Beaver: Estimating Future Risks at Scale in Real-World Deployments

Marco Balduzzi
Trend Micro Research

Roel Reyes
Trend Micro Research

Jessica Balaquit
Trend Micro Research

Ryan Flores
Trend Micro Research

***Abstract*—Malware continues to pose a significant threat to organizations worldwide, with various forms of malicious software enabling criminal activities. To protect against these threats, security solutions such as anti-malware and intrusion-detection-systems have been introduced over the years. However, while these solutions work well, especially when combined, they tend to detect attacks only when they are already happening. In this paper, we adopt a proactive strategy aimed at anticipating threats before they occur. We introduce a system that leverages the activities of users on their machines and over the Internet to predict future malware outbreaks. Our solution estimates the risk for different classes of malware, enabling organizations to proactively implement mitigation strategies tailored to their risk profiles. We deploy our implementation in a real-world setting and conduct a large-scale risk study across 10.7 million endpoints collected over a period of one month. Our empirical study provides insights on the behaviors that most significantly put users at risk, the categories of endpoints that are most vulnerable to specific malware, the distribution mechanisms used to operate malware campaigns, among other findings we share with the community.**

I. Introduction

Malware outbreaks still represent a major problem for organizations worldwide [1], [2] with ransomware infections alone costing millions of dollars on average to affected organizations [33]. Criminals are still leveraging malware, and other various forms of malicious software to enable their attacks: trojans, for example, are used in maintain persistence into compromised networks, while hacktools in conducting lateral movement from initial infection to primary targets such as confidential information stored in internal databases.

To protect organizations against cyber-crime, the industry has introduced several security solutions over the years, with anti-malware playing a central role in defending against outbreaks. This involves detecting the presence of malicious software on an endpoint, such as upon download, and then quarantining it. Intrusion-detection-systems and network-detection-response platforms inspect network traffic through known signatures of attack patterns to detect anomalies, which could be due to the presence of an infection or ongoing attack. More recently, in response to the fact that modern attacks are becoming more sophisticated and are challenging the capabilities of traditional solutions [60], platforms such as EDRs have been progressively introduced in the market.

While all these solutions, especially when combined, provide a comprehensive strategy against cyber-attacks, they share a common shortfall: they employ a *reactive approach*. Anti-malware, anti-spam, web-application-firewalls, and intrusion-detection-systems all operate by waiting for evidence of an attack (i.e., detected by a signature) before triggering an alert or initiating remediation. While this is effective in most cases, it does not allow an organization to strategically plan its defense. In contrast, a *proactive approach* enables an organization to *anticipate* potential outbreaks and stay ahead of future attacks performed by miscreants.

In this work, we build upon our previous study [62] and introduce a system that estimates the risk of future malware outbreaks. Our system not only forecasts potential outbreaks within the next 30 days, but also provides a breakdown of the risk probabilities by malware class. Additionally, it offers explainable reasoning behind these predictions.

We believe this aspect is very important: each organization is susceptible to different classes of attacks e.g. hospitals are known to be a primary target for ransomware due to the sensitive nature of the data they handle (i.e., patient data); similarly, each machine is also susceptible to different classes of infection e.g. Windows desktops operated by regular employees in an organization are often targeted by PUAs or adware. Being able to profile the risk for different classes of malware enables organizations to better strategize their defenses.

Our approach consists of employing behavioral user data to estimate the risk of future attacks. Studies such as [62] have previously shown that improper user behaviors, such as visiting shady websites, put users at risk of infection. We build on top of these findings by extending the methodology and introducing a system for analyzing real-time data at scale.

We introduce a novel tool, which we name *Beaver*, to conduct a large-scale study on a population of 10.7 million endpoints collected from our global-scale telemetry in one month. We provide insights into the effective risks faced by 822 organizations with respect to various classes of malware. Additionally, we discuss the behaviors that put these organizations at risk and the categories of endpoints that are most vulnerable to specific classes of malware.

To summarize, our contributions are the following:

- We build upon our previous study and introduce a system designed to estimate future outbreaks and associated risks.

979-8-3315-8969-1/25/$31.00 ©2025 IEEE

- We implement our approach in a tool named *Beaver*, which automates data collection, analysis, and risk prediction.
- We conduct a large-scale study involving 10.7 million endpoints, collected from our telemetry over one month.
- We discuss our findings, including the behaviors that most significantly put users at risk and the categories of endpoints that are most vulnerable to specific classes of malware.

Our paper is organized as follows: Section II introduces the architecture of our system, consisting of a data acquisition and pre-processing layer, and two macro-components designed to estimate the risk of future malware outbreaks. Section III provides a summary of the datasets employed for both training the system and for conducting our large-scale study. Section IV describes in detail the implementation of our system and how we conducted training (and testing). Section V explains how the system was used to predict future malware outbreaks and discusses the results of our study. We provide the related work in Section VI and conclude in Section VII. Further details are included in the Appendix.

II. System Overview

We begin our work by giving an overview of the approach we introduced for estimating future malware outbreaks.

Given a time t, our approach consists of training a system on the past activities of the users (at $t - 30days$), and use current and future malware incidents (at $t + 30days$) for labeling. This strategy enables the system to estimate a prediction for an endpoint e at time t for the upcoming window.

The system operates in two phases. As shown in Figure 1, one is used for training (on the left) and one for estimating the risk of future malware outbreaks (on the right).

While several other components are needed to handle the processing e.g. one to extract the users' behavior from their activities on the endpoints, the main tasks are handled by the *Odds-Ratios Generator (ODG)* and the *Multi-Label Classifier (MLC)*.

The Odds-Ratios Generator is built on top a previous study [62] which extends with the risk estimation module (i.e., *ODG prediction* in Figure 1) and other internals we detail later in Section II-A. In our system, we combine this component with a new Multi-Label Classifier which leverages supervised-learned models to refine the risk estimation R. This is made for each malware class under analysis, which in our current setup consists of coinminer, hacktool, PUA, ransomware, trojan, and virus as per definition in Table I.

Both these macro-components (i.e., ODG and MLC) rely on a pre-processing phase in which activity data (that we discuss in detail in Section III) are loaded from our telemetry and pre-processed accordingly. A first component extracts the users' behavior from the activity of the users on the endpoint, e.g. the network traffic generated from the applications installed on the machines, the software regularly downloaded from the Internet, the categories of software executed and websites visited, the frequency of machine utilization, etc.. These information are passed to a second component which generates the features used by ODG and MLC. Note that both these components rely on the same set of features (presented in Table IV), but the way in which are used is different – as we discuss later.

Malware Class	Description
Coinminer	Mines cryptocurrencies
Hacktool	Used for local hacking or lateral movement
PUA	Potentially Unwanted Application
Ransomware	Ransoms the user to request a payment
Trojan	Disguises as a legitimate program
Virus	Infects endpoints automatically

TABLE I: Malware classes considered in this work and their definition.

Concurrently with this execution, another component labels the endpoint as "infected" or "not infected" (and the related malware class) based on whether the endpoint encounters malware in the following 30 days. Since our goal is to capture the effect where *it is* the user's behavior that triggers a malware infection (e.g., visiting shady websites or installing untrusted applications), we focus exclusively on behaviors directly related to the user, disregarding any automated executions by applications. In addition, we only consider first-stage infections; that is, we disregard endpoints that were previously infected.

The labeling is used in the training phase, i.e. by the odds-ratios generator for generating the case-control groups as discussed in Section II-A and by the multi-label classifier to train the models; the features are instead used both in the training phase as well as during the risk estimation phase to compute the risk score for a given endpoint e at real-time.

In this phase, the system operates by fetching the activity of the endpoint's user from the telemetry in *real-time*, extracting the user's behavior, generating the features, and using that information as input for the system to generate the risk score $R = \{R_1, R_2, ...R_i\}$, where i represents the number of malware classes taken into consideration[1] as an average of the risk scores estimated by the individual components (i.e., ODG and MLC).

A. Odds-Ratios Generator (ODG) - Design

This component is tasked with generating the odds-ratios for the dataset under analysis, which, in our case, consists of behavioral user data and malware infection. In statistics, odds-ratios are mathematical formulations that quantify the association between two events. Making an analogy with the medical domain, this is, for example, the likelihood that smoking a high quantity of tobacco per day would result in a lung cancer. The process of generating an odds-ratio consists of dividing the training population in two groups: those ones that are exposed to a risk, for example because smoking more than 10 cigarettes a day, from the others. These groups are then divided in two more groups, i.e. those ones that developed cancer in the future, and those that did not. This process

[1] In this work, we consider 6 malware classes.

