Machine Learning-Based Malware Detection in a Production Setting

Sven Krasser, Joel Spurlock, Marian Radu, Brad Moon, Arnd Korn, Madhavi Seth, and Christoph Bausewein

CrowdStrike

Abstract

Machine Learning-based approaches for detection of malware files or execution of malware have been getting increased attention in both academia and in the security industry. With an ever-increasing flood of new threats, the promise of these approaches is to establish a more proactive posture as compared to other methods such as signatures and heuristics. However, operating Machine Learning systems in a production environment is not a trivial task and often overlooked in academic works. In this work, we are giving an overview of the additional requirements, constraints, and complications stemming from running a Machine Learning model as part of such a larger system in an industry setting. This includes model-specific requirements such as target false positive rates, corpus size, and corpus diversity, and system-specific needs such as false positive/negative mitigation, robustness and cost of the model generation process, establishment of feedback mechanisms, deployment considerations, and implementation constraints. Lastly, we touch on common compliance and contractual considerations.

This is a preprint of the following chapter: Sven Krasser, Joel Spurlock, Marian Radu, Brad Moon, Arnd Korn, Madhavi Seth, and Christoph Bausewin, Machine Learning-Based Malware Detection in a Production Setting, published in Malware: Handbook of Prevention and Detection, edited by Dimitris Gritzalis, Kim-Kwang Raymond Choo, and Constantinos Patsakis, 2024, Springer, reproduced with permission of Springer Nature Switzerland AG. The final authenticated version is available online at https://doi.org/10.1007/978-3-031-66245-4_5.

1 Introduction

In the early days of computing, the primary intent of malware was to achieve infamy or to vex computer users. For example, Elk Cloner—initially released in 1982 and widely considered to be the first computer virus to spread in the wild—infected Apple II home computers with a harmless (but annoying) boot sector virus that would display a poem on the computer's screen every 50th time someone booted from an infected disk [22].

These relatively innocent times didn't last for long as businesses adopted PCs en masse and computers were networked. Smelling opportunity, criminal elements were soon releasing viruses and other malware for financial gain. The floodgates were opened in the 1990s with the advent of the internet as a global commercial network.

Today, cybercrime is a multi-billion dollar industry responsible for causing economic damage measured in the trillions of dollars. According to projections, the cost of cybercrime is on track to hit $10.5 trillion annually by 2025 [26]. Motivated by strong financial incentives, threat actors from wide-ranging backgrounds continuously seek out new ways to abuse computing infrastructure. However, the damage caused by cybercrime often goes beyond a dollar value, making it an even bigger concern.

Some nation-states are known to leverage malicious code as a tool for industrial espionage and theft of intellectual property. Recently it was reported that hackers linked to the Chinese government had undertaken a three-year campaign to steal trade secrets from dozens of technology and manufacturing firms in the U.S., Europe, and Asia, targeting the aerospace and pharmaceutical sectors [23]. As 2023 wound down, a joint investigation by South Korea and the FBI uncovered a cyberattack by hackers connected to North Korea's intelligence office that resulted in the theft of sensitive data related to anti-aircraft defense systems [30].

Nation-state hackers have also been implicated in attacks on critical infrastructure. For example, in 2017 in what has been described as the "worst cyberattack in history," Russian military hackers known as Sandworm (also known as Voodoo Bear) released the NotPetya malware against Ukrainian targets. The devastating malware knocked out infrastructure throughout that country including transport hubs, at least four hospitals, six power companies, two airports, and most Ukrainian government agencies. NotPetya quickly propagated globally and in just one casualty beyond Ukraine's borders, left nearly 800 cargo ships (one fifth of the global shipping capacity) dead in the water [16].

Criminals use ransomware for the extortion of businesses or password stealers to gain access to victims' bank accounts. The business of cybercrime is so lucrative—and established—that some criminals started specializing in providing tools and infrastructure to others for such illicit purposes, in the form of "Ransomware-as-a-Service" or RaaS frameworks. RaaS has now matured to the point that a range of business models are being used including affiliate programs and monthly subscription fees [3]. LockBit is one of the better known RaaS variants and has been responsible for a large percentage of ransomware attacks across the U.S., Canada, Australia, and New Zealand since first being spotted in 2020 [13].

Then there is the issue of hands-on-keyboard attacks. Sophisticated cybercriminals are able to use legitimate tools to infiltrate networks without the use of malware. These so-called fileless attacks are extremely hard to detect, especially for traditional signature-based security software that relies on the detection of known malware. Once considered a novel threat, fileless, hands-on keyboard attacks now make up the majority of observed cyberattacks [11].

In the latest cybercrime development, the advent of Generative AI is providing attackers with advanced tools such as FraudGPT and WormGPT, which can be used to generate much more convincing phishing emails and could even be used to detect network vulnerabilities or create novel malware variants [7].

The financial incentives, potential geopolitical implications, and the improved tooling that attackers now have at their fingertips have a serious implication for defenders in the security industry or in academia: we are facing a larger onslaught of new threats that are unleashed by motivated criminals that seek to identify evasions of security systems for financial gain. To stem this flood of new threats, the industry has turned away from an overreliance on traditional, signature-based detections and begun to embrace Machine Learning (ML)-based approaches.

The use of ML for detection of malware files or execution of malware has been getting significant attention in both academia and in the security industry—and for good reason. With an ever-increasing flood of new threats, the promise of Machine Learning is to establish a more proactive posture as compared to traditional cybersecurity methodology including signatures and heuristics. In addition, Machine Learning offers defensive advantages such as the ability to detect and react to an attack in real-time (without the need of a signature update) and the ability to identify patterns of attack to predict the behavior of malware. At a time when cyberattacks from both criminal elements and state-sponsored groups are on the rise while organizations also struggle with a global cybersecurity workforce shortage [17], ML has the ability to augment human analysts and even automate repetitive tasks—freeing up analyst time. It represents a compelling solution to some of the biggest challenges faced by the cybersecurity industry.

However, operating Machine Learning systems in a production environment is not a trivial task and this challenge is often overlooked in academic works.

To address this frequent shortcoming, we are providing an overview of the additional requirements, constraints, and complications that stem from running a Machine Learning model as part of such a larger system in an industry setting. This includes model-specific requirements such as target false positive rates, corpus size, and corpus diversity, in addition to system-specific needs such as false positive/negative mitigation, robustness and cost of the model generation process, establishment of feedback mechanisms, deployment considerations, and implementation constraints. Lastly, we touch on common compliance and contractual considerations.

