Measuring Malware Detection Capability for Security Decision Making

Muhammad Yasir Muzayan Haq
University of Twente, NL
m.y.m.haq@utwente.nl

Abhishta Abhishta
University of Twente, NL
s.abhishta@utwente.nl

Sander Zeijlemaker
Cybersecurity at MIT Sloan,
Massachusetts Institute of Technology, US
szeijl@mit.edu

Annette Chau
Cybersecurity at MIT Sloan,
Massachusetts Institute of Technology, US
ac129@wellesley.edu

Michael Siegel
Cybersecurity at MIT Sloan,
Massachusetts Institute of Technology, US
msiegel@mit.edu

L.J.M. Nieuwenhuis
University of Twente, NL
l.j.m.nieuwenhuis@utwente.nl

***Abstract*—Organizations face an urgent need to bolster their cybersecurity defenses against the rising threat of ransomware. Implementing advanced antivirus and anti-malware tools is crucial for proactive identification and mitigation of malicious software. However, adversaries constantly refine malware to evade detection increasing the complexity of the threat. Hence, developing an effective strategy is nontrivial. To address this challenge, this study conducts various analyses on scan results of publicly shared malware samples. Utilizing metadata from 635K samples sourced from MalwareBazaar and scan results from VirusTotal, we assign family labels using AVClass. Additionally, we examine a 90-day longitudinal dataset alongside the main dataset. Our findings demonstrate that while over 60% of scanner engines detect 67% of samples, certain malware families consistently exhibit lower detection rates. Detection capability improves over time, particularly within the initial 30 days, but remains inadequate for specific families. Furthermore, we observe that some scanner engines demonstrate nearly flawless detection capability across all malware families, while the majority struggle with efficiently detecting certain types. Moreover, we performed Monte Carlo simulations and revealed that employing multiple scanner engines substantially enhances detection capability, with 3 to 7 scanners being optimal. Finally, simulation analysis in a case study highlights the significant impact of hard-to-detect malware on risk and performance, underscoring the importance of effective malware strategies.**

***Index Terms*—VirusTotal, malware detection, defense strategy, decision-making, security investment**

1. Introduction

Given the growing threat of ransomware, it becomes imperative for organizations to fortify their cybersecurity frameworks to ward off malware threats. A robust defense mechanism often involves the implementation of advanced antivirus and anti-malware tools, tailored to proactively identify the presence of such malicious software within enterprise systems. Despite the important role that these scanners fulfill in the early detection of malware and preventing ransomware incidents, depending solely on the recommendations of a singular scanner proves to be insufficient [8, 10].

Adversaries continually refine and release updated versions of malware, specifically designed to evade these detection mechanisms. While developers of anti-malware tools strive to release frequent updates for these scanners, not every tool shows the same level of effectiveness in recognizing every variant of malware [7]. Their detection capabilities tend to improve progressively, as updates from malware authors are integrated [2].

A failure in detection and subsequent response enables malware to propagate swiftly throughout an organization. Incidents leading to a successful breach lead to profound repercussions on the financial stability and revenue-generating capacities of the organization [4, 5]. Faced with such complexity, organizational leaders find themselves deliberating over optimal investment allocations across various defensive layers, acceptable levels of risk appetite, and dedicated mitigation efforts to secure specific industries.

Determining a viable risk management strategy against malware demands an evaluation of the precision of existing anti-malware tools and an analysis of their evolving efficacy. This paper leverages a comprehensive large dataset of malware scanner evaluations to explore the aggregate efficiency of these tools in identifying diverse malware families. Furthermore, it assesses the accuracy of malware scanners in evaluating a set of 1,500 recently identified malware samples over 90 days. Using this empirical evidence, we conduct Monte Carlo simulations to measure the effectiveness of employing either singular or multiple scanner strategies. We further present a case of a simulation exercise (built on empirical data) predicated on a recognized digital twinning method [9, 11] to ascertain the effectiveness of cybersecurity risk management strategies in mitigating malware risks. We use the simulation model proposed by Jalali et al. [6] including all the assumptions they made.

The key findings from our analyses are as follows:

- Despite over 60% of scanner engines detecting 67% of samples, certain malware families consistently show lower detection scores, persisting even with frequent occurrences in the dataset.

TABLE 1: Summary of two datasets.

	Dataset	
	Large	Longitudinal
Malware Samples		
# of Unique Samples	635,478	1,500
Earliest Submission	2020-09-14	2023-11-10
Latest Submission	2023-11-17	2023-11-17
Scanning Results		
# of Unique Engine	114	79
# of Scan Results	47,452,360	10,219,889
Average Scans per Sample	74	6,817
Data Frequency		
# of Days Retrieved	1	90
First Retrieval		2023-11-21
Last Retrieval		2024-02-18

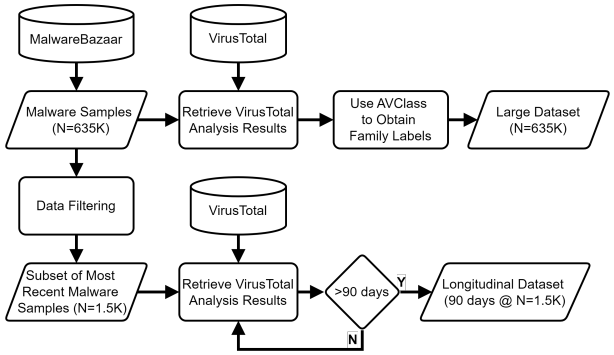


Figure 1: Flowchart of datasets development.

- Detection capability generally improves over time, particularly within the initial 30 days, yet remains inadequate for specific families, leading to sustained low detection rates over an extended duration.
- While some scanner engines exhibit nearly flawless detection across all malware families, the majority still struggle to efficiently detect certain types.
- Utilizing multiple scanner engines significantly enhances detection capability, with the optimal range being 3 to 7 scanners, as employing more than 7 results in diminishing returns. Detection capability in a multi-scanner setup improves gradually over time but becomes insignificant after the initial 30 days.
- In a case study involving 14 security and IT executives, it was observed that most struggled with effective malware strategies initially but showed improvement through simulations. Analysis revealed that hard-to-detect malware significantly impacts risk and performance, even with a decent strategy.