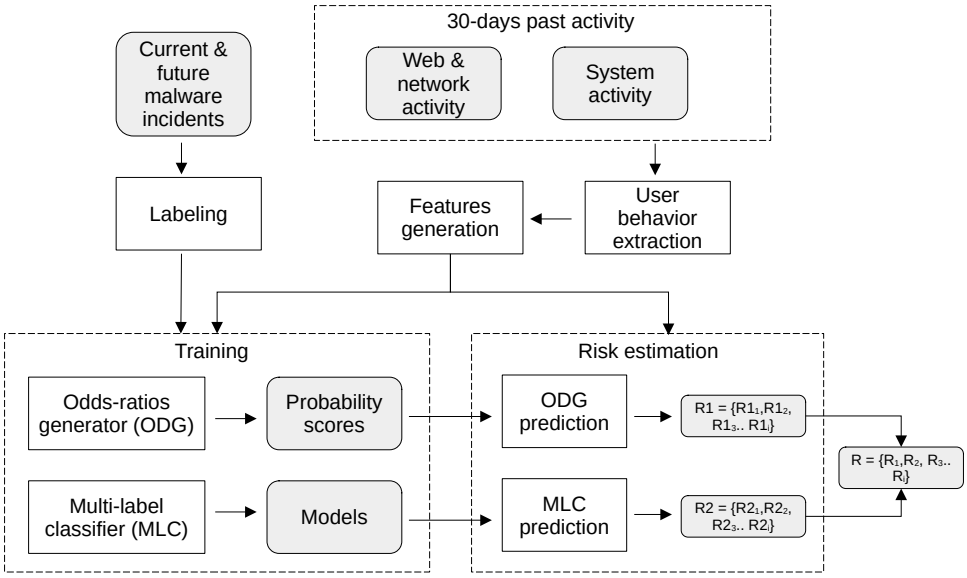


Fig. 1: *Beaver*: System Architecture.

Visited domains per day (avg)	Encountered ransomware	
	Yes	Not
>200	a	b
<=200	c	d

TABLE II: Example of *contingency table* used as input for generating the odds-ratios.

results in so-called contingency table, which is used as input to compute the odds-ratio.

Table II provide an example that applies to this work: the case group is composed of endpoints that encountered ransomware ($a + c$), while the control group consists of endpoints that did not encounter the malware ($b+d$). The odds ratio OR is the ratio between the odds of exposure among cases ($\frac{a}{c}$) and the odds of exposure among controls ($\frac{b}{d}$). As an example, an OR of 2.3 indicates that visiting more than 200 domains per day increases the probability of being infected by ransomware of 230%.

During training, this process is computed through all the features and all malware classes. The implementation of this execution is described in Section IV-B, and the results are presented in Tables IV and V.

Following the training phase, the same component is used to estimate the risk of endpoints at real-time. In this configuration, the system operates by fetching the past 30-days of endpoint's activity from the telemetry, extracting the user behavior and computing the features. The estimate risk R_i for a class of malware i is computed with the following formula:

$$\begin{aligned} S_i &= \sum \left(\log\left(\text{OR}_j\right) \times \text{SignificantFeature}_j\right) \\ R_i &= \tfrac{1}{1+e^{-S_i}} \end{aligned} \tag{1}$$

Note that only the features that are statistically significant are used (i.e. those ones with p-value < 0.05 in our implementation, as explained later in Section IV-B).

B. Multi-Label Classifier (MLC) - Design

This component implements a collection of independent binary classifiers, each trained to distinguish between a specific malware class and normal (i.e, control) endpoint's behavior. Unlike traditional multiclass classification approaches where the probabilities across all classes sum to 1, this architecture treats each malware prediction task as an independent binary classification problem. This design enables the simultaneous prediction of multiple malware classes, and delivers estimations that can collectively exceed unity when multiple threats are simultaneously present.

The training process begins by constructing datasets for each malware class using a 1:2 ratio of endpoints that encountered malware to normal samples. For a given malware class i with n_i endpoint samples, we create a corresponding dataset by pairing all n_i malware encountering samples with a randomly sampled subset of $2n_i$ control samples from the control population. This approach ensures class balance while preserving the statistical properties of normal endpoint behavior. The sampling is performed without replacement across different malware classes to maintain dataset independence and prevent overfitting to specific normal behavior patterns. For computational efficiency, control endpoints are randomly sampled, acknowledging that this may introduce some demographic bias compared to the approach adopted by ODG (i.e., case-control groups).

During the risk estimation phase, the system processes endpoint's behavioral data through all trained classifiers simultaneously. Following, the random forest methodology proposed by Breiman [62], each independent binary classifier for malware class i outputs a risk score R_i representing the likelihood that the endpoint exhibits behavior consistent with that specific malware encountered. The risk score is computed as the proportion of trees in the forest that vote for the positive

class (malware), which converges to the true probability as the number of trees approaches infinity. Each risk assessment is computed independently as:

$$R_i = \frac{1}{T} \sum_{t=1}^{T} I(h_t(\mathbf{x}) = i) \tag{2}$$

where R_i is the independent risk score for malware class i, T is the number of trees in the forest, $h_t(\mathbf{x})$ is the prediction of the t-th tree for input $\mathbf{x}$, and $I(\cdot)$ is the indicator function. This approach allows for simultaneous detection of multiple malware classes, as each classifier operates independently without requiring probabilities to sum to unity, enabling security analysts to assess and prioritize response efforts based on individual threat-specific risk scores.

III. Datasets Overview

In this Section, we provide an overview of the datasets that we used for both computing the odds-ratios and for training (and testing) the multi-label classifier. The datasets are collected by our telemetry though a series of sensors which are integrated in anti-malware and similar endpoint solutions deployed on Windows endpoints.

The process is the same for all endpoints and the data format is consistent across the different versions of the solutions. Overall, we leveraged the following telemetry datasets:

- The first dataset (named *WRS*) consists of information associated with the web reputation system of the security solutions. This datasets include the URLs contacted by an endpoint, e.g. using a browser or any HTTP-enabled application, together with the timestamp and the associated user-agent. Each request is associated with both a safety score and a category score assigned by an internal classification mechanism known as web reputation system.
  The safety score indicates the risk level associated with the visited website and takes the following values: "normal" (non-malicious website), "suspicious" (potentially linked to spam or compromise), "dangerous" (confirmed to be malicious), or "unclassified". The category score indicated the category the contacted URL belongs to. For example, the category "Adult" is associated with sites that may be considered inappropriate for children, while "Business" includes sites related to business, employment, or commerce. To determine the popularity of the visited websites, we rely on Tranco Top 10k [44].
  Note that since we are interested in collecting activities from the users only, this dataset is processed by the system to only retain user-generated traffic i.e. excluding automated-genterated traffic like OS updates. We discuss more about this in Section IV.
- The second dataset (named *CS*) include information on all binaries (i.e., .exes) executed on the machine and the associated loaded dynamic libraries (i.e., dlls) together with their timestamp, signer information and version.
  We also collect information on the operating system, the country, and the industry (e.g. healthcare or manufacturing) of the organization in which the endpoint is deployed.
  We enriched this dataset with information about the application category tied to an executable. For this, we adopted the approach of Bilge et al. [3] and compared the executable names against Capterra (a service that classifies popular enterprise software) to identify their categories.
- The third dataset (named *VS*) consists of malware events extracted from security incidents detected on the endpoint by the security solutions. Some examples are malware dropped through drive-by-downloads or attached to emails. These information include the name of the detected malware, including class (e.g. ransomware), SHA1 and timestamp. We adopted CARO's naming convention [65].

A. Ethical Considerations

Similar to prior research in this field [40]–[42], we believe that realistic experiments are essential for conducting reliable research, such as studies in real-world deployments. Aware of this aspect, we prioritized user privacy and the sensitivity of the collected data. Specifically, we excluded endpoint identifiers (such as IP addresses) from our datasets and anonymized or removed any information that could reveal user identity. Users had given explicit opt-in consent for the collection of telemetry data, intended for post-sale support, threat intelligence, and additional product improvement research. There was no correlation among datasets, for example, from outside the company, which could assist in deanonymizing individuals. The datasets were only accessible within the company's secure environment to prevent unauthorized data extraction, and aggregated data was accessed by researchers through a secure repository requiring individual credentials.

IV. Training

In this Section, we discuss how we conducted the training for our system. For both *ODG* and *MLC*, we employ an initial set of 2 months of data (November and December 2024), totaling a number of 12,320,026 endpoints. These endpoints consist of Windows OS versions 7, 8, 10, 11 and Vista in all their flavor (e.g., Home, Pro, Enterprise, etc..) and languages. The dataset span over 208 countries, making it representative of the general behavior of users world-wide. Figure 2 provide an overview of the top 20 countries.