2 Considering the Attacker's Perspective

The purpose of Machine Learning in cybersecurity is ultimately to detect malicious payloads and behaviors in order to prevent an adversary from achieving their objective. It is important to recognize this as a conflict where the attacker is human and acting on specific objectives. Such objectives can be either criminal (such as a ransomware actor or access broker) or those of a nation state with goals varying from espionage to sabotage. In this conflict, it is important to recognize several important facts.

First, the adversary is actively working to avoid detection by defenders. Stopping the execution of a single malware file is not stopping the adversary.

Second, as in any conflict, the adversary will adapt to the behavior of the defenders—often in real-time—through command and control systems during an intrusion. A classic example of this can be seen in the behavior of the adversary group known as Scattered Spider. In a 2023 joint Cybersecurity Advisory (CSA), the Federal Bureau of Investigation (FBI) and Cybersecurity and Infrastructure Security Agency (CISA) warned: "To determine if their activities have been uncovered and maintain persistence, Scattered Spider threat actors often search the victim's Slack, Microsoft Teams, and Microsoft Exchange online for emails or conversations regarding the threat actor's intrusion and any security response. The threat actors frequently join incident remediation and response calls and teleconferences, likely to identify how security teams are hunting them and proactively develop new avenues of intrusion in response to victim defenses."[14]

Third, adversaries are persistent and will spend months or even years developing and gaining access in order to achieve an objective. We saw just how devastating this persistence can be with the infamous SolarWinds hack in 2020. The supply chain attack quickly became known as one of the biggest cybersecurity breaches of the 21st century. Attackers targeted SolarWinds' Orion IT management system, which was used by over 30,000 organizations (including some U.S. government agencies) to manage their IT infrastructure. The campaign began with reconnaissance of the Orion build chain. Then in September 2019 adversaries used the SUNSPOT malware to insert the SUNBURST backdoor into Orion [12]. The code was not detected by Orion developers and the backdoor malware ended up being deployed to SolarWinds customers in an Orion update starting in March 2020. The attack was finally uncovered in December 2020.

We see active avoidance in many different ways. One way to represent active avoidance is to consider tactics, techniques, and procedures (TTPs), or the methods that adversaries use to execute malicious activity. The MITRE Corporation maintains a knowledge base of tactics and techniques used by adversaries in public reporting. As of 2023, this library comprehensively documents 201 different high level techniques and 424 subtechniques in MITRE ATT&CK Enterprise. These techniques encompass a variety of TTPs, from manipulating certificates to obfuscating files to disabling security systems in an effort to avoid detection. One specific technique of note is indicator removal, which is a form of operational security, essentially removing the indicators of compromise, files, logs, or other evidence. This is especially relevant given that to create Machine Learning solutions, it is necessary to capture examples of the prediction target, such as a compiled file, script, or log. Further, there are multiple documented techniques of adversarial ML, essentially attacking the capabilities of the Machine Learning model itself in an effort to bypass detection.

Hands-on-keyboard attacks require remote access through a reverse shell, some other command and control system, or a remote administration tool such as Remote Desktop. Gaining remote access enables the threat actor to interact with the system directly, without having to be on the physical premises. They are able to respond to detection by any signature or Machine Learning model by pivoting to alternate techniques. Of note, these fileless attacks leverage live-off-the-land binaries, or LOLBins; they use legitimate tools to accomplish malicious activity. Usage of these tools is effective because the behavior looks extremely similar to legitimate activity of an IT administrator, such as using PSExec or WMI to remotely execute a process across the network. From a Machine Learning perspective, differentiating classes is a harder problem when the clean and dirty classes are almost indistinguishable, especially if instances for training cannot be obtained at scale.

For an adversary to be successful, they only have to be successful once, and they will continually adapt, creating or mutating payloads. Most Indicators of Compromise (IoCs) are exposed to a security solution for the first time during an attack—this makes sense because an IoC represents forensic evidence that security has already been breached (i.e., the attack was not detected and prevented by the security solution). This is a byproduct of continual modification in an attempt to avoid detection and operational security. Training a model or writing a signature that can successfully classify a months-old threat is not valuable in cyberdefense unless that same payload or a similar (to the model) payload is used again in an attempted intrusion. This conflict between adversaries and defenders shows in the data over time through concept drift and increased cost of security such as false positives or disruptions to productivity as the adversary works to avoid detection.

3 File-Based versus Fileless Attacks

Traditionally, most cyberattacks have relied on delivering and executing a malware file on the victim's machine. Such files can be delivered by download through a website, as an email attachment, on a USB thumb drive, or through a variety of other methods. It is this file-based malware that signature-based security solutions are designed to detect and block.

However, in recent years there has been a growing trend toward employing fileless (or hands-on-keyboard) attacks. These attacks do not rely on a file being installed on disk, making them extremely difficult to detect. Signature-based security solutions that scan for known malware files have no visibility into such an attack: there is no malware file to find. The only way to effectively detect a fileless attack is to monitor for suspicious behaviors, which is made even more difficult because attackers in this case are (mis)using legitimate tools and processes.

We track this data for business and enterprise users. Over the past five years, the percentage of observed attacks that are hands-on-keyboard rather than file-based has grown rapidly. In 2019, 40% of attacks were fileless. The following year, fileless attacks surpassed half of all those observed. By 2023 these hands-on-keyboard attacks dominated, accounting for 75% of the total [11]. Figure 1 shows this trend over time.

4 Industry Challenges

Operating Machine Learning-based detectors in an industry setting poses a number of challenges. Not all of these challenges are unique to industry deployments, but they may slant what a successful classifier looks like for industry when compared to academia.

