Brought to you by:

CYLANCE

Next-Generation Anti-Malware Testing

for dummies®
A Wiley Brand

Learn to combat modern malware

Do your own anti-malware testing

Turn test results into a protection plan

Cylance® Special Edition

Chad Skipper

Carl Gottlieb

Lawrence C. Miller

Note to the Reader

We thank you for your interest in this book and sincerely hope it will start you down the path of testing for yourself. What we describe here is not the “ be all and end all” of testing. It’s a starting point. We invite you to share your results, share your testing methods, and share your stories with us. Here’s how to contact us:

Chad Skipper via Twitter @chadskipper or on LinkedIn.

Carl Gottlieb via Twitter @CarlGottlieb or on LinkedIn

Test for Yourself. *Res Ipsa Loquitur!* (“The thing speaks for itself!”)

Next-Generation Anti-Malware Testing

Cylance Special Edition

by Chad Skipper, Carl Gottlieb, and Lawrence C. Miller

for dummies®
A Wiley Brand

Next-Generation Anti-Malware Testing For Dummies®, Cylance Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub. For information about licensing the *For Dummies* brand for products or services, contact BrandedRights&Licenses@Wiley.com.

ISBN 978-1-119-45062-7 (pbk), ISBN 978-1-119-45064-1 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

We're proud of this book and of the people who worked on it. Some of the people who helped bring this book to market include the following:

Project Editor: Martin V. Minner

Editorial Manager: Rev Mengle

Senior Acquisitions Editor: Amy Fandrei

Business Development Representative: Karen Hattan

Production Editor: Tamilmani Varadharaj

Images: Drew Hoffman

Cylance Reviewers: Stuart McClure, Malcolm Harkins, and Scott Marcks

Table of Contents

Foreword

Walk into the exhibit hall at any major security conference and it's easy to understand why managers and technology people get excited by what they see. With statistics showing an average of 300,000 to 1 million malware samples created on a daily basis, it's understandable why corporations are looking to next-generation endpoint security solutions to help solve their security challenges.

Let's face it, the endpoint protection space is a marketing team's dream. Who wouldn't want to purchase a solution with cool terminology like "machine learning," "artificial intelligence," "behavioral detection," and the ability to stop "advanced persistent threats (APTs)"? Sounds like something that can fix all your security problems and finally let you sleep at night, right? However, this space can be overwhelming. Knowing where to start can be a real point of confusion because many of us have a "set it and forget it" mentality when it comes to anti-malware products.

When I started my own personal journey back in November 2016, I wanted to believe all the marketing, hype, statistics, and demos, but the neurons in my brain said I had to do my due diligence and take these new shiny cars for a test drive to check out their top speed claims. We do it for everything else in life, so why should endpoint security software be any different? Maybe because it sounds difficult to do, but it isn't!

I remember when I asked a vendor what their recommended system resource specification was for their agent. Their answer was "We run on anything," so of course I deliberately ran it on an old operating system with one processor and 1 GB of memory. Needless to say, nothing worked and it was painfully slow.

Another vendor said, "We catch all APTs." I thought, "Cool!" When I asked them about their memory-based detection, they said "We haven't implemented that capability yet." After mutating a malware sample set twice and retesting it over a three-week period, I saw a repeat of the same malware executing with zero detections — let alone any type of prevention. When I asked for details about their machine learning, I was told it would be updated in the next agent revision.

These are some of the reasons I tested everything for myself. It isn't that I didn't want to believe, but that I wanted to speak with 100 percent conviction about my findings — rather than basing my decision on marketing slogans.

The unique style of testing, determination, and persistence with bulk and targeted scenarios received recognition at this year's Black Hat conference in Las Vegas, where a collegue and I had the honor of presenting this work. The title of the presentation was "Lies and Damn Lies: Getting Past the Hype of Endpoint Security Solutions." We didn't simply test 10 to 12 samples and close the test out. We used different virtualization platforms and different user profiles, simulated remote non-connected users and no existence of cloud, turned layers off, and mutated the samples multiple times with the objective of replicating what happens in real commercial companies. The solution policy settings were never modified.

Here's a link to all our testing data: `http://pinktangent.me/publications`.

Passion, problem solving, and natural-born curiosity are why those of us in technology love doing what we do. Testing endpoint solutions is a natural extension of these skills. We owe it to ourselves to be confident in our decision-making process when determining which of these solutions will best fit our organizational business policies and values.

There is no substitute for doing your own testing. Yes, listen to your vendor. Many have amazing researchers, but verify and test yourself. You know your environment better than anyone else, and it's important that you trust your own judgment. Some may question your approach along the way, but there is no right or wrong way to test. After all, that's why it's called *testing.*

Lidia Giuliano
Independent Researcher

Introduction

How did you choose your anti-malware solution? Did you put it through the same rigorous process as your other security solutions? Or, did you simply renew your current product licensing? Perhaps you went with something you had used at a previous job. Maybe you even went so far as to read a few product reviews and third-party test results or evaluations. In other words, did you test for yourself? Did you test the anti-malware solution in your lab?

Anti-malware protection just hasn't been very exciting — until now. In this book, we explain how artificial intelligence (AI) and machine learning (ML) can help your enterprise combat malware threats in a more preventative, proactive, and radically better way than with legacy anti-malware products. We explain why you need to not take someone else's word for it (including ours and your vendor's). Instead, you need to test different solutions for yourself, just as you would with any other major security investment.

About This Book

Next-Generation Anti-Malware Testing For Dummies, Cylance Special Edition, consists of six short chapters that explore the following:

- Why legacy anti-malware techniques are limited, and how artificial intelligence and machine learning combat modern malware more effectively (Chapter 1)
- Why you should test for yourself (Chapter 2)
- How to set up your own anti-malware testing environment (Chapter 3)
- How to safely obtain malware samples and test anti-malware products yourself (Chapter 4)
- How to take action on your anti-malware testing results (Chapter 5)
- What to consider when choosing an anti-malware solution for your organization (Chapter 6)

Foolish Assumptions

In this book, we assume that you are an IT manager or security administrator responsible for server and endpoint security in your organization. Thus, we assume you are a somewhat technical reader with some knowledge of security issues, specifically malware and common malware detection methods, as well as popular anti-malware products.

If none of these assumptions describe you, keep reading anyway. We won't get too technical, and when you finish the book, you'll know a few things about malware and anti-malware testing.

Icons Used in This Book

Throughout this book, we occasionally use special icons to call attention to important information. Here's what to expect:

REMEMBER

This icon points out information you should commit to your nonvolatile memory, your gray matter, or your noggin — along with anniversaries and birthdays.

TECHNICAL STUFF

You won't find a map of the human genome here, but if you seek to attain the seventh level of NERD-vana, perk up! This icon explains the jargon beneath the jargon.

TIP

Tips are appreciated, never expected — and we sure hope you'll appreciate these tips. This icon points out useful nuggets of information.

WARNING

These alerts point out the stuff your mother warned you about (well, probably not), but they do offer practical advice to help you avoid potentially costly or frustrating mistakes.

Beyond the Book

There's only so much we can cover in a short book, so if you find yourself at the end, thinking "Where can I learn more?" just go to `www.cylance.com/tfy`.

IN THIS CHAPTER

- **Looking at "old world" anti-malware approaches**
- **Leveraging AI and machine learning in "new world" solutions**

Chapter 1
Treating Malware as a Data Problem

In this chapter, you learn about legacy anti-malware techniques and their limitations, as well as advances in anti-malware protection utilizing artificial intelligence (AI) and machine learning (ML).

Recognizing the Limitations of Legacy Anti-Malware Approaches

For decades, the entire anti-malware industry has been built on a reactionary model: A system, also referred to as "patient-zero," must be infected with malware before it can be detected and prevented. Thus, protection requires a "sacrificial lamb," or first victim. Even the most advanced techniques of signature-based detection, exploit prevention, whitelisting, application controls, and endpoint detection and response all fall into this "sacrificial lamb" reactionary model (to learn more, see www.cylance.com/en_us/blog/no-more-sacrificial-lambs.html).

Today, the reactive approach is the anti-malware industry's greatest weakness.

The model of reacting to what has already been seen, experienced, or known is limiting. Because of all the known "unknowns," many organizations, both large and small, have acquiesced when it comes to trying to prevent malware infections. The perception is that there are simply too many new techniques and variants of malware, so people relegate themselves to lean heavily on a response-only mode. They pour precious time and resources into building the fastest response team possible. Remediation is the success measure of the "It's not if, but when" mentality.

In many ways, the legacy anti-malware industry has encouraged this way of thinking. For all the "new" software and solutions built over the decades, no unique, new concepts or ways of thinking have entered the marketplace — until recently (more on that later in this chapter).

The problem is that the vast amount of malware being released in the wild today is drowning the legacy anti-malware industry and its reactive nature (see Figure 1-1). You may recall seeing presentations by legacy anti-malware vendors that detail their response timelines, boasting about their ability to provide a signature within 12 hours of a new infection. But 12 hours is an eternity in today's threat landscape. Remember, SQL Slammer? In 2003, SQL Slammer infected 75,000 victims in just 10 minutes! And that was more than 14 years ago — the Stone Age of technology (we didn't even have iPhones back then). Human creation of new signature files simply can't keep up with today's explosion of malware threats.