For each endpoint, the training set consists of the user's behavior over the previous 30 days, such as the webpages visited by the user and the applications installed, followed by a subsequent event suggesting a malware incident, e.g., a binary quarantined by the anti-malware solution. We did our best to ensure that the events are related to first infections (i.e., excluding malware dropped by previous infections) and to only retain behaviors directly attributable to the user (i.e.,

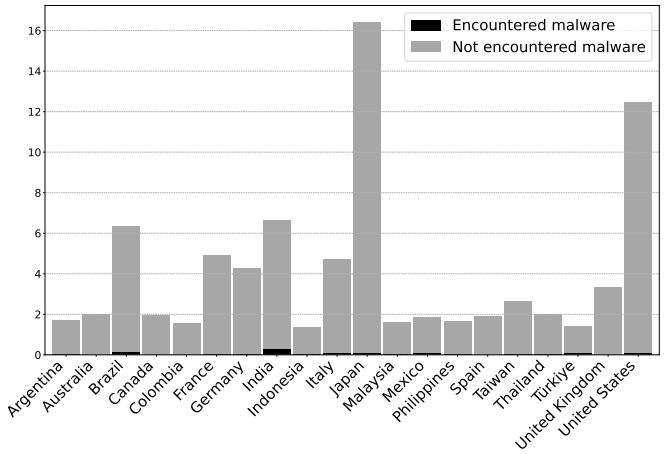


Fig. 2: Country distribution for endpoints that *encountered* malware and endpoints that *did not* (top 20). Y-axis = % of population.

excluding any activity performed by software installed on the machine).

Following this pre-processing phase, the cleaned datasets are passed to the odds-ratios generator and the multi-label classifier. The result of the training (in form of probability scores and models) is used in the risk estimation phase to compute the risk score for a given endpoint e at real-time. We discuss later in Section V how we used our tool to conduct the large-scale study across 10.7 million endpoints in an automated way.

A. Experimental Setup

In developing our system, we designed an architecture that guarantees quick training and classification operations. In this section, we outline the setup, resources, and configuration utilized in this process.

The data repository for our models was located in Amazon's S3, and we utilized an EC2 instance for computation. This instance featured 32 cores, 256GB of RAM, and a primary storage of 1TB. We allocated an additional 2TB of drive space for PySpark, which eliminate storage capacity challenges while transferring data from S3 to local storage. The EC2 instance was provisioned with the permissions to access the S3 bucket, essential for PySpark operations. JupyterHub was installed to enable coding and testing, employing Python version 3.12 along with several modules such as pandas, numpy, pyspark, and s3fs. This configuration guarantees efficient data retrieval and processing within the Spark environment.

We also optimized PySpark at different stages of data processing, including collection, preparation, sanitation, feature extraction, and training, with the goal of reducing memory allocation and improving file handling capabilities. For example, we increased both executor memory and executor cores to support operations related to data preprocessing and feature extraction (i.e., +50% and +30% respectively).

Performance The collection and preprocessing of the training set took 30 minutes for one day of *WRS* logs, 20 minutes for *CS* logs, and 10 minutes for *VS* logs, totaling 61 hours (approximately 2.5 days) for the entire training set. Extracting user behaviors and generating features required an additional 1.5 days. The ODG component took 2 days to generate the case-control groups and an additional day to compute the odds-ratios.

The MLC component exhibited the following performance metrics: hyperparameter tuning, which optimizes parameters for each malware class independently using cross-validation, required a total of 10 hours across all six malware classes. The data processing and training phases for each independent classifier took approximately 7 minutes. This duration included feature preparation, dataset balancing, and the training process itself. The per-classifier training time supports a model-rolling architecture, allowing classifiers to be retrained as new training data becomes available.

During prediction, the system estimates risk scores Ri with an average time of *155ms* per endpoint e and malware class i. This processing time results from optimizations including vectorization of the feature scaling step. Thus, it can be concluded that the prediction time supports real-time risk estimation deployment.

B. Odds-Ratios Generator - Implementation

As we discussed in Section II-A, the approach we adopted for generating the odds-ratios (i.e. the *training* phase of the odds-ratio generator (ODG) component) follows previous work [62]. Some notable improvements are: we extended the labeling period D_P from 2 weeks to 1 month (i.e. the month of December), we extended the number of features to 28 to better capture the behaviors of the users, we included a wider set of 218 countries in the training, and we leveraged a wider population of 12,320,026 endpoints and 1,472 organizations.

Following the pre-processing phase, in which we only included endpoints initially compromised (i.e., we did not consider those endpoints that could have been "re-compromised" due to a previous infection) and sampled the initial set with a ratio of 1/1,000 to reduce computation overhead, we constructed the case-control groups by considering a ratio of control to cases of 3 to 1, i.e. for each case we enrolled 3 control endpoints matching the following endpoint characteristics: country, type of organization (large or SMB), operating system (flavor, version and language), and industry sector.

Table III provides an overview of the 102,424 case-control groups used for the generation of the odds-ratios. While PUAs and trojans account for a good amount of the groups because of their large presence in the malware ecosystem (i.e., numerous families and high popularity of adoption by attackers), minor and more specialized malware classes such as coinminers and ransomware account for about a thousand groups each making our training set large enough for generating the odds-ratios.

	Case	Control	Total
Coinminer	249	747	996
Hacktool	5,148	15,444	20,592
PUA	12,483	37,449	49,932
Ransomware	217	651	868
Trojan	5,781	17,343	23,124
Virus	1,728	5,184	6,912
Total	25,606	76,818	102,424

TABLE III: Case-control groups created during training. Alphabetically ordered per class of malware.

We now report the thresholds used for discriminating between the population exposed to risky behaviors (e.g., visiting a large number of distinct categories of websites per day) and safe behaviors (e.g., visiting trusted websites only). As shown in Table III, different features have different thresholds: we employ both *static thresholds* for features where, for example, visiting even a single malicious website would put the user at risk, and *dynamic thresholds*, which are computed by the system using the 3rd quartile of the feature's distribution. Note that for some features, we employ both types of thresholds: this is because we conduct a first pass to only retain those endpoints that exceed normal activity (e.g., visiting a number of URLs per day that surpasses what would be typical OS updates) and then perform a second pass to actually identify the endpoints at risk.

As an example, feature #21 captures the average number of distinct software products used by the user in a day. What happens in practice is that if a user regularly employs more than 119 applications, this behavior is considered *exposed* to risk because above threshold. This information is used by the algorithm in generating the contingency table needed for computing the odds-ratios (ref. Table II).

The last step of the training consists of computing the odds-ratios for the different classes of malware. These probabilities express the *likelihood* that a certain behavior puts the user at risk. The algorithm works by taking the contingency tables for each malware class and risk factors, and compute the odds-ratio using the χ^2-test. To avoid division by zero in computing the ratios, we applied Haldane's correction (add 0.5 to all cell if any of them is zero). We considered the exposure effect to be statistically significant if the p-value obtained from the chi-square test is < 0.05. In addition to the p-value, we also computed the 95% Confidence Interval (CI) [39] that provides information about the range in which the odd lays with 95% probability. Given a CI interval, we can determine if the test is statistically significant if it does not include the value 1, called value of zero effect.

The result of the odds-ratios generation is shown in Table V. For example, installing a large variety of software applications (more than 159) increases the probability of encountering backdoored software, e.g., with trojans, by 161% (feature #22). This is understandable because in a desktop environment where users are allowed to download and install any application found over the Internet (i.e., no enterprise policy defining application whitelists is in place), there is a higher chance that some of these applications would contain malware.

Similarly, endpoints found to have a multitude of different concurrent signed applications on disk, including self-signed ones (i.e., more than 31), have double the chances of encountering hacktools – this is because an attacker could use that machine to disguise his presence and conduct lateral movement (feature #23).

Interestingly, visiting gambling websites (feature 10) exposes to the risk of PUA, Trojan and Hacktool, but not to Coinminer, Ransomware or Virus. Our explaination for this is that certain classes of malware are more likely to be associated to certain categories of websites, e.g. because of the distribution model of the cybercriminals behind the malware campaigns.

C. Multi-Label Classifier - Implementation

As we discussed in Section II-B, the multi-label component consists of independent binary classifiers i.e. one for each class of malware. The training methodology follows a similar data pre-processing pipeline of the ODG component, i.e. utilizing the same set of 28 behavioral features extracted from endpoint telemetry data.

For training and validation, we performed an 80/20 split on the November-December 2024 dataset. For testing, we employed the January 2025 dataset after removing the malware-positive samples.

The training process begins with hyper-parameter tuning performed independently for each malware class, which evaluates different combinations of model parameters using cross-validation. Each binary classifier is then trained independently using the optimized hyper-parameters, allowing specialized decision boundaries tailored to each malware class.

Following initial training, comprehensive performance evaluation was performed for each binary classifier through ROC curve analysis. These curves represent the true positive rate against the false positive rate across all possible classification thresholds, providing insight into each classifier's discriminative ability. AUC was calculated for each classifier, which serves as a threshold-independent measure of classification performance. The analysis revealed AUC scores above 0.9 for all malware classes, indicating good ranking ability of the classifiers – as shown in Figure 3.