For example, in a production environment, practical issues like being able to handle a large volume of users or maintaining a high response speed and system reliability are very important. Concept drift—the phenomenon where trained ML models can degrade in performance when exposed to constantly changing real-world data inputs—has to be accounted for. Of course, ML models used in production environments will typically have very aggressive detection rate requirements, with minimal false positives and false negatives allowable. In addition, there

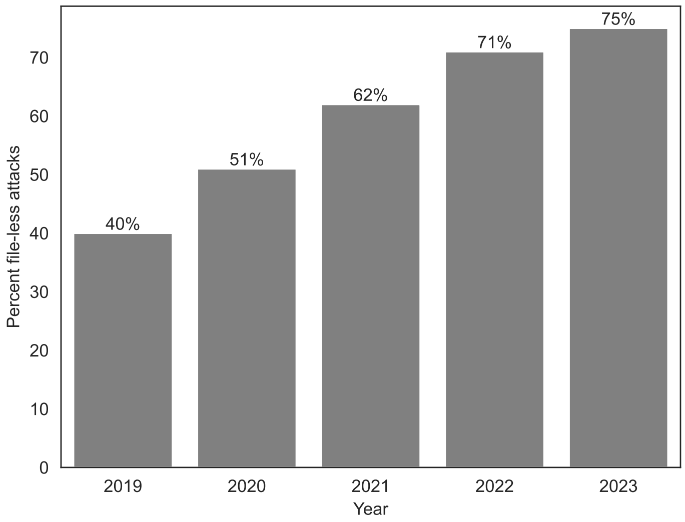

Figure 1: Fraction of fileless attacks year over year.

are many issues specific to data handling (such as privacy and geographic location) that may have little impact in a research or academic setting but are critical when ML is deployed in a production environment.

We will explore some of the challenges in the following sections.

4.1 Data Preparation and Modeling

When defending against cyberthreats, parts of the data utilized are adversary-controlled, meaning that the adversary can choose to a certain degree how to populate it. This has implications on efficacy and possible bypasses, which we explore in Section 5.3, but it also complicates data processing and data preparation.

4.1.1 Outliers and Errors

Let's explore executable malware files as an example. Adversaries are largely free to choose what the file contains, within limitations that the file format imposes and with the constraint that the malicious code needs to be included in some form. For example, a malware author has considerable freedom to choose values like file size, number of resources, or number of sections, which all are commonly used in some form in typical format-specific ML models. This has an impact on data scaling as the adversary can introduce a larger number of outliers[1]. Note that it is acceptable to the adversary if the outlier-producing malware files are easily detected since their intent is to poison the corpus. Naive min/max-based normalization or

[1] The term "outlier" is a slight misnomer since the adversary can easily produce files with such values at high quantities.

standardization to $\mu = 0$ and $\sigma^2 = 1$ can compress the intrinsic distribution of a feature so that it is not useful to a classifier. Therefore, it is crucial to use robust scaling methods such as quantile transforms, or to leverage classifiers that do not rely on scaling such as tree-based methods.

Next, the ability to control the file content also allows the malware author to interfere with parsing and *featurizing* (translating file properties into feature vectors for ML) the file. Parsing data is a notoriously difficult undertaking, and an adversary has several routes to evade detection or to increase the cost to the defender. First, the adversary can introduce errors in the file that break the parsing process of the detector but that are accepted by the operating system loader. The result is that the parser rejects the file or that the parser crashes if sufficient error handling is missing. In the latter case, the detector may be vulnerable to arbitrary code injection or other attacks[35]. For the former case, the parser should not reject files that the operating system loader accepts. However, that is a moving target and can change between operating system versions.

Even compliant files with no format-specific errors can be problematic. As an example, the Windows Portable Format supports resources organized in a tree that can have up to 2^{31} levels [5]. A naive parser may expect a much smaller set of resources. An adversary can then take advantage of this assumption by creating a malware file that takes a very long time to parse, as the parser investigates the large number of resource objects in the file. Depending on the implementation, parsing could time out as a result (analysis of the file fails), or the detector process could be tied up for an excessively long time (a denial of service attack). Defenders need to anticipate such compliant but unusual instances.

4.1.2 Volume, Prevalence, and Efficacy

While sample files to build a training corpus for static analysis are easily obtained, building a corpus for behavioral analysis requires more effort. There are plenty of malware files to be found, but there are comparably fewer malware executions occurring in the field. Furthermore, in a real-world setting many potential malware executions will have already been thwarted by static analysis-based detectors before a malware file started to execute. A possible remedy is to execute malware in a sandbox environment, but care needs to be taken to ensure that environment-specific clues do not leak into the behavioral features extracted from the execution data [21].

In the case of static analysis, files seen in the real world are very diverse, and modeling such diversity requires larger corpora. Fitting a model to a smaller corpus that samples some clusters of that diversity can lead to deceptively positive results, so care needs to be taken that the corpus runs the gamut of real-world diversity.

Given the resources at the disposal of adversaries, detectors need high true positive rates (TPR). Consider two detectors with TPRs of 99% and 99.9%, respectively. Figure 2 shows the probability that after n independent attempts, at least one attempt is successful (for example, at least one malware file executes). For a TPR of 99%, after about $n = 300$ attempts, the probability of at least one success is over 95%. For a TPR of 99.9%, that threshold is crossed around $n = 3000$.

For a state-affiliated threat actor or an eCrime ransomware group, creating 300 unique and unrelated malware files is well within their capabilities. Therefore, a single model—even one with a reasonably high TPR—is not a sufficient detector. Note that models with significantly lower TPRs still have utility, for example as part of an ensemble classifier, or as a component of a larger system ("defense in

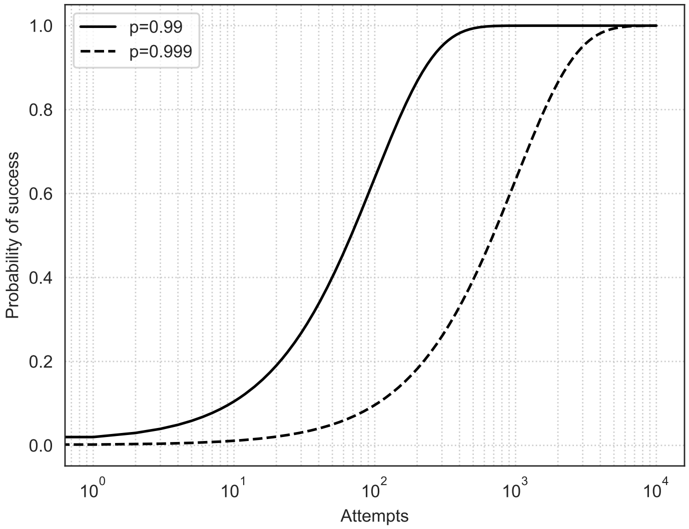

Figure 2: Probability of at least a single success after repeated attempts.

depth").

Next, a detector suitable for field deployment also needs an extremely low false positive rate (FPR). FPs are disruptive for workers, for example preventing them from running productivity applications, and they cause alert fatigue for security analysts. Even for a detector with a relatively low FPR, the detector is prone to producing a large number of FPs due to the low prevalence of malware.

