

# The real reason why malware detection is hard

---

 [gdatasoftware.com/blog/2022/06/37445-malware-detection-is-hard](https://gdatasoftware.com/blog/2022/06/37445-malware-detection-is-hard)

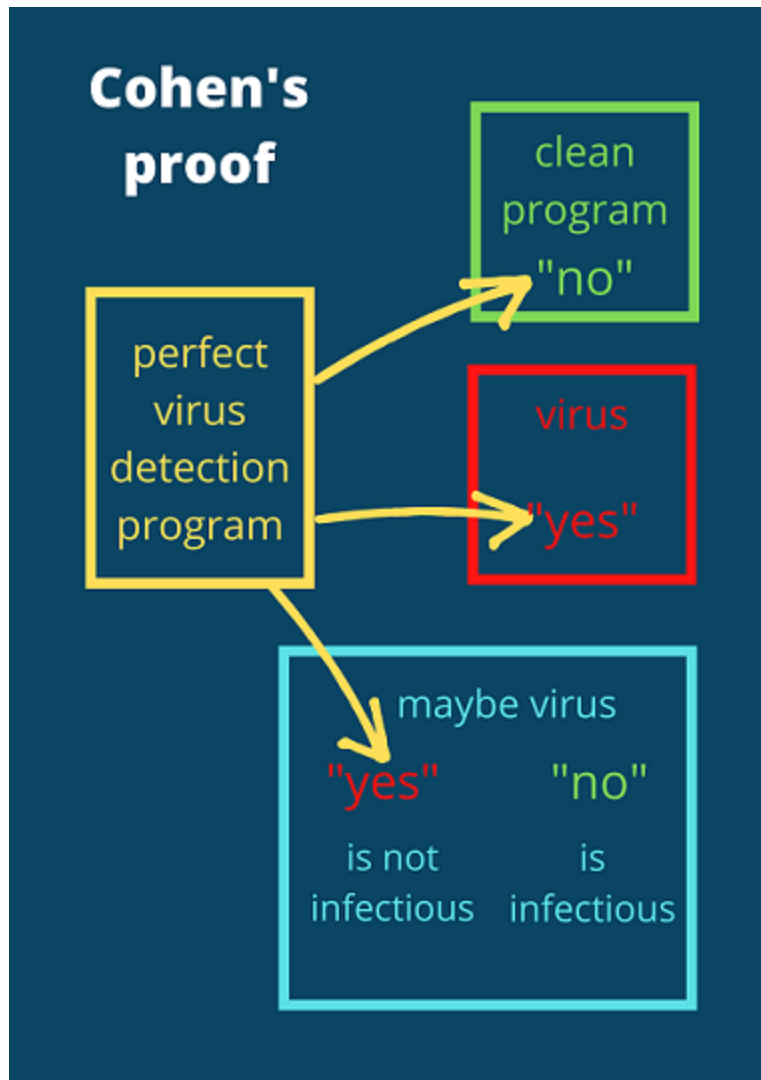


## Malware

Researchers develop an AI with a 98% malware detection rate and 5% false positive rate. If you think this is a splendid technology for antivirus software, this article might change your mind.

## **Malware detection is undecidable but not impractical**

---



Indirect proof for undecidability of virus detection by Fred Cohen. The "maybe virus" is only infectious if the perfect detection program decides it is clean.

Fred Cohen already gave proof in 1984 that virus detection is an undecidable problem. He did this by assuming that a perfect virus detection program exists, then he constructed a potential virus for which this detection program cannot provide a correct answer because it acts only infectious if the perfect detection program thinks it is clean [cohen84]. With all viruses being malware, an analogue proof by contradiction can be constructed for malware detection.

But "undecidable" in the mathematical sense does not mean there is no practical solution for it. An undecidable problem is a decision problem that cannot provide a correct answer in every case. So if you have a program that answers 99.9999% of all cases correctly but 0.0001% of them incorrectly, the problem might be both, undecidable and still practically solvable. We have sufficient solutions for similar problems. One example is the travelling salesman problem which has applications in logistics for package routing systems or in circuit manufacturing processes (a three minute video explanation is here). Although there is

**yet** no known algorithm that can solve the travelling salesman problem in a reasonable time (it is NP-hard), there are sufficient approximate solutions to calculate routes for package delivery.

