API-import representation. Transfer to richer PE, byte-level, or behavioral features is untested, and differences in category size have not been isolated.

## 5 Conclusion

A single benign label counts malware errors but hides differences within benign software. The five functional labels show both which benign outputs are strongest for existing false negatives and which category-associated APIs most easily change an initially correct decision. These two evaluations lead to different category rankings: Administration is most strongly associated with existing errors, whereas Multimedia produces the highest decision-change rates. Real category API lists also outperform matched random benign groups, supporting the diagnostic value of these labels in this API-import setting. These results support model inspection, case selection, and repeatable testing after model updates. Future work could test whether training on malware with added category-associated APIs improves detection without increasing false positives.

## Acknowledgments

This project has received funding from the Research Council of Lithuania (LMTLT), agreement No. S-MIP-24-116.

## References

[1] Hyrum S. Anderson and Phil Roth. 2018. EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. arXiv:1804.04637

[2] Juozas Dautartas, Juozapas Rokas Čypas, Olga Kurasova, and Viktor Medvedev. 2026. Evasion of Malware Classifiers by Injecting Category-Specific Benign Features. In *2026 International Conference on Advances in Artificial Intelligence and Machine Learning*. IEEE, Piscataway, NJ, USA, 102–109. doi:10.1109/AAIML67890.2026.11498090

[3] Juozas Dautartas, Olga Kurasova, Juozapas Rokas Čypas, and Viktor Medvedev. 2026. Learning to Look Benign: Targeted Evasion of Malware Detectors via API Import Injection. arXiv:2605.18624

[4] Juozas Dautartas, Olga Kurasova, Juozapas Rokas Čypas, and Viktor Medvedev. 2026. WinAPI-AdvMal: A Six-Class Windows API Import Dataset for Adversarial Malware. doi:10.5281/zenodo.20208958

[5] Jialai Wang, Wenjie Qu, Yi Rong, Han Qiu, Qi Li, Zongpeng Li, and Chao Zhang. 2023. MPass: Bypassing Learning-based Static Malware Detectors. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, Piscataway, NJ, USA, 1–6. doi:10.1109/DAC56929.2023.10247858

[6] Limin Yang, Arridhana Ciptadi, Ihar Laziuk, Ali Ahmadzadeh, and Gang Wang. 2021. BODMAS: An Open Dataset for Learning Based Temporal Analysis of PE Malware. In *2021 IEEE Security and Privacy Workshops*. IEEE, Piscataway, NJ, USA, 78–84. doi:10.1109/SPW53761.2021.00020