Consider a behavioral detector designed to stop malware executions that correctly identifies each one. In other words, it has a TPR of 100%. Table 1 shows the prevalence of attempted malware launches per platform. For Windows, malware files account for roughly one execution in 31 million. At an FPR of 0.00001%, this detector would produce one TP and 3 FPs and hence would only have a precision of 25% (one in four detections are correct).

Table 1: Malware file executions per platform

Platform	Prevalence of Malware File Executions
Windows	1: 31,000,000
Macintosh	1: 72,000,000

The numbers are more favorable if we look at unique files, deduplicated by their hash value. Table 2 shows this data. For Windows, one in about 4000 files that were attempted to launch was malicious. As an example, an FPR of 0.1% would result in 4 unique files to be erroneously identified as malicious, resulting in a precision of 20%.

Table 2: Unique malware files launched per platform

Platform	Prevalence of Unique Malware Files Launched
Windows	1: 4,000
Macintosh	1: 18,000

Therefore, when designing ML-based detectors, care needs to be taken with respect to the expected prevalence. Contextual cues can help, however, in creating more favorable biases. For example, a file already stored on the hard disk is less likely to be malware than a file that was just downloaded.

4.2 Operations

Machine Learning Operations (MLOps) are a critical aspect of successfully utilizing malware classifiers. In a research setting, we commonly focus on creating the best classifier on a specific dataset. In industry, it is instead critical to create a *process* that *robustly* yields new viable updated models.

Once a model is in production, any possible false positives or false negatives (FNs) need to be rapidly addressed, generally before there is time to retrain the model. Hash-based block or allowlisting is an easy option, but malware often adds randomizations to alter file checksums. Fuzzy hashes such as ssdeep, peHash, or Lempel-Ziv Jaccard Distance are alternatives that are more robust to such alterations [29, 34]. Approaches in the model's feature space are also possible, such as utilizing the distance from a given centroid. Distance-based algorithms tend to require pairwise comparison, so a lookup against an allowlist does require $O(n)$ time. Algorithms such as Locality-Sensitive Hashing only require an $O(1)$ lookup to determine set membership.

Some FPs and FNs are identified as part of field feedback, but many new gaps are uncovered using automated analysis and by human analysts. All this data needs to be included into the training round for future models. This can shift the distribution of types of files in the corpus in undesired ways. Therefore, managing the composition of the corpus is critical to ensure robust delivery of future models. While a great deal of attention is devoted to improving classifiers and their training, this critical aspect of corpus management is garnering less consideration.

Upon deployment of a new model, a fraction of detections tends to change. This results in new FPs and new FNs. First, the modeling process needs to reduce these changes. Second, prior to release, any remaining FPs and FNs need to be mitigated lest they cause problems in the field.

Another important metric is training speed. Models that can be trained faster allow for easier experimentation. If a model shows issues during field testing, a shorter training time reduces the time to remedy the issues with a replacement model.

Feature extraction and inference speeds are also critical—especially for models deployed on endpoints—as they have direct impact on the user experience. The same goes for the model memory footprint, which corresponds to memory no longer available to the user of the protected system. A model that requires excessive memory use can impact system performance and slow other applications, frustrating users and adding to the workload of IT staff who respond to complaints. Non-ML antivirus solutions in particular have attracted the ire of users due to their large signature databases containing numerous rules that require slow, sequential evaluation.

4.3 Cost Considerations

False positives occur when a cybersecurity solution flags a file or action as being a threat, when in fact it is harmless. The problem with FPs is that analysts must investigate them as though they were a true positive (TP), and that has an impact on an organization's security team. False positives increase stress levels (are we under attack?) and contribute to alert fatigue—this becomes dangerous when those alerts are ignored. There is also the issue of opportunity cost: any time that analysts spend investigating FPs is time that they are not available to work on an actual breach. Analyst shortages and skills gaps only make the problem of chasing false positives worse.

In 2015, SecurityWeek attempted to put a dollar value to the cost of triaging and investigating false alerts [20]. This number includes both false positives and false negatives and doesn't account for factors like opportunity cost. In addition, between inflation and increased attack volume, the numbers would be much higher today. However, it does drive home the point that FPs are costly: according to the report, U.S. organizations received an average of 16,937 security alerts per week. They only investigated 705 of those alerts (just 4% of the total), but ended up wasting 21,000 hours per year at a cost of roughly $1.3 million annually on what turned out to be false alarms.

The catch is, cybersecurity solutions will have some false positives if they are to be proactive and able to detect novel threats; an ML model that only detects known malware is not significantly better than signature-based approaches. The key is to reach a balance so that an ML model's FP rate is manageable, yet it is still able to alert to previously unknown threats. Since ML models are probabilistic in nature, some security products allow organizations to configure this trade-off in accordance with their capabilities.

Regardless of whether an organization is a small or medium-sized business, multinational enterprise, government agency, or a not-for-profit, it faces an environment where cybersecurity attacks have increased in frequency as well as complexity. What makes this situation even more challenging is that the relentless escalation of threats is occurring concurrently with an ongoing global shortage of trained cybersecurity professionals.

According to the World Economic Forum's 2023 Global Cybersecurity Outlook, sectors including electricity, payments, and hospitals are most at risk because they are experiencing the largest critical gaps in their ability to onboard skilled cybersecurity professionals [6]. As the report points out, this shortage leaves key parts of our society—critical infrastructure, banking/retail, and healthcare—especially vulnerable to cyberattacks. To put the scale of this shortage in more concrete terms, the World Economic Forum report estimates the global shortage of cybersecurity workers to be 3.4 million.

In addition, retaining cybersecurity professionals is a problem. The existing talent pool is feeling significant workload pressure, with 70% of cybersecurity workers reporting feeling overworked and 24% of cybersecurity leaders preparing to leave their jobs as a result of work-related stress [19]. Burnout needs to be addressed as part of any solution.

The promise of incorporating Machine Learning in cybersecurity solutions is that doing so augments the capabilities of an organization's human analysts—increasing their effectiveness, reducing response times, automating some repetitive tasks, and helping to fill the gaps left by a shortage of trained cybersecurity workers. This includes traditional ML and AI approaches that allow the analysis of large amounts of complex data as well as Generative AI, which can help with automation and automated reasoning during investigations and response.