2. Methodology

In this study, we combine multiple data from different data sources, namely, metadata of malware samples, scan results from various anti-malware scanners, and malware family labels. Figure 1 shows the process we use to collect and analyze this data.

Gather metadata of malware samples. We use metadata of malware samples collected from MalwareBazaar[1]. MalwareBazaar is a crowd-sourcing platform that collects malware samples from the information security community and shares them to help the community against malware threats. We gather metadata from 635K unique malware samples submitted to the platform from September 2020 to November 2023. The metadata consists of sample SHA256 hashes and submission dates used in this work.

Collect scanning results from anti-malware scanners. Second, we collect scanning results from VirusTotal[2]. VirusTotal is an online service that analyses files and URLs for malware and security threats by aggregating multiple antivirus engines and detection tools. Users can upload files or URLs, and VirusTotal scans them using various antivirus engines and tools, providing detailed reports including detection results, file behavior, relationships, and associated domains/IP addresses. Reports may vary over time as the platform regularly reanalyses files.

In this work, we retrieve the scan results from VirusTotal using the SHA256 hashes as identifiers of 635K malware samples from our metadataset from MalwareBazaar. If VirusTotal already knows the hash, the platform will return the analysis result of that file consisting of scan results from various contributing scanner engines. In this work, we obtained the scan results of all 635K malware samples from VirusTotal. The result also contains other metadata for each scan result, e.g., engine name, engine last update, detection result, and category. VirusTotal suggests a threat label by aggregating threat labels given by each engine to the sample.

A scan result might yield one of these categories for the sample: `"malicious"`, `"suspicious"`, `"harmless"`, `"undetected"`, `"confirmed-timeout"`, `"failure"`, `"timeout"`, and `"type-unsupported"`. Table 1 summarizes the dataset used in this work.

Populate with malware family labels. We used AVClass to label the samples according to their family signatures. AVClass[3] is a tool used in cybersecurity to classify malware based on labels assigned by antivirus engines [14, 15]. It normalizes and clusters these labels to identify common types of malware, making it easier for analysts to understand and respond to threats. Each sample only gets one family label from the tool, based on the scan results for the samples from various scanner engines obtained from VirusTotal reports. AVClass was able to label all 635K samples in our dataset.

Define metrics and analyse data to compare collective scanner performance across malware families.. To avoid doubts, we provide definitions of the important concepts and metrics that we use in this work as follows:

- A *malware analysis* refers to an analysis of a malware sample against all contributing scanner engines of VirusTotal at a certain time which consists of multiple scan results

1. https://bazaar.abuse.ch
2. https://docs.virustotal.com/docs/how-it-works
3. https://github.com/malicialab/avclass

- A *scan result* is a result of a single scan of a sample by a version of a scanner engine in an analysis.
- *Detected* and *positive result* refer to a scan result with a "malicious" category
- *Detection score* of a sample is the proportion of "malicious" scan results in an analysis
- *Detection capability* of a scanner engine is the proportion of samples that it can detect as malicious.
- *Elapsed time* of a scan result refers to the days between the date the sample was first seen by VirusTotal and the date of the engine's last update
- *Malware family* refers to the family label given to a sample by AVClass based on the threat labels from all scanner engines.

Longitudinal analysis to measure the improvement in scanner performance. To study how time affects the detection capability of the scanners, we also conducted a longitudinal analysis. From the main dataset of 635K samples, we created a subset of 1,500 of the most recently submitted malware samples. We retrieved the analysis results from VirusTotal of all malware samples in the subset, every day, for 90 consecutive days. Table 1 compares the large (main) and longitudinal datasets.

Monte Carlo simulation to compare the performance of single and multi-scanner strategies. We conduct Monte Carlo simulations to investigate how the number of scanner engines chosen affects the detection capability. We carried out two simulations using the large and longitudinal datasets.

To evaluate the detection performance, we initiate the process by randomly choosing a collection of 20 different scanner engines for each evaluation cycle. The initial step involved assessing the detection efficiency of the first scanner engine selected. Following this, we introduced the second engine to the mix and assessed the cumulative detection capacity of both engines together. This assessment utilized the 'OR' criteria, meaning a sample was considered malicious if either of the engines identified it as malicious. We sequentially integrated additional engines from the set of 20, one by one, into this evaluation process, recalculating the collective detection capability at each step. This procedure was repeated until we had incorporated all 20 scanner engines into the analysis, thereby providing a comprehensive view of their combined effectiveness in identifying malicious samples. We run these simulations in two settings; 1) 100 evaluation cycles on the large sample of 635K samples; and 2) 10K evaluation cycles on longitudinal data from 1, 7, 30 and 60 days.

3. Results and Discussion

In this section, we present the results of our analyses and discuss our observations.

3.1. Overall Detection Capabilities of Various Scanners

Figure 2 illustrates the distribution of detection scores of all malware samples in the large dataset. From the plot, we observe that the overall detection score of the samples is relatively high, namely, around 80% on average which means that most malware samples are recognized by 80% of the engines. As seen in Figure 2, around 67% of the samples (∼426K) have a detection score higher than 60%. In other words, around 67% of the samples are recognized by at least 60% scanner engines. Despite that, around 33% of the samples (∼209K) have a relatively lower detection score indicating that these samples are not widely recognized and are only detected by less than 60% of the engines.

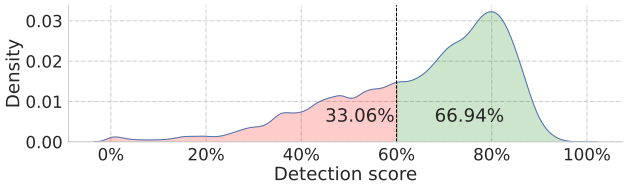


Figure 2: Distribution of detection score, i.e., the proportion of scanners that can detect a malware sample, among 635K malware samples. Area Under Curve (AUC) values of red and green shaded areas indicate the proportion of malware samples with lower and higher detection scores than the threshold of 60%, respectively.

