

## Past its Prime: Is Anti-Virus Scanning Obsolete?

*Paul Schmehl* 2002-04-01

### Past its Prime: Is Anti-Virus Scanning Obsolete?

by *Paul Schmehl*

last updated April 1, 2002

---

The title and topic of this article is clearly controversial. It is guaranteed to get a strong reaction from the anti-virus industry, which is firmly convinced it sees clear sailing ahead. So, *is* anti-virus scanning obsolete? In a word, yes - but don't throw out your scanner. Its replacement hasn't been created yet. In this article we will examine the weaknesses of virus scanning that will cause its eventual downfall.

Anti-virus scanning is based upon the age-old principle of Newton's law; for every action there is an equal and opposite reaction. Each time a new virus, or a new viral approach, is discovered, anti-virus scanners must be updated. To be sure, this isn't always true. Heuristic scanning does have the capability to recognize some attacks as viral without having specific detection for the virus it has alerted on. In general, however, each new virus discovery requires an update of the scanning software's "virus definition" files in order for the scanner to recognize the new virus.

In some cases (Melissa, for example) the scan "engine" (the algorithm that does the comparison between the virus' behavior and the virus definition files and identifies viral content) must also be updated for the anti-virus scanner to be effective at detection (and hopefully eradication.) This constant updating process has several flaws. We'll look at these flaws in detail.

#### **The Arms Race. Or Is It The Rat Race?**

To keep up with constantly emerging viral attacks, vendors must devise new definition files and sometimes new scan engines. In order to design a new virus definition file, a virus researcher has to obtain a "live" copy of a new virus (sometimes not the easiest thing to do), determine its behaviors through testing, decipher its algorithms to determine precisely what it does, decompile it if its code is compiled and determine how best to detect its presence. In some cases a virus will use encryption ([Hybris](#) is a good example), and the researcher cannot decrypt the virus. This leaves behavioral verification as the only method of determining how best to

detect the virus.

Once the researcher understands what the virus does and how it works, he or she must design a new definition that will work with the existing scan engine or perhaps even design a new scan engine that will detect the virus. All of this takes time, yet every minute that passes the vendor's customers are at risk of being infected. When the researcher finishes his or her work, the quality control department enters the picture.

Software updates cannot be released without testing to ensure that they work as designed and don't degrade the performance of the existing software, the underlying operating system or other programs installed on the computer. Anti-virus software is, by necessity, a low level process that interacts with crucial parts of an operating system. It can easily interfere with other essential processes and render them ineffective or even inaccessible. So, the researcher may have to correct problems in the code, and then the quality control process will begin again.

Even this doesn't ensure that the released update will not cause problems. It simply isn't possible to test every probable configuration that the software runs on. It isn't even possible to know what they are. The number of potential combinations of hardware, operating systems and programs means that the best a quality control department can do is ensure that the software works as expected, with no adverse behavior, on *typical* configurations. Only when the software is released to the general public will a vendor truly know if a problem exists. And anti-virus vendors have to complete this cycle in one week! Some even do it several times a week or even daily! Yet each iteration of the software is more complex than the last, more prone to unexpected interactions, more likely to produce unexpected results.

So the problem is two-fold. Firstly, when releasing updates to detect newly discovered viruses time is of the essence; and secondly, quality is required for those updates to be effective. These two factors work at odds to each other. Quality demands time for it to be done right, and urgency demands that quality be sacrificed in the rush to get the updates out to customers. It doesn't take a degree in mathematics to see that the more new viruses are released, the more quality will suffer because timeliness is the overriding factor in anti-virus updates.

Indeed, most major anti-virus vendors have suffered from quality problems recently, clear evidence that this problem is coming to a head. Problems that have occurred have ranged from updates that don't detect what they're supposed to detect, to updates that cause the client computer to crash, to updates that cause noticeable degradation of system performance.

While all this is happening, the customers are clamoring for an update so they can be protected from the latest threat. It isn't hard to see that there is a point of diminishing returns, where updating is no longer feasible because testing takes too long. At that point, wherever it is, customers begin to look to other solutions to overcome the problem of malicious threats. In fact, that point has already been reached, to some degree, and in the past twelve months many enterprises have turned to other methods to supplement anti-virus protection.

### **The Industry's Dirty Little Secret**

The constant pressure to update has put the anti-virus companies in an awkward ethical position as well. In order to try and stay ahead of the zero-sum game they have developed, they have to lurk in the shadows of the virus underworld, constantly on the lookout for the next variation that comes along. In some cases, virus writers send their newest creations to the anti-virus vendors directly. In others, the vendors "harvest" them from virus writers' Web sites. All this interaction between virus companies and virus writers has led some to question whether anti-virus companies actually employ virus writers to keep the game going. No one has ever proved a single instance of a virus writer working for an anti-virus company, but the accusation reveals an awareness of the ethical dilemma that plays behind the scenes.