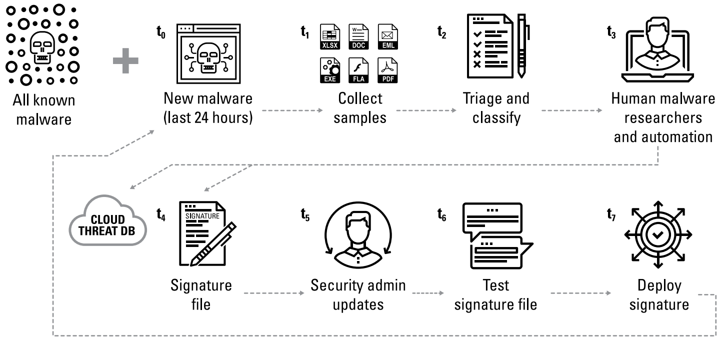


FIGURE 1-1: Legacy signature-based anti-malware is reactive and drowning in a sea of malware.

Thus, security is now a data problem. Can artificial intelligence and machine learning help with this data problem? There's a test for that (see Chapter 4). Can algorithmic science reliably predict and prevent known and unknown malware from executing? There's a test for that (see Chapter 4). Can there be a predictive advantage with AI/ML whereby it can prevent tomorrow's malware yesterday? Yes, there's a test for that (see Chapter 4). Numerous reports suggest that nearly 700,000 new malicious programs are created every day. That fact, in and of itself, is a data problem that can no longer be reactively addressed by humans. AI/ML is the only viable way to combat malware threats today and in the future.

Signature Updates: Teaching an Old Dog "New" Tricks

Signature-based anti-malware products are the most prevalent type of anti-malware products on the market today. Yet, despite their popularity, signature-based products have several inherent weaknesses that severely limit their effectiveness.

First, as discussed earlier, the signature-based model is reactive in nature: It requires a "sacrificial lamb" to first be infected with a new zero-day malware threat before a signature can be developed to detect it. Of course, in today's hyperconnected world, it's rarely a single lamb — more like an entire flock (or enterprise network). Which may be okay — so long as the sacrificial lamb isn't ewe.

Second, a new anti-malware signature must be effective at detecting the new malware. That's a tall order given the number of malware variants that can easily be rapidly and automatically generated, and the intense pressure on anti-malware vendors to develop accurate signatures in a timely manner and deliver them on a nearly continuous basis.

Then, the signature files must be reliably delivered to your endpoints over the network. Lots of things can go wrong. An endpoint might simply be powered down for a week or so while the user is on vacation. Your network connection might be slow or down. Or your anti-malware vendor could have network issues of its own. Stranger things have happened.

Finally, what if all else works as it's supposed to? Say a zero-day threat is quickly discovered and its initial impact is limited (only a handful of someone else's lambs are led to the slaughter). Your anti-malware vendor's developers have had their recommended daily allowance of hyper-caffeinated energy drinks, and their signatures are "spot on." Your network is humming "zippity doo dah," so all your endpoints get updated before the new malware ever has a chance to ruin your day. Even if all those things are working in your favor, something else can, and eventually will, go wrong.

REMEMBER

Relying on continuous updates to protect your endpoints from malware is like installing a web browser that requires constant patching to find new websites on the Internet!

REMEMBER

Establishing persistence (typically with malware) on a victim's endpoint is a key objective for adversaries.

The problem is that legacy signature-based anti-malware products rely on continuous and effective updates — delivered over the Internet — to work. Take away the updates and/or Internet access, and these products just fall apart.

TIP

Read Chapter 3 to learn how to set up a testing environment, and read Chapter 4 to learn how to perform various anti-malware effectiveness tests.

Anti-malware products should provide complete protection, even when they are offline, without requiring a continuous online connection and updates to be effective. Then, when a major feature release is delivered, software vendors and end users can take the time to fully QA the release before it gets deployed.

Exploring Artificial Intelligence and Machine Learning

The theory of predictive advantage is the ability of an anti-malware solution to prevent tomorrow's malware with today's machine learning model. This means having the ability to prevent future malware infections — by malware that doesn't yet exist — from executing on endpoints with today's artificial intelligence model. Can your current anti-malware product predict and prevent future malware from executing today? It's hard to fathom, but until now the attacker has always had the advantage because the available security technologies were all reactive to new malware threats — which meant succumbing to the "sacrificial lamb" model.

Artificial intelligence and its mathematical subset, machine learning, are radically changing the "old world" mode of cybersecurity. With new industry priorities and greater demand for security, what's needed is more than just another tool, technology, solution, layers, or perceived best practice. A radical new way of thinking is needed to redefine the security industry.

That's where proactive, predictive, and preventative protection through machine learning comes into play. Machine learning uses algorithms to build models that uncover patterns and continually refine them with its learning capabilities. By using machine learning, organizations can make better decisions at a speed and scale that surpass human capabilities. This capability comes from being able to predict based on experiences from the past.

Part of the new proactive method includes focusing on the endpoint. Rather than adding more layers of reactionary technology to the network and endpoint, the focus instead should be on the new perimeter of the modern era — the user. Where is the user? At the endpoint.

Machine learning offers a comprehensive, granular approach to malware prevention at the equivalent to the DNA-level of code. With human DNA, you have a complex set of instructions which, in blocks (genes), interact with other blocks to create patterns for building a living organism. Machine learning can analyze similarly interrelating blocks of code and file characteristics at a rate and volume that manual analysis by humans cannot begin to match, which means the performance of systems and workflows is not adversely affected.

One of the marvels of machine learning is that, unlike human analysis, once malware is deconstructed, views of statistically similar blocks of code can be analyzed to identify the presence of malicious code ("bad genes") without even having to execute the file first. AI can determine malicious files through observation, pattern recognition, and predictive analytics. With AI, existing and never-before-seen malware threats can be — and are — prevented.

To achieve this level of success, machine learning algorithms can be placed on an endpoint to conduct pre-execution static analysis. This technique can quickly determine if a file is malicious or benign. As opposed to relying on cloud-based analysis techniques, the endpoint can venture off-network and benefit from the same level of protection because the algorithm runs on the endpoint. Unlike signature-based techniques, which require network connectivity to obtain frequent updates, machine learning algorithms can run off-network for months at a time and be more effective than signature-based products that are fully up to date.

Although the concept of AI sounds more like science fiction than science, the precepts are powerfully simple. Machine learning breaks down into four phases (see Figure 1-2):

- **Collection:** The first step in machine learning involves collecting as much data as possible. Effective machine learning requires vast amounts of data. Virtually hundreds of millions of good and bad files are compiled from multiple sources, which include live data feeds, government databases, and proprietary repositories, as well as research and scientific surveys that are open source. The ability to gather and store data in the cloud, as well as extract data from mobile, Internet of Things (IoT) devices, and embedded systems, facilitates taking data collection to new heights that were not possible even a few years ago. Gleaning data from all these sources ensures a relevant sample size that represents the broadest range of file types and authors. To train a model, it must be known which files are good and which are bad.
- **Extraction:** The feature extraction process deconstructs a single file into a variety of characteristics that number in the millions. Each characteristic is analyzed against millions of characteristics derived from other files. Thus, millions of records gathered during the collection phase are then individually deconstructed into millions of variables that are transformed into vectors.

  Extraction plays a key role in applying science and engineering capabilities to the process, then scales them by a factor of millions. In the end, it yields the observations that train the AI, and it is through those learned patterns that AI can determine if a new file is benign or malicious.

- **Classification:** Building better statistical models allows for highly-tuned classification and clustering. In the end, machine learning relies on precise content categorization. In the case of endpoint protection, the categories are malicious or benign. In addition, classification includes organizing files based on what a file is intended to do — for example, whether a file is intended to perform as a key logger or Trojan.

  AI can detect subtle statistical connections that, to a human, may appear innocuous or go unnoticed altogether. The analysis takes milliseconds and is extremely precise because of the breadth of the files and file characteristics analyzed.

  The analysis provides a confidence score as part of the classification process. The score gives additional insight that can be used to weigh decisions around a single file — such as whether to block, quarantine, monitor, or analyze it further.

- **Learning:** After data is collected and features are extracted from each file, the millions of attributes are ready for the learning process. The attributes are converted to numerical values in the form of vectors, which are used in model training. Dozens of models are created with measurements to ensure the accuracy of prediction, and the testing process itself helps identify ineffective models. Hundreds of millions of files are used to test and validate models. Tested, refined, and ready for action, the final models are loaded for use.

  As the files and file attributes go through the learning process, the models develop an understanding of the intention of a sample, which can be used in a predictive fashion to determine the potential risk a new file may pose without having to execute the file itself.

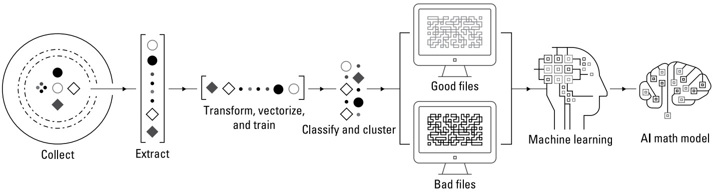


FIGURE 1-2: The machine learning process.