Detection across malware families. We further analyze the large dataset by comparing the detection scores of the samples with different family signatures. As mentioned, we label all the samples with an individual family signature using AVClass. Table 2 presents a brief statistics of two lists of top 10 families with 1) the most frequent samples and, 2) the lowest mean detection score. We highlighted certain families that organizations should be more concerned about, e.g. `mirai`, `qbot`, and `sload`, since they have a high number of samples but with a low mean detection score. In other words, these families are ubiquitous but at the same time hard to detect.

Figure 3 shows how the detection score of the malware samples are distributed across different families. We may observe from the plot that there are some families, such as `emotet`, `agenttesla`, and `dridex`, that are well recognized by most of the scanner engines, i.e., less than 20% samples from these families with a detection score less than 60%. Their high occurrences and high detection score contribute to a high overall detection score as seen in Figure 2. On the other hand, other families shown in Figure 3 are harder to detect by most of the scanner engines, especially malware samples from `sload`, `encdoc`, and `sneaky` families. Malware samples from these families exhibit low detection score (<60%) in majority indicating that only a few scanner engines can detect them.

Key takeaway: *67% samples can be detected by more than 60% scanner engines, however, samples of certain families still have lower detection scores despite being frequently present in the dataset.*

3.2. Time and Detection Capability

From the dataset, we observe that most of the scanner engines in VirusTotal use the blacklist method which

TABLE 2: Summary of family signatures with the most number of samples and lowest detection score. Bold-faced families are those in need of special attention considering their high frequency (>1%) and low detection score (<60%).

Most frequent				Lowest detection			
Family	N	%	MDS	Family	N	%	MDS
emotet	78K	(12.34%)	72%	sneaky	4K	(0.61%)	33%
mirai	**41K**	**(6.40%)**	**59%**	nastya	2K	(0.35%)	37%
agenttesla	39K	(6.11%)	73%	**sload**	**21K**	**(3.37%)**	**39%**
qbot	**39K**	**(6.07%)**	**51%**	**encdoc**	**9K**	**(1.46%)**	**45%**
dridex	34K	(5.42%)	77%	sagent	3K	(0.51%)	47%
sload	**21K**	**(3.37%)**	**39%**	agentxlm	3K	(0.41%)	51%
reline	18K	(2.80%)	77%	**qbot**	**39K**	**(6.07%)**	**51%**
noon	14K	(2.17%)	73%	valyria	2K	(0.32%)	53%
gafgyt	10K	(1.64%)	61%	minerva	2K	(0.24%)	59%
taskun	9K	(1.49%)	74%	icedid	3K	(0.41%)	59%

MDS:Mean Detection Score

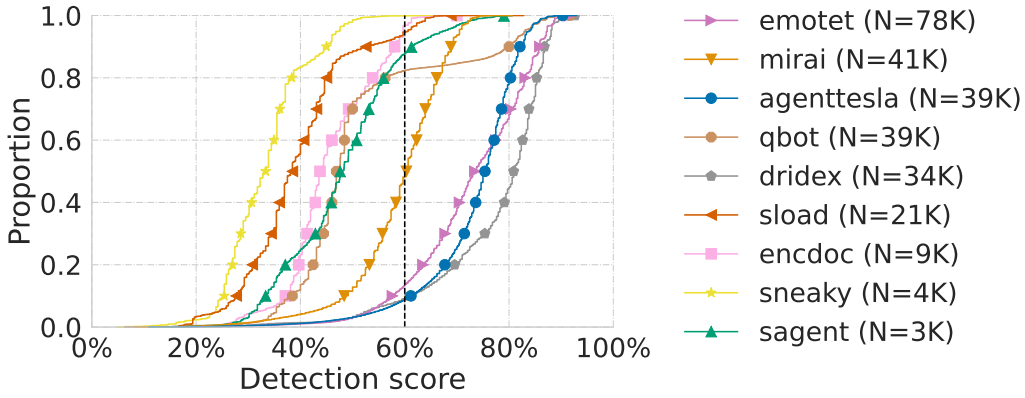


Figure 3: Cumulative distribution of detection score of 635K malware samples from selected malware families. The dashed vertical line indicates the detection score threshold of 60%, i.e., at least 60% of the engines recognize the sample as malicious.

means that they keep a list of signatures known as malicious. With this approach, the detection capability of an engine is improved as new hashes are added to the list. Hence, the timeliness of the list-updating process is crucial to maintain the engine's detection capability. Here, we discuss how time affects the detection capability of the scanner engines in the two datasets.

Detection and analysis age. Using the large dataset, we calculated the elapsed time between the **latest** analysis date and the day VirusTotal first received the sample. In other words, it reflects the age of the analysis result. In Figure 4, we compared the elapsed time of an analysis of a sample with its detection score, i.e., how many engines detect it as malicious. We excluded the analyses aged more than 1,000 days to maintain interpretability.

We can observe from Figure 4 that most analysis results in the large dataset are relatively 'young' compared to the entire dataset, specifically, 60% of the analysis results are younger than 71 days old. The plot also shows that these 'young' analyses (<71 days old) also have a relatively good detection score, namely, >60% of them have at least 60% detection score. Meanwhile, despite not being very significant, we can observe that the age of analysis positively affects the detection score as reflected by an ascending regression line. This is also reflected by less concentrated data points in the lower area for older analysis results.

Analysis age in different families. We introduced the family signatures to the analysis to observe how time affects the detection capability against certain malware families. Figure 6 illustrates the results of this analysis on selected malware families with low detection scores. As we can see from the regression lines in the plot, the detection capability against samples with `mirai`, `qbot`, and `sload` signatures improve over time. However, the improvement in `sload` family is still insufficient to make it significantly more recognizable by all scanner engines. Meanwhile, samples with a `valyria` signature show no visible improvement in the detection score even after a longer period.