The same is true for antivirus products which are shielding systems against malware as best as possible. But unlike package recipients, malware developers are actively trying to create malware in such a way that antivirus products make wrong decisions and do not detect these programs. That would be akin to people deliberately and consistently trying to find package delivery routing inputs that are not solved optimally in a reasonable time.

Consequently malware detection is hard by nature, and needs constant work and improvements to stay useful. Yet, antivirus programs have acquired a bad reputation as being outdated, inflexible and old-fashioned. Is there any truth to that?

## Others say they can do it better

---

Judging by the usual claims of media and advertisers, one gets the impression that antivirus products are deliberately using outdated technology and refusing to build their systems with those new AI and other technology advancements that have been researched. How often do we see articles like "Traditional antivirus is dead" and "This AI is better than antivirus"?

That begs the question: If antivirus products could be better that easily, why wouldn't they adapt the new technologies? The truth is:

1. They do.
2. They cannot afford false positives and are unfairly being compared to applications that can.

## Myth 1: Antivirus products use outdated technologies

---

According to the media, antivirus products only search for patterns in files and compare file hashes to blocklists. These detection mechanisms are still there, but antivirus products have been using other malware detection technologies for at least two decades.

## Why Antivirus is Not Enough

One of the most common cybersecurity technologies used for malware detection is the antivirus solution. Antivirus is a program located at an endpoint on the customer's machine that continuously checks a library of known threat signatures to identify suspicious or malicious files, in order to quarantine or destroy them before they can cause damage. However, even when antivirus solutions are used correctly and kept up to date, they can only detect known threats. This means that antivirus solutions cannot protect networks against undisclosed or zero-day attacks, which

Article from 2021 claims antivirus only detect known threats because it relies on signatures [art1]. Both is wrong: Antivirus software does not rely solely on signatures and signatures do not only detect known malware. (click to enlarge)

The best defense for enterprises is to be informed of the evolving threats, act quickly in the event of an attack, and be proactive with advanced security solutions. “You have to have a next-gen solution. Antivirus is dead,” Meyers said.

Article stating that next-gen is better than antivirus [art2] (click to enlarge)

Antivirus software is struggling to keep up because the primary strategy on which it relies—signature detection—is based on the outdated assumption that the malware you saw yesterday will look the same today. Generally speaking, when a cybersecurity company sees

This article wrongly assumes that signatures only detect known malware, probably because it equates signatures with pattern based detection [art3]. If this seems confusing, please read the Glossary entry for signature. (click to enlarge)

When I became interested in malware analysis, one of the very first books I read on that topic was Péter Ször's *The Art of Computer Virus Research and Defense* from 2005 [szor05]. This by now 17-year-old book describes methods that go far beyond pattern signatures and hashes. Ször mentions, among others, emulation, X-Ray scanning, in-memory scanning, algorithm based signatures, behavior blockers, network scanning [szor05]. These technologies have been refined and new ones have been developed over the years.

Artificial intelligence (AI) technologies were not mentioned by Ször, but have been in use for some years as well. They serve many purposes, including malware clustering, malware detection on client systems as well as automatic signature creation. GDATA's very own DeepRay stems from 2018 and we are certainly not the first AV manufacturer to augment their detection capabilities by using AI.

Despite all these advancements, the myth stubbornly persists to this day. I can think of two reasons why it's still sticking around.

**Firstly**, security product marketing actively reinforces it to give the impression that the advertised product is new and better. NextGen antivirus products are such an example. They claim to have new and different detection and protection technologies but they use the same ones as the antivirus products that have been around for longer. Artificial differentiation between anti malware and antivirus is trying the same by giving the (false) impression that antivirus products do not protect against malware.

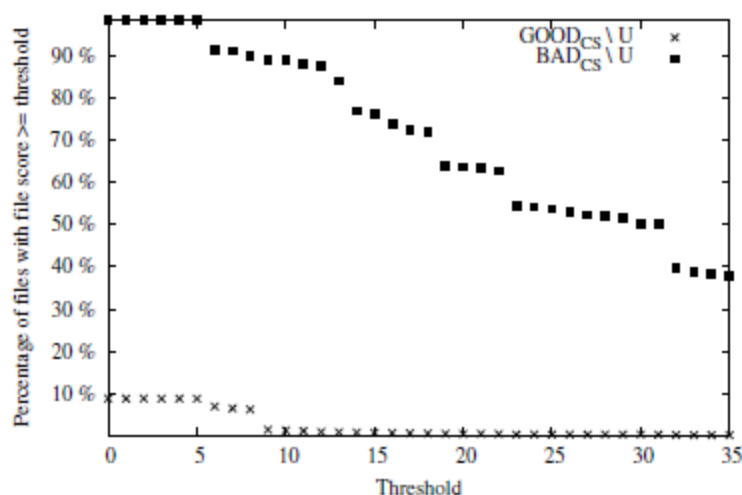
**Secondly**, people experiment with detections on VirusTotal thinking the scanning engines used there reflect the real antivirus products. If they perform small changes on a detected file and these result in lower detection rates on VirusTotal, they assume they could successfully evade antivirus products. This testing strategy is flawed because scanning engines on VirusTotal only support a fraction of the capabilities that the real products have and indeed

