REBUILDING ANTI-MALWARE TESTING FOR THE FUTURE

Dr Igor Muttik, James Vignoles
McAfee Avert, Alton House, Gatehouse Way,
Aylesbury, Bucks HP19 8XD, UK

Tel +44 1296 318700
Email {igor_muttik, james_vignoles}@mcafee.com

ABSTRACT

This paper discusses several aspects related to testing the ability of security products to detect malware.

The complexity of malware and of security solutions continues to increase extremely quickly, so we present arguments as to why we believe that traditional comprehensive testing of anti-virus products (QA) is no longer viable and why a different approach is in order.

We look into the problem of compiling a representative 'next-generation' sample test set:

- Balancing the test speed with the breadth and depth of testing.
- Ranking threats and removing or downgrading the rank of legacy threats (e.g. DOS and *Word 6* viruses).
- Removing short-lived and inactive threats (e.g. spammed downloaders where the site has been shut down).
- Tracking the history and relationship of malware samples (downloader of what? where from? is URL still alive? gaming password stealer for *WoW* or for *Zhengtu*?).
- Excluding most HTMLs (are encrypted URLs malicious code or are they just obfuscated data?).
- Downgrading or excluding downloaders for the sake of what they download.
- Ranking clean data and false alarms (just like malware, clean programs are not equal).
- Attempting better separation of the malware samples and spam (encrypted URLs could be tricky to classify).
- Considering fair representation of local threats (e.g. could there be too many Brazilian password stealers vs oriental trojans related to gaming?).

We present topological and percolation models of malware distribution and arguments as to why the user profile should be part of the test.

We discuss potential solutions to QA problems:

- Running different tests for different user profiles.
- Organizing collections in attack sample groups rather than individual samples.
- Collecting telemetry data via testing/reporting plug-ins to security products.
- Using live telemetry to collect malware execution data (frequencies, geo-location information, etc.).
- Using telemetry to rank malware attacks.
- Standardizing the format of telemetry data and sharing it within the industry.
- Testing complete security products (e.g. AV bundled with anti-spam rather than pure AV).

THE ISSUES

Malware sample growth

One of the biggest issues impacting the testing of malware is the level of resources required to prepare for and conduct testing, in terms of human time and hardware.

It has been shown before that for test results to be fair one has to use a complete set of malware samples [1]. At the same time the number of malware samples is growing at an increasing rate. Figure 1 shows a year-to-year comparison of the number of malware families registered by *McAfee Avert Labs* from 1997 to 2007 and Figure 2 shows the running total of malware samples recorded in our samples database over the last 12 months.

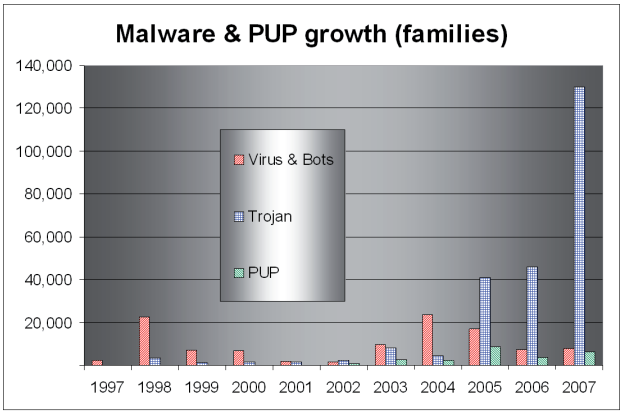

Figure 1: Year-to-year comparison of the number of malware families (source: McAfee Avert Labs).