Longitudinal analysis of detection capability. It remains inadequate to fully understand how time improves the detection capability by analyzing only one analysis per sample. Therefore, we isolated the detection capability of the engines against a fixed set of samples over time to observe the unbiased effect of time in building the capability. Using the longitudinal dataset, we compared the detection score of the samples from analyses performed in 90 consecutive days as seen in Figure 5. The average

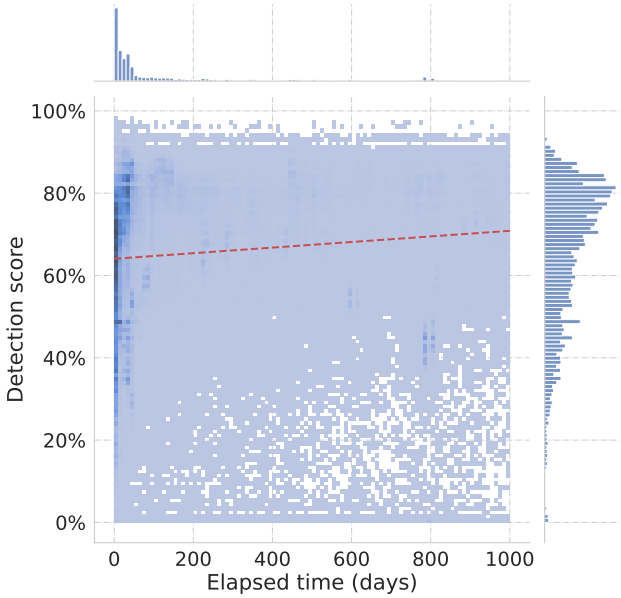


Figure 4: Detection score and elapsed time of 635K malware samples. The red dashed line indicates the linear regression. Horizontal and vertical histograms illustrate the distribution of elapsed time and detection score of analysis results in the large dataset, respectively.

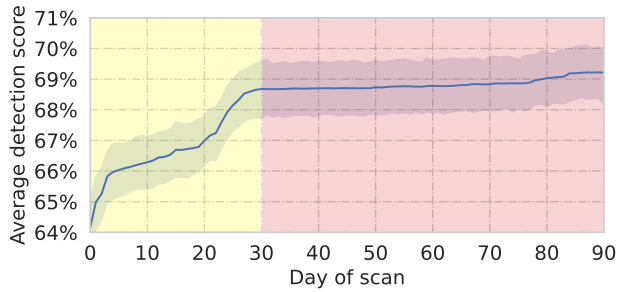


Figure 5: Average detection scores of 1,500 malware samples scanned within 90 consecutive days with 95% confidence interval. The yellow and red shades indicate significant and non-significant periods of improvement in detection capability, respectively.

detection score of 1,500 malware samples starts at around 64% on the first day. It improves significantly to almost 69% within the first 30 days. This clearly shows how time positively affects the detection capability against the samples. However, the average detection score remains almost constant after 30 days with no significant improvement. It might reflect that the blacklist updating process of the scanner engines mainly happens within the first month of the sample's first occurrence.

Key takeaway: *Detection capability improves over time. The most significant increase in detection capability is observed in the first 30 days.*

3.3. Coverage of Individual Scanner Engines

When using a scanner engine, there is a concern about the capability of that individual engine to detect as many malicious files as possible, especially with the large and growing number of new malicious files. Hence, following our previous discussions about the samples, we focus on the detection capability of individual scanner engines. Our objective is to understand the coverage of an engine against different malware families and compare it with other scanner engines. We respect the terms of using VirusTotal data by not disclosing the engine identities anywhere in this work. We calculated the detection capability of each engine against each malware family, i.e., the percentage of malware samples from the family that the engine can detect as malicious. Figure 7 illustrates the results from all scanner engines against samples from the top 20 most frequent families. Each vertical column represents a scanner engine while each horizontal row denotes a malware family.

We can observe from Figure 7 that the detection capability of the scanner engines differs quite significantly. There are at least three types of scanner engines regarding their detection capability. Uniformly dark columns in the left area of the plot suggest the least effective scanner engines with poor detection capability in all 20 families. On the contrary, entirely light-colored columns in the right area illustrate the most effective engines with very high detection capability against samples from all 20 families implying that there exists 'one-fits-all' solutions to tackle malware threats. However, most scanners still possess weaknesses against different malware families, indicated by columns with both dark and light colors.

With regards to the families, some families are only well detected by a small group of scanner engines, such as `dridex`, `reine`, and `encdoc`, denoted by horizontal rows with predominant darker colors.

Key takeaway: *A few scanner engines show almost perfect detection capability (9/87 scanners show above 95% average detection capability) for the top 20 malware families found in our dataset, but most scanner engines (median detection capability of 72%) remain inefficient even for these families.*

3.4. Multi-scanner Strategy

We have demonstrated how most individual scanner engines might possess weaknesses against certain families. Hence, using multiple scanner engines can be a good strategy to increase malware detection capability in an organization. To explore this option, we ran Monte Carlo simulations to portray scenarios in which multiple scanner engines are used at the same time. We defined detection capability in a multi-scanner setup as the proportion of malware samples detected as malicious by at least one scanner engine. Figure 8 presents the results of our simulations with up to twenty scanner engines using scan results from the large and longitudinal datasets. This simulation aims to find the optimum number of scanners to balance the benefit, i.e., detection capability, and the cost of using more scanners, i.e., licensing fee.