As further optimization, we adopted dynamic thresholds. In fact, initial evaluation using the classifiers' default threshold (i.e. 0.5) yielded suboptimal accuracy, despite the high AUC values. This discrepancy highlighted the need for dynamic threshold optimization, as the default threshold may not be optimal for imbalanced datasets. To address this discrepancy, Youden's J statistic method [64] was used to determine optimal classification thresholds for each binary classifier. The threshold θ was thus computed for each classifier, where TPR(θ) and FPR(θ) are the true positive and false positive rate at threshold θ respectively:

$$\theta_{\text{optimal}} = \arg\max_{\theta}[\text{TPR}(\theta) - \text{FPR}(\theta)] \tag{3}$$

Feature ID	Description	Category	Static Threshold	Dynamic Threshold
1	Suspicious and dangerous sites visited per day	Content	>0	-
2	Presence of shady traffic	Content	>0	-
3	Undefined / unknown sites visited per day	Content	>0	>12
4	Suspicious sites visited per day	Content	>0	-
5	Peer to peer sites visited per day	Content	>0	>5
6	Malicious sites visited per day	Content	>0	-
7	Illegal gambling sites visited per day	Content	>0	-
8	Illegal sites visited per day	Content	>0	-
9	Entertainment sites visited per day	Content	>0	>9
10	Business sites visited per day	Content	>0	>45
11	Number of adult sites visited per day	Content	>0	-
12	Gambling sites visited per day	Content	>0	-
13	Number of distinct countries	Diversity	-	>1
14	Avg distinct sha1 by path	Diversity	-	>7.76
15	Avg distinct domains	Diversity	>4.5	>51.45
16	Number of distinct domains	Diversity	>18	>145
17	Number of distinct categories site	Diversity	>2	>38
18	Ratio not in tranco diff	Popularity	-	>0.20
19	Product version percentile rank	Popularity	-	>0.75
20	Ratio not in tranco	Popularity	>1%	>15.24%
21	Avg daily distinct product	Volume	-	>119.25
22	Number of product names	Volume	-	>159
23	Number of file signers	Volume	-	>31
24	Night ratio	Volume	-	>0.85
25	Median of file prevalence	Volume	-	>1.93
26	Avg daily night ratio	Volume	>1%	>28.51%
27	Number of refined filepaths	Volume	-	>52
28	Transformed median file prevalence	Volume	-	>0.07

TABLE IV: Features employed by *ODG* together with the thresholds used to discriminate between *exposed* and *not exposed* behaviors.

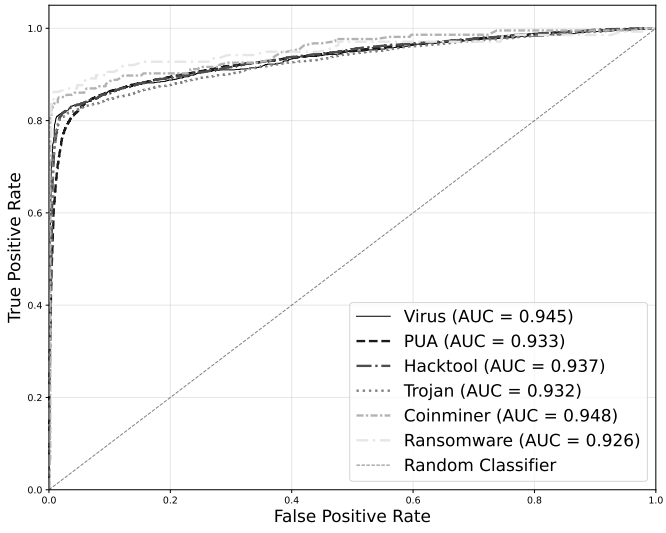


Fig. 3: ROC curves for *MLC*.

As shown in Table VI, the adoption of dynamic thresholds resulted in significant accuracy improvements across all malware classes, with F1, precision, and recall above 0.9. This indicates that greater accuracy is achieved by requiring higher confidence in classifying an endpoint as at risk. That is, a certain degree of conservatism in the judgment of the classifiers in the decision point lends to a higher overall accuracy.

After optimizing the thresholds, we computed the feature importance for each malware class. This analysis helped us understand which user behaviors contribute most significantly to risk prediction. The feature importance for each feature is calculated using *GINI Impurity* [63]:

$$\text{Gini}(D) = 1 - \sum_{i=1}^{c} p_i^2 \tag{4}$$

D represents the training set, p_i denotes the proportion of samples belonging to the malware class i, c is the total number of classes, and the sum is calculated over all classes. To compare the feature importance across all malware classes, we calculated the importance scores by treating all different malware classes as *exposed* and classifying them collectively against the normal class. These measures reveal the relative contribution of each feature to the resulting risk score, providing interpretable insights into the behavioral patterns that help anticipate future malware outbreaks. Feature importance for the generic class "encountered malware" is depicted in Figure 4, while the breakdown for all classes is left for the Appendix (Figure 9).

The results demonstrate a convergence between the MLC and ODG methodologies, despite their differing approaches. Both methods identify system- and network-reputation activities as the primary discriminative features. Notably, the feature *product version percentile rank* stands out as a significant predictor in both models. This is identified as a contributing feature in five out of six malware classes for ODG (as shown in feature number 19 in Table V), and it is ranked among the top 10 features in MLC with an importance score of 0.064, as illustrated in Figure 4.

ID	Feature	Coinminer	Hacktool	PUA	Ransom	Trojan	Virus
1	Suspicious and dangerous sites visited per day	4.74 (**)	2.81 (**)	2.98 (**)	2.73	3.97 (**)	2.80 (**)
2	Presence of shady traffic	5.28 (**)	2.81 (**)	2.98 (**)	3.07 (*)	3.97 (**)	2.83 (**)
3	Undefined / unknown sites visited per day	0.71	1.74 (**)	2.15 (**)	1.71 (**)	1.80 (**)	1.13 (*)
4	Suspicious sites visited per day	0.94	1.37 (**)	1.91 (**)	0.68	1.78 (**)	0.77
5	Peer to peer sites visited per day	0.56 (**)	1.56 (**)	1.81 (**)	0.85	1.53 (**)	0.80 (**)
6	Malicious sites visited per day	2.21 (**)	2.80 (**)	2.90 (**)	1.48	3.01 (**)	2.52 (**)
7	Illegal gambling sites visited per day	0.91	1.47 (**)	2.00 (**)	0.68	1.83 (**)	0.92
8	Illegal sites visited per day	1.00	8.51 (**)	8.06 (**)	2.99	5.83 (**)	4.28 (**)
9	Entertainment sites visited per day	0.77	1.62 (**)	1.95 (**)	1.22	1.76 (**)	1.05
10	Business sites visited per day	0.58 (**)	1.41 (**)	1.82 (**)	1.45 (*)	1.46 (**)	0.75 (**)
11	Number of adult site visited per day	2.31	1.78 (**)	2.37 (**)	1.00	2.29 (**)	1.15
12	Gambling sites visited per day	0.94	1.37 (**)	1.91 (**)	0.68	1.78 (**)	0.77
17	Number of distinct categories site	0.79	1.58 (**)	1.97 (**)	1.47 (*)	1.68 (**)	0.85 (*)
16	Number of distinct domains	0.84	1.50 (**)	1.89 (**)	1.31	1.57 (**)	0.84 (**)
14	Avg distinct sha1 by path	1.13	1.26 (**)	1.46 (**)	1.30	1.22 (**)	1.29 (**)
13	Number of distinct countries	0.87	1.03	0.99	0.94	1.18 (*)	1.49 (**)
15	Avg distinct domains	0.77	1.49 (**)	1.77 (**)	1.10	1.53 (**)	0.86 (*)
18	Ratio not in tranco diff	1.06	1.08 (*)	1.19 (**)	0.77	1.19 (**)	1.08
19	Product version percentile rank	2.84 (**)	1.01	1.08 (**)	1.71 (**)	1.29 (**)	1.76 (**)
20	Ratio not in tranco	1.17	0.99	1.13 (**)	0.75	1.13 (**)	1.11
27	Number of refined filepaths	1.50 (*)	1.94 (**)	1.95 (**)	2.08 (**)	1.88 (**)	1.84 (**)
21	Avg daily distinct product	1.46 (*)	1.63 (**)	1.78 (**)	2.29 (**)	1.46 (**)	1.80 (**)
22	Number of product names	1.21	1.59 (**)	1.85 (**)	2.45 (**)	1.61 (**)	1.64 (**)
23	Number of file signers	1.07	2.02 (**)	2.10 (**)	1.37	1.72 (**)	1.83 (**)
24	Night ratio	1.07	1.13 (**)	1.26 (**)	1.37	1.15 (**)	1.33 (**)
25	Median of file prevalence	0.90	0.75 (**)	0.66 (**)	0.47 (**)	0.85 (**)	0.83 (**)
26	Avg daily night ratio	1.03	1.02	1.27 (**)	1.92 (**)	1.19 (**)	1.16 (*)
28	Transformed median file prevalence	0.95	1.64 (**)	1.70 (**)	1.73 (**)	1.38 (**)	1.71 (**)

TABLE V: Odds-ratios generated by *ODG* during training. Only the probabilities with p-value less than 0.05 are considered statistically significant and used for generating the risk scores Ri. $(*)$ $p < 0.05$; $(**)$ $p < 0.01$

	F1 Score	Precision	Recall	AUC
Coinminer	0.938	0.941	0.940	0.948
Hacktool	0.924	0.927	0.925	0.937
PUA	0.904	0.904	0.904	0.933
Ransomware	0.923	0.926	0.924	0.926
Trojan	0.926	0.928	0.927	0.932
Virus	0.936	0.938	0.937	0.945

TABLE VI: Accuracy for *MLC*: F1 Score, Precision, Recall and AUC for individual classifiers.