mostly restrict themselves to pattern scanning and blocklisting of files, thereby giving people a false sense of how the full products actually work. This is also stated on VirusTotal's website (see image below).

- VirusTotal should not be used to generate comparative metrics between different antivirus products. Antivirus engines can be sophisticated tools that **have additional detection features that may not function within the VirusTotal scanning environment**. Because of this, VirusTotal scan results aren't intended to be used for the comparison of the effectiveness of antivirus products.

## Myth 2: False postive rate of 5% is great

---



False positive (crosses) and true positive (black squares) rates for a heuristic detection based on PE file anomalies [p.107, hahn14]

While writing my master's thesis in 2014, I developed a malware detection heuristic based on file anomalies. My work was based on Ange Albertini's contributions, namely finding and documenting file anomalies, which he collected in his project Corkami. Ange, who was working as malware analyst for Avira at the time, caught wind of my work, so he visited me in Leipzig for my master's thesis defense. When I asked him, what false positive rate was acceptable for my heuristic, I was taken aback by his answer. *Zero*, he said.

Zero was a terrible answer to work with because the detection rate plummeted as soon as I tried to get the false positive rate close to zero. The rate was never actually zero. I used 49,814 clean samples and 103,275 malicious samples for my tests. The picture on the right side shows the graph of false positive (crosses) and true positive (black squares) rates [p.107, hahn14]. E.g. at 8.81% false positive rate there is a 98.47% true positive detection rate [p.108, hahn14]. That means 8.81% of the clean files were erroneously assumed to be malware, whereas 1.53% of malware could not be detected. Keeping the false positive rate as low as possible still resulted in 0.17% of clean files receiving a wrong verdict. The downside of this approach was a reduced detection rate of only 37.80% detected malware files [p.109, hahn14].

At the time I did not understand why it had to be zero. Surely, some rate of errors must be acceptable? Antivirus products are not perfect after all. Everyone has heard of and maybe even encountered false positives from antivirus products. Now that I am a malware analyst, I know the answer: Yes, false positives are something antivirus products live with, but the acceptable false positive rate is way lower than you think.

Current malware detection research papers often have the same misconception about acceptable false positive rates. A lot of them assume that the number of clean files and malware files is roughly equal aka balanced. In reality computer systems rarely see any malware but deal with clean files the majority of the time. E.g. for Windows 10 the folder *C:/Windows* contains roughly 500,000 files. If we assume only 5% false positive rate, the detection technology will determine 25,000 of those files as malicious. Unless you are ready to bet that your system still works fine after deleting 25,000 arbitrary files from *C:/Windows* this is an unacceptable number. How many files would you be ready to bet on?