In the simulations with the large dataset, we ran 100 individual simulations involving up to twenty randomly selected scanners sampled without replacement. Then, we calculated the detection capability using the scan results of all 635K samples. Figure 8a summarizes the average

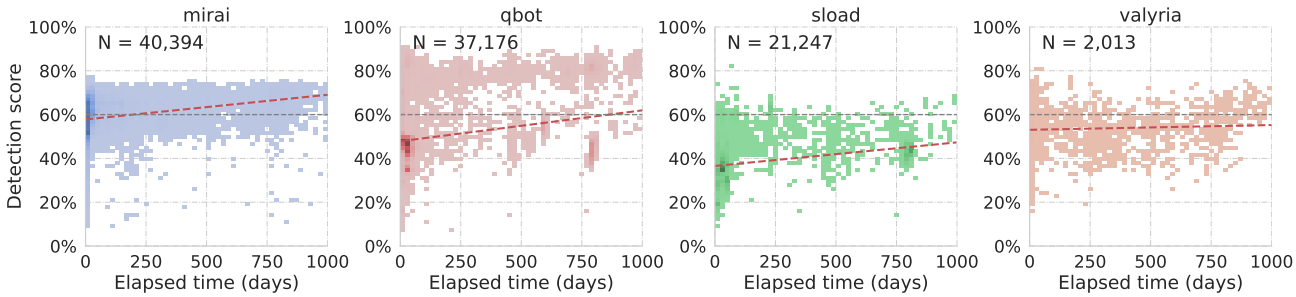


Figure 6: Distribution of malware samples from selected families across detection score and elapsed time. Red and black dashed lines indicate the regressions and the 60% detection threshold.

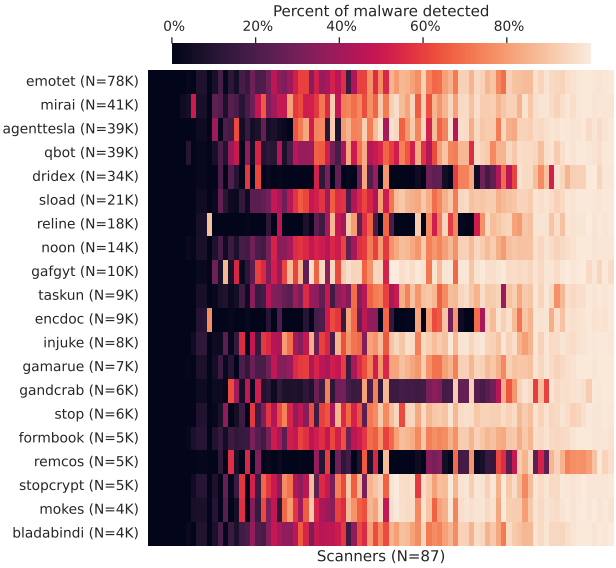
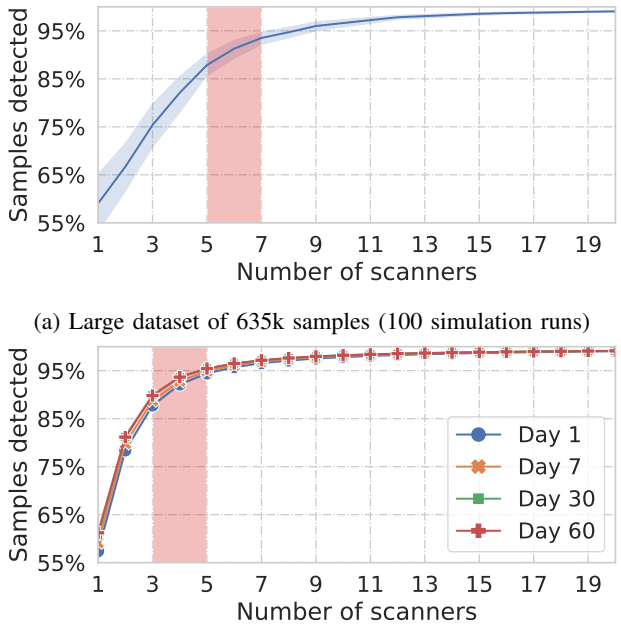


Figure 7: Detection capability of various scanner engines against top 20 malware families by number of samples.



(a) Large dataset of 635k samples (100 simulation runs)

(b) Longitudinal dataset of 1,500 samples (10K simulation runs)

Figure 8: Percentage of malware detected with different numbers of scanners from Monte Carlo simulation on the large dataset (a) and the longitudinal dataset (b). The line plots show the mean values with 95% confidence interval. Red-shaded areas indicate the optimum number of scanners.

detection capability with 95% confidence interval (line shade). We can observe from the plot that the capability increases significantly at a decreasing rate as more engines are used. Looking at the elbow of the curve, i.e., the turning point of the curve, we can identify that the optimum number of scanners lies between 5 and 7. Using more than 7 scanners would be suboptimum due to diminishing returns, i.e., the improvement in detection capability starts to diminish and no longer justifies the cost of using additional scanner engines.

In the simulations with the longitudinal dataset, we compared the scan results from 1^{st}, 7^{th}, 30^{th}, and 60^{th} days to explore the effect of time on the detection capability in a multi-scanner setup. We used the same approach but ran more simulations, namely, 10,000 runs per day, since the data is only 1,500 samples which is much smaller. Figure 8b summarizes the average detection capability in the four selected days. We can spot a similar observation with a slight difference regarding the optimum number of scanners which lies between 3 to 5. The detection capability also improves over time despite not significantly: the results slightly improve from day 1^{st} to day 30^{th}, but the results of the day 30^{th} and 60^{th} are almost identical.

Key takeaway: *Using multiple scanner engines can significantly increase the detection capability with the optimum number of scanners between 3 and 7 while using more than 7 scanners is not recommended due to diminishing returns. Detection capability in multi-scanner setup improves negligibly over time.*

4. Case Study: Integrating Empirical Measures in a System Dynamics Model

Strategic decisions made in an organization to combat cyber threats should not only safeguard its information security but should also fit with its business objectives. These decisions often involve investments of large funding with the expectation of a profitable return. Therefore, we further employed the results from our empirical data

analyses to assist in estimating the impact of strategic security investments on business operations represented by two main aspects: risk and performance. To imitate a more realistic scenario of malware threats, we fed the results from our empirical analyses, including average detection scores of malware and detection capability and timeliness of scanner engines, to the simulation model.