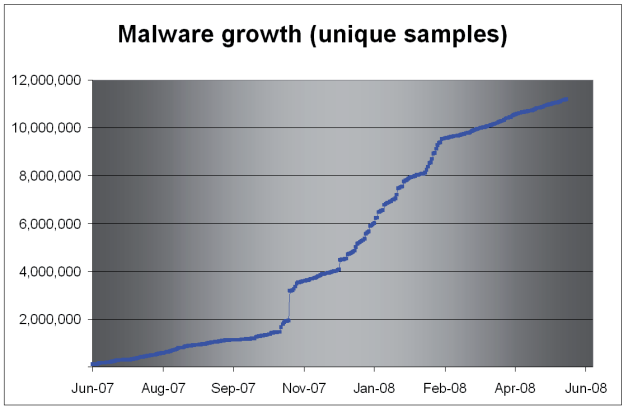

Figure 2: Running total of malware samples (source: McAfee Avert Labs).

Test sample reduction

From Figure 2 we can see that there were approximately five million malware samples at the beginning of 2008, and just six months later this number had exceeded 11 million. Obviously, a test involving all 11+ million samples could be conducted if enough computers were used in parallel, but we have to ask ourselves if this is necessary.

In the 1990s AV scanners were on demand only and the culture of running on-demand scanning (ODS) tests was born. It represented the end-user experience pretty well. With the introduction of on-access scanners, on-access scanning (OAS) was added to the test protocols where ODS and OAS were tested separately over similar (or identical) sets of test samples. Further developments in security software were not easy to reflect in testing. By this we mean such features of contemporary security suites as firewalls, behavioural and access protection rules, anti-spam blocking, whitelisting, URL filtering and similar. Testing of all these features is a significantly more complex task and discussions of corresponding methodologies have only recently started [2].

Do we need to test as many samples as we possibly can to achieve a fair and accurate test reflecting end-user experience? We do not believe so. There are several reasons for this:

- The lifetime of malware has become shorter, so older samples are becoming less and less relevant.
- The increase of custom malware (this includes server-side polymorphism, locale-specific malware and samples for the users of specific software – e.g. online games) makes many samples irrelevant because detecting them bears no relation to the ability of a product to protect from the same attack again or protect another user or group.

We can represent the above points on a graphic (Figure 3) that shows the changes in the landscape over time. Initially malware was visible and propagated slowly (no network propagation, mostly via floppy disks) – the AV solution was simple. Distribution (number of affected users for a single malware item) was fairly high as malware was almost exclusively viral at the time and there were not many field viruses.

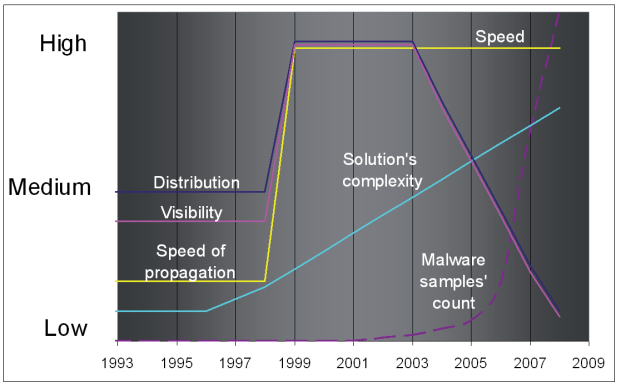

Figure 3: Changes over time to malware properties (speed of propagation, visibility and distribution) as well as the malware sample count and the complexity of the solution.

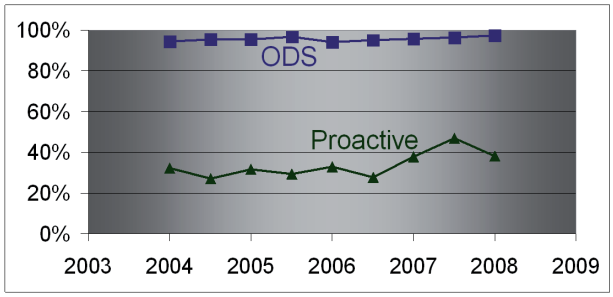

Figure 4: Industry average detection rates of known malware samples (ODS) and unknown ones (proactive) for nine AV products that participated in all AV-comparatives tests from 2004–2008.