5 Malware as an Adversarial Domain

5.1 The Adversary Problem

The domains of malware development and malware-free intrusion are fundamentally driven by adversarial motives. In fact, it has been posited before that:

> *Individuals and organizations do not have a malware problem. They have an adversary problem.*

Historically, when malware detection was heavily dependent on digital signatures of malware samples, evasion could be as easy as employing a small-scale function-preserving modification of a sample (hash busting). The adversary was actively taking measures to evade detection and/or to conduct their activity without detection or interruption, but the method of attack was a malware file—even if it was one that had been slightly modified to avoid signature-based detection. The introduction of Machine Learning models (and their ability to rapidly detect such attacks) has made tactics like these largely ineffective, at least in attacks on environments protected by commercial cybersecurity products. As a consequence, evasion requires more sophistication now. On the one hand, this drives the shift from malware to hands-on-keyboard and fileless attacks. On the other hand, threat actors can employ weaknesses in ML-based detectors, both for file-based malware and fileless attacks.

As adversaries explore novel routes to accomplish their objectives, it is critical for defenders to think beyond specific technical approaches and adopt an adversary-centric approach to cyber defense. Such an approach seeks to understand the characteristics of known adversaries, including:

1. Their capabilities (such as technical sophistication and ability to attack different operating systems)
2. Their goals (for example information extraction or functional disruption)
3. The adversary's available resources and limitations (developers, infrastructure)
4. The motivations driving their cyberattack campaigns (in most cases the driver will be financial, the search for trade secrets, or political activism)

5.2 Malicious Uses of Artificial Intelligence

Adversaries are starting to leverage AI as well. The National Cyber Security Centre (NCSC) is a UK government organization tasked with protecting the private and public sectors in that country from computer security threats. In a January 2024 report on the near-term impact of AI on the efficacy of cyber operations, the NCSC came to the conclusion that Artificial Intelligence will "almost certainly increase the volume and heighten the impact of cyber attacks over the next two years." [27]

Among the highlights from the report:

- The AI-based threat through 2025 will come from using AI to evolve and enhance existing tactics, techniques and procedures (TTPs).
- All types of adversaries (state-sponsored and criminal) at various degrees of skill level are already employing AI to some degree.
- AI is providing capability uplift for reconnaissance and social engineering (as seen, for example, in the use of Generative AI to craft much more sophisticated phishing emails).
- More sophisticated use of AI is highly likely to be restricted to more advanced adversaries for the short term as they have access to both cyberattack and AI expertise as well as quality training data for ML models.
- These advanced adversaries will be able to use AI to analyze exfiltrated data more rapidly and more effectively, then use that data to train ML models.
- The commoditization of AI-enabled capability in criminal and commercial markets will almost certainly make improved capability available to cyber crime and state actors by 2025.

In addition, the NCSC report notes that "AI has the potential to generate malware that could evade detection by current security filters, but only if it is trained on quality exploit data. There is a realistic possibility that highly capable states have repositories of malware that are large enough to effectively train an AI model for this purpose."

We appear to be rapidly approaching a tipping point in the weaponization of Generative AI. On February 14, 2024, OpenAI announced that it, in partnership with Microsoft, had shut down the ChatGPT accounts for five different state-sponsored threat actors, including China and North Korea-based groups. These adversaries were misusing the AI tool for tasks that included debugging code, generating scripts, researching ways malware can evade detection, and to understand publicly available vulnerabilities [28]. In short, this incident shows that adversaries have already progressed beyond using AI and ML tools to simply write more convincing phishing emails.

Given the threat that Machine Learning as a cyber weapon poses, it is understandable that the cybersecurity community is working hard to stay on top of what adversaries are doing in this space.

5.3 Adversarial Machine Learning

With ML becoming a critical technology for defenders, it is also moving into the crosshairs of adversaries. The field of Adversarial Machine Learning (AML) investigates the attack methods for different phases of ML operations [33].

In the cybersecurity application of predictive ML (for example to determine whether a behavior or a file is clean or suspicious/malicious), the phases of data collection, model training, and model serving offer possible attack surfaces. These attack surfaces could include data poisoning, model extraction, or model evasion.

5.3.1 Creating Evasive Files

Looking again at the classification of files using static analysis, while ML is resilient to traditional hash busting techniques, AML offers novel ML-specific routes to evade detection. In 2019, researchers from Skylight Cyber provided a real-world example of how an ML-based security product can be attacked [2]. By reverse engineering the product, they identified an allowlist of applications. They inferred that files on the allowlist are also likely to be in the training dataset. Furthermore, they identified that the detector utilized the presence of specific strings as features. By adding strings of allowlisted files to a malware file, they were able to successfully trick the product to misclassify the file.

While this evasion was achieved with significant direction from the attacker, more automated approaches are possible. For example, if the attacker has access to the ML model's confidence value, then the attacker can employ small changes to a malware file that change the feature vector but not the file behavior and observe if these changes reduce the confidence the model outputs. Executing this procedure a large number of times will often eventually yield an evasive file.

If the detector outputs only a yes/no output whether a file is malware, then the attacker can use these outputs to train a proxy model that outputs a confidence. A file that successfully evades the proxy model will very often also evade the original model.

A growing body of research investigates various techniques that can be used to reliably create evasive files, including:

- Deep reinforcement learning (DRL) such as actor-critic with experience replay (ACER) was used to generate evasive portable executable (PE) samples [1]: the authors used different datasets of malware families to train agents and tested on holdout samples not seen during the training phase. The evasion rate on the holdout samples ranged between 10% and 24% and were found to be greater than for a random agent (9% to 23%).
- Double deep Q-network (DDQN) was used for content injection in PDFs [25] and compared to Deep q-network (DQN) and Deep SARSA (state, action, reward, state, action): while all three methods showed good learning outcomes, the average steps required to generate an evasive sample was different ranging from 14 for DDQN, 15 for deep SARSA, and 17 for DQN. Importantly, the authors also investigated the evasive success over time, decreasing from 93% after one year to 87% after five years, and demonstrated that regular training continuation (or online fine-tuning) alleviates this performance decay somewhat, achieving 99% and 97% evasion rates after one and five years, respectively.
- More classic reinforcement learning (RL) methods such as the multi-armed bandit (MAB) were applied for PE [31]: the authors achieved evasion rates of 74% to 98% for the EMBER and MalConv datasets, respectively, which compare favorably with the success rates of a random agent (15% to 33%). Importantly, the authors suggest an algorithm to post-process an evasive

sample that was generated with a sequence of modification steps from an original sample: the action minimization seeks to (1) remove actions from the sequence that are not needed to obtain a classification evasion and (2) replace an action with a microaction, if applicable (for example, append several bytes of overlay at the end of binary ⟶ minimize to append only one byte) if the new sample evades classification. In this manner, the adversarial samples obtained are more similar to the original, unmodified sample.

