

FEATURE 2

REAL-WORLD TESTING OF EMAIL ANTI-VIRUS SOLUTIONS

Dr Adam J. O'Donnell
Cloudmark, Inc., USA

Testing security products can be a very complex task – especially validating the effectiveness of technology against threats that are either difficult to enumerate, or which evolve at an extremely rapid rate.

If you are testing a source code flaw-finder that examines code for specific classes of programming flaws, you can create a test set with a large number of example errors and then refine the code until all the flaws are detected and none of the correct segments are caught as false positives. Likewise, if you are writing a vulnerability scanner that purports to detect a list of known security and configuration holes, you can construct a pool of example systems where the issues can be found and confirm that the scanner detects the complete list.

In many situations, however, test vectors that are representative of the threat environment cannot be created, making the validation of a security technology somewhat challenging. What happens when the product you are testing is designed to catch threats that are evolving rapidly? Building a corpus of known threats for testing against a live system is futile if the time between creation and testing is long enough for the threat to evolve significantly. Either the live system must be tested against live data, or the system, which most likely has been fetching updates, must be 'rolled back' to its state at the time each element in the test corpora first appeared.

ANTI-SPAM TESTING METHODOLOGIES

Consider anti-spam systems, for example. The most essential ingredient for an accurate test of an anti-spam product is a stream of *live* email, rather than a stale, pre-screened corpus. If spam did not evolve at a high rate, then corpus-based testing of an anti-spam product would provide catch-rate figures that were commensurate with those seen when the filter is put into production at the mail gateway. The rate of change of spam content is so dramatic, however, that accuracy figures provided by corpus testing become misleading as the corpus ages by the hour.

The 'accuracy drift' of the anti-spam system under test would be insignificant if not for the fact that spam evolves



at such an incredibly fast rate. If spam did not mutate so quickly, then Bayesian filters and sender blacklists would have been the final solution for the entire messaging abuse problem.

In the past year, *Cloudmark* has seen more and more customers realize that a live data stream is essential for evaluating a new anti-spam solution even before we begin our initial engagement. It is our experience that the differences in the expected performance, as derived from corpus testing, and the performance realized once the filter is put into production, are driving customers independently to adopt more stringent testing methodologies.

GENERAL SECURITY PRODUCT TESTING

I began this article intending to discuss security products, and thus far I have only mentioned anti-spam systems. The issues with testing became apparent in this area first because of the number of eyes watching the performance of anti-spam systems: almost every email user on the planet.

What about other filtration methods? In the past, anti-virus systems had to contend with several hundred new viruses a year. A set of viruses could easily be created that would be fairly representative of what a typical user would face for many days or weeks, as long as the rates of emergence and propagation of new viruses were low enough.

The assumption that a regularly generated virus corpus could be representative of the current threat state was mostly accurate in the days when amateurs created viruses with no motive other than fame. However, contemporary viruses are not written by 'script kiddies' trying to outdo each other, but by organized professionals attempting to build large networks of compromised desktops with the intention of leasing them as automated fraud platforms for profit.

The profit motive drives a much higher rate of malware production than previously seen, as exemplified by the volume of Stration/Warezov and CME-711/Storm variants which caused difficulties for many of the AV companies that attempted to catch the outbreaks.

IMPLICATIONS

What's the big deal if people don't perform testing correctly, and what does this have to do with new virus outbreaks? Engineers typically design and build systems to meet a specification, and they place their system under test to verify that this specification is being met. If their testing methodology is flawed, then they cannot detect design flaws that are likely to exist in the product. Eventually, these flaws will emerge in the public eye and, in the case of AV

products, consumers will start to realize that products with claims of 100% accuracy have been allowing viruses through.

In other words, by testing against even a slightly stale corpus, and not against new variants, AV filter designs are able to claim considerably higher accuracy than their products actually provide. This is not to say that on-demand testing is completely neglected; rather it is relegated to a secondary role behind the de-facto industry standard of corpus testing.

I am by no means the first person to discuss the testing of anti-virus products. The subject received quite a bit of attention when *Consumer Reports* attempted to test anti-virus systems using a set of newly created viruses rather than a standard corpus. While their attempt at devising a new testing methodology may have been well intentioned for the testing of the heuristic components of scanners, it was not representative of how threats appear on the Internet. Using new, non-propagating viruses to test an AV system is almost equivalent to the proverbial tree that falls in a forest that no one is around to hear.

Additionally, the incremental changes that are usually detected by heuristics are not often characteristics of the viruses that become significant threats – it is the radical evolution in viruses and the time required for the anti-virus vendors to react that are of more concern to us. These are things that cannot be modelled via corpus testing, but only via extended testing on live traffic.