The prominence of network behavior indicators, such as the average number of domains visited per day by a user (*average distinct domains*) or their lack of popularity in white lists (*ratio not in Tranco*), underscores the importance of incorporating users' web behavior into malware risk quantification. With these features, we capture the behavior of malware operators who tend to leverage unconventional Internet facilities for their campaigns.

V. Risk Estimation

With *Beaver* well-trained on both system and web activities of millions of users and associated malware infections, we proceed to estimate the risk of *future* malware outbreaks for *new* endpoints. In this phase, we deployed our system in a real-world setting and utilized it to assess the risk of all endpoints collected through our telemetry over a one-month period (January 2025).

In total, we analyzed 10,680,183 endpoints across 217 countries and 822 organizations. A majority of these machines

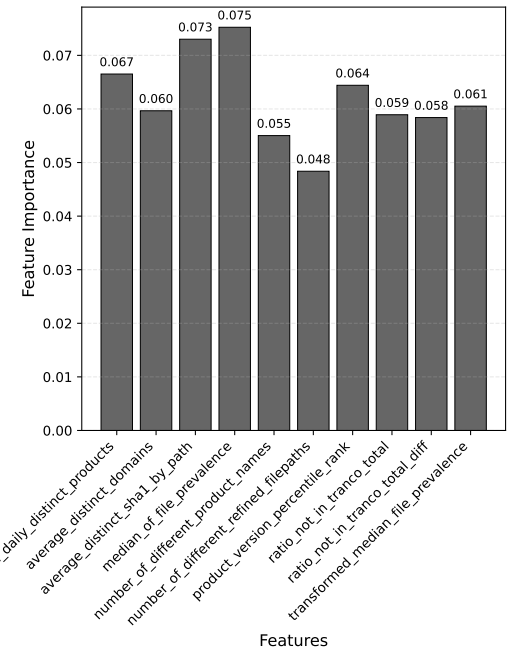


Fig. 4: Feature importance for the generic class 'encountered malware' *(MLC)*.

(73.48%) were situated in very large enterprises (VLEs)[2] consisting of networks used for business purposes, including

[2]VLEs are defined as organizations with an average of 10,000+ deployed endpoints.

Malware Class	Estimated Risk R_i 0-20	21-40	41-60	61-80	81-100
Coinminer	45.91	44.75	8.42	0.88	0.04
Hacktool	23.37	40.55	22.21	13.08	0.79
PUA	27.82	30.85	18.82	21.60	0.91
Ransomware	32.53	39.08	20.26	7.79	0.34
Trojan	21.82	40.84	21.83	14.85	0.66
Virus	38.23	43.85	13.80	3.99	0.13
Overall Risk R	31.61	39.99	17.56	10.37	0.48

TABLE VII: Overall *risk estimation* for the 10.7M endpoints included in our study.

regular desktop computers for day-to-day office tasks, high-end machines for specialized operations, and servers.

As we presented in Section II, the estimation of the risk (in a scale between 0 and 100) consists of a combination of a statistical estimator (i.e., odds-ratios representing the association between an exposure and the outcome) and a multi-label classifier. These two approaches are combined to generate the final risk score $R = \{R_1, R_2, ...R_i\}$ for an endpoint e as an average of the single scores $R1$ and $R2$ (ref. Figure 1).

We left our system processing endpoint data collected in real-time for one month, and then drew cumulative results. The risks associated with various classes of malware are detailed in Table VII and visually represented through a heatmap in Figure 5.

A first, general observation is that 31.61% of the population falls within the 0-20 risk range. This result supports our initial assumption. In fact, the majority of the machines in our study are part of enterprise networks that utilize security solutions, primarily the anti-malware platform from where we collect our telemetry data. Furthermore, it is reasonable to assume that these organizations are mandated by regulatory frameworks such as NIS2 or NIST to enforce corporate security policies, including web filtering and application whitelisting, which inherently reduce the overall risk a priori.

Over 60% of the machines are at risk, with 10.37% and 0.48% respectively falling within the 61-80 and 81-100 risk ranges.

When examining the different malware classes predicted by our system, PUA emerges as the most risky class in the ecosystem. This is attributable to the widespread presence of unwanted programs in all their forms, particularly on dubious websites that offer malicious applications disguised as legitimate programs, such as streaming websites providing applications for watching copyrighted content for free. This observation is further confirmed by Table V, which shows PUA as the leading malware class when it comes to the highest prevalence of significant odds-ratios (i.e., marked with a single or double $*$).

Vice-versa, coinminer is a very "specialized" form of malware, which leverages the computational resources of unaware users for mining cryptocurrencies. Our study revealed that the risk of encountering one of these is the lowest when compared with other classes. However, when examining the features contributing to the risk of future coinminer incidents as reported in

Fig. 5: Heatmap displaying the risk estimation per malware class, along with the associated number of endpoints for each risk range and class.

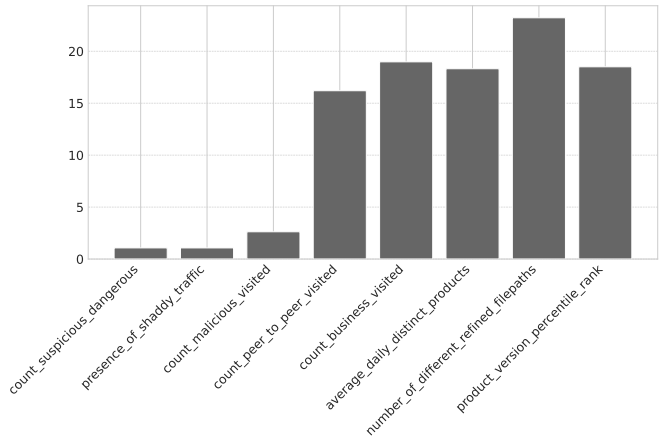


Fig. 6: Contributing features for the malware class 'Coinminer'. Y-axis represents the percentage of the population leveraging each feature for risk estimation.

Figure 6, visiting P2P websites (for example, those hijacked with crypto scams) or business sites (e.g., offering crypto-like applications) significantly put a user at risk. This is also confirmed by all other features that capture the high number of software applications downloaded, installed, and executed by the user on the machine.

Next, we analyzed the endpoints identified as highly exposed to malware outbreaks (i.e., scores of 81-100 in Figure 5) to understand their usage within corporate networks. Our goal was to validate our assumptions, i.e. that the way in which a machine is utilized is a main factor for becoming victim of a specific class of malware.

For each class, we randomly sampled 500 high-risk endpoints and extracted the list of installed applications such as engineering software, ERP platforms, games, etc.. from the system activity logs (i.e., before feature extraction). We

	Coinminer	Hacktool	PUA	Ransomware	Trojan	Virus
Business & Enterprise	12.00	12.00	2.00	48.00	22.00	4.00
Desktop & General	39.56	13.19	5.49	15.93	18.13	7.69
Finance	0.00	0.00	16.67	75.00	8.33	0.00
Gaming	0.00	22.22	66.67	0.00	11.11	0.00

TABLE VIII: Amount of endpoints at very high risk, categorized by endpoint's usage. Values expressed in %.

refined this list by excluding DLL files, drivers, and known OS applications, and removed redundant product names to improve computational performance.

We fed the curated dataset to OpenAI GPT-4o [61] which we used to identify the intended use of each endpoint such as a computer graphics workstation, a database server, or a regular office endpoint. For this to work, we designed a system prompt aimed at enabling the LLM to categorize the endpoint.

The results of our analysis are reported in Table VIII. Ransomware tend to target endpoints holding valuable business information or critical to the operation of an organization (e.g., ERP or financial systems) – this because they represent attractive targets for ransom demands.

Coinminers tend to spread indiscriminately across the largest number of enterprise machines, which normally consist of regular desktop endpoints used for office and generic tasks. In this case, the malware authors are primarily interested in leveraging the largest amount of machines for distributed computation of cryptocurrencies. PUAs are found targeting machines that run gaming software, most likely because their users (gamers) are used to download a large variety of games, including untrusted software.

This shows that the way in which a user utilizes the endpoint is a main factor for categorizing the risk of future malware outbreaks.

Finally, we show how *Beaver* renders in practice. The dashboard (ref. Figure 7) consists of a world map visualizing the aggregated risk scores per country and class of malware. An analyst can interact with the map to expand on different regions, or utilize the search functionality to look up specific endpoints, networks, or organizations. Other general information such as the number of endpoints processed by the system, the associated organizations, and the endpoints with the highest risks are also presented.