Yet anti-virus companies cannot afford *not* to frequent virus writers' Web sites and lurk in the shadows of that world, because every moment of advance notice is critical in getting their updates out on time. In fact, battles go on constantly between companies who get copies of new viruses and companies who want them so their researchers can get to work. Sometimes these battles take on childish proportions as accusations fly back and forth and companies are accused of "holding out" or "taking advantage" of other companies by not releasing virus copies to researchers. None of this would be necessary if anti-virus software didn't require constant updating.

### **Won't You PLEASE Update Now!!!**

Another problem with constant updating is that the customers get tired of doing it and begin to skip updates. They begin making decisions, uninformed as they are, about when updates are really necessary and put off updates when they don't consider the threat critical. Or they simply don't see the need to update so often, so they don't bother with them until they get infected. The outbreaks of Code Red and Nimda proved the folly of this approach, and now most enterprises are updating regularly. Individual users, however, continue to risk infection by not

updating regularly.

To combat this problem, some in the anti-virus industry have recently advocated *forcing* end users to update by taking their choice to update away and pushing updates to them whether they want them or not. Of course this is fraught with peril. What happens if the vendor's site is compromised and the updates themselves become infected? What if the update itself has problems that cause machines to crash or misbehave? How do you prevent man-in-the-middle attacks? What happens when the customer isn't on-line long enough for the update to complete? What about invasion of privacy issues? As much as some in the industry would like to implement them, forced updates probably will never see the light of day.

### **These Problems Aren't Exactly A Secret**

In a long rant on the subject (see [Face It, We're All a Bunch of Addicts](#), Rob Rosenberger, the [Virus Myths](#) Web site owner discusses the problem of updates and suggests that generic detection would be better. Rob points out that generic detection has been around for a long time, but the anti-virus industry claims customers didn't want generic protection. They wanted viruses to be detected by name. While Rob is correct in his assertions, I don't believe generic detection is the answer.

In a [January 15, 2002 article](#), published in Silicon News, Simon Perry, vice president of security for the software giant, [Computer Associates](#), criticized the anti-virus industry (of which CA is a part) for "not doing enough to protect their customers" and claimed that companies needed to adopt a "holistic" approach to virus protection. He stated, "The approach to security should be a holistic one - which is not what the anti-virus vendors can offer."

In January 2002, SecurityFocus published an article entitled "[Holistic" Enterprise Anti-Virus Protection](#)", written by this author, on the same subject. In that article, I argue that enterprises need to use all the resources they have to fight the virus battle. Intrusion detection, firewalls, e-mail blocks, switches and routers, gateway filters and anti-virus scanners can all play a role in keeping the enterprise safe from viruses.

It is obvious that the game of definition updating will collapse under the load of an increasing number of viruses. Another solution is needed, and the sooner the better. What should that solution be?

## Behavioral Blocking

Viruses (and for purposes of this article, viruses are defined as “malicious code” and include viruses, worms and trojans) all have one thing in common – undesirable behavior. They alter, delete or render useless files on your computer. They send files from your computer to other computers without your permission. They send e-mail to the people in your address book, and they disrupt the normal functionality of your computer. All of these actions are behavioral, and it’s possible to define those behaviors in such a way that a program can look for them. For example, no program should need to alter critical operating system files. Any program that attempts to should at the very least generate a warning if not be prevented from making the alteration.

Interestingly, viruses are detected now (and always have been) by behavioral recognition. Unfortunately, the customers are the ones who have been forced to perform this function. When a new, previously unknown virus infects a customer’s machine, the customer will recognize that something has changed. At that point, an interaction between the customer and the anti-virus vendor begins and the process of updating is initiated.

It seems quite logical then, that if the customer has to perform the behavioral recognition process anyway, that software should be designed to perform that task *before* the malicious code reaches the customer’s desktop. Indeed some development has been ongoing in this area. A January 28, 2002 Network World article, entitled [Behavior blocking repels new viruses](#), discusses the existing products, all of which address parts of the problem, but none of them address the issue of behavioral blocking of all data coming in to a network.

What is needed is a system that will address all active and executable content arriving at the doorstep, before it ever gets to the users’ desktops. Due to speed and bandwidth requirements, it may be necessary that this process be a “quarantine” of sorts, where all potentially “bad” content goes for inspection before arriving at its final destination. Once there, the content can be run in a virtual environment, simulating all the operating systems that an enterprise uses. In my next article, I will discuss behavioral blocking in detail; what’s available now, what is needed for it to replace anti-virus scanners, and what its parameters need to be for it to successfully defeat malicious code.

*Paul Schmehl is a Technical Support Services Manager with over 25 years experience. He is currently employed in IT management in higher education, in enterprise-wide technical support, help desk management and anti-virus protection. Involved in many new technology projects, web site development and security-*

*related issues. Paul is also a founding member of [AVIEN](#).*

#### Relevant Links

[Behavior Blocking: The Next Step in Anti-Virus Protection](#)

*Carey Nachenberg, SecurityFocus*

["Holistic" Enterprise Anti-Virus Protection](#)

*Paul Schmehl, SecurityFocus*

[Privacy Statement](#)

Copyright 2006, SecurityFocus