With machine learning, models are built to determine if a file is malicious or benign, and whether it should be identified as suspicious based on the confidence score. Through the identification of known malicious and benign files, organizations benefit from the powerful capabilities of AI while retaining the ability to isolate the few outliers that may require some manual analysis.

In this approach, the enterprise gets the best of both worlds — powerful technology that automates, accelerates, and dramatically improves processes, with the option for integrated human expertise when desired. The advantage is that your security staff can move away from responding to thousands of alerts that range in severity, and concentrate on attending the few activities that require their expertise. In addition, you remove from the process the possible bias that affects legacy anti-malware methodologies. In other words, let computers do what computers do best, and let humans do what humans do best.

TIP

For more information, please see *Introduction to Artificial Intelligence for Security Professionals* (`https://pages.cylance.com/2017-07-25CNTIntrotoAIBook_LP-Download.html`).

Machine learning is itself not a new technique, but its quality, maturity, and implementation vary greatly within the anti-malware industry. Anti-malware vendors generally use machine learning in one of two ways:

- To help automate and augment humans in the creation of heuristics and signatures of specific, previously seen malware for future detection by the product.
- To create a predictive mathematical model that will allow the product to prevent execution of known and future malware.

These two implementations are polar opposites, with the former simply speeding up existing human leg work in creating signature detections after the fact, and the latter creating a paradigm shift for creating malware prevention intelligence before the malware has even been created. Very few anti-malware vendors use the latter technique, and when comparing prevention rates in your lab (see Chapter 4), you'll see for yourself the huge difference this approach makes.

IN THIS CHAPTER

- » Recognizing the limitations of third-party testing
- » Trusting your own testing results

Chapter 2 Understanding Why You Should Test for Yourself

Your choice of anti-malware software for your organization is too important a decision to be based solely on your experience with a given vendor, customer testimonials (they'll only share the good ones with you anyway!), product reviews, or industry testing reports. In this chapter, our goal is to get you to attest to the virtues of testing anti-malware products for yourself.

Third-Party Malware Testing

Let's start by pointing out that there are some reputable testing houses out there. In these cases, you can find examples of positivity and growth, usually coupled with the willingness to change and work with new security technologies. However, there are also plenty of testing houses that are unduly motivated or influenced by factors that can negatively affect the objectivity of their testing.

This unfortunate reality is true not only of anti-malware testing but of practically any product testing, whether hardware or software and technology or non-technology related. How many movies have you walked out of thinking "Wow, the critics got that one wrong — two thumbs down!" And have you ever wondered who the nine out of ten dentists are who recommend your brand of toothpaste — and more importantly why the one dentist didn't recommend it?

TIP

Cylance is a member of the Anti-Malware Standards Testing Organization (AMTSO). AMTSO members consist of vendors and testers working together to create objective and effective industry testing standards. These standards aim to provide testing protocols and behavior expectations for testers and vendors to follow when testing anti-malware solutions. You can learn more about AMTSO at `http://amtso.org`.

To remedy the status quo in the anti-malware testing industry, testing organizations must address some of the ever-changing testing of threat vectors, including the following:

- **Define a curation strategy for the malware repository:** Define the process used to collect the malware used in testing and how the malware is maintained over time.
- **Explain the conviction process of malware:** Explain the steps used in providing evidence on how the malware was convicted or charged as being malware.
- **Enable malware creation for testing purposes:** Allow any testing organization to create its own malware, also known as *zero-days,* to be used in testing.
- **Obfuscate and/or modify malware:** Change the malware from its original form for the purposes of changing the underlying properties of the malware.
- **Test in both offline and online states:** Test the anti-malware architecture and the online dependency of its solutions.

WARNING

Using a malware list to which vendors and testers can contribute (a common practice in the anti-malware testing industry) is simply hygiene testing. This method does not detect and defeat prevalent and unknown malware because everyone already knows about the malware in question.

Learning a Lesson About Testing

The genesis of this book started in November of 2014 when one of the authors, Chad Skipper, began looking for a new anti-malware security vendor that his employer, Dell, could partner with. Chad was in the office of the CTO, and the team's job was to set and incubate the security strategy for the millions of endpoints Dell delivered to its commercial customers each year. This particular project was to find a partner that could prevent known and unknown malware from executing on the endpoint. Chad set a high bar: The vendor chosen would not subscribe to the "sacrificial lamb" theorem described in Chapter 1.

He began to scour the Internet and found that more than 60 vendors claimed the ability to predict and prevent unknown, zero-day malware. The challenge was that neither analyst reports nor third-party testing proved this. Chad could not go to his peers and superiors and recommend a product for partnering based upon nothing more than analyst reports and third-party testing results. He needed to test for himself.

Chad still remembers the day his team tested. He and his colleague were in awe of what artificial intelligence (AI) and machine learning (ML) could do. They had seen nothing like it in their careers of more than 20 years. They had 460 zero-day pieces of malware and thousands of known pieces of malware. The Cylance AI solution prevented all but 12, and the traditional solutions missed all the zero-days and hundreds to thousands of pieces of known malware.

This is why we share this story and why Chad is at Cylance today. Nothing better shows the results of predictive advantage than testing for yourself. This book details some of the testing methods Chad and his colleague used throughout their several rounds of testing. In the end, they had a scientific methodology with real results that clearly showed the differences across all the anti-malware technologies they tested.

REMEMBER

We encourage you to *test for yourself.* Res Ipsa Loquitur! ("The thing speaks for itself!")

Testing for Yourself

How do you go about choosing the best anti-malware protection for your organization? You should test for yourself!

You shouldn't dismiss public test results outright, but they should be taken with a grain of salt. Don't blindly accept public test results or arbitrarily dismiss a product that isn't recommended by one of the testing houses. It's possible — even likely — that the vendor simply didn't "pay to play."

Instead, you should select your anti-malware protection as you would any other technology. Talk with different vendors, peers, and customer organizations, but most importantly, evaluate your options for yourself. Do a "proof of concept" for anti-malware protection by testing the different products for yourself under a variety of real-world conditions.

It's also important to recognize that there is no such thing as a 100 percent efficacy rate in security. There is no single silver bullet that will provide total, unbreachable protection against every type of malware in every situation. But there is such a thing as *predictive advantage*. Predictive advantage is the ability to detect tomorrow's malware today. It's the ability to detect unknown, yet-to-be seen, and yet-to-be discovered malware that is lurking all over the Internet (see Figure 2-1).

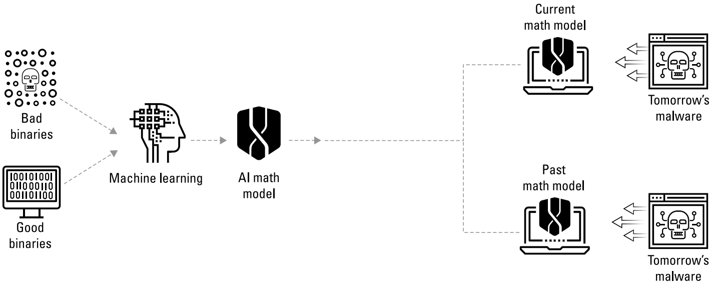


FIGURE 2-1: The predictive advantage theorem protects the "sacrificial lamb" (or "patient-zero").

TIP

In a recent CarbonView survey, 62 percent of respondents said they conduct their own endpoint anti-malware testing. Among those who don't do their own testing, 76 percent said they would like to do so in the future. Across all respondents, "not enough resources" and "lack of expertise" are seen as the biggest barriers to testing. We hope this book helps with the "lack of expertise."

The bottom line is: Don't trust a vendor or third-party testing report at its word. Test for yourself. Trust yourself. Don't base your buying decision entirely on analyst and testing reports. Get the product into *your* lab. The only true test is the one that replicates your real-world production environment.

IN THIS CHAPTER

- Using virtual machines to test
- Isolating your test environment
- Obtaining malware

Chapter 3

Setting Up an Anti-Malware Testing Environment

Testing anti-malware products can be performed in a safe and secure manner if the tester follows best practices. Testing in a virtual machine (VM) that is isolated from the host device, as well as isolated from the production network, ensures that a security analyst can execute malware safely and in a manner that yields the most accurate test results.

In this chapter, we explain best practices for setting up your testing environment.

Configuring the Hosts

Testing in a virtual environment is the single biggest advantage of testbed virtualization. A security analyst can quickly snapshot a VM and conduct tests, confident that any changes made can be easily reverted by restoring to the VM state captured in the snapshot.

WARNING

Some malware is designed to detect virtual machines and therefore might not execute correctly.

In practice, accurate results are a product of re-creating production environments as accurately as possible. It follows that accurate reproduction of a production environment warrants the accurate reproduction of attacks against that environment. This outcome can be achieved with software that virtualizes a physical machine. Alternatively, you can create a fresh VM with a base operating system (OS) image.

TIP

If a VM is created from an OS image, be sure to install all software that is in the base image to best mirror the production environment.

The VM should next be updated to include all recent security patches. Some operating systems have built-in security features, which may interfere with the results of test output data and should therefore be disabled. Be sure to remove or disable any other anti-malware product that you do not want to evaluate. Only one product at a time should be tested. This step is critical in order to demonstrate the true ability of the product being tested.