Around 1998 Internet-based propagation kicked in and since then the speed of malware distribution has remained high. The AV solution incorporated firewall features, outbreak management and so on. During this period global outbreaks were common and most malware was very visible.

Around 2003 the malware scene started to become a more commercial operation – malware became increasingly covert (rootkits can make malicious programs almost invisible) and distribution decreased – we started seeing an asymptotic tendency for a single sample to target a single computer. This targeting of samples (server-side polymorphism or a specific targeted attack onto an individual user or small group) means that the likelihood of one or more AV vendor obtaining a sample is greatly reduced.

In fact, over the last three to five years the coverage, complexity and functionality of security solutions have grown significantly. At the same time, the effectiveness of pure AV has not changed much. Figure 4 shows the average industry detection rates over the last few years – both over known samples and unknown ones.

We can see that these rates have not changed much over the last several years. But bearing in mind the extreme growth in the number of attacks in recent years, we can conclude that the number of missed attacks (by pure AV, not counting other means of protection) is increasing rapidly.

Everything in, nothing out

Due to the fact that in the 90s threats were predominantly viral (so they persisted for a long period of time) people got used to the idea that malware collections would only grow and that nothing would ever be excluded – either from AV detections or from the test sets. Today, when viruses are outnumbered by short-lived trojans, the AV developers and even the testers are tempted to drop inactive old threats. A bigger problem here is the users (or small magazine reviewers), many of whom have their own collections of prehistoric malware and who assume old stuff will always be detected. It is a very difficult problem to solve and this puts unnecessary pressure on AV update files as they only grow and grow. That eats memory, consumes disk space and bandwidth. Just as a comparison, anti-spam software appeared much later and so there is no expectation to detect spam from, say, the previous year. We believe AV solutions

should move in this direction too and a concerted move of the AV vendors to retire detection of the legacy threats sounds like the only possible way forward.

Another concern in this area is the principle of 'easier to add than to argue', which has resulted in the generally accepted practice of adding the detection of samples regardless of their viability. This puts further pressure on the AV update files.

This principle has produced, and will continue producing, a snowball effect whereby detection of a 'clean' sample (false alarm) can propagate through the industry – effectively a poisoning of the sample set.

Distribution of samples

A large part of any comparative test is compiling a test set. Let us have a closer look at the whole process of sample distribution in order to understand how the compilation of sample sets for testing can be optimized.

We mentioned that malware is now more targeted and that means that the likelihood of AV vendors getting hold of a sample is reduced. Unfortunately, in the case of a successful targeted attack it is not likely that any vendor would get a sample at all (attackers use special clean-up measures to delete temporary programs and files; they use fake error messages to conceal any malware activity and rootkits to hide active malware components).

The AV companies have been exchanging samples for a long time. For many years they swapped monthly virus collections. Following the general increase in malware attacks the swaps became weekly, daily and multi-daily (essentially live feeds).

Our colleagues from a number of AV companies provided us with information about the frequency and volume of collection exchange. We were pleased to find out that on average each AV company regularly receives samples from 25.9 sources – the number was higher than we expected (although there could be some degree of over-estimatation here, as not everybody revealed their sources and it is possible that those who did had more than those who declined to share this information). The average count of samples received in a single monthly collection is approximately 24,500 samples (for January–June 2008 monthly collections – this does not include daily sample feeds). The average number of samples from all sources (monthlies, live daily feeds, customers, honeypots) for the January–June 2008 period was approximately 600,000 unique samples a month or more than 20,000 a day. The average sample size is currently around 240 kbytes and it is slowly but constantly growing over time.