Figure 8 presents a breakdown for one specific endpoint: Figure 8a depicts the estimated risk for the 6 malware classes considered in this work, and Figure 8b summarizes the behaviors contributing to the risk. These visualizations help analysts assess the factors that pose the organization at risk and to prevent possible future incidents.

VI. Related Work

Over the past few years, several work has been conducted on identifying factors that contribute to security risks.

Woods et al. [19] presented a systematization of knowledge on quantifying cyber risk. Their work synthesizes existing approaches and methodologies for measuring and modeling cybersecurity risk, aiming to provide a structured understanding of the challenges and advancements in this complex field. The paper discusses various metrics, frameworks, and data sources used to quantify cyber risk, highlighting their strengths, limitations, and applicability in different contexts.

Lévesque et al. conducted two related clinical trials investigating human factors in home cybersecurity. Initially, in 2017 [16], they explored how user characteristics and behaviors contribute to malware risk in home networks. Building on this, their 2018 study [31] involved 50 home users, assessing antivirus performance alongside user traits like technical proficiency, browse habits, and security alert interaction to determine their influence on malware encounter risk, even with AV software present. Both studies highlight the critical interplay between human factors and technological defenses in shaping domestic cybersecurity.

Canali et al. [6] explored the effectiveness of predicting security risks based on users' web browsing behaviors, with data collected by an antivirus vendor. Their research aimed to determine whether it was feasible to identify users who were at a higher risk of falling victim to web-based attacks by analyzing their browsing patterns. The study utilized a large dataset of anonymized user behavior collected by the AV vendor, which included details such as the types of websites visited, frequency of visits, and the user's engagement with potentially risky sites.

Yen et al. [5] investigated security risks within a large enterprise, revealing two critical insights. They found geographical variations in compromise risk, indicating location's influence. Moreover, devices faced an elevated malware risk when operating outside the enterprise's network, underscoring the protective role of organizational security perimeters.

Sanchez-Rola et al. [30] investigated the security implications associated with mouse clicks on web links. The focus of their investigation was to understand how users can be misled about the destinations of links they click on websites. This misleading information can significantly elevate the risks users face, particularly in relation to security threats such as phishing attacks, where users are directed to malicious sites under false pretenses.

Thonnard et al. [27] demonstrated that the risk of encountering targeted attacks is not uniform across all industries. Their findings indicate that certain sectors are inherently more vulnerable to these sophisticated and deliberate breaches due to factors such as the value of their data, their role in critical infrastructure, or their involvement in sensitive technologies. This heightened risk necessitates a tailored approach to cybersecurity, recognizing that a one-size-fits-all defense strategy is insufficient for protecting industries facing specialized threats.

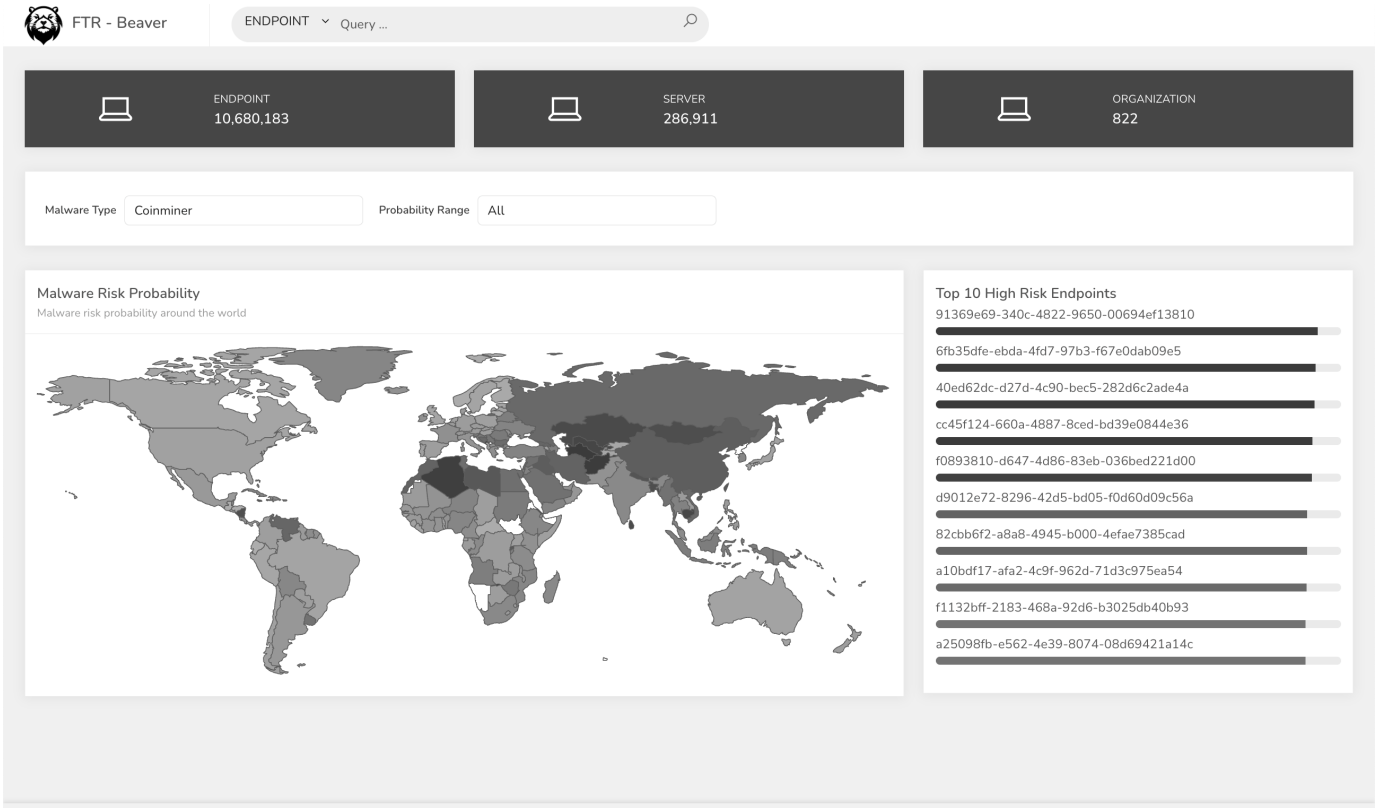


Fig. 7: *Beaver*'s dashboard showing the risk distribution for malware class 'coinminer'.

Allodi et al. [18] explored the various factors leading to vulnerability exploitation. Utilizing a case-control methodology, they found that simply relying on a high CVSS score for patching offers minimal risk reduction. In contrast, the existence of a public exploit was a significantly better indicator of potential exploitation, while the presence of an exploit on cybercrime black markets proved to be the most reliable factor for predicting actual exploitation in the wild. This suggests that monitoring black markets for exploit availability could lead to more effective and economically sensible patching strategies compared to prioritizing solely based on CVSS scores.

Vasek et al. [21] identified specific web server characteristics that increase their likelihood of being compromised for phishing campaigns. Their research pointed to factors like the content management system used, the web server software, and whether that software was outdated. For example, servers running popular CMS platforms like WordPress and Joomla, along with Apache and Nginx, showed higher susceptibility.

In a similar work, Di Tizio et al. [22] conducted a case-control study to identify characteristics of websites that increase their likelihood of being compromised for cryptojacking campaigns. Their preliminary analysis indicated associations between certain website features and cryptojacking.

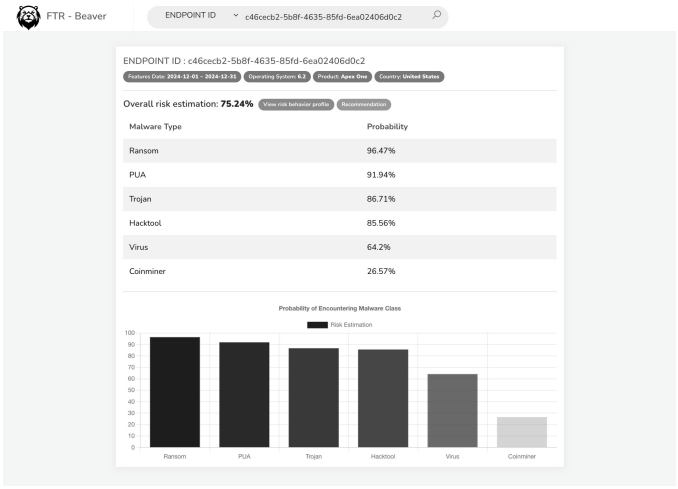
Tajalizadehkhoob et al. [28] investigated how web security features and patching practices influence compromise rates in shared hosting environments. Their analysis identified that webmaster and web application security efforts, including patching, significantly reduce phishing and malware incidents.

Fang et al. [29] developed a comprehensive model to assess and quantify the risk of enterprise data breaches. Their work focuses on understanding the various factors that contribute to breach likelihood and impact, moving beyond qualitative assessments to provide a more data-driven approach. The model considers aspects such as an enterprise's security posture, the value of its assets, the nature of potential threats, and the effectiveness of implemented security controls.