Once the virtual environment has been established, install the anti-malware product of your choice for testing and ensure it has been configured and updated with the policies you intend to run within your environment. Check to ensure the anti-malware product is up to date and running the policy of choice. Virtualization software, such as VMware, should also be updated to the most recent version.

WARNING

Isolating the VM from the host device is crucial to ensure that if the malware infection is undetected, it will be fully contained. Device isolation is accomplished via the VM's settings and network configuration. Always check the VM settings to make sure any shared folders are set up with read-only (RO) permission. Also verify that drag-and-drop and/or copy-and-paste features are disabled.

Building the Network

Typically, virtualization software provides three types of networking interfaces:

- **Network address translation (NAT):** A NAT interface allows access to the physical network by sharing the address of the hosting machine. This setup provides access to the Internet through the physical infrastructure. Typically, your physical device is assigned the address of `X.X.X.1` on the NAT network. This puts the physical device and your VM on the same virtual network through virtual interfaces on the physical device. Thus, the VM can only communicate out of the virtual network. The hosting device is performing NAT, and the external devices to the virtual network have no routing information back to the VM. This network configuration is best used to allow VMs access to the physical network while limiting inbound network traffic.
- **Host-only or custom network:** A host-only or custom network interface allows communications between all devices located on that network segment. It is important that these devices should not have access to the Internet or physical network. To better isolate the VM, the physical device's virtual interface can be removed from these network segments. This configuration is best used to set up a virtual network that will be isolated from the physical network.
- **Bridged:** A bridged interface will place the VM's virtual network directly in the physical network. This configuration allows all devices on the physical network to communicate with the VM. There are almost no circumstances where this configuration should be used while testing an anti-malware product or any malware.

REMEMBER

Always test on a network that is separated from production. Host-only or custom network interfaces are established if the hosting device is segmented from the production network and no Internet connectivity is necessary. Configure a NAT interface if Internet connectivity is required for testing.

For more complex and secure setups, you can use a hybrid of the two interfaces. For example, the testing virtual machine and an OS firewall VM, such as PfSense, could be placed on a host-only or custom network segment. Once this setup has been established, configure the firewall VM to have an additional interface attached to the NAT network. The firewall VM should be configured to route anti-malware test VM traffic from the host-only or custom network through to the NAT network. Strict firewall rules should be in place to ensure that traffic does not communicate with any of your physical devices (see Figure 3-1).

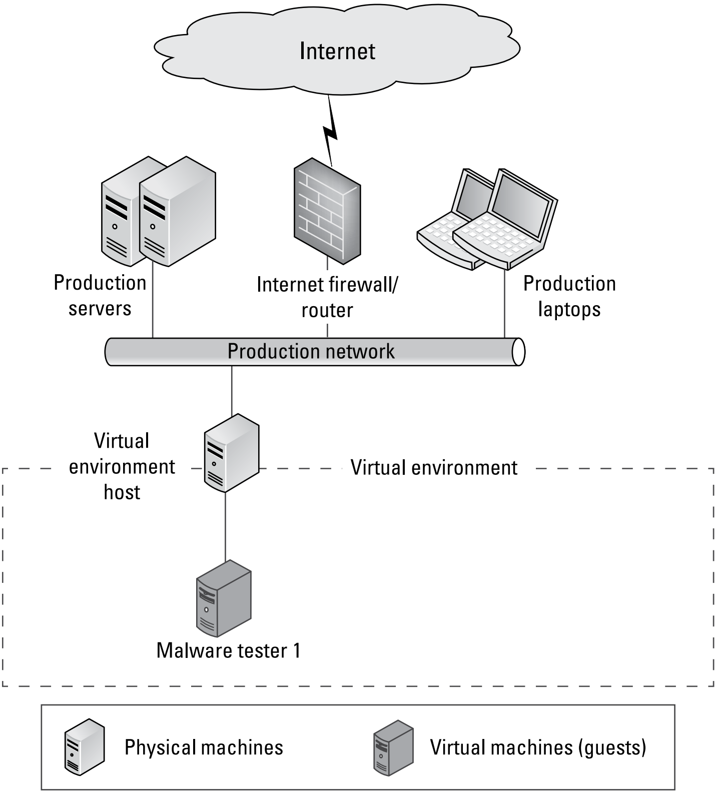


FIGURE 3-1: A simple testing lab setup.

A snapshot of the VM should be taken once all available updates have been applied and all recommended VM configuration, up to this point, has been completed. It is now safe to introduce malware to the VM.

REMEMBER

Accurate test results allow security professionals to properly vet available solutions for their networks and devices, and it is therefore imperative that these tests reflect the environment that the tested products will eventually protect. Therefore, the test environment must accurately mimic your organization's production environments.

Getting the Malware

Now, where can you get malware? There are many legitimate sources for obtaining malware samples. If you don't have a "zoo" (your own library of malware), then visit `https://testmyav.com`. TestMyAV is a website with a single purpose — to enable people to test anti-malware solutions for themselves. Rather than trusting vendors, testing companies, and salespeople at their word, TestMyAV knows that testing isn't hard and believes that everyone should have the ability to evaluate which solutions are best for their organization.

Your malware samples should include different malware types like portable executables (Pes), compressed files, Visual Basic scripts, javascript, and browser-based exploits, among others.

Malware can be introduced onto an endpoint via a number of different channels. This can happen through email, files downloaded from the Internet, infected USB drives, scripts, or Powershell, or as malicious files stored in shared folders. Always zip and password-protect malware while transferring malicious files between devices.

SAFELY HANDLING MALWARE

Let's state the obvious: The *mal* in malware is an abbreviation for "malicious." I'll spare you the dictionary definition, but the short version of the story is that it is out to hurt your machine. It therefore stands to reason that you should be extremely careful while handling malware. Here are a few measures you can take:

- **Always keep files zipped and password-protected when moving between machines. The industry-standard password is "infected".** This helps ensure that the malicious files are not accidentally executed. All malware samples on the `testmyav.com` website use the default password "testmyav".
- **Never send malware samples via email.** Email provides opportunities for samples to be released to unintended parties. There is also a risk that your intention to share testing resources will be

(continued)

(continued)

construed as an attempt to infect the recipient. Organizations typically deploy anti-malware measures on mail servers, so this practice could get you flagged. It's better to share samples via repositories or carefully secured USB drives.

- **Keep a working directory and a storage directory in your test environment.** This ensures that you are being intentional about the malicious files you are testing. To maintain hygiene, follow these rules:
  - Move malware you intend to test to your working directory.
  - Only detonate malware from your working directory.
  - Always move malware you do not intend to test back to storage.
  - Consider removing or altering file extensions (see below).
- **Remove file extensions or add an invalid file extension to malicious files.** In Windows Explorer, be sure to have file extensions visible. You can simply delete file extensions by highlighting them. This method, however, is pretty unwieldy because you might be dealing with hundreds or possibly thousands of files. To do so in the command window, follow the steps provided at `testmyav.com`.
- **Work in an AV excluded directory if you operate malware on your host.** Do not operate malware unless you're in a virtualized environment. Be sure to exclude the directory in which the malware resides if you're going to do so. Be very, very careful if you do this.
- **Remove executable rights from the directory you use to store malware.** This provides an extra layer of protection in that you cannot accidentally detonate stored malware. This technique is especially helpful if you decide against changing the file extensions.

TIP

As a security professional, learning to mutate malware allows you to better vet endpoint protection solutions because you can create unique malware — from a hash and signature perspective — for your tests.

IN THIS CHAPTER

- **Emulating random malware mutations**
- **Testing effectiveness with and without the cloud**
- **Simulating a zero-day threat**
- **Going fileless with malware testing**

Chapter 4

Exploring Anti-Malware Testing Methodologies

The simple objective in testing an anti-malware product is to verify that it stops execution of malware on the endpoint. Testing in this case is not about features and functions, it's about preventing the malware from executing. That's what an anti-malware product is designed to do — stop malware — and these tests are designed to measure their pre-execution stopping power. In this chapter, you learn about four different testing methodologies for portable executables (PEs) and fileless malware.

Random Mutation

Mutating malware is the process of changing existing malicious software without altering its functionality. This is often performed to change a piece of malware's hash (also known as the *message digest*). Mutation allows malware to evade signature-based anti-malware solutions, which typically rely heavily upon a collection of hashes to identify malware threats.

This random mutation test is designed to emulate real-world conditions. In the real world, adversaries pack their malware to evade signature-based detection. A *packer* is software that takes the original malware file and compresses it, which makes the original code and data unreadable, and more importantly changes the hash of the file. The packed program can still execute despite compression. By doing this, the attacker evades signature-based hash detection.

Figure 4-1 illustrates why signature-based anti-malware products fail to detect malware in the real world, as demonstrated by random mutation testing. In panel A, legacy anti-virus software has a detection signature for known malware and can therefore detect the malware. However, the same "known" malware can be easily mutated (as demonstrated in the steps that follow). In panel B, the mutated version becomes "unknown" malware, thereby bypassing the anti-virus signature and infecting the endpoint.