We utilized simulation techniques to create a digital replica or twin of a company and incorporated a gaming functionality in it. We adopted the simulation model including all the assumptions from Jalali et al. [6]. Unlike the real world, wherein bad decisions can cause businesses to fail, simulation environments allow decision-makers to explore and evaluate how their malware strategies could evolve in real life before implementing them [1, 6]. In this research, we used simulation techniques to evaluate the effectiveness of cyber risk management strategies in combating malware.

Replicating a Company in the Simulation. We used a system dynamic approach to build this simulation environment, leveraging the work of Jalali et al. [6]. However, we also considered the epidemic properties of malware through interconnected systems and related the counteractions of anomaly detection and network segmentation [18], as well as ransom payment dynamics [3].

This simulation-aided approach translated system science, control theory, and simulation modeling into a learning experience that captures decision-making behavior in an environment that mimics strategic investment decisions in cyber risk management to drive business performance [6, 17, 19].

In this simulation environment, decision-makers can decide to invest in prevention, detection, response, and recovery on a yearly basis for a period of five years, while maximizing the financial performance (measured in accumulated profits) and minimizing threat exposure (measured in percentage of compromised systems). Each year, 0–5% of the total IT costs can be invested in security capabilities.

The core structure of this simulation is based on the fact that systems not at risk could become systems at risk over time, systems at risk can be compromised by adversaries, and when adversarial threats are properly mitigated, compromised systems can become systems not at risk again [6, 16, 19]. The amount of investment made in prevention, detection, response, and recovery will drive this sequence. Compromised systems as well as investment in cyber security can affect the profits being generated over time.

Strategy Design by Experts. Fourteen security & IT executives played 77 simulations on three different scenarios where they faced regular cyber threats, malware threats, and malware threats under the condition of paying a ransom. These simulations were run based on the assumption of average detection capabilities (see §3.1). Figure 9 shows the results of these simulations.

More than 50% of the participants were unable to define an effective risk management strategy to combat the malware. The others had to complete three full simulation games on average to be able to design an effective cyber risk management strategy. This experiment allowed us to define a base case to come up with an effective strategy to combat malware and malware.

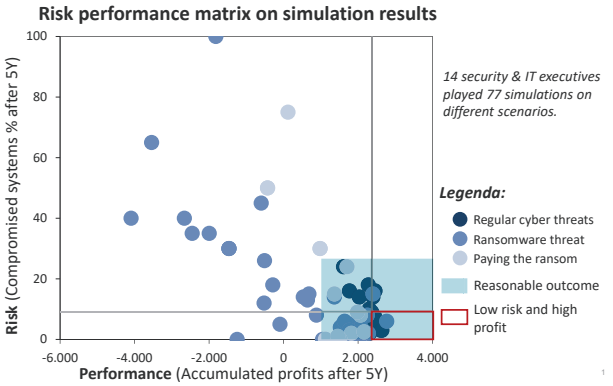


Figure 9: Risk performance matrix of using cyber risk management strategies to combat malware

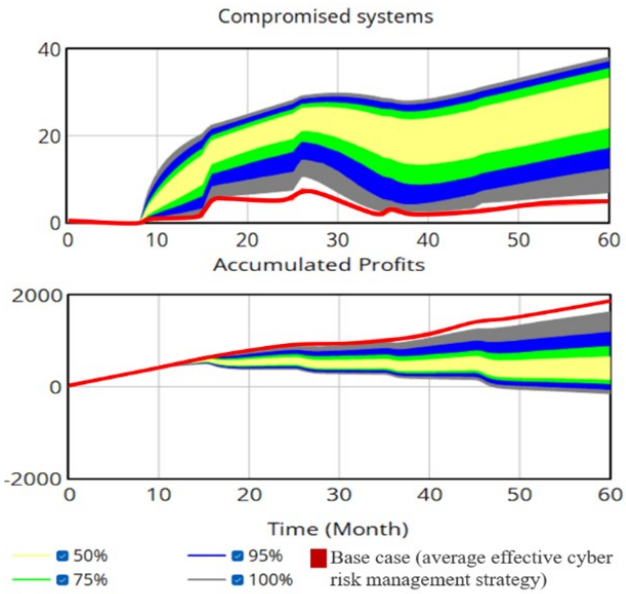


Figure 10: Sensitivity analysis for risk and performance using the simulation model from Jalali et al. [6]. The horizontal axis indicates months after the investment. The top plot illustrates the number of compromised systems over time. The bottom plot shows the accumulated profit from the amount of money that organizations lose or do not lose to pay for ransoms.

Malware with Low Detection Rates. To explore the effectiveness of security investments against malware with low detection rates, we performed a sensitivity analysis. We drew a uniform distribution on a Monte Carlo simulation setup, consisting of randomly launched attacks, which included 2,000 random attack patterns launched with different attack strengths and timing, under the assumption of applying the base case cyber security strategy while varying the detection strength (1% to 60%) and detection/response time (27 days to 1,000 days), reflecting attacks of hard-to-detect malware. Over the full game period, the total strength of all the attacks combined remained the same.

The sensitivity analysis, shown in Figure 10, uses different colors (yellow, green, blue, and gray) to show the distribution of possible outcomes. The results show that hard-to-detect malware can bypass a reasonable cyber risk management strategy and significantly impact risk and performance.

Key takeaway: *After an experiment involving 14 security and IT executives using simulation gaming enhanced with empirical data, we revealed that most decision-makers struggled to define effective strategies against malware threats but gained a better understanding through a series of simulations. We learned from a sensitivity analysis that the threat of hard-to-detect malware might significantly impact risk and performance despite a decent strategy.*

5. Related Works

Several works have inspired us to do this work. Zeijlemaker et al. [19] used simulation gaming to help managers understand the impact of strategic decisions related to security investments on their organizations. They concluded that it is essential to provide managers with supporting tools to help them make strategic security decisions. Jalali et al. [6] applied a similar approach, i.e., using simulation gaming, to investigate the effectiveness of managers in making cybersecurity-related decisions when confronted with two complexities: potential delays in detection capability advancement and uncertain cyber incidents. Between these two factors, their study revealed that uncertainty in cyber incidents was more problematic and caused decision errors for both experienced and inexperienced participants. Our work aims to shed light on this uncertainty problem by quantifying malware threats using empirical data.