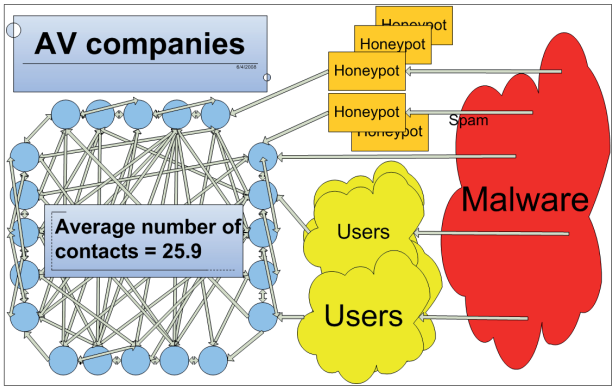

Figure 5: The topology of sample exchange.

Figure 5 depicts the topology of sample exchange. Some samples reach AV companies directly (mass mailers and spammed links to malware end up in our submission mailboxes all the time), other samples are obtained from our own honeypots and crawlers, but the majority of samples come from other AV companies via sample exchange. That means a unique sample, submitted to a single AV company will be distributed to other vendors at some point after receipt (as part of a monthly collection or a sample feed). Then it will, in turn, be redistributed until it reaches all the recipients. This propagation of a sample can be described in terms of percolation theory [3] but we could not investigate this model in depth as most of the information about the exact nature of the graph is private data that was not available to us.

As discussed above, the more or less constant proactive detection rates (see Figure 4) coupled with a rapid growth in malware attacks means that pure AV is becoming less effective. In our opinion this degradation could be for the following reasons:

- A sample was not available to build protection against before the attack (the exact sample for unique detection; a similar sample for a generic family detection or a similar kind of sample for heuristic protection)
- A sample was available, but was sitting in a queue (or we can call it a backlog if it is a long queue).

In any case early availability of a sample is the crucial factor in building protection. So we can conclude that to protect the public, collection exchanges (and especially live sample feeds) are becoming more important because they can provide samples earlier. It is also important, of course, to process these samples and build the actual detection.

For testers to be able to compile independent test sets they need as many of the sample sources as possible. It would be best if they had even more sample sources than the AV vendors whose products are under test – an ideal which is probably hard to achieve.

Do no harm

A crucial aspect of any testing is the protection of the public from any accidental release of malware. Security processes need to be in place and enforced to ensure due diligence is carried out regarding the prevention of 'leaks' of malware into the public domain. This has historically been achieved through the use of isolated networks.

If AV companies start running big honeypot computer farms connected to the Internet then these computers, when infected, can be open to misuse and contribute to the problem (being used for DDoS attacks, relaying spam and similar). Certain controls to limit this possibility can be put in place but they are likely to impair the ability of these honeypots to catch malware successfully. Finding a balance between these two factors is non-trivial.

Going forward we will need to investigate the use of:

- One-way networks
- Firewalls with filters (possibly with human review when a filter triggers)
- Allowing for specific protocols only
- Logging and playback on isolated networks.

Accuracy and reproducibility

Although the complex nature of the malware environment limits the ability of testers to ensure absolute adherence to the scientific principles of repeatability, verifiability and controlled changes, it is still important to ensure that testing is conducted in as scientific a manner as possible.

It is desirable to run comparative tests in a reproducible manner. To be able to do that, all parameters of the test should be fixed (samples, updates, test computer hardware and software, etc.). Sometimes this is not possible though – the products increasingly use dynamic detection methods (behavioural, 'herd intelligence', etc.) which cannot be 'frozen' because, for example, they require a live Internet connection to a remote server (which may hold a live database of behaviours, blacklists, whitelists, fingerprints, hashes or similar). In such situations it is very important to keep the test logs to the maximum detail. It may not be possible to re-run the test but if all logs and network captures are kept, then at least all results would be fully supported by this documentation. That makes results verifiable instead of reproducible – which is a lot better than neither of the two.

Foundation for rebuilding

It is our belief that the issues discussed here are common to all types of testing of anti-malware solutions, whether this is product testing prior to release, competitive testing conducted internally or the comparative testing conducted by independent companies. We are therefore exploring the generic foundations required to move forward with the rebuilding of anti-malware testing.