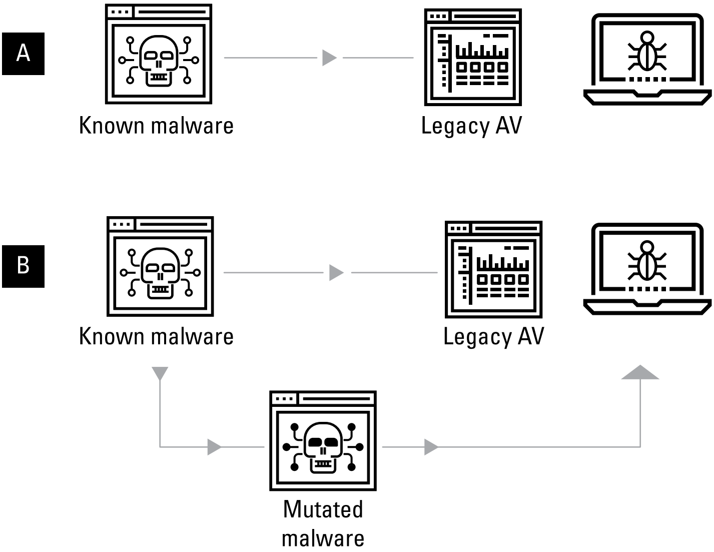


FIGURE 4-1: Why signature-based anti-malware products fail random mutation testing.

Run your initial test with the packed malware and record your results. Wait a week, update the anti-malware signatures and run the same packed malware against these updated signature files. You'll be shocked by the results!

TIP

Go to testmyav.com for a list of packer websites.

The following example uses AegisCrypter (www.aegiscrypter.com) to pack malware:

1. **Create a fully patched Windows 10 virtual image and snapshot it with the name of the packer you will be using.** This snapshot will be used as a workbench for creating your packed malware.
2. **Go to www.aegiscrypter.com to download and install the packer.** Note that some anti-malware products have a generic signature for packers. It's best to use multiple packers in testing. Disable any anti-malware products on the packer image, including Windows Defender. Save the snapshot.

   TIP

   Consider mutating your malware sample two or three times using various packers. Always test the packed file on a victim machine to verify the functionality of the malware sample. Some packers don't play together nicely when you layer them and can neutralize the malware sample entirely.
3. **Go to testmyav.com to download the latest samples of malware and save the malware on your packer image.**
4. **Run AegisCrypter.** In the AegisCrypter window, enter/select the following options and click the Protect button:

   Filepath: Select the malware you want to pack (hold down the Shift key to select multiple files).

   StubPath: Select a stub to be used. Note that packing malware could make it inert. If this happens, select a different stub until the malware executes.

   IconPath: This auto-populates (leave it as is).

   EnKey: Click the ellipsis (. . .) to auto-populate.

   Mutex: This auto-populates.
5. **Select the boxes to agree to the license terms and click the Agree button.**
6. **Type in a filename (for example,** clickme**) and click the Save button.** You have now created a packed piece of zero-day malware.

   TIP

   A *stub* is code that contains the routine used to load the original malware file into memory. The stubs used in AegisCrypter sometimes make the malware inert. You should execute your AegisCrypter-packed malware sample on a victim virtual image without any security software, to verify that the malware is still working properly (that is, it's still malicious). If the malware sample does not execute properly, then select another stub and test again until you find a stub that allows the malware sample to properly execute. You can find additional AegisCrypter stubs at **`www.testmyav.com`**.

7. **Bring up your clean virtual image snapshot of the various anti-malware products you are testing.** Update all operating system (OS) software patches and obtain the latest updates for your anti-malware products.
8. **Take a snapshot of the virtual images with the various anti-malware products and name the snapshot.** For example, name it "Packed-VendorName-Nov-13-2017-Clean".
9. **Bring up the clean snapshot and keep it online.**
10. **Copy the packed malware created in Steps 3 through 6.**
11. **Execute the malware.**
12. **Record the result.** Did the anti-malware product prevent the malware from executing?
13. **Revert to a clean snapshot.** For example, revert to "Packed-VendorName-Nov-13-2017-Clean".
14. **Repeat Steps 9 through 13 until you have exhausted the malware sample set.**

Online/Offline Testing

Testing an endpoint security technology offline can reveal a lot about the product's architecture and capabilities, which is important in order to make a well-informed decision.

Solutions that rely on cloud lookups may leave the customer at risk by allowing — intentionally or not — a "patient zero" scenario and by potentially introducing delays on the endpoint as a result of the latency associated with cloud processing. If a solution requires cloud lookups to process never-before-seen malware, then it implies that the solution relies on either cloud intelligence (file reputation) or cloud-based emulation.

Cloud intelligence typically uses anti-malware multi-scanner results. While this approach arguably offers better detection for known threats compared to traditional signature-based anti-malware, it remains ineffective against new, never-before-seen malware. Why? Because if the payload is truly new and has never been seen before, then it doesn't matter how many anti-malware engines you query, most are going to return a "not malicious" answer.

Cloud-based code emulation and/or sandbox detonation derives behavioral data and indicators of compromise (IOCs). This approach can be interesting from the point of view of analyzing files post-detection, but it requires the file to run, detonate, and/or execute to some extent before a result can be produced. As such, it tends to require time to perform this analysis, which in turn implies either significant performance impact to the end-user and/or the possibility of a "patient zero" situation.

Some technologies leverage sandbox detonation to analyze a potential threat. This can pose a challenge when considering how easy it is for a malware author to create a payload that is sandbox-aware. Sandbox-aware payloads may opt to sleep indefinitely when executed in a sandbox, or simply run a completely benign piece of its code in order to fool the sandbox into thinking the payload is benign, and therefore let it into the environment.

Finally, if an endpoint security technology requires a persistent and reliable connection to the cloud in order to properly protect on the endpoint, where does that leave the end user in the event of a significant Internet outage or a saturated or unreliable Internet connection? Although this admittedly does not happen every day, there are plenty of examples of major denial-of-service attacks affecting otherwise robust cloud-based applications.

Figure 4-2 illustrates why signature-based anti-malware products fail to detect malware in the real world, as demonstrated by online/offline testing. In panel A, legacy anti-virus software has a detection signature for known malware and can therefore detect the malware. However, if connectivity to the cloud isn't available, as shown in panel B, the legacy anti-malware product can't get any signature and threat intelligence updates, so it is unable to detect any new malware threats.

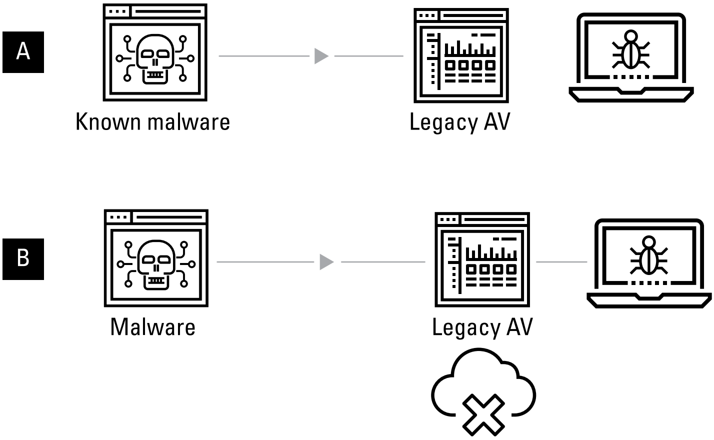


FIGURE 4-2: Why signature-based anti-malware products fail online/offline testing.

It's important to test an anti-malware product offline using a mutated sample, then test it again online, using the same sample set. If there is a significant difference between the offline and online test result, then the anti-malware product can only function properly when online. The reason for testing a product's capabilities while offline is to shed light on its architecture and capabilities.

To perform an offline test:

1. **Bring up your clean virtual image snapshot of the various anti-malware products you are testing.** Update all OS software patches and obtain the latest updates for your anti-malware products.
2. **Take a snapshot of the virtual images with the various anti-malware products and name the snapshot.** For example, name it "Offline-VendorName-Nov-13-2017-Clean".
3. **Download new malware from `testmyav.com` and optionally pack the malware as described in the preceding section.**
4. **Disconnect your offline images from the Internet.** This is necessary to test the efficacy of the client itself, without help from the cloud. Bring up a command window and ping a remote site to verify that the product is offline.
5. **Copy the same instance of malware or optionally mutated malware to each image.**

6. **Execute the malware.**
7. **Record the result.** Did the anti-malware product prevent the malware from executing without being online?
8. **Revert to a clean snapshot.** For example, revert to "Offline-VendorName-Nov-13-2017-Clean".
9. **Repeat Steps 4 through 8 until you have exhausted the malware sample set.**

Now with the same malware used in your offline testing, test your anti-malware product online, using the following steps:

1. **Bring up your clean virtual image snapshot of the various anti-malware products you are testing.** Update all OS software patches and obtain the latest updates for your anti-malware products.
2. **Take a snapshot of the virtual images with the various anti-malware products and name the snapshot.** For example, name it "Online-VendorName-Nov-13-2017-Clean".
3. **Use the malware that was not detected in your offline testing.**
4. **Connect your online images to the Internet.** This is necessary to test the efficacy of the client with help from the cloud. Bring up a command window and ping a remote site to verify that the product is online.
5. **Copy the offline undetected malware to each image.**
6. **Execute the malware.**
7. **Record the result.** Did the anti-malware product prevent the malware from executing while online?
8. **Revert to a clean snapshot.** For example, revert to "Online-VendorName-Nov-13-2017-Clean".
9. **Repeat Steps 5 through 9 until you have exhausted the malware sample set.**

The Holiday Test (Zero-Day Simulation)

Most organizations can't guarantee that all of their corporate endpoints and systems are 100 percent up to date with all the latest anti-malware signature updates. In fact, most can't guarantee that all their systems even have anti-malware software installed, let alone that it is up to date. This is often due to one or more of the following reasons:

- Users are out of the office on vacation, so their endpoints don't connect to the Internet daily.
- Corporate policy is set to only deliver signature updates on a weekly basis to minimize the impact on user productivity.
- Errors in transmission cause the update to be disrupted or corrupt.