- Population-based training (PBT) such as Genetic Programming (GP) for PE [8]: for each PE sample from a training dataset, an initial population was generated by applying different adversarial modifications. Then, for several iteration steps, the modified sample population was evaluated (according to a fitness function including whether the modified sample is evades detection or not), subjected to a selection process (removing a proportion of samples with lower fitness scores), and processed via crossover (swapping modification steps between samples), mutation (replacing a proportion of modification steps with a random modification) and reproduction (to bring the population size back up). This multi-step workflow depends on a number of hyperparameters that warrant exploring and tuning: population size, mutation rate, crossover rate, reproduction rate. The authors demonstrated that their approach allows them to generate more varied and more valid malware samples than can be obtained with random modifications. They also highlighted that samples obtained from learning to evade one commercial classifier lead to high evasion rates with two other commercial classifiers with cross-evasion rates ranging from 25% to 83%.

5.3.2 Adversarial Training

It is possible to evade at least some real-world detectors by repeatedly employing some modification and obfuscation techniques that result in a file's feature vector to change. For PE files, examples of modifications include:

- Modify header: for example, change metadata or file characteristics
- Add overlay: for example, add bytes to the end of the file
- Add section: adding a block of code or data to the file
- Rename section: changing the name of a section
- Obfuscate imports: hiding what library functions the executable invokes

One way to increase the robustness of a classifier with respect to evasive samples is the enrichment of the model training corpus with modified samples, so called adversarial training. Our production pipeline includes a common framework that automates the creation and inclusion of such samples, applying more than 100,000 permutations of obfuscation techniques to various file formats. For comparison, we observed that some real-world detectors fail to detect adversarial samples with as little as 30 bytes of modifications to the sample.

On the one hand, the common adversarial framework allows us to anticipate possible susceptibility of a candidate classifier to particular AML attacks. Furthermore, results can highlight possible new avenues for feature engineering, if needed. On the other hand, the generation of modified samples in bulk allows enrichment of the model training corpus.

Table 3: Partial area under the curve (pAUC) for a binary classifier trained with standard samples or standard and adversarially modified samples and evaluated on a separate holdout set of unmodified or modified samples (closer to one is better).

	Holdout Set	
Training Set	Standard	Adversarial
Standard	0.97175	0.92012
Std + Adv	0.97700	0.98486

In our analyses we have repeatedly observed that an adversarially trained classifier not only shows better detection rates for new holdout adversarial samples, but also demonstrates better predictive power on the holdout set of unmodified samples. Table 3 shows the results of one such study. Introducing adversarial samples during training increased the partial area under the curve (pAUC) by more than 5 in 1,000. Additionally, the enrichment can improve the training corpus quality, for example with respect to label balance of a nascent malware family with fewer than 1,000 samples collected from sources in the wild.

During the data collection phase, data scientists and security/malware researchers work closely together to combine near real-time sample feeds and process them via triage, evaluation, analysis, and ultimately selection for inclusion in a classifier model training and evaluation phase.

The integrity and validation of the processing workflow are important to reduce the risk of data poisoning. The latter refers to adversary methods aiming to compromise the model performance during the serving phase by introducing targeted samples into the training process. Continuous monitoring of feature drift and anomalies or outliers together with similarity and clustering supports scalable analysis and processing of a large sample corpus. The sample similarity analysis can be performed at various levels: fuzzy hash similarity, static feature similarity (based on static sample analysis), behavioral feature similarity (for example, based on events recorded during detonation in a sandbox environment [15]), or statistical model decision path similarity (for example, leaf similarity in tree-based classifiers such as xgboost [10]).

6 Compliance

When operating in a production environment, data processing activities must be compliant with various laws and regulations. This includes adherence to international privacy laws (e.g. GDPR) as well as upcoming AI regulation and guidance. The activities must also fulfill contractual obligations. These requirements have impacts on building corpora and on inference, which we explore in this section.

6.1 Storage of Sensitive Information

Looking at static analysis of executable files, many files for malware and clean applications can be easily obtained to build a corpus. Websites such as Hybrid

Analysis or VirusTotal are popular sources for files. However, it is not always possible to get samples of internal applications of customers, for example in cases where these applications contain confidential intellectual property or trade secrets. This issue is even more pronounced when classifying document files. As a result, mitigating misclassifications is more complicated and files cannot easily be analyzed to validate that they are indeed of the correct class.

If there are no changes to the feature space, building a corpus of feature vectors instead of samples can be a workaround to get such files into training. Files are transformed into feature vectors, which are transmitted to the cloud without revealing sensitive or protected information (such as personal information) that may be contained within these files. However, while this approach enables training based on customer feedback, it does not allow analysts to validate the correctness of the feedback by reverse engineering of the sample. This is especially of concern with surreptitious attacks in mind, where deeper validation is often required.

For behavioral classification, parts of the pertinent data can be subject to legal or contractual data retention and deletion requirements. Those parts cannot be leveraged in a compliant manner for building up a training corpus due to their ephemeral nature unless all legitimate interests and objectives are considered in a positive-sum "win-win" manner as outlined by Ann Cavoukian in her 7 Foundational Principles of Privacy by Design for Implementation and Mapping of Fair Information Practices [9]. Privacy is often positioned in a zero-sum manner as having to compete with other legitimate interests, design objectives, and technical capabilities in a given domain. Ann Cavoukian's Privacy by Design idea rejects taking such an approach—it embraces legitimate non-privacy objectives and accommodates them in an innovative positive-sum manner.

For the data that can be retained, it needs to be ensured that personal identifiable information is removed, in order to mitigate the risk of legal challenges associated with reidentification of individuals within datasets. For instance, this can be achieved by scrubbing based on known/expected locations for personally identifiable information/sensitive content. File subpaths under `\Users`, `/Users`, `/home`, etc. that commonly contain user names must be replaced with a generic string. Telemetry containing customer IDs or endpoint IDs should be removed or replaced in an irreversible manner in order to dissociate the data from a particular customer environment. In addition to the scrubbing routines employed, a regular review cadence can help to identify any potentially missed personally identifiable information/sensitive content for manual removal.

In parallel, for the data that can be retained and should be used, further legal conditions and limitations must be reflected.