It is worth noting that the current situation has resulted in the importance of all the proactive detection methods growing considerably. Although this is what the generic, heuristic, sandboxing, behavioural and other similar methods of detection in AV software have been doing for a long time, it is even more critical today.

A big addition to traditional AV comes also from complementary security solutions like anti-spam (AS) and firewall solutions. For example, if a spammed email with a link to malware is blocked by the anti-spam component of a security suite then AV does not have to be able to detect the piece of malware that this bad link points to. Obviously, it is best if it does, but combined protection provided by AS+AV is clearly likely to be higher than that provided by pure AV. Another example is Internet browsing – if web-filtering software is in place (for example [4]), this greatly reduces the risk of contracting malware from the web. The same is true when any other additional protective measures are in place.

Ideally the testing should incorporate these multi-dimensional solutions even though this is very difficult due to a lack of common methodology and the specific nature of some of the technologies used.

The first foundation of any test is the test set of samples used. The issues and the solutions mostly differ in magnitude today vs the past but are worth reiterating.

DEFINING THE TEST SET

In addition to the requirement to remove any duplicates, non-viable samples and similar from the test set, it has previously been shown that selection criteria have an enormous effect on the outcome of any comparative test [1, 5].

Whilst the filtering of the sample set will remove a fair number of samples we will still be left with too many for the purposes of testing. The level of resources required to conduct a scan of such an extensive test set would be unacceptably high and, given the rapid malware creation rate, it is a safe assumption that new important samples would arrive during the test run. Either adding these fresh samples to the test set or ignoring them would impact the results so neither is completely scientific.

So how do we approach the problem of reducing the test set size to make it concise and, at the same time, relevant? The first part is to ensure that we are balancing the test execution speed against the breadth and depth of testing. We do not want to have a very fast test against a small test set, producing inaccurate results, nor do we expect a very slow test against a test set containing 'all known' viruses, which although accurate would require significant resources. Secondly, we must recognize that the size of the test set is very important to produce a fair result. A small test set will produce random results depending on the selection of the samples (see [1]). That means as many samples should be tested as possible to avoid this kind of bias. This, naturally, leads to a contradiction between the accelerating number of malware samples and the limited resources available for testing.

The first suggestion would be to carry out further analysis and segregation of the collection. This would allow prioritized testing of the collection and the removal of lowest priority samples if appropriate. Some examples would be:

- Ranking threats and removing or downgrading the rank of legacy threats (e.g, DOS and *Word 6* viruses).
- Removing short-lived and inactive threats (e.g. spammed downloaders where the site has been shut down).
- Excluding most HTMLs. (The majority of HTMLs in collections contain encrypted URLs, and while one can say they contain malicious code another will insist it is just obfuscated data. We believe the latter point of view to be more practical for the simple reason that AV should not overlap anti-spam products; detecting unwanted URLs is largely the area covered by anti-spam solutions.)
- Downgrading or excluding downloaders completely for the sake of what they download (downloaders are frequently only malicious by association with a specific URL that

hosts malware and many legitimate downloaders differ from malicious ones only because they access reputable websites).

- 'Chopping the tail' (removing samples older than, say, two or three years).

To be able to build such test sets one needs a lot of meta-data stored in a sample database (DB). Very little of this information is readily available (except, maybe, the timestamp of the first appearance of a sample). Generating this information from the samples themselves is not always possible (e.g. checking if URLs are active requires an active research environment – this is equivalent to re-analysing the threats, which is not a very practical solution when the inflow is so significant).

A sensible compromise would be to switch gradually to a new method of populating a sample database. Old samples can be left as they are – without the meta-data in the DB. For new samples the important metrics can be collected at the time of receipt and saved into the DB. Then all this meta-data would be readily available for extraction when we build our test set.