Figure 4-3 illustrates why signature-based anti-malware products fail to detect malware in the real world, as demonstrated by the holiday test. In panel A, legacy anti-virus software has a detection signature for known malware and can therefore detect the malware. However, if the endpoint is offline for a period of time, and therefore unable to get signature updates, it is unable to detect any new malware threats, as shown in panel B.

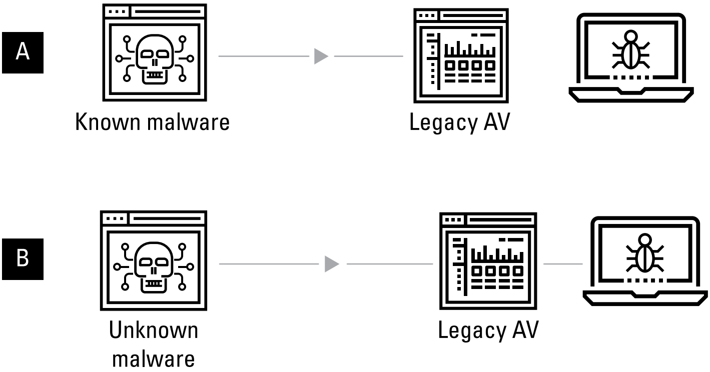


FIGURE 4-3: Why signature-based anti-malware products fail random mutation testing.

This test will also simulate prevention of zero-day-like malware. If the anti-malware product you are testing claims the ability to detect unknown and zero-day malware, then the easiest way to verify such a claim is to put the product out of date. Rather than developing a new zero-day malware threat, which is difficult, you can simulate a zero-day environment to determine a product's ability to detect unknown malware, as follows:

1. **Bring up your clean virtual image snapshot of the various anti-malware products you are testing.** Update all OS software patches and obtain the latest updates for your anti-malware products.
2. **Take a snapshot of the virtual images with the various anti-malware products and name the snapshot.** For example, name it "Holiday-Test-VendorName-Nov-13-2017-Clean". Take the system down.
3. **Wait five days, then download some new malware from `testmyav.com`.**
4. **Mutate the malware using the steps described in the Random Mutation test (optional, but highly recommended).**
5. **Bring up your out-of-date images but *do not* bring them online.** You don't want them to update. Keep the images offline.
6. **Copy the malware or optionally mutated malware to each image.**
7. **Execute the malware.**
8. **Record the result.** Did the anti-malware product prevent the malware from executing?
9. **Revert to a clean snapshot.** For example, revert to "Holiday-Test-VendorName-Nov-13-2017-Clean".
10. **Repeat Steps 5 through 9 until you have exhausted the malware sample set.**

Testing with Fileless Malware

To test the effectiveness of endpoint anti-malware products against fileless malware, you can use Powershell with various malicious payloads. There are numerous ways to test fileless malware. This test is not meant to be exhaustive. Instead, its focus is on ease of use and basic prevention.

WARNING

Powershell is an incredibly powerful tool for administering and controlling a Windows system — that's exactly why its use has become so prevalent within many malware strains.

You can find various Powershell test samples at `testmyav.com`. Each sample contains a notable action (such as file creation, system modification, or reboot) to represent the malicious actions within real malware, such as file encryption and theft.

The Powershell samples at `testmyav.com` are provided as raw text on the webpage for you to copy and paste onto your local test machine. This method ensures there are no actual files to detect. To test fileless malware, follow these steps:

1. **Choose a sample you want to test with.**
2. **Open Powershell on your test machine.** For example, open a command prompt and type **powershell**.
3. **Copy the Powershell text from the `testmyav.com` webpage and paste it directly into your Powershell window.**
4. **Press Enter.**
5. **Observe if the malicious action (described by the selected Powershell text) occurs, or if your anti-malware product detects and blocks it.**
6. **Restore your test machine to a clean snapshot and repeat Steps 1 through 5 with various Powershell samples.**

IN THIS CHAPTER

- » Understanding the challenges and risks of legacy approaches
- » Looking to the future with a proactive strategy

Chapter 5 Turning Your Results into Action

This chapter contrasts three distinct security strategies: legacy prevention, a reactive "detect-and-respond" strategy, and a proactive "prevent" strategy. Legacy prevention succumbs to the sacrificial lamb model. *Detect-and-respond* relies on detection of malware as it executes on a system, then responds with predefined actions after the malware has already run on the system. A *prevent* approach blocks malware before it executes on a system.

Legacy Prevention

Early malware detection technologies used generic and heuristic anti-virus (AV) signatures to detect known malware as soon as it was written to disk. Anti-malware software vendors used to be able to get away with manually writing signatures because malware families did not change that often. Other commonly employed techniques (see Figure 5-1) included host-based intrusion prevention systems (HIPS), sandboxing, behavioral heuristics, and endpoint detection and response (EDR).

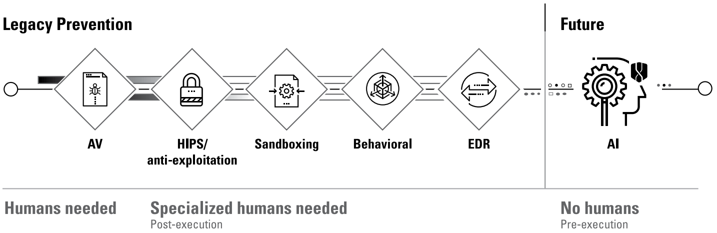


FIGURE 5-1: Legacy malware prevention techniques.

For a brief time, the cost of managing anti-malware security solutions in the enterprise was largely fixed. Once installed, an anti-malware solution would be on autopilot and detect and remediate known malware.

The mid-2000s brought a wave of polymorphic, rapidly changing malware and rootkits, with many tricks for bypassing traditional anti-malware. To keep detection rates up, security vendors created automated malware processing solutions with hashing technologies at their helm. The fatal shortcoming of hashing is that every malware specimen, no matter how slight its differences, looks completely new and different when hashed. The next wave of polymorphic and hash-busting malware preyed upon this limitation with automated hashing techniques. The cost of missed detections rose rapidly for enterprises, so vendors began to offer repair and remediation services. To counter the detection challenges, the industry added memory analysis and behavioral technologies. These security layers kept increasing over time.

This period also saw the introduction of pure policy-based solutions where only known files could execute *(whitelisting)*. This approach was typically limited to specific use cases that had very restrictive change control, such as point-of-sale (POS) systems.

Detect-and-Respond

Today, many state-of-the-art security solutions are responding to the increase in cyberattacks via post-execution malware analysis that includes continuous endpoint monitoring and rapid reactive response to attacks. However, malware execution on an endpoint has inherent risks.

Post-execution monitoring analyzes and logs application behavior, and in many cases also analyzes and stores most of the network traffic. This is all done to detect and eventually recover from the inevitable worst-case compromise scenario.

Coming back to the dilemma of malware execution and analysis on an endpoint, we must answer the question: What should these solutions monitor? What to monitor is an important issue. If a solution is logging most of the network, operating system, and application behavior data in anticipation of the worst, it is collecting massive amounts of data.

Post-execution solutions are noisy. With limited autonomy, they simply can't risk missing something important, so they seek to collect and analyze an avalanche of data. This includes disk writes (and parent processes), execution events, some subset of registry keys, remote procedure call (RPC) communications, user activities (including websites visited and cookies written), domain name service (DNS) requests, and network trace data like packet capture (pcap) or NetFlow for every operation.

The amount of data quickly adds up. For example, suppose these systems collect 1MB per hour per host (or 1,000 1KB records after compression). If you multiply that by 24 hours for 1,000 hosts, you have 24 million events, or 24GB of data a day. After 90 days, you amass 2.1 billion records, or 2.1TB of data. Imagine how much data an enterprise with 50,000 to 100,000 hosts might collect. It's like looking for the proverbial needle in a haystack (see Figure 5-2).

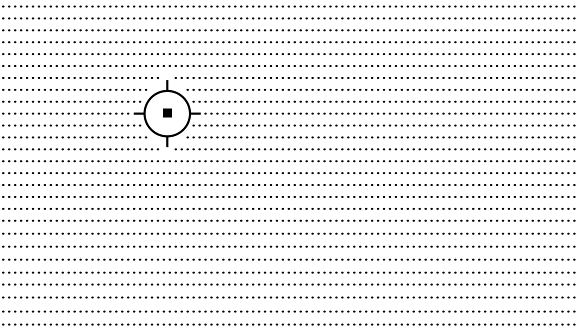


FIGURE 5-2: Modern "detect-and-respond" anti-malware techniques must find the needle in the haystack.