6.1.1 Cross-Border Data Transfer

Data collection and use can be limited by cross-border data transfer regulation. Various jurisdictions and the European Union (EU) require certain safeguards for transferring personal information of their residents to third countries. To that end, data collection must be well prepared based on diligent legal review and actions for safeguarding international data transfers.

6.1.2 Data Localization

To date, only very few jurisdictions (e.g. the Russian Federation) apply data residence legal requirements that require processing data including personal information in a certain territory. However, several jurisdictions and supranational organizations such as the European Union are considering data localization requirements, which can impose that certain data may be processed only within a specific territory. This would pose several burdens on use of data for training and Machine Learning and AI approaches. First, at training time, several corpora (one per affected territory) would be required, reducing the size of the available training dataset. Second, at inference time, features at global scale could no longer be effectively computed [32].

6.2 Lawful Processing

Globally accepted privacy principles require the lawful processing of personal information. For instance, the EU General Data Protection Regulation (GDPR) only allows processing of personal information if processing is agreed between parties and necessary for the performance of a contract, the data subject has given consent to the processing of his or her personal data for Machine Learning purposes, or processing is presumably necessary for the purposes of the legitimate interests pursued by the body who intends to use the data for Machine Learning purposes. However, relying on legitimate interest requires accountability in terms of having the ability to demonstrate that there are no overriding interests or fundamental rights and freedoms of the data subject, which require protection of its personal data. In this respect, it shall be taken into account that the French data protection authority (Commission Nationale de l'Informatique et des Libertés - CNIL) issued guidance on the reuse of personal data by service providers for their own purposes under the GDPR. The guidance addresses the use of personal data for purposes broader than just strictly providing services; for example, to improve products or services or train artificial intelligence and machine learning algorithms [24].

Finally, GDPR's emphasis on transparency and accountability requires organizations to provide clear and understandable information to individuals about how their data is being used in AI systems, including the purpose of processing, the legal basis for processing, and individuals' rights about their data.

6.3 AI Regulation

Similarly, the EU AI Act (the "Act") will regulate and govern the development and deployment of Artificial Intelligence systems within the European Union. The Act defines covered parties as either providers or deployers of AI systems and applies if such a party is established in the EU, or if the output produced by such party's AI system is used in the EU.

The Act adopts a classification system and imposes different compliance obligations on covered parties, depending on the risks presented by the AI system in question. AI systems can be classified as "Prohibited" or "High-Risk" under the Act. Even where an AI system does not fall under one of these risk categorizations (and is therefore not a regulated system under the Act), it may still be subject to overarching transparency obligations (for example, where it interacts with humans or produces AI-generated content).

Under the Act, organizations are required to adhere to stringent guidelines when using personal data to train and deploy AI models, ensuring transparency, fairness, and respect for individuals' privacy rights.

Developers of AI systems would be prohibited from deploying AI systems that present unacceptable risks to the fundamental rights of individuals. Examples of "prohibited AI systems" include systems that are used for social scoring based on social behavior or personal characteristics; AI systems designed to explore vulnerabilities that result in significant harm and the material distortion of behavior; and AI systems that engage in the untargeted scraping of facial images from the internet or CCTV footage to create facial recognition databases [18].

Deployers of "high-risk AI systems" will have a significant number of direct obligations under the AI Act, including the obligations to implement risk mitigation, ensure that datasets are high quality, log activity with detailed documentation, assign human oversight of the AI system to a person with the necessary competence, training, authority and support; and to provide clear information for users [4].

Examples of high-risk AI systems (as listed in Annex III of the Act) include, but are not limited to, systems intended to be used as safety components in the management and operation of critical digital infrastructure, road traffic, and the supply of water, gas, heating, or electricity; AI systems used to monitor prohibited behavior of students during tests in the context of/within education and vocational training institutions; AI systems intended to be used for recruitment of and evaluating the performance of employees; and AI systems intended to be used by public authorities to evaluate the eligibility of individuals for essential public assistance benefits and services, including healthcare services [18].

AI systems that are not high-risk but pose transparency risks will be subject to specific transparency requirements under the Act. Deployers of AI systems that are intended to directly interact with individuals and act like a human (such as chatbots) or AI systems designed to generate content (e.g., to prepare news articles) will be required to inform users they are interacting with a machine. Deepfakes must be labeled, and users must be informed when a biometric categorization or emotion recognition system is used [18]. This information must be provided to users in a clear and distinguishable manner at the latest when the user first interacts with the AI system or is exposed to the AI-generated content.

7 Conclusions

As an increasing number of financially or politically motivated adversaries create more plentiful and more sophisticated threats, Machine Learning-based malware and cyberattack detection is becoming an increasingly important tool to defend systems and networks.

Machine Learning is proving to be the most valuable new weapon in the cyberdefense arsenal, acting as a force multiplier that also supports a much more proactive approach to defending against cyberattacks. The promise of ML is considerable, however what is often missing from the conversation is the added complexity of deploying a Machine Learning model in a production environment. The reality is that bringing ML to the frontlines in the real world requires additional considerations beyond creating a single suitable model. Success in an academic or research setting does not necessarily translate to an ML model that will perform to high standards in real life use.

We have provided a detailed explanation of the many additional requirements, constraints, and complications that will typically be encountered for an ML model to be successfully deployed in production, as well as the relevant context for each. This includes the rapidly growing range of TTPs employed by attackers, the need for extremely high true positive rates, the cost of false positives, adversarial ML, risk of data poisoning during model training, legal regulations for data processing/storage requirements, and more.

We believe that by highlighting the additional challenges of productionizing Machine Learning models, further avenues for research in this exciting field will be explored and the cause of deploying ML in a cybersecurity setting will be advanced.

8 Acknowledgments

The authors would like to thank Nick Yarbrough, Jeff Kaplan, Noah Bason, and Jared Engstrom for their help and support.

References

[1] Hyrum S. Anderson, Anant Kharkar, Bobby Filar, David Evans, and Phil Roth. Learning to evade static PE machine learning malware models via reinforcement learning, 2018.

[2] Adi Ashkenazy and Shahar Zini. Attacking machine learning, 2019. URL https://skylightcyber.com/2019/07/18/cylance-i-kill-you/Cylance-AdversarialMachineLearningCaseStudy.pdf.

[3] Kurt Baker. Ransomware as a service (RaaS) explained, 2023. URL https://www.crowdstrike.com/cybersecurity-101/ransomware/ransomware-as-a-service-raas/.