Naturally, for this meta-data (a.k.a. telemetry data) to be used in the industry-based tests we have to find a way to standardize and communicate it. This is difficult for two reasons – firstly, agreeing on the technicalities and, secondly, the AV industry is highly competitive and some of the information may be considered too sensitive to share. We have to note here that in the past the same arguments were put forward against sharing malware samples. In the current interconnected (or should we say Internet-connected?) world no single security vendor would have access to information that others cannot get eventually. So our belief is that to ensure the best protection for the customer security vendors will need to co-operate and start exchanging more than just the malware samples – extending the sharing to the URLs, the commonality, attack vectors and similar information that would help prioritize the samples. This will serve two purposes – internal prioritization will improve the response time to more prevalent or important attacks, and it will also help build concise and representative sample test sets. At a minimum the sharing of this telemetry data with the independent testers would help build the representative collection while protecting vendor confidentiality. We are hoping that AMTSO can serve the role of a body that would standardize the format of the telemetry data because without a common standard, sharing is not likely to be useful.

The second suggestion would be the use of 'hard core' collections so that, rather than trying to test against all known malware, the time would be spent testing what is not commonly known. This could be achieved by each sample being scanned by a small number of reference scanners; those samples identified by a majority would be removed, leaving a smaller subset of 'hard core' samples to be tested against the wider spread of products. The use of a small number of reference scanners in this manner would avoid introducing any bias as the scanners would not be evaluated against the samples removed from the collection. It is worth considering how AMTSO could be leveraged in this area, perhaps through being a sample collection body or the creation of a sample collection plug-in available to all.

Our third suggestion is to organize the samples into 'attack sample groups'. In the past most of the threats were represented by a single sample. These days, multi-component malware is a lot more common. For example, a downloader-based attack always involves at least two files. Frequently a downloader installs many files on a local computer. The tendency to be more modular is due to a shift from parasitic to static malware – it is awkward for a parasitic virus to be multi-file but for static malware it is quite convenient. Modularization of malware follows common software design principles – this reduces the development and maintenance costs. One example is having a rootkit component as a separate program – this allows the rootkit component to be re-used in many different attacks. Our point is that individual samples no longer adequately represent an attack. It makes a lot more sense to organize samples into groups of related samples.

From the point of view of testing this creates two problems. Firstly, organizing the sample test set into groups requires knowledge of the relationships. And again, it is impossible to achieve 100% coverage of all historic samples so a transitional period would be required.

The fourth aspect of defining the test set is recognizing and resolving the bias (natural or otherwise) in acquiring samples. This bias may be regarding the geographic source of the samples; the vendor located in the same region as the majority of the samples could be assumed to have an advantage. For example, because contemporary threats are more targeted and also do not replicate, it would be natural to assume that samples coming from somewhat isolated environments (e.g. China or Eastern Europe) would appear in, say, western Europe with a delay, if they appeared at all. So, a password-stealing program for the Chinese game *Zhengtu* [6] would predominantly affect users in China. The few users in other regions may not be sufficient for such a sample to appear on the radar of non-Chinese AV vendors. That means security companies who do not have big representation in China may only get a sample in a monthly collection from those AV companies who had an affected customer. The source of the samples for a test set also introduces a bias: for example, samples frequently come from the same companies that supply the products to test. It is quite obvious that the correlation between the two will affect the results compared to a test where all samples came from independent sources. The final factor influencing the bias of the test set is the timing. If most of the test samples were collected over a short timeframe, the fluctuations of inflow from various sources can create an imbalance in the test set. During this period some sources may have had fewer samples added to the set.

The fifth and final consideration: in the past there was an assumption that a product which found one piece of malware is likely to identify others belonging to the same family. However, the increasing specificity of contemporary threats makes it more difficult to justify their inclusion in the test because such threats, being unique, would rarely (if ever!) reoccur in the future. And as we all know, past successes do not guarantee future performance. Yet these threats are still valid for their lifetime so need to be included to represent what the end-user sees. This specificity is causing a change to AV technology because,

historically, it was based on an assumption that any piece of malware would be re-used. If this assumption is no longer true then adding detection for every piece of malware could cause issues with the technology. As far as we are aware there are no mechanisms at the disposal of testers to allow the tracking of the ageing and relevance of threats which would aid in the reduction of this issue.