Even with this heavy-handed approach to data siphoning, defenders may never be able to find the needle in the haystack. This burdensome approach is extremely complex and wastes power, memory, disk space, network resources, and human capital.

Gathering and maintaining the volume of information needed to operate a detect-and-respond solution is a commitment that grows with time, as does the cost of extracting value from the information. Enterprises should be aware of the following hidden costs and concerns:

- **Security event analysis:** More security events lead to more analysis and increased costs.
- **Endpoint system performance:** Continuous endpoint monitoring leads to performance bottlenecks, while unnecessary data collection further strains the endpoint.
- **Cloud lookups/network bandwidth:** Although the security vendor is paying for cloud storage, the enterprise pays for network data usage.
- **On-premises analysis:** Hosting and managing a big data solution on-premises to deal with the volume of data adds complexity, as well as hard and soft costs.
- **Privacy concerns:** Solutions that collect and store most of the system events for detection and response may end up collecting more information than is necessary or desired. Access controls, locality, retention periods, and encryption policies on the collected data may vary by vendor.

Some security solutions rely on open-source data to get information about suspicious files. These sources often do not turn up useful data because pre-execution malware detection has become a lost art and security vendors are not investing sufficiently in improving file detection capabilities.

For example, if the Dyre family of malware is detected on June 1, a post-execution system querying any vendor on June 4 will not recognize the sample as malware, based on the industry's collective knowledge at the time. Malware often is not initially identified as malicious and fails to get reported as bad for weeks, months, or even years. Such delays highlight the need for systems to fill the gap in malware identification without relying on reactive file-detection scanners.

Permitting malware execution creates major technical challenges by expanding the playing field for malware instead of limiting its options. Some examples of weaknesses in newer technologies that are based on detect-and-respond include the following:

- **Good/bad behavior:** During malware post-execution analysis, endpoint security solutions need to monitor the suspect in its natural environment to detect, log, and block events in order to recover from attacks. However, even under monitoring, it is very hard to predict when a malware specimen may reveal its ugly side. Days may go by before it executes its malicious code, or it may be dependent on some user action to trigger the malware. Lots of new research has tried to solve some of these problems, only to be circumvented again. An alternative would be to watch all applications all the time in anticipation of a security event, which increases security management costs.
- **How late is too late?:** Can a monitoring technology detect the first bad event? Is installing a driver a malicious event by itself? Most often the answer is no, but the moment a malicious kernel driver runs, it's probably too late to save the system. These are just some of the disadvantages of post-execution monitoring. More often than not, a series of behaviors constitutes a malicious behavior. However, it may be too late to block the malware if that determination is not made in time and, more importantly, every time. This again presents the old signature detection cat-and-mouse game, where defenders try to detect as early as possible and attackers try to evade by mixing good and bad events, and sometimes gray events.

Prevention

Legacy prevention and detect-and-respond technologies have had their merits during the evolution of malware. However, today's malware threats are far more advanced and prolific than ever before, and such techniques are no longer sufficient to protect your systems and network. Relying solely on post-execution detection is a risky proposition for your enterprise.

The basic requirement of any anti-malware solution is as follows: If a file is bad, block it. Although this principle is simple enough, the solution has been elusive. Until now.

With the evolution of artificial intelligence (AI) and machine learning (ML)-based pre-execution prevention, an efficient anti-malware solution is now a reality.

For far too long, we've been conditioned to believe that our defense against malware will be adequate, as long as we keep our anti-malware software continuously updated to detect known threats. When that approach increasingly began to fail, we were then conditioned to believe that a breach is inevitable, so we must add complex layers of defense to respond. This approach is inherently reactive.

It's now clear that relying on solutions that only seek to detect known malware and respond after it has executed is not a viable solution. When the security industry started pulling away from pre-execution containment, technologies became reactionary and too dependent on manual sample analysis and signature creation. Security vendors hoped post-execution analysis and solutions would give them the necessary respite from the malware problem, only to find it made the system more complex, more expensive, more difficult to manage, and more prone to attacks and bypasses.

Think of the choice in terms of your own network environment. Would you rather *prevent* malware from executing by not allowing the malware to run in the first place? Or would you rather *detect* malware by letting the malware run on your systems, then *respond* IF suspicious behavior is identified. *If,* because in order to react to suspicious behavior, you have to allow it to occur and know what it looks like. So, in other words, in a reactive world you have to know what to look for before you know what to look for.

Perhaps if Benjamin Franklin were alive today and working in the anti-malware industry, he would offer up the following sage advice: "An ounce of prevention is worth a pound of detection." True, prevention lowers security costs and complexities, and is thus the best cure for malware. Pre-execution detection environments that leverage artificial intelligence and machine learning offer a solution to this challenge. The main challenge in the pre-execution environment is to analyze the program and determine if a file is good or bad, based purely on the information in the file itself, and then do that at a sustainable, massive scale.

The ability to do this across a huge number of samples is important because modern malware creation is automated. Today, it requires very little effort for attackers to mutate a piece of malware. The Satan ransomware-as-a-service is one example of ransomware that requires absolutely no skill to execute. An attacker simply selects the target, pays the service fee, and collects the ransom.

To be able to go back to the basics and stop malware before it ever gets a chance to execute, machine learning generates models that can predict if a program is malicious. This approach for file detection has proven extremely effective at stopping malware. With machine learning, it is possible to identify malware with astonishing accuracy, utilizing low system resources without ever having seen the malware before.

For example, the recent WannaCry ransomware exploited a Windows vulnerability that Microsoft had released a patch for two months earlier. Yet, WannaCry successfully infected nearly a quarter million computers in more than 150 countries worldwide. However, Cylance customers were protected against unknown malware threats like WannaCry since November 2015 — without ever having to apply a single patch or update (see Figure 5-3). It's a predictive advantage over the adversary — instead of the adversary having a predictive advantage over us.

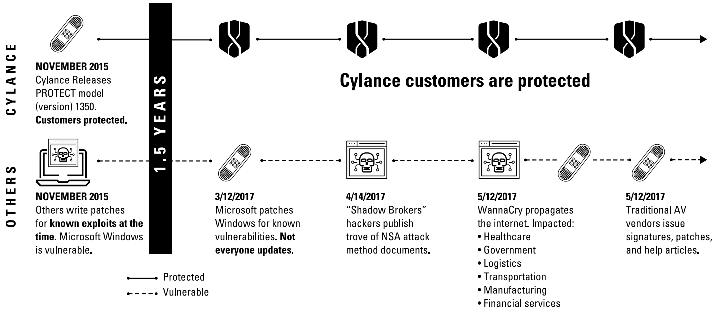

CYLANCE
NOVEMBER 2015
Cylance Releases PROTECT model (version) 1350. Customers protected.
1.5 YEARS
Cylance customers are protected
OTHERS
NOVEMBER 2015
Others write patches for known exploits at the time. Microsoft Windows is vulnerable.
3/12/2017
Microsoft patches Windows for known vulnerabilities. Not everyone updates.
4/14/2017
"Shadow Brokers" hackers publish trove of NSA attack method documents.
5/12/2017
WannaCry propagates the internet. Impacted:
• Healthcare
• Government
• Logistics
• Transportation
• Manufacturing
• Financial services
5/12/2017
Traditional AV vendors issue signatures, patches, and help articles.
Protected
Vulnerable

FIGURE 5-3: The predictive advantage.

TIP

See the following testing reports to learn more:

- **AV-Test Advanced Threat Prevention Test Results** (`https://pages.cylance.com/2017-02-08-CNT-AV-TEST-Report-2017-2092.html`)

- **Black Hat: Lies, and Damn Lies: Getting Past the Hype of Endpoint Security Solutions**

Pre-execution malware prevention is not a silver bullet that no malware can ever bypass. No single solution can be infallible.

REMEMBER

A pre-execution strategy is the first step in building an effective security portfolio. Identifying malicious applications before they get a chance to execute helps limit security management costs and system performance overhead. It can also reduce the challenges posed to post-execution analysis environments, greatly reducing both the number of samples that need post-execution monitoring and the odds that a malicious sample will ever make it past that final layer of defense. That can help reduce the number of security layers needed to successfully thwart our adversaries.

IN THIS CHAPTER

- » Making sure your anti-malware solution works
- » Minimizing the impact to your end users
- » Maximizing platform coverage and deployment options
- » Looking at the complete picture — integration and total cost of ownership

Chapter 6
Ten Anti-Malware Buying Criteria

This chapter offers ten important criteria for you to consider when evaluating anti-malware options for your enterprise.

Addressing the Attack Vectors

First, and foremost, an anti-malware solution must address all attack vectors. This includes effectively preventing malware and exploits, and all their variants — both known and unknown (zero-day) — from executing on your servers and endpoints. Your anti-malware solution should also provide forensic analysis of detected malware and exploits.