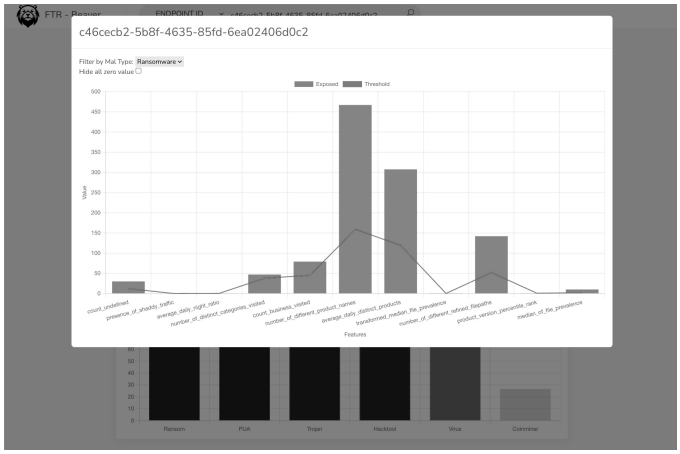
Bilge et al. [3] developed a predictive model to identify enterprise machines at high risk of malware infection. Their approach leverages a specific set of binary-related features, such as characteristics derived from executable files, to determine the likelihood of a machine being compromised. By analyzing these low-level attributes, their model aims to provide early warnings and enable proactive intervention to prevent widespread infections within an enterprise network.

In a similar work, Ovelgönne et al. [10] utilized the WINE dataset to analyze the risk of malware attacks. Their research focused on identifying correlations between malware encounter rates and both user categories (e.g., gamers, software developers) and user system behaviors (e.g., number of binaries, binary prevalence). This allowed them to measure how different user profiles and their digital habits influence their susceptibility to malware.

Research into risk prediction extends to mobile devices, with studies by Dambra et al. [8] and Sharif et al. [9]. Dambra



(a) Estimated scores Ri per malware class i.



(b) Contributing features for malware class 'ransomware'.

Fig. 8: Example of risk estimation for endpoint `c46cecb2-5b8f-4635-85fd-6ea02406d0c2`.

et al. focused on Android applications, analyzing the risk of malware and Potentially Unwanted Applications (PUAs) based on user profiles, considering factors like application volume, diversity, and geographic features. Complementing this, Sharif et al. developed a system that predicts a user's exposure to malicious content solely by analyzing their HTTP traffic, allowing for proactive identification of risky behaviors.

Dambra et al. [7] investigated the varying incidence of malware classes (like PUA, adware, ransomware) and families between consumer and enterprise environments. Their research revealed distinct patterns of malware encounters in these two user groups. Employing a Generalized Linear Model (GLM), they quantified the risk of these encounters based on several key characteristics, including the number of active days, geographical location, the count of installed software vendors, and file-request volume. This approach provides a granular understanding of how different factors contribute to malware risk across diverse user contexts.

The research most closely related to ours is conducted by Meschini et al. [62], who performed a case-control study to understand how user behaviors within organizations influence the risk of malware encounters. This research identified specific network and system behaviors that increase the odds of encountering various malware. Our work builds on this previous study, which significantly extends in different directions including a system that automates the estimation of future risks and malware outbreaks at scale. We leverage a wider set of features, and we refine the estimation by coupling the component responsible for generating the odds-ratios with a multi-label classifier.

Overall, while existing literature has explored the influence of user behaviors on malware risk within organizational contexts, our research advances the field in several ways. We provide a practical, large-scale assessment of the security risks across 10 million+ endpoints used by regular users globally, far exceeding the scope of previous studies.

Our work introduces a novel approach that combines statistical inference and machine learning to estimate the risk of future malware outbreaks. We believe we are the first to conduct such an extensive study that examines the combined effect of both web behaviors and system behaviors on the risks of a wide range of malware classes, including coinminer, hacktool, PUA, ransomware, trojan, and virus. This comprehensive behavioral analysis, coupled with our tool's implementation for practical assessment, sets our work apart from prior research that focuses on specific behavioral aspects, smaller datasets, or treats the different malware classes as a whole.

VII. Conclusions

In this work, we explored the link between user activities and malware outbreaks. We introduced a system that we deployed in a real-world setting to conduct a large-scale study on millions of endpoints and their associated user activities. Our findings confirm that malware authors operate their campaigns in different forms depending on the class of malware they employ. At the same time, the way users utilize their machines becomes a primary factor for incidents, and the behaviors adopted both on the machine and on the Internet can be leveraged to estimate the risk for specific classes of malware. Our system has proven helpful to analysts in predicting risk outbreaks, which we believe represents a significant advancement compared to traditional security solutions that rely on reactive approaches to detect already-occurring attacks rather than anticipating future compromises. We hope our work will foster further research in this field.

References

[1] A. Acar, et al., "An analysis of malware trends in enterprise networks," in *Proc. of ISC-19*, 2019.

[2] P. Kotzias, et al., "Mind your own business: A longitudinal study of threats and vulnerabilities in enterprises," in *Proc. of NDSS-19*, 2019.

[3] L. Bilge, Y. Han, and M. Dell'Amico, "Riskteller: Predicting the risk of cyber incidents," in *Proc. of CCS-17*, 2017.

[4] S. Dambra, L. Bilge, and D. Balzarotti, "Sok: Cyber insurance - technical challenges and a system security roadmap," in *Proc. of SS&P-20*, 2020.

[5] T. Yen, et al., "An epidemiological study of malware encounters in a large enterprise," in *Proc. of CCS-14*, 2014.

[6] D. Canali, L. Bilge, and D. Balzarotti, "On the effectiveness of risk prediction based on users browsing behavior," in *Proc. of AsiaCCS-14*, 2014.

[7] S. Dambra, L. Bilge, and D. Balzarotti, "A comparison of systemic and systematic risks of malware encounters in consumer and enterprise environments," *ACM Trans. Priv. Secur.*, vol. 26, no. 2, 2023.

[8] S. Dambra, et al., "One size does not fit all: Quantifying the risk of malicious app encounters for different android user profiles," in *Proc. of USENIX-23*, 2023.

[9] M. Sharif, et al., "Predicting impending exposure to malicious content from user behavior," in *Proc. of CCS-18*, 2018.

[10] M. Ovelgönne, et al., "Understanding the relationship between human behavior and susceptibility to cyber attacks: A data-driven approach," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 4, 2017.

[11] R. van Wegberg, et al., "Plug and prey? measuring the commoditization of cybercrime via online anonymous markets," in *Proc. of USENIX-18*, 2018.

[12] E. Cozzi, et al., "Understanding linux malware," in *Proc. of SS&P-18*, 2018.

[13] E. Cozzi, et al., "The tangled genealogy of iot malware," in *Proc. of ACSAC-20*, 2020.

[14] E. Avllazagaj, et al., "When malware changed its mind: An empirical study of variable program behaviors in the real world," in *Proc. of USENIX-21*, 2021.

[15] NIST, "Sp 800-30: Guide for conducting risk assessments," 2012, https://csrc.nist.gov/pubs/sp/800/30/r1/final. Accessed: 2023-10-01.

[16] F. L. Lévesque, J. M. Fernandez, and D. Batchelder, "Age and gender as independent risk factors for malware victimisation," in *Proc. of BCS HCI-17*, 2017.

[17] R. Doll and A. B. Hill, "Smoking and carcinoma of the lung," *British medical journal*, vol. 2, no. 4682, 1950.

[18] L. Allodi and F. Massacci, "Comparing vulnerability severity and exploits using case-control studies," *ACM Trans. Inf. Syst. Secur.*, vol. 17, no. 1, 2014.

[19] D. W. Woods and R. Böhme, "Sok: Quantifying cyber risk," in *Proc. of SS&P-21*, 2021.

[20] D. W. Woods and L. Walter, "Reviewing estimates of cybercrime victimisation and cyber risk likelihood," in *Proc. of IEEE EuroS&PW-22*, 2022.

[21] M. Vasek, J. Wadleigh, and T. Moore, "Hacking is not random: A case-control study of webserver-compromise risk," *IEEE Trans. Dependable Secur. Comput.*, vol. 13, no. 2, 2016.

[22] G. D. Tizio and C. N. Ngo, "Are you a favorite target for cryptojacking? A case-control study on the cryptojacking ecosystem," in *Proc. of IEEE EuroS&PW-20*, 2020.

[23] G. Mezzour, K. M. Carley, and L. R. Carley, "An empirical study of global malware encounters," in *Proc. of HotSoS-15*, 2015.

[24] X. Ugarte-Pedrero, M. Graziano, and D. Balzarotti, "A close look at a daily dataset of malware samples," *ACM Trans. Priv. Secur.*, vol. 22, no. 1, 2019.