We took inspiration regarding the use of empirical data in investigating malware threats from other works with similar approaches. Zhu et al. [20] explored the performance and stability of scanner engines in detecting malicious samples using VirusTotal data. With daily scan results of 14K malware samples, they showed that even 'trusted' individual engines can still provide poor and inconsistent detection against the samples which supports our results in §3.3. Leka et al. [7] conducted a comparison study between scan results from VirusTotal and desktop-based antivirus scanners. Despite the limited number of samples, their work discovered that the detection capability of desktop scanners is relatively better than scanner engines used in VirusTotal. Peng et al. [12] performed several analyses using VirusTotal data to evaluate its performance and consistency in detecting malicious URLs. They set up their own phishing sites for VirusTotal to analyze and found out that due to inconsistent scan results, we need a better way to combine the detection labels from the platform. Sakib et al. [13] proposed different methods to maximize scan results from a multi-scanner setup. Using both malicious and benign samples, they measured the performance of each method using simulations. They also proposed and validated methods to find the optimum configuration of scanner engines.

6. Conclusions

We retrieved the scan results of 635K malware samples from the multi-engine scanning platform VirusTotal and used AVClass to assign family labels based on the multiple tags provided by the scanners. In addition, we collected a 90-day longitudinal dataset with analysis results from the latest 1,500 samples. These two datasets allow us to explore how detection capability changes over time and across different malware families. We compared the coverage and performance of individual scanners and used simulations to determine the optimal number of scanners for the best multi-scanner strategy. Following the simulation model from Jalali et al. [6], we integrated our empirical findings to assess how security investments affect organizational risk and profit.

The results of our analyses include the following points. *First*, although over 60% of scanner engines can detect 67% of the samples, certain malware families consistently exhibit lower detection scores, despite their frequent presence in the dataset. *Second*, detection capability shows improvement over time, although it is notably significant only within the initial 30 days. Nevertheless, for specific families, this improvement is inadequate, resulting in sustained low detection capability even over an extended duration. *Third*, while some scanner engines demonstrate nearly flawless detection capability across all malware families, the majority of scanner engines still struggle with efficiently detecting certain types of malware families. *Fourth*, employing multiple scanner engines can substantially enhance detection capability, with the optimal number of scanners ranging from 3 to 7. However, utilizing more than 7 scanners is discouraged due to diminishing returns. In a multi-scanner setup, detection capability improves gradually over time but becomes insignificant after the initial 30 days. *Fifth,* in a case study with 14 security and IT executives, we found that most struggled with effective malware strategies but improved through simulations. Our analysis showed that hard-to-detect malware significantly impacts risk and performance, even with a decent strategy.

6.1. Limitations

There are some limitations in our work. *First,* considering the nature of crowd-sourcing platforms, we cannot guarantee that the malware sample data that we use in this work is complete, i.e., has included all malware samples out there, neither do we know how representative the samples are. They were submitted voluntarily by people from the security community, hence, many malware samples might be absent from the data set. *Second,* despite their instructions only to submit malicious samples to the platform, some samples in the data might be benign. As a community-based and non-profit project, the platform has been receiving support from several security companies[4] to improve the data quality. *Third,* to respect the data use agreement, we cannot disclose the names of the scanner engines in this work. Moreover, we do not encourage readers to use a scanner engine from a specific provider.

4. https://bazaar.abuse.ch/about/

Instead, we want to emphasize the effectiveness of adopting a multi-scanner strategy as it provides significantly better detection capability than relying on a single engine according to our simulation results.

6.2. Future Works

We observed that simulation gaming could improve decision-makers' understanding of security investment against malware since it allows them to evaluate the effectiveness of their strategy before implementing them. Hence, we believe that improving the simulation model would also improve the strategic decisions quality. Investigating more empirical data allows us to model more probability distributions into the simulation, gradually eliminating assumptions, creating a more realistic scenario, and producing less biased conclusions.] Hence, we plan to further develop our simulation gaming model in the future. With the results of our empirical analyses from this work, we could model more uncertainties as probability distributions in the simulation model For example, we can model the probability of encountering certain malware families in each attack as well as their detection scores according to the statistics, We can further model the detection results based on the number of scanners used in the simulation setup.

We also plan to introduce the cost aspect into our analysis. For example, by collecting more data about the scanner engines such as licensing fees, we can extend our study on the effectiveness of security investments. Later, we can incorporate the result into the simulation model as well to provide a more accurate estimate of the financial impact of the strategies.

Acknowledgements

This work is part of NWO:MASCOT (CS.014) project funded by the Netherlands Organization for Scientific Research

References

[1] Stefano Armenia, Eduardo Ferreira Franco, Fabio Nonino, Emanuele Spagnoli, and Carlo M. Medaglia. Towards the Definition of a Dynamic and Systemic Assessment for Cybersecurity Risks. *Systems Research and Behavioral Science*, 36(4):404–423, 2019. ISSN 1099-1743. doi: 10.1002/sres.2556. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sres.2556.

[2] Ömer Aslan Aslan and Refik Samet. A Comprehensive Review on Malware Detection Approaches. *IEEE Access*, 8:6249–6271, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2963724. URL https://ieeexplore.ieee.org/abstract/document/8949524. Conference Name: IEEE Access.

[3] Renee Dudley. The extortion economy: How insurance companies are fueling a rise in ransomware attacks. *Pro Publica*, 2019.

[4] Andy Greenberg. Ransomware Payments Hit a Record $1.1 Billion in 2023. *Wired*, February 2024. ISSN 1059-1028. URL https://www.wired.com/story/ransomware-payments-2023-breaks-record/. Section: tags.