Effectiveness

Effectiveness describes the accuracy of your anti-malware solution in preventing adversary attacks. How effective is your anti-malware solution in preventing known and unknown malware from executing? Can your anti-malware solution provide a predictive advantage whereby it can prevent tomorrow's malware yesterday?

Effectiveness is a single source of truth. Either it has predictive capabilities or it doesn't. It's no longer necessarily about efficacy percentage against known malware — it's about the ability to effectively predict and prevent unknown malware from executing.

TIP

The "holiday test" described in Chapter 4 provides a scientific method for testing effectiveness.

Performance

Anti-malware products that introduce significant performance overhead in the servers and endpoints on which they are installed tend to get disabled. Look for solutions that have a minimal impact on CPU and memory utilization, particularly on specialized systems such as point-of-sale (POS) systems, medical devices, and supervisory control and data acquisition (SCADA) systems.

Additionally, your anti-malware solution should work even when it isn't connected to the network or the Internet. It shouldn't be dependent on continuous signature and software updates to be effective.

Ease of Use

Anti-malware products shouldn't require your end-users to be security experts and shouldn't require specialized training to understand and use. The user interface should be simple and intuitive, and most anti-malware activity should take place in the background — without requiring user interaction or knowledge. In fact, the end-user experience when it comes to the anti-malware solution should be silence!

Non-intrusive

Anti-malware should run seamlessly in the background, without interrupting the user experience or negatively impacting productivity (see "Performance" and "Ease of Use"). Most anti-malware actions should happen automatically, based on a centrally managed policy, without user intervention. When it comes to identifying threats and how to protect the endpoint, this truly is a case where the anti-malware product should know better than the user.

Platform Coverage

Today's enterprise computing environment is comprised of a variety of servers, endpoints, and devices running different operating systems. Deploying a different anti-malware solution on various platforms is not a viable option for a comprehensive and robust enterprise security strategy.

Look for a solution that provides proactive, predictive, and preventative techniques in anti-malware protection that can be applied across platforms, operating systems, file types, and devices.

Deployment

Rolling out a new anti-malware solution to all your enterprise endpoints can be a daunting task. Don't let it discourage you from deploying the best anti-malware protection available. Instead, look for an anti-malware solution that can be easily deployed across platforms and locations, without requiring a complex backend support infrastructure such as on-premises central management and update servers. Given today's Software-as-a-Service (SaaS) capabilities, look for solutions that require little (if any) capital investment and avoid extensive on-premises management solutions.

Simplicity

"Defense in depth" doesn't mean deploying multiple security products to compensate for inadequate protection provided in a single product. Rather than deploying layers upon layers of legacy security products with complex, overlapping policies, look for an anti-malware solution that is simple, yet complete.

Management, Reporting, and Third-Party Integration

Infecting a target system with malware is rarely, if ever, an attacker's ultimate objective. Rather, it is the means to an end, and, as such, it's a pretty unmistakable indicator of compromise (IoC).

TIP

Unfortunately, given their experience with past technologies, many incident response teams treat malware infections as a fact of life in an enterprise environment and consider it to be a relatively low-impact incident — as long as the infection is contained and cleaned before other systems on the network are affected.

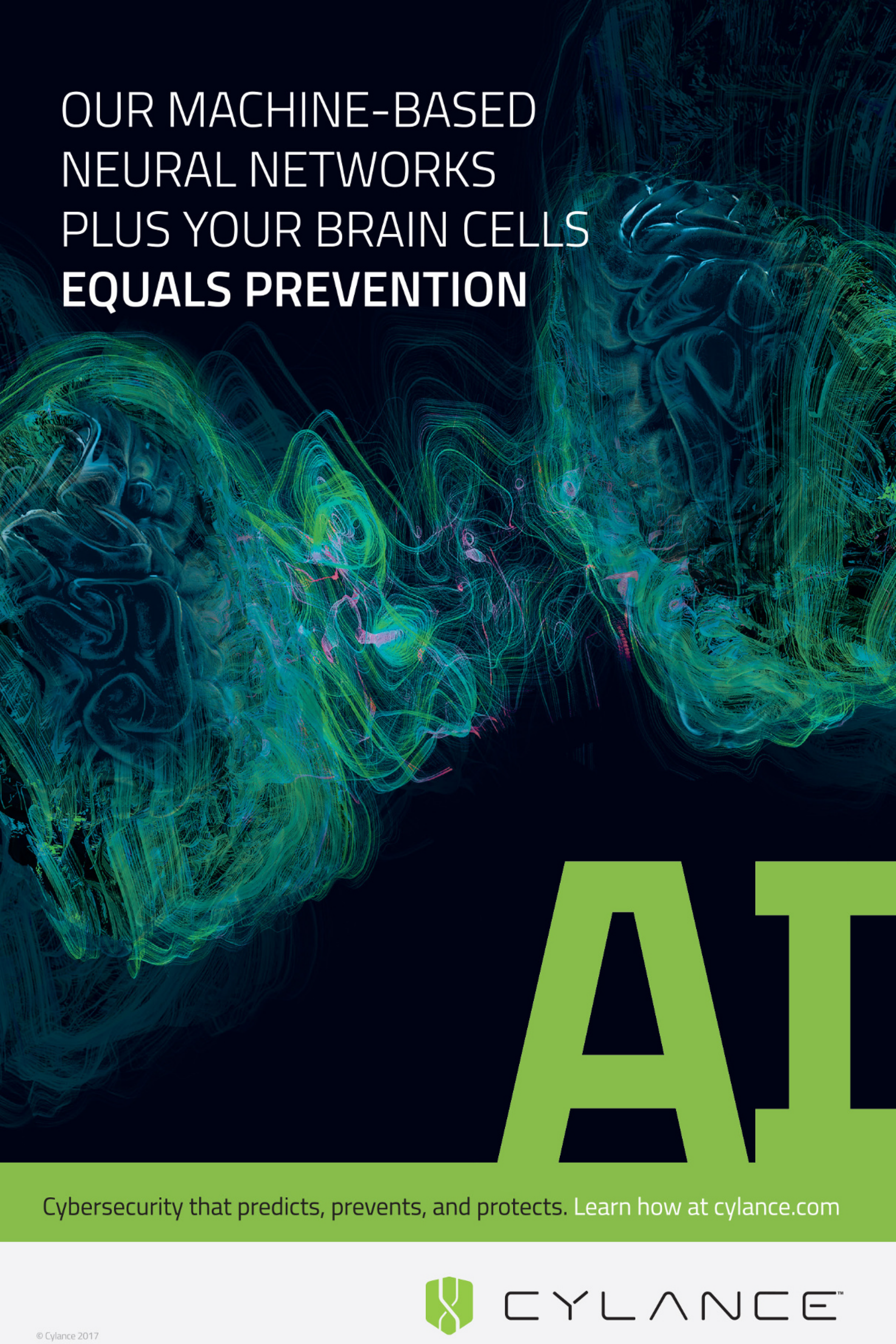
However, this approach to malware infections is dangerous. Just as infecting a target system with malware is just part of the early stages of a cyberattack, cleaning malware from an infected system should only be the beginning of your incident response and forensic analysis efforts.

Look for anti-malware solutions that provide robust management and reporting capabilities and can easily integrate into existing security information and event management (SIEM) platforms. Having open application programming interfaces (APIs) to provide your incident response teams with orchestration capabilities to obtain a complete picture with as much forensic data as possible is important.

Cost and Support

It's important to consider the total cost of ownership for your enterprise anti-malware solution. Beyond the initial acquisition cost, look at the ongoing maintenance and support costs. Do you have to maintain an ongoing subscription to receive continuous updates and access to real-time threat intelligence? Also, what are the potential costs related to lost productivity due to a malware infection? Or lost or compromised information due to a data breach or ransomware? How many personnel does the solution require to manage?

When organizations change their cybersecurity approach to pre-execution, they begin to remove layers of technology, resources, and people dedicated to continuous response, and they can instead redirect them to other strategic projects that generate revenue or create value for the business. As a result, costs are significantly lowered. As companies remove layers and solve issues at the core, they begin to discover ways to consolidate infrastructure.


OUR MACHINE-BASED
NEURAL NETWORKS
PLUS YOUR BRAIN CELLS
EQUALS PREVENTION
AI
Cybersecurity that predicts, prevents, and protects. Learn how at cylance.com
CYLANCE™
© Cylance 2017

Proactively test anti-malware products

Traditional anti-malware products rely on a "sacrificial lamb" to provide malware from an infected host and create a signature for the masses. That's why traditional AV is no longer effective at protecting against new and advanced malware. How can you effectively test traditional and next-generational anti-virus products for yourself without relying on third-party vendors? This book outlines ways to test both traditional and next-generational AV products to protect against advanced attacks.

Inside...

- See the limits of legacy AV methods
- Pair AI with machine learning to combat modern malware
- Set up a testing environment
- Do your own anti-malware testing
- Explore testing methodologies
- Stop malware before it executes

CYLANCE

Chad Skipper is Vice President, Industry Relations and Product Testing at Cylance. **Carl Gottlieb** is Consulting Director at Cognition. **Lawrence C. Miller** is the author of more than 60 Dummies books.

for dummies®
A Wiley Brand

ISBN: 978-1-119-45062-7
Not for resale