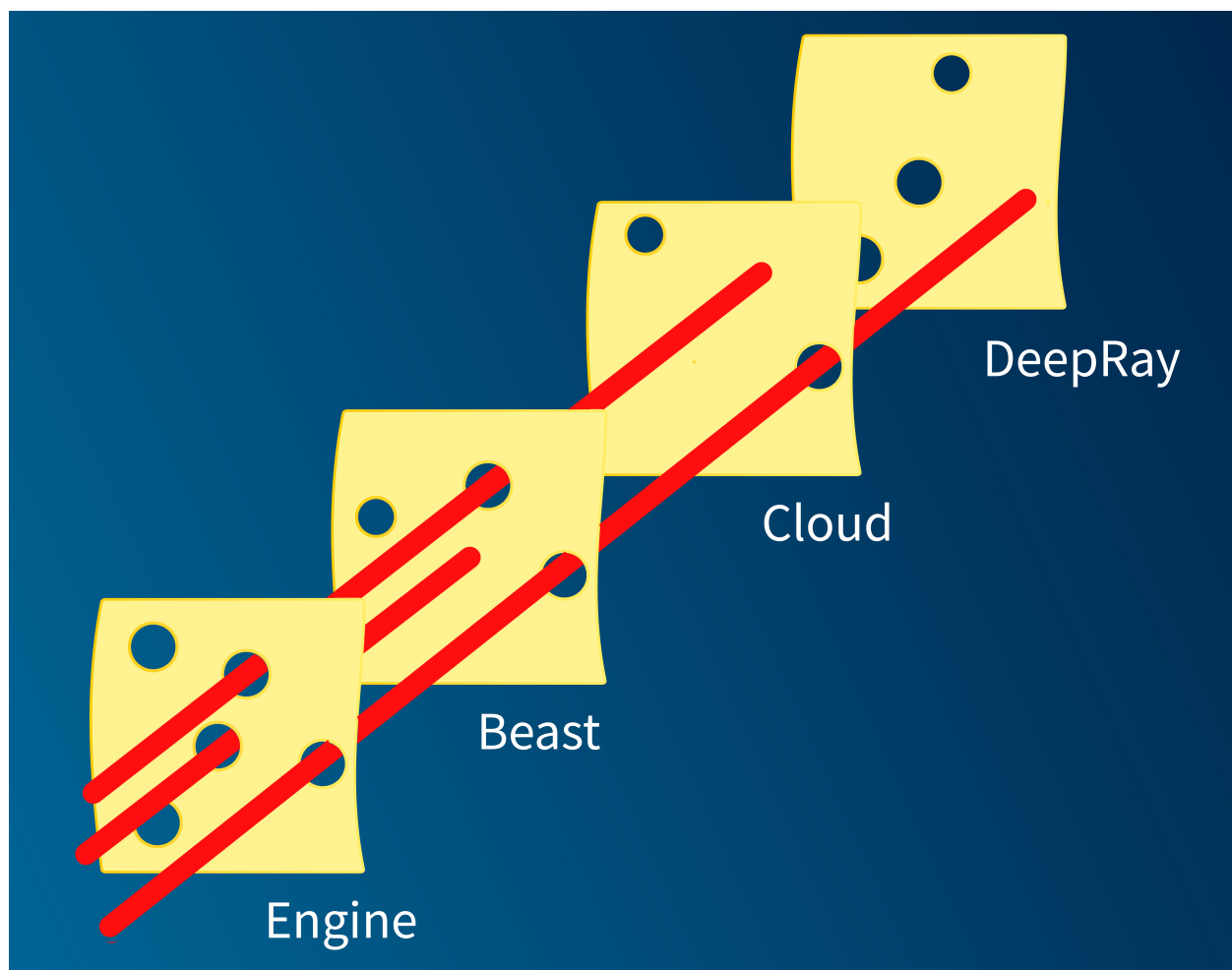
This misunderstanding of imbalanced problems has a name: **Base Rate Fallacy**. It is so common that several studies were conducted to raise awareness. Jan Brabec and Lukas Machlica concluded *"We encountered [a] non-insignificant amount of recent papers, in which improper evaluation methods are used"* [p.1, bramac18] and *"bad practices may heavily bias the results in favour of inappropriate algorithms"* [p.1, bramac18]. The study *Dos and Don'ts of Machine Learning in Computer Security* [arp20] examined common pitfalls in 30 papers published over the course of ten years. Eleven of those papers suffered from the Base Rate Fallacy [p.7, arp20].

What would be an acceptable false positive rate, though? Stefan Axelsson analyzed this question for intrusion detection. Axelsson says *"the factor limiting the performance of an intrusion detection system is not the ability to identify behavior correctly as intrusive, but rather its ability to suppress false alarms"* [p.202, axelsson00] and concludes that *"less than 1 / 100,000 per 'event' [...] will have to be reached for the intrusion detection system"* [p.202, axelsson00]. This is a false positive rate of 0.001% for intrusion detection. Solutions with a higher false positive alarm rate not only result in more work for the person tasked with mitigating the alerts, they become *The Boy Who Cried Wolf*—no one takes them seriously anymore.

A viable false positive rate must be even lower for antivirus products which respond to threats automatically. In contrast to intrusion detection systems, false positives in automatic prevention can wreck entire systems or halt a production pipeline. For research studies that use clean sample set sizes smaller than 100,000 samples, the false positive rate must be effectively zero.