LIVE TESTING

We should ask why testing is not done primarily on live data as opposed to corpus-based analysis. I suspect there are two reasons: labour and repeatability.

With corpus testing, the tester hand-verifies that each element in the corpus is a virus. This is done once, and that cost is amortized over every test conducted using the corpus. This isn't a realistic option with live testing, since every message that is either blocked or passed by the filter must be examined by hand. Collecting repeatable test results is also challenging because, to be meaningful, the test must be conducted over an extended period of time to cover multiple, large and unique virus outbreaks. However, just because something is difficult, does not mean it shouldn't be done.

TOWARDS ACCURACY METRICS

In situations where there are a limited number of security vendors and adversaries, even live testing becomes extremely difficult. Consider the following hypothetical

situation, where there is only one security vendor and multiple adversaries. Every client system is identical and running current anti-virus packages.

From the standpoint of the testing and user community, the accuracy of the system is perfect; no viruses are seen by the system since they don't even have an opportunity to propagate. At the same time, virus writers realize there is a huge, untapped source of machines just waiting to be compromised, if they can just gain a foothold. These individuals sit around and hack code until a vulnerability is found in the AV system, and upon finding it, release a virus that exploits it in the wild.

Before the attackers uncovered the hole in the AV engine, the system could be viewed as being 100% accurate, since no viruses propagated.

After the virus is released, havoc breaks out as 5% of all computers worldwide are infected before the vendor releases a patch. If the vendor was able to move faster, the number of compromised systems may have been only 1%; left to its own devices, with no patches applied, the virus would have compromised every system connected to the net. In this situation, the accuracy of the system is even more difficult to quantify.

Consider the three following accuracy measures:

1. Accuracy = 0%. No viruses were in circulation at the time except for the malware from the recent outbreak, on which the scanner had zero accuracy.
2. $\left(\frac{\text{Virus Corpus Size} - 1}{\text{Virus Corpus Size}} \right) * 100\%$.
Several viruses were in circulation at the time. Detection accuracy was perfect on all viruses except for the latest outbreak.
3. $\text{Pr}(\text{a given system is not infected}) * 100\%$. The probability that any given system was not infected by the contagion.

The third of these measures seems the most appropriate, and the most flexible, given a variety of network and economic conditions and adversary styles. Anti-spam system evaluators use the measure, which is effectively the expectation of exploitation for a given host. It is a slightly more sophisticated way of expressing the probability that a piece of spam will get through.

RESPONSE TIME AND ZERO-DAY AV

From a general security standpoint, however, this measure covers a difficult and often ignored parameter that is critical to the accuracy of a security product: vendor response time. If the window of vulnerability between when the virus first

appears and when signatures are issued is reduced, the accuracy expressed by this metric improves.

The Zero-Day Anti-Virus (ZDAV) industry is an emerging sub-industry that attempts to address this issue directly by shortening the time between outbreak and AV coverage by using fingerprints that are generated and issued automatically.

Although the technology cannot be used for infection remediation, ZDAV's utility for keeping a message stream clean of emergent viruses before desktop AV systems are capable of catching the content makes it an incredibly effective means of reducing the number of infected systems in the wild.

While many methods of providing so-called zero-day coverage exist, they all revolve around removing a traditionally critical component from the fingerprint-generation loop, namely the small team of highly trained malware analysts.

For example, *Cloudmark* correlates reports from a large pool of both ordinary and trusted honeypots and human respondents, and allows a decision on the disposition of the new sample content in a handful of minutes.

Both honeypot and human submitters who provided a report that agrees with the overall community's assessment gain the system's trust, which is used by the system to (1) issue fingerprints originating from those reporters more quickly in the future and (2) remove reporters who submit bogus content.

Many of these zero-day technologies are being used primarily in the message stream, but this restriction probably won't last for long. The technology appeared for messaging first because of the high rate of emergence of email virus variants, as well as the ease with which service providers – who ultimately fund these technologies – can quantify its cost. Mail is a store-and-forward technology that provides managers an opportunity to examine the number of viruses, unlike web-based trojans that fly through alongside legitimate traffic and don't provide much opportunity for analysis.

CONCLUSION

As consumers begin to demand performance estimates that match their real-world experience, technologies similar to the zero-day methodologies described will appear in areas outside of the message stream. Testing methodologies for anti-virus products must become much more rigorous and focused upon real-world scenarios such as live-stream testing, rather than a second-tier test metric compared to corpus testing.