Once the 'dirty' test set has been defined, the next stage is to gather a 'clean' test set to provide the balance. To quote Dr Alan Solomon, 'if something is superb at detecting viruses, it's no use if it gives a lot of false alarms.' While this is a separate test from the 'dirty' test, it is an essential but often forgotten part. Currently the principle of gathering a clean set is often simply to grab as much clean data as possible from any source. The main principles of compiling a 'clean' test set are the same as for a 'dirty' test set. While the data can come from any source, the grading of the data is essential as every clean file is not equal and a false alarm on *Excel* is worse than a false alarm on a *Microsoft Chinese* DLL.

So the creation of a test set (and its maintenance) is a hugely complex issue and would benefit from being a specific role, separate from the testing role.

TESTING PROTOCOL

Once the test sets are defined and available the next step is to ensure that the test protocol is appropriate, representative and fair. We will not go into detail here but simply state that it is important to adhere to scientific methodology as much as possible, essential to define the methodology to be used and for the test to be appropriate.

SUGGESTIONS FOR THE FUTURE

One suggestion to aid in the definition of the test methodology is to actually identify what is to be tested. By this we mean target the tests at:

- the solutions used by the product
- the threat vector used by the malware test set
- the protection at the point of delivery
- the end-user profile.

If we targeted the solutions used by the products then the testing may look at a company's complete security offering (e.g. AV bundled with anti-spam) or may look at only one small part of the solution (e.g. the on-demand scanner). We would expect testing to look at the 'whole solution' since this is what the customer wants to know about, but if a component can be isolated then there could be a reason to focus on this with a specific test. However, this must be made absolutely clear in the methodology and any results reported from the test.

Testing the class of malware would require ensuring that the test set consists of only one type of threat vector, such as rootkits or trojans, and would then define the tests around these rather than the solutions available. Obviously the settings would need to be identified in the methodology, but it could be a case of simply taking the default settings of a product that claims to handle rootkits and then testing against this collection.

Basing the tests around the 'delivery' of the sample allows for more real-life tests and an example of this would be testing samples that propagate through email by sending them to the product in email form rather than by scanning the saved contents of the email. This is not a theoretical requirement – there are products which are context-specific and activate appropriate detection capabilities only in the correct circumstances (e.g. detections are protocol-specific: SMTP/POP3, HTTP, IM, etc.).

Finally, the testing could be defined through user profiles. It should be relatively easy to represent the user's behaviour in a particular simulation script – for example, an Internet surfer script can just do browsing. Another script can represent a P2P user and so on. A particular honeypot/goat computer can have a complete set of security software and it would be relatively easy to determine how long a particular suite can hold the fort against an intrusion. Changing a profile script on our honeypot computer would expose the strengths and weaknesses of different security products in a situation very close to real-world use.

An altogether different methodology for testing would be to measure the protection in a live system rather than in a lab environment. This could take the form of special security testing software (a security version of SETI@Home). Such software could install alongside, say, an AV product and record the actions of that AV product. The logs could be submitted to a central server and the effectiveness of the product could be determined (sometimes instantly but frequently later, only when the nature of a specific attack is fully known) and appropriate comparisons made. This protection-evaluation software could be made available, perhaps leveraging AMTSO and even via open source.

Given the mantra that the customer is always right and following on from the above idea, an automatic system to collect users' feedback could be created and distributed. The quality of AV products can be determined from the level of user satisfaction and perception. If this is gathered and coupled with the hard data of intrusions (via the open-source AMTSO plug-in) this could provide not just an opinion, but the users' view of real attacks (or false alarms – this too can be determined retrospectively by analysing the logs). This would combine a subjective level on top of an objective one, and would allow a more complete picture to be seen when comparing products.