## How Antivirus products do it

---



The key to effective detections is a layering of different specialized technologies. (click to enlarge)

Now that we know how low the false positive rate of an antivirus program must be to remain serviceable, it becomes clear why detection heuristics are not that simple anymore.

Many people can come up with ideas for heuristic detection, e.g., to detect ransomware, you "only" have to check for programs that rename lots of files at once and raise their entropy due to encryption. But when such heuristics are put into practice, you realize how many legitimate programs display similar behavior. Focusing on the ransomware heuristic example it is, e.g., backup programs that do the very same thing: renaming personal files en masse and raising their entropy with compression.

Antivirus products solve this by layering defense mechanisms. Technologies are stacked on top of each other to achieve the best coverage. Some of them might only detect 20% of samples because they are specific to certain types of attacks or environments, e.g., to file formats, behavior or other properties that are a prerequisite. But if a subset of these samples is otherwise not detected, the layer is worth implementing.

Everyone is probably familiar with the swiss cheese model of defense, the implications of which are of interest here.

With that model in mind it makes sense why false positives are way worse than false negatives. Undetected malware can be caught by other layers eventually. So a low detection rate for one individual layer is perfectly fine as long as other layers cover the holes. But for false positives there is no such network of layers. "But what about allowlisting?", you may ask.

While allowlisting exists, it must be seen as last resort and used with caution. This has several reasons:

**Firstly**, allowlisting can open doors for malware to evade antivirus detection. If a certificate, keyword, behavior or other characteristic of a program is used to allowlist it, it can also be abused by malware. For the same reason certain programs cannot be allowlisted, e.g., because they are part of an execution environment that is used by both, legitimate and malware files; or because they are only clean in a certain context like, for example, remote access tools, which are fine if used for actual assistance but much less so if installed silently by an attacker.

**Secondly**, legitimate programs evolve and new versions or similar programs appear daily. Therefore a single version-specific allowlist entry is not a long-lasting solution. Adding one allowlist entry to adjust a false-positive-prone detection heuristic almost certainly entails having to add others regularly, making this a maintenance-heavy endeavour.

So usually a false positive of a prevention layer is a false positive for the whole product. By contrast the false negative rate can be arbitrarily high for any such layer as long as any performance impact is justified by the newly covered malware samples.

## Takeaways

---

Malware detection heuristics seem easy on the surface but are tricky in practice. While many believe that high malware detection rates are the primary goal, a low false positive rate is the most important indicator for detection heuristic quality. The impact of false positive rates is frequently underestimated because of the base rate fallacy.

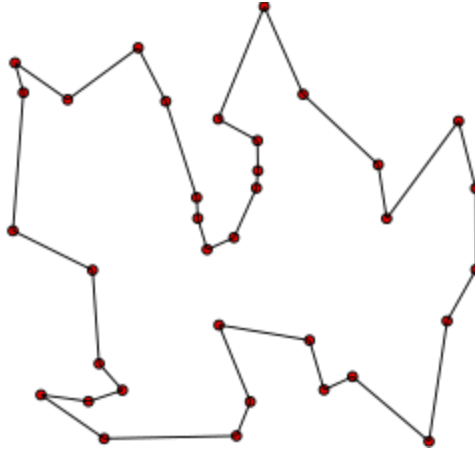
When it comes to antivirus products, false positives must be manageable and a corresponding tolerance rate definitely has to be lower than 0.001%. The false negative rate of individual technologies does not matter that much as long as the detection heuristic covers some swiss cheese holes.

I'd wish for detection technology research to specifically concentrate on samples that are hard to detect and find practical solutions for those instead of trying to create what we Germans call an "egg-laying, milk-bearing woolly sow". There will never be the one technology that can rule them all.

I also want journalists and security influencers to understand the state of antivirus technologies and actual value of detection research so that they stop reinforcing myths. The same is true for antivirus marketing where the myth of outdated detection technologies may work great to discredit competitors but has adverse affects on the antivirus industry in the long run.

## Glossary

---



A solution of a travelling salesman problem (image from Wikipedia, [click to enlarge](#))

**Anti malware:** See antivirus

**Antivirus:** Program that protects against malware, including but not limited to viruses. The term "anti malware" is more accurate but antivirus is more widely used.