[4] Jedidiah Bracy and Alex LaCasse. EU reaches deal on world's first comprehensive AI regulation, 12 2023. URL https://iapp.org/news/a/eu-reaches-deal-on-worlds-first-comprehensive-ai-regulation/.

[5] Karl Bridge et al. PE format, 2023. URL https://learn.microsoft.com/en-us/windows/win32/debug/pe-format.

[6] Gretchen Bueermann et al. Global cybersecurity outlook 2023, 2023. URL https://www.weforum.org/publications/global-cybersecurity-outlook-2023/.

[7] Matt Burgess. Criminals have created their own ChatGPT clones, 2023. URL https://www.wired.co.uk/article/chatgpt-scams-fraudgpt-wormgpt-crime.

[8] Raphael Labaca Castro, Corinna Schmitt, and Gabi Dreo. AIMED: Evolving malware with genetic programming to evade detection. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 240–247, 2019. doi: 10.1109/TrustCom/BigDataSE.2019.00040.

[9] Ann Cavoukian. The 7 foundational principles: Implementation and mapping of fair information practices, 2010. URL https://iapp.org/media/pdf/resource_center/pbd_implement_7found_principles.pdf.

[10] Dragos Georgian Corlatescu, Alexandru Dinu, Mihaela Gaman, and Paul Sumedrea. EMBERSim: A large-scale databank for boosting similarity search in malware analysis, 2023.

[11] CrowdStrike. 2024 Global Threat Report, 2024. URL https://www.crowdstrike.com/global-threat-report/.

[12] CrowdStrike Intelligence Team. SUNSPOT: An implant in the build process, 2021. URL https://www.crowdstrike.com/blog/sunspot-malware-technical-analysis/.

[13] Cybersecurity & Infrastructure Security Agency. Understanding ransomware threat actors: LockBit, 2023. URL https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-165a.

[14] Cybersecurity & Infrastructure Security Agency. Cybersecurity advisory Scattered Spider, 2023. URL https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-320a.

[15] Anusha Damodaran, Fabio Di Troia, Visaggio Aaron Corrado, Thomas H. Austin, and Mark Stamp. A comparison of static, dynamic, and hybrid analysis for malware detection, 2022. URL https://doi.org/10.48550/arXiv.2203.09938.

[16] Andy Greenberg. The untold story of NotPetya, the most devastating cyberattack in history, 2018. URL https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/.

[17] Michael Hill. Cybersecurity workforce shortage reaches 4 million despite significant recruitment drive, 2023. URL https://www.csoonline.com/article/657598/cybersecurity-workforce-shortage-reaches-4-million-despite-significant-recruitment-drive.html.

[18] Hunton Andrews Kurth LLP. Final draft of EU AI Act leaked, 2 2024. URL https://www.huntonprivacyblog.com/2024/02/01/final-draft-of-eu-ai-act-leaked/.

[19] Akshay Joshi, Sean Doyle, and Natasa Perucica. The cybersecurity skills gap is a real threat——here's how to address it, 2023. URL https://www.weforum.org/agenda/2023/05/the-cybersecurity-skills-gap-is-a-real-threat-heres-how-to-address-it/.

[20] Eduard Kovacs. False positive alerts cost organizations $1.3 million per year: Report, 2015. URL https://www.securityweek.com/false-positive-alerts-cost-organizations-13-million-year-report/.

[21] Sven Krasser, Brett Meyer, and Patrick Crenshaw. Valkyrie: Behavioral malware detection using global kernel-level telemetry data. In *Proceedings of the 2015 IEEE International Workshop on Machine Learning for Signal Processing*, September 2015.

[22] John Leyden. The 30-year-old prank that became the first computer virus, 2012. URL https://www.theregister.com/2012/12/14/first_virus_elk_cloner_creator_interviewed/.

[23] Sean Lyngaas. Chinese hackers cast wide net for trade secrets in US, Europe and Asia, researchers say, 2022. URL https://www.cnn.com/2022/05/04/politics/china-hackers-economic-espionage-manufacturing/index.html.

[24] Gabe Maldoff and Omer Tene. CNIL sets parameters for processors' reuse of data for product improvement, 2022. URL https://iapp.org/news/a/cnil-sets-parameters-for-processors-reuse-of-data-for-product-improvement/.

[25] Zhengyang Mao, Zhiyang Fang, Meijin Li, and Yang Fan. EvadeRL: Evading PDF malware classifiers with deep reinforcement learning, 2022. URL https://doi.org/10.1155/2022/7218800.

[26] Robert Muggah and Mac Margolis. Why we need global rules to crack down on cybercrime, 2023. URL https://www.weforum.org/agenda/2023/01/global-rules-crack-down-cybercrime/.

[27] National Cyber Security Centre. The near-term impact of AI on the cyber threat, 2024. URL https://www.ncsc.gov.uk/report/impact-of-ai-on-cyber-threat.

[28] OpenAI. Disrupting malicious uses of AI by state-affiliated threat actors, 2024. URL https://openai.com/blog/disrupting-malicious-uses-of-ai-by-state-affiliated-threat-actors.

[29] Edward Raff and Charles Nicholas. Lempel-Ziv Jaccard distance, an effective alternative to ssdeep and sdhash. *Digital Investigation*, 24:34–49, 2018.

[30] James Reddick. North Korean hackers stole anti-aircraft system data from South Korean firm, 2023. URL https://therecord.media/north-korea-hackers-stole-anti-aircraft-system-data.

[31] Wei Song, Xuezixiang Li, Sadia Afroz, Deepali Garg, Dmitry Kuznetsov, and Heng Yin. MAB-Malware: A reinforcement learning framework for attacking static malware classifiers, 2021.

[32] Peter Swire, DeBrae Kennedy-Mayo, Drew Bagley, Avani Modak, Sven Krasser, and Christoph Bausewein. Risks to cybersecurity from data localization, organized by techniques, tactics, and procedures, June 2023. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4466479.

[33] Apostol Vassilev, Alina Oprea, Alie Fordyce, and Hyrum Anderson. Adversarial machine learning: A taxonomy and terminology of attacks and mitigations, 2024. URL https://doi.org/10.6028/NIST.AI.100-2e2023.

[34] Georg Wicherski. peHash: A novel approach to fast malware clustering. *LEET*, 9:8, 2009.

[35] Feng Xue. Attacking antivirus. In *Black Hat Europe*, 2008. URL https://www.blackhat.com/presentations/bh-europe-08/Feng-Xue/Whitepaper/bh-eu-08-xue-WP.pdf.