Our final suggestion would be to create a formula for the analysis of test results: this would a) give the consumer an indication of how relevant the test was to the product being tested and b) attempt to normalize the results to allow competitive or comparative review of matching security solutions which use differing technologies. This could also be extended to become a common ranking system. Taking the simple case of indicating how relevant the test is to each product, this could be done by defining the number of components used by the product under test and then taking the test result, and dividing by the components.

As an example:

Product A - *consists of an on-demand scanner only*
- *scores 97%*

Product B - *consists of an on-demand scanner, an on-access scanner and an email scanner*

- *scores 96%*

Currently these products would be ranked with product A first and product B second. However, given that product B (assuming default settings) could potentially prevent the malware from ever arriving on the machine to be caught by the on-demand scan, this ranking is misleading. To be truly accurate we would then have to test the on-access and email scanner of product B and also isolate the missed samples from all three tests to confirm whether these were truly missed by product B. In most cases this would be too resource intensive and would not be done. However, by linking a relevance score to the test we could indicate to the consumer that, while product A is better in this specific area, it should not be assumed that it is better overall than product B due to the more complex nature of product B.

Test relevance = Tests exercising discreet components / number of components

So product A scores 1 and product B scores 0.33, showing that only a third of the capability of product B had been exercised in this test.

This is a very simple example and with more time, and perhaps more importantly with some input from our colleagues at the other AV companies, we could come up with further formulas that allow for the complexity of anti-malware solutions. It would be useful to have a standard for ranking products independent of their chosen solution which would also incorporate the false alarm rate.

CONCLUSION

The obvious conclusion is that 10 years ago the testing of security products was a simpler task than the development of them. Today, however, the situation has reversed and it is now far more complex to test than to develop a product. When comparative testing of security products is conducted then the complexity increases again. Due to this complexity the number of arguments around the methodologies and the results will only increase and it is to be hoped that the timely creation of AMTSO will aid in the 'mediation' of these differences in opinions.

It can also be concluded that there is a need for an independent body to assist in the collection of samples as well as providing tools for the tester to benefit from some of the suggestions in this document. Our belief is that it would make sense for AMTSO to take on this role with open-source plug-ins such as sample collection (record and playback), user feedback, protection evaluation and intrusion logging, and by becoming an independent sample authority.

We are very optimistic about the future despite all the complexities. The creation of the Anti-Spyware Coalition (ASC) did help in controlling PUPs – the amount of adware started decreasing after a few court cases in the US used the industry-agreed definitions developed by the ASC. This makes us believe that AMTSO also has a good chance of succeeding because there is a need for a change, everybody agrees on the main principles (both technical and ethical) and we are seeing strong cohesive cooperation in this area.

ACKNOWLEDGEMENTS

We are grateful to our colleagues at *Alwil*, *Ikarus*, *Kaspersky Lab*, *Microsoft*, *Rising* and *Sophos* for information about their sample exchange.

The statistics about the malware growth and the number of samples are courtesy of our colleagues Dmitry Gryaznov and François Paget.

We appreciate the input of all AMTSO members in general and are particularly thankful to Michael Parsons, Mark Kennedy and Matt Williamson. These discussions helped to shape some ideas and served as an impetus to write this whole paper.

REFERENCES

[1] http://www.mcafee.com/us/local_content/white_papers/threat_center/wp_imuttik_vb_conf_2001.pdf.

[2] http://www.amtso.org/.

[3] http://www.siteadvisor.com/.

[4] http://en.wikipedia.org/wiki/Percolation_theory.

[5] Harley, D.; Lee, A. Who Will Test The Testers? Proceedings of the 18th Virus Bulletin International Conference, 2008.

[6] http://en.wikipedia.org/wiki/Zhengtu.