[5] NCC Group Security Services, Inc. NCC Group Annual Threat Monitor Report 2023, February 2024. URL https://www.nccgroup.com/us/newsroom/annual-ransomware-attacks-increased-by-84-in-2023.

[6] Mohammad S. Jalali, Michael Siegel, and Stuart Madnick. Decision-making and biases in cybersecurity capability development: Evidence from a simulation game experiment. *The Journal of Strategic Information Systems*, 28(1):66–82, March 2019. ISSN 0963-8687. doi: 10.1016/j.jsis.2018.09.003. URL https://www.sciencedirect.com/science/article/pii/S0963868717304353.

[7] Christian Leka, Christoforos Ntantogian, Stylianos Karagiannis, Emmanouil Magkos, and Vassilios S. Verykios. A Comparative Analysis of VirusTotal and Desktop Antivirus Detection Capabilities. In *2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–6, July 2022. doi: 10.1109/IISA56318.2022.9904382. URL https://ieeexplore.ieee.org/abstract/document/9904382.

[8] Eitan Menahem, Asaf Shabtai, Lior Rokach, and Yuval Elovici. Improving malware detection by applying multi-inducer ensemble. *Computational Statistics & Data Analysis*, 53(4):1483–1494, February 2009. ISSN 0167-9473. doi: 10.1016/j.csda.2008.10.015. URL https://www.sciencedirect.com/science/article/pii/S0167947308004763.

[9] Stefan Mihai, Mahnoor Yaqoob, Dang V. Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu, Balbir Barn, Dattaprasad Shetve, Raja V. Prasad, Hrishikesh Venkataraman, Ramona Trestian, and Huan X. Nguyen. Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. *IEEE Communications Surveys & Tutorials*, 24(4):2255–2291, 2022. ISSN 1553-877X. doi: 10.1109/COMST.2022.3208773. URL https://ieeexplore.ieee.org/abstract/document/9899718.

[10] Jose Morales, Shouhuai Xu, and Ravi Sandhu. Analyzing malware detection efficiency with multiple anti-malware programs. *ASE Science Journal*, 1(2):56–66, 2012.

[11] Georgios Mylonas, Athanasios Kalogeras, Georgios Kalogeras, Christos Anagnostopoulos, Christos Alexakos, and Luis Muñoz. Digital Twins From Smart Manufacturing to Smart Cities: A Survey. *IEEE Access*, 9:143222–143249, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3120843. URL https://ieeexplore.ieee.org/abstract/document/9576739. Conference Name: IEEE Access.

[12] Peng Peng, Limin Yang, Linhai Song, and Gang Wang. Opening the Blackbox of VirusTotal: Analyzing Online Phishing Scan Engines. In *Proceedings of the Internet Measurement Conference*, IMC '19, pages 478–485, New York, NY, USA, October 2019. Association for Computing Machinery. ISBN 978-1-4503-6948-0. doi: 10.1145/3355369.3355585. URL https://dl.acm.org/doi/10.1145/3355369.3355585.

[13] Muhammad N. Sakib, Chin-Tser Huang, and Ying-Dar Lin. Maximizing accuracy in multi-scanner malware detection systems. *Computer Networks*, 169:107027, March 2020. ISSN 1389-1286. doi: 10.1016/j.comnet.2019.107027. URL https://www.sciencedirect.com/science/article/pii/S1389128619302518.

[14] Marcos Sebastián, Richard Rivera, Platon Kotzias, and Juan Caballero. AVclass: A Tool for Massive Malware Labeling. In Fabian Monrose, Marc Dacier, Gregory Blanc, and Joaquin Garcia-Alfaro, editors, *Research in Attacks, Intrusions, and Defenses*, pages 230–253, Cham, 2016. Springer International Publishing. ISBN 978-3-319-45719-2. doi: 10.1007/978-3-319-45719-2_11.

[15] Silvia Sebastián and Juan Caballero. AVclass2: Massive Malware Tag Extraction from AV Labels. In *Proceedings of the 36th Annual Computer Security Applications Conference*, ACSAC '20, pages 42–53, New York, NY, USA, December 2020. Association for Computing Machinery. ISBN 978-1-4503-8858-0. doi: 10.1145/3427228.3427261. URL https://doi.org/10.1145/3427228.3427261.

[16] Daniel A. Sepúlveda Estay. A system dynamics, epidemiological approach for high-level cyber-resilience to zero-day vulnerabilities. *Journal of Simulation*, 17(1):1–16, January 2023. ISSN 1747-7778. doi: 10.1080/17477778.2021.1890533. URL https://doi.org/10.1080/17477778.2021.1890533.

[17] J. Sterman. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. McGraw-Hill Higher Education. Irwin/McGraw-Hill, 2000. ISBN 978-0-07-231135-8.

[18] Sander Zeijlemaker and Michael Siegel. *Capturing the Dynamic Nature of Cyber Risk: Evidence from an Explorative Case Study*. January 2023. ISBN 978-0-9981331-6-4. URL https://hdl.handle.net/10125/103372.

[19] Sander Zeijlemaker, Etiënne A. J. A. Rouwette, Giovanni Cunico, Stefano Armenia, and Michael von Kutzschenbach. Decision-Makers' Understanding of Cyber-Security's Systemic and Dynamic Complexity: Insights from a Board Game for Bank Managers. *Systems*, 10(2):49, April 2022. ISSN 2079-8954. doi: 10.3390/systems10020049. URL https://www.mdpi.com/2079-8954/10/2/49. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[20] Shuofei Zhu, Jianjun Shi, Limin Yang, Boqin Qin, Ziyi Zhang, Linhai Song, and Gang Wang. Measuring and Modeling the Label Dynamics of Online {Anti-Malware} Engines. pages 2361–2378, 2020. ISBN 978-1-939133-17-5. URL https://www.usenix.org/conference/usenixsecurity20/presentation/zhu.