**Base rate fallacy:** Neglecting differences in base rates of imbalanced problems while calculating probabilities. E.g. if 1 out of 10000 people are infected, and we calculate the probability that a person with positive test result is infected, it would be a base rate fallacy to assume that the number of infected and not-infected people is the same.

**Entropy:** A measure of how much information or how much randomness is in a set of data. Compressed data and encrypted data have high entropy. Data with repeating patterns have low entropy.

**False positive:** A clean file that is wrongly detected as malware by an antivirus product

**False negative:** A malware that is not detected by an antivirus product.

**Heuristic detection:** Heuristics in general are approximate solutions. Heuristic detection describes all malware detection methodologies that have a risk for false positives. How the detection is done does not matter. It can be based on pattern signatures, machine learning, behavior, context information ...

**Intrusion detection system:** Monitors activity on systems and networks and forwards suspicious activity, potential malware outbreaks and intrusion attempts to an event management system or administrator. These systems can tolerate higher false positive rates than antivirus products with automatic prevention response.

**Signature:** An independent unit of detection logic combined with a detection name. The detection logic can, e.g, consist of a set of *patterns*, *behaviors*, *circumstances* and a *condition* that must be satisfied to declare a piece of data as malware. Detection logic can also be an algorithm that decides when a piece of data is detected as malware. Detection signatures are deployed independently from the antivirus product code, so that they can be updated more frequently. What distinguishes an algorithmic signature from a non-signature detection is mainly how it is deployed: A hardcoded algorithm for malware detection is not a signature. Media often equates the term signature with pattern-based detection logic.

**Travelling Salesman Problem:** A problem in theoretical computer science that can be described as follows: "Given a list of cities and their distances, find the optimal, shortest route to visit each city exactly once." This problem is **NP-hard**. In layman's terms there is as of **yet** no optimal solution that always works within a reasonable time.

**True positive:** A malware that is detected by an antivirus program

**True negative:** A clean file that is not detected by an antivirus program

**Undecidable problem:** A decision problem in mathematics and theoretical computer science for which it is impossible to construct an algorithm that always provides a correct answer (yes or no). In layman's terms a perfect answering machine is not possible for these kind of problems. One example is the decision whether a file is malware.

**Virus:** Special kind of malware that self-replicates and infects files.

## Bibliography

---

[ange] Ange Albertini, *Corkami*, Google Code project,  
<https://code.google.com/archive/p/corkami/> now moved to <https://github.com/corkami>

[arp20] Daniel Arp et al, 2020, *Dos and Don'ts of Machine Learning in Computer Security*,  
[https://www.researchgate.net/publication/344757244\\_Dos\\_and\\_Don'ts\\_of\\_Machine\\_Learning\\_in\\_Computer\\_Security](https://www.researchgate.net/publication/344757244_Dos_and_Don'ts_of_Machine_Learning_in_Computer_Security)

[art1] <https://securityboulevard.com/2021/09/why-antivirus-is-not-enough-the-sandbox-is-dead-turning-the-corner-with-malware-prevention>

[art2] <https://venturebeat.com/2021/02/22/antivirus-is-dead-the-rising-enterprise-security-threats-for-2021-and-how-to-protect-against-them/>

[art3] <https://slate.com/technology/2017/02/why-you-cant-depend-on-antivirus-software-anymore.html>

[axelsson00] Stefan Axelsson, 2000, *The Base-Rate Fallacy and the Difficulty of Intrusion Detection*, <https://dl.acm.org/doi/pdf/10.1145/357830.357849>

[bramac18] Jan Brabec and Lukas Machlica, 2018, *Bad practices in evaluation methodology relevant to class-imbalanced problems*, <https://arxiv.org/abs/1812.01388>

[cohen84] Fred Cohen, 1984, *Prevention of Computer Viruses*, <https://web.eecs.umich.edu/~aparakash/eecs588/handouts/cohen-viruses.html>

[hahn14] Karsten Philipp Boris Hahn (previously Katja Hahn), 2014, *Robust Static Analysis of Portable Executable Malware*, [https://www.researchgate.net/publication/350722779\\_Robust\\_Static\\_Analysis\\_of\\_Portable\\_Executable\\_Malware](https://www.researchgate.net/publication/350722779_Robust_Static_Analysis_of_Portable_Executable_Malware)

[szor05] Péter Ször, February 2005, *The Art of Computer Virus Research and Defense*, Addison Wesley Professional

© 2025 G DATA CyberDefense AG. All rights reserved.