[25] M. Botacin, et al., "One size does not fit all: A longitudinal analysis of brazilian financial malware," *ACM Trans. Priv. Secur.*, vol. 24, no. 2, pp. 11:1–11:31, 2021.

[26] A. Küchler, et al., "Does every second count? time-based evolution of malware behavior in sandboxes," in *Proc. of NDSS-21*, 2021.

[27] O. Thonnard, et al., "Are you at risk? profiling organizations and individuals subject to targeted attacks," in *Proc. of FC-15*, 2015.

[28] S. Tajalizadehkhoob, et al., "Herding vulnerable cats: A statistical approach to disentangle joint responsibility for web security in shared hosting," in *Proc. of CCS-17*, 2017.

[29] Z. Fang, et al., "A framework for predicting data breach risk: Leveraging dependence to cope with sparsity," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, 2021.

[30] I. Sánchez-Rola, et al., "Dirty clicks: A study of the usability and security implications of click-related behaviors on the web," in *Proc. of WWW-20*, 2020.

[31] F. L. Lévesque, et al., "Technological and human factors of malware attacks: A computer security clinical trial approach," *ACM Trans. Priv. Secur.*, vol. 21, no. 4, 2018.

[32] "Trend micro reports earnings results for q2 2023 marking 99th quarter of profitability," 2023, https://newsroom.trendmicro.com/2023-08-08-Trend-Micro-Reports-Earnings-Results-for-Q2-2023-Marking-99th-Quarter-of-Profitability.

[33] "The Average Cost Of Ransomware Attacks (Updated 2025)", https://purplesec.us/learn/average-cost-of-ransomware-attacks/

[34] "Trend micro blocks over 94 billion threats in 2021," 2021, https://newsroom.trendmicro.com/2022-01-24-Trend-Micro-Blocks-Over-94-Billion-Threats-in-2021.

[35] N. Pearce, "Analysis of matched case-control studies," *BMJ*, vol. 352, 2016.

[36] K. J. Rothman, et al., *Modern epidemiology*, 2008, vol. 3.

[37] S. Bird, I. Segall, and M. Lopatka, "Replication: Why we still can't browse in peace: On the uniqueness and reidentifiability of web browsing histories," in *Proc. of SOUPS-20*, 2020.

[38] P. A. Lachenbruch, "Analysis of data with excess zeros," *Statistical methods in medical research*, vol. 11, no. 4, 2002.

[39] J.-B. Du Prel, et al., "Confidence interval or p-value?: part 4 of a series on evaluation of scientific publications," *Deutsches Ärzteblatt International*, vol. 106, no. 19, 2009.

[40] M. Jakobsson and J. Ratkiewicz, "Designing ethical phishing experiments: a study of (ROT13) ronl query features," in *Proc. of WWW-06*, 2006.

[41] M. Jakobsson, P. Finn, and N. A. Johnson, "Why and how to perform fraud experiments," *IEEE Secur. Priv.*, vol. 6, no. 2, 2008.

[42] T. Yen and M. K. Reiter, "Traffic aggregation for malware detection," in *Proc. of DIMVA-08*, 2008.

[43] "Trend Micro Site Safety Center," 2025, https://global.sitesafety.trendmicro.com/.

[44] V. L. Pochat, T. van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proc. of NDSS-19*, 2019.

[45] F. L. Lévesque, et al., "A clinical study of risk factors related to malware infections," in *Proc. of CCS-13*, 2013.

[46] V. Le Pochat, et al., "A practical approach for taking down avalanche botnets under real-world constraints." in *Proc. of NDSS-20*, 2020.

[47] "Symantec Internet Security Threat Report," Tech. Rep. Volume 24, Feb. 2019.

[48] "Wannacry malware profilemitre att&ck tactics," 2017, https://www.mandiant.com/resources/blog/wannacry-malware-profile. Accessed: 2023-10-01.

[49] B. A. AlAhmadi, L. Axon, and I. Martinovic, "99% false positives: A qualitative study of SOC analysts' perspectives on security alarms," in *Proc. of USENIX-22*, 2022.

[50] A. Nappa, et al., "The attack of the clones: A study of the impact of shared code on vulnerability patching," in *Proc. of SSP-15*, 2015.

[51] G. D. Tizio, M. Armellini, and F. Massacci, "Software updates strategies: A quantitative evaluation against advanced persistent threats," *IEEE Trans. Software Eng.*, vol. 49, no. 3, 2023.

[52] S. Englehardt and A. Narayanan, "Online tracking: A 1-million-site measurement and analysis," in *Proc. of CCS-16*, 2016.

[53] L. Bilge and T. Dumitraş, "Before we knew it: an empirical study of zero-day attacks in the real world," in *Proc. of CCS-12*, 2012, pp. 833–844.

[54] A. Forget, et al., "Security behavior observatory: Infrastructure for long-term monitoring of client machines," Tech. Rep. CMU-CyLab-14-009, 2014.

[55] A. Forget, et al., "Do or do not, there is no try: user engagement may not improve security outcomes," in *Proc. of SOUPS-16)*, 2016, pp. 97–111.

[56] S. Pearman, et al., "Let's go in for a closer look: Observing passwords in their natural habitat," in *Proc. of CCS-17*, 2017, pp. 295–310.

[57] M. Sharif, et al., "Predicting impending exposure to malicious content from user behavior," in *Proc. of CCS-18*, 2018, pp. 1487–1501.

[58] L. Allodi, F. Massacci, and J. Williams, "The work-averse cyberattacker model: theory and evidence from two million attack signatures," *Risk Analysis*, vol. 42, no. 8, pp. 1623–1642, 2022.

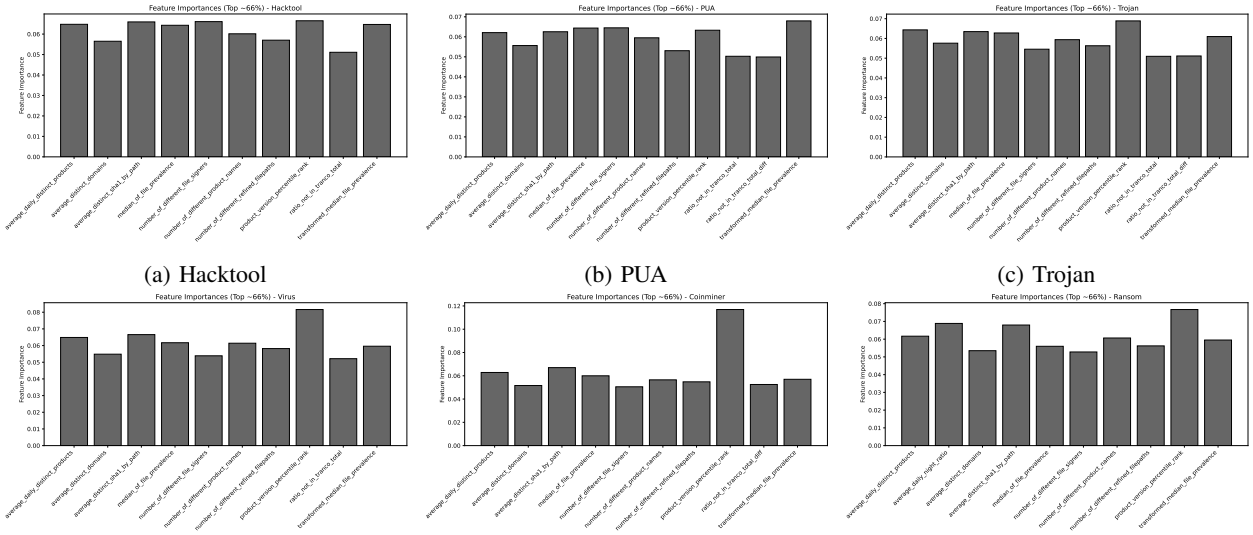
[59] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proc. of SS&P-18*. 2008, pp. 111–125.

[60] "EDR vs. Antivirus – Why Traditional Security Isn't Enough", https://inventivehq.com/edr-vs-antivirus-why-traditional-security-isnt-enough/

[61] "Introducing GPT-4o: OpenAI's new flagship multimodal model now in preview on Azure", https://azure.microsoft.com/en-us/blog/introducing-

gpt-4o-openais-new-flagship-multimodal-model-now-in-preview-on-azure/
[62] M. Meschini, G. Tizio, M. Balduzzi, & F. Massacci, “A Case-Control Study to Measure Behavioral Risks of Malware Encounters in Organizations”, *IEEE Transactions On Information Forensics And Security*, 19 pp. 9419-9432 (2024).
[63] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[64] Youden, W.J. (1950). Index for rating diagnostic tests. *Cancer*, 3(1), 32–35.
[65] Vesselin Bontchev, “Current Status of the CARO Malware Naming Scheme”, https://bontchev.nlcv.bas.bg/papers/pdfs/caroname.pdf

VIII. Appendix



(a) Hacktool (b) PUA (c) Trojan

(d) Virus (e) Coinminer (f) Ransomware

Fig. 9: Feature importance for